Analysis of Finite Word-Length Effects

Introduction

Finite wordlength effects are caused by:

- Quantization of the filter coefficients
- Rounding / truncation of multiplication results
- Quantization of the input signal
- Dynamic range constraints of the implementation

Analysis of Finite Wordlength Effects

- Ideally, the system parameters along with the signal variables have infinite precision taking any value between $-\infty$ and ∞
- In practice, they can take only discrete values within a specified range since the registers of the digital machine where they are stored are of finite length
- The discretization process results in nonlinear difference equations characterizing the discrete-time systems

Analysis of Finite Wordlength Effects

- These nonlinear equations, in principle, are almost impossible to analyze and deal with exactly
- However, if the quantization amounts are small compared to the values of signal variables and filter parameters, a simpler approximate theory based on a statistical model can be applied

Analysis of Finite Wordlength Effects

- Using the statistical model, it is possible to derive the effects of discretization and develop results that can be verified experimentally
- Sources of errors -

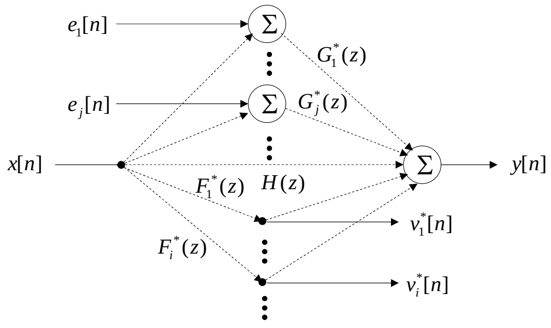
(1) Filter coefficient quantization

(2) A/D conversion

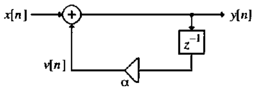
(3) Quantization of arithmetic operations

(4) Limit cycles

Analysis of Noise Properties and Dynamic Range Constraints



Example: First Order IIR Filter



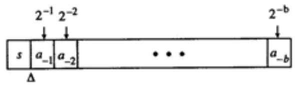
$y[n] = \alpha y[n-1] + x[n]$

$$H(z) = \frac{1}{1-\alpha z^{-1}} = \frac{z}{z-\alpha}$$

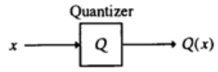
- Quantization of coefficients α: $H'(z) = \frac{1}{1-\alpha' z^{-1}}$
- Quantization of input $x[n]$: $x'[n] = x[n] + e[n]$
- Rounding/truncation of $v[n]$: $v'[n] = v[n] + e_\alpha[n]$
- Output $y[n]$ with finite wordlength: $y'[n] = y[n] + \eta[n]$

The Quantization Process and Errors

- Fractional numbers (sign bit + fractional part)

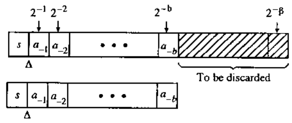


- The quantization process model

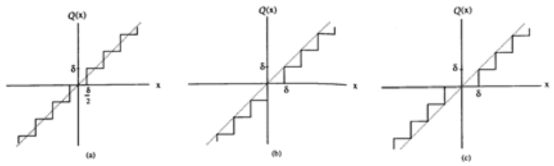


Error: $\varepsilon = Q(x) - x$

- Rounding / Truncation:



The Quantization Errors



Rounding

$-\frac{1}{2}\left(2^{-b} - 2^{-\beta}\right) \le \varepsilon_r \le \frac{1}{2}\left(2^{-b} - 2^{-\beta}\right)$

Two's complement truncation

$-\left(2^{-b} - 2^{-\beta}\right) \le \varepsilon_t \le 0$

Sign-magnitude and one's complement truncation

$-\left(2^{-b} - 2^{-\beta}\right) \le \varepsilon_t \le 0 \quad \text{for } x > 0$

$0 \le \varepsilon_t \le \left(2^{-b} - 2^{-\beta}\right) \quad \text{for } x < 0$

Quantization Error

Table 9.1 Range of quantization error.

Type of quantization	Number representation	Range of Error $Q(x) - x$
Truncation	Positive number Two's-complement negative number	$-\delta < \varepsilon_t \le 0$
Truncation	Sign-magnitude negative number Ones'-complement negative number	$0 \le \varepsilon_t < \delta$
Rounding	All positive and negative numbers	$-\frac{\delta}{2} < \varepsilon_r \le \frac{\delta}{2}$

Note: $\delta = 2^{-b}$.

Quantization of Floating-Point Numbers

- Only mantissa is quantized; the relative error is relevant!

$x = 2^E M, \quad Q(x) = 2^E Q(M)$ ➡ Error: $\mathrm{e} = \frac{Q(x)-x}{x} = \frac{Q(M)-x}{x}$

Table 9.2 Range of relative error $\mathrm{e} = (Q(x) - x)/x$.

Type quantization	Number representation	Range of relative error
Truncation	Two's-complement	$-2\delta < \mathrm{e}_t \le 0, \quad x > 0$ $0 \le \mathrm{e}_t < 2\delta, \quad x < 0$
Truncation	Sign-magnitude Ones'-complement	$-2\delta < \mathrm{e}_t \le 0$
Rounding	All numbers	$-\delta < \mathrm{e}_r \le \delta$

Note: $\delta = 2^{-b}$.

Analysis of Coefficient Quantization Effects

- The transfer function $\hat{H}(z)$ of the digital filter implemented with quantized coefficients is different from the desired transfer function $H(z)$
- Main effect of coefficient quantization is to move the poles and zeros to different locations from the original desired locations

Analysis of Coefficient Quantization Effects

- The actual frequency response $\hat{H}(e^{j\omega})$ is thus different from the desired frequency response $H(e^{j\omega})$
- In some cases, the poles may move outside the unit circle causing the implemented digital filter to become unstable even though the original transfer function $H(z)$ is stable

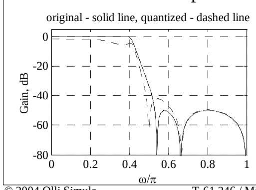
Analysis of Coefficient Quantization Effects

- Direct form realizations are more sensitive to coefficient quantization than cascade or parallel forms
- The sensitivity increases with increasing filter order
- Usually second order blocks in cascade or parallel are used

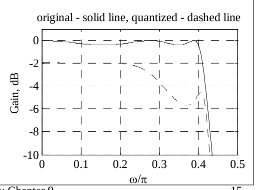
Coefficient Quantization Effects On a Direct Form IIR Filter

- Gain responses of a 5-th order elliptic lowpass filter with unquantized and quantized coefficients

Fullband Gain Response

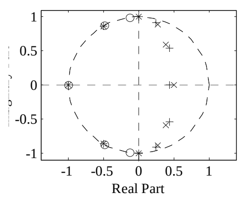


Passband Details



Coefficient Quantization Effects On a Direct Form IIR Filter

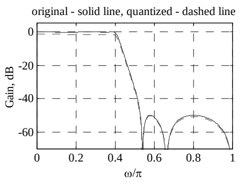
- Pole and zero locations of the filter with quantized coefficients (denoted by “x” and “o”) and those of the filter with unquantized coefficients (denoted by “+” and “*”)



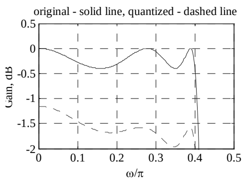
Coefficient Quantization Effects On a Cascade Form IIR Filter

- Gain responses of a 5-th order elliptic lowpass filter implemented in a cascade form with unquantized and quantized coefficients

Fullband Gain Response



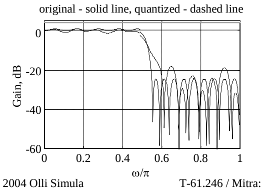
Passband Details



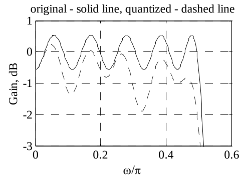
Coefficient Quantization Effects On A Direct Form FIR Filter

- Gain responses of a 39-th order equiripple lowpass FIR filter with unquantized and quantized coefficients

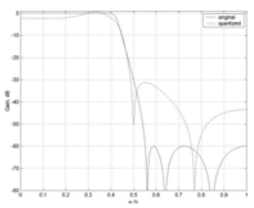
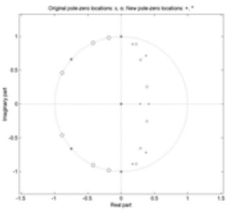
Fullband Gain Response



Passband details

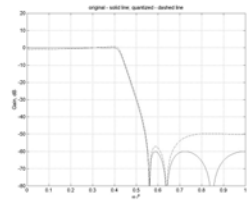
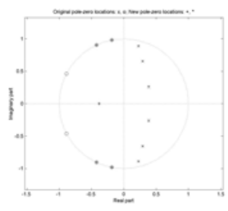


Example of Coefficient Quantization in 6th Order Direct Form Realization

Amplitude responses Pole-zero locations

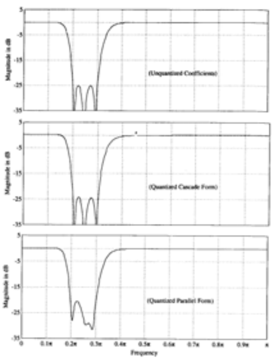
Example of Coefficient Quantization in 6th Order Cascade Form Realization

Amplitude responses Pole-zero locations

<u>Example:</u>

- 6th order bandstop filter with unquantized coefficients
- Cascade form with coefficients quantized to 6 bits
- Parallel form with coefficients quantized to 6 bits



Coefficient Quantization in FIR Filters

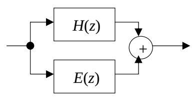
- Consider an $(M-1)$th order FIR transfer function

$$H(z)=\sum_{n=0}^{M-1} h[n]z^{-n}$$

- Quantization of the filter coefficients results in a new transfer function

$$H'(z)=\sum_{n=0}^{M-1} h'[n]z^{-n}=\sum_{n=0}^{M-1}\big(h[n]+e[n]\big)z^{-n}$$

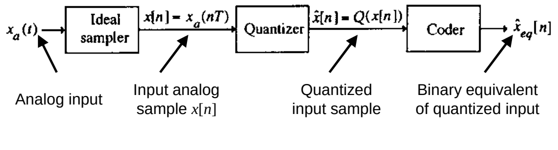
$$\Rightarrow \quad H'(z)=H(z)+E(z)$$



- <u>Linear phase</u>: $h[n] = \pm h[N-1-n]$

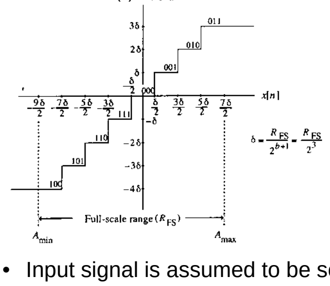
  Symmetry of the impulse response not affected by quantization

A/D Conversion Noise Analysis



- Quantization of the input signal introduces error at the input of the filter
- This error is propagated through the filter together with the input signal
- Affects the signal-to-noise ratio of the system

Quantization Noise Model



- Two's complement representation

$$-1 \le \hat{x}_{eq}[n] < 1$$

$$\hat{x}_{eq}[n]=\frac{2\hat{x}[n]}{R_{FS}}$$

$$R_{FS}=2^{b+1}\delta$$

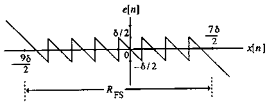
- Input signal is assumed to be scaled to be in the range of ± 1 by dividing its amplitude by $R_{FS}/2$

Quantization Error

- The quantization error $e[n]$:

$$e[n] = Q(x[n]) - x[n] = \hat{x}[n] - x[n]$$

- For two's complement rounding: $-\frac{\delta}{2} < e[n] \leq \frac{\delta}{2}$



- $e[n]$ is called ***granular noise***
- Outside R_{FS} the error increases linearly; $e[n]$ is called the ***saturation error*** or the ***overload noise***
- The output value is clipped to the maximum value

Model of the Quantization Error

$x[n]$ → (+) → $\hat{x}[n] = x[n] + e[n]$, with $e[n]$ added at the summing node

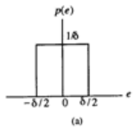
Assumptions:

1) The error sequence $\{e[n]\}$ is a sample sequence of a wide-sense stationary (WSS) white noise process, with each sample $e[n]$ being uniformly distributed over the quantization error
2) The error sequence is uncorrelated with its corresponding input sequence $\{x[n]\}$
3) The input sequence is a sample sequence of a stationary random process

The assumptions hold in most practical situations with rapidly changing input signals

Quantization Error Distributions

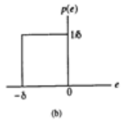
(a) Rounding



$$m_e = \frac{(\delta/2) - (\delta/2)}{2} = 0$$

$$\sigma_e^2 = \frac{((\delta/2) - (-\delta/2))^2}{12} = \frac{\delta^2}{12}$$

(b) Two's complement truncation



$$m_e = \frac{0 - \delta}{2} = -\frac{\delta}{2}$$

$$\sigma_e^2 = \frac{(0 - \delta)^2}{12} = \frac{\delta^2}{12}$$

- The variance represents the noise power

Signal-to-Noise ratio

- Additive quantization noise $e[n]$ on the signal $x[n]$
- Signal-to-quantization noise ratio in dB is defined as

$$SNR = 10\log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) \text{ dB}$$

where

- σ_x^2 is the signal variance (power) and
- σ_e^2 is the noise variance (power)

Signal-to-Noise Ratio

A/D conversion:

- (b+l) bits: $\delta = 2^{-(b+1)} R_{FS}$, where R_{FS} is the full-scale range

$$\sigma_e^2 = \frac{\delta^2}{12} = \frac{2^{-2(b+1)} R_{FS}^2}{12} = \frac{2^{-2b} R_{FS}^2}{48}$$

$$SNR_{A/D} = 10\log_{10}\left(\frac{48\sigma_x^2}{2^{-2b} R_{FS}^2}\right) = 6.02\text{b} + 16.81 - 20\log_{10}\left(\frac{R_{FS}}{\sigma_x}\right) \text{ dB}$$

- Thus, SNR increases 6 dB for each added bit in the wordlength

Effect of Input Scaling on *SNR*

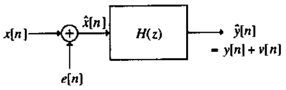
- Let the input scaling factor be A with $A>0$
- The variance of the scaled input $Ax[n]$ is $A^2\sigma_x^2$
- The SNR changes to

$$SNR_{A/D} = 6.02\text{b} + 16.81 - 20\log_{10}\left(\frac{R_{FS}}{A\sigma_x}\right)$$
$$= 6.02b + 16.81 - 20\log_{10}(K) + 20\log_{10}(A)$$

where $R_{FS} = K\sigma_x$ (σ_x is the RMS value of the signal)

- Scaling down the input signal ($A<1$) decreases the SNR
- Scaling up the input signal ($A>1$) increases the possibility to exceed the full-scale range R_{FS} resulting in clipping SNR

Propagation of Input Quantization Noise to Digital Filter Output



- Due to linearity of $H(z)$ and the assumption that $x[n]$ and $e[n]$ are uncorrelated the output can be expressed as a linear combination (sum) of two sequences:

$$\hat{y}[n] = h[n] * \hat{x}[n]$$
$$= h[n] * \left[x[n] + e[n]\right] = h[n] * x[n] + h[n] * e[n]$$

- The output noise is: $v[n] = \sum_{m=-\infty}^{\infty} e[m]h[n-m]$

Propagation of Input Quantization Noise to Digital Filter Output

- The mean and variance of $v[n]$ characterize the output noise
- The mean m_v is: $m_v = m_e H(e^{j0})$
- The noise variance σ_v^2 is:

$$\sigma_v^2 = \frac{\sigma_e^2}{2\pi} \int_{-\pi}^{\pi} \left| H(e^{j\omega}) \right|^2 d\omega$$

- The output noise power spectrum is:

$$P_{vv}^2(\omega) = \sigma_e^2 \left| H(e^{j\omega}) \right|^2$$

Propagation of Input Quantization Noise to Digital Filter Output

- The normalized output noise variance is given by

$$\sigma_{v,n}^2 = \frac{\sigma_v^2}{\sigma_e^2} = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| H(e^{j\omega}) \right|^2 d\omega$$

which can be written as:

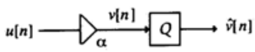
$$\sigma_{v,n}^2 = \frac{1}{2\pi j} \oint_C H(z) H(z^{-1}) z^{-1} dz$$

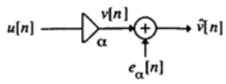
- An equivalent expression is:

$$\sigma_{v,n}^2 = \sum_{n=-\infty}^{\infty} \left| h[n] \right|^2$$

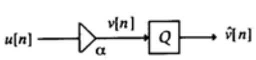
Analysis of

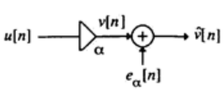
Arithmetic Round-Off Errors





Quantization of Multiplication Results





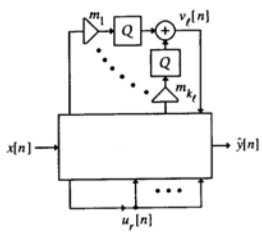
Assumptions:

1) The error sequence $\{e_\alpha[n]\}$ is a sample sequence of a stationary white noise process, with each sample $e_\alpha[n]$ being uniformly distributed
2) The quantization error sequence $\{e_\alpha[n]\}$ is uncorrelated with the signal $\{v[n]\}$, the input sequence $\{x[n]\}$ to the filter, and all other quantization errors

The assumption of $\{e_\alpha[n]\}$ being uncorrelated with $\{v[n]\}$ holds for rounding and two's complement truncation

Quantization of Multiplication Results

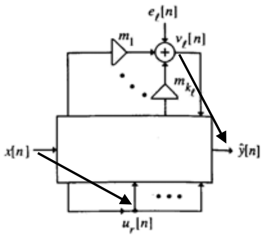
- The quantization model can be used to analyze the quantization effects at the filter output



- Quantization before summation
- The number of multiplications k_l at adder inputs
- The rth branch node with signal value $u_r[n]$ needs to be scaled to prevent overflow

Quantization of Multiplication Results

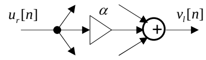
- Statistical model of the filter:



- $f_r[n]$
  Impulse response from filter input to branch node r
- $g_l[n]$
  Impulse response from input of lth adder to filter output

Quantization of Multiplication Results

- Branch nodes to be scaled lead to multipliers and are outputs of summations:



- Scaling transfer function: $F_r(z)$
- Noise transfer function: $G_l(z)$
- Let σ_0^2 be the variance of each individual noise source; then $k_l\sigma_0^2$ is the noise variance of $e_l[n]$
- The output noise variance is:

$$\sigma_0^2\left[k_l\left(\tfrac{1}{2\pi j}\oint_C G_l(z)G_l(z^{-1})z^{-1}dz\right)\right] = \sigma_0^2\left[k_l\left(\tfrac{1}{2\pi}\int_{-\pi}^{\pi}\left|G(e^{j\omega})\right|^2 d\omega\right)\right]$$

Quantization of Multiplication Results

- The total output noise variance:

$$\sigma_\gamma^2 = \sigma_0^2\sum_{l=1}^{L} k_l\left(\tfrac{1}{2\pi j}\oint_C G_l(z)G_l(z^{-1})z^{-1}dz\right)$$

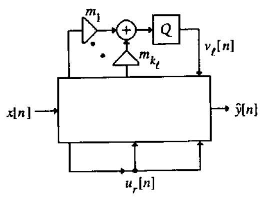
where L is the number of summation nodes to which noise sources are connected

- The noise variance can also be written as

$$\sigma_\gamma^2 = \sigma_0^2\sum_{l=1}^{L} k_l\sum_{n=0}^{\infty}\left|g_l^*[n]\right|^2$$

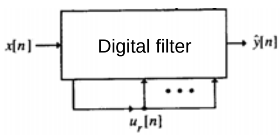
The Output Quantization Noise

- The amount of noise depends on the implementation



- Quantization of multiplication results after summation reduces the number of noise sources to one
- The variance of the noise source $e_l[n]$ is now σ_0^2
- DSP processor carry out multiply-accumulate operation using double precision arithmetic

Dynamic Range Scaling

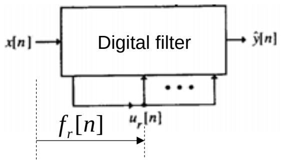


- The rth node value $u_r[n]$ has to be scaled
- Assume that the input sequence is bounded by unity, i.e., $|x[n]| \leq 1$ for all values of n
- The objective of scaling is toensure that
  $|u_r[n]| \leq 1$ for all r and all values of n

Dynamic Range Scaling

- Three different conditions to ensure that $u_r[n]$ satisfies the conditions:
  1) An absolute bound
  2) $L_{infinity}$ -bound
  3) L_2 -bound
- Different bounds are applicable under certain input signal conditions

An Absolute Bound



- $F_r(z)$ is the scaling transfer function
- The node value $u_r[n]$ is determined by the convolution

$$u_r[n] = \sum_{k=-\infty}^{\infty} f_r[k]x[n-k]$$

An Absolute Bound

- Assuming that $x[n]$ satisfies the dynamic range constraint $|x[n]| \le 1$

$$|u_r[n]| = \left| \sum_{k=-\infty}^{\infty} f_r[k]x[n-k] \right| \le \sum_{k=-\infty}^{\infty} |f_r[k]|$$

- The node value $u_r[n]$ now satisfies the dynamic range constraint, i.e., $|u_r[n]| \le 1$ if

$$\sum_{k=-\infty}^{\infty} |f_r[k]| \le 1 \quad \text{for all } r$$

- This is both necessary and sufficient condition to guarantee that there will be no overflow

Scaling with the Absolute Bound

- If the dynamic range constraint is not satisfied the filter input has to be scaled with the multiplier K

$$K = \frac{1}{\max_r \sum_{k=-\infty}^{\infty} |f_r[k]|}$$

- The scaling rule based on the absolute bound is too pessimistic and reduces the SNR significantly
- More practical and easy to use scaling rules can be derived in the frequency domain if some information about the input signal is known a priori

Scaling Norms

- Define the L_p-norm of a Fourier transform $F(e^{j\omega})$ as

$$\|F\|_p = \left(\frac{1}{2\pi} \int_{-\pi}^{\pi} |F(e^{j\omega})|^p d\omega \right)^{1/p}$$

- L_2-norm, $\|F\|_2$, is the root-mean-square (RMS) value of $F(e^{j\omega})$, and
- L_1-norm, $\|F\|_1$, is the mean absolute value of $F(e^{j\omega})$ over ω
- Moreover, $\lim_{p\to\infty} \|F\|_p$ exists for a continuous $F(e^{j\omega})$ and is given by its peak

$$\|F\|_\infty = \max_{-\pi \le \omega \le \pi} |F(e^{j\omega})|$$

Scaling Norms: L_∞-Bound

$$U_r(e^{j\omega}) = F_r(e^{j\omega})X(e^{j\omega})$$

- An inverse Fourier transform

$$u_r[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_r(e^{j\omega})X(e^{j\omega})e^{j\omega n} d\omega$$

$$|u_r[n]| \le \frac{1}{2\pi} \int_{-\pi}^{\pi} |F_r(e^{j\omega})||X(e^{j\omega})| d\omega$$

$$\le \|F_r(e^{j\omega})\|_\infty \left[\frac{1}{2\pi} \int_{-\pi}^{\pi} |X(e^{j\omega})| d\omega \right]$$

$$\le \|F_r(e^{j\omega})\|_\infty \|X(e^{j\omega})\|_1$$

Scaling Norms: L_∞-Bound

- If $\|X\|_1 \le 1$, then the dynamic range constraints satisfied if

$$\|F\|_\infty \le 1$$

- If the mean absolute value of the input spectrum is bounded by unity, then there will be no adder overflow if the peak gains from the filter input to all adder output nodes are scaled satisfying the above bound
- The scaling rule is rarely used since with most input signals encountered in practice $\|X\|_1 \le 1$ does not hold

Scaling Norms: L_2-Bound

$$u_r[n]=\frac{1}{2\pi}\int_{-\pi}^{\pi}F_r(e^{j\omega})X(e^{j\omega})e^{j\omega n}d\omega$$

- Applying Schwarz inequality

$$|u_r[n]|^2\le\left(\frac{1}{2\pi}\int_{-\pi}^{\pi}\left|F_r(e^{j\omega})\right|^2d\omega\right)\left(\frac{1}{2\pi}\int_{-\pi}^{\pi}\left|X(e^{j\omega})\right|^2d\omega\right)$$

or equivalently $\quad |u_r[n]|\le\left\|F_r(e^{j\omega})\right\|_2\left\|X(e^{j\omega})\right\|_2$

- If the filter input has finite energy bounded by unity, i.e., $\|X\|_2\le 1$, then the adder overflow can be prevented by scaling the filter such that the RMS value of the scaling transfer functions are bounded by unity: $\quad \|F\|_2\le 1,\quad r=1,2,...,R$

A General Scaling Rule

- A more general scaling rule is obtained using Holder's inequality

$$|u_r[n]|\le\left\|F_r(e^{j\omega})\right\|_p\left\|X(e^{j\omega})\right\|_q$$

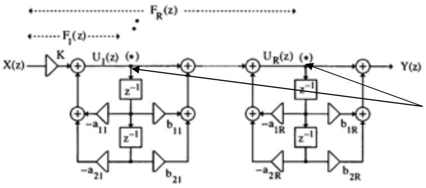
for all $p,q\ge 1$, with $\quad (1/p)+(1/q)=1$

- After the scaling the transfer functions become $\|F'\|_p$ and the scaling constants should be chosen such that

$$\|F'\|_p\le 1,\quad r=1,2,...,R$$

- In many structures the scaling multipliers can be absorbed to the existing feedforward multipliers

Scaling of a Cascade Form IIR Filter



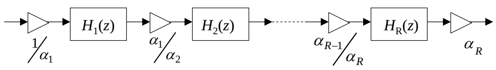
The nodes (•) need to be scaled

$$H(z)=K\prod_{i=1}^{R}H_i(z),\qquad \text{where}\quad H_i(z)=\frac{B_i(z)}{A_i(z)}=\frac{1+b_{1i}z^{-1}+b_{2i}z^{-2}}{1+a_{1i}z^{-1}+a_{2i}z^{-2}}$$

- Scaling transfer functions: $\quad F_r(z)=\frac{K'}{A_r(z)}\prod_{l=1}^{r-1}H'_l(z),\quad r=1,2,...,R$
- $F_r(z)$ can be expressed by poles and zeros of the original $H(z)$

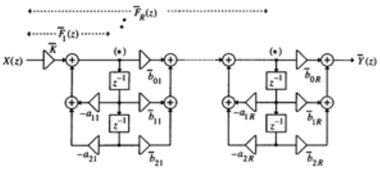
Scaling - Back-Scaling

- The effect of input scaling is compensated by back-scaling at the output of the filter
- Scaling block-by-block in cascade realization forms



- Each second order block is scaled individually
- The scaling coefficients between the blocks contain the back-scaling of the previous block and the scaling of the the next block

Scaled Cascade Form IIR Filter Structure



Scaling transfer functions:

$$\overline{F}_r(z)=\frac{\overline{K}}{A_r(z)}\prod_{l=1}^{r-1}\overline{H}_l(z),\quad r=1,2,...,R$$

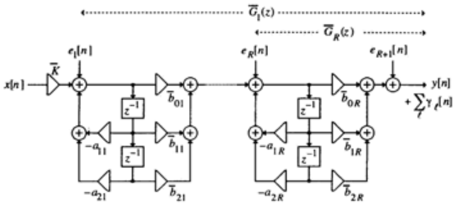
$$\overline{H}(z)=\overline{K}\prod_{l=1}^{R}\overline{H}_l(z),\qquad \text{where}\quad \overline{H}_l(z)=\frac{1+\overline{b}_{1i}z^{-1}+\overline{b}_{2l}z^{-2}}{1+\overline{a}_{1l}z^{-1}+\overline{a}_{2l}z^{-2}}$$

- The scaled structure has new values of the coefficients in the feed-forward branches
- Only one critical branch node in each second order block has to be checked for overflow

Optimum Section Ordering and Pole-Zero Pairing of a Cascade Form IIR Digital Filter

Ordering of second-order sections as well as pairing of poles and zeros affects the output noise power of the filter

Noise Transfer Functions



- The noise transfer functions can be expressed using the transfer functions of the cascaded second-order blocks
- The scaled noise transfer functions are given by

$$\overline{G}_l(z) = \overline{K}\prod_{i=l}^{R}\overline{H}_i(z) = \left(\prod_{i=l}^{R}\beta_i\right)G_l(z), \quad l = 1,2,...,R; \text{ and } \quad \overline{G}_{R+1}(z) = 1$$

Noise Transfer Functions

- The output noise power spectrum due to product round-off is given by

$$P_{yy}(\omega) = \sigma_0^2\left[\sum_{l=1}^{R+1}k_l\left|\overline{G}_l(e^{j\omega})\right|^2\right]$$

and output noise variance is

$$\sigma_y^2 = \sigma_0^2\left[\sum_{l=1}^{R+1}k_l\left(\frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\overline{G}_l(e^{j\omega})\right|^2 d\omega\right)\right] = \sigma_0^2\left[\sum_{l=1}^{R+1}k_l\left\|\overline{G}_l\right\|_2^2\right]$$

where the integral in the parenthesis is the square of the L_2-norm of the noise transfer function

Noise Model of Second-Order Blocks

- The noise model introduces noise sources to the input/output summation of each block
- The number of elementary noise sources, k_l, has different values depending on the location of rounding (before or after the summation) and depending on the block (first, intermediate, last)
- Let k_l be the total number multipliers connected to the l^{th} adder
  - Rounding before summation: $k_1 = k_{R+1} = 3$, $k_l = 5$, for $l = 2, 3,...,R$
  - Rounding after summation: $k_l = 1$, for $l = 1, 2,...,R+1$

Noise Transfer Functions

- The scaling coefficients are

$$\prod_{i=l}^{R}\beta_i = \frac{\alpha_l}{\alpha_{R+1}} = \frac{\left\|F_l\right\|_p}{\left\|H\right\|_p}$$

- The output noise power spectrum of the scaled filter is

$$P_{yy}(\omega) = \frac{\sigma_0^2}{\left\|H\right\|_p^2}\left[k_{R+1}\left\|H\right\|_p^2 + \sum_{l=1}^{R+1}k_l\left\|F_l\right\|_p^2\left|G_l(e^{j\omega})\right|^2\right]$$

and output noise variance is

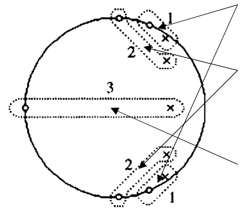
$$\sigma_y = \frac{\sigma_0^2}{\left\|H\right\|_p^2}\left[k_{R+1}\left\|H\right\|_p^2 + \sum_{l=1}^{R+1}k_l\left\|F_l\right\|_p^2\left\|G_l\right\|_2^2\right]$$

Minimizing the Output Round-Off Noise

- The scaling transfer function $F_l(z)$ contains sections $H_i(z)$, $i = 1, 2,..., l-1$
- The noise transfer function $G_l(z)$ contains sections $H_i(z)$, $i = l, l+1,..., R$
- Every term in the sum for the noise power or the noise variance includes the transfer function of all R sections in the cascade realization
- To minimize the output noise power the norms of $H_i(z)$ should be minimized for all values of i by appropriately pairing the poles and zeros

Pairing the Poles and Zeros

- Poles close to unit circle introduce gain and zeros (on the unit circle) introduce attenuation



1) First, the poles closest to the unit circle should be paired with the nearest zeros
2) Next, the poles closest to the previous set of poles should be paired with the next closest zeros
3) This process is continued until all poles and zeros are paired

Section Ordering

- A section in the front part of the cascade has its transfer function $H_i(z)$ appearing more frequently in the scaling transfer functions
- A section near the output end of the cascade has its transfer function $H_i(z)$ appearing more frequently in the noise transfer function expressions

=> **The best location for $H_i(z)$ depends on the type of norms being applied to the scaling and noise transfer functions**

Section Ordering

L_2 scaling:

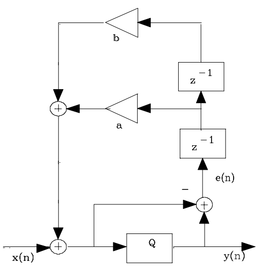
- The ordering of paired sections does not influence too much the output noise power since all norms in the expressions are L_2-norms

L_∞ scaling:

- The sections with poles closest to the unit circle exhibit a peaking magnitude response and should be placed closer to the output end

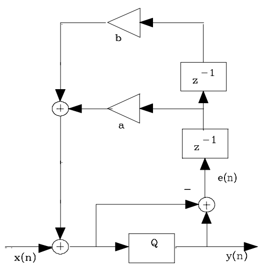
  => The ordering should be from least-peaked to most-peaked
- On the other hand, the ordering scheme is exactly opposite if the objective is to minimize the peak noise $\|P_{yy}(\omega)\|_\infty$ and L_2-scaling is used
- The ordering has no effect on the peak noise with L_∞-scaling

Error Spectrum Shaping



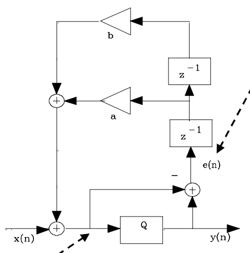
- Quantization error can be compensated using the so called error-feedback (or error spectrum shaping)
- The filtered error signal is added to the signal branch before quantization (Q[.]).

Error Spectrum Shaping



- Without error-feedback the error signal $e[n]$ is the pure quantization error, i.e., $e[n] = y[n] - x[n]$
- In the compensated structure the error signal is the difference between the output $y[n]$ and the compensated input signal

Error Spectrum Shaping



$e[n] = y[n] - w[n]$

- Substituting $w[n]$:

  $e[n] = y[n] - x[n] - ae[n-1] - be[n-2]$
- Total error between output and input is still:

  $e[n] = y[n] - x[n]$

$w[n] = x[n] + ae[n-1] + be[n-2]$

Error Spectrum Shaping

- Solving $y[n]$ - $x[n]$:

  $y[n] - x[n] = e[n] + ae[n-1] + be[n-2]$
- Taking the z-transform:

$$Y(z) - X(z) = E(z) + az^{-1}E(z) + bz^{-2}E(z)$$
$$= (1 + az^{-1} + bz^{-2})E(z) = G(z)E(z)$$

where $G(z)$ is the error shaping transfer function

Error Spectrum Shaping

- Example: a=-2 and b=1

$$G(z) = \left(1 + az^{-1} + bz^{-2}\right)$$
$$= \left(1 - 2z^{-1} + z^{-2}\right) = \left(1 - z^{-1}\right)^2$$

- Double zero is at z=1
- Noise spectrum is modified by attenuating noise at low frequencies