Digital Audio Signal Processing

Udo Zölzer
Technical University of Hamburg-Harburg, Germany

JOHN WILEY & SONS, LTD
Chichester • New York • Weinheim • Brisbane • Singapore • Toronto

First published under the title *Digitale Audiosignalverarbeitung*

Baffins Lane, Chichester,
West Sussex PO19 1UD, England
National 01243 779777
International (+44) 1243 779777

Reprinted July 1998, December 1999

e-mail (for orders and customer service enquiries): cs-books@wiley.co.uk

Visit our Home Page on http://www.wiley.co.uk
or
http://www.wiley.com

Other Wiley Editorial Offices

John Wiley & Sons, Inc., 605 Third Avenue,
New York, NY 10158-0012, USA

VCH Verlagsgesellschaft mbH, Pappelallee 3,
D-69469 Weinheim, Germany

Jacaranda Wiley Ltd, 33 Park Road, Milton,
Queensland 4064, Australia

John Wiley & Sons (Asia) Pte Ltd, 2 Clementi Loop #02-01,
Jin Xing Distripark, Singapore 129809

John Wiley & Sons (Canada) Ltd, 22 Worcester Road,
Rexdale, Ontario M9W 1L1, Canada

British Library Cataloguing in Publication Data

A catalogue record for this book is available from the British Library

ISBN 0 471 97226 6

Produced from PostScript files supplied by the author.
Printed and bound in Great Britain by Bookcraft (Bath) Ltd, Midsomer Norton, Somerset
This book is printed on acid-free paper responsibly manufactured from sustainable forestry, in which at least two trees are planted for each one used for paper production.

Contents

Preface

Digital audio signal processing is employed in recording and storing music and speech signals, for sound mixing and production of digital programs, in digital transmission to broadcast receivers as well as in consumer products like CDs, DATs and PCs. In the latter case, the audio signal is in a digital form all the way from the microphone right up to the loudspeakers, enabling real-time processing with fast digital signal processors.

This book provides the basis of an advanced course in *Digital Audio Signal Processing* which I have been giving since 1992 at the Technical University Hamburg-Harburg. It is directed at students studying engineering, computer science and physics but, also for professionals who look for solutions to problems in audio signal processing like in the fields of studio engineering, consumer electronics and multimedia. The mathematical and theoretical fundamentals of digital audio signal processing systems will be presented and typical applications with an emphasis on realization aspects will be discussed. Prior knowledge of systems theory, digital signal processing and multirate signal processing are taken as a prerequisite.

The book is divided into two parts. The first part (chapters 1-4) presents a basis for hardware systems used in digital audio signal processing. The second part (chapters 5-9) discusses algorithms for processing digital audio signals. Chapter 1 describes the course taken by an audio signal from its recording in a studio up to its reproduction at home. Chapter 2 contains a representation of signal quantization, dither techniques and spectral shaping of quantization errors used for reducing the nonlinear effects of quantization. In the end, a comparison is made between the fixed-point and floating-point number representations as well as their associated effects on format conversion and algorithms. Chapter 3 describes methods for AD/DA conversion of signals, starting with Nyquist sampling, methods for oversampling techniques and delta-sigma modulation. The chapter closes with a presentation of some circuit design of AD/DA converters. After an introduction to digital signal processors and digital audio interfaces, chapter 4 describes simple hardware systems based on a single- and multiprocessor solutions. The algorithms introduced in the following chapters 5-9 are, to a great extent, implemented in real-time on hardware platforms presented in chapter 4. Chapter 5 describes digital audio equalizers. Apart from the implementation aspects of recursive audio filters, nonrecursive linear phase filters based on fast convolution and filter banks are introduced. Filter designs, parametric filter structures and precautions for reducing quantization errors in recursive filters are dealt with in detail. Chapter 6 deals with room simulation. Methods for simulation of artificial room impulse response and methods for approximation of measured impulse responses are discussed. In chapter 7 the dynamic range control of audio signals is described. These

methods are applied at several positions in the audio chain from the microphone up to the loudspeakers in order to adapt to the dynamics of the recording, transmission and listening environment. Chapter 8 contains a presentation of methods for synchronous and asynchronous sampling rate conversion. Efficient algorithms are described which are suitable for real-time processing as well as off-line processing. Both lossless and lossy data compression of digital audio signals are discussed in chapter 9. Lossless data compression is applied for storing of higher word-lengths. Lossy data compression, on the other hand, plays a significant role in communication systems.

I would like to thank Prof. Fliege (University of Mannheim), Prof. Kammeyer (University of Bremen) and Prof. Heute (University of Kiel) for comments and support. I am also grateful to my colleagues at the TUHH and especially Dr. Alfred Mertins, Dr. Thomas Boltze, Dr. Bernd Redmer, Dr. Martin Schönle, Dr. Manfred Schusdziarra, Dr. Tanja Karp, Georg Dickmann, Werner Eckel, Thomas Scholz, Rüdiger Wolf, Jens Wohlers, Horst Zölzer, Bärbel Erdmann, Ursula Seifert and Dieter Gödecke. Apart from these, I would also like to say a word of gratitude to all those students who helped me in carrying out this work successfully.

Special thanks go to Saeed Khawaja for his help during translation and to Dr. Anthony Macgrath for proof-reading the text. I also would like to thank Jenny Smith, Colin McKerracher, Ian Stoneham and Christian Rauscher (Wiley).

My special thanks are directed to my wife Elke and my daughter Franziska.

Hamburg, July 1997 Udo Zölzer

Chapter 1

Introduction

In this introductory chapter, the fields of application for digital audio signal processing are presented. Starting from recording in a studio or in a concert hall, the whole chain of signal processing is shown, up to the reproduction at home or in a car. The fields of application can be divided into the following areas:

- studio technology
- digital transmission systems
- storage media
- audio components for home entertainment

The basic principles of the above-mentioned fields of application will be presented as an overview in order to exhibit the uses of digital signal processing.

1.1 Studio Technology

While recording speech or music in a studio or in a concert hall, the analog signal from a microphone is first digitized, fed to a digital mixing console and then stored on a digital storage medium. A digital sound studio is shown in Fig. 1.1. Besides the analog sources (microphones), digital sources are fed to the digital mixing console over multichannel MADI interfaces [AES91]. Digital storage media like the digital multitrack tape machines and digital hard disc recording systems are also connected via multichannel MADI interfaces to the mixing console. The final stereo

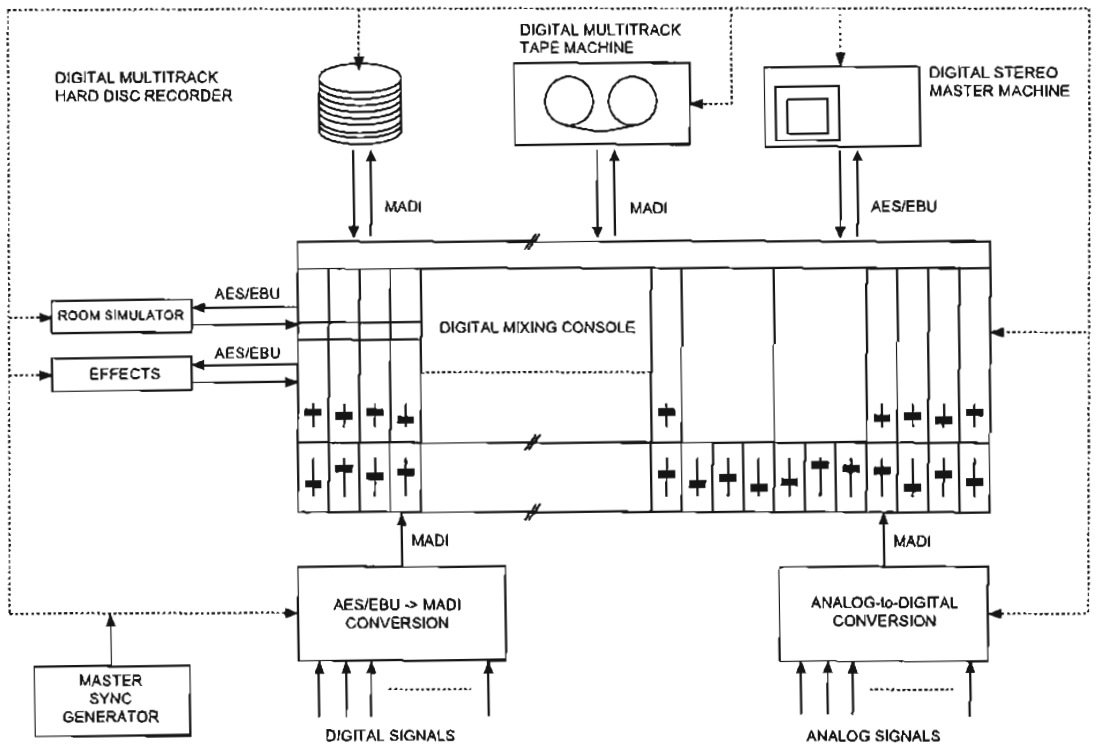


Figure 1.1 Digital sound studio.

mix is stored via a two-channel AES/EBU interface [AES92] on a two-channel MASTER machine. External appliances for effects or room simulators are also connected to the mixing console via a two-channel AES/EBU interface. All systems are synchronized by a MASTER clock reference. In digital audio technology, the sampling rates[1] $f_S = 48$ kHz for professional studio technology, $f_S = 44.1$ kHz for compact disc and $f_S = 32$ kHz for broadcasting applications are established. The sound mixing console plays a central role in a digital sound studio. Fig. 1.2 shows the functional units. The N input signals are processed individually. After level and panorama control, all signals are summed up to give a stereo mix. The summation is carried out several times so that other auxiliary stereo and/or mono signals are available for other purposes. In a sound channel (see Fig. 1.3), an equalizer unit (EQ), a dynamic unit (DYN), a delay unit (DEL), a gain element (GAIN) and a panorama element (PAN) are used. In addition to input and output signals in an audio channel, inserts as well as auxiliary or direct outputs are required.

[1] data rate: 16 bit x 48 kHz = 768 kbit/s
data rate (AES/EBU signal): 2 x (24+8) bit x 48 kHz = 3.072 Mbit/s
data rate (MADI signal): 56 x (24+8) bit x 48 kHz = 86.016 Mbit/s

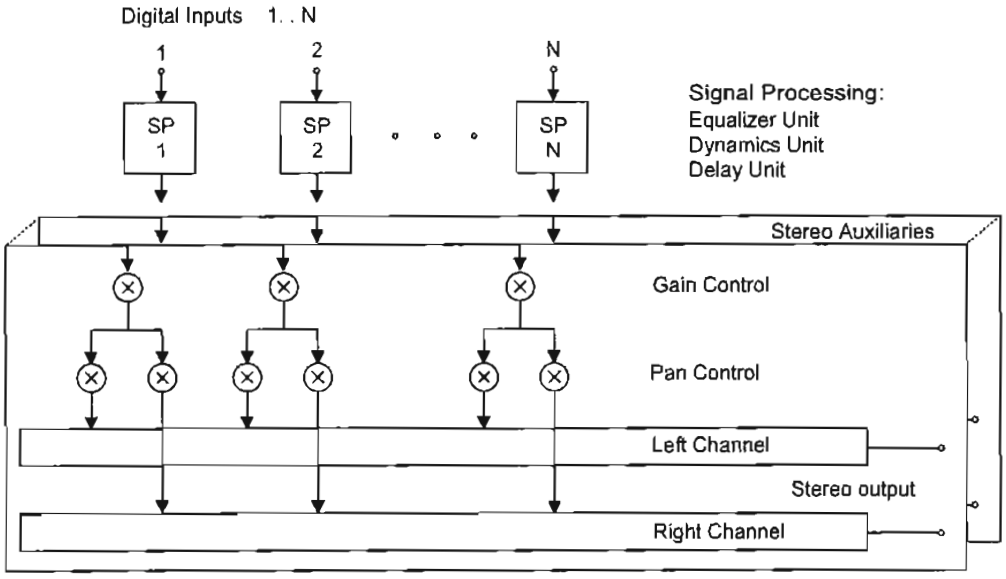


Figure 1.2 N-channel sound mixing console.

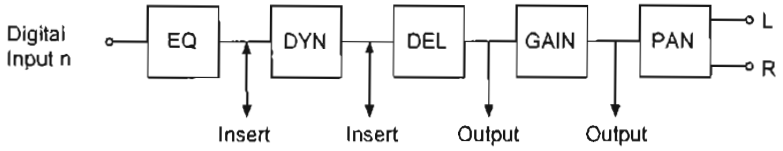


Figure 1.3 Sound channel.

1.2 Digital Transmission Systems

For radio broadcasting, there are no digital transmission techniques to complement the analog techniques (LF, MF, HF and VHF) covering wide broadcasting areas. Initial solutions are introduced with Digital Satellite Broadcasting (Digital Satellite Radio). A long-term replacement of stationary and mobile receiving of FM signals is planned from 1995 onwards with a digital technique called DAB (Digital Audio Broadcasting, Terrestrial + Satellite). A list of broadcasting systems is given in Table 1.1 [Ple91].

Digital Satellite Broadcasting

Digital Satellite Broadcasting operates at a sampling rate of $f_S = 32$ kHz. Digital sound signals of an AES/EBU interface are reduced (see Fig. 1.4, [Ple85]) by a

Table 1.1 Comparison of different broadcasting systems (n = number of programs, B = audio bandwidth, SNR = signal-to-noise ratio, M = mono, S = stereo, S+ = stereo + additional information, m = mobile, s = stationary).

frequency range	receiving area	parameters				
		n	B [kHz]	SNR [dB]	M+S	s;m
LF	nationwide and nearby areas	3-5	< 4.5	20	M	s+m
MF	nationwide and nearby areas	8	< 4.5	20	M	s+m
HF	worldwide	?	< 4.5	0	M	s(m)
VHF	regional and local	5-10	15	50	S	s+m
DSR	nationwide and nearby areas	16	15	70	S	s
DAB-T	regional and local	≥ 4 +12	> 15	70	S+	s+m
DAB-T+S	nat. + nearby reg., local	?	> 15	70	S+	s+m

coder (DCA) to a data rate of 1.024 Mbit/s (DS1 interface[2]). The data reduction is carried out by means of a floating-point representation with a 14 bit mantissa and a scaling factor for a block of 64 samples. The transmission to a ground station takes place through a digital signal connection with a bit rate of 2.048 Mbit/sec (DS2 interface). Here the signals of two DS1 interfaces are combined together. A total of 16 programs are transmitted to the satellite TV-Sat 1. At the receiver side, the satellite signal is fed to a DSR receiver via the coaxial network of the Deutsche Bundespost at 118 MHz. As an alternative, it is also possible to receive DSR with a personal satellite receiver (Fig. 1.5, [Ple85]).

Terrestrial Digital Broadcasting (DAB)

With the introduction of terrestrial digital broadcasting, the quality standards of a compact disc will be achieved for mobile and stationary reception of radio signals [Ple91]. Therefore, the data rate of a two-channel AES/EBU signal from a transmitting studio is reduced with the help of a source coder [Bra94] (see Fig. 1.6). Following the source coder (SC), additional information (AI) like the type of program (music/speech) and traffic information is added. A multicarrier technique is applied for digital transmission to stationary and mobile receivers. At the transmitter, several broadcasting programs are combined in a multiplexer

[2]data rate: 2x(14 data bits + 1 parity bit)x 32 kHz + 2x4x 8 kHz = 1.024 Mbit/s

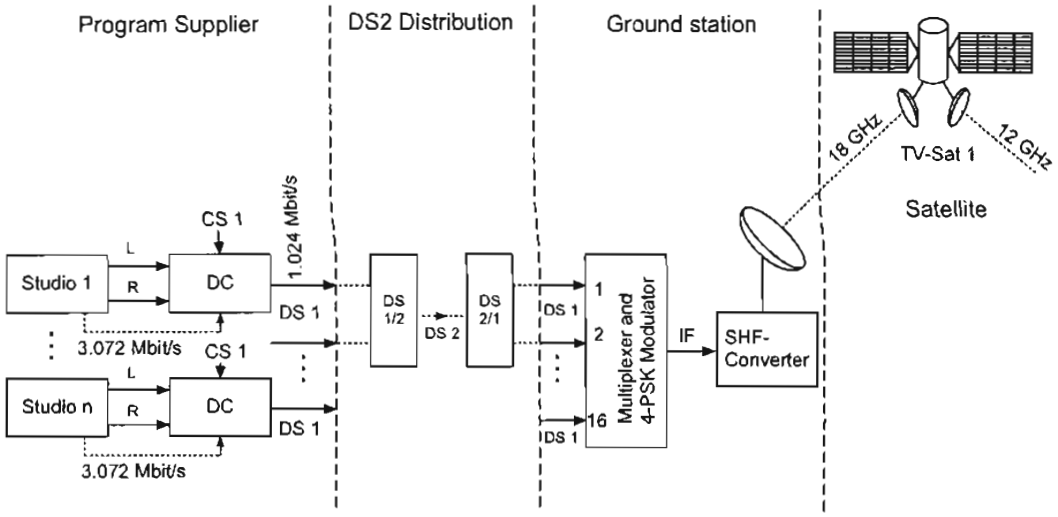


Figure 1.4 Studio - ground station (TS1 = clock signal 1024 kHz).

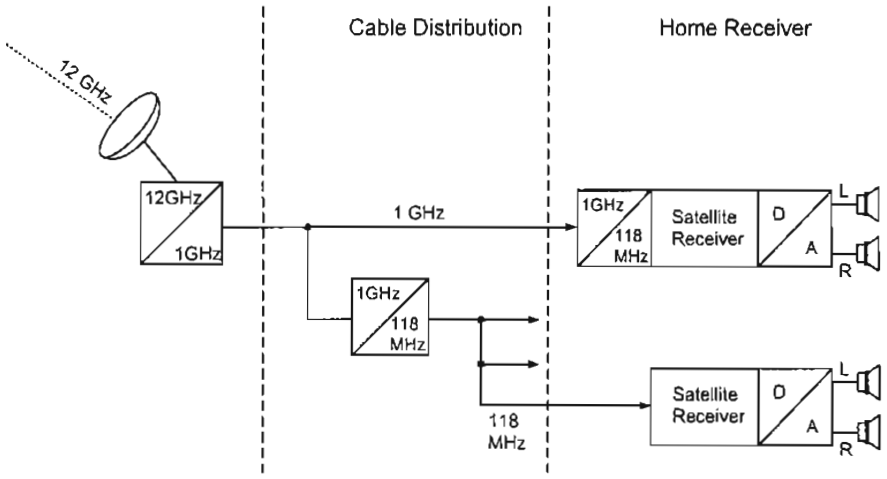


Figure 1.5 Digital Satellite Broadcasting - receiving system.

(MUX) to form a multiplex signal. The channel coding and modulation is carried out with a multi-carrier transmission technique (Coded Orthogonal Frequency Division Multiplex, [Ala87],[Kam92a],[Kam92b],[Kam93],[Tui93]).

The DAB receiver (Fig. 1.7) consists of the demodulator (DMOD), the demultiplexer (DMUX) and the source decoder (SD). The SD provides a linearly quantized PCM signal (Pulse Code Modulation [Jay84]). The PCM signal is fed over a Digital-to-Analog Converter (DA converter) to an amplifier connected to loudspeakers.

For a more detailed description of the DAB transmission technique, an illustration based on filter banks is presented (see Fig. 1.8). The audio signal at a

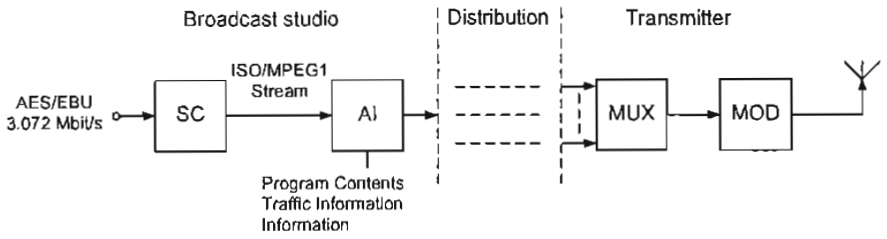


Figure 1.6 DAB transmitter.

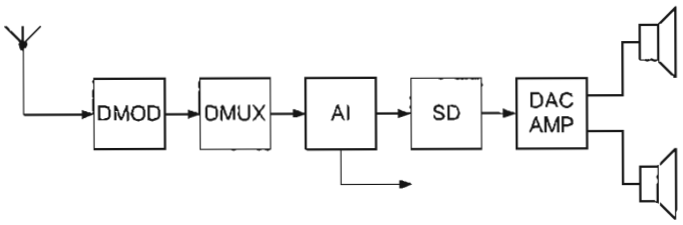


Figure 1.7 DAB receiver.

data rate of 768 kbit/s is decomposed into subbands with the help of an analysis filter bank (AFB). Quantization and coding based on psychoacoustic models is carried out within each subband. The data reduction leads to a data rate of 96-192 kbit/s. The quantized subband signals are provided with additional information (header) and combined together in a frame. This so-called ISO-MPEG1 frame [ISO92] is first subjected to channel coding (CC). Time-interleaving (T-IL) follows and will be described later on. The individual transmitting programs are combined in frequency multiplex (frequency-interleaving F-IL) with a synthesis filter bank (SFB) to one broadband transmitting signal. The synthesis filter bank has several complex-valued input signals and one complex-valued output signal. The real-valued band-pass signal is obtained by modulating with $e^{j\omega_c t}$ and taking the real part. At the receiver, the complex-valued baseband signal is obtained by demodulation followed by low-pass filtering. The complex-valued analysis filter bank provides the complex-valued band-pass signals from which the ISO-MPEG1 frame is formed after frequency and time deinterleaving and channel decoding. The PCM signal is combined using the synthesis filter bank after extracting the subband signals from the frame.

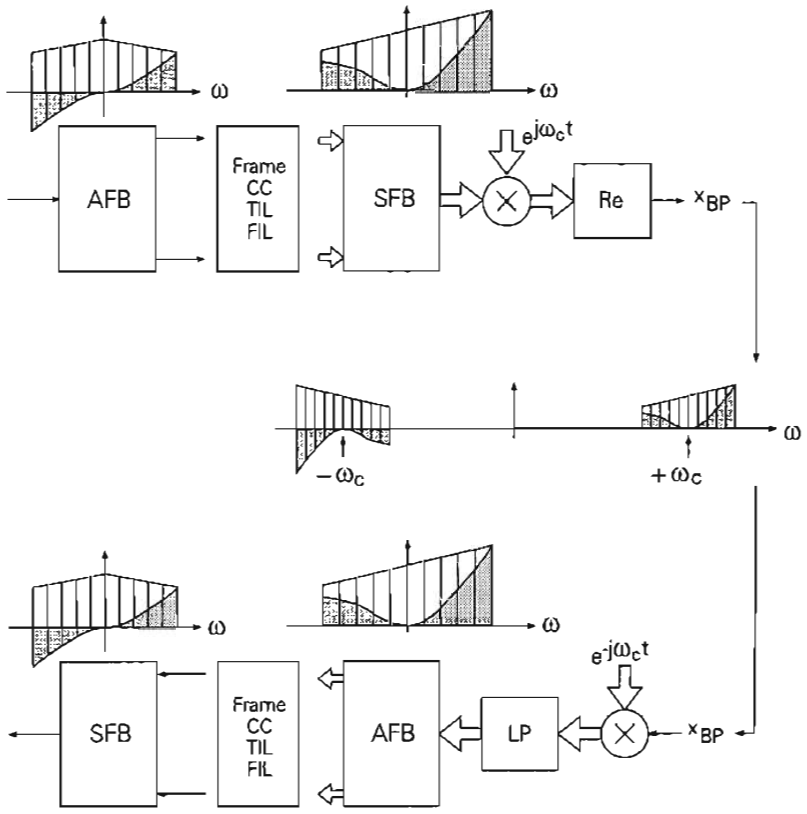


Figure 1.8 Filter banks within DAB.

DAB Transmission Technique. The special problems of mobile communications [Kam92a, Pro89] are dealt with using a combination of the OFDM transmission technique with DPSK modulation and time and frequency interleaving. Possible disturbances are minimized by consecutive channel coding. The schematic diagram in Fig. 1.9 shows the relevant subsystems.

For example, the transmission of a program P_1 which is delivered as an ISO-MPEG1 stream is shown in Fig. 1.9. The channel coding doubles the data rate. The typical characteristics of a mobile communication channel like time and frequency selectivity are handled by using time and frequency interleaving with the help of a multicarrier technique. The burst disturbances of consecutive bits are reduced to single bit errors by spreading the bits over a longer period of time. The narrow-band disturbances affect only individual carriers by spreading the transmitter program P_1 in the frequency domain, i.e. distribution of transmitter programs of carrier frequencies at a certain displacement. The remaining disturbances of the mobile channel are suppressed with the help of channel coding, i.e. by adding redundancy, and decoding with a Viterbi decoder. The implementation of an OFDM transmission is discussed in the following.

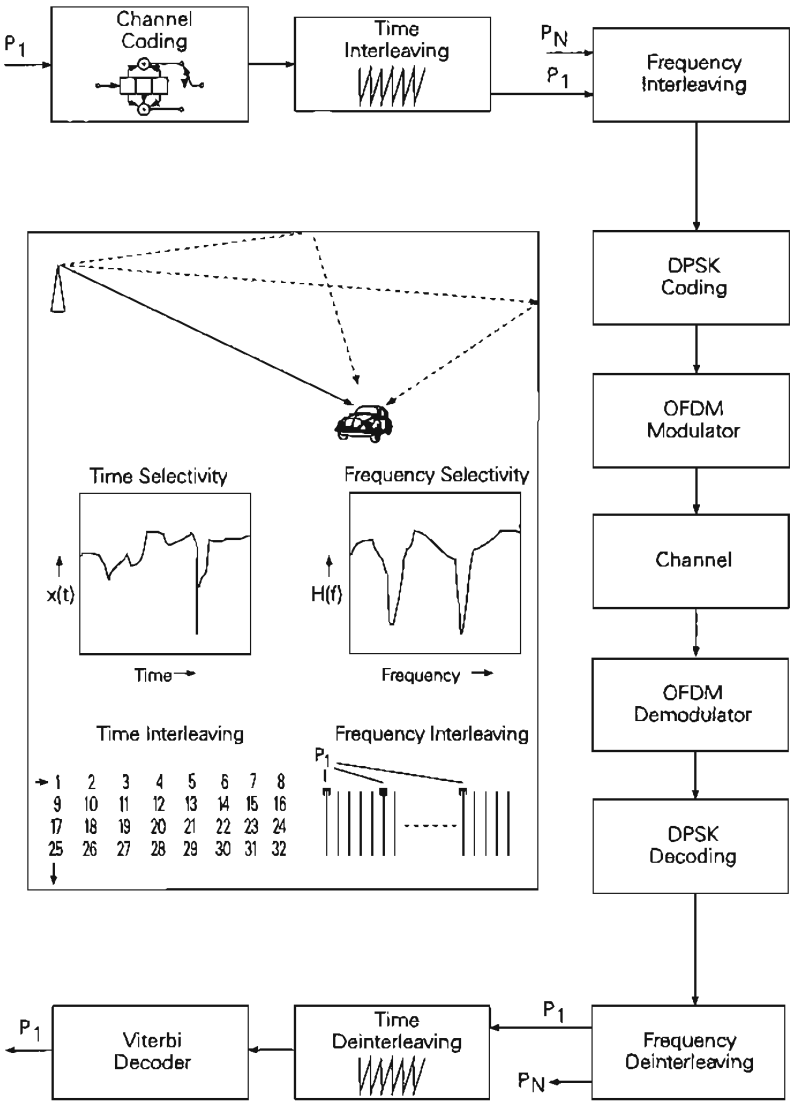


Figure 1.9 DAB transmission technique.

OFDM Transmission. The OFDM transmission technique is shown in Fig. 1.10. The technique stands out owing to its simple implementation in the digital domain. The data sequence $c_t(k)$ which is to be transmitted, is written blockwise into a register of length $2M$. The complex numbers from $d_1(m)$ to $d_M(m)$ are formed from two consecutive bits (dibits). Here the first bit corresponds to the real part and the second to the imaginary part. The signal space shows the four states for the so-called QPSK [Kam92a, Pro89]. The vector $\mathbf{d}(m)$ is transformed with an inverse FFT (Fast Fourier Transform, [Gab87, Fli91]) into a vector $\mathbf{e}(m)$ which describes the values of the transmitted symbol in the time domain. The transmitted symbol $x_t(n)$ with period T_{sym} is formed by the transmission of the M complex

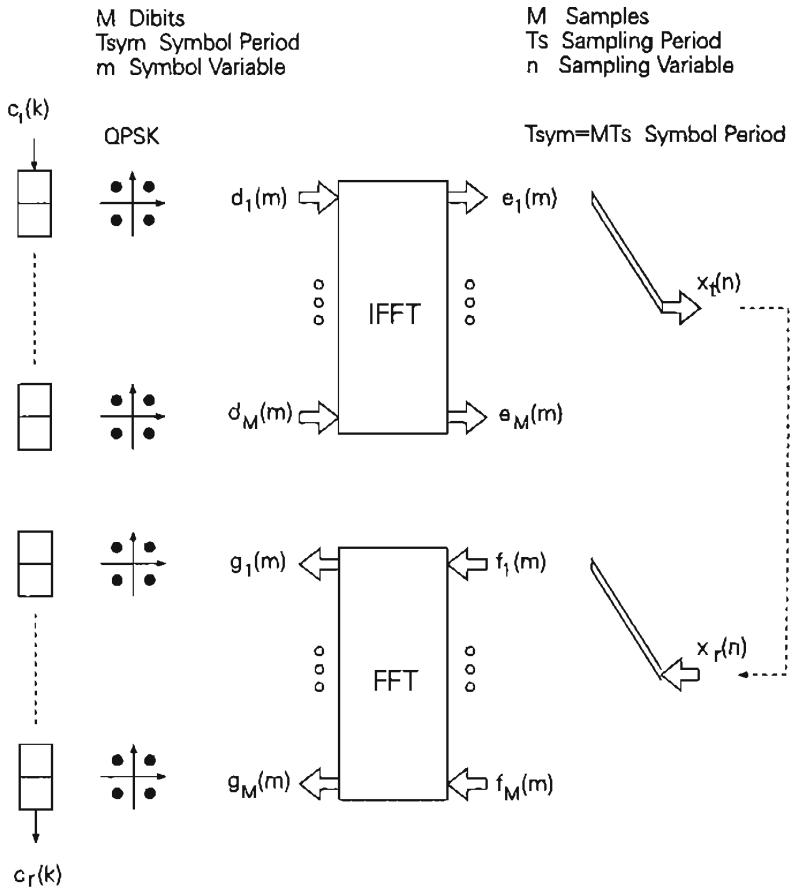


Figure 1.10 OFDM transmission.

numbers $e_i(m)$ at sampling period T_S. The real-valued band-pass signal is formed at high frequency after DA conversion of the quadrature signals, modulation by $e^{j\omega_c t}$ and by taking the real part. At the receiver, the transmitted symbol becomes a complex-valued sequence $x_r(n)$ by demodulation with $e^{-j\omega_c t}$ and AD conversion of the quadrature signal. M samples of the received sequence $x_r(n)$ are distributed over the M input values $f_i(m)$ and transformed into the frequency domain with the help of FFT. The resulting complex numbers $g_i(m)$ are again converted to dibits and provide the received sequence $c_r(k)$. Without the influence of the communication channel, the transmitted sequence can be reconstructed exactly.

OFDM Transmission with a Guard Interval. In order to describe the OFDM transmission with a guard interval, the schematic diagram in Fig. 1.11 is considered. The transmission of a symbol of length M over a channel with impulse response $h(n)$ of length L leads to a received signal $y(n)$ of length $M + L - 1$. This means that the received symbol is longer than the transmitted signal. The exact reconstruction of the transmitted symbol is disturbed because of the overlapping

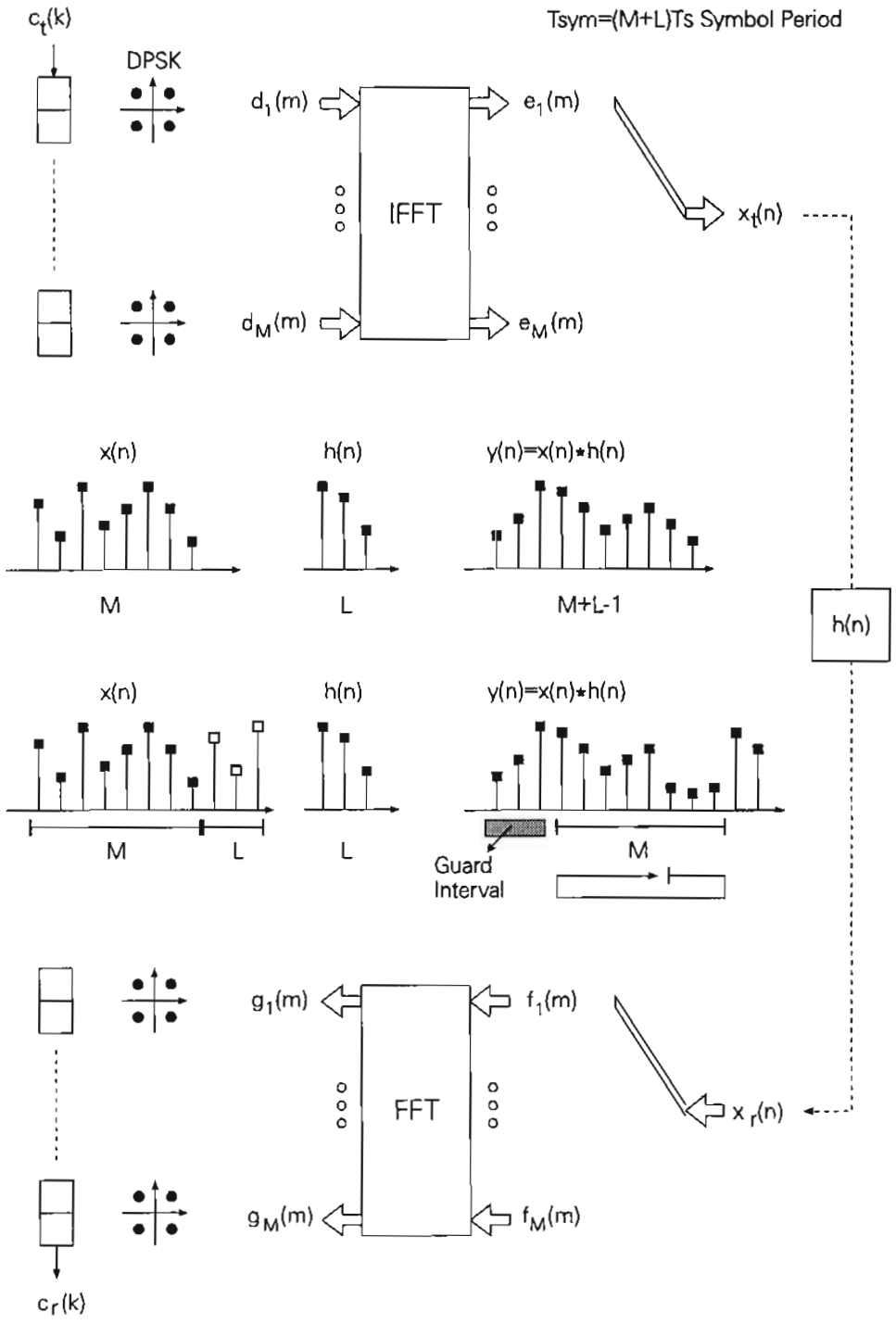


Figure 1.11 OFDM transmission with a guard interval.

of received symbols. Reconstruction of the transmitted symbol is possible by cyclic continuation of the transmitted symbol. Here, the complex numbers from the

vector $\mathbf{e}(m)$ are repeated so as to give a symbol period of $T_{sym} = (M + L)T_S$. Each of the transmitted symbols is, therefore, extended to a length of $M + L$. After transmission over a channel with impulse response of length L, the response of the channel is periodic with length M. After the initial transient state of the channel, i.e. after the L samples of the *guard interval*, the following M samples are written into a register. Since a time delay occurs between the start of the transmitted symbol and the sampling shifted by L displacements, it is necessary to shift the sequence of length M cyclically by L displacements. The inverse FFT provides the received vector $\mathbf{g}(m)$. The complex values $g_i(m)$ do not correspond to the exact transmitted values $d_i(m)$ because of the transmission channel $h(n)$. However, there is no influence of neighboring carrier frequencies. Every received value $g_i(m)$ is weighted with the corresponding magnitude and phase of the channel at the specific carrier frequency. The influence of the communication channel can be eliminated by differential coding of consecutive dibits [Kam92a, Pro89]. The decoding process can be done according to $z_i(m) = g_i(m)g_i^*(m-1)$. The dibit corresponds to the sign of the real and imaginary parts. The DAB transmission technique presented stands out owing to its simple implementation with the help of FFT algorithms. The extension of the transmitted symbol by a length L of the channel impulse response and the synchronization to collect the M samples out of the received symbol have still to be carried out. The length of the guard interval must be matched to the maximum echo delay of the multipath channel. Owing to differential coding of the transmitted sequence, an equalizer at the receiver is not necessary.

1.3 Storage Media

Compact Disc

The technological advances in the semiconductor industry have led to economical storage media for digitally encoded information. Independently of the developments in the computer business, the compact disc system was introduced by Philips and Sony in 1982. The storage of digital audio data is carried out on an optical storage medium. The compact disc operates at a sampling rate of $f_S = 44.1$ kHz[3]. The essential specifications are summed up in Table 1.2.

[3] 3 x 490 x 30 Hz (NTSC) = 3 x 588 x 25 Hz (CCIR) = 44.1 kHz

Table 1.2 Specifications of the CD system [Ben88].

Type of recording	
Signal recognition	optical
Storage density	682 Mbit/in^2
Audio specification	
Number of channels	2
Duration	approx. 60 min
Frequency range	20-20000 Hz
Dynamic range	> 90 dB
THD	< 0.01 %
Signal format	
Sampling rate	44.1 kHz
Quantization	16-Bit PCM (2's-complement)
Preemphasis	none or 50/15 μs
Error Correction	CIRC
Data rate	2.034 Mbit/s
Modulation	EFM
Channel bit rate	4.3218 Mbit/s
Redundancy	30 %
Mechanical specification	
Diameter	120 mm
Thickness	1.2 mm
Diameter of the inner hole	15 mm
Program range	50-116 mm
Reading speed	1.2 - 1.4 m/s
	500 - 200 r/min

DASH (Digital Audio Stationary Head)

The DASH system serves to record multiple audio channels in the field of professional audio production. It is based on longitudinal track recording (Table 1.3) on magnetic tape. Apart from analog interfaces with band-limiting and reconstruction filters and AD/DA converters, purely digital interfaces are also available.

R-DAT (Rotary-Head Digital Audio on Tape)

The R-DAT system makes use of the heliscan method for two-channel recording. The available devices enable the recording of 16 bit PCM signals with all three sampling rates (Table 1.4) on a tape. R-DAT recorders are used in studio recording as well as in consumer applications.

Table 1.3 Specifications of the DASH system [Ben88].

Type of recording	
Signal recognition	magnetic
storage capacity	> 16 GB
Audio specification	
Number of channels	24, 48
Signal format	
Sampling rate	48, 44.1, 32 kHz
Quantization	16 bit, 20 bit PCM (2s complement)
Error correction	CRC
Mechanical specification	
Tapewidth of magnet	1/2 in
Lin. trackspeed	19.05, 38.1, 76.2 cm/s (48 kHz)

Table 1.4 Specifications of the R-DAT system [Ben88].

Type of recording	
Signal recognition	magnetic
storage capacity	2 GB
Audio specification	
Number of channels	2
Duration	max. 120 min
Frequency range	20-20000 Hz
Dynamic range	> 90 dB
THD	< 0.01 %
Signal format	
Sampling rate	48, 44.1, 32 kHz
Quantization	16 bit PCM (2s complement)
Error correction	CIRC
Channel coding	8/10 modulation
Data rate	2.46 Mbit/s
Channel bit rate	9.4 Mbit/s
Mechanical specification	
Tapewidth of magnet	3.8 mm
Thickness	13 μm
Diameter of head drum	3 cm
Revolutions per min	2000 r/min
Rel. track speed	3.133 m/s
	500 - 200 r/min

DCC (PASC) and Mini Disc (ATRAC)

The techniques of storage for the DCC system (Digital Compact Cassette) and the Mini Disc system are based on source coding techniques that make use of psychoacoustic effects for reducing data rates. The DCC system uses the PASC technique (Precision Adaptive Subband Coding [Wir91]) and operates at 2·192 kbit/s for a stereo channel. A compact cassette (magnetic tape) serves as a storage medium for analog and digital signals. The Mini Disc system operates with the ATRAC technique (Adaptive Transform Acoustic Coding, [Tsu92]) and has a data rate of about 2·140 kbit/s for a stereo channel. A magneto-optical storage medium is used for recording.

Hard Disc Recording Systems

Apart from the specially developed digital recording systems for audio purposes, hard disc storage media enable new recording concepts. Recording systems based on magnetic and magneto-optical concepts provide new operating philosophies with different recording strategies. This is because spooling times of tape machines do not occur and fast access to audio signals is possible. Moreover, the provisions of editing in the digital domain are very useful. Hence, as well as the acoustic control, the visual presentation of audio signals on a screen simplifies and improves the processing.

1.4 Audio Components at Home

The domestic digital storage media already in use, like compact discs, DAT recorders and DCC/Mini Disc, which have digital outputs, can be connected to digital post-processing systems right up to the loudspeakers. The individual tone control consists of the following processing.

Equalizer

Spectral modification of the music signal in amplitude and phase and the automatic correction of the frequency response from loudspeaker to listening environment is desired.

Room Simulation

The simulation of room impulse responses and the processing of music signals with special room impulse response are used to give an impression of a room like a concert hall, a cathedral or a jazz club.

Surround Systems

Besides the reproduction of stereo signals from a CD over two frontal loudspeakers, more than two channels will be recorded in the prospective digital recording systems [Lin93]. This is already illustrated in the sound production for cinema movies where besides the stereo signal (L, R), a middle channel (M) and two additional room signals (L_B, R_B) are recorded. These *surround* systems are also used in the prospective digital television systems. The *ambisonics* technique [Ger85] is a recording technique that allows three-dimensional recording and reproduction of sound.

Digital Amplifier Concepts

The basis of a digital amplifier is pulse width modulation as shown in Fig. 1.12. With the help of a fast counter, a pulse width modulated signal is formed out of the w bit linearly quantized signal. Single-sided and double-sided modulated conversion are used and they are represented by two and three states respectively. Single-sided modulation (2 states, -1 and +1) is performed by a counter which counts upward from zero with multiples of the sampling rate. The number range of the PCM signal from -1 to +1 is directly mapped onto the counter. The duration of the pulse width is controlled by a comparator. For pulse width modulation with three states (-1, 0, +1), the sign of the PCM signal determines the state. The pulse width is determined by a mapping of the number range from 0 to 1 onto a counter. For double-sided modulation, an upward/downward counter is needed which has to be clocked at twice the rate compared with single-sided modulation. The allocation of pulse widths is shown in Fig. 1.12. In order to reduce the clock rate for the counter, pulse width modulation is carried out after oversampling (Oversampling) and noise shaping (Noise Shaping) of the quantization error (see Fig. 1.13, [Gol90]). Thus the clock rate of the counter is reduced to 180.6 MHz. The input signal is first upsampled by a factor of 16 and then quantized to 8 bits with third-order noise shaping. The use of pulse shaping with delta-sigma modulation is shown in Fig. 1.14 [And92]. Here a direct conversion of the delta-sigma modulated 1 bit signal is performed. The pulse converter shapes the envelope of the serial

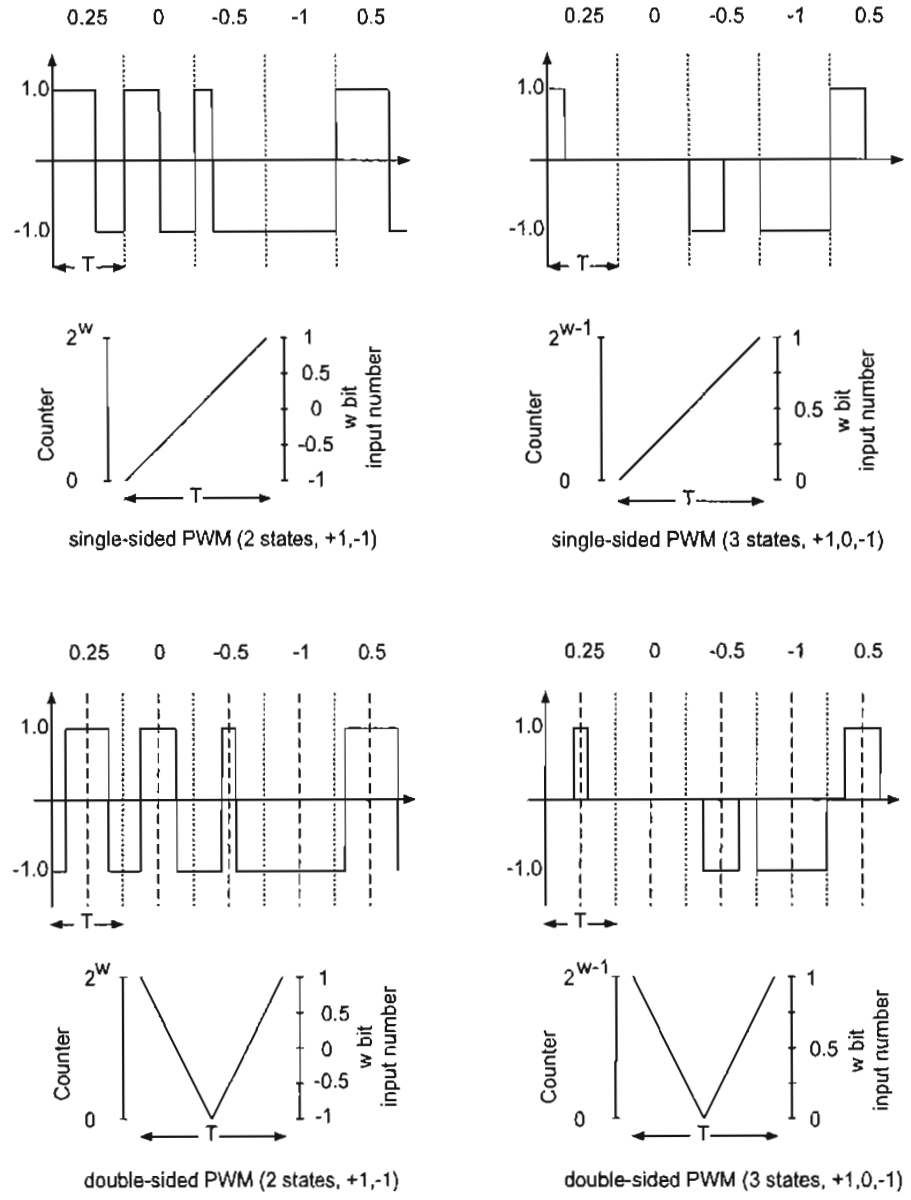


Figure 1.12 Pulse width modulation.

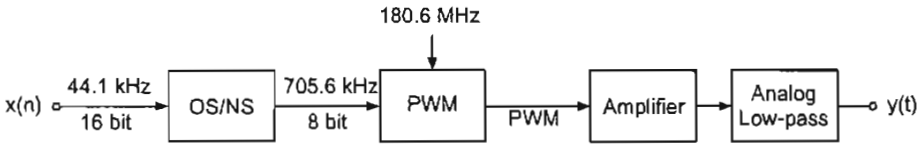


Figure 1.13 Pulse width modulation with oversampling and noise shaping.

data bits. The low-pass filter reconstructs the analog signal. In order to reduce nonlinear distortion, the output signal is fed back (see Fig. 1.15, [Klu92]).

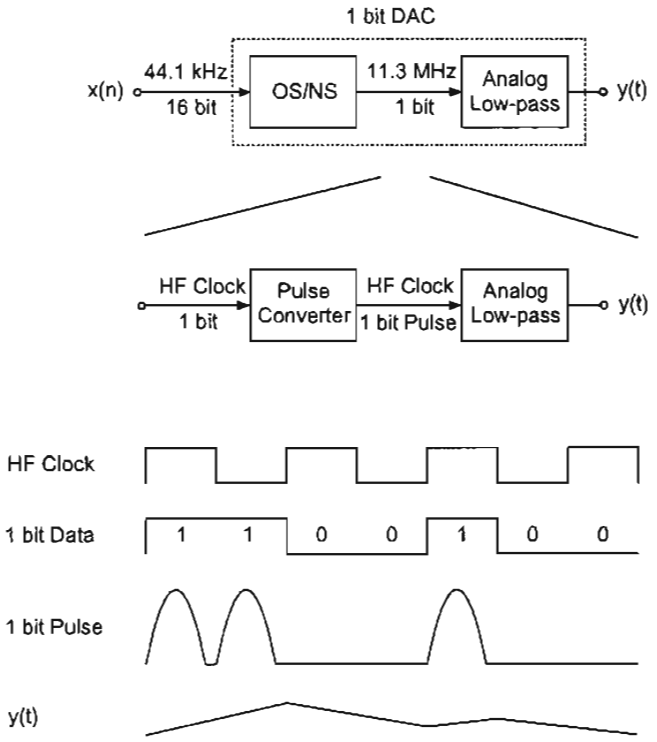


Figure 1.14 Pulse shaping after delta-sigma modulation.

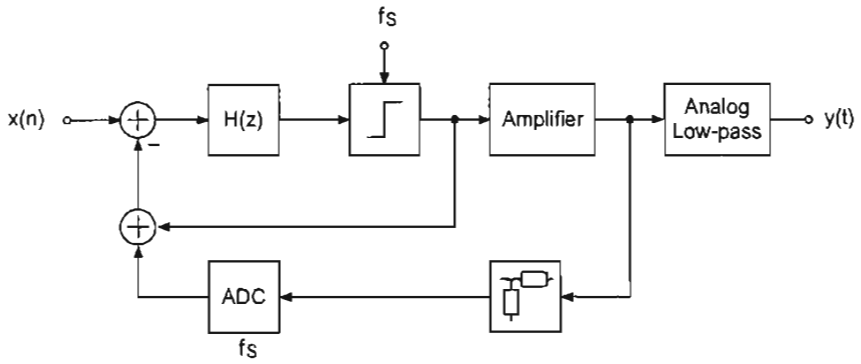


Figure 1.15 Delta-sigma modulated amplifier with feedback.

Digital Crossover

In order to perform digital crossovers for loudspeakers, a linear phase decomposition of the signal with a special filter bank [Zöl92] is done (Fig. 1.16). In a first step, the input signal is decomposed into its high-pass and low-pass components and the high-pass signal is fed to a DAC over a delay unit. In the next step, the low-pass signal is further decomposed. The individual band-pass signals and the low-pass signal are then fed to the respective loudspeaker.

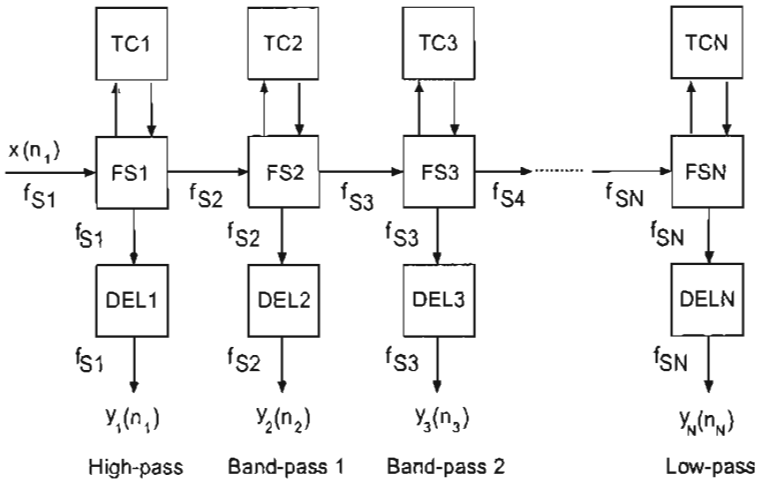


Figure 1.16 Digital crossover (FS_i frequency splitting, TC_i transition bandwidth control, DEL_i delay).

Digital Audio Systems in an Automobile

The special listening conditions in a car demand the matching of reproduction dynamics to velocity-dependent noise, as well as improving the room acoustics inside a car [Scp92].

Chapter 2

Quantization

The digitization of a sampled signal with continuous amplitude is called quantization. The effects of quantization starting with the classical quantization model are discussed in the first section. In the second section dither techniques are presented which, for low-level signals, linearize the process of quantization. In the third section, spectral shaping of quantization errors is described. The last section deals with number representation for digital audio signals and their effects on algorithms.

2.1 Signal Quantization

2.1.1 Classical Quantization Model

Quantization is described by *Widrow's Quantization Theorem* [Wid61]. It says that a quantizer can be modeled (see Fig. 2.1) as the addition of a uniform distributed random signal e and the original unquantized signal x. In order to simplify the notation, the time indices n of x, e and x_Q have been omitted. This linear model of the output x_Q is only then valid when the input amplitude has a wide dynamic range and the quantization error is not correlated with the signal x. Owing to the statistical independence of consecutive quantization errors, the power density spectrum is constant over all frequencies.

The nonlinear process of quantization is described by a nonlinear characteristic curve as shown in Fig. 2.2a where Q denotes the quantization step. The difference between output and input of the quantizer provides the quantization error

$$e = x_Q - x, \tag{2.1}$$

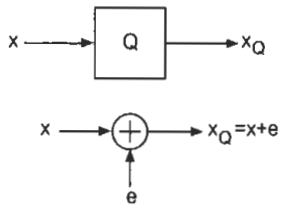


Figure 2.1 Quantization.

which is shown in Fig. 2.2b. The uniform probability density function (PDF) of the quantization error is given (see Fig. 2.2b) by

$$p_E(e) = \frac{1}{Q}\text{rect}(\frac{e}{Q}). \tag{2.2}$$

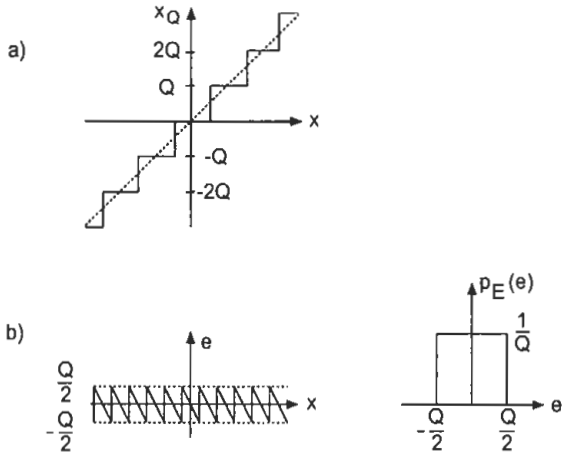


Figure 2.2 a) Nonlinear characteristic curve of a quantizer. b) Quantization error e and its probability density function (PDF) $p_E(e)$.

The mth moment of a random variable $\mathbf{E}$ with a PDF $p_E(e)$ is defined as the expected value of $\mathbf{E}^m$:

$$E[\mathbf{E}^m] = \int_{-\infty}^{\infty} e^m p_E(e) de. \tag{2.3}$$

For a uniform distributed random process, as in Equation (2.2), the first two moments are given by

$$m_E = E[\mathbf{E}] = 0 \quad \text{mean value} \tag{2.4}$$

$$\sigma_E^2 = E[\mathbf{E}^2] = \frac{Q^2}{12} \quad \text{variance.} \tag{2.5}$$

The signal-to-noise ratio (Signal-to-Noise Ratio)

$$\text{SNR} = 10\log_{10}\left(\frac{\sigma_X^2}{\sigma_E^2}\right) \qquad \text{[dB]} \tag{2.6}$$

is defined as the ratio of signal power to error power.

For a quantizer with input range $\pm x_{max}$ and word-length w, the quantization step size can be expressed as

$$Q = 2x_{max}/2^w. \tag{2.7}$$

By defining a peak factor

$$P_F = \frac{x_{max}}{\sigma_X} = \frac{2^{w-1}Q}{\sigma_X} \tag{2.8}$$

the variances of the input and the quantization error can be written as

$$\sigma_X^2 = \frac{x_{max}^2}{P_F^2} \qquad \text{and} \tag{2.9}$$

$$\sigma_E^2 = \frac{Q^2}{12} = \frac{1}{12}\frac{x_{max}^2}{2^{2w}}2^2 = \frac{1}{3}x_{max}^2 2^{-2w} \quad . \tag{2.10}$$

The signal-to-noise ratio is then given by

$$\begin{aligned} \text{SNR} &= 10\log_{10}\left(\frac{x_{max}^2/P_F^2}{\frac{1}{3}x_{max}^2 2^{-2w}}\right) = 10\log_{10}\left(2^{2w}\frac{3}{P_F^2}\right) \\ &= 6.02\,w - 10\log_{10}(P_F^2/3) \qquad \text{[dB]}. \end{aligned} \tag{2.11}$$

A sinusoidal signal (PDF as in Fig. 2.3) with $P_F = \sqrt{2}$, gives

$$\text{SNR} = 6.02\,w + 1.76 \qquad \text{[dB]}. \tag{2.12}$$

For a signal with uniform PDF (see Fig. 2.3) and $P_F = \sqrt{3}$ we can write

$$\text{SNR} = 6.02\,w \qquad \text{[dB]} \tag{2.13}$$

and for a Gaussian distributed signal (probability of overload $< 10^{-5}$ leads to $P_F = 4.61$, see Fig. 2.4), it follows that

$$\text{SNR} = 6.02\,w - 8.5 \qquad \text{[dB]}. \tag{2.14}$$

It is obvious that the signal-to-noise ratio depends on the PDF of the input. For digital audio signals that exhibit nearly Gaussian distribution, the maximum signal-to-noise ratio for given word-length w, is 8.5 dB lower than the *rule of thumb* formula (2.13) for the signal-to-noise ratio.

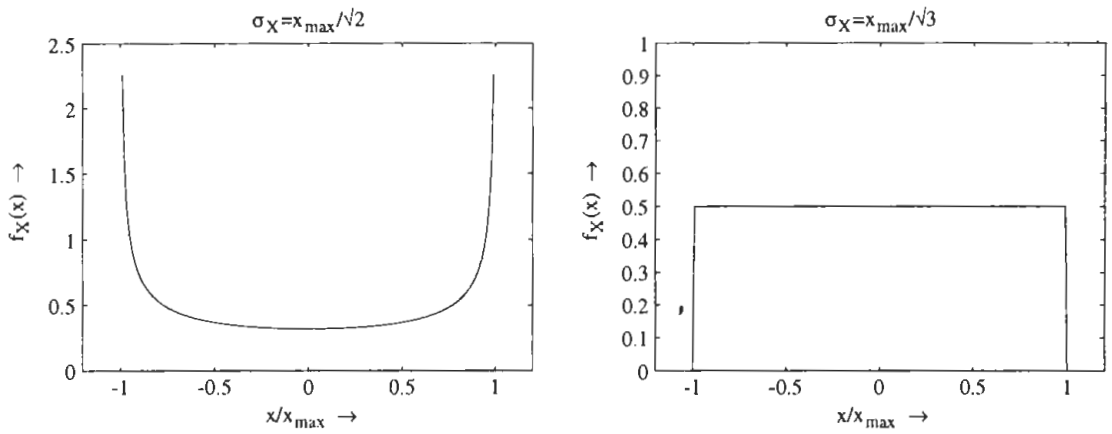


Figure 2.3 Probability density function (sinusoidal signal and signal with uniform PDF).

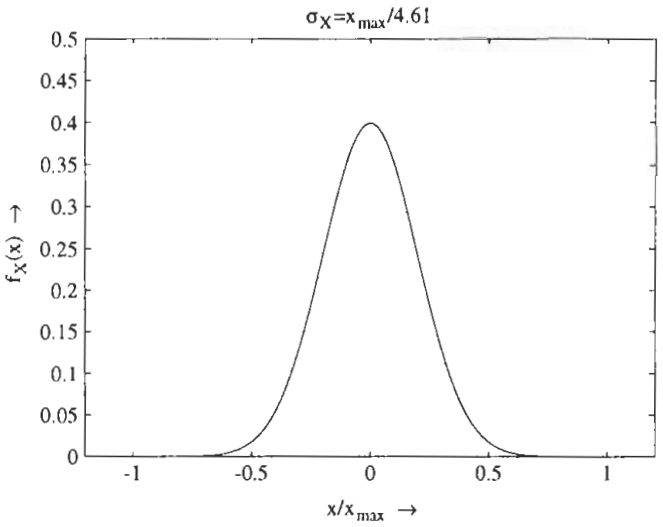


Figure 2.4 Probability density function (signal with Gaussian PDF).

2.1.2 Quantization Theorem

The statement of the *quantization theorem* for amplitude sampling (digitizing the amplitude) of signals has been given by Widrow [Wid61]. The analogy for digitizing the time axis is the *sampling theorem* given by Shannon [Sha48]. First of all, the PDF of the output signal of a quantizer is determined in terms of the PDF of the input signal. The respective characteristic functions (Fourier transform of a PDF) of the input and output signals form the basis for Widrow's *quantization theorem*.

First-order Statistics of the Quantizer Output

Quantization of a continuous-amplitude signal x with PDF $p_X(x)$ leads to a discrete-amplitude signal y with PDF $p_Y(y)$ (see Fig. 2.5). The continuous PDF of the input is sampled by integrating over all quantization intervals (zone sampling). This leads to a discrete PDF of the output.

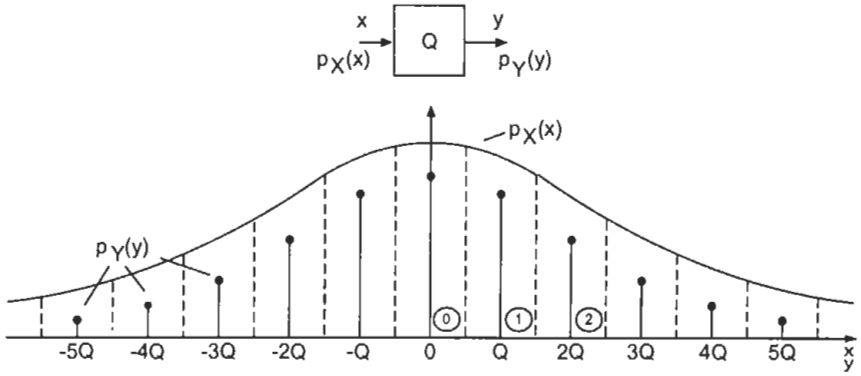


Figure 2.5 Zone sampling of the PDF.

In the quantization intervals, the discrete PDF of the output is determined by the probability

$$W[kQ] = W[-\frac{Q}{2} + kQ \leq x < \frac{Q}{2} + kQ] = \int_{-\frac{Q}{2}+kQ}^{\frac{Q}{2}+kQ} p_X(x)dx. \tag{2.15}$$

For the intervals $k = 0, 1, 2$, it follows that

$$\begin{aligned} p_Y(y) &= \delta(0) \int_{-\frac{Q}{2}}^{\frac{Q}{2}} p_X(x)dx && -\tfrac{Q}{2} \leq y < \tfrac{Q}{2}, \\ &= \delta(y-Q) \int_{-\frac{Q}{2}+Q}^{\frac{Q}{2}+Q} p_X(x)dx && -\tfrac{Q}{2}+Q \leq y < \tfrac{Q}{2}+Q, \\ &= \delta(y-2Q) \int_{-\frac{Q}{2}+2Q}^{\frac{Q}{2}+2Q} p_X(x)dx && -\tfrac{Q}{2}+2Q \leq y < \tfrac{Q}{2}+2Q. \end{aligned}$$

The summation over all intervals gives the PDF of the output

$$p_Y(y) = \sum_{k=-\infty}^{\infty} \delta(y - kQ)W(kQ) \tag{2.16}$$

$$= \sum_{k=-\infty}^{\infty} \delta(y - kQ)W(y) \tag{2.17}$$

where

$$W(kQ) = \int_{-\frac{Q}{2}+kQ}^{\frac{Q}{2}+kQ} p_X(x)dx \quad , \tag{2.18}$$

$$W(y) = \int_{-\infty}^{\infty} \text{rect}(\frac{y-x}{Q}) p_X(x)dx \tag{2.19}$$

$$= \text{rect}(\frac{y}{Q}) \star p_X(y). \tag{2.20}$$

Using

$$\delta_Q(y) = \sum_{k=-\infty}^{\infty} \delta(y - kQ) \tag{2.21}$$

the PDF of the output is given by

$$p_Y(y) = \delta_Q(y)[\text{rect}(\frac{y}{Q}) \star p_X(y)]. \tag{2.22}$$

The PDF of the output can hence be determined by convolution of a rect-function [Lip92] with the PDF of the input. This is followed by an *amplitude sampling* with resolution Q as described in (2.22) (see Fig. 2.6).

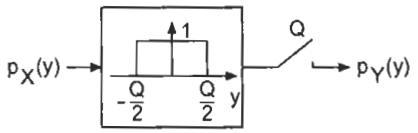


Figure 2.6 Determining PDF of the output.

Using $FT\{f_1(t) \cdot f_2(t)\} = \frac{1}{2\pi} F_1(j\omega) \star F_2(j\omega)$, the characteristic function (Fourier transform of $p_Y(y)$) can be written as

$$P_Y(ju) = \frac{1}{2\pi} u_o \sum_{k=-\infty}^{\infty} \delta(u - ku_o) \star \left[Q \frac{\sin(u\frac{Q}{2})}{u\frac{Q}{2}} \cdot P_X(ju) \right] \tag{2.23}$$

$$\text{with} \quad u_o = \frac{2\pi}{Q}$$

$$= \sum_{k=-\infty}^{\infty} \delta(u - ku_o) \star \left[\frac{\sin(u\frac{Q}{2})}{u\frac{Q}{2}} \cdot P_X(ju) \right] \tag{2.24}$$

$$\boxed{P_Y(ju) = \sum_{k=-\infty}^{\infty} P_X(ju - jku_o) \frac{\sin[(u - ku_o)\frac{Q}{2}]}{(u - ku_o)\frac{Q}{2}}} \tag{2.25}$$

Equation (2.25) describes the sampling of the continuous PDF of the input. If the quantization frequency $u_o = 2\pi/Q$ is twice the highest frequency of the characteristic function $P_X(ju)$ then periodically recurring spectra do not overlap. Hence, a reconstruction of the PDF of the input $p_X(x)$ from the quantized PDF of the output $p_Y(y)$ is possible (see Fig. 2.7). This is known as the *Quantization Theorem* of Widrow. Contrary to the first sampling theorem (Shannon's Sampling Theorem,

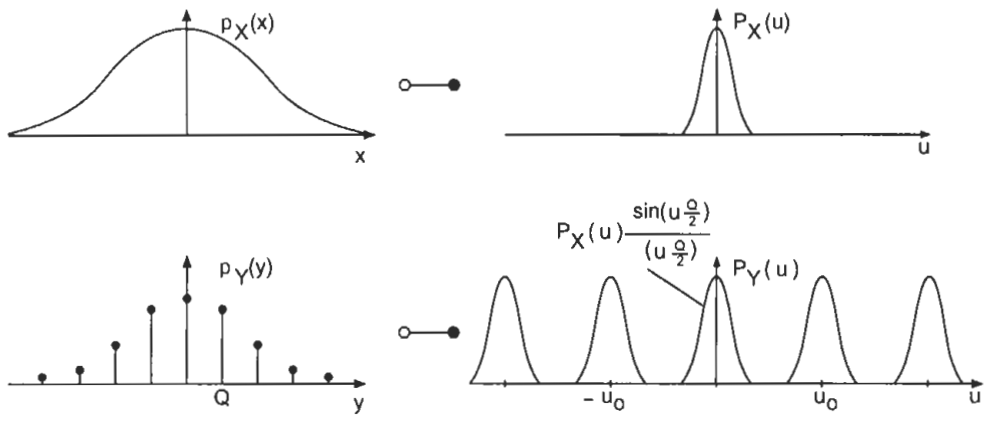


Figure 2.7 Spectral representation.

ideal amplitude sampling in the time domain) $F^A(j\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} F(j\omega - jk\omega_o)$, it can be observed that there is an additional multiplication of the periodically characteristic function with $\frac{\sin[(u-ku_o)\frac{Q}{2}]}{(u-ku_o)\frac{Q}{2}}$ (see Equ. (2.25)).

If the baseband of the characteristic function ($k = 0$)

$$P_Y(ju) = P_X(ju) \underbrace{\frac{\sin(u\frac{Q}{2})}{u\frac{Q}{2}}}_{P_E(ju)} \tag{2.26}$$

is considered, it is observed that it is a product of two characteristic functions. The multiplication of characteristic functions leads to the convolution of PDFs from which the addition of two statistically independent signals can be concluded. The characteristic function of the quantization error is hence

$$P_E(ju) = \frac{\sin(u\frac{Q}{2})}{u\frac{Q}{2}} \tag{2.27}$$

and the PDF

$$p_E(e) = \frac{1}{Q}\mathrm{rect}(\frac{e}{Q}) \tag{2.28}$$

(see Fig. 2.8).

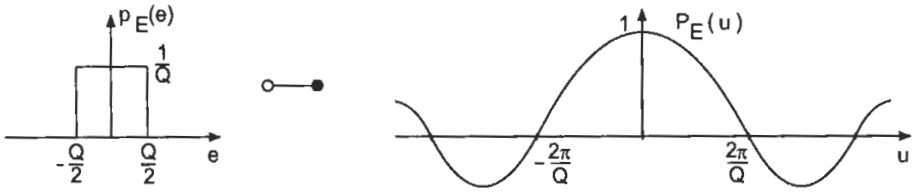


Figure 2.8 PDF and characteristic function of quantization error.

The modeling of the quantization process as an addition of a statistically independent noise signal to an unquantized input signal leads to a continuous PDF of the output (see Fig. 2.9, convolution of PDFs and sampling in the interval Q gives the discrete PDF of the output). The PDF of the discrete-valued output comprises

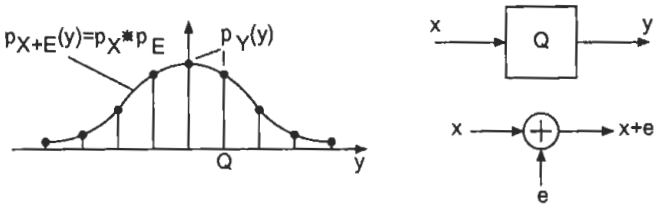


Figure 2.9 PDF of model.

Dirac pulses at distance Q with values equal to the continuous PDF (see Equation (2.22)). Only if the *Quantization Theorem* is valid, the continuous PDF can be reconstructed from the discrete PDF.

In many cases, it is not necessary to reconstruct the PDF of the input. It is sufficient to calculate the moments of the input from the output. The mth moment can be expressed in terms of the PDF or the characteristic function:

$$E[\mathbf{Y}^m] = \int_{-\infty}^{\infty} y^m p_Y(y)dy \tag{2.29}$$

$$= (-j)^m \frac{d^m P_Y(ju)}{du^m}\bigg|_{u=0}. \tag{2.30}$$

If the *Quantization Theorem* is satisfied then the periodic terms in (2.25) do not overlap and the mth derivative of $P_Y(ju)$ is solely determined by the baseband[1] so that with (2.25), it can be written

$$E[\mathbf{Y}^m] = (-j)^m \frac{d^m}{du^m} P_X(ju) \frac{\sin(u\frac{Q}{2})}{u\frac{Q}{2}}\bigg|_{u=0}. \tag{2.31}$$

[1]This is also valid owing to the weaker condition of Sripad and Snyder [Sri77] discussed in the next section.

With Equation (2.31), the first two moments can be determined as

$$m_Y = E[\mathbf{Y}] = E[\mathbf{X}], \tag{2.32}$$

$$\sigma_Y^2 = E[\mathbf{Y}^2] = \underbrace{E[\mathbf{X}^2]}_{\sigma_X^2} + \underbrace{\frac{Q^2}{12}}_{\sigma_E^2}. \tag{2.33}$$

Second-order Statistics of Quantizer Output

In order to describe the properties of the output in the frequency domain, two output values y_1 (at time t_1) and y_2 (at time t_2) are considered [Lip92]. For the joint density function:

$$p_{Y_1Y_2}(y_1, y_2) = \delta_{QQ}(y_1, y_2)[\mathrm{rect}(\frac{y_1}{Q}, \frac{y_2}{Q}) \star p_{X_1X_2}(y_1, y_2)] \tag{2.34}$$

with

$$\delta_{QQ}(y_1, y_2) = \delta_Q(y_1) \cdot \delta_Q(y_2) \tag{2.35}$$

and

$$\mathrm{rect}(\frac{y_1}{Q}, \frac{y_2}{Q}) = \mathrm{rect}(\frac{y_1}{Q}) \cdot \mathrm{rect}(\frac{y_2}{Q}). \tag{2.36}$$

For the two-dimensional Fourier transform, it follows that

$$\begin{aligned} P_{Y_1Y_2}(ju_1, ju_2) &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \delta(u_1 - ku_o)\delta(u_2 - lu_o) \\ &\quad \star \left[\frac{\sin(u_1 \frac{Q}{2})}{u_1 \frac{Q}{2}} \cdot \frac{\sin(u_2 \frac{Q}{2})}{u_2 \frac{Q}{2}} \cdot P_{X_1X_2}(ju_1, ju_2) \right] \qquad (2.37) \\ &= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} P_{X_1X_2}(ju_1 - jku_o, ju_2 - jlu_o) \\ &\quad \frac{\sin[(u_1 - ku_o)\frac{Q}{2}]}{(u_1 - ku_o)\frac{Q}{2}} \cdot \frac{\sin[(u_2 - lu_o)\frac{Q}{2}]}{(u_2 - lu_o)\frac{Q}{2}}. \qquad (2.38) \end{aligned}$$

Similar to the one-dimensional *Quantization Theorem*, a two-dimensional theorem [Wid61] can be formulated: the joint density function of the input can be reconstructed from the joint density function of the output, if $P_{X_1X_2}(ju_1, ju_2) = 0$ for $u_1 \geq u_o/2$ and $u_2 \geq u_o/2$. Here again, the moments of the joint density function can be calculated as follows:

$$E[\mathbf{Y_1}^m \mathbf{Y_2}^n] = (-j)^{m+n} \frac{\partial^{m+n}}{\partial u_1^m \partial u_2^n} P_{X_1X_2}(ju_1, ju_2) \frac{\sin(u_1 \frac{Q}{2})}{u_1 \frac{Q}{2}} \frac{\sin(u_2 \frac{Q}{2})}{u_2 \frac{Q}{2}} \Bigg|_{u_1=0, u_2=0} \tag{2.39}$$

From this, the autocorrelation function with $\kappa = t_2 - t_1$ can be written as

$$r_{yy}(\kappa) = E[\mathbf{Y_1 Y_2}](\kappa) = \begin{cases} E[\mathbf{X}^2] + \frac{Q^2}{12} & \kappa = 0 \\ E[\mathbf{X_1 X_2}](\kappa) & \text{elsewhere} \end{cases} \tag{2.40}$$

(for $\kappa = 0$ we obtain Equation (2.33)).

2.1.3 Statistics of Quantization Error

First-order Statistics of Quantization Error

The probability density function (PDF) of the quantization error depends on the PDF of the input and is dealt with in the following. The quantization error $e =$

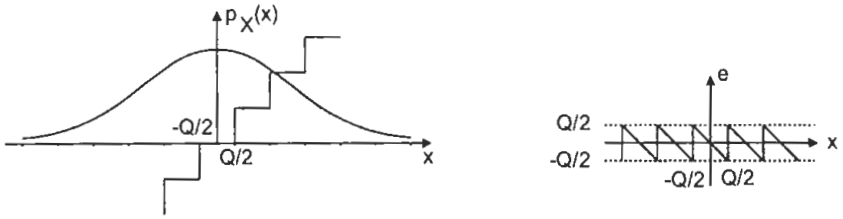


Figure 2.10 Probability density function and quantization error.

$x_Q - x$ is restricted to the interval $[-\frac{Q}{2}, \frac{Q}{2}]$. It depends linearly on the input (see Fig.2.10). If the input value lies in the interval $[-\frac{Q}{2}, \frac{Q}{2}]$ then the error is $e = 0 - x$. For the PDF we obtain $p_E(e) = p_X(e)$. If the input value lies in the interval $[-\frac{Q}{2} + Q, \frac{Q}{2} + Q]$ then the quantization error is $e = Q\lfloor Q^{-1}x + 0.5 \rfloor - x$ and is again restricted to $[-\frac{Q}{2}, \frac{Q}{2}]$. The PDF of the quantization error is consequently $p_E(e) = p_X(e + Q)$ and is added to the first term. For the sum over all intervals we can write

$$p_E(e) = \begin{cases} \sum_{k=-\infty}^{\infty} p_X(e - kQ) & -\frac{Q}{2} \le e < \frac{Q}{2} \\ 0 & \text{elsewhere} \end{cases} . \tag{2.41}$$

Because of the restricted values of the variable of the PDF, we can write

$$p_E(e) = \text{rect}(\frac{e}{Q}) \sum_{k=-\infty}^{\infty} p_X(e - kQ) \tag{2.42}$$

$$= \text{rect}(\frac{e}{Q})[p_X(e) \star \delta_Q(e)]. \tag{2.43}$$

The PDF of the quantization error is determined by the PDF of the input and can be computed by shifting and windowing a zone. All individual zones are summed up for calculating the PDF of the quantization error [Lip92]. A simple graphical interpretation of this overlapping is shown in Fig. 2.11. The overlapping leads to

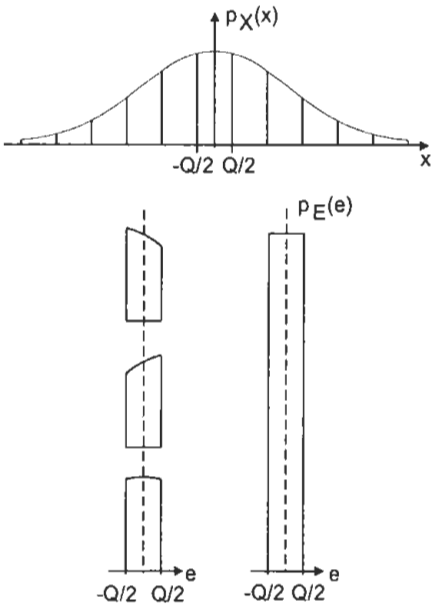


Figure 2.11 Probability density function of the quantization error.

a uniform distribution of the quantization error if the input PDF $p_X(x)$ is spread over a sufficient number of quantization intervals.

For the Fourier transform of the PDF from (2.43) follows

$$P_E(ju) = \frac{1}{2\pi} Q \frac{\sin(u\frac{Q}{2})}{u\frac{Q}{2}} \star \left[P_X(ju) \frac{2\pi}{Q} \sum_{k=-\infty}^{\infty} \delta(u - ku_o) \right] \tag{2.44}$$

$$= \frac{\sin(u\frac{Q}{2})}{u\frac{Q}{2}} \star \left[\sum_{k=-\infty}^{\infty} P_X(jku_o)\delta(u - ku_o) \right] \tag{2.45}$$

$$= \sum_{k=-\infty}^{\infty} P_X(jku_o) \left[\frac{\sin(u\frac{Q}{2})}{u\frac{Q}{2}} \star \delta(u - ku_o) \right] \tag{2.46}$$

$$\boxed{P_E(ju) = \sum_{k=-\infty}^{\infty} P_X(jku_o) \frac{\sin[(u - ku_o)\frac{Q}{2}]}{(u - ku_o)\frac{Q}{2}}} \tag{2.47}$$

If the Quantization Theorem is satisfied, i.e. if $P_X(ju) = 0$ for $u > u_o/2$, then there is only one non-zero term ($k = 0$ in Equ. (2.47)). The characteristic function of the quantization error is reduced, with $P_X(0) = 1$, to

$$P_E(ju) = \frac{\sin(u\frac{Q}{2})}{u\frac{Q}{2}}. \tag{2.48}$$

Hence, for the quantization error:

$$p_E(e) = \frac{1}{Q}\text{rect}(\frac{e}{Q}). \tag{2.49}$$

Sripad and Snyder [Sri77] have modified the sufficient condition of Widrow (band-limited characteristic function of input) for a quantization error of uniform PDF by the weaker condition

$$\boxed{P_X(jku_o) = P_X(j\frac{2\pi k}{Q}) = 0 \qquad \text{for all } k \neq 0.} \tag{2.50}$$

The uniform distribution of the input PDF

$$p_X(x) = \frac{1}{Q}\text{rect}(\frac{x}{Q}) \tag{2.51}$$

with characteristic function

$$P_X(ju) = \frac{\sin(u\frac{Q}{2})}{u\frac{Q}{2}} \tag{2.52}$$

does not satisfy Widrow's condition for a band-limited characteristic function, but instead the weaker condition

$$P_X(j\frac{2\pi k}{Q}) = \frac{\sin(\pi k)}{\pi k} = 0 \qquad \text{for all } k \neq 0 \tag{2.53}$$

is fulfilled. From this follows the uniform PDF (2.48) of the quantization error. The weaker condition from Sripad and Snyder extends the class of input signals for which a uniform PDF of the quantization error can be assumed.

In order to show the deviation from the uniform PDF of the quantization error as a function of the PDF of the input, (2.47) can be written as

$$\begin{aligned} P_E(ju) &= P_X(0)\frac{\sin[u\frac{Q}{2}]}{u\frac{Q}{2}} + \sum_{k=-\infty,k\neq 0}^{\infty} P_X(j\frac{2\pi k}{Q})\frac{\sin[(u-ku_o)\frac{Q}{2}]}{(u-ku_o)\frac{Q}{2}} \\ &= \frac{\sin[u\frac{Q}{2}]}{u\frac{Q}{2}} + \sum_{k=-\infty,k\neq 0}^{\infty} P_X(j\frac{2\pi k}{Q})\frac{\sin[u\frac{Q}{2}]}{u\frac{Q}{2}} \star \delta(u-ku_0). \end{aligned} \tag{2.54}$$

The inverse Fourier transform yields

$$p_E(e) = \frac{1}{Q}\text{rect}(\frac{e}{Q})[1 + \sum_{k=-\infty,k\neq 0}^{\infty} P_X(j\frac{2\pi k}{Q})\exp(j\frac{2\pi ke}{Q})] \quad (2.55)$$

$$= \begin{cases} \frac{1}{Q}[1 + \sum_{k\neq 0}^{\infty} P_X(j\frac{2\pi k}{Q})\exp(j\frac{2\pi ke}{Q})] & -\frac{Q}{2} \leq e < \frac{Q}{2} \\ 0 & \text{elsewhere.} \end{cases} \quad (2.56)$$

Equation (2.55) shows the effect of the input PDF on the deviation from a uniform PDF.

Second-order Statistics of Quantization Error

For describing the spectral properties of the error signal, two values e_1 (at time t_1) and e_2 (at time t_2) are considered [Lip92]. The joint PDF is given by

$$p_{E_1E_2}(e_1, e_2) = \text{rect}(\frac{e_1}{Q}, \frac{e_2}{Q})[p_{X_1X_2}(e_1, e_2) \star \delta_{QQ}(e_1, e_2)]. \quad (2.57)$$

Here $\delta_{QQ}(e_1, e_2) = \delta_Q(e_1) \cdot \delta_Q(e_2)$ and $\text{rect}(\frac{e_1}{Q}, \frac{e_2}{Q}) = \text{rect}(\frac{e_1}{Q}) \cdot \text{rect}(\frac{e_2}{Q})$. For the Fourier transform of the joint PDF, a similar procedure to that shown by (2.44)-(2.47) leads to

$$P_{E_1E_2}(ju_1, ju_2) = \sum_{k_1=-\infty}^{\infty}\sum_{k_2=-\infty}^{\infty} P_{X_1X_2}(jk_1u_o, jk_2u_o) \frac{\sin[(u_1 - k_1u_o)\frac{Q}{2}]}{(u_1 - k_1u_o)\frac{Q}{2}} \frac{\sin[(u_2 - k_2u_o)\frac{Q}{2}]}{(u_2 - k_2u_o)\frac{Q}{2}} . \quad (2.58)$$

If the quantization theorem and/or the Sripad-Snyder condition

$$\boxed{P_{X_1X_2}(jk_1u_o, jk_2u_o) = 0 \qquad \text{for all } k_1, k_2 \neq 0} \quad (2.59)$$

are satisfied

$$P_{E_1E_2}(ju_1, ju_2) = \frac{\sin[u_1\frac{Q}{2}]}{u_1\frac{Q}{2}} \frac{\sin[u_2\frac{Q}{2}]}{u_2\frac{Q}{2}}. \quad (2.60)$$

For the joint PDF of the quantization error, it hence holds that

$$p_{E_1E_2}(e_1, e_2) = \frac{1}{Q}\text{rect}(\frac{e_1}{Q}) \cdot \frac{1}{Q}\text{rect}(\frac{e_2}{Q}) \qquad -\frac{Q}{2} \leq e_1, e_2 < \frac{Q}{2} \quad (2.61)$$

$$= p_{E_1}(e_1) \cdot p_{E_2}(e_2). \quad (2.62)$$

Due to the statistical independence of quantization errors (Equation (2.62)),

$$E[\mathbf{E_1}^m\mathbf{E_2}^n] = E[\mathbf{E_1}^m] \cdot E[\mathbf{E_2}^n]. \quad (2.63)$$

For the moments of the joint PDF,

$$E[\mathbf{E_1}^m \mathbf{E_2}^n] = (-j)^{m+n} \frac{\partial^{m+n}}{\partial u_1^m \partial u_2^n} P_{E_1 E_2}(u_1, u_2)\Big|_{u_1=0, u_2=0}. \tag{2.64}$$

From this, it follows for the autocorrelation function with $\kappa = t_2 - t_1$

$$r_{ee}(\kappa) = E[\mathbf{E_1}\mathbf{E_2}](\kappa) = \begin{cases} E[\mathbf{E}^2] & \kappa = 0 \\ E[\mathbf{E_1}\mathbf{E_2}](\kappa) & \text{elsewhere} \end{cases} \tag{2.65}$$

$$= \begin{cases} \frac{Q^2}{12} & \kappa = 0 \\ 0 & \text{elsewhere} \end{cases}. \tag{2.66}$$

Correlation of Signal and Quantization Error

For describing the correlation of the signal and the quantization error [Sri77], the second moment of the output with Equation (2.25) is derived as follows:

$$E[\mathbf{Y}^2] = (-j)^2 \frac{d^2 P_Y(ju)}{du^2}\Big|_{u=0} \tag{2.67}$$

$$= (-j)^2 \sum_{k=-\infty}^{\infty} \Big[\ddot{P}_X(-\frac{2\pi k}{Q}) \frac{\sin(\pi k)}{\pi k} + Q\dot{P}_X(-\frac{2\pi k}{Q}) \frac{\sin(\pi k) - \pi k \cos(\pi k)}{\pi^2 k^2} + \frac{Q^2}{4} P_X(-\frac{2\pi k}{Q}) \frac{(2 - \pi^2 k^2)\sin(\pi k) - 2\pi k \cos(\pi k)}{\pi^3 k^3} \Big] \tag{2.68}$$

$$= E[\mathbf{X}^2] + \frac{Q}{\pi} \sum_{k=-\infty, k\neq 0}^{\infty} \frac{(-1)^k}{k} \dot{P}_X(-\frac{2\pi k}{Q}) + E[\mathbf{E}^2]. \tag{2.69}$$

With the quantization error $e = y - x$,

$$E[\mathbf{Y}^2] = E[\mathbf{X}^2] + 2E[\mathbf{XE}] + E[\mathbf{E}^2], \tag{2.70}$$

where the term $E[\mathbf{XE}]$, with (2.69), is written as

$$E[\mathbf{XE}] = \frac{Q}{2\pi} \sum_{k=-\infty, k\neq 0}^{\infty} \frac{(-1)^k}{k} \dot{P}_X(-\frac{2\pi k}{Q}). \tag{2.71}$$

With the assumption of a Gaussian PDF of the input we obtain

$$p_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(\frac{-x^2}{2\sigma^2}) \tag{2.72}$$

with the characteristic function

$$P_X(ju) = \exp(\frac{-u^2\sigma^2}{2}). \tag{2.73}$$

Using (2.56) the PDF of the quantization error is then given by

$$p_E(e) = \begin{cases} \frac{1}{Q}\left[1 + 2\sum_{k=1}^{\infty}\cos(\frac{2\pi ke}{Q})\exp(-\frac{2\pi^2k^2\sigma^2}{Q^2})\right] & -\frac{Q}{2} \le e < \frac{Q}{2} \\ 0 & \text{elsewhere.} \end{cases} \tag{2.74}$$

Figure 2.12a shows the PDF (2.74) of the quantization error for different variances of the input.

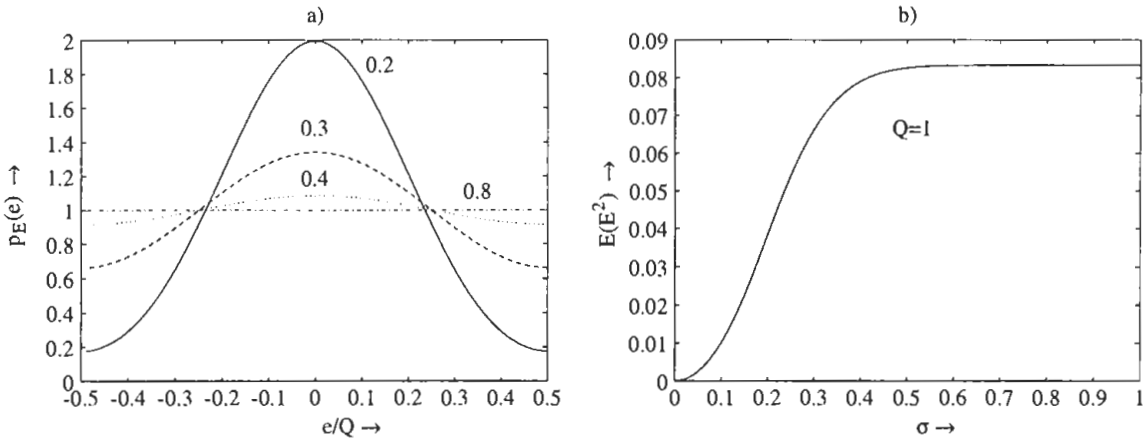


Figure 2.12 a) PDF of quantization error for different standard deviations of a Gaussian PDF input. b) Variance of quantization error for different standard deviation of a Gaussian PDF input.

For the mean value and the variance of a quantization error, it follows with Equation (2.74) that $E[\mathbf{E}] = 0$ and

$$E[\mathbf{E}^2] = \int_{-\infty}^{\infty} e^2 p_E(e)de = \frac{Q^2}{12}\left[1 + \frac{12}{\pi^2}\sum_{k=1}^{\infty}\frac{(-1)^k}{k^2}\exp(-\frac{2\pi^2k^2\sigma^2}{Q^2})\right]. \tag{2.75}$$

Figure 2.12b shows the variance of the quantization error (2.75) for different variances of the input.

For a Gaussian PDF input as given by (2.72) and (2.73), the correlation (see Equation (2.71)) between input and quantization error is expressed as

$$E[\mathbf{XE}] = 2\sigma^2\sum_{k=1}^{\infty}(-1)^k\exp(-\frac{2\pi^2k^2\sigma^2}{Q^2}). \tag{2.76}$$

The correlation is negligible for large values of $\frac{\sigma}{Q}$.

2.2 Dither

2.2.1 Basics

The *requantization* (renewed quantization of already quantized signals) to limited word-lengths occurs repeatedly during storage, format conversion and signal processing algorithms. Here, small signal levels lead to error signals which depend on the input. Owing to quantization, nonlinear distortion occurs for low level signals. The conditions for the classical quantization model are not satisfied anymore. To reduce these effects for signals of small amplitude, a linearization of the nonlinear characteristic curve of the quantizer is performed. This is done by adding a random sequence $d(n)$ to the quantized signal $x(n)$ (see Fig. 2.13) before the actual quantization process. The specification of the word-length is shown in Fig. 2.14. This random signal is called dither. The statistical independence of the error signal from the input is not achieved, but the conditional moments of the error signal can be affected [Lip92, Ger89, Wan92].

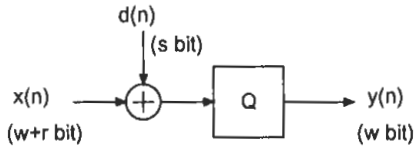


Figure 2.13 Addition of a random sequence before a quantizer.

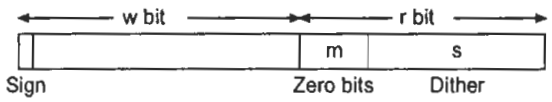


Figure 2.14 Specification of the word-length.

The sequence $d(n)$, with amplitude range $(-\frac{Q}{2} \leq d(n) \leq \frac{Q}{2})$, is generated with the help of a random number generator and is added to the input. For a dither value with $Q = 2^{-(w-1)}$:

$$d_k = k2^{-r}Q \qquad -2^{s-1} \leq k \leq 2^{s-1} - 1. \tag{2.77}$$

The index k of the random number d_k characterizes the value from the set of $N = 2^s$ possible numbers with the probability

$$P(d_k) = \begin{cases} 2^{-s} & -2^{s-1} \leq k \leq 2^{s-1} - 1 \\ 0 & \text{elsewhere} \end{cases} . \tag{2.78}$$

With the mean value $\overline{d} = \sum_k d_k P(d_k)$, the variance $\sigma_d^2 = \sum_k [d_k - \overline{d}]^2 P(d_k)$ and the quadratic mean $\overline{d^2} = \sum_k d_k^2 P(d_k)$, we can rewrite the variance as $\sigma_d^2 = \overline{d^2} - \overline{d}^2$.

For a static input amplitude V and the dither value d_k the rounding operation [Lip86] is expressed as

$$g(V + d_k) = Q \left\lfloor \frac{V + d_k}{Q} + 0.5 \right\rfloor . \tag{2.79}$$

For the mean of the output $\overline{g}(V)$ as a function of the input V, we can write

$$\overline{g}(V) = \sum_k g(V + d_k) P(d_k). \tag{2.80}$$

The quadratic mean of the output $\overline{g^2}(V)$ for input V is given by

$$\overline{g^2}(V) = \sum_k g^2(V + d_k) P(d_k). \tag{2.81}$$

For the variance $d_R^2(V)$ for input V

$$d_R^2(V) = \sum_k \{g(V + d_k) - \overline{g}(V)\}^2 P(d_k) = \overline{g^2}(V) - \{\overline{g}(V)\}^2. \tag{2.82}$$

The above-mentioned equations have the input V as a parameter. Figures 2.15 and 2.16 illustrate the mean output $\overline{g}(V)$ and the standard deviation $d_R(V)$ within a quantization step size, which are given by Equations (2.80), (2.81) and (2.82). The

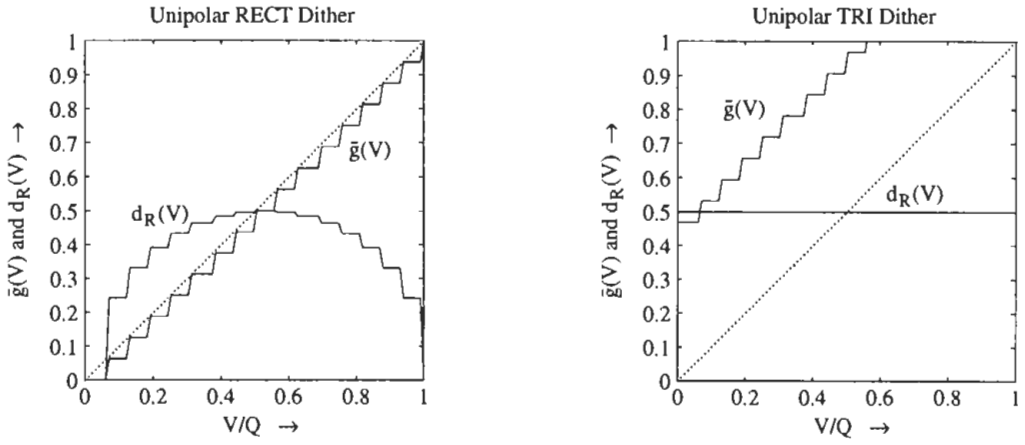


Figure 2.15 Truncation - linearizing and suppression of noise modulation ($s = 4$, $m = 0$).

examples of rounding and truncation demonstrate the linearization of the characteristic curve of the quantizer. The coarse step size is replaced by a finer one. The quadratic deviation from the mean output $d_R^2(V)$ is termed *noise modulation*. For a uniform PDF dither, this noise modulation depends on the amplitude (see Fig.

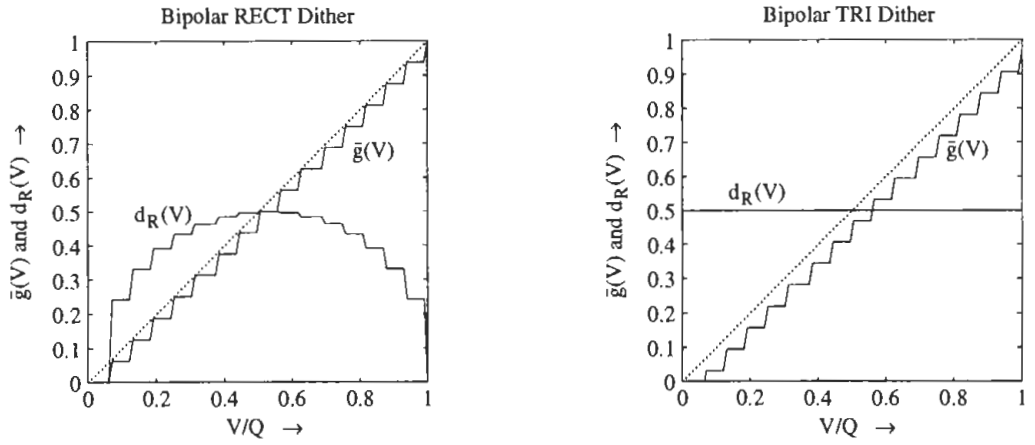


Figure 2.16 Rounding - linearizing and suppression of noise modulation ($s = 4$, $m = 1$).

2.15 and 2.16). It is maximum in the middle of the quantization step size and approaches zero towards the end. The linearization and the suppression of the noise modulation can be achieved by a triangular PDF dither with bipolar characteristic [Van89] and rounding operation (see Fig. 2.16). Triangular PDF dither is obtained by adding two statistically independent dither signals with uniform PDF (convolution of PDFs). A dither signal with higher-order PDF is not necessary for audio signals [Lip92].

The total noise power for this quantization technique consists of the dither power and the power of the quantization error [Lip86]. The following noise powers are obtained by integration with respect to V as follows:

1. Mean dither power d^2:

$$d^2 = \frac{1}{Q}\int_0^Q d_R^2(V)dV \tag{2.83}$$

$$= \frac{1}{Q}\int_0^Q \sum_k \{g(V+d_k) - \overline{g}(V)\}^2 P(d_k)\, dV \tag{2.84}$$

(This is equal to the deviation from mean output in accordance with Equation (2.80).)

2. Mean of total noise power d_{tot}^2:

$$d_{tot}^2 = \frac{1}{Q}\int_0^Q \sum_k \{g(V+d_k) - V\}^2 P(d_k)\, dV \tag{2.85}$$

(This is equal to the deviation from an ideal straight line.)

In order to derive a relationship between d_{tot}^2 and d^2, the quantization error given by

$$Q(V + d_k) = g(V + d_k) - (V + d_k) \tag{2.86}$$

is used to rewrite (2.85) as

$$\begin{aligned} d_{tot}^2 &= \sum_k P(d_k) \frac{1}{Q} \int_0^Q (Q^2(V + d_k) + 2d_k Q(V + d_k) + d_k^2)\, dV \qquad (2.87) \\ &= \sum_k P(d_k) \frac{1}{Q} \int_0^Q Q^2(V + d_k)\, dV \\ &\quad + 2 \sum_k d_k P(d_k) \frac{1}{Q} \int_0^Q Q(V + d_k)\, dV \\ &\quad + \sum_k d_k^2 P(d_k) \frac{1}{Q} \int_0^Q dV. \qquad (2.88) \end{aligned}$$

The integrals in (2.88) are independent of d_k. Moreover $\sum_k P(d_k) = 1$. With the mean value of the quantization error

$$\overline{e} = \frac{1}{Q} \int_0^Q Q(V)\, dV \tag{2.89}$$

and the quadratic mean error

$$\overline{e^2} = \frac{1}{Q} \int_0^Q Q^2(V)\, dV, \tag{2.90}$$

it is possible to rewrite (2.88) as

$$d_{tot}^2 = \overline{e^2} + 2\overline{d}\overline{e} + \overline{d^2}. \tag{2.91}$$

With $\sigma_e^2 = \overline{e^2} - \overline{e}^2$ and $\sigma_d^2 = \overline{d^2} - \overline{d}^2$, Equation (2.80) can be written as

$$\boxed{d_{tot}^2 = \sigma_e^2 + (\overline{d} + \overline{e})^2 + \sigma_d^2}. \tag{2.92}$$

Equation (2.91) and (2.92) describe the total noise power as a function of the quantization $(\overline{e}, \overline{e^2}, \sigma_e^2)$ and the dither addition $(\overline{d}, \overline{d^2}, \sigma_d^2)$. It can be seen that for zero-mean quantization, the middle term in Equation (2.92) results in $\overline{d} + \overline{e} = 0$. The acoustically perceptible part of the total error power is represented by σ_e^2 and σ_d^2.

2.2.2 Implementation

The random sequence $d(n)$ is generated with the help of a random number generator with uniform PDF. For generating a triangular PDF random sequence, two independent uniform PDF random sequences $d_1(n)$ and $d_2(n)$ can be added. In order to generate a triangular high-pass dither, the dither value $d_1(n)$ is added to $-d_1(n-1)$. Thus, only one random number generator is required. In conclusion, the following dither sequences can be implemented:

$$d_{\mathrm{RECT}}(n) = d_1(n) \tag{2.93}$$

$$d_{\mathrm{TRI}}(n) = d_1(n) + d_2(n) \tag{2.94}$$

$$d_{\mathrm{HP}}(n) = d_1(n) - d_1(n-1). \tag{2.95}$$

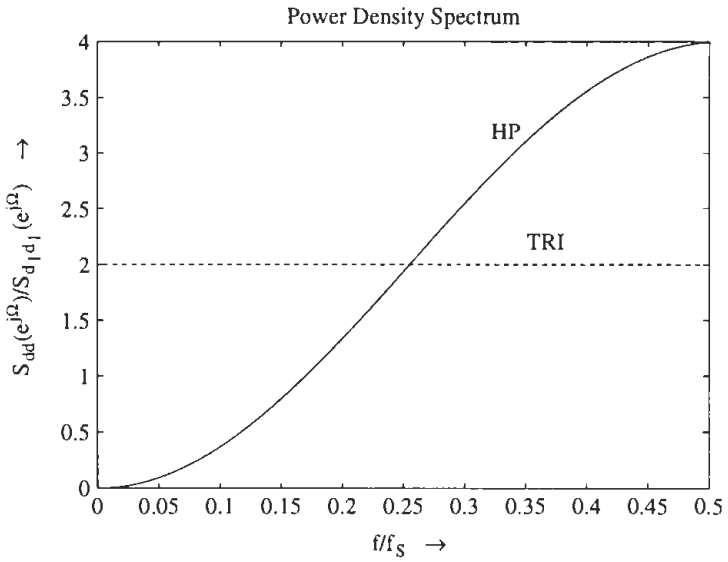


Figure 2.17 Normalized power density spectrum for triangular PDF dither (TRI) with $d_1(n) + d_2(n)$ and triangular PDF high-pass dither (HP) with $d_1(n) - d_1(n-1)$.

The power density spectra of triangular PDF dither and triangular PDF HP dither are shown in Fig. 2.17. Figure 2.18 shows histograms of a uniform PDF dither and a triangular PDF high-pass dither together with their respective power density spectra. The amplitude range of a uniform PDF dither lies between $\pm Q/2$ whereas it lies between $\pm Q$ for triangular PDF dither. The total noise power for triangular PDF dither is doubled.

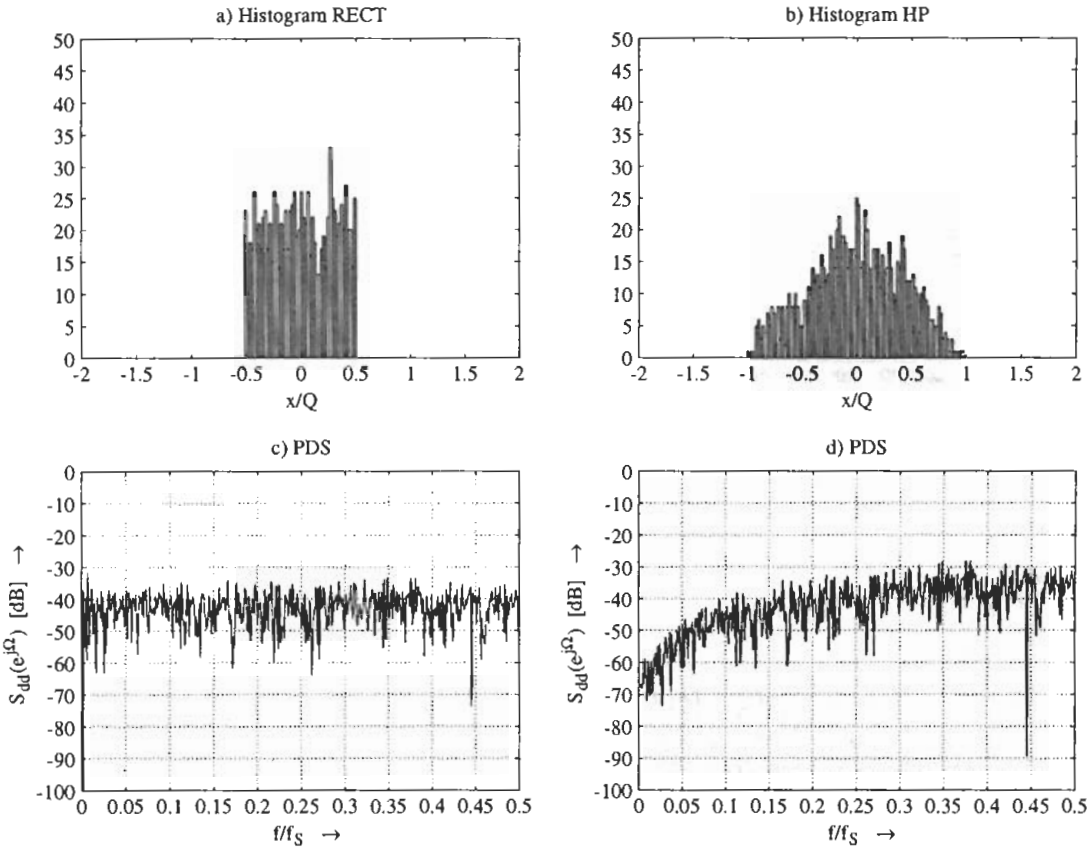


Figure 2.18 a,d) Histogram and c,d) power density spectrum of uniform PDF dither (RECT) with $d_1(n)$ and triangular PDF high-pass dither (HP) with $d_1(n) - d_1(n-1)$.

2.2.3 Examples

The effect of the input amplitude of the quantizer is shown in Fig. 2.19 for a 16 bit quantizer ($Q = 2^{-15}$). A quantized sinusoidal signal with amplitude 2^{-15} (1 bit amplitude) and frequency $f/f_S = 64/1024$ is shown in Fig. 2.19a,b for rounding and truncation. Figure 2.19c,d show their corresponding spectra. For truncation, Fig. 2.19c shows the spectral line of the signal and the spectral distribution of the quantization error with the harmonics of the input signal. For rounding Fig. 2.19d with special signal frequency $f/f_S = 64/1024$, the quantization error is concentrated in uneven harmonics.

In the following, only the rounding operation is used. By adding a uniform PDF random signal to the actual signal before quantization, the quantized signal shown in Fig. 2.20a results. The corresponding power density spectrum is illustrated in Fig. 2.20c. In the time domain, it is observed that the 1 bit amplitudes approach zero so that the regular pattern of the quantized signal is affected. The resulting

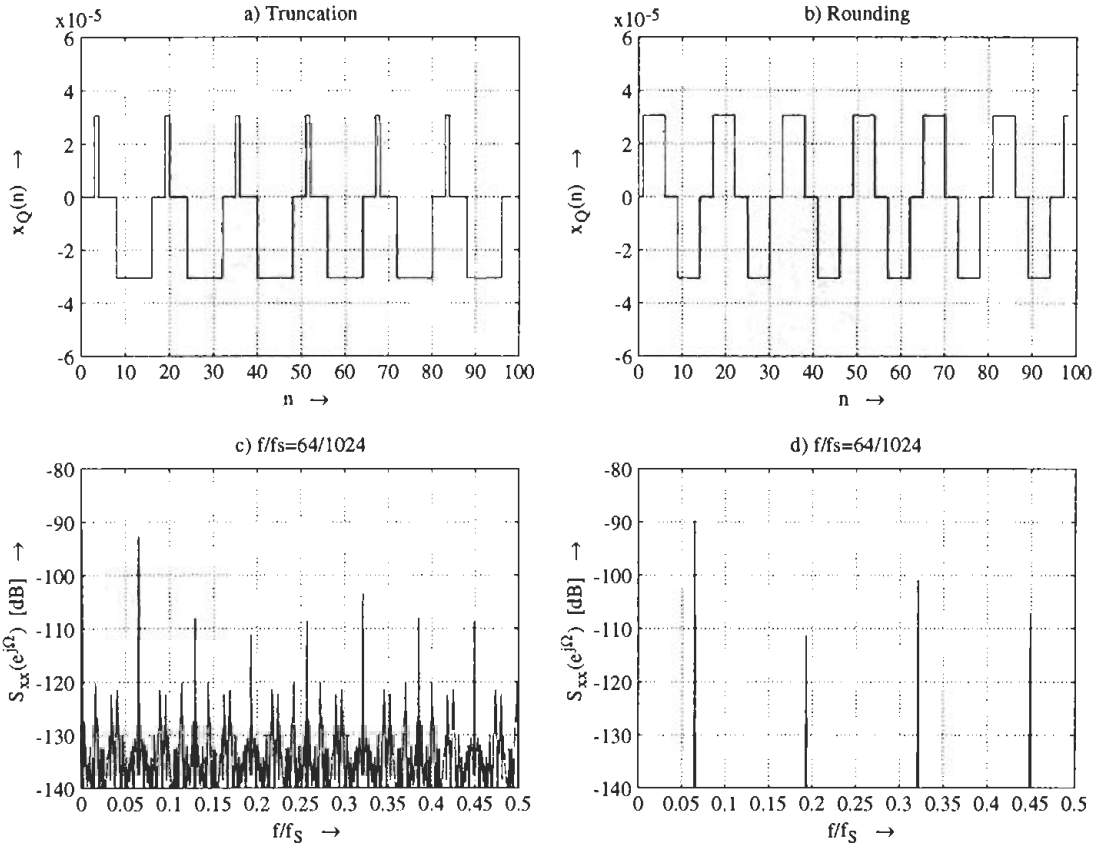


Figure 2.19 1 bit amplitude - quantizer with truncation (a,c) and rounding (b,d).

power density spectrum in Fig. 2.20c shows that the harmonics do not occur anymore and the noise power is uniformly distributed over the frequencies. For triangular PDF dither, the quantized signal is shown in Fig. 2.20b. Owing to triangular PDF, amplitudes $\pm 2Q$ occur besides the signal values $\pm Q$ and zero. Figure 2.20d shows the increase of the total noise power.

In order to illustrate the noise modulation for uniform PDF dither, the amplitude of the input is reduced to $A = 2^{-18}$ and the frequency is chosen as $f/f_S = 14/1024$. This means that input amplitude to the quantizer is 0.25 bit. For a quantizer without additive dither, the quantized output signal is zero. For RECT dither, the quantized signal is shown in Fig. 2.21a. The unquantized input signal with amplitude $0.25Q$ is also shown. The power density spectrum of the quantized signal is shown in Fig. 2.21c. The spectral line of the signal and the uniform distribution of the quantization error can be seen. But in the time domain, a correlation between positive and negative amplitudes of the input and the quantized positive and negative values of the output can be observed. In hearing tests this noise modulation occurs if the amplitude of the input is decreased

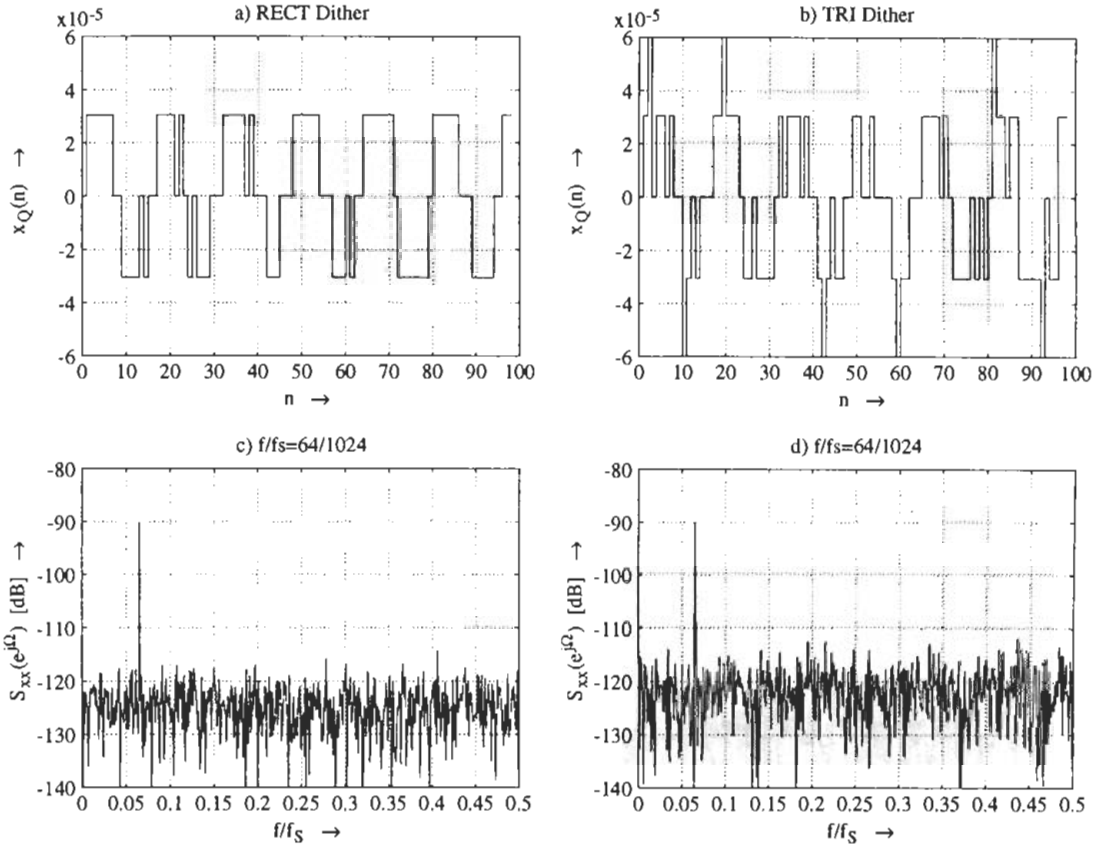


Figure 2.20 1 bit amplitude - rounding with RECT dither (a,c) and TRI dither (b,d).

continuously and falls below the amplitude of the quantization step. This process occurs for all fade-out processes that occur in speech and music signals. For positive low-amplitude signals, two output states, zero and Q, occur, and for negative low-amplitude signals, the output states zero and -Q, occur. This is observed as a disturbing rattle which is overlapped to the actual signal. If the input level is further reduced the quantized output approaches zero.

In order to reduce this noise modulation at low levels, a triangular PDF dither is used. Figure 2.21b shows the quantized signal and Fig. 2.21d shows the power density spectrum. It can be observed that the quantized signal has an irregular pattern. Hence a direct association of positive half-waves with the positive output values as well as vice versa is not possible. The power density spectrum shows that spectral line of the signal along with an increase in noise power owing to triangular PDF dither. In acoustic hearing tests, the use of triangular PDF dither results in a constant noise floor even if the input level is reduced to zero.

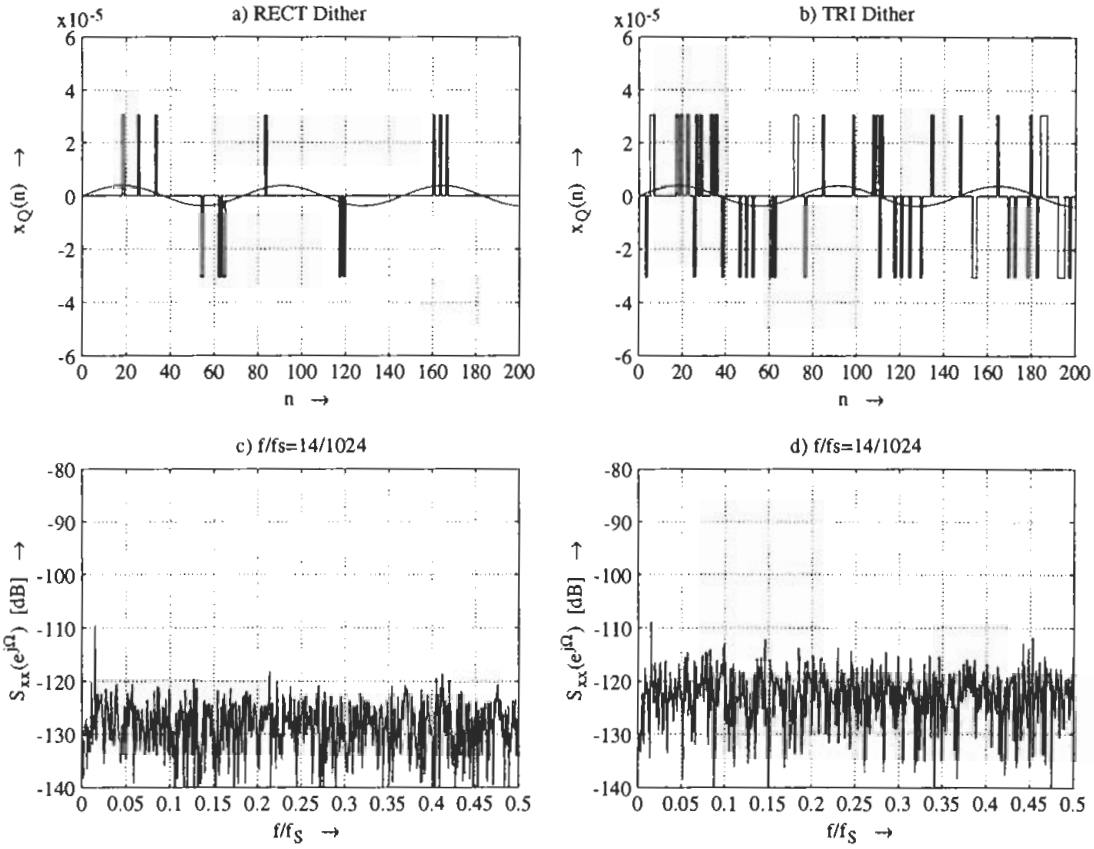


Figure 2.21 0.25 bit amplitude - rounding with RECT dither (a,c) and TRI dither (b,d).

2.3 Spectrum Shaping of Quantization - Noise Shaping

Using the linear model of a quantizer in Fig. 2.22 and the relations

$$\begin{aligned} e(n) &= y(n) - x(n) && (2.96)\\ y(n) &= [x(n)]_Q && (2.97)\\ &= x(n) + e(n), && (2.98) \end{aligned}$$

the quantization error $e(n)$ may be isolated and fed back through a transfer function $H(z)$ as shown in Fig. 2.23. This leads to the spectral shaping of the quantization error as given by

$$\begin{aligned} y(n) &= [x(n) - e(n) \star h(n)]_Q && (2.99)\\ &= x(n) + e(n) - e(n) \star h(n) && (2.100) \end{aligned}$$

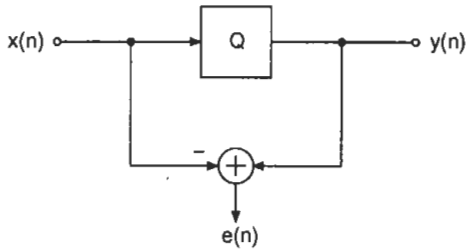


Figure 2.22 Linear model of quantizer.

$$e_1(n) = y(n) - x(n) \tag{2.101}$$
$$= e(n) \star [1 - h(n)] \tag{2.102}$$

and the corresponding Z-transforms

$$Y(z) = X(z) + E(z)[1 - H(z)] \tag{2.103}$$
$$E_1(z) = E(z)[1 - H(z)]. \tag{2.104}$$

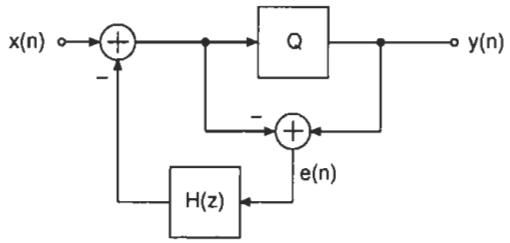


Figure 2.23 Spectrum shaping of quantization error.

A simple spectrum shaping of the quantization error $e(n)$ is achieved by feeding back with $H(z) = z^{-1}$ as shown in Fig. 2.24, and leads to

$$y(n) = [x(n) - e(n-1)]_Q \tag{2.105}$$
$$= x(n) - e(n-1) + e(n) \tag{2.106}$$
$$e_1(n) = y(n) - x(n) \tag{2.107}$$
$$= e(n) - e(n-1) \tag{2.108}$$

and the Z-transforms

$$Y(z) = X(z) + E(z)[1 - z^{-1}] \tag{2.109}$$
$$E_1(z) = E(z)[1 - z^{-1}]. \tag{2.110}$$

Equation (2.110) shows a high-pass weighting of the original error signal $e(n)$. By

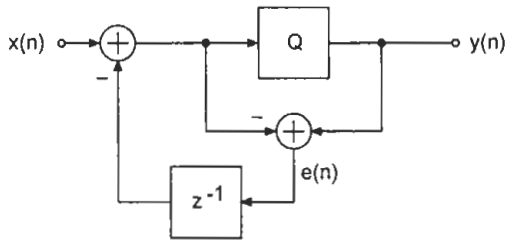


Figure 2.24 High-pass spectrum shaping of quantization error.

choosing $H(z) = z^{-1}(-2 + z^{-1})$, second-order high-pass weighting given by

$$E_2(z) = E(z)[1 - 2z^{-1} + z^{-2}] \tag{2.111}$$

can be achieved. The power density spectrum of the error signal for the two cases is given by

$$S_{e_1e_1}(e^{j\Omega}) = |1 - e^{-j\Omega}|^2 S_{ee}(e^{j\Omega}) \tag{2.112}$$

$$S_{e_2e_2}(e^{j\Omega}) = |1 - 2e^{-j\Omega} + e^{-j2\Omega}|^2 S_{ee}(e^{j\Omega}). \tag{2.113}$$

Figure 2.25 shows the weighting of power density spectrum by this noise shaping technique.

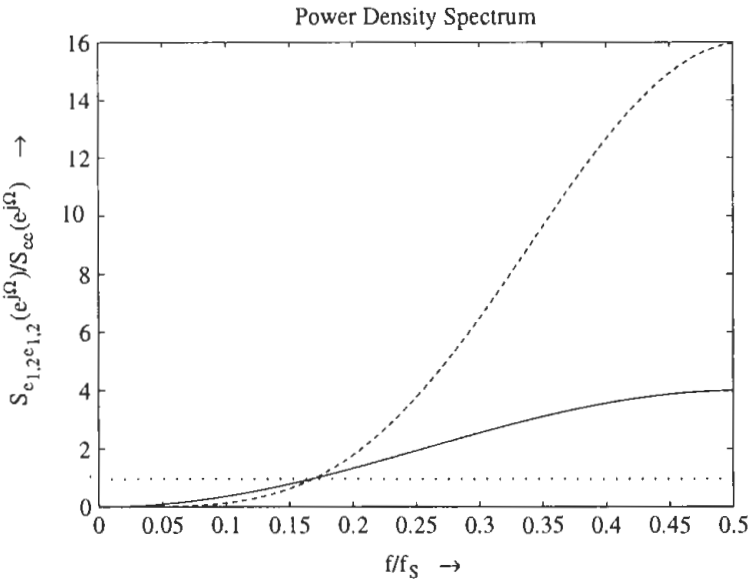


Figure 2.25 Spectrum shaping ($S_{ee}(e^{j\Omega})$ $\cdots$, $S_{e_1e_1}(e^{j\Omega})$ —, $S_{e_2e_2}(e^{j\Omega})$ - - -).

By adding a dither signal $d(n)$ (see Fig. 2.26), the output and the error are given by

$$y(n) = [x(n) + d(n) - e(n-1)]_Q \tag{2.114}$$

$$= x(n) + d(n) - e(n-1) + e(n) \tag{2.115}$$

and

$$e_1(n) = y(n) - x(n) \tag{2.116}$$
$$= d(n) + e(n) - e(n-1). \tag{2.117}$$

For the Z-transforms we write

$$Y(z) = X(z) + E(z)[1 - z^{-1}] + D(z) \tag{2.118}$$
$$E_1(z) = E(z)[1 - z^{-1}] + D(z). \tag{2.119}$$

The modified error signal $e_1(n)$ consists of the dither and the high-pass shaped quantization error.

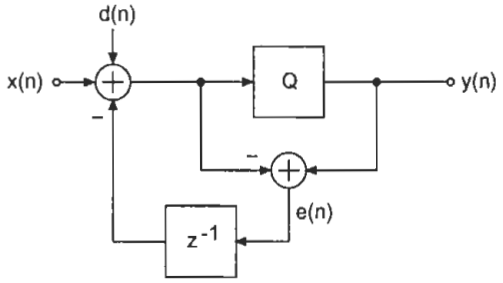


Figure 2.26 Dither and spectrum shaping.

By moving the addition (Fig. 2.27) of the dither directly before the quantizer, a high-pass spectrum shaping is obtained for both the error signal and the dither. Here the following relationships hold

$$y(n) = [x(n) + d(n) - e_0(n-1)]_Q \tag{2.120}$$
$$= x(n) + d(n) - e_0(n-1) + e(n) \tag{2.121}$$
$$e_0(n) = y(n) - [x(n) - e_0(n-1)] \tag{2.122}$$
$$= d(n) + e(n) \tag{2.123}$$
$$y(n) = x(n) + d(n) - d(n-1) + e(n) - e(n-1) \tag{2.124}$$
$$e_1(n) = d(n) - d(n-1) + e(n) - e(n-1) \tag{2.125}$$

with the Z-transforms given by

$$Y(z) = X(z) + E(z)[1 - z^{-1}] + D(z)[1 - z^{-1}] \tag{2.126}$$
$$E_1(z) = E(z)[1 - z^{-1}] + D(z)[1 - z^{-1}]. \tag{2.127}$$

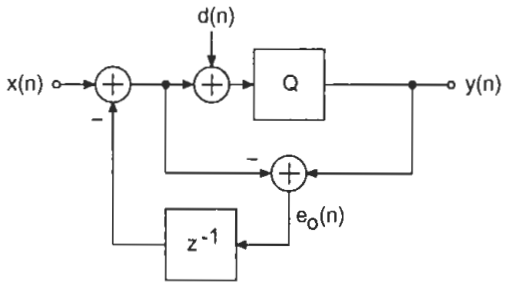


Figure 2.27 Modified dither and spectrum shaping.

Apart from the discussed feedback structures which are easy to implement on a digital signal processor and which lead to high-pass noise shaping, there are psychoacoustic-based noise shaping methods proposed in the literature [Ger89, Wan92]. These methods use special approximations of the hearing threshold (threshold in quiet, absolute threshold) for the feedback structure $1 - H(z)$. Figure 2.28a shows the hearing threshold as a function of frequency. It can be seen that the

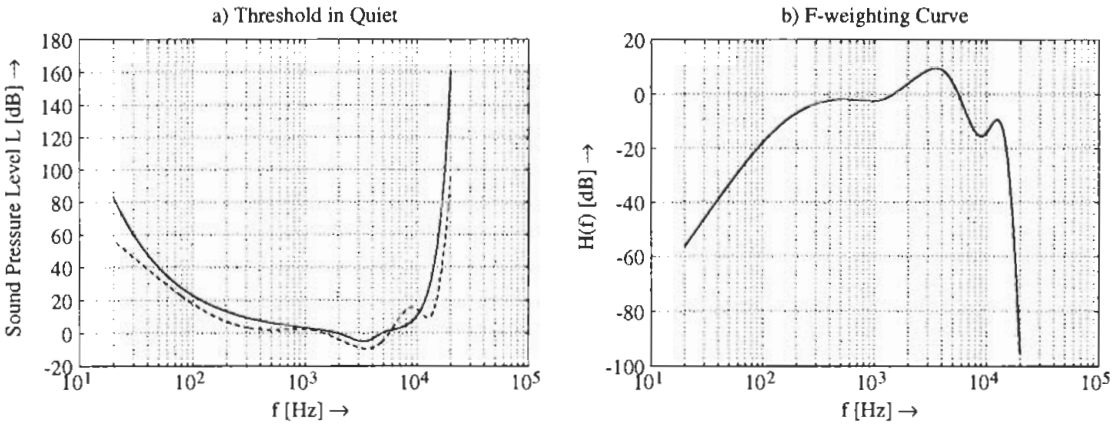


Figure 2.28 a) Threshold in quiet (—) and inverse F-weighting curve (···). b) F-weighting curve.

sensitivity of human hearing is high for frequencies between 2 and 6 kHz and sharply decreases for high and low frequencies. Figure 2.28a also shows an inverse F-weighting curve which represents an approximation of the threshold in quiet. The feedback filter should affect the quantization error with the inverse F-weighting curve. Hence, the noise power in the frequency range with high sensitivity should be reduced and shifted towards lower and higher frequencies. Figure 2.29a shows the power density spectrum of the quantization error for two special filters $H(z)$ [Wan92]. The power density spectrum with the F-weighting is illustrated in Fig. 2.29b. It can be seen that the perceived noise power is reduced by 15 dB for the

dashed curve. This is equivalent to a 2.5 bit increase in word-length. Gerzon has given a theoretical limit of 27 dB, which can be achieved by a direct weighting with the filter characteristic of the threshold in quiet [Ger89].

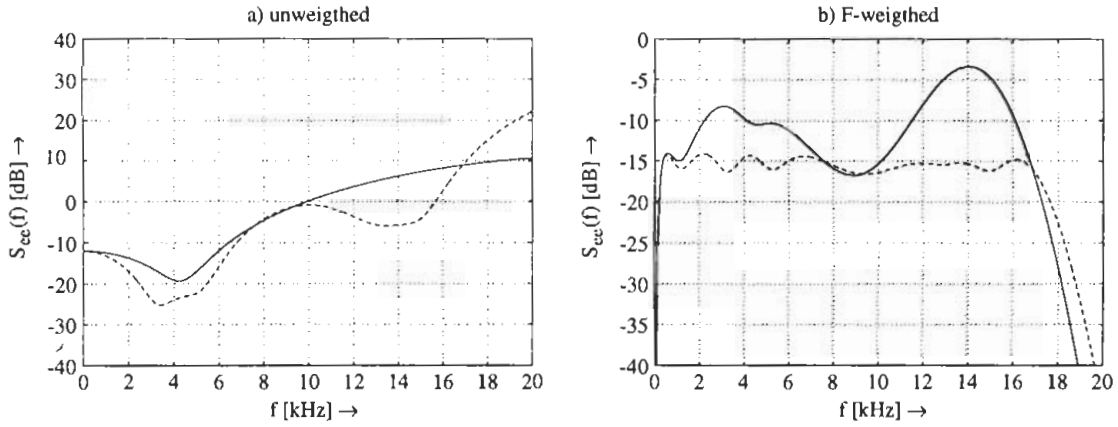


Figure 2.29 Power density spectrum of two filter approximations (filter 1 ···, filter 2 —): a) unweighted, b) F-weighted.

2.4 Number Representation

The different applications in digital signal processing and transmission of audio signals leads to the question of the type of number representation for digital audio signals. In this section, basic properties of fixed-point and floating-point number representation in the context of digital audio signal processing are presented.

2.4.1 Fixed-point Number Representation

In general, an arbitrary real number x can be approximated by a finite summation

$$x_Q = \sum_{i=0}^{w-1} b_i 2^i, \tag{2.128}$$

where the possible values for b_i are 0 and 1.

The fixed-point number representation with a finite number w of binary places leads to four different interpretations of the number range (see Table 2.1 and Fig. 2.30).

Table 2.1 Bit location and range of values.

type	bit location	range of values
signed 2s c.	$x_Q = -b_0 + \sum_{i=1}^{w-1} b_{-i}2^{-i}$	$-1 \leq x_Q \leq 1 - 2^{-(w-1)}$
unsigned 2s c.	$x_Q = \sum_{i=1}^{w} b_{-i}2^{-i}$	$0 \leq x_Q \leq 1 - 2^{-w}$
signed int.	$x_Q = -b_{w-1}2^{w-1} + \sum_{i=0}^{w-2} b_i 2^i$	$-2^{w-1} \leq x_Q \leq 2^{w-1} - 1$
unsigned int.	$x_Q = \sum_{i=0}^{w-1} b_i 2^i$	$0 \leq x_Q \leq 2^w - 1$

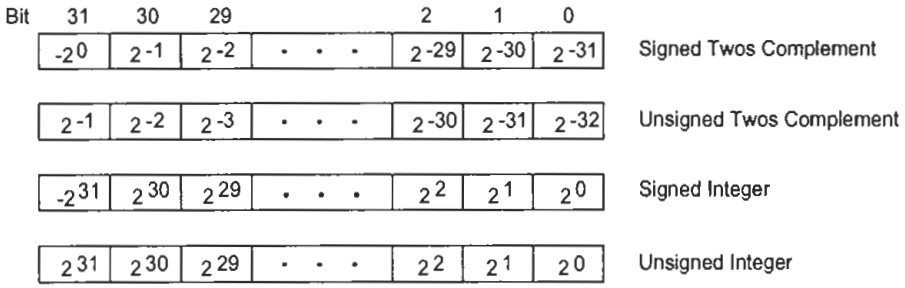


Figure 2.30 Fixed-point formats.

The signed fractional representation (2s complement) is the usual format for digital audio signals and for algorithms in fixed-point arithmetic. For address and modulo operation, the unsigned integer is used. Owing to finite word-length w, overflow occurs as shown in Fig. 2.31. These curves have to be taken into consideration while carrying out operations, especially additions in 2s complement arithmetic.

Quantization is carried out with techniques as shown in Table 2.2 for rounding and truncation. The quantization step size is characterized by $Q = 2^{-(w-1)}$ and the operation $\lfloor x \rfloor$ denotes an integer smaller than x. Figure 2.32 shows the rounding

Table 2.2 Rounding and truncation of 2s complement numbers.

type	quantization	error limits
2s c. (r)	$x_Q = Q\lfloor Q^{-1}x + 0.5 \rfloor$	$-Q/2 \leq x_Q - x \leq Q/2$
2s c. (t)	$x_Q = Q\lfloor Q^{-1}x \rfloor$	$-Q \leq x_Q - x \leq 0$

and truncation curves for 2s complement number representation. The absolute

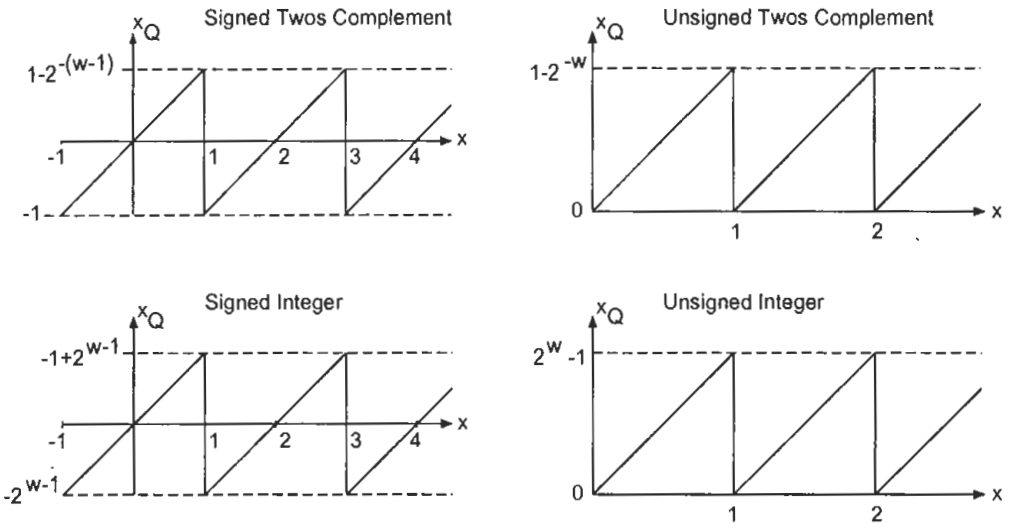


Figure 2.31 Number range.

error shown in Fig. 2.32 is given by

$$e = x_Q - x. \tag{2.129}$$

Digital audio signals are coded in the 2s complement number representation. For 2s complement representation, the range of values from $-X_{max}$ to $+X_{max}$ is normalized to the range -1 to +1 and is represented by the weighted finite sum $x_Q = -b_0 + b_1 \cdot 0.5 + b_2 \cdot 0.25 + b_3 \cdot 0.125 + \cdots + b_{w-1} \cdot 2^{-(w-1)}$. The variables b_0 to b_{w-1} are called bits and can take the values 1 or 0. The bit b_0 is called MSB (most significant bit) and b_{w-1} is called LSB (least significant bit). For positive numbers, b_0 is equal to 0 and for negative numbers b_0 equals 1. For a 3 bit quantization (see Fig. 2.33), a quantized value can be represented by $x_Q = -b_0 + b_1 \cdot 0.5 + b_2 \cdot 0.25$. The smallest quantization step size is 0.25. For a positive number 0.75 it follows that $0.75 = -0 + 1 \cdot 0.5 + 1 \cdot 0.25$. The binary coding for 0.75 is 011.

Dynamic Range. The dynamic range of a number representation is defined as the ratio of maximum to minimum number. For fixed-point representation with

$$x_{Qmax} = (1 - 2^{-(w-1)}) \tag{2.130}$$

$$x_{Qmin} = 2^{-(w-1)} \tag{2.131}$$

the dynamic range is given by

$$\begin{aligned} \mathrm{DR}_F &= 20 \log_{10} \left(\frac{x_{Qmax}}{x_{Qmin}} \right) = 20 \log_{10} \left(\frac{1-Q}{Q} \right) \\ &= 20 \log_{10}(2^{w-1} - 1) \quad [\mathrm{dB}]. \end{aligned} \tag{2.132}$$

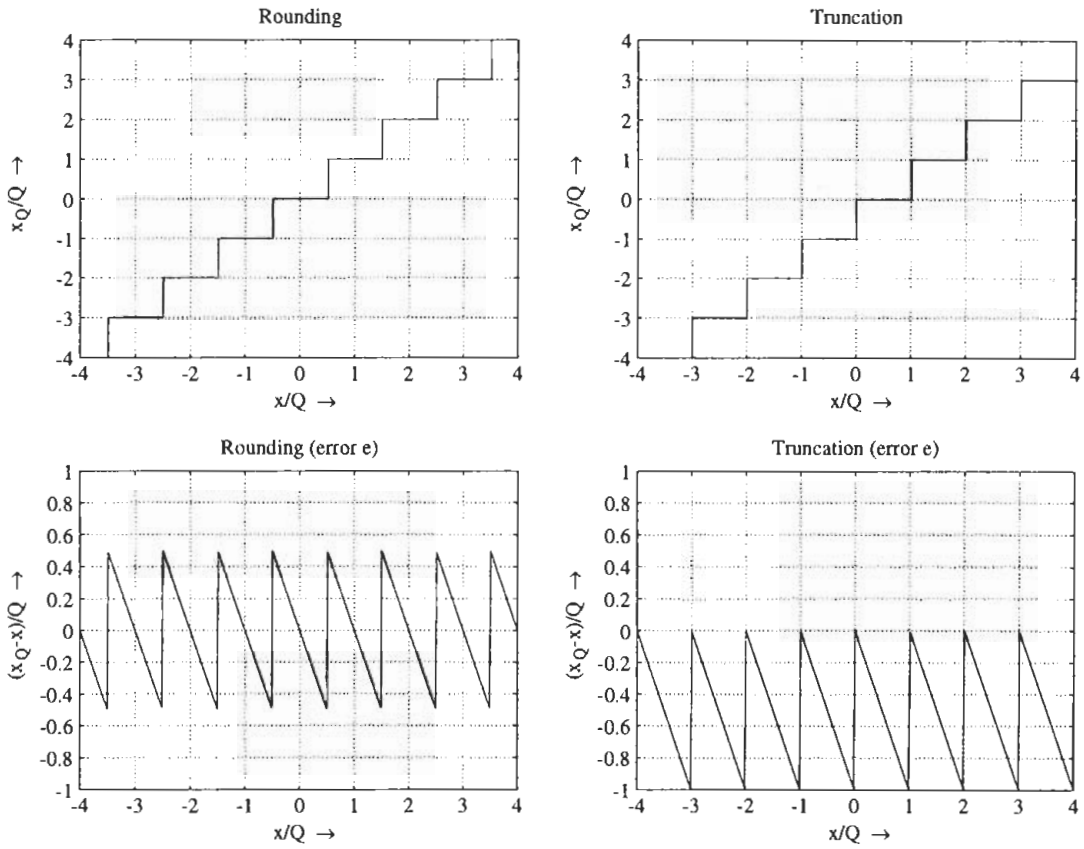


Figure 2.32 Rounding and truncation curves.

Multiplication and Addition of Fixed-point Numbers. For the multiplication of two fixed-point numbers in the range from -1 to +1, the result is always lesser than 1. For the addition of two fixed-point numbers, care must be taken for the result to remain in the range from -1 to +1. An addition of $0.6 + 0.7 = 1.3$ must be carried out in the form $0.5(0.6 + 0.7) = 0.65$. This multiplication with the factor 0.5 or generally 2^{-s} is called scaling. An integer in the range from 1 to, for instance, 8 is chosen for the scaling coefficient s.

Error Model. The quantization process for fixed-point numbers can be approximated as an the addition of an error signal e to the signal x (see Fig. 2.34). The error signal is a random signal with white power density spectrum.

Signal-to-noise Ratio. The signal-to-noise ratio for a fixed-point quantizer is defined by

$$\mathrm{SNR} = 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_e^2} \right), \tag{2.133}$$

where σ_x^2 is the signal power and σ_e^2 is the noise power.

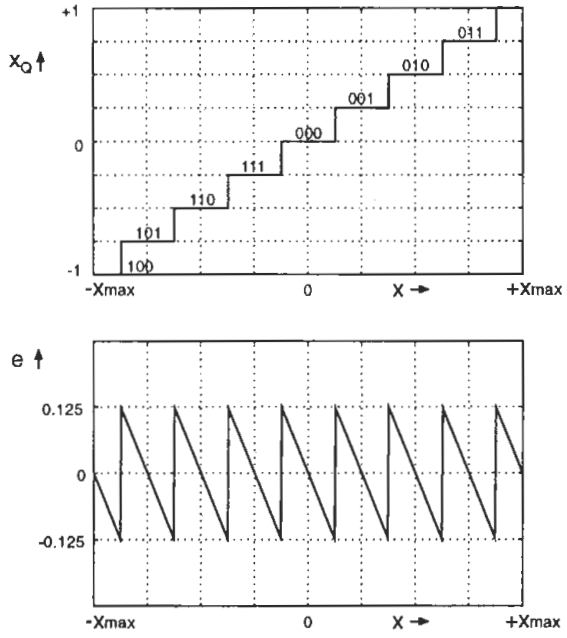


Figure 2.33 Rounding curve and error signal for $w = 3$ bit.

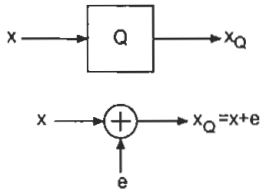


Figure 2.34 Model of a fixed-point quantizer.

2.4.2 Floating-point Number Representation

The representation of a floating-point number is given by

$$x_Q = M_G \, 2^{E_G} \tag{2.134}$$

with

$$0.5 \leq M_G < 1, \tag{2.135}$$

where M_G denotes the normalized mantissa and E_G the exponent. The normalized standard format (IEEE) is shown in Fig. 2.35 and special cases are given in Table 2.3. The mantissa M is implemented with a word-length of w_M bits and is in fixed-point number representation. The exponent E is implemented with a word-length of w_E bits and is an integer in the range from $-2^{w_E-1}+2$ to $2^{w_E-1}-1$. For an exponent word-length of $w_E = 8$ bits, its range of values lies between -126 and

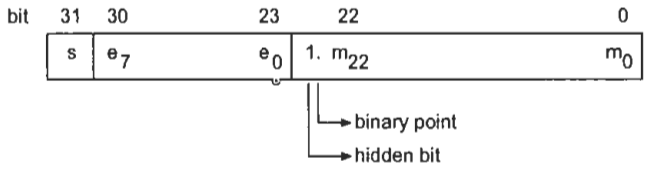


Figure 2.35 Floating-point number representation.

Table 2.3 Special cases of floating-point number representation.

type	exponent	mantissa	value
NAN	255	$\neq 0$	undefined
infinity	255	0	$(-1)^s$ infinity
normal	$1 \leq e \leq 254$	any	$(-1)^s(0.m)2^{e-127}$
zero	0	0	$(-1)^s \cdot 0$

+127. The range of values of the mantissa lies between 0.5 and 1. This is denoted as the normalized mantissa and is responsible for a unique representation of a number. For a fixed-point number in the range between 0.5 and 1, it follows that the exponent of the floating-point number representation is $E = 0$. For representing a fixed-point number in the range between 0.25 and 0.5 in floating-point representation, the range of values of the normalized mantissa M lies between 0.5 and 1, and for the exponent it follows $E = -1$. As an example, for a fixed-point number 0.75 the floating-point number $0.75 \cdot 2^0$ results. The fixed-point number 0.375 is not represented as the floating-point number $0.375 \cdot 2^0$. With the normalized mantissa, the floating-point number is expressed as $0.75 \cdot 2^{-1}$. Owing to normalization, the ambiguity of floating-point number representation is avoided. Numbers > 1 can be represented. For example, 1.5 becomes $0.75 \cdot 2^1$ in floating-point number representation.

Figure 2.36 shows the rounding and truncations curves for floating-point representation and the absolute error $e = x_Q - x$. The curves for floating-point quantization show that for small amplitudes small quantization steps sizes occur. In contrast to fixed-point representation the absolute error is dependent on the input signal.

In the interval

$$2^{E_G} \leq x < 2^{E_G+1} \tag{2.136}$$

the quantization step is given by

$$Q_G = 2^{-(w_M-1)}2^{E_G}. \tag{2.137}$$

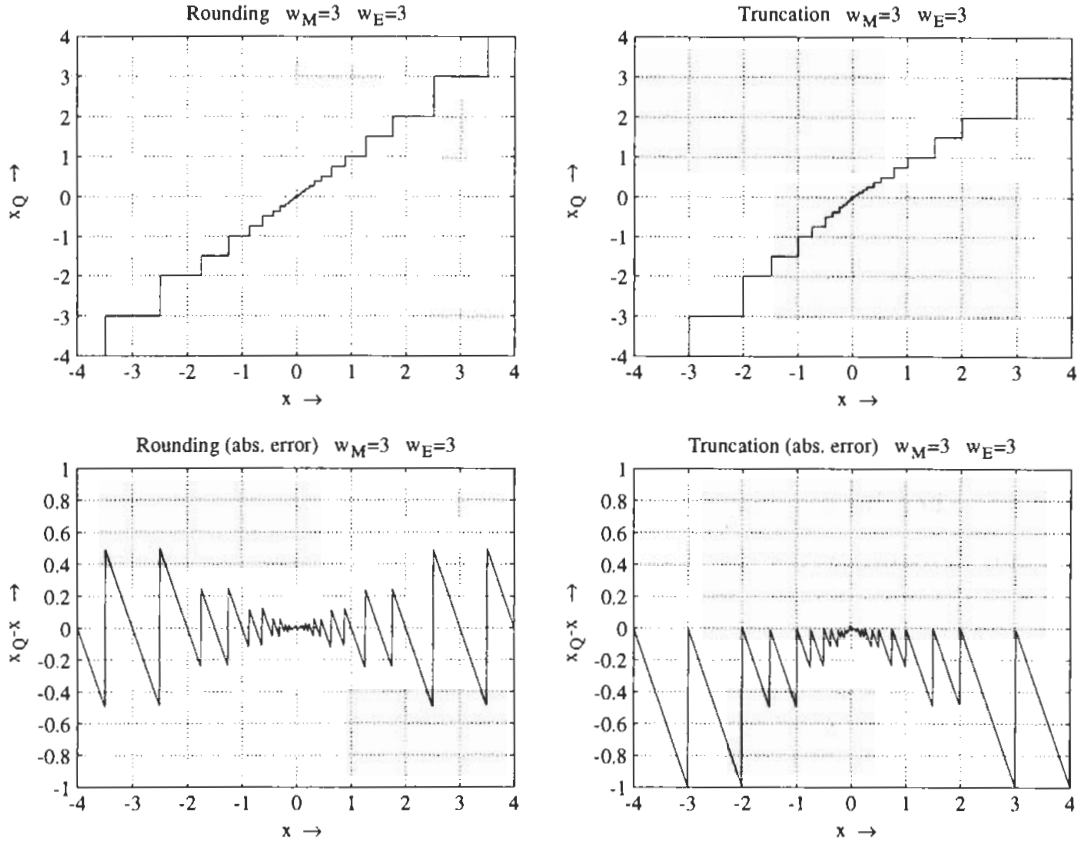


Figure 2.36 Rounding and truncation curves for floating-point representation.

For the relative error

$$e_r = \frac{x_Q - x}{x}, \tag{2.138}$$

of the floating-point representation, a constant upper limit can be stated as

$$|e_r| \leq 2^{-(w_M - 1)}. \tag{2.139}$$

Dynamic Range. With the maximum and minimum numbers given by

$$x_{Qmax} = (1 - 2^{-(w_M - 1)})2^{E_{Gmax}} \tag{2.140}$$

$$x_{Qmin} = 0.5\,2^{E_{Gmin}} \tag{2.141}$$

and

$$E_{Gmax} = 2^{w_E - 1} - 1 \tag{2.142}$$

$$E_{Gmin} = -2^{w_E - 1} + 2 \tag{2.143}$$

the dynamic range for floating-point representation is given by

$$\begin{aligned} \mathrm{DR}_G &= 20\log_{10}\left(\frac{(1-2^{-(w_M-1)})2^{E_{Gmax}}}{0.5\,2^{E_{Gmin}}}\right) \\ &= 20\log_{10}(1-2^{-(w_M-1)})2^{E_{Gmax}-E_{Gmin}+1} \\ &= 20\log_{10}(1-2^{-(w_M-1)})2^{2^{w_E}-2} \quad [\mathrm{dB}]. \end{aligned} \tag{2.144}$$

Multiplication and Addition of Floating-point Numbers. For multiplications with floating-point numbers, the exponents of both numbers $x_{Q1} = M_1 2^{E_1}$ and $x_{Q2} = M_2 2^{E_2}$ are added and the mantissas are multiplied. The resulting exponent $E_G = E_1 + E_2$ is adjusted so that $M_G = M_1 M_2$ lies in the interval $0.5 \leq M_G < 1$. For additions the smaller number is denormalized to get the same exponent. Then both mantissa are added and the result is normalized.

Error Model. With the definition of the relative error $e_r = \frac{x_Q - x}{x}$ the quantized signal can be written as

$$x_Q = x(1+e_r) = x + x \cdot e_r. \tag{2.145}$$

Floating-point quantization can be modeled as an additive error signal $e = x \cdot e_r$ to the signal x (see Fig. 2.37).

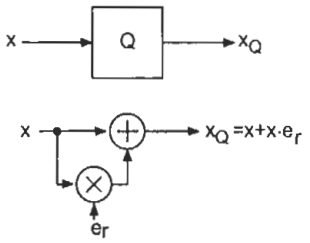


Figure 2.37 Model of a floating-point quantizer.

Signal-to-noise Ratio. Under the assumption that the relative error is independent of the input x, the noise power of the floating-point quantizer can be written as

$$\sigma_e^2 = \sigma_x^2 \cdot \sigma_{e_r}^2. \tag{2.146}$$

For the signal-to-noise-ratio, we can derive

$$\begin{aligned} \mathrm{SNR} &= 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) \\ &= 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_x^2 \cdot \sigma_{e_r}^2}\right) \\ &= 10\log_{10}\left(\frac{1}{\sigma_{e_r}^2}\right). \end{aligned} \tag{2.147}$$

Equation (2.147) shows that the signal-to-noise-ratio is independent of the level of the input. It is only dependent on the noise power $\sigma_{e_r}^2$ which, in turn, is only dependent on the word-length w_M of the mantissa of the floating-point representation.

2.4.3 Effects on Format Conversion and Algorithms

First, a comparison of signal-to-noise ratio is made for fixed-point and floating-point number representation. Figure 2.38 shows the signal-to-noise ratio as a function of input level for both number representations. The fixed-point word-length is $w = 16$ bits. The word-length of the mantissa in floating-point representation is also $w_M = 16$ bits whereas that of the exponent is $w_E = 4$ bits. The signal-to-

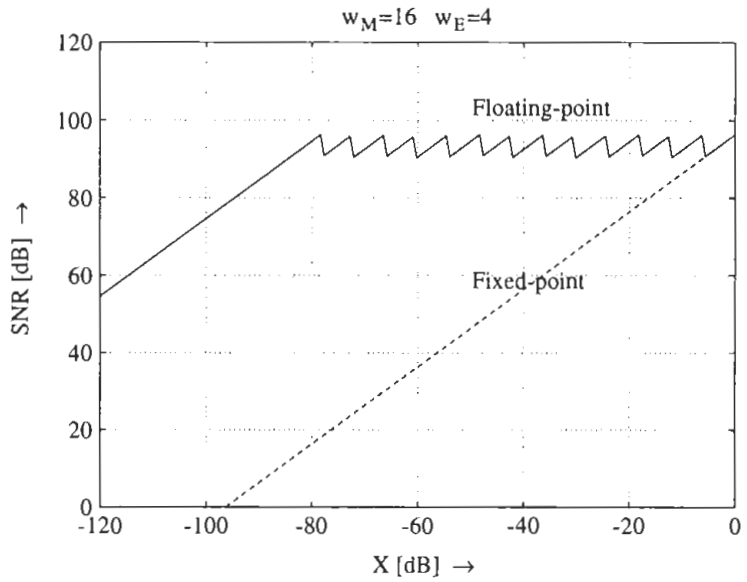


Figure 2.38 Signal-to-noise ratio for an input level.

noise ratio for floating-point representation shows that it is independent of input level and varies as a saw-tooth curve in a 6 dB grid. If the input level is so low that a normalization of the mantissa due to finite number representation is not possible, then the signal-to-noise ratio is comparable to fixed-point representation. While using the full range, both fixed-point and floating-point result in the same signal-to-noise ratio. It can be noticed that the signal-to-noise ratio for fixed-point representation depends on the input level. This signal-to-noise ratio in the digital domain is an exact image of the level-dependent signal-to-noise ratio of an analog signal in the analog domain. A floating-point representation cannot improve this

signal-to-noise ratio. Rather, the floating-point curve is vertically shifted downwards to the value of signal-to-noise ratio of an analog signal.

AD/DA Conversion. Before processing, storing and transmission of audio signals, the analog audio signal is converted into a digital signal. The precision of this conversion depends on the word-length w of the AD converter. The resulting signal-to-noise ratio is $6w$ dB for uniform PDF inputs. The signal-to-noise ratio in the analog domain depends on the level. This linear dependence of signal-to-noise ratio on level is preserved after AD conversion with subsequent fixed-point representation.

Digital Audio Formats. The basis for established digital audio transmission formats is provided in the previous section on AD/DA conversion. The digital two-channel AES/EBU-interface [AES92] and 56-channel MADI-interface [AES91] both operate with fixed-point representation with a word-length of, at the most, 24 bits per channel.

Storage and Transmission. Besides the established storage media like compact disc and DAT which were exclusively developed for audio application, there are storage systems like hard discs in computers. These are based on magnetic or magneto-optic principles. The systems operate with fixed-point number representation. With regard to the transmission of digital audio signals for band-limited transmission channels like satellite broadcasting (Digital Satellite Radio, DSR) or terrestrial broadcasting, it is necessary to reduce bit rates. For this, a conversion of a block of linearly coded samples is carried out in a so-called block floating-point representation in DSR. In the context of DAB, a data reduction of linear coded samples is carried out based on psychoacoustic criteria.

Equalizers. While implementing equalizers with recursive digital filters, the signal-to-noise ratio depends on the choice of the recursive filter structure. By a suitable choice of a filter structure and methods to spectrally shape the quantization errors, optimal signal-to-noise ratios are obtained for a given word-length. The signal-to-noise ratio for fixed-point representation depends on the word-length and for floating-point representation on the word-length of the mantissa. For filter implementations with fixed-point arithmetic, boost filters have to be implemented with a scaling within the filter algorithm. The properties of floating-point representation take care of automatic scaling in boost filters. If an insert I/O in fixed-point representation follows a boost filter in floating-point representation then the same scaling as in fixed-point arithmetics has to be done.

Dynamic Range Control. Dynamic range control is performed by a simple multiplicative weighting of the input signal with a control factor. The latter follows from calculating the peak and RMS value (root mean square) of the input signal.

The number representation of the signal has no influence on the properties of the algorithm. Owing to the normalized mantissa in floating-point representation some simplifications are produced while determining the control factor.

Mixing/Summation. While mixing signals to a stereo image, only multiplications and additions occur. Under the assumption of incoherent signals, an overload reserve can be estimated. This implies a reserve of 20/30 dB for 48/96 sources. For fixed-point representation the overload reserve is provided by a number of overflow bits in the accumulator of a DSP (Digital Signal Processor). The properties of automatic scaling in floating-point arithmetic provide for overload reserves. For both number representations, the summation signal must be matched with the number representation of the output. While dealing with AES/EBU outputs or MADI outputs, both number representations are adjusted to fixed-point format. Similarly, within heterogeneous system solutions, it is logical to make heterogeneous use of both number representations though corresponding number representations have to be converted.

Since the signal-to-noise ratio in fixed-point representation depends on the input level, a conversion from fixed-point to floating-point representation does not lead to a change of signal-to-noise ratio, i.e. the conversion does not improve the signal-to-noise ratio. Further signal processing with floating-point or fixed-point arithmetic does not alter the signal-to-noise ratio as long as the algorithms are chosen and programmed accordingly. Reconversion from floating-point to fixed-point representation again leads to a level-dependent signal-to-noise ratio.

As a consequence, for two-channel DSP systems which operate with AES/EBU or with analog inputs and outputs, and which are used for equalization, dynamic range control, room simulation etc., the above-mentioned holds. These conclusions are also valid for digital mixing consoles for which digital inputs from AD converters or from multitrack machines are represented in fixed-point format (AES/EBU or MADI). The number representation for inserts and auxiliaries is specific to a system. Digital AES/EBU (or MADI) inputs and outputs are realized in fixed-point number representation.

Chapter 3

AD/DA Conversion

The conversion of a continuous-time function $x(t)$ (voltage, current) into a sequence of numbers $x(n)$ is called analog-to-digital conversion (AD conversion). The reverse process is known as digital-to-analog conversion (DA conversion). The time-sampling of a function $x(t)$ is described by *Shannon's Sampling Theorem.* It states that a continuous-time signal with bandwidth f_B can be sampled with a sampling rate $f_S > 2f_B$ without changing the content of information in the signal. The original analog signal is reconstructed by low-pass filtering with bandwidth f_B. Besides time-sampling, the nonlinear procedure of digitizing the continuous-valued amplitude (quantization) of the sampled signal occurs. In the first section, basic concepts of Nyquist sampling, oversampling and delta-sigma modulation are presented. In the second and third sections, principles of AD and DA converter circuits are discussed.

3.1 Methods

3.1.1 Nyquist Sampling

The sampling of a signal with sampling rate $f_S > 2f_B$ is called Nyquist sampling. The schematic diagram in Fig. 3.1 shows the procedure. The band-limiting of the input at $f_S/2$ is carried out by an analog low-pass filter (Fig. 3.1a). The following sample-and-hold circuit samples the band-limited input at a sampling rate f_S. The constant amplitude of the time function over the sampling period $T_S = 1/f_S$ is converted to a number sequence $x(n)$ by a quantizer (Fig. 3.1b). This number

sequence is fed to a digital signal processor (DSP) which performs signal processing algorithms. The output sequence $y(n)$ is delivered to a DA converter which gives a staircase as its output (Fig.3.1c). Following this, a low-pass filter gives the analog output $y(t)$ (Fig. 3.1d). Figure 3.2 demonstrates each step of AD/DA conversion

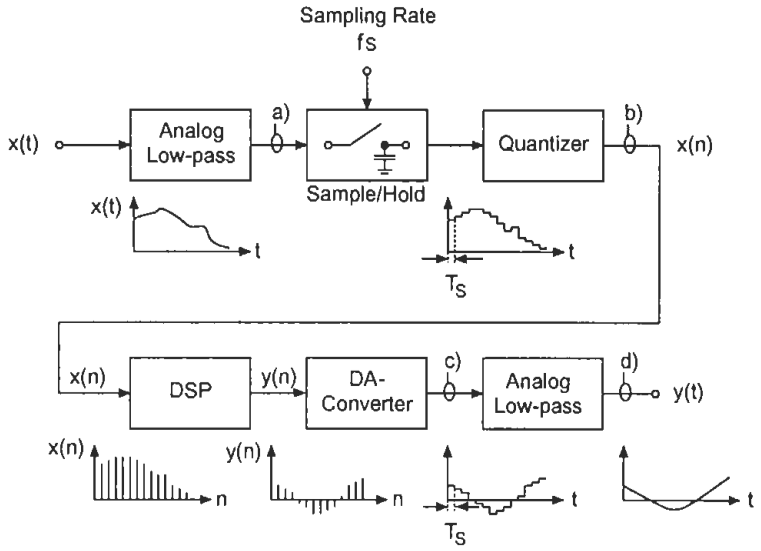


Figure 3.1 Schematic diagram of Nyquist sampling.

in the frequency domain. The individual spectra in Fig. 3.2a...d correspond to the outputs a...d in Fig. 3.1.

After band-limiting (Fig. 3.2a) and sampling, a periodic spectrum with period f_S of the sampled signal is obtained as in Fig. 3.2b. Assuming that consecutive quantization errors $e(n)$ are statistically independent of each other, the noise power has a spectral uniform distribution in the frequency domain $0 \leq f \leq f_S$. The output of the DA converter still has a periodic spectrum. However, this is weighted with the sinc-function (sinc=$\frac{\sin(x)}{x}$, [Fli91, Gab87] of the sample-and-hold circuit (Fig. 3.2c). The zeros of the sinc-function are at multiples of the sampling rate f_S. In order to reconstruct the output (Fig. 3.2d), the image spectra are eliminated by an analog low-pass of sufficient stop-band attenuation.

The problems of Nyquist sampling lie in the steep band-limiting filter characteristic (anti-aliasing filter) of the analog input filter and the analog reconstruction filter (anti-imaging filter) of similar filter characteristic and sufficient stop-band attenuation. Further, sinc-distortion due to the sample-and-hold circuit needs to be compensated for.

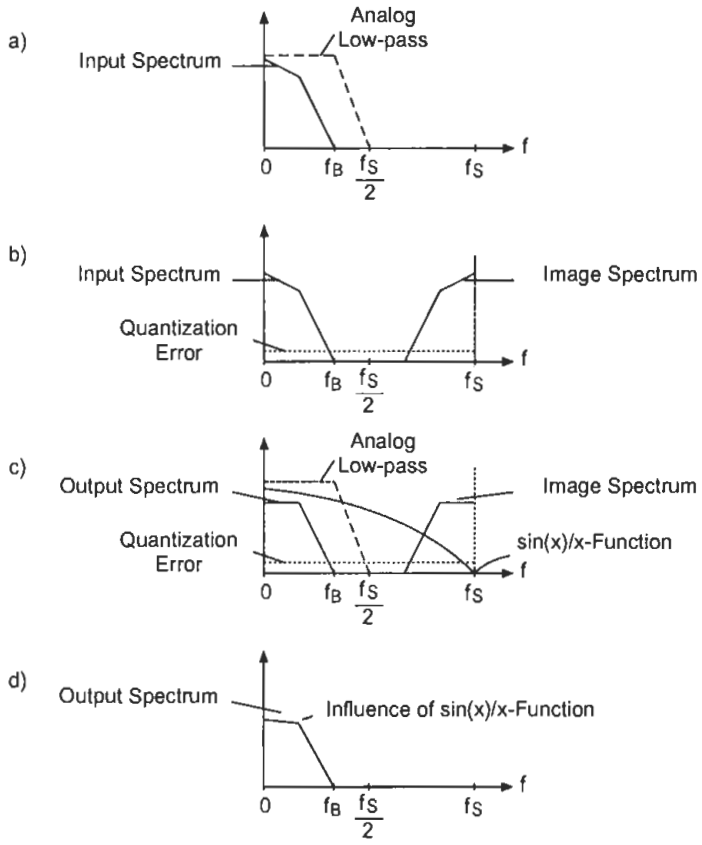


Figure 3.2 Nyquist sampling - interpretation in the frequency domain.

3.1.2 Oversampling

In order to increase the resolution of the conversion process and reduce the complexity of the analog filters, oversampling techniques are employed. Owing to the spectral uniform distribution of quantization error between 0 and f_S (see Fig. 3.3a), it is possible to reduce the power spectral density in the pass-band through oversampling by a factor L, i.e. with the new sampling rate Lf_S (see Fig. 3.3b). For identical quantization step size Q, the shaded areas (quantization error power σ_E^2) in Fig. 3.3a and Fig. 3.3b are equal. The increase in the signal-to-noise ratio is also noticed from Fig. 3.3.

It follows that in the pass-band at a sampling rate of $f_S = 2f_B$ the power spectral density given by

$$S_{ee}(f) = \frac{Q^2}{12f_S} \tag{3.1}$$

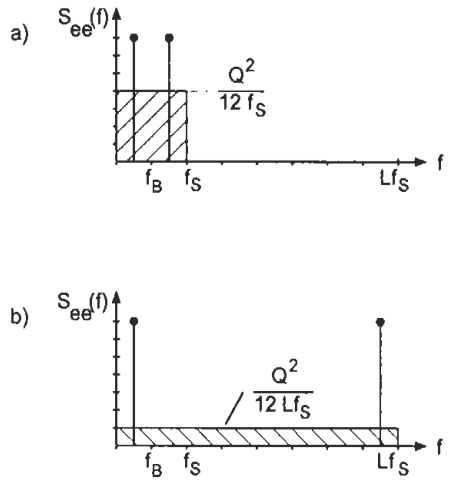


Figure 3.3 Influence of oversampling on power spectral density of quantization error.

leads to the noise power

$$N_B^2 = \sigma_E^2 = 2\int_0^{f_B} S_{ee}(f)df = \frac{Q^2}{12}. \tag{3.2}$$

Owing to oversampling by a factor of L, a reduction of the power spectral density given by

$$S_{ee}(f) = \frac{Q^2}{12Lf_S} \tag{3.3}$$

is obtained (see Fig. 3.3). The signal-to-noise ratio (with $P_F = \sqrt{3}$) owing to oversampling can be expressed as

$$\text{SNR} = 6.02 \cdot w + 10\log_{10}(L) \qquad \text{[dB]}. \tag{3.4}$$

Figure 3.4a shows a schematic diagram of an oversampling AD converter. Owing to oversampling, the analog band-limiting low-pass filter can have a wider transition bandwidth as shown in Fig. 3.4b. The quantization error power is distributed between 0 and the sampling rate Lf_S. To reduce the sampling rate, it is necessary to limit the bandwidth with a digital low-pass filter (see Fig. 3.4c). After this, the sampling rate is reduced by a factor L (see Fig. 3.4d) by taking every Lth output sample of the digital low-pass filter [Vai93, Fli94].

Figure 3.5a shows the schematic diagram of an oversampling DA converter. The sampling rate is first increased by a factor of L. For this, $L-1$ zeros are introduced between two consecutive input values [Vai93, Fli94]. The following digital filter

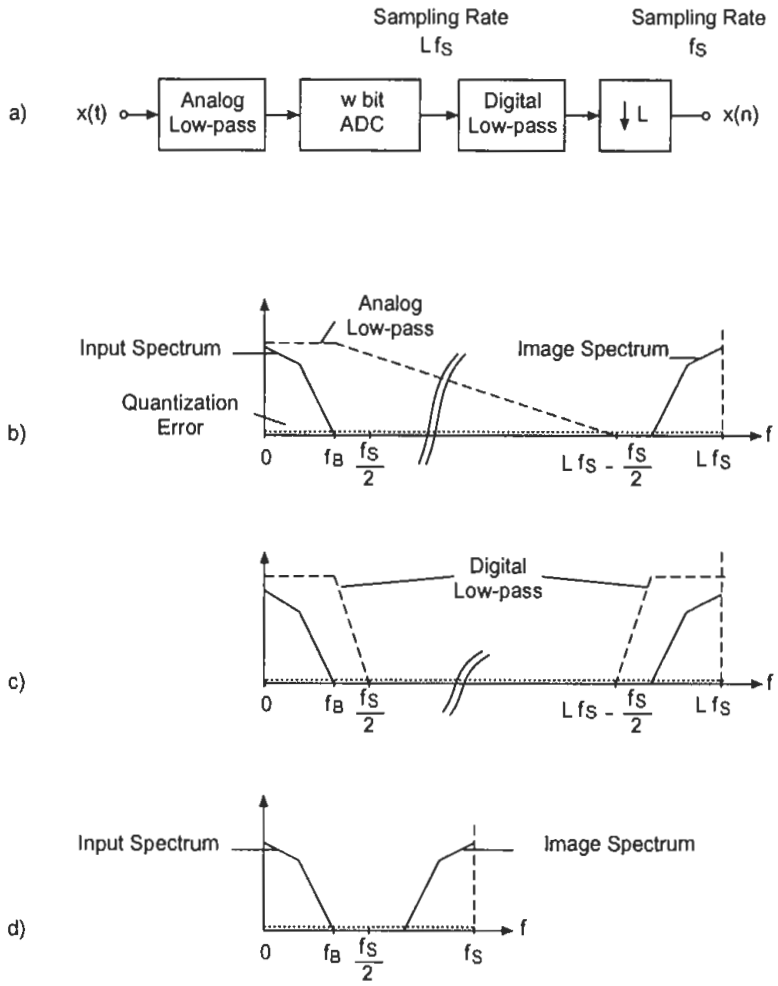


Figure 3.4 Oversampling AD converter and sampling rate reduction.

eliminates all image spectra (Fig. 3.5b) except the baseband spectrum and spectra at multiples of Lf_S (Fig. 3.5c). It interpolates $L-1$ samples between two input samples. The w bit DA converter operates at a sampling rate Lf_S. Its output is fed to an analog reconstruction filter which eliminates the image spectra at multiples of Lf_S.

3.1.3 Delta-sigma Modulation

Delta-sigma modulation using oversampling is a conversion strategy derived from delta modulation. In delta modulation (Fig. 3.6a), the difference between the input $x(t)$ and signal $x_1(t)$ is converted into a 1 bit signal $y(n)$ at a very high sampling rate Lf_S. The sampling rate is higher than the necessary Nyquist rate f_S. The quantized signal $y(n)$ gives the signal $x_1(t)$ via an analog integrator. The

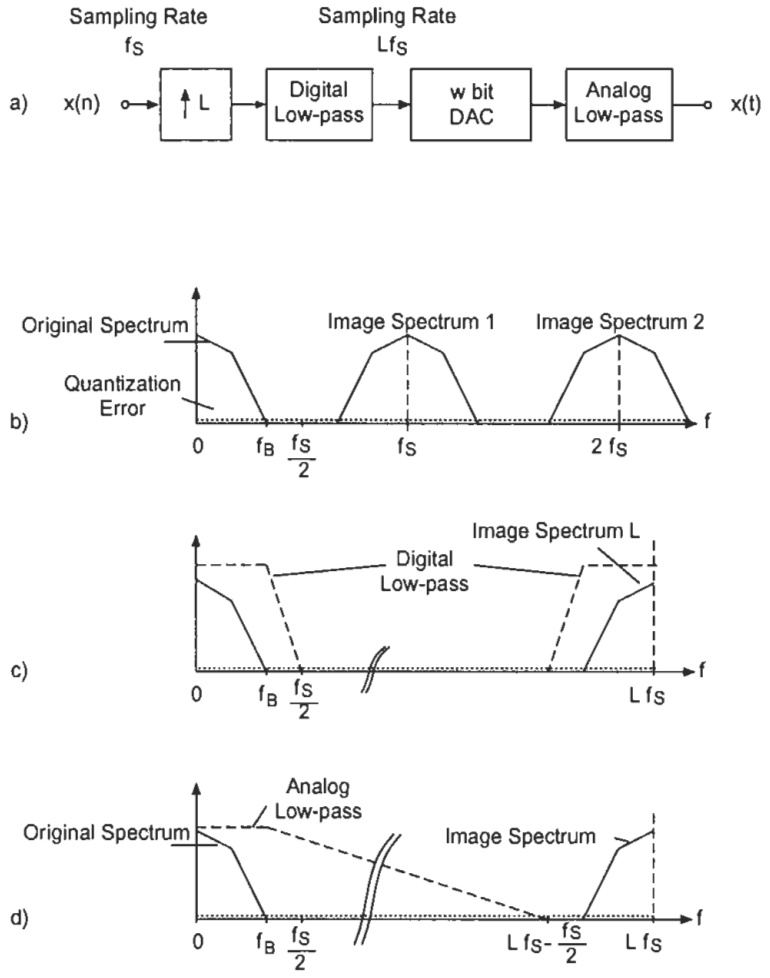


Figure 3.5 Oversampling and DA conversion.

demodulator consists of an integrator and a reconstruction low-pass filter. The corresponding signals are shown in Fig. 3.7.

The extension to delta-sigma modulation [Ino63] consists of shifting the integrator from the demodulator to the input of the modulator (see Fig. 3.6b). With this, it is possible to combine the two integrators as a single integrator after addition (see Fig. 3.8a). The corresponding signals are shown in Fig. 3.9.

A time-discrete model of the delta-sigma modulator is given in Fig. 3.8b. The Z-transform of the output signal $y(n)$ is given by

$$\begin{aligned} Y(z) &= \frac{H(z)}{1+H(z)}X(z) + \frac{1}{1+H(z)}E(z) \\ &\approx X(z) + \frac{1}{1+H(z)}E(z). \end{aligned} \tag{3.5}$$

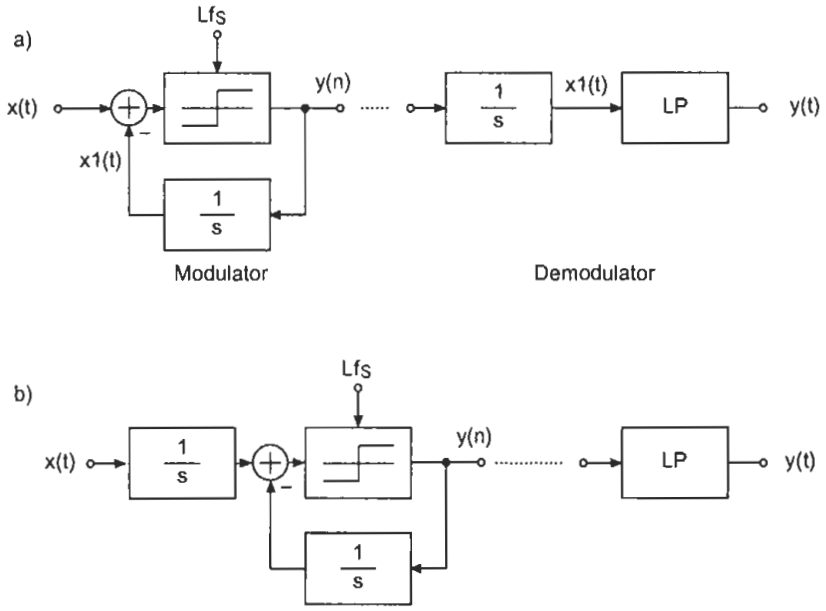


Figure 3.6 Delta modulation and displacement of integrator.

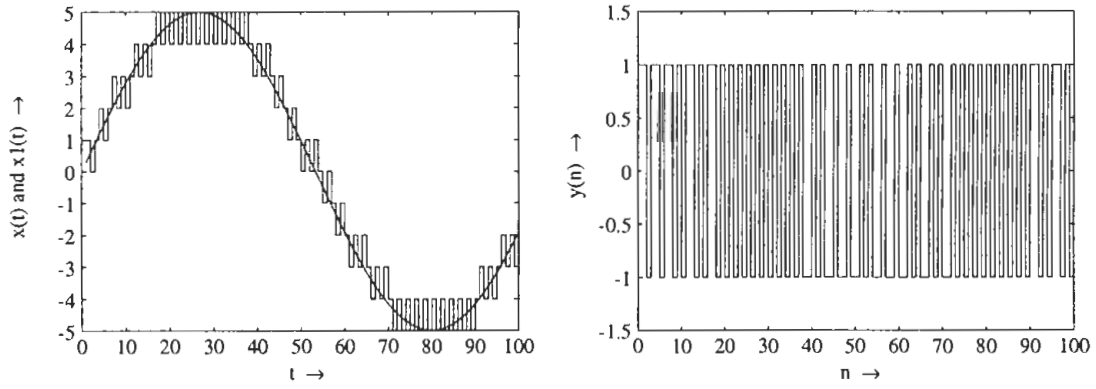


Figure 3.7 Signals in delta modulation.

For a large gain factor of the system $H(z)$, the input signal will not be affected. In contrast, the quantization error is shaped by the filter term $1/[(1 + H(z)]$.

The schematic diagrams of delta-sigma AD/DA conversion are shown in Figs. 3.10 and 3.11. For delta-sigma AD converters, a digital low-pass filter and a down-sampler with factor L are used to reduce the sampling rate Lf_S down to f_S. The 1 bit input to the digital low-pass filter leads to a w bit output $x(n)$ at a sampling rate f_S. The delta-sigma DA converter consists of an upsampler with factor L, a digital low-pass filter to eliminate the mirror spectra and a delta-sigma modulator followed by an analog reconstruction low-pass filter. In order to illustrate noise

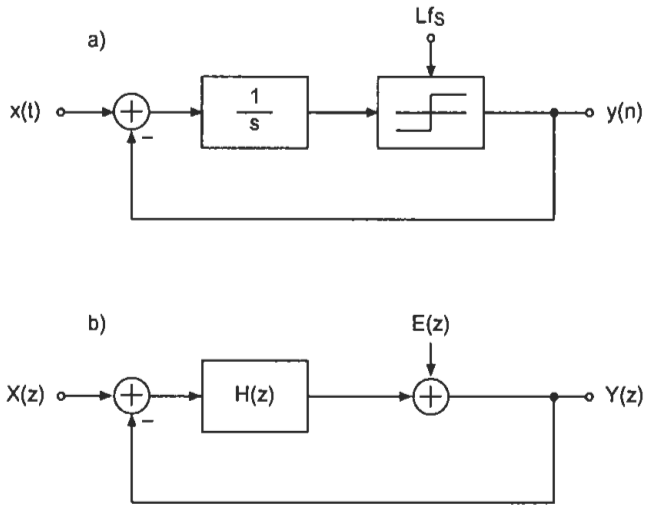


Figure 3.8 Delta-sigma modulation and time-discrete model.

shaping in delta-sigma modulation in detail, first- and second-order systems as well as multistage techniques are investigated in the following sections.

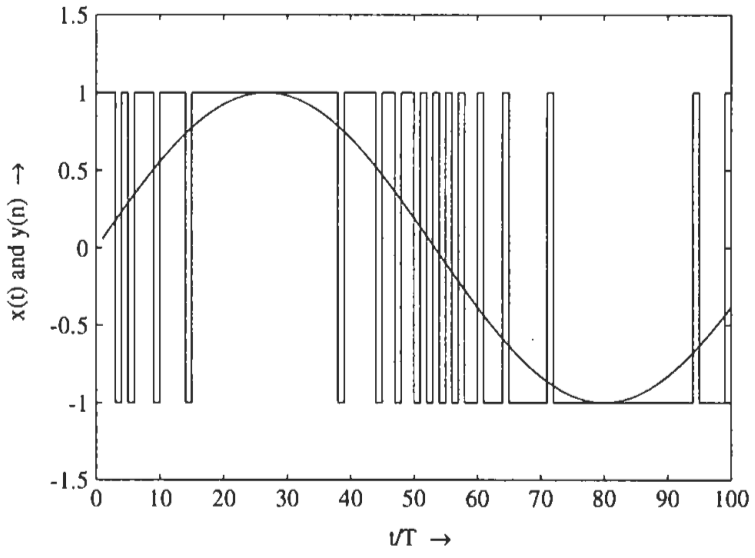


Figure 3.9 Signals in delta-sigma modulation.

First-order Delta-sigma Modulator

A time-discrete model of a first-order delta-sigma modulator is shown in Fig. 3.12. The difference equation for the output $y(n)$ is given by

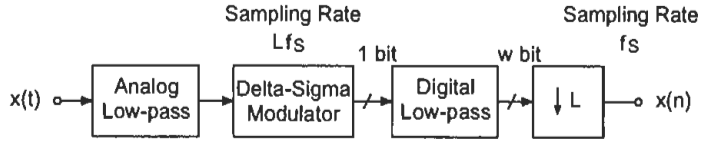


Figure 3.10 Oversampling delta-sigma AD converter.

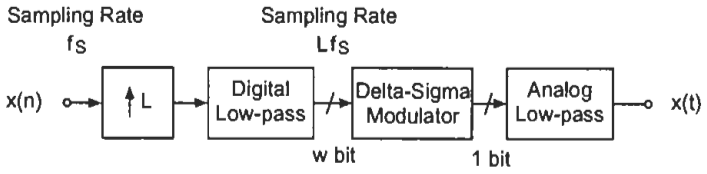


Figure 3.11 Oversampling delta-sigma DA converter.

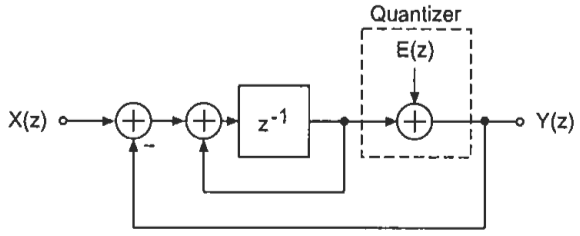


Figure 3.12 Time-discrete model of a first-order delta-sigma modulator.

$$y(n) = x(n-1) + e(n) - e(n-1). \tag{3.6}$$

The corresponding Z-transform leads to

$$Y(z) = z^{-1}X(z) + E(z)\underbrace{(1-z^{-1})}_{H_E(z)}. \tag{3.7}$$

The power density spectrum of the error signal $e_1(n) = e(n) - e(n-1)$ is

$$\begin{aligned} S_{e_1e_1}(e^{j\Omega}) &= S_{ee}(e^{j\Omega})\left|1-e^{-j\Omega}\right|^2 \\ &= S_{ee}(e^{j\Omega})4\sin^2(\frac{\Omega}{2}), \end{aligned} \tag{3.8}$$

where $S_{ee}(e^{j\Omega})$ denotes the power density spectrum of the quantization error $e(n)$. The error power in the frequency band $[-f_B, f_B]$, with $S_{ee}(f) = \frac{Q^2}{12Lf_S}$, can be written as

$$N_B^2 = S_{ee}(f)2\int_0^{f_B} 4\sin^2(\pi\frac{f}{Lf_S})df \tag{3.9}$$

$$\simeq \frac{Q^2}{12}\frac{\pi^2}{3}\left(\frac{2f_B}{Lf_S}\right)^3. \tag{3.10}$$

With $f_S = 2f_B$ we get

$$N_B^2 = \frac{Q^2}{12}\frac{\pi^2}{3}\left(\frac{1}{L}\right)^3. \tag{3.11}$$

Second-order Delta-sigma Modulator

For the second-order delta-sigma modulator [Can85], shown in Fig. 3.13, the

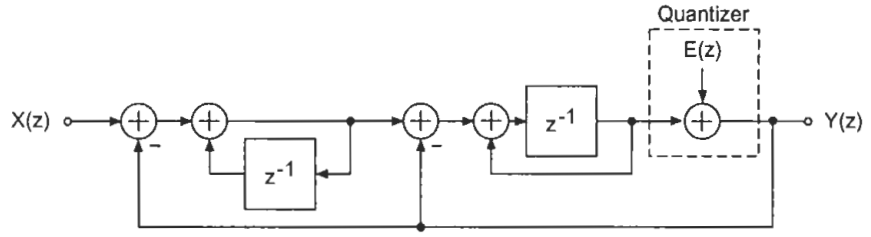


Figure 3.13 Time-discrete model of a second-order delta-sigma modulator.

difference equation is expressed as

$$y(n) = x(n-1) + e(n) - 2e(n-1) + e(n-2) \tag{3.12}$$

and the Z-transform is given by

$$Y(z) = z^{-1}X(z) + E(z)\underbrace{(1 - 2z^{-1} + z^{-2})}_{H_E(z)=(1-z^{-1})^2}. \tag{3.13}$$

The power density spectrum of the error signal $e_1(n) = e(n) - 2e(n-1) + e(n-2)$ can be written as

$$\begin{aligned} S_{e_1e_1}(e^{j\Omega}) &= S_{ee}(e^{j\Omega})\left|1 - e^{-j\Omega}\right|^4 \\ &= S_{ee}(e^{j\Omega})[4\sin^2(\frac{\Omega}{2})]^2 \\ &= S_{ee}(e^{j\Omega})4[1 - \cos(\Omega)]^2. \end{aligned} \tag{3.14}$$

The error power in the frequency band $[-f_B, f_B]$ is given by

$$N_B^2 = S_{ee}(f)2\int_0^{f_B} 4[1 - \cos(\Omega)]^2 df \tag{3.15}$$

$$\simeq \frac{Q^2}{12}\frac{\pi^4}{5}\left(\frac{2f_B}{Lf_S}\right)^5 \tag{3.16}$$

and with $f_S = 2f_B$ we obtain

$$N_B^2 = \frac{Q^2}{12}\frac{\pi^4}{5}\left(\frac{1}{L}\right)^5. \tag{3.17}$$

Multistage Delta-sigma Modulator

A multistage delta-sigma modulator (MASH, [Mat87]) is shown in Fig. 3.14.

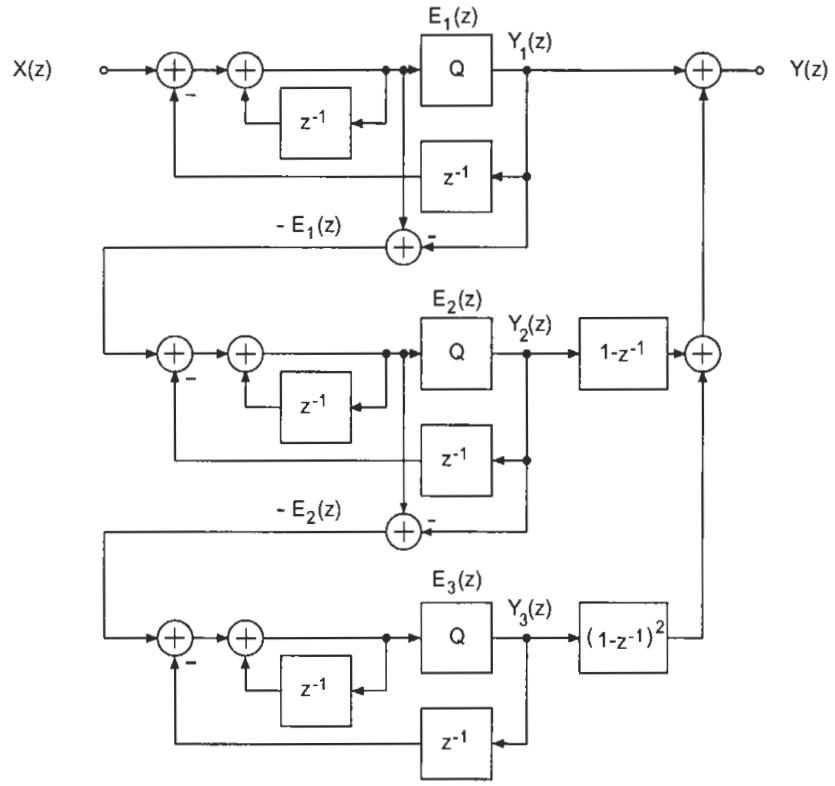


Figure 3.14 Time-discrete model of a multistage delta-sigma modulator.

The Z-transforms of the output signal $y_{1..3}(n)$ are given by

$$Y_1(z) = X(z) + (1 - z^{-1})E_1(z) \tag{3.18}$$
$$Y_2(z) = -E_1(z) + (1 - z^{-1})E_2(z) \tag{3.19}$$
$$Y_3(z) = -E_2(z) + (1 - z^{-1})E_3(z). \tag{3.20}$$

The Z-transform of the output results by addition and filtering in

$$\begin{aligned} Y(z) &= Y_1(z) + (1 - z^{-1})Y_2(z) + (1 - z^{-1})^2 Y_3(z) \\ &= X(z) + (1 - z^{-1})E_1(z) - (1 - z^{-1})E_1(z) \\ &\quad +(1 - z^{-1})^2 E_2(z) - (1 - z^{-1})^2 E_2(z) + (1 - z^{-1})^3 E_3(z) \\ &= X(z) + \underbrace{(1 - z^{-1})^3}_{H_E(z)} E_3(z). \end{aligned} \tag{3.21}$$

The error power in the frequency band $[-f_B, f_B]$

$$N_B^2 = \frac{Q^2}{12}\frac{\pi^6}{7}\left(\frac{2f_B}{Lf_S}\right)^7 \tag{3.22}$$

with $f_S = 2f_B$ gives the following total noise power

$$N_B^2 = \frac{Q^2}{12}\frac{\pi^6}{7}\left(\frac{1}{L}\right)^7. \tag{3.23}$$

The error transfer functions in Fig. 3.15 show the noise shaping for three types of delta-sigma modulations as discussed before. The error power is shifted towards higher frequencies.

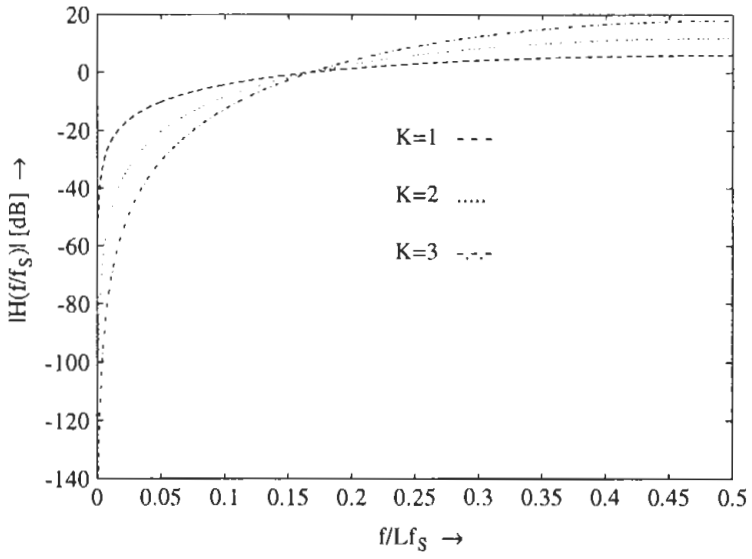


Figure 3.15 $H_E(z) = (1 - z^{-1})^K$ with $K = 1, 2, 3$.

The improvement of signal-to-noise ratio by pure oversampling and delta-sigma modulation (first-, second- and third-order) is shown in Fig. 3.16.

Higher-order Delta-sigma Modulator

A widening of the stop-band for the high-pass transfer function of the quantization error is achieved with higher-order delta-sigma modulation [Cha90]. Besides the zeros at $z = 1$, additional zeros are placed on the unit circle. Also, poles are integrated into the transfer function. A time-discrete model of a higher-order delta-sigma modulator is shown in Fig. 3.17.

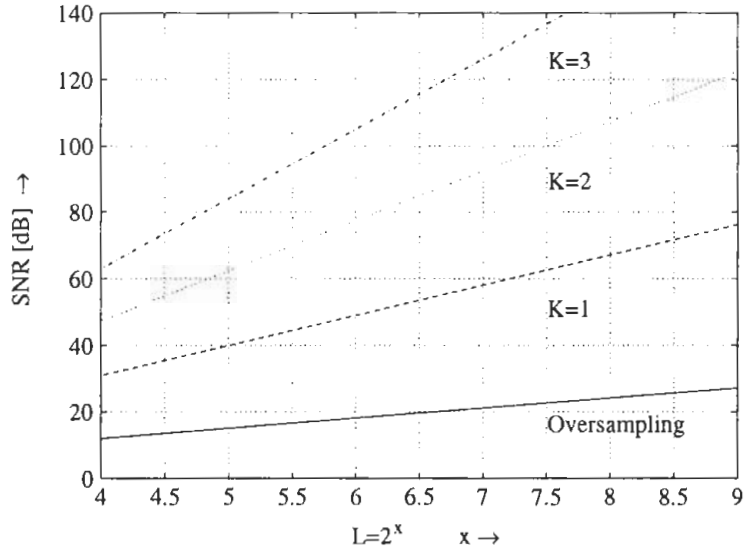


Figure 3.16 Improvement of signal-to-noise ratio as a function of oversampling and noise shaping ($L = 2^x$).

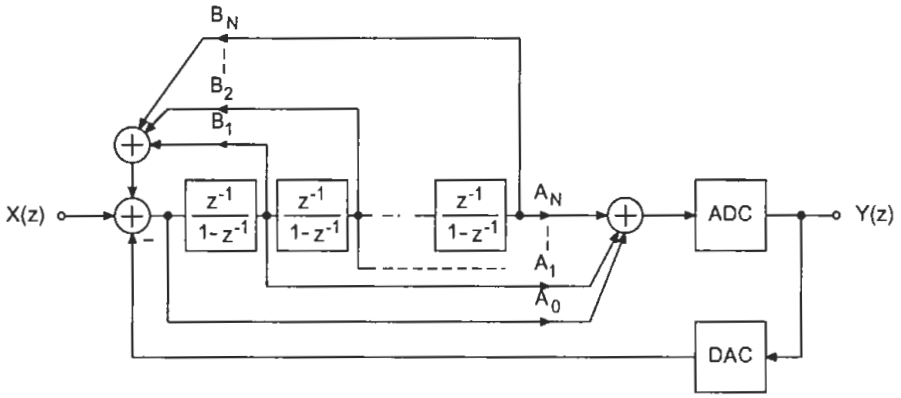


Figure 3.17 Higher-order delta-sigma modulator.

The transfer function in Fig. 3.17 can be written as

$$\begin{aligned} H(z) &= \frac{A_0 + A_1 \frac{z^{-1}}{1-z^{-1}} + A_2 \left(\frac{z^{-1}}{1-z^{-1}}\right)^2 + \ldots}{1 - B_1 \frac{z^{-1}}{1-z^{-1}} - B_2 \left(\frac{z^{-1}}{1-z^{-1}}\right)^2 + \ldots} \\ &= \frac{A_0(z-1)^N + A_1(z-1)^{N-1} + \ldots + A_N}{(z-1)^N - B_1(z-1)^{N-1} - \ldots - B_N} \\ &= \frac{\sum_{i=0}^{N} A_i (z-1)^{N-i}}{(z-1)^N - \sum_{i=1}^{N} B_i (z-1)^{N-i}}. \end{aligned} \tag{3.24}$$

The Z-transform of the output is given by

$$Y(z) = \frac{H(z)}{1+H(z)}X(z) + \frac{1}{1+H(z)}E(z) \tag{3.25}$$

$$= H_X(z)X(z) + H_E(Z)E(z). \tag{3.26}$$

The transfer function for the input is

$$H_X(z) = \frac{\sum_{i=0}^{N} A_i(z-1)^{N-i}}{(z-1)^N - \sum_{i=1}^{N} B_i(z-1)^{N-i} + \sum_{i=0}^{N} A_i(z-1)^{N-i}}, \tag{3.27}$$

and the transfer function for the error signal is given by

$$H_E(z) = \frac{(z-1)^N - \sum_{i=1}^{N} B_i(z-1)^{N-i}}{(z-1)^N - \sum_{i=1}^{N} B_i(z-1)^{N-i} + \sum_{i=0}^{N} A_i(z-1)^{N-i}}. \tag{3.28}$$

For Butterworth or Chebychev filter designs, the frequency responses as shown in 3.18 are obtained for the error transfer function. As a comparison, the frequency responses of first-, second- and third-order delta-sigma modulation are shown. The widening of the stop-band for Butterworth and Chebychev filters can be observed from Fig. 3.19.

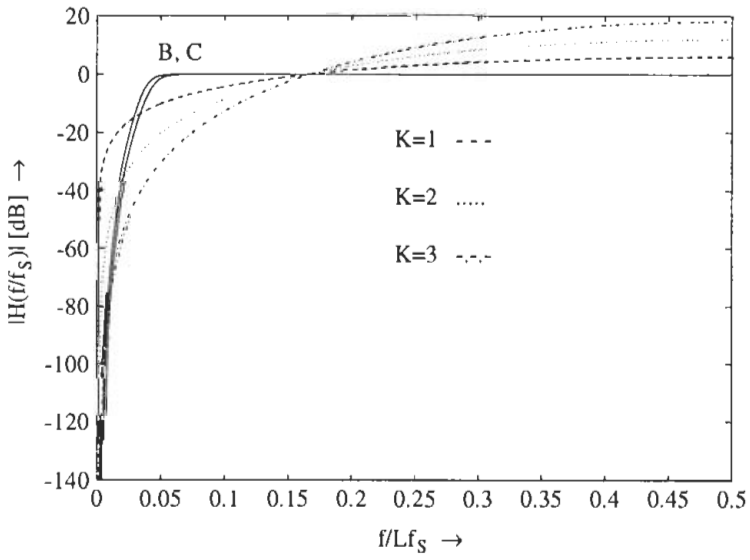


Figure 3.18 Comparison of different transfer functions of error signal.

Decimation Filter

The implementation of decimation filters for AD conversion and interpolation filters for DA conversion are performed with multirate systems [Fli94]. The necessary

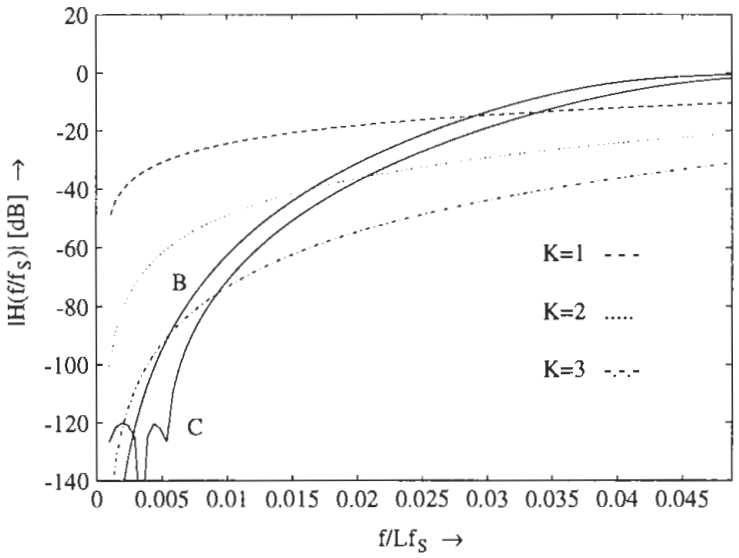


Figure 3.19 Transfer function of the error signal in stop-band.

downsampler and upsampler are simple systems. For the former, every nth sample is taken out of the input sequence. For the latter, $(n-1)$ zeros are inserted between two input samples. For decimation, band-limiting is performed by $H(z)$ followed by sampling rate reduction by a factor L. This procedure can be implemented in stages (see Fig. 3.20). The use of easy-to-implement filters structures at high

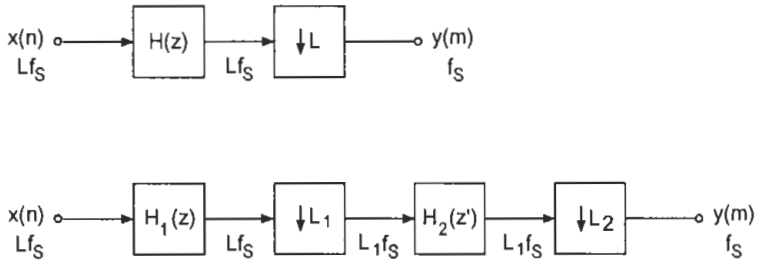


Figure 3.20 Several stages for sampling rate reduction.

sampling rates, like comb filters with transfer function

$$H_1(z) = \frac{1}{L}\frac{1-z^{-L}}{1-z^{-1}} \tag{3.29}$$

(shown in Fig. 3.21), allows simple implementation needing only delay systems and additions. In order to increase the stop-band attenuation, a series of comb

filters is used so that

$$H_1^M(z) = \left[\frac{1}{L}\frac{1-z^{-L}}{1-z^{-1}}\right]^M \tag{3.30}$$

is obtained.

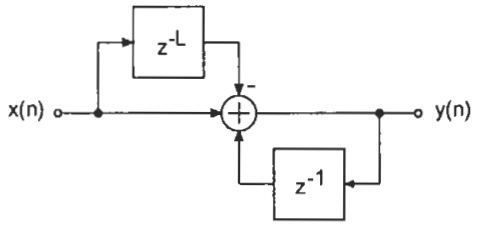


Figure 3.21 Signal flow diagram of a comb filter.

Besides additions at high sampling rates, complexity can be reduced further. Owing to sampling rate reduction by a factor of L, the numerator $(1 - z^{-L})$ can be moved so that it is placed after the downsampler (see Fig. 3.22). For a series of comb filters, the structure in Fig. 3.23 results. M simple recursive accumulators have to be performed at the high sampling rate Lf_S. After this, downsampling by a factor L is carried out. The M nonrecursive systems are calculated with the output sampling rate f_S.

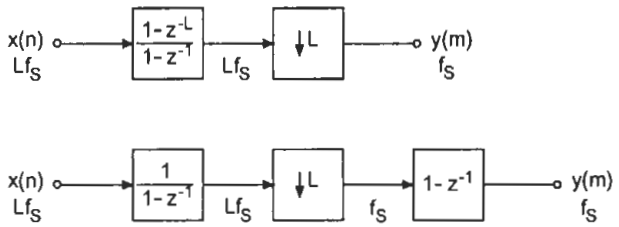


Figure 3.22 Comb filter for sampling rate reduction.

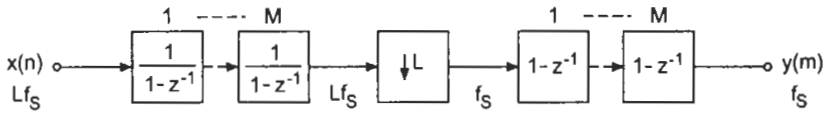


Figure 3.23 Series of comb filters for sampling rate reduction.

Figure 3.24a shows the frequency responses of a series of comb filters ($L = 16$). Figure 3.24b shows the resulting frequency response for the quantization error of a third-order delta-sigma modulator connected in series with a comb filter $H_1^4(z)$. The system delay owing to filtering and sampling rate reduction is given by

$$t_D = \frac{N-1}{2} \cdot \frac{1}{Lf_S}. \tag{3.31}$$

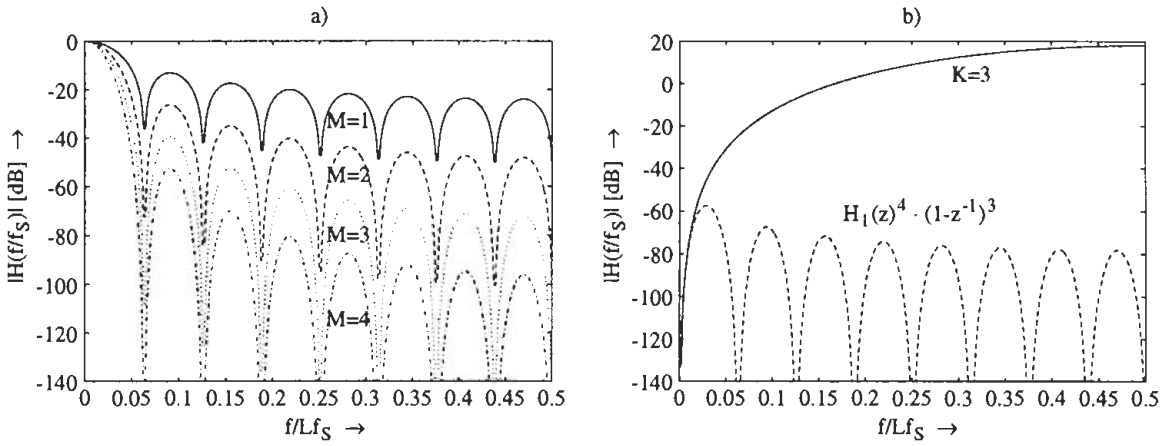


Figure 3.24 a) Transfer function $H_1^M(z) = \left[\frac{1}{16}\frac{1-z^{-16}}{1-z^{-1}}\right]^M$ with $M = 1 \ldots 4$. b) Third-order delta-sigma modulation and in series with $H_1^4(z)$.

Example: Delay time of conversion process (latency time)

1. Nyquist conversion

$$\begin{aligned} f_S &= 48 \quad \text{kHz} \\ t_D &= \frac{1}{f_S} = 20.83 \quad \mu s \end{aligned}$$

2. Delta-sigma modulation with single-stage downsampling

$$\begin{aligned} L &= 64 \\ f_S &= 48 \quad \text{kHz} \\ N &= 4096 \\ t_D &= 665 \quad \mu s \end{aligned}$$

3. Delta-sigma modulation with two-stage downsampling

$$\begin{aligned} L &= 64 \\ f_S &= 48 \quad \text{kHz} \\ L_1 &= 16 \\ L_2 &= 4 \\ N_1 &= 61 \\ N_2 &= 255 \\ t_{D_1} &= 9.76 \quad \mu s \\ t_{D_2} &= 662 \quad \mu s \end{aligned}$$

3.2 AD Converters

The choice of an AD converter for a certain application is influenced by a number of factors. It mainly depends on the necessary resolution for a given conversion time. Both of these depend upon each other and are decisively influenced by the architecture of the AD converter. For this reason, the specifications of an AD converter are first discussed. This is followed by circuit principles which influence the mutual dependence of resolution and conversion time.

3.2.1 Specifications

In the following, the most important specifications for AD conversion are presented.

Resolution. The resolution for a given word-length w of an AD converter determines the smallest amplitude

$$x_{min} = Q = x_{max}\ 2^{-(w-1)}, \tag{3.32}$$

which is equal to the quantization step Q.

Conversion Time. The minimum sampling period $T_S = 1/f_S$ between two samples is called conversion time.

Sample-and-hold Circuit. Before quantization, the time-continuous function is sampled with the help of a sample-and-hold circuit, as shown in Fig. 3.25. The sampling period T_S is divided into sampling time t_S in which the output voltage U_2 follows the input voltage U_1, and hold-time t_H. During the hold-time the output voltage U_2 is constant and is converted into a binary word by quantization.

Aperture Delay. The time t_{AD} elapsed between start of hold and actual hold mode (see Fig. 3.25) is called aperture delay.

Aperture Jitter. The variation of aperture delay from sample to sample is called aperture jitter t_{ADJ}. The influence of aperture jitter limits the useful bandwidth of the sampled signal. This is because at high frequency a deterioration of the signal-to-noise ratio occurs. Assuming a Gaussian PDF aperture jitter, the signal-to-noise ratio owing to aperture jitter as a function of frequency f can be written as

$$\mathrm{SNR}_J = -20\log_{10}(2\pi f t_{ADJ}) \qquad [\mathrm{dB}]. \tag{3.33}$$

Offset Error and Gain Error. The offset and gain errors of an AD converter are shown in Fig. 3.26. The offset error results in a horizontal displacement of the

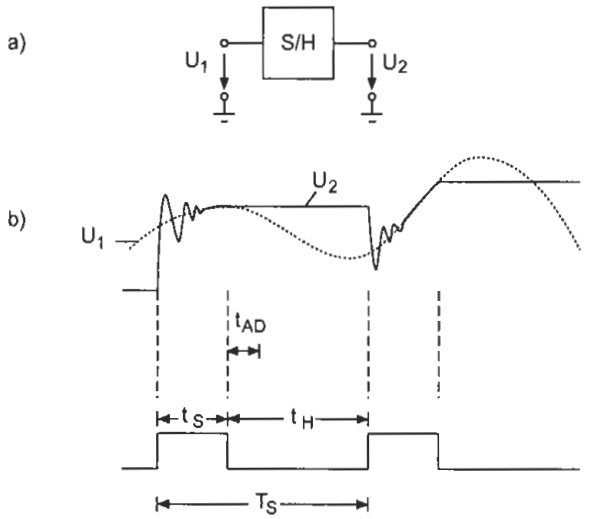


Figure 3.25 a) Sample-and-hold circuit. b) Input and output with clock signal. (t_S=sampling time, t_H=hold-time, t_{AD}=aperture delay).

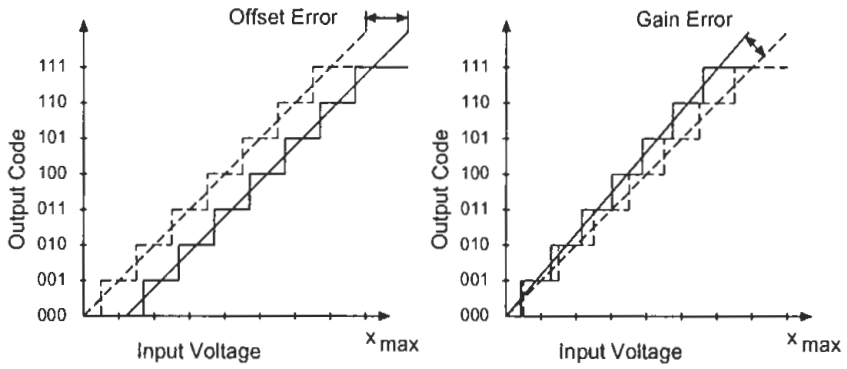


Figure 3.26 Offset error and gain error.

real curve compared with the dashed ideal curve of an AD converter. The gain error is expressed as the deviation from the ideal gradient of the curve.

Differential Nonlinearity. The differential nonlinearity

$$\mathrm{DNL} = \frac{\Delta x/Q}{\Delta x_Q} - 1 \qquad [\mathrm{LSB}] \tag{3.34}$$

describes the error of the step size of a certain code word in LSB units. For ideal quantization, the increase Δx of the input voltage up to the next output code x_Q is equal to the quantization step Q (see Fig. 3.27). The difference of two consecutive output codes is denoted as Δx_Q. When the output code changes from 010 to 011, the step size is 1.5 LSB and therefore the differential nonlinearity DNL=0.5 LSB. The step size between the codes 011 and 101 is 0 LSB and the code 200 is missing.

The differential nonlinearity is DNL=-1 LSB.

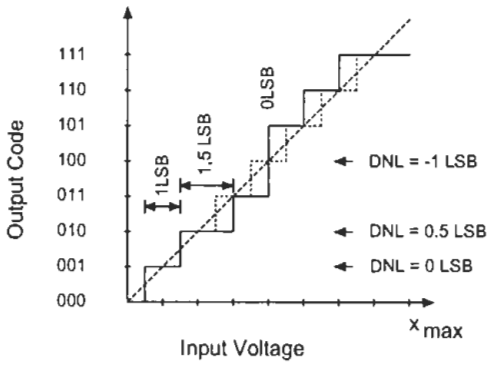


Figure 3.27 Differential Nonlinearity.

Integral Nonlinearity. The integral nonlinearity (INL) describes the error between the quantized and the ideal continuous value. This error is given in LSB units. It arises owing to the accumulated error of the step size. This (see Fig. 3.28) changes itself continuously from one output code to another.

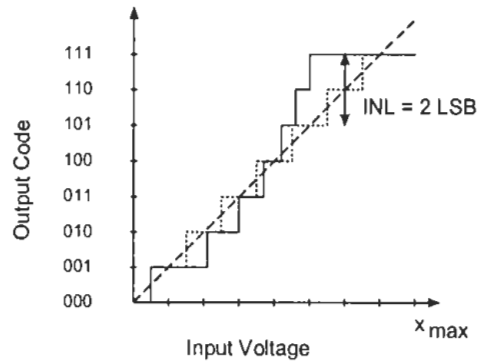


Figure 3.28 Integral nonlinearity.

Monotonicity. The progressive increase in quantizer output code for a continuously increasing input voltage and progressive decrease in quantizer output code for a continuously decreasing input voltage is called monotonicity. An example of non-monotonic behavior is shown in Fig. 3.29 where one output code does not occur.

Total Harmonic Distortion. The harmonic distortion is calculated for an AD converter at full range with a sinusoid ($X_1 = 0$ dB) of given frequency. The selective measurement of harmonics of the second- to the ninth-order are used to

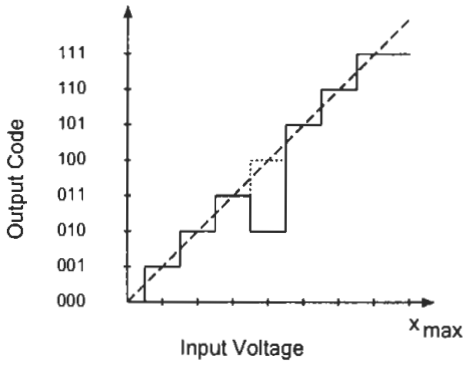


Figure 3.29 Monotonicity.

compute

$$\text{THD} = 20 \log \sqrt{\sum_{n=2}^{\infty} [10^{(-X_n/20)}]^2} \quad [\text{dB}] \tag{3.35}$$

$$= \sqrt{\sum_{n=2}^{\infty} [10^{(-X_n/20)}]^2} \cdot 100\% \tag{3.36}$$

where X_n are the harmonics in dB.

THD+N: Total Harmonic Distortion plus Noise. For the calculation of harmonic distortion plus noise, the test signal is suppressed by a stop-band filter. The measurement of harmonic distortion plus noise is performed by measuring the remaining broad-band noise signal which consists of integral and differential nonlinearity, missing codes, aperture jitter, analog noise and quantization error.

3.2.2 Parallel Converter

Parallel Converter. A direct method for AD conversion is called parallel conversion (flash converter). In parallel converters, the output voltage of the sample-and-hold circuit is compared with a reference voltage U_R with the help of $2^w - 1$ comparators (see Fig. 3.30). The sample-and-hold circuit is controlled with sampling rate f_S so that during the hold-time t_H, a constant voltage at the output of the sample-and-hold circuit is available. The outputs of the comparators are fed at sampling clock rate into a $2^w - 1$ bit register and converted by a coding logic to a w bit data word. This is fed at sampling clock rate to an output register. The sampling rates that can be achieved lie between 1 and 500 MHz for a resolution of up to 10 bits. Owing to the large number of comparators, the technique is not feasible for high precision.

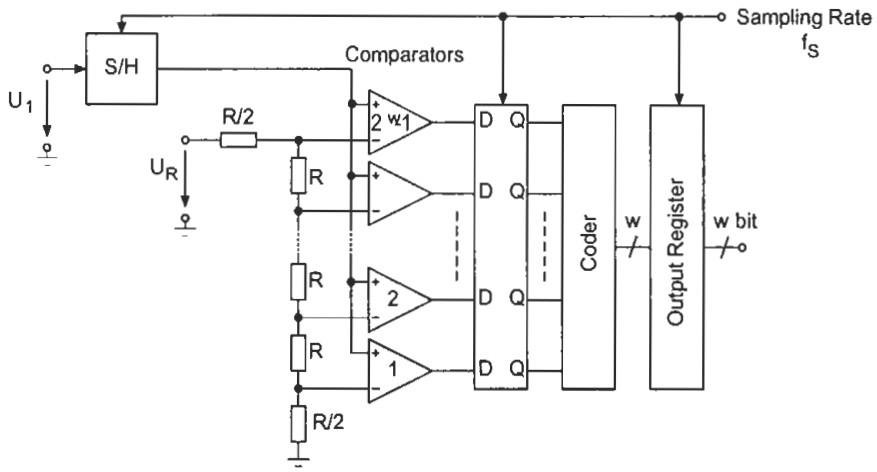


Figure 3.30 Parallel converter.

Half-flash Converter. In half-flash AD converters (Fig. 3.31), two m bit parallel converters are used in order to convert two different ranges. The first m bit AD converter gives a digital output word which is converted into an analog voltage using an m bit DA converter. This voltage is now subtracted from the output voltage of the sample-and-hold circuit. The difference voltage is digitized with a second m bit AD converter. The rough and fine quantization leads to a w bit data word with a subsequent logic.

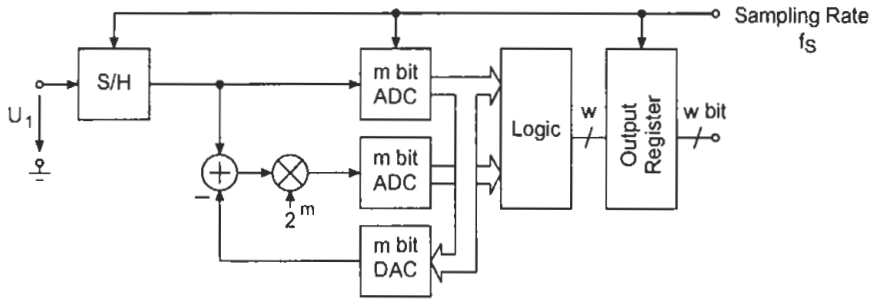


Figure 3.31 Half-flash AD converter.

Subranging Converter. A combination of direct conversion and sequential procedure is carried out for subranging AD converters (see Fig. 3.32). In contrast to the half-flash converter, only one parallel converter is required. The switches S_1 and S_2 take the values of 0 and 1. First, the output voltage of a sample-and-hold circuit and then the difference voltage amplified by a factor 2^m is fed to an m bit AD converter. The difference voltage is formed with the help of the output voltage of an m bit DA converter and the output voltage of the sample-and-hold circuit. The conversion rates lie between 100 kHz and 40 MHz where a resolution of up to

16 bits is achieved.

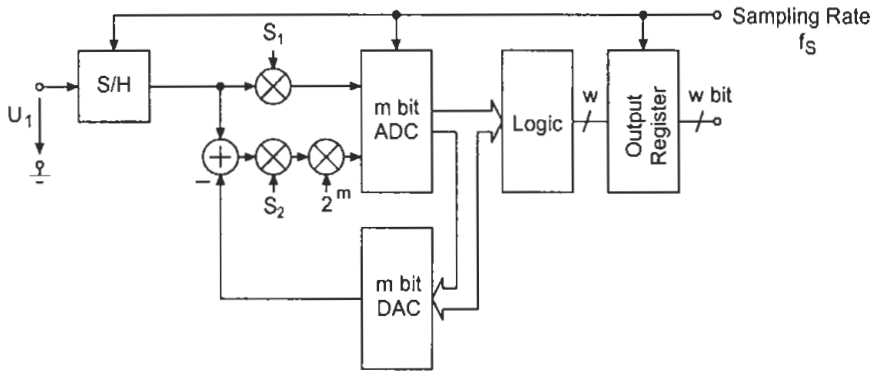


Figure 3.32 Subranging AD converter.

3.2.3 Successive Approximation

AD converters with successive approximation consist of the functional modules shown in Fig. 3.33. The analog voltage is converted into a w bit word within w cycles. The converter consists of a comparator, a w bit DA converter and logic for controlling the successive approximation.

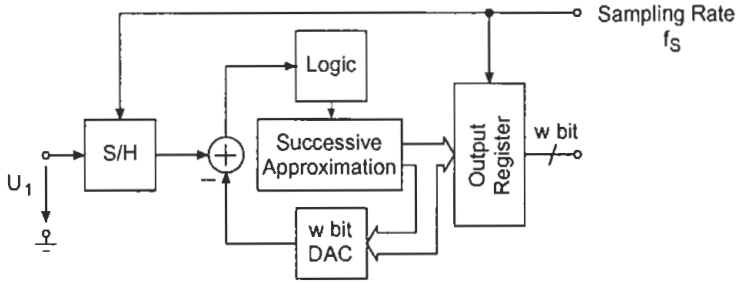


Figure 3.33 AD converter with successive approximation.

The conversion process is explained with the help of Fig. 3.34. First, it is checked whether a positive or negative voltage is present at the comparator. If it is positive, the output $+0.5U_R$ is fed to a DA converter to check whether the output voltage of the comparator is greater or less than $+0.5U_R$. After that, the output of $(+0.5 \pm 0.25)U_R$ is fed to the DA comparator. The output of the comparator is then evaluated. This procedure is performed w times and leads to a w bit word.

For a resolution of 12 bits, sampling rates of up to 1 MHz can be achieved. Higher resolutions of more than 16 bits are possible at a lower sampling rates.

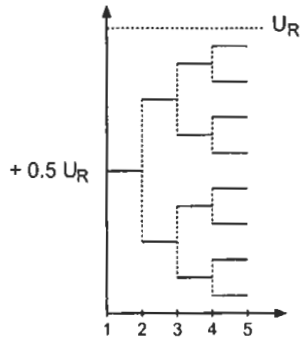


Figure 3.34 Successive approximation.

3.2.4 Counter Methods

In contrast to the conversion techniques of the previous sections for high conversion rates, the following techniques are used for sampling rates smaller than 50 kHz.

Forward-backward Counter. A technique which operates like successive approximation is the forward-backward counter shown in Fig. 3.35. A logic controls a clocked forward-backward counter whose output data word provides an analog output voltage via a w bit DA converter. The difference signal between this voltage and the output voltage of the sample-and-hold circuit determines the direction of counting. The counter stops when the corresponding output voltage of the DA converter is equal to the output voltage of the sample-and-hold circuit.

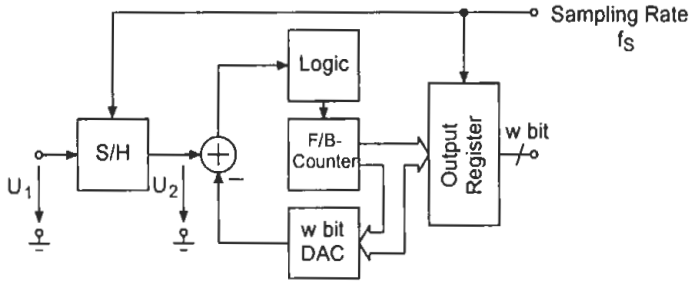


Figure 3.35 AD converter with forward-backward counter.

Single-slope Counter. The single-slope AD converter shown in Fig. 3.36 compares the output voltage of the sample-and-hold circuit with a voltage of a sawtooth generator. The sawtooth generator is started every sampling period. As long as the input voltage is greater than the sawtooth voltage, the clock impulses are counted. The counter value corresponds to the digital value of the input voltage.

Dual-slope Converter. A dual-slope AD converter is shown in Fig. 3.37. In the first phase in which a switch S_1 is closed for a counter period t_1, the output

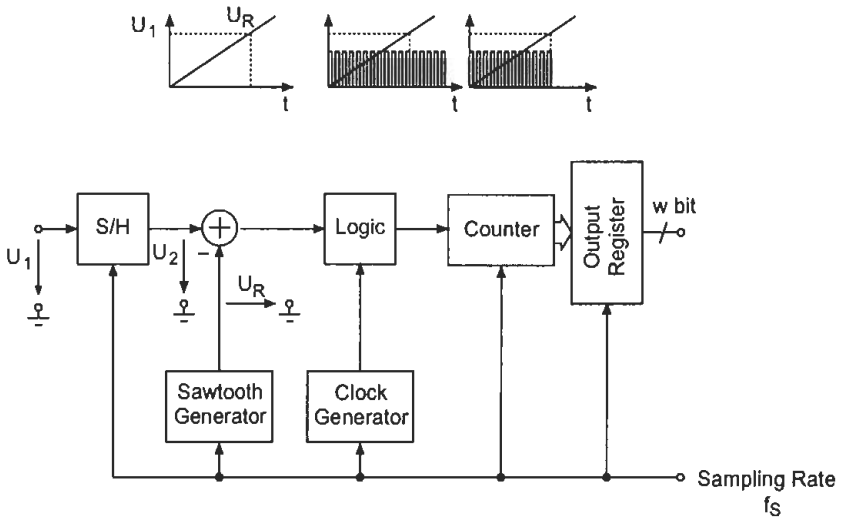


Figure 3.36 Single-slope AD converter.

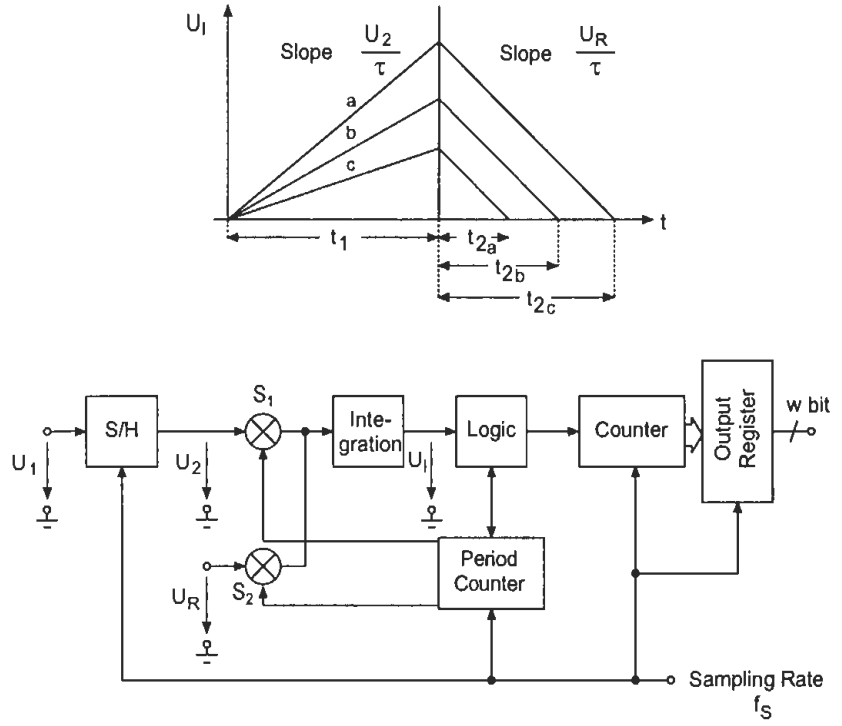


Figure 3.37 Dual-slope AD converter.

voltage of the sample-and-hold circuit is fed to an integrator of time-constant τ. During the second phase, the switch S_2 is closed and the switch S_1 is opened. The reference voltage is switched to the integrator and the time to reach a threshold is determined by counting the clock impulses by a counter. Figure 3.37 demonstrates

this for three different voltages U_2. The slope during time t_1 is proportional to the output voltage U_2 of the sample-and-hold circuit whereas the slope is constant when the reference voltage U_R is connected to the integrator. The ratio $U_2/U_R = t_2/t_1$ leads to the digital output word.

3.2.5 Delta-sigma AD Converter

The delta-sigma AD converter in Fig. 3.38 requires no sample-and-hold circuit owing to its high conversion rate. The analog band-limiting low-pass filter and the digital low-pass filter for downsampling to a sampling rate f_S are usually on the same circuit. The linear phase nonrecursive digital low-pass filter in Fig. 3.38 has a 1 bit input signal and leads to a w bit output signal owing to the N filter coefficients $h_0, h_1, \ldots, h_{N-1}$ which are implemented with a word-length of w bits. The output signal of the filter results from the summation of the filter coefficients of the nonrecursive low-pass filter with either 0 or 1. The downsampling by a factor L is performed by taking every Lth sample out of the filter and writing to the output register. In order to reduce the number of operations the filtering and downsampling can be performed only every Lth input sample.

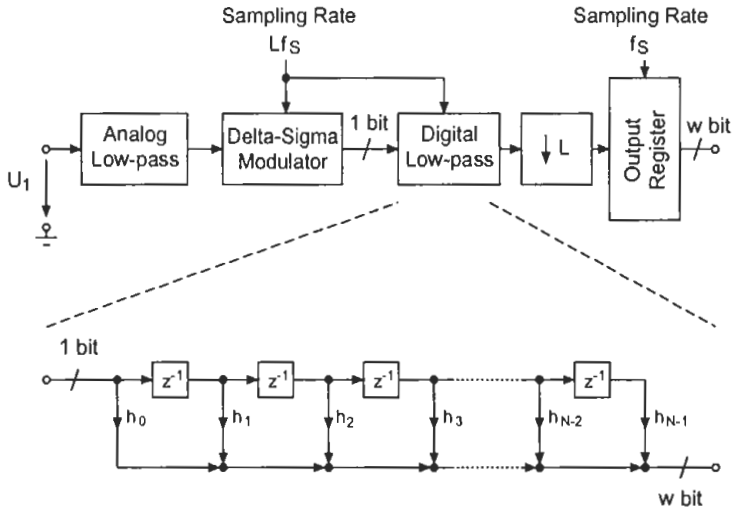


Figure 3.38 Delta-sigma AD converter.

The applications of delta-sigma AD converters are found at sampling rates of up to 100 kHz with a resolution of up to 24 bits.

3.3 DA Converters

Circuit principles for DA converters are mainly based on direct conversion techniques of the input code. Therefore, the achievable sampling rates are accordingly high.

3.3.1 Specifications

The definitions of resolution, total harmonic distortion (THD) and total harmonic distortion plus noise (THD+N) correspond to those for AD converters. Further specifications are discussed in the following.

Settling Time. The time interval between transferring a binary word and achieving the analog output value within a specific error range is called the settling time t_{SE}. The settling time determines the maximum conversion frequency $f_{S_{max}} = 1/t_{SE}$. Within this time, glitches between consecutive amplitude values can occur (see Fig. 3.39). With the help of a sample-and-hold circuit (deglitcher), the output voltage of the DA converter is sampled after the settling time and held.

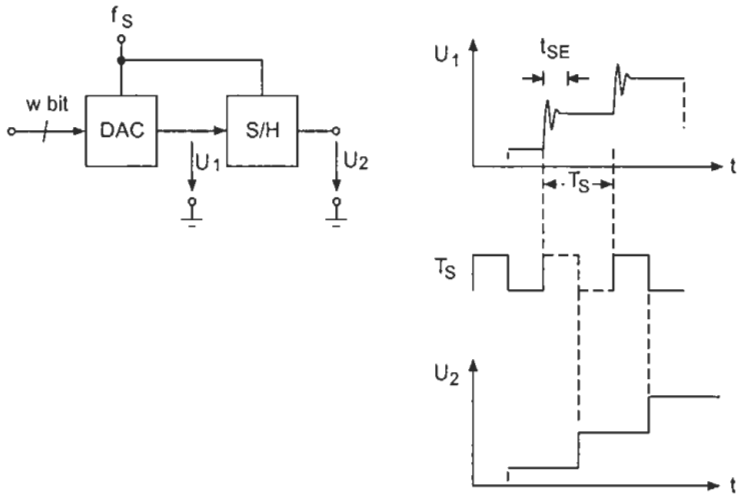


Figure 3.39 Settling time and sample-and-hold function.

Offset and Gain Error. The offset and gain errors of a DA converter are shown in Fig. 3.40.

Differential Nonlinearity. The differential nonlinearity for DA converters describes the step size error of a code word in LSB. For ideal quantization, the increase Δx of the output voltage until the next code word corresponding to the

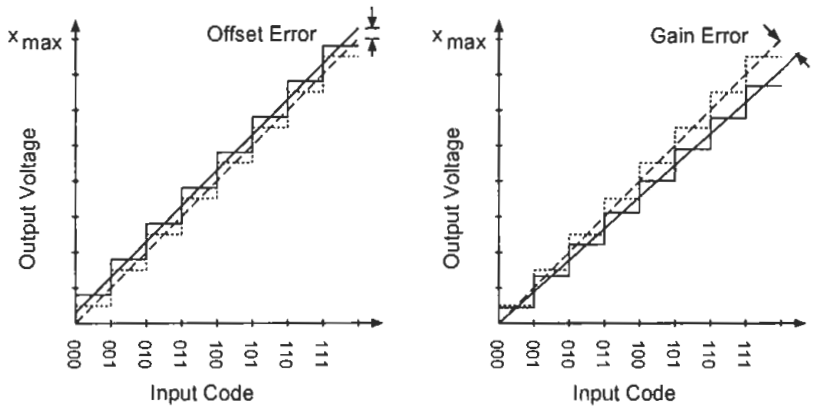


Figure 3.40 Offset and gain error.

output voltage is equal to the quantization step size Q (see Fig. 3.41). The difference of two consecutive input codes is termed Δx_Q. Differential nonlinearity is given by

$$\text{DNL} = \frac{\Delta x/Q}{\Delta x_Q} - 1 \qquad \text{[LSB]}. \tag{3.37}$$

For the code step from 001 to 010 as shown in Fig. 3.41, the step size is 1.5 LSB, and therefore the differential nonlinearity DNL = 0.5 LSB. The step size between the codes 010 and 100 is 0.75 LSB and it follows for DNL = -0.25. The step size for the code change from 011 to 100 is 0 LSB (DNL = -1 LSB).

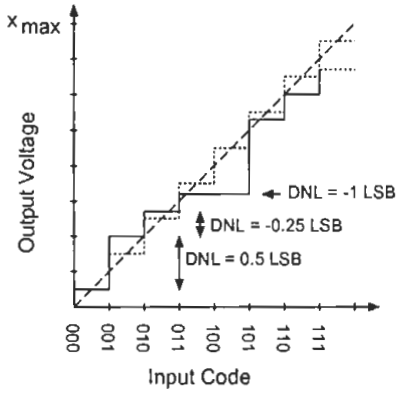


Figure 3.41 Differential nonlinearity.

Integral Nonlinearity. The integral nonlinearity describes the maximum deviation of the output voltage of a real DA converter from the ideal straight line (see Fig. 3.42).

Monotonicity. The continuous increase of the output voltage with increasing input code and the continuous decrease of the output voltage with decreasing input

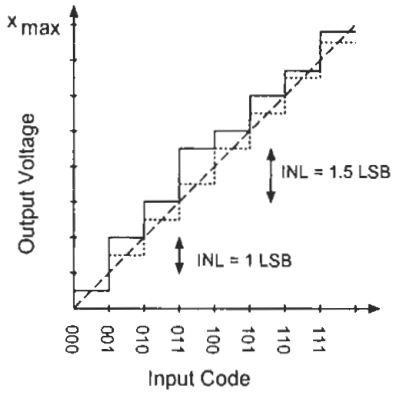


Figure 3.42 Integral nonlinearity.

code is called monotonicity. A non-monotonic behavior is presented in Fig. 3.43.

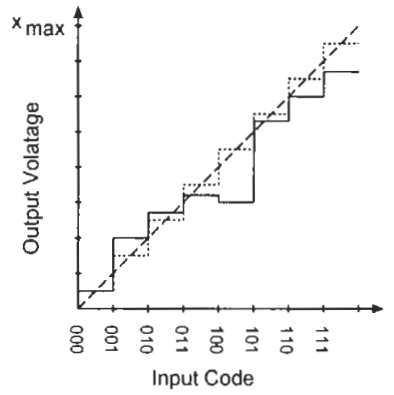


Figure 3.43 Monotonicity.

3.3.2 Switched Voltage and Current Sources

Switched Voltage Sources. The DA conversion with switched voltage sources shown in Fig. 3.44a is carried out with a reference voltage connected to a resistor network. The resistor network consists of 2^w resistors of equal resistance and is switched in stages to a binary-controlled decoder so that at the output, a voltage U_2 is present corresponding to the input code. Figure 3.44b shows the decoder for a 3 bit input code 101.

Switched Current Sources. DA conversion with 2^w switched current sources is shown in Fig. 3.45. The decoder switches the corresponding number of current sources onto the current-voltage converter. The advantage of both techniques is the monotonicity which is guaranteed for ideal switches but also for slightly deviating

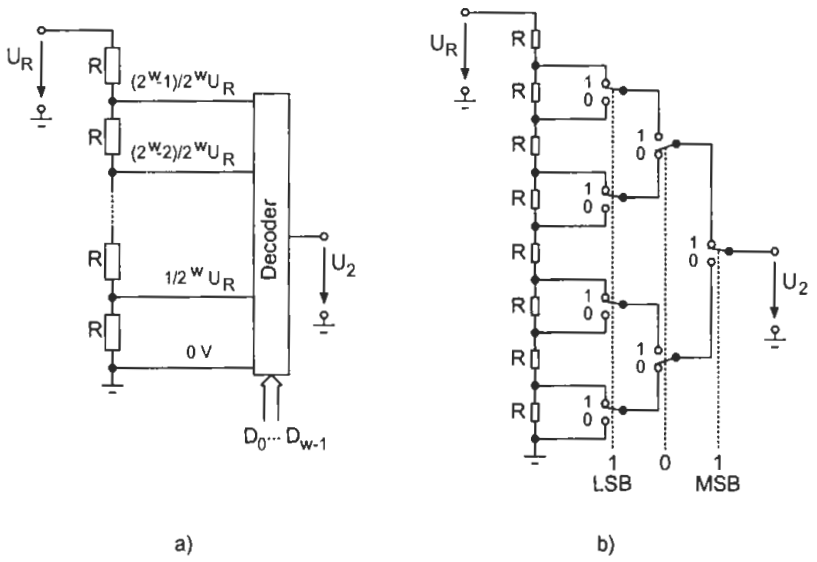


Figure 3.44 Switched voltage sources.

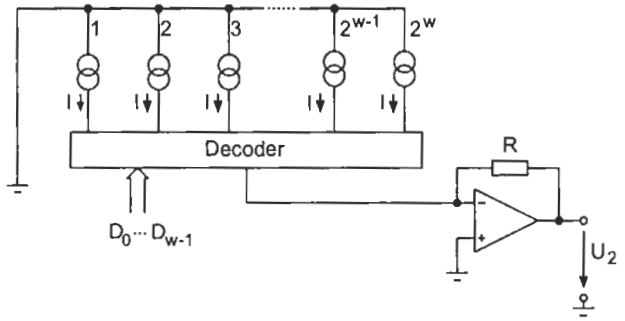


Figure 3.45 Switched current sources.

resistances. The large number of resistors in switched current sources or the large number of switched current sources causes problems for long word-lengths. The techniques are used in combination with other a methods for DA conversion of higher significant bits.

3.3.3 Weighted Resistors and Capacitors

A reduction in the number of identical resistors or current sources is achieved with the following method.

Weighted Resistors. DA conversion with w switched current sources which are weighted according to

$$I_1 = 2I_2 = 4I_3 = \ldots = 2^{w-1} I_w \tag{3.38}$$

is shown in Fig. 3.46. The output voltage is

$$U_2 = -R \cdot I = -R \cdot (b_1 I_1 2^0 + b_2 I_2 2^1 + b_3 I_3 2^2 + \ldots + b_w I_w 2^{w-1}), \qquad (3.39)$$

where b_n takes values 0 or 1. The implementation of DA conversion with switched

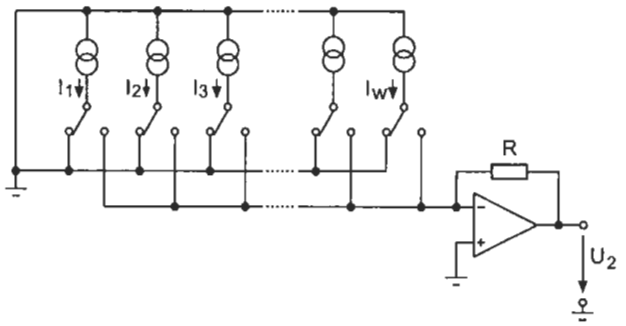


Figure 3.46 Weighted current sources.

current sources is carried out with weighted resistors as shown in Fig. 3.47. The

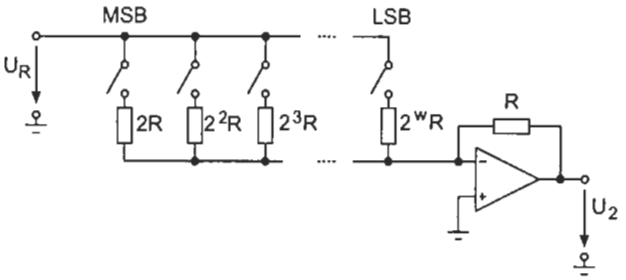


Figure 3.47 DA conversion with weighted resistors.

output voltage is

$$U_2 = R \cdot I = R\left(\frac{b_1}{2R} + \frac{b_2}{4R} + \frac{b_4}{8R} + \ldots + \frac{b_w}{2^w R}\right) U_R \qquad (3.40)$$

$$= (b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + \ldots b_w 2^{-w}) U_R. \qquad (3.41)$$

Weighted Capacitors. DA conversion with weighted capacitors is shown in Fig. 3.48. During the first phase (switch position 1 in Fig. 3.48) all capacitors are discharged. During the second phase, all capacitors that belong to 1 bits, are connected to a reference voltage. Those capacitors belonging to 0 bits are connected to ground. The charge on the capacitors C_a that are connected with the reference voltage can be set equal to the total charge on all capacitors C_g, which leads to

$$U_R\, C_a = U_R\left(b_1 C + \frac{b_2 C}{2} + \frac{b_3 C}{2^2} + \ldots + \frac{b_w C}{2^{w-1}}\right) = C_g\, U_2 = 2\, C\, U_2. \qquad (3.42)$$

Hence, the output voltage is

$$U_2 = (b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + \ldots + b_w 2^{-w}) U_R. \tag{3.43}$$

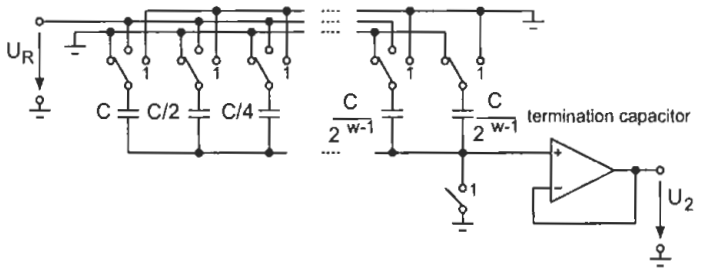


Figure 3.48 DA conversion with weighted capacitors.

3.3.4 R-2R Resistor Networks

The DA conversion with switched current sources can also be carried with an R-2R resistor network shown in Fig. 3.49. In contrast to the method with weighted resistors, the ratio of the smallest to largest resistor is reduced to 2:1.

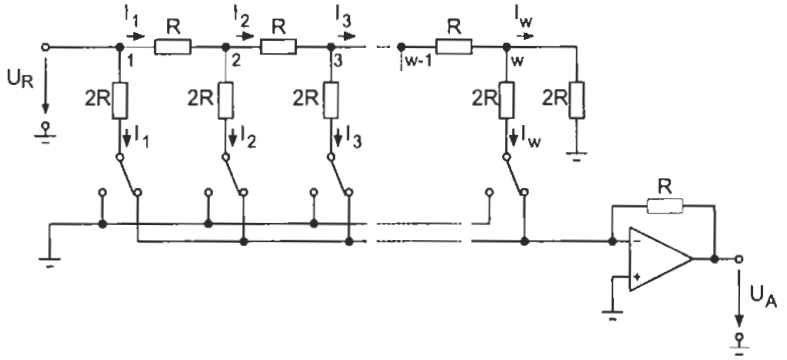


Figure 3.49 Switched current sources with R-2R resistor network.

The weighting of currents is achieved by a current division in every junction. Looking right from every junction, a resulting resistance $R + 2R \parallel 2R = 2R$ is found which is equal to the resistance in vertical direction downwards from the junction. For the current from junction 1 follows $I_1 = \frac{U_R}{2R}$, and for the current from junction 2 $I_2 = \frac{I_1}{2}$. Hence, a binary weighting of the w currents is given by

$$I_1 = 2I_2 = 4I_3 = \ldots = 2^{w-1} I_w. \tag{3.44}$$

The output voltage U_2 can be written as

$$U_2 = -RI = -R(\frac{b_1}{2R} + \frac{b_2}{4R} + \frac{b_3}{8R} + \ldots + \frac{b_w}{2^{w-1}R})U_R \quad (3.45)$$

$$= -U_R(b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + \ldots + b_w 2^{-w}). \quad (3.46)$$

3.3.5 Delta-sigma DA Converter

A delta-sigma DA converter is shown in Fig. 3.50. The converter is provided with w bit data words by an input register with the sampling rate f_S. This is followed by a sample rate conversion up to Lf_S by upsampling and a digital low-pass filter. A delta-sigma modulator converts the w bit input signal into a 1 bit output signal. The delta-sigma modulator corresponds to the model in section 3.1.3. Subsequently, the DA conversion of the 1 bit signal is performed followed by the reconstruction of the time-continuous signal by an analog low-pass filter.

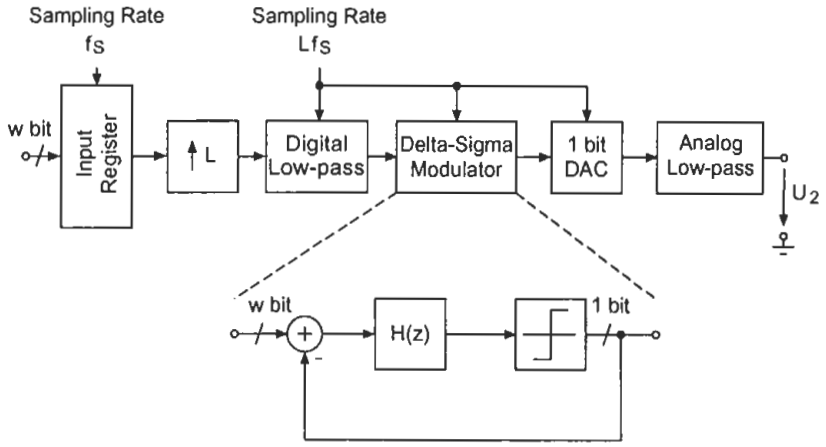


Figure 3.50 Delta-sigma DA converter.

Chapter 4

Audio Processing Systems

4.1 Digital Signal Processors

Digital signal processors (DSP) are used for discrete-time signal processing. Their architecture and instruction set is specially designed for real-time processing of signal processing algorithms. Digital signal processors of different manufacturers and their use in practical circuits will be discussed. The restriction to the architecture and practical circuits shall provide the user with the criteria necessary for selecting a DSP for a particular application. From the architectural features of different DSPs, the advantages of a certain processor with respect to fast execution of algorithms (digital filter, adaptive filter, FFT etc.) automatically result. The programming methods and application programs are not dealt with here, because the DSP user guides from different manufacturers provide adequate information in the form of sample programs for a variety of signal processing algorithms.

After comparing DSPs with other microcomputers, the following topics will be discussed in the forthcoming sections:

- Fixed-point DSPs
- Floating-point DSPs
- Development tools
- Single-processor systems
  (peripherals, control principles)

- Multi-processor systems
  (coupling principles, control principles)

The internal design of microcomputers is mainly based on two architectures; the *von Neumann* architecture which uses shared instruction/data bus; and the *Harvard* architecture which has separate buses for instructions and data. Processors based on these architectures are CISCs, RISCs and DSPs. Their characteristics are given in Table 4.1. Besides the internal properties listed in the table, DSPs

Table 4.1 CISC, RISC and DSP.

type	characteristics
CISC	Complex Instruction Set Computer - von Neumann architecture - assembler programming - large number of instructions - computer families - compilers - application: universal microcomputers
RISC	Reduced Instruction Set Computer - von Neumann architecture/Harvard architecture - number of instructions < 50 - number of address modes < 4, instruction formats < 4 - hard wired instruction (no microprogramming) - processing most of the instructions in one cycle - optimizing compilers for high-level programming languages - application: workstations
DSP	Digital Signal Processor - Harvard architecture - several internal data buses - assembler programming - parallel processing of several instructions in one cycle - optimizing compilers for high-level programming languages - real-time operating systems - application: real-time signal processing

have special on-chip peripherals which are suited to signal processing applications. The fast response to external interrupts enables their use in real-time operating systems.

4.1.1 Fixed-point DSPs

The discrete-time and discrete-amplitude output of an AD converter is usually represented in 2s complement format. The processing of these number sequences is carried out with fixed-point or floating-point arithmetic. The output of a processed signal is again in 2s complement format and is fed to a DA converter. The signed fractional representation (2s complement) is the common way for algorithms in fixed-point number representation. For address generation and modulo operations unsigned integers are used. Figure 4.1 shows a schematic diagram of a typical fixed-point DSP. The main building blocks are program controller, arithmetic lo-

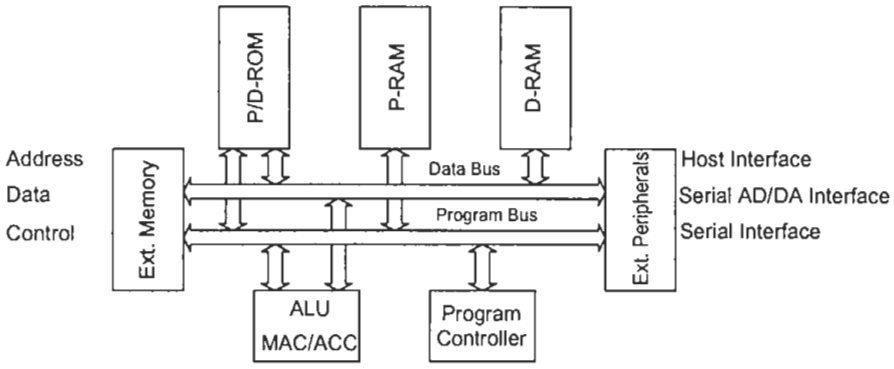


Figure 4.1 Schematic diagram of a fixed-point DSP.

gic unit (ALU) with a multiplier-accumulator (MAC), program and data memory and interfaces to external memory and peripherals. All blocks are connected with each other by an internal bus system. The internal bus system has separate instruction and data buses. The data bus itself can consist of more than one parallel bus enabling it, for instance, to transmit both operands of a multiplication instruction to the MAC in parallel. The internal memory consists of instruction and data RAM and additional ROM memory. This internal memory permits a fast execution of internal instructions and data transfer. For increasing memory space, address/control and data buses are connected to external memories like EPROM, ROM and RAM. The connection of the external bus system to the internal bus architecture has great influence on efficient execution of external instructions as well as on processing external data. In order to connect serially operating AD/DA converters, special serial interfaces with high transmission rates are offered by several DSPs. Moreover, some processors support direct connection to an RS232 interface. The control from a microprocessor can be achieved via a host interface with a word-length of 8 bits.

An overview of fixed-point DSPs with respect to word-length and cycle time is shown in Table 4.2. Basically, the precision of the arithmetic can be doubled if

quantization affects the stability and numeric precision of the applied algorithm. The cycle time in connection with processing time (in processor cycles) of a combined multiplication and accumulation command gives insight into the computing power of a particular processor type. The cycle time directly results from the maximum clock frequency. The instruction processing time depends mainly on the internal instruction and data structure as well as on the external memory connections of the processor. Table 4.3 contains the internal memory partitioning of several

Table 4.2 Fixed-point DSPs.

type	word-length	cycle time
ADSP-2100	16	60/77/80/100 ns
AT&T DSP16	16	25/33/55/75 ns
Motorola DSP56156	16	33/50 ns
Motorola DSP56001/2	24	60/74 ns
NEC 77C25	16	100/122 ns
NEC 77220	24	100/122 ns
TI 320C1x	16	114/120/200/280 ns
TI 320C2x/5x	16	C2x 78/98/125 ns C5x 35/50 ns

DSPs. A large on-chip memory for data and instructions is a precondition for efficient programming of algorithms. Data and instruction transfer from external memories can hence be avoided. The availability of special tables (cosine, sine) in ROM supports algorithms like FFT.

Table 4.3 Internal memory structure (P = program, D = data).

type	on-chip D-RAM	on-chip P-RAM	on-chip ROM
ADSP-2100	-	16	-
ADSP-2101/2/11	1k	2k	-
AT&T DSP16A	2k	-	4k (P/D)
DSP56156	2k	2k	64 (P)
DSP56001/2	2x256	512	2x256 (D)
NEC 77C25	256	-	1k (D) 2k (P)
NEC 77220	2x256	-	1k (D) 2k (P)
TI 320C1x	256	-	4k (P)
TI 320C25	288	256 (P/D)	4k (P/D)
TI 320C26	-	1.5k (D/P)	256 (P)
TI 320C50/51	1k	9k/1k (D/P)	2k (P)/8k (P)

The fast access to external instruction and data memories is of special significance in complex algorithms and in processing huge data loads. Further attention has to be paid to the linking of serial data connections with AD/DA converters and the control by a host computer over a special host interface (Table 4.4). Complex interface circuits could therefore be avoided. For stand-alone solutions, program loading from a simple external EPROM can also be done.

Table 4.4 External peripherals (xS = x serial interface, xP = x parallel interface).

type	external memory	on-chip peripherals
ADSP-2100	32k (P), 16k (D)	-
ADSP-2101/2	16k (P), 16k (D)	2S
ADSP-2111	16k (P), 15k (D)	2S, 1P
AT&T DSP16A	64k	1S, 1P
DSP56156	64k (P), 64k (D)	2S, 1P
DSP56001/2	64k (P), 128k (D)	2S, 1P
NEC 77C25	-	1S, 1P
NEC 77220	8k (P), 8k (D)	1S, 1P
TI 320C1x	4k	
TI 320C25/26	64k (P), 64k (D)	1S
TI 320C50/51	64k (P), 64k (D)	2S

For signal processing algorithms, the following software commands are necessary:

1. MAC (multiply and accumulate)
   → combined multiplication and addition command

2. simultaneous transfer of both operands for multiplication to the MAC (parallel move)

3. bit-reversed addressing (for FFT)

4. modulo addressing (for windowing and filtering)

Different signal processors have different processing times for FFT implementations. The latest signal processors with improved architecture have shorter processing times. The instruction cycles for the combined multiplication and accumulation command (application: windowing, filtering) are approximately equal for different processors, but processing cycles for external operands have to be considered.

4.1.2 Floating-point DSPs

Figure 4.2 shows the block diagram of a typical floating-point DSP. The main characteristics of the different architectures are the dual-port principle (Motorola, Texas Instruments) and the external *Harvard* architecture (Analog Devices, NEC). Floating-point DSPs internally have multiple bus systems in order to accelerate data transfer to the processing unit. An On-chip DMA controller and cache-memory support higher data transfer rates. An overview of floating-point DSPs is shown in Table 4.5. Besides the standardized floating-point representation IEEE-754, there are also manufacturer-dependent number representations. The internal memory structure is given in Table 4.6.

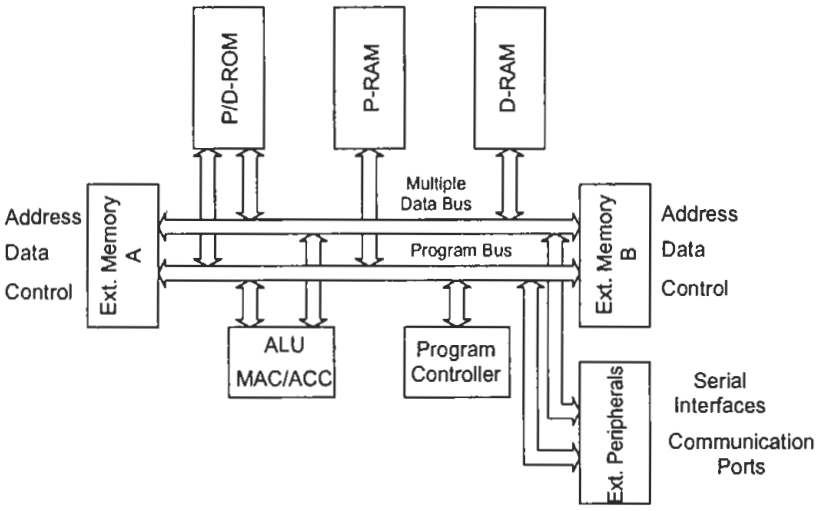


Figure 4.2 Block diagram of a floating-point digital signal processor.

Table 4.5 Floating-point DSPs.

type	word-length	cycle time
ADSP-21060	32 (IEEE-754)	25 ns
AT&T DSP32C	32	80/100 ns
Motorola DSP96002	32 (IEEE-754)	50/60/74 ns
NEC 77230	32	150 ns
NEC 77240	32	90 ns
TI 320C3x	32	50/60/74 ns
TI 320C40	32	40/50 ns

An overview of external address space and on-chip peripherals is given in Table 4.7. The external dual-port architecture of some floating-point DSPs supports the design of multiprocessor systems.

Table 4.6 Internal memory structure (C = cache memory).

type	on-chip D/P-RAM	on-chip ROM
ADSP-21060	4 Mbit (P/D)	
AT&T DSP32C	2x512 (P/D)	4k (P/D) or 512 P/D-RAM
M DSP96002	2x512 (D), 1k (P)	64 Boot, 2x512 (D)
NEC 77230	2x512 (D)	2k (P), 1k (D)
NEC 77240	2x512 (D)	2k (P), 1k (D)
TI 320C3x	2x1k (P/D) 64 (C)	4k (P/D)
TI 320C40	2x1k (P/D) 128 (C)	4k (P/D)

Table 4.7 External peripherals (xS = x serial interface, xP = x parallel interface).

type	ext. buses	on-chip peripherals
ADSP-21060	4Gx32/48 (P/D)	2S/6P
AT&T DSP32C	4Mx32 (P/D)	1S, 1P
DSP96002	2x 32 bit (A), 2x 32 bit (D) dual-port architecture	-
NEC 77230	8kx32 (P/D)	1S
NEC 77240	64kx32 (P), 16Mx32 (D) external Harvard architecture	
TI 320C30	16Mx32 (P/D), 8kx32 (P/D) dual-port architecture	2S
TI 320C40	2x 32 bit (A), 2x 32 bit (D) dual-port architecture	6P

4.1.3 Development Tools

The rapid development of hard- and software for a certain application is supported by development tools which are listed below:

- Manufacturers' Literature (Data Books): Data books of manufacturers, application examples and detailed program libraries.
- Assembler/Compiler/Linker: Tools for developing software.
- High-level Language Compilers: The use of higher language compilers allows a fast software development without special knowledge of the architecture

and the instruction set of the processor. The generated assembler code can be optimized with respect to processing speed. The advantage of using higher language compilers is the compatibility of the code for different signal processors and the associated fast access to algorithms for future DSPs.

- Real-time Operating Systems: Special operating systems for DSPs with a core consisting of memory management and hardware-interrupt handling, a programming interface for linking application programs and a real-time multitasking core.
- Software Simulator: A software simulator simulates the modules of a DSP and the execution of programs. All registers, memories and interfaces are accessible. Therefore, the programs can be tested under the boundary conditions of the DSP.
- Hardware Emulator (In-circuit Emulation): In-circuit emulation serves for testing the DSP in the target hardware. With the help of a special target cable, the hardware emulator is connected into the socket for the DSP.
- On-chip Emulation: The advantage of a DSP with integrated on-chip emulation is the possibility of self-testing the hardware and software in the target application with the DSP.
- EPROM Simulator: Besides the on-chip emulation, the use of EPROM simulators (or program loading by a control processor) supports the hardware and software development in the target hardware.

4.2 Digital Audio Interfaces

For transferring digital audio signals, two transmission standards have been established by the AES (Audio Engineering Society) and the EBU (European Broadcasting Union) respectively. These standards are for two-channel transmission [AES92] and for multichannel transmission of up to 56 audio signals.

4.2.1 Two-channel AES/EBU Interface

For the two-channel AES/EBU interface, professional and consumer modes are defined. The outer frame is identical for both modes and is shown in Fig. 4.3. For a sampling period a frame is defined so that it consists of two subframes, for channel 1 with preamble X, and for channel 2 with preamble Y. A total of 192 frames form a block, the block start is characterized by a special preamble Z. The

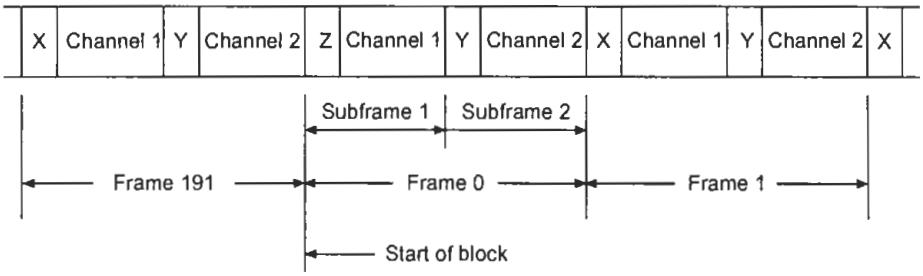


Figure 4.3 Two-channel format.

bit allocation of a subframe consists of 32 bits as in Fig. 4.4. The preamble consists of 4 bits (bit 0...3) and the audio data of up to 24 bits (bit 4...27). The last four bits of the subframe characterize Validity (validity of data word or error), User Status (usable bit), Channel Status (from 192 bits/block=24 bytes coded status information for the channel) and Parity (even parity). The transmission of the

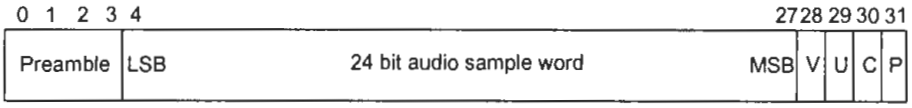


Figure 4.4 Two-channel format (subframe).

serial data bits is carried out with a biphase code. This is done with the help of an XOR relationship between clock (of double bit rate) and the serial data bits (Fig. 4.5). At the receiver, clock retrieval is achieved by detecting the preamble (X=11100010, Y=11100100, Z=11101000) as it violates the coding rule (see Fig. 4.6). The meaning of the 24 bytes for channel status information is summarized in

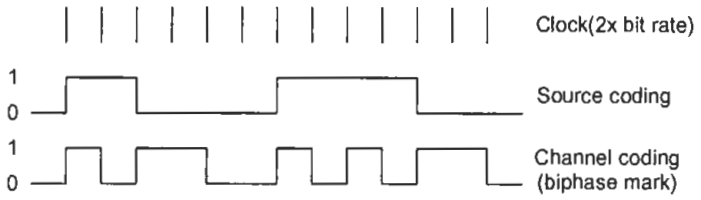


Figure 4.5 Channel coding.

Table 4.8. An exact bit allocation of the first three important bytes of this channel status information is presented in Fig. 4.7. In the individual fields of byte 0, preemphasis and sampling rate are specified besides professional/consumer modes and the characterization of data/audio (see Tables 4.9 and 4.10). Byte 1 determines the channel mode (Table 4.11). The consumer format (often labeled as SPDIF = Sony/Philips Digital Interface Format) differs from the professional format in the definition of the channel status information and the technical specifications

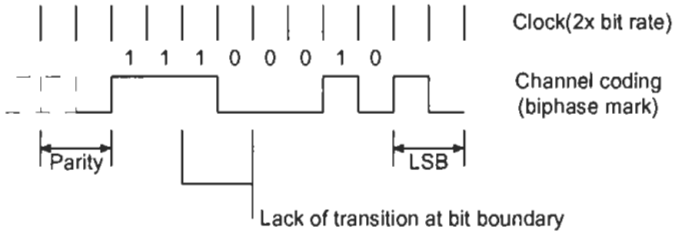


Figure 4.6 Preamble X.

Table 4.8 Channel status bytes.

byte	description
0	emphasis, sampling rate
1	channel use
2	sample length
3	vector for byte 1
4	reference bits
5	reserved
6-9	4 bytes of ASCII origin
10-13	4 bytes of ASCII destination
14-17	4 bytes of local address
18-21	time code
22	flags
23	CRC

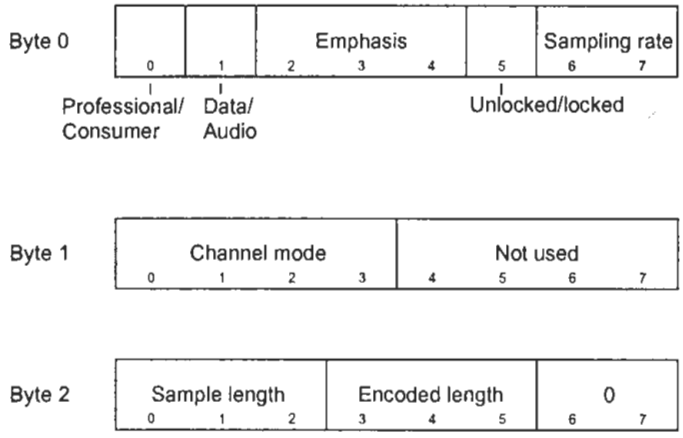


Figure 4.7 Bytes 0...2 of channel status information.

for inputs and outputs. The bit allocation for the first four bits of the channel information is shown in Fig. 4.8. For consumer applications, two-wired leads with RCA connectors are used. The inputs and outputs are asymmetrical. Also, optical

Table 4.9 Emphasis field.

0	none indicated, override enabled
4	none indicated, override disabled
6	50/15 μs emphasis
7	CCITT J.17 emphasis

Table 4.10 Sampling rate field.

0	none indicated (48 kHz default)
1	48 kHz
2	44.1 kHz
3	32 kHz

Table 4.11 Channel mode.

0	none indicated (2 channel default)
1	two channel
2	monaural
3	primary/secondary (A=primary, B=secondary)
4	stereo (A=left, B=right)
7	vector to byte 3

connectors exist. For professional use, shielded two-wired leads with XLR connectors and symmetrical inputs and outputs (professional format) are used. Table 4.12 shows the electrical specifications for professional AES/EBU interfaces.

Table 4.12 Electrical specifications of professional interfaces.

output impedance	signal amplitude	jitter
110 Ω	2-7 V	max. 20 ns
input impedance	**signal amplitude**	**connect.**
110 Ω	min. 200 mV	XLR

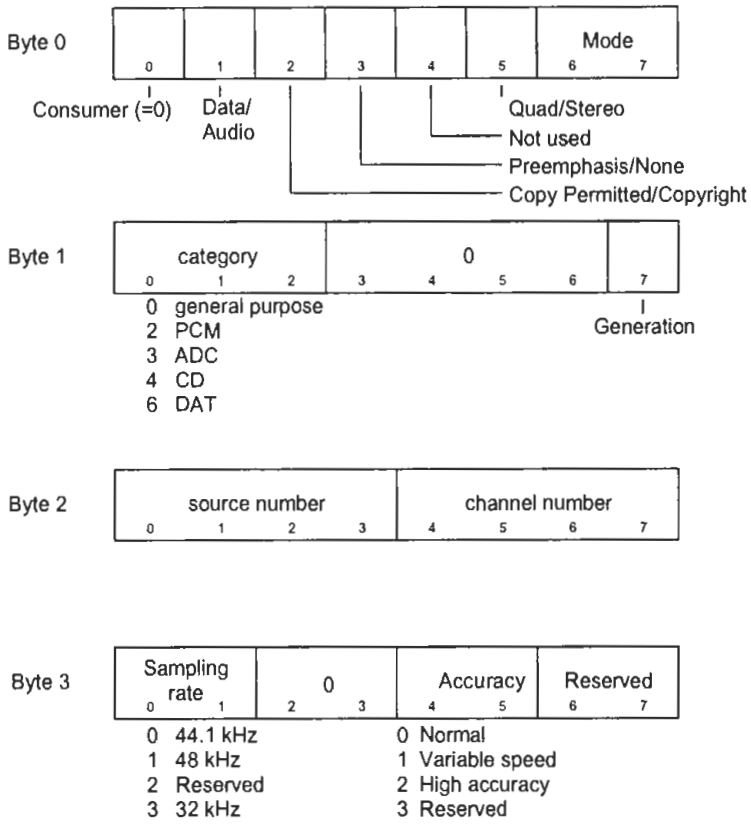


Figure 4.8 Bytes 0...3 (consumer format).

4.2.2 MADI Interface

For connecting an audio processing system at different locations, a MADI interface (Multichannel Audio Digital Interface) is used. A system link by MADI is presented in Fig. 4.9. Analog/digital I/O systems consisting of AD/DA converters, AES/EBU interfaces (AES) and sampling rate converters (SRC) are connected to digital distribution systems with bi-directional MADI links. The actual audio signal processing is performed in special DSP systems which are connected to the digital distribution systems by MADI links. The MADI format is derived from the two-channel AES/EBU format and allows the transmission of 56 digital mono channels (see Fig. 4.10) within a sampling period. The MADI frame consists of 56 AES/EBU subframes. Each channel has a preamble containing the information shown in Fig. 4.10. The bit 0 is responsible for identifying the first MADI channel (MADI Channel 0). Table 4.13 shows the sampling rates and the corresponding data transfer rates. The maximum data rate of 96.768 Mbit/s is required at sampling rate of 48 kHz+12.5%. Data transmission is done by FDDI techniques (Fiber Distributed Digital Interface). The transmision rate of 125 Mbit/s is implemented

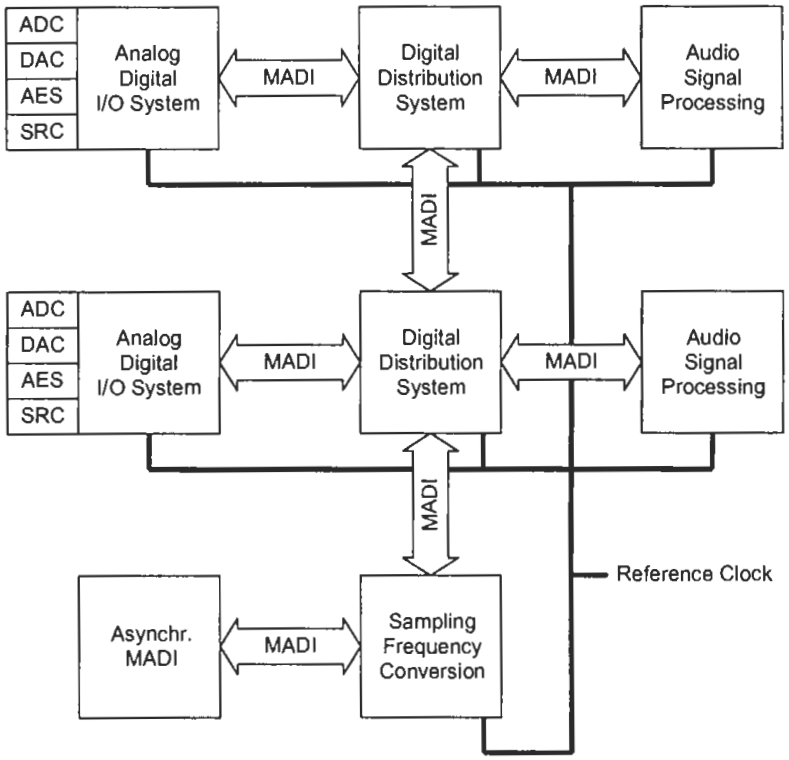


Figure 4.9 A system link by MADI.

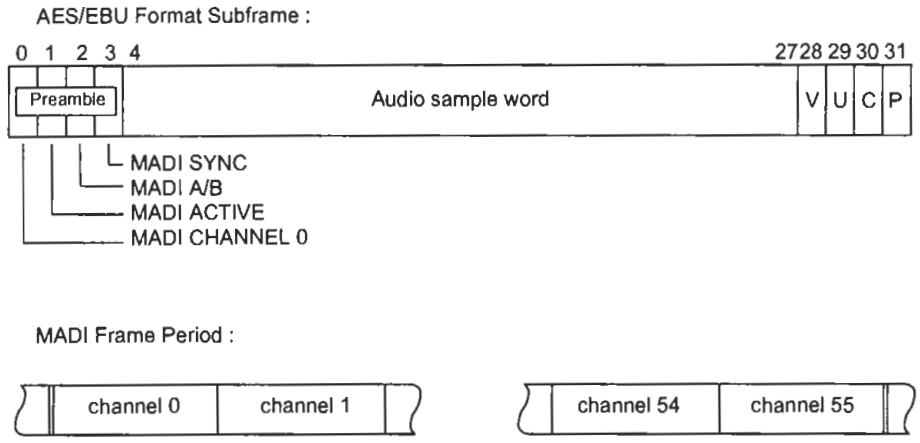


Figure 4.10 MADI frame format.

with special TAXI chips. The transmission for a coaxial cable is already specified (see Table 4.14). The optical transmission medium for audio applications is not yet defined.

A unidirectional MADI link is shown in Fig. 4.11. The MADI transmitter and receiver must be synchronized by a common master clock. The transmission

between FDDI chips is performed by a transmitter with integrated clock generation and clock retrieval at the receiver.

Table 4.13 MADI specifications.

sampling rate	32 kHz - 48 kHz ± 12.5%
transmission rate	125 Mbit/s
data transfer rate	100 Mbit/s
max. data transfer rate	96.768 Mbit/s (56 channels at 48 kHz+12.5%)
min. data transfer rate	50.176 Mbit/s (56 channels at 32 kHz−12.5%)

Table 4.14 Electrical specifications (MADI).

output impedance	signal ampl.	cable length	connect.
75 Ω	0.3-0.7 V	50m (coaxial cable)	BNC

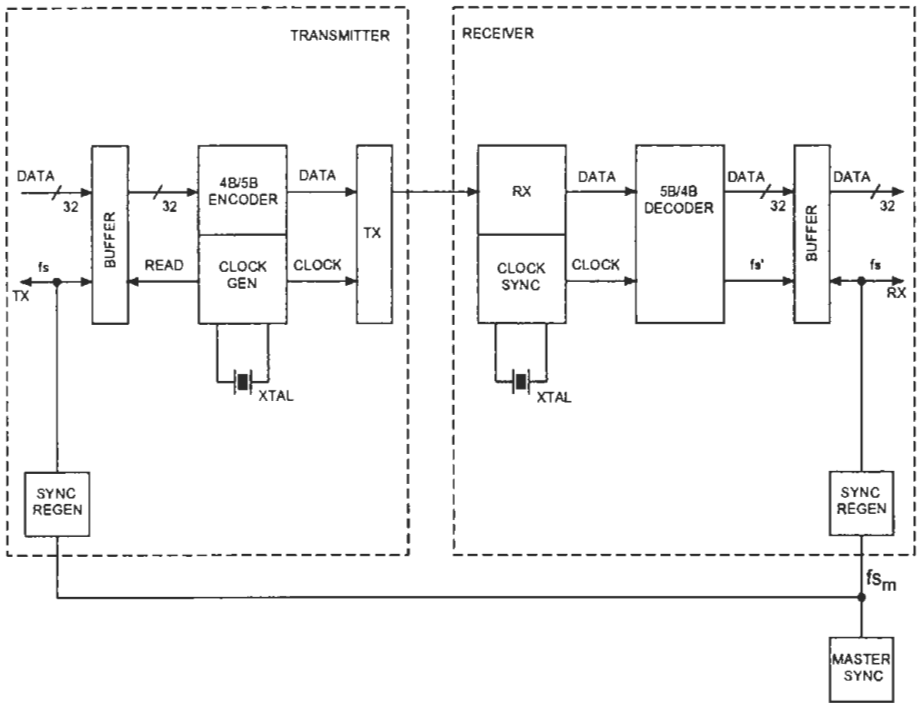


Figure 4.11 MADI link.

4.3 Single-processor Systems

4.3.1 Peripherals

A common system configuration is shown in Fig. 4.12. It consists of a DSP, clock generation, instruction and data memory and a BOOT-EPROM. After RESET, the program is loaded into the internal RAM of the signal processor. The loading is done byte by byte so that only an EPROM with 8 bit data word-length is necessary. In terms of circuit complexity the connection of AD/DA converters

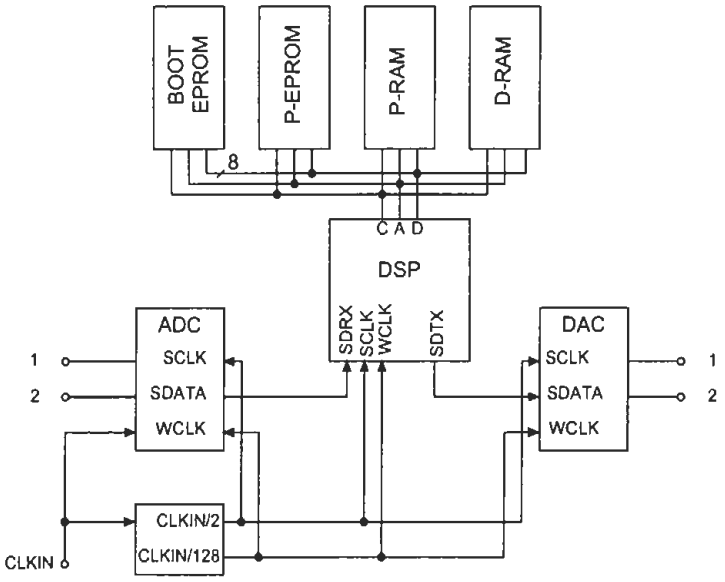


Figure 4.12 DSP system with two-channel AD/DA converters (C = control, A = address, D = data, SDATA = serial data, SCLK = bit clock, WCLK = word clock, SDRX = serial input, SDTX = serial output).

over serial interfaces is the simplest solution. Most fixed-point signal processors support serial connection where a lead for bit clock SCLK, sampling clock/word clock WCLK, and the serial input and output data SDRX/SDTX are used. The clock signals are obtained from a higher reference clock CLKIN (see Fig. 4.13). For non-serially operating AD/DA converters, parallel interfaces can also be connected to the DSP.

4.3.2 Control

For controlling digital signal processors and data exchange with host processors, some DSPs provide a special host interface that can be read and written directly

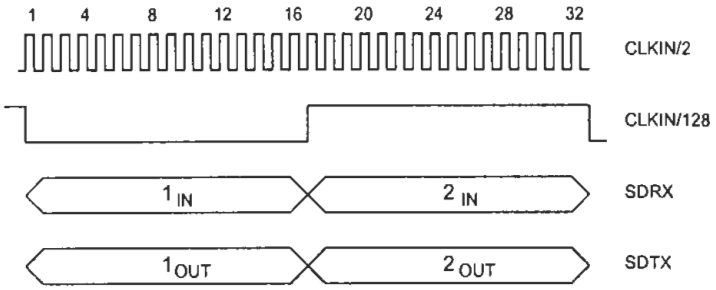


Figure 4.13 Serial transmission format.

(see Fig. 4.14). The data word-length depends on the processor. The host interface is included in the external address space of the host or is connected to a local bus system, for instance a PC bus.

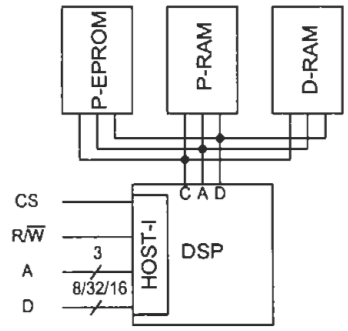


Figure 4.14 Control via a host interface of the DSP (CS = chip select, R/$\overline{W}$ = read/write, A = address, D = data).

A DSP as a coprocessor for special signal processing problems can be used by connecting it with a dual-port RAM and additional interrupt logic to a host processor. This enables data transmission between the DSP system and host processor(see Fig. 4.15). This results in a complete separation from the host processor. The communication can either be interrupt-controlled or carried out by polling a memory address in a dual-port RAM.

A very simple control can be done directly via an RS232-interface. This is can be carried out via an additional asynchronous serial interface (Serial Communication Interface) of the DSP (see Fig. 4.16).

4.4 Multiprocessor Systems

The design of multiprocessor systems can be carried out by linking signal processors by serial or parallel interfaces. Besides purely multiprocessor DSP systems,

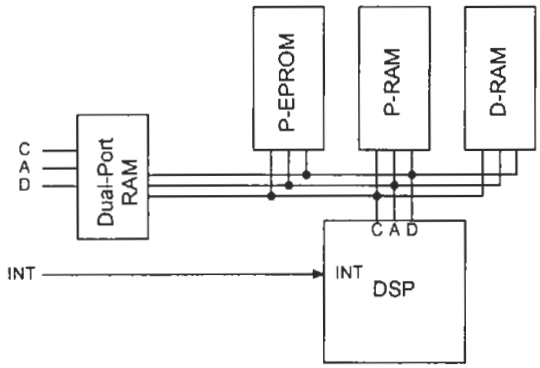


Figure 4.15 Control over a dual-port RAM and interrupt.

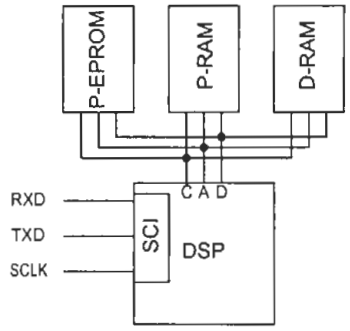


Figure 4.16 Control over a serial interface (RS232, RS422).

an additional connection to standard bus systems can be made as well.

4.4.1 Connection via Serial Links

In connecting via serial links, signal processors are cascaded so that different program segments are distributed over different processors (see Fig. 4.17). The serial output data is fed into the serial input of the following signal processor. A synchronous bit clock and a common synchronization SYNC control the serial interface. With the help of a serial time-multiplex mode (Fig. 4.18) a parallel configuration

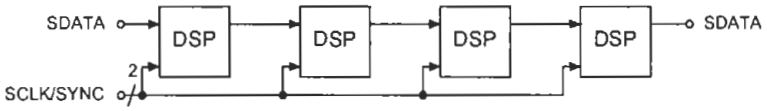


Figure 4.17 Cascading and pipelining (SDATA = serial data, SCLK = bit clock, SYNC = synchronization).

can be designed which, for instance, feeds several parallel signal processors with

serial input data. The serial outputs of signal processors provide output data in time-multiplex. A complete time-multiplex connection via the serial interface of

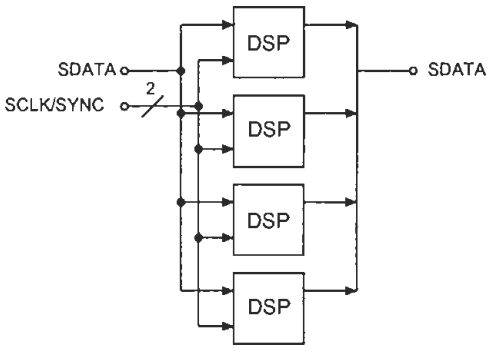


Figure 4.18 Parallel configuration with output time-multiplex.

the signal processor is shown in Fig. 4.19. The allocation of a signal processor at a particular time slot can either be fixed or carried out by an address control ADR.

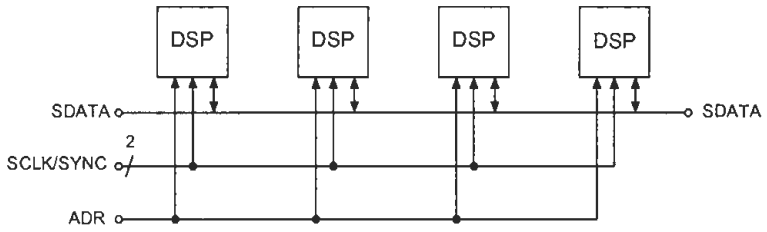


Figure 4.19 Time-multiplex connection (ADR = address at a particular time).

4.4.2 Connection via Parallel Links

The connection via parallel links is possible with dual-port processors as well as with dual-port RAMs (see Fig. 4.20). A parallel configuration of signal processor

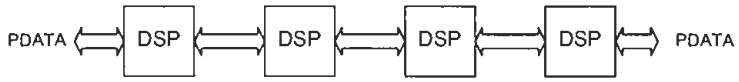


Figure 4.20 Cascading and pipelining.

systems with a local bus is shown in Fig. 4.21. The connection to the local bus is done either over a dual-port RAM or directly with a second signal processor port. Another possible configuration is the use of a 4-port RAM as shown in Fig. 4.22. Here, one processor serves as a connector to a system bus and feeds three other processors over a 4-port RAM with control and data information.

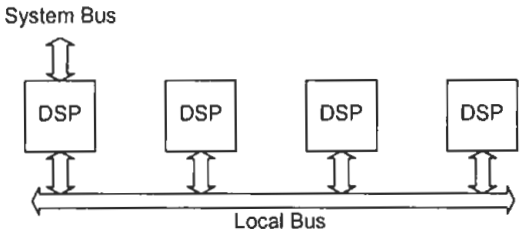


Figure 4.21 Parallel configuration.

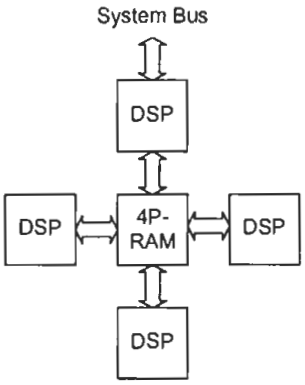


Figure 4.22 Connection over a 4-port RAM.

4.4.3 Connection via Standard Bus Systems

The use of standard bus systems (VME bus, MULTIBUS, PC bus) to control multiprocessor systems is presented in Fig. 4.23. The connection of signal processors can either be carried out directly over a control bus or with the help of a special data bus. This parallel data bus can operate in time-multiplex. Hence control information and data are separated. A few of the criteria for standard bus systems

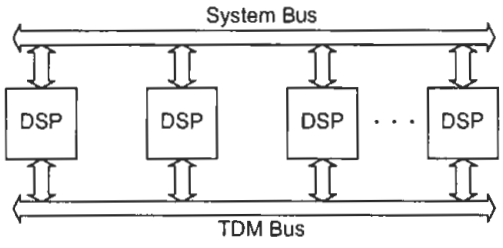


Figure 4.23 Signal processor systems based on standard bus system.

are data transfer rate, interrupt request and processing, the option of several masters, auxiliary functions (power supply, bus error, battery buffer) and mechanical requirements.

4.4.4 Scalable Audio System

The functional segmentation of an audio system into different stages, the analog, interface, digital and man-machine stages, is shown in Fig. 4.24. All stages are con-

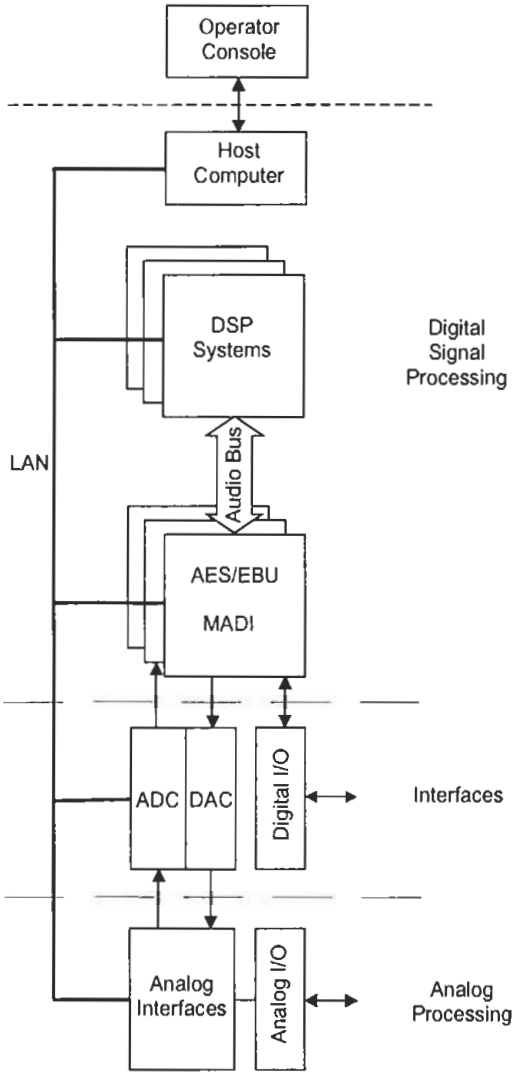


Figure 4.24 Audio system.

trolled by a LAN (Local Area Network). In the analog domain, crosspoint switches and microphone amplifiers are controlled. In the interface domain AD/DA converters and sampling rate converters are used. The connection to a signal processing system is done by AES/EBU and MADI interfaces. A host computer with a control console for the sound engineer serves as the central control unit.

The realization of the digital domain with the help of a standard bus system is shown in Fig. 4.25. A central mixing console controls several subsystems over

a host. These subsystems have special control computers which control several DSP modules. The system concept is scalable (extendable in modules) within a subsystem and by extension to several subsystems. Audio data transfer between subsystems is performed by AES/EBU and MADI interfaces. The segmentation within a subsystem is shown in Fig. 4.26. Here, besides DSP modules, digital interfaces (AES/EBU, MADI, sampling rate converters, etc.) and AD/DA converters can be integrated.

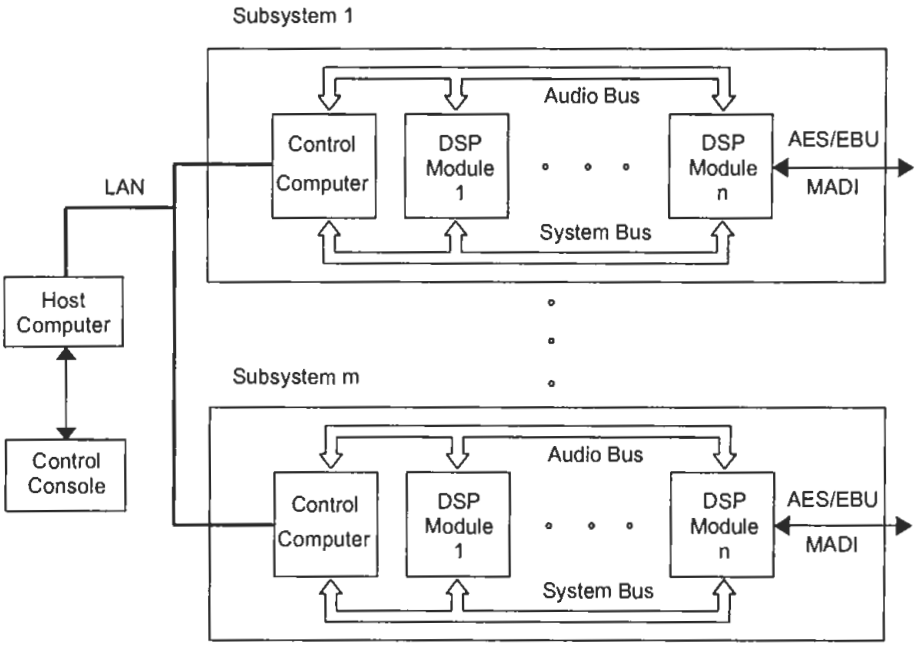


Figure 4.25 Scalable digital audio system.

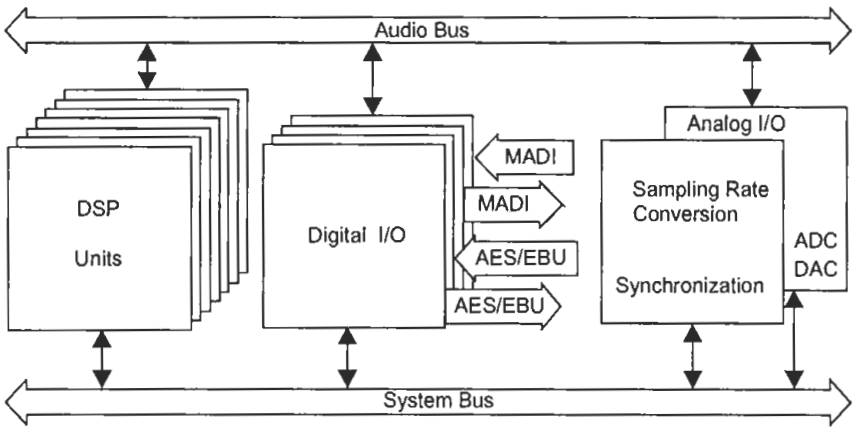


Figure 4.26 Subsystem.

Chapter 5

Equalizers

Spectral sound equalization is one of the most important methods for processing audio signals. Equalizers are found in various forms in the transmission of audio signals from a sound studio to the listener. The more complex filter functions are used in sound studios. But in almost every consumer product like car radios, hifi-amplifiers etc., simple filter functions are used for sound equalization. The first section of this chapter discusses the design and the implementation of recursive audio filters. In the second and third sections, linear phase nonrecursive filter structures and their implementation are introduced.

5.1 Recursive Audio Filters

5.1.1 Design

A certain filter response can be approximated by two kinds of transfer function. On the one hand, the combination of poles and zeros leads to a very low-order transfer function $H(z)$ in fractional form, which solves the given approximation problem. The digital implementation of this transfer function needs recursive procedures owing to its poles. On the other hand, the approximation problem can be solved by placing only zeros in the z-plane. This transfer function $H(z)$ has, besides its zeros, a corresponding number of poles at the origin of the z-plane. The order of this transfer function, for same approximation conditions, is substantially higher than for transfer functions consisting of poles and zeros. In view of an economical implementation of a filter algorithm in terms of complexity, recursive filters achieve shorter computing time owing to their lower order. For a sampling

rate of 48 kHz, the algorithm has 20.83 μs processing time available. With the DSPs presently available it is easily possible to implement recursive digital filters for audio applications within this sampling period using only one DSP. Starting with the design of typical equalizers in the S-domain, these filters will be mapped to the Z-domain by the bilinear transformation.

Low-pass/High-pass Filters. In order to limit the audio spectrum, low-pass and high-pass filters with Butterworth response are used in analog mixers. They offer a monotonic pass-band and a monotonically decreasing stop-band attenuation per octave ($n \cdot 6$ dB/oct.) that is determined by the filter order. Low-pass filters of the second and fourth order are commonly used. The normalized second-order low-pass transfer function is given by

$$H_{LP}(s) = \frac{1}{s^2 + \frac{1}{Q_\infty}s + 1}. \tag{5.1}$$

The Q-factor Q_∞ of a Butterworth approximation is equal to $1/\sqrt{2}$. The corresponding second-order high-pass transfer function

$$H_{HP}(s) = \frac{s^2}{s^2 + \frac{1}{Q_\infty}s + 1} \tag{5.2}$$

is obtained by a low-pass to high-pass transformation. Figure 5.1 shows the pole-zero locations in the s-plane. The amplitude frequency response of a high-pass

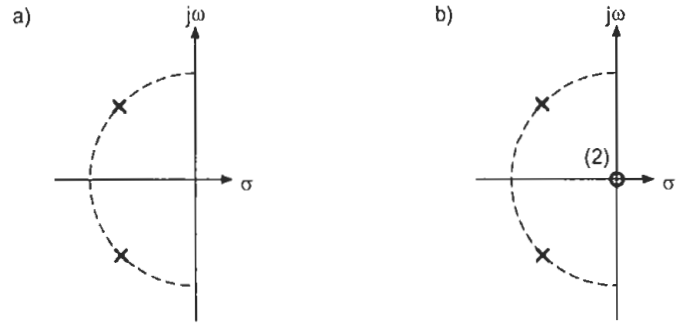


Figure 5.1 Pole-zero location for a) second-order low-pass and b) second-order high-pass.

filter with a 3 dB cutoff frequency of 50 Hz and a low-pass filter with a 3 dB cutoff frequency of 5000 Hz are shown in Fig. 5.2. Second- and fourth-order filters are shown.

Table 5.1 summarizes the transfer functions of low-pass and high-pass filters with Butterworth response.

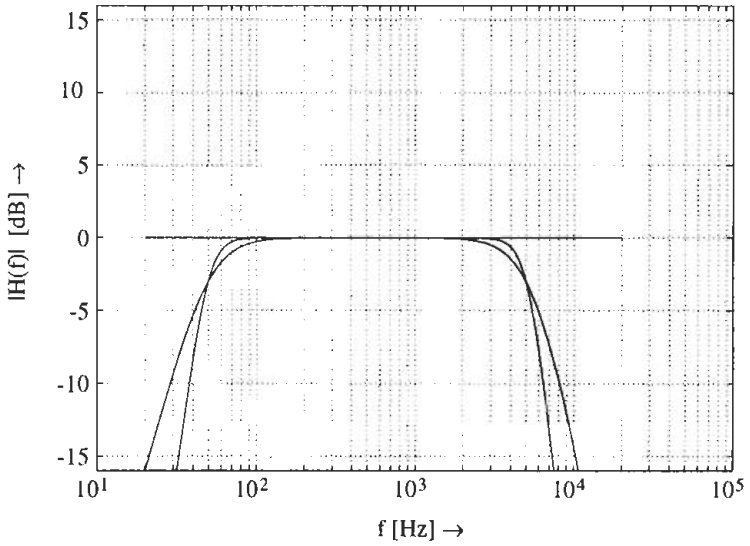


Figure 5.2 Frequency response of low-pass and high-pass filters - high-pass $f_c = 50$ Hz (second-/fourth-order), low-pass $f_c = 5000$ Hz (second-/fourth-order).

Table 5.1 Transfer functions of low-pass and high-pass filters.

Low-pass	$H(s) = \frac{1}{s^2+\sqrt{2}s+1}$	second-order
	$H(s) = \frac{1}{(s^2+1.848s+1)(s^2+0.765s+1)}$	fourth-order
High-pass	$H(s) = \frac{s^2}{s^2+\sqrt{2}s+1}$	second-order
	$H(s) = \frac{s^4}{(s^2+1.848s+1)(s^2+0.765s+1)}$	fourth-order

Shelving Filters. Besides the purely band-limiting filters like low-pass and high-pass filters shelving filters are used to perform weighting of certain frequencies. A simple approach for a first-order shelving filter is given by

$$H(s) = 1 + \frac{H_0}{s+1}. \tag{5.3}$$

It consists of a first-order low-pass filter with dc amplification of H_0 connected in parallel with an all-pass system of transfer function $H(s) = 1$. Equation (5.3) can

be written as

$$H(s) = \frac{s + (1 + H_0)}{s + 1} = \frac{s + V_0}{s + 1} \tag{5.4}$$

where V_0 determines the amplification at $\omega = 0$. By changing the parameter V_0, any desired boost ($V_0 > 1$) and cut ($V_0 < 1$) level can be adjusted. Fig. 5.3 shows the asymptotes of the frequency response. For $V_0 < 1$, the cutoff frequency ω_c is moved to lower frequencies.

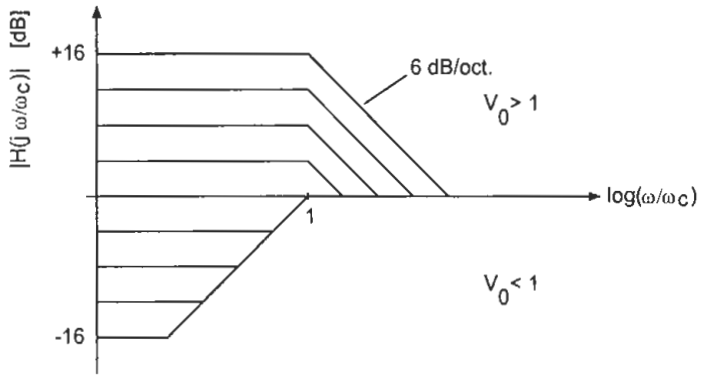


Figure 5.3 Asymptotes of the frequency response with transfer function (5.4).

In order to obtain a symmetrical amplitude frequency response relative to the frequency axis without changing the cutoff frequency, it is necessary to invert the transfer function (5.4) in the case of cut ($V_0 < 1$). This has the effect of swapping poles with zeros and leads to the transfer function

$$H(s) = \frac{s + 1}{s + V_0} \tag{5.5}$$

for cut (Fig. 5.4).

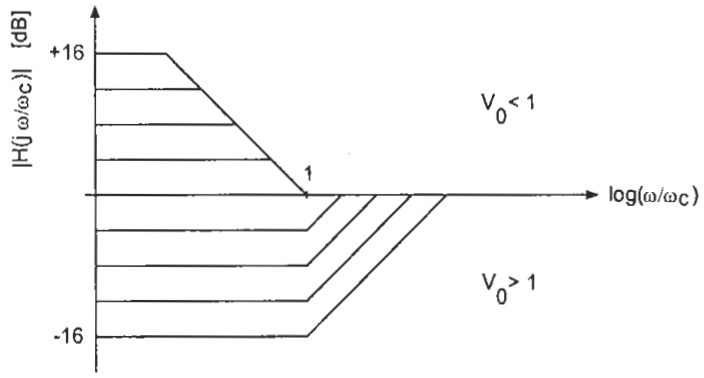


Figure 5.4 Asymptotes of the frequency response with transfer function (5.5).

Figure 5.5 shows the locations of poles and zeros for both the boost and the cut case. By moving zeros and poles on the negative σ-axis, boost and cut can be

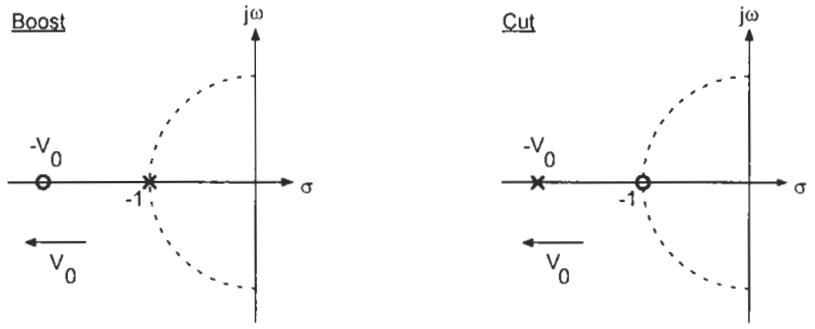


Figure 5.5 Pole-zero locations of a first-order low-frequency shelving filter.

adjusted. The equivalent shelving filter for high frequencies is obtained by means of a low-pass to high-pass transformation. In the case of boost, the transfer function is given by

$$H(s) = \frac{sV_0 + 1}{s + 1}, V_0 > 1 \tag{5.6}$$

and for cut we get

$$H(s) = \frac{s + 1}{sV_0 + 1}, V_0 > 1. \tag{5.7}$$

The parameter V_0 determines the value of the transfer function $H(s)$ at $\omega = \infty$ for high-frequency shelving filters.

In order to increase the slope of the filter response in the transition band, a general second-order transfer function

$$H(s) = \frac{a_2 s^2 + a_1 s + a_0}{s^2 + \sqrt{2}s + 1} \tag{5.8}$$

is considered, in which complex zeros are added to the complex poles. The calculation of poles leads to

$$s_{\infty\,1/2} = \sqrt{\frac{1}{2}}(-1 \pm j). \tag{5.9}$$

If the complex zeros

$$s_{0\,1/2} = \sqrt{\frac{V_0}{2}}(-1 \pm j) \tag{5.10}$$

are moved on a straight line with the help of the parameter V_0 (see Fig. 5.6), the transfer function

$$H(s) = \frac{s^2 + \sqrt{2V_0}s + V_0}{s^2 + \sqrt{2}s + 1} \tag{5.11}$$

of a second-order low-frequency shelving filter is obtained. The parameter V_0 determines the boost for low frequencies. The cut case can be achieved by inversion of Equation (5.11).

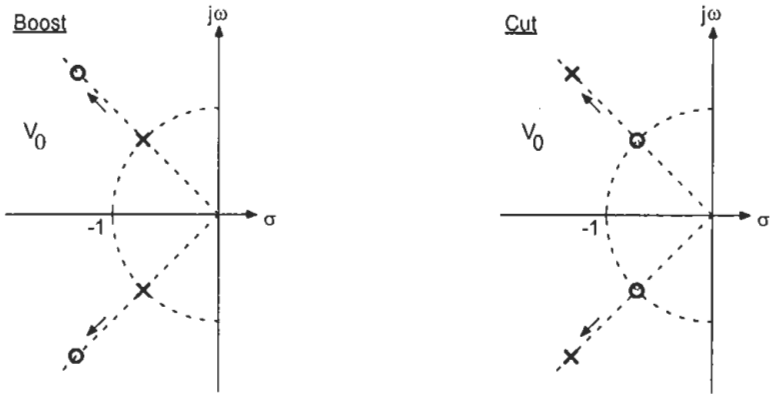


Figure 5.6 Pole-zero locations of a second-order low-frequency shelving filter.

A low-pass to high-pass transformation of (5.11) provides the transfer function

$$H(s) = \frac{V_0 s^2 + \sqrt{2V_0}s + 1}{s^2 + \sqrt{2}s + 1} \tag{5.12}$$

of a second-order high-frequency shelving filter. The zeros

$$s_{o\,1/2} = \sqrt{\frac{1}{2V_0}}(-1 \pm j) \tag{5.13}$$

are moved on a straight line towards the origin with increasing V_0 (see Fig. 5.7). The cut case is obtained by inverting the transfer function (5.12). Figure 5.8 shows the amplitude frequency response of a low-frequency shelving filter with cutoff frequency 100 Hz and a high-frequency shelving filter with cutoff frequency 5000 Hz (parameter V_0).

Peak Filter. Another equalizer used for boosting or cutting any desired frequency is the peak filter. With the help of a second-order band-pass transfer function

$$H_{BP}(s) = \frac{(H_0/Q_\infty)s}{s^2 + \frac{1}{Q_\infty}s + 1} \tag{5.14}$$

the transfer function

$$\begin{aligned} H(s) &= 1 + H_{BP}(s) \\ &= \frac{s^2 + \frac{1+H_0}{Q_\infty}s + 1}{s^2 + \frac{1}{Q_\infty}s + 1} \end{aligned}$$

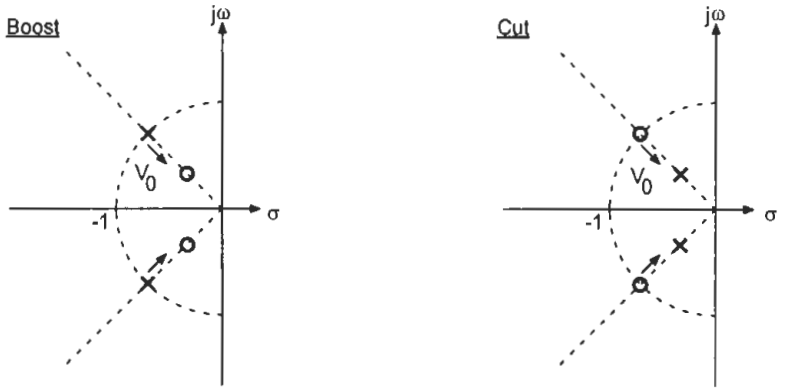


Figure 5.7 Pole-zero locations of second-order high-frequency shelving filter.

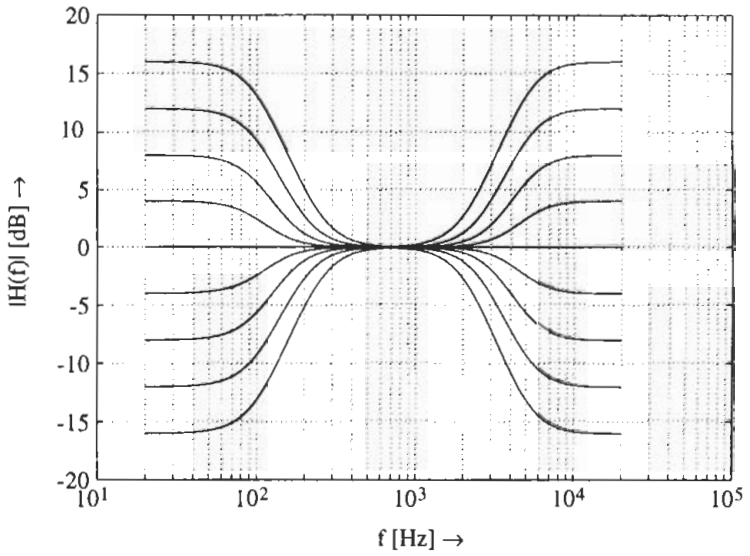


Figure 5.8 Frequency responses of low-/high-frequency shelving filters - low-frequency shelving filter $f_c = 100$ Hz (parameter V_0), high-frequency shelving filter $f_c = 5000$ Hz (parameter V_0).

$$= \frac{s^2 + \frac{V_0}{Q_\infty}s + 1}{s^2 + \frac{1}{Q_\infty}s + 1} \tag{5.15}$$

of a peak filter can be derived. It can be shown that the maximum of the amplitude frequency response at the center frequency is determined by the parameter V_0. The relative bandwidth is fixed by the Q-factor. The geometrical symmetry of the frequency response relative to the center frequency remains constant for the transfer function of a peak filter (5.15). The poles and zeros lie on the unit circle. By adjusting the parameter V_0, the complex zeros are moved with respect to the

complex poles. Figure 5.9 shows this for the boost and cut cases. With increasing Q-factor, the complex poles move towards the $j\omega$-axis on the unit circle.

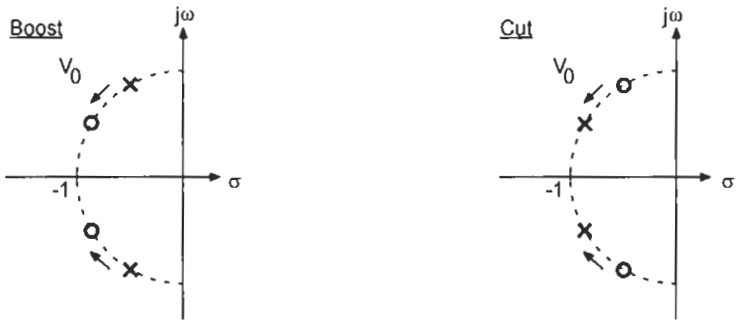


Figure 5.9 Pole-zero locations of a second-order peak filter.

Figure 5.10 shows the amplitude frequency response of a peak filter by changing the parameter V_0 at a center frequency of 500 Hz and a Q-factor of 1.25. Figure 5.11 shows the variation of the Q-factor Q_∞ at a center frequency of 500 Hz, a boost/cut of ±16 dB and Q-factor of 1.25.

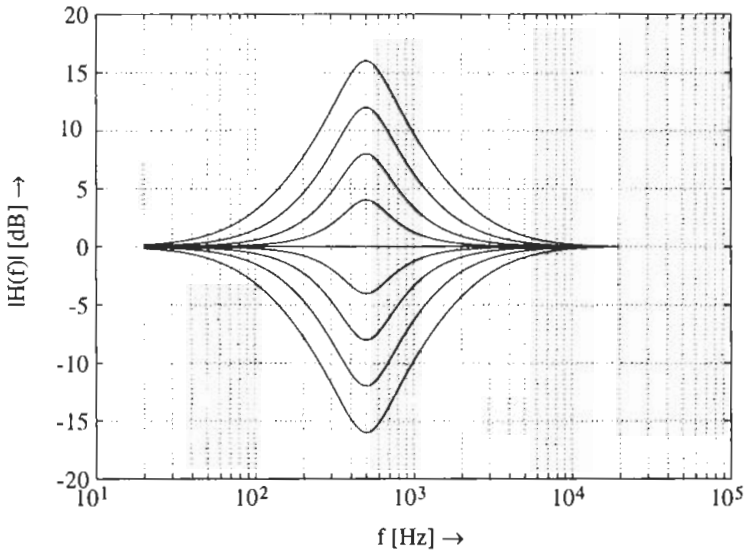


Figure 5.10 Frequency response of a peak filter - $f_c = 500$ Hz, $Q_\infty = 1.25$, cut parameter V_0.

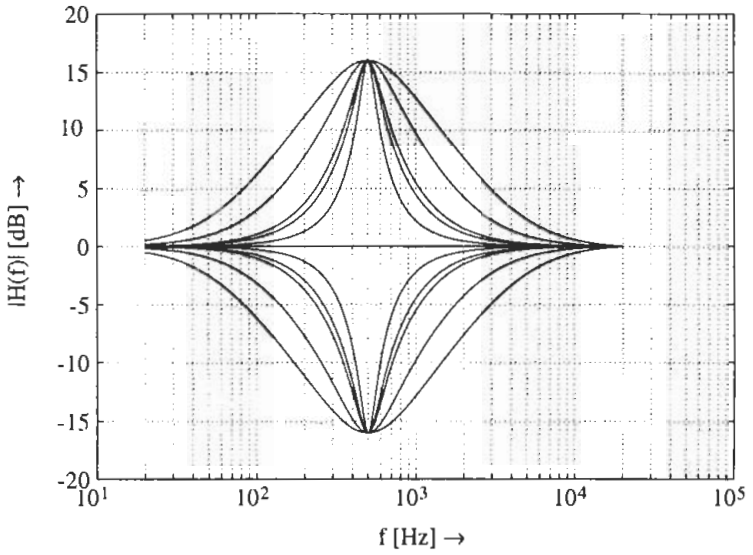


Figure 5.11 Frequency responses of peak filters - f_c = 500Hz, boost/cut ±16 dB Q_∞ = 0.707, 1.25, 2.5, 3, 5.

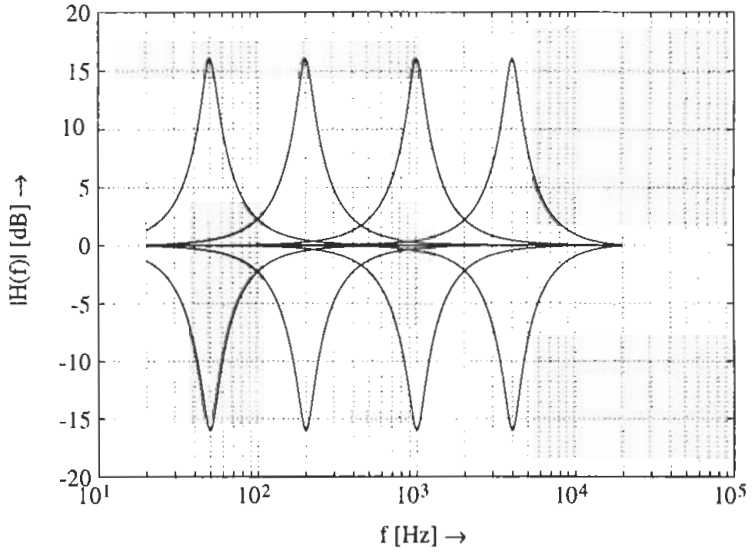


Figure 5.12 Frequency responses of peak filters - boost/cut ±16 dB, Q_∞ = 1.25 f_c = 50, 200, 1000, 4000 Hz.

Mapping to Z-domain. In order to implement a digital filter, the filter designed in the S-domain with transfer function $H(s)$ is converted to the Z-domain with the help of a suitable transformation to obtain the transfer function $H(z)$. The impulse-invariant transformation is not suitable as it leads to overlapping ef-

fects if the transfer function $H(s)$ is not band-limited to half the sampling rate. An independent mapping of poles and zeros from the S-domain into poles and zeros in the Z-domain is possible with help of the bilinear transformation given by

$$s = \frac{2}{T}\frac{z-1}{z+1}. \tag{5.16}$$

Tables 5.2, 5.3 and 5.4 contain the coefficients of the second-order transfer function

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}, \tag{5.17}$$

which are determined by the bilinear transformation and the auxiliary variable $K = \tan(\omega_c T/2)$ for different filter types. Strategies for time-variant switching of audio filters can be found in [Zöl93].

Table 5.2 Low-pass/high-pass filter design.

low-pass (second-order)				
a_0	a_1	a_2	b_1	b_2
$\frac{K^2}{1+\sqrt{2}K+K^2}$	$\frac{2K^2}{1+\sqrt{2}K+K^2}$	$\frac{K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2}K+K^2}$
high-pass (second-order)				
a_0	a_1	a_2	b_1	b_2
$\frac{1}{1+\sqrt{2}K+K^2}$	$\frac{-2}{1+\sqrt{2}K+K^2}$	$\frac{1}{1+\sqrt{2}K+K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2}K+K^2}$

Table 5.3 Peak filter design.

peak (boost $V_0 = 10^{G/20}$)				
a_0	a_1	a_2	b_1	b_2
$\frac{1+\frac{V_0}{Q_\infty}K+K^2}{1+\frac{1}{Q_\infty}K+K^2}$	$\frac{2(K^2-1)}{1+\frac{1}{Q_\infty}K+K^2}$	$\frac{1-\frac{V_0}{Q_\infty}K+K^2}{1+\frac{1}{Q_\infty}K+K^2}$	$\frac{2(K^2-1)}{1+\frac{1}{Q_\infty}K+K^2}$	$\frac{1-\frac{1}{Q_\infty}K+K^2}{1+\frac{1}{Q_\infty}K+K^2}$
peak (cut $V_0 = 10^{-G/20}$)				
a_0	a_1	a_2	b_1	b_2
$\frac{1+\frac{1}{Q_\infty}K+K^2}{1+\frac{V_0}{Q_\infty}K+K^2}$	$\frac{2(K^2-1)}{1+\frac{V_0}{Q_\infty}K+K^2}$	$\frac{1-\frac{1}{Q_\infty}K+K^2}{1+\frac{V_0}{Q_\infty}K+K^2}$	$\frac{2(K^2-1)}{1+\frac{V_0}{Q_\infty}K+K^2}$	$\frac{1-\frac{V_0}{Q_\infty}K+K^2}{1+\frac{V_0}{Q_\infty}K+K^2}$

Table 5.4 Shelving filter design.

low-frequency shelving (boost $V_0 = 10^{G/20}$)				
a_0	a_1	a_2	b_1	b_2
$\frac{1+\sqrt{2V_0}K+V_0K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(V_0K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2V_0}K+V_0K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2}K+K^2}$
low-frequency shelving (cut $V_0 = 10^{-G/20}$)				
a_0	a_1	a_2	b_1	b_2
$\frac{1+\sqrt{2}K+K^2}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{2(V_0K^2-1)}{1+\sqrt{2V_0}K+V_0K^2}$	$\frac{1-\sqrt{2V_0}K+V_0K^2}{1+\sqrt{2V_0}K+V_0K^2}$
high-frequency shelving (boost $V_0 = 10^{G/20}$)				
a_0	a_1	a_2	b_1	b_2
$\frac{V_0+\sqrt{2V_0}K+K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(K^2-V_0)}{1+\sqrt{2}K+K^2}$	$\frac{V_0-\sqrt{2V_0}K+K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2}K+K^2}$
high-frequency shelving (cut $V_0 = 10^{-G/20}$)				
a_0	a_1	a_2	b_1	b_2
$\frac{1+\sqrt{2}K+K^2}{V_0+\sqrt{2V_0}K+K^2}$	$\frac{2(K^2-1)}{V_0+\sqrt{2V_0}K+K^2}$	$\frac{1-\sqrt{2}K+K^2}{V_0+\sqrt{2V_0}K+K^2}$	$\frac{2(K^2/V_0-1)}{1+\sqrt{2/V_0}K+K^2/V_0}$	$\frac{1-\sqrt{2/V_0}K+K^2/V_0}{1+\sqrt{2/V_0}K+K^2/V_0}$

5.1.2 Parametric Filter Structures

Parametric filter structures allow direct access to the parameters of the transfer function, like center/cutoff frequency, bandwidth and gain, via control of associated coefficients. To modify one of these parameters, it is therefore not necessary to compute a complete set of coefficients for a second-order transfer function, but instead only one coefficient in the filter structure is calculated.

An independent control of gain, cutoff/center frequency and bandwidth is achieved by a feed forward (FW) structure for *boost* and a feed backward (FB) structure for *cut* as shown in Fig. 5.13. The corresponding transfer functions are:

$$G_{FW}(z) = 1 + H_0H(z) \tag{5.18}$$

$$G_{FB}(z) = \frac{1}{1 + H_0H(z)}. \tag{5.19}$$

The boost/cut factor is $V_0 = 1 + H_0$. For digital filter implementations, it is necessary for the feed backward case that the inner transfer function be of the form

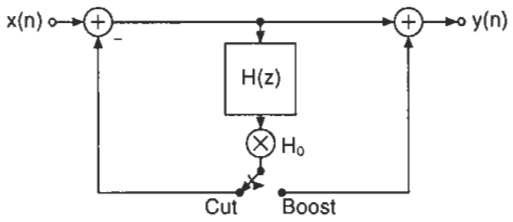


Figure 5.13 Filter structure for implementing boost and cut filters.

$H(z) = z^{-1}H_1(z)$ to ensure stability. A parametric filter structure proposed by Harris [Har93] is based on the feed forward/feed backward technique, but the frequency response shows slight deviations near $z = 1$ and $z = -1$ from the desired one. This is due to the z^{-1} in the FF/FB branch. It is possible to implement typical audio filters with only a feed forward structure. The complete decoupling of the control parameters is possible for the boost case, but there remains a coupling between bandwidth and gain factor for the cut case. In the following, two approaches for parametric audio filter structures based on an all-pass decomposition of the transfer function will be discussed.

Regalia filter [Reg87]. The denormalized transfer function of a first-order shelving filter is given by

$$H(s) = \frac{s + V_0\omega_c}{s + \omega_c} \tag{5.20}$$

with

$$\begin{aligned} H(0) &= V_0 \\ H(\infty) &= 1. \end{aligned}$$

A decomposition of (5.20) leads to

$$H(s) = \frac{s}{s + \omega_c} + V_0 \frac{\omega_c}{s + \omega_c}. \tag{5.21}$$

The low-pass and high-pass transfer functions in (5.21) can be expressed by an all-pass decomposition of the form

$$\frac{s}{s + \omega_c} = \frac{1}{2}\left[1 + \frac{s - \omega_c}{s + \omega_c}\right] \tag{5.22}$$

$$\frac{V_0\omega_c}{s + \omega_c} = \frac{V_0}{2}\left[1 - \frac{s - \omega_c}{s + \omega_c}\right]. \tag{5.23}$$

With the all-pass transfer function

$$A_B(s) = \frac{s - \omega_c}{s + \omega_c} \tag{5.24}$$

for *boost*, (5.20) can be rewritten as

$$H(s) = \frac{1}{2}[1 + A_B(s)] + \frac{1}{2}V_0[1 - A_B(s)]. \tag{5.25}$$

The bilinear transformation $s = \frac{2}{T}\frac{z-1}{z+1}$ leads to

$$H(z) = \frac{1}{2}[1 + A_B(z)] + \frac{1}{2}V_0[1 - A_B(z)] \tag{5.26}$$

with

$$A_B(z) = -\frac{z^{-1} + a_B}{1 + a_B z^{-1}} \tag{5.27}$$

and the frequency parameter

$$a_B = \frac{\tan(\omega_c T/2) - 1}{\tan(\omega_c T/2) + 1}. \tag{5.28}$$

A filter structure for direct implementation of (5.26) is presented in Fig. 5.14a. Other possible structures can be seen in Fig. 5.14b,c. For the cut case $V_0 < 1$, the cutoff frequency of the filter moves towards lower frequencies [Reg87].

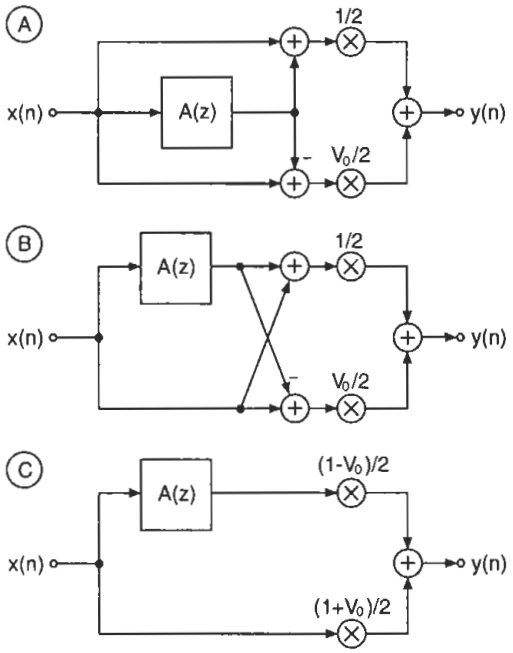


Figure 5.14 Filter structures by Regalia.

In order to retain the cutoff frequency for the cut case [Zöl95], the denormalized transfer function of a first-order shelving filter (cut)

$$H(s) = \frac{s + \omega_c}{s + \omega_c/V_0} \tag{5.29}$$

with the boundary conditions

$$\begin{aligned} H(0) &= V_0 \\ H(\infty) &= 1 \end{aligned}$$

can be decomposed as follows:

$$H(s) = \frac{s}{s + \omega_c/V_0} + \frac{\omega_c}{s + \omega_c/V_0}. \tag{5.30}$$

With the all-pass decompositions

$$\frac{s}{s + \omega_c/V_0} = \frac{1}{2}\left[1 + \frac{s - \omega_c/V_0}{s + \omega_c/V_0}\right] \tag{5.31}$$

$$\frac{\omega_c}{s + \omega_c/V_0} = \frac{V_0}{2}\left[1 - \frac{s - \omega_c/V_0}{s + \omega_c/V_0}\right] \tag{5.32}$$

and the all-pass transfer function

$$A_C(s) = \frac{s - \omega_c/V_0}{s + \omega_c/V_0} \tag{5.33}$$

for *cut*, (5.29) can be rewritten as

$$H(s) = \frac{1}{2}[1 + A_C(s)] + \frac{V_0}{2}[1 - A_C(s)]. \tag{5.34}$$

The bilinear transformation leads to

$$H(z) = \frac{1}{2}[1 + A_C(z)] + \frac{V_0}{2}[1 - A_C(z)] \tag{5.35}$$

with

$$A_C(z) = -\frac{z^{-1} + a_C}{1 + a_C z^{-1}} \tag{5.36}$$

and the frequency parameter

$$a_C = \frac{\tan(\omega_c T/2) - V_0}{\tan(\omega_c T/2) + V_0}. \tag{5.37}$$

Due to (5.35) and (5.26), boost and cut can be implemented with the same filter structure (see Fig. 5.14). However, it has to be noted that the frequency parameter a_C as in (5.37) for cut depends on the cutoff frequency and gain.

A second-order peak filter is obtained by a low-pass to high-pass transformation according to

$$z^{-1} \rightarrow -z^{-1}\frac{z^{-1}+d}{1+dz^{-1}}. \tag{5.38}$$

For an all-pass as given in (5.27) and (5.36), the second-order all-pass is given by

$$A_{BC}(z) = \frac{z^{-2}+d(1+a_{BC})z^{-1}+a_{BC}}{1+d(1+a_{BC})z^{-1}+a_{BC}z^{-2}} \tag{5.39}$$

with parameters (cut as in [Zöl95])

$$\begin{aligned} d &= -cos(\Omega_c) && (5.40)\\ V_0 &= H(e^{j\Omega_c}) && (5.41)\\ a_B &= \frac{1-\tan(\omega_b T/2)}{1+\tan(\omega_b T/2)} && (5.42)\\ a_C &= \frac{V_0-\tan(\omega_b T/2)}{V_0+\tan(\omega_b T/2)}. && (5.43) \end{aligned}$$

The center frequency f_c is fixed by the parameter d, the bandwidth f_b by the parameters a_B and a_C and gain by the parameter V_0.

Simplified All-pass Decomposition [Zöl95]. The transfer function of a first-order low-frequency shelving filter can be a decomposed as

$$\begin{aligned} H(s) &= \frac{s+V_0\omega_c}{s+\omega_c}\\ &= 1+H_0\frac{\omega_c}{s+\omega_c} && (5.44)\\ &= 1+\frac{H_0}{2}\left[1-\frac{s-\omega_c}{s+\omega_c}\right] && (5.45) \end{aligned}$$

with

$$\begin{aligned} V_0 &= H(s=0) && (5.46)\\ H_0 &= V_0-1 && (5.47)\\ V_0 &= 10^{\frac{G}{20}} \qquad \text{(G [dB])}. && (5.48) \end{aligned}$$

The transfer function (5.45) is composed of a direct branch and a low-pass filter. The first-order low-pass filter is again implemented by an all-pass decomposition. Applying the bilinear transformation to (5.45) leads to

$$H(z) = 1+\frac{H_0}{2}\left[1-A(z)\right] \tag{5.49}$$

with

$$A(z) = -\frac{z^{-1} + a_B}{1 + a_B z^{-1}}. \tag{5.50}$$

For cut, the following decomposition can be derived:

$$\begin{aligned} H(s) &= \frac{s + \omega_c}{s + \omega_c/V_0} && (5.51)\\ &= 1 + \underbrace{(V_0 - 1)}_{H_0} \frac{\omega_c/V_0}{s + \omega_c/V_0} && (5.52)\\ &= 1 + \frac{H_0}{2}\left[1 - \frac{s - \omega_c/V_0}{s + \omega_c/V_0}\right]. && (5.53) \end{aligned}$$

The bilinear transformation applied to (5.53) again gives (5.49). The filter structure is identical for boost and cut. The frequency parameter a_B for boost and a_C for cut can be calculated as

$$\begin{aligned} a_B &= \frac{\tan(\omega_c T/2) - 1}{\tan(\omega_c T/2) + 1} && (5.54)\\ a_C &= \frac{\tan(\omega_c T/2) - V_0}{\tan(\omega_c T/2) + V_0}. && (5.55) \end{aligned}$$

The transfer function of a first-order low-frequency shelving filter can be calculated as

$$H(z) = \frac{1 + (1 + a_{BC})\frac{H_0}{2} + (a_{BC} + (1 + a_{BC})\frac{H_0}{2})z^{-1}}{1 + a_{BC} z^{-1}}. \tag{5.56}$$

With $A_1(z) = -A(z)$ the signal flow chart in Fig. 5.15 shows a first-order low-pass filter and a first-order low-frequency shelving filter.

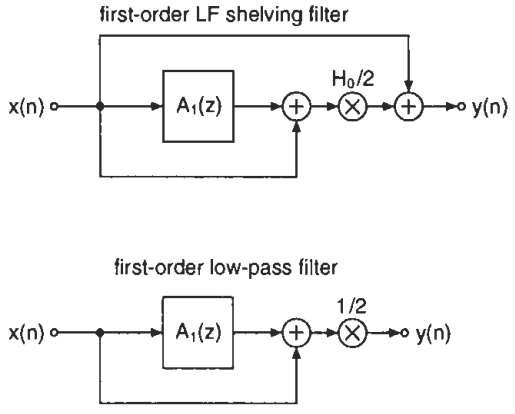


Figure 5.15 Low-frequency shelving filter and first-order low-pass filter.

The decomposition of a denormalized transfer function of a first-order high-frequency shelving filter can be given in the form of

$$\begin{aligned} H(s) &= \frac{sV_0 + \omega_c}{s + \omega_c} \\ &= 1 + H_0 \frac{s}{s + \omega_c} \qquad (5.57) \\ &= 1 + \frac{H_0}{2}\left[1 + \frac{s - \omega_c}{s + \omega_c}\right] \qquad (5.58) \end{aligned}$$

where

$$\begin{aligned} V_0 &= H(s = \infty) \qquad (5.59) \\ H_0 &= V_0 - 1. \qquad (5.60) \end{aligned}$$

The transfer function results by adding a high-pass filter to a constant. Applying the bilinear transformation to (5.58) gives

$$H(z) = 1 + \frac{H_0}{2}\left[1 + A(z)\right] \qquad (5.61)$$

with

$$A(z) = -\frac{z^{-1} + a_B}{1 + a_B z^{-1}}. \qquad (5.62)$$

For cut, the decomposition can be given by

$$\begin{aligned} H(s) &= \frac{s + \omega_c}{s/V_0 + \omega_c} \qquad (5.63) \\ &= 1 + \underbrace{(V_0 - 1)}_{H_0} \frac{s}{s + V_0\omega_c} \qquad (5.64) \\ &= 1 + \frac{H_0}{2}\left[1 + \frac{s - V_0\omega_c}{s + V_0\omega_c}\right] \qquad (5.65) \end{aligned}$$

which in return results in Equation (5.61) after a bilinear transformation. The boost and cut parameters can be calculated as

$$\begin{aligned} a_B &= \frac{\tan(\omega_c T/2) - 1}{\tan(\omega_c T/2) + 1} \qquad (5.66) \\ a_C &= \frac{V_0 \tan(\omega_c T/2) - 1}{V_0 \tan(\omega_c T/2) + 1}. \qquad (5.67) \end{aligned}$$

The transfer function of a first-order high-frequency shelving filter can then be written as

$$H(z) = \frac{1 + (1 - a_{BC})\frac{H_0}{2} + (a_{BC} + (a_{BC} - 1)\frac{H_0}{2})z^{-1}}{1 + a_{BC} z^{-1}}. \qquad (5.68)$$

With $A_1(z) = -A(z)$ the signal flow chart in Fig. 5.16 shows a first-order high-pass filter and a high-frequency shelving filter.

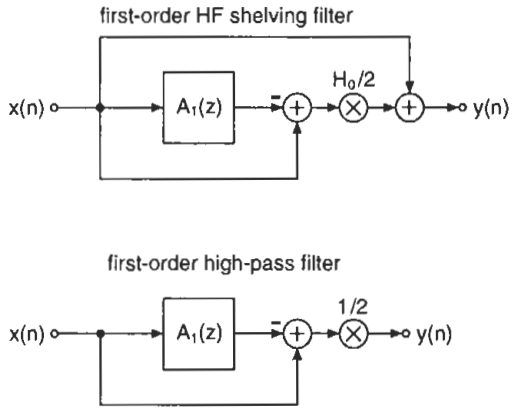


Figure 5.16 First-order high-frequency shelving and high-pass filters.

The implementation of a second-order peak filter can be carried out with a low-pass to high-pass transformation of a first-order shelving filter. But the addition of a second-order band-pass filter to a constant branch also results in a peak filter. With the help of an all-pass implementation of a band-pass filter as given by

$$H(z) = \frac{1}{2}[1 - A_2(z)] \tag{5.69}$$

and

$$A_2(z) = \frac{-a_B + (d - da_B)z^{-1} + z^{-2}}{1 + (d - da_B)z^{-1} - a_B z^{-2}} \tag{5.70}$$

a second-order peak filter can be expressed as

$$H(z) = 1 + \frac{H_0}{2}[1 - A_2(z)]\,. \tag{5.71}$$

The bandwidth parameters a_B and a_C for boost and cut are given

$$a_B = \frac{\tan(\omega_b T/2) - 1}{\tan(\omega_b T/2) + 1} \tag{5.72}$$

$$a_C = \frac{\tan(\omega_b T/2) - V_0}{\tan(\omega_b T/2) + V_0}. \tag{5.73}$$

The center frequency parameter d and the coefficient H_0 are given by

$$d = -cos(\Omega_c) \tag{5.74}$$

$$V_0 = H(e^{j\Omega_c}) \tag{5.75}$$

$$H_0 = V_0 - 1. \tag{5.76}$$

The transfer function of a second-order peak filter results in

$$H(z) = \frac{1 + (1 + a_{BC})\frac{H_0}{2} + d(1 - a_{BC})z^{-1} + (-a_{BC} - (1 + a_{BC})\frac{H_0}{2})z^{-2}}{1 + d(1 - a_{BC})z^{-1} - a_{BC}z^{-2}}. \tag{5.77}$$

The signal flow charts for a second-order peak filter and a second-order band-pass filter are shown in Fig. 5.17.

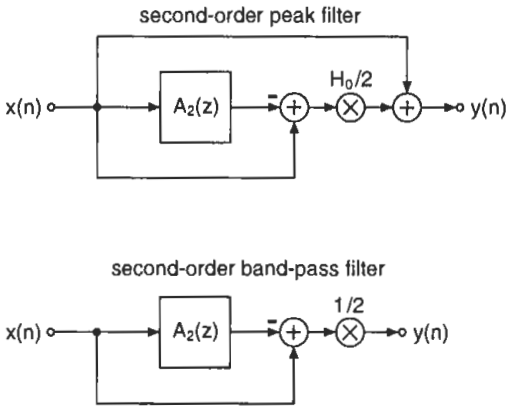


Figure 5.17 Second-order peak filter and band-pass filter.

The frequency responses for high-frequency, low-frequency shelving and peak filters are shown in Figs. 5.18, 5.19 and 5.20.

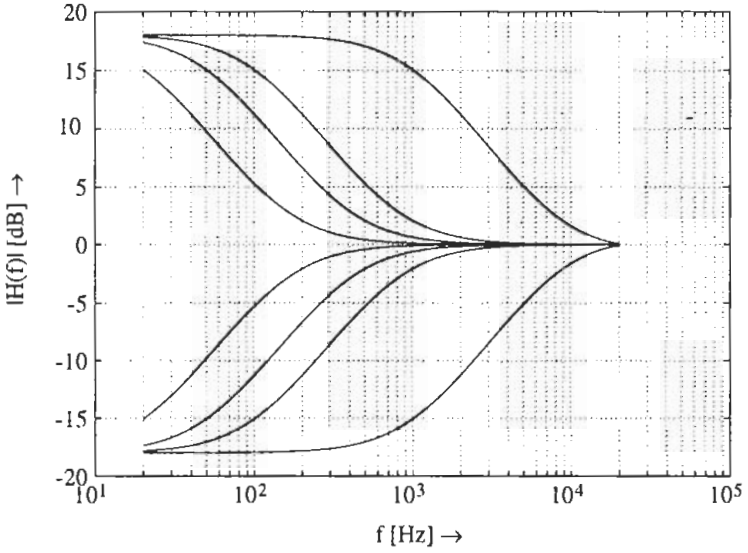


Figure 5.18 Low-frequency first-order shelving filter ($G = \pm 18$ dB - $f_c = 20{,}50{,}100{,}1000$ Hz).

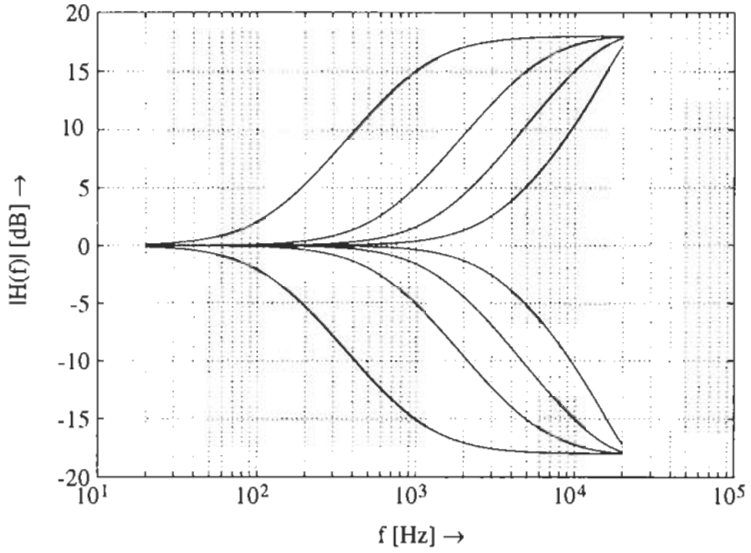


Figure 5.19 First-order high-frequency shelving filter ($G = \pm 18$ dB - f_c = 1,3,5,10,16 kHz).

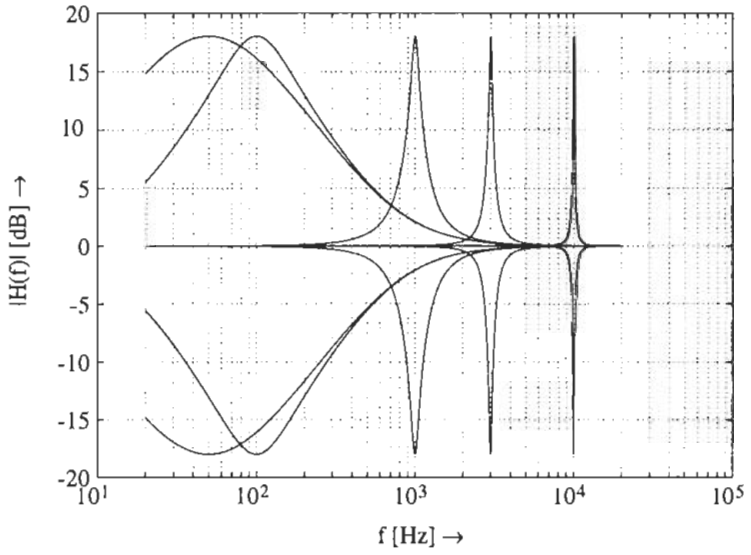


Figure 5.20 Second-order peak filter ($G = \pm 18$ dB - f_c=50,100,1000,3000,10000 Hz - f_b = 100 Hz).

5.1.3 Quantization Effects

The limited word-length for digital recursive filters leads to two different types of quantization error. The quantization of the coefficients of a digital filter results

in linear distortion which can be noticed as a deviation from the ideal frequency response. The quantization of the signal inside a filter structure is responsible for the maximum dynamic range and determines the noise behavior of the filter. Owing to rounding operations in a filter structure, round-off noise is produced. Another effect of the signal quantization is limit cycles. They can be classified as overflow limit cycles, small-scale limit cycles and limit cycles correlated with the input signal. Limit cycles are very disturbing owing to their small-band (sinusoidal) nature. The overflow limit cycles can be avoided by suitable scaling of the input signal. The effects of other errors mentioned above can be reduced by increasing the word-lengths of the coefficient and the state variables of the filter structure.

The noise behavior and coefficient sensitivity of a filter structure depend on the topology and the cutoff frequency (position of the poles in the Z-domain) of the filter. Since common audio filters operate between 20 Hz and 20 kHz at a sampling rate of 48 kHz, the filter structures are subjected to specially strict criteria with respect to error behavior. The frequency range for equalizers is between 20 Hz and 4...6 kHz because the human voice and many musical instruments have their formants in that frequency region. For given coefficient and signal word-lengths (like in a digital signal processor), a filter structure with low round-off noise for audio application can lead to a suitable solution. For this, the following second-order filter structures are compared.

The basis of the following considerations is the relationship between the coefficient sensitivity and round-off noise. This was first stated by Fettweis [Fet72]. By increasing the pole density in a certain region of the z-plane, the coefficient sensitivity and the round-off noise of the filter structure are reduced. Owing to these improvements, the coefficient word-length as well as signal word-length can be reduced. Work in designing digital filters with minimum word-length for coefficients and state variables was first carried out by Avenhaus [Ave71].

Typical audio filters like high-/low-pass, peak/shelving filters can be described by the second-order transfer function

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}. \tag{5.78}$$

The recursive part of the difference equation which can be derived from the transfer function (5.78) is considered more closely, since it plays a major role in affecting the error behavior. Owing to the quantization of the coefficients in the denominator in (5.78), the distribution of poles in the z-plane is restricted (see Fig. 5.21 for 6 bit quantization of coefficients). The pole distribution in the second quadrant of the z-plane is the mirror image of the first quadrant. Figure 5.22 shows a block diagram of the recursive part. Another equivalent representation of the denominator is

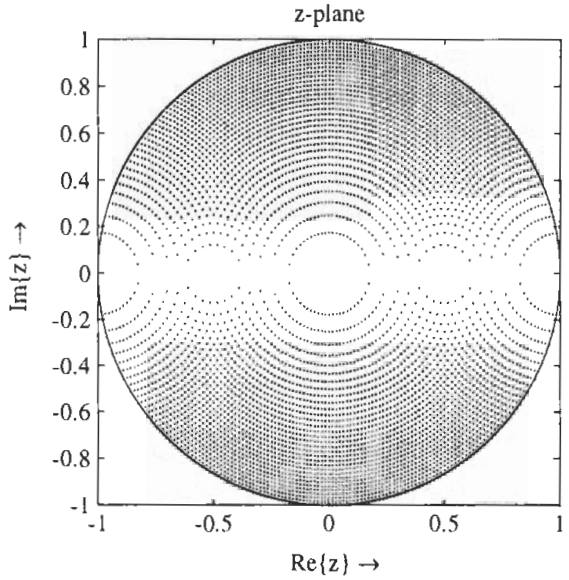


Figure 5.21 Direct-form structure - pole distribution (6 bit quantization).

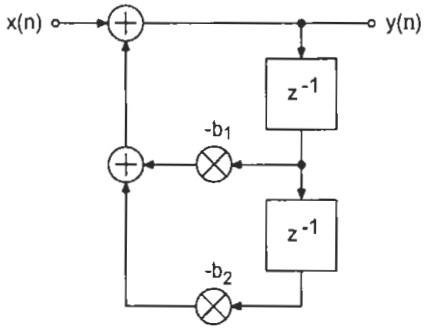


Figure 5.22 Directi-form structure - block diagram of recursive part.

given by

$$H(z) = \frac{N(z)}{1 - 2r\cos\varphi z^{-1} + r^2 z^{-2}}. \tag{5.79}$$

Here r is the radius and φ the corresponding phase of the complex poles. By quantizing these parameters, the pole distribution is altered in contrast to the case where b_1 and b_2 are quantized as in Equation (5.78).

The state variable structure [Mul76, Bom85] is based on the approach by Gold and Rader [Gol67], which is given by

$$H(z) = \frac{N(z)}{1 - 2\mathrm{Re}\{z_\infty\}z^{-1} + (\mathrm{Re}\{z_\infty\}^2 + \mathrm{Im}\{z_\infty\}^2)z^{-2}}. \tag{5.80}$$

The possible pole locations are shown in Fig. 5.23 for 6 bit quantization (block

diagram of recursive part is shown in Fig. 5.24). Owing to the quantization of

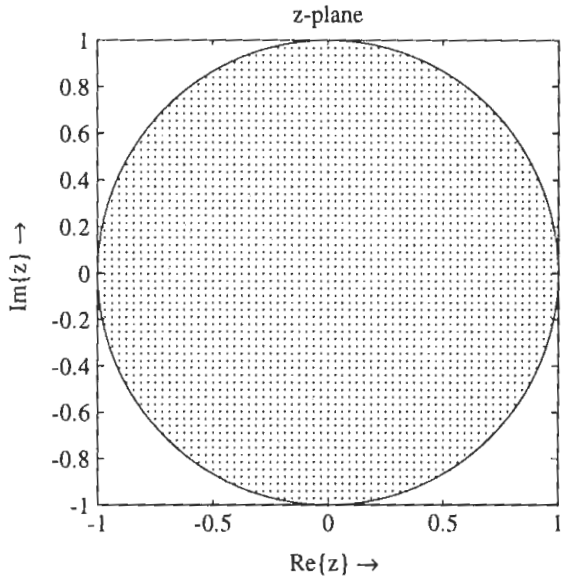


Figure 5.23 Gold and Rader - pole distribution (6 bit quantization).

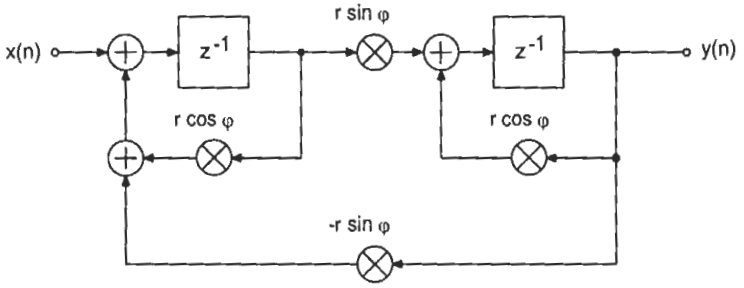


Figure 5.24 Gold and Rader - block diagram of recursive part.

real and imaginary parts, a uniform grid of different pole locations results. In contrast to direct quantization of the coefficients b_1 und b_2 in the denominator, the quantization of the real and imaginary parts leads to an increase in the pole density at $z = 1$. The possible pole locations in the second quadrant in the z-plane are the mirror images of the ones in the first quadrant.

In [Kin72] a filter structure is suggested which has a pole distribution as shown in Fig. 5.25 (block diagram of recursive part see Fig. 5.26).

The corresponding transfer function

$$H(z) = \frac{N(z)}{1 - (2 - k_1 k_2 - k_1^2) z^{-1} + (1 - k_1 k_2) z^{-2}}. \tag{5.81}$$

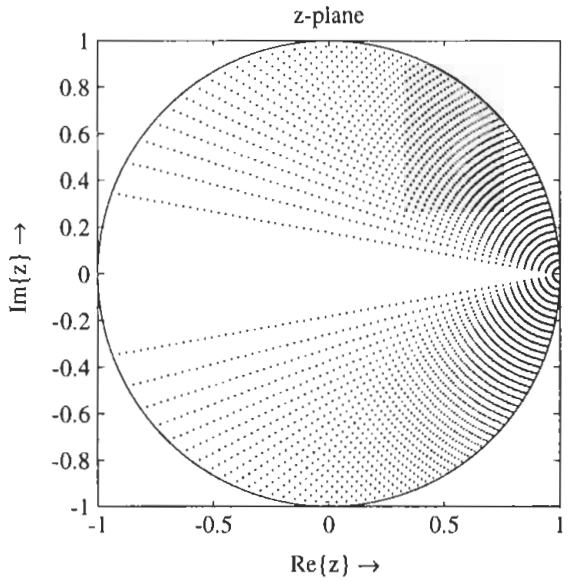


Figure 5.25 Kingsbury - pole distribution (6 bit quantization).

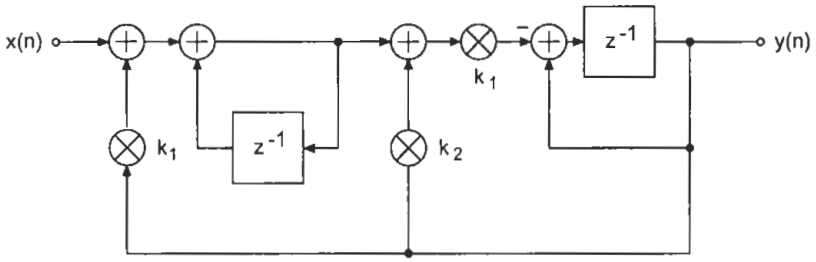


Figure 5.26 Kingsbury - block diagram of recursive part.

shows that in this case the coefficients b_1 and b_2 can be obtained by a linear combination of the quantized coefficients k_1 and k_2. The distance d of the pole from the point $z = 1$ determines the coefficients

$$k_1 = d = \sqrt{1 - 2r\cos\varphi + r^2} \tag{5.82}$$

$$k_2 = \frac{1 - r^2}{k_1} \tag{5.83}$$

as illustrated in Fig. 5.27.

The filter structures under consideration showed that by a suitable linear combination of quantized coefficients, any desired pole distribution can be obtained. An increase of the pole density at $z = 1$ can be achieved by influencing the linear relationship between the coefficient k_1 and the distance d from $z = 1$ [Zöl89]. The nonlinear relationship of the new coefficients gives the following structure with the

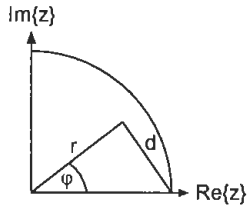


Figure 5.27 Geometric interpretation.

transfer function

$$H(z) = \frac{N(z)}{1 - (2 - z_1 z_2 - z_1^3)z^{-1} + (1 - z_1 z_2)z^{-2}} \tag{5.84}$$

and coefficients

$$z_1 = \sqrt[3]{1 + b_1 + b_2} \tag{5.85}$$

$$z_2 = \frac{1 - b_2}{z_1}, \tag{5.86}$$

with

$$z_1 = \sqrt[3]{d^2}. \tag{5.87}$$

The pole distribution of this structure is shown in Fig. 5.28. The block diagram of the recursive part is illustrated in Fig. 5.29. The increase in the pole density at $z = 1$ in contrast to previous pole distributions is noticed. The pole distributions of the Kingsbury and Zölzer structures show a decrease in the pole density for higher frequencies. For the pole density, a symmetry with respect to the imaginary axis as in the case of the direct-form structure and the Gold and Rader structure is not possible. But changing the sign in the recursive part of the difference equation results in a mirror image of the pole density. The mirror image can be achieved through a change of sign in the denominator polynomial. The denominator polynomial

$$D(z) = 1 \overset{!}{\overbrace{\pm}} (2 - z_1 z_2 - z_1^3)z^{-1} + (1 - z_1 z_2)z^{-2} \tag{5.88}$$

shows that the real part depends on the coefficient of z^{-1}.

Analytical Comparison of Noise Behavior of Different Filter Structures

In this section, recursive filter structures are analyzed in terms of their noise behavior in fixed-point arithmetic [Zöl89, Zöl94]. The block diagrams provide the

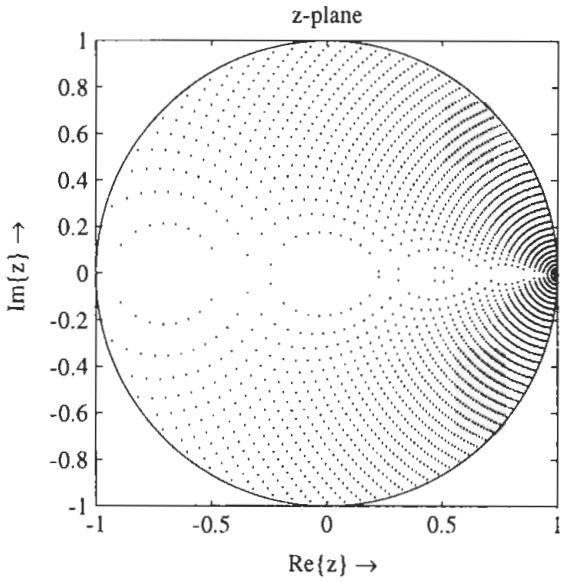


Figure 5.28 Zölzer - pole distribution (6 bit quantization).

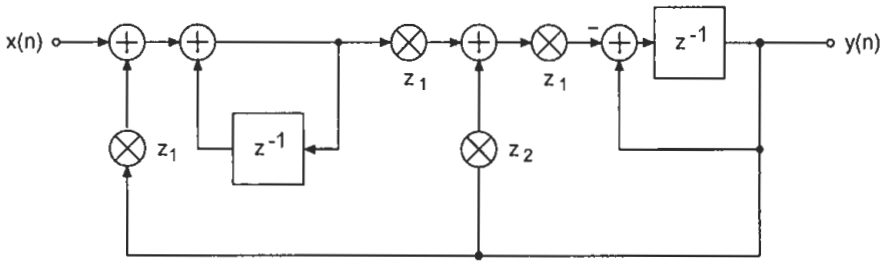


Figure 5.29 Zölzer - block diagram of recursive part.

basis of an analytical calculation of noise power owing to the quantization of state variables. First of all, the general case is considered in which quantization is performed after multiplication. For this purpose, the transfer function $G_i(z)$ of every multiplier output to the output of the filter structure is determined.

For this error analysis it is assumed that the signal within the filter structure covers the whole dynamic range so that the quantization error $e_i(n)$ is not correlated with the signal. Consecutive quantization error samples are not correlated with each other so that a uniform power density spectrum results [Sri77, Schü94]. It can also be assumed that different quantization errors $e_i(n)$ are uncorrelated within the filter structure. Owing to the uniform distribution of the quantization error, the variance can be given by

$$\sigma_e^2 = \frac{Q^2}{12}. \tag{5.89}$$

The quantization error is added at every point of quantization and is filtered by the corresponding transfer function $G(z)$ to the output of the filter. The variance of the output quantization noise (due to the noise source $e(n)$) is given by

$$\sigma_{ye}^2 = \sigma_e^2 \frac{1}{2\pi j} \oint\limits_{z=e^{j\Omega}} G(z)G(z^{-1})z^{-1}dz. \tag{5.90}$$

Exact solutions for the ring integral (5.90) can be found in [Jur64] for transfer functions up to the fourth order. With the L_2 norm of a periodic function

$$\| G \|_2 = \left[\frac{1}{2\pi} \int\limits_{-\pi}^{\pi} |G(e^{j\Omega})|^2 d\Omega \right]^{\frac{1}{2}} \tag{5.91}$$

the superposition of the noise variances leads with (5.91) to the total output noise variance

$$\sigma_{ye}^2 = \sigma_e^2 \sum_i \| G_i \|_2^2 \,. \tag{5.92}$$

The signal-to-noise ratio (SNR) for a full-range sinusoid can be written as

$$\text{SNR} = 10 \log_{10} \frac{0.5}{\sigma_{ye}^2} \quad [\text{dB}]. \tag{5.93}$$

The ring integral

$$I_n = \frac{1}{2\pi j} \oint\limits_{z=e^{j\Omega}} \frac{A(z)A(z^{-1})}{B(z)B(z^{-1})} z^{-1} dz \tag{5.94}$$

is given in [Jur64] for first-order systems by

$$G(z) = \frac{a_0 z + a_1}{b_0 z + b_1} \tag{5.95}$$

$$I_1 = \frac{(a_0^2 + a_1^2)b_0 - 2a_0 a_1 b_1}{b_0(b_0^2 - b_1^2)} \tag{5.96}$$

and for second-order systems by

$$G(z) = \frac{a_0 z^2 + a_1 z + a_2}{b_0 z^2 + b_1 z + b_2} \tag{5.97}$$

$$I_2 = \frac{A_0 b_0 c_1 - A_1 b_0 b_1 + A_2(b_1^2 - b_2 c_1)}{b_0[(b_0^2 - b_2^2)c_1 - (b_0 b_1 - b_1 b_2)b_1]} \tag{5.98}$$

$$A_0 = a_0^2 + a_1^2 + a_2^2 \tag{5.99}$$

$$A_1 = 2(a_0 a_1 + a_1 a_2) \tag{5.100}$$

$$A_2 = 2a_0 a_2 \tag{5.101}$$

$$c_1 = b_0 + b_2. \tag{5.102}$$

In the following, an analysis of the noise behavior for different recursive filter structures will be made. The noise transfer functions of individual recursive parts are responsible for noise shaping.

The error transfer function of a second-order direct-form structure (see Fig. 5.30) has only complex poles (see Table 5.5).

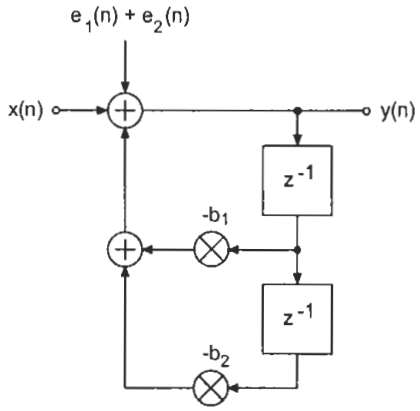


Figure 5.30 Direct-form with additive error signal.

Table 5.5 Direct-form - a) noise transfer function, b) quadratic L_2 norm and c) output noise variance in the case of quantization after every multiplication.

a)	$G_1(z) = G_2(z) = \dfrac{z^2}{z^2 + b_1 z + b_2}$
b)	$\|G_1\|_2^2 = \|G_2\|_2^2 = \dfrac{1+b_2}{1-b_2} \dfrac{1}{(1+b_2)^2 - b_1^2}$
c)	$\sigma_{ye}^2 = \sigma_e^2 2 \dfrac{1+b_2}{1-b_2} \dfrac{1}{(1+b_2)^2 - b_1^2}$

The implementation of poles near the unit circle leads to high amplification of the quantization error. The effect of the pole radius on the noise variance can be observed in the equation for output noise variance. The coefficient $b_2 = r^2$ approaches 1 leading to a huge increase in the output noise variance.

The Gold and Rader filter structure (Fig. 5.31) has an output noise variance that depends on the pole radius (see Table 5.6) and is independent of the pole phase. The latter fact is because of the uniform grid of the pole distribution. An additional zero on the real axis ($z = r\cos\varphi$) directly beneath the poles reduces the effect of the complex poles.

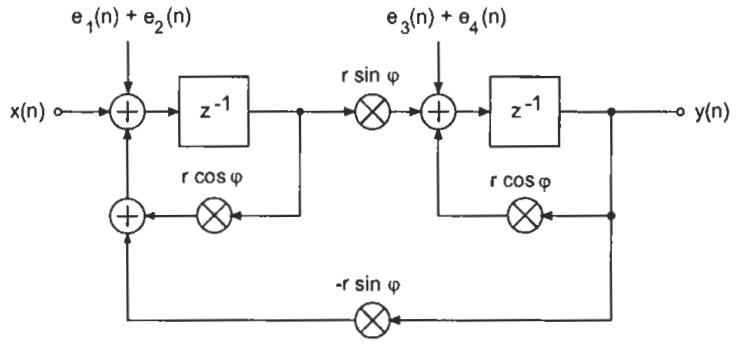


Figure 5.31 Gold and Rader structure with additive error signals.

Table 5.6 Gold and Rader - a) noise transfer function, b) quadratic L_2 norm and c) output noise variance in the case of quantization after every multiplication.

a)	$G_1(z) = G_2(z) = \dfrac{r \sin\varphi}{z^2 - 2r\cos\varphi z + r^2}$ $G_3(z) = G_4(z) = \dfrac{z - r\cos\varphi}{z^2 - 2r\cos\varphi z + r^2}$
b)	$\|G_1\|_2^2 = \|G_2\|_2^2 = \dfrac{1+b_2}{1-b_2}\dfrac{(r\sin\varphi)^2}{(1+b_2)^2 - b_1^2}$ $\|G_3\|_2^2 = \|G_4\|_2^2 = \dfrac{1}{1-b_2}\dfrac{[1+(r\sin\varphi)^2](1+b_2)^2 - b_1^2}{(1+b_2)^2 - b_1^2}$
c)	$\sigma_{ye}^2 = \sigma_e^2 2\dfrac{1}{1-b_2}$

The Kingsbury filter (Fig. 5.32 and Table 5.7) and the Zölzer filter (Fig. 5.33 and Table 5.8), which is derived from it, show that the noise variance depends on the pole radius. The noise transfer functions have a zero at $z = 1$ in addition to the complex poles. This zero reduces the amplifying effect of the pole near the unit circle at $z = 1$.

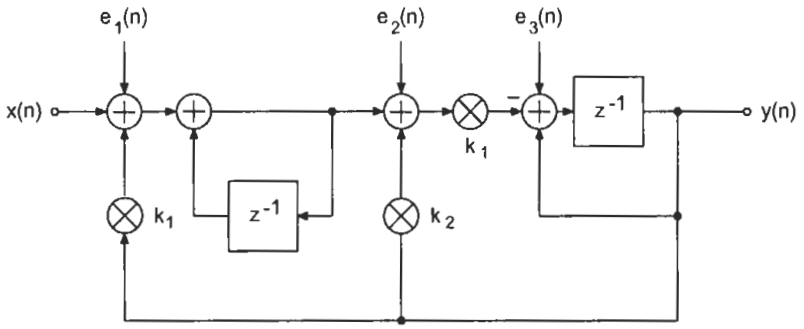


Figure 5.32 Kingsbury structure with additive error signals.

Table 5.7 Kingsbury - a) noise transfer function, b) quadratic L_2 norm and c) output noise variance in the case of quantization after every multiplication.

a)	$G_1(z) = \dfrac{-k_1 z}{z^2 - (2 - k_1k_2 - k_1^2)z + (1 - k_1k_2)}$ $G_2(z) = \dfrac{-k_1(z-1)}{z^2 - (2 - k_1k_2 - k_1^2)z + (1 - k_1k_2)}$ $G_3(z) = \dfrac{z-1}{z^2 - (2 - k_1k_2 - k_1^2)z + (1 - k_1k_2)}$
b)	$\|G_1\|_2^2 = \dfrac{1}{k_1k_2}\dfrac{2 - k_1k_2}{2(2 - k_1k_2) - k_1^2}$ $\|G_2\|_2^2 = \dfrac{k_1}{k_2}\dfrac{2}{2(2 - k_1k_2) - k_1^2}$ $\|G_3\|_2^2 = \dfrac{1}{k_1k_2}\dfrac{2}{2(2 - k_1k_2) - k_1^2}$
c)	$\sigma_{ye}^2 = \sigma_e^2 2\dfrac{5 + 2b_1 + 3b_2}{(1 - b_2)(1 + b_2 - b_1)}$

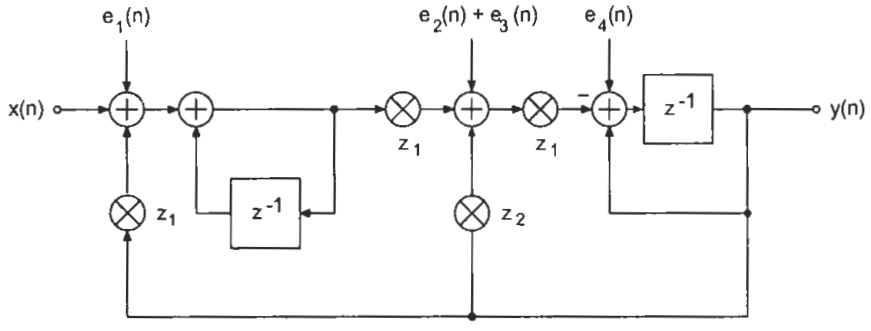


Figure 5.33 Zölzer structure with additive error signals.

Table 5.8 Zölzer - a) noise transfer function, b) quadratic L_2 norm and c) output noise variance in the case of quantization after every multiplication.

a)	$G_1(z)$	$= \dfrac{-z_1^2 z}{z^2 - (2 - z_1 z_2 - z_1^3)z + (1 - z_1 z_2)}$
	$G_2(z) = G_3(z)$	$= \dfrac{-z_1(z-1)}{z^2 - (2 - z_1 z_2 - z_1^3)z + (1 - z_1 z_2)}$
	$G_4(z)$	$= \dfrac{z-1}{z^2 - (2 - z_1 z_2 - z_1^3)z + (1 - z_1 z_2)}$
b)	$\|G_1\|_2^2$	$= \dfrac{z_1^4}{z_1 z_2} \dfrac{2 - z_1 z_2}{2z_1^3(2 - z_1 z_2) - z_1^6}$
	$\|G_2\|_2^2 = \|G_3\|_2^2$	$= \dfrac{z_1^6}{z_1 z_2} \dfrac{2}{2z_1^3(2 - z_1 z_2) - z_1^6}$
	$\|G_4\|_2^2$	$= \dfrac{z_1^3}{z_1 z_2} \dfrac{2}{2z_1^3(2 - z_1 z_2) - z_1^6}$
c)	$\sigma_{ye}^2 = \sigma_e^2 2 \dfrac{6 + 4(b_1 + b_2) + (1 + b_2)(1 + b_1 + b_2)^{1/3}}{(1 - b_2)(1 + b_2 - b_1)}$	

Figure 5.34 shows the signal-to-noise ratio versus the cutoff frequency for the four filter structures presented above. The signals are quantized to 16 bit. Here,

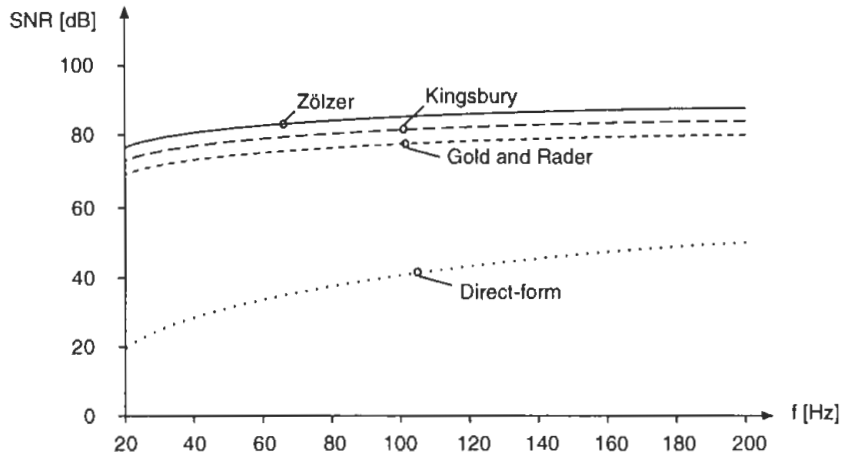


Figure 5.34 SNR vs. cutoff frequency - quantization of products ($f_c < 200$ Hz).

the poles move with increasing cutoff frequency on the curve characterized by the Q-factor $Q_\infty = 0.7071$ in the z-plane. For very small cutoff frequencies, the Zölzer filter shows an improvement of 3 dB in terms of signal-to-noise ratio compared

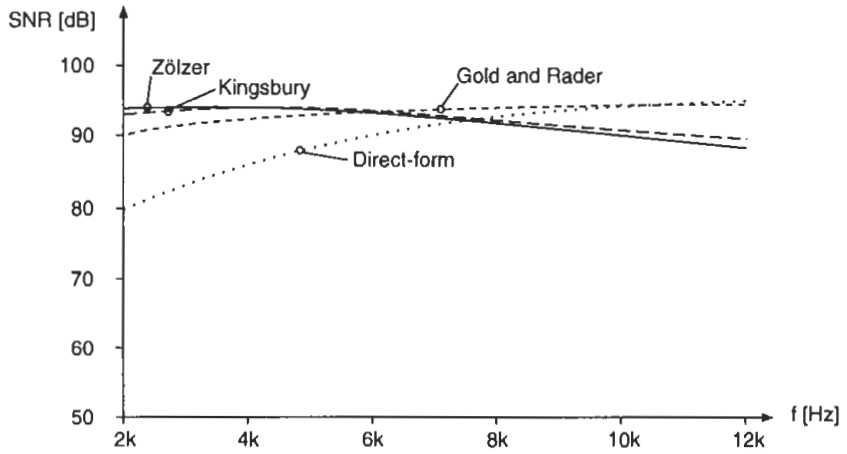
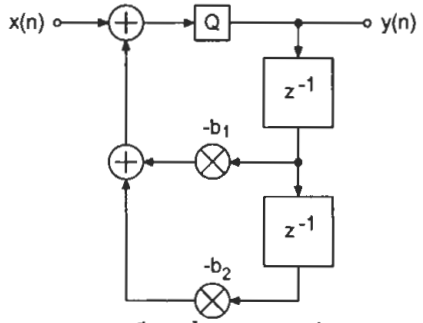


Figure 5.35 SNR vs. cutoff frequency - quantization of products ($f_c > 2$ kHz).

with the Kingsbury filter and an improvement of 6 dB compared with the Gold and Rader filter. Up to 5 kHz, the Zölzer filter yields better results (see Fig. 5.35). From 6 kHz onwards, the reduction of pole density in this filter leads to a decrease in the signal-to-noise ratio (see Fig. 5.35).

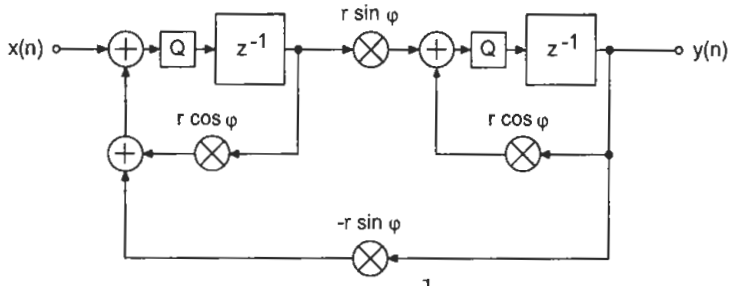
With regard to the implementation of the these filters with digital signal processors a quantization after every multiplication is not necessary. Quantization takes place when the accumulator has to be stored to memory. This can be seen in Figs. 5.36, 5.37, 5.38 and 5.39 by introducing quantizers where they really occur. The resulting output noise variances are also shown. The signal-to-noise ratio is



$$\sigma_{ye}^2 = \sigma_e^2 \frac{1+b_2}{1-b_2} \frac{1}{(1+b_2)^2 - b_1^2}$$

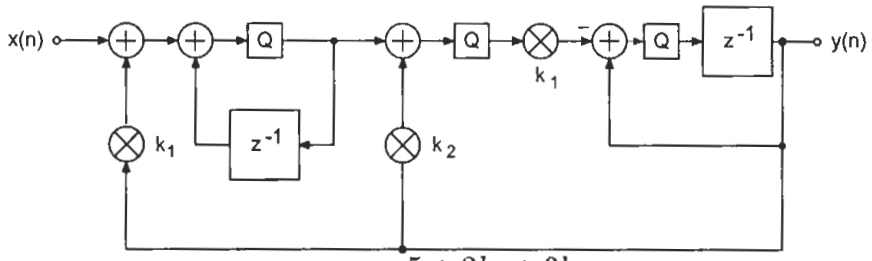
Figure 5.36 Direct-form filter - quantization after accumulator.

plotted versus the cutoff frequency in Figs. 5.40 and 5.41. In the case of direct-form and Gold and Rader filters, the signal-to-noise ratio increases by 3 dB whereas the output noise variance for the Kingsbury filter remains unchanged. The Kingsbury filter and the Gold and Rader filters exhibit similar results up to a frequency of



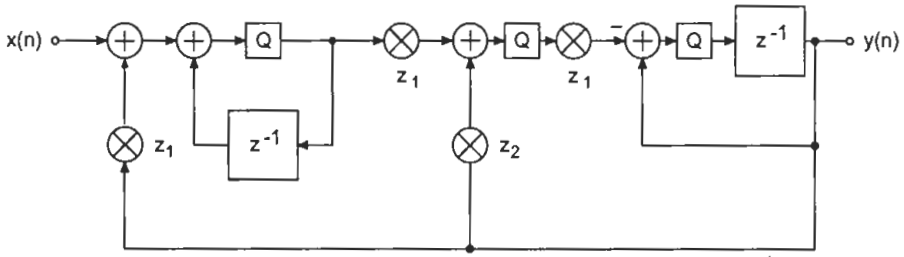
$$\sigma_{ye}^2 = \sigma_e^2 \frac{1}{1 - b_2}$$

Figure 5.37 Gold and Rader filter - quantization after accumulator.



$$\sigma_{ye}^2 = \sigma_e^2 \frac{5 + 2b_1 + 3b_2}{(1 - b_2)(1 + b_2 - b_1)}$$

Figure 5.38 Kingsbury filter - quantization after accumulator.



$$\sigma_{ye}^2 = \sigma_e^2 \frac{2(2 + b_1 + b_2) + (1 + b_2)(1 + b_1 + b_2)^{1/3}}{(1 - b_2)(1 + b_2 - b_1)}$$

Figure 5.39 Zölzer filter - quantization after accumulator.

200 kHz (see Fig. 5.40). The Zölzer filter demonstrates an improvement of 3 dB compared with these structures. For frequencies of up to 2 kHz (see Fig. 5.41) it is seen that the increased pole density leads to an improvement of the signal-to-noise ratio as well as a reduced effect due to coefficient quantization.

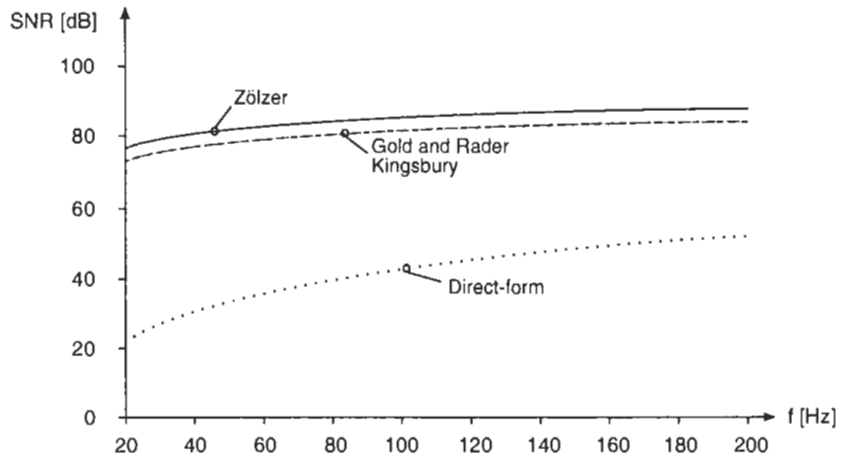


Figure 5.40 SNR vs. cutoff frequency - quantization after accumulator ($f_c < 200$ Hz).

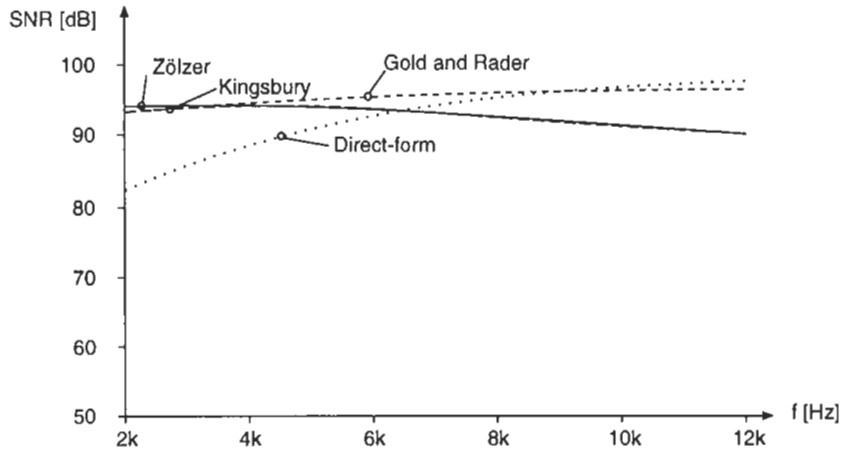


Figure 5.41 SNR vs. cutoff frequency - quantization after accumulator ($f_c > 2$ kHz).

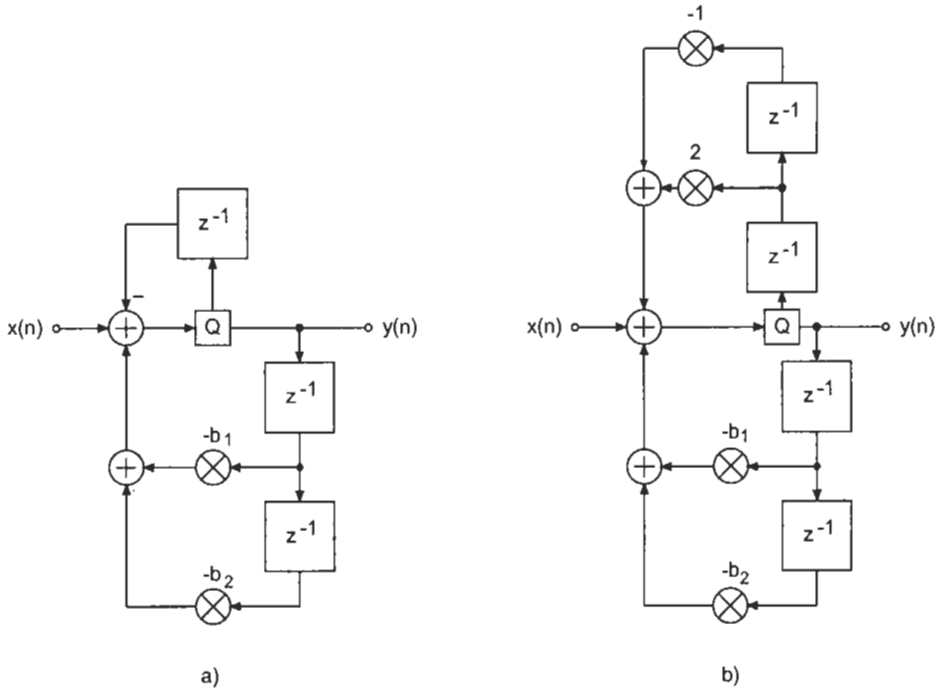
Noise Shaping in Recursive Filters

The analysis of the noise transfer function of different structures shows that for three structures with low round-off noise a zero at $z = 1$ occurs in the transfer functions $G(z)$ of the error signals in addition to the complex poles. This zero near the poles reduces the amplifying effect of the pole. If it is now possible to introduce another zero into the noise transfer function then the effect of the poles is compensated for to a larger extent. The procedure of feeding back the quantization error as shown in chapter 2 produces an additional zero in the noise transfer function [Tra77, Cha78, Abu79, Bar82, Zöl89]. The feedback of the quantization

error is first demonstrated with the help of the direct-form structure as shown in Fig. 5.42. This generates a zero at $z = 1$ in the noise transfer function given by

$$G_{1.O}(z) = \frac{1 - z^{-1}}{1 + b_1 z^{-1} + b_2 z^{-2}}. \tag{5.103}$$

The resulting variance σ^2 of the quantization error at the output of the filter is



$$\sigma^2_{DF1} = \sigma^2_e \frac{2}{(1 - b_2)(1 + b_2 - b_1)}$$
$$\sigma^2_{DF2} = \sigma^2_e \frac{6 + 2b_1 - 2b_2}{(1 - b_2)(1 + b_2 - b_1)}$$

Figure 5.42 Direct-form with noise shaping.

presented in Fig. 5.42. In order to produce two zeros at $z = 1$, the quantization error is fed back over two delays weighted with 2 and -1 (see Fig. 5.42b). The noise transfer function is, hence, given by

$$G_{2.O}(z) = \frac{1 - 2z^{-1} + z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}. \tag{5.104}$$

The signal-to-noise ratio of the direct-form is plotted versus the cutoff frequency in Fig. 5.43. Even a single zero significantly improves the signal-to-noise ratio in

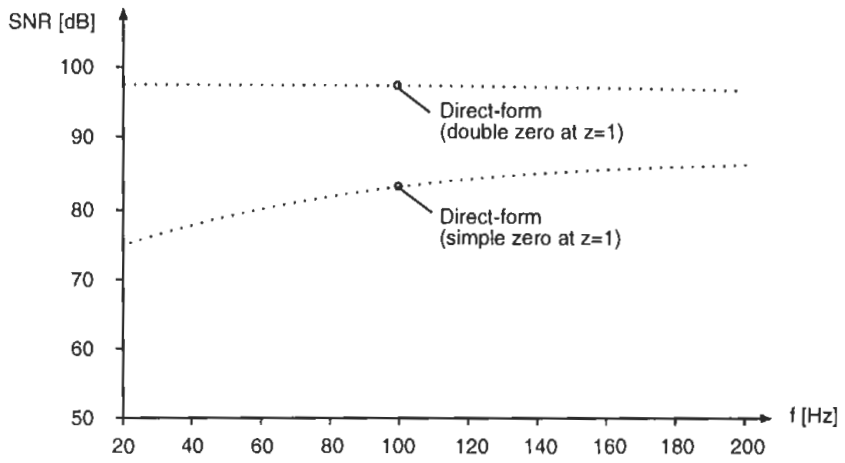
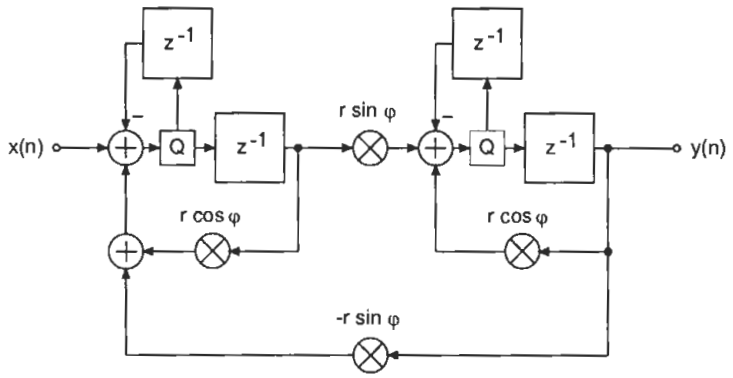


Figure 5.43 SNR - Noise shaping in direct-form filter structures.

the direct-form. The coefficients b_1 and b_2 approach -2 and 1 respectively with the decrease of the cutoff frequency. With this, the error is filtered with a second-order high-pass. The introduction of the additional zeros in the noise transfer function only affects the noise signal of the filter. The input signal is only affected by the transfer function $H(z)$. If the feedback coefficients are chosen equal to the coefficients b_1 and b_2 in the denominator polynomial, complex zeros are produced that are identical with the complex poles. The noise transfer function $G(z)$ is then reduced to unity. The choice of complex zeros directly at the location of the complex poles corresponds to double-precision arithmetic.

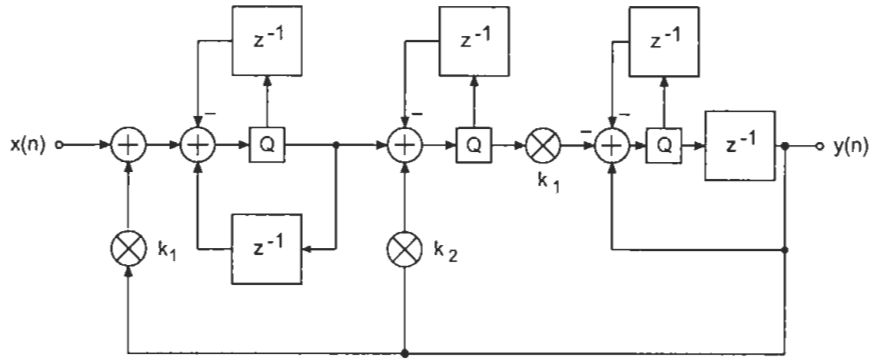


$$\sigma_{ye}^2 = \sigma_e^2 \frac{2 + b_1}{1 - b_2}$$

Figure 5.44 Gold and Rader filter with noise shaping.

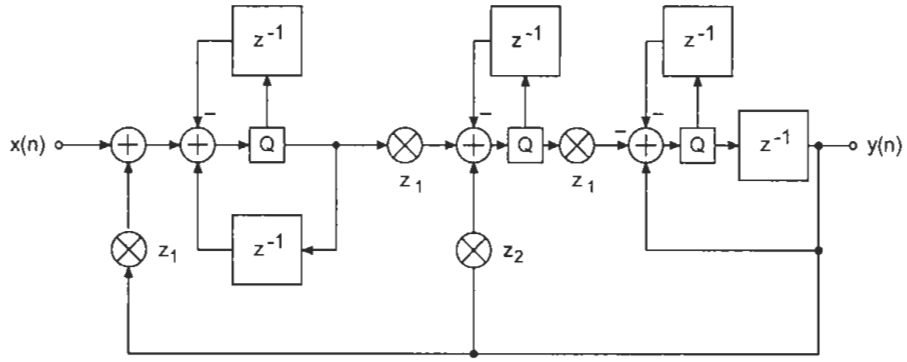
In [Abu79], an improvement of noise behavior for the direct-form in any desired location of the z-plane is achieved by placing additional simple-to-implement

complex zeros near the poles. For implementing filter algorithms with digital signal processors, these kinds of suboptimal zero are easily realized. Since the Gold and Rader, Kingsbury and Zölzer filter structures already have zeros in their respective noise transfer functions, it is sufficient to use a simple feedback for the quantization error. By virtue of this extension, the block diagrams in Figs. 5.44, 5.45 and 5.46 are obtained.



$$\sigma_{ye}^2 = \sigma_e^2 \frac{(1+k_1^2)((1+b_2)(6-2b_2)+2b_1^2+8b_1)+2k_1^2(1+b_1+b_2)}{(1-b_2)(1+b_2-b_1)(1+b_2-b_1)}$$

Figure 5.45 Kingsbury filter with noise shaping.



$$\sigma_{ye}^2 = \sigma_e^2 \frac{(1+z_1^2)((1+b_2)(6-2b_2)+2b_1^2+8b_1)+2z_1^4(1+b_1+b_2)}{(1-b_2)(1+b_2-b_1)(1+b_2-b_1)}$$

Figure 5.46 Zölzer filter with noise shaping.

The effect of noise shaping on signal-to-noise ratio is shown in Figs. 5.47 and 5.48. The almost ideal noise behavior of all filter structures for 16 bit quantization and very small cutoff frequencies can be observed. The effect of this noise shaping for increasing cutoff frequencies is shown in Fig. 5.48. The compensating effect of the two zeros at $z = 1$ is reduced.

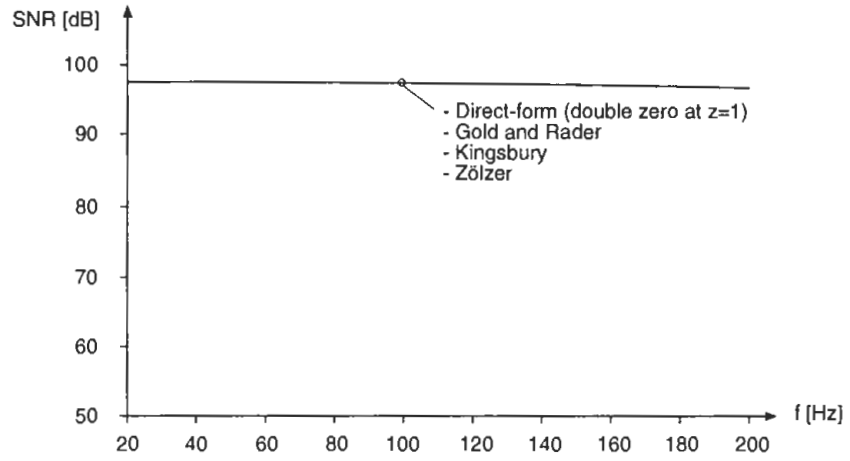


Figure 5.47 SNR - noise shaping (20 Hz .... 200 Hz).

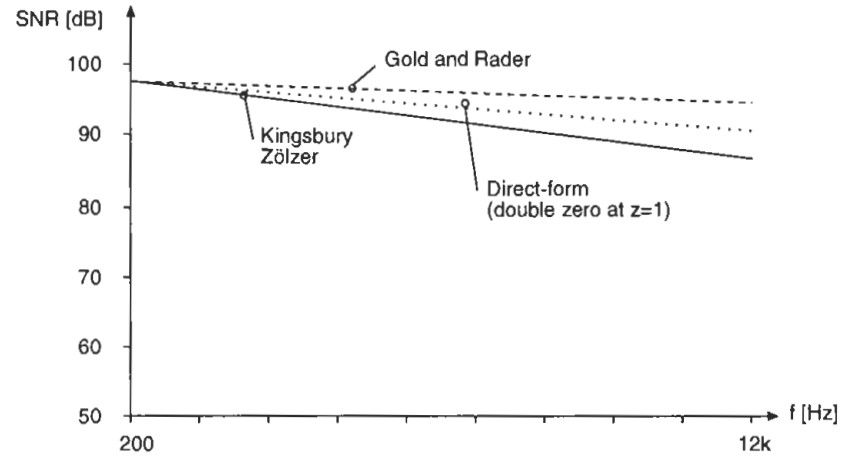


Figure 5.48 SNR - noise shaping (200 Hz ... 12 kHz).

Scaling

In a fixed-point implementation of a digital filter, a transfer function from the input of the filter to a junction within the filter has to be determined as well as the transfer function from the input to the output. By scaling the input signal, it has to be guaranteed that the signals remain within the number range at each junction or at the output.

In order to calculate scaling coefficients, different criteria can be used. The L_p norm is defined as

$$L_p = \|H\|_p = \left[\frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\Omega})|^p d\Omega\right]^{1/p} \tag{5.105}$$

and an expression for the L_∞ norm follows for $p = \infty$:

$$L_\infty = \|H(e^{j\Omega})\|_\infty = \max_{0 \le \Omega \le \pi} |H(e^{j\Omega})|. \tag{5.106}$$

The L_∞ norm represents the maximum of the amplitude frequency response. In general, the modulus of the output is

$$|y(n)| \le \|H\|_p \|X\|_q \tag{5.107}$$

with

$$\frac{1}{p} + \frac{1}{q} = 1 \qquad p, q \ge 1. \tag{5.108}$$

For the L_1, L_2 and L_∞ norms the explanations in Table 5.9 can be used.

Table 5.9 Commonly used scaling.

p	q	
1	∞	given max. value of input spectrum scaling w.r.t. the L_1 norm of $H(e^{j\Omega})$
∞	1	given L_1 norm of input spectrum $X(e^{j\Omega})$ scaling w.r.t. the L_∞ norm of $H(e^{j\Omega})$
2	2	given L_2 norm of input spectrum $X(e^{j\Omega})$ scaling w.r.t. the L_2 norm of $H(e^{j\Omega})$

With

$$|y_i(n)| \le \|H_i(e^{j\Omega})\|_\infty \|X(e^{j\Omega})\|_1 \tag{5.109}$$

the L_∞ norm is given by

$$L_\infty = \|h_i\|_\infty = \max_{k=0}^{\infty} |h_i(k)|. \tag{5.110}$$

For a sinusoidal input signal of amplitude 1 $\|X(e^{j\Omega})\|_1 = 1$. For $|y_i(n)| \le 1$ to be valid, the scaling factor must be chosen to be

$$S_i = \frac{1}{\|H_i(e^{j\Omega})\|_\infty}. \tag{5.111}$$

The scaling of the input signal is carried out with the maximum of the amplitude frequency response with the goal that for $|x(n)| \le 1$, $|y_i(n)| \le 1$. As a scaling coefficient for the input signal the highest scaling factor S_i is chosen. For determining the maximum of the transfer function

$$\|H(e^{j\Omega})\|_\infty = \max_{0 \le \Omega \le \pi} |H(e^{j\Omega})| \tag{5.112}$$

of a second-order system

$$\begin{aligned} H(z) &= \frac{a_0 + a_1 z^{-1} + a_2 z^{-1}}{1 + b_1 z^{-1} + b_2 z^{-1}} \\ &= \frac{a_0 z^2 + a_1 z + a_2}{z^2 + b_1 z + b_2} \end{aligned}$$

the maximum value can be calculated as

$$|H(e^{j\Omega})|^2 = \frac{\overbrace{\frac{a_0 a_2}{b_2}}^{\alpha_0} \cos^2(\Omega) + \overbrace{\frac{a_1(a_0 + a_2)}{2b_2}}^{\alpha_1} \cos(\Omega) + \overbrace{\frac{(a_0 - a_2)^2 + a_1^2}{4b_2}}^{\alpha_2}}{\cos^2(\Omega) + \underbrace{\frac{b_1(1 + b_2)}{2b_2}}_{\beta_1} \cos(\Omega) + \underbrace{\frac{(1 - b_2)^2 + b_1^2}{4b_2}}_{\beta_2}} = S^2. \tag{5.113}$$

With $x = \cos(\Omega)$ it follows that

$$(S^2 - \alpha_0)x^2 + (\beta_1 S^2 - \alpha_1)x + (\beta_2 S^2 - \alpha_2) = 0. \tag{5.114}$$

The solution of the Equation (5.114) leads to $x = \cos(\Omega_{max/min})$ which must be real ($-1 \leq x \leq 1$) for the maximum/minimum to occur at a real frequency. For a single solution (repeated roots) of the above quadratic equation, the discriminant must be $D = (p/2)^2 - q = 0$ ($x^2 + px + q = 0$). It follows that

$$D = \frac{(\beta_1 S^2 - \alpha_1)^2}{4(S^2 - \alpha_0)^2} - \frac{\beta_2 S^2 - \alpha_2}{S^2 - \alpha_0} = 0 \tag{5.115}$$

and

$$S^4(\beta_1^2 - 4\beta_2) + S^2(4\alpha_2 + 4\alpha_0\beta_2 - 2\alpha_1\beta_1) + (\alpha_1^2 - 4\alpha_0\alpha_2) = 0. \tag{5.116}$$

The solution of (5.116) gives two solutions for S^2. The solution with the larger value is chosen. If the discriminant D is not greater than zero, the maximum lies at $x = 1$ ($z = 1$) or $x = -1$ ($z = -1$) as given by

$$S^2 = \frac{\alpha_0 + \alpha_1 + \alpha_2}{1 + \beta_1 + \beta_2} \tag{5.117}$$

or

$$S^2 = \frac{\alpha_0 - \alpha_1 + \alpha_2}{1 - \beta_1 + \beta_2}. \tag{5.118}$$

Limit Cycles

Limit cycles are periodic processes in a filter which can be measured as sinusoidal signals. They arise owing to the quantization of state variables. The different types of limit cycle and the methods necessary to prevent them are briefly listed below:

- overflow limit cycles
  → saturation curve
  → scaling
- limit cycles for vanishing input
  → noise shaping
  → dithering
- limit cycles correlated with the input signal
  → noise shaping
  → dithering

5.2 Nonrecursive Audio Filters

For implementing linear phase audio filters, nonrecursive filters are used. The basis of an efficient implementation is the *fast convolution* [Rab75, Kam89] which will be discussed in the first section. The filter design is carried out by sampling a prescribed amplitude frequency response and linear phase constraints.

5.2.1 Fast Convolution

IDFT Implementation with DFT Algorithm. The discrete Fourier Transformation (DFT) is described by

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} = \mathrm{DFT}_k[x(n)] \tag{5.119}$$

$$W_N = e^{-j2\pi/N} \tag{5.120}$$

and the inverse discrete Fourier Transformation (IDFT) by

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}. \tag{5.121}$$

Without scaling factor $1/N$ we write

$$x'(n) = \sum_{k=0}^{N-1} X(k) W_N^{-nk} = \mathrm{IDFT}_n[X(k)], \tag{5.122}$$

so that the following symmetrical transformation algorithms hold

$$X'(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n) W_N^{nk} \tag{5.123}$$

$$x(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X'(k) W_N^{-nk}. \tag{5.124}$$

The IDFT differs from the DFT only by its sign in the exponential term.

An alternative approach for calculating the IDFT with the help of a DFT is described as follows [Cad87, Duh88]. We will make use of the relationships

$$x(n) = a(n) + j \cdot b(n) \tag{5.125}$$

$$j \cdot x^*(n) = b(n) + j \cdot a(n). \tag{5.126}$$

Conjugating (5.122) gives

$$x'^*(n) = \sum_{k=0}^{N-1} X^*(k) W_N^{nk}. \tag{5.127}$$

The multiplication of (5.127) by j leads to

$$j \cdot x'^*(n) = \sum_{k=0}^{N-1} j \cdot X^*(k) W_N^{nk}. \tag{5.128}$$

Conjugating and multiplying (5.128) by j results in

$$x'(n) = j \cdot \left[\sum_{k=0}^{N-1} (j \cdot X^*(k) W_N^{nk} \right]^*. \tag{5.129}$$

An interpretation of (5.126) and (5.129) suggests the following way of performing the IDFT with the DFT algorithm:

1. exchange the real with the imaginary part of the spectral sequence

$$Y(k) = Y_I(k) + jY_R(k)$$

2. transformation with DFT algorithm

$$\mathrm{DFT}[Y(k)] = y_I(n) + jy_R(n)$$

3. exchange the real with the imaginary part of the time sequence

$$y(n) = y_R(n) + jy_I(n)$$

For implementation on a digital signal processor, the use of DFT saves memory for IDFT.

Discrete Fourier Transformation of two Real Sequences. In many applications, stereo signals that consist of a left and right channel are processed. With the help of the DFT, both channels can be transformed simultaneously into the frequency domain [Sor87, Ell82].

For a real sequence $x(n)$ follows

$$\begin{aligned} X(k) &= X^*(-k) \qquad k = 0, 1, \ldots, N-1 && (5.130)\\ &= X^*(N-k). && (5.131) \end{aligned}$$

For a discrete Fourier transformation of two real sequences $x(n)$ and $y(n)$, a complex sequence is first formed according to

$$z(n) = x(n) + jy(n). \tag{5.132}$$

The Fourier transformation gives

$$\begin{aligned} \mathrm{DFT}[z(n)] &= \mathrm{DFT}[x(n) + jy(n)] \\ &= Z_R(k) + jZ_I(k) && (5.133)\\ &= Z(k), && (5.134) \end{aligned}$$

where

$$\begin{aligned} Z(k) &= Z_R(k) + jZ_I(k) && (5.135)\\ &= X_R(k) + jX_I(k) + j[Y_R(k) + jY_I(k)] && (5.136)\\ &= X_R(k) - Y_I(k) + j[X_I(k) + Y_R(k)]. && (5.137) \end{aligned}$$

Since $x(n)$ and $y(n)$ are real sequences, it follows from (5.131) that

$$\begin{aligned} Z(N-k) &= Z_R(N-k) + jZ_I(N-k) = Z^*(k) && (5.138)\\ &= X_R(k) - jX_I(k) + j[Y_R(k) - jY_I(k)] && (5.139)\\ &= X_R(k) + Y_I(k) - j[X_I(k) - Y_R(k)]. && (5.140) \end{aligned}$$

Considering the real part of $Z(k)$, adding (5.137) and (5.140) gives

$$2X_R(k) = Z_R(k) + Z_R(N-k) \tag{5.141}$$

$$\rightarrow \quad X_R(k) = \frac{1}{2}[Z_R(k) + Z_R(N-k)] \tag{5.142}$$

and subtraction of Equation (5.140) from (5.137) results in

$$2Y_I(k) = Z_R(N-k) - Z_R(k) \tag{5.143}$$

$$\rightarrow \quad Y_I(k) = \frac{1}{2}[Z_R(N-k) - Z_R(k)]. \tag{5.144}$$

Considering the imaginary part of $Z(k)$, adding (5.137) and (5.140) gives

$$2Y_R(k) = Z_I(k) + Z_I(N-k) \tag{5.145}$$

$$\rightarrow \quad Y_R(k) = \frac{1}{2}[Z_I(k) + Z_I(N-k)] \tag{5.146}$$

and subtraction of Equation (5.140) from (5.137) results in

$$2X_I(k) = Z_I(k) - Z_I(N-k) \tag{5.147}$$

$$\rightarrow \quad X_I(k) = \frac{1}{2}[Z_I(k) - Z_I(N-k)]. \tag{5.148}$$

Hence the spectral functions are given by

$$X(k) = \mathrm{DFT}[x(n)] = X_R(k) + jX_I(k) \tag{5.149}$$

$$= \frac{1}{2}[Z_R(k) + Z_R(N-k)]$$

$$+j\frac{1}{2}[Z_I(k) - Z_I(N-k)] \tag{5.150}$$

$$k = 0, 1, \ldots, \frac{N}{2}$$

$$Y(k) = \mathrm{DFT}[y(n)] = Y_R(k) + jY_R(k) \tag{5.151}$$

$$= \frac{1}{2}[Z_I(k) + Z_I(N-k)]$$

$$+j\frac{1}{2}[Z_R(N-k) - Z_R(k)] \tag{5.152}$$

$$k = 0, 1, \ldots, \frac{N}{2}$$

and

$$X_R(k) + jX_I(k) = X_R(N-k) - jX_I(N-k) \tag{5.153}$$

$$Y_R(k) + jY_I(k) = Y_R(N-k) - jY_I(N-k) \tag{5.154}$$

$$k = \frac{N}{2} + 1, \ldots, N-1.$$

Fast Convolution if Spectral Functions are Known. The spectral functions $X(k)$, $Y(k)$ and $H(k)$ are known. With the help of (5.137), the spectral sequence can be formed by

$$\begin{aligned} Z(k) &= Z_R(k) + jZ_I(k) && (5.155)\\ &= X_R(k) - Y_I(k) + j[X_I(k) + Y_R(k)] && (5.156) \end{aligned}$$

$$k = 0, 1, \ldots, N-1.$$

Filtering is done by multiplication in the frequency domain:

$$\begin{aligned} Z'(k) &= [Z_R(k) + jZ_I(k)][H_R(k) + jH_I(k)] \\ &= Z_R(k)H_R(k) - Z_I(k)H_I(k) \\ &\quad + j[Z_R(k)H_I(k) + Z_I(k)H_R(k)]. && (5.157) \end{aligned}$$

The inverse transformation gives

$$\begin{aligned} z'(n) &= [x(n) + jy(n)] * h(n) = x(n) * h(n) + jy(n) * h(n) && (5.158)\\ &= \mathrm{IDFT}[Z'(k)] \\ &= z'_R(n) + jz'_I(n), && (5.159) \end{aligned}$$

so that the filtered output sequence is given by

$$\begin{aligned} x'(n) &= z'_R(n) && (5.160)\\ y'(n) &= z'_I(n). && (5.161) \end{aligned}$$

The filtering of a stereo signal can hence be done by transformation into the frequency domain, multiplication of the spectral functions and inverse transformation of left and right channels.

5.2.2 Fast Convolution of Long Sequences

The fast convolution of two real input sequences $x_l(n)$ and $x_{l+1}(n)$ of length N_1 with the impulse response $h(n)$ of length N_2 leads to the output sequences

$$\begin{aligned} y_l(n) &= x_l(n) * h(n) && (5.162)\\ y_{l+1}(n) &= x_{l+1}(n) * h(n) && (5.163) \end{aligned}$$

of length $N_1 + N_2 - 1$. The implementation of a nonrecursive filter with fast convolution becomes more efficient than the direct implementation of an FIR filter for filter lengths $N > 30$ [Rab75, Kam89]. Therefore the following procedure will be performed:

- Formation of a complex sequence

$$z(n) = x_l(n) + jx_{l+1}(n) \tag{5.164}$$

- Fourier transformation of the impulse response $h(n)$ that is padded with zeros to a length $N \geq N_1 + N_2 - 1$

$$H(k) = \mathrm{DFT}[h(n)] \quad (\text{FFT-length } N) \tag{5.165}$$

- Fourier transformation of the sequence $z(n)$ that is padded with zeros to a length $N \geq N_1 + N_2 - 1$

$$Z(k) = \mathrm{DFT}[z(n)] \quad (\text{FFT-length } N) \tag{5.166}$$

- Formation of a complex output sequence

$$\begin{aligned} e(n) &= \mathrm{IDFT}[Z(k)H(k)] && (5.167)\\ &= z(n) * h(n) && (5.168)\\ &= x_l(n) * h(n) + jx_{l+1}(n) * h(n) && (5.169) \end{aligned}$$

- Formation of a real output sequence

$$\begin{aligned} y_l(n) &= \mathrm{Re}\{e(n)\} && (5.170)\\ y_{l+1}(n) &= \mathrm{Im}\{e(n)\}. && (5.171) \end{aligned}$$

For the convolution of an infinite-length input sequence with an impulse response $h(n)$, the input sequence is partitioned into sequences $x_m(n)$ of length L:

$$x_m(n) = \begin{cases} x(n) & (m-1)L \leq n \leq mL-1 \\ 0 & \text{otherwise} \end{cases} . \tag{5.172}$$

The input sequence is given by superposition of finite-length sequences according to

$$x(n) = \sum_{m=1}^{\infty} x_m(n). \tag{5.173}$$

The convolution of the input sequence with the impulse response $h(n)$ of length M gives

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{5.174}$$

$$= \sum_{k=0}^{M-1} h(k) \sum_{m=1}^{\infty} x_m(n-k) \tag{5.175}$$

$$= \sum_{m=1}^{\infty} \left[\sum_{k=0}^{M-1} h(k)x_m(n-k) \right]. \tag{5.176}$$

The term in brackets corresponds to the convolution of a finite-length sequence $x_m(n)$ of length L with the impulse response of length M. The output signal can be given as superposition of convolution products of length $L+M-1$. With these partial convolution products

$$y_m(n) = \begin{cases} \sum_{k=0}^{M-1} h(k)x_m(n-k) & (m-1)L \le n \le mL+M-2 \\ 0 & \text{otherwise} \end{cases} \tag{5.177}$$

the output signal can be written as

$$y(n) = \sum_{m=1}^{\infty} y_m(n). \tag{5.178}$$

If the length M of the impulse response is very long, it can be similarly partitioned into P parts each of length M/P. With

$$h_p(n-(p-1)\frac{M}{P}) = \begin{cases} h(n) & (p-1)\frac{M}{P} \le n \le p\frac{M}{P}-1 \\ 0 & \text{otherwise} \end{cases} \tag{5.179}$$

it follows that

$$h(n) = \sum_{p=1}^{P} h_p \left(n-(p-1)\frac{M}{P} \right). \tag{5.180}$$

With $M_p = pM/P$ and (5.178) the following partitioning can be done

$$y(n) = \sum_{m=1}^{\infty} \Bigg[\underbrace{\sum_{k=0}^{M-1} h(k)x_m(n-k)}_{y_m(n)} \Bigg] \tag{5.181}$$

$$= \sum_{m=1}^{\infty} \left[\sum_{k=0}^{M_1-1} h(k)x_m(n-k) + \sum_{k=M_1}^{M_2-1} h(k)x_m(n-k) + \ldots \right.$$

$$+ \sum_{k=M_{P-1}}^{M-1} h(k)x_m(n-k)\Big]. \tag{5.182}$$

This can be rewritten as

$$\begin{aligned} y(n) &= \sum_{m=1}^{\infty} \Big[\underbrace{\sum_{k=0}^{M_1-1} h_1(k)x_m(n-k)}_{y_{m1}} + \underbrace{\sum_{k=0}^{M_1-1} h_2(k)x_m(n-M_1-k)}_{y_{m2}} \\ &\quad + \underbrace{\sum_{k=0}^{M_1-1} h_3(k)x_m(n-2M_1-k)}_{y_{m3}} \\ &\quad \ldots + \underbrace{\sum_{k=0}^{M_1-1} h_P(k)x_m(n-(P-1)M_1-k)}_{y_{mP}} \Big] \\ &= \sum_{m=1}^{\infty} \underbrace{[y_{m1}(n) + y_{m2}(n-M_1) + \ldots + y_{mP}(n-(P-1)M_1)]}_{y_m(n)}. \end{aligned} \tag{5.183}$$

An example of partitioning the impulse response into $P = 4$ parts is graphically shown in Fig. 5.49. This leads to

$$\begin{aligned} y(n) &= \sum_{m=1}^{\infty} \Big[\underbrace{\sum_{k=0}^{M_1-1} h_1(k)x_m(n-k)}_{y_{m1}} + \underbrace{\sum_{k=0}^{M_1-1} h_2(k)x_m(n-M_1-k)}_{y_{m2}} \\ &\quad + \underbrace{\sum_{k=0}^{M_1-1} h_3(k)x_m(n-2M_1-k)}_{y_{m3}} + \underbrace{\sum_{k=0}^{M_1-1} h_4(k)x_m(n-3M_1-k)}_{y_{m4}} \Big] \\ &= \sum_{m=1}^{\infty} \underbrace{[y_{m1}(n) + y_{m2}(n-M_1) + y_{m3}(n-2M_1) + y_{m4}(n-3M_1)]}_{y_m(n)}. \end{aligned} \tag{5.184}$$

The procedure of a fast convolution by partitioning the input sequence $x(n)$ as well as the impulse response $h(n)$ is given in the following for the example in Fig. 5.49.

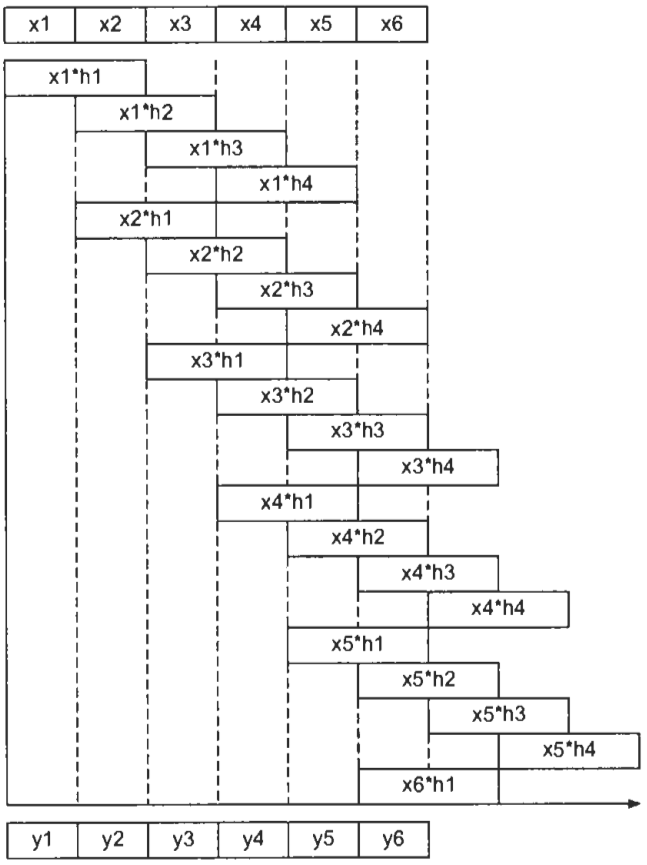


Figure 5.49 Scheme for a fast convolution with $P = 4$.

1. Decomposition of the impulse response $h(n)$ of length $4M$:

$$h_1(n) = h(n) \qquad 0 \leq n \leq M-1 \tag{5.185}$$

$$h_2(n-M) = h(n) \qquad M \leq n \leq 2M-1 \tag{5.186}$$

$$h_3(n-2M) = h(n) \qquad 2M \leq n \leq 3M-1 \tag{5.187}$$

$$h_4(n-3M) = h(n) \qquad 3M \leq n \leq 4M-1 \tag{5.188}$$

2. Zero-padding of partial impulse responses up to a length $2M$:

$$h_1(n) = \begin{cases} h_1(n) & 0 \leq n \leq M-1 \\ 0 & M \leq n \leq 2M-1 \end{cases} \tag{5.189}$$

$$h_2(n) = \begin{cases} h_2(n) & 0 \leq n \leq M-1 \\ 0 & M \leq n \leq 2M-1 \end{cases} \tag{5.190}$$

$$h_3(n) = \begin{cases} h_3(n) & 0 \leq n \leq M-1 \\ 0 & M \leq n \leq 2M-1 \end{cases} \tag{5.191}$$

$$h_4(n) = \begin{cases} h_4(n) & 0 \leq n \leq M-1 \\ 0 & M \leq n \leq 2M-1 \end{cases} \tag{5.192}$$

3. Calculating and storing

$$H_i(k) = \mathrm{DFT}[h_i(n)] \quad i = 1, \ldots, 4 \quad \text{(FFT-length } 2M) \tag{5.193}$$

4. Decomposition of the input sequence $x(n)$ into partial sequences $x_l(n)$ of length M:

$$x_l(n) \;=\; x(n) \qquad (l-1)M \leq n \leq lM-1 \quad l = 1, \ldots \infty \tag{5.194}$$

5. Nesting partial sequences:

$$z_m(n) \;=\; x_l(n) + jx_{l+1}(n) \quad m, l = 1, \ldots, \infty \tag{5.195}$$

6. Zero-padding of complex sequence $z_m(n)$ up to a length $2M$:

$$z_m(n) \;=\; \begin{cases} z_m(n) & (l-1)M \leq n \leq lM-1 \\ 0 & lM \leq n \leq (l+1)M-1 \end{cases} \tag{5.196}$$

7. Fourier transformation of the complex sequences $z_m(n)$:

$$Z_m(k) = \mathrm{DFT}[z_m(n)] = Z_{mR}(k) + jZ_{mI}(k) \quad \text{(FFT-length } 2M) \tag{5.197}$$

8. Multiplication in the frequency domain:

$$\begin{aligned} &[Z_R(k) + jZ_I(k)][H_R(k) + jH_I(k)] = \\ &\quad Z_R(k)H_R(k) - Z_I(k)H_I(k) \\ &\quad +j[Z_R(k)H_I(k) + Z_I(k)H_R(k)] \end{aligned} \tag{5.198}$$

$$E_{m1}(k) \;=\; Z_m(k)H_1(k) \quad k = 0, 1, \ldots, 2M-1 \tag{5.199}$$
$$E_{m2}(k) \;=\; Z_m(k)H_2(k) \quad k = 0, 1, \ldots, 2M-1 \tag{5.200}$$
$$E_{m3}(k) \;=\; Z_m(k)H_3(k) \quad k = 0, 1, \ldots, 2M-1 \tag{5.201}$$
$$E_{m4}(k) \;=\; Z_m(k)H_4(k) \quad k = 0, 1, \ldots, 2M-1 \tag{5.202}$$

9. Inverse transformation:

$$e_{m1}(n) \;=\; \mathrm{IDFT}[Z_m(k)H_1(k)] \quad n = 0, 1, \ldots, 2M-1 \tag{5.203}$$
$$e_{m2}(n) \;=\; \mathrm{IDFT}[Z_m(k)H_2(k)] \quad n = 0, 1, \ldots, 2M-1 \tag{5.204}$$
$$e_{m3}(n) \;=\; \mathrm{IDFT}[Z_m(k)H_3(k)] \quad n = 0, 1, \ldots, 2M-1 \tag{5.205}$$
$$e_{m4}(n) \;=\; \mathrm{IDFT}[Z_m(k)H_4(k)] \quad n = 0, 1, \ldots, 2M-1 \tag{5.206}$$

10. Determination of partial convolutions:

$$\begin{aligned}
\mathrm{Re}\{e_{m1}(n)\} &= x_l * h_1 && (5.207)\\
\mathrm{Im}\{e_{m1}(n)\} &= x_{l+1} * h_1 && (5.208)\\
\mathrm{Re}\{e_{m2}(n)\} &= x_l * h_2 && (5.209)\\
\mathrm{Im}\{e_{m2}(n)\} &= x_{l+1} * h_2 && (5.210)\\
\mathrm{Re}\{e_{m3}(n)\} &= x_l * h_3 && (5.211)\\
\mathrm{Im}\{e_{m3}(n)\} &= x_{l+1} * h_3 && (5.212)\\
\mathrm{Re}\{e_{m4}(n)\} &= x_l * h_4 && (5.213)\\
\mathrm{Im}\{e_{m4}(n)\} &= x_{l+1} * h_4 && (5.214)
\end{aligned}$$

11. *Overlap-Add* of partial sequences, increment from $l = l + 2$ and $m = m + 1$ and back to step 5.

5.2.3 Filter Design by Frequency Sampling

Audio filter design for nonrecursive filter realizations by fast convolution can be carried out by the frequency sampling method [Opp75, Rab75]. For linear phase systems follows

$$H(e^{j\Omega}) = H_O(e^{j\Omega})e^{-j\frac{N_F-1}{2}\Omega}, \tag{5.215}$$

where $H_O(e^{j\Omega})$ is a real valued function and N_F is the length of the impulse response. The magnitude $|H(e^{j\Omega})|$ is calculated by sampling in the frequency domain at equidistant places

$$\frac{f}{f_A} = \frac{k}{N_F} \qquad \text{with} \qquad k = 0, 1, ..., N_F - 1 \tag{5.216}$$

according to

$$|H(e^{j\Omega})| = H_O(e^{j2\pi k/N_F}) \qquad k = 0, 1, ..., \frac{N_F}{2} - 1. \tag{5.217}$$

Hence, a filter can be designed by fulfilling conditions in the frequency domain. The linear phase is determined as

$$\begin{aligned}
e^{-j\frac{N_F-1}{2}\Omega} &= e^{-j2\pi\frac{N_F-1}{2}\frac{k}{N_F}} && (5.218)\\
&= \cos(2\pi\frac{N_F-1}{2}\frac{k}{N_F}) - j\sin(2\pi\frac{N_F-1}{2}\frac{k}{N_F}) && (5.219)\\
&\qquad k = 0, 1, ..., \frac{N_F}{2} - 1.
\end{aligned}$$

Owing to the real transfer function $H(z)$ for an even filter length, we have to fulfill

$$H(k = \frac{N_F}{2}) = 0 \tag{5.220}$$

and

$$H(k) = H^*(N_F - k) \qquad k = 0, 1, ... \frac{N_F}{2} - 1. \tag{5.221}$$

This has to be taken into consideration while designing filters of even length N_F. The impulse response $h(n)$ is obtained through an N_F-point IDFT of the spectral sequence $H(k)$. This impulse response is extended with *Zero-padding* [Rab75, Cad87, Kam89] to the length N and then transformed by an N-point DFT resulting in the spectral sequence $H(k)$ of the filter.

Example: For $N_F = 8$, $|H(k)| = 1$ $(k = 0, 1, 2, 3, 5, 6, 7)$ and $|H(4)| = 0$, the group delay is $t_G = 3.5$. Figure 5.50 shows the amplitude, real part and imaginary part of the transfer function and the impulse response $h(n)$.

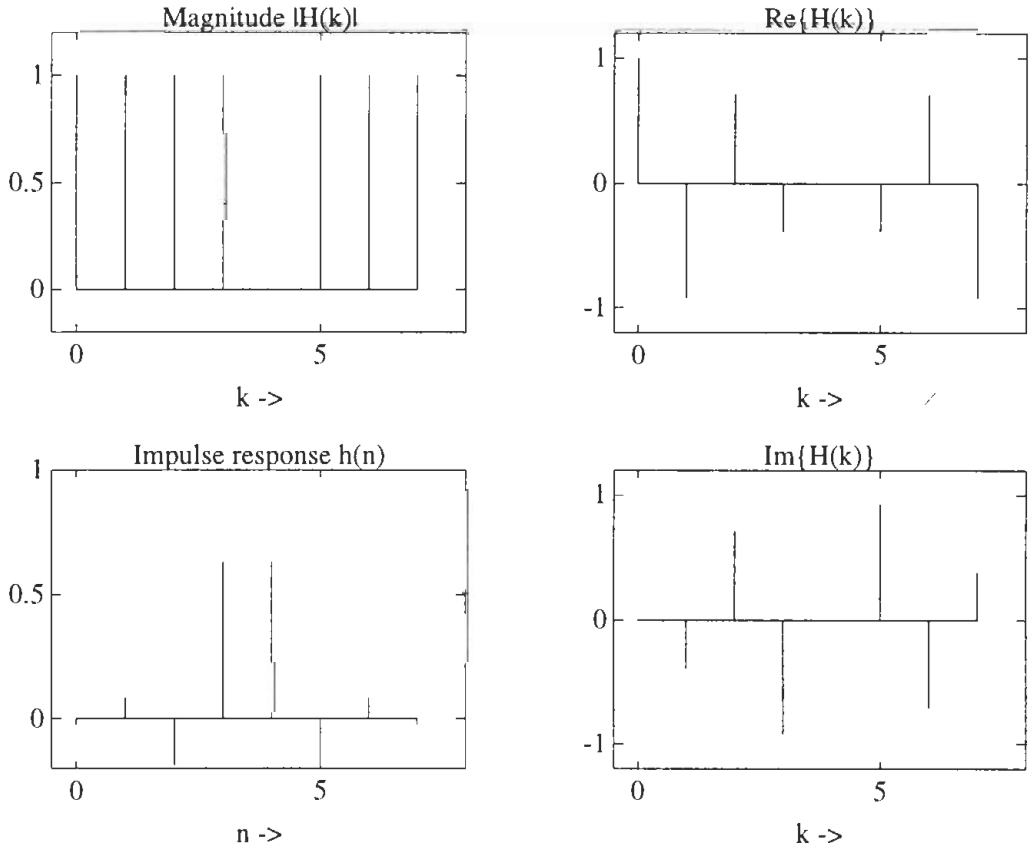


Figure 5.50 Filter design by frequency sampling (N_F even).

5.3 Multi-complementary Filter Bank

The subband processing of audio signals is mainly used in source coding applications for efficient transmission and storing. The basis for the subband decomposition is critically sampled filter banks [Vai93, Fli94]. These filter banks allow a perfect reconstruction of the input provided there is no processing within the subbands. They consist of an analysis filter bank for decomposing the signal in critically sampled subbands and a synthesis filter bank for reconstructing the broadband output. The aliasing in the subbands is eliminated by the synthesis filter bank. Nonlinear methods are used for coding the subband signals. The reconstruction error of the filter bank is negligible compared with the errors due to the coding/decoding process. Using a critically sampled filter bank as a multi-band equalizer, multi-band dynamic range control or multi-band room simulation, the processing in the subbands leads to aliasing at the output. In order to avoid aliasing, a multi-complementary filter bank [Fli92, Zöl92, Fli93] is presented which enables an aliasing-free processing in the subbands and leads to a perfect reconstruction of the output. It allows a decomposition into octave frequency bands which are matched to the human ear.

5.3.1 Principles

Figure 5.51 shows an octave-band filter bank with *critical sampling*. It performs a successive low-pass/high-pass decomposition into half-bands followed by down-

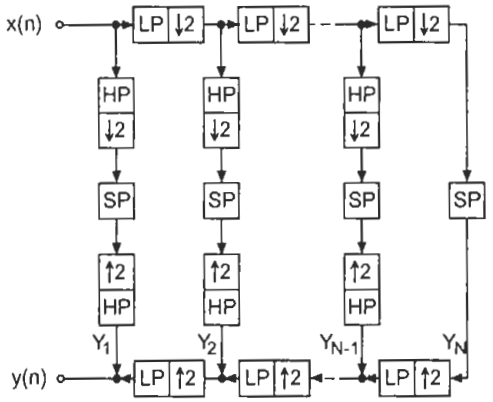


Figure 5.51 Octave-band QMF filter bank (SP = signal processing, LP = low-pass, HP = high-pass).

sampling by a factor 2. The decomposition leads to the subbands Y_1 to Y_N (see

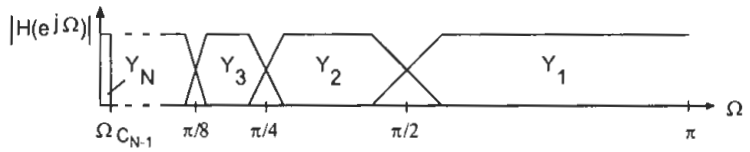


Figure 5.52 Octave-frequency bands.

Fig. 5.52). The transition frequencies of this decomposition are given by

$$\Omega_{Ck} = \frac{\pi}{2} 2^{-k+1} \quad \text{with} \quad k = 1, 2, \cdots, N-1. \tag{5.222}$$

In order to avoid aliasing in subbands, a modified octave-band filter bank is considered which is shown in Fig. 5.53 for a 2-band decomposition. The cutoff frequency

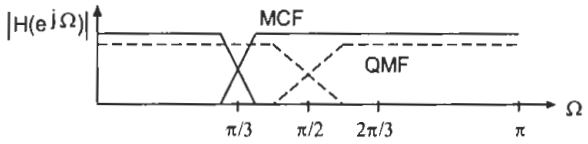


Figure 5.53 2-band decomposition.

of the modified filter bank is moved from $\frac{\pi}{2}$ to a lower frequency. This means, that in downsampling the low-pass branch, no aliasing occurs in the transition band (e.g. cutoff frequency $\frac{\pi}{3}$). The broader high-pass branch cannot be downsampled. A continuation of the described 2-band decomposition leads to the modified octave-band filter bank shown in Fig. 5.54. The frequency bands are depicted in Fig. 5.55 showing that besides the cutoff frequencies

$$\Omega_{Ck} = \frac{\pi}{3} 2^{-k+1} \quad \text{with} \quad k = 1, 2, \cdots, N-1 \tag{5.223}$$

the bandwidth of the subbands is reduced by a factor 2. The high-pass subband Y_1 is an exception.

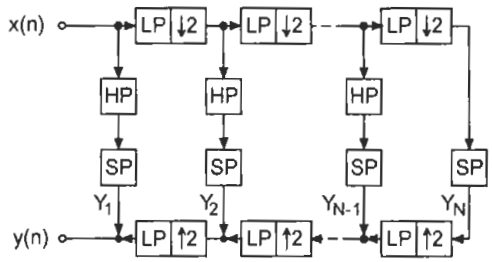


Figure 5.54 Modified octave-band filter bank.

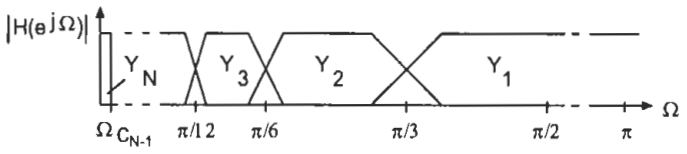


Figure 5.55 Modified octave decomposition.

The special low-pass/high-pass decomposition is carried out by a 2-band complementary filter bank as shown in Fig. 5.56. The frequency responses of a decimation filter $H_D(z)$, interpolation filter $H_I(z)$ and kernel filter $H_K(z)$ are shown in Fig. 5.57.

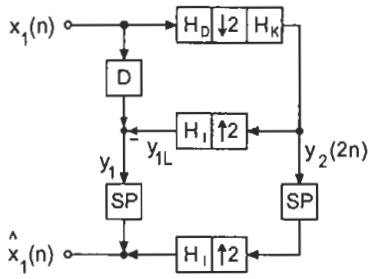


Figure 5.56 2-band complementary filter bank.

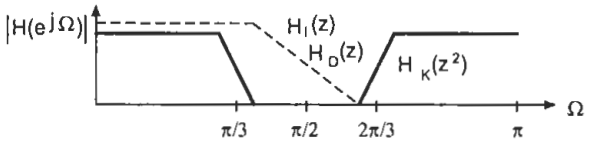


Figure 5.57 Design of $H_D(z)$, $H_I(z)$ and $H_K(z)$.

The low-pass filtering of a signal $x_1(n)$ is done with the help of a decimation filter $H_D(z)$, the downsampler of factor 2 and the kernel filter $H_K(z)$ and leads to $y_2(2n)$. The Z-transform of $y_2(2n)$ is given by

$$\begin{aligned} Y_2(z) &= \frac{1}{2}[H_D(z^{\frac{1}{2}})X_1(z^{\frac{1}{2}})H_K(z) \\ &\quad + H_D(-z^{\frac{1}{2}})X_1(-z^{\frac{1}{2}})H_K(z)]. \end{aligned} \tag{5.224}$$

The interpolated low-pass signal $y_{1L}(n)$ is generated by upsampling by a factor 2 and filtering with the interpolation filter $H_I(z)$. The Z-transform of $y_{1L}(n)$ is given by

$$Y_{1L}(z) = Y_2(z^2)H_I(z) \tag{5.225}$$

$$= \underbrace{\frac{1}{2}H_D(z)H_I(z)H_K(z^2)}_{G_1(z)}X_1(z) + \underbrace{\frac{1}{2}H_D(-z)H_I(z)H_K(z^2)}_{G_2(z)}X_1(-z). \quad (5.226)$$

The high-pass signal $y_1(n)$ is obtained by subtracting the interpolated low-pass signal $y_{1L}(n)$ from the delayed input signal $x_1(n-D)$. The Z-transform of the high-pass signal is given by

$$Y_1(z) = z^{-D}X_1(z) - Y_{1L}(z) \quad (5.227)$$
$$= [z^{-D} - G_1(z)]X_1(z) - G_2(z)X_1(-z). \quad (5.228)$$

The low-pass and high-pass signals are processed individually. The output signal $\hat{x}_1(n)$ is formed by adding the high-pass signal to the upsampled and filtered low-pass signal. With (5.226) and (5.228) the Z-transform of $\hat{x}_1(n)$ can be written as

$$\hat{X}_1(z) = Y_{1L}(z) + Y_1(z) = z^{-D}X_1(z) \quad (5.229)$$

Equation (5.229) shows the perfect reconstruction of the input signal which is delayed by D sampling units.

The extension to N subbands and performing the kernel filter using complementary techniques [Ram88, Ram90] leads to the multi-complementary filter bank as shown in Fig. 5.58. Delays are integrated in the high-pass (Y_1) and band-pass subbands (Y_2 to Y_{N-2}) in order to compensate the group delay. The filter structure consists of N horizontal stages. The kernel filter is implemented as a complementary filter in S vertical stages. The design of the latter shall be discussed later on. The vertical delays in the extended kernel filters (EKF_1 to EKF_{N-1}) compensate group delays caused by forming the complementary component. At the end of each of these vertical stages is the kernel filter H_K. With

$$z_k = z^{2^{-(k-1)}} \quad \text{and} \quad k = 1, \cdots, N \quad (5.230)$$

the signals $\hat{X}_k(z_k)$ can be written as a function of the signals $X_k(z_k)$ as

$$\hat{\mathbf{X}} = \mathrm{diag}\,[z_1^{-D_1} \quad z_2^{-D_2} \ldots \quad z_N^{-D_N}]\mathbf{X}, \quad (5.231)$$

with

$$\hat{\mathbf{X}} = [\hat{X}_1(z_1) \quad \hat{X}_2(z_2) \quad \ldots \quad \hat{X}_N(z_N)]^T$$
$$\mathbf{X} = [X_1(z_1) \quad X_2(z_2) \quad \ldots \quad X_N(z_N)]^T$$

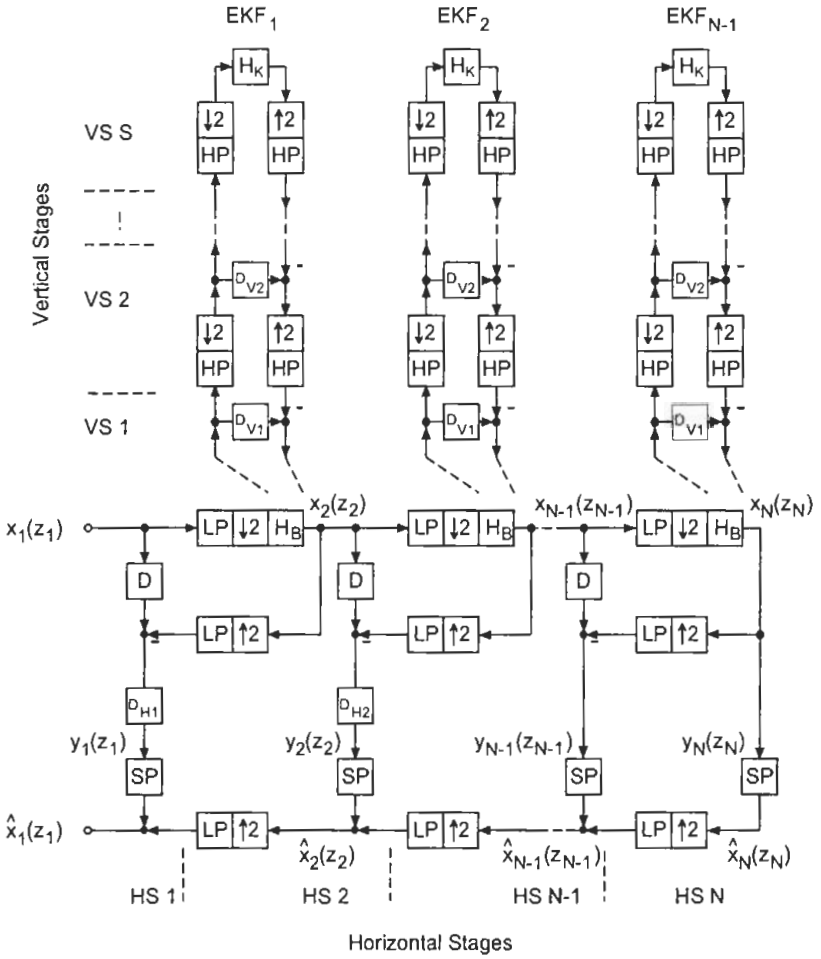


Figure 5.58 Multi-complementary filter bank.

and with $k = N - l$ the delays are given by

$$D_{k=N} = 0 \tag{5.232}$$

$$D_{k=N-l} = 2D_{N-l+1} + D \qquad l = 1, ..., N-1. \tag{5.233}$$

Perfect reconstruction of the input signal can be achieved if the horizontal delays D_{Hk} are given by

$$\begin{aligned} D_{H_{k=N}} &= 0 \\ D_{H_{k=N-1}} &= 0 \\ D_{H_{k=N-l}} &= 2D_{N-l+1} \qquad l = 2, ..., N-1. \end{aligned}$$

The implementation of the extended vertical kernel filters is done by calculating complementary components as shown in Fig. 5.59. After upsampling, interpolating

with a high-pass HP (Fig. 5.59b) and forming the complementary component, the kernel filter H_K with frequency response as in Fig. 5.59a becomes low-pass with frequency response as illustrated in Fig. 5.59c. The slope of the filter characteristic remains constant whereas the cutoff frequency is doubled. A subsequent upsampling with an interpolation high-pass (Fig. 5.59d) and complement filtering leads to the frequency response in Fig. 5.59e. With the help of this technique, the kernel filter is implemented at a reduced sampling rate. The cutoff frequency is moved to a desired cutoff frequency by using decimation/interpolation stages with complement filtering.

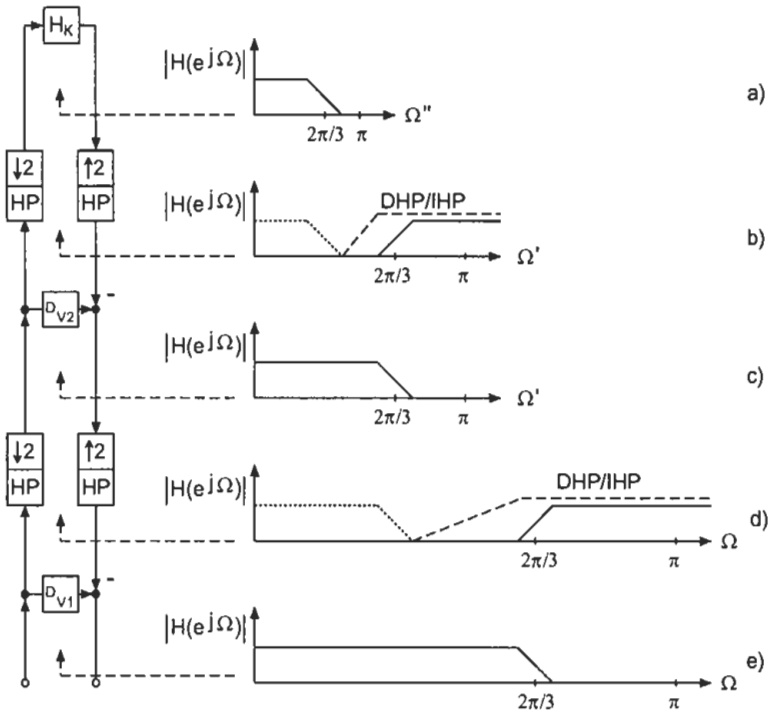


Figure 5.59 Multirate complementary filter.

Computational Complexity. For an N-band multi-complementary filter bank with $N-1$ decomposition filters where each is implemented by a kernel filter with S stages, the horizontal complexity is given by:

$$\mathrm{HC} = \mathrm{HC}_1 + \mathrm{HC}_2 \left(\frac{1}{2} + \frac{1}{4} + \ldots + \frac{1}{2^N} \right). \tag{5.234}$$

HC_1 denotes the number of operations that are carried out at the input sampling rate. These operations occur in the horizontal stage HS_1 (see Fig. 5.58). HC_2 denotes the number of operations (horizontal stage HS_2) that are performed at half of the sampling rate. The number of operations in the stages from HS_2 to

HS_N are approximately identical but are calculated at sampling rates that are successively halved.

The complexities VC_1 to VC_{N-1} of the vertical kernel filters EKF_1 to EKF_{N-1} are calculated as

$$\begin{aligned}
VC_1 &= \frac{1}{2}V_1 + V_2\left(\frac{1}{4}+\frac{1}{8}+\ldots+\frac{1}{2^{S+1}}\right)\\
VC_2 &= \frac{1}{4}V_1 + V_2\left(\frac{1}{8}+\frac{1}{16}+\ldots+\frac{1}{2^{S+2}}\right) = \frac{1}{2}VC_1\\
VC_3 &= \frac{1}{8}V_1 + V_2\left(\frac{1}{16}+\frac{1}{32}+\ldots+\frac{1}{2^{S+3}}\right) = \frac{1}{4}VC_1\\
VC_{N-1} &= \frac{1}{2^{N-1}}V_1 + V_2\left(\frac{1}{2^N}+\ldots+\frac{1}{2^{S+N-1}}\right) = \frac{1}{2^{N-1}}VC_1,
\end{aligned}$$

where V_1 depicts the complexity of the first stage VS_1 and V_2 is the complexity of the second stage VS_2 (see Fig. 5.58). It can be seen that the total vertical complexity is given by

$$VC = VC_1\left(1+\frac{1}{2}+\frac{1}{4}+\ldots+\frac{1}{2^{N-1}}\right). \tag{5.235}$$

The upper bound of the total complexity results is the sum of horizontal and vertical complexities and can be written as

$$C_{tot} = HC_1 + HC_2 + 2VC_1. \tag{5.236}$$

The total complexity C_{tot} is independent of the number of frequency bands N and vertical stages S. This means that for real-time implementation with finite computation power, any desired number of subbands with however narrow transition bands can be implemented!

5.3.2 Example: 8-Band Multi-complementary Filter Bank

In order to implement the frequency decomposition into the 8 bands shown in Fig. 5.60, the multirate filter structure of Fig. 5.61 is employed. The individual parts of the system provide means of downsampling (D=decimation), upsampling (I=interpolation), kernel filtering (K), signal processing (SP), delays (N_1=Delay 1, N_2=Delay 2) and group delay compensation M_i in the ith band. The frequency decomposition is carried out successively from the highest to the lowest frequency band. In the two lowest frequency bands, a compensation for group delay is not required.

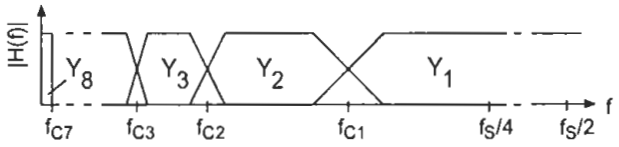


Figure 5.60 Modified octave decomposition of the frequency band.

The slope of the filter response can be adjusted with the complementary filter shown in Fig. 5.61 which consists of one stage. The specifications of an 8-band equalizer are listed in Table 5.10. The stop-band attenuation of the subband filters is chosen to be 100 dB.

Table 5.10 Transition frequencies f_{Ci} and transition bandwidths TB in an 8-band equalizer.

$f_S[kHz]$	$f_{C1}[Hz]$	$f_{C2}[Hz]$	$f_{C3}[Hz]$	$f_{C4}[Hz]$	$f_{C5}[Hz]$	$f_{C6}[Hz]$	$f_{C7}[Hz]$
44.1	7350	3675	1837.5	918.75	459.375	≈ 230	≈ 115
TB$[Hz]$	1280	640	320	160	80	40	20

Filter Design

The design of different decimation and interpolation filters is mainly determined by the transition bandwidth and the stop-band attenuation for the lower frequency band. As an example, a design is made for an 8-band equalizer. The filter structure for both lower frequency bands is illustrated in Fig. 5.62. The design specifications for the kernel low-pass, decimation and interpolation filters are presented in Fig. 5.63.

Kernel Filter Design. The transition bandwidth of the kernel filter is known if the transition bandwidth is given for the lower frequency band. This kernel filter must be designed for a sampling rate of $f_S'' = 44100/(2^8)$. For a given transition bandwidth f_{TB} at a frequency $f'' = f_S''/3$ the normalized pass-band frequency is

$$\frac{\Omega_{Pb}''}{2\pi} = \frac{f'' - f_{TB}/2}{f_S''} \tag{5.237}$$

and the normalized stop-band frequency

$$\frac{\Omega_{Sb}''}{2\pi} = \frac{f'' + f_{TB}/2}{f_S''}. \tag{5.238}$$

With the help of these parameters the filter can be designed.

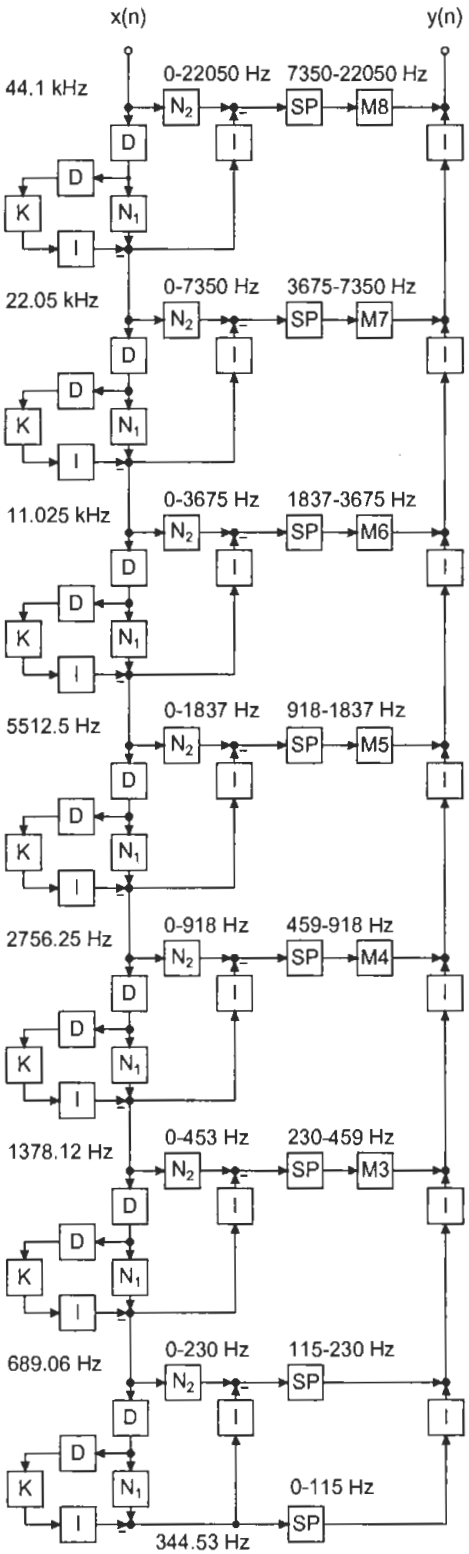


Figure 5.61 Linear phase 8-band equalizer.

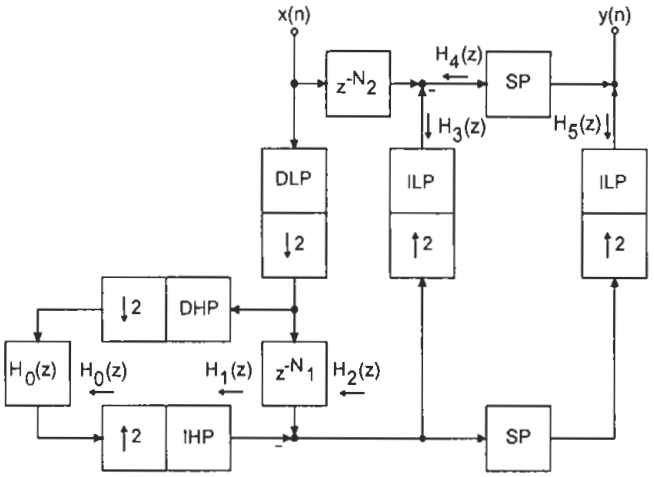


Figure 5.62 Part of a system.

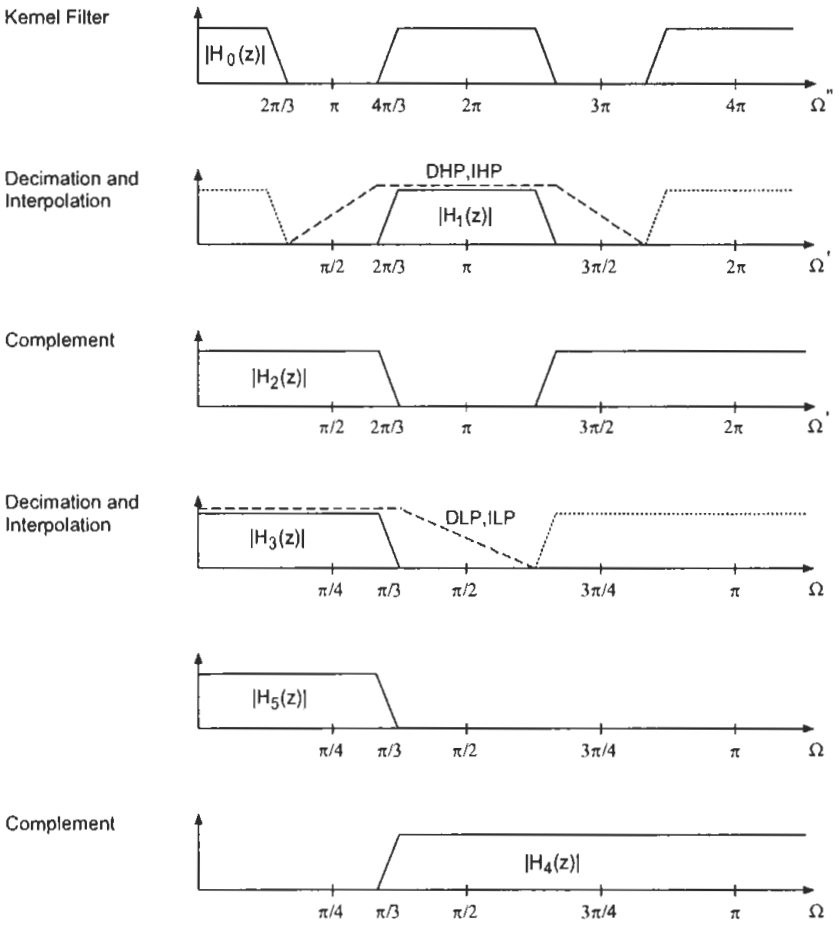


Figure 5.63 Decimation and interpolation filters.

Making use of the Parks-McClellan program the frequency response shown in

Fig. 5.64 is obtained for a transition bandwidth of $f_{TB} = 20$ Hz. The necessary filter length for a stop-band attenuation of 100 dB is 53 taps.

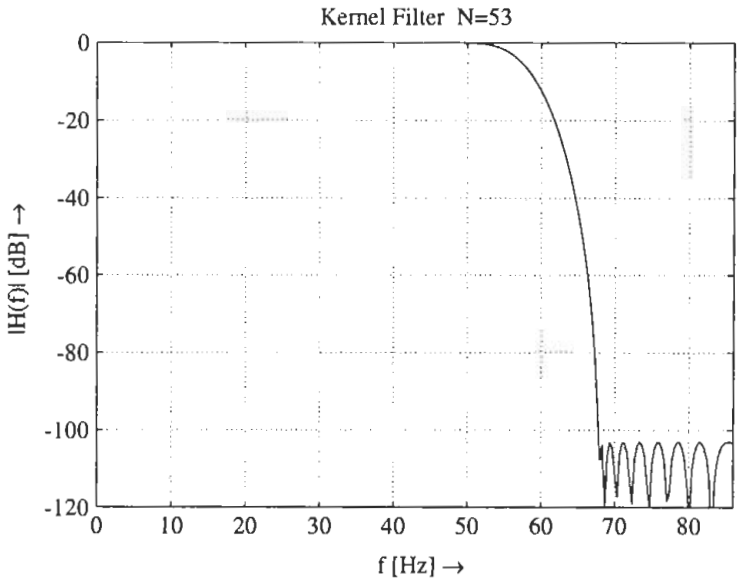


Figure 5.64 Kernel low-pass filter with a transition bandwidth of 20 Hz.

Decimation and Interpolation High-pass Filter. These filters are designed for a sampling rate of $f_S' = 44100/(2^7)$ and are half-band filters as illustrated in Fig. 5.63. First a low-pass filter is designed followed by a high-pass to low-pass transformation. For a given transition bandwidth f_{TB}, the normalized pass-band frequency is

$$\frac{\Omega_{Pb}'}{2\pi} = \frac{f'' + f_{TB}/2}{f_S'} \tag{5.239}$$

and the normalized stop-band frequency is given by

$$\frac{\Omega_{Sb}'}{2\pi} = \frac{2f'' - f_{TB}/2}{f_S'}. \tag{5.240}$$

With these parameters the design of a half-band filter is carried out. Figure 5.65 shows the frequency response. The necessary filter length for a stop-band attenuation of 100 dB is 55 taps.

Decimation and Interpolation Low-pass Filter. These filters are designed for a sampling rate of $f_S = 44100/(2^6)$ and are also half-band filters. For a given transition bandwidth f_{TB}, the normalized pass-band frequency is

$$\frac{\Omega_{Pb}}{2\pi} = \frac{2f'' + f_{TB}/2}{f_S} \tag{5.241}$$

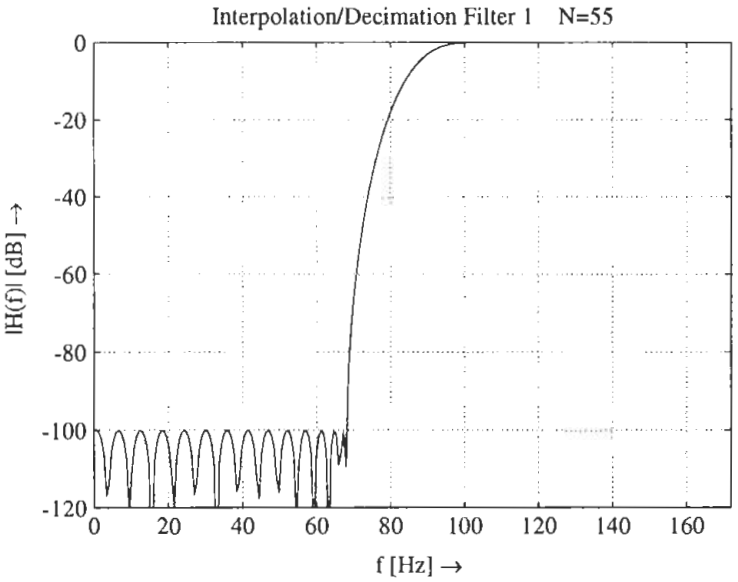


Figure 5.65 Decimation and interpolation high-pass filter.

and the normalized stop-band frequency is given by

$$\frac{\Omega_{Sb}}{2\pi} = \frac{4f'' - f_{TB}/2}{f_S}. \tag{5.242}$$

With these parameters the design of a half-band filter is carried out. Figure 5.66 shows the frequency response. The necessary filter length for a stop-band attenuation of 100 dB is 43 taps. These filter designs are used in every decomposition stage so that the transition frequencies and bandwidths are obtained as listed in Table 5.10.

Memory Requirements and Latency Time. The memory requirements depend directly on the transition bandwidth and the stop-band attenuation. Here, the memory operations for the actual kernel, decimation and interpolation filters have to be differentiated from the group delay compensations in the frequency bands. The compensating group delay N_1 for decimation and interpolation high-pass filters of order $O_{\text{DHP/IHP}}$ is calculated with the help of the kernel filter order O_{KF} according to

$$N_1 = O_{\text{KF}} + O_{\text{DHP/IHP}}. \tag{5.243}$$

The group delay compensation N_2 for the decimation and interpolation low-pass filters of order $O_{\text{DLP/ILP}}$ is given by

$$N_2 = 2N_1 + O_{\text{DLP/ILP}}. \tag{5.244}$$

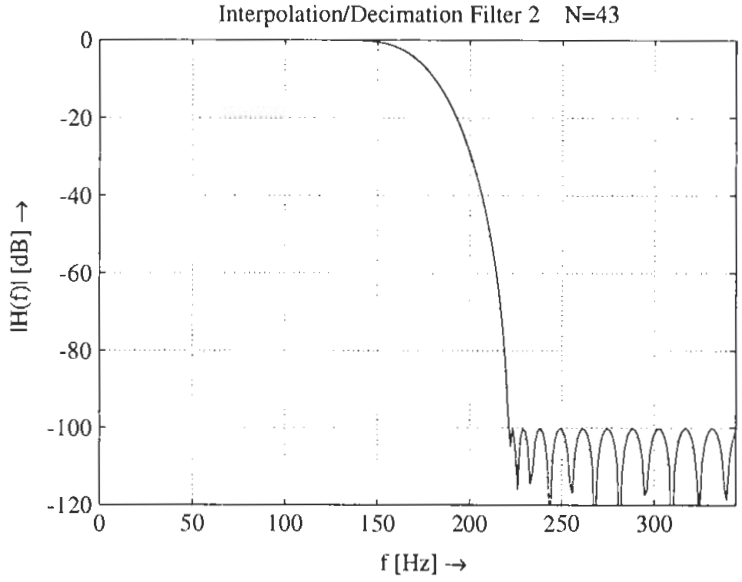


Figure 5.66 Decimation and interpolation low-pass filter.

The delays $M_3 \ldots M_8$ in the individual frequency bands are calculated recursively starting from the two lowest frequency bands:

$$M_3 = 2N_2 \tag{5.245}$$
$$M_4 = 6N_2 \tag{5.246}$$
$$M_5 = 14N_2 \tag{5.247}$$
$$M_6 = 30N_2 \tag{5.248}$$
$$M_7 = 62N_2 \tag{5.249}$$
$$M_8 = 126N_2. \tag{5.250}$$

Memory Requirement (Static RAM) per Decomposition Stage.

Table 5.11 Memory requirements for static RAM.

Kernel filter	O_{KF}
DHP/IHP	$2 \cdot O_{\mathrm{DHP/IHP}}$
DLP/ILP	$3 \cdot O_{\mathrm{DLP/ILP}}$
N_1	$O_{\mathrm{KF}} + O_{\mathrm{DHP/IHP}}$
N_2	$2 \cdot N_1 + O_{\mathrm{DLP/ILP}}$

Memory Requirement for Group Delay (Dynamic RAM).

$$\mathrm{CGD} = \sum M_i = 240N_2 \tag{5.251}$$

Latency Time.

$$t_L = \frac{M_8}{44100} 10^3 \quad [\text{ms}] \tag{5.252}$$

The presented example requires SRAM = 4522 and DRAM = 60960 memory locations. The latency time is $t_L = 725$ ms.

Chapter 6

Room Simulation

Room simulation artificially reproduces the acoustics of a room. The foundations of room acoustics are found in [Cre78, Kut91]. Room simulation is mainly used for post-processing signals in which a microphone is located in the vicinity of an instrument or a voice. The direct signal, without additional room impression, is mapped to a certain acoustical room, for example a concert hall or a church. In terms of signal processing, the post-processing of an audio signal with room simulation corresponds to the convolution of the audio signal with a room impulse response. The room impulse response between two points in a room can be classified as shown in Fig. 6.1. The impulse response consists of the direct signal, early reflections (from walls) and subsequent reverberation. The number of early

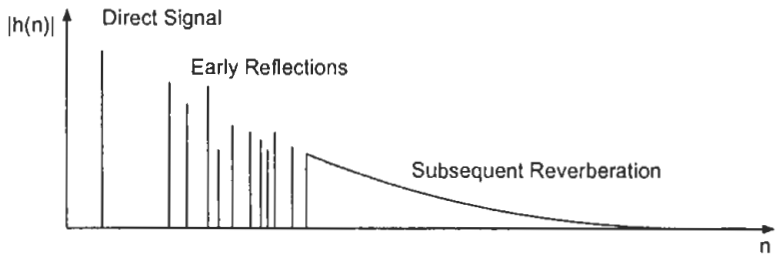


Figure 6.1 Classification of room impulse response as direct signal, early reflections and subsequent reverberation.

reflections continuously increases with time and leads to a random signal with exponential decay called subsequent reverberation. The *reverberation time* (decrease of sound pressure level by 60 dB) can be calculated using the geometry of the room

and the partial areas that absorb sound in the room according to

$$T_{60} = 0.163\frac{V}{\alpha S} = \frac{0.163}{[\mathrm{m/s}]}\frac{V}{\sum_n \alpha_n S_n} \tag{6.1}$$

T_{60} = reverberation time in [s]
V = volume of the room [m^3]
S_n = partial areas [m^2]
α_n = absorption coefficient of partial area S_n.

The geometry of the room also determines the eigenfrequencies of a three-dimensional rectangular room:

$$f_e = \frac{c}{2}\sqrt{\left(\frac{n_x}{l_x}\right)^2 + \left(\frac{n_y}{l_y}\right)^2 + \left(\frac{n_z}{l_z}\right)^2} \tag{6.2}$$

with

n_x, n_y, n_z	integer number of half waves (0,1,2,...)
l_x, l_y, l_z	dimensions of a rectangular room
c	sound velocity.

For larger rooms, the eigenfrequencies start from very low frequencies. In contrast, the lowest eigenfrequencies of smaller rooms are shifted towards higher frequencies. The mean frequency between two extrema of the frequency response of a large room is approximately inversely proportional to the reverberation time [Schr87]:

$$\Delta f \sim 1/T_{60}. \tag{6.3}$$

The distance between two eigenfrequencies decreases with increasing number of half waves. Above a *critical frequency*

$$f_c > 4000\sqrt{T_{60}/V} \tag{6.4}$$

the density of eigenfrequencies becomes so large that they overlap each other [Schr87].

Calculation of room impulse responses with model-based methods. The methods for analytically determining a room impulse response are based on the ray tracing model [Schr70] or image model [All79]. In case of the ray tracing model, a point source with radial emission is assumed. The path length of rays and the absorption coefficients of walls, roofs and floors are used to determine the room impulse response (see Fig. 6.2). For the image model, image rooms with

secondary image sources are formed which in turn have further image rooms and image sources. The summation of all image sources with corresponding delays and attenuations provides the estimated room impulse response. Both methods are applied in room acoustics to get insight into the acoustical properties when planning concert halls, theaters etc.

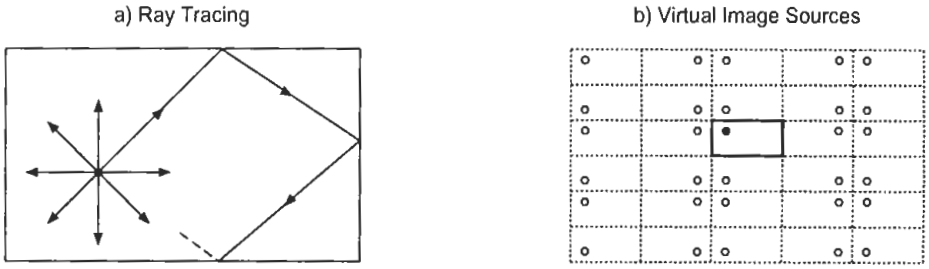


Figure 6.2 Model-based methods for calculating room impulse responses.

Measurement of room impulse response by pseudo-random sequence. The direct measurement of a room impulse response is carried out by impulse excitation. Better measurement results are obtained by correlation measurement of room impulse responses by using pseudo-random sequences as the excitation signal. Pseudo-random sequences can be generated by feedback shift registers [Mac76]. The pseudo-random sequence is periodic with period $L = 2^N - 1$ where N is the number of states of the shift register. The autocorrelation function (ACF) of such a random sequence is given by

$$\tilde{r}_{xx}(n) = \begin{cases} a^2 & n = 0, L, 2L, \ldots \\ \frac{-a^2}{L} & \text{elsewhere} \end{cases}, \tag{6.5}$$

where a is the maximum value of the pseudo-random sequence. The ACF also has a period L. After going through a DA converter, the pseudo-random signal is fed through a loudspeaker into a room (see Fig. 6.3).

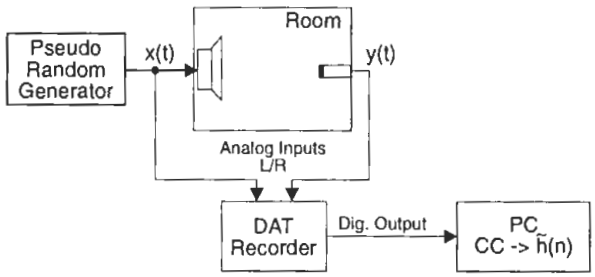


Figure 6.3 Measurement of room impulse response with pseudo-random signal $x(t)$.

At the same time, the pseudo-random signal and the room signal captured by a microphone are recorded on a DAT recorder. The impulse response is obtained

with the cyclic cross-correlation

$$\tilde{r}_{xy}(n) = \tilde{r}_{xx}(n) * h(n) \approx \tilde{h}(n). \tag{6.6}$$

For the measurement of room impulse responses it has to be considered that the periodic length of the pseudo-random sequence must be longer than the length of the room impulse response. Otherwise, aliasing in the periodic cross-correlation occurs. For improving the signal-to-noise ratio of the measurement, the average of several periods of the cross-correlation is calculated.

The just described methods provide means for calculating the impulse response out of the geometry of a room and for measuring the impulse response of a real room. The reproduction of such an impulse response is basically possible with the help of the *fast convolution* method as described in chapter 5. However, owing to the computational complexity, the implementation requires a multiprocessor system or specially manufactured integrated circuits. In contrast, the following sections deal with networks which do not generate the exact room impulse response, but offer, with reasonable complexity, a satisfactory solution in terms of acoustic aspects.

6.1 Early Reflections

Early reflections decisively affect room perception. *Spatial impression* is produced by early reflections which reach the listener laterally. The significance of lateral reflections in creating *spatial impression* was investigated by Barron [Bar71, Bar82]. Fundamental investigations of concert halls and their different acoustics are described by Ando [And85].

6.1.1 Ando's Investigations

The results of the investigations by Ando are summarized in the following:

- Preferred *delay time of a single reflection*: with the ACF of the signal, the delay is determined from $|r_{xx}(\Delta t_1)| = 0.1 \cdot r_{xx}(0)$.
- Preferred *direction of a single reflection*: $\pm(55° \pm 20°)$.
- Preferred *amplitude of a single reflection*: $A_1 = \pm 5$ dB.
- Preferred *spectrum of a single reflection*: no spectral shaping.

- Preferred *delay time of a second reflection*: $\Delta t_2 = 1.8 \cdot \Delta t_1$.
- Preferred *reverberation time*: $T_{60} = 23 \cdot \Delta t_1$.

These results show that in terms of perception, a preferred pattern of reflections as well as the reverberation time depend decisively on the audio signal. Hence for different audio signals like classical music, pop music, speech or musical instruments entirely different requirements for early reflections and reverberation time have to be considered.

6.1.2 Gerzon Algorithm

The commonly used method of simulating early reflections is shown in Figs. 6.4 and 6.5. The signal is weighted and fed into a system generating early reflections, followed by an addition to the input signal. The first M reflections are imple-

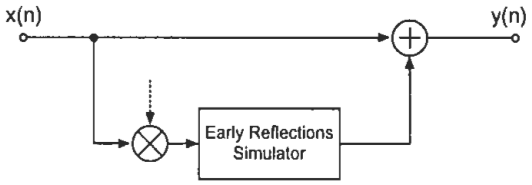


Figure 6.4 Simulation of early reflections.

mented by reading samples from a delay line and weighting these samples with a corresponding factor g_i (see Fig. 6.5). The design of a system for simulating early

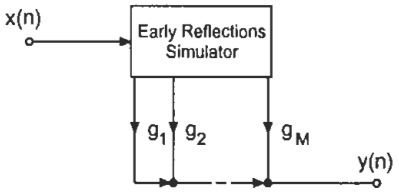


Figure 6.5 Early reflections.

reflections will now be described as proposed by Gerzon [Ger92].

Craven Hypothesis. The Craven hypothesis [Ger92] states that the human perception of the distance to a sound source is evaluated with the help of the amplitude and delay time ratios of the direct signal and early reflections as given by

$$g = \frac{d}{d'} \tag{6.7}$$

$$T_D = \frac{d' - d}{c} \tag{6.8}$$

$$\Rightarrow d = \frac{cT_D}{g^{-1} - 1} \tag{6.9}$$

with

d	distance of source
d'	distance of image source of the first reflection
g	relative amplitude of direct signal to first reflection
c	sound velocity
T_D	relative delay time of first reflection to direct signal.

Without a reflection, human beings are not able to determine the distance d to a sound source. The extended Craven hypothesis includes the absorption coefficient r for determining

$$g = \frac{d}{d'}\exp(-rT_D) \tag{6.10}$$

$$T_D = \frac{d' - d}{c} \tag{6.11}$$

$$\rightarrow d = \frac{cT_D}{g^{-1}\exp(-rT_D) - 1} \tag{6.12}$$

$$\rightarrow g = \frac{\exp(-rT_D)}{1 + cT_D/d}. \tag{6.13}$$

For a given reverberation time T_{60}, the absorption coefficient can be calculated by using $\exp(-rT_{60}) = 1/1000$ according to

$$r = (\ln 1000)/T_{60}. \tag{6.14}$$

With the relationships (6.11) and (6.13), the parameters for an early reflections simulator as shown in Fig. 6.4 can be determined.

Gerzon's Distance Algorithm. For a system simulating early reflections produced by more than one sound source, Gerzon's distance algorithm can be used [Ger92], where several sound sources are placed with different distances as well as in the stereo position into a stereophonic sound field. An application of this technique is mainly used in multichannel mixing consoles.

By shifting a sound source by $-\delta$ (decrease of relative delay time) it follows that from the relative delay time of the first reflection $T_D - \delta/c = \frac{d' - (d+\delta)}{c}$, and

the relative amplitude according to (6.13)

$$g_\delta = \frac{1}{1+\frac{c(T_D-\delta/c)}{d+\delta}}\exp(-r(T_D-\delta/c)) = \left[\frac{d+\delta}{d}\exp(r\delta/c)\right]\frac{\exp(-rT_D)}{1+cT_D/d}. \tag{6.15}$$

This results in a delay and a gain factor for the direct signal (see Fig. 6.6) as given by

$$d_2 = d+\delta \tag{6.16}$$

$$t_D = \delta/c \tag{6.17}$$

$$g_D = \frac{d}{d+\delta}\exp(-r\delta/c). \tag{6.18}$$

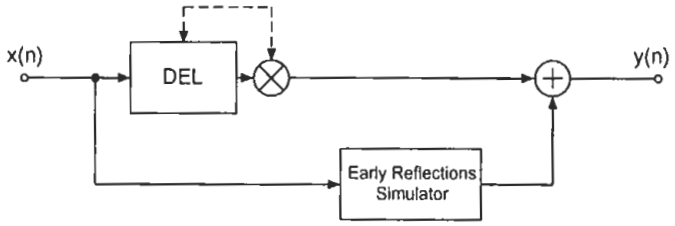


Figure 6.6 Delay and weighting of the direct signal.

By shifting a sound source by $+\delta$ (increase of relative delay time) the relative delay time of the first reflection is $T_D - \delta/c = \frac{d'-(d-\delta)}{c}$. As a consequence, a delay and a gain factor for the effect signal (see Fig. 6.7) are given by

$$d_2 = d-\delta \tag{6.19}$$

$$t_E = \delta/c \tag{6.20}$$

$$g_E = \frac{d}{d+\delta}\exp(-r\delta/c). \tag{6.21}$$

Using two delay systems in the direct signal as well as in the reflection path, two

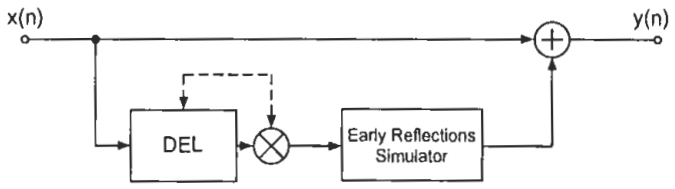


Figure 6.7 Delay and weighting of effect signal.

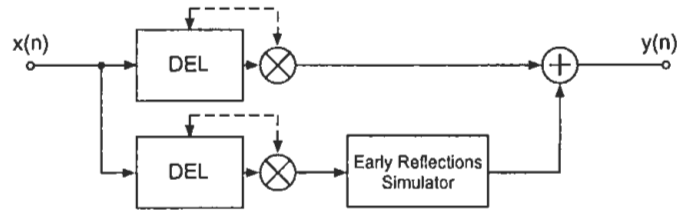


Figure 6.8 Coupled factors and delays.

coupled weighting factors and delay lengths (see Fig. 6.8) can be obtained. For multichannel applications like digital mixing consoles, the scheme in Fig. 6.9 is suggested by Gerzon [Ger92]. Only one system for implementing early reflections is necessary.

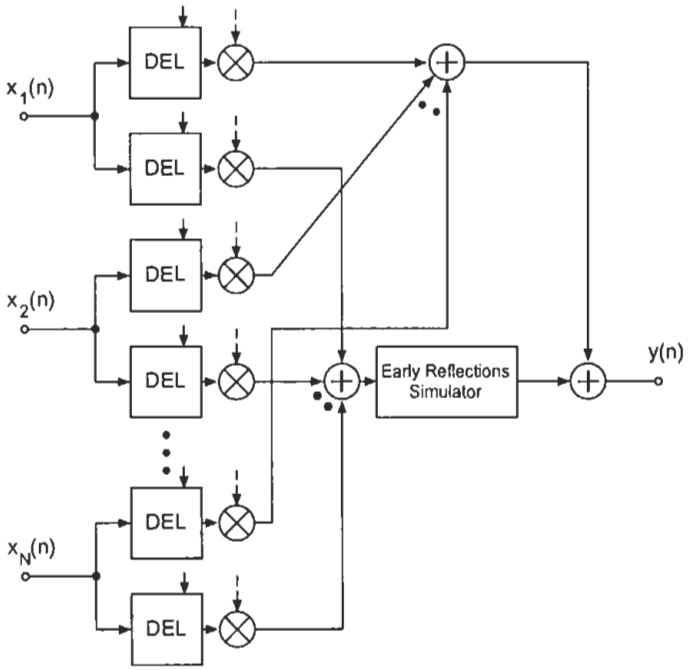


Figure 6.9 Multichannel application.

Stereo Implementation. In many applications, stereo signals have to be processed (see Fig. 6.10). For this, reflections from both sides with positive and negative angles are implemented to avoid stereo displacements. The weighting is done with

$$\begin{aligned} g_i &= \frac{\exp(-rT_i)}{1 + cT_i/d} \\ \mathbf{G}_i &= g_i \begin{pmatrix} \cos\Theta_i & -\sin\Theta_i \\ \sin\Theta_i & \cos\Theta_i \end{pmatrix}. \end{aligned} \tag{6.22}$$

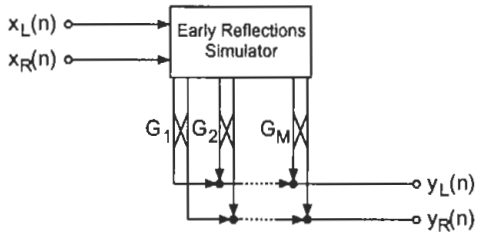


Figure 6.10 Stereo reflections.

For each reflection, a weighting factor and an angle have to be considered.

Generation of early reflection with increasing time density. In [Schr61] it is stated that the time density of reflections increases proportional to the square of time:

$$\text{Number of reflections per second} = (4\pi c^3/V) \cdot t^2. \tag{6.23}$$

After time t_C the reflections have a statistical decay behavior. For a pulse width of Δt, individual reflections overlap after

$$t_C = 5 \cdot 10^{-5} \sqrt{V/\Delta t}. \tag{6.24}$$

For avoiding overlap of reflections, Gerzon [Ger92] suggests the increase of the density of reflections with t^p (for example $p = 1, 0.5$ leads to t or $t^{0.5}$). In the interval $(0, 1]$, with initial value x_0 and a number k between 0.5 and 1 the following procedure is performed:

$$y_i = x_0 + ik (\mathrm{mod}\, 1) \qquad i = 0, 1, \ldots, M-1. \tag{6.25}$$

The numbers y_i in the interval $(0, 1]$ are now transformed to time delays T_i in the interval $[T_{min}, T_{min} + T_{max}]$ by

$$b = T_{min}^{1+p} \tag{6.26}$$

$$a = (T_{max} + T_{min})^{1+p} - b \tag{6.27}$$

$$T_i = (ay_i + b)^{1/(1+p)}. \tag{6.28}$$

The increase of the density of reflections is shown by the example in Fig. 6.11.

6.2 Subsequent Reverberation

This section deals with techniques for reproducing subsequent reverberation. The first approaches by Schroeder [Schr61, Schr62] and their extension by Moorer [Moo78] will be described. Further developments by Stautner and Puckette [Sta82] led to general feedback networks [Ger71, Ger76, Jot91, Rco97] which have a random impulse response with exponential decay.

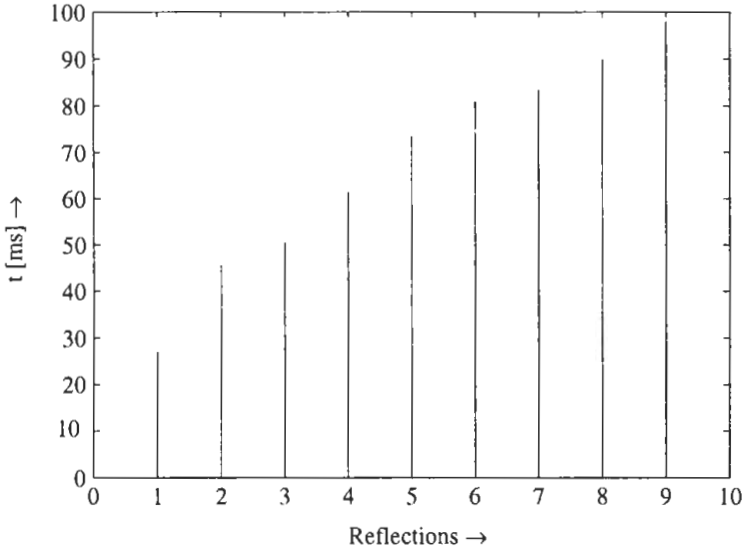


Figure 6.11 Increase of density for 9 reflections.

6.2.1 Schroeder Algorithm

The first software implementations of room simulation algorithms were carried out in 1961 by Schroeder. The basis for simulating an impulse response with exponential decay is a recursive comb filter shown in Fig. 6.12.

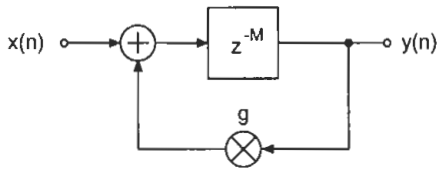


Figure 6.12 Recursive comb filter (g = feedback factor, M = delay length).

The transfer function is given by

$$H(z) = \frac{z^{-M}}{1 - gz^{-M}} \tag{6.29}$$

$$= \sum_{k=0}^{M-1} \frac{A_k}{z - z_k} \tag{6.30}$$

with

$$A_k = \frac{z_k}{Mg} \qquad \text{residues} \tag{6.31}$$

$$z_k = re^{j2\pi k/M} \quad \text{poles} \tag{6.32}$$

$$r = g^{1/M} \quad \text{pole radius.} \tag{6.33}$$

With the correspondence of the Z-transform $a/(z-a) \circ\!\!-\!\!\bullet\, \epsilon(n-1)a^n$ the impulse response is given by

$$H(z) \circ\!\!-\!\!\bullet\, h(n) = \frac{\epsilon(n-1)}{Mg} \sum_{k=0}^{M-1} z_k^n$$

$$h(n) = \frac{\epsilon(n-1)}{Mg} r^n \sum_{k=0}^{M-1} e^{j\Omega_k n}. \tag{6.34}$$

The complex poles are combined as pairs so that the impulse response can be written as

$$h(n) = \frac{\epsilon(n-1)}{Mg} r^n \sum_{k=1}^{\frac{M}{2}-1} \cos\Omega_k n \qquad M \text{ even} \tag{6.35}$$

$$= \frac{\epsilon(n-1)}{Mg} r^n \left[1 + \sum_{k=1}^{\frac{M+1}{2}-1} \cos\Omega_k n\right] \qquad M \text{ uneven.} \tag{6.36}$$

The impulse response is expressed as a summation of cosine oscillations with frequencies Ω_k. These frequencies correspond to the eigenfrequencies of a room. They decay with an exponential envelope r^n, where r is the damping constant (see Fig. 6.14a). The overall impulse response is weighted by $\frac{1}{Mg}$. The frequency response of the comb filter is shown in Fig. 6.14c and is given by

$$|H(e^{j\Omega})| = \sqrt{\frac{1}{1 - 2g\cos(\Omega M) + g^2}}. \tag{6.37}$$

It shows maxima at $\Omega = 2\pi k/M \quad (k = 0, 1, \ldots, M-1)$ of magnitude

$$|H(e^{j\Omega})|_{\max} = \frac{1}{1-g} \tag{6.38}$$

and minima at $\Omega = (2k+1)\pi/M \quad (k = 0, 1, \ldots, M-1)$ of magnitude

$$|H(e^{j\Omega})|_{\min} = \frac{1}{1+g}. \tag{6.39}$$

Another basis of the Schroeder algorithm is the all-pass filter shown in Fig. 6.13 with transfer function

$$H(z) = \frac{z^{-M} - g}{1 - gz^{-M}} \tag{6.40}$$

$$= \frac{z^{-M}}{1 - gz^{-M}} - \frac{g}{1 - gz^{-M}}. \tag{6.41}$$

From Equation (6.41) it can be seen that the impulse response can also be expressed as a summation of cosine oscillations.

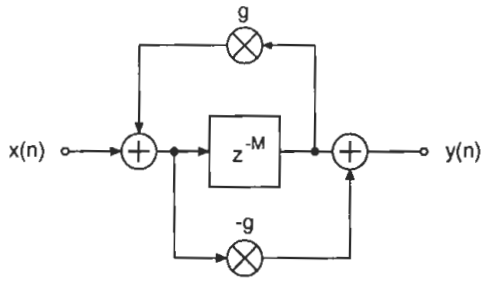


Figure 6.13 All-pass filter (M = delay length).

The impulse responses and the frequency responses of a comb filter and an all-pass filter are presented in Fig. 6.14. Both impulse responses show an exponential decay. A sample in the impulse response occurs every M sampling periods. The density of samples in the impulse responses does not increase with time. For the recursive comb filter, spectral shaping due to the maxima at the corresponding poles of the transfer function is observed.

Frequency Density

The frequency density describes the number of eigenfrequencies per Hertz and is defined for a comb filter [Jot91] as

$$D_f = M \cdot T_S \quad [1/\text{Hz}]. \tag{6.42}$$

A single comb filter gives M resonances in the interval $[0, 2\pi]$, which are separated by a frequency distance of $\Delta f = \frac{f_S}{M}$. In order to increase the frequency density, a parallel circuit (see Fig. 6.15) of P comb filters is used which leads to

$$H(z) = \sum_{p=1}^{P} \frac{z^{-M_p}}{1 - g_p z^{-M_p}} = \left[\frac{z^{-M_1}}{1 - g_1 z^{-M_1}} + \frac{z^{-M_2}}{1 - g_2 z^{-M_2}} + \cdots \right]. \tag{6.43}$$

The choice of the delay systems [Schr62] is suggested as

$$M_1 : M_P = 1 : 1.5 \tag{6.44}$$

and leads to a frequency density

$$D_f = \sum_{p=1}^{P} M_p \cdot T_S = P \cdot \overline{M} \cdot T_S. \tag{6.45}$$

In [Schr62] a necessary frequency density of $D_f = 0.15$ eigenfrequencies per Hertz is proposed.

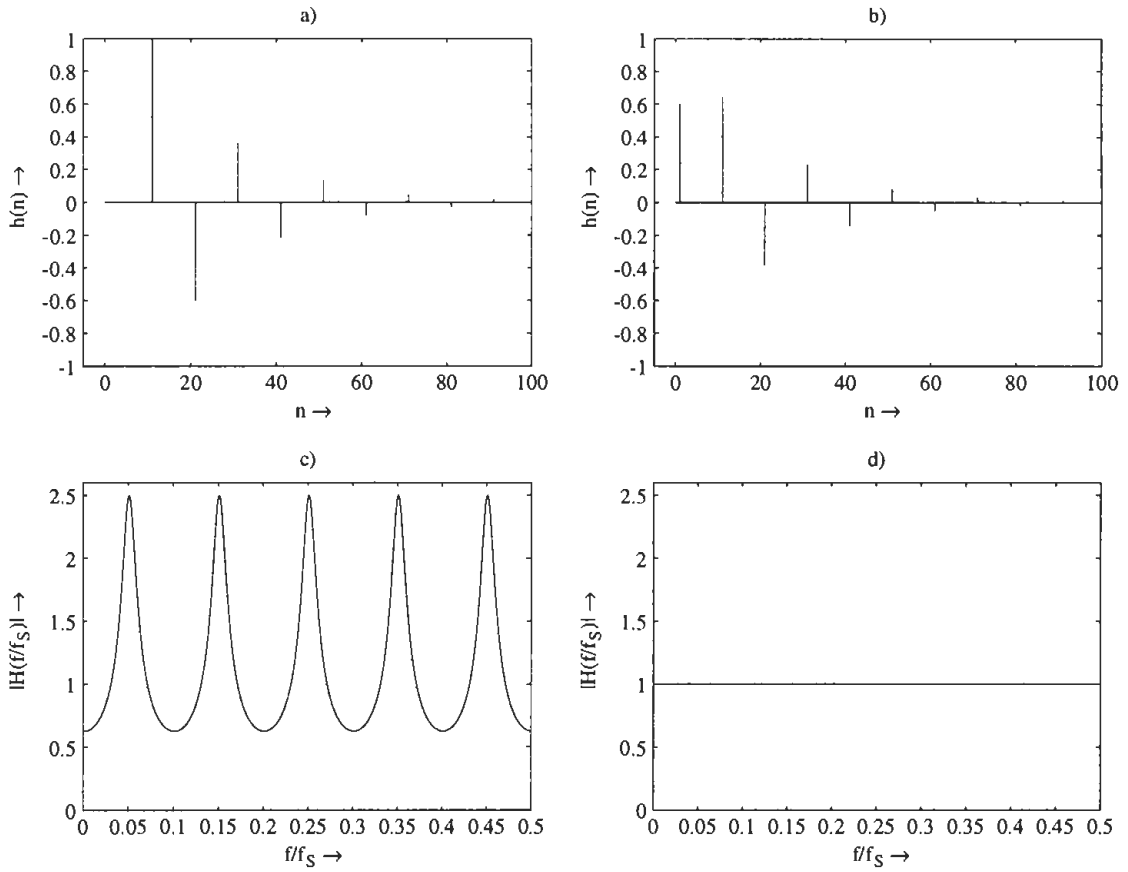


Figure 6.14 a) Impulse response of a comb filter (M = 10, g = -0.6). b) Impulse response of an all-pass filter (M = 10, g = -0.6). c) Frequency response of a comb filter. d) Frequency response of an all-pass filter.

Echo Density

The echo density is the number of reflections per second and is defined for a comb filter [Jot91] as

$$D_t = \frac{1}{M \cdot T_S} \quad [1/\mathrm{s}]. \tag{6.46}$$

For a parallel circuit of comb filters, the echo density is given by

$$D_t = \sum_{p=1}^{P} \frac{1}{M_p \cdot T_S} = P \frac{1}{\overline{M} \cdot T_S}. \tag{6.47}$$

With (6.45) and (6.47), the number P of parallel comb filters and the mean delay length M

$$P = \sqrt{D_f \cdot D_t} \tag{6.48}$$

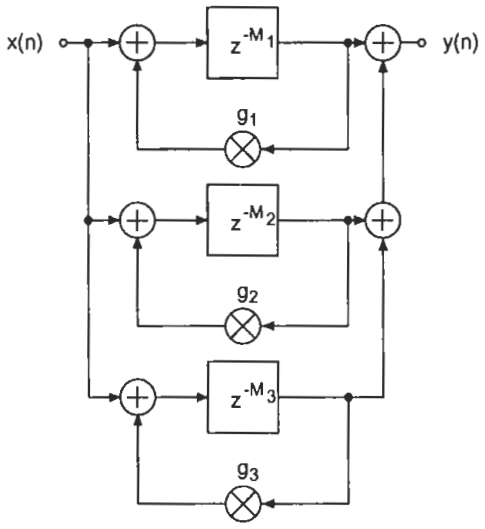


Figure 6.15 Parallel circuit of comb filters.

$$\overline{M}T_S = \sqrt{D_f/D_t} \tag{6.49}$$

are obtained. For a frequency density $D_f = 0.15$ and an echo density $D_t = 1000$ it can be concluded that the number of parallel comb filters is $P = 12$ and the mean delay length is $\overline{M}T_S = 12$ ms. Since the frequency density is proportional to the reverberation time, the number of parallel comb filters has to be increased accordingly.

A further increase of the echo density is achieved by a cascade circuit of P_A all-pass filters (see Fig. 6.16) with transfer function

$$H(z) = \prod_{p=1}^{P_A} \frac{z^{-M_p} - g_p}{1 - g_p z^{-M_p}}. \tag{6.50}$$

These all-pass sections are connected in series with the parallel circuit of comb filters. For a sufficient echo density, 10000 reflections per second are necessary [Gri89].

Avoiding Unnatural Resonances

Since the impulse response of a single comb filter can be described as a sum of M (delay length) decaying sinusoidal oscillations, the short-time FFT of consecutive parts from this impulse response gives the frequency response shown in Fig. 6.17 in the time-frequency domain. Only the maxima are presented. The parallel circuit of comb filters with the condition (6.44) leads to radii of pole distribution as given by

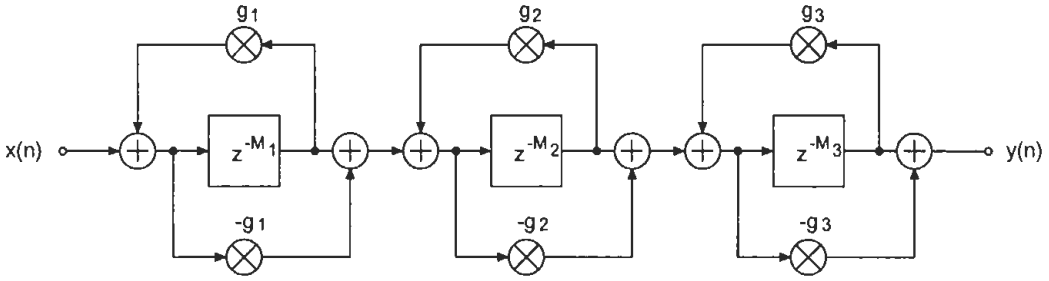


Figure 6.16 Cascade circuit of all-pass filters.

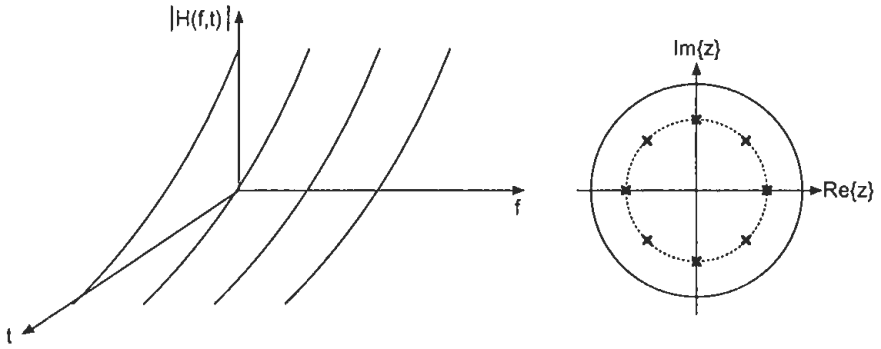


Figure 6.17 Short-time spectra of a comb filter ($M = 8$).

$r_p = g_p^{1/M_p}$ $(p = 1, 2, \ldots, P)$. In order to avoid unnatural resonances, the radii of the pole distribution of a parallel circuit of comb filters must satisfy the condition

$$r_p = \text{const.} = g_p^{1/M_p} \quad \text{for} \quad p = 1, 2, \ldots, P. \tag{6.51}$$

This leads to the short-time spectra and the pole distribution as shown in Fig. 6.18. Figure 6.19 shows the impulse response and the echogram (logarithmic pre-

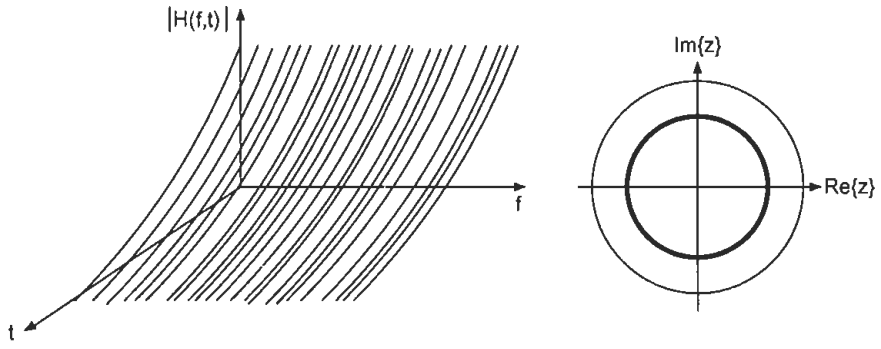


Figure 6.18 Short-time spectra of a parallel circuit of comb filters.

sentation of the amplitude of the impulse response) of a parallel circuit of comb

filters with equal and unequal pole radii. For unequal pole radius, the different decay times of the eigenfrequencies can be seen.

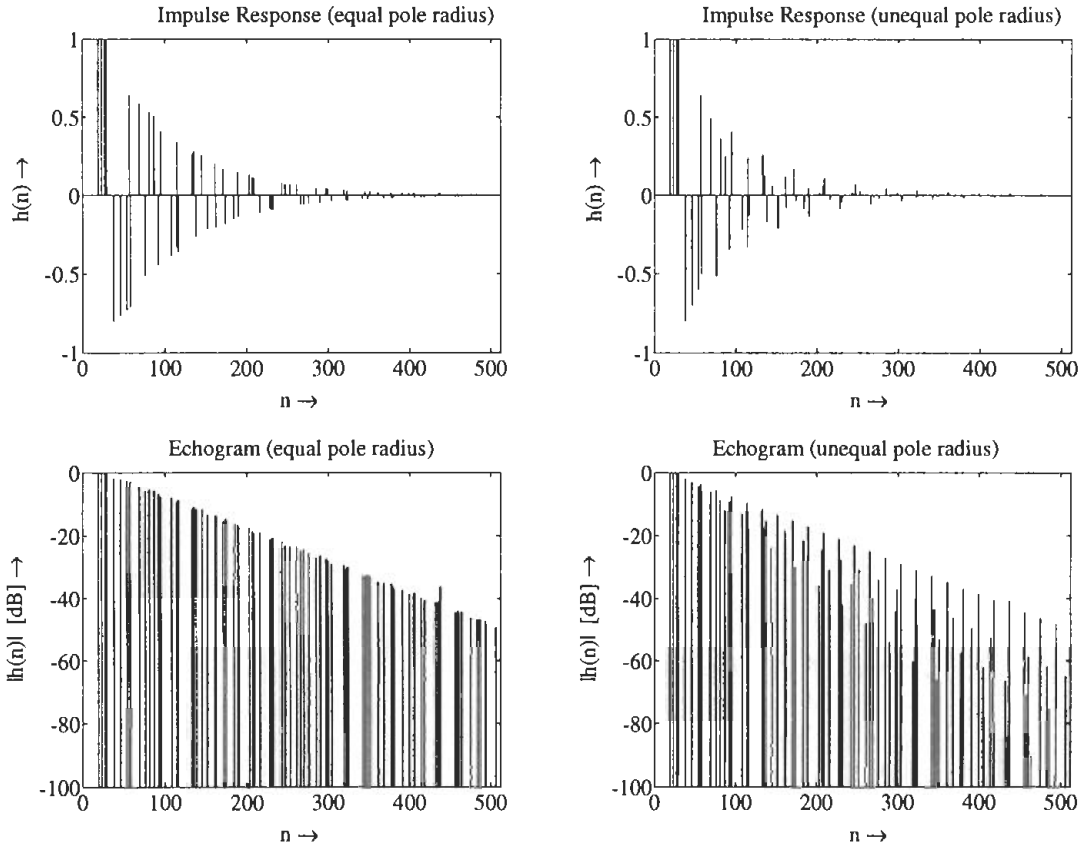


Figure 6.19 Impulse response and echogram.

Reverberation Time

The reverberation time of a recursive comb filter can be adjusted with the feedback factor g which describes the ratio

$$g = \frac{h(n)}{h(n-M)} \tag{6.52}$$

of two different nonzero samples of the impulse response separated by M sampling periods. The factor g describes the decay constant per M samples. The decay constant per sampling period can be calculated from the pole radius $r = g^{1/M}$ and is defined as

$$r = \frac{h(n)}{h(n-1)}. \tag{6.53}$$

The relationship between feedback factor g and pole radius r can also be expressed using (6.52) and (6.53) and is given by

$$g = \frac{h(n)}{h(n-M)} = \frac{h(n)}{h(n-1)} \cdot \frac{h(n-1)}{h(n-2)} \cdots \frac{h(n-(M-1))}{h(n-M)} = r \cdot r \cdot r \cdots r = r^M. \tag{6.54}$$

With the constant radius $r = g_p^{1/M_p}$ and the logarithmic parameters $R = 20\log_{10} r$ and $G_p = 20\log_{10} g_p$, the attenuation per sampling period is given by

$$R = \frac{G_p}{M_p}. \tag{6.55}$$

The reverberation time is defined as decay time of the impulse response to -60 dB. With $\frac{-60}{T_{60}} = \frac{R}{T_S}$, the reverberation time can be written as

$$T_{60} = -60\frac{T_S}{R} = -60\frac{T_S M_p}{G_p} = \frac{3}{\log_{10}|1/g_p|} M_p \cdot T_S. \tag{6.56}$$

The control of reverberation time can either be carried out with the feedback factor g or the delay parameter M. The increase of the reverberation time with factor g is responsible for a pole radius close to the unit circle and, hence, leads to an amplification of maxima of the frequency response (see Equation (6.38)). This leads to a *coloring* of the sound impression. The increase of the delay parameter M, on the other hand, leads to an impulse response whose nonzero samples are far apart from each other, so that individual echoes can be heard. The discrepancy between echo density and frequency density for a given reverberation time can be solved by a sufficient number of parallel comb filters.

Frequency-dependent Reverberation Time

The eigenfrequencies of rooms have a rapid decay for high frequencies. A frequency-dependent reverberation time can be implemented with a low-pass filter

$$H_1(z) = \frac{1}{1 - az^{-1}} \tag{6.57}$$

in the feedback loop of a comb filter. The modified comb filter in Fig. 6.20 has transfer function

$$H(z) = \frac{z^{-M}}{1 - gH_1(z)z^{-M}} \tag{6.58}$$

with the stability criterion

$$\frac{g}{1-a} < 1. \tag{6.59}$$

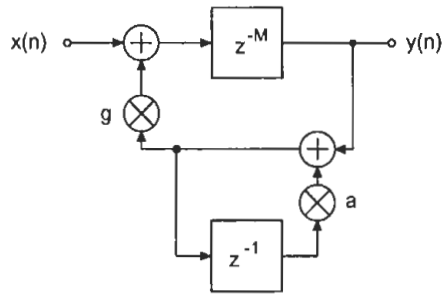


Figure 6.20 Modified low-pass comb filter.

The short-time spectra and the pole distribution of a parallel circuit with low-pass comb filters are presented in Fig. 6.21. Low eigenfrequencies decay slower than higher ones. The circular pole distribution becomes an elliptical distribution where the low-frequency poles are moved towards the unit circle.

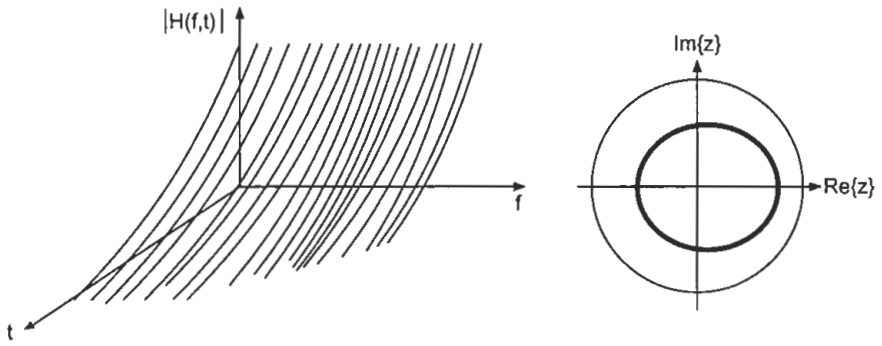


Figure 6.21 Short-time spectra of a parallel circuit of low-pass comb filters.

Stereo Room Simulation

An extension of the Schroeder algorithm was suggested by Moorer [Moo78]. In addition to a parallel circuit of comb filters in series with a cascade of all-pass filters, a pattern of early reflections is generated. Figure 6.22 shows a room simulation system for a stereo signal. The generated room signals $e_L(n)$ and $e_R(n)$ are added to the direct signals $x_L(n)$ and $x_R(n)$. The input of the room simulation is the mono signal $x_M(n) = x_L(n) + x_R(n)$ (sum signal). This mono signal is added to the left and right room signals after going through a delay line DEL1. The total sum of all reflections is fed via another delay line DEL2 to a parallel circuit of comb filters which implements subsequent reverberation. In order to get a high

quality spatial impression, it is necessary to decorrelate the room signals $e_L(n)$ and $e_R(n)$ [Bla74, Bla85]. This can be achieved by taking left and right room signals at different points out of the parallel circuit of comb filters. These room signals are then fed to an all-pass section for increasing the echo density.

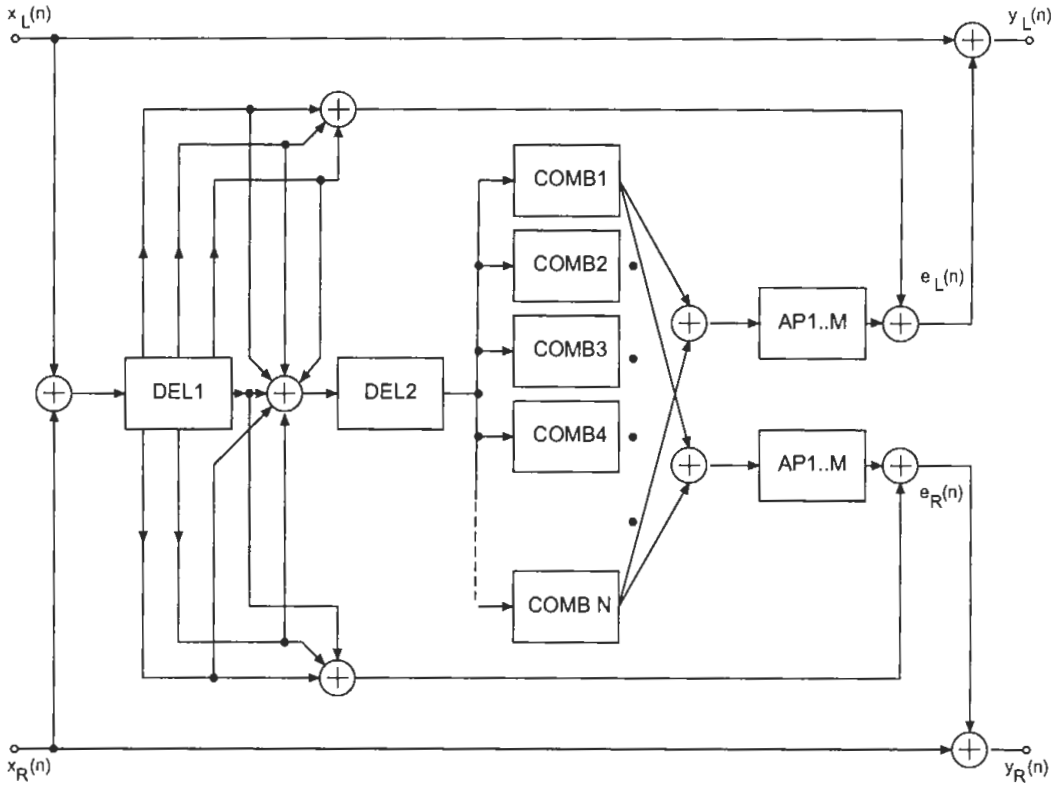


Figure 6.22 Stereo room simulation.

Besides the described system for stereo room simulation in which the mono signal is processed with a room algorithm, it is also possible to perform complete stereo processing of $x_L(n)$ und $x_R(n)$, or to process a mono signal $x_M(n) = x_L(n) + x_R(n)$ and a side (difference) signal $x_S(n) = x_L(n) - x_R(n)$ individually.

6.2.2 General Feedback Systems

Further developments of the comb filter method by Schroeder tried to improve the acoustic quality of reverberation and especially the increase of echo density [Ger71, Ger76, Sta82, Jot91, Jot92, Roc97]. With respect to [Jot91], the general feedback system in Fig. 6.23 is considered. For simplification only three delay systems are shown. The feedback of output signals is carried out with the help of a matrix **A** which feeds back each of the three outputs to the three inputs.

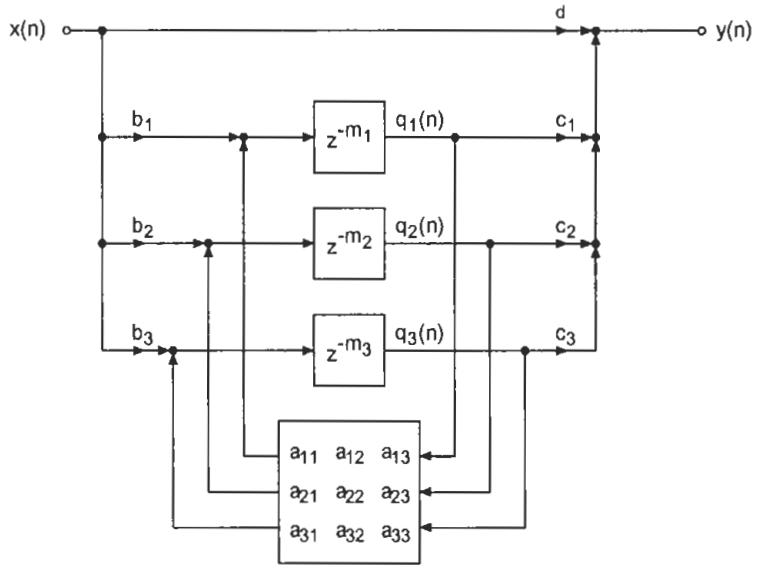


Figure 6.23 General feedback system.

In general, for N delay systems we can write

$$y(n) = \sum_{i=1}^{N} c_i q_i(n) + dx(n) \tag{6.60}$$

$$q_j(n+m_j) = \sum_{i=1}^{N} a_{ij} q_i(n) + b_j x(n) \qquad 1 \le j \le N. \tag{6.61}$$

The Z-transform leads to

$$Y(z) = \mathbf{c}^T \mathbf{Q}(z) + d \cdot X(z) \tag{6.62}$$

$$\begin{aligned} \mathbf{D}(z) \cdot \mathbf{Q}(z) &= \mathbf{A} \cdot \mathbf{Q}(z) + \mathbf{b} \cdot X(z) \\ \rightarrow \mathbf{Q}(z) &= [\mathbf{D}(z) - \mathbf{A}]^{-1} \mathbf{b} \cdot X(z) \end{aligned} \tag{6.63}$$

with

$$\mathbf{Q}(z) = \begin{bmatrix} Q_1(z) \\ \vdots \\ Q_N(z) \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_N \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_N \end{bmatrix} \tag{6.64}$$

and the diagonal delay matrix

$$\mathbf{D}(z) = \mathrm{diag}[z^{-m_1} \cdots z^{-m_N}]. \tag{6.65}$$

With (6.63) the Z-transform of the output is given by

$$Y(z) = \mathbf{c}^T [\mathbf{D}(z) - \mathbf{A}]^{-1} \mathbf{b} \cdot X(z) + d \cdot X(z) \tag{6.66}$$

and the transfer function by

$$H(z) = \mathbf{c}^T[\mathbf{D}(z) - \mathbf{A}]^{-1}\mathbf{b} + d. \tag{6.67}$$

The system is stable if the feedback matrix $\mathbf{A}$ can be expressed as a product of unitary matrix $\mathbf{U}$ ($\mathbf{U}^{-1} = \overline{\mathbf{U}}^T$) and a diagonal matrix with $g_{ii} < 1$ (derivation in [Sta82]). Figure 6.24 shows a general feedback system with input vector $\mathbf{X}(z)$, the output vector $\mathbf{Y}(z)$, a diagonal matrix $\mathbf{D}(z)$ consisting of purely delay systems z^{-m_i} and a feedback matrix $\mathbf{A}$. This feedback matrix consists of an orthogonal matrix $\mathbf{U}$ multiplied by the matrix $\mathbf{G}$ which results in a weighting of the feedback matrix $\mathbf{A}$.

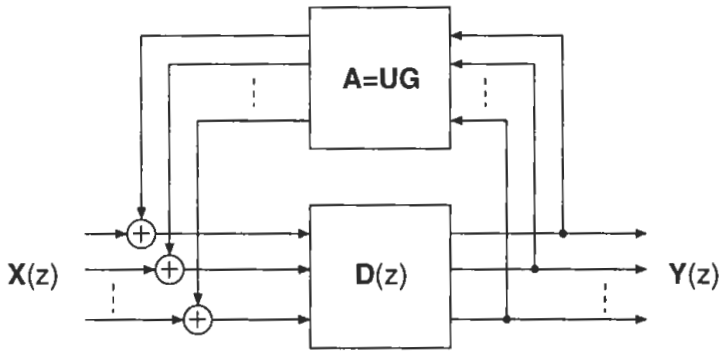


Figure 6.24 Feedback system.

If an orthogonal matrix $\mathbf{U}$ is chosen and the weighting matrix is equal to the unit matrix $\mathbf{G} = \mathbf{I}$, the system in Fig. 6.24 implements a white-noise random signal with Gaussian distribution when a pulse excitation is applied to the input. The time density of this signal slowly increases with time. If the diagonal elements of the weighting matrix $\mathbf{G}$ are less than one, a random signal with exponential amplitude decay results. With the help of the weighting matrix $\mathbf{G}$, the reverberation time can be adjusted. Such a feedback system performs the convolution of an audio input signal with an impulse response of exponential decay.

The effect of the orthogonal matrix $\mathbf{U}$ on the subjective sound perception of subsequent reverberation is of particular interest. A relationship between the distribution of the eigenvalues of the matrix $\mathbf{U}$ on the unit circle and the poles of the system transfer function cannot be described analytically, owing to the high order of the feedback system. In [Her94], it is shown experimentally that the distribution of eigenvalues within the right-hand or left-hand complex plane produces a uniform distribution of poles of the system transfer function. Such a feedback matrix leads to an acoustically improved reverberation. The echo density rapidly increases to the maximum value of one sample per sampling period for a uniform distribution

of eigenvalues. Besides the feedback matrix, additional digital filtering is necessary for spectrally shaping the subsequent reverberation and for implementing frequency-dependent decay times (see [Jot91]). The following example illustrates the increase of the echo density.

Example: First, a system with only one feedback path per comb filter is considered. The feedback matrix is then given by

$$\mathbf{A} = \frac{g}{\sqrt{2}}\mathbf{I}. \tag{6.68}$$

Figure 6.25 shows the impulse response and the amplitude frequency response.

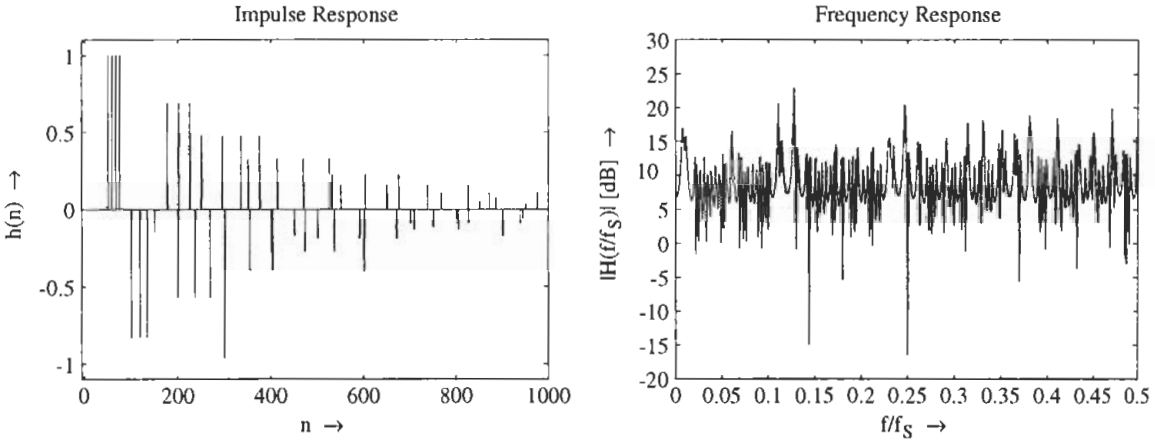


Figure 6.25 Impulse response and frequency response of 4-delay system with a unit matrix as unitary feedback matrix ($g = 0.83$).

With the feedback matrix

$$\mathbf{A} = \frac{g}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix} \tag{6.69}$$

from [Sta82], the impulse response and the corresponding frequency response shown in Fig. 6.26 are obtained. In contrast to Fig. 6.25 an increase of the echo density of the impulse response is noticed.

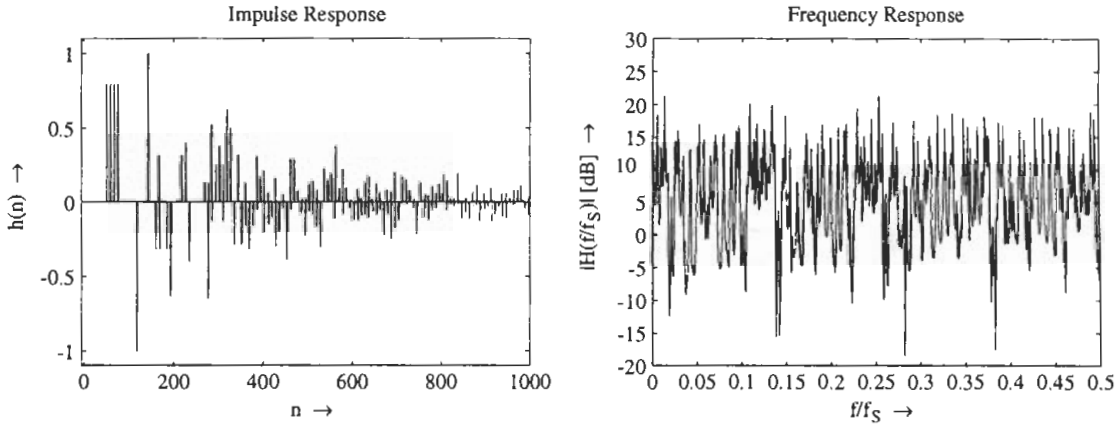


Figure 6.26 Impulse response and frequency response of a 4-delay system with unitary feedback matrix ($g = 0.63$).

6.3 Approximation of Room Impulse Responses

In contrast to the systems for simulation of room impulse responses discussed up to this point, a method is now presented that measures and approximates the room impulse response in one step [Zöl90b, Sch92, Sch93] (see Fig. 6.27). Moreover, it leads to a parametric representation of the room impulse response. Since the decay times of room impulse responses decrease for high frequencies, use is made of multirate signal processing.

The analog system that is to be measured and approximated is excited with a binary pseudo-random sequence $x(n)$ via a DA converter. The resulting room signal gives a digital sequence $y(n)$ after AD conversion. The discrete-time sequence $y(n)$ and the pseudo-random sequence $x(n)$ are each decomposed by an analysis filter bank into subband signals $y_1, ..., y_P$ and $x_1, ..., x_P$ respectively. The sampling rate is reduced in accordance with the bandwidth of the signals. The subband signals $y_1, ..., y_P$ are approximated by adjusting the subband systems $H_1(z) = A_1(z)/B_1(z), ..., H_P(z) = A_P(z)/B_P(z)$. The outputs $\hat{y}_1, ..., \hat{y}_P$ of these subband systems give an approximation of the measured subband signals. With this procedure the impulse response is given in parametric form (subband parameters) and can be directly simulated in the digital domain.

By suitably adjusting the analysis filter bank [Sch94] the subband impulse responses are obtained directly from the cross-correlation function

$$h_i \approx r_{x_i y_i}. \tag{6.70}$$

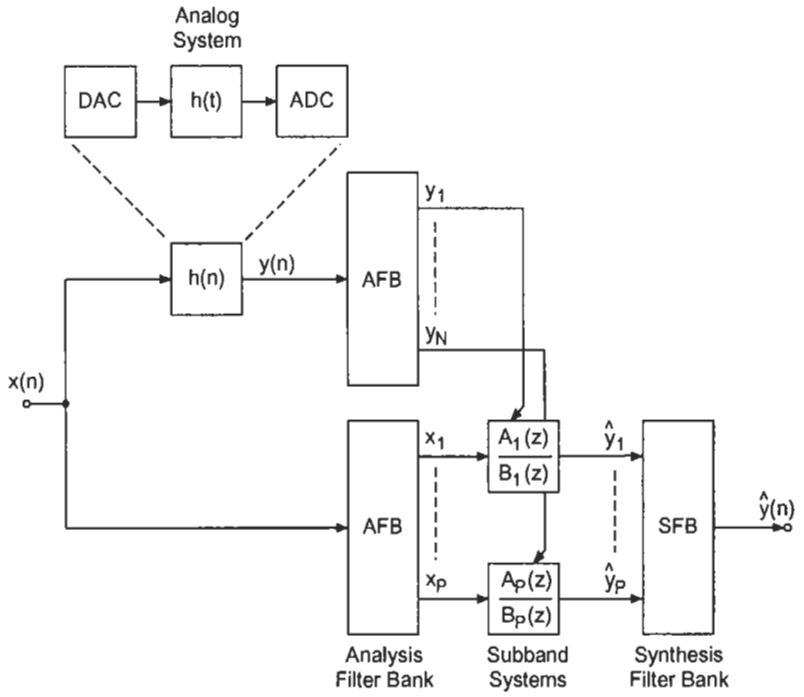


Figure 6.27 System measuring and approximating room impulse responses.

The subband impulse responses are approximated by a nonrecursive filter and a recursive comb filter. The cascade of both filters leads to the transfer function

$$H_i(z) = \frac{b_0 + \ldots + b_{M_i} z^{-M_i}}{1 - g_i z^{-N_i}} = \sum_{n_i=0}^{\infty} h_i(n_i) z^{-n_i}, \tag{6.71}$$

which is set equal to the impulse response in subband i. Multiplying both sides of (6.71) by the denominator $1 - g_i z^{-N_i}$ gives

$$(b_0 + \ldots + b_{M_i} z^{-M_i}) = \left(\sum_{n_i=0}^{\infty} h_i(n_i) z^{-n_i} \right) (1 - g_i z^{-N_i}). \tag{6.72}$$

Truncating the impulse response of each subband to K samples and comparing the coefficients of powers of z on both sides of the equation the following set of equations is obtained:

$$\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_M \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} h_0 & 0 & 0 & \cdots & 0 \\ h_1 & h_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ h_M & h_{M-1} & h_{M-2} & \cdots & h_{M-N} \\ h_{M+1} & h_M & h_{M-1} & \cdots & h_{M-N+1} \\ \vdots & \vdots & \vdots & & \vdots \\ h_K & h_{K-1} & h_{K-2} & \cdots & h_{K-N} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ -g \end{bmatrix}. \tag{6.73}$$

The coefficients $b_0 \ldots b_M$ and g in the above equation are determined in two steps. First, the coefficient g of the comb filter is calculated from the exponentially decaying envelope of the measured subband impulse response. The vector $[1\,0 \ldots g]^T$ is then used to determine the coefficients $[b_0 b_1 \ldots b_M]^T$.

For the calculation of the coefficient g, we start with the impulse response of the comb filter $H(z) = 1/(1 - gz^{-N})$ given by

$$h(l = Nn) = g^l. \tag{6.74}$$

We further make use of the *integrated impulse response*

$$h_e(k) = \sum_{n=k}^{\infty} h(n)^2 \tag{6.75}$$

defined in [Schr65]. It describes the rest energy of the impulse response at time k. By taking the logarithm of $h_e(k)$, a straight line over time index k is obtained. From the slope of the straight line we use

$$\ln g = N \cdot \frac{\ln h_e(n_1) - \ln h_e(n_2)}{n_1 - n_2} \qquad \text{with} \quad n_1 < n_2 \tag{6.76}$$

to determine the coefficient g [Sch94]. For $M = N$, the coefficients in (6.73) of the numerator polynomial are obtained directly from the impulse response

$$\begin{aligned} b_n &= h_n \qquad \text{for} \quad n = 0, 1, \ldots, M-1 \\ b_M &= h_M - g h_0. \end{aligned} \tag{6.77}$$

Hence, the numerator polynomial of (6.71) is a direct reproduction of the first M samples of the impulse response (see Fig. 6.28). The denominator polynomial approximates the further exponentially decaying impulse response. This method is applied to each subband. The implementation complexity can be reduced by

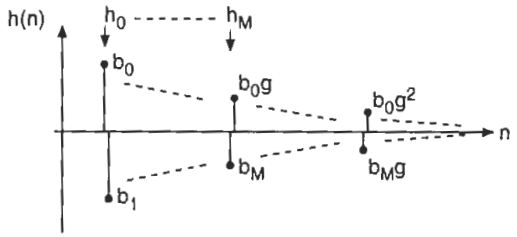


Figure 6.28 Determining model parameters from the measured impulse response.

a factor 10 compared with the direct implementation of the broad-band impulse response [Sch94]. However, owing to the group delay caused by the filter bank, this method is not so suitable for real-time applications.

Chapter 7

Dynamic Range Control

Dynamic range control of audio signals is used in many applications to match the dynamic behavior of the audio signal to different requirements. While recording, dynamic range control protects the AD converter from overload or it is used in the signal path to optimally use the full amplitude range of a recording system. For suppressing low-level noise, so-called noise gates are used so that the audio signal is passed through only from a certain level onwards. While reproducing music and speech in a car, the dynamics have to match the special noise characteristic inside a car.

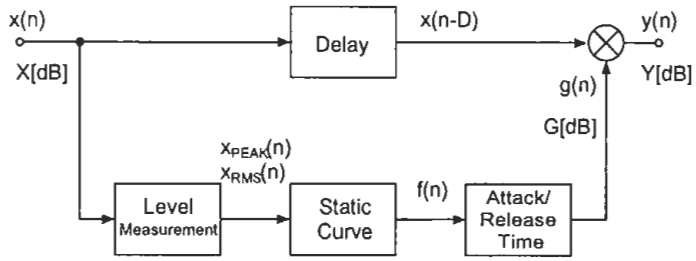


Figure 7.1 System for dynamic range control.

Figure 7.1 shows a block diagram of system for dynamic range control. After measuring the input level X[dB], the output level Y[dB] is affected by multiplying the delayed input signal $x(n)$ by a factor $g(n)$ according to

$$y(n) = g(n) \cdot x(n - D). \tag{7.1}$$

The delay of the signal $x(n)$ compared with the control signal $g(n)$ allows predictive control of the output signal level. This multiplicative weighting is carried out

with corresponding attack and release time. Multiplication leads, in terms of a logarithmic representation, to the addition of the weighting level G[dB] to the input level X[dB] giving the output level Y[dB].

7.1 Static Curve

The relationship between input level and weighting level is defined by a static level curve $G[\text{dB}] = f(X[\text{dB}])$. An example of such a static curve is given in Fig. 7.2. Here, the output level and the weighting level are given as functions of the input level.

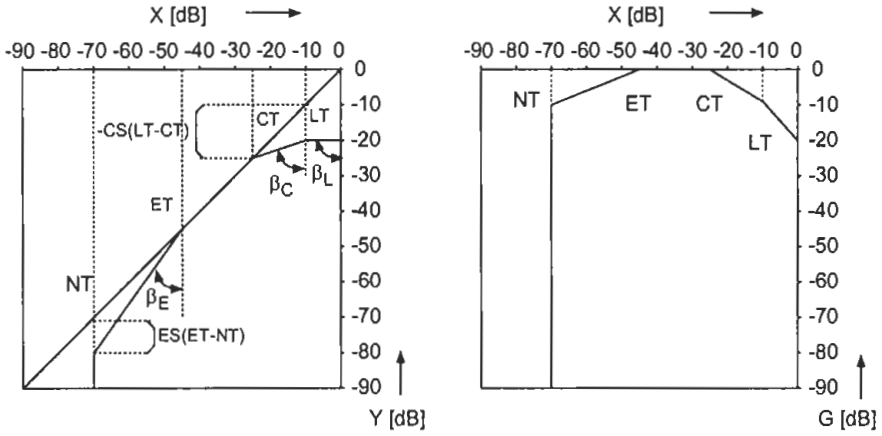


Figure 7.2 Static curve with the parameters LT=*Limiter threshold*, CT=*Compressor threshold*, ET=*Expander threshold* and NT=*Noise gate threshold*.

With the help of a limiter, the output level is limited when the input level exceeds the limiter threshold LT. All input levels above this threshold lead to a constant output level. The compressor maps a change of input level on a certain smaller change of output level. In contrast to a limiter, the compressor increases the loudness of the audio signal. The expander increases changes in the input level to larger changes in the output level. With this, an increase of the dynamics for low levels is achieved. The noise gate is used to suppress low-level signals, for noise reduction and is also used for sound effects like truncating the decay of room reverberation. Every threshold used in particular parts of the static curve is defined as the lower limit for limiter and compressor and upper limit for expander and noise gate.

In the logarithmic representation of the static curve the compression factor R (*Ratio*) is defined by the ratio of the input level change ΔP_I to the output level

change ΔP_O as given by

$$\mathrm{R} = \frac{\Delta P_I}{\Delta P_O}. \tag{7.2}$$

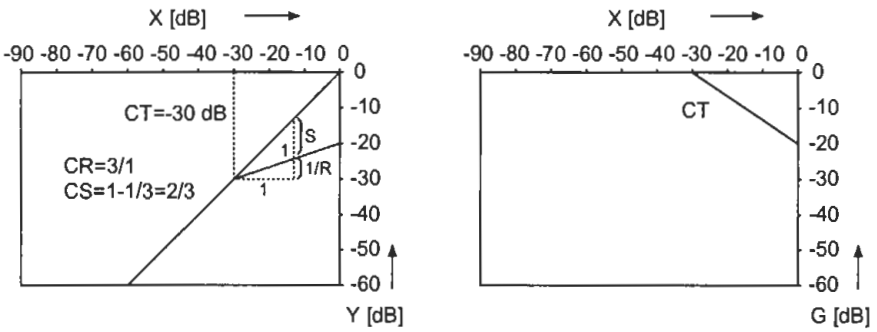


Figure 7.3 Compressor Curve (Compressor Ratio CR/Slope CS).

With the help of Fig. 7.3 the straight line equation $Y = \mathrm{CT} + \frac{1}{\mathrm{R}}(X - \mathrm{CT})$ and the compression factor

$$\mathrm{R} = \frac{X - \mathrm{CT}}{Y - \mathrm{CT}} = \tan\beta_C, \tag{7.3}$$

are obtained, where the angle β is defined as shown in Fig. 7.2. The relationship between the *ratio* R and the *slope* S can also be derived from Fig. 7.3 and is expressed as

$$\mathrm{S} = 1 - \frac{1}{\mathrm{R}} \tag{7.4}$$

$$\mathrm{R} = \frac{1}{1 - \mathrm{S}}. \tag{7.5}$$

Typical compression factors are

$$\begin{array}{rrrll} & \mathrm{R} & = & \infty & \text{limiter} \\ & \mathrm{R} & > & 1 & \text{compressor (CR: compressor ratio)} \\ 0 < & \mathrm{R} & < & 1 & \text{expander (ER: expander ratio)} \\ & \mathrm{R} & = & 0 & \text{noise gate.} \end{array} \tag{7.6}$$

The transition from logarithmic to linear representation leads, from (7.3), to

$$\mathrm{R} = \frac{\log_{10}\frac{\hat{x}}{c_T}}{\log_{10}\frac{\hat{y}}{c_T}}, \tag{7.7}$$

where $\hat{x}$ and $\hat{y}$ are the linear levels and c_T denotes the linear compressor threshold. Rewriting (7.7) gives the linear output level

$$\begin{aligned} \frac{\hat{y}}{c_T} &= 10^{\frac{1}{\mathrm{R}}\log_{10}(\frac{\hat{x}}{c_T})} = (\frac{\hat{x}}{c_T})^{\frac{1}{\mathrm{R}}} \\ \hat{y} &= c_T^{1-\frac{1}{\mathrm{R}}} \cdot \hat{x}^{\frac{1}{\mathrm{R}}} \end{aligned} \tag{7.8}$$

as a function of input level. The control factor $g(n)$ can be calculated by the quotient

$$\begin{aligned} g(n) &= \frac{\hat{y}}{\hat{x}} \\ &= \left(\frac{\hat{x}}{c_T}\right)^{\frac{1}{R}-1}. \end{aligned} \tag{7.9}$$

With the help of tables and interpolation methods, it is possible to determine the control factor without taking logarithms and antilogarithms. The implementation described as follows, however, makes use of the logarithm of the input level and calculates the control level G[dB] with the help of the straight line equation. The antilogarithm leads to the value $f(n)$ which gives the control factor $g(n)$ with corresponding attack and release time (see Fig. 7.1).

7.2 Dynamic Behavior

Besides the static curve of dynamic range control, the dynamic behavior in terms of attack and release times plays a significant role in sound quality. The rapidity of dynamic range control depends also on the measurement of PEAK and RMS values [McN84, Sti86].

7.2.1 Level Measurement

Level measurements [McN84] can be made with the systems shown in Figs. 7.4 and 7.5. For PEAK measurement, the absolute value of the input is compared with the peak value $x_{\text{PEAK}}(n)$. If the absolute value is greater than the peak value, the difference is weighted with the coefficient AT (attack time) and added to $(1 - \text{RT}) \cdot x_{\text{PEAK}}(n)$ (RT = release time). If the absolute value of the input is smaller than the peak value, the new peak value is equal to $(1 - \text{RT}) \cdot x_{\text{PEAK}}(n)$. The difference equation for the block diagram in Fig. 7.4 is given by

$$x_{\text{PEAK}}(n) = (1 - \text{AT} - \text{RT}) \cdot x_{\text{PEAK}}(n-1) + \text{AT} \cdot |x(n)| \tag{7.10}$$

with the transfer function

$$H(z) = \frac{\text{AT}}{1 - (1 - \text{AT} - \text{RT})z^{-1}}. \tag{7.11}$$

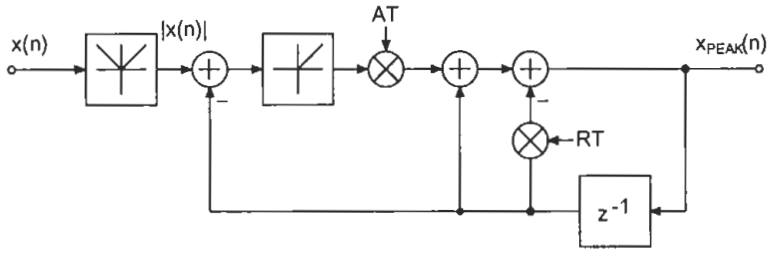


Figure 7.4 PEAK measurement.

The RMS measurement shown in Fig. 7.5, uses the square of the input and performs averaging with a first-order low-pass filter. The averaging coefficient TAV is determined in section 7.2.3. The difference equation is given by

$$x_{\mathrm{RMS}}(n) = (1 - \mathrm{TAV}) \cdot x_{\mathrm{RMS}}(n-1) + \mathrm{TAV} \cdot x^2(n) \tag{7.12}$$

with the transfer function

$$H(z) = \frac{\mathrm{TAV}}{1 - (1 - \mathrm{TAV})z^{-1}}. \tag{7.13}$$

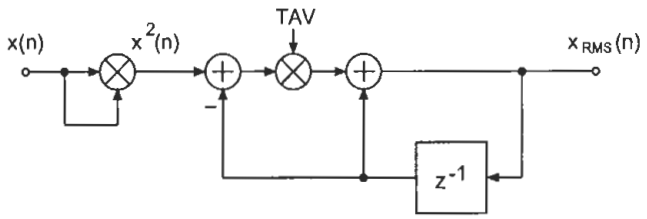


Figure 7.5 RMS measurement (TAV = averaging coefficient).

7.2.2 Gain Factor Smoothing

Attack and release times can be implemented by the system shown in Fig. 7.6 [McN84]. The attack coefficient AT or release coefficient RT is obtained by comparing the input control factor and the previous one. A small hysteresis curve determines whether the control factor is in the attack or release status and hence gives the coefficient AT or RT. The system also serves to smooth the control signal. The difference equation is given by

$$g(n) = (1 - k) \cdot g(n-1) + k \cdot f(n), \tag{7.14}$$

with $k = \mathrm{AT}$ or $k = \mathrm{RT}$ and the corresponding transfer function leads to

$$H(z) = \frac{k}{1 - (1 - k)z^{-1}}. \tag{7.15}$$

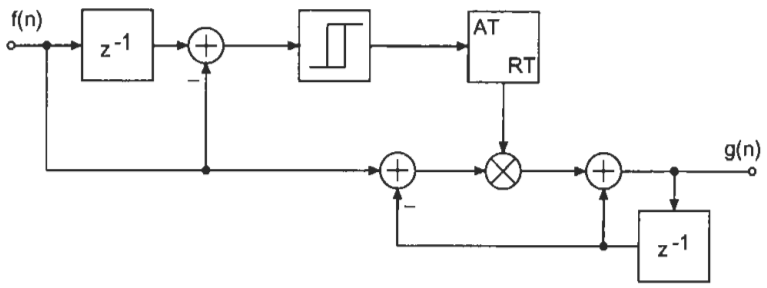


Figure 7.6 Implementing attack and release time or gain factor smoothing.

7.2.3 Time Constants

If the step response of a continuous-time system is

$$g(t) = 1 - e^{-t/\tau} \qquad \tau = \text{time constant}, \tag{7.16}$$

then sampling (step-invariant transform) the step response gives the discrete-time step response

$$g(nT_S) = \varepsilon(nT_S) - e^{-nT_S/\tau} = 1 - z_\infty^n \qquad \text{with} \quad z_\infty = e^{-T_S/\tau}. \tag{7.17}$$

The Z-transform leads to

$$\begin{aligned} G(z) &= \frac{z}{z-1} - \frac{1}{1 - z_\infty z^{-1}} \\ &= \frac{1 - z_\infty}{(z-1)(1 - z_\infty z^{-1})}. \end{aligned} \tag{7.18}$$

With the definition of attack time $t_a = t_{90} - t_{10}$, we derive

$$\begin{aligned} 0.1 &= 1 - e^{-t_{10}/\tau} \qquad \leftarrow t_{10} = 0.1\tau \\ 0.9 &= 1 - e^{-t_{90}/\tau} \qquad \leftarrow t_{90} = 0.9\tau. \end{aligned} \tag{7.19}$$

The relationship between attack time t_a and the time constant τ of the step response is obtained as follows:

$$\begin{aligned} 0.9/0.1 &= e^{(t_{90} - t_{10})/\tau} \\ \ln(0.9/0.1) &= (t_{90} - t_{10})/\tau \\ t_a &= t_{90} - t_{10} = 2.2\tau. \end{aligned} \tag{7.20}$$

Hence, the pole is calculated as

$$\boxed{z_\infty = e^{-2.2T_S/t_a}} \tag{7.21}$$

A system for implementing the given step response is obtained by the relationship between the Z-transform of the impulse response and the Z-transform of the step response:

$$H(z) = \frac{z-1}{z} G(z). \tag{7.22}$$

The transfer function can now be written as

$$H(z) = \frac{(1-z_\infty)z^{-1}}{(1-z_\infty z^{-1})}. \tag{7.23}$$

7.3 Implementation

The programming of a system for dynamic range control is described in the following sections.

7.3.1 Limiter

The block diagram of a limiter is presented in Fig. 7.7. The signal $x_{\mathrm{PEAK}}(n)$ is determined from the input with variable attack and release time. The logarithm to the base 2 of this peak signal is taken and compared with the limiter threshold. If the signal is above the threshold, the difference is multiplied by the negative slope of the limiter LS. After this, the antilogarithm of the result is taken. The obtained control factor $f(n)$ is then smoothed with a first-order low-pass filter (SMOOTH). If the signal $x_{\mathrm{PEAK}}(n)$ lies below the limiter threshold, the signal $f(n)$ is set to $f(n) = 1$. The delayed input $x(n - D_1)$ is multiplied by the smoothed control factor $g(n)$ to give the output $y(n)$.

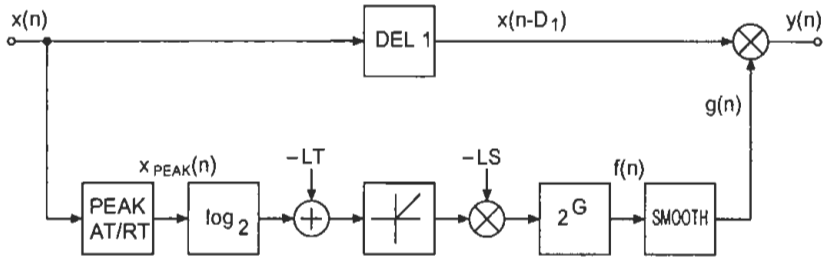


Figure 7.7 Limiter.

7.3.2 Compressor, Expander, Noise Gate

The block diagram of a compressor/expander/noise gate is shown in Fig. 7.8. The basic structure is similar to the limiter. In contrast to the limiter, the logarithm of

the signal $x_{\mathrm{RMS}}(n)$ is taken and multiplied by 0.5. The obtained value is compared with three thresholds in order to determine the operating range of the static curve. If one of the three thresholds is crossed, the resulting difference is multiplied by the corresponding slope (CS, ES, NS) and the antilogarithm of the result is taken. A following first-order low-pass filter provides the attack and release time.

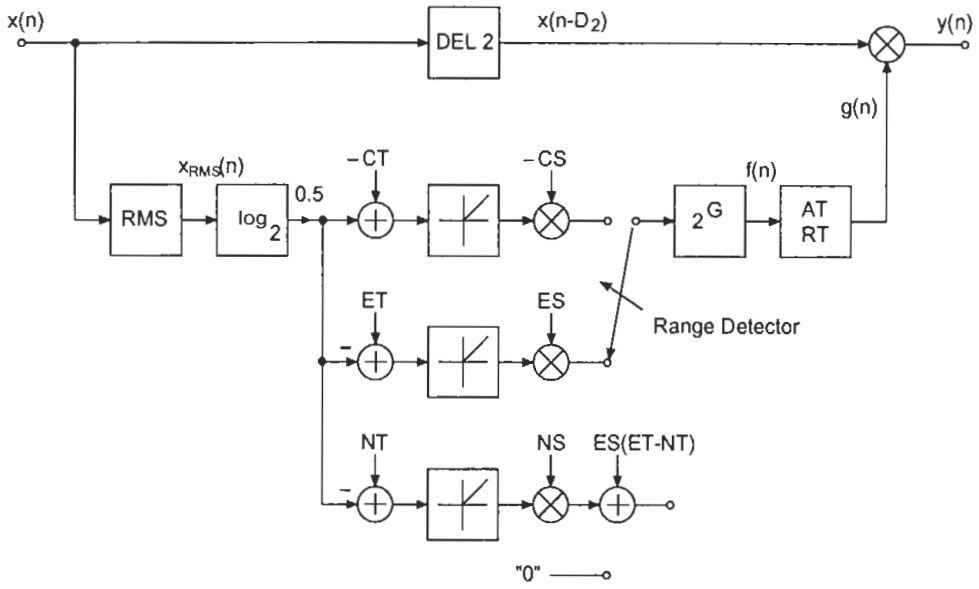


Figure 7.8 Compressor/expander/noise gate.

7.3.3 Combination System

A combination of a limiter that uses PEAK measurement, and a compressor/expander/noise gate that is based on RMS measurement, is presented in Fig. 7.9. The PEAK and RMS values are measured simultaneously. If the linear threshold of the limiter is crossed, the logarithm of the peak signal $x_{\mathrm{PEAK}}(n)$ is taken and the upper path of the limiter is used for calculating the characteristic curve. If the limiter threshold is not crossed, the logarithm of the RMS value is taken and one of the three lower paths is used. The additive terms in the limiter and noise gate paths result from the static curve. After going through the range detector, the antilogarithm is taken. The sequence $f(n)$ is smoothed with a SMOOTH filter in the limiter case, or weighted with corresponding attack and release times of the relevant operating range (compressor, expander or noise gate). By limiting the maximum level, the dynamic range is reduced. As a consequence, the overall static curve can be shifted up by a gain factor. Figure 7.10 demonstrates this with a gain factor equal to 10 dB. This static parameter value is directly included in the control factor $g(n)$.

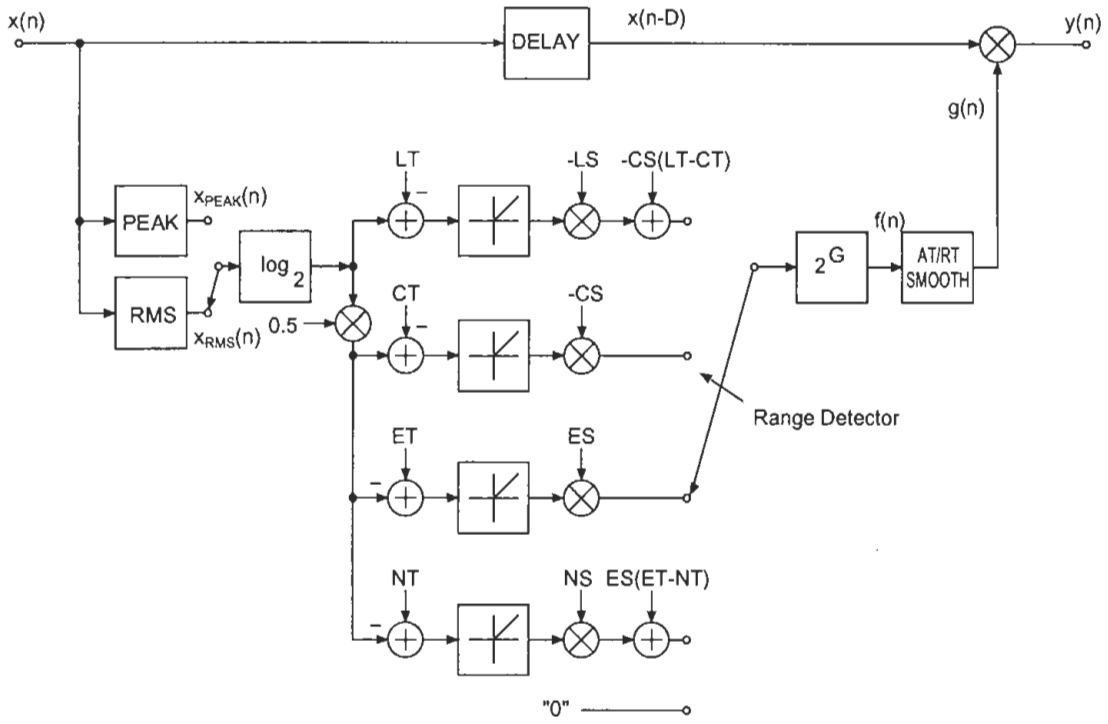


Figure 7.9 Limiter/compressor/expander/noise gate.

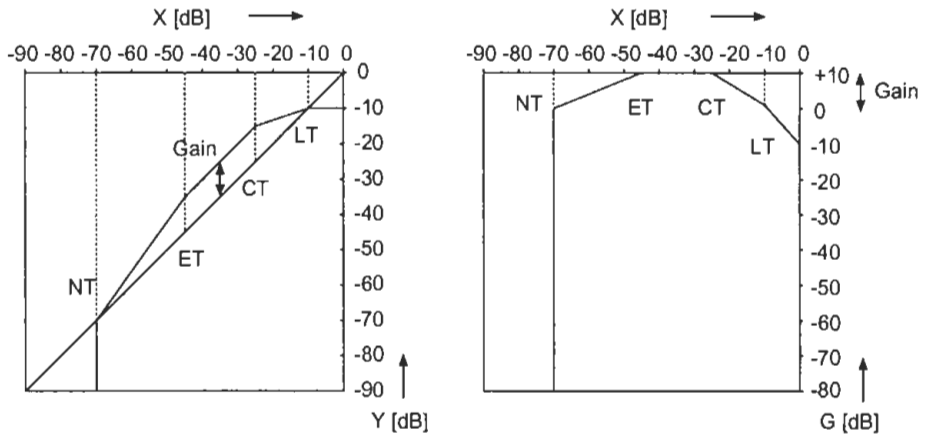


Figure 7.10 Shifting the static curve by a gain factor.

As an example, Fig. 7.11 illustrates the input $x(n)$, the output $y(n)$ and the control factor $g(n)$ of a compressor/expander system. It is observed that signals with high amplitude are compressed and the ones with low amplitude are expanded. An additional gain of 12 dB shows the maximum value of 4 for the control factor $g(n)$. The compressor/expander system operates in the linear region of the static curve if the control factor is equal to 4. If the control factor is between 1

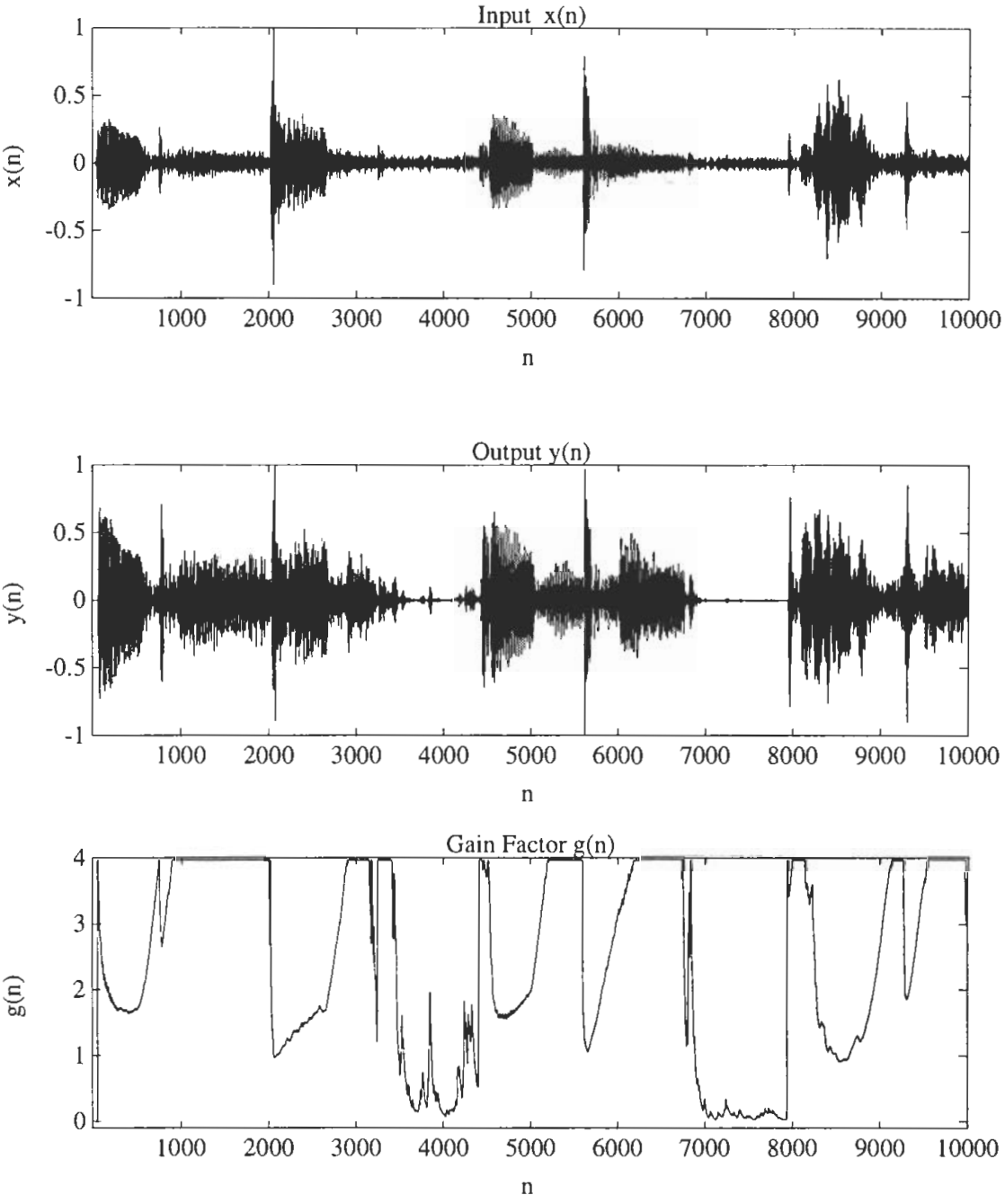


Figure 7.11 Signals $x(n)$, $y(n)$ and $g(n)$ for dynamic range control.

and 4, the system operates as a compressor. For control factors lower than 1, the system works as an expander ($3500 < n < 4500$ and $6800 < n < 7900$). The compressor is responsible for increasing the loudness of the signal whereas the expander increases the dynamic range for signals of small amplitude.

7.4 Realization Aspects

7.4.1 Sampling Rate Reduction

In order to reduce the computational complexity, downsampling can be carried out after calculating the PEAK/RMS value (see Fig. 7.12). As the signals $x_{\mathrm{PEAK}}(n)$ and $x_{\mathrm{RMS}}(n)$ are already band-limited, they can be directly downsampled by taking every second or fourth value of the sequence. This downsampled signal is then processed by taking its logarithm, calculating the static curve, taking the anti-logarithm and filtering with corresponding attack and release time with reduced sampling rate. The following upsampling by a factor of 4 is achieved by repeating the output value four times. This procedure is equivalent to upsampling by a factor 4 followed by a sample-and-hold transfer function.

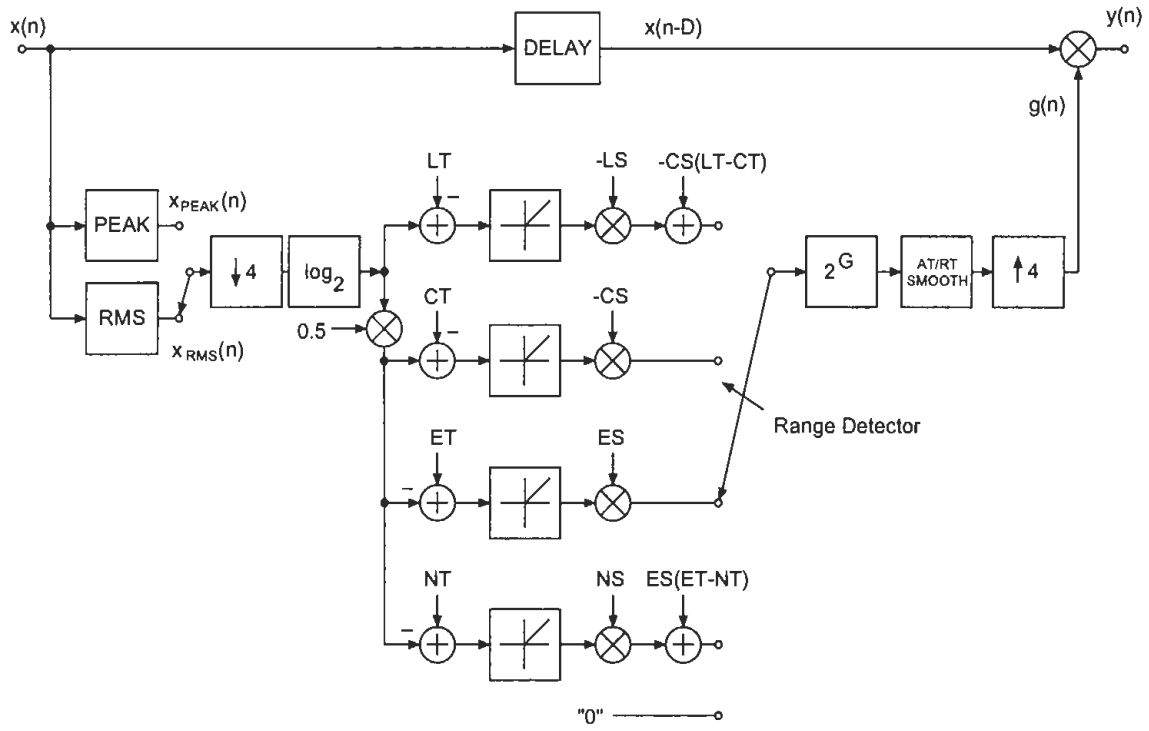


Figure 7.12 Dynamic system with sampling rate reduction.

The nesting and spreading of partial program modules over four sampling periods is shown in Fig. 7.13. The modules PEAK/RMS (i.e. PEAK/RMS calculation) and MULT (delay of input and multiplication with $g(n)$) are performed every input sampling period. The number of processor cycles for PEAK/RMS and MULT are denoted by Z1 and Z3 respectively. The modules LD(x), CURVE, 2^x and SMO have a maximum number of processor cycles of Z2 and are processed

consecutively in the given order. This procedure is repeated every four sampling periods. The total number of processor cycles per sampling period for the complete dynamics algorithm results from the sum of all three modules.

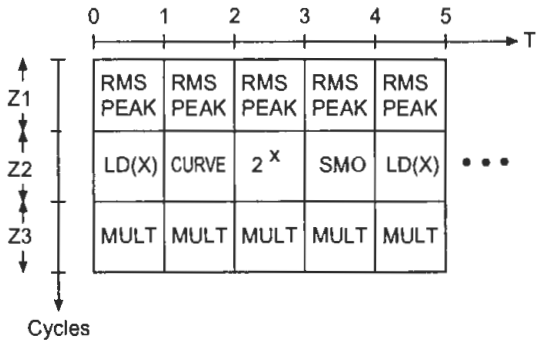


Figure 7.13 Nesting technique.

7.4.2 Curve Approximation

Besides taking logarithm and antilogarithm, other simple operations like comparisons and addition/multiplication occur in calculating the static curve. The logarithm of the PEAK/RMS value is taken as follows:

$$x = M \cdot 2^E \tag{7.24}$$

$$\mathrm{ld}(x) = \mathrm{ld}(M) + E. \tag{7.25}$$

First, the mantissa is normalized and the exponent is determined. The function $\mathrm{ld}(M)$ is then calculated by a series expansion. The exponent is simply added to the result.

The logarithmic weighting factor G and the antilogarithm 2^G are given by

$$G = -E - M \tag{7.26}$$

$$2^G = 2^{-E} \cdot 2^{-M}. \tag{7.27}$$

$$\tag{7.28}$$

Here, E is a natural number and M is a fractional number. The antilogarithm 2^G is calculated by expanding the function 2^{-M} in a series and multiplication by 2^{-E}. A reduction of computational complexity can be achieved by directly using tables for log and antilog.

7.4.3 Stereo Processing

For stereo processing, a common control factor $g(n)$ is needed. If different control factors are used for both channels, limiting or compressing one of the two stereo signals causes a displacement of the stereo balance. Figure 7.14 shows a stereo dynamic system in which the sum of the two signals is used for calculating a common control factor $g(n)$. The following processing steps of measuring the PEAK/RMS value, downsampling, taking logarithm, calculating static curve, taking antilogarithm attack and release time and upsampling with a sample-and-hold function remain the same. The delay (DEL) in the direct path must be the same for both channels.

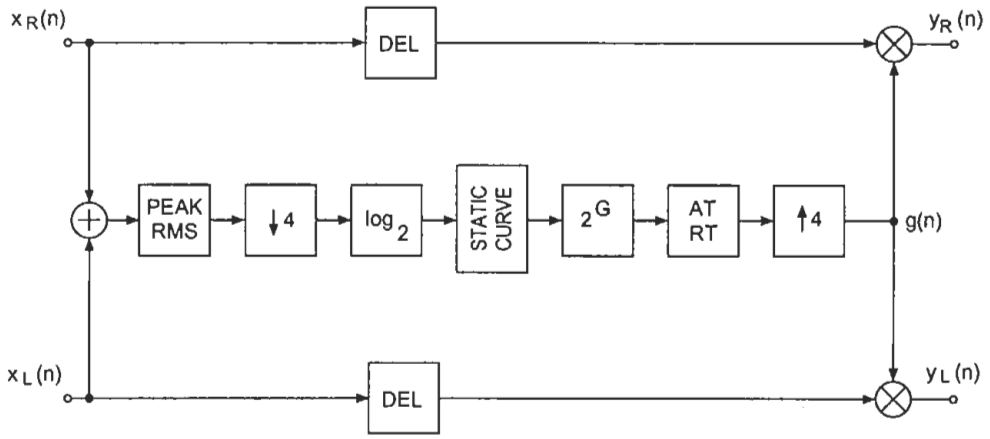


Figure 7.14 Stereo dynamic system.

Chapter 8

Sampling Rate Conversion

Several different sampling rates are established for digital audio applications. For broadcasting, professional and consumer audio, sampling rates of 32, 48 and 44.1 kHz are used. Moreover, other sampling rates are derived from different frame rates for film and video. In connecting systems with different uncoupled sampling rates, there is a need for sampling rate conversion. In this chapter, synchronous sampling rate conversion with rational factor L/M for coupled clock rates and asynchronous sampling rate conversion will be discussed where the different sampling rates are not synchronized with each other.

8.1 Synchronous Conversion

Sampling rate conversion for coupled sampling rates by a rational factor L/M can be performed by the system shown in Fig. 8.1. After upsampling by a factor L, anti-image filtering at Lf_S is done followed by downsampling by factor M. Since after upsampling and filtering only every Mth sample is used, it is possible to develop efficient algorithms that reduce complexity. In this respect two methods are in use; one is based on a time-domain interpretation [Cro83] and the other one [Hsi87] uses Z-domain fundamentals. Owing to its computational efficiency, only the method in the Z-domain will be considered.

Starting with the finite impulse response $h(n)$ of length N and its Z-transform

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}, \tag{8.1}$$

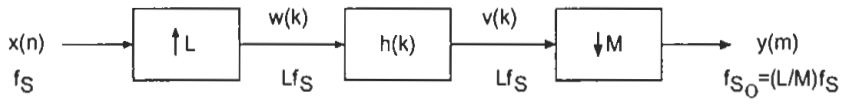


Figure 8.1 Sampling rate conversion by factor L/M.

the polyphase representation [Cro83, Vai93, Fli94] with M components can be expressed as

$$H(z) = \sum_{k=0}^{M-1} z^{-k} E_k(z^M) \qquad \text{type 1} \tag{8.2}$$

$$\text{with} \quad e_k(n) = h(nM+k) \quad , \quad k = 0, 1, ..., M-1 \tag{8.3}$$

or

$$H(z) = \sum_{k=0}^{M-1} z^{-(M-1-k)} R_k(z^M) \qquad \text{type 2} \tag{8.4}$$

$$\text{with} \quad r_k(n) = h(nM-k) \quad , \quad k = 0, 1, ..., M-1. \tag{8.5}$$

The polyphase decomposition as given in (8.2) and (8.4) is denoted as type 1 and 2 respectively. The type 1 polyphase decomposition corresponds to a commutator model in the anti clockwise direction whereas the type 2 is in the clockwise direction. The relationship between $R(z)$ and $E(z)$ is described by

$$R_k(z) = E_{M-1-k}(z). \tag{8.6}$$

With the help of the identities [Vai93] shown in Fig. 8.2 and the decomposition (Euclid's theorem)

$$z^{-1} = z^{-pL} z^{qM}, \tag{8.7}$$

it is possible to move the inner delay elements of Fig. 8.3. Equation (8.7) is valid if M and L are prime numbers. In a cascade of upsampling and downsampling, the order of functional blocks can be exchanged (see Fig. 8.3b).

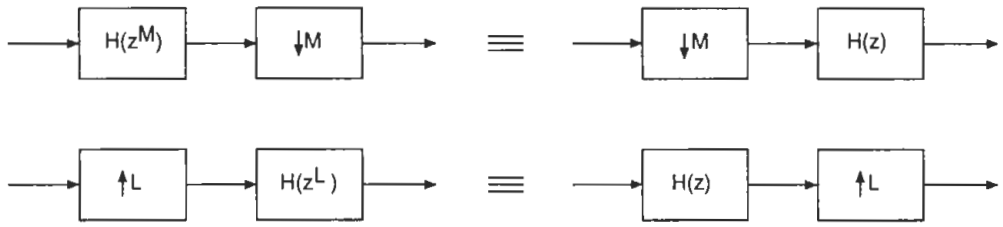


Figure 8.2 Identities for sampling rate conversion.

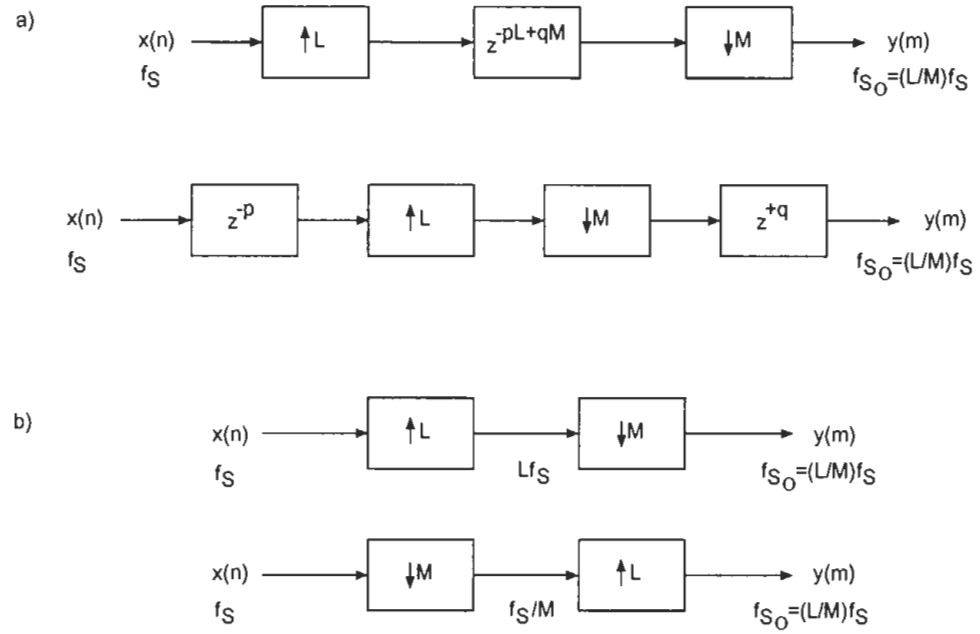


Figure 8.3 Decomposition in accordance with Euclid's theorem.

The use of polyphase decomposition can be demonstrated with the help of an example for $L = 2$ and $M = 3$. This implies a sampling rate conversion from 48 kHz to 32 kHz. Figures 8.4 and 8.5 show two different solutions for polyphase decomposition of sampling rate conversion by 2/3.

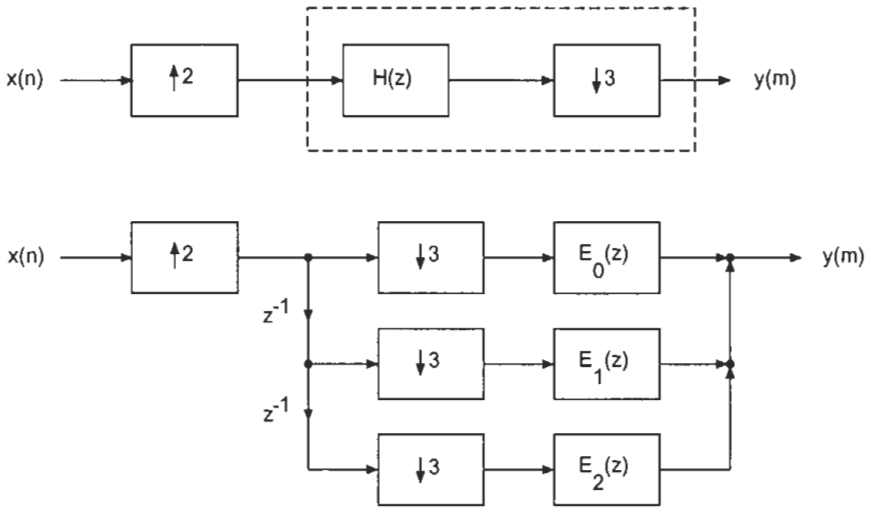


Figure 8.4 Polyphase decomposition for downsampling $L/M = 2/3$.

Further decompositions of the upsampling decomposition of Fig. 8.5 are demonstrated in Fig. 8.6. First, interpolation is implemented with a polyphase decompo-

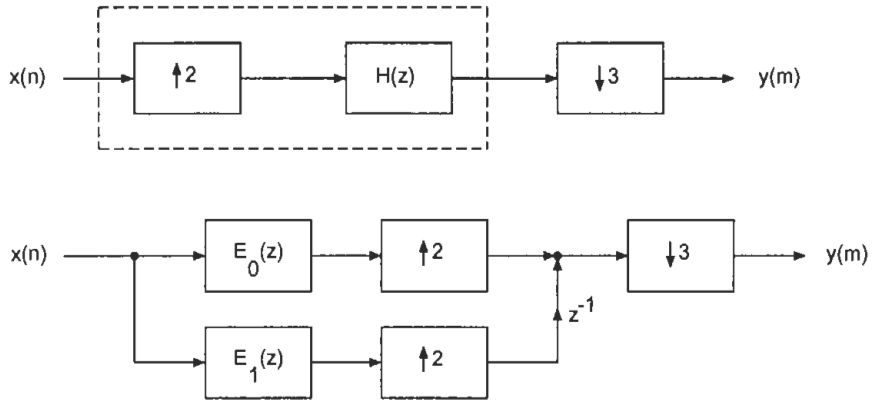


Figure 8.5 Polyphase decomposition for upsampling $L/M = 2/3$.

sition and the delay z^{-1} is decomposed to $z^{-1} = z^{-2}z^3$. Then, the downsampler of factor 3 is moved through the adder into the two paths (Fig. 8.6b) and the delays are moved according to the identities of Fig. 8.2. In Fig. 8.6c, the upsampler is exchanged with the downsampler and in a last step (Fig. 8.6d) another polyphase decomposition of $E_0(z)$ and $E_1(z)$ is carried out. The actual filter operations $E_{0k}(z)$ and $E_{1k}(z)$ with $k = 0, 1, 2$ are performed at $\frac{1}{3}$ of the input sampling rate.

8.2 Asynchronous Conversion

Plesiochronous systems consist of partial systems with different and uncoupled sampling rates. Sampling rate conversion between such systems can be achieved through a DA conversion with the sampling rate of the first systems followed by an AD conversion with sampling rate of the second system. A digital approximation of this approach is made with a multirate system [Lag81, Lag82a,b,c, Lag83, Ram82, Ram84]. Figure 8.7a shows a system for increasing the sampling rate by a factor L followed by an anti-image filter $H(z)$ and a resampling of the interpolated signal $y(k)$. The samples $y(k)$ are held for a clock period (see Fig. 8.7c) and then sampled with output clock period $T_{S_O} = 1/f_{S_O}$. The interpolation sampling rate must be increased so far that the difference of two consecutive samples $y(k)$ is smaller than the quantization step Q. The sample-and-hold function applied to $y(k)$ suppresses the spectral images at multiples of Lf_S (see Fig. 8.7b). The now obtained signal is a bandlimited continuous-time signal which can sampled with output sampling rate f_{S_O}.

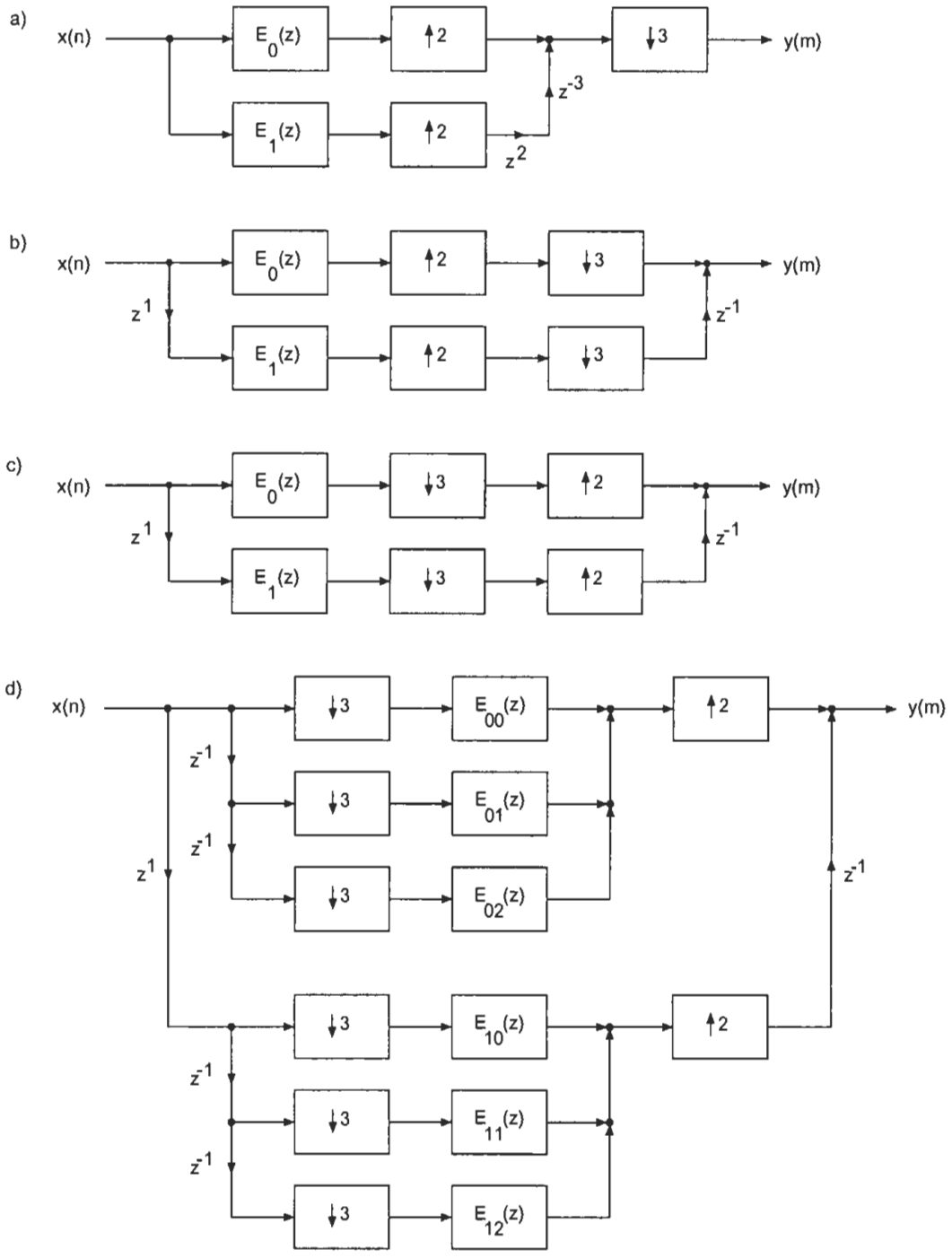


Figure 8.6 Sampling rate conversion by factor 2/3.

For the calculation of the necessary oversampling rate, the problem is considered in the frequency-domain. The sinc-function of a sample-and-hold system (see

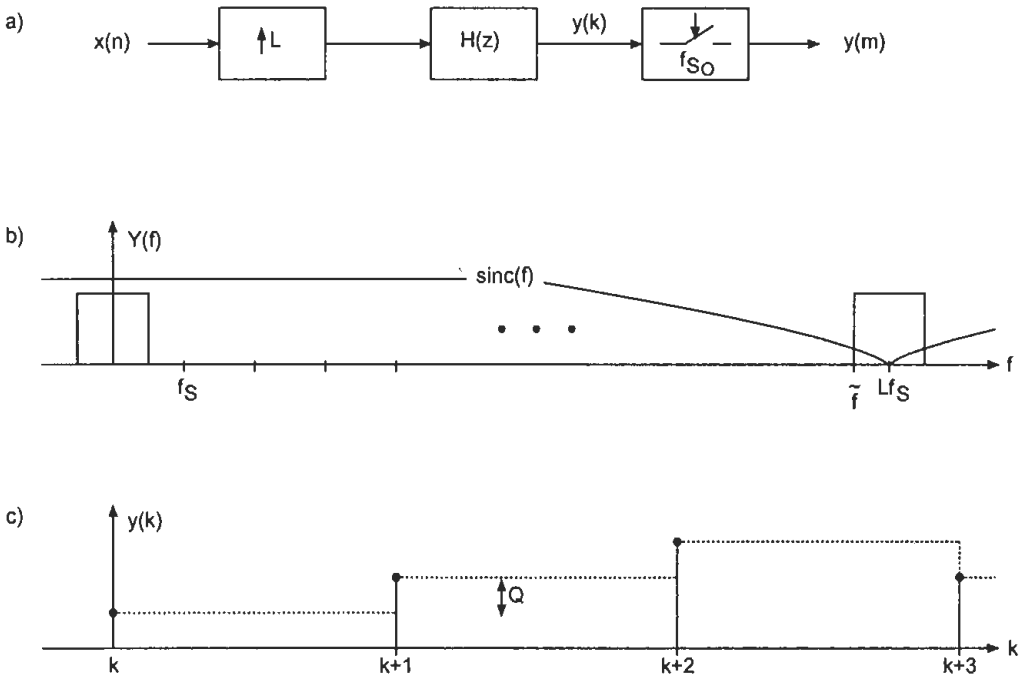


Figure 8.7 Approximation of DA/AD conversions.

Fig. 8.7b) at frequency $\tilde{f} = (L - \frac{1}{2})f_S$ is given by

$$E(\tilde{f}) = \frac{\sin\left(\frac{\pi\tilde{f}}{Lf_S}\right)}{\frac{\pi\tilde{f}}{Lf_S}} \tag{8.8}$$

$$= \frac{\sin\left(\frac{\pi(L-\frac{1}{2})f_S}{Lf_S}\right)}{\frac{\pi(L-\frac{1}{2})f_S}{Lf_S}} \tag{8.9}$$

$$= \frac{\sin\left(\pi - \frac{\pi}{2L}\right)}{\pi - \frac{\pi}{2L}}. \tag{8.10}$$

With $\sin(\alpha - \beta) = \sin(\alpha)\cos(\beta) - \cos(\alpha)\sin(\beta)$ we derive

$$E(\tilde{f}) = \frac{\sin\left(\frac{\pi}{2L}\right)}{\pi\left(1 - \frac{1}{2L}\right)} \tag{8.11}$$

$$\approx \frac{\pi/2L}{\pi\left(1 - \frac{1}{2L}\right)} \tag{8.12}$$

$$\approx \frac{1}{2L-1} \approx \frac{1}{2L}. \tag{8.13}$$

For a given word-length w and quantization step Q, the necessary interpolation

rate L is calculated by:

$$\frac{Q}{2} \geq \frac{1}{2L} \tag{8.14}$$

$$\frac{2^{-(w-1)}}{2} \geq \frac{1}{2L} \tag{8.15}$$

$$L \geq 2^{w-1}. \tag{8.16}$$

For a linear interpolation between upsampled samples $y(k)$, we can derive

$$E(\tilde{f}) = \frac{\sin^2\left(\frac{\pi\tilde{f}}{Lf_S}\right)}{\left(\frac{\pi\tilde{f}}{Lf_S}\right)^2} \tag{8.17}$$

$$= \frac{\sin^2\left(\frac{\pi(L-\frac{1}{2})f_S}{Lf_S}\right)}{\left(\frac{\pi(L-\frac{1}{2})f_S}{Lf_S}\right)^2} \tag{8.18}$$

$$\approx \frac{1}{(2L)^2}. \tag{8.19}$$

With this it is possible to reduce the necessary interpolation rate to

$$L_1 \geq 2^{\frac{w}{2}-1}. \tag{8.20}$$

Figure (8.8) demonstrates this with a two-stage block diagram. First, interpolation up to a sampling rate $L_1 f_S$ is performed by conventional filtering. In a second stage upsampling by factor L_2 is done by linear interpolation. The two-stage approach must satisfy the sampling rate $Lf_S = (L_1 L_2) f_S$.

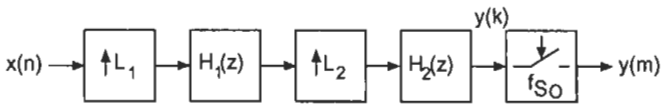


Figure 8.8 Linear interpolation before virtual sample-and-hold function.

The choice of the interpolation algorithm in the second stage enables the reduction of the first oversampling factor. More details are discussed in section 8.2.2.

8.2.1 Single-stage Methods

Direct conversion methods implement the block diagram [Lag83, Smi84, Par90, Par91a,b, Ada92, Ada93] shown in Fig. 8.7a. The calculation of a discrete sample

on an output grid of sampling rate f_{S_O} from samples $x(n)$ at sampling rate f_{S_I}, can be written as

$$\begin{aligned} \mathrm{DFT}[x(n-\alpha)] &= X(e^{j\Omega})e^{-j\alpha\Omega} \\ &= X(e^{j\Omega})H_\alpha(e^{j\Omega}), \end{aligned} \tag{8.21}$$

where $0 \leq \alpha < 1$. With the transfer function

$$H_\alpha(e^{j\Omega}) = e^{-j\alpha\Omega} \tag{8.22}$$

and the properties

$$H(e^{j\Omega}) = \begin{cases} 1 & 0 \leq |\Omega| \leq \Omega_c \\ 0 & \Omega_c < |\Omega| < \pi \end{cases} \tag{8.23}$$

the impulse response is given by

$$h_\alpha = h(n-\alpha) = \frac{\Omega_c}{\pi} \frac{\sin[\Omega_c(n-\alpha)]}{\Omega_c(n-\alpha)}. \tag{8.24}$$

From (8.21) the convolution sum

$$x(n-\alpha) = \sum_{m=-\infty}^{\infty} x(m)h(n-\alpha-m) \tag{8.25}$$

$$= \sum_{m=-\infty}^{\infty} x(m) \frac{\Omega_c}{\pi} \frac{\sin[\Omega_c(n-\alpha-m)]}{\Omega_c(n-\alpha-m)} \tag{8.26}$$

is obtained. Figure 8.9 illustrates this convolution in the time-domain for a fixed α.

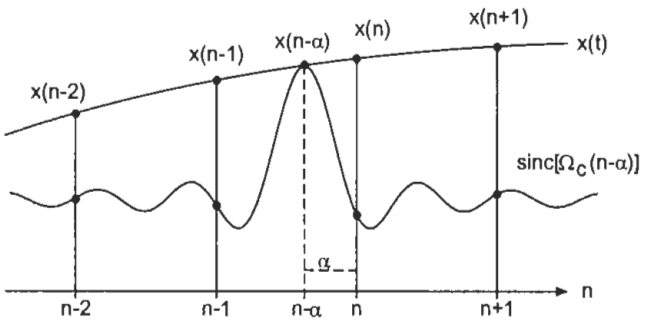


Figure 8.9 Convolution sum in the time-domain.

Fig. 8.10 shows the coefficients $h(n-\alpha_i)$ for discrete α_i $(i = 0, \ldots, 3)$ which are obtained from the intersection of the sinc-function with the discrete samples $x(n)$.

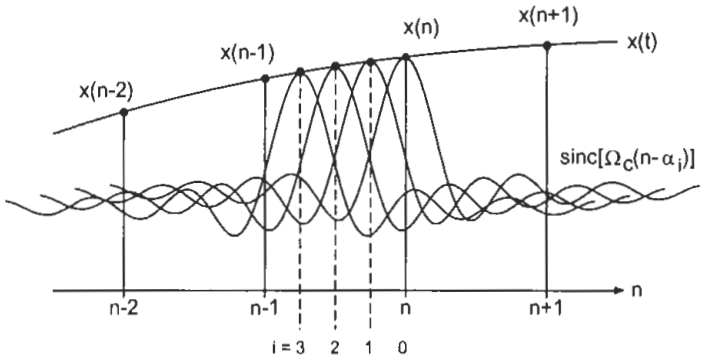


Figure 8.10 Convolution sum for different α_i.

In order to limit the convolution sum, the impulse response is windowed, which gives

$$h_W(n-\alpha_i) = w(n)\frac{\Omega_c}{\pi}\frac{\sin[\Omega_c(n-\alpha_i)]}{\Omega_c(n-\alpha_i)} \qquad n = 0,\ldots,2M. \tag{8.27}$$

From this, the sample estimate

$$\hat{x}(n-\alpha_i) = \sum_{m=-M}^{M} x(m)h_W(n-\alpha_i-m) \tag{8.28}$$

results. A graphical interpretation of the time-variant impulse response which depends on α_i is shown in Fig. 8.11. The discrete segmentation between two input samples into N intervals, leads to N partial impulse responses of length $2M+1$.

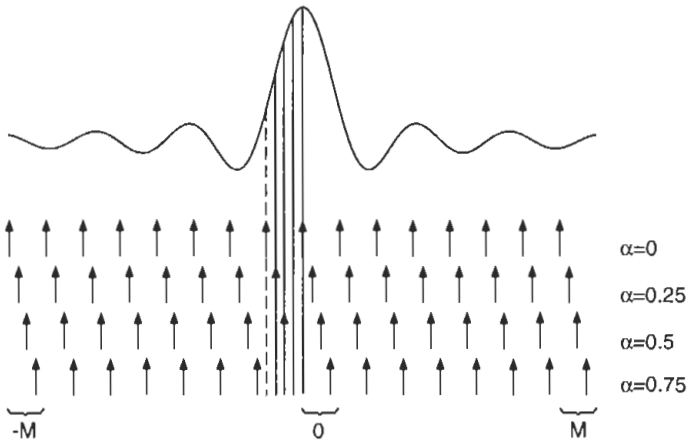


Figure 8.11 Sinc-function and different impulse responses.

If the output sampling rate is smaller than the input sampling rate ($f_{S_O} < f_{S_I}$), band-limiting (anti-aliasing) to the output sampling rate has to be done. This can

be achieved with factor $\beta = \frac{f_{S_O}}{f_{S_I}}$ and leads, with the the scaling theorem of the Fourier transform, to

$$h(n-\alpha) = \frac{\beta\Omega_c}{\pi} \frac{\sin[\beta\Omega_c(n-\alpha)]}{\beta\Omega_c(n-\alpha)}. \tag{8.29}$$

This time-scaling of the impulse response has the consequence that the number of coefficients of the time-variant partial impulse responses is increased. The number of required states also increases. Figure 8.12 shows the time-scaled impulse response and elucidates the increase of the number M of the coefficients.

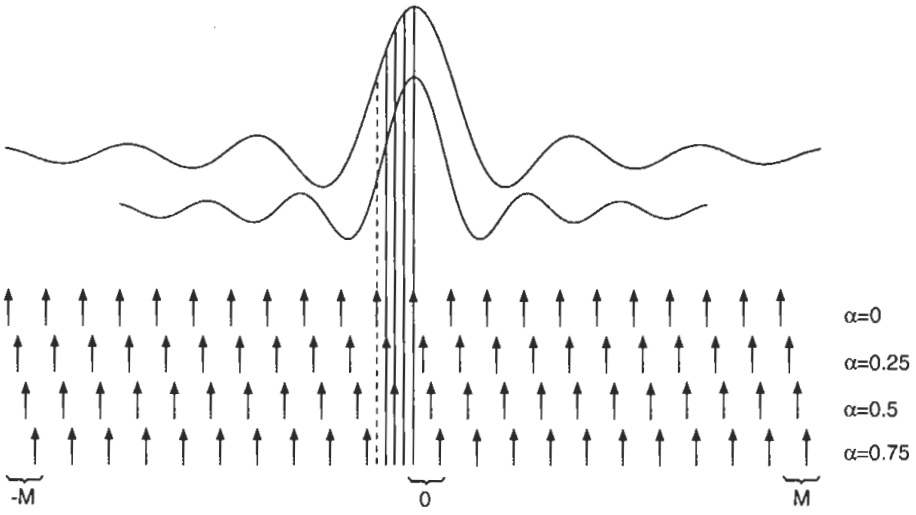


Figure 8.12 Time-scaled impulse response.

8.2.2 Multistage Methods

The basis of a multistage conversion method [Lag81, Lag82, Kat85, Kat86] is shown in Fig. 8.13a and will be described in the frequency-domain as shown in Fig. 8.13b-d.

The increase of the sampling rate up to the rate Lf_S before the sample-and-hold function is done in four stages. In the first two stages, the sampling rate is increased by a factor 2 followed by an anti-imaging filter (see Fig. 8.13b,c), which leads to a 4 times oversampled spectrum (Fig. 8.13d). In the third stage, the signal is upsampled by a factor 32 and the image spectra are suppressed (see Fig. 8.13d,e). In the fourth stage (Fig. 8.13e) the signal is upsampled to a sampling rate of Lf_S by factor 256 and a linear interpolator. The sinc^2-function of the linear interpolator suppresses the images at multiples of $128f_S$ up to the spectrum at Lf_S.

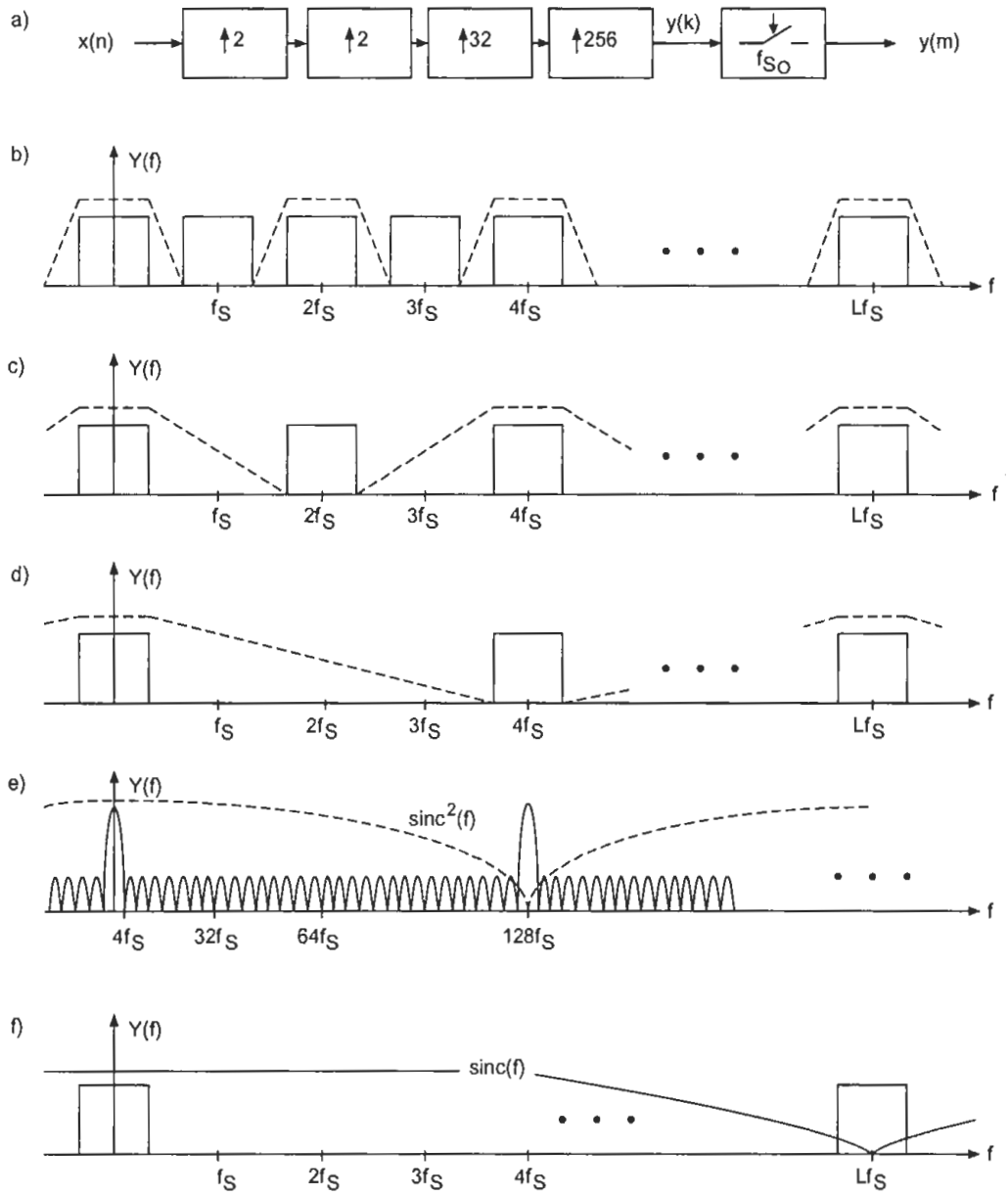


Figure 8.13 Multistage conversion - frequency-domain interpretation.

The virtual sample-and-hold function is shown in Fig. 8.13f, where resampling at the output sampling rate is performed. A direct conversion of this kind of cascaded interpolation structure requires anti-image filtering after every upsampling with the corresponding sampling rate. Although the necessary filter order decreases owing to a decrease of requirements for filter design, an implementation of the filters in the third and fourth stages is not possible directly. After a suggestion by Lagadec [Lag82c] the measurement of the ratio of input to output rate is used to

control the polyphase filters in the third and fourth stage (see Fig. 8.14a, CON = control) to reduce complexity. Figures 8.14b..d illustrate an interpretation in the time-domain. Figure 8.14b shows the interpolation of three samples between two

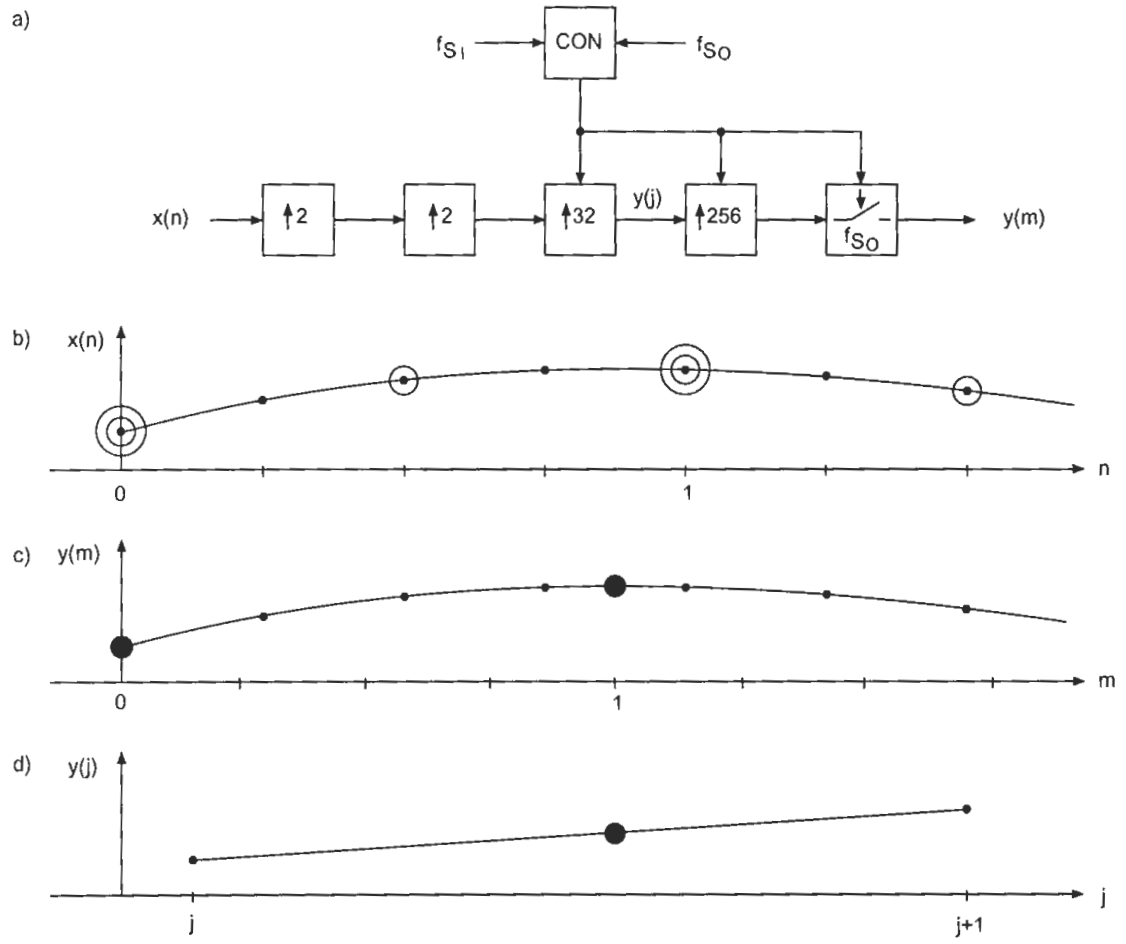


Figure 8.14 Time-domain interpretation.

input samples $x(n)$ with the help of the first and second interpolation stage. The abscissa represents the intervals of the input sampling rate and the sampling rate is increased by factor 4. In Fig. 8.14c the 4 times oversampled signal is shown. The abscissa shows the 4 times oversampled output grid. It is assumed that output sample $y(m = 0)$ and input sample $x(n = 0)$ are identical. The output sample $y(m = 1)$ is now determined in such a form that with the interpolator in the third stage only two polyphase filters just before and after the output sample need to be calculated. Hence, only 2 out of a total of 31 possible polyphase filters are calculated in the third stage. Fig. 8.14d shows these two polyphase output samples. Between these two samples, the output sample $y(m = 1)$ is obtained with a linear interpolation on a grid of 255 values.

Instead of the third and fourth stages, special interpolation methods can be used to calculate the output $y(m)$ directly from the 4 times oversampled input signal (see Fig. 8.15) [Sti91 , Cuc91, Liu92]. Section 8.3 is devoted to different

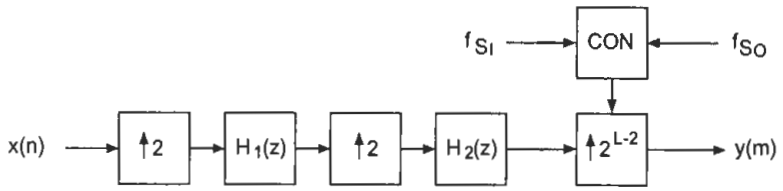


Figure 8.15 Sampling rate conversion with interpolation for calculating coefficients of a time-variant interpolation filter.

interpolation methods which allow a real-time calculation of filter coefficients. This can be interpreted as time-variant filters in which the filter coefficients are derived from the ratio of sampling rates. The calculation of one filter coefficient set for the output sample at the output rate is done by measuring the ratio of input to output sampling rate as described in the next section.

8.2.3 Control of Interpolation Filters

The measurement of the ratio of input and output sampling rate is used for controlling the interpolation filters [Lag82a]. By increasing the sampling rate by a factor of L the input sampling period is divided into $L = 2^{w-1} = 2^{15}$ parts for a signal word-length of $w = 16$ bits. The time instant of the output sample is calculated on this grid with the help of the measured ratio of sampling periods T_{S_O}/T_{S_I} as follows.

A counter is clocked with Lf_{S_I} and reset by every new input sampling clock. A sawtooth curve of the counter output versus time is obtained as shown in Fig. 8.16. The counter runs from 0 to $L-1$ during one input sampling period. At time t_{i-2} which corresponds to counter output z_{i-2}, the output sampling period T_{S_O} starts, and stops at time t_{i-1} with counter output z_{i-1}. The difference between both counter measurements allows the calculation of the output sampling period T_{S_O} with a resolution of Lf_{S_I}.

The new counter measurement is added to the difference of previous counter measurements. As a result, the new counter measurement is obtained as

$$t_i = (t_{i-1} + T_{S_O}) \oplus T_{S_I}. \tag{8.30}$$

The modulo operation can be carried out with an accumulator of word-length $w - 1 = 15$. The resulting time t_i determines the time instant of the output

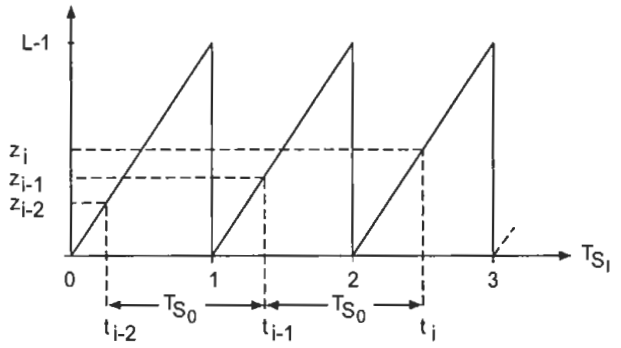


Figure 8.16 Calculation of t_i.

sample at the output sampling rate and therefore the choice of the polyphase filter in a single-stage conversion or the time instant for a multistage conversion.

The measurement of T_{S_O}/T_{S_I} is illustrated in Fig. 8.17:

- The input sampling rate f_{S_I} is increased to $M_Z f_{S_I}$ using a frequency multiplier where $M_Z = 2^w$. This increased input clock by the factor M_Z triggers a w bit counter. The counter output z is evaluated every M_O output sampling periods.
- Counting of M_O output sampling periods.
- Simultaneous counting of the M_I input sampling periods.

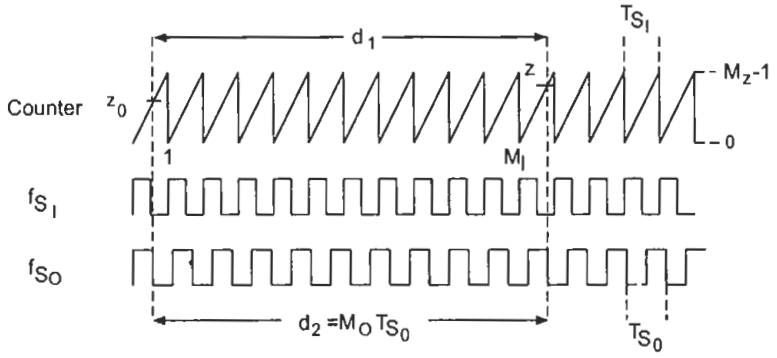


Figure 8.17 Measurement of T_{S_O}/T_{S_I}.

The time intervals d_1 and d_2 (see Fig. 8.17) are given by

$$d_1 = M_I T_{S_I} + \frac{z - z_0}{M_Z} T_{S_I} = (M_I + \frac{z - z_0}{M_Z}) T_{S_I} \tag{8.31}$$

$$d_2 = M_O T_{S_O}, \tag{8.32}$$

and with the requirement $d_1 = d_2$ we can write

$$\begin{aligned} M_O T_{S_O} &= (M_I + \frac{z - z_0}{M_Z}) T_{S_I} \\ \frac{T_{S_O}}{T_{S_I}} &= \frac{M_I + (z - z_0)/M_Z}{M_O} = \frac{M_Z M_I + (z - z_0)}{M_Z M_O}. \end{aligned} \tag{8.33}$$

- Example 1: $w = 0 \rightarrow M_Z = 1$

$$\frac{T_{S_O}}{T_{S_I}} = \frac{M_I}{2^{15}} \tag{8.34}$$

With a precision of 15 bits, the averaging number is chosen as $M_O = 2^{15}$ and the number M_I has to be determined.

- Example 2: $w = 8 \rightarrow M_Z = 2^8$

$$\frac{T_{S_O}}{T_{S_I}} = \frac{2^8 M_I + (z - z_0)}{2^8 2^7} \tag{8.35}$$

With a precision of 15 bits, the averaging number is chosen as $M_O = 2^7$ and the number M_I, as well as the counter outputs, has to be determined.

The sampling rates at the input and output of a sampling rate converter can be calculated by evaluating the 8 bit increment of the counter for each output clock with

$$z = \frac{T_{S_O}}{T_{S_I}} M_Z = \frac{f_{S_I}}{f_{S_O}} 256 \tag{8.36}$$

as seen from Table 8.1.

Table 8.1 Counter increments for different sampling rate conversions.

Conversion/kHz			8 bit counter increment.
32	→	48	170
44.1	→	48	235
32	→	44.1	185
48	→	44.1	278
48	→	32	384
44.1	→	32	352

8.3 Interpolation Methods

In the following sections, special interpolation methods are discussed. These methods enable the calculation of time-variant filter coefficients for sampling rate conversion and need an oversampled input sequence as well as the time instant of the output sample. A convolution of the oversampled input sequence with time-variant filter coefficients gives the output sample at the output sampling rate. This real-time computation of filter coefficients is not based on popular filter design methods like windowing or the Remez Exchange Algorithm [Rab75]. On the contrary, methods are presented for calculating filter coefficient sets for every input clock cycle where the filter coefficients are derived from the distance of output samples to the time grid of the oversampled input sequence.

8.3.1 Polynomial Interpolation

The aim of a polynomial interpolation [Liu92] is to determine a polynomial

$$p_N(x) = \sum_{i=0}^{N} a_i x^i \tag{8.37}$$

of Nth order representing exactly a function $f(x)$ at $N+1$ uniformly spaced x_i, i.e. $p_N(x_i) = f(x_i) = y_i$ for $i = 0, \ldots, N$. This can be written as a set of linear equations

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^N \\ 1 & x_1 & x_1^2 & \cdots & x_1^N \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_N & x_N^N & \cdots & x_N^N \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_N \end{bmatrix}. \tag{8.38}$$

The polynomial coefficients a_i as functions of $y_0 \ldots y_N$ are obtained with the help of Cramer's Rule according to

$$a_i = \frac{\begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & \overset{\text{ith column}}{y_0} & \cdots & x_0^N \\ 1 & x_1 & x_1^2 & \cdots & y_1 & \cdots & x_1^N \\ \vdots & \vdots & \vdots & & \vdots & \cdots & \vdots \\ 1 & x_N & x_N^2 & \cdots & y_N & \cdots & x_N^N \end{vmatrix}}{\begin{vmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^N \\ 1 & x_1 & x_1^2 & \cdots & x_1^N \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^N \end{vmatrix}}, \qquad i = 0, 1, \ldots, N. \tag{8.39}$$

For uniformly spaced $x_i = i$ with $i = 0, 1, \ldots, N$ the interpolation of an output sample with distance α gives

$$y(n+\alpha) = \sum_{i=0}^{N} a_i (n+\alpha)^i. \tag{8.40}$$

In order to determine the relationship between the output sample $y(n+\alpha)$ and y_i, a set of time-variant coefficients c_i needs to be determined such that

$$y(n+\alpha) = \sum_{i=-N/2}^{N/2} c_i(\alpha) y(n+i). \tag{8.41}$$

The calculation of time-variant coefficients $c_i(\alpha)$ will be illustrated by an example.

Example: Figure 8.18 shows the interpolation of an output sample of distance α with $N = 2$ and using 3 samples which can be written as

$$y(n+\alpha) = \sum_{i=0}^{2} a_i (n+\alpha)^i. \tag{8.42}$$

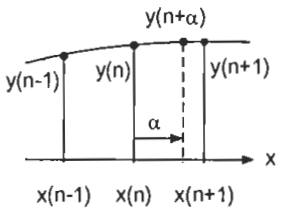


Figure 8.18 Polynomial interpolation with 3 samples.

The samples $y(n+i)$, with $i = -1, 0, 1$, can be expressed as

$$\begin{aligned} y(n+1) &= \sum_{i=0}^{2} a_i (n+1)^i \qquad \alpha = 1 \\ y(n) &= \sum_{i=0}^{2} a_i n^i \qquad \alpha = 0 \\ y(n-1) &= \sum_{i=0}^{2} a_i (n-1)^i \qquad \alpha = -1 \end{aligned} \tag{8.43}$$

or in matrix notation

$$\begin{bmatrix} 1 & (n+1) & (n+1)^2 \\ 1 & n & n^2 \\ 1 & (n-1) & (n-1)^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y(n+1) \\ y(n) \\ y(n-1) \end{bmatrix}. \tag{8.44}$$

The coefficients a_i as functions of y_i are then given by

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \frac{n(n-1)}{2} & 1-n^2 & \frac{n(n+1)}{2} \\ -\frac{2n-1}{2} & 2n & -\frac{2n+1}{2} \\ \frac{1}{2} & -1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} y(n+1) \\ y(n) \\ y(n-1) \end{bmatrix}, \tag{8.45}$$

such that

$$y(n+\alpha) = a_0 + a_1(n+\alpha) + a_2(n+\alpha)^2. \tag{8.46}$$

is valid. The output sample $y(n+\alpha)$ can be written as

$$\begin{aligned} y(n+\alpha) &= \sum_{i=-1}^{1} c_i(\alpha) y(n+i) \\ &= c_{-1}y(n-1) + c_0 y(n) + c_1 y(n+1). \end{aligned} \tag{8.47}$$

Equation (8.46) with a_i from Equation (8.45) leads to

$$\begin{aligned} y(n+\alpha) &= \left[\frac{1}{2}y(n+1) - y(n) + \frac{1}{2}y(n-1)\right](n+\alpha)^2 \\ &\quad + \left[-\frac{2n-1}{2}y(n+1) + 2ny(n) - \frac{2n+1}{2}y(n-1)\right](n+\alpha) \\ &\quad + \frac{n(n-1)}{2}y(n+1) + (1-n^2)y(n) + \frac{n(n+1)}{2}y(n-1). \end{aligned} \tag{8.48}$$

Comparing the coefficients from (8.47) and (8.48) for $n = 0$ gives the coefficients

$$\begin{aligned} c_{-1} &= \frac{1}{2}\alpha(\alpha-1) \\ c_0 &= -(\alpha-1)(\alpha+1) = 1-\alpha^2 \\ c_1 &= \frac{1}{2}\alpha(\alpha+1). \end{aligned}$$

8.3.2 Lagrange Interpolation

Lagrange interpolation for $N+1$ samples makes use of the polynomials $l_i(x)$ which have the following properties (see Fig. 8.19):

$$l_i(x_k) = \delta_{ik} = \begin{cases} 1 & i = k \\ 0 & \text{elsewhere} \end{cases} . \tag{8.49}$$

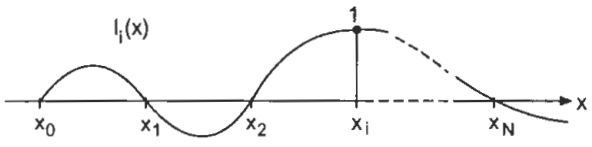


Figure 8.19 Lagrange polynomial.

Based on the zeros of the polynomial $l_i(x)$, it follows that

$$l_i(x) = a_i(x - x_0)\dots(x - x_{i-1})(x - x_{i+1})\dots(x - x_N). \tag{8.50}$$

With $l_i(x_i) = 1$ the coefficients are given by

$$a_i(x_i) = \frac{1}{(x_i - x_0)\dots(x_i - x_{i-1})(x_i - x_{i+1})\dots(x_i - x_N)}. \tag{8.51}$$

The interpolation polynomial is expressed as

$$\begin{aligned} p_N(x) &= \sum_{i=0}^{N} l_i(x) y_i \\ &= l_0(x)y_0 + \dots + l_N(x)y_N. \end{aligned} \tag{8.52}$$

With $a = \prod_{j=0}^{N}(x - x_j)$, (8.50) can be written as

$$\begin{aligned} l_i(x) &= a_i \frac{a}{x - x_i} = \frac{1}{\prod_{j=0, j\neq i}^{N} x_i - x_j} \frac{\prod_{j=0}^{N} x - x_j}{x - x_i} \\ &= \prod_{j=0, j\neq i}^{N} \frac{x - x_j}{x_i - x_j}. \end{aligned} \tag{8.53}$$

For uniformly spaced samples

$$x_i = x_0 + ih \tag{8.54}$$

and with the new variable α as given by

$$x = x_0 + \alpha h \tag{8.55}$$

we get

$$\frac{x - x_j}{x_i - x_j} = \frac{(x_0 + \alpha h) - (x_0 + jh)}{(x_0 + ih) - (x_0 + jh)} = \frac{\alpha - j}{i - j} \tag{8.56}$$

and hence

$$l_i(x(\alpha)) = \prod_{j=0, j\neq i}^{N} \frac{\alpha - j}{i - j}. \tag{8.57}$$

For even N we can write

$$l_i(x(\alpha)) = \prod_{j=-\frac{N}{2}, j\neq i}^{\frac{N}{2}} \frac{\alpha - j}{i - j} \tag{8.58}$$

and for odd N

$$l_i(x(\alpha)) = \prod_{j=-\frac{N-1}{2}, j\neq i}^{\frac{N+1}{2}} \frac{\alpha - j}{i - j}. \tag{8.59}$$

The interpolation of an output sample is given by

$$y(n + \alpha) = \sum_{i=-N/2}^{N/2} l_i(\alpha) y(n + i). \tag{8.60}$$

Example: $N = 2$, 3 samples

$$\begin{aligned} l_{-1}(x(\alpha)) &= \prod_{j=-1, j\neq -1}^{1} \frac{\alpha - j}{-1 - j} = \frac{1}{2}\alpha(\alpha - 1) \\ l_0(x(\alpha)) &= \prod_{j=-1, j\neq 0}^{1} \frac{\alpha - j}{0 - j} = -(\alpha - 1)(\alpha + 1) = 1 - \alpha^2 \\ l_1(x(\alpha)) &= \prod_{j=-1, j\neq 1}^{1} \frac{\alpha - j}{1 - j} = \frac{1}{2}\alpha(\alpha + 1). \end{aligned}$$

8.3.3 Spline Interpolation

The interpolation using piecewise defined functions that only exist over finite intervals is called Spline Interpolation [Cuc91].

A B-Spline $M_k^N(x)$ of Nth order using $m + 1$ samples is defined in the interval $[x_k, \ldots, x_{k+m}]$ by

$$M_k^N(x) = \sum_{i=k}^{k+m} a_i \phi_i(x) \tag{8.61}$$

with the truncated power functions

$$\phi_i(x) = (x - x_i)_+^N = \begin{cases} 0 & x < x_i \\ (x - x_i)^N & x \geq x_i \end{cases} . \tag{8.62}$$

In the following $M_0^N(x) = \sum_{i=0}^{m} a_i \phi_i(x)$ will be considered for $k = 0$ where $M_0^N(x) = 0$ for $x < x_0$ and $M_0^N(x) = 0$ for $x \geq x_m$. Figure 8.20 shows the truncated power functions and the B-Spline of Nth order. With the definition of

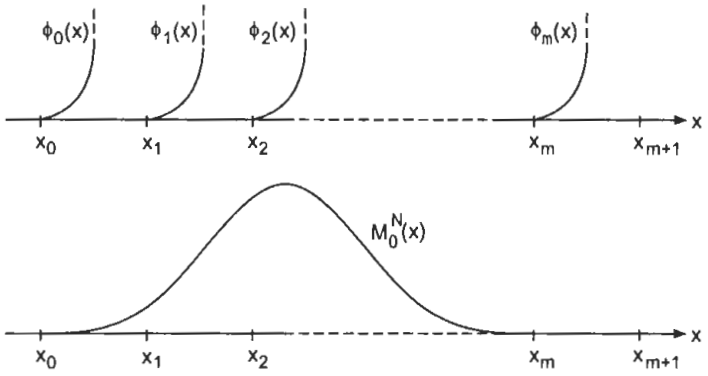


Figure 8.20 Truncated power functions and the B-Spline of Nth order.

the truncated power functions we can write

$$\begin{aligned} M_0^N(x) &= a_0\phi_0(x) + a_1\phi_1(x) + \ldots + a_m\phi_m(x) \\ &= a_0(x - x_0)_+^N + a_1(x - x_1)_+^N + \ldots + a_m(x - x_m)_+^N \end{aligned} \tag{8.63}$$

and after some calculations we get

$$\begin{aligned} M_0^N(x) = \; & a_0(x_0^N + c_1 x_0^{N-1} x + \ldots + c_{N-1} x_0 x^{N-1} + x^N) \\ & + a_1(x_1^N + c_1 x_1^{N-1} x + \ldots + c_{N-1} x_1 x^{N-1} + x^N) \\ & \vdots \\ & + a_m(x_m^N + c_1 x_m^{N-1} x + \ldots + c_{N-1} x_m x^{N-1} + x^N). \end{aligned} \tag{8.64}$$

With the condition $M_0^N(x) = 0$ for $x \geq x_m$, the following set of linear equations can be written with (8.64) and the coefficients of the powers of x:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_m \\ x_0^2 & x_1^2 & \cdots & x_m^2 \\ \vdots & \vdots & & \vdots \\ x_0^N & x_1^N & \cdots & x_m^N \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \tag{8.65}$$

The homogeneous set of linear equations has non-trivial solutions for $m > N$. The minimum requirement results in $m = N + 1$. For $m = N + 1$, the coefficients [Boe93] can be obtained as follows

$$a_i = \frac{\begin{vmatrix} 1 & 1 & 1 & \cdots & \overset{i\text{th column}}{0} & \cdots & 1 \\ x_0 & x_1 & x_2 & \cdots & 0 & \cdots & x_{N+1} \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ x_0^N & x_1^N & x_2^N & \cdots & 0 & \cdots & x_{N+1}^N \end{vmatrix}}{\begin{vmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_0 & x_1 & x_2 & \cdots & x_{N+1} \\ \vdots & \vdots & \vdots & & \vdots \\ x_0^{N+1} & x_1^{N+1} & x_2^{N+1} & \cdots & x_{N+1}^{N+1} \end{vmatrix}}, \qquad i = 0, 1, \ldots, N+1. \tag{8.66}$$

This can be rewritten as

$$a_i = \frac{1}{\prod_{j=0, i \neq j}^{N+1} (x_i - x_j)} \tag{8.67}$$

and hence

$$M_0^N(x) = \sum_{i=0}^{N+1} \frac{(x - x_i)_+^N}{\prod_{j=0, i \neq j}^{N+1} (x_i - x_j)}. \tag{8.68}$$

For some k

$$M_k^N(x) = \sum_{i=k}^{k+N+1} \frac{(x - x_i)_+^N}{\prod_{j=0, i \neq j}^{N+1} (x_i - x_j)}. \tag{8.69}$$

Since the functions $M_k^N(x)$ decrease with increasing N, a normalization of the form

$$N_k^N(x) = (x_{k+N+1} - x_k) M_k^N \tag{8.70}$$

is performed. The next example illustrates the computation of B-Splines.

Example: $N = 3$, $m = 4$, 5 samples

With the help of the matrix

$$\mathbf{U} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ x_0 & x_1 & x_2 & x_3 & x_4 \\ x_0^2 & x_1^2 & x_2^2 & x_3^2 & x_4^2 \\ x_0^3 & x_1^3 & x_2^3 & x_3^3 & x_4^3 \\ x_0^4 & x_1^4 & x_2^4 & x_3^4 & x_4^4 \end{bmatrix} \tag{8.71}$$

with corresponding determinant

$$\begin{aligned} \det \mathbf{U} \quad = \quad & (x_4 - x_3)(x_4 - x_2)(x_4 - x_1)(x_4 - x_0) \ldots \\ & (x_3 - x_2)(x_3 - x_1)(x_3 - x_0)(x_2 - x_1)(x_2 - x_0)(x_1 - x_0) \end{aligned}$$

and the matrix

$$\mathbf{M} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ x_0 & x_1 & x_2 & x_3 & x_4 \\ x_0^2 & x_1^2 & x_2^2 & x_3^2 & x_4^2 \\ x_0^3 & x_1^3 & x_2^3 & x_3^3 & x_4^3 \\ \phi_0(x) & \phi_1(x) & \phi_2(x) & \phi_3(x) & \phi_4(x) \end{bmatrix} \tag{8.72}$$

with corresponding determinant

$$\begin{aligned} \det \mathbf{M} \;=\; & +\phi_0(x)[(x_4-x_3)(x_4-x_2)(x_4-x_1)(x_3-x_2)(x_3-x_1)(x_2-x_1)] \\ & -\phi_1(x)[(x_4-x_3)(x_4-x_2)(x_4-x_0)(x_3-x_2)(x_3-x_0)(x_2-x_0)] \\ & +\phi_2(x)[(x_4-x_3)(x_4-x_1)(x_4-x_0)(x_3-x_1)(x_3-x_0)(x_1-x_0)] \\ & -\phi_3(x)[(x_4-x_2)(x_4-x_1)(x_4-x_0)(x_2-x_1)(x_2-x_0)(x_1-x_0)] \\ & +\phi_4(x)[(x_3-x_2)(x_3-x_1)(x_3-x_0)(x_2-x_1)(x_2-x_0)(x_1-x_0)] \end{aligned}$$

the B-spline can be expressed as

$$M_0^N(x) = \frac{\det \mathbf{M}}{\det \mathbf{U}}. \tag{8.73}$$

The coefficients can be derived by

$$a_i = \frac{\phi_i(x)[\ldots]}{\det \mathbf{U}} \tag{8.74}$$

and hence we get

$$\begin{aligned} a_0 &= \frac{1}{(x_0-x_4)(x_0-x_3)(x_0-x_2)(x_0-x_1)} \\ a_1 &= \frac{1}{(x_1-x_4)(x_1-x_3)(x_1-x_2)(x_1-x_0)} \\ a_2 &= \frac{1}{(x_2-x_4)(x_2-x_3)(x_2-x_1)(x_2-x_0)} \\ a_3 &= \frac{1}{(x_3-x_4)(x_3-x_2)(x_3-x_1)(x_3-x_0)} \\ a_4 &= \frac{1}{(x_4-x_3)(x_4-x_2)(x_2-x_1)(x_3-x_0)}. \end{aligned}$$

Figure 8.21a,b shows the truncated power functions and their summation for calculating $N_0^3(x)$. In Fig. 8.21c the horizontally shifted $N_i^3(x)$ are depicted.

A linear combination of B-splines is called a spline. Figure 8.22 shows the interpolation of sample $y(n+\alpha)$ for splines of second and third order. The shifted B-splines $N_i^N(x)$ are evaluated at the vertical line representing the distance α.

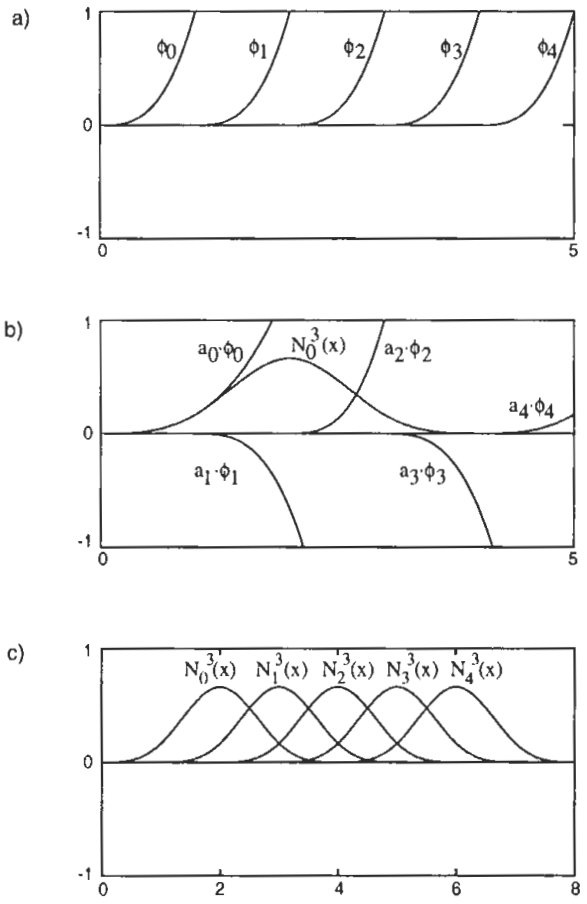


Figure 8.21 Third-order B-spline ($N = 3$, $m = 4$, 5 samples).

With sample $y(n)$ and the normalized B-splines $N_i^N(x)$, the second- and third-order splines are expressed as

$$y(n+\alpha) = \sum_{i=-1}^{1} y(n+i) N_{n-1+i}^2(\alpha) \tag{8.75}$$

$$y(n+\alpha) = \sum_{i=-1}^{2} y(n+i) N_{n-2+i}^3(\alpha). \tag{8.76}$$

Owing to the symmetrical characteristic of B-splines, the time-variant coefficients of the second-order B-spline can be derived:

$$N_3^2(\alpha) = h(1) = -\frac{1}{2}\alpha^2 \tag{8.77}$$

$$N_2^2(\alpha) = h(2) = -\frac{1}{2}(1+\alpha)^2 + \frac{3}{2}\alpha^2 \tag{8.78}$$

$$N_1^2(\alpha) = h(3) = -\frac{1}{2}(1-\alpha)^2. \tag{8.79}$$

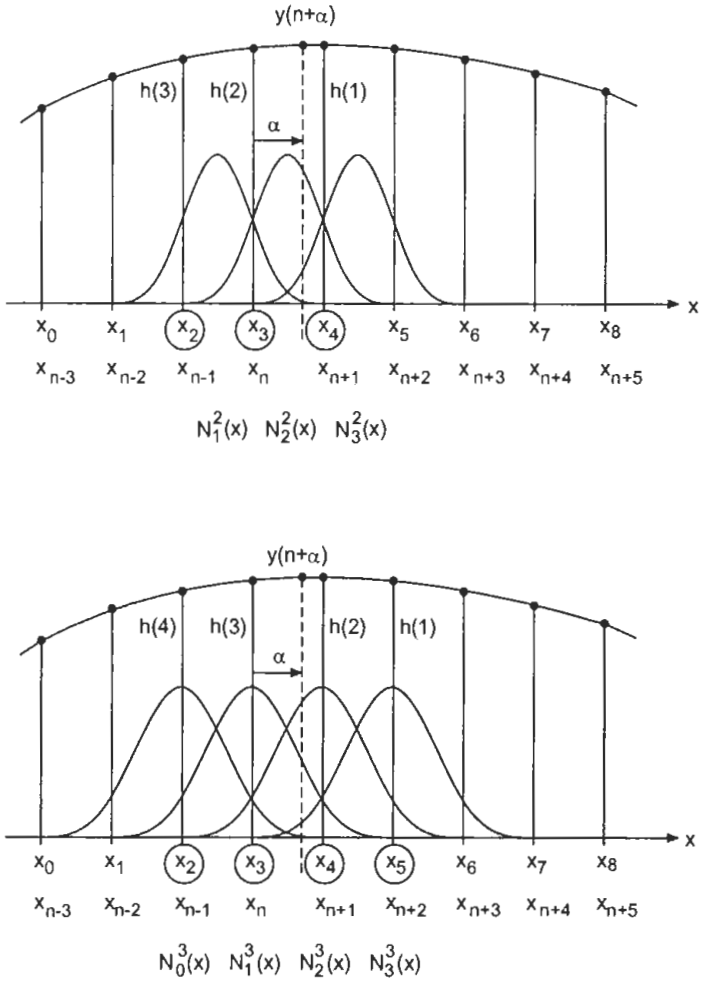


Figure 8.22 Interpolation with B-splines of second and third order.

The time-variant coefficients of a third-order B-spline are given by

$$N_3^3(\alpha) = h(1) = \frac{1}{6}\alpha^3 \tag{8.80}$$

$$N_2^3(\alpha) = h(2) = \frac{1}{6}(1+\alpha)^3 - \frac{2}{3}\alpha^3 \tag{8.81}$$

$$N_1^3(\alpha) = h(3) = \frac{1}{6}(2-\alpha)^3 - \frac{2}{3}(1-\alpha)^3 \tag{8.82}$$

$$N_0^3(\alpha) = h(4) = \frac{1}{6}(1-\alpha)^3. \tag{8.83}$$

Higher-order B-splines are given by:

$$y(n+\alpha) = \sum_{i=-2}^{2} y(n+i) N_{n-2+i}^4(\alpha) \tag{8.84}$$

$$y(n+\alpha) = \sum_{i=-2}^{3} y(n+i) N_{n-3+i}^{5}(\alpha) \tag{8.85}$$

$$y(n+\alpha) = \sum_{i=-3}^{3} y(n+i) N_{n-3+i}^{6}(\alpha). \tag{8.86}$$

Similar sets of coefficients can be derived here as well. Figure 8.23 illustrates this for fourth- and sixth-order B-splines.

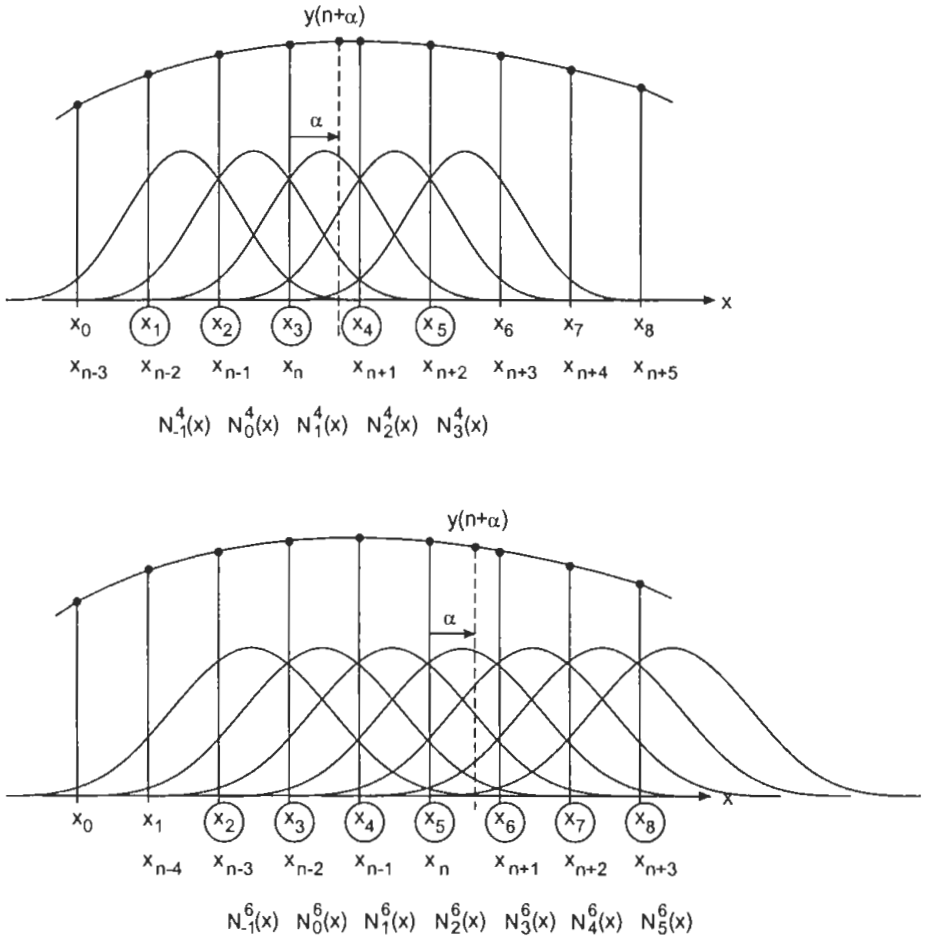


Figure 8.23 Interpolation with B-splines of fourth and sixth order.

Generally, for even orders we get

$$y(n+\alpha) = \sum_{i=-N/2}^{N/2} y(n+i) N_{n-N/2+i}^{N}(\alpha) \tag{8.87}$$

and for odd orders

$$y(n+\alpha) = \sum_{i=-(N-1)/2}^{(N+1)/2} y(n+i) N^N_{n-(N+1)/2+i}(\alpha). \tag{8.88}$$

For the application of interpolation the properties in the frequency-domain are important. The zero-order B-spline is given by

$$N^0_0(x) = \sum_{i=0}^{1} a_i \phi_i(x) = \begin{cases} 0 & x < 0 \\ 1 & 0 \le x < 1 \\ 0 & x \ge 1 \end{cases} \tag{8.89}$$

and the Fourier transform gives the sinc-function in the frequency-domain. The first-order B-spline given by

$$N^1_0(x) = 2\sum_{i=0}^{2} a_i \phi_i(x) = \begin{cases} 0 & x < 0 \\ \frac{1}{2}x & 0 \le x < 1 \\ 1 - \frac{1}{2}x & 1 \le x < 2 \\ 0 & x \ge 2 \end{cases}, \tag{8.90}$$

leads to a sinc^2-function in the frequency-domain. Higher-order B-splines can be derived by repeated convolution [Chu92] as given by

$$N^N(x) = N^0(x) * N^{N-1}(x). \tag{8.91}$$

Thus, the Fourier transform leads to

$$\mathrm{FT}[N^N(x)] = \mathrm{sinc}^{N+1}(f). \tag{8.92}$$

With the help of the properties in the frequency-domain, the necessary order of the spline interpolation can be determined. Owing to the attenuation properties of the $\mathrm{sinc}^{N+1}(f)$-function and the simple real-time calculation of the coefficients, spline interpolation is well suited to time-variant conversion in the last stage of a multistage sampling rate conversion system [Zöl94b].

Chapter 9

Data Compression

For transmission and storage of audio signals, different methods for compressing data have been investigated besides the PCM representation. Data compression can be divided into two types: *lossless* and *lossy* data compression.

9.1 Lossless Data Compression

Lossless data compression is based on linear prediction followed by entropy coding [Jay84] as shown in Fig. 9.1:

- Linear Prediction. A quantized set of coefficients P for a block of M samples is determined which leads to an estimate $\hat{x}(n)$ of the input sequence $x(n)$. The aim is to minimize the power of the difference signal $d(n)$ without any additional quantization errors, i.e. the word-length of the signal $\hat{x}(n)$ must be equal to the word-length of the input.
- Entropy Coding. Quantization of signal $d(n)$ due to the probability density function of the block. Samples $d(n)$ of greater probability are coded with shorter data words whereas samples $d(n)$ of lesser probability are coded with longer data words [Huf52].
- Frame Packing.

The attainable compression rates depend on the statistics of the audio signal and allow a compression rate of up to 2 [Bra92, Cel93].

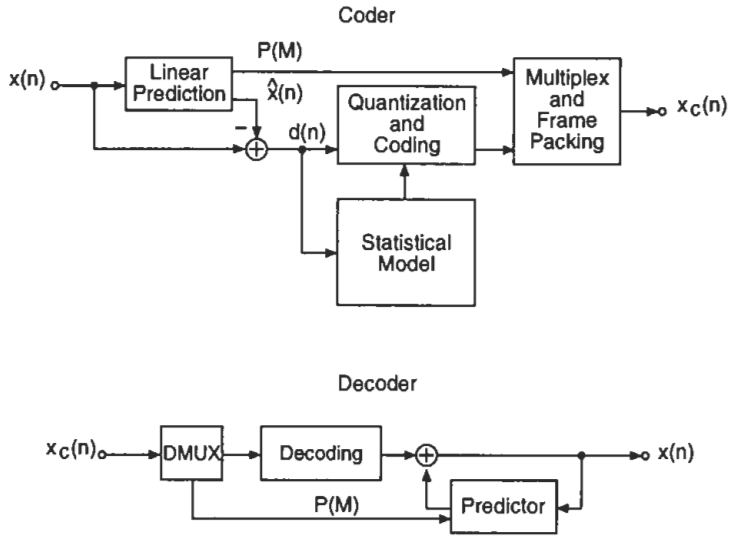


Figure 9.1 Lossless data compression based on linear prediction and entropy coding.

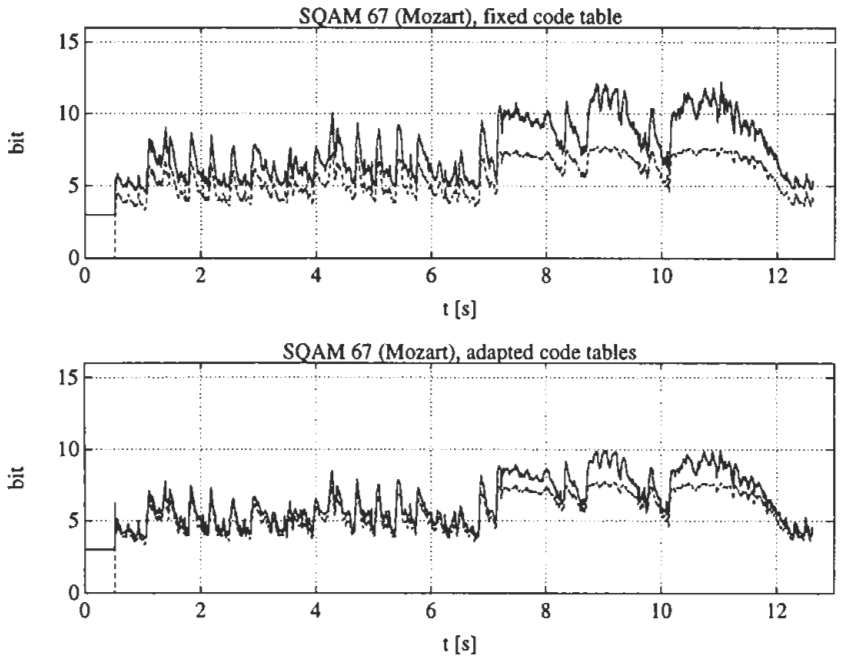


Figure 9.2 Lossless data compression (Mozart): word-length [bit] versus time (entropy - - , linear prediction with Huffman coding —).

Figure 9.2 illustrates examples of the necessary word-length for lossless data compression [Blo95, Sqa88]. Besides the entropy of the signal, results for linear prediction followed by Huffman coding [Huf52] are presented. Huffman coding is carried out with a fixed code table [Pen93] and a power-controlled choice of adapted

code tables. It is observed from Fig. 9.3 that for high signal powers, a reduction of word-length is possible if the choice is made from several adapted code tables.

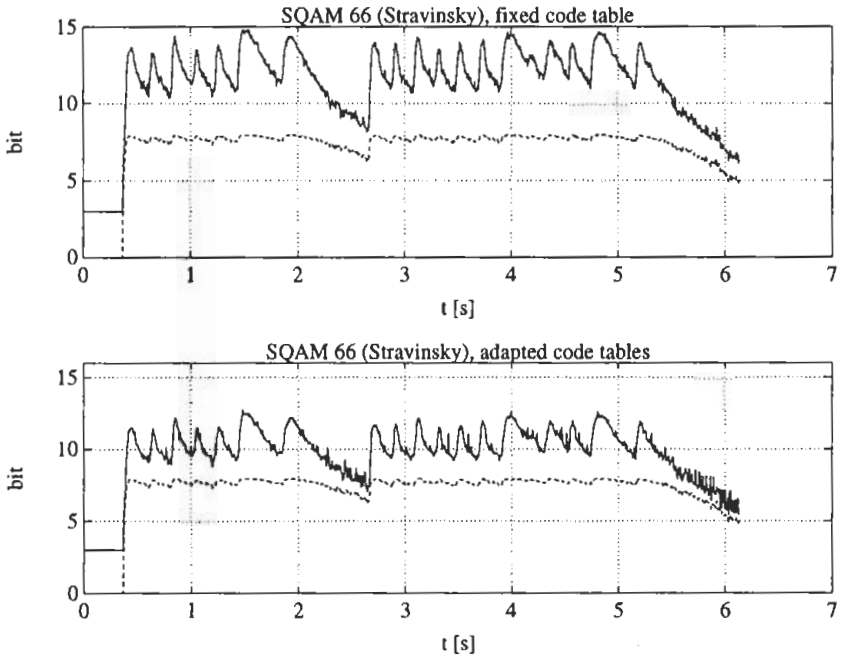


Figure 9.3 Lossless data compression (Stravinsky): word-length [bit] versus time (entropy - - , linear prediction with Huffman coding —).

Lossless compression methods are used for storage media with limited word-length (16 bit in CD and DAT) which are used for recording audio signals of higher word-lengths (> 16 bit). Further applications are in the transmission and archiving of audio signals.

9.2 Lossy Data Compression

Significantly higher compression rates (of factor 4 to 8) can be obtained with *lossy* coding methods. Psychoacoustic phenomena of human hearing are used for signal compression. The fields of application have a wide range, from professional audio like source coding for DAB to audio transmission via ISDN and home entertainment like DCC and MiniDisc.

An outline of the coding methods [Bra94] is standardized in an international specification ISO/IEC 11172-3 [ISO92], which is based on the following processing (see Fig. 9.4).

- Subband decomposition with filter banks of short latency time

- Calculation of psychoacoustic model parameters based on short-time FFT
- Dynamic bit allocation due to psychoacoustic model parameters (signal-to-mask ratio SMR_i)
- Quantization and coding of subband signals
- Multiplex and frame packing.

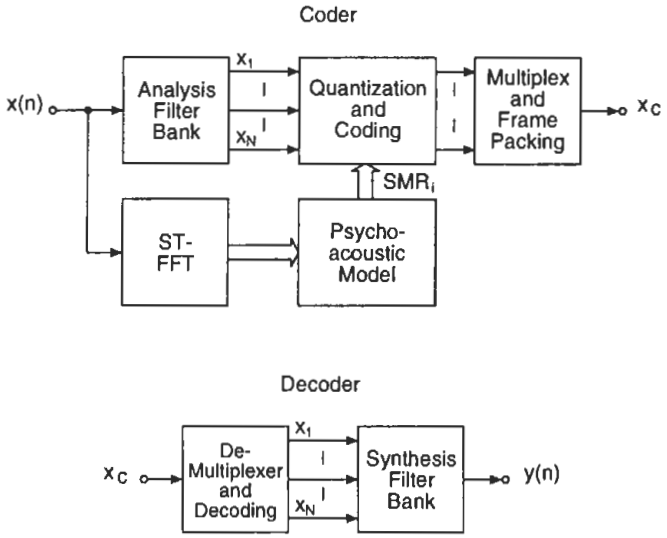


Figure 9.4 Lossy data compression based on subband coding and psychoacoustic models.

Owing to lossy data compression, post-processing of such signals or several coding/decoding steps is associated with some additional problems. The high compression rates justify the use of lossy data compression techniques in applications like transmission. In the next section, basic principles of psychoacoustics are presented followed by a description of the ISO-MPEG1 audio coding techniques.

9.3 Psychoacoustics

The results of psychoacoustic investigations by Zwicker [Zwi82, Zwi90] form the basis for audio data compression based on models of human perception. These coded audio signals have a significantly reduced data rate compared with the linearly quantized PCM representation. The human auditory system analyzes broad-band signals in so-called critical bands. The aim of psychoacoustic coding of audio signals is to decompose the broad-band audio signal into subbands which are matched to

the critical bands and then perform quantization and coding of these subband signals [Joh88a, Joh88b, Thei88]. Since the perception of sound below the absolute threshold of hearing is not possible, subband signals below this threshold need neither be coded nor transmitted. In addition to the perception in critical bands and the absolute threshold, the effects of signal masking in human perception play an important role in signal compression. These will be explained in the following and their application to psychoacoustic coding will be discussed.

9.3.1 Critical Bands and Absolute Threshold

Critical Bands. Critical bands as investigated by Zwicker are listed in Table 9.1.

Table 9.1 Critical bands as given by Zwicker 1982.

z/Bark	f_l/Hz	f_u/Hz	Δf_G/Hz	f_c/Hz
0	0	100	100	50
1	100	200	100	150
2	200	300	100	250
3	300	400	100	350
4	400	510	110	450
5	510	630	120	570
6	630	770	140	700
7	770	920	150	840
8	920	1080	160	1000
9	1080	1270	190	1170
10	1270	1480	210	1370
11	1480	1720	240	1600
12	1720	2000	280	1850
13	2000	2320	320	2150
14	2320	2700	380	2500
15	2700	3150	450	2900
16	3150	3700	550	3400
17	3700	4400	700	4000
18	4400	5300	900	4800
19	5300	6400	1100	5800
20	6400	7700	1300	7000
21	7700	9500	1800	8500
22	9500	1200	2500	10500
23	12000	15500	3500	13500
24	15500			

A transformation of the linear frequency scale into a hearing adapted scale is given by Zwicker [Zwi90] (units of z in Bark)

$$\frac{z}{\text{Bark}} = 13 \arctan(0.76 \frac{f}{\text{kHz}}) + 3.5 \arctan(\frac{f}{7.5\text{kHz}})^2. \tag{9.1}$$

The individual critical bands have the following bandwidths

$$\Delta f_G = 25 + 75(1 + 1.4(\frac{f}{\text{kHz}})^2)^{0.69}. \tag{9.2}$$

Absolute Threshold. The absolute threshold L_{T_q} (threshold in quiet) denotes the curve of sound pressure level L [Zwi82] versus frequency, which leads to the perception of a sinusoidal tone. The absolute threshold is given by [Ter79]

$$\frac{L_{T_q}}{\text{dB}} = 3.64(\frac{f}{\text{kHz}})^{-0.8} - 6.5\exp(-0.6(\frac{f}{\text{kHz}} - 3.3)^2) + 10^{-3}(\frac{f}{\text{kHz}})^4. \tag{9.3}$$

Below the absolute threshold, no perception of signals is possible. Figure 9.5 shows the absolute threshold versus frequency. Band-splitting in critical bands and the

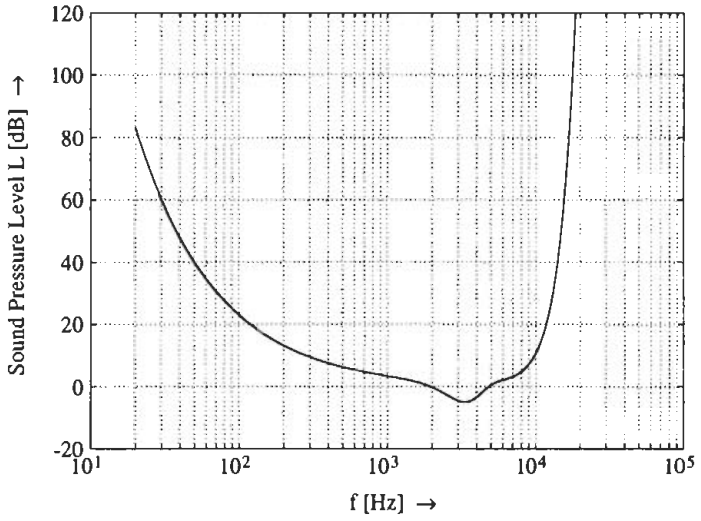


Figure 9.5 Absolute threshold (threshold in quiet).

absolute threshold allow the calculation of an offset between the signal level and the absolute threshold for every critical band. This offset is responsible for choosing appropriate quantization steps per critical band.

9.3.2 Masking

For data compression the use of sound perception in critical bands and absolute threshold only is not sufficient for high compression rates. The basis for further data reduction is the masking effects investigated by Zwicker. For band-limited noise or a sinusoidal signal, frequency-dependent masking thresholds can be given.

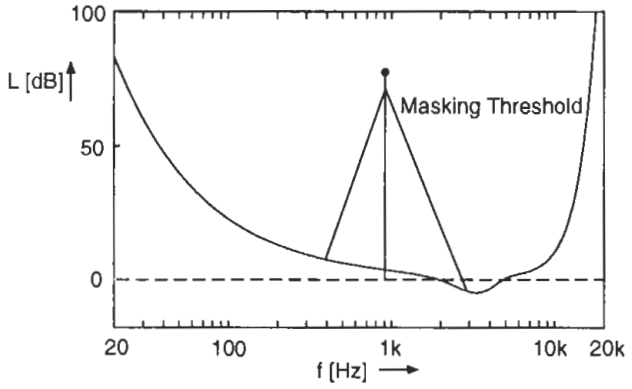


Figure 9.6 Masking threshold of band-limited noise.

These thresholds perform masking of frequency components if these components are below a masking threshold (see Fig. 9.6). The application of masking for perceptual coding is described in the following.

Calculation of Signal Power in Band i. First, the sound pressure level within a critical band is calculated. The short-time spectrum $X(k) = \mathrm{DFT}[x(n)]$ is used for calculating the power density spectrum

$$S_p(e^{j\Omega}) = S_p(e^{j\frac{2\pi k}{N}}) = X_R^2(e^{j\frac{2\pi k}{N}}) + X_I^2(e^{j\frac{2\pi k}{N}}) \tag{9.4}$$

$$S_p(k) = X_R^2(k) + X_I^2(k) \quad 0 \leq k \leq N-1 \tag{9.5}$$

with the help of an N-point FFT. The signal power in band i is calculated by the sum

$$S_p(i) = \sum_{\Omega=\Omega_{li}}^{\Omega_{ui}} S_p(k) \tag{9.6}$$

from the lower frequency up to the upper frequency of critical band i. The sound level pressure in band i is given by $L_S(i) = 10\log_{10} S_p(i)$.

Absolute Threshold. The absolute threshold is set such that a 4 kHz signal with peak amplitude ± 1 LSB for a 16 bit representation lies at the lower limit of the absolute threshold curve. Every masking threshold calculated in individual critical bands, which lies below the absolute threshold, is set to a value equal to the absolute threshold in the corresponding band. Since the absolute threshold within a critical band varies for low and high frequencies, it is necessary to make use of the mean absolute threshold within a band.

Masking Threshold. The offset between signal level and the masking threshold in critical band i (see Fig. 9.7) is given by [Hel72]

$$\frac{O(i)}{\mathrm{dB}} = \alpha(14.5 + i) + (1 - \alpha)a_v, \tag{9.7}$$

where α denotes the tonality index and a_v is the masking index. The masking

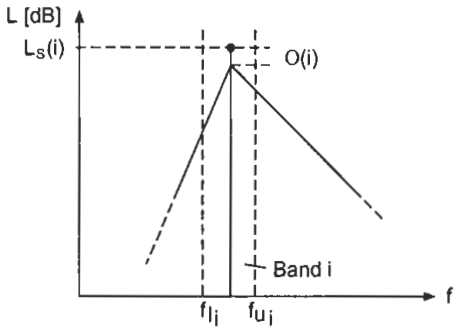


Figure 9.7 Offset between signal level and masking threshold.

index [Kap92] is given by

$$a_v = -2 - 2.05 \arctan(\frac{f}{4 \text{ kHz}}) - 0.75 \arctan(\frac{f^2}{2.56 \text{ kHz}^2}). \tag{9.8}$$

As an approximation

$$\frac{O(i)}{\text{dB}} = \alpha(14.5 + i) + (1 - \alpha)5.5 \tag{9.9}$$

can be used [Joh88a,b]. If a tone is masking a noise-like signal ($\alpha = 1$), the threshold is set $14.5 + i$ dB below the value of $L_S(i)$. If a noise-like signal is masking a tone ($\alpha = 0$), the threshold is set $5.5 + i$ dB below $L_S(i)$. In order to recognize a tonal or noise-like signal within a certain number of samples, the *Spectral Flatness Measure* SFM is estimated. The SFM is defined by the ratio of the geometric to arithmetic mean value of $S_p(i)$ according to

$$\text{SFM} = 10 \log_{10} \left(\frac{\left[\prod_{k=1}^{\frac{N}{2}} S_p(e^{j\frac{2\pi k}{N}}) \right]^{1/\frac{N}{2}}}{\frac{1}{N/2} \sum_{k=1}^{\frac{N}{2}} S_p(e^{j\frac{2\pi k}{N}})} i \right). \tag{9.10}$$

The SFM is compared with the SFM of a sinusoidal signal (definition $\text{SFM}_{\text{max}} = -60\text{dB}$) and the tonality index is calculated [Joh88a,b] by

$$\alpha = \text{MIN} \left(\frac{\text{SFM}}{\text{SFM}_{\text{max}}}, 1 \right). \tag{9.11}$$

SFM = 0 dB corresponds to a noise-like signal and leads to $\alpha = 0$, whereas an SFM = 75 dB gives a tone-like signal ($\alpha = 1$). With the sound pressure level $L_S(i)$ and the offset $O(i)$ the masking threshold is given by

$$T(i) = 10^{[L_S(i) - O(i)]/10}. \tag{9.12}$$

Masking Across Critical Bands. Masking across critical bands can be carried out with the help of the Bark scale. The masking threshold is of a triangular form which decreases with S_1 dB per Bark for the lower slope and with S_2 dB per Bark for the upper slope, depending on the sound pressure level L_i and the center frequency f_{c_i} in band i (see [Ter79]) according to

$$S_1 = 27 \quad \text{dB/Bark} \tag{9.13}$$

$$S_2 = 24 + 0.23(\frac{f_{c_i}}{\text{kHz}})^{-1} - 0.2\frac{L_S(i)}{\text{dB}} \quad \text{dB/Bark.} \tag{9.14}$$

An approximation of the minimum masking within a critical band can be made using Fig. 9.8 [Thei88, Sauv90]. Masking at the upper frequency f_{u_i} in the critical band i is responsible for masking the quantization noise with approximately 32 dB using the lower masking threshold that decreases by 27 dB/Bark. The upper slope has a steepness which depends on sound pressure level. This steepness is lower than the steepness of the lower slope.

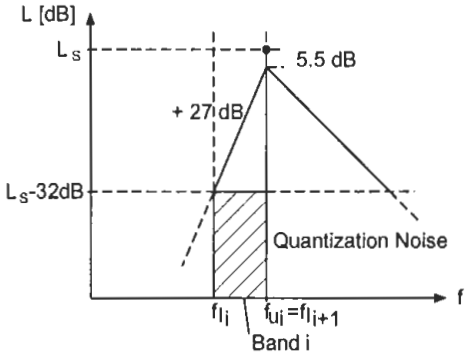


Figure 9.8 Masking within a critical band.

Masking across critical bands is presented in Fig. 9.9. The masking signal in critical band $i-1$ is responsible for masking the quantization noise in critical band i as well as the masking signal in critical band i. This kind of masking across critical bands further reduces the number of quantization steps within critical bands.

An analytical expression for masking across critical bands [Schr79] is given by

$$10\log_{10}[B(\Delta i)] = 15.81 + 7.5(\Delta i + 0.474) - 17.5[1 + (\Delta i + 0.474)^2]^{\frac{1}{2}}. \tag{9.15}$$

Δi denotes the distance between two critical bands in Bark. Expression (9.15) is called *spreading* function. With the help of this *spreading* function, masking of critical band i by critical band j can be calculated [Joh88a,b] with $\text{abs}(i-j) \leq 25$ such that

$$S_m(i) = B_{ij} \cdot S_p(i). \tag{9.16}$$

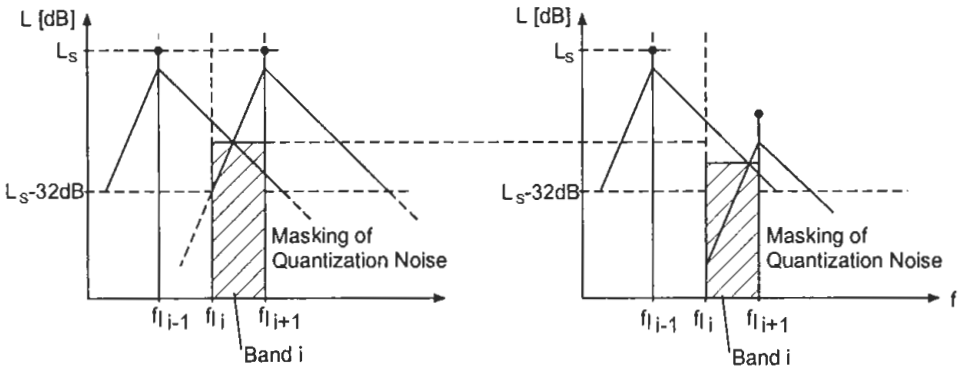


Figure 9.9 Masking across critical bands.

The masking across critical bands can therefore be expressed as a matrix operation given by

$$\begin{bmatrix} S_m(1) \\ S_m(2) \\ \vdots \\ S_m(25) \end{bmatrix} = \begin{bmatrix} B(0) & B(-1) & B(-2) & \cdots & B(-24) \\ B(1) & B(0) & B(-1) & \cdots & B(-23) \\ \vdots & \vdots & \vdots & & \vdots \\ B(24) & B(23) & B(22) & \cdots & B(0) \end{bmatrix} \begin{bmatrix} S_p(1) \\ S_p(2) \\ \vdots \\ S_p(25) \end{bmatrix}. \quad (9.17)$$

A renewed calculation of the masking threshold with (9.16) leads to the global masking threshold

$$T_m(i) = 10^{\log_{10} S_m(i) - O(i)/10}. \quad (9.18)$$

Figure 9.10 shows the absolute threshold and the individual masking thresholds of each masking signal as well as the global masking threshold.

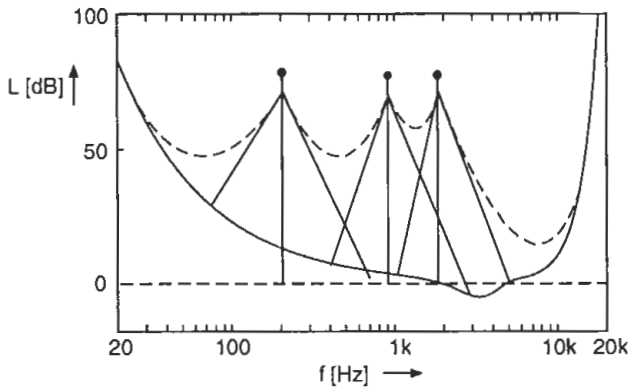


Figure 9.10 Masking thresholds for three masking signals.

For data compression, the following calculations for every critical band i have to be performed.

- Calculation of the signal power $S_p(i)$
  ($L_S(i)$ [dB])
- Absolute threshold
  ($L_{T_q}(i)$ [dB])
- Calculation of masking thresholds $T(i)$
  ($L_T(i)$ [dB])
- Calculation of global masking thresholds $T_m(i)$
  ($L_{T_m}(i)$ [dB])
- Calculation of the signal-to-mask ratio

$$\mathrm{SMR}_i = L_S(i) - L_{T_m}(i). \tag{9.19}$$

Owing to the signal-to-mask ratio, quantization in each of the critical bands and a dynamic bit allocation can be performed (see Fig. 9.11).

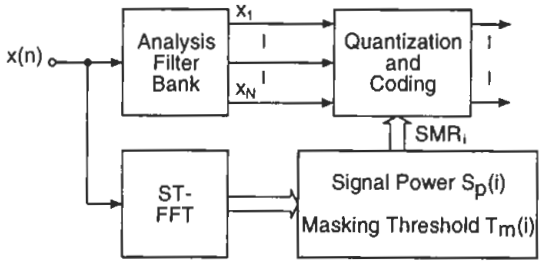


Figure 9.11 Calculation of signal-to-mask ratio SMR_i, quantization and dynamic bit allocation.

9.4 ISO-MPEG1 Audio Coding

In this section, the coding method for digital audio signals is described which is specified in the standard ISO/IEC 11172-3 [ISO92]. The filter banks used for subband decomposition, the psychoacoustic models, dynamic bit allocation and coding are discussed. A simplified block diagram of the coder for implementing layers I and II of the standard is shown in Fig. 9.12. The corresponding decoder is shown in Fig. 9.13. It uses the information from the ISO-MPEG1 frame and

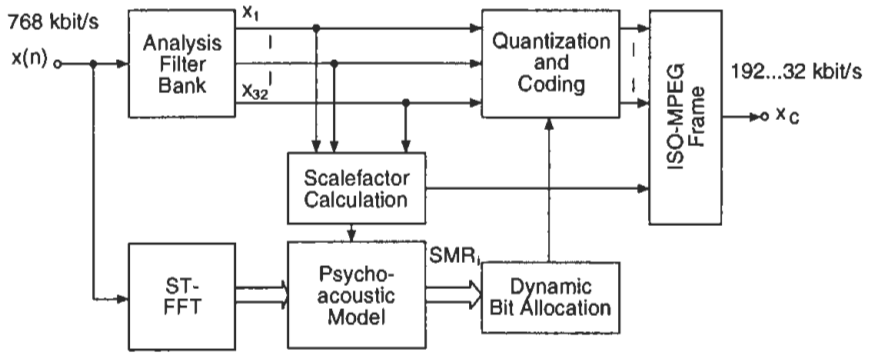


Figure 9.12 Simplified block diagram of a ISO-MPEG1 coder.

feeds the decoded subband signals to a synthesis filter bank for reconstructing the broad-band PCM signal. The complexity of the decoder is, in contrast to the coder, significantly lower. Prospective improvements of the coding method are being made entirely for the coder.

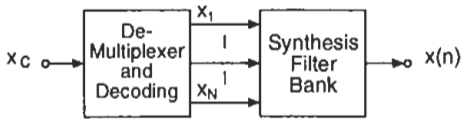


Figure 9.13 Simplified block diagram of a ISO-MPEG1 decoder.

9.4.1 Filter Banks

The subband decomposition is done with a Pseudo-QMF filter bank. The theoretical background is found in the related literature [Rot83, Mas85, Vai93]. The Pseudo-QMF filter bank is marked by its low complexity.

The decomposition of the broad-band signal is made into M uniformly spaced subbands. The subbands are processed further after a sampling rate reduction by a factor of M. The individual band-pass filters $H_0(z) \ldots H_{M-1}(z)$ are designed using a prototype low-pass filter $H(z)$ and frequency shifted versions. The frequency shifting of the prototype with cutoff frequency $\pi/2M$ is done by modulating the impulse response $h(n)$ with a cosine term. The band-pass filters have band-width π/M. For the synthesis filter bank, corresponding filters $F_0(z) \ldots F_{M-1}(z)$ give outputs which are added together resulting in a broad-band PCM signal. The implementation of an ISO-MPEG1 coder is based on $M = 32$ frequency bands. The Pseudo-QMF filter bank can be implemented by the combination of a polyphase filter structure followed by a discrete cosine transform [Rot83, Vai93, Kon94].

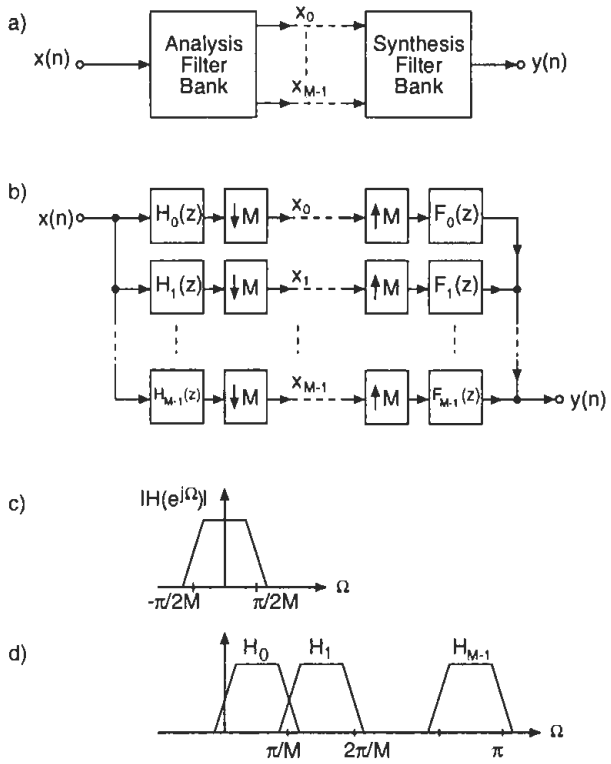


Figure 9.14 Pseudo-QMF filter bank.

For increasing the frequency resolution, layer III of the standard decomposes each of the 32 subbands further into a maximum of 18 uniformly spaced subbands (see Fig. 9.15). The decomposition is carried out with the help of an overlap-

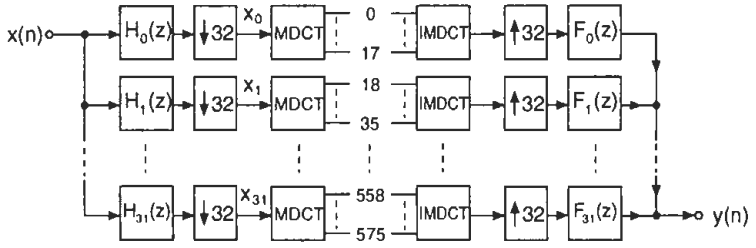


Figure 9.15 Polyphase/MDCT hybrid filter bank.

ped transform of windowed subband samples. The method is based on a modified discrete cosine transform, also known as the TDAC filter bank (Time Domain Aliasing Cancellation) and MLT (Modulated Lapped Transform). An exact description is found in [Pri87, Mal92]. This extended filter bank is denoted as the polyphase/MDCT hybrid filter bank [Bra94]. The higher frequency resolution enables a higher coding gain but has the disadvantage of having a worse time resolution. This is observed for impulse-like signals. In order to minimize these artifacts, the

number of subbands per subband can be altered from 18 down to 6. Subband decompositions that are matched to the signal can be obtained by specially designed window functions with overlapping transforms [Edl89].

9.4.2 Psychoacoustic Models

Two psychoacoustic models have been developed for layers I to III of the ISO-MPEG1 standard. Both models can be used independently of each other for all three layers. Psychoacoustic model 1 is used for layers I and II whereas model 2 is used for layer III. Owing to the numerous applications of layers I and II, we will discuss psychoacoustic model 1 in the following.

Psychoacoustic Model 1. Bit allocation in each of the 32 subbands is carried out using the signal-to-mask ratio SMR_i. This is based on the minimum masking threshold and the maximum signal level within a subband. In order to calculate this ratio, the power density spectrum is estimated with the help of a short-time FFT in parallel with the analysis filter bank. As a consequence, a higher frequency resolution is obtained for estimating the power density spectrum in contrast to the frequency resolution of the 32-band analysis filter bank. The signal-to-mask ratio for every subband is determined as follows:

1. Calculating the power density spectrum of a block of N samples using FFT. After windowing a block of $N = 512$ ($N = 1024$ for layer II) input samples, the power density spectrum

$$X(k) = 10 \log_{10} \left| \frac{1}{N} \sum_{n=0}^{N-1} h(n)x(n)e^{-jnk2\pi/N} \right|^2 \text{[dB]} \qquad (9.20)$$

   is calculated. After this, the window $h(n)$ is displaced by 384 ($12 \cdot 32$) samples and the next block is processed.

2. Determination of sound pressure level in every subband. The sound pressure level is derived from the calculated power density spectrum and by calculating a scaling factor in the corresponding subband as given by

$$L_S(i) = \mathrm{MAX}[X(k), 20 \log_{10}[SCF_{max}(i) * 32768] - 10] \quad \text{[dB]}. \qquad (9.21)$$

   For $X(k)$, the maximum of the spectral lines in a subband is used. The scaling factor SCF_i for subband i is calculated from the absolute value of the maximum of 12 consecutive subband samples. A nonlinear quantization to 64 levels is carried out (layer I). For layer II, the sound pressure level is determined by choosing the largest of the three scaling factors from $3 \cdot 12$ subband samples.

3. Considering the absolute threshold. The absolute threshold $LT_q(m)$ is specified for different sampling rates in [ISO92]. The frequency index m is based on a reduction of $N/2$ relevant frequencies with the FFT of index k (see Fig. 9.16). The subband index is still i.

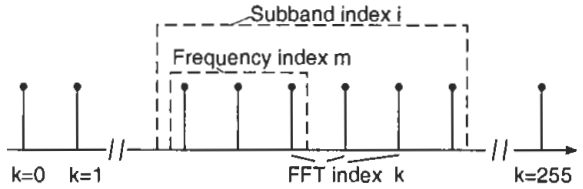


Figure 9.16 Nomenclature of frequency indices.

4. Calculating tonal $X_{tm}(k)$ or non-tonal $X_{nm}(k)$ masking components and determining relevant masking components (for details see [ISO92]). These masking components are denoted as $X_{tm}[z(j)]$ and $X_{nm}[z(j)]$. With the index j, tonal and non-tonal masking components are labeled. The variable $z(m)$ is listed for reduced frequency indices m in [ISO92]. It allows a finer resolution of the 24 critical bands with the frequency group index z.

5. Calculating the individual masking thresholds. For masking thresholds of tonal and non-tonal masking components $X_{tm}[z(j)]$ and $X_{nm}[z(j)]$, the following calculation is performed:

$$LT_{tm}[z(j), z(m)] = X_{tm}[z(j)] + a_{v_{tm}}[z(j)] + v_f[z(j), z(m)] \quad \text{[dB]} \quad (9.22)$$

$$LT_{nm}[z(j), z(m)] = X_{nm}[z(j)] + a_{v_{nm}}[z(j)] + v_f[z(j), z(m)] \quad \text{[dB]}. \quad (9.23)$$

The masking index for tonal masking components is given by

$$a_{v_{tm}} = -1.525 - 0.275 \cdot z(j) - 4.5 \quad \text{[dB]} \quad (9.24)$$

and the masking index for non-tonal masking components is

$$a_{v_{nm}} = -1.525 - 0.175 \cdot z(j) - 0.5 \quad \text{[dB]}. \quad (9.25)$$

The masking function $v_f[z(j), z(m)]$ with distance $\Delta z = z(m) - z(j)$ is given by

$$v_f = \begin{cases} 17 \cdot (\Delta z + 1) - (0.4 \cdot X[z(j)] + 6) & -3 \leq \Delta z < -1 \\ (0.4 \cdot X[z(j)] + 6) \cdot \Delta z & -1 \leq \Delta z < 0 \\ -17 \cdot \Delta z & 0 \leq \Delta z < 1 \\ -(\Delta z - 1) \cdot (17 - 0.15 \cdot X[z(j)]) - 17 & 1 \leq \Delta z < 8 \\ \text{[dB]} & \text{[Bark]}. \end{cases}$$

This masking function $v_f[z(j), z(m)]$ describes the masking of the frequency index $z(m)$ by the masking component $z(j)$.

6. Calculating the global masking threshold. For frequency index m, the global masking threshold is calculated as the sum of all contributing masking components according to

$$LT_g(m) = 10\log_{10}\left[10^{LT_q(m)/10} + \sum_{j=1}^{T_m} 10^{LT_{tm}[z(j),z(m)]/10} + \sum_{j=1}^{R_m} 10^{LT_{nm}[z(j),z(m)]/10}\right]. \quad (9.26)$$

The total number of tonal and non-tonal masking components are denoted as T_m and R_m respectively. For a given subband i, only masking components that lie in the range -8 to +3 Bark will be considered. Masking components outside this range are neglected.

7. Determination of the minimum masking threshold in every subband:

$$LT_{min}(i) = \text{MIN}[LT_g(m)] \quad [\text{dB}]. \quad (9.27)$$

Several masking thresholds $LT_g(m)$ can occur in a subband as long as m lies within the subband i.

8. Calculation of the signal-to-mask ratio SMR_i in every subband:

$$\text{SMR}_i = L_S(i) - LT_{min}(i) \quad [\text{dB}]. \quad (9.28)$$

The signal-to-mask ratio determines the dynamic range that has to be quantized in the particular subband so that the level of quantization noise lies below the masking threshold. The signal-to-mask ratio is the basis for the bit allocation procedure for quantizing the subband signals.

9.4.3 Dynamic Bit Allocation and Coding

Dynamic Bit Allocation. Dynamic bit allocation is used to determine the number of bits that are necessary for the individual subbands so that a transparent perception is possible. The minimum number of bits in subband i can be determined from the difference between scaling factor SCF_i and the absolute threshold $LT_q(i)$ as $b_i = SCF_i - LT_q(i)$. With this quantization noise remains under the masking threshold. Masking across critical bands is used for the implementation of the ISO-MPEG1 coding method.

For a given transmission rate, the maximum possible number of bits B_m for coding subband signals and scaling factors is calculated as

$$B_m = \sum_{i=1}^{32} b_i + SCF_i \cdot + \text{additional information.} \tag{9.29}$$

The bit allocation is performed within an allocation frame consisting of 12 subband samples ($384 = 12 \cdot 32$ PCM samples) for layer I and 36 subband samples ($1152 = 36 \cdot 32$ PCM samples) for layer II.

The dynamic bit allocation for the subband signals is carried out as an iterative procedure. At the beginning, the number of bits per subband is set to zero. First, the mask-to-noise ratio

$$\text{MNR}_i = \text{SNR}_i - \text{SMR}_i \tag{9.30}$$

is determined for every subband. The signal-to-mask ratio SMR_i is the result of the psychoacoustic model. The signal-to-noise ratio SNR_i is defined by a table in [ISO92], in which for every number of bits a corresponding signal-to-noise ratio is specified. The number of bits must be increased as long as the mask-to-noise ratio MNR is less than zero.

The iterative bit allocation is performed by the following steps:

1. Determination of the minimum MNR_i of all subbands.
2. Increasing the number of bits of these subbands on to the next stage of the MPEG1 standard. Allocation of 6 bits for the scaling factor of the MPEG1 standard when the number of bits is increased for the first time.
3. New calculation of MNR_i in this subband.
4. Calculation of the number of bits for all subbands and scaling factors and comparison with the maximum number B_m. If the number of bits is smaller than the maximum number, the iteration starts again with step 1.

Quantization and Coding of Subband Signals. The quantization of the subband signals is done with the allocated bits for the corresponding subband. The 12 (36) subband samples are divided by the corresponding scaling factor and then linearly quantized and coded (for details see [ISO92]). This is followed by a frame packing. In the decoder, the procedure is reversed. The decoded subband signals with different word-lengths are reconstructed to a broad-band PCM signal with a synthesis filter bank (see Fig. 9.13).

References

[Abu79] A.I. Abu-El-Haija, A.M. Peterson: *An Approach to Eliminate Roundoff Errors in Digital Filters*, IEEE Trans. ASSP, pp. 195-198, April 1979.

[Ada92] R. Adams, T. Kwan: *VLSI Architectures for Asynchronous Sample-Rate Conversion*, Proc. 93rd AES Convention, San Francisco, Preprint No. 3355, October 1992.

[Ada93] R. Adams, T. Kwan: *Theory and VLSI Implementations for Asynchronous Sample-Rate Conversion*, Proc. 94th AES Convention, Berlin, Preprint No. 3570, March 1993.

[AES91] AES10-1991 (ANSI S4.43-1991): AES Recommended Practice for Digital Audio Engineering - Serial Multichannel Audio Digital Interface (MADI).

[AES92] AES3-1992 (ANSI S4.40-1992): AES Recommended Practice for Digital Audio Engineering - Serial Transmission Format for Two-Channel Linearly Represented Digital Audio.

[Ala87] M. Alard, R. Lasalle: *Principles of Modulation and Channel Coding for Digital Broadcasting for Mobile Receivers*, EBU Review, No. 224, pp. 168-190, Aug. 1987.

[All79] J.B. Allen, D.A. Berkeley: *Image Method for Efficient Simulating Small Room Acoustics*, J. Acoust. Soc. Am., Vol. 65 , No. 4, pp. 943-950, 1979.

[And85] Y. Ando: *Concert Hall Acoustics*, Springer-Verlag, 1985.

[And92] M. Andersen: *New Principle for Digital Audio Power Amplifiers*, Proc. 92nd AES Convention, Preprint No. 3226, Vienna 1992.

[Ave71] E. Avenhaus: *Zum Entwurf digitaler Filter mit minimaler Speicherwortlänge für Koeffizienten und Zustandsgrössen*, Ausgewählte Arbeiten über Nachrichtensysteme, Nr. 13, herausgegeben von Prof. Dr.-Ing. H.W. Schüssler, Erlangen 1971.

[Bar82] C.W. Barnes: *Error Feedback in Normal Realizations of Recursive Digital Filters*, IEEE Trans. Circuits and Systems, pp. 72-75, Jan. 1982.

[Bar71] M. Barron: *The Subjective Effects of First Reflections in Concert Halls - The Need for Lateral Reflections*, J. Sound and Vibration 15, pp. 475-494, 1971.

[Bar81] M. Barron, A.H. Marschall: *Spatial Impression Due to Early Lateral Reflections in Concert Halls: The Derivation of a Physical Measure*, J. Sound and Vibration 77, pp. 211-232, 1981.

[Ben88] K.B. Benson: *Audio Engineering Handbook*, McGraw-Hill, New York 1988.

[Bla74] J. Blauert: *Räumliches Hören*, S. Hirzel Verlag, Stuttgart, 1974.

[Bla85] J. Blauert: *Räumliches Hören*, Nachschrift-Neue Ergebnisse und Trends seit 1972, S. Hirzel Verlag, Stuttgart, 1985.

[Blo95] T. Block: *Untersuchung von Verfahren zur verlustlosen Datenkompression von digitalen Audiosignalen*, Studienarbeit, TU Hamburg-Harburg, 1995.

[Bom85] B.W. Bomar: *New Second-Order State-Space Structures for Realizing Low Roundoff Noise Digital Filters*, IEEE Trans. ASSP, pp. 106-110, Feb. 1985.

[Boe93] W. Boehm, H. Prautzsch: *Numerical Methods*, AK Peters-Vieweg, Braunschweig 1993.

[Bra92] K. Brandenburg, J. Herre: *Digital Audio Compression for Professional Applications*, Proc. 92nd AES Convention, Preprint No. 3330, Vienna 1992.

[Bra94] K. Brandenburg, G. Stoll: *ISO/MPEG-1 Audio: A Generic Standard for Coding of High Quality Digital Audio*, J. Audio Eng. Soc., Vol. 42, No. 10, pp. 780-792, October 1994.

[Cad87] J.A. Cadzow: *Foundations of Digital Signal Processing and Data Analysis*, New York: Macmillan Publishing Company, 1987.

[Can85] J.C. Candy: *A Use of Double Integration in Sigma Delta Modulation*, IEEE Trans. Commun., vol. COM-37, pp. 249-258, March 1985.

[Can92] J.C. Candy, G.C. Temes, Ed.: *Oversampling Delta-Sigma Data Converters*, IEEE Press, Piscataway, NJ, 1992.

[Cel93] C. Cellier, P. Chenes, M. Rossi: *Lossless Audio Data Compression for Real-Time Applications*, Proc. 95th AES Convention, Preprint No. 3780, New York 1993.

[Cha78] T.L. Chang: *A Low Roundoff Noise Digital Filter Structure*, Proc. Int. Symp. on Circuits and Systems, pp. 1004-1008, May 1978.

[Cha90] K. Chao et al: *A High Order Topology for Interpolative Modulators for Oversampling A/D Converters*, IEEE Trans. Circuits and Syst., vol. CAS-37, pp. 309-318, March 1990.

[Chu92] C.K. Chui (ed.): *Wavelets: A Tutorial in Theory and Applications*, Volume 2, Academic Press, Boston, 1992.

[Cre78] L. Cremer, H.A. Müller: *Die wissenschaftlichen Grundlagen der Raumakustik - Bd. 1 u. 2*, S. Hirzel Verlag, Stuttgart, 1978/76.

[Cro83] R.E. Crochiere, L.R. Rabiner: *Multirate Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, 1983.

[Cuc91] S. Cucchi, F. Desinan, G. Parladori, G. Sicuranza: *DSP Implementation of Arbitrary Sampling Frequency Conversion for High Quality Sound Application*, Proc. IEEE ICASSP-91, Toronto, pp. 3609-3612, May 1991.

[Duh88] P. Duhamel, B. Piron, J.M. Etcheto: *On Computing the Inverse DFT*, IEEE Trans. Acoust., Speech, Signal Processing, Vol. 36, No. 2, pp. 285-286, February 1988.

[Edl89] B. Edler: *Codierung von Audiosignalen mit überlappender Transformation und adaptiven Fensterfunktionen*, Frequenz, Vol. 43, pp. 252-256, 1989.

[Ell82] D.F. Elliott, K.R. Rao: *Fast Transforms: Algorithms, Analyses, Applications*, New York: Academic Press, 1982.

[Fet72] A. Fettweis: *On the Connection Between Multiplier Wordlength Limitation and Roundoff Noise in Digital Filters*, IEEE Trans. Circuit Theory, pp. 486-491, Sept. 1972.

[Fli91] N. Fliege: *Systemtheorie*, B.G. Teubner, Stuttgart 1991.

[Fli92] N.J. Fliege, U. Zölzer: *Multi-Complementary Filter Bank: A New Concept with Aliasing-Free Subband Signal Processing and Perfect Reconstruction*, Proc. EUSIPCO-92, Brussels, pp. 207-210, August 1992.

[Fli93] N.J. Fliege, U. Zölzer: *Multi-Complementary Filter Bank*, Proc. IEEE ICASSP-93, Minneapolis, pp. 193-196, April 1993.

[Fli94] N.J. Fliege: *Multirate Digital Signal Processing*, John Wiley & Sons, Chicester 1994.

[Gab87] R.A. Gabel, R.A. Roberts: *Signals and Linear Systems*, John Wiley & Sons, New York 1987.

[Ger71] M.A. Gerzon: *Synthetic Stereo Reverberation*, Studio Sound, No. 13, pp. 632-635, 1971 and No. 14, pp. 24-28, 1972.

[Ger76] M.A. Gerzon: *Unitary (Energy-Preserving) Multichannel Networks with Feedback*, Electronics Letters, Vol. 12, No. 11, pp. 278-279, 1976.

[Ger85] M.A. Gerzon: *Ambisonics in Multichannel Broadcasting and Video*, J. Audio Eng. Soc., Vol. 33, No. 11, pp. 859-871, November 1985.

[Ger89] M.A. Gerzon, P.G. Craven: *Optimal Noise Shaping and Dither of Digital Signals*, Proc. 87th AES Convention, New York, Preprint No. 2822, October 1989.

[Ger92] M.A. Gerzon: *The Design of Distance Panpots*, Proc. 92nd AES Convention, Preprint No. 3308, Vienna, 1992.

[Gol67] B. Gold, C.M. Rader: *Effects of Parameter Quantization on the Poles of a Digital Filter*, Proc. IEEE, pp. 688-689, May 1967.

[Gol90] J.M. Goldberg, M.B. Sandler: *New Results in PWM for Digital Power Amplification*, Proc. 89th AES Convention, Preprint No. 2959, Los Angeles 1990.

[Gri89] D. Griesinger: *Practical Processors and Programs for Digital Reverberation*, Proc. AES 7th Int. Conf., pp. 187-195, Toronto, 1989.

[Har93] F.J. Harris, E. Brooking: *A Versatile Parametric Filter Using an Embedded All-Pass Sub-Filter to Independently Adjust Bandwidth, Center Frequency and Boost or Cut*, Proc. 95th AES Convention, San Francisco, Preprint No. 3757, 1993.

[Hel72] R.P. Hellman: *Asymmetry in Masking Between Noise and Tone*, Perception and Psychophys., Vol. 11, pp. 241-246, 1972.

[Her94] T. Hertz: *Implementierung und Untersuchung von Rückkopplungssystemen zur digitalen Raumsimulation*, Diplomarbeit, TU Hamburg-Harburg, 1994.

[Hsi87] C.-C. Hsiao: *Polyphase Filter Matrix for Rational Sampling Rate Conversions*, Proc. IEEE ICASSP-87, Dallas, pp. 2173-2176, April 1987.

[Huf52] D.A. Huffman: *A Method for the Construction of Minimum-Redundancy Codes*, Proc. of the IRE, pp. 1098-1101, 1952.

[Ino63] H. Inose, Y. Yasuda: *A Unity Bit Coding Method by Negative Feedback*, Proc. IEEE, vol. 51, pp. 1524-1535, November 1963.

[ISO92] ISO/IEC 11172-3: *Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to 1.5 Mbits/s - Audio Part*, International Standard, 1992.

[Jay84] N.S. Jayant, P. Noll: *Digital Coding of Waveforms*, Prentice-Hall, New Jersey, 1984.

[Joh88a] J.D. Johnston: *Transform Coding of Audio Signals Using Perceptual Noise Criteria*, IEEE Journal on Selected Areas in Communications, Vol. 6, No. 2, pp. 314-323, February 1988.

[Joh88b] J.D. Johnston: *Estimation of Perceptual Entropy Using Noise Masking Criteria*, Proc. IEEE ICASSP-88, pp. 2524-2527, 1988.

[Jot91] J.M. Jot, A. Chaigne: *Digital Delay Networks for Designing Artificial Reverberators*, Proc. 94th AES Convention, Preprint No. 3030, 1991.

[Jot92] J.M. Jot: *An Analysis/Synthesis Approach to Real-Time Artificial Reverberation*, Proc. IEEE ICASSP-92, pp. 221-224, San Francisco, 1992.

[Jur64] E.I. Jury: *Theory and Application of the z-Transform Method*, Wiley, 1964.

[Kam89] K.D. Kammeyer, K. Kroschel: *Digitale Signalverarbeitung*, B.G. Teubner, Stuttgart, 1989.

[Kam92a] K.D. Kammeyer: *Nachrichtenübertragung*, Stuttgart, B.G. Teubner, 1992.

[Kam92b] K.D. Kammeyer, U. Tuisel, H. Schulze, H. Bochmann: *Digital Multicarrier-Transmission of Audio Signals over Mobile Radio Channels*, Europ. Trans. on Telecommun. ETT, vol. 3, pp. 243-254, May-June 1992.

[Kam93] K.D. Kammeyer, U. Tuisel: *Synchronisationsprobleme in digitalen Multiträgersystemen*, Frequenz, vol. 47, pp. 159-166, Mai 1993.

[Kap92] R. Kapust: *A Human Ear Related Objective Measurement Technique Yields Audible Error and Error Margin*, Proc. 11th Int. AES Conference - Test&Measurement, Portland, pp. 191-202, 1992.

[Kat85] Y. Katsumata, O. Hamada: *A Digital Audio Sampling Frequency Converter Employing New Digital Signal Processors*, Proc. 79th AES Convention, New York, Preprint No. 2272, October 1985.

[Kat86] Y. Katsumata, O. Hamada: *An Audio Sampling Frequency Conversion Using Digital Signal Processors*, Proc. IEEE ICASSP-86, Tokyo, pp. 33-36, 1986.

[Kin72] N.G. Kingsbury: *Second-Order Recursive Digital Filter Element for Poles Near the Unit Circle and the Real z-Axis*, Electronic Letters, pp. 155-156, March 1972.

[Klu92] J. Klugbauer-Heilmeier: *A Sigma Delta Modulated Switching Power Amp*, Proc. 92nd AES Convention, Preprint No. 3227, Vienna 1992.

[Kon94] K. Konstantinides: *Fast Subband Filtering in MPEG Audio Coding*, IEEE Signal Processing Letters, Vol. 1, No. 2, pp. 26-28, February 1994.

[Kut91] H. Kuttruff: *Room Acoustics*, 3rd Edition, Elsevier Applied Sciences, London, 1991.

[Lag81] R. Lagadec, H.O. Kunz: *A Universal, Digital Sampling Frequency Converter for Digital Audio*, Proc. IEEE ICASSP-81, Atlanta, pp. 595-598, April 1981.

[Lag82a] R. Lagadec, D. Pelloni, D. Weiss: *A Two-Channel Professional Digital Audio Sampling Frequency Converter*, Proc. 71st AES Convention, Montreux, Preprint No. 1882, March 1982.

[Lag82b] D. Lagadec, D. Pelloni, D. Weiss: *A 2-Channel, 16-Bit Digital Sampling Frequency Converter for Professional Digital Audio*, Proc. IEEE ICASSP-82, Paris, pp. 93-96, May 1982.

[Lag82c] R. Lagadec: *Digital Sampling Frequency Conversion*, Digital Audio, Collected Papers from the AES Premier Conference, pp. 90-96, June 1982.

[Lag83] R. Lagadec, D. Pelloni, A. Koch: *Single-Stage Sampling Frequency Conversion*, Proc. 74th AES Convention, New York, Preprint No. 2039, October 1983.

[Lin93] B. Link, D. Mandell: *A DSP Implementation of a Pro Logic Surround Decoder*, Proc. 95th AES Convention, Preprint No. 3758, New York 1993.

[Lip86] S.P. Lipshitz, J. Vanderkoy: *Digital Dither*, Proc. 81st AES Convention, Los Angeles, Preprint No. 2412, November 1986.

[Lip92] S.P. Lipshitz, R.A. Wannamaker, J. Vanderkoy: *Quantization and Dither: A Theoretical Survey*, J. Audio Eng. Soc., Vol. 40, No. 5, pp. 355-375, May 1992.

[Liu92] G.-S. Liu, C.-H. Wei: *A New Variable Fractional Delay Filter with Nonlinear Interpolation*, IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing, vol. 39, no. 2, pp. 123-126, February 1992.

[Mac76] F.J. MacWilliams, N.J.A. Sloane: *Pseudo-Random Sequences and Arrays*, IEEE Proceedings, Vol. 64, pp. 1715-1729, 1976.

[Mal92] H.S. Malvar: *Signal Processing with Lapped Transforms*, Artech House, Norwood, 1992.

[Mas85] J. Masson, Z. Picel: *Flexible Design of Computationally Efficient Nearly Perfect QMF Filter Banks*, Proc. IEEE ICASSP-85, pp. 541-544, 1985.

[Mat87] Y. Matsuya et al: *A 16-bit Oversampling A-to-D Conversion Technology Using Triple-Integration Noise Shaping*, IEEE J. Solid-State Circuits, vol. SC-22, pp. 921-929, Dec. 1987.

[McN84] G.W. McNally: *Dynamic Range Control of Digital Audio Signals*, J. Audio Eng. Soc., Vol. 32, No. 5, pp. 316-327, 1984.

[Moo78] J.A. Moorer: *About this Reverberation Business*, Computer Music Journal, Vol. 3, No. 2, pp. 13-28, 1978.

[Mul76] C.T. Mullis, R.A. Roberts: *Synthesis of Minimum Roundoff Noise Fixed Point Digital Filters*, IEEE Trans. Circuits and Systems, pp. 551-562, Sept. 1976.

[Opp75] A.V. Oppenheim, R.W. Schafer: *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs 1975.

[Par90] S. Park, R. Robles: *A Real-Time Method for Sample-Rate Conversion from CD to DAT*, Proc. IEEE Int. Conf. Consumer Electronics, Chicago, pp. 360-361, June 1990.

[Par91a] S. Park: *Low Cost Sample Rate Converters*, Proc. NAB Broadcast Engineering Conference, Las Vegas, April 1991.

[Par91b] S. Park, R. Robles: *A Novel Structure for Real-Time Digital Sample-Rate Converters with Finite Precision Error Analysis*, Proc. IEEE ICASSP-91, Toronto, pp. 3613-3616, May 1991.

[Pen93] W.B. Pennebaker, J.L. Mitchell: *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.

[Ple85] G. Plenge: *The German DBS Digital Sound Broadcasting System*, Proc. 77th AES Convention, Hamburg, Preprint No. 2203, March 1985.

[Ple91] G. Plenge: *DAB - Ein neues Hörrundfunksystem - Stand der Entwicklung und Wege zu seiner Einführung*, Rundfunktechnische Mitteilungen, Jahrg. 35 (1991), H. 2, S. 46-66.

[Pri87] J. Princen, A.W. Johnston, A. Bradley: *Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancelation*, Proc. IEEE ICASSP-87, pp. 2161-2164, 1987.

[Pro89] J.G. Proakis: *Digital Communications*, McGraw-Hill, New York 1989.

[Rab75] L.R. Rabiner, B. Gold: *Theory and Application of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs 1975.

[Ram82] T.A. Ramstad: *Sample-Rate Conversion by Arbitrary Ratios*, Proc. IEEE ICASSP-82, Paris, pp. 101-104, May 1982.

[Ram84] T.A. Ramstad: *Digital Methods for Conversion Between Arbitrary Sampling Frequencies*, IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-32, no. 3, pp. 577-591, June 1984.

[Ram88] T.A. Ramstad, T. Saramäki: *Efficient Multirate Realization for Narrow Transition-Band FIR Filters*, Proc. IEEE Int. Symp. on Circuits and Syst. (Espoo, Finland), pp. 2019-2022, June 1988.

[Ram90] T.A. Ramstad, T. Saramäki: *Multistage, Multirate FIR Filter Structures for Narrow Transition-Band Filters*, Proc. IEEE Int. Symp. on Circuits and Syst. (New Orleans, USA), pp. 2017-2021, May 1990.

[Reg87] P.A. Regalia, S.K. Mitra: *Tunable Digital Frequency Response Equalization Filters*, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, No. 1, pp. 118-120, January 1987.

[Roc97] D. Rocchesso, J.O. Smith: *Circulant and Elliptic Feedback Delay Networks for Artificial Reverberation*, IEEE Transactions on Speech and Audio Processing, Vol. 5, No. 1, pp. 51-63, January 1997.

[Rot83] J.H. Rothweiler: *Polyphase Quadrature Filters - A New Subband Coding Technique*, Proc. IEEE ICASSP-87, pp. 1280-1283, 1983.

[Sauv90] U. Sauvagerd: *Bitratenreduktion hochwertiger Musiksignale unter Verwendung von Wellendigitalfiltern*, VDI-Verlag, Düsseldorf 1990.

[Sch92] M. Schönle, U. Zölzer, N. Fliege: *Modeling of Room Impulse Responses by Multirate Systems*, Proc. 93rd AES Convention, Preprint No. 3447, San Francisco 1992.

[Sch93] M. Schönle, N.J. Fliege, U. Zölzer: *Parametric Approximation of Room Impulse Responses by Multirate Systems*, Proc. IEEE ICASSP-93, Vol. 1, pp. 153-156, 1993.

[Sch94] M. Schönle: *Wavelet-Analyse und parametrische Approximation von Raumimpulsantworten*, Dissertation, TU Hamburg-Harburg, 1994.

[Scp92] H. Schöpp, H. Hetzel: *New Methods of Adaptive Sound Reinforcement in Car Environment*, Proc. 92nd AES Convention, Preprint No. 3246, Vienna 1992.

[Schr61] M.R. Schroeder, B.F. Logan: *Colorless Artificial Reverberation*, J. Audio Eng. Soc., Vol. 9(3), pp. 192-197, 1961.

[Schr62] M.R. Schroeder: *Natural Sounding Artificial Reverberation*, J. Audio Eng. Soc., Vol. 10(3), pp. 219-223, 1962.

[Schr65] M.R. Schroeder: *New Method of Measuring Reverberation Time*, J. Acoust. Soc. Am., pp. 409-412, 1965.

[Schr70] M.R. Schroeder: *Digital Simulation of Sound Transmission in Reverberant Spaces*, J. Acoust. Soc. Am., Vol. 47 , No. 2, pp. 424-431, 1970.

[Schr79] M.R. Schroeder, B.S. Atal, J.L. Hall: *Optimizing Digital Speech Coders by Exploiting Masking Properties of the Human Ear*, J. Acoust. Soc. Am., Vol. 66, No. 6, pp. 1647-1652, Dec. 1979.

[Schr87] M.R. Schroeder: *Statistical Parameters of the Frequency Response Curves of Large Rooms*, J. Audio Eng. Soc., Vol. 35 , No. 5, pp. 299-305, 1987.

[Schü94] H.W. Schüssler: *Digitale Signalverarbeitung 1*, Analyse diskreketer Signale und Systeme, Springer-Verlag, Berlin, 4. Aufl. 1994.

[Sha48] C.E. Shannon: *A Mathematical Theory of Communication*, Bell Syst, Techn. J., pp. 379-423, pp. 623-656, 1948.

[Smi84] J.O. Smith, P. Gossett: *A Flexible Sampling-Rate Conversion Method*, Proc. IEEE ICASSP-84, pp. 19.4.1-19.4.4, 1984.

[Sor87] H.V. Sorensen, D.J. Jones, M.T. Heideman, C.S. Burrus: *Real-Valued Fast Fourier Transform Algorithms*, IEEE Trans. Acoust., Speech, Signal Processing, Vol. 35, No. 6, pp. 849-863, June 1987.

[Sqa88] EBU-SQAM: *Sound Quality Assessment Material*, Recordings for Subjective Tests, CompactDisc, 1988.

[Sri77] A.B. Sripad, D.L. Snyder: *A Necessary and Sufficient Condition for Quantization Errors to be Uniform and White*, IEEE Trans. ASSP, Vol. 25, pp. 442-448, Oct. 1977.

[Sta82] J. Stautner, M. Puckette: *Designing Multi-Channel Reverberators*, Computer Music Journal, Vol. 6, No. 1, pp. 56-65, 1982.

[Sti86] E. Stikvoort: *Digital Dynamic Range Compressor for Audio*, J. Audio Eng. Soc., Vol. 34, No. 1/2, pp. 3-9, 1986.

[Sti91] E.F. Stikvoort: *Digital Sampling Rate Converter with Interpolation in Continuous Time*, Proc. 90th AES Convention, Paris, Preprint No. 3018, Feb. 1991.

[Ter79] E. Terhardt: *Calculating Virtual Pitch*, Hearing Res., Vol. 1, pp. 155-182, 1979.

[Thei88] G. Theile, G. Stoll, M. Link: *Low Bit-Rate Coding of High-quality Audio Signals*, EBU Review, No. 230, pp. 158-181, August 1988.

[Tra77] Tran-Thong, B. Liu: *Error Spectrum Shaping in Narrow Band Recursive Filters*, IEEE Trans. ASSP, pp. 200-203, April 1977.

[Tsu92] K. Tsutsui, H. Suzuki, O. Shimoyoshi, M. Sonohara, K. Akagiri, R. Heddle: *ATRAC: Adaptive Transform Coding for MiniDisc*, Proc. 91st AES Convention, Preprint No. 3216, New York 1991.

[Tui93] U. Tuisel: *Multiträgerkonzepte für die digitale, terrestrische Hörrundfunk-"ubertragung*, Dissertation, TU Hamburg-Harburg, 1993.

[Van89] J. Vanderkoy, S.P. Lipshitz: *Digital Dither: Signal Processing with Resolution Far below the Least Significant Bit*, Proc. AES Int. Conf. on Audio in Digital Times, pp. 87-96, May 1989.

[Vai93] P.P. Vaidyanathan: *Multirate Systems and Filter Banks*, Prentice-Hall, Englewood Cliffs, 1993.

[Wan92] R.A. Wannamaker: *Psychoacoustically Optimal Noise Shaping*, J. Audio Eng. Soc., Vol. 40, No. 7/8, pp. 611-620, July/August 1992.

[Wid61] B. Widrow: *Statistical Analysis of Amplitude-Quantized Sampled-Data Systems*, Trans. AIEE, Pt. II, Vol. 79, pp. 555-568, Jan. 1961.

[Wir91] G.C. Wirtz: *Digital Compact Cassette: Audio Coding Technique*, Proc. 91st AES Convention, Preprint No. 3216, New York 1991.

[Zöl89] U. Zölzer: *Entwurf digitaler Filter für die Anwendung im Tonstudiobereich* , pp. 43-45, Wissenschaftliche Beiträge zur Nachrichtentechnik und Signalverarbeitung, TU Hamburg-Harburg, Juni 1989.

[Zöl90a] U. Zölzer: *A Low Roundoff Noise Digital Audio Filter*, Proc. EUSIPCO-90, Barcelona, pp. 529-532, 1990.

[Zöl90b] U. Zölzer, N.J. Fliege, M. Schönle, M. Schusdziarra: *Multirate Digital Reverberation System*, Proc. 89th AES Convention, Preprint No. 2968, Los Angeles 1990.

[Zöl92] U. Zölzer, N. Fliege: *Logarithmic Spaced Analysis Filter Bank for Multiple Loudspeaker Channels*, Proc. 93rd AES Convention, Preprint No. 3453, San Francisco 1992.

[Zöl93] U. Zölzer, B. Redmer, J. Bucholtz: *Strategies for Switching Digital Audio Filters*, Proc. 95th AES Convention, New York, Preprint No. 3714, October 1993.

[Zöl94a] U. Zölzer: *Roundoff Error Analysis of Digital Filters*, J. Audio Eng. Soc., Vol. 42, No. 4, pp. 232-244, April 1994.

[Zöl94b] U. Zölzer, T. Boltze: *Interpolation Algorithms: Theory and Application*, Proc. 97th AES Convention, San Francisco, Preprint No. 3898, November 1994.

[Zöl95] U. Zölzer, T. Boltze: *Parametric Digital Filter Structures*, Proc. 99th AES Convention, New York, Preprint No. 4099, October 1995.

[Zwi82] E. Zwicker: *Psychoakustik*, Springer-Verlag, Berlin, 1982.

[Zwi90] E. Zwicker, H. Fastl: *Psychoacoustics*, Springer-Verlag, Berlin, 1990.

Index