80509 LINEAR DIGITAL FILTERING I

PART V: Finite word length effects in digital filters

1) Output noise due to the multiplication roundoff errors.
2) Filter scaling: various scaling norms.
3) Coefficient quantization errors.
4) Various kinds of oscillations.

● What to read for the examination ?:

1) How to scale a digital filter; the basic scaling norms and their differences in practice?

2) How to estimate output noise due to the multiplication roundoff errors?

3) It is very likely that in the examination there is a simple filter which must be scaled (according to some norm) and then the output noise must be estimated.

Please study carefully exercises in Appendix B.

FINITE WORD LENGTH EFFECTS IN DIGITAL FILTERS

- Digital filters are implemented using finite word lengths for both the data and the filter coefficients.
- The main errors caused by the use of finite word length are as follows:
  - Noise generated in the analog-to-digital conversion, resulting from representing the samples of the input data by only a few bits;
  - Coefficient quantization errors, caused by representing the filter coefficients by a finite number of bits;
  - Various kinds of oscillations, such as parasitic oscillations, overflow oscillations, and limit cycles;
  - Output noise due to multiplication roundoff errors, resulting from the rounding or truncation of multiplication products within the filter.

FIXED-POINT ARITHMETIC

- We consider only filters where both the coefficients and data samples are given using a fixed-point representation.

- In this case, each data sample is represented by a sign bit and b decimal bits and it is required that inside the filter all the data samples are within the range $[-1, 1]$.

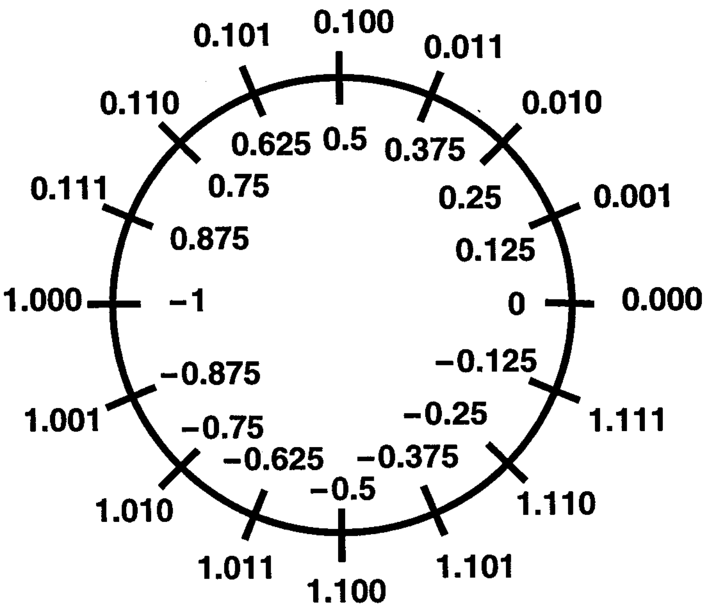
- For the coefficients, some integer bits are sometimes required.

- The following table shows several number systems for the (1+3)-bit representation.

	Interpretation		
Binary Number	*Sign and Magnitude*	*Two's-Complement*	*One's-Complement*
$0_\Delta 111$	7/8	7/8	7/8
$0_\Delta 110$	6/8	6/8	6/8
$0_\Delta 101$	5/8	5/8	5/8
$0_\Delta 100$	4/8	4/8	4/8
$0_\Delta 011$	3/8	3/8	3/8
$0_\Delta 010$	2/8	2/8	2/8
$0_\Delta 001$	1/8	1/8	1/8
$0_\Delta 000$	0	0	0
$1_\Delta 000$	−0	−1	−7/8
$1_\Delta 001$	−1/8	−7/8	−6/8
$1_\Delta 010$	−2/8	−6/8	−5/8
$1_\Delta 011$	−3/8	−5/8	−4/8
$1_\Delta 100$	−4/8	−4/8	−3/8
$1_\Delta 101$	−5/8	−3/8	−2/8
$1_\Delta 110$	−6/8	−2/8	−1/8
$1_\Delta 111$	−7/8	−1/8	−0

TWO'S COMPLEMENT ARITHMETIC

- This arithmetic has several attractive properties:
  - When numbers are added, the sign bit is treated in the same manner as the other bits.
  - When several numbers are added, overflows (jumps outside the range $[-1, 1)$ are allowed provided that the final result is within the desired range.
  - The following circle illustrates how additions are performed in the case of two's complement arithmetic for the (1+3)-bit representation. The following transparency gives an example.



TWO'S COMPLEMENT ARITHMETIC

- When a positive number is added with amplitude equal to $a/8$, we travel counterclockwise along the circle $(a/8) \cdot 180$ degrees.

- When a negative number is added with magnitude equal to $a/8$, we travel clockwise along the circle $(a/8) \cdot 180$ degrees.

- As an example, consider the case $-0.75 - 0.75 + 0.5 + 0.5$.

- When adding -0.75 and -0.75, we travel twice clockwise 135 degrees, arriving at the angle of -270 degrees or 90 degrees. This corresponds to 0.5, instead of -1.5 (an overflow happened!!). When adding twice 0.5, we travel counterclockwise twise 90 degrees, arriving at the angle of 170 degree. The corresponding number is -0.5 that is the desired correct result, even though an overflow took place earlier.

- Note that for two's complement arithmetic the largest positive number is $1 - 2^{-b}$ with b being the number of fractional bits. Thus, $+1$ does not exist!

- In practice, in the case where b is large enough, for example, $b = 15$, it can be required that the numbers must stay within $[+1, -1]$, intead of $[1 - 2^{-b},\ -1]$. For simplicity, this assumption is used for the filter scaling to be described later.

ROUNDING AND TRUNCATION

- If a $(1+b)$-bit data sample is multiplied by a $(1+a)$-bit coefficient, then the result, denoted by x, is a $(1+a+b)$-bit number, which must be represented again by a $(1+b)$-bit number.

- This operation can be performed either using a rounding or a truncation.

- In order to explain these operations, we express a $(1+a+b)$-bit number as

$$x = s_{\Delta} d_1 d_2 .... d_{a+b},$$

where s is the sign bit ($s = 0$ and $s = 1$ for positive and negative numbers, respectively) and d_k's for $k = 1, 2, \cdots, a+b$ have either the value of zero or unity.

- We concentrate only on **two's complement arithmetic** for which (see transparency 2 for a (1+3)-bit representation):

$$x = -s + x_d,$$

where

$$x_d = \sum_{k=1}^{a+b} d_k 2^{-k}. \qquad (A)$$

- In other words, for positive numbers ($s = 0$),

$$x = x_d, \qquad (B)$$

whereas for negative numbers ($s = 1$),

$$x = -1 + x_d. \qquad (C)$$

- As a matter of fact, the term "two's complement arithmetic" comes from the following properties:

  1) First s is interpreted as an integer bit so that $\widetilde{x} = 1^s + x_d$. $\widetilde{x}$ is thus in the range $[0,\ 2 - 2^{a+b}]$.

  2) Then, for $s = 0$, $x \equiv \widetilde{x}$, whereas for $s = 1$, $x \equiv 2 - \widetilde{x}$.

- Hence, the resulting numbers stay within the range $[-1,\ 1 - 2^{-(a+b)}]$ when $a + b$ decimal bits are in use (see again transparency 2).

TRUNCATION IN TWO'S COMPLEMENT ARITHMETIC

- In the case of truncation, a $(1+a+b)$-bit number is expressed as the following $(1+b)$-bit number:

$$\widehat{x} = Q[x] = Q[s_\Delta d_1 d_2 .... d_{b+a}] = s_\Delta d_1 d_2 .... d_b,$$

that is, the d_k's for $k = b+1, b+2, \cdots, b+a$ are simply disregarded.

- In the above, $Q[\;]$ stands for the quantization operation.

- This means that

$$e = \widehat{x}_d - x_d,$$

where

$$\widehat{x}_d = 0_\Delta d_1 d_2 .... d_b$$

and x_d is given by equation (A) of transparency 5, satisfies

$$-\Delta < e \leq 0,$$

where

$$\Delta = 2^{-b}.$$

- Because of equations (B) and (C) of transparency 5, the following is valid for truncation in two's complement arithmetic:

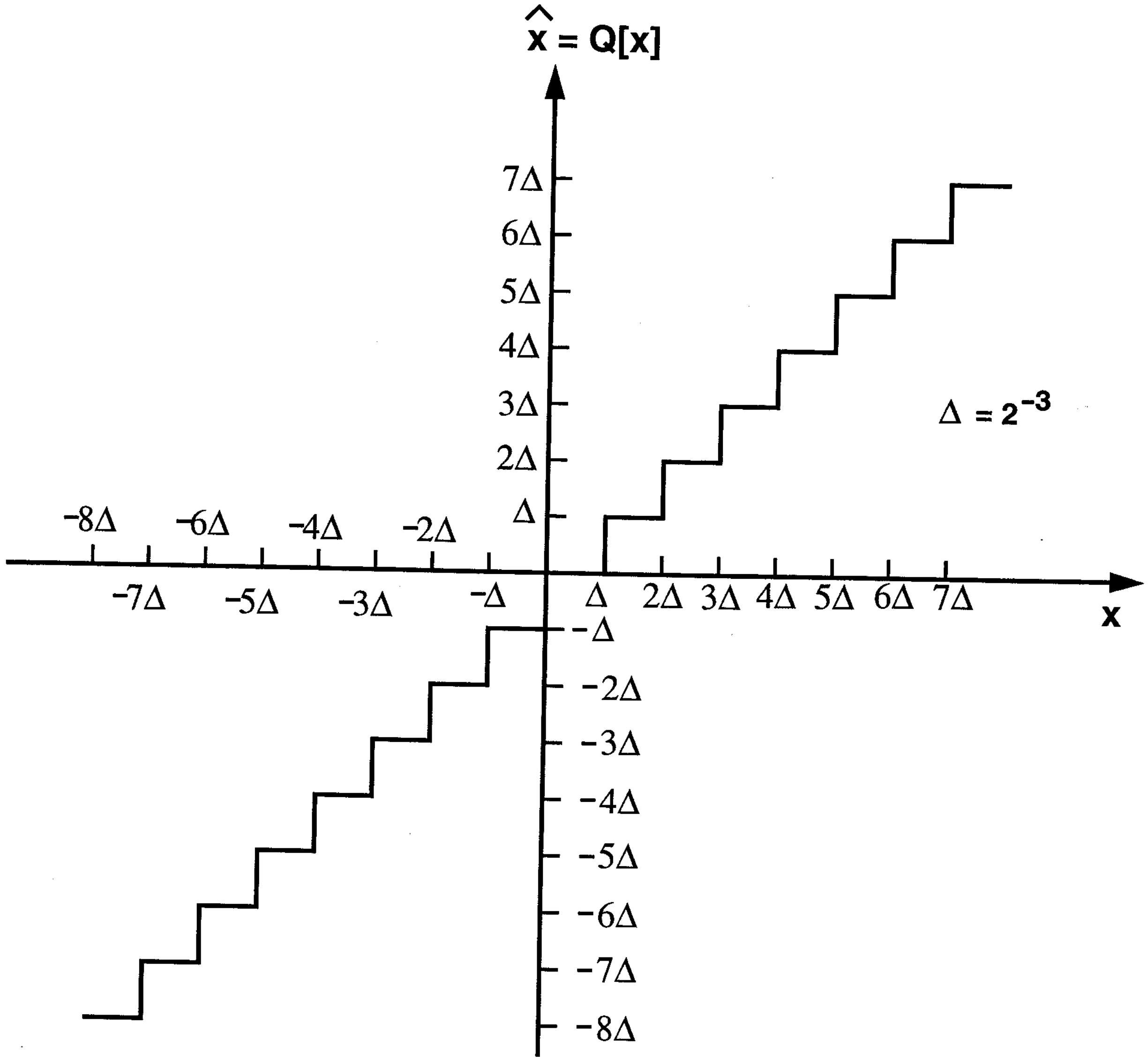
  $$\widehat{x} = Q[x] = x + e$$

  where

  $$-\Delta = -2^{-b} < e \leq 0.$$

- The following transparency shows the relation between $\widehat{x} = Q[x]$ and x for $b = 3$.

TRUNCATION IN TWO'S COMPLEMENT ARITHMETIC FOR $b = 3$: $-\Delta < \widehat{x} - x = Q[x] - x \leq 0$, $\Delta = 2^{-b}$



ROUNDING IN TWO'S COMPLEMENT ARITHMETIC

- In the case of rounding, a $(1+a+b)$-bit number is expressed as the following $(1+b)$-bit number:

  $\widehat{x} = Q[x] = Q[s_\Delta d_1 d_2 .... d_{b+a}] = s_\Delta d_1 d_2 .... d_b + 0_\Delta c_1 c_2 ... c_b,$

  where $c_k = 0$ for $k = 1, 2, \cdots, b-1$ and

$$c_b = \begin{cases} 1, & \text{for } d_{b+1} = 1 \\ 0, & \text{for } d_{b+1} = 0. \end{cases}$$

- In other words, rounding is performed to $(1-b)$-bits using the following two steps:

1) Perform truncation to $(1-b)$-bits.

2) Add 2^{-b} to the result of Step 1 if $d_{b+1} = 1$. Otherwise, keep the truncated result of Step 1.

- For rounding in two's complement arithmetic the following is thus valid:

$$\widehat{x} = Q[x] = x + e,$$

where

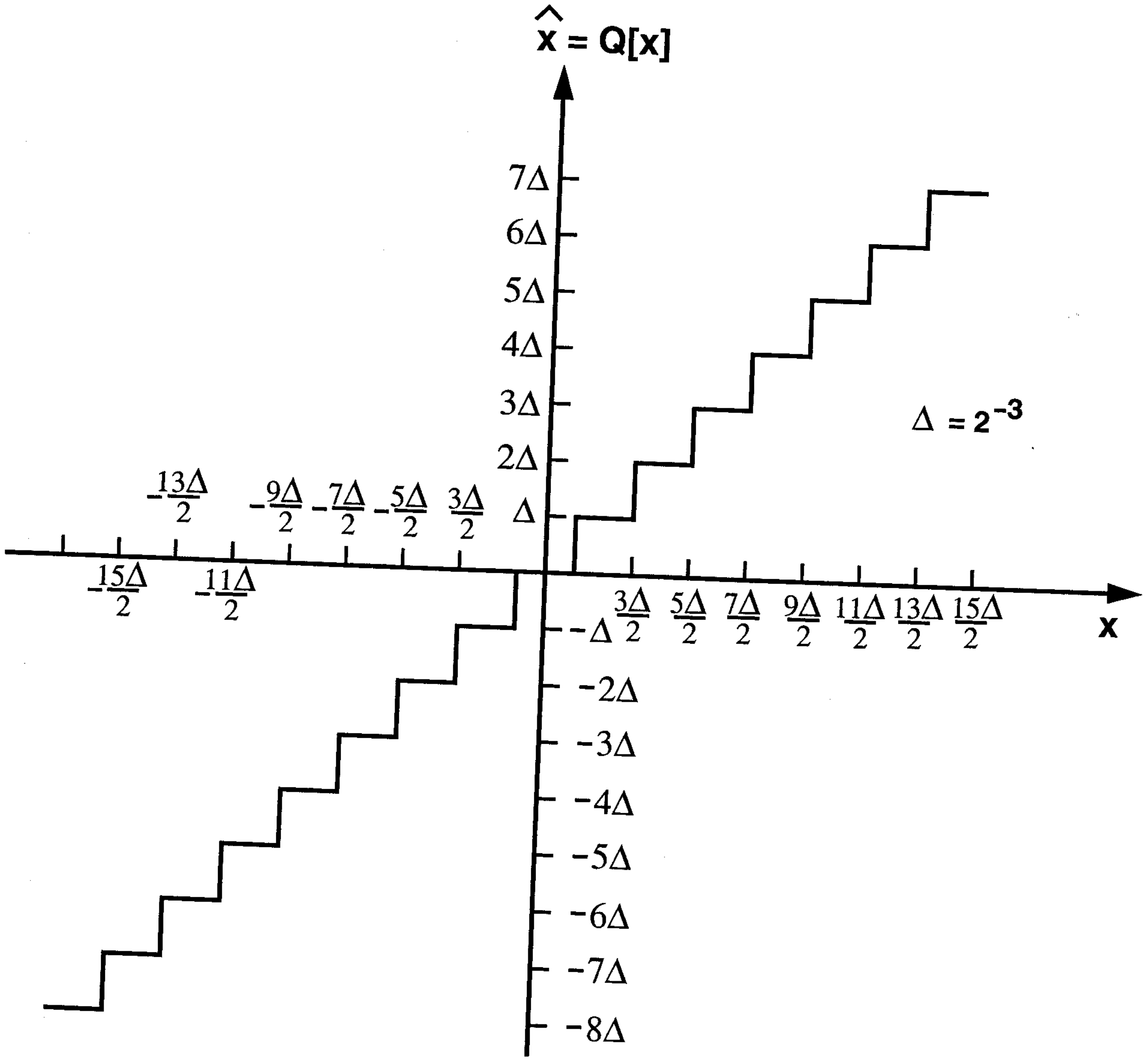
$$-\Delta/2 < e \leq \Delta/2,$$

with

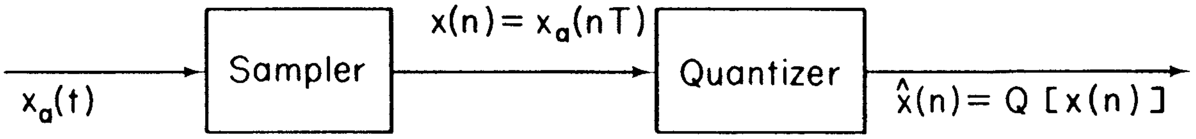
$$\Delta = 2^{-b}.$$

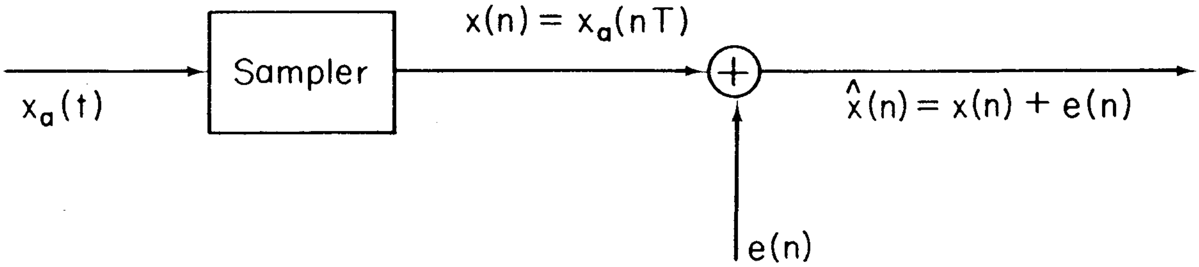
- The following transparency shows the relation between $\widehat{x} = Q[x]$ and x for $b = 3$.

ROUNDING IN TWO'S COMPLEMENT ARITHMETIC FOR $b = 3$: $-\Delta/2 < \widehat{x} - x = Q[x] - x \leq \Delta/2$, $\Delta = 2^{-b}$



QUANTIZATION IN SAMPLING ANALOG SIGNALS





- Consider the above figure, where it is assumed that the unquantized samples $x_a(n)$ are within the range the $(1+b)$-bit number, that is, for two's complement arithmetic, they satisfy

$$-1 \leq x_a(nT) \leq (1 - 2^{-b}).$$

- This corresponds to a proper scaling of the input signal to the A/D-converter. For instance, if the input varies between ±5 volts, we can scale the signal in such a manner, that 5 volts corresponds to unity and -5 volts to minus unity.

- Before feeding the samples $x(n) = x_a(nT)$ to a digital filter they must be quantized to $(1+b)$-bit numbers $\widehat{x}(n)$, as indicated in Figure (a) above.

- For the quantized $(1+b)$-bit samples in the case of two's complement arithmetic, it is valid

$$\widehat{x}(n) = Q[x(n)] = x(n) + e(n),$$

where

$$-\Delta/2 < e(n) \leq \Delta/2$$

for rounding and

$$-\Delta < e(n) \leq 0$$

for truncation with

$$\Delta = 2^{-b}.$$

- The quantization process can be modeled by representing the quantization effect by including an additive noise sourse $e(n)$ as shown in Figure (b) of the previous transparency.

QUANTIZATION ERROR

- It is common to make the following assumptions:

  1. The sequence of the error samples $e(n)$ is a sample sequence of a stationary random process.

  2. The error sequence is uncorrelated with the sequence of exact samples $x(n)$.

  3. The random variables of the error process are uncorrelated, that is, the error is a white-noise process.

  4. The probability distribution of the error process is uniform over the range of quantization error.

- The following figure shows the probability distributions for both rounding and truncation.

PROBABILITY DENSITY FUNCTIONS FOR ROUNDING AND TRUNCATION

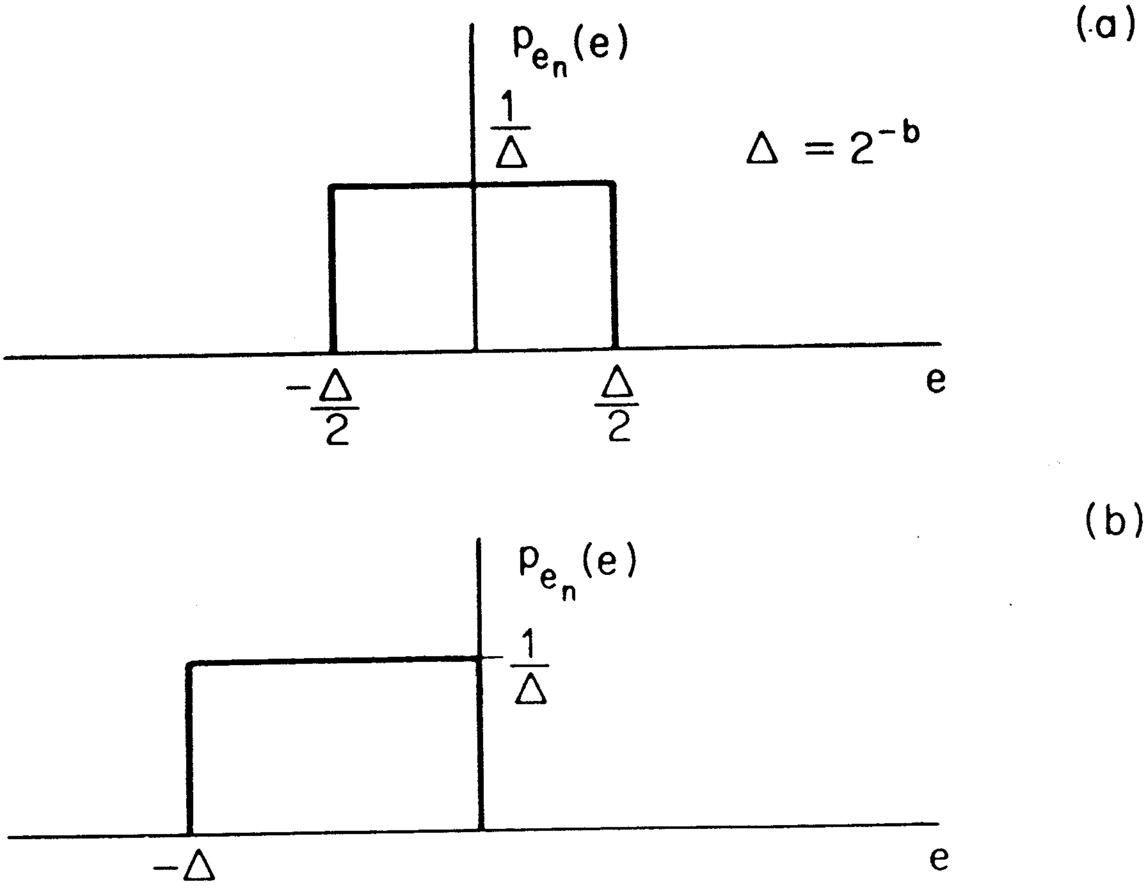


Fig. 9.4 Probability density functions for (a) rounding; (b) truncation.

QUANTIZATION NOISE

- Based on the above assumptions, the following is valid (see Appendix A: Discete-Time Random Signals in the end of this pile):

- For **rounding**, the mean and variance of the quantization noise are given by

$$m_e = 0$$

and

$$\sigma_e^2 = \frac{1}{\Delta}\int_{-\Delta/2}^{\Delta/2} e^2 de = \frac{\Delta^2}{12} = \frac{2^{-2b}}{12}.$$

- For **truncation**, the corresponding quantities are

$$m_e = -\frac{2^{-b}}{2}$$

and

$$\sigma_e^2 = \frac{2^{-2b}}{12}.$$

- In both cases, the autocovariance sequence satisfies

$$c_{ee}(n) = \sigma_e^2 \delta(n),$$

indicating that there is no correlation between $e(n)$ and $e(n+l)$ for $l \neq 0$.

- At the output of a filter output with the impulse reponse $h(n)$ and the transfer function $H(z)$, the effect of $e(n)$ can be seen as a random process $f(n)$ (see Appendix A for details) with the mean and variance (power), respectively, given by

$$m_f = m_e \sum_{n=0}^{\infty} h(n) = m_e H(e^{j0})$$

and

$$\sigma_f^2 = \sigma_e^2 \sum_{n=0}^{\infty} h^2(n) = \frac{\sigma_e^2}{2\pi} \int_{-\pi}^{\pi} |H(e^{j\omega})|^2 d\omega.$$

SIGNAL-TO-NOISE RATIO

- If the power of the incoming signal is σ_x^2, then the signal-to-noise ratio (SNR) is

$$\frac{\sigma_x^2}{\sigma_e^2} = (12 \cdot 2^{2b})\sigma_x^2.$$

- On the logarithmic scale,

$$\text{SNR} = 10\log_{10}(\frac{\sigma_x^2}{\sigma_e^2}) = 6.02b + 10.79 + 10\log_{10}(\sigma_x^2).$$

- The signal entering the A/D-conveter must be scaled such that with high probability $|x(n)| < 1$.

- Often the scaling is performed such that $4\sigma_x = 1$, giving

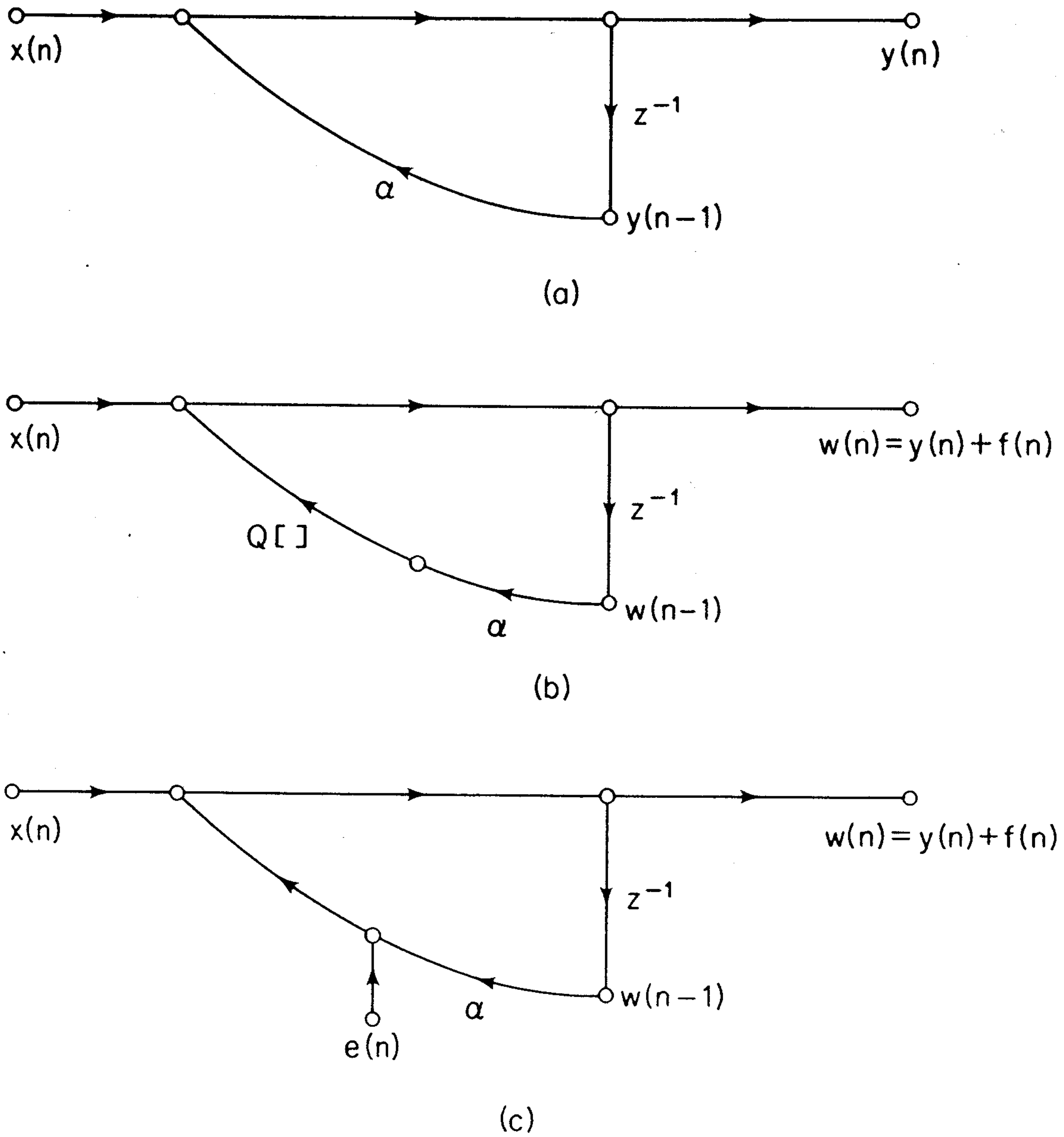
$$\text{SNR} = 6b - 1.24 \quad \text{dB}.$$

- SNR ≥ 80 requires $b = 14$ bits.

- To increase SNR by 6 dB, we need one more bit.

MULTIPLICATION ROUNDOFF ERROR: SIMPLE EXAMPLE

- Consider a simple first-order filter with transfer function $H(z) = 1/(1 - \alpha z^{-1})$. The following figure gives several flow graphs for this system. (If two signals enter a dot, it means that we add those signals. An arrow and α means multiplication by α.)



(a)

(b)

(c)

Fig. 9.7 Flow graphs for a first-order IIR system: (a) ideal linear system; (b) nonlinear system; (c) statistical model for fixed-point roundoff noise.

DIFFERENT MODELS

- Figure (a) depicts the ideal system described by the difference equation $y(n) = x(n) + \alpha y(n-1)$.

- Figure (b) shows the practical system. If the data sample $w(n-1)$ has a $(1+b)$-bit representation and α has a $(1+a)$-bit representation, then $\alpha w(n-1)$ has a $(1+b+a)$-bit representation. This number must be rounded or truncated to a $(1+b)$-bit number in order to avoid the number of bits for the data representation from growing. $Q[\ ]$ denotes this operation and has the same meaning as in the case of the A/D-conversion considered earlier.

- In Figure (c), the same system is depicted with the effect of quantizer being represented by the additive noise source

$$e(n) = Q[\alpha w(n-1)] - \alpha w(n-1).$$

ASSUMPTIONS

- Again the following assumptions are made:

  1. The error sequence $e(n)$ is a white-noise sequence.
  2. The error sequence has a uniform distribution over one quantization interval (see transparency 15).
  3. The error sequence is uncorrelated with the input $x(n)$ and $\alpha w(n-1)$.

- These assumptions are valid when the input signal as well as $w(n-1)$ vary from sample to sample in a sufficiently complex manner.

- It has been experimentally observed that in the case where the input signal $x(n)$ itself is a white-noise process, these assumptions apply extremely well. However, if $x(n)$ is a sinusoidal or a sum of sinusoidals these assumtions are not valid. In this case, $e(n)$ is also a sum of sinusoidals.

- These facts are considered in more details in the course "System Level DSP Algorithms".

MULTIPLICATION ROUNDOFF ERROR AT THE FILTER OUTPUT

- **For rounding**, $e(n)$ satisfies $-\frac{1}{2}2^{-b} < e(n) \leq \frac{1}{2}2^{-b}$, $m_e = 0$, and $\sigma_e^2 = 2^{-2b}/12$.

- **For truncation**, $e(n)$ satisfies $2^{-b} < e(n) \leq 0$, $m_e = \frac{1}{2}2^{-b}$, and $\sigma_e^2 = 2^{-2b}/12$.

- If $y(n)$ denotes the output signal of the ideal system, then the actual output is $w(n) = y(n) + f(n)$, where $f(n)$ represents the output error due to the noise source $e(n)$. If $h_e(n)$ is the impulse response from the input of $e(n)$ to the overall filter output, then the mean and variance (power) of $f(n)$ are, respectively, given by

$$m_f = m_e \sum_{n=0}^{\infty} h_e(n)$$

and

$$\sigma_f^2 = \sigma_e^2 \sum_{n=0}^{\infty} h_e^2(n).$$

- $h_e(n)$ is the same as the impulse response of the overall filter, that is, $h_e(n) = \alpha^n u(n)$, giving

$$\sigma_f^2 = \sigma_e^2/(1-\alpha^2) = \frac{1}{12}2^{-2b}/(1-\alpha^2).$$

MULTIPLICATION ROUNDOFF ERROR: SECOND-ORDER FILTER

- Consider a second-order filter with one complex pole pair at $z = re^{\pm j\theta}$. This system is represented by the following diffrerence equation:

$$y(n) = x(n) + 2r\cos\theta y(n-1) - r^2 y(n-2).$$

- With rounding of products, we obtain the nonlinear difference equation

$$w(n) = x(n) + Q[2r\cos\theta w(n-1)] - Q[r^2 y(n-2)].$$

- Since there are two multiplications, two noise sources are introduced as depicted in the following figure, denoted by $e_1(n)$ and $e_2(n)$. Since $e_1(n)$ and $e_2(n)$ are uncorrelated, there are two output errors due to these noise sources, denoted by $f_1(n)$ and $f_2(n)$.

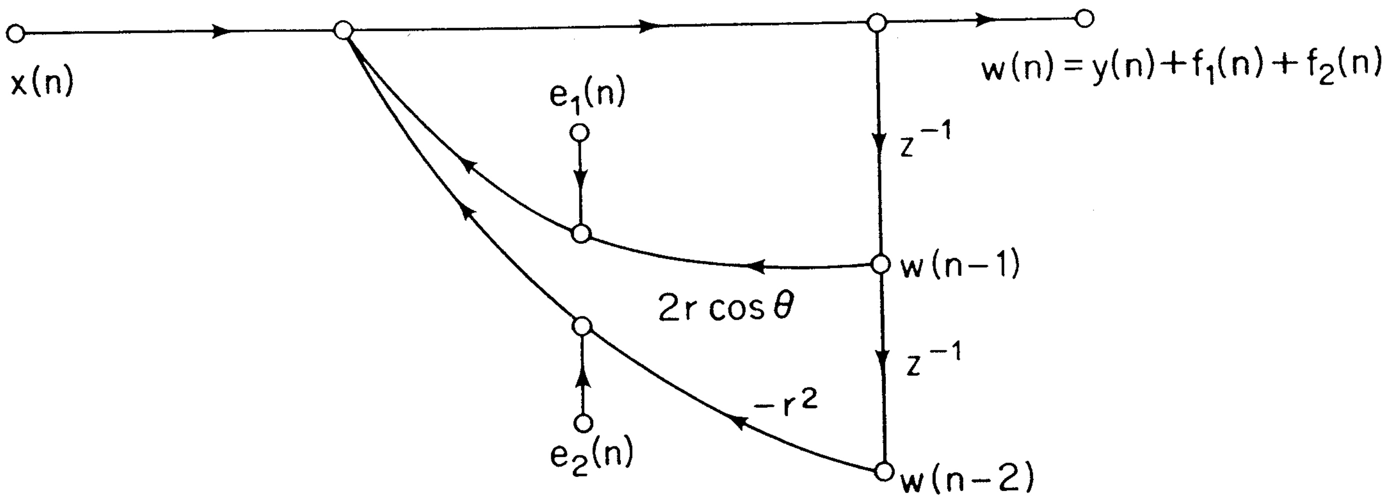


Fig. 9.8 Statistical model for fixed-point roundoff noise in a second-order IIR system.

- As a consequence of this fact, the overall output noise variance is

$$\sigma_f^2 = \sigma_{f_1}^2 + \sigma_{f_2}^2,$$

where

$$\sigma_{f_1}^2 = \sigma_e^2 \sum_{n=0}^{\infty} h_1^2(n)$$

and

$$\sigma_{f_2}^2 = \sigma_e^2 \sum_{n=0}^{\infty} h_2^2(n)$$

with $h_1(n)$ and $h_2(n)$ denoting the unit sample response from the noise source inputs.

- For this example,

$$h_1(n) = h_2(n) = \frac{1}{\sin\theta} r^n \sin[(n+1)\theta]u(n),$$

giving

$$\sum_{n=0}^{\infty} h_1^2(n) = \sum_{n=0}^{\infty} h_2^2(n) = [\frac{1+r^2}{1-r^2}][\frac{1}{r^4+1-2r^2\cos 2\theta}].$$

- Based on the above facts, we finally arrive at

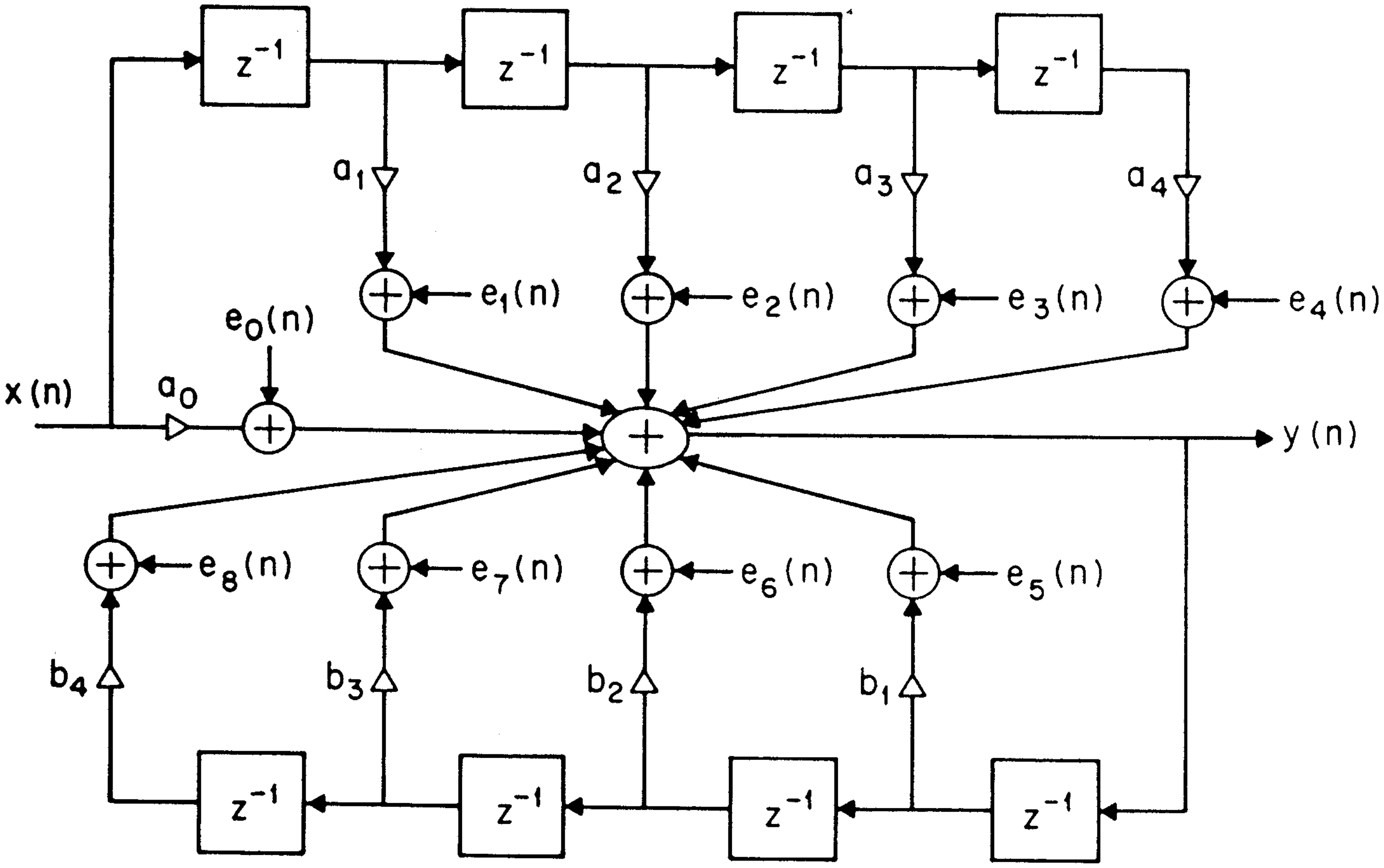
$$\sigma_f^2 = \frac{2}{12}2^{-2b}[\frac{1+r^2}{1-r^2}][\frac{1}{r^4+1-2r^2\cos 2\theta}].$$

GENERAL FILTER STRUCTURE

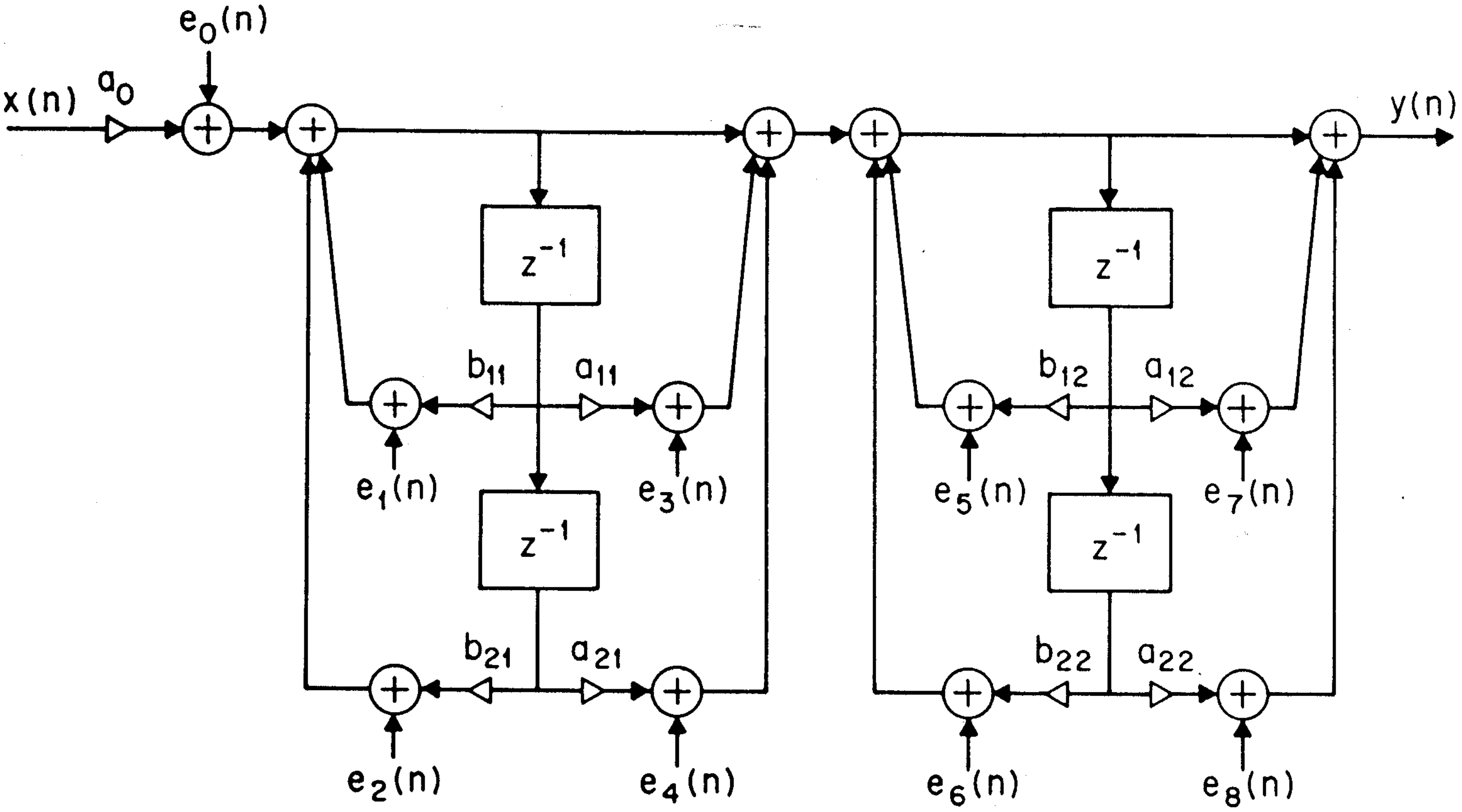
- Add after each multiplier an error source $e(n)$.
- If $h_e(n)$ is the impulse response from the input of $e(n)$ to the overall filter output, then the output noise variance due to the error source $e(n)$ is given by

$$\sigma_f^2 = \sigma_e^2 \sum_{n=0}^{\infty} h_e^2(n).$$

- The overall output noise is the direct sum of the output noise variances caused by the individual error sources.
- Direct-form structure:

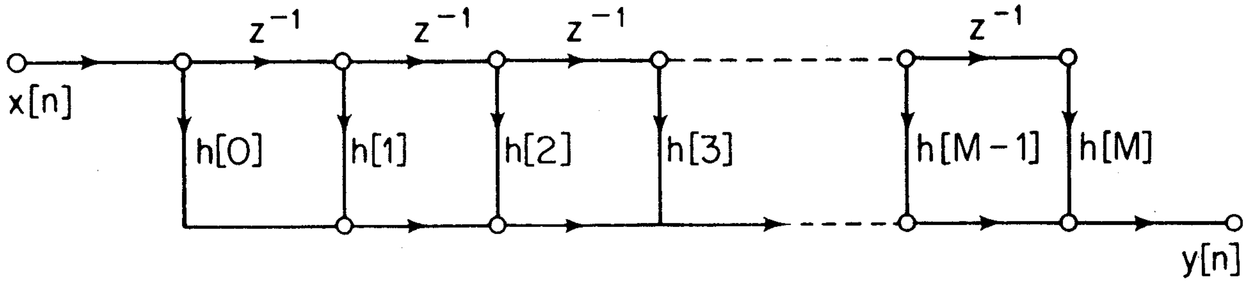


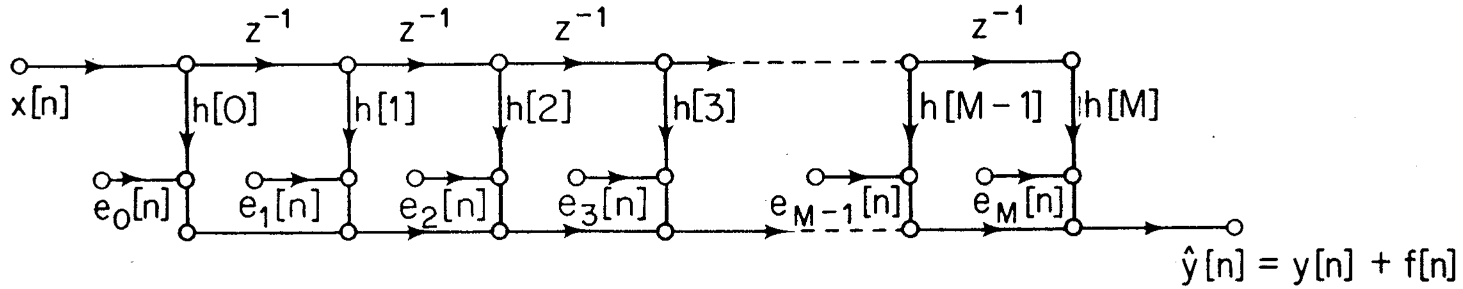
CASCADE-FORM STRUCTURE



DIRECT-FORM FIR FILTER

- The following figures depict both the infinite-precision model and linear noise model for a direct-form FIR filter of order M.





- $\sigma_f^2 = (M+1)2^{-2b}/12$

- If a double length accumulator is available, then there is only a single error source at the filter output and $\sigma_f^2 = 2^{-2b}/12$. In this case, the multiplication results are added using the double precision arithmetic and the final result is rounded or truncated. This is true for most signal processors.

FILTER SCALING

- The purpose of scaling a digital filter in proper manner is two-fold.

- First, in order to avoid overflows (jumps over the range $[-1,\ 1-2^{-b}]$), it is required that the data samples inside the filter are in this range **before multipliers** for two's complement arithmetic.

- Second, in order to reduce the output noise due to the multiplication roundoff errors, it is desired to keep the signal values inside the filter as high as possible.

- There exist several scaling norms making compromizes between the probability of overflows and the value of the output noise.

- We start by the definitions of these scaling norms.

- Then, several examples are included illustrating the fact that keeping the signal levels high inside the filter results in a small output noise level.

- We concentrate on cascade-form structures since for them the output noise is significantly lower than for the direct-form structures considered in Part I of these lecture notes.

SIMPLE EXAMPLE

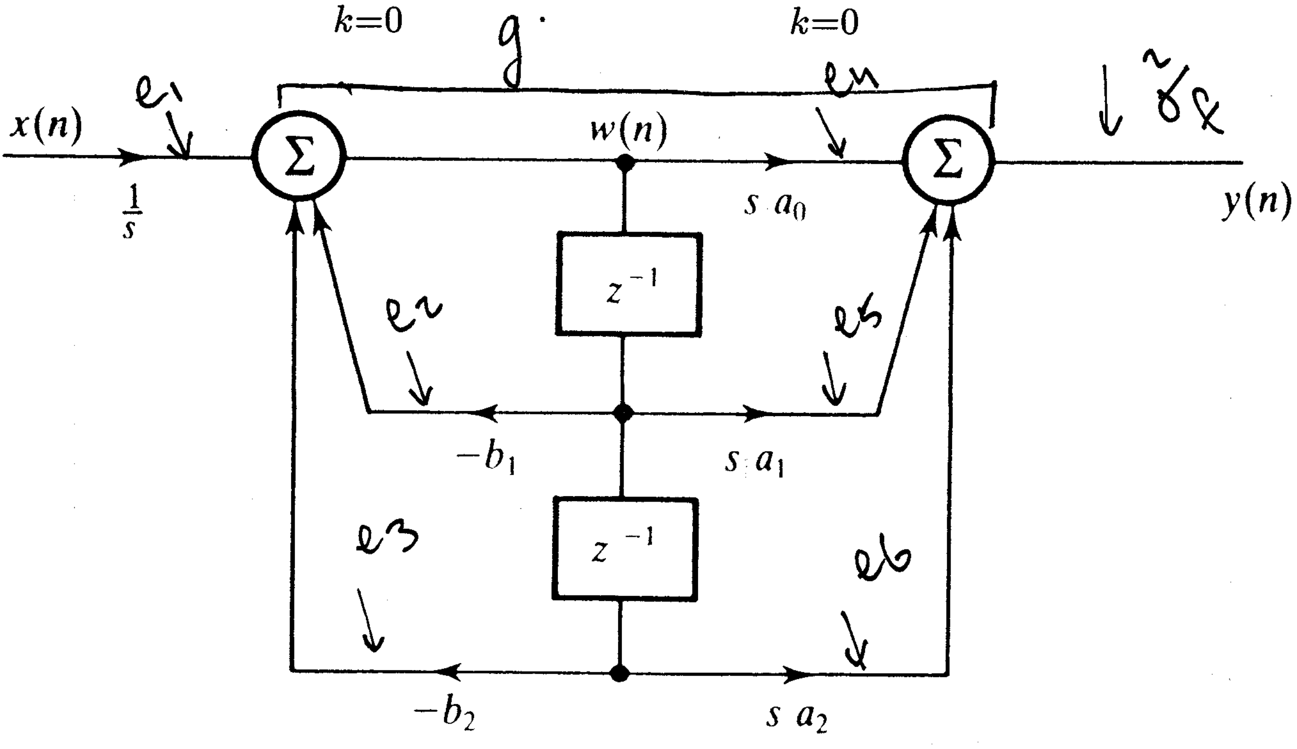
- Consider the following filter structure, where the scaling multiplier $1/s$ is added at the filter input, whereas the feedforward coefficients are multiplied by s in order to keep the overall transfer function the same.

- The role of the scaling multiplier is to keep $w(n)$ within the range $[-1, 1)$ with high probability (no overflows) when the input signal $x(n)$ is also in this range.

- The transfer function from the input to $w(n)$ is $E(z) = (1/s)F(z)$, where $F(z) = 1/(1 + b_1 z^{-1} + b_2 z^{-2})$. For $b_1 = -2r\cos\theta$ and $b_2 = r^2$ (pole pair is located at $z = r\exp(\pm j\theta)$), the unit sample response is $e(n) = (1/s)f(n)$, where $f(n) = \{r^n \sin[(n+1)\theta]/\sin\theta\}u(n)$.

- The relation between $x(n)$ and $w(n)$ is given by

$$w(n) = (1/s)\sum_{k=0}^{\infty} f(k)x(n-k) = \sum_{k=0}^{\infty} e(k)x(n-k)$$



WORST-CASE SCALING

- It is required that $|w(n)| \leq 1$ for all n and for all inputs, that is,

$$|w(n)| = (1/s)\sum_{k=0}^{\infty}|f(k)||x(n-k)| \leq 1.$$

- Since $|x(n-k)| \leq 1$, this implies that s has to satisfy

$$s = \sum_{k=0}^{\infty}|f(k)|.$$

- In other words, the overall impulse response $e(n) = (1/s)f(n)$ satisfies

$$\sum_{k=0}^{\infty}|e(k)| = 1$$

- $w(n)$ achieves the value $\sum_{k=0}^{\infty}|e(k)|$ if for $e(k) = f(k)/s$ positive, the corresponding $x(n-k) = 1$ and for $e(k)$ negative, the corresponding $x(n-k) = -1$.

- Similarly, $w(n)$ achieves the value $-\sum_{k=0}^{\infty}|e(k)|$ if for $e(k)$ positive, $x(n-k) = -1$ and for $e(k)$ negative, $x(n-k) = 1$.

- In this case, there are no overflows at all provided that $|x(n)| \leq 1$.

- In many cases, this is too pessimistic and the output SNR can be increased by increasing the probability of overflows.

L_∞ NORM

- In this case, it is required that

$$\max_{\omega\in[0,\ \pi]} |E(e^{j\omega})| = \max_{\omega\in[0,\ \pi]} (1/s)|F(e^{j\omega})| = 1$$

$$\Rightarrow \quad s = \max_{\omega\in[0,\ \pi]} |F(e^{j\omega})|$$

- We know that the response of a filter with transfer function $E(z)$ to a sinusoidal exitation $x(n) = A\cos(n\omega_0 + \phi)$ is given by

$$y(n) = A|E(e^{j\omega_0})|\cos([n - \tau_p(\omega_0)]\omega_0 + \phi).$$

- Therefore, if $|x(n)| \leq 1$, that is, $A \leq 1$, then the oscillation amplitude of the output signal satisfies $A|E(e^{j\omega_0})| \leq 1$ for $0 \leq \omega_0 \leq \pi$.

- This means that for the L_∞ norm scaling or the so-called peak scaling, a single sinusoidal signal causes no overflows.

- In many cases, L_∞ norm is a good selection especially for IIR filters.

L_2 NORM

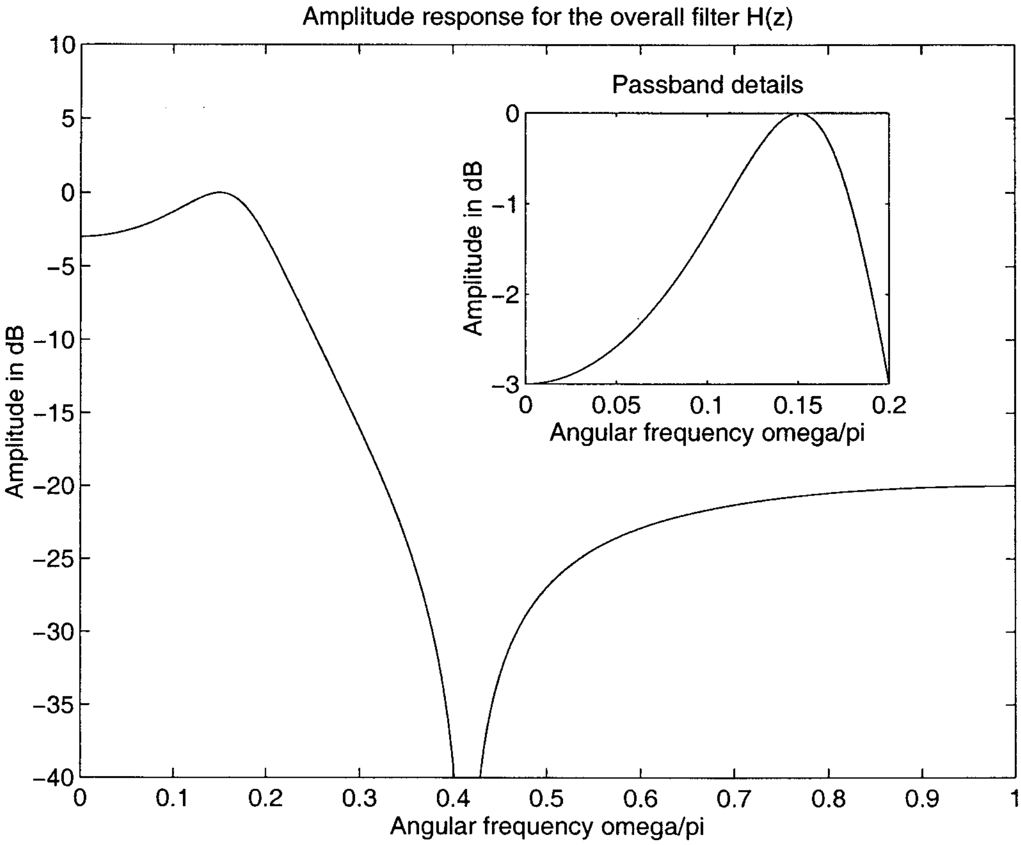
- In this case, it is required that

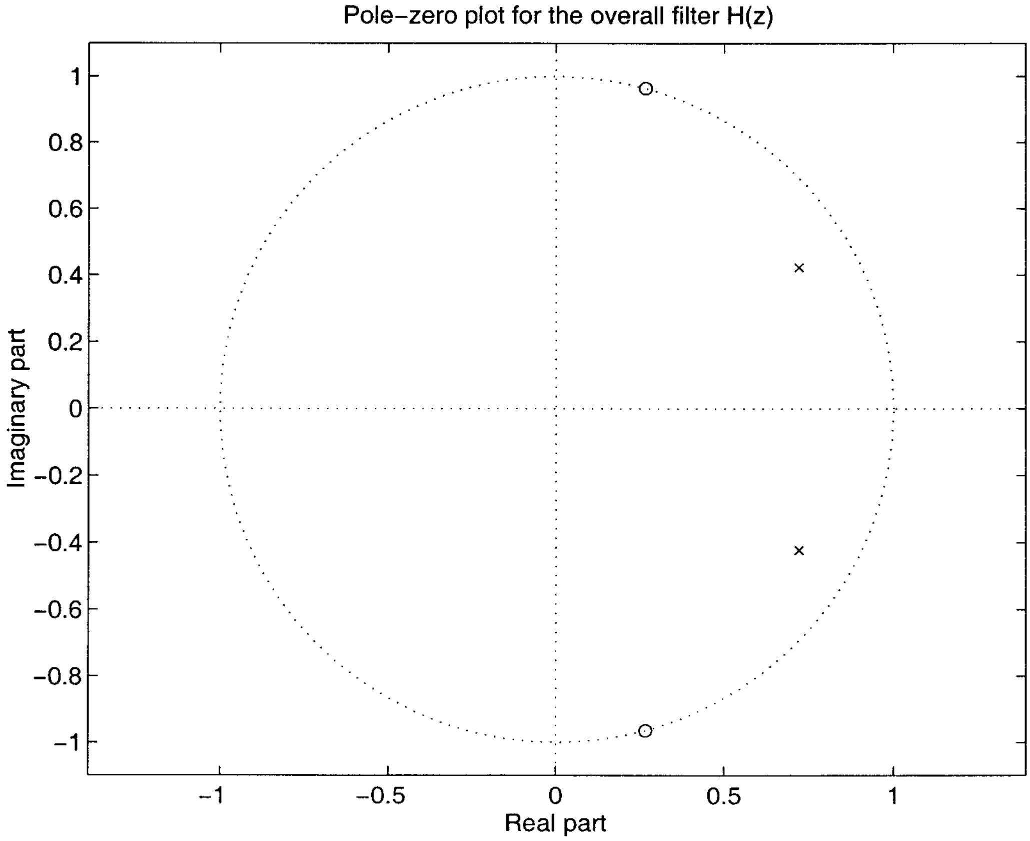
$$\sum_{k=0}^{\infty} e^2(k) = (1/s)^2 \sum_{k=0}^{\infty} f^2(k) = 1$$

$$\Rightarrow \quad s = \sqrt{\sum_{k=0}^{\infty} f^2(k)}.$$

- This scaling can be used when $x(n)$ is a random signal.
- The probability of overflows is the highest among the three scaling norms.

SIMPLE EXAMPLE: SECOND-ORDER ELLIPTIC FILTER





SIMPLE EXAMPLE: SECOND-ORDER ELLIPTIC FILTER

- $b_1 = -1.4410226$, $b_2 = 0.69785000$, $a_1 = a_3 = 0.12391452$, $a_2 = -0.06600919$.

- The pole pair is located at $z = r\exp(\pm j\theta)$ with $r = 0.83537416$ and $\theta = 0.16889666$, whereas the zeros lie on the unit circle at $\omega = \pm 0.41418240$.

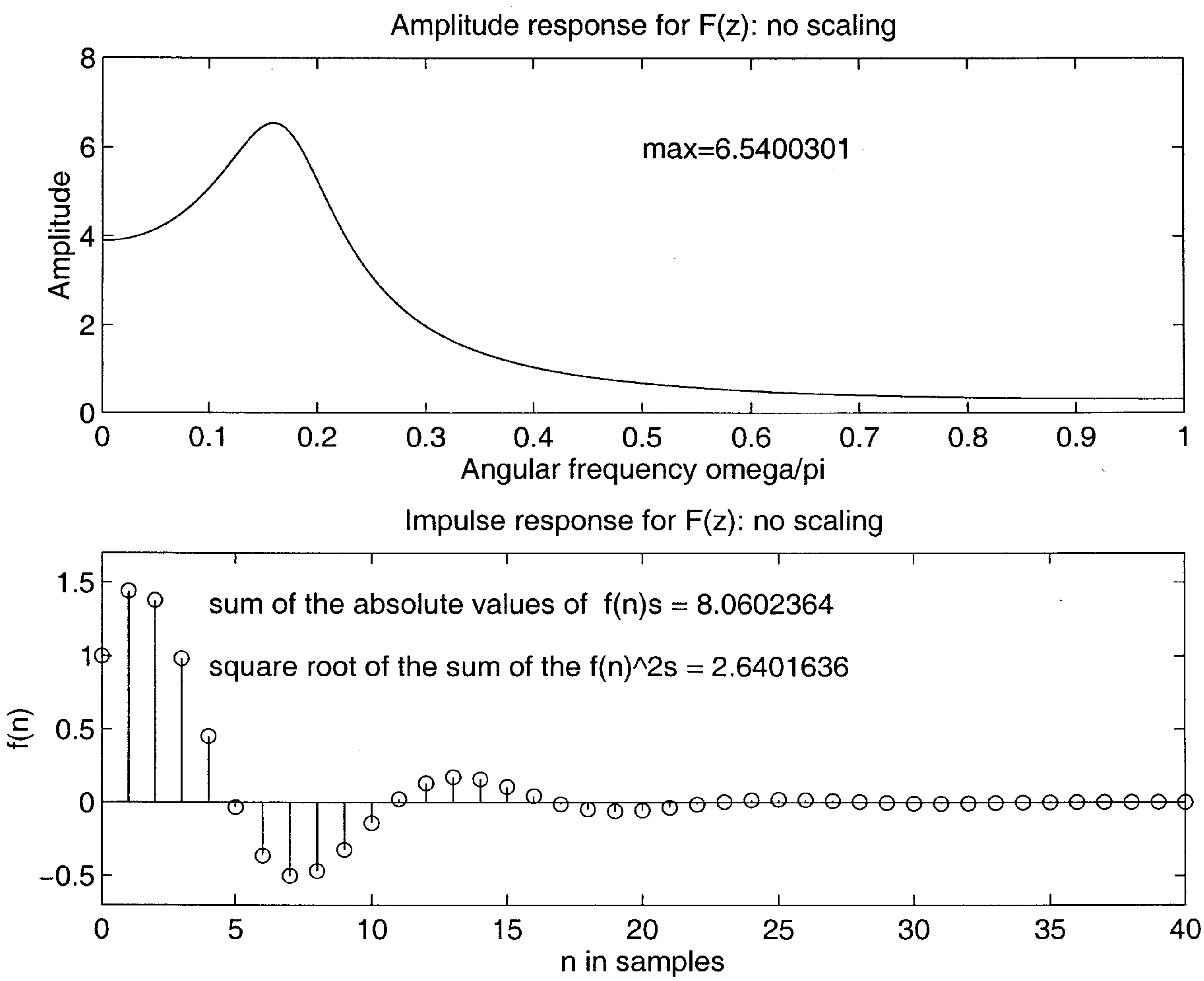
- In this case (see the figures of the next page),

$$s = \begin{cases} 8.0602364 & \text{for worst-case} \\ 6.5400301 & \text{for } L_\infty \\ 2.6401636 & \text{for } L_2. \end{cases}$$

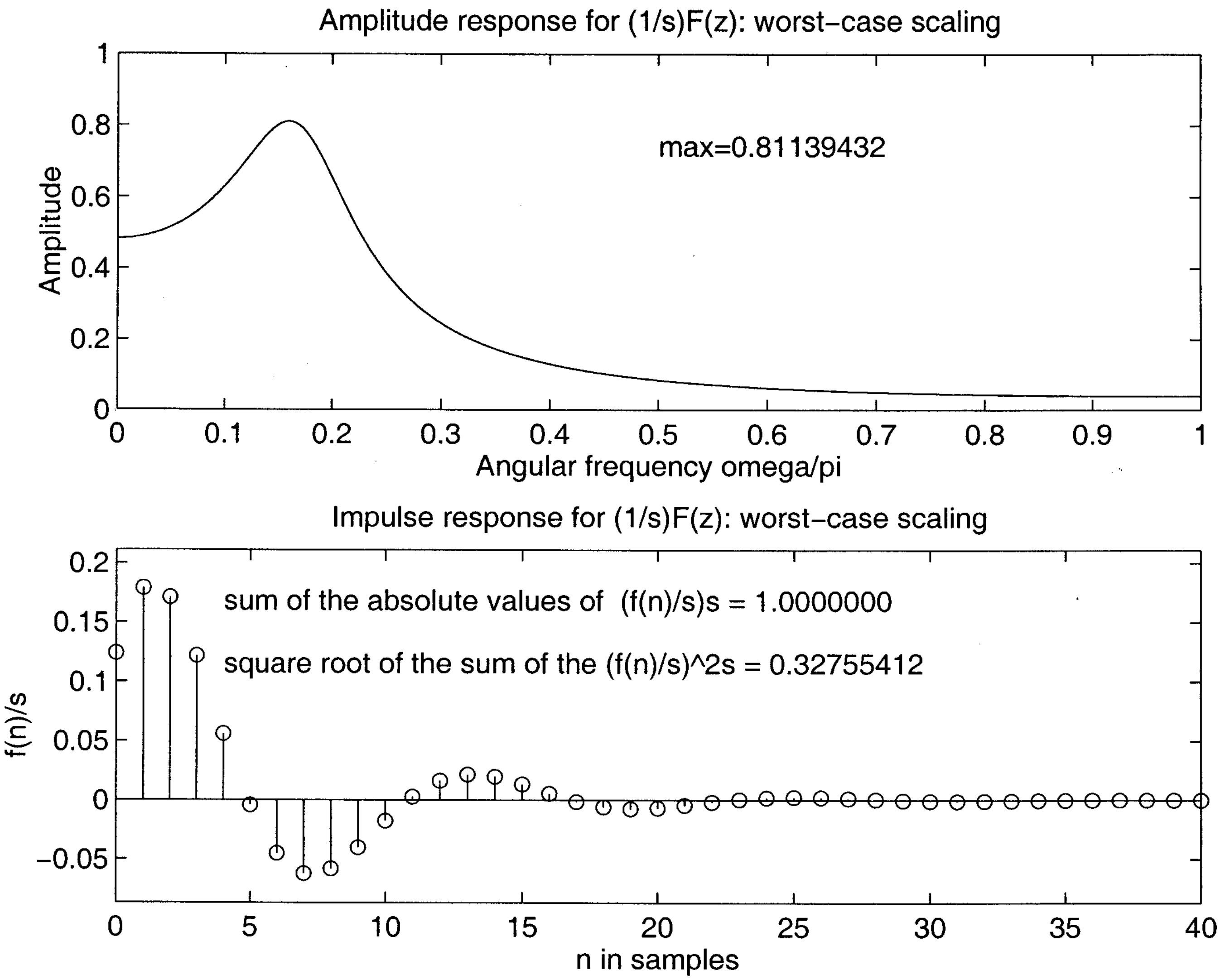
- Pages 28, 29, and 30 show the the amplitude and impulse responses for the resulting $(1/s)F(z)$ in the three cases.

- Pages 42 and 43 show the matlab code for generating the figures.

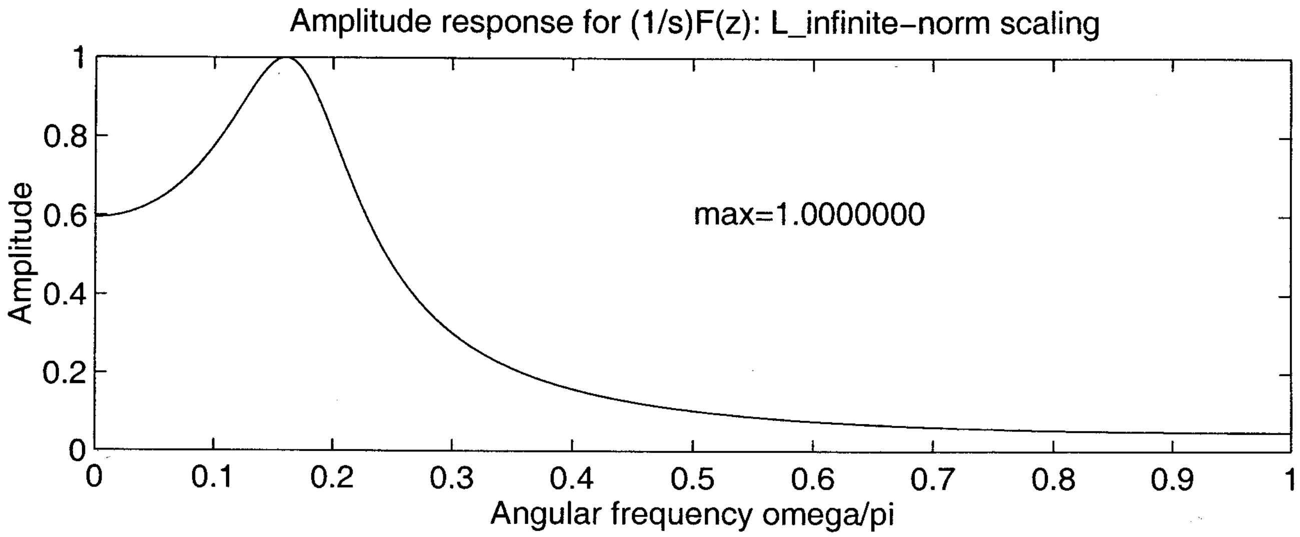
AMPLITUDE AND IMPULSE RESPONSES FOR $F(z)$

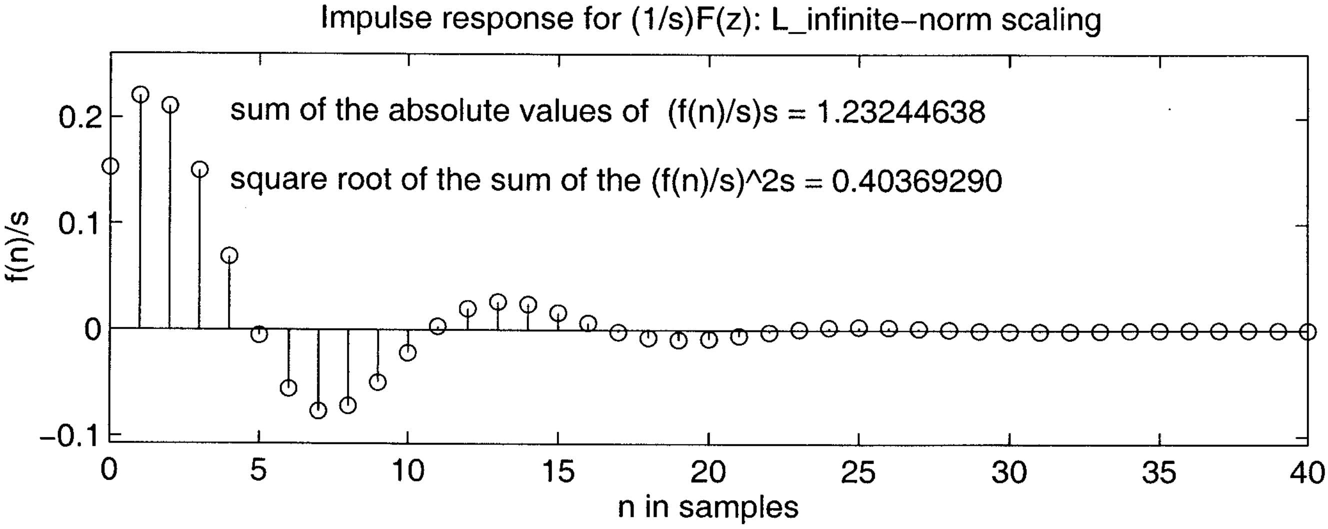


AMPLITUDE AND IMPULSE RESPONSES FOR $(1/s)F(z)$: WORST-CASE SCALING

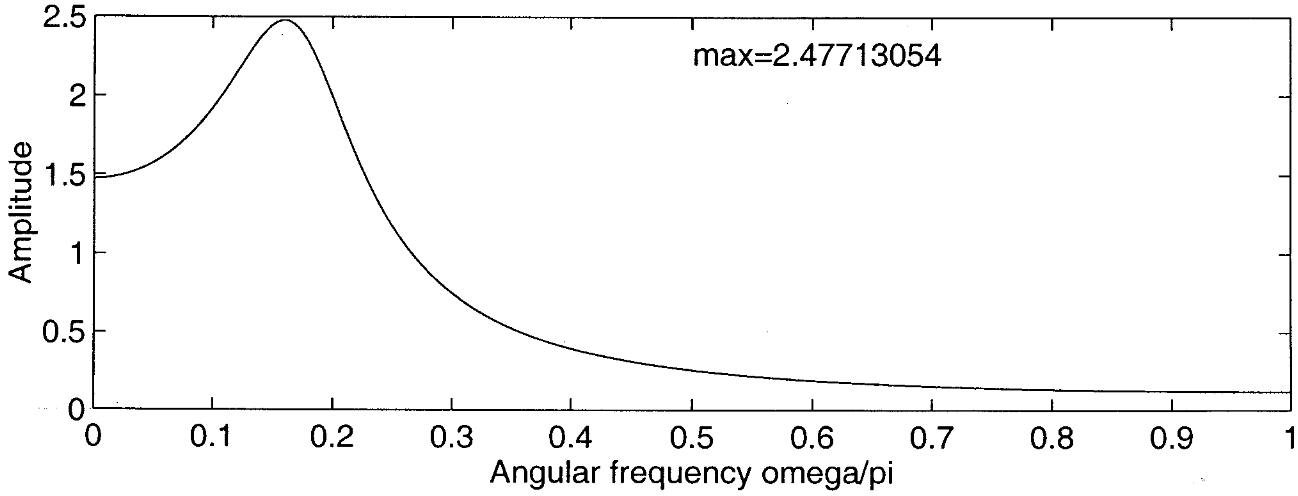


AMPLITUDE AND IMPULSE RESPONSES FOR $(1/s)F(z)$: L_∞-norm SCALING



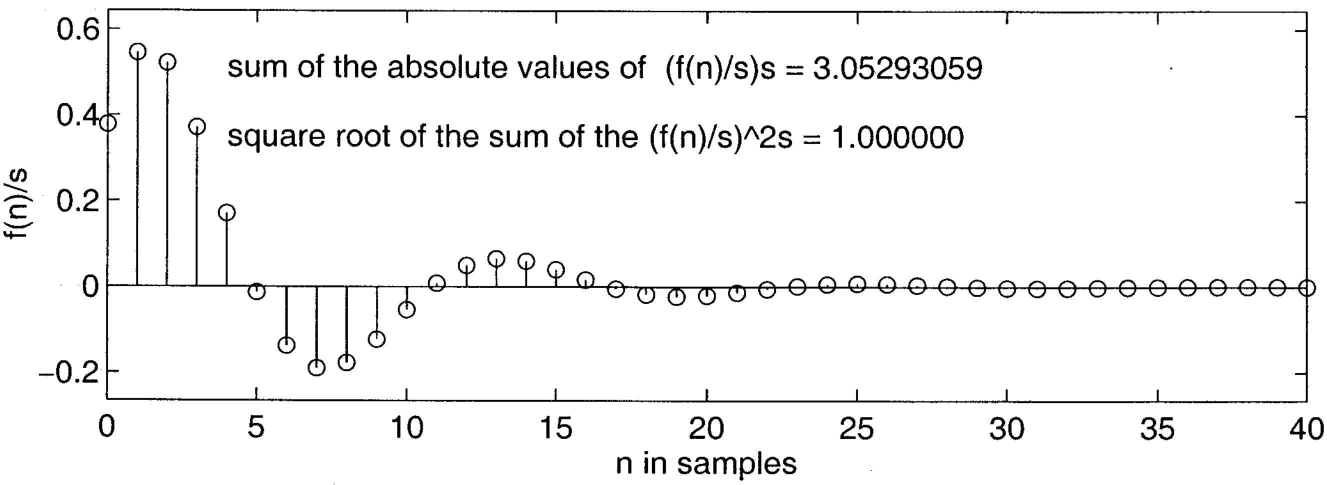


AMPLITUDE AND IMPULSE RESPONSES FOR $(1/s)F(z)$: L_2-norm SCALING



Amplitude response for (1/s)F(z): L_2−norm scaling

max=2.47713054



Impulse response for (1/s)F(z): L_2−norm scaling

sum of the absolute values of (f(n)/s)s = 3.05293059

square root of the sum of the (f(n)/s)^2s = 1.000000

OUTPUT NOISE VARIANCE DUE TO THE MULTIPLICATION ROUNDOFF ERRORS

- Assuming that after each multiplier there is an error source with variance $\sigma_e^2 = 2^{-2b}/12$, the output noise variance for the filter of page 29 is given by

$$\sigma_f^2 = \sigma_e^2[3\sum_{n=0}^{\infty} g^2(n) + 3].$$

- Here, $g(n)$ is the impulse response from the outputs of the multipliers $1/s$, $-b_1$, and $-b_2$ to the filter output. After multipliers sa_0, sa_1, and sa_2, the impulse response is simply an impulse.

- The transfer function corresponding to the impulse response is

$$G(z) = s(a_0 + a_1 z^{-1} + a_2 z^{-2})/(1 + b_1 z^{-1} + b_2 z^{-2}).$$

- For $b_1 = -2r\cos\theta$ and $b_2 = r^2$ (pole pair is located at $z = \exp(\pm j\theta)$),

$$g(n) = s(a_0 f(n) + a_1 f(n-1) + a_2 f(n-2)$$

where

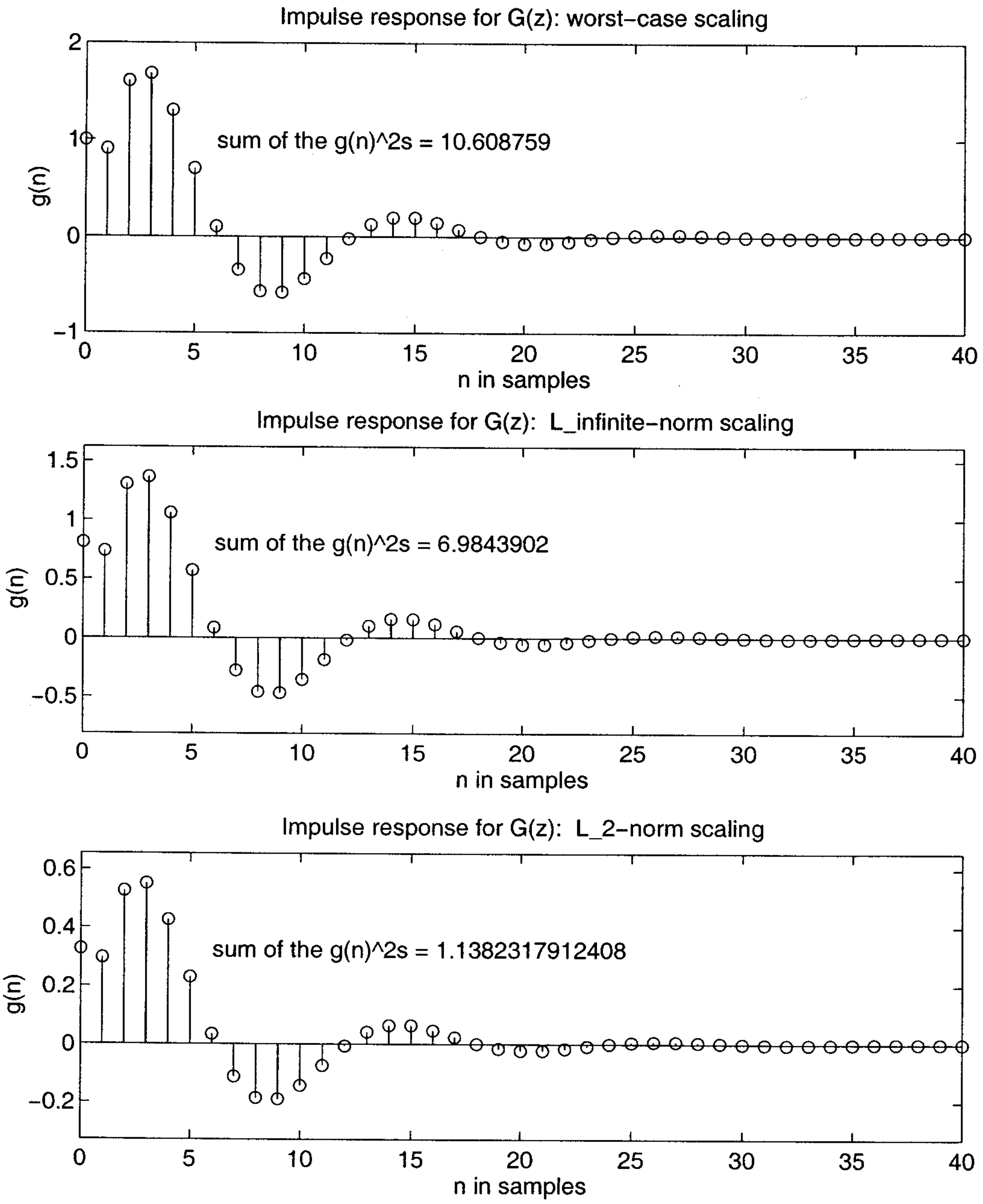
$$f(n) = \{r^n \sin[(n+1)\theta]/\sin\theta\}u(n).$$

- For the three scaling norms (see the figures on the next page),

$$\sigma_f^2 = \begin{cases} 34.82627621\sigma_e^2 & \text{for worst-case} \\ 23.95317083\sigma_e^2 & \text{for } L_\infty \\ 6.41469537\sigma_e^2 & \text{for } L_2. \end{cases}$$

- As expected, the noise is the highest for the worst-case scaling and the lowest for the L_2-norm scaling, whereas the L_∞-norm scaling is between these two cases.

IMPULSE RESPONSES $g(n)$ FOR THE THREE SCALING NORMS



Impulse response for G(z): worst-case scaling

Impulse response for G(z): L_infinite-norm scaling

Impulse response for G(z): L_2-norm scaling

```
%Simple example on scaling : Elliptic filter with 3-dB passband ripple and 20-dB stopband
%ripple. Passband edge=0.2pi, whereas the stopband edge is as narrow as possible to meet the
%stopband criteria
%Tapio Saram"aki 2.2.1996
%Can be found in SUN’s: ~ts/matlab/dsp/scasim.m
%
[B,A] = ellip(2,3.,20.,.2);
[HH,w]=freqz(B,A,8*1024);
figure(1)
plot(w/pi,20*log10(abs(HH)));axis([0 1 -40 10]);
title(‘Amplitude response for the overall filter H(z)’)
ylabel(‘Amplitude in dB’);xlabel(‘Angular frequency omega/pi’); hold on;
axes(‘position’, [.5 .55 .3 .3]);plot(w/pi,20*log10(abs(HH)));axis([0 .2 -3. 0]);
title(‘Passband details’):ylabel(‘Amplitude in dB’);
xlabel(‘Angular frequency omega/pi’);hold off
figure(2)
zplane(B,A);title(‘Pole-zero plot for the overall filter H(z)’)

%Scaling transfer functions

FN1(1)=1;FD1=A;[H1,w]=freqz(FN1,FD1,8*1024);
%L_infinite-norm: s=
inff=max(abs(H1(1:8192)))
[hh1,t]=impz(FN1,FD1,501);
%Worst-case: s=
worf=sum(abs(hh1))
%L_2-norm: s=
l2f=sqrt(sum(hh1.*hh1))

%Amplitude and impulse responses; no scaling
figure(3)
subplot(2,1,1);plot(w/pi,abs(H1));title(‘Amplitude response for F(z): no scaling’)
ylabel(‘Amplitude’);xlabel(‘Angular frequency omega/pi’);text(0.5,6.,’max=6.5400301)
subplot(2,1,2);impz(hh1);title(‘Impulse response for F(z): no scaling’);
axis([0 40 -.7 1.7]);ylabel(‘f(n)’);xlabel(‘n in samples’);
text(4,.8*1.7,’sum of the absolute values of  f(n)s = 0.80602364’);
text(4,.55*1.7,’square root of the sum of the f(n)^2s = 2.6401636’)

%Scaled filters: divide the responses by the corresponding value of s
figure(4)
subplot(2,1,1);plot(w/pi,abs(H1/worf));
title(‘Amplitude response for (1/s)F(z): worst-case scaling’)
ylabel(‘Amplitude’);xlabel(‘Angular frequency omea/pi’);
text(0.5,6./worf,’max=0.81139432’)
subplot(2,1,2);impz(hh1/worf);title(‘Impulse response for (1/s)F(z): worst-case scaling’);
axis([0 40 -.7/worf 1.7/worf]);ylabel(‘f(n)/s’);xlabel(‘n in samples’);
text(4,.8*1.7/worf,’sum of the absolute values of  (f(n)/s)s = 1.0000000’);
text(4,.55*1.7/worf,’square root of the sum of the (f(n)/s)^2s = 0.32755412’)
figure(5)
subplot(2,1,1);plot(w/pi,abs(H1/inff));
```

```
title(‘Amplitude response for (1/s)F(z): L_infinite-norm scaling’)
ylabel(‘Amplitude’);xlabel(‘Angular frequency omega/pi’);
text(0.5,4./inff,’max=1.0000000’)
subplot(2,1,2);impz(hh1/inff);title(‘Impulse response for (1/s)F(z): L_infinite-norm scaling’);
axis([0 40 -.7/inff 1.7/inff]);ylabel(‘f(n)/s’);xlabel(‘n in samples’);
text(4,.8*1.7/inff,’sum of the absolute values of  (f(n)/s)s = 1.23244638’);
text(4,.55*1.7/inff,’square root of the sum of the (f(n)/s)^2s = 0.40369290’)
figure(6)
subplot(2,1,1);plot(w/pi,abs(H1/l2f));
title(‘Amplitude response for (1/s)F(z): L_2-norm scaling’)
ylabel(‘Amplitude’);xlabel(‘Angular frequency omega/pi’);
text(0.5,6./l2f,’max=2.47713054’)
subplot(2,1,2);impz(hh1/l2f);title(‘Impulse response for (1/s)F(z): L_2-norm scaling’);
axis([0 40 -.7/l2f 1.7/l2f]);ylabel(‘f(n)/s’);xlabel(‘n in samples’);
text(4,.8*1.7/l2f,’sum of the absolute values of  (f(n)/s)s = 3.05293059’);
text(4,.55*1.7/l2f,’square root of the sum of the (f(n)/s)^2s = 1.000000’)

%Noise
%First impulse response for the overall filter

GN1=B;GD1=A;
[gg,t]=impz(GN1,GD1,501);
sqg=(sum(gg.*gg))

%For each scaling norm, g(n) is obtained by multiplying gg by the corresponding
%value of s
figure(7)
subplot(2,1,1);impz(gg*worf);title(‘Impulse response for G(z): worst-case scaling ‘);
ylabel(‘g(n)’);xlabel(‘n in samples’);axis([0 40 -1. 2.]);
text(6,1.,’sum of the g(n)^2s = 10.608759’);
subplot(2,1,2);impz(gg*inff);title(‘Impulse response for G(z):  L_infinite-norm scaling’)
ylabel(‘g(n)’);xlabel(‘n in samples’);axis([0 40  -1.*inff/worf 2.*inff/worf]);
text(6,1.*inff/worf,’sum of the g(n)^2s = 6.9843902’);
figure(8)
subplot(2,1,1);impz(gg*12f);title(‘Impulse response for G(z):  L_2-norm scaling’);
ylabel(‘g(n)’);xlabel(‘n in samples’);axis([0 40 -1.*12f/worf 2.*12f/worf]);
text(6,1.*12f/worf,’sum of the g(n)^2s = 1.1382317912408’);

%Output noise variances: multiply sqg by s^2

%worst-case
noworst=3*(1+worf*worf*sqg)

%L_infinte-norm
noinfinite=3*(1+inff*inff*sqg)

%L_infinte-norm
no2=3*(1+12f*12f*sqg)
```

SATURATION ARITHMETIC

- The effect of a possible overflow can be killed in a finite time using the saturation arithmetic (see the following figure).

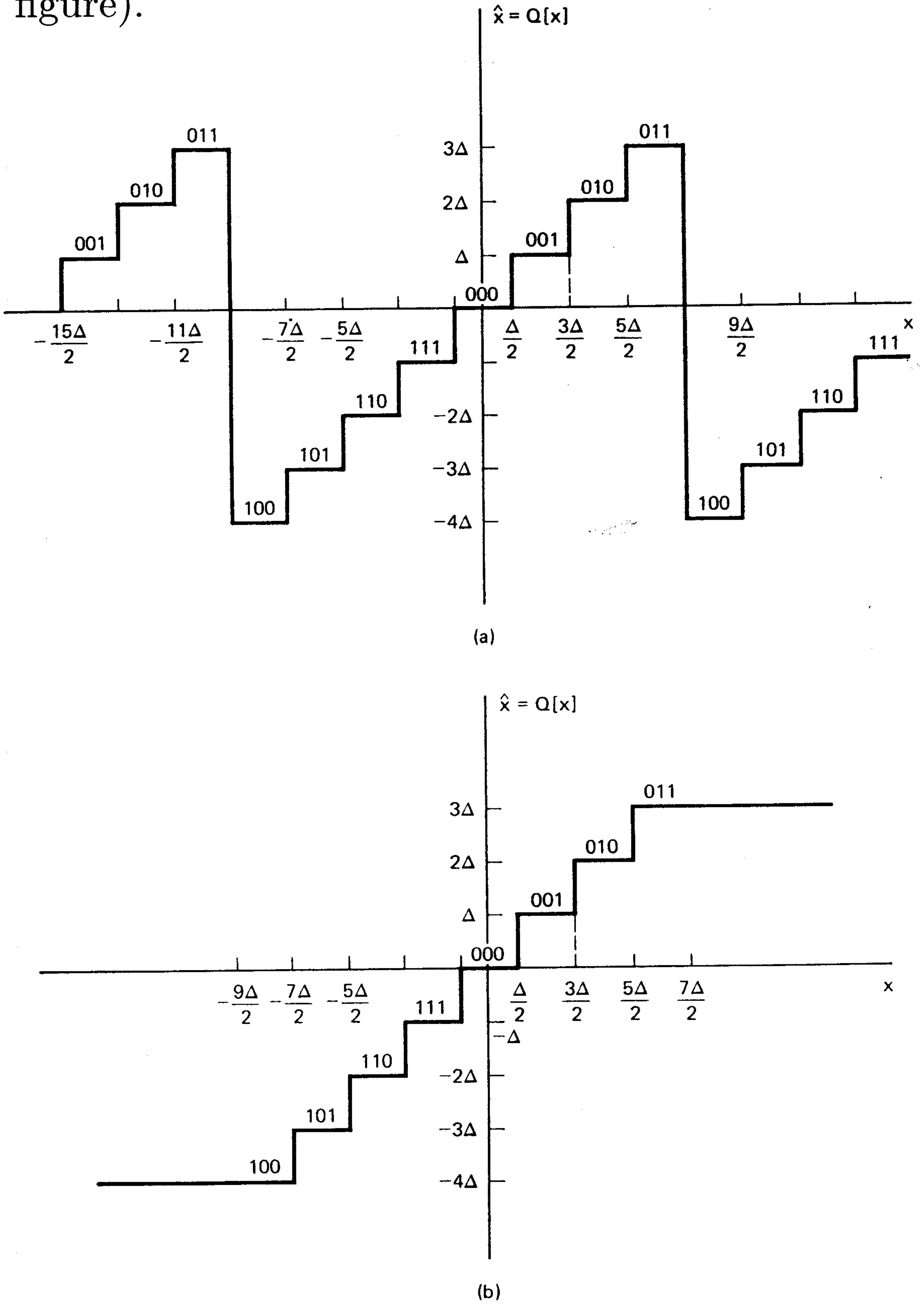
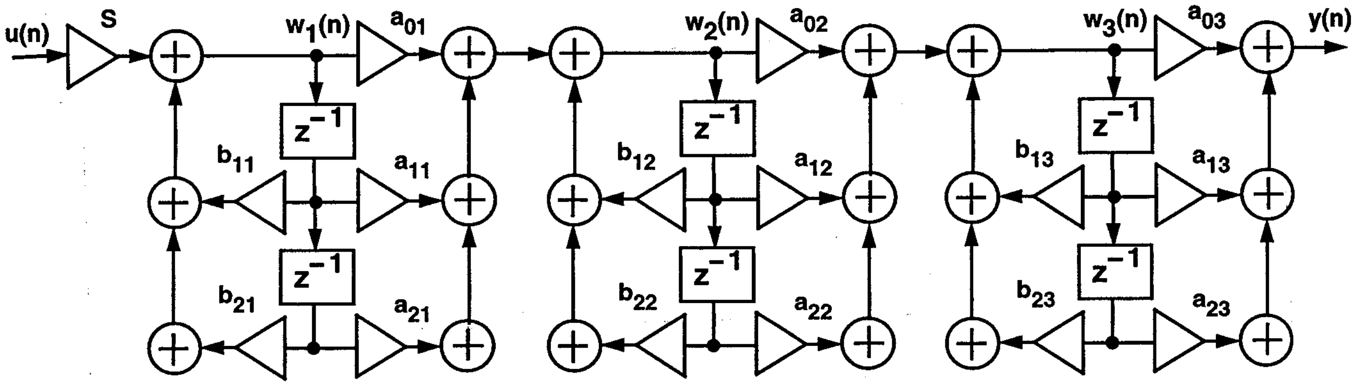


Figure 6.44 Two's-complement rounding. (a) Natural overflow. (b) Saturation.

ILLUSTRATIVE EXAMPLE ON SCALING: A CASCADE OF THREE SECOND-ORDER SECTIONS; A lowpass filter with $\omega_p = 0.4\pi$, $\omega_s = 0.6\pi$, $A_p = 0.5$ dB, $A_s = 80$ dB



- Coefficients of the unscaled filter:

$$S = 0.00811165$$

$$b_{11} = 1.165885, \quad b_{12} = 0.814467 \quad b_{13} = 0.564167$$

$$b_{21} = -0.417533, \quad b_{22} = -0.640952 \quad b_{23} = -0.883560$$

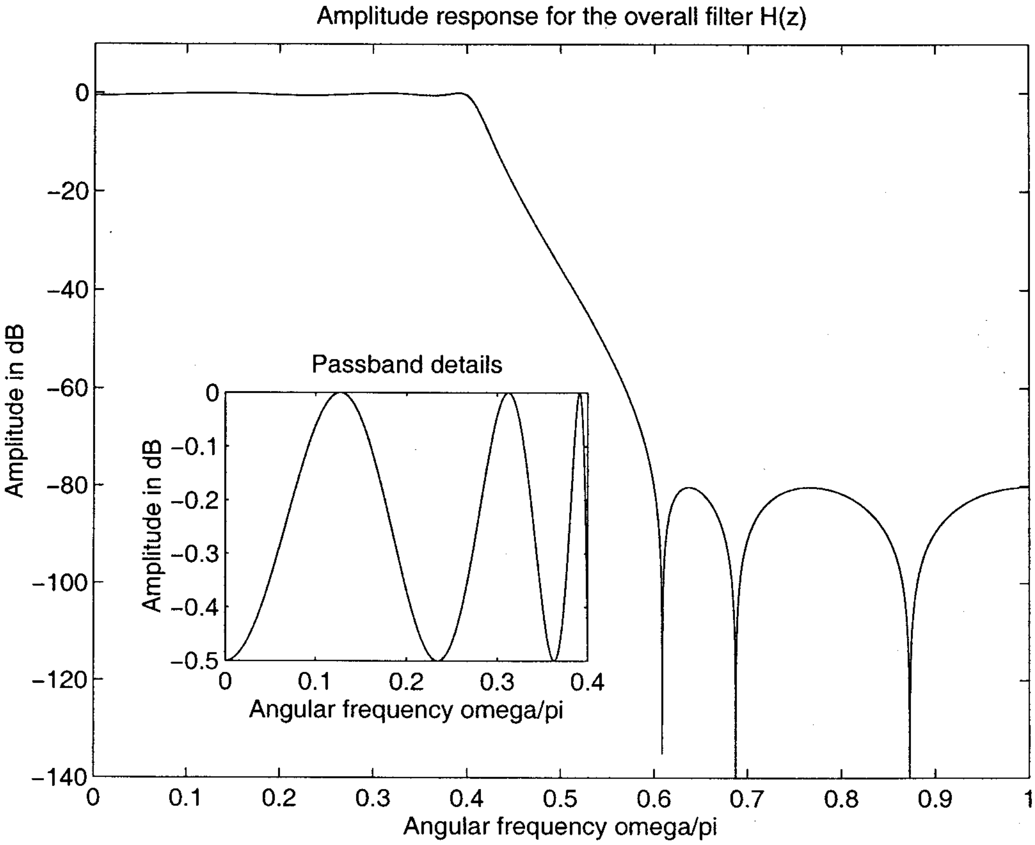
$$a_{0k} = a_{2k} = 1, \quad k = 1, 2, 3$$

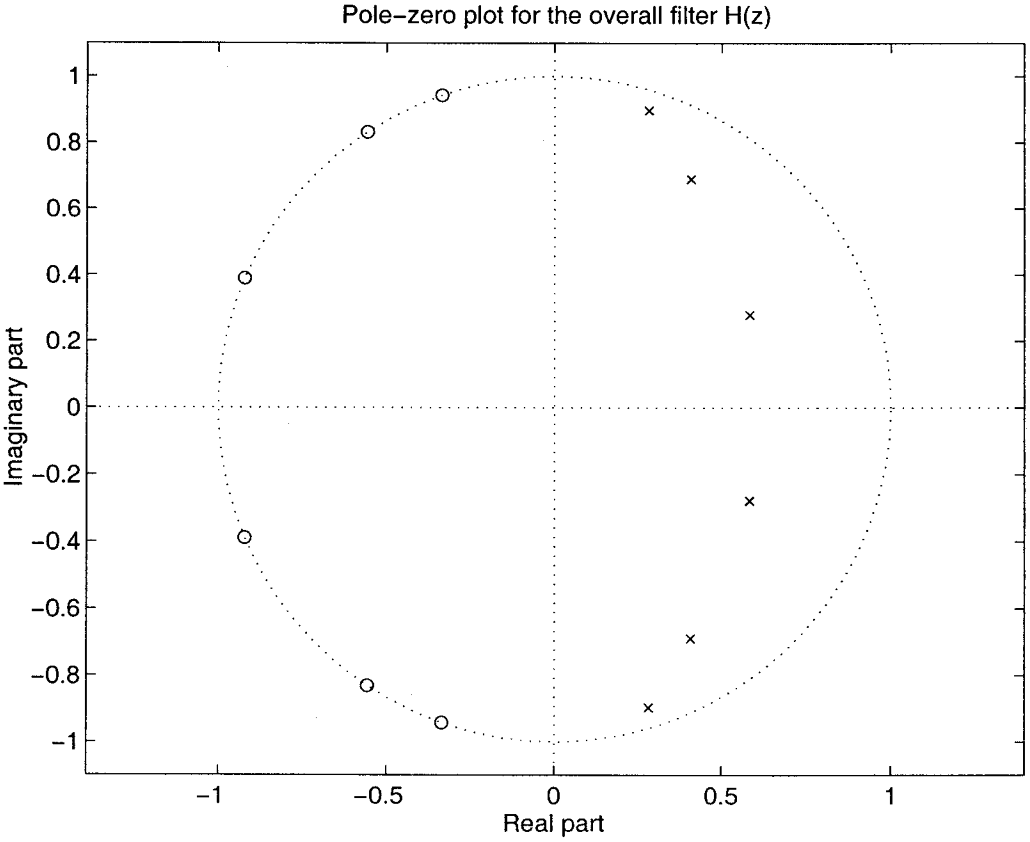
$$a_{11} = 1.842348, \quad a_{12} = 1.111859 \quad a_{13} = 0.671013$$

- The amplitude response and the pole-zero plot for this filter are shown on the next page.

- For two's complement arithmetic, the variables $w_1(n)$, $w_2(n)$, and $w_3(n)$ must be kept in the range $[-1, 1)$.

EXAMPLE ELLIPTIC FILTER





FILTER WITHOUT SCALING

- We denote by $F_k(z)$ and $f_k(n)$, $k = 1, 2, 3$ the transfer functions and impulse responses from the input to the variables $w_1(n)$, $w_2(n)$, and $w_3(n)$. The overall transfer function and the impulse response is denoted by $H(z)$ and $h(n)$, respectively.

$$\begin{aligned} F_1(z) &= S/B_1(z) \\ F_2(z) &= [SA_1(z)]/[B_1(z)B_2(z)] \\ F_3(z) &= [SA_1(z)A_2(z)]/[B_1(z)B_2(z)B_3(z)] \\ H(z) &= [SA_1(z)A_2(z)A_3(z)]/[B_1(z)B_2(z)B_3(z)], \end{aligned}$$

where for $k = 1, 2, 3$

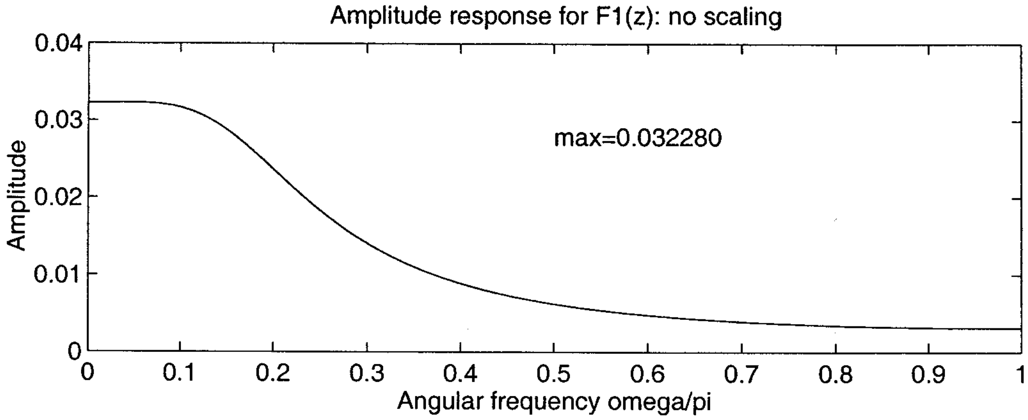
$$A_k(z) = a_{0k} + a_{1k}z^{-1} + a_{2k}z^{-2}$$

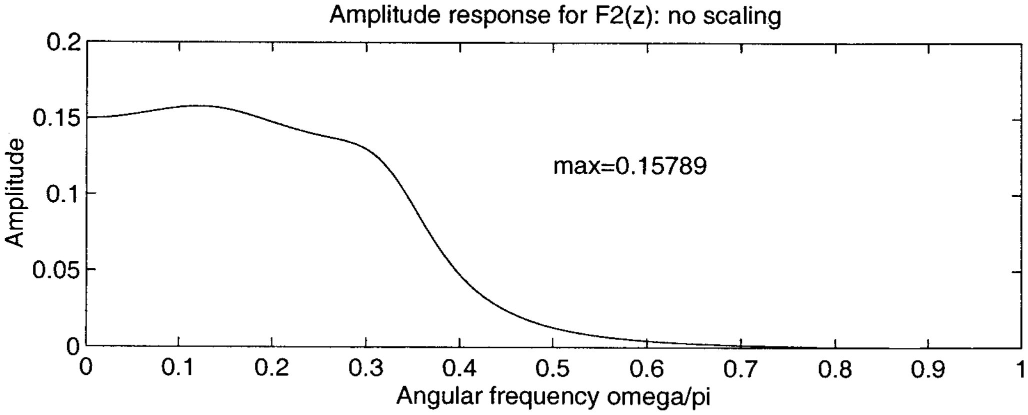
and

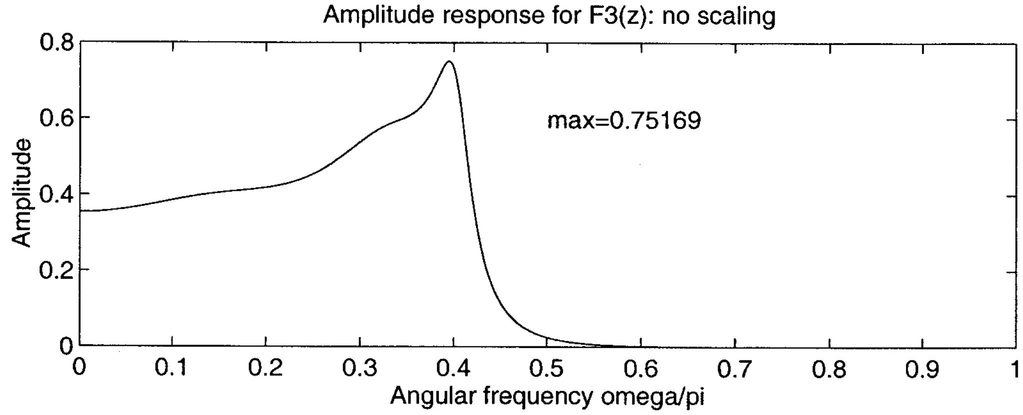
$$B_k(z) = 1 - b_{1k}z^{-1} - b_{2k}z^{-2}.$$

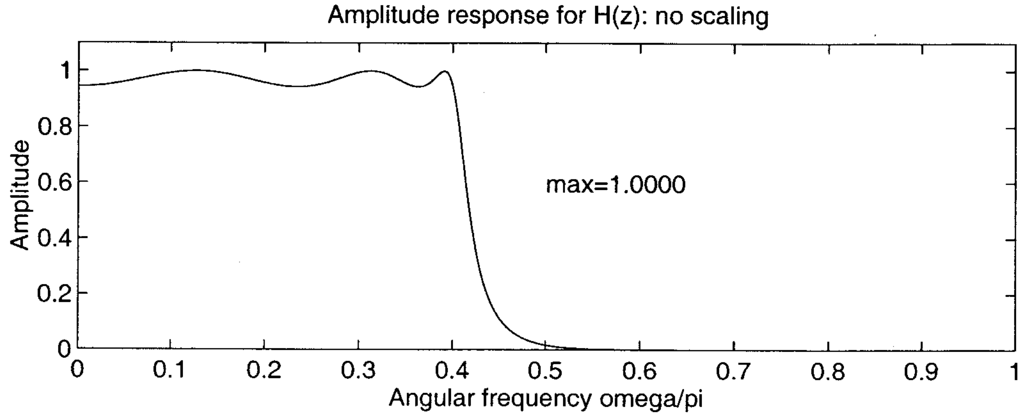
- The following two pages show the amplitude and impulse responses for the $F_k(z)$'s and $H(z)$.

UNSCALED FILTER

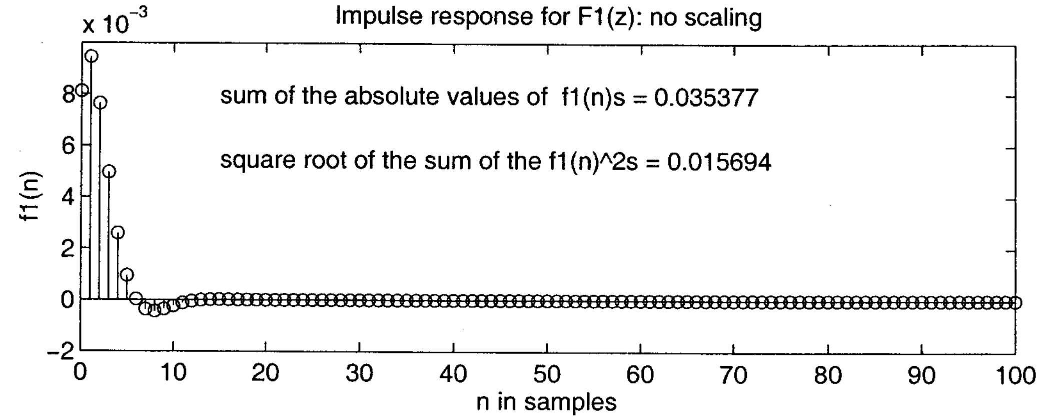


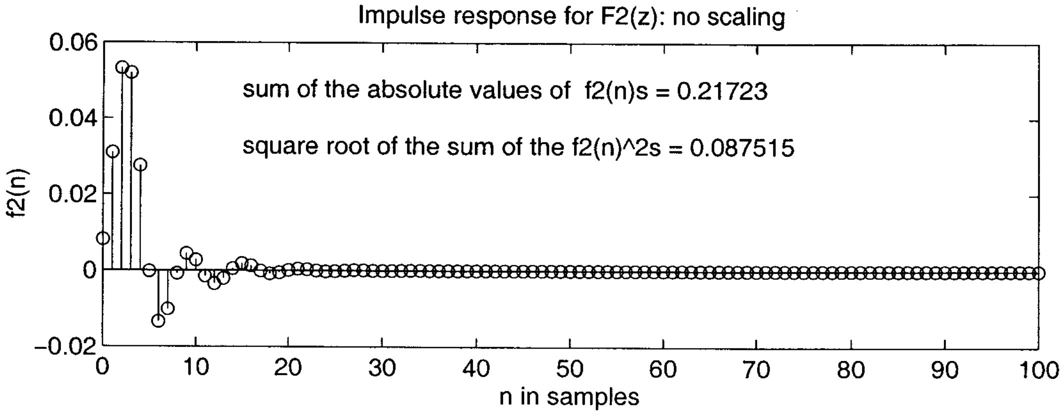


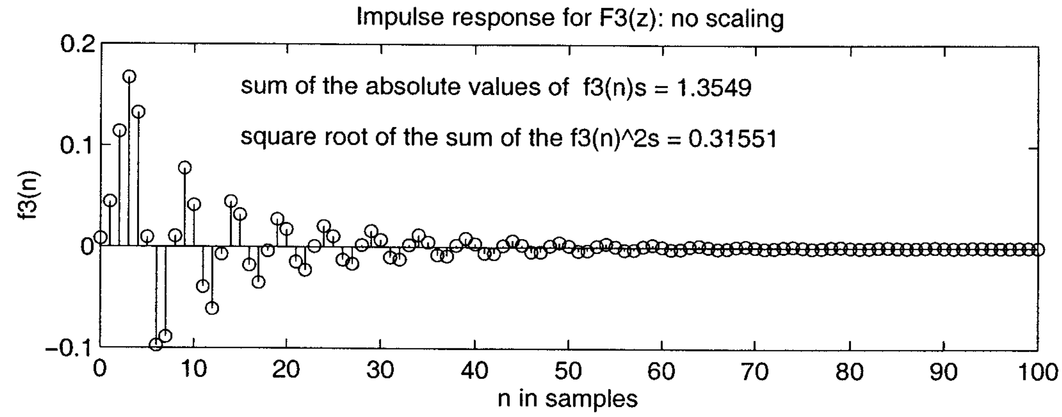


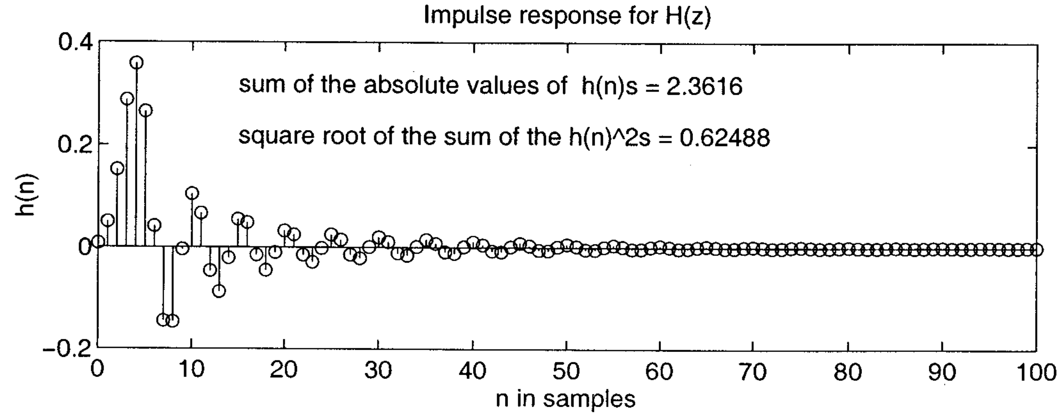


UNSCALED FILTER









WORST-CASE SCALING

- For the unscaled filter,

$$d_1 = \sum_{k=0}^{\infty} |f_1(k)| = 0.035377$$

$$d_2 = \sum_{k=0}^{\infty} |f_2(k)| = 0.21723$$

$$d_3 = \sum_{k=0}^{\infty} |f_3(k)| = 1.3549$$

$$d_4 = \sum_{k=0}^{\infty} |h(k)| = 2.3617.$$

- For the filter scaled according to the worst-case scaling, it is required that that $d_1 = d_2 = d_3 = 1$.
- The scaling is performed by changing the filter coefficients as follows:

$$S^* = SC_1$$

$$a_{k1}^* = C_2 a_{k1}, \quad k = 0, 1, 2$$

$$a_{k2}^* = C_3 a_{k3}, \quad k = 0, 1, 2$$

$$a_{k3}^* = \frac{a_{k3}}{C_1 C_2 C_3}, \quad k = 0, 1, 2,$$

where the C_k's are selected such that $C_1 d_1 = 1$, $C_1 C_2 d_2 = 1$, and $C_1 C_2 C_3 d_3 = 1$.

- That is, $C_1 = 1/d_1$, $C_2 = 1/(d_2C_1)$, and $C_3 = 1/(C_1C_2d_3)$.

- Note that the change of the a_{k3}'s in the above manner guarantees that the transfer function remains the same even though at the overall filter output an overflow may occur ($d_4 = 2.3617$).

- To properly treat the possible overflow at the filter output, extra integer bits are needed. If this is not possible and no overflows are allowed, the coefficient a_{k3} values must be further divided by $d_4 = 2.3617$, resulting in a filter with the passband maximum equal to $1/d_4$. **This problem has not been considered in the literature!!**

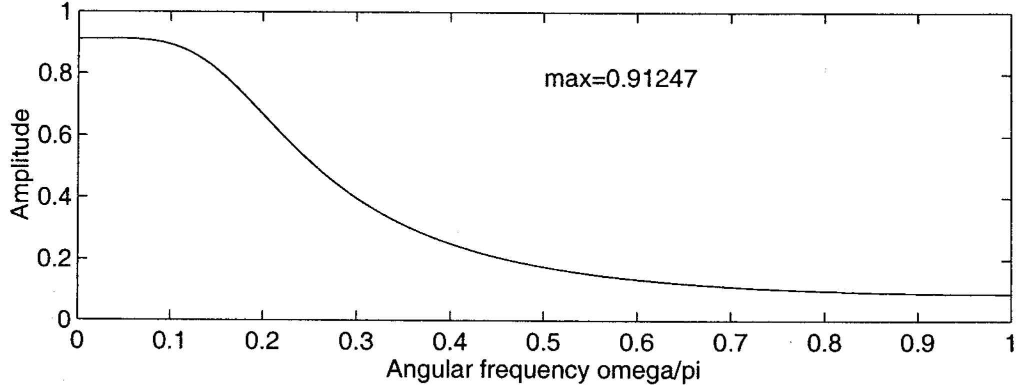
- For the scaled filter,

$$S^* = SC_1 = 0.229288$$

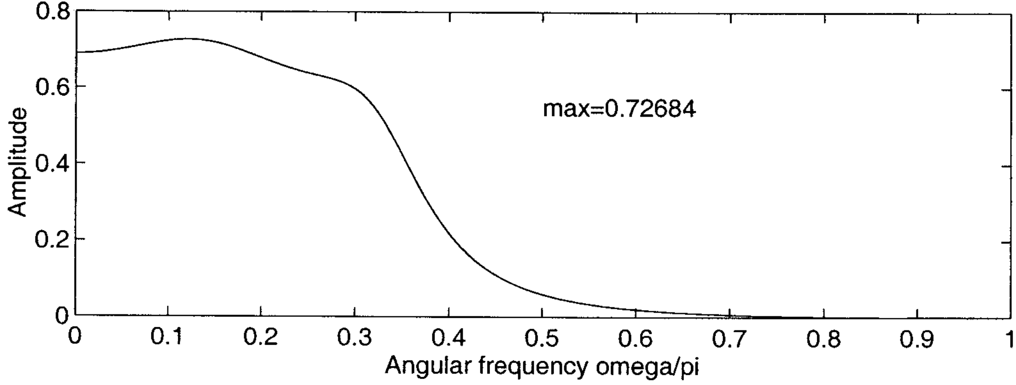
$$\begin{aligned} a_{01}^* = a_{21}^* &= 0.162853, \quad a_{11}^* = 0.300032 \\ a_{02}^* = a_{22}^* &= 0.160329, \quad a_{12}^* = 0.178264 \\ a_{03}^* = a_{23}^* &= 1.354926, \quad a_{13}^* = 0.909173. \end{aligned}$$

- The following two pages show the amplitude and impulse responses for the resulting $F_k(z)$'s and $H(z)$.

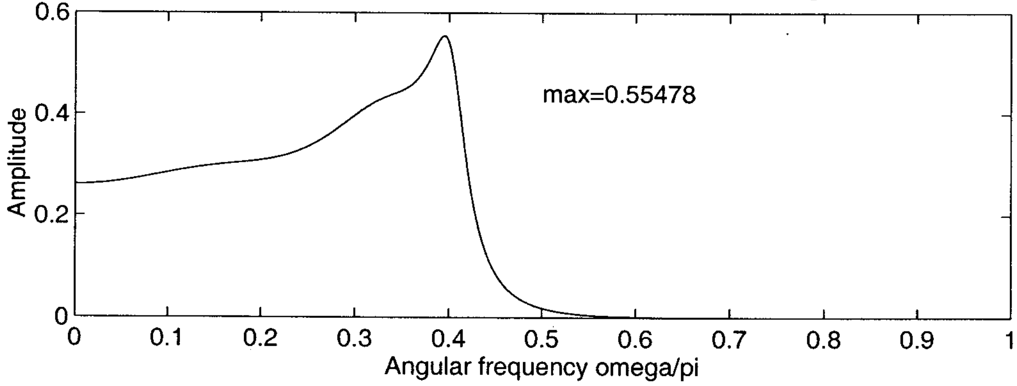
FILTER WITH THE WORST-CASE SCALING



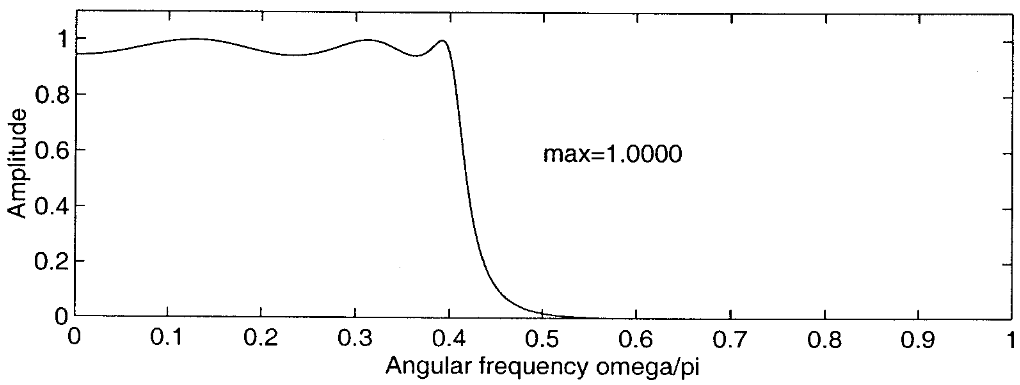
Amplitude response for F1(z): worst-case scaling



Amplitude response for F2(z): worst-case scaling



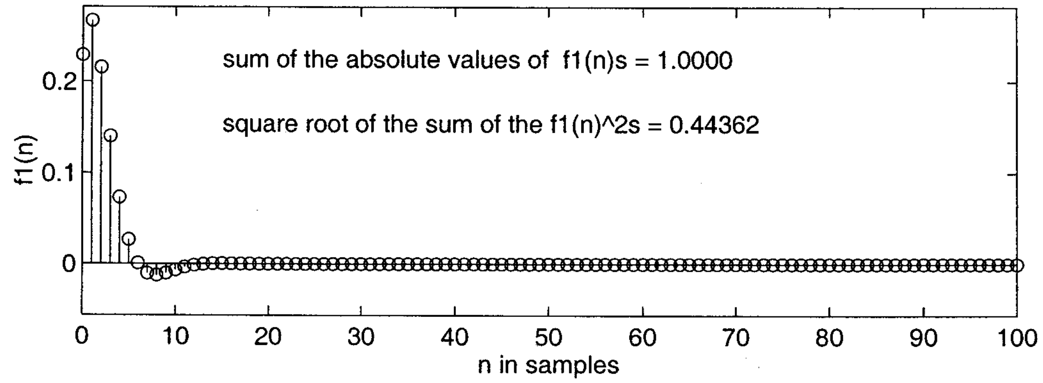
Amplitude response for F3(z): worst-case scaling



Amplitude response for H(z): worst-case scaling

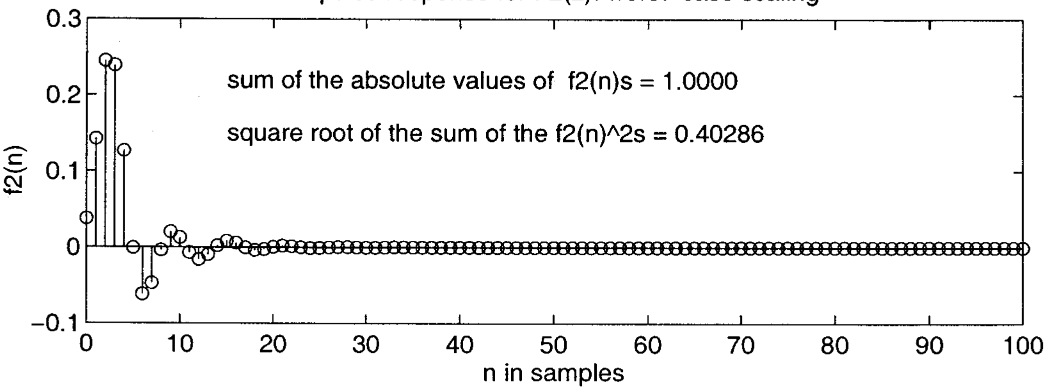
FILTER WITH THE WORST-CASE SACLING

Impulse response for F1(z): worst-case scaling


sum of the absolute values of f1(n)s = 1.0000

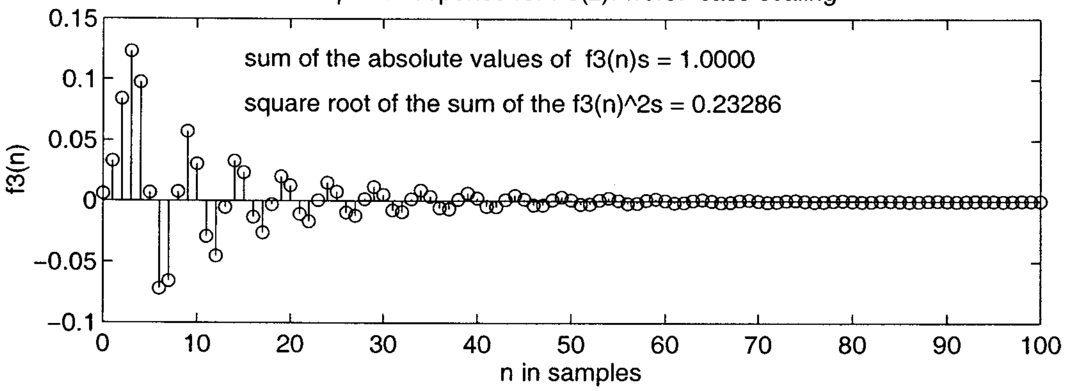
square root of the sum of the f1(n)^2s = 0.44362

Impulse response for F2(z): worst-case scaling


sum of the absolute values of f2(n)s = 1.0000

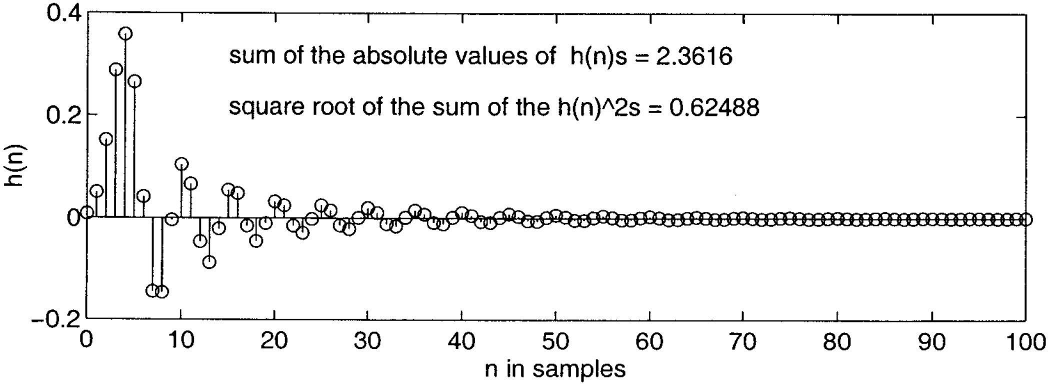
square root of the sum of the f2(n)^2s = 0.40286

Impulse response for F3(z): worst-case scaling


sum of the absolute values of f3(n)s = 1.0000

square root of the sum of the f3(n)^2s = 0.23286

Impulse response for H(z)


sum of the absolute values of h(n)s = 2.3616

square root of the sum of the h(n)^2s = 0.62488

L_∞-NORM SCALING

- For the unscaled filter,

$$d_1 = \max_{\omega\in[0,\ \pi]} |F_1(e^{j\omega})| = 0.032281$$

$$d_2 = \max_{\omega\in[0,\ \pi]} |F_2(e^{j\omega})| = 0.15789$$

$$d_3 = \max_{\omega\in[0,\ \pi]} |F_3(e^{j\omega})| = 0.75169$$

$$d_4 = \max_{\omega\in[0,\ \pi]} |H(e^{j\omega})| = 1.000.$$

- For the filter scaled according to the L_∞-norm, it is desired that $d_1 = d_2 = d_3 = 1$.

- The scaling is performed by changing the filter coefficients as follows:

$$S^* = SC_1$$

$$\begin{aligned} a^*_{k1} &= C_2 a_{k1}, \quad & k = 0, 1, 2 \\ a^*_{k2} &= C_3 a_{k3}, \quad & k = 0, 1, 2 \\ a^*_{k3} &= \frac{a_{k3}}{C_1 C_2 C_3}, \quad & k = 0, 1, 2, \end{aligned}$$

where the C_k's are selected such that $C_1 d_1 = 1$, $C_1 C_2 d_2 = 1$, and $C_1 C_2 C_3 d_3 = 1$.

- That is, $C_1 = 1/d_1$, $C_2 = 1/(d_2C_1)$, and $C_3 = 1/(C_1C_2d_3)$.

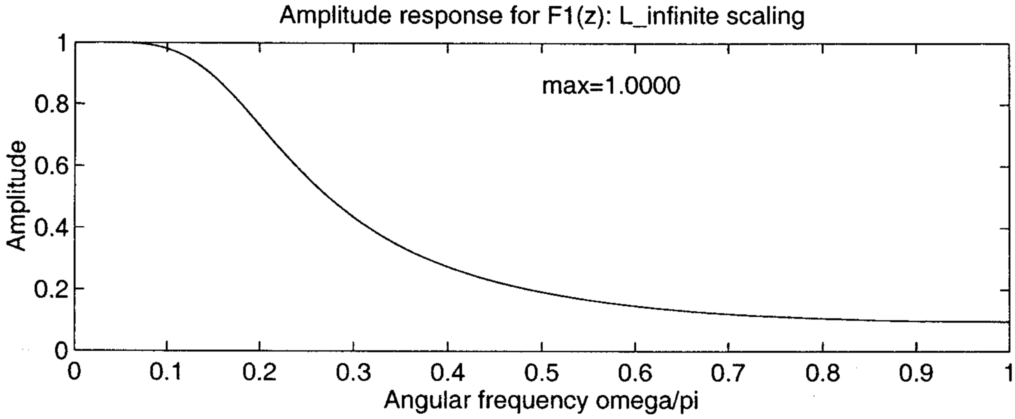
- For the scaled filter,

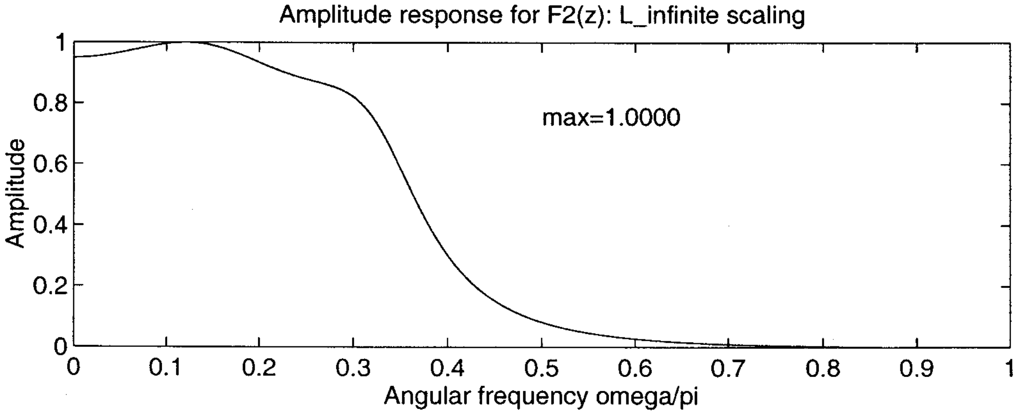
$$S^* = SC_1 = 0.251285$$

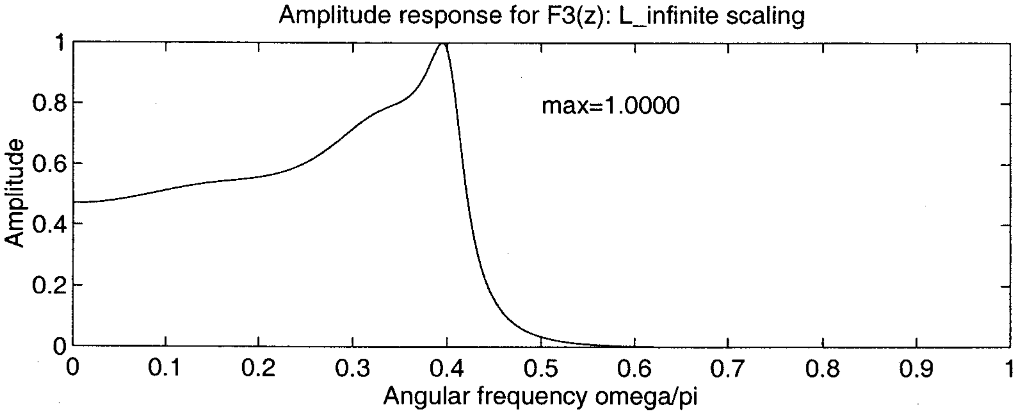
$$a_{01}^* = a_{21}^* = 0.204444, \quad a_{11}^* = 0.376659$$
$$a_{02}^* = a_{22}^* = 0.2100524, \quad a_{12}^* = 0.233549$$
$$a_{03}^* = a_{23}^* = 0.751689, \quad a_{13}^* = 0.504393.$$

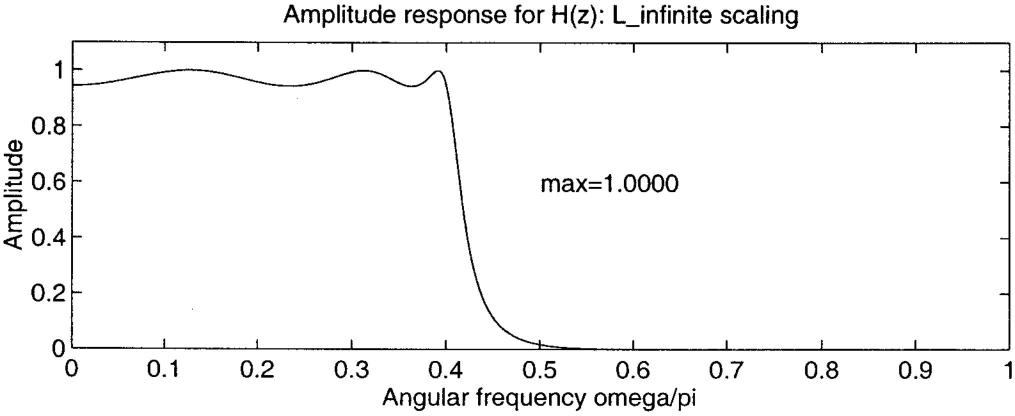
- The following two pages show the amplitude and impulse responses for the resulting $F_k(z)$'s and $H(z)$.

FILTER WITH THE L_∞-NORM SCALING

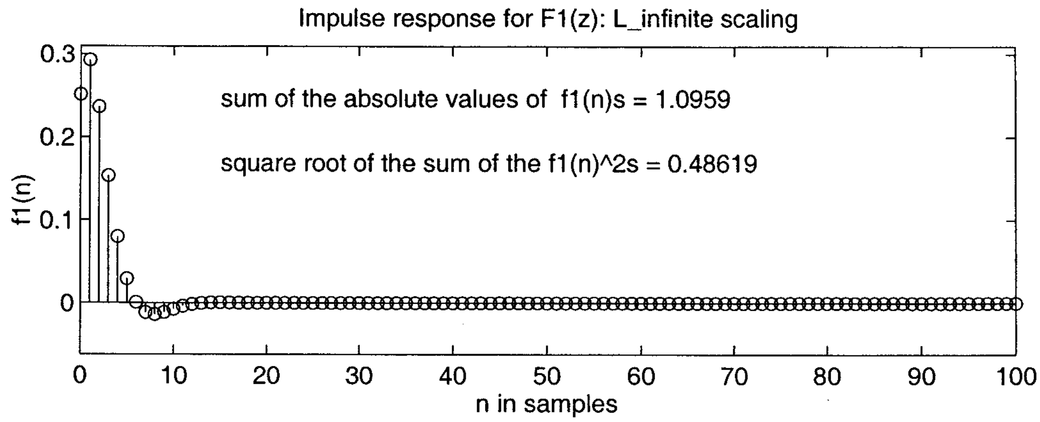


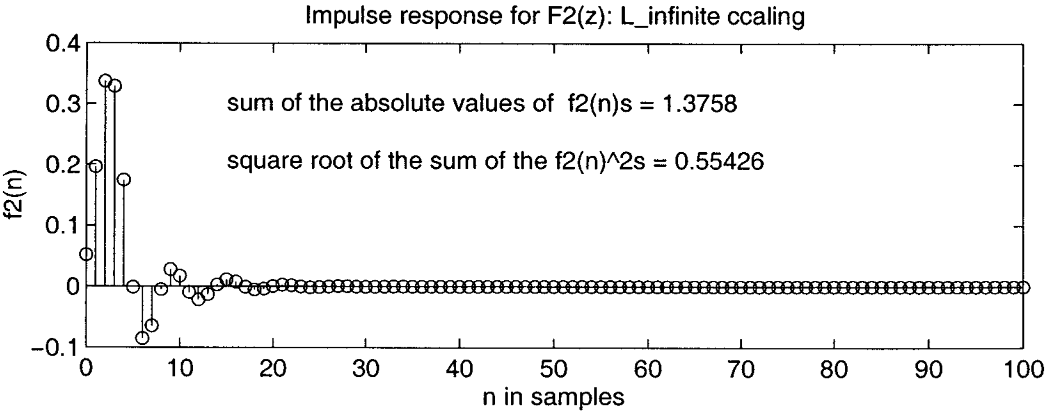


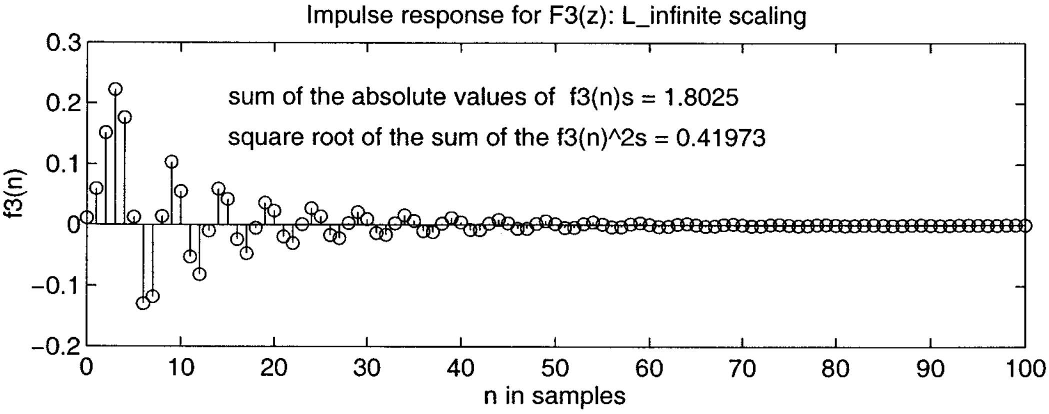


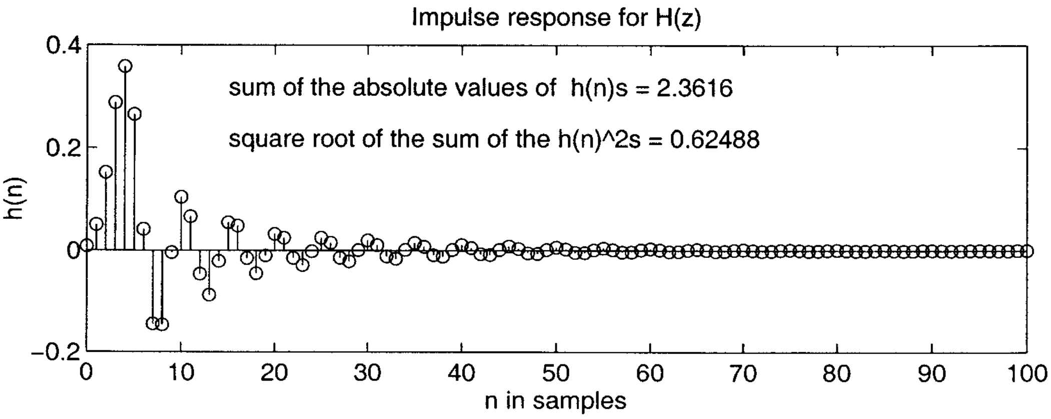


FILTER WITH THE L_∞-NORM SACLING









L_2-NORM SCALING

- For the unscaled filter,

$$d_1 = \sqrt{\sum_{k=0}^{\infty} f_1^2(k)} = 0.015694$$

$$d_2 = \sqrt{\sum_{k=0}^{\infty} f_2^2(k)} = 0.087515$$

$$d_3 = \sqrt{\sum_{k=0}^{\infty} f_3^2(k)} = 0.31551$$

$$d_4 = \sqrt{\sum_{k=0}^{\infty} h^2(k)} = 0.62488.$$

- For the filter scaled according to the L_2-norm scaling, it is desired that $d_1 = d_2 = d_3 = 1$.

- The scaling is performed by changing the filter coefficients as follows:

$$S^* = SC_1$$

$$\begin{aligned} a_{k1}^* &= C_2 a_{k1}, \quad & k = 0, 1, 2 \\ a_{k2}^* &= C_3 a_{k3}, \quad & k = 0, 1, 2 \\ a_{k3}^* &= \frac{a_{k3}}{C_1 C_2 C_3}, \quad & k = 0, 1, 2, \end{aligned}$$

where the C_k's are selected such that $C_1 d_1 = 1$, $C_1 C_2 d_2 = 1$, and $C_1 C_2 C_3 d_3 = 1$.

- That is, $C_1 = 1/d_1$, $C_2 = 1/(d_2C_1)$, and $C_3 = 1/(C_1C_2d_3)$.

- For the scaled filter,

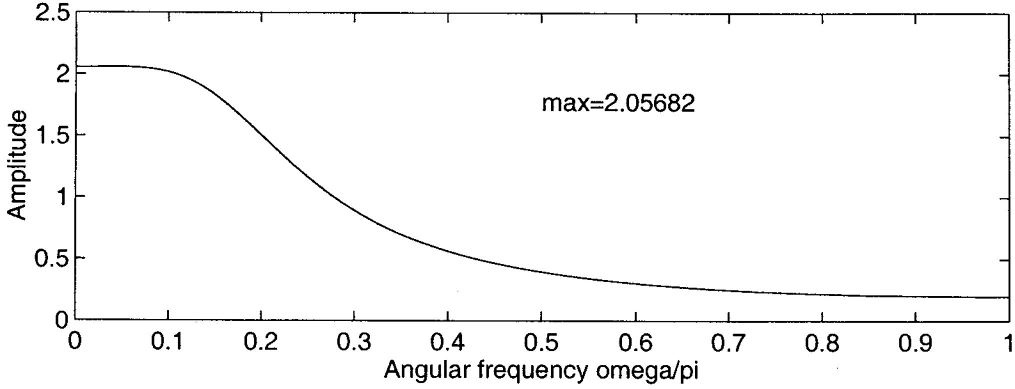
$$S^* = SC_1 = 0.516848$$

$$a_{01}^* = a_{21}^* = 0.179334, \quad a_{11}^* = 0.330396$$
$$a_{02}^* = a_{22}^* = 0.277379, \quad a_{12}^* = 0.308406$$
$$a_{03}^* = a_{23}^* = 0.315508, \quad a_{13}^* = 0.211710.$$

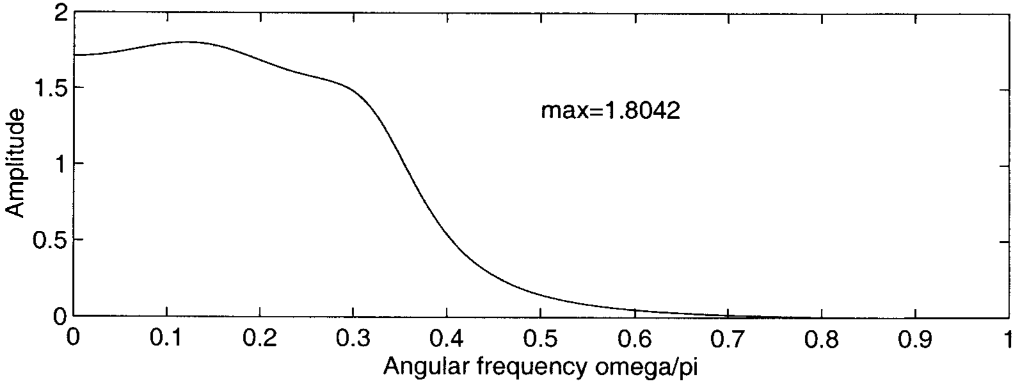
- The following two pages show the amplitude and impulse responses for the resulting $F_k(z)$'s and $H(z)$.

FILTER WITH THE L_2-NORM SCALING

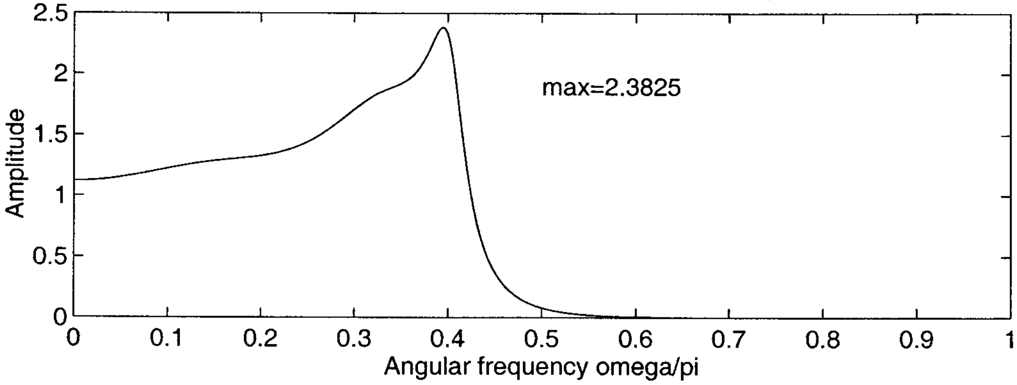
Amplitude response for F1(z): L_2 scaling



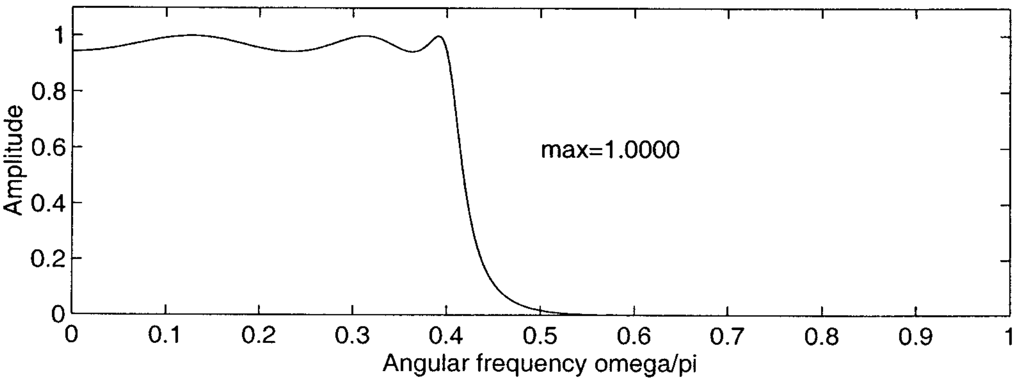
Amplitude response for F2(z): L_2 scaling



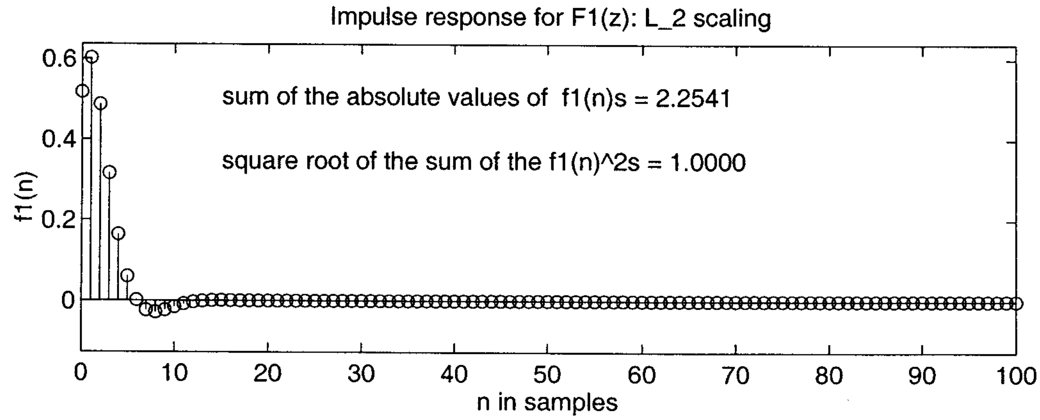
Amplitude response for F3(z): L_2 scaling

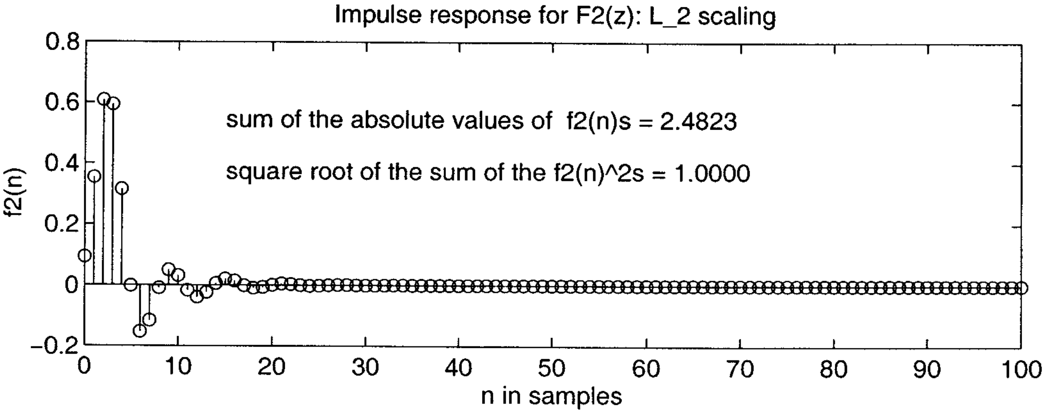


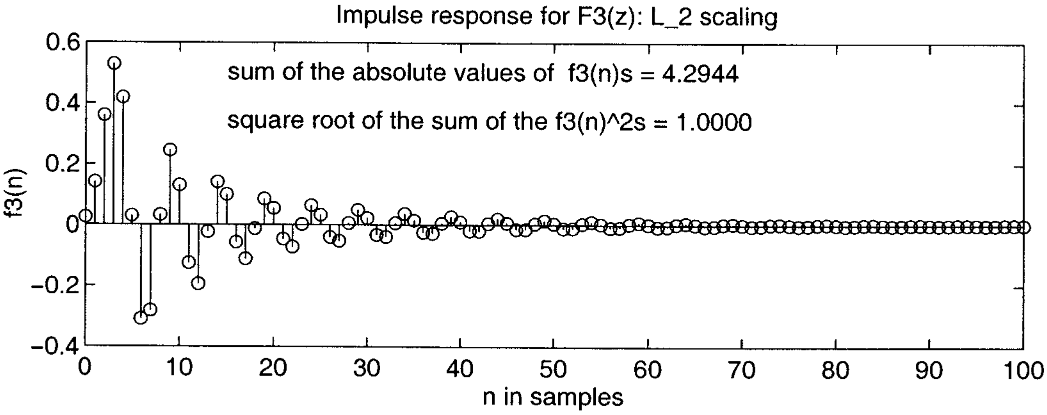
Amplitude response for H(z): no scaling

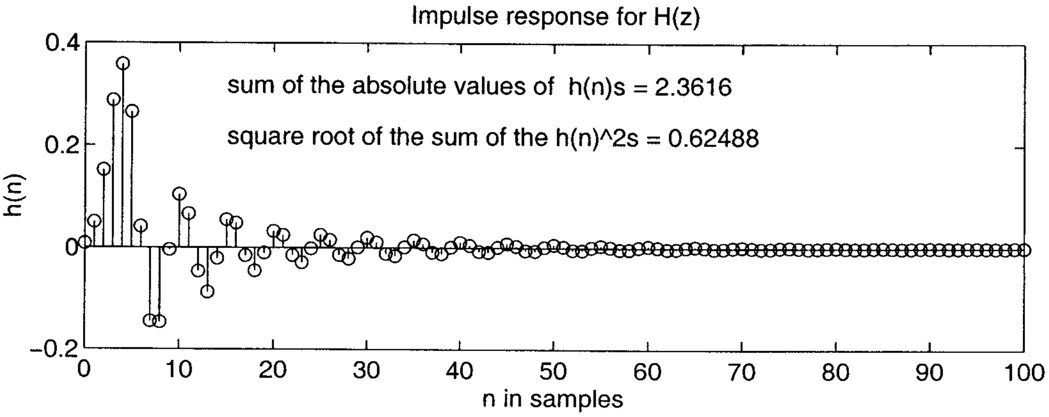


FILTER WITH THE L_2-NORM SACLING









OUTPUT NOISE VARIANCES DUE TO THE MULTIPLICATION ROUNDOFF ERRORS

- It is assumed that after each multiplier there is an error source with variance $\sigma_e^2 = 2^{-2b}/12$.

- For the unscaled filter, the output noise variance ($a_{0k} = 1$ and $a_{2k} = 1$ for $k = 1, 2, 3$ cause no noise) is given by

$$\sigma_f^2 = 3\sigma_{f1}^2 + 3\sigma_{f2}^2 + 3\sigma_{f3}^2 + 1\sigma_e^2,$$

where

$$\sigma_{fk}^2 = \sigma_e^2 \sum_{n=0}^{\infty} g_k^2(n), \quad k = 1, 2, 3.$$

- Here, $g_1(n)$ is the impulse response to the output after the multipliers S, b_{11}, and b_{21}. $g_2(n)$ is the impulse response to the output after the multipliers b_{12}, b_{22}, and a_{11}. $g_3(n)$ is the impulse response to the output after the multipliers b_{13}, b_{23}, and a_{12}. The impulse after the multiplier a_{13} is an impulse.

- The corresponding transfer functions are

$$G_1(z) = [A_1(z)A_2(z)A_3(z)]/[B_1(z)B_2(z)B_3(z)]$$
$$G_2(z) = [A_2(z)A_3(z)]/[B_2(z)B_3(z)]$$
$$G_3(z) = A_3(z)/B_3(z),$$

where for $k = 1, 2, 3$

$$A_k(z) = a_{0k} + a_{1k}z^{-1} + a_{2k}z^{-2}$$

and

$$B_k(z) = 1 - b_{1k}z^{-1} - b_{2k}z^{-2}.$$

- For the scaled filter, the output noise variance is given by

$$\sigma_f^2 = (3\sigma_{f1}^2 + 5\sigma_{f2}^2 + 5\sigma_{f3}^2 + 3\sigma_e^2.$$

- $g_k(n)$'s for the unscaled filter as well as for the scaled filters are given on pages 65, 66, 67, and 68.

- The output noise variances are

$$\sigma_f^2 = \begin{cases} 18390\sigma_e^2 & \text{for unscaled} \\ 153.16\sigma_e^2 & \text{for worst-case} \\ 70.604\sigma_e^2 & \text{for } L_\infty \\ 19.071\sigma_e^2 & \text{for } L_2. \end{cases}$$

- In terms of the quantity

$$\text{NOISE GAIN} = 10\log_{10}(\sigma_f^2/\sigma_e^2),$$

the results are

$$\text{NOISE GAIN} = \begin{cases} 42.6 \text{ dB} & \text{for unscaled} \\ 21.9 \text{ dB} & \text{for worst-case} \\ 18.5 \text{ dB} & \text{for } L_\infty \\ 12.8 \text{ dB} & \text{for } L_2. \end{cases}$$

- It is seen that the noise gain is 21 - 31 dB less for the scaled filters. This corresponds to 3 to 5 bit improvement in the computation accuracy.

- This means that required number of bits for the data representation is 3 to 5 bits less to achieve the same output noise level.

- A 6-dB reduction in the noise gain means that one bit less is required for the data representatio in order to achieve the same output noise level.

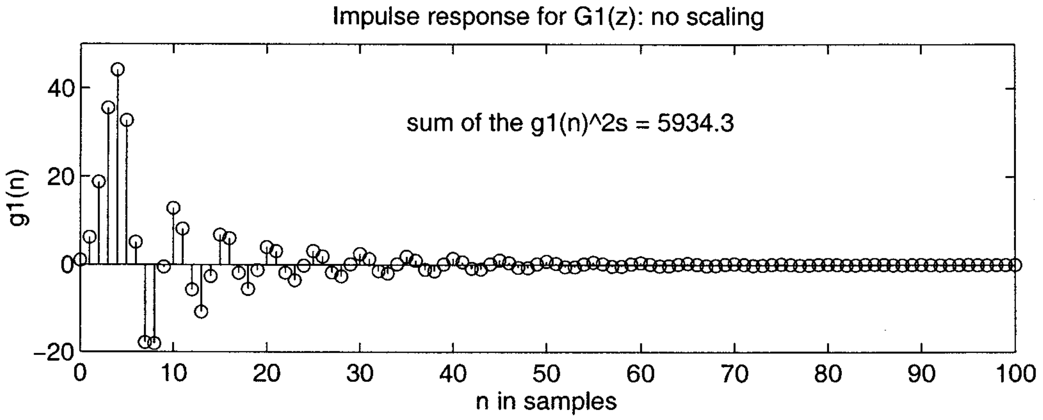
- In many cases, it is desired to increase the number of bits at the filter input for internal calculations.

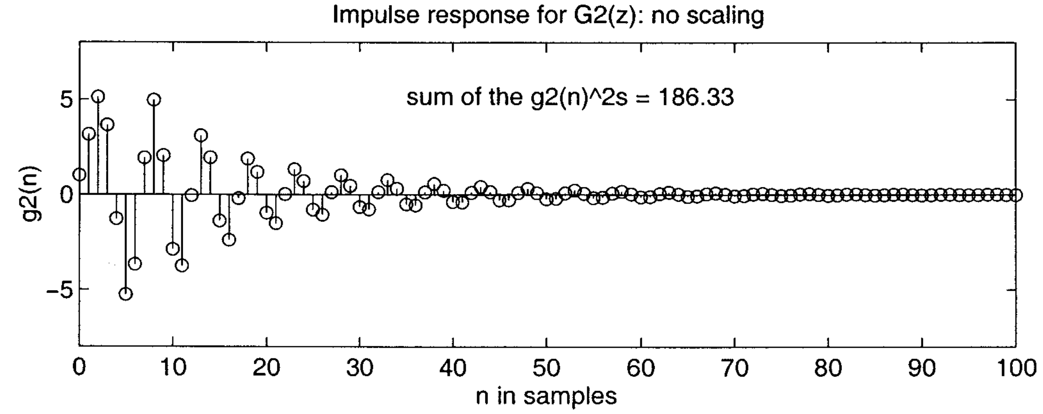
- The purpose is to select the number of extra bits in such a way that when at the output of the filter the data sequence is rounded back to the original number of bits, the overall system corresponds to an ideal filter with one additive noise sourse at the filter output.

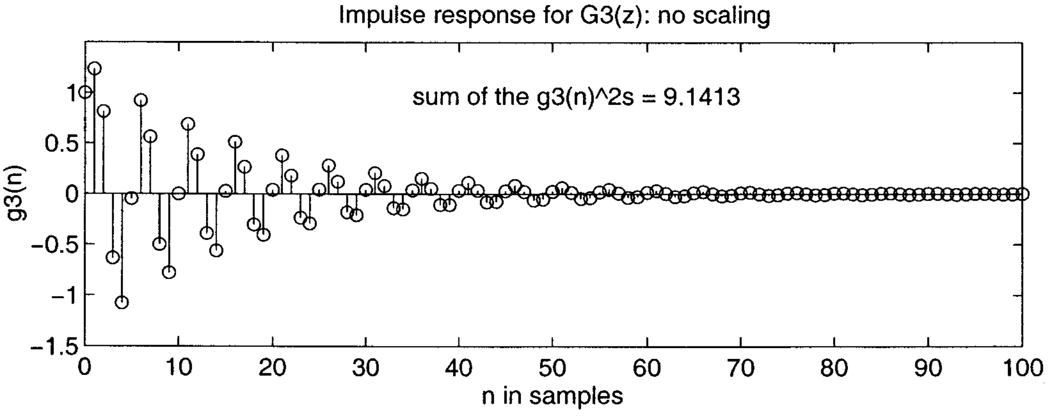
- The rule of thumb for the number of extra bits is the noise gain divided by 6.

- The file generating the above figures is ts/matlab/dsp/ellisca.m.

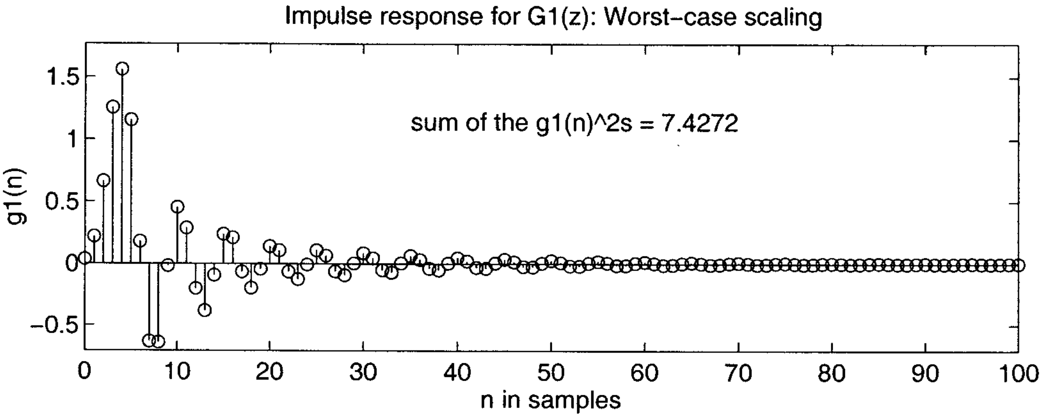
IMPULSE RESPONSES OF NOISE SOURCES: UNSCALED FILTER

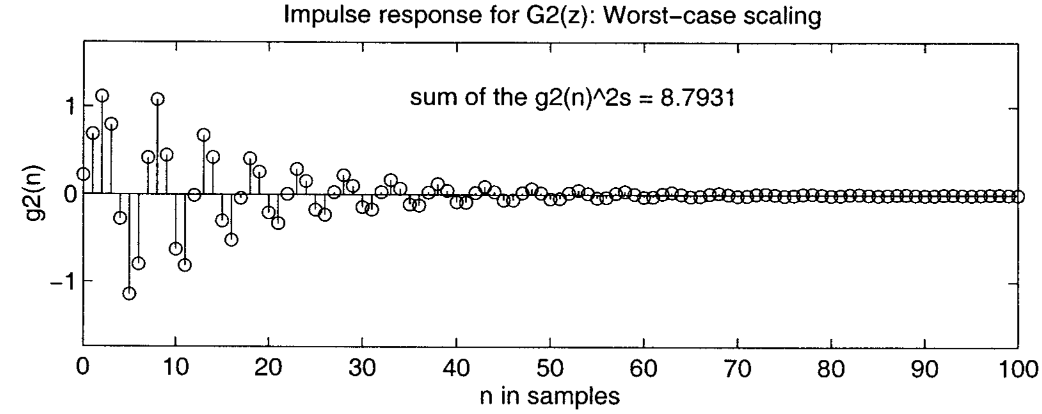

Impulse response for G1(z): no scaling

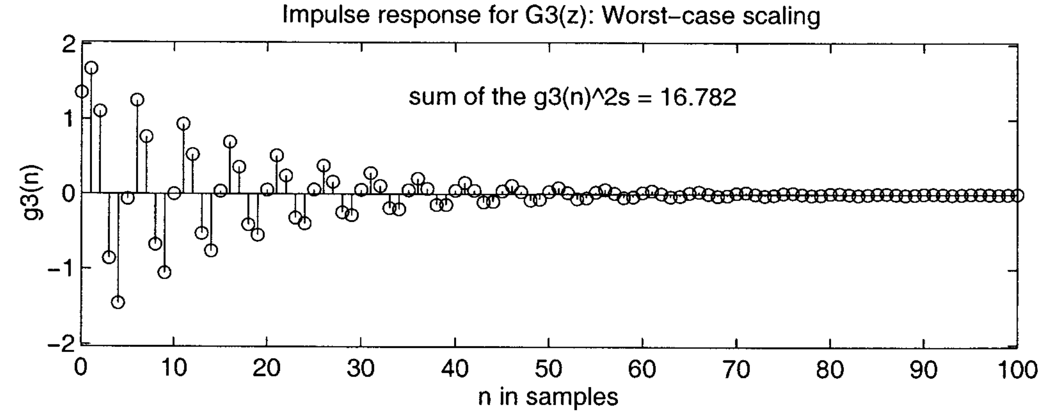

Impulse response for G2(z): no scaling


Impulse response for G3(z): no scaling

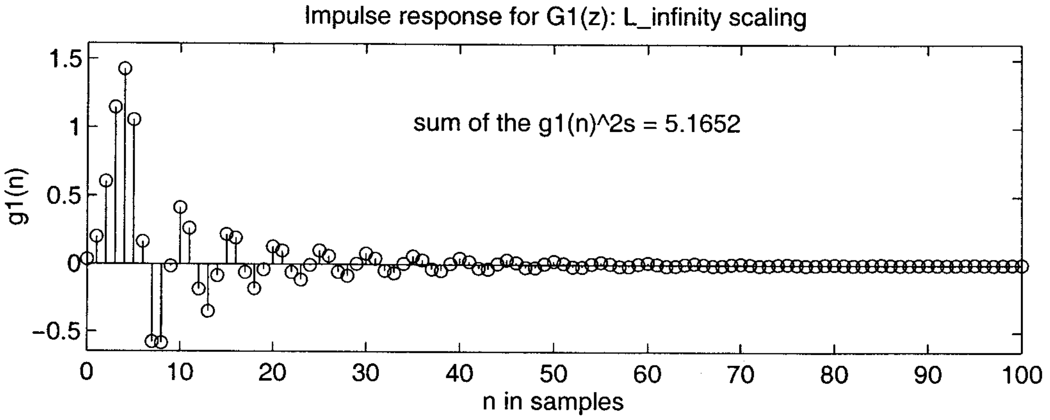
IMPULSE RESPONSES OF NOISE SOURCES: FILTER WITH THE WORST-CASE SCALING

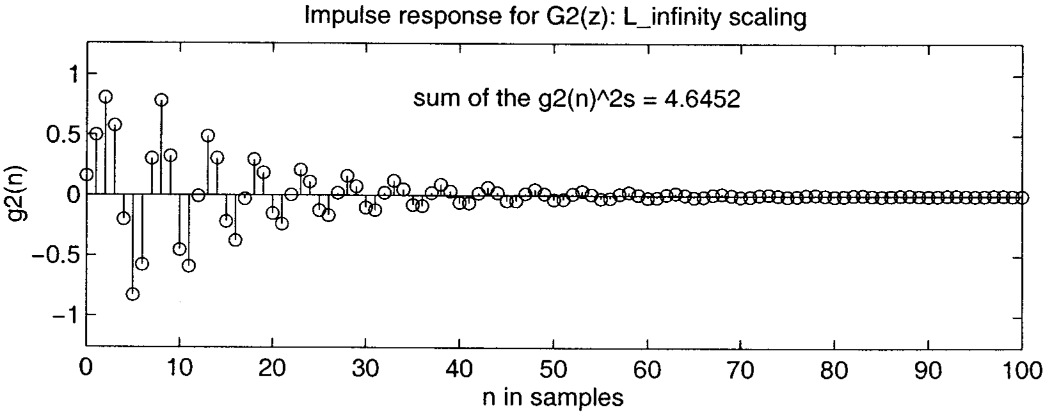


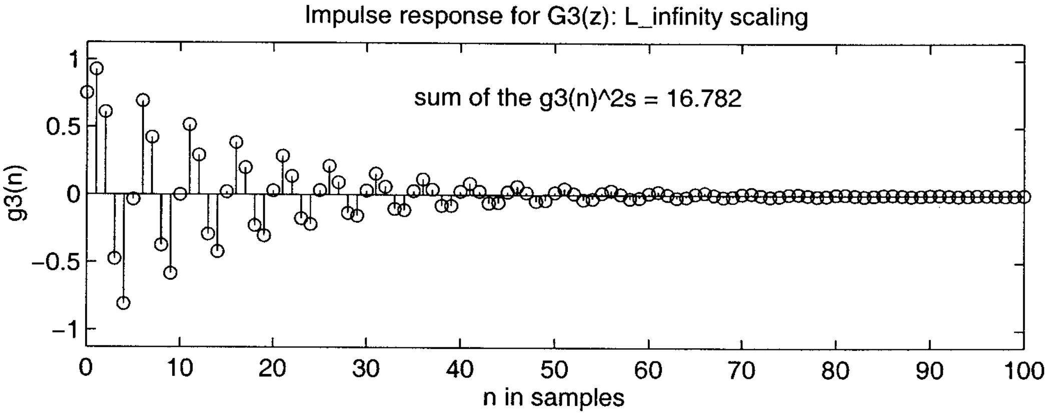




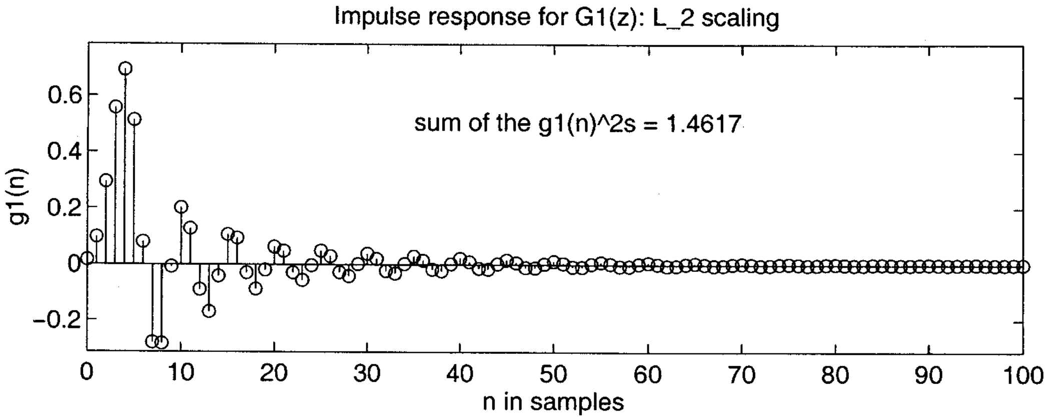
IMPULSE RESPONSES OF NOISE SOURCES: FILTER WITH THE L_∞-NORM SCALING

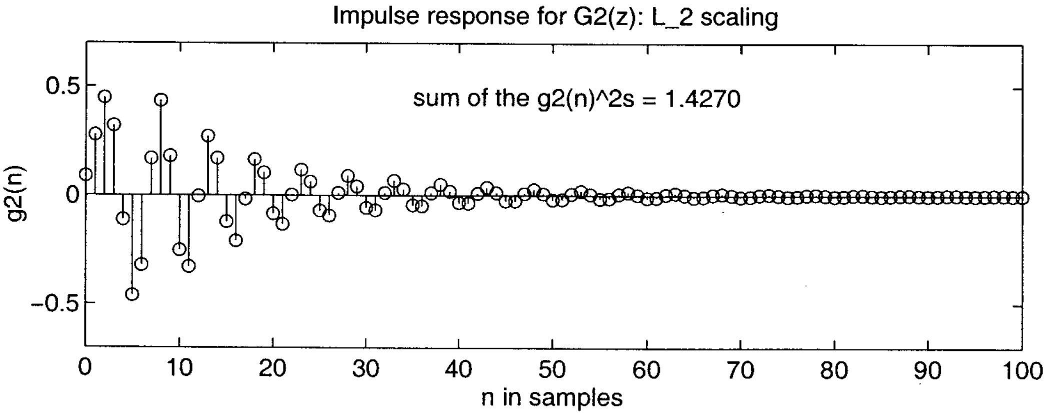


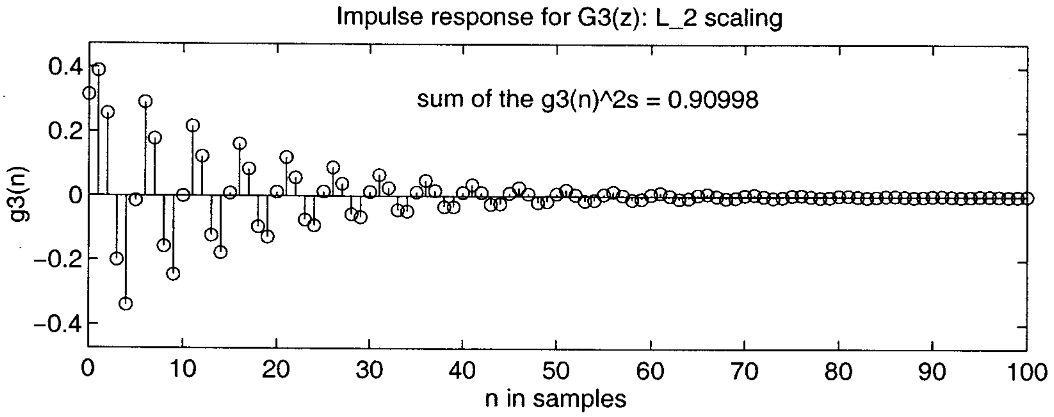




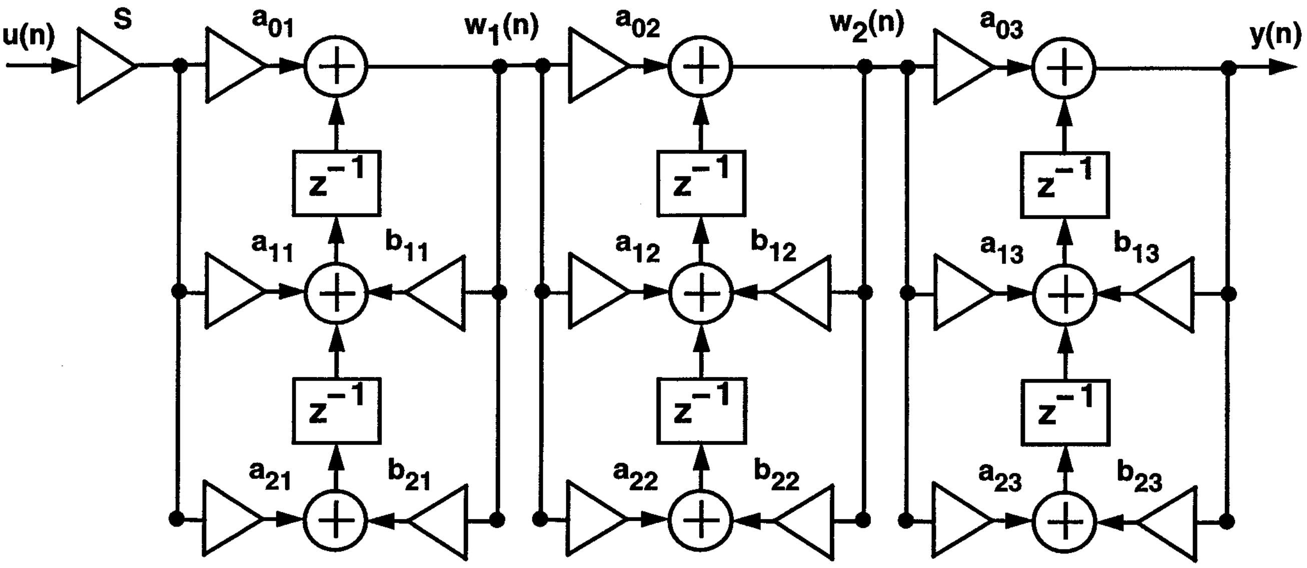
IMPULSE RESPONSES OF NOISE SOURCES: FILTER WITH THE L_2-NORM SCALING







CASCADE OF SECOND-ORDER TRANSPOSED DIRECT-FORM II SECTIONS



- The above figure shows an implementation of our filter as a cascade of transposed direct-form II sections.

- Coefficients of the unscaled filter:

$$S = 0.00811165$$

$$b_{11} = 1.165885, \quad b_{12} = 0.814467 \quad b_{13} = 0.564167$$

$$b_{21} = -0.417533, \quad b_{22} = -0.640952 \quad b_{23} = -0.883560$$

$$a_{0k} = a_{2k} = 1, \quad k = 1, 2, 3$$

$$a_{11} = 1.842348, \quad a_{12} = 1.111859, \quad a_{13} = 0.671013.$$

- For L_∞-norm scaling, it is required that the maximum amplitude value of the transfer functions from the input to the variables $w_1(n)$ and $w_2(n)$, denoted by $F_1(z)$ and $F_2(z)$, is less than or equal to unity.

- Since the maximum of the overall amplitude response is unity, the filter output is automatically L_∞-norm scaled.

FILTER SCALING

- The two scaling transfer function as well as the overall transfer function are given by

$$F_1(z) = [SA_1(z)]/B_1(z)$$

$$F_2(z) = [SA_1(z)A_2(z)]/[B_1(z)B_2(z)]$$

$$H(z) = [SA_1(z)A_2(z)A_3(z)]/[B_1(z)B_2(z)B_3(z)],$$

where for $k = 1, 2, 3$

$$A_k(z) = a_{0k} + a_{1k}z^{-1} + a_{2k}z^{-2}$$

and

$$B_k(z) = 1 - b_{1k}z^{-1} - b_{2k}z^{-2}.$$

- As seen from the following page, the maximum amplitude values of $F_1(z)$, $F_2(z)$, and $H(z)$ are $d_1 = 0.12385489$, $d_2 = 0.47436880$, and $d_3 = 1.0000000$, respectively.

- By selecting $S_1 = 1$, $Sa_{0k}/d_1 \mapsto a_{0k}$, $d_1a_{1k}/d_2 \mapsto a_{1k}$, and $d_2a_{2i}/ \mapsto a_{2k}$ for $k = 1, 2, 3$, the a_{ik}'s in the figure of the previous page become

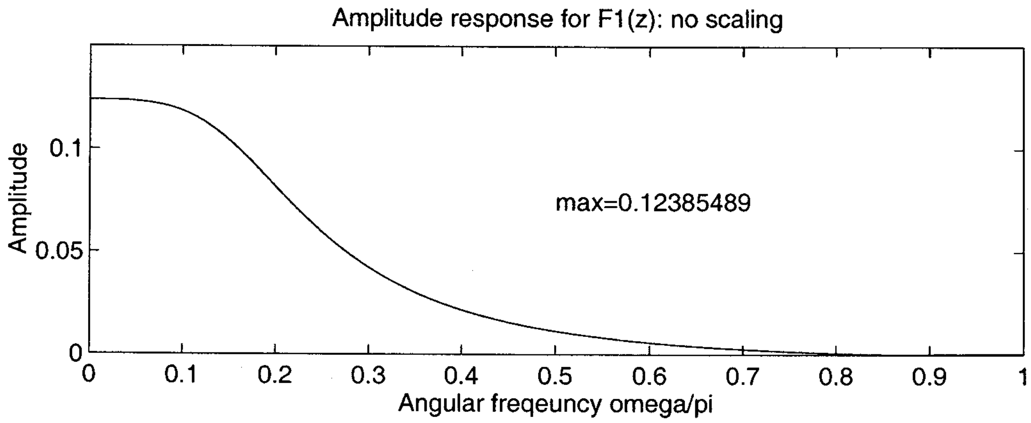
$$a_{01} = a_{21} = 0.065493181, a_{11} = 0.120661225$$

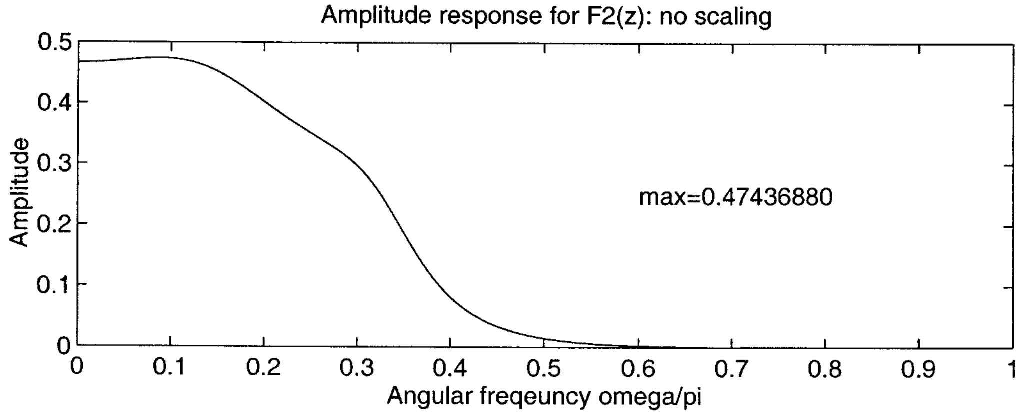
$$a_{02} = a_{22} = 0.261094093, a_{12} = 0.290299894$$

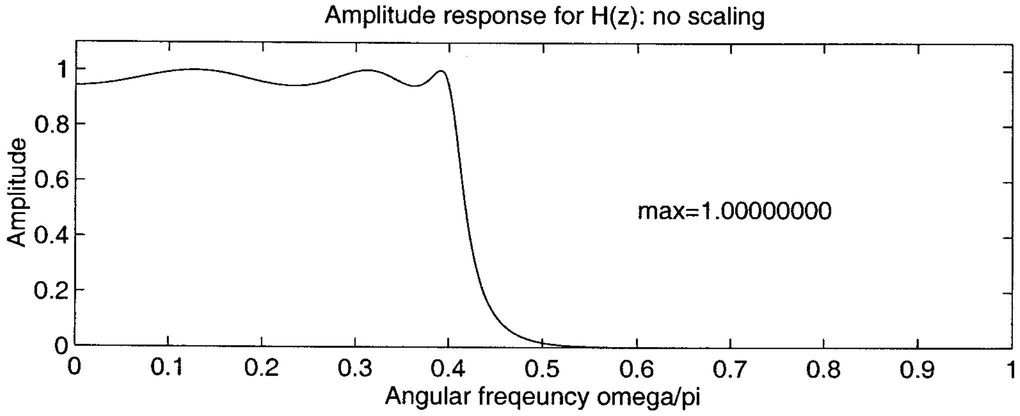
$$a_{03} = a_{23} = 0.474368802, a_{13} = 0.474368802.$$

- As seen from the figure on page 71, the filter is in this case L_∞-norm scaled.

SCALING TRANSFER FUNCTIONS: UNSCALED FILTER

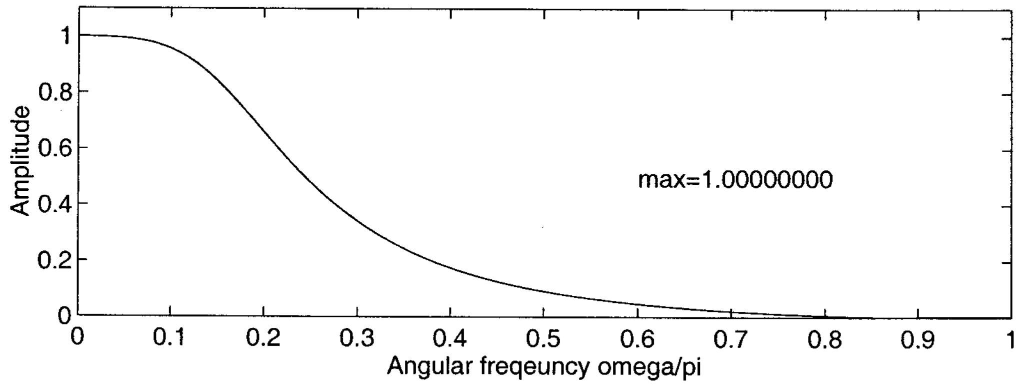

Amplitude response for F1(z): no scaling


Amplitude response for F2(z): no scaling

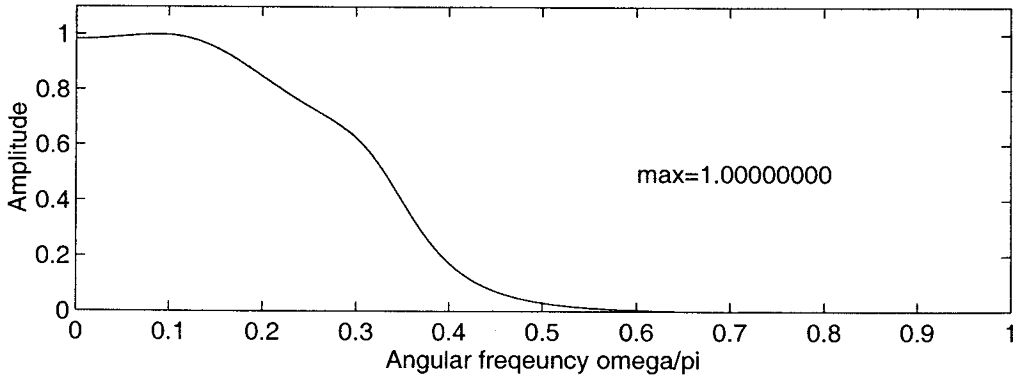

Amplitude response for H(z): no scaling

SCALING TRANSFER FUNCTIONS: SCALED FILTER

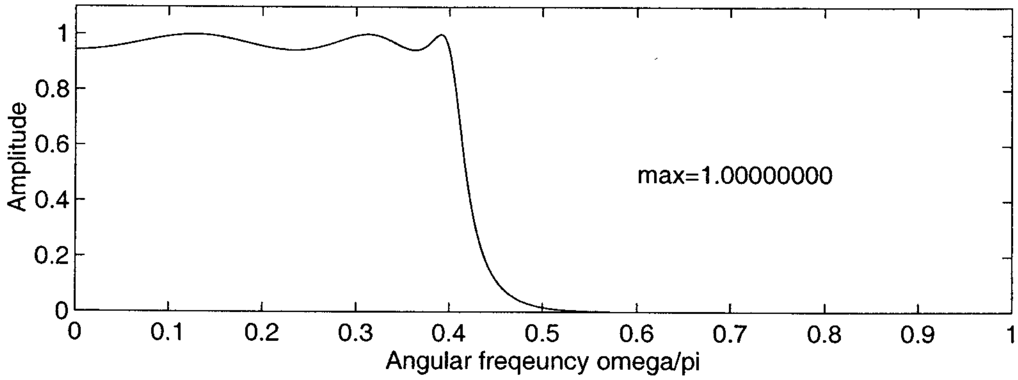
Amplitude response for F1(z): L_infinite scaling



Amplitude response for F2(z): L_infinite scaling



Amplitude response for H(z): L_infinite scaling



OUTPUT NOISE VARIANCES DUE TO THE MULTIPLICATION ROUNDOFF ERRORS

- It is assumed that after each multiplier there is an error source with variance $\sigma_e^2 = 2^{-2b}/12$

- For the unscaled filter, the scaling constant is included in the multipliers a_{01}, a_{11}, and a_{21} by multiplying them by S.

- For this filter, the output noise variance ($a_{0k} = 1$ and $a_{2k} = 1$ for $k = 2, 3$ cause no noise) is given by
$$\sigma_f^2 = 5\sigma_{f1}^2 + 3\sigma_{f2}^2 + 3\sigma_{f3}^2,$$
where
$$\sigma_{fk}^2 = \sigma_e^2 \sum_{n=0}^{\infty} g_k^2(n), \quad k = 1, 2, 3.$$

- Here, $g_1(n)$ is the impulse response to the output after the multipliers b_{11}, b_{21}, a_{01}, a_{11}, and a_{21}. $g_2(n)$ is the impulse response to the output after the multipliers b_{12}, b_{22}, and a_{12}. $g_3(n)$ is the impulse response to the output after the multipliers b_{13}, b_{23}, and a_{13}.

- The corresponding transfer functions are ($S = 1$)
$$G_1(z) = [A_2(z)A_3(z)]/[B_1(z)B_2(z)B_3(z)]$$
$$G_2(z) = [A_3(z)]/[B_2(z)B_3(z)]$$
$$G_3(z) = 1/B_3(z),$$
where for $k = 1, 2, 3$

$$A_k(z) = a_{0k} + a_{1k}z^{-1} + a_{2k}z^{-2}$$

and

$$B_k(z) = 1 - b_{1k}z^{-1} - b_{2k}z^{-2}.$$

- For the scaled filter, the output noise variance is given by

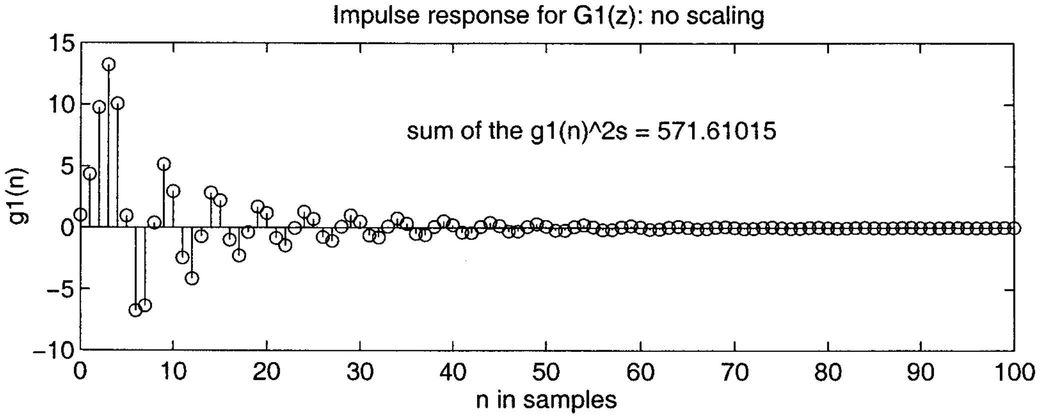
$$\sigma_f^2 = 5(\sigma_{f1}^2 + \sigma_{f2}^2 + \sigma_{f3}^2)\sigma_e^2.$$

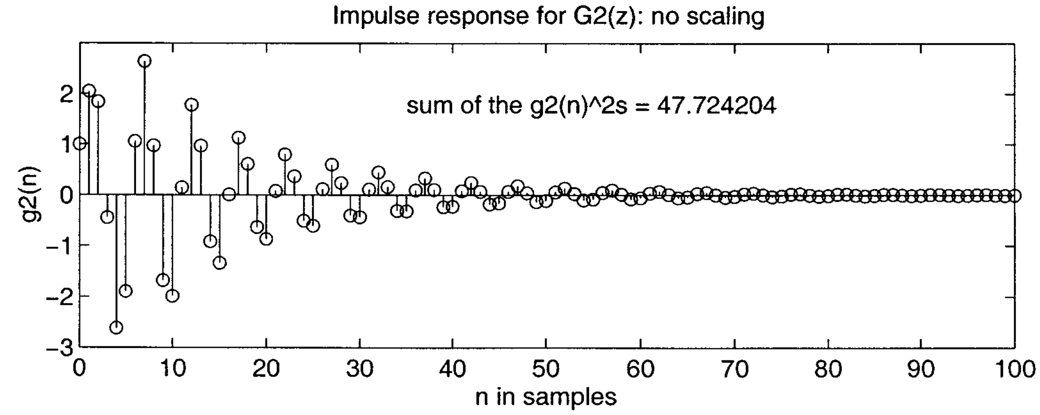
- $g_k(n)$'s for the unscaled filter as well as for the scaled filter are given on pages 75 and 76.

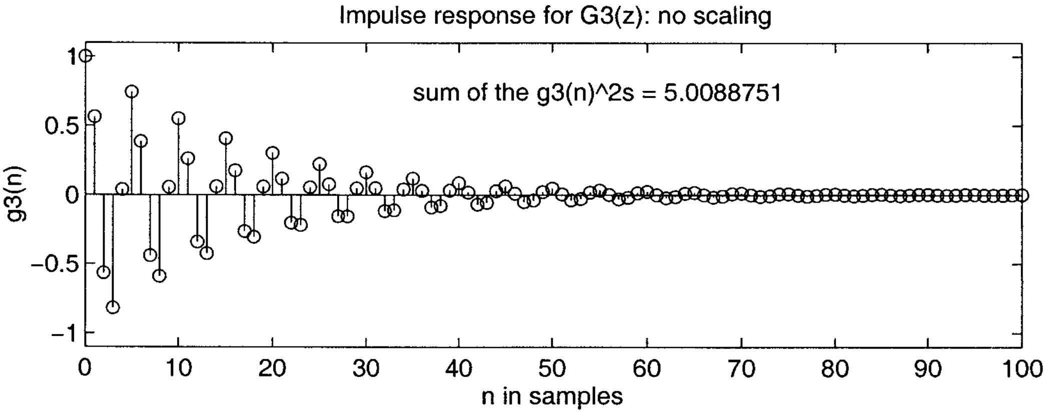
- The output noise variances are

$$\sigma_f^2 = \begin{cases} 1873.02968\sigma_e^2 & \text{for unscaled} \\ 122.58284\sigma_e^2 & \text{for } L_\infty. \end{cases}$$

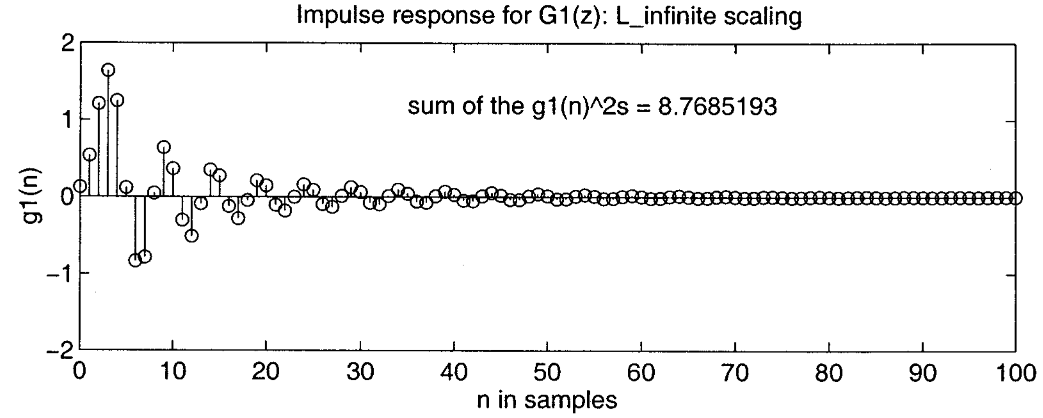
NOISE TRANSFER FUNCTIONS: UNSCALED FILTER

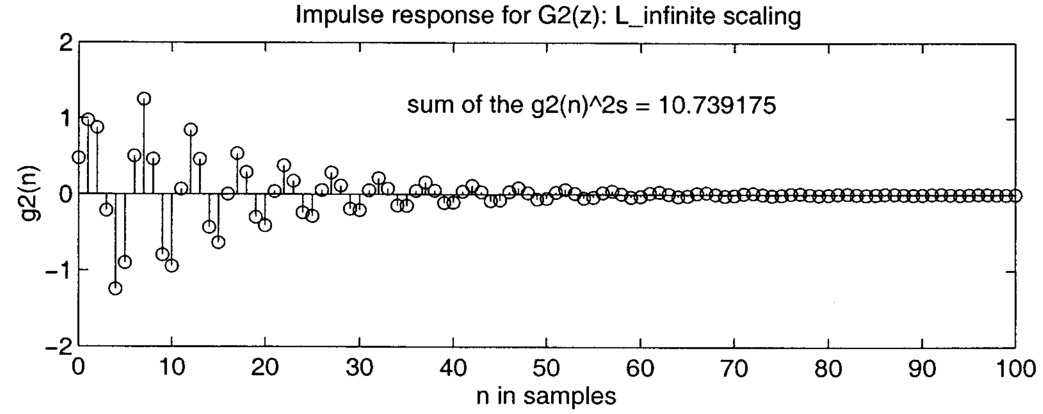


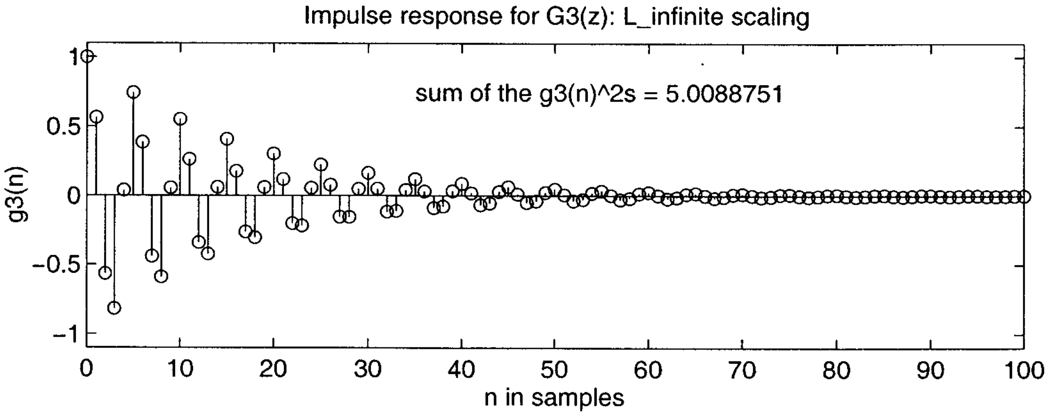




NOISE TRANSFER FUNCTIONS: SCALED FILTER







COMMENTS

- There are several ways of forming second-order blocks for the cascade realization.

- It has been observed experimentally that a low output noise variance is obtained by applying rules of the following form:

1. The pole pair that is closest to the unit circle should be paired with the zero pair that is closest to it in the z-plane.

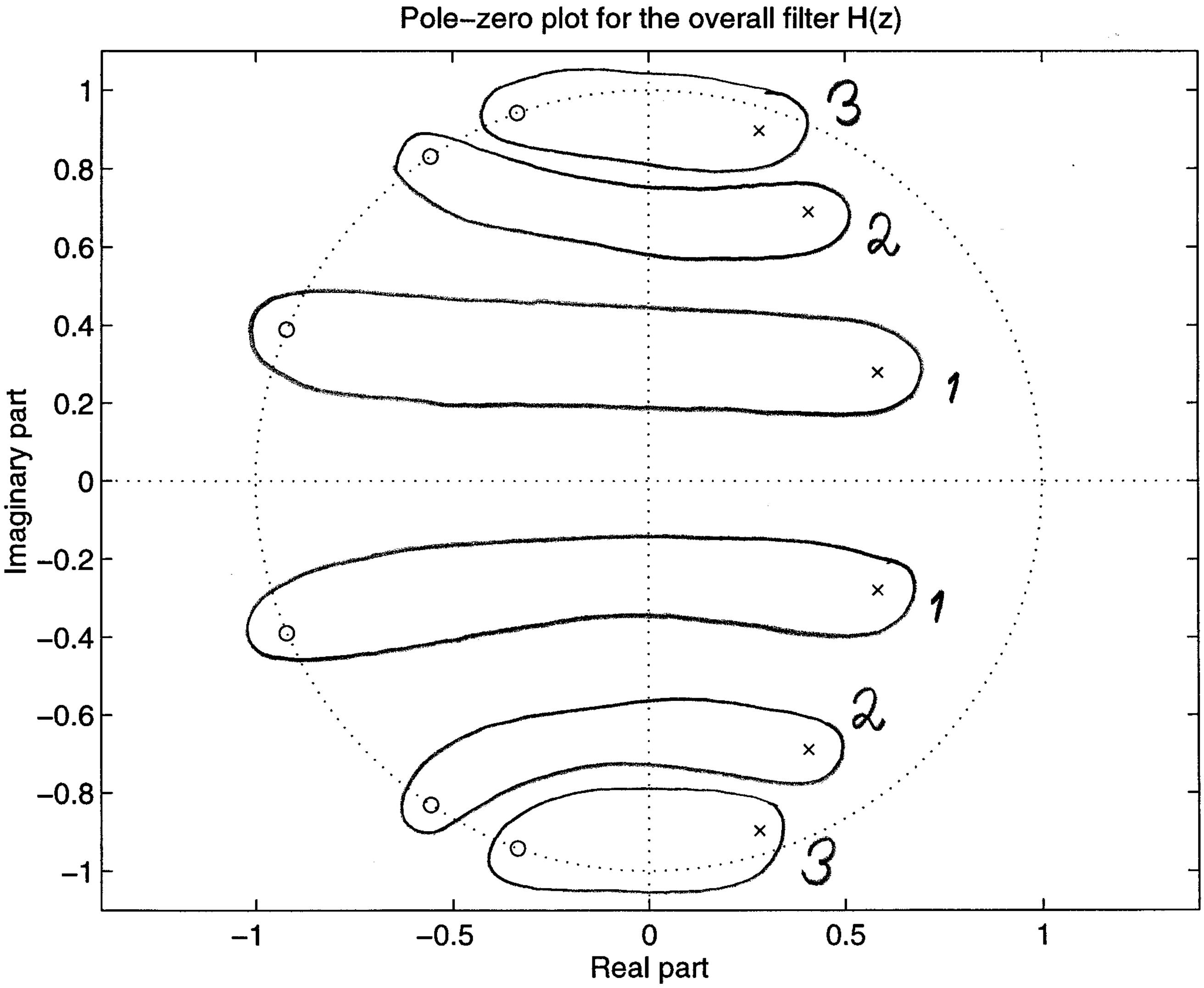
2. Rule 1 should be repeatedly applied until all the poles and zeros have been paired.

3. The resulting second-order sections should be ordered according to the closeness of the poles to the unit circle, either in order of increasing closeness to the unit circle or decreasing closeness to the unit circle.

- For our filter, the second-order sections have been selected as shown on the next page.

POLE-ZERO PLOT FOR OUR SIXTH-ORDER FILTER SHOWING PAIRING OF THE POLES AND ZEROS

- The zeros are located at $z = \exp(\pm j0.608907580\pi)$, $z = \exp(\pm j0.68763761\pi)$, and $z = \exp(\pm j0.87276841\pi)$, whereas the poles are located at $z = 0.93997874\exp(\pm j0.40298141\pi)$, $z = 0.80059474\exp(\pm j0.33013970\pi)$, and $z = 0.64616766\exp(\pm j0.14198545\pi)$.

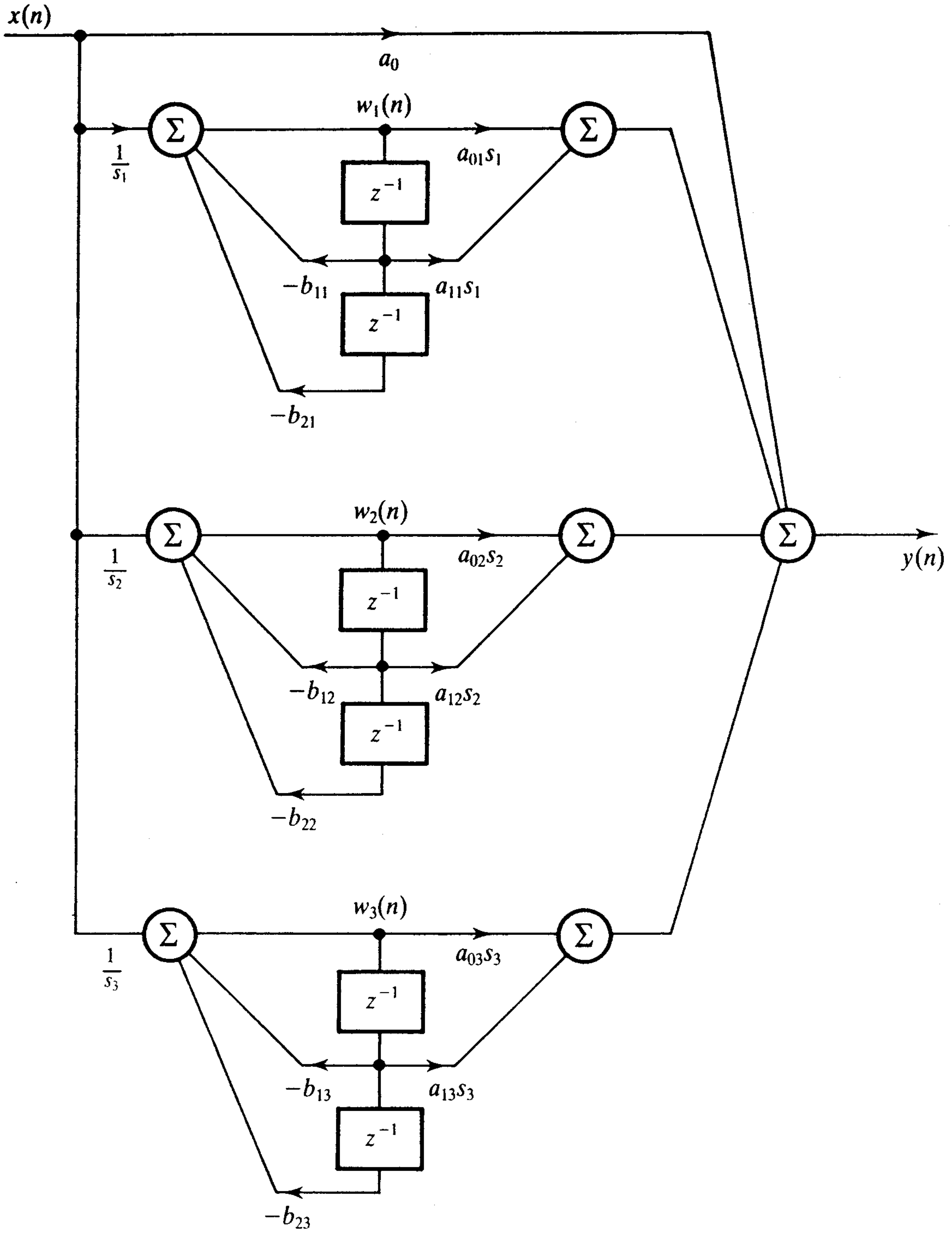


```
%This file illustrates the scaling of a casced structure with 3 second-order sections
%transposed direct-form II sections. Passband ripple is 0.5dB, stopband attenuation
%80 dB and passband and stopband edges at 0.4pi and 0.6pi.
%Can be found in SUN’s: ~ts/matlab/dsp/sca.m
%Matlab likes to make the stopband edge as small
%as possible!!
[N, Wn] = ellipord(.4, .6, 0.5, 80.24234);
[B,A] = ellip(N,.5,80.24234,Wn);
[HH,w]=freqz(B,A,8*1024);
figure(1);plot(w/pi,20*log10(abs(HH)));axis([0 1 -140 10]);
title(‘Amplitude response for the overall filter H(z)’)
ylabel(‘Amplitude in dB’);xlabel(‘Angular freqeuncy omega/pi’);
hold on;
axes(‘position’, [.24 .24 .3 .3]);plot(w/pi,20*log10(abs(HH)));
axis([0 .4 -0.5 0]);title(‘Passband details’)
ylabel(‘Amplitude in dB’);xlabel(‘Angular freqeuncy omega/pi’);hold off
figure(2);zplane(B,A);title(‘Pole-zero plot for the overall filter H(z)’)

%Poles and zeros
zer=roots(B);pol=roots(A);

%Three numerator terms
%Note that zer(1) and zer(2) are complex conjugates
%The same is true for zer(3) and zer(4); and for
%zer(5) and zer(6)
a1(1)=1;a1(3)=1;a1(2)=-2*real(zer(1));
a2(1)=1;a2(3)=1;a2(2)=-2*real(zer(3));
a3(1)=1;a3(3)=1;a3(2)=-2*real(zer(5));

%Three denominator terms
b1(1)=1;b1(2)=-2*real(pol(5));b1(3)=real(pol(5))^2+imag(pol(5))^2;
b2(1)=1;b2(2)=-2*real(pol(3));b2(3)=real(pol(3))^2+imag(pol(3))^2;
b3(1)=1;b3(2)=-2*real(pol(1));b3(3)=real(pol(1))^2+imag(pol(1))^2;
S=B(1);

%Scaling transfer functions

FN1=S*a1;FD1=b1;
FN2=conv(FN1,a2);FD2=conv(FD1,b2);
FN3=conv(FN2,a3);FD3=conv(FD2,b3);

[H1,w]=freqz(FN1,FD1,8*1024);[H2,w]=freqz(FN2,FD2,8*1024);
[H3,w]=freqz(FN3,FD3,8*1024);

inff1=max(abs(H1(1:8192)));inff2=max(abs(H2(1:8192)));inff3=max(abs(H3(1:8192)));

figure(3);subplot(2,1,1);plot(w/pi,abs(H1));
axis([0 1 0 .15]);
title(‘Amplitude response for F1(z): no scaling’)
ylabel(‘Amplitude’);xlabel(‘Angular freqeuncy omega/pi’);
```

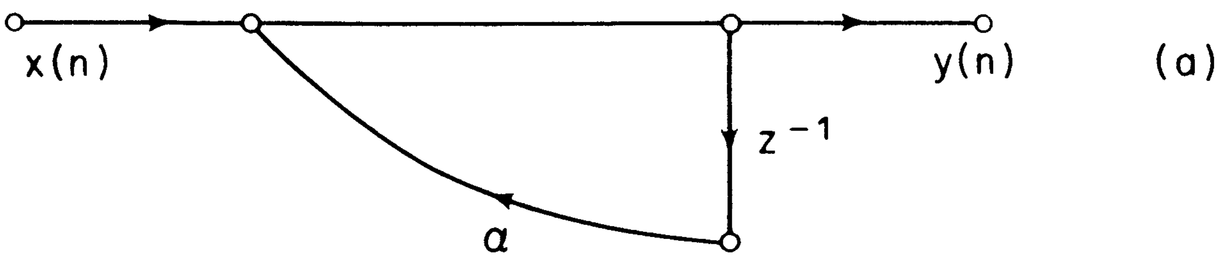
```
text(0.5,.075,’max=0.12385489’)
subplot(2,1,2);plot(w/pi,abs(H2));axis([0 1 0 .5]);
title(‘Amplitude response for F2(z): no scaling’)
ylabel(‘Amplitude’);xlabel(‘Angular freqeuncy omega/pi’);
text(0.6,.25,’max=0.47436880’)

figure(4);subplot(2,1,1);plot(w/pi,abs(H3));axis([0 1 0 1.1]);
title(‘Amplitude response for H(z): no scaling’)
ylabel(‘Amplitude’);xlabel(‘Angular freqeuncy omega/pi’);
text(0.6,.5,’max=1.00000000’)

%Noise when a_i1 --> Sa_i1, i=1,2,3
GN3(1)=1;GD3=b3;
GN2=conv(GN3,a3);GD2=conv(GD3,b2);
GN1=conv(GN2,a2);GD1=conv(GD2,b1);

%Corresponding impulse responses

[gg1,t]=impz(GN1,GD1,501);[gg2,t]=impz(GN2,GD2,501);
[gg3,t]=impz(GN3,GD3,501);

%Sum of the squared responses

sqg1=(sum(gg1.*gg1));sqg2=(sum(gg2.*gg2));sqg3=(sum(gg3.*gg3));

%Overall noise
noise1=5*sqg1+3*sqg2+3*sqg3

%Noise
%Impulse responses: no scaling
figure(5);subplot(2,1,1);impz(gg1);title(‘Impulse response for G1(z): no scaling’);
ylabel(‘g1(n)’);xlabel(‘n in samples’);axis([0 100 -10 15]);
text(35,8,’sum of the g1(n)^2s = 571.61015’);
subplot(2,1,2);impz(gg2);title(‘Impulse response for G2(z): no scaling’)
ylabel(‘g2(n)’);xlabel(‘n in samples’);axis([0 100 -3 3]);
text(35,1.8,’sum of the g2(n)^2s = 47.724204’);

figure(6);subplot(2,1,1);impz(gg3);title(‘Impulse response for G3(z): no scaling’);
ylabel(‘g3(n)’);xlabel(‘n in samples’);axis([0 100 -1.1 1.1]);
text(35,.75,’sum of the g3(n)^2s = 5.0088751’);

%Filter scaling
a1=S*a1/inff1;a2=inff1*a2/inff2;a3=inff2*a3;S=1

%Scaling transfer functions

FN1=S*a1;FD1=b1;
FN2=conv(FN1,a2);FD2=conv(FD1,b2);
FN3=conv(FN2,a3);FD3=conv(FD2,b3);
```

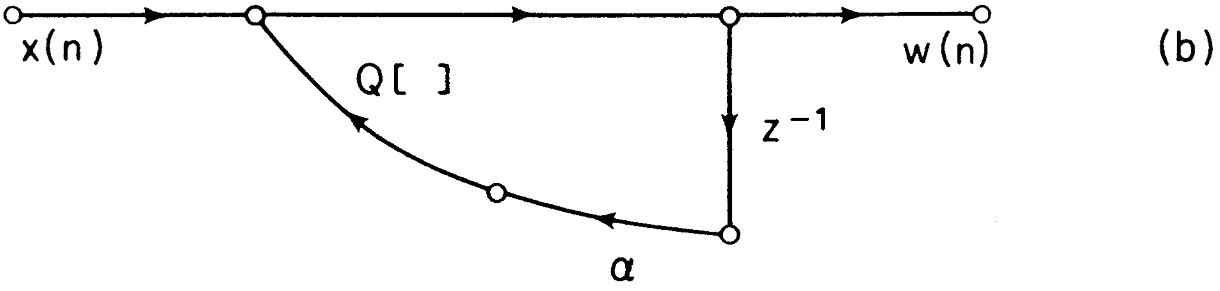
```
[H1,w]=freqz(FN1,FD1,8*1024);[H2,w]=freqz(FN2,FD2,8*1024);
[H3,w]=freqz(FN3,FD3,8*1024);

figure(7);subplot(2,1,1);plot(w/pi,abs(H1));axis([0 1 0 1.1]);
title('Amplitude response for F1(z): L_infinite scaling')
ylabel('Amplitude');xlabel('Angular freqeuncy omega/pi');
text(0.6,.5,'max=1.00000000')
subplot(2,1,2);plot(w/pi,abs(H2));axis([0 1 0 1.1]);
title('Amplitude response for F2(z): L_infinite scaling')
ylabel('Amplitude');xlabel('Angular freqeuncy omega/pi');
text(0.6,.5,'max=1.00000000')

figure(8);subplot(2,1,1);plot(w/pi,abs(H3));axis([0 1 0 1.1]);
title('Amplitude response for H(z): L_infinite scaling')
ylabel('Amplitude');xlabel('Angular freqeuncy omega/pi');
text(0.6,.5,'max=1.00000000')

GN3(1)=1;GD3=b3;
GN2=conv(GN3,a3);GD2=conv(GD3,b2);
GN1=conv(GN2,a2);GD1=conv(GD2,b1);

%Corresponding impulse responses

[gg1,t]=impz(GN1,GD1,501);[gg2,t]=impz(GN2,GD2,501);
[gg3,t]=impz(GN3,GD3,501);

%Sum of the squared responses

sqg1=(sum(gg1.*gg1));sqg2=(sum(gg2.*gg2));sqg3=(sum(gg3.*gg3));

%Overall noise
noise2=5*(sqg1+sqg2+sqg3);

%Noise
%Impulse responses: L_infinite-norm scaling
figure(9);subplot(2,1,1);impz(gg1);title('Impulse response for G1(z): L_infinite scaling');
ylabel('g1(n)');xlabel('n in samples');axis([0 100 -2 2]);
text(35,1.2,'sum of the g1(n)^2s = 8.7685193');
subplot(2,1,2);impz(gg2);title('Impulse response for G2(z): L_infinite scaling')
ylabel('g2(n)');xlabel('n in samples');axis([0 100 -2 2]);
text(35,1.2,'sum of the g2(n)^2s = 10.739175');

figure(10);subplot(2,1,1);impz(gg3);
title('Impulse response for G3(z): L_infinite scaling');
ylabel('g3(n)');xlabel('n in samples');axis([0 100 -1.1 1.1]);
text(35,.75,'sum of the g3(n)^2s = 5.0088751');
```

SCALING IN THE PARALLEL-FORM STRUCTURE



LIMIT CYCLES





- Consider a first-order system (see Figure (a)) described by the difference equation

$$y(n) = \alpha y(n-1) + x(n).$$

- In practical implementation, we use a (1+3)-bit representation for α, $x(n)$, and $y(n)$. In this case, $\alpha y(n-1)$ must be rounded to a (1+3)-bit representation, before addition to $x(n)$.

- The actual implementation is depicted in Figure (b), where the actual output $w(n)$ satisfies

$$w(n) = Q[\alpha w(n-1)] + x(n),$$

where $Q[\ ]$ represents the rounding operation.

- Assume that $x(n) = 7/8\delta(n) = 0_{\Delta}111\delta(n)$, $\alpha = 1/2 = 0_{\Delta}100$, and $w(-1) = 0$. Then, $w(0) = x(0) = 7/8 = 0_{\Delta}111$ and $\alpha w(0) = 0_{\Delta}011100 = 7/16$. Since $x(1) = 0$, $w(1) = Q[\alpha w(0)] = 0_{\Delta}100 = 1/2$. Continuing, $w(2) = Q[\alpha w(1)] = 0_{\Delta}010 = 1/4$ and $w(3) = 0_{\Delta}001 = 1/8$.

- To obtain $w(4)$, we must round a seven-bit number $\alpha w(3) = 0_{\Delta}000100$ to $0_{\Delta}001 = 1/8$. The same result is obtained for $n \geq 3$ (see the figure of the next page).

- For $\alpha = -1/2$, the same procedure can be carried out and finally the output oscillates between $1/8$ and $-1/8$ (see the figure of the next page). Note that for $\alpha = -1/2$, this result is obtained by rounding the magnitude, that is, for rounding for the sign and magnitude arithmetic (see transparency 2).

- These oscillations occuring at the filter output for ever when the exitation remains to be zero are called **limit cycle oscillations**.

RESULTING RESPONSES FOR $\alpha = 1/2$ and $\alpha = -1/2$

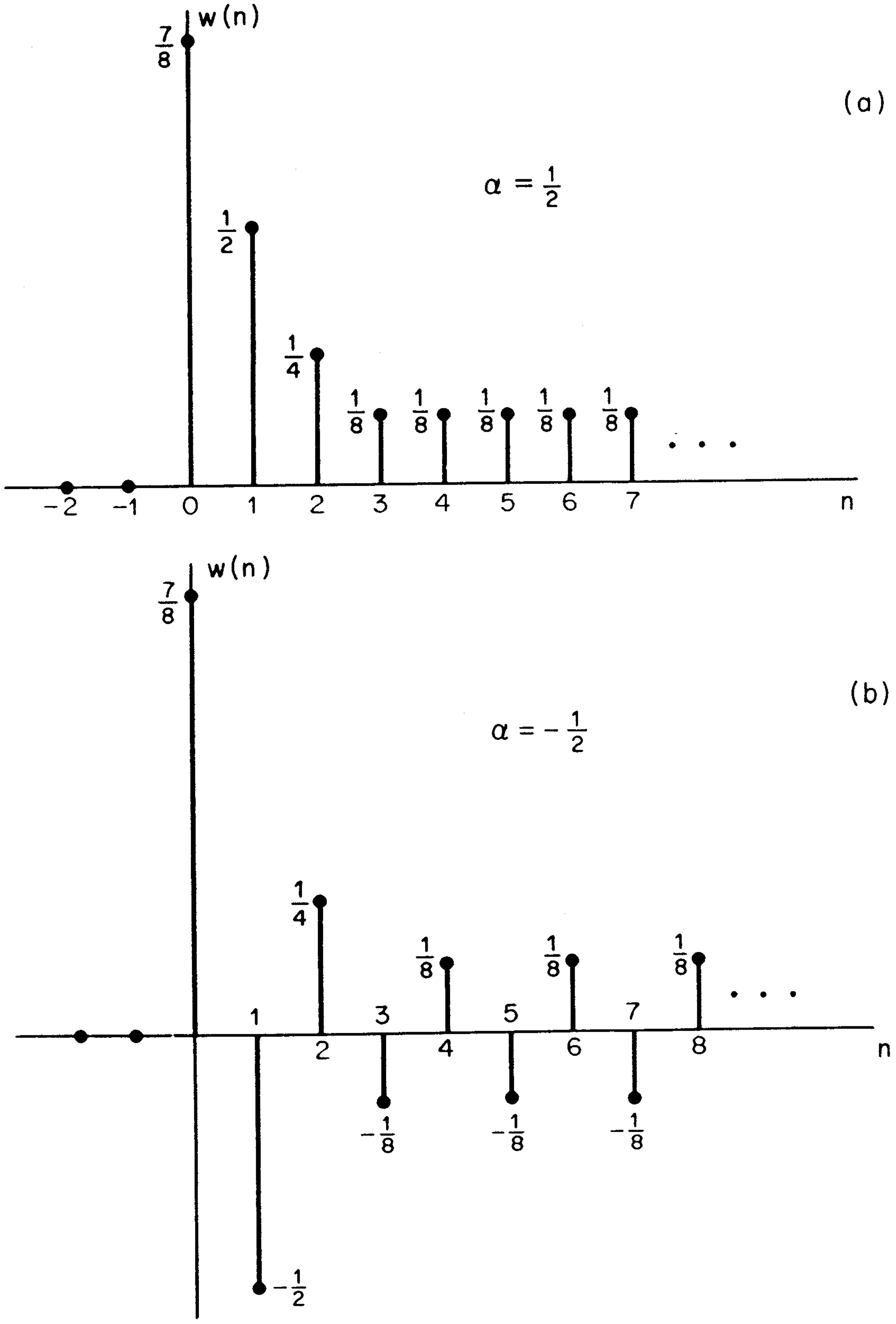


Fig. 9.6 Response of a first-order quantized system to a unit sample: (a) $\alpha = \frac{1}{2}$; (b) $\alpha = -\frac{1}{2}$.

LIMIT CYCLES DUE TO OVERFLOW

- Consider a second-order system characterized by the difference equation
$$w(n) = Q[a_1 w(n-1)] + Q[a_2 w(n-2)] + x(n),$$
where $Q[\ ]$ represents two's complement rounding with a (1+3)-bit wordlength. It is assumed that for negative numbers rounding is performed such that the magnitude is rounded. (This is not always the case!!)

- $a_1 = 3/4 = 0_\Delta 110$, $a_2 = -3/4 = 1_\Delta 010$, $w(-1) = 3/4 = 0_\Delta 110$, $w(-2) = -3/4 = 1_\Delta 010$, $x(n) = 0,\ n \geq 0$.

- Now, the output is
$$\begin{aligned} w(0) =& Q[9/16] + Q[9/16] \\ =& Q[0_\Delta 100100] + Q[0_\Delta 100100] \\ =& 0_\Delta 101 + 0_\Delta 101 = 1_\Delta 010 = -3/4 \\ w(1) =& Q[-9/16] + Q[-9/16] \\ =& Q[1_\Delta 011100] + Q[1_\Delta 01110] \\ =& 1_\Delta 011 + 1_\Delta 011 = 0_\Delta 110 = +3/4 \end{aligned}$$

- The output will continue oscillating between $-3/4$ and $3/4$, that is, between very high amplitude values, until an input is applied.

- The oscillations of the above kinds can be avoided by using more complicated filter structures, using the saturation arithmetic, and using more bits for internal calculations (one topic of the course “System Level DSP Algorithms”).

COMMENTS

- The above examples on limit cycles were for very simple filters.
- For more complicated filters, they can be more severe.
- The most severe are huge overflow oscillations which may occur in badly scaled filters. In this case, the filter output is useless and in many cases these oscillations can be killed only by resetting the state variables (data samples in delays).
- In addition to limit cycles, there are parasitic oscillations inside the filter.
- These are extra oscillations due to the finite word length effects, but they are not usually visible at the filter output in the normal operation.
- They can be found out by simulations and are considered in more details in the course “System Level DSP Algorithms”.

COEFFICIENT ROUNDING EFFECTS IN CASCADE-FORM IIR FILTERS

- Several examples are given in the couse "System Level DSP Algorithms".

- Here we consider the filter of transparency 45 that is scaled according to the L_∞ norm (see transparency 55).

- For the infinite-precision filter, the passband and stopband ripples are $A_p = 0.5$ dB and $A_s = 80.24$ dB, respectively.

- A satisfactory result is obtained by using 6 decimal bits for the coefficients. The resulting values are

$$S = 2^{-2},$$

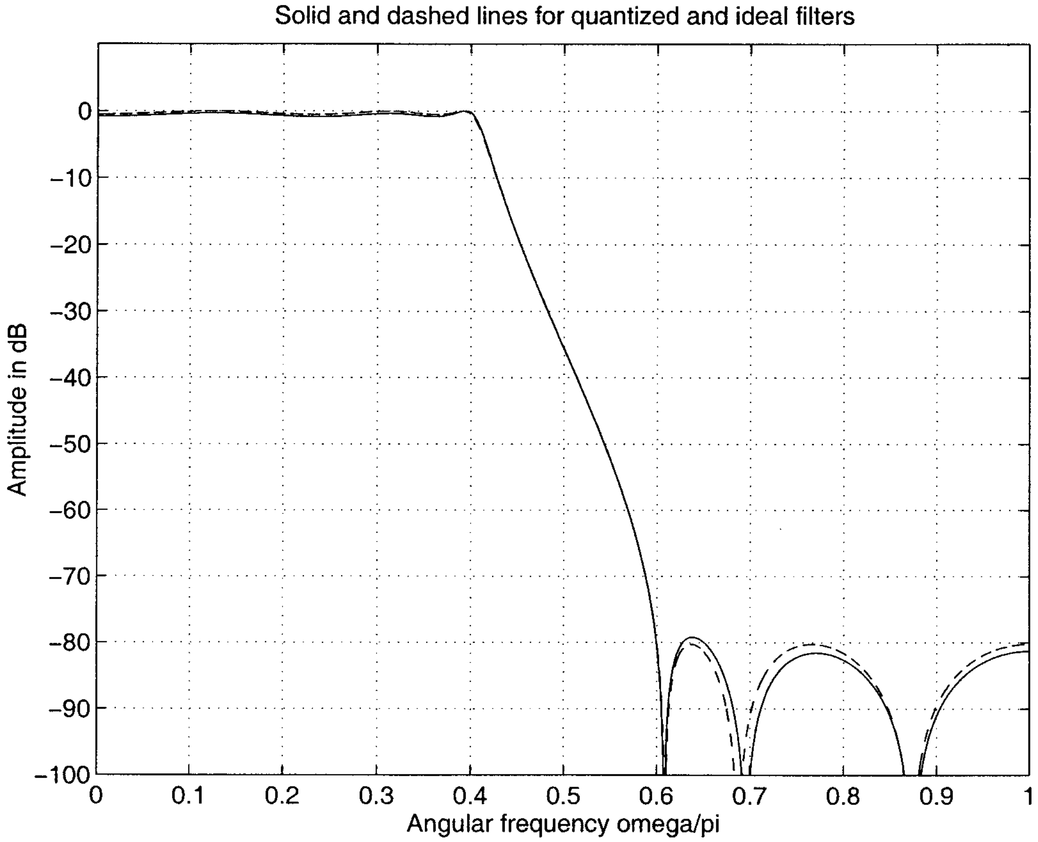
$$\begin{aligned} a_{01} = a_{21} &= 13 \cdot 2^{-6}, \quad a_{11} = 24 \cdot 2^{-6}, \\ a_{02} = a_{22} &= 13 \cdot 2^{-6}, \quad a_{12} = 15 \cdot 2^{-6}, \\ a_{03} = a_{23} &= 48 \cdot 2^{-6}, \quad a_{13} = 32 \cdot 2^{-6}, \\ b_{11} = 75 \cdot 2^{-6}, &\quad b_{21} = -27 \cdot 2^{-6}, \\ b_{12} = 52 \cdot 2^{-6}, &\quad b_{22} = -41 \cdot 2^{-6}, \end{aligned}$$

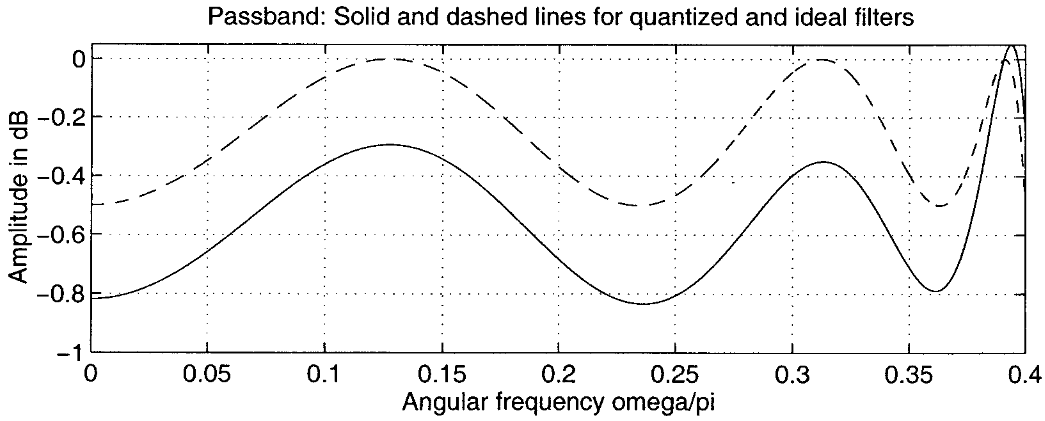
and

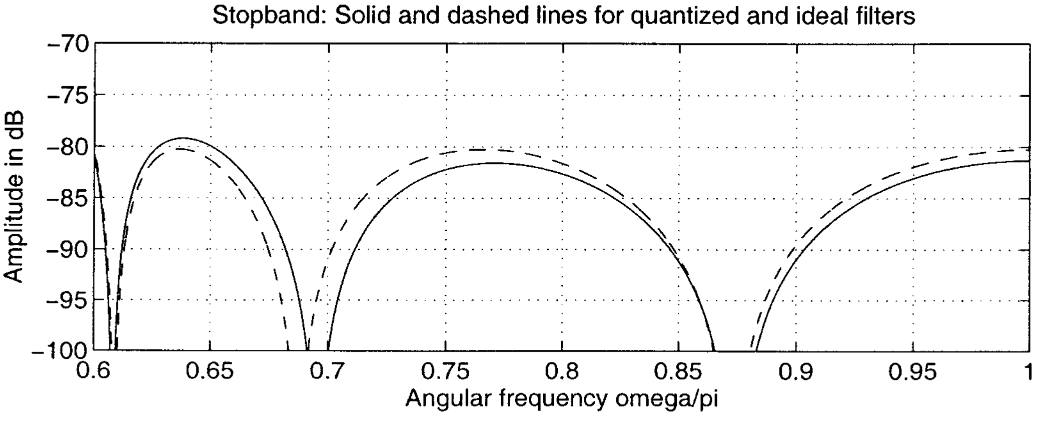
$$b_{13} = 36 \cdot 2^{-6}, \quad b_{23} = -57 \cdot 2^{-6}.$$

- As seen from the next transparency, in the passband the resulting response varies between 0.05 dB and -0.83 dB, and the minimum stopband attenuation is 79.2 dB.

RESPONSES FOR QUANTIZED (SOLID LINE) AND UNQUANTIZED (DASHED LINE) IIR FILTERS


Solid and dashed lines for quantized and ideal filters


Passband: Solid and dashed lines for quantized and ideal filters


Stopband: Solid and dashed lines for quantized and ideal filters

COMMENTS

- The above filter had a rather wide passband region. This is why the number of decimal bits was rather small to achieve a satisfactory overall response.

- For very narrowband cases, significantly more bits are required.

- It should also be pointed out that the direct-form structures require significantly more bits for the coefficient representations, and due to the fact that some coefficient values are very large, several integer bits are required.

COEFFICIENT ROUNDING EFFECTS IN LINEAR-PHASE FIR FILTERS

- In the course “System Level DSP Algorithms” it is shown that a filter with b decimal bits for the rounded coefficient values can be designed in most cases as follows:

Step 1: Determine

$$\delta_e = 2^{-(b+1)}[(N+1)\log_e(N+1)/3]^{1/2},$$

where N is the filter order.

Step 2: Design the minimum-order linear phase FIR filter for the passband and stopband ripples of $\delta_p - \delta_e$ and $\delta_s - \delta_e$, respectively.

Step 3: Round the coefficients of this filter to b decimal bits and check whether the resulting filter meets the original ripple requirements of δ_p and δ_s.

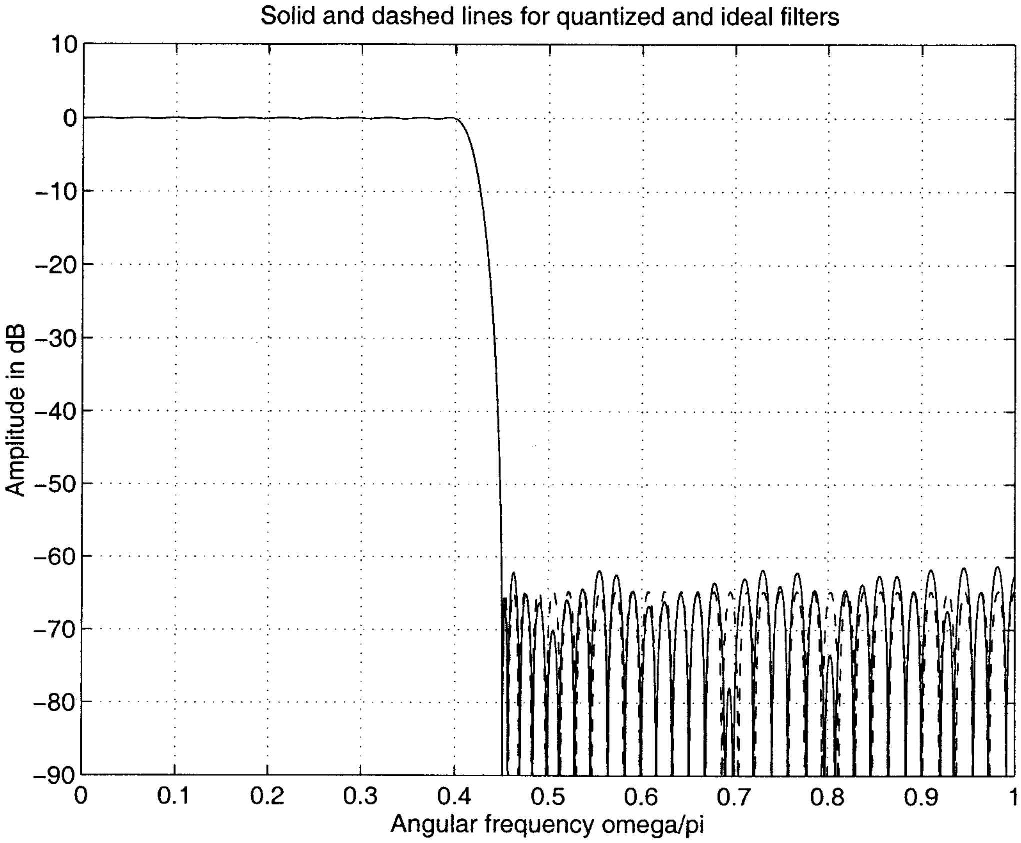
- It is desired to design a lowpass filter with edges at $\omega_p = 0.4\pi$ and $\omega_s = 0.45\pi$ and ripples of $\delta_p = 0.01$ and $\delta_s = 0.001$. The minimum order to meet these criteria is $N = 104$

- For $b = 14$, the above formula gives approximately $\delta_e = 0.0004$ so that the passband and stopband ripples for the infinite-precision filter are 0.0096 and 0.006, respectively. The minimum order to meet these criteria is 110.

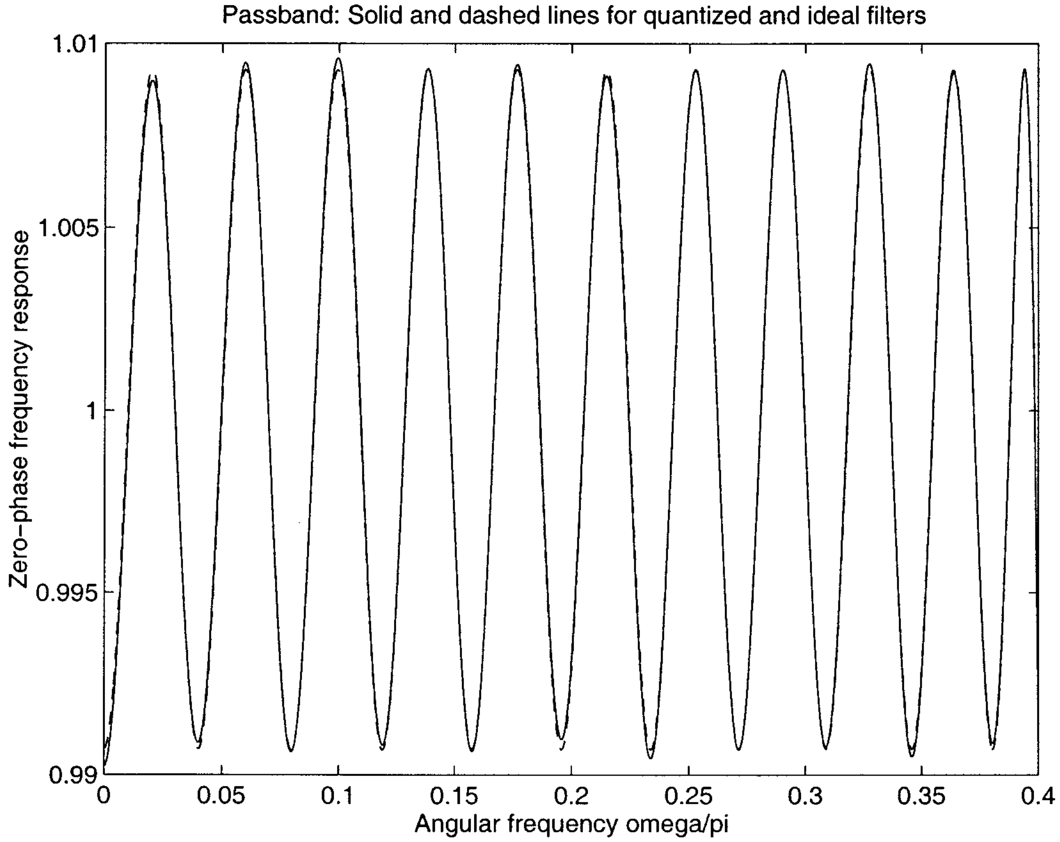
- The following two transparencies show the responses for the quantized and unquantized filter and indicate that the quantized filter meets the given criteria.

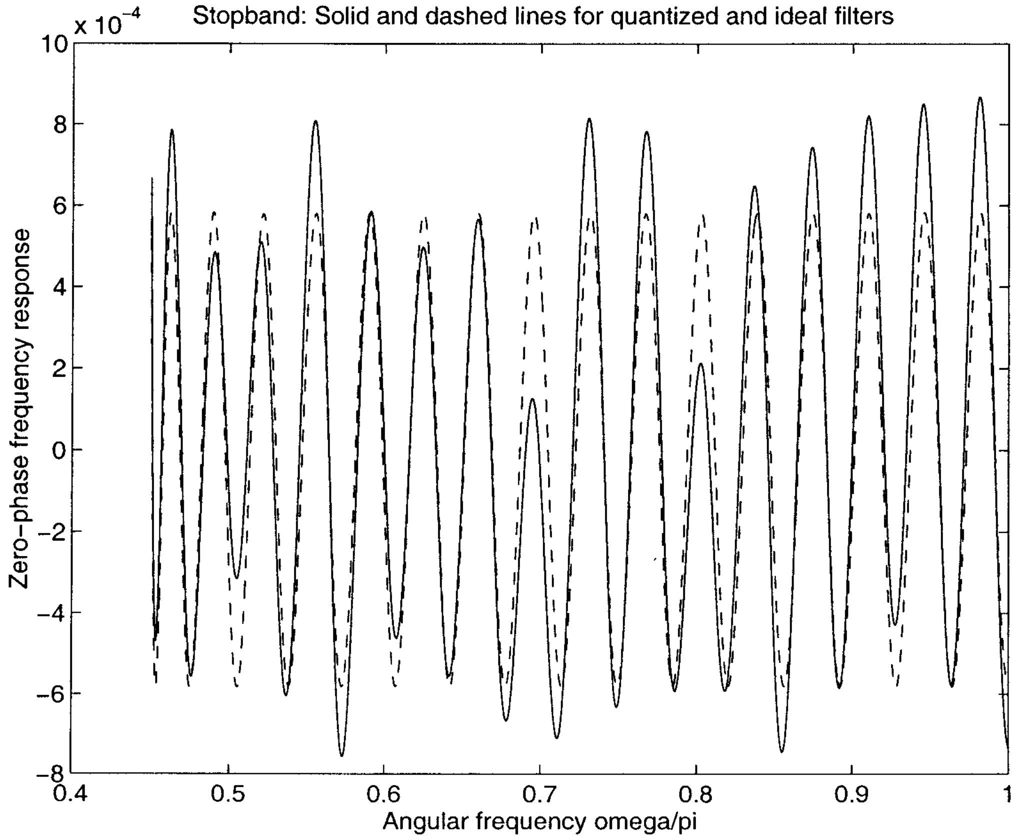
vfill

RESPONSES FOR QUANTIZED (SOLID LINE) AND UNQUANTIZED (DASHED LINE) FIR FILTERS



RESPONSES FOR QUANTIZED (SOLID LINE) AND UNQUANTIZED (DASHED LINE) FIR FILTERS





APPENDIX A: DISCRETE-TIME RANDOM SIGNALS

- Up to now, we have assumed that the discrete-time signals have been determisitic, that is, each value of the sequence under consideration is uniquely determined by a mathematical expression, a table of data, or a rule of some type.

- However, there are several situations, such as the error signal resulting after truncation or rounding a signal to a fewer number of bits than the original signal, where the signal is not at all deterministic.

- The purpose of this appendix is to give a short review on how to treat mathematically signals that are random, that is, they do not obey any strict mathematical expression.

- It should be pointed out that the purpose of this appendix is not to give a deep mathematical frameform on how to model discrete-time random signals.

- We just concentrate on the basics on how to model the error signal resulting when quantizing the signal obtained by multiplying it by a coefficient. In addition, it is considered what happens to this error signal when it travels to the output through a filter with a transfer function $H_e(z)$ or an impulse response $h_e(n)$.

RANDOM SIGNALS

- Consider a random signal $e(n)$. There exist several ways of modeling this signal.

- The **mean value** m_e of the signal $e(n)$ is the following quantity
$$m_e = E[e(n)],$$
where $E[x]$ stands for the expected value of x.

- The **covariance sequence** $c_{ee}(l)$ and the **autocorrelation sequence** $\phi_{ee}(l)$ of our signal are defined by
$$c_{ee}(l) = E[\{e(n) - m_e\}\{e(n+l) - m_e\}]$$
and
$$\phi_{ee}(l) = E[e(n)e(n+l)],$$
respectively.

- The **variance** (power) σ_e^2 of the signal $e(n)$ is the following quantity:
$$\sigma_e^2 = c_{ee}(0) = E[\{e(n) - m_e\}^2].$$

- The **power density spectrum** of our signal $e(n)$ is defined by
$$\Phi_{ee}(\omega) = \sum_{l=-\infty}^{\infty} \phi_{ee}(l)e^{-jl\omega}.$$

- If $\Phi_{ee}(\omega)$ is known, then
$$\phi_{ee}(l) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \Phi_{ee}(\omega)e^{jl\omega}d\omega.$$

- If $e(n)$ is filtered with a filter with the transfer function $H(z)$, then the power density spectrum of the output random signal, denoted by $f(n)$, is given by

$$\Phi_{ff}(\omega) = |H(e^{j\omega})|^2 \Phi_{ee}(\omega).$$

ROUNDING AND TRUNCATION IN TWO'S COMPLEMENT ARITHMETIC: OUTPUT NOISE

- When $(1+b)$-bit data samples $w(n)$ are multiplied by a $(1+a)$-bit coefficient α, then the resulting sequence $\alpha w(n)$ consists of $(1+a+b)$-bit numbers.

- If this number is rounded back to a $(1+b)$-bit data sample in two's complement arithemetic, then it is common to assume that we generate an error sequence $e(n)$ satisfying

$$m_e = 0,$$

$$\sigma_e^2 = 2^{-2b}/12,$$

  and

$$c_{ee}(l) = \sigma_e^2 \delta(l),$$

  indicating that there is no correlation between $e(n)$ and $e(n+l)$ for $l \neq 1$.

- For truncation the same is valid except that

$$m_e = -2^{-b}/2.$$

- For the above two cases, the autocorrelation sequency is given by

$$\phi_{ee}(l) = \sigma_e^2 \delta(l) + m_e^2.$$

- In the frequency domain, our error sequence $e(n)$ consists of the following two parts:

1) It has a DC-component equal to m_e.

2) The autocorrelation sequence of $e(n) - m_e$ is

$$\widehat{\phi}_{ee}e(l) = \sigma_e^2 \delta(l),$$

so that the corresponding power density spectrum is given by

$$\widehat{\Phi}_{ee}(\omega) = \sum_{l=-\infty}^{\infty} \widehat{\phi}_{ee}(l) e^{-jl\omega} \equiv \sigma_e^2.$$

- We next consider the properties of a signal $f(n)$ which is related to $e(n)$ through

$$f(n) = \sum_{k=0}^{\infty} h_e(k) e(n-k),$$

that is, our random signal goes trough a filter with an impulse response $h_e(n)u(n)$ or a transfer function given by

$$H_e(z) = \sum_{k=0}^{\infty} h_e(k) z^{-k}.$$

- The mean value of $f(n)$ is given by

$$\begin{aligned} m_f = E[f(n)] &= E[\sum_{k=0}^{\infty} h_e(k) e(n-k)] \\ &= \sum_{k=0}^{\infty} h_e(k) E[e(n-k)]. \end{aligned}$$

- Since $E[e(n-k)] \equiv m_e$, we obtain
$$m_f = m_e \sum_{k=0}^{\infty} h(k) = m_e H_e(e^{j0}).$$

- The covariance sequence of $f(n)$ is given by
$ccff(l) = E[\{f(n) - m_f\}\{f(n+l) - m_f\}]$
$$=E[\sum_{k=0}^{\infty}\sum_{r=0}^{\infty} h_e(k)h_e(r)\{e(n-k) - m_e\}\{e(n+l-r) - m_e\}]$$
$$=\sum_{k=0}^{\infty} h_e(k) \sum_{r=0}^{\infty} h_e(r)E[\{e(n-k) - m_e\}\{e(n+l-r) - m_e\}]$$

- Since $E[\{e(n-k) - m_e\}\{e(n+l-r) - m_e\}] = c_{ee}(k+l-r)$,
$$c_{ff}(l) = \sum_{k=0}^{\infty} h_e(k) \sum_{r=0}^{\infty} h_e(r)c_{ee}(k+l-r).$$

- By making the substitution $s = r - k$, we obtain
$$c_{ff}(l) = \sum_{s=-k}^{\infty} c_{ee}(l-s) \sum_{k=0}^{\infty} h(k)h(s+k) = \sum_{s=-k}^{\infty} c_{ee}(l-s)C(s),$$
where
$$C(s) = \sum_{k=0}^{\infty} h(k)h(s+k).$$

- Since $c_{ee}(l-s) = \sigma_e^2\delta(l-s)$, $c_{ff}(l)$ can be expressed as
$$c_{ff}(l) = \sigma_e^2 \sum_{s=-k}^{\infty} \delta(l-s)C(s) = \sigma_e^2 C(l).$$

- This shows that there is a correlation between $f(n)$ and $f(n+l)$.

- The autocorrelation sequence of $f(n)$ is then

$$\phi_{ff}(l) = m_f^2 + c_{ff}(l) = m_f^2 + \sigma_e^2 C(l).$$

- The variance of $f(n)$ is thus given by

$$\sigma_f^2 = c_{ff}(0) = \sigma_e^2 \sum_{k=0}^{\infty} h_e^2(k).$$

- In the frequency domain, the output error sequence $f(n)$ consists of following two parts:

  1) It has a DC-component equal to m_f.

  2) The power density spectrum of $f(n) - m_f$ is given by

$$\widehat{\Phi}_{ff}(\omega) = |H_e(e^{j\omega})|^2 \widehat{\Phi}_{ee}(\omega) = |H_e(e^{j\omega})|^2 \sigma_e^2.$$

- If desired, then the variance σ_f^2 of $f(n)$ can be evaluated from

$$\sigma_f^2 = \frac{\sigma_e^2}{2\pi} \int_{-\pi}^{\pi} |H_e(e^{j\omega})|^2 d\omega \equiv \sigma_e^2 \sum_{n=0}^{\infty} h_e^2(n).$$

- The equivalence

$$\int_{-\pi}^{\pi} |H_e(e^{j\omega})|^2 d\omega = \sum_{n=0}^{\infty} h_e^2(n)$$

is the Parseval theorem for the discrete-time signals.

80509 LINEAR DIGITAL FILTERING I

PART V: Finite word length effects in digital filters:

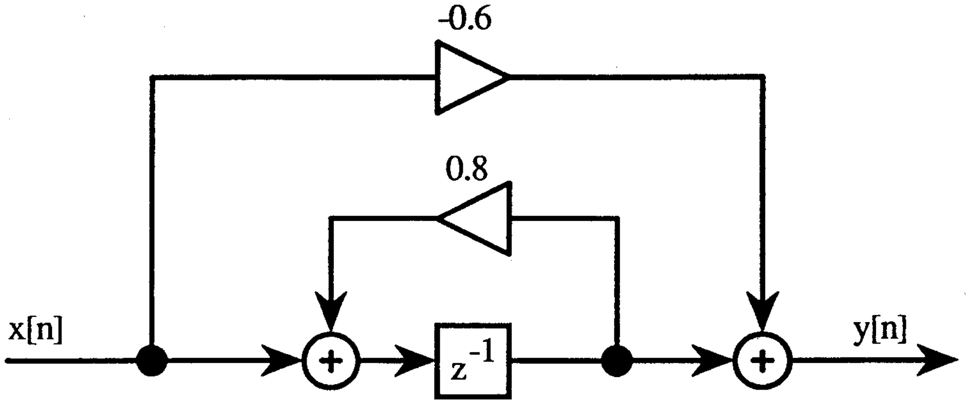
APPENDIX B

Two simple exercises on scaling a filter and evaluating the output noise due to the multiplication roundoff errors

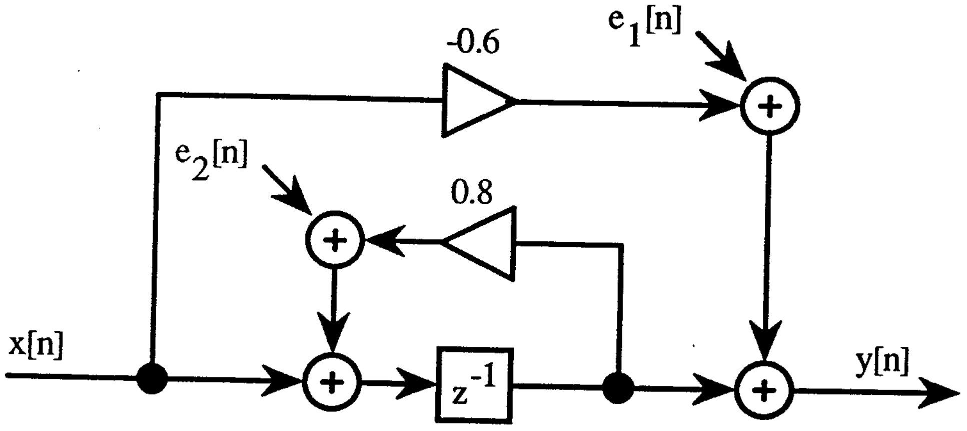
Example 1: The filter shown below is implemented using (8+1)-bit fixed-point arithmetic.

(a) Evaluate the output noise variance due to the multiplication roundoff errors.

(b) The input signal is $x[n] = A\sin(n\pi/10)$. Determine the highest value of A for which there are no overflows. For this value of A, determine the signal-to-noise ratio at the filter output.



1(a):



$$\sigma_{e_1}^2 = \sigma_{e_2}^2 = \frac{2^{-2b}}{12} = \frac{2^{-16}}{12}.$$

The output noise variance due to the multiplication roundoff errors is

$$\sigma_f^2 = \frac{2^{-16}}{12}\Big(\sum_{k=0}^{\infty} h_1^2[k] + \sum_{k=0}^{\infty} h_2^2[k]\Big),$$

where $h_1[k]$ ja $h_2[k]$ are the impulse responses from the noise sources $e_1[n]$ ja $e_2[n]$ to the filter output.

$$h_1[n] = \delta[n] \Rightarrow \sum_{n=0}^{\infty} h_1^2[n] = 1.$$

$$h_2[n] = (8/10)^{n-1} u[n-1]$$

$$\Rightarrow h_2^2[n] = ((8/10)^2)^{n-1} u[n-1]$$

$$\Rightarrow \sum_{n=0}^{\infty} h_2^2[n] = \frac{1}{1-(8/10)^2} = 2.778.$$

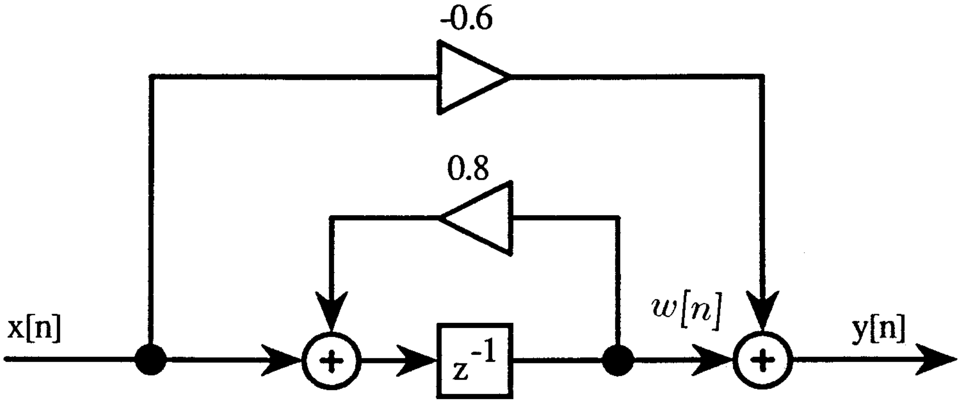
The output noise variance is thus given by

$$\sigma_f = (1+2.778)\frac{2^{-16}}{12} = 3.778\frac{2^{-16}}{12}.$$

1(b):

$$x[n] = A\sin(\frac{\pi}{10}n)$$

There are no overflows if $|w[n]| < 1$ and $|y[n]| < 1$ for all n (see the following figure).



$w[n]$ and $y[n]$ are given by

$$w[n] = A|H'(e^{j\pi/10})|\sin\Big(\frac{\pi}{10}n + \arg H'(e^{j\pi/10})\Big)$$

and

$$y[n] = A|H(e^{j\pi/10})|\sin\Big(\frac{\pi}{10}n + \arg H(e^{j\pi/10})\Big),$$

respectively, where

$$H'(z) = \frac{z^{-1}}{1 - 0.8z^{-1}}$$

and

$$H(z) = \frac{z^{-1}}{1 - 0.8z^{-1}} - 0.6 = \frac{-0.6 + 1.48z^{-1}}{1 - 0.8z^{-1}}$$

are the transfer functions from the filter input to $w[n]$ and $y[n]$, respectively. Therefore, it is required that

$$A|H'(e^{j\pi/10})| < 1 \quad \text{and} \quad A|H(e^{j\pi/10})| < 1,$$

that is,

$$A < \min\{1/|H'(e^{j\pi/10})|,\ 1/|H(e^{j\pi/10})|\}.$$

$$\begin{aligned}|H'(e^{j\pi/10})| &= \frac{|\cos(\pi/10) - j\sin(\pi/10)|}{|1 - 0.8[\cos(\pi/10) - j\sin(\pi/10)]|} \\ &= \frac{1}{|0.2392 + j0.2472|} = 2.907\end{aligned}$$

and

$$\begin{aligned}|H(e^{j\pi/10})| &= \frac{|-0.6 + 1.48[\cos(\pi/10) - j\sin(\pi/10)]|}{|1 - 0.8[\cos(\pi/10) - j\sin(\pi/10)]|} \\ &= \frac{|0.8076 - j0.4573|}{|0.2392 + j0.2472|} = 2.698.\end{aligned}$$

$\Rightarrow$ There are no overflows if $A_{\text{max}} < 1/2.907 = 0.3440$.

At the filter output, the amplitude of the response of the exitation $x[n] = A_{\text{max}}\sin(\frac{\pi}{10}n)$ is given by

$$A_{max}|H(e^{j\pi/10})| = 0.3440 \cdot 2.698 = 0.9281.$$

The power of this signal is

$$A^2/2 = 0.4307,$$

whereas the output noise power is

$$\sigma_f = 3.778\frac{2^{-16}}{12}$$

Therefore, the signal-to-noise ratio is given by

$$S/N \text{ [dB]} = 10 \cdot \log_{10} \frac{0.4307}{3.778 \cdot \frac{2^{-16}}{12}} = 49.5 \text{ dB}.$$

Example 2: It is desired to implement the transfer function

$$H(z) = \frac{0.02}{(1 - 0.9z^{-1})(1 - 0.8z^{-1})}$$

using the fixed-point arithmetic.

(a) The filter is desired to be implemented as a cascade of two first-order blocks. Scale the filter using the worst-case scaling and select the order of the blocks to minimize the output noise variance. Repeat the problem using L_∞- and L_2-norm scalings.

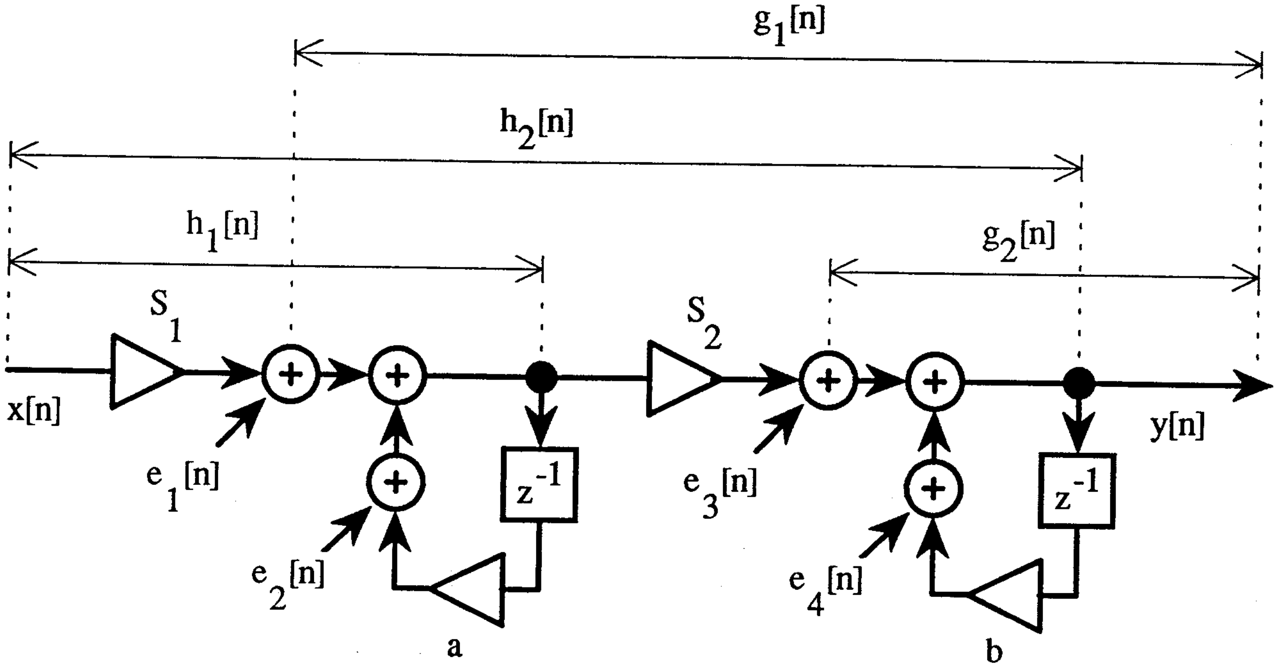
(b) The filter is implemented directly as a single second-order block. Scale the filter and evaluate the output noise variance.

2(a)

The scaling constants as well as the noise sources are shown in the figure of the next page. $h_1[n]$ ja $h_2[n]$ are the impulse responses of the scaling transfer functions, whereas $g_1[n]$ ja $g_2[n]$ are the impulse responses of the noise transfer functions. Note that also the scaling coefficients cause noise.

Consider first the scaling transfer functions which are exploited later.

$$h_1(n) = S_1 a^n u[n]$$



$$Z(h_2[n]) = H_2(z) = \frac{S_1S_2}{(1-az^{-1})(1-bz^{-1}}$$
$$= \frac{S_1S_2}{1-a/b}\frac{1}{1-bz^{-1}} + \frac{S_1S_2}{1-b/a}\frac{1}{1-az^{-1}}$$

⇒

$$h_2[n] = \frac{S_1S_2}{b-a}\Big(b^{n+1}u[n] - a^{n+1}u[n]\Big).$$

When determining the output noise variance it is observed that the impulse response from the noise sources $e_1[n]$ ja $e_2[n]$ to the filter output are the same ($g_1[n]$). Also, $e_3[n]$ and $e_4[n]$ have the same impulse response ($g_2[n]$).

The overall output noise variance is given by

$$\sigma_f^2 = 2\sigma_e^2 \sum_{n=0}^{\infty} \Big(g_1^2[n] + g_2^2[n]\Big).$$
$$g_1[n] = h_2[n]/S_1$$

⇒

$$\sum_{n=0}^{\infty} g_1^2[n] = \Big(\frac{S_2}{b-a}\Big)^2 \sum_{n=0}^{\infty} \Big(b^{n+1} - a^{n+1}\Big)^2$$

$$= \Big(\frac{S_2}{b-a}\Big)^2 \sum_{n=0}^{\infty}\Big((b^2)^{n+1} + (a^2)^{n+1} - 2(ab)^{n+1}\Big)^2$$

$$= \Big(\frac{S_2}{b-a}\Big)^2 \Big[\frac{b^2}{1-b^2} + \frac{a^2}{1-a^2} - \frac{2ab}{1-ab}\Big].$$

For both $a = 0.8$, $b = 0.9$ and $a = 0.9$, $b = 0.8$,

$$\sum_{n=0}^{\infty} g_1^2[n] = 89.82S_2^2.$$

$$g_2(n) = b^n u[n]$$

⇒

$$\sum_{n=0}^{\infty} g_2^2[n] = \sum_{n=0}^{\infty}(b^2)^n = \frac{1}{1-b^2}.$$

The overall output noise variance is thus given by

$$\sigma_f^2 = \sigma_e^2\Big(179.62S_2^2 + \frac{2}{1-b^2}\Big).$$

Consider first the worst-case scaling, where the coefficients S_1 ja S_2 are determined such that

$$\sum_{n=0}^{\infty} |h_1[n]| = \sum_{n=0}^{\infty} |h_2[n]| = 1.$$

In this case there are no overflows at all.

Case $a = 0.9$ and $b = 0.8$:

$$\sum_{n=0}^{\infty} |h_1[n]| = S_1 \sum_{n=0}^{\infty} 0.9^n = S_1\frac{1}{1-0.9} = 10S_1 = 1$$

⇒

$$S_1 = 0.1.$$

$$\sum_{n=0}^{\infty} |h_2[n]| = 10S_1S_2 \sum_{n=0}^{\infty} |0.8^{n+1} - 0.9^{n+1}|$$

$$= 10S_1S_2 \sum_{n=0}^{\infty} (0.9^{n+1} - 0.8^{n+1})$$

$$= S_2\Big(\frac{0.9}{1-0.9} - \frac{0.8}{1-0.8}\Big) = 5S_2$$

$\Rightarrow$

$$S_2 = 0.2.$$

We get

$$\sigma_f^2 = 12.74\sigma_e^2.$$

Case $a = 0.8$ and $b = 0.9$:

$$\sum_{n=0}^{\infty} |h_1[n]| = 5S_1$$

$\Rightarrow$

$$S_1 = 0.2.$$

$$\sum_{n=0}^{\infty} |h_2[n]| = 10S_1S_2 \cdot 5 = 10S_2$$

$\Rightarrow$

$$S_2 = 0.1.$$

We get

$$\sigma_f^2 = 12.32\sigma_e^2.$$

Case $a = 0.8$ and $b = 0.9$ is thus slightly better. Note that since in both of the above cases $S_1S_2 = 0.02$ no additional scaling coefficient is needed at the filter output to make the numerator of the overall transfer function equal to 0.02.

Consider next the L_∞-norm scaling, where S_1 ja S_2 are determined such that

$$\max_{\omega \in [0,\ \pi]} |H_1(e^{j\omega})| = 1$$

and

$$\max_{\omega \in [0,\ \pi]} |H_2(e^{j\omega})| = 1,$$

where

$$H_1(z) = \frac{S_1}{(1 - az^{-1})}$$

and

$$H_2(z) = \frac{S_1 S_2}{(1 - az^{-1})(1 - bz^{-1})}.$$

In this case, no single sinusoidal signal causes overflows. The maxima of both $|H_1(e^{j\omega})|$ and $|H_2(e^{j\omega})|$ occur at $\omega = 0$. Therefore, it is required that $H_1(1) = S_1/(1-a)$ and $H_2(1) = S_1 S_2/[(1-a)(1-b)]$, yielding $S_1 = (1-a)$ and $S_2 = (1-b)$. These are the same scaling constants as for the worst-case scaling. Therefore, in this example, the scaling for both the worst-case and L_∞-norm scalings is the same and the above results apply also to the L_∞-norm scaling. The explanation to this is that the worst-case input signal is $x[n] = 1$ or $x[n] = -1$ which are also sinusoidal signals of frequency $\omega = 0$.

Consider finally the L_2-norm scaling, where S_1 ja S_2 are determined such that

$$\sum_{n=0}^{\infty} h_1^2[n] = \sum_{n=0}^{\infty} h_2^2[n] = 1.$$

In this case, overflows are possible. However, if the input signal is random enough, overflows are not likely to occur.

Case $a = 0.9$ and $b = 0.8$:

$$\sum_{n=0}^{\infty} h_1^2[n] = S_1^2 \sum_{n=0}^{\infty} 0.9^{2n} = S_1^2 \frac{1}{1-0.9^2} = 5.263 S_1^2 = 1$$

$\Rightarrow$

$$S_1 = 0.4359.$$

According to the previous considerations,

$$h_2[n] = S_1 g_1[n] \quad \text{and} \quad \sum_{n=0}^{\infty} g_1^2[n] = 89.81 S_1^2.$$

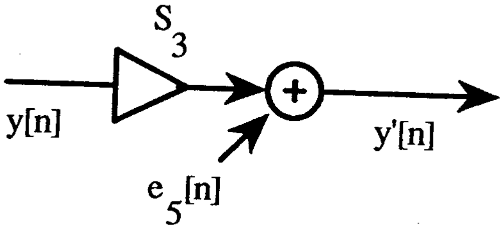
Hence,

$$\sum_{n=0}^{\infty} h_2^2[n] = S_1^2 S_2^2 89.81 = 17.06 S_2^2 = 1$$

$\Rightarrow$

$$S_2 = 0.2421.$$

In order to guarantee that the overall transfer has the desired constant 0.02 in the numerator, there is a need to have at the output an additional coefficient as shown in the following figure.



S_3 is determined such that $S_1 S_2 S_3 = 0.02$.

$\Rightarrow S_3 = 0.1895.$

The output noise variance is now given by

$$(\sigma_f')^2 = \sigma_e^2 + S_3^2 \sigma_f^2 = \sigma_e^2 (1 + S_3^2 (179.62 S_2^2 + \frac{2}{1-b^2})) = 1.578 \sigma_e^2.$$

Case $a = 0.8$ and $b = 0.9$:

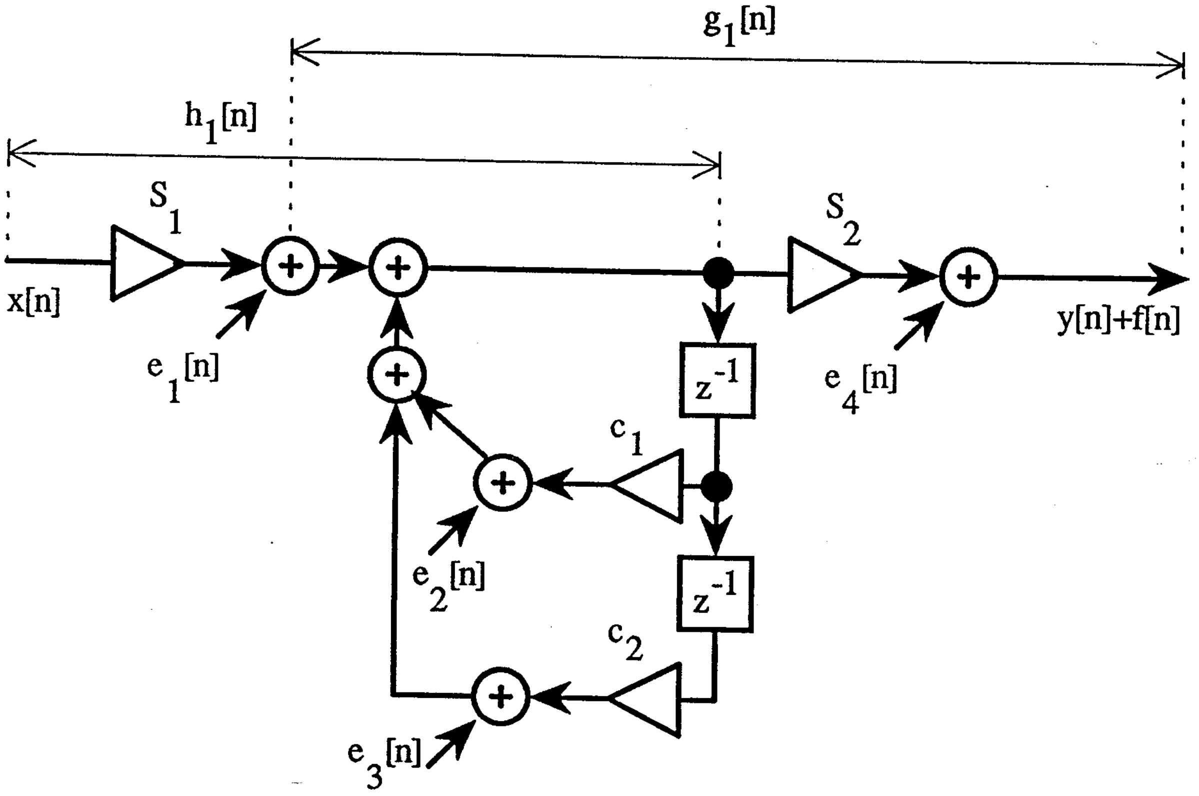
We obtain $S_1 = 0.6$, $S_2 = 0.1759$, $S_3 = 0.1895$, and

$$(\sigma'_f)^2 = 1.578\sigma_e^2.$$

Both of the above two cases give thus the same output noise variance.

It is observed that with the L_2-norm scaling the output noise variance is approximately 8 dB lower than with the worst-case or L_∞-norm scaling at the expense of possible overflows.

2(b):



$$H(z) = \frac{0.02}{(1-0.9z^{-1})(1-0.8z^{-1})} = \frac{0.02}{1-1.7z^{-1}+0.72z^{-2}}$$

Scaling:

$$H(z) = \frac{S_1}{(1-0.9z^{-1})(1-0.8z^{-1})}$$

⇒

$$\begin{aligned} h[n] &= \frac{S_1}{b-a}(b^{n+1} - a^{n+1})u[n] \\ &= \frac{10S_1}{b-a}(0.9^{n+1} - 0.8^{n+1})u[n]. \end{aligned}$$

Consider first the worst-case scaling, where S_1 is determined to give

$$\sum_{n=0}^{\infty} |h[n]| = 1.$$

$$\sum_{n=0}^{\infty} |h[n]| = 50S_1 = 1.$$

⇒

$$S_1 = 0.02.$$

$S_1 S_2 = 0.02 \Rightarrow S_2 = 1$. Since $S_2 = 1$, this multiplier as well as $e_4[n] = 0$ are absent.

Consider next the L_∞-norm scaling, where S_1 is determined such that

$$\max_{\omega \in [0,\ \pi]} |H(e^{j\omega})| = H(1) = \frac{S_1}{0.1 \cdot 0.2} = 1$$

⇒

$$S_1 = 0.02.$$

Again, the scaling for the worst-case and the L_∞-norm scalings is the same.

Consider finally the L_2-norm scaling, where S_1 is determined such that

$$\sum_{n=0}^{\infty} h^2[n] = 1.$$

$$\sum_{n=0}^{\infty} h^2[n] = 90S_1^2 = 1.$$

$\Rightarrow$

$$S_1 = 0.1111, \quad S_2 = 0.18$$

Output noise:

$$G(z) = \frac{S_2}{(1-0.9z^{-1})(1-0.8z^{-1})}$$

$$\sum_{n=0}^{\infty} g^2[n] = 90S_2^2.$$

The output noise variance for both the worst-case scaling and the L_∞-norm scaling is given by ($S_2 \equiv 1$ as well as $e_4[n] = 0$ are absent) is given by

$$\sigma_f^2 = 3\sigma_e^2 \cdot 90S_2^2 = 270\sigma_e^2$$

The corresponding variance for the L_2-norm scaling is

$$\sigma_f^2 = 3\sigma_e^2 \cdot 90S_2^2 + \sigma_e^2 = 9.748\sigma_e^2.$$

It is seen that for all the scaling norms, the cascade-form realization gives a lower output noise variance. For the worst-case and L_∞-norm scalings, the output noise variance is $10 \cdot \log_{10} 270/12.32 = 13.4$ dB lower for the cascade-form realization. Since a 6-dB reduction means a one-bit saving in the number of data bits, the cascade-form realization requires two bits less.