Chapter 1

Introduction

1.1 The z-Transform

The z-transform is used to describe linear time-invariant systems (LTI) for discrete-time signals as the Laplace transform does for the analysis of continous-time signals in LTI systems. The transform simplifies the signal analysis and makes it possible to characterize a LTI system. The z-transform for a known sequence x(n) where $-\infty \leq n \leq \infty$ is defined by

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)\ z^{-n} \tag{1.1}$$

where z is a complex variable. The z-transform of a sequence can be viewed as a unique representation of the signal sequence x(n) in the complex z-plane. Knowing the pole-zero locations, the system can be estimated with regard to stability. Herein the unit circle plays an important role.

The z-transform is an infinite power series and converges everywhere in the z-plane only if x(n) is of finite duration. The z-transform converges everywhere outside a circle of radius R_1 if the sequence x(n) is causal, what means $x(n) \neq 0$ for $0 \leq N_1 \leq n \leq \infty$. X(z) converges inside a circle of radius R_2, if the sequence x(n) is noncausal, or in a more formal expression for $-\infty \leq n \leq N_2 < 0$ is $x(n) \neq 0$. Finally, if x(n) is defined over $-\infty \leq N_1 \leq n \leq N_2 \leq \infty$, then X(z) converges between these circles.

Chapter 1

Introduction

1.1 The z-Transform

The z-transform is used to describe linear time-invariant systems (LTI) for discrete-time signals as the Laplace transform does for the analysis of continous-time signals in LTI systems. The transform simplifies the signal analysis and makes it possible to characterize a LTI system. The z-transform for a known sequence x(n) where $-\infty \leq n \leq \infty$ is defined by

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)\, z^{-n} \tag{1.1}$$

where z is a complex variable. The z-transform of a sequence can be viewed as a unique representation of the signal sequence x(n) in the complex z-plane. Knowing the pole-zero locations, the system can be estimated with regard to stability. Herein the unit circle plays an important role.

The z-transform is an infinite power series and converges everywhere in the z-plane only if x(n) is of finite duration. The z-transform converges everywhere outside a circle of radius R_1 if the sequence x(n) is causal, what means $x(n) \neq 0$ for $0 \leq N_1 \leq n \leq \infty$. X(z) converges inside a circle of radius R_2, if the sequence x(n) is noncausal, or in a more formal expression for $-\infty \leq n \leq N_2 < 0$ is $x(n) \neq 0$. Finally, if x(n) is defined over $-\infty \leq N_1 \leq n \leq N_2 \leq \infty$, then X(z) converges between these circles.

A digital system is generally described by its transfer function

$$H(z) = \frac{Y(z)}{X(z)}. \tag{1.2}$$

Furthermore, a digital system with feedback and an additive noise source as shown in Figure 1.1 is described by the signal transfer function (STF) and noise transfer function (NTF). The STF is given by

$$H_{STF}(z) = \frac{H_t(z)}{1 - H_t(z)}, \qquad Q(z) = 0 \tag{1.3}$$

and the NTF is determined by

$$H_{NTF}(z) = \frac{1}{1 - H_t(z)}, \qquad X(z) = 0 \tag{1.4}$$

Figure 1.1 shows the block diagram of a digital system with feedback.

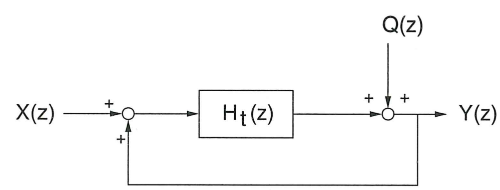

Figure 1.1: Block Diagram of a Digital System

1.2 Digital Filter Fundamentals

A digital filter is completely characterized by its difference equation. The difference equation describes the relationship between input and output.

$$y(n) = a_0 \cdot x(n) + a_1 \cdot x(n-1) + a_2 \cdot x(n-2) + \cdots + a_N \cdot x(n-N)$$

$$- b_1 \cdot y(n-1) - b_2 \cdot y(n-2) - \cdots - b_L \cdot y(n-L) \tag{1.5}$$

where x(n) is the discrete-time input signal and y(n) the output sequence. The transfer function of a digital filter is generally given by

$$H(z) = \frac{\sum_{j=0}^{L} b_j \, z^{-j}}{1 - \sum_{i=1}^{N} a_i \, z^{-i}} \tag{1.6}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}}{1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_L z^{-L}}. \tag{1.7}$$

The transfer function of the nonrecursive FIR filter is

$$H(z) = \frac{Y(z)}{X(z)} = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N} \tag{1.8}$$

A FIR filter is characterized by the impulse response written as a finite convolution sum

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{1.9}$$

where x(n) is the input signal, h(n) the impulse response of the filter and y(n) the output signal. The FIR filter is sometimes also called a convolution filter, because of the method of realization. Viewing the FIR filter from the time-domain, the system is also called moving-average filter. Obviously, the transfer function of a FIR filter in the frequency-domain is given by

$$H(z) = \sum_{n=0}^{N-1} h(n)z^{-n} \tag{1.10}$$

Due to its nonrecursive structure, FIR filters are always stable and provide linear

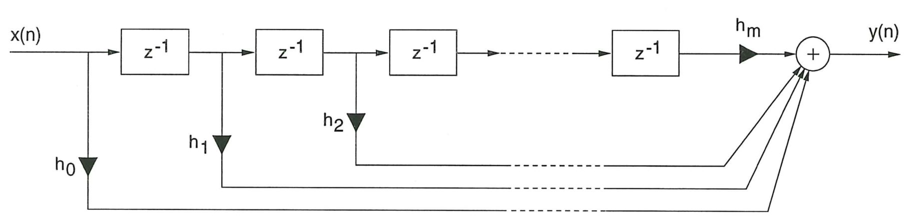

Figure 1.2: FIR Filter in Direct Form

phase if the coefficients are symmetric. They are well suited for applications in which an arbitrary magnitude response is desired and frequency distortion due to nonlinear phase must be avoided, e.g. in applications of speech processing or acoustics in general.

The performance of a FIR filter is bounded firstly by the available filter taps and secondly by finite word length effects. Let us first consider the trade-off between the width of the transition band, stopband attenuation and filter length. The filter length is a function of the allowed stopband and passband ripple (δ_s, δ_p) and the width of the transition band Δf. The peak-to-peak passband ripple in decibels is given by

$$A_p = 20 \cdot \log_{10}\left(\frac{1+\delta_p}{1-\delta_p}\right) \quad [dB] \tag{1.11}$$

and the stopband attenuation is determined by

$$A_s = 20 \cdot \log_{10} \delta_s \quad [dB] \tag{1.12}$$

where δ_p is the passband ripple and δ_s the stopband ripple.

The order of a digital lowpass filter is determined by the empirical Kaiser relation [1].

$$N = \frac{D_\infty(\delta_p \delta_s)}{\Delta f} \tag{1.13}$$

with

$$\begin{aligned} D_\infty(\delta_p\delta_s) = {} & \log_{10}\delta_s[a_1(\log_{10}\delta_p)^2 + a_2\log_{10}\delta_p + a_3] \\ & + [a_4(\log_{10}\delta_p)^2 + a_5\log_{10}\delta_p + a_6] \end{aligned} \tag{1.14}$$

and

δ_p : passband ripple

δ_s : stopband ripple

$a_1 =$ 0.005309

$a_2 =$ 0.07114

$a_3 =$ -0.4761

$a_4 =$ -0.00266

$a_5 =$ -0.5941

The transition band is normalized relative to the input sampling frequency and given by

$$\Delta f = \frac{f_s - f_p}{f_{sa}} \tag{1.15}$$

where f_p is the passband frequency, f_s the stopband frequency and f_{sa} the sampling frequency. A more useful equivalent equation to determine the filter length is

$$\mathrm{N} = \frac{-20\log_{10}\sqrt{\delta_p\delta_s} - 13}{14.6 \cdot \Delta f} + 1 \tag{1.16}$$

As equation (1.16) demonstrates, the filter length highly depends on the width of the transition band Δf. Table 1.1 shows this trade-off for a FIR lowpass filter with 100dB stopband attenuation. The higher the filter specifications are, the higher

f_p [1]	f_s [1]	F_p	F_s	Δf	N [100dB]
0.5	0.6	78125	93750	0.003125	1908
0.5	0.55	78125	85937.5	0.0015625	3814
0.5	0.53	78125	82812.5	0.0009375	(6358)
0.5	0.525	78125	82031.25	0.00078125	(7628)

Table 1.1: Filter Specifications, with Filter Order N

the filter order. In a multistage filter structure, every single stage has its own specifications. Figure 1.7 shows a multistage filter cascade and (1.16) becomes

$$\mathrm{N_i} = \frac{-20\log_{10}\sqrt{\delta_{p,i}\delta_{s,i}} - 13}{14.6 \cdot \Delta f_i} + 1 \tag{1.17}$$

for the ith filter stage. The transition band becomes

$$\Delta f_i = \frac{f_{s,i} - f_{p,i}}{f_{sa,i}} \tag{1.18}$$

where $f_{s,i}$ is the stopband frequency, $f_{p,i}$ the passband frequency and $f_{sa,i}$ the input sampling rate for stage i, respectively.

A digital filter is furthermore characterized by the required computation in multiplications per second [2]

$$\mathrm{R} = \frac{\mathrm{N} \cdot f_{sa}}{2\mathrm{D}} \quad \text{[multiplications/sample]} \tag{1.19}$$

where f_{sa} denotes the sampling frequency, N the filter length and D the decimation ratio. The above presented one-stage FIR filter (N=3814, f_{sa}=5MHz) requires 297 968 750 multiplications per second.

1.3 Decimation Filters

In many digital signal processing applications, the sampling rate has to be reduced. The process of decimation, obtaining a signal with a lower sampling rate, is also called sampling rate conversion. In applications using oversampling techniques, decimation will furthermore reduce the quantization noise. Figure 1.3 shows the block diagram of a single-stage decimator to illustrate the decimation process. In an efficient architecture, the decimator is located before the coefficient multiplier as depicted in Figure 1.4. Hence, the multiplication is performed at the reduced sampling rate of f_s/D. Since having symmetric coefficients, further savings in complexity are possible. Figure 1.6 shows this approach.
Decimation by D means that every Dth output sample is required. This means for the filter realization, that only every Dth sample need to be computed. In order to avoid aliasing, the signal must be band-limeted to $\Omega = \pi/\mathrm{D}$. Figure 1.5 shows the magnitude spectrum of the band-limeted signal. The original spectrum is periodic in $\Omega = 2\pi$. The downsampled signal is described by

$$\mathrm{Y}(\mathrm{e}^{\mathrm{j D}\Omega}) = \frac{1}{\mathrm{D}} \sum_{\mathrm{i}=0}^{\mathrm{D}-1} \mathrm{X}(\mathrm{e}^{\mathrm{j}(\Omega - 2\pi \mathrm{i}/\mathrm{D})}) \tag{1.20}$$

The resulting downsampled spectrum is periodical in $\Omega = 2\pi/\mathrm{D}$. This can be regarded as a new sampling rate of Ω'=D$\cdot\Omega$. In Figure 1.5 is a new axis with Ω' depicted.
The designed decimation filter must have two major properties. First, it must fulfill the demands in attenuating the out-of-band signals and the modulator quantization noise. Second, the noise of the filter itself must be sufficiently low. The noise caused within the filter is essentially coefficient quantization noise and roundoff noise.

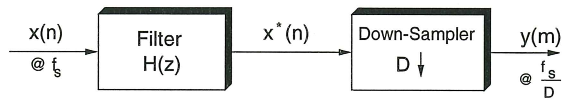

Figure 1.3: Single Stage Digital Filter and D to 1 Decimator

The downsampling process is described with the following equations. The sequence at the output of the filter is given by (ref. Figure 1.3)

$$x^*(n) = \sum_{i=0}^{N} h(i)x(n-i) \tag{1.21}$$

The final decimated signal is hence

$$y(m) = x^*(mD) \tag{1.22}$$

with $n = m \cdot D$. The output, depending on the input signal, can be written as

$$y(m) = \sum_{i=0}^{N} h(i)x(mD-i) \tag{1.23}$$

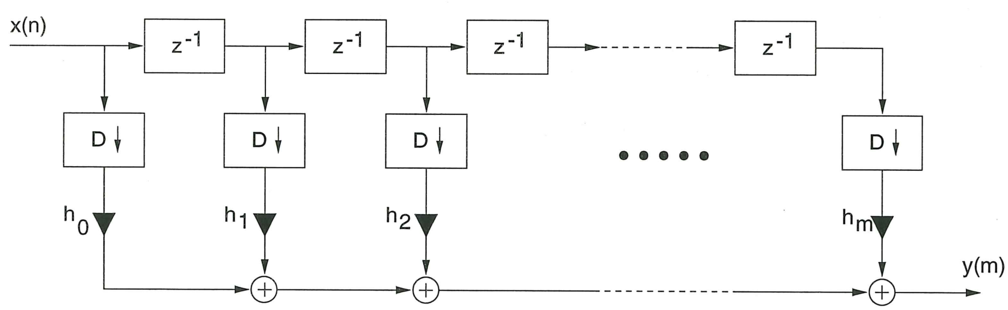

Figure 1.4: Direct Form of a FIR Filter with Decimation

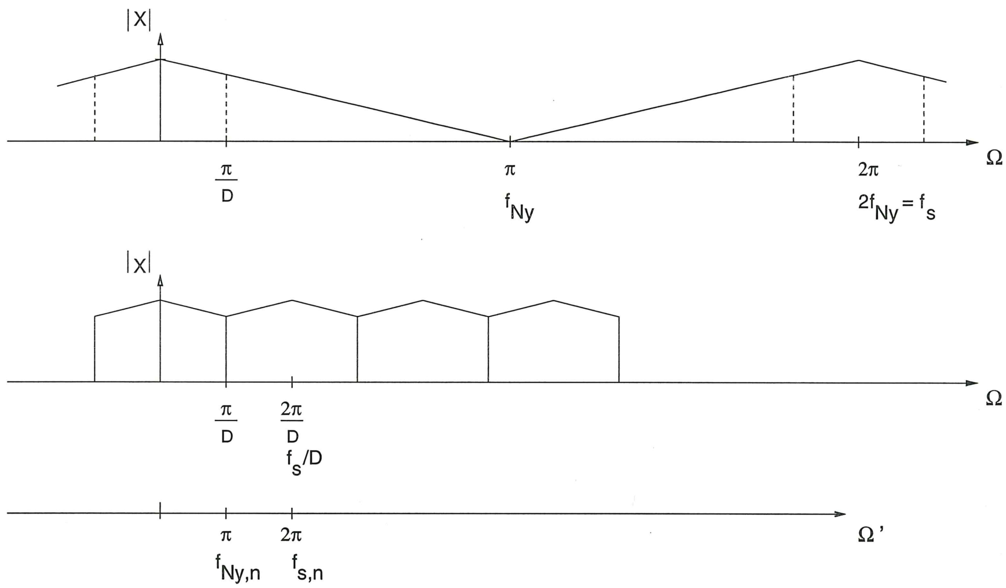

Figure 1.5: Magnitude Spectrum in the Decimation Process

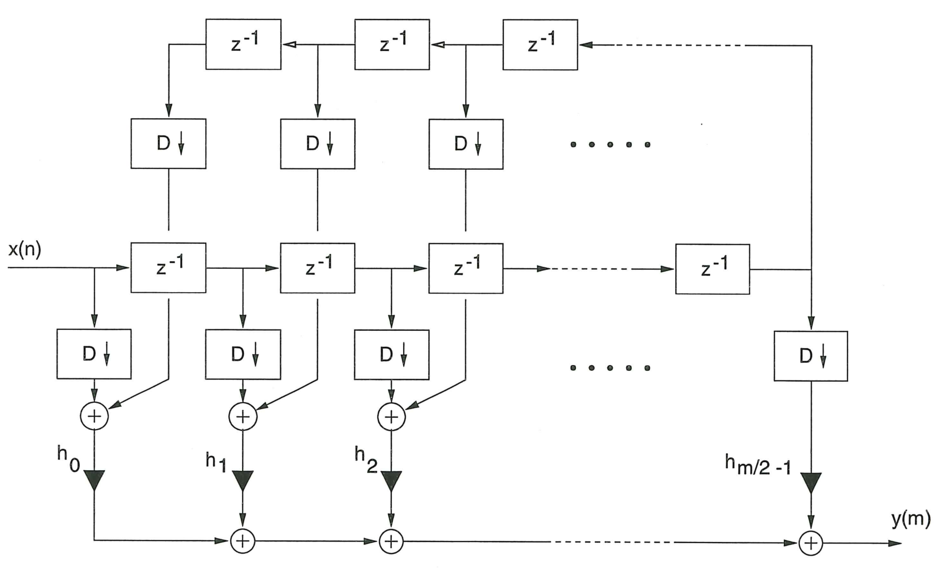

Figure 1.6: Direct Form of a FIR Filter with Symmetric Coefficients

1.3.1 Multistage Decimation Filters

When large downsampling ratios must be realized, the filter requirements can be great. Large oversampling ratios and large decimation ratios cause a very narrow transition band with regard to the overall frequency range $[0; f_{sa}]$. This leads to a prohibitively large filter length, as mentioned before. Moreover, in a single-stage filter structure, a large word length is required to avoid quantization noise and roundoff errors. In a multistage architecture sampling frequency is decimated in steps. Every single stage has its own specifications. Due to the lower input sampling rate at the intermediate stages, Δf_i is not as narrow as in the one-stage case. Figure 1.7 shows a two-stage filter structure. The frequency bands are subdivided in

passband:

$$0 \leq \mathrm{f} \leq \mathrm{f_{p,i}} \tag{1.24}$$

and transition band:

$$\mathrm{f_{p,i}} \leq \mathrm{f} \leq \mathrm{f_{s,i}} = \frac{\mathrm{f_i}}{2} \tag{1.25}$$

where i is the stage index.

At the output of the ith stage, the sampling frequency becomes

$$\mathrm{f_i} = \frac{\mathrm{f_{i-1}}}{\mathrm{D_i}}. \tag{1.26}$$

Due to an overall decimation of

$$\mathrm{D} = \prod \mathrm{D_i}, \tag{1.27}$$

the final output sampling frequency will be

$$\mathrm{f_o} = \frac{\mathrm{f_s}}{\prod \mathrm{D_i}} = \frac{\mathrm{f_s}}{\mathrm{D}}. \tag{1.28}$$

Figure 1.8 illustrates the steps of decimation. The frequency breakpoints in the final stage are the same as in the one-stage case. The only diffenence is that the

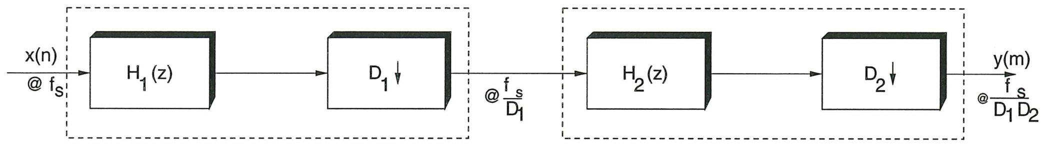

Figure 1.7: Cascaded Decimation Filter

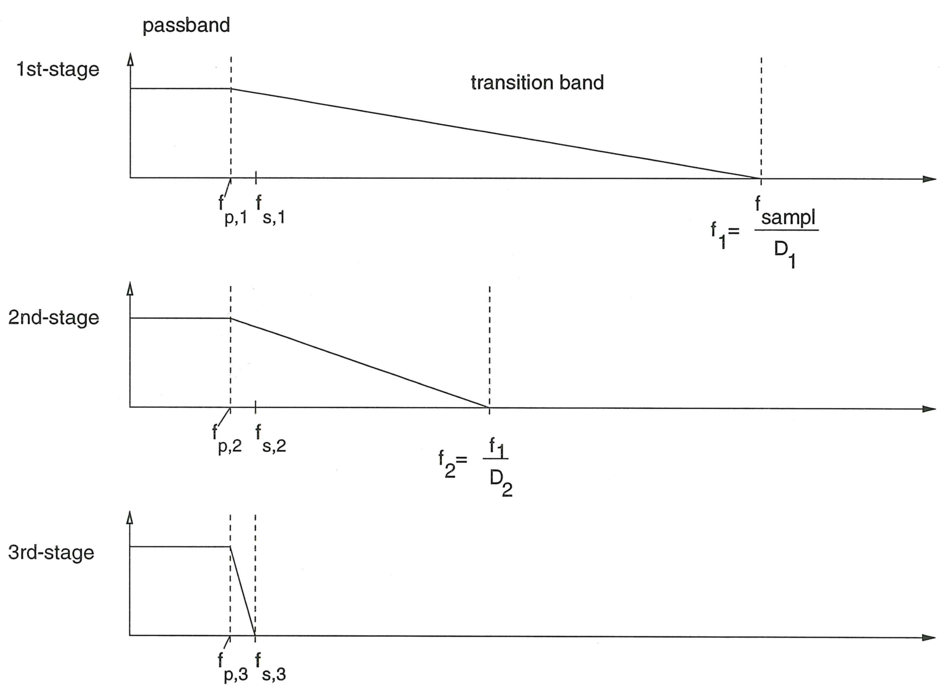

Figure 1.8: Frequency Decimation in a Cascaded Filter Structure

final stage has a lower input frequency, which means a smaller filter order.

The most important advantages of a cascaded FIR structure are: [2]

- Reduced overall filter order and therefore reduced storage requirements
- Significantly reduced computation to implement the architecture
- Reduced finite word length effects (roundoff noise, bit-sensitivity)

1.4 Comb Filters

If we realize a decimation filter cascade, the first stage should be suitable for the "bulk" of decimation. The first stage operates at a high frequency while good silicon utilization should be achieved. Especially for this purpose comb filters (sometimes called sinc^K filters, where K is the order of the comb filter) are well suited. They have good properties for decimation purposes and a simple structure [24]. The advantages of the comb filter compared with the FIR filter are:

- No multipliers are required
- A simple structure can be designed
- No coefficient storage
- The architecture is independent to the decimation ratio
- The early decimation leads to lesser dynamic power consumption for the following stages

The comb can be treated as notch filter with zeros at $\omega_k = \frac{2\pi \cdot k}{D}$ for k = 1, 2, ..., D/2. The comb filter is a recursive filter whose coefficients b_j are equal to one. One obtains a conditionally stable liner phase filter with length D. The transfer function is

$$H(z) = \sum_{n=0}^{D-1} z^{-n} = \frac{1 - z^{-D}}{1 - z^{-1}} \tag{1.29}$$

The transfer function of the comb filter is derived from the moving average filter by rewriting it in a recursive form. In order to derive equation (1.29), let us consider the moving average process

$$y(n) = \sum_{i=0}^{D-1} x(n-i) \tag{1.30}$$

and rewrite it as

$$y(n) = \sum_{i=-1}^{D-2} x(n-i) + x(n) - x(n-D) \tag{1.31}$$

$$= y(n-1) + x(n) - x(n-D). \tag{1.32}$$

Obviously, equation (1.29) and (1.32) describe the same system. This expression leads to equation (1.29). Every new output sample can be determined by adding the previous output sample to the new input value and subtracting the input value, that occured D samples ago. We obtain a significant reduction in computation time with the recursive structure [32]. The computation of a new output sample requires $t_c = \frac{1}{2 \cdot f_{sa}}$ seconds. The conventional moving average filter consisting of D delay units needs a computation time of $t_c = \frac{1}{D \cdot f_{sa}}$ seconds for every new output sample, where f_{sa} is the input sampling frequency.

Due to its recursive structure, the comb filter is only conditionally stable. Furthermore, a DC or a low frequency input signal will cause initial values because of the IIR part. Figure 1.9 shows the magnitude response of a comb filter with D=16. The magnitude converges to zero at multiples of $2\pi/D$. That is the most important property for using comb filters in a decimation filter. If we choose the filter length equal to the decimation ratio, the attenuation in the aliased bands can be sufficiently high. Of course the attenuation depends on the filter order. The *aliased bands* are those frequency bands which will be folded back after decimation. This property is sometimes called *'natural anti-aliasing'*. Figure 1.21 illustrates the effect of comb filter anti-aliasing. Another reason that makes comb filter interesting is that no multipliers are required. The disadvantage, on the other hand, is the

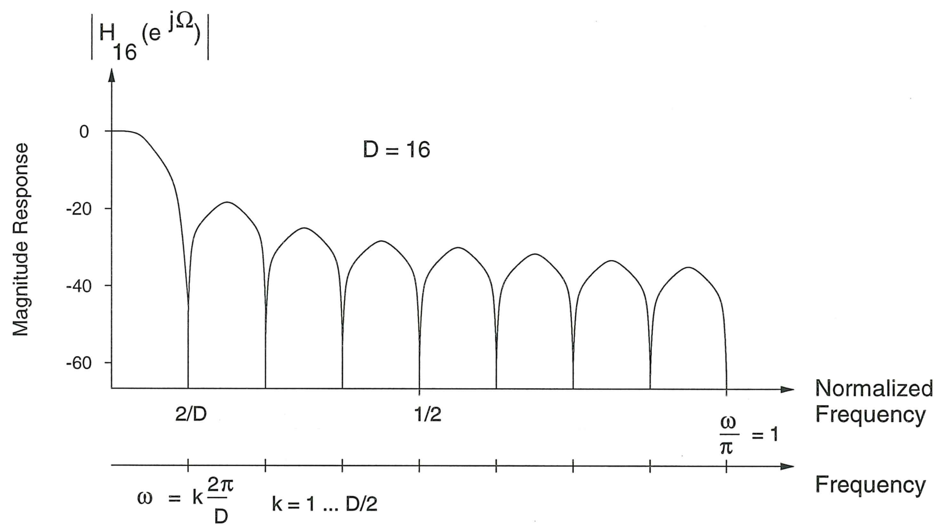

Figure 1.9: Magnitude Response of a Comb Filter with D=16

relatively low attenuation. We are not able to achieve sufficient alias rejection with one comb filter. A cascade of comb filters is necessary. The disatvantage of the multiple-comb filter is the inherent passband droop. Figure 1.11 makes this clear. The passband droop increases with the order and slightly with the decimation ratio, as can be seen in Table 6.3. Considering the overall perfomance, we cannot neglect the loss in magnitude in the passband section. A subsequent FIR or IIR filter stage is required to correct this deviation. Those filters are therefore often called compensation filters [27].

The system function of the conventional comb filter in the z-domain is

$$H(z) = \frac{1}{D}\sum_{k=0}^{D-1} z^{-k} = \frac{1}{D} \cdot \frac{1 - z^{-D}}{1 - z^{-1}} \tag{1.33}$$

and in the frequency domain

$$H(e^{j\Omega}) = \frac{1}{D} \cdot \frac{\sin(\Omega \cdot D/2)}{\sin(\Omega/2)} \tag{1.34}$$

with

$$\Omega = \frac{2\pi \cdot f}{f_{sa}} \tag{1.35}$$

Figure 1.10 shows the block diagram of a comb filter. In order to reach a more efficient implementation, the basic structure can be redrawn using the commutative rule, as shown in Figure 1.10. The differentiation $(1 - z^{-1})$ is now performed at a

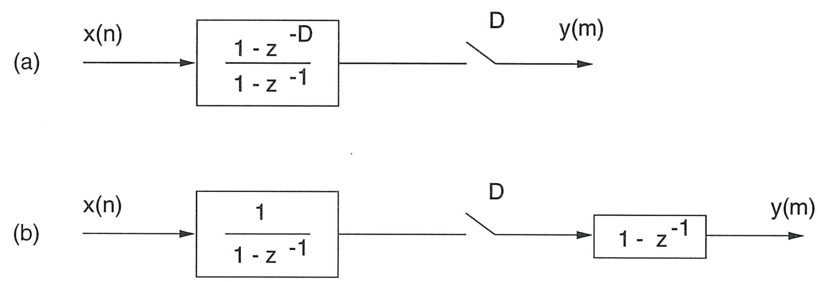

Figure 1.10: Comb Filter with Decimation

lower sampling rate. With this modification, we are able to reduce the number of registers and the processing rate [26].

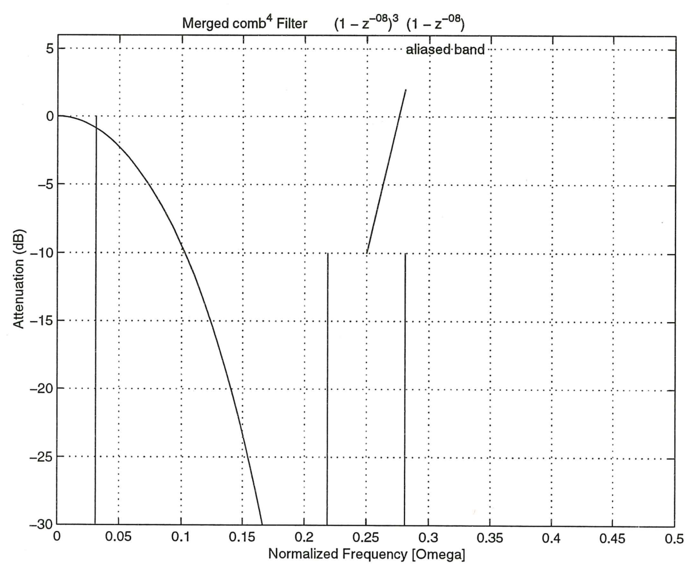

Figure 1.11: Passband Droop

1.4.1 Cascaded Comb Filters

With a single comb filter, sufficient stopband attenuation is not achievable. Therefore, a cascaded comb structure is often applied, as mentioned before. The cascaded integrators are usually followed by the intermediate downsampler and finally by the FIR section. Figure 1.12 shows this approach. Equation (1.36) is the transfer function of the cascaded comb filter.

$$H(z) = \left[\frac{1}{D} \cdot \frac{1 - z^{-D}}{1 - z^{-1}}\right]^K \tag{1.36}$$

The transfer function in the frequency domain is, respectively

$$H(e^{j\Omega}) = \left[\frac{1}{D} \cdot \frac{\sin(\Omega \cdot D/2)}{\sin(\Omega/2)}\right]^K \tag{1.37}$$

with

$$\Omega = \frac{2\pi \cdot f}{f_{sa}} \tag{1.38}$$

where f_{sa} denotes the sampling frequency. Equation (1.37) describes a lowpass filter with linear phase. Cascading must be continued until the desired stopband attenuation at $\Omega = \frac{2}{D} - \Omega_p$ is reached. Figure 1.13 shows the magnitude responses

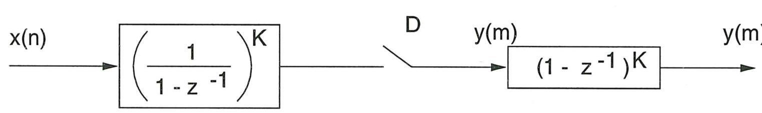

Figure 1.12: Cascaded Comb Filter with Decimation

of a two stage and five stage comb filter. The worst case aliasing will occur at $\omega_{fb} = 2/D - \omega_p$. Hence, the worst alias attenuation is given by

$$A_{alias}(\omega_{fb}) = 20 \cdot \log \left(\frac{\sin(\omega_{fb} \cdot D/2)}{D \cdot \sin(\omega_{fb}/2)}\right)^K_{|f_p} \tag{1.39}$$

where f_p is the passband frequency.

With (1.39) we can determine the maximum alias rejection as a function of the

D	K = 1	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	Ω_p
2	26dB	52dB	79dB	105dB	131dB	157dB	183dB	0.96875π
4	23dB	46dB	68dB	91dB	114dB	137dB	159dB	0.46875π
8	17dB	34dB	51dB	68dB	85dB	102dB	119dB	0.21875π
16	10dB	21dB	31dB	42dB	52dB	63dB	73dB	0.09375π

Table 1.2: Alias Attenuation for the Comb Cascade

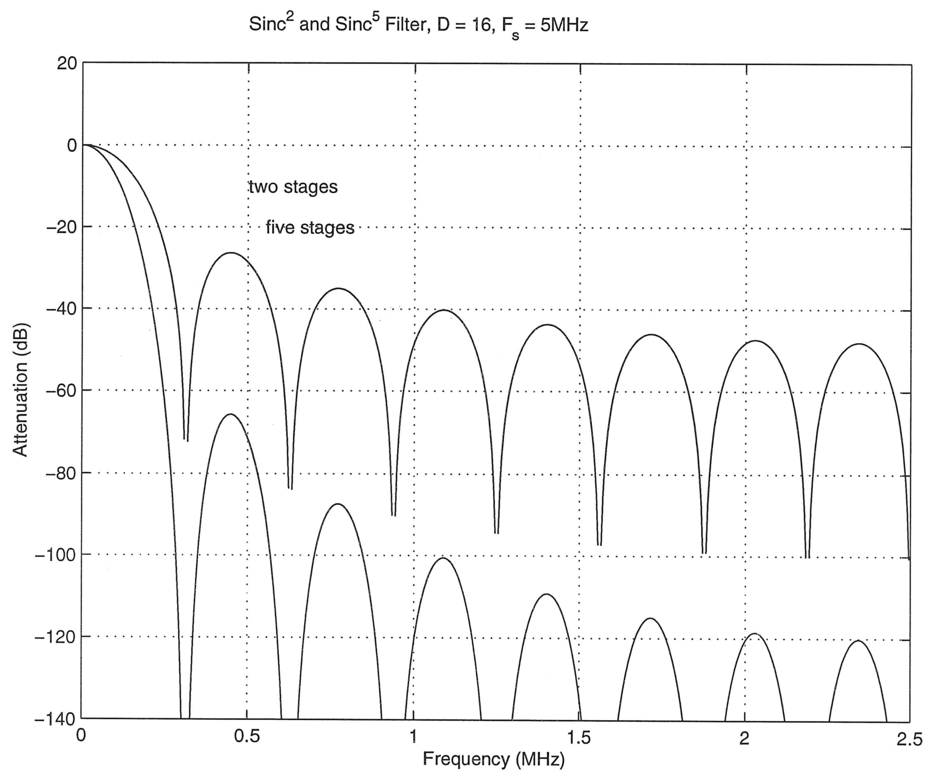

Figure 1.13: Magnitude Response of a Cascaded Comb Filter with Decimation

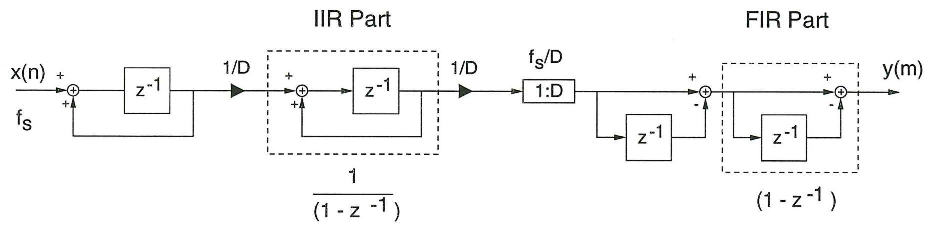

Figure 1.14: Block Diagram of the two Stage Comb Filter

decimation ratio D, the filter order K and the passband frequency. Unfortunately, the passband droop increases with the number of comb filters K. Figure 1.14 shows the realization of a 2nd order comb filter cascade. Applying the commutative rule, the structure is split into a FIR and IIR part, corresponding to Figure 1.12.

Within the class of comb filters, a modification of the structure should be mentioned. Basically, this is the conventional merged filter structure with an additional delay in the forward path [26]. Figure 1.15 shows this approach. The transfer function for a single stage is given by

$$\mathrm{H(z)} = \frac{1 - \mathrm{z}^{-\mathrm{D}}}{1 - \mathrm{z}^{-1}} + \mathrm{z}^{-\mathrm{D}} \qquad = \qquad \frac{1 - \mathrm{z}^{-(\mathrm{D}+1)}}{1 - \mathrm{z}^{-1}} \tag{1.40}$$

The transfer function for a Kth-order cascade is given by

$$\mathrm{H(z)} = \frac{1 - \mathrm{z}^{-(\mathrm{D}+1)}}{1 - \mathrm{z}^{-1}} \cdot \left[\frac{1 - \mathrm{z}^{-\mathrm{D}}}{1 - \mathrm{z}^{-1}}\right]^{\mathrm{K}-1} . \tag{1.41}$$

This is basically a conventional (K-1)th-order comb filter supperposed with the modified comb filter (1.40) with single order. Figure 1.16 shows the frequency response for an example. An additional narrow notch is inserted on the left of the $\frac{k \cdot 2}{D}$ notch. The shown example is a length-(D+1) comb filter with an order of K=6 and a decimation ratio of 8. The attenuation to the left of the notch at Ω=2/D is slightly increased. In some cases, the specifications can be achieved by a lower order. The length-(D+1) filter is always single order. The changes in the overall frequency behavior depend strongly on the order K.

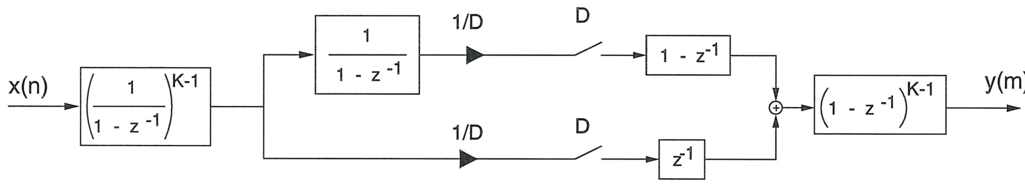

Figure 1.15: Block Diagram of the length-(D+1) Comb Filter K*th* order

The transfer function for the example shown in Figure 1.16 is

$$H(z) = \frac{(1 - z^{-8})^5 (1 - z^{-9})}{(1 - z^{-1})^6} \tag{1.42}$$

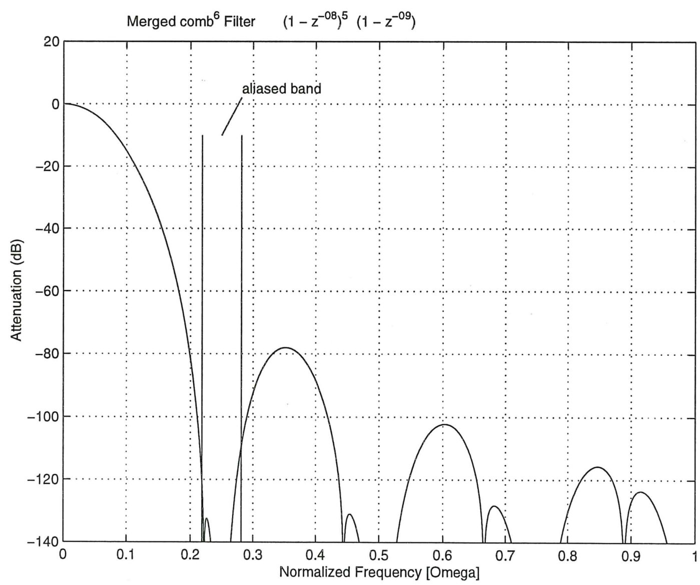

Figure 1.16: Frequency Response of the length-(D+1) Comb Filter 6*th* order

Multistage filter design using comb filters means usually an architecture of one or more comb filters followed by a FIR or IIR compensation filter. Figure

6.3 shows the block diagram of a comb - FIR filter cascade with specifications for the design example. For audio applications, where a linear phase is required, FIR compensators are mandatory.

1.4.2 Sharpened Comb Filter

In recent years a modified comb structure denoted as sharpened comb filter has also been used in decimation filters. The properties are described in [29], [30]. With the sharpened comb filter, we are able to reduce the passband droop and can achieve significant alias rejection. The sharpened comb filter requires three copies of the conventional comb filter, one adder, two scaling multiplier and one delay unit. The overall transfer function becomes $H_{ov}(z)=H^2(z)\cdot[3 - 2\cdot H(z)]$, see (1.44). The used building blocks H(z) have linear phase, hence the overall filter will also have linear phase. The hardware requirements for the implementation increase compared to the conventional comb filter. Figure 1.18 shows the topology of the filter. Estimating the requirements in silicon versus perfomance is part of a later investigation. The transfer function of the sharpened comb filter in the frequency domain is [30]

$$H(e^{j\Omega}) = 3 \cdot \left(\frac{\sin\frac{\Omega D}{2}}{D \cdot \sin\frac{\Omega}{2}}\right)^{2K} - 2 \cdot \left(\frac{\sin\frac{\Omega D}{2}}{D \cdot \sin\frac{\Omega}{2}}\right)^{3K} \tag{1.43}$$

where Ω is the frequency, D the decimation ratio and K the order. With the sharpened comb filter, we are able to achieve about 120dB attenuation in the alias band with D=8 and K=4. Figure 1.17 shows the frequency response for this case. Assuming $H(z) = \left[\frac{1}{D} \cdot \frac{1-z^{-D}}{1-z^{-1}}\right]^K$ and a delay unit $z^{-(D-1)}$, the overall transfer function in the z-domain is given by

$$H_{ov}(z) = 3 \cdot \left[\frac{1}{D} \cdot \frac{1-z^{-D}}{1-z^{-1}}\right]^{2K} - 2 \cdot \left[\frac{1}{D} \cdot \frac{1-z^{-D}}{1-z^{-1}}\right]^{3K} \tag{1.44}$$

If we consequently apply the commutative rule, the decimation is performed in the intermediate stage. The following stages are running at a lower clock rate, which leads to lesser power consumption. Figure 1.19 shows the sharpened comb structure.

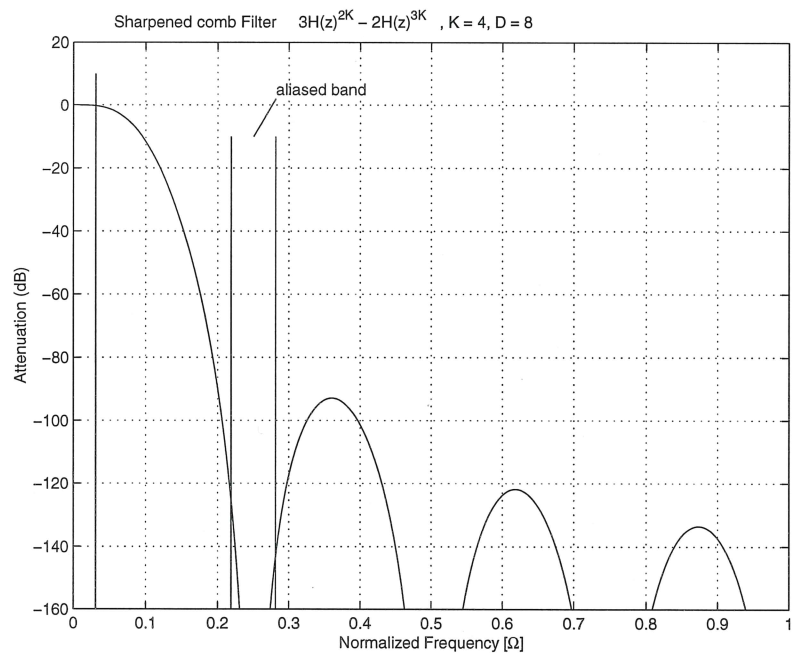

Figure 1.17: Sharpened Comb Filter K = 4, D = 8

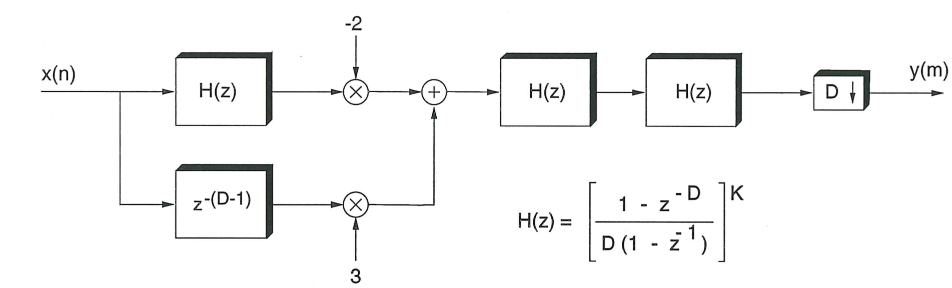

Figure 1.18: Block Diagram of the Sharpened Comb Filter

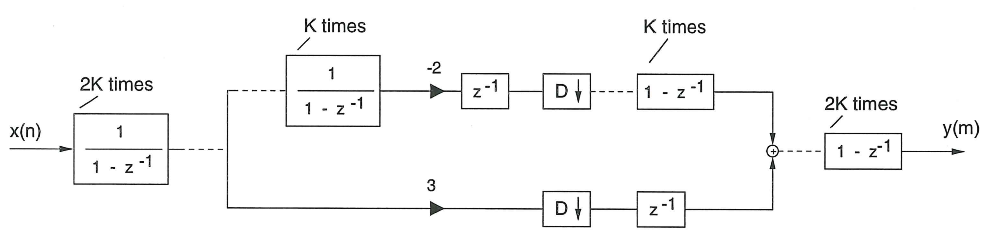

Figure 1.19: Redrawn Block Diagram of the Sharpened Comb Filter

1.5 Anti-Aliasing

Sigma-Delta modulators belong to the class of oversampling A/D converters. Compared with Nyquist-rate ADCs, the sampling frequency is many times higher than the Nyquist-rate. This leads to some advantages concerning aliasing effects and quantization noise.

In the process of decimation, care must be taken to prevent unwanted signals in the desired signal band. The anti-alias filter is a lowpass filter with a flat frequency response, which sufficiently attenuates the frequencies above the Nyquist frequency. To estimate the noise in the passband, we must consider the amount of alias distortion. If we decimate simply by keeping every Nth sample, an undesired signal with a foldering frequency of $\frac{f_s}{2D}$ will appear in the band of interest. Therefore, we must first bound the bandwith of the incoming signal to $f_c = \frac{f_s}{2D}$. The overall filter process consists of a digital lowpass filter and a following decimation by D. Figure 1.3 illustrates the decimation process. Considering the frequency domain only, the range of interest is $|\,\omega\,| \leq \pi/D$. The lowpass filter must meet the following condition

$$\mathbf{H}(\omega) = \begin{cases} 1 & |\,\omega\,| \leq \pi/\mathrm{D} \\ 0 & \text{otherwise} \end{cases}$$

where π refers to the Nyquist frequency. Figure 1.20 illustrates the principle of the anti-alias filter. A narrow transition band and high stopband attenuation leads to large filter lengths. An analog anti-aliasing filter used in Nyquist-rate A/D converters probably can not meet the specifications.

1.5.1 Alias Rejection using Comb Filters

Figure 1.21 shows the frequency response for a signgle-stage comb filter. The aliased band is bounded by $\frac{k \cdot f_s}{D} - f_p < f_n < \frac{k \cdot f_s}{D} + f_p$. The remaining frequency range is often denoted as a *'don't care'* region.

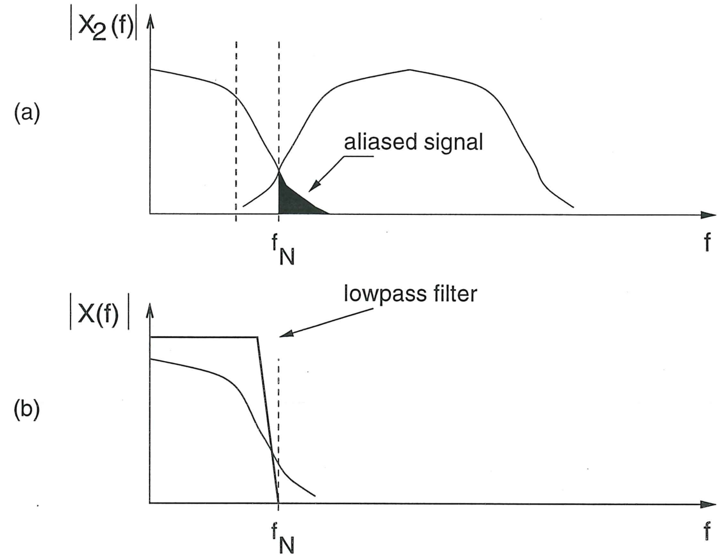

Figure 1.20: Alias Prevention in Oversampled A/D Converters, sampled signal without (a) and with (b) band-limitation

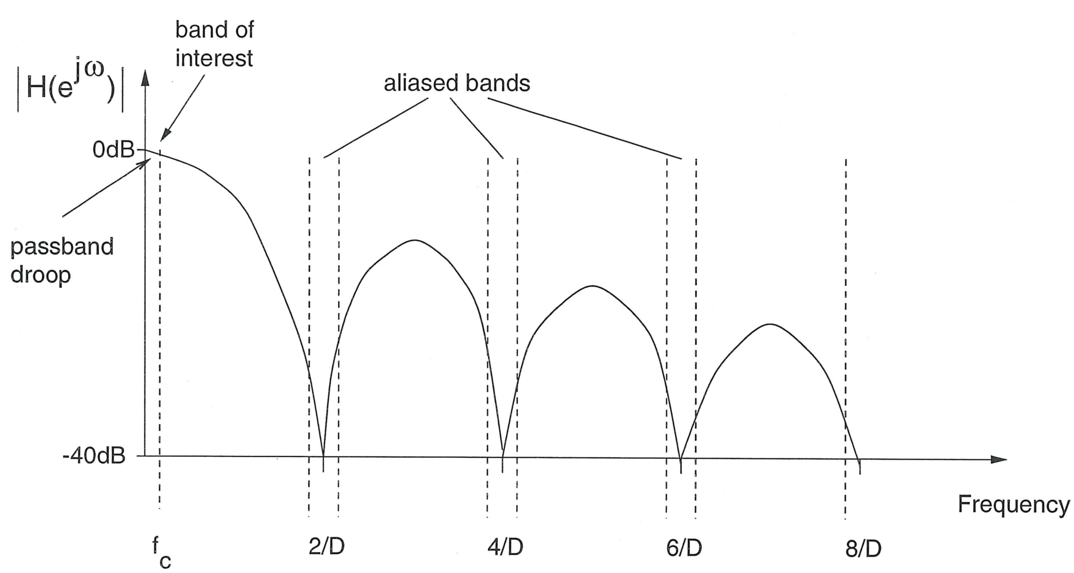

Figure 1.21: Alias Rejection with Comb Filters

1.6 Finite Word-Length Effects

The coefficients of the difference equation, which represent the digital filter, are subject to amplitude quantization errors. The applied design algorithm yields very exact results which are very close to the desired impulse response. Every digital filter must be implemented using a fixed number of bits to represent the values. This concerns the filter coefficients, the input signal and output signal. Finite word length effects cause coefficient quantization noise, roundoff noise and overflow oscillations. Due to the coefficient quantization error, the desired frequency response may not be achieved. The roundoff noise is a low-level noise that occurs since the result of each calculation is truncated or rounded within a digital filter.

1.6.1 Number Representation

The binary number represents a fraction, an integer or a mixed number. The rightmost bit is called the least significant bit (LSB) while the leftmost bit is called the most significant bit (MSB). In the most applications a sign bit a_0 is neccessary. Available are various binary code representations such as one's complement, two's complement, sign-magnitude or offset binary. In digital signal processing, a number is usually represented in a signed two's complement digital format as shown in (1.45).

$$\mathrm{y} = -\mathrm{a}_0 + \mathrm{a}_{-1} \cdot 2^{-1} + \mathrm{a}_{-2} \cdot 2^{-2} + \cdots + \mathrm{a}_{\mathrm{k}-1} \cdot 2^{-(\mathrm{k}-1)} \tag{1.45}$$

where k is the number of bits and a_0 the sign bit.
Considering (1.45), the quantization step is obviously

$$\Delta \mathrm{Q} = 2^{-(\mathrm{k}-1)} \tag{1.46}$$

The numbers representable by these k bits are called quantization levels and the gap between two of these levels is called the quantization step ΔQ.

1.6.2 Roundoff Noise

Realizing the filter in a convolution sum (1.47) means that roundoff noise will occure after each multiplication. Since we implement the filter with a multiply and accumulate unit, where no intermediate quantization must be done, just the final output has to be limited. The roundoff noise power is given by (1.48), where ΔQ is the quantization step. This error, typical for digital systems, can be considered as white Gaussian noise.

$$\mathrm{y(n)} = \sum_{\mathrm{k=0}}^{\mathrm{N-1}} \mathrm{h(k)x(n-k)} \tag{1.47}$$

$$\mathrm{E_Q} = \frac{\Delta \mathrm{Q}^2}{12} \tag{1.48}$$

with $\Delta Q = 2^{-(k-1)}$ and k bits used to represent the number.

1.6.3 Truncation and Rounding Errors

The limitation to a fixed number of bits is done by truncating or rounding the exact value y. We obtain a quantizated value y_q:

$$y_q = \quad 0.1101010110$$

from

$$y = \quad 0.11010101101011100111011$$

The exact value is recomposed as follows

$$\mathrm{y} = \mathrm{y_q} + \Delta \mathrm{y} \tag{1.49}$$

where Δy is the truncated part. Figure 1.22 shows the differences in quantization by rounding and truncating. We can determine the maximum error for both cases from this figure. The rounding error lies between

$$-\frac{1}{2} \cdot (2^{-\mathrm{m}} - 2^{\mathrm{m_q}}) \leq \mathrm{E_r} \leq \frac{1}{2} \cdot (2^{-\mathrm{m}} - 2^{\mathrm{m_q}}) \tag{1.50}$$

and the truncation error falls in the range of

$$-(2^{-m} - 2^{m_q}) \leq E_t \leq 0. \tag{1.51}$$

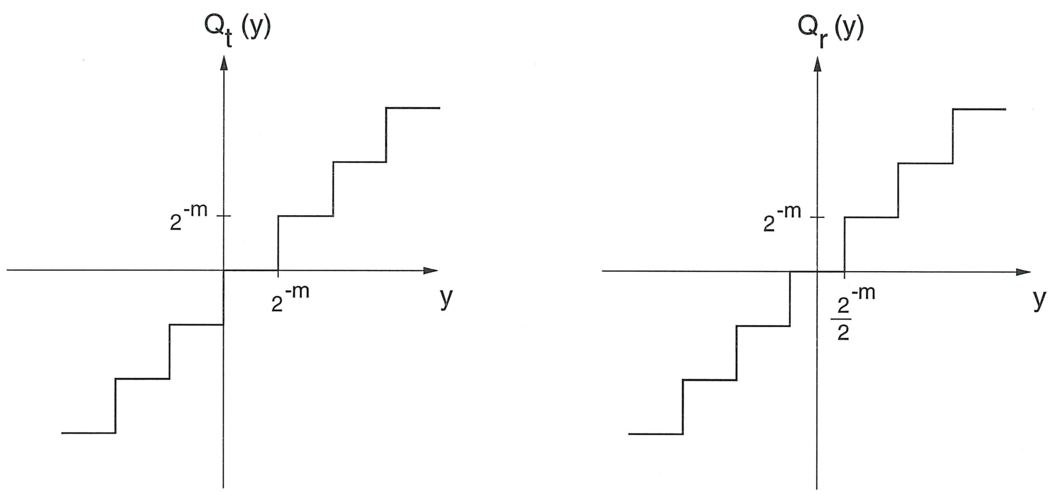

Figure 1.22: Truncating and Rounding

1.7 PTV(D)-Filter

Subject of recent investigations is a FIR filter realization using radix-r number representation and periodically time-varying (PTV) coefficients. The PTV or PTVD (PTV with downsampler) filter are different in the way of their coefficient encoding and their modified topology.

This relatively new approach for multiplier-free FIR filter implementations is described in [34],[35] and [36]. The PTV filter is based on the well known direct form FIR filter structure. Conspicuous is the representation of the coefficients in a set of signed digit numbers (SD). The coefficients are restricted to ternary values $\{\pm1, 0\}$ (radix-2/-3) or a quinary set $\{\pm2, \pm1, 0\}$ (radix-4/-5). Since the required scaling factors are represented in powers-of-two, no multipliers are needed. The radix-3 number system is the traditional signed binary representation and can be implemented with simple add/subtract operations. The radix-4 representation contains also ±2 (power-of-two) and can be implemented in add/subtarct and a one bit shift operation. This leads to a multiplier-free implementation of the target filter. In fact, the FIR-PTV filter is basically no new filter. We need to design a filter with target coefficients using traditional design methods. Really new is only the decoding of the coefficients, which ultimately mandates a modified structure. This architecture is suitable for multi-bit digital inputs like they arise in multistage structures.

In order to realize the FIR-PTV filter, we first must encode the known target coefficients h(n) to the radix-r signed-digit number representation. Available is a coefficient set of $\{0, \pm1\}$ if r = 2,3 or $\{0, \pm1, \pm2\}$ if r = 4,5. Let us assume a target impulse response $h_t(n)$ [36]

$$h(n) = C_{out} \sum_{i=0}^{N-1} (K_{in})^i \sum_{j=0}^{N-1} (K_{out})^j c_{nN-j-i}(N-j) \qquad (1.52)$$

In [35] and [36], the FIR-PTV filter is described in detail with an up- and down-sampling unit. In applications where oversampled A/D converters are used, usually only the downsamlpling unit is of interest. Figure 1.23 shows the block diagram of the FIR-PTV filter. The filter output block is scaled by a harmonic function S(n) with period D.

The target filter coefficients can then be expressed in relation to the PTV filter coefficients, by

$$\mathrm{h(n)} = \mathrm{C_0} \sum_{\mathrm{j=0}}^{\mathrm{D-1}} \mathrm{K^j_{out} c_{m+D-j}(-j)} \tag{1.53}$$

The filter length is given by

$$\mathrm{L} = \eta(\mathrm{N} - 1 + \mathrm{D}) + 1 \tag{1.54}$$

where N is the length of the target filter and D the decimation ratio. A way to make the radix-r representation more efficient is to scale the coefficients by a factor (power-of-two). The available range is efficiently used. This is very useful for small coefficients, where the largest coefficient is smaller than $\simeq \frac{1}{8}$, $\simeq \frac{1}{4}$ or $\simeq \frac{1}{2}$. To nullify this process, the output needs to be divided by the scaling factor. Figure 1.24 shows this process in principle.

Ghanekar and Tantaratana presented in their publications an algorithm to compute the radix-r represented coefficients [34], [35] and [36].

```
h(n) = h_t(n-2)
f = h(n)(r')N/2
do j=0:N-1
    do i=0:N-1
        d'_{N-1-i+Nj} = (f/[3^{N-1-i}(r')^{-Nj}])_{roundet}
        c_{nN-i-j}(N-j) = d'_{N-1-i+Nj}
        f=f-d'_{N-1-i+Nj} · 3^{N-1-i}(r')^{-Nj}
    end
end
```

With a word length of q=8, we can already achieve results which are very close to the desired filter.

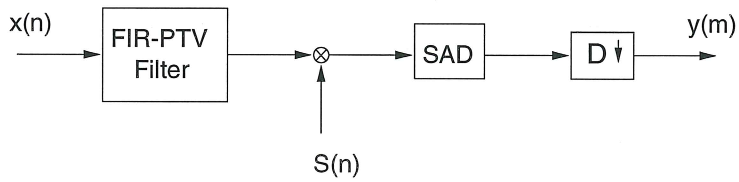

Figure 1.23: Block Diagram of the FIR-PTV Filter

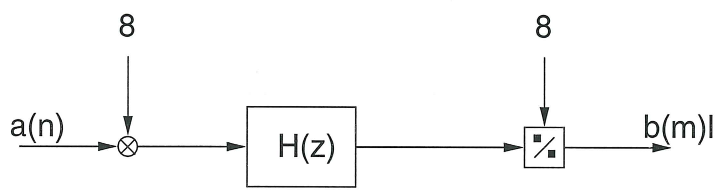

Figure 1.24: Scaling to exploit the digit range

1.7.1 Radix-r Signed Digit Number Representation

The representation of a radix-r encoded number can be expressed as

$$\mathrm{a} = \mathrm{C} \cdot \sum_{\mathrm{i}=0}^{\mathrm{q}-1} \mathrm{d_i} \cdot \mathrm{r}^{-\mathrm{i}} \tag{1.55}$$

where r is the radix and $d_i \in \{-I, \cdots, -1, 0, 1, \cdots, I\}$ are the coefficients.

with
q : word length of the representation
$C = \frac{1-r^{-1}}{I}$
d_i : signed digits

and

$$\mathbf{I} = \begin{cases} 1, & \text{for radix-2/-3 representation} \\ 2, & \text{for radix-4/-5 representation} \end{cases} \tag{1.56}$$

The PTV coefficients derived from the target coefficients need to be encoded in a binary digit representation $\{1, 0\}$. We require two bits for the radix-2/-3 and

I	Coefficient Set	radix	C_0	K_{in}	K_{out}	Bit-Precision
1	$\{\pm1, 0\}$	2	2^{-1}	1	2^{-1}	D + 1
1	$\{\pm1, 0\}$	3	$2^{-1} + 2^{-3}$	1	$2^{-2} + 2^{-3}$	1.415·D + 0.263
2	$\{\pm2, \pm1, 0\}$	4	2^{-1}	1	2^{-2}	2·D
2	$\{\pm2, \pm1, 0\}$	5	$2^{-1} - 2^{-4}$	1	$2^{-2} - 2^{-5}$	2.193·D

Table 1.3: Parameters and Bit-Precision for the PTV Decimator

three bits for the radix-4/-5 representation. This is shown in Table 1.6 and Table 1.7. The word length of representation q must not be mixed up with the number of bits we need to realize the overall structure.

The entire PTV filter structure is a two dimensional array. It consists of q columns and w_{in} rows, where w_{in} denotes the input signal word length. Figure 1.25 shows the realization of the FIR-PTV filter with radix-2/3 encoded coefficients. The building block is the add/subtract cell (ASC). Figure 1.26 shows the block diagram of one ASC. One add/subtract cell consist of one delay unit, one XOR, five OR and three AND gates, where five OR and two AND gates compose one full adder.

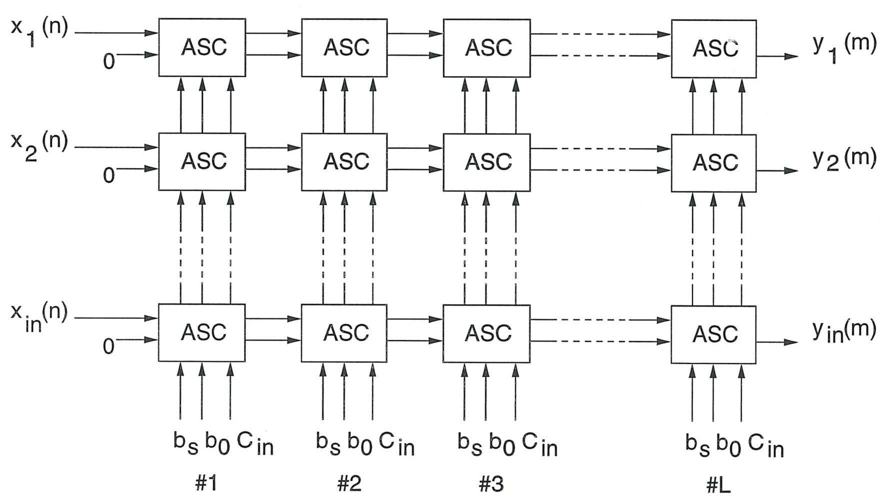

Figure 1.25: Overall FIR-PTV Filter

The entire hardware requirements for a radix-2/-3 number representation are listed

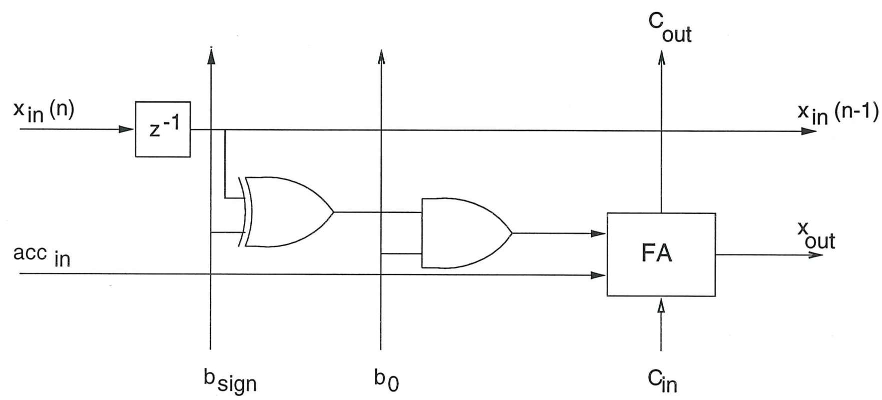

Figure 1.26: Single Add/Subtract Cell for a Ternary Coefficient Set

in Table 1.4, where the full-adder is also decomposed into its basic blocks.

Delay	AND	OR	XOR
L·w_{in}	3L·w_{in}	5L·w_{in}	L·w_{in}

Table 1.4: Hardware Requirements for a Radix-2/-3 encoded FIR-PTV Filter, q is the word length of the representation and w_{in} is the input signal word length

1.7.2 Quantization Error

The quantization error depends on the word length q and the radix. Table 1.5 lists the quantization step.

Radix	ΔQ	ΔQ (q=8)	ΔQ (q=12)
2	$\pm\,(K_{out})^q = \pm\left(\frac{1}{2}\right)^q$	$7.81{\cdot}10^{-3}$	$4.88{\cdot}10^{-4}$
3	$\pm\,(K_{out})^q = \pm\left(\frac{3}{8}\right)^q$	$1.04{\cdot}10^{-3}$	$2.06{\cdot}10^{-5}$
4	$\pm\,(K_{out})^q = \pm\left(\frac{1}{4}\right)^q$	$6.10{\cdot}10^{-5}$	$2.38{\cdot}10^{-7}$
5	$\pm\,(K_{out})^q = \pm\left(\frac{7}{32}\right)^q$	$2.40{\cdot}10^{-5}$	$5.49{\cdot}10^{-8}$

Table 1.5: Quantization Step for Radix-r represented Coefficients

1.7.3 Radix-3 SD Representation

The coefficients vary periodically with the period N to $c_k(m+N) = c_k(m)$, where c_k is the kth coefficient.

Coefficient	b_{sign}	b_0
-1	1	1
0	0	0
+1	0	1
$K_{in} = 3$	$K_{out} = 3^{-N}$	$C_{out} = 2K_{out} = 2/3^N$

Table 1.6: Encoding for the Coefficients in Radix-3 Representation

1.7.4 Radix-4 SD Representation

Coefficient	b_{sign}	b_{sign}	b_0
-2	1	1	1
-1	0	1	1
0	0	0	0
+1	0	0	1
+2	1	0	1
$K_{in} = 4$	$K_{out} = 4^{-N}$	$C_{out}=\frac{3}{2} K_{out} = \frac{3}{2} 4^{-N}$	

Table 1.7: Encoding for the Coefficients in Radix-4 Representation

The constant scale C_{out} requires a multi-shift and add operation.

1.7.5 The Design Flow for a PTV Filter

- Design a target FIR filter which meet the desired specifications
- Set $K_{in} = 3$ (radix-2/-3); $K_{in} = 4$ (radix-4/-5)
- Set $K_{out} = 3^{-N}$ (radix-2/-3); $K_{out} = 4^{-N}$ (radix-4/-5)
- Determine the constant K_{out} depending upon the desired precision.

  Set $C_{out} = 2 \cdot K_{out}$.
- Length of the PTV filter

  $L = N_{target} + D$
- Approximate h_t(n-2) by h(n)