Advanced Digital Signal Processing

Abdellatif Zaidi †
Department of Electrical Engineering
University of Notre Dame
azaidi@nd.edu

Outline:

1. **Introduction**
2. **Digital processing of continuous-time signals**
   - Retition: Sampling and sampling theorem
   - Quantization
   - AD- and DA-conversion
3. **DFT and FFT**
   - Leakage effect
   - Windowing
   - FFT structure
4. **Digital filters**
   - FIR-filters: Structures, linear phase filters, least-squares frequency domain design, Chebyshev approximation
   - IIR-filters: Structures, classical analog lowpass filter approximations, conversion to digital transfer functions
   - Finite word-length effects
5. **Multirate digital signal processing**
   - Decimation and interpolation
   - Filters in sampling rate alteration systems
   - Polyphase decomposition and efficient structures
   - Digital filter banks

Parts of this textbook have been realized in close collaboration with Dr. Joerg Kliewer whom I warmly thank.

6. **Spectral estimation**
   - Periodogram, Bartlett's method, Welch's method, Blackman-Tukey method
   - ARMA modeling, Yule-Walker equation and solution

Literature

- J. G. Proakis, D. G. Manolakis: *Digital Signal Processing: Principles, Algorithms, and Applications*, Prentice Hall, 2007, 4th edition
- S. K. Mitra: Digital Signal Processing: *A Computer-Based Approach*, McGraw Hill Higher Education, 2006, 3rd edition
- A. V. Oppenheim, R. W. Schafer: *Discrete-time signal processing*, Prentice Hall, 1999, 2nd edition
- M. H. Hayes: *Statistical Signal Processing and Modeling*, John Wiley and Sons, 1996 (chapter 6).

1. Introduction

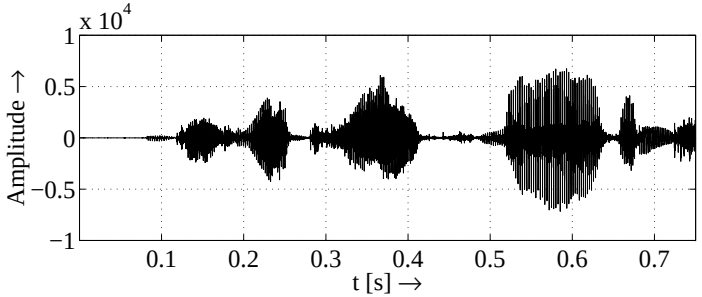
1.1 Signals, systems and signal processing

What does “Digital Signal Processing” mean?

Signal:

- Physical quantity that varies with time, space or any other independent variable
- Mathematically: Function of one or more independent variables, $s_1(t) = 5\,t$, $s_2(t) = 20\,t^2$
- Examples: Temperature over time t, brightness (luminance) of an image over (x, y), pressure of a sound wave over (x, y, z) or (x, y, z, t)

  Speech signal:



Signal Processing:

- Passing the signal through a system
- Examples:
  - Modification of the signal (filtering, interpolation, noise reduction, equalization, . . .)
  - Prediction, transformation to another domain (e.g. Fourier transform)

  - Numerical integration and differentiation
  - Determination of mean value, correlation, p.d.f., . . .
- Properties of the system (e.g. linear/nonlinear) determine the properties of the whole processing operation
- System: Definition also includes:
  - *software* realizations of operations on a signal, which are carried out on a digital computer ($\Rightarrow$ software implementation of the system)
  - digital *hardware* realizations (logic circuits) configured such that they are able to perform the processing operation, or
  - most general definition: a combination of both

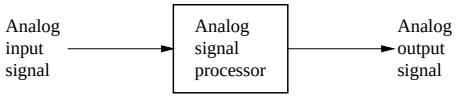
Digital signal processing: Processing of signals by digital means (software and/or hardware)

Includes:

- Conversion from the analog to the digital domain and back (physical signals are analog)
- Mathematical specification of the processing operations $\Rightarrow$ *Algorithm*: method or set of rules for implementing the system by a program that performs the corresponding mathematical operations
- Emphasis on computationally efficient algorithms, which are fast and easily implemented.

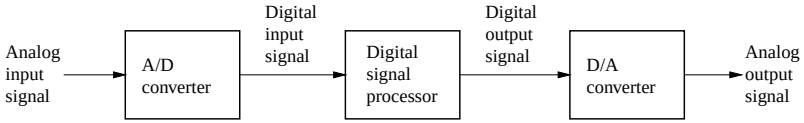
Basic elements of a digital signal processing system

Analog signal processing:



Digital signal processing:

(A/D: Analog-to-digital, D/A: Digital-to-analog)



Why has digital signal processing become so popular?

Digital signal processing has many advantages compared to analog processing:

Property	Digital	Analog
Dynamics	only limited by complexity	generally limited
Precision	generally unlimited (costs, complexity $\sim$ precision)	generally limited (costs increase drastically with required precision)
Aging	without problems	problematic
Production costs	low	higher
Frequency range	$\omega_{dmin} \ll \omega_{amin}, \omega_{dmax} \ll \omega_{amax}$	
Linear-phase frequency responses	exactly realizable	approximately realizable
Complex algorithms	realizable	strong limitations

1.2 Digital signal processors (DSPs)

- Programmable microprocessor (more flexibility), or hardwired digital processor (*ASIC*, *application specific integrated circuit*) (faster, cheaper)

Often programmable DSPs (simply called ”DSPs”) are used for evaluation purposes, for prototypes and for complex algorithms:

- Fixed-point processors: Twos-complement number representation.
- Floating-point processors: Floating point number representation (as for example used in PC processors)

Overview over some available DSP processors see next page.

Performance example: 256-point complex FFT

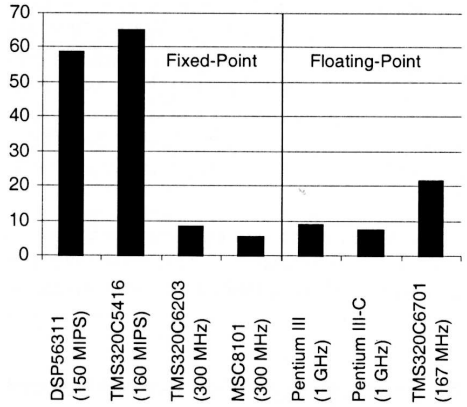
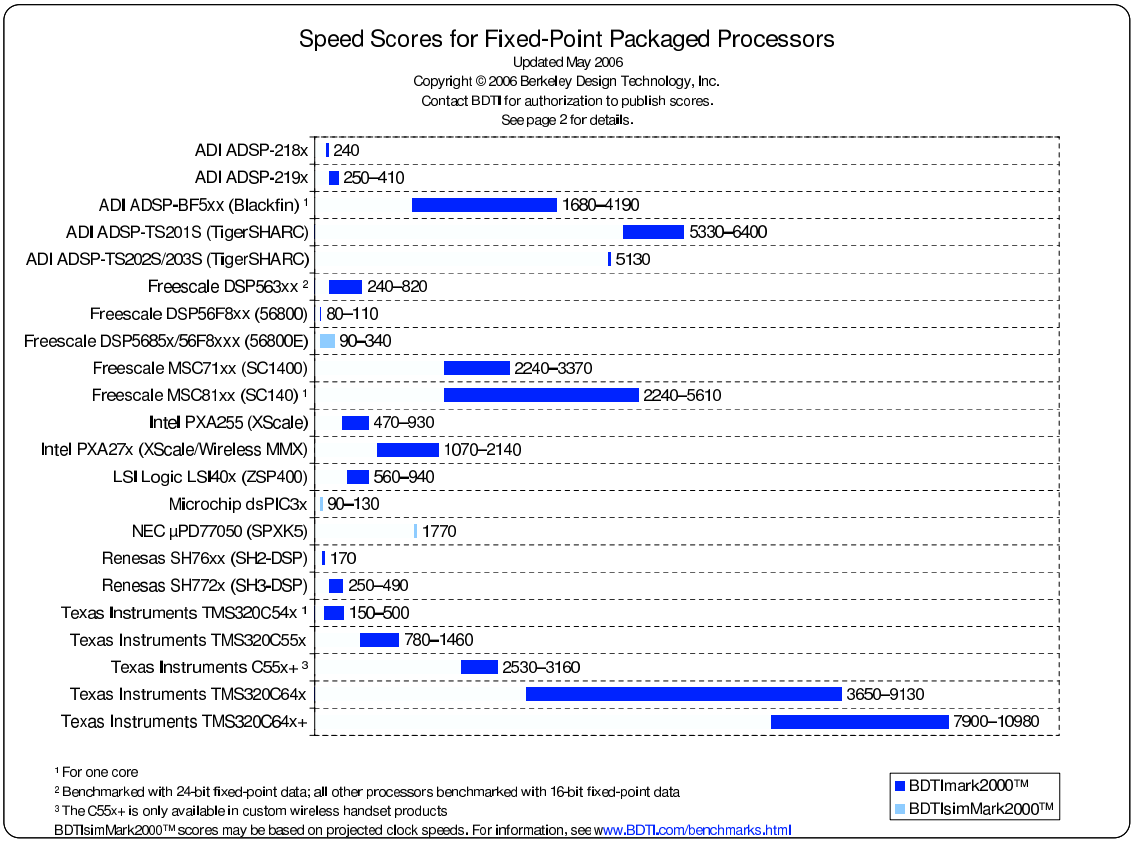


FIGURE 1. Execution times for a 256-point complex FFT, in microseconds (lower is better).

(from [Evaluating DSP Processor Performance, Berkeley Design Technology, Inc., 2000])

Some currently available DSP processors and their properties (2006):

Manufacturer	Family	Arithmetic	Data width (bits)	BDTImark 2000(TM)	Core clock speed	Unit price qty. 10000
Analog Devices	ADSP-219x	fixed-point	16	410	160 MHz	$11-26
	ADSP-2126x	floating-point	32/40	1090	200 MHz	$5-15
	ADSP-BF5xx	fixed-point	16	4190	750 MHz	$5-32
	ADSP-TS20x	floating/fixed-point	8/16/32/40	6400	600 MHz	$131-205
Freescale	DSP563xx	fixed-point	24	820	275 MHz	$4-47
	DSP568xx	fixed-point	16	110	80 MHz	$3-12
	MSC71xx	fixed-point	16	3370	300 MHz	$13-35
	MSC81xx	fixed-point	16	5610	500 MHz	$77-184
Texas-Instuments	TMS320C24x	fixed-point	16	n/a	40 MHz	$2-8
	TMS320C54x	fixed-point	16	500	160 MHz	$3-54
	TMS320C55x	fixed-point	16	1460	300 MHz	$4-17
	TMS320C64x	fixed-point	8/16	9130	1 GHz	$15-208
	TMS320C67x	floating-point	32	1470	300 MHz	$12-31



Speed Scores for Fixed-Point Packaged Processors

Updated May 2006

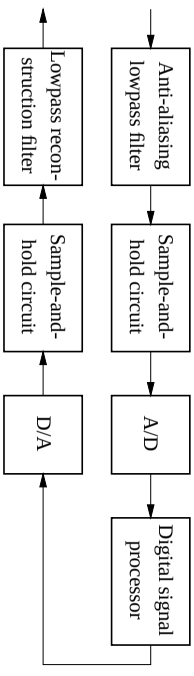
Processor Family	Clock Rate (min-max)	BDTImark2000™, *BDTIsimMark2000™* (min-max)
ADI ADSP-218x	80 MHz	240
ADI ADSP-219x	100–160 MHz	250–410
ADI ADSP-BF5xx (Blackfin) [1]	300–750 MHz	1680–4190
ADI ADSP-TS201S (TigerSHARC)	500–600 MHz	5330–6400
ADI ADSP-TS202S/203S (TigerSHARC)	500 MHz	5130
Freescale DSP563xx [2]	80–275 MHz	240–820
Freescale DSP56F8xx (56800)	60–80 MHz	80–110
Freescale DSP5685x/56F8xxx (56800E)	32–120 MHz	*90–340*
Freescale MSC71xx (SC1400)	200–300 MHz	2240–3370
Freescale MSC81xx (SC140) [1]	200–500 MHz	2240–5610
Intel PXA255 (XScale)	200–400 MHz	470–930
Intel PXA27x (XScale/Wireless MMX)	312–624 MHz	1070–2140
LSI Logic LSI40x (ZSP400)	120–200 MHz	560–940
Microchip dsPIC3x	30–40 MHz	*90–130*
NEC µPD77050 (SPXK5)	250 MHz	*1770*
Renesas SH76xx (SH2-DSP)	62.5 MHz	170
Renesas SH772x (SH3-DSP)	100–200 MHz	250–490
Texas Instruments TMS320C54x [1]	50–160 MHz	150–500
Texas Instruments TMS320C55x	160–300 MHz	780–1460
Texas Instruments C55x+ [3]	400–500 MHz	2530–3160
Texas Instruments TMS320C64x	400–1000 MHz	3650–9130
Texas Instruments TMS320C64x+	720–1000 MHz	7900–10980

[1] For one core
[2] Benchmarked with 24-bit fixed-point data; all other processors benchmarked with 16-bit fixed-point data
[3] The C55x+ is only available in custom wireless handset products

BDTImark2000™, BDTIsimMark2000™: The BDTImark2000™ and BDTIsimMark2000™ provide a summary measure of signal processing speed. BDTIsimMark2000™ scores may be based on projected clock speeds. For more info and scores see:
www.BDTI.com/benchmarks.html

2. Digital Processing of Continuous-Time Signals

Digital signal processing system from above is refined:



2.1 Sampling

⇒ Generation of discrete-time signals from continuous-time signals

Ideal sampling

Ideally sampled signal $x_s(t)$ obtained by multiplication of the continuous-time signal $x_c(t)$ with the periodic impulse train

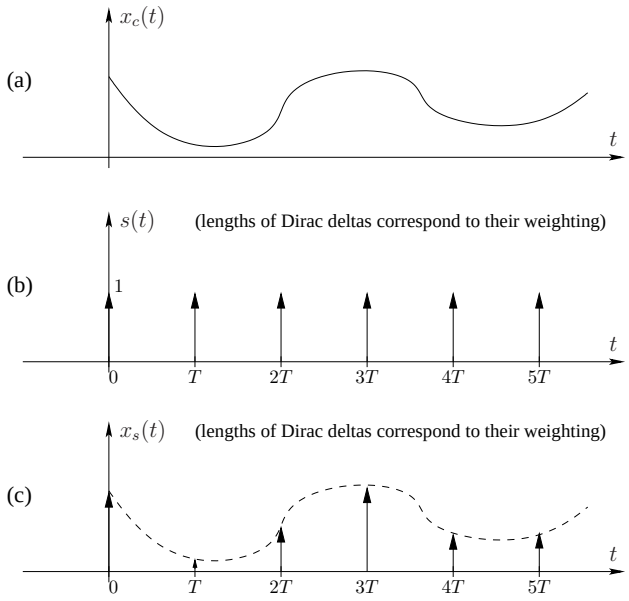
$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT),$$

where $\delta(t)$ is the *unit impulse function* and T the sampling period:

$$x_s(t) = x_c(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{2.1}$$

$$= \sum_{n=-\infty}^{\infty} x_c(nT)\, \delta(t - nT) \tag{2.2}$$

("sifting property" of the impulse function)



How does the **Fourier transform** $\mathcal{F}\{x_s(t)\} = X_s(j\Omega)$ look like?

Fourier transform of an impulse train:

$$s(t) \circ\!\!-\!\!\bullet\, S(j\Omega) = \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s) \qquad (2.3)$$

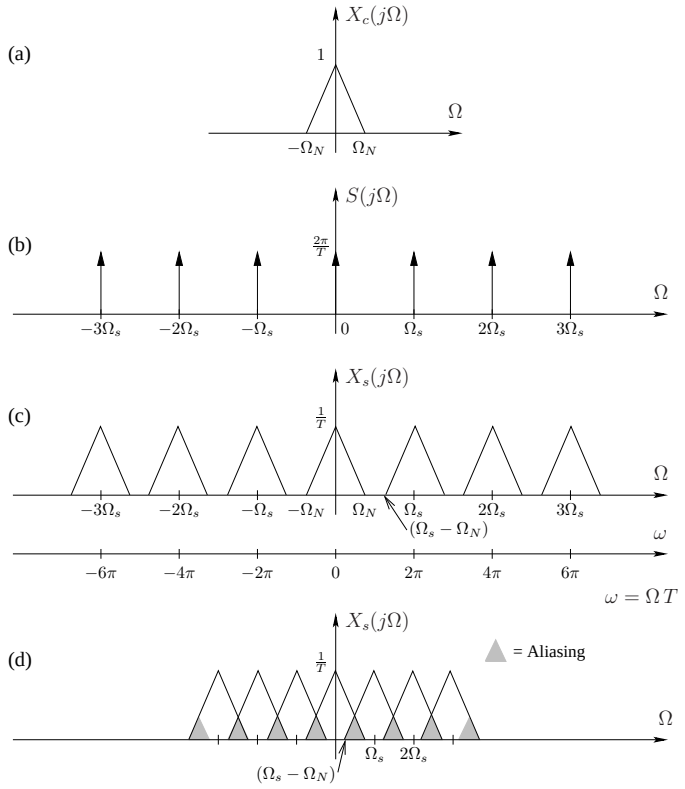
$\Omega_s = 2\pi/T$: sampling frequency in radians/s.
Writing (2.1) in the Fourier domain,

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega),$$

we finally have for the Fourier transform of $x_s(t)$

$$\boxed{X_s(j\Omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_c(j(\Omega - k\Omega_s)).} \qquad (2.4)$$

$\Rightarrow$ Periodically repeated copies of $X_s(j\Omega)/T$, shifted by integer multiples of the sampling frequency



(a) Fourier transform of a bandlimited continuous-time input signal $X_c(j\Omega)$, highest frequency is Ω_N

(b) Fourier transform of the Dirac impulse train

(c) Result of the convolution $S(j\Omega) * X_c(j\Omega)$. It is evident that when

$$\Omega_s - \Omega_N > \Omega_N \quad \text{or} \quad \Omega_s > 2\Omega_N \tag{2.5}$$

the replicas of $X_c(j\Omega)$ do *not* overlap.
$\Rightarrow$ *$x_c(t)$ can be recovered with an ideal lowpass filter!*

(d) If (2.5) does not hold, i.e. if $\Omega_s \leq 2\Omega_N$, the copies of $X_c(j\Omega)$ overlap and the signal $x_c(t)$ cannot be recovered by lowpass filtering. The distortions in the gray shaded areas are called *aliasing distortions* or simply *aliasing*.

Also in (c): Representation with the discrete (normalized) frequency $\omega = \Omega T = 2\pi f T$ (f frequency in Hz) for the discrete signal $x_c(nT) = x(n)$ with $\mathcal{F}_*\{x(n)\} = X(e^{j\omega})$, $\mathcal{F}_*\{\cdot\}$ denoting the Fourier transform for discrete-time aperiodic signals (DTFT)

Nonideal sampling

$\Rightarrow$ Modeling the sampling operation with the Dirac impulse train is not a feasible model in real life, since we always need a finite amount of time for acquiring a signal sample.

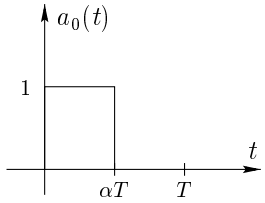
Nonideally sampled signal $x_n(t)$ obtained by multiplication of a continuous-time signal $x_c(t)$ with a periodic rectangular window function $a_n(t)$: $x_n(t) = x_c(t) \cdot a_n(t)$ where

$$a_n(t) = a_0(t) * \sum_{n=-\infty}^{\infty} \delta(t - n\,T) = \sum_{n=-\infty}^{\infty} a_0(t - nT) \tag{2.6}$$

$a_0(t)$ denotes the rectangular prototype window:

$$a_0(t) = \text{rect}\left(\frac{t - \alpha T/2}{\alpha T}\right) \tag{2.7}$$

$$\text{with} \quad \text{rect}(t) := \begin{cases} 0 & \text{for} \quad |t| > 1/2 \\ 1 & \text{for} \quad |t| < 1/2 \end{cases} \tag{2.8}$$



$\text{rect}(t) \circ\!\!-\!\!\bullet \text{sinc}(\Omega/2)$,
$\text{sinc}(x) := \sin(x)/x$

Fourier transform of $a_n(t)$:

Fourier transform of the rectangular time window in (2.7) (see properties of the Fourier transform)

$$A_0(j\Omega) = \mathcal{F}\{a_0(t)\} = \alpha\, T \cdot \text{sinc}(\Omega\alpha T/2) \cdot e^{-j\Omega\alpha T/2} \tag{2.9}$$

Fourier transform of $a_n(t)$ in (2.6) (analog to (2.3)):

$$\begin{aligned} A_n(j\Omega) &= A_0(j\Omega) \cdot \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s) \\ &= 2\pi\alpha \sum_{k=-\infty}^{\infty} \text{sinc}(k\Omega_s\alpha\, T/2)\, e^{-jk\Omega_s\alpha T/2}\, \delta(\Omega - k\Omega_s) \\ &= 2\pi\alpha \sum_{k=-\infty}^{\infty} \text{sinc}(k\pi\alpha)\, e^{-jk\pi\alpha}\, \delta(\Omega - k\Omega_s) \end{aligned} \tag{2.10}$$

Since

$$x_n(t) = x_c(t)\, a_n(t) \;\circ\!\!-\!\!\bullet\; X_n(j\Omega) = \frac{1}{2\pi}\,(X_c(j\Omega) * A_n(j\Omega))$$

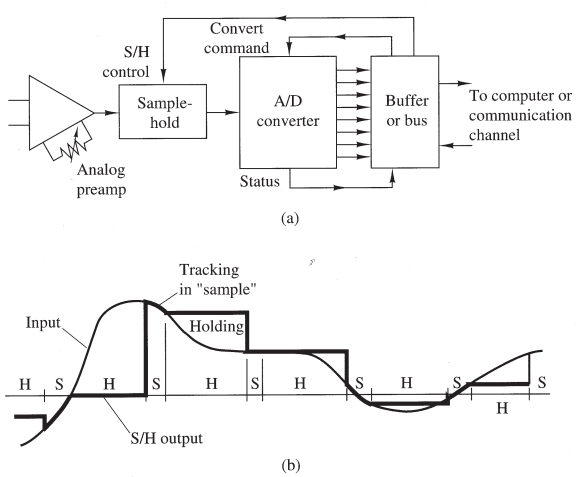
we finally have by inserting (2.10)

$$X_n(j\Omega) = \alpha \sum_{k=-\infty}^{\infty} \operatorname{sinc}(k\pi\alpha)\, e^{-jk\pi\alpha} X_c(j(\Omega - k\Omega_s)). \tag{2.11}$$

From (2.11) we can deduce the following:

- Compared to the result in the ideal sampling case (cp. (2.4)) here each repeated spectrum at the center frequency $k\Omega_s$ is weighted with the term $\operatorname{sinc}(k\pi\alpha)\, e^{-jk\pi\alpha}$.
- The energy $|X_n(j\Omega)|^2$ is proportional α^2: This is problematic since in order to approximate the ideal case we would like to choose the parameter α as small as possible.

Solution: Sampling is performed by a *sample-and-hold* (S/H) circuit



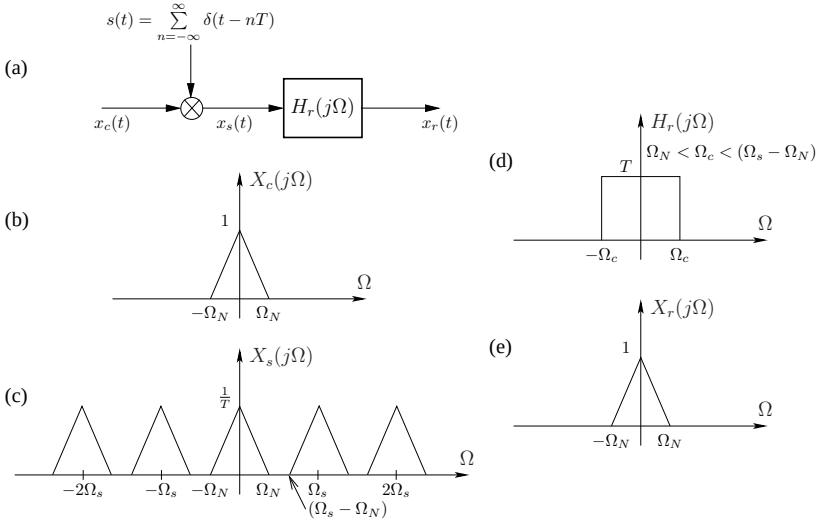
(from [Proakis, Manolakis, 1996])

(a) Basic elements of an A/D converter, (b) time-domain response of an ideal S/H circuit

- The goal is to continously sample the input signal and to hold that value constant as long as it takes for the A/D converter to obtain its digital representation.
- Ideal S/H circuit introduces no distortion and can be modeled as an ideal sampler.
  $\Rightarrow$ Drawbacks for the nonideal sampling case can be avoided (all results for the ideal case hold here as well).

2.2 Sampling Theorem

Reconstruction of an ideally sampled signal by ideal lowpass filtering:



In order to get the input signal $x_c(t)$ back after reconstruction, i.e. $X_r(j\Omega) = X_c(j\Omega)$, the conditions

$$\Omega_N < \frac{\Omega_s}{2} \quad \text{and} \quad \Omega_N < \Omega_c < (\Omega_s - \Omega_N) \tag{2.12}$$

have both to be satisfied. Then,

$$X_c(j\Omega) = X_r(j\Omega) = X_s(j\Omega) \cdot H_r(j\Omega) \quad \bullet\!\!-\!\!\circ \quad x_c(t) = x_r(t) = x_s(t) * h_r(t). \tag{2.13}$$

We now choose the cutoff frequency Ω_c of the lowpass filter as $\Omega_c = \Omega_s/2$ (satisfies both inequalities in (2.12)).

Then, with the definition of the rect$(\cdot)$ function in (2.8) we have

$$H_r(j\Omega) = T\,\mathrm{rect}(\Omega/\Omega_s) \bullet\!\!-\!\!\circ h_r(t) = \mathrm{sinc}(\Omega_s t/2). \tag{2.14}$$

Combining (2.13), (2.14), and (2.2) yields

$$\begin{aligned} x_c(t) &= \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x_c(nT)\,\delta(\tau - nT)\,\mathrm{sinc}\left(\frac{1}{2}\,\Omega_s(t-\tau)\right)\,d\tau \\ &= \sum_{n=-\infty}^{\infty} x_c(nT) \int_{-\infty}^{\infty} \delta(\tau - nT)\,\mathrm{sinc}\left(\frac{1}{2}\,\Omega_s(t-\tau)\right)\,d\tau \\ &= \sum_{n=-\infty}^{\infty} x_c(nT)\,\mathrm{sinc}\left(\frac{1}{2}\,\Omega_s(t-nT)\right). \end{aligned}$$

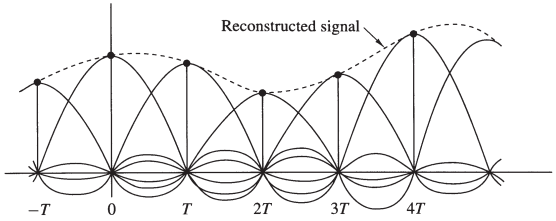
Sampling theorem:
Every bandlimited continuous-time signal $x_c(t)$ with $\Omega_N < \Omega_s/2$ can be uniquely recovered from its samples $x_c(nT)$ according to

$$\boxed{x_c(t) = \sum_{n=-\infty}^{\infty} x_c(nT)\,\mathrm{sinc}\left(\frac{1}{2}\,\Omega_s(t-nT)\right).} \tag{2.15}$$

Remarks:

- Eq. (2.15) is called the *ideal interpolation formula*, and the sinc-function is named *ideal interpolation function*

- Reconstruction of a continuous-time signal using ideal interpolation:



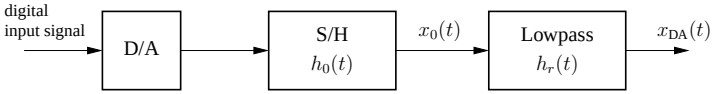
(from [Proakis, Manolakis, 1996])

Anti-aliasing lowpass filtering:

In order to avoid aliasing, the continuous-time input signal has to be bandlimited by means of an *anti-aliasing lowpass-filter* with cut-off frequency $\Omega_c \leq \Omega_s/2$ prior to sampling, such that the sampling theorem is satisfied.

2.3 Reconstruction with sample-and-hold circuit

In practice, a reconstruction is carried out by combining a D/A converter with a S/H circuit, followed by a lowpass reconstruction (smoothing) filter



- D/A converter accepts electrical signals that correspond to binary words as input, and delivers an output voltage or current being proportional to the value of the binary word for every clock interval nT
- Often, the application on an input code word yields a high-amplitude transient at the output of the D/A converter (”glitch”)

$\Rightarrow$ S/H circuit serves as a ”deglitcher”:
Output of the D/A converter is held constant at the previous output value until the new sample at the D/A output reaches steady state

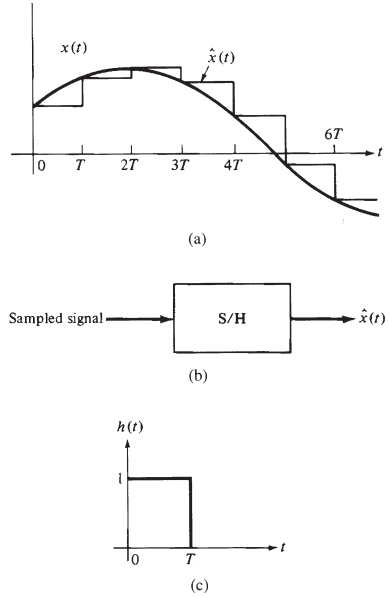


Figure 9.22 (a) Approximation of an analog signal by a staircase; (b) linear filtering interpretation; (c) impulse response of the S/H.

(from [Proakis, Manolakis, 1996])

Analysis:
The S/H circuit has the impulse response

$$h_0(t) = \text{rect}\left(\frac{t - T/2}{T}\right) \tag{2.16}$$

which leads to a frequency response

$$H_0(j\Omega) = T \cdot \text{sinc}(\Omega T/2) \cdot e^{-j\Omega T/2} \tag{2.17}$$

- No sharp cutoff frequency response characteristics $\Rightarrow$ we have undesirable frequency components (above $\Omega_s/2$), which can be removed by passing $x_0(t)$ through a lowpass reconstruction filter $h_r(t)$. This operation is equivalent to smoothing the staircase-like signal $x_0(t)$ after the S/H operation.
- When we now suppose that the reconstruction filter $h_r(t)$ is an ideal lowpass with cutoff frequency $\Omega_c = \Omega_s/2$ and an amplification of one, the only distortion in the reconstructed signal $x_{\text{DA}}(t)$ is due to the S/H operation:

$$|X_{\text{DA}}(j\Omega) = |X_c(j\Omega)| \cdot |\text{sinc}(\Omega T/2)| \qquad (2.18)$$

$|X_c(j\Omega)|$ denotes the magnitude spectrum for the ideal reconstruction case.

$\Rightarrow$ Additional distortion when the reconstruction filter is not ideal (as in real life!)
$\Rightarrow$ Distortion due to the sinc-function may be corrected by pre-biasing the frequency response of the reconstruction filter

Spectral interpretation of the reconstruction process (see next page):

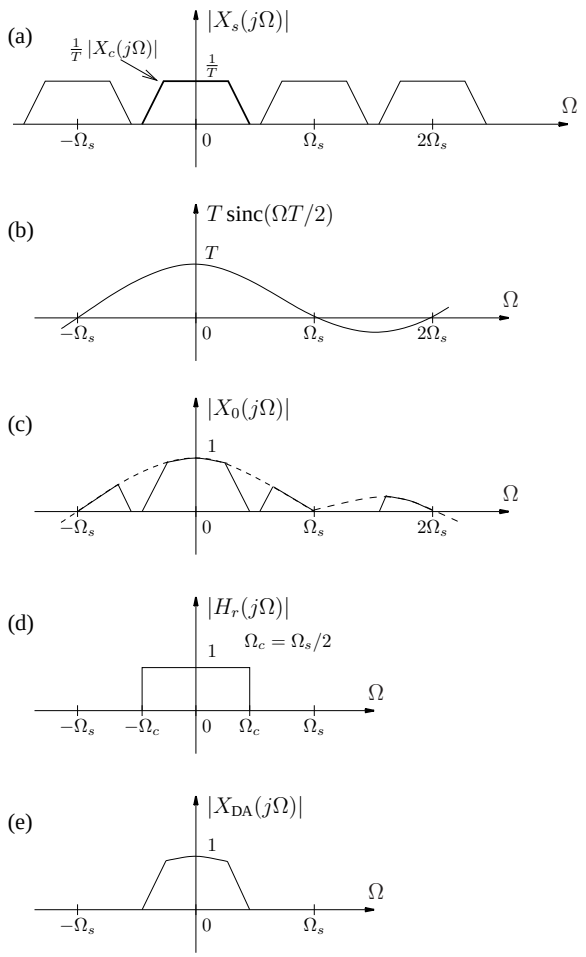
(a) Magnitude frequency response of the ideally sampled continuous-time signal

(b) Frequency response of the S/H circuit (phase factor $e^{-j\Omega T/2}$ omitted)

(c) Magnitude frequency response after the S/H circuit

(d) Magnitude frequency response: lowpass reconstruction filter

(e) Magnitude frequency response of the reconstructed continuous-time signal



2.4 Quantization

Conversion carried out by an A/D converter involves *quantization* of the sampled input signal $x_s(nT)$ and the *encoding* of the resulting binary representation

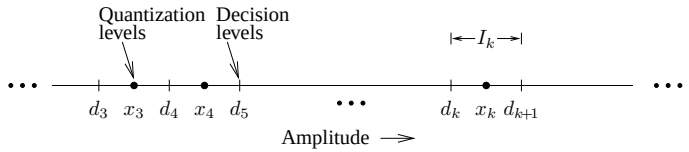
- Quantization is a *nonlinear* and *noninvertible* process which realizes the mapping

$$x_s(nT) = x(n) \longrightarrow x_k \in \mathcal{I}, \tag{2.19}$$

  where the amplitude x_k is taken from a finite alphabet $\mathcal{I}$.

- Signal amplitude range is divided into L intervals I_n using the $L+1$ decision levels $d_1, d_2, \ldots, d_{L+1}$:

$$I_n = \{d_k < x(n) \leq d_{k+1}\}, \quad k = 1, 2, \ldots, L$$

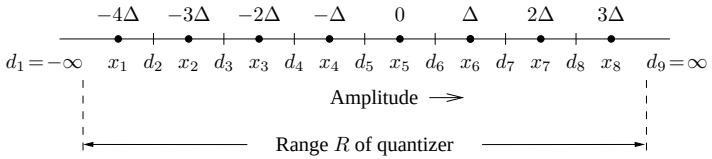


- Mapping in (2.19) is denoted as $\hat{x}(n) = Q[x(n)]$
- *Uniform* or *linear* quantizers with constant *quantization step size* Δ are very often used in signal processing applications:

$$\Delta = x_{k+1} - x_k = \text{const.}, \quad \text{for all } k = 1, 2, \ldots, L-1 \tag{2.20}$$

- *Midtreat* quantizer: Zero is assigned a quantization level
  Midrise quantizer: Zero is assigned a decision level

- Example: midtreat quantizer with $L = 8$ levels and *range* $R = 8\Delta$

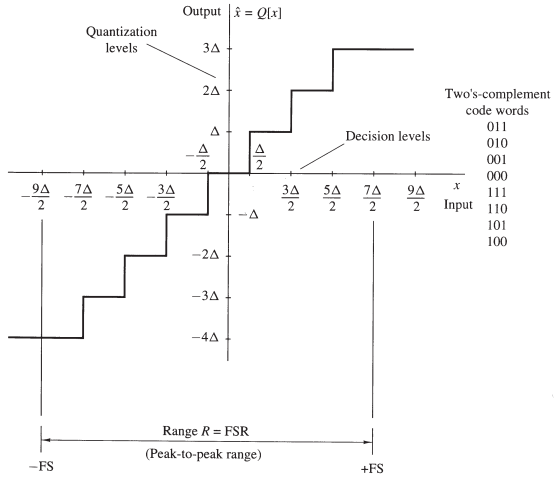


- *Quantization error* $e(n)$ with respect to the unquantized signal

$$-\frac{\Delta}{2} < e(n) \leq \frac{\Delta}{2} \tag{2.21}$$

  If the dynamic range of the input signal $(x_{\max} - x_{\min})$ is larger than the range of the quantizer, the samples exceeding the quantizer range are clipped, which leads to $e(n) > \Delta/2$.

- *Quantization characteristic function* for a midtreat quantizer with $L = 8$:



(from [Proakis, Manolakis, 1996])

Coding

The coding process in an A/D converter assigns a binary number to each quantization level.

- With a wordlength of b bits we can represent $2^b > L$ binary numbers, which yields

$$b \geq \log_2(L). \tag{2.22}$$

- The step size or the *resolution* of the A/D converter is given as

$$\Delta = \frac{R}{2^b} \quad \text{with the range } R \text{ of the quantizer.} \tag{2.23}$$

- Commonly used bipolar codes:

Number	Positive Reference	Negative Reference	Sign + Magnitude	Two's Complement	Offset Binary	One's Complement
+7	$+\frac{7}{8}$	$-\frac{7}{8}$	0 1 1 1	0 1 1 1	1 1 1 1	0 1 1 1
+6	$+\frac{6}{8}$	$-\frac{6}{8}$	0 1 1 0	0 1 1 0	1 1 1 0	0 1 1 0
+5	$+\frac{5}{8}$	$-\frac{5}{8}$	0 1 0 1	0 1 0 1	1 1 0 1	0 1 0 1
+4	$+\frac{4}{8}$	$-\frac{4}{8}$	0 1 0 0	0 1 0 0	1 1 0 0	0 1 0 0
+3	$+\frac{3}{8}$	$-\frac{3}{8}$	0 0 1 1	0 0 1 1	1 0 1 1	0 0 1 1
+2	$+\frac{2}{8}$	$-\frac{2}{8}$	0 0 1 0	0 0 1 0	1 0 1 0	0 0 1 0
+1	$+\frac{1}{8}$	$-\frac{1}{8}$	0 0 0 1	0 0 0 1	1 0 0 1	0 0 0 1
0	0+	0−	0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0
0	0−	0+	1 0 0 0	(0 0 0 0)	(1 0 0 0)	1 1 1 1
−1	$-\frac{1}{8}$	$+\frac{1}{8}$	1 0 0 1	1 1 1 1	0 1 1 1	1 1 1 0
−2	$-\frac{2}{8}$	$+\frac{2}{8}$	1 0 1 0	1 1 1 0	0 1 1 0	1 1 0 1
−3	$-\frac{3}{8}$	$+\frac{3}{8}$	1 0 1 1	1 1 0 1	0 1 0 1	1 1 0 0
−4	$-\frac{4}{8}$	$+\frac{4}{8}$	1 1 0 0	1 1 0 0	0 1 0 0	1 0 1 1
−5	$-\frac{5}{8}$	$+\frac{5}{8}$	1 1 0 1	1 0 1 1	0 0 1 1	1 0 1 0
−6	$-\frac{6}{8}$	$+\frac{6}{8}$	1 1 1 0	1 0 1 0	0 0 1 0	1 0 0 1
−7	$-\frac{7}{8}$	$+\frac{7}{8}$	1 1 1 1	1 0 0 1	0 0 0 1	1 0 0 0
−8	$-\frac{8}{8}$	$+\frac{8}{8}$		(1 0 0 0)	(0 0 0 0)	

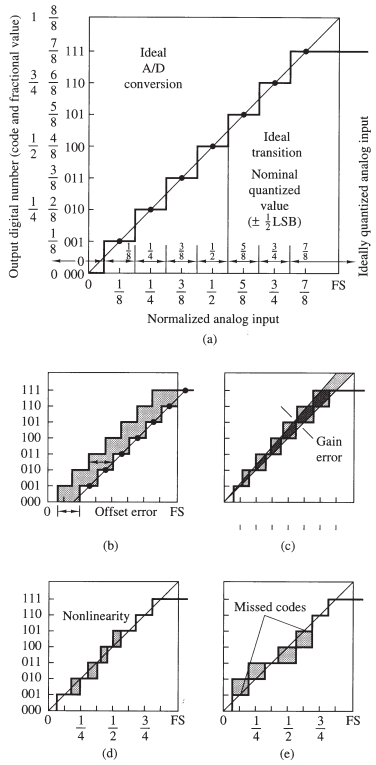
(from [Proakis, Manolakis, 1996])

- Two's complement representation is used in most fixed-point DSPs: A b-bit binary fraction $[\beta_0\beta_1\beta_2 \ldots \beta_{b-1}]$, β_0 denoting the *most significant bit* (MSB) and β_{b-1} the *least significant bit* (LSB), has the value

$$x = -\beta_0 + \sum_{\ell=1}^{b-1} \beta_\ell \, 2^{-\ell} \tag{2.24}$$

- Number representation has no influence on the performance of the quantization process!

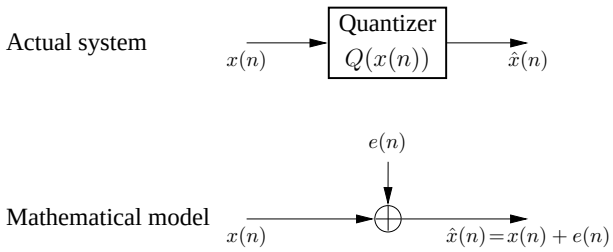
Performance degradations in practical A/D converters:



(from [Proakis, Manolakis, 1996])

Quantization errors

Quantization error is modeled as noise, which is added to the unquantized signal:



Assumptions:

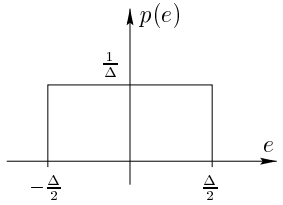
- The quantization error $e(n)$ is uniformly distributed over the range $-\frac{\Delta}{2} < e(n) < \frac{\Delta}{2}$.
- The error sequence $e(n)$ is modeled as a stationary white noise sequence.
- The error sequence $e(n)$ is uncorrelated with the signal sequence $x(n)$.
- The signal sequence is assumed to have zero mean.

Assumptions do not hold in general, but they are fairly well satisfied for large quantizer wordlengths b.

Effect of the quantization error or *quantization noise* on the resulting signal $\hat{x}(n)$ can be evaluated in terms of the *signal-to-noise ratio* (SNR) in Decibels (dB)

$$\boxed{\text{SNR} := 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_e^2} \right) ,} \tag{2.25}$$

where σ_x^2 denotes the signal power and σ_e^2 the power of the quantization noise.
Quantization noise is assumed to be uniformly distributed in the range $(-\Delta/2, \Delta/2)$:



$\Rightarrow$ Zero mean, and a variance of

$$\boxed{\sigma_e^2 = \int_{-\Delta/2}^{\Delta/2} e^2 \, p(e) \, de = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} e^2 \, de = \frac{\Delta^2}{12}} \tag{2.26}$$

Inserting (2.23) into (2.26) yields

$$\sigma_e^2 = \frac{2^{-2b} \, R^2}{12}, \tag{2.27}$$

and by using this in (2.25) we obtain

$$\begin{aligned} \text{SNR} &= 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_e^2} \right) = 10 \log_{10} \left(\frac{12 \, 2^{2b} \, \sigma_x^2}{R^2} \right) \\ &= 6.02 \, b + 10.8 \underbrace{-20 \log_{10} \left(\frac{R}{\sigma_x} \right)}_{(*)} \text{ dB}. \end{aligned} \tag{2.28}$$

Term $(*)$ in (2.28):

- σ_x root-mean-square (RMS) amplitude of the signal $v(t)$
- σ_x to small $\Rightarrow$ decreasing SNR
- Furthermore (not directly from $(*)$): σ_x to large $\Rightarrow$ range R is exceeded

$\Rightarrow$ Signal amplitude has to be carefully matched to the range of the A/D converter
For music and speech a good choice is $\sigma_x = R/4$. Then the SNR of a b-bit quantizer can be approximately determined as

$$\boxed{\text{SNR} = 6.02\,b - 1.25\,\text{dB}.} \qquad (2.29)$$

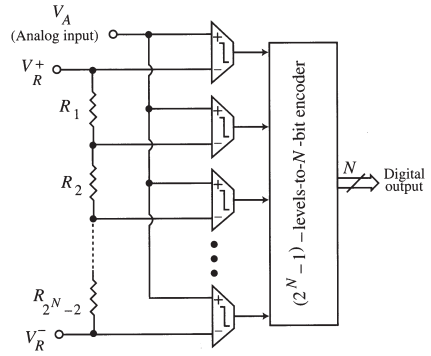
Each additional bit in the quantizer increases the signal-to-noise ratio by 6 *dB!*

Examples:

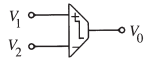
Narrowband speech: $b = 8$ Bit $\Rightarrow$ SNR $= 46.9$ dB
Music (CD): $b = 16$ Bit $\Rightarrow$ SNR $= 95.1$ dB
Music (Studio): $b = 24$ Bit $\Rightarrow$ SNR $= 143.2$ dB

2.5 Analog-to-digital converter realizations

Flash A/D converter



Analog comparator:



(from [Mitra, 2000], $N = b$: resolution in bits)

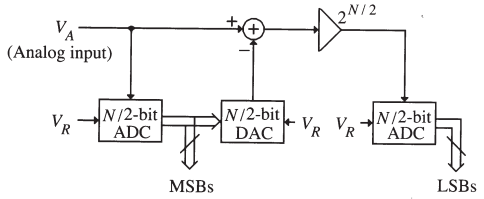
- Analog input voltage V_A is simultaneously compared with a set of $2^b - 1$ separated reference voltage levels by means of a set of 2^b–1 analog comparators $\Rightarrow$ locations of the comparator circuits indicate range of the input voltage.
- All output bits are developed simultaneously $\Rightarrow$ very fast conversion
- Hardware requirements increase exponentially with an increase in resolution

$\Rightarrow$ Flash converters are used for low-resultion ($b < 8$ bit) and high-speed conversion applications.

Serial-to-parallel A/D converters

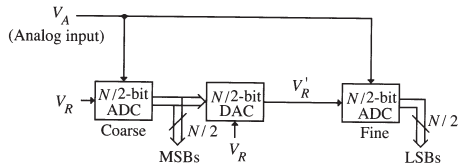
Here, two $b/2$-bit flash converters in a serial-parallel configuration are used to reduce the hardware complextity at the expense of a slightly higher conversion time

Subranging A/D converter:



(from [Mitra, 2000], $N = b$: resolution in bits)

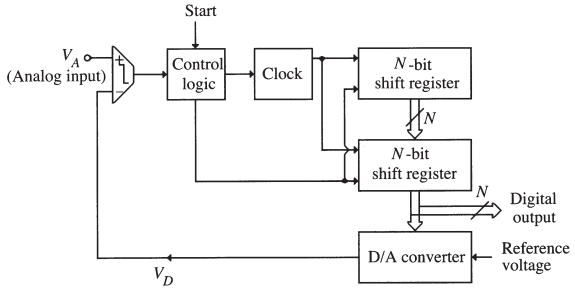
Ripple A/D converter:



(from [Mitra, 2000], $N = b$: resolution in bits)

Advantage of both structures: Always one converter is idle while the other one is operating
$\Rightarrow$ Only one $b/2$-bit converter is necessary

Sucessive approximation A/D converter



(from [Mitra, 2000], $N = b$: resolution in bits)

Iterative approach: At the k-th step of the iteration the binary

approximation in the shift register is converted into an (analog) reference voltage V_D by D/A conversion (binary representation $[\beta_0\beta_1 \ldots \beta_k\beta_{k+1} \ldots \beta_{b-1}]$, $\beta_k \in \{0, 1\} \forall k$):

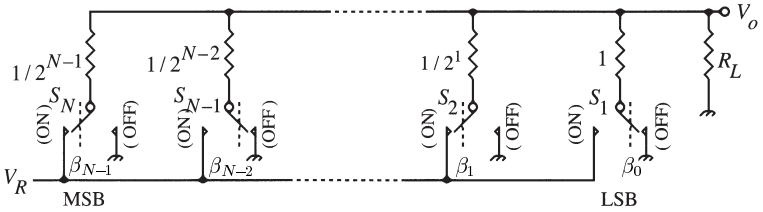
- Case 1: Reference voltage $V_D < V_A \Rightarrow$ increase the binary number by setting both the k-th bit and the $(k+1)$-th bit to 1
- Case 2: Reference voltage $V_D \geq V_A \Rightarrow$ decrease the binary number by setting the k-th bit to 0 and the $(k+1)$-th bit to 1

$\Rightarrow$ High resolution and fairly high speed at moderate costs, widely used in digital signal processing applications

Oversampling sigma-delta A/D converter, to be discussed in Section 5...

2.6 Digital-to-analog converter realizations

Weighted-resistor D/A converter



(from [Mitra, 2000], $N = b$: resolution in bits)

Output V_o of the D/A converter is given by

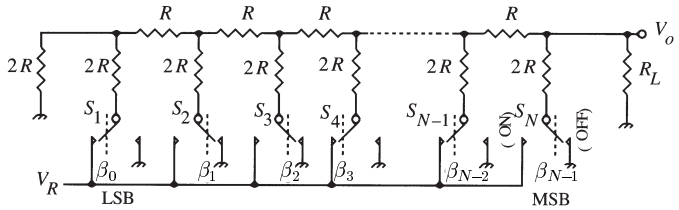
$$V_o = \sum_{\ell=0}^{N-1} 2^\ell \beta_\ell \frac{R_L}{(2^N - 1)R_L + 1} V_R$$

V_R: reference voltage
Full-scale output voltage $V_{o,FS}$ is obtained when $\beta_\ell = 1$ for all $\ell = 0, \ldots, N-1$:

$$V_{o,FS} = \frac{(2^N - 1)R_L}{(2^N - 1)R_L + 1} V_R \approx V_R, \text{ since } (2^N\text{–}1)R_L \gg 1$$

Disadvantage: For high resolutions the spread of the resistor values becomes very large

Resistor-ladder D/A converter



(from [Mitra, 2000], $N = b$: resolution in bits)

$\Rightarrow$ R–$2R$ ladder D/A converter, most widely used in practice.
Output V_o of the D/A converter:

$$V_o = \sum_{\ell=0}^{N-1} 2^\ell \beta_\ell \frac{R_L}{2(R_L + R)} \cdot \frac{V_R}{2^{N-1}}$$

In practice, often $2R_L \gg R$, and thus, the full-scale output voltage $V_{o,FS}$ is given as

$$V_{o,FS} \approx \frac{(2^N - 1)}{2^N} V_R$$

Oversampling sigma-delta D/A converter, to be discussed in Section 5…

3. DFT and FFT

3.1 DFT and signal processing

Definition DFT from Signals and Systems:

$$\text{DFT:} \quad v(n) \circ\!\!-\!\!\bullet\, V_N(k) = \sum_{n=0}^{N-1} v(n)\, W_N^{kn} \tag{3.1}$$

$$\text{IDFT:} \quad V_N(k) \bullet\!\!-\!\!\circ\, v(n) = \frac{1}{N} \sum_{k=0}^{N-1} V_N(k)\, W_N^{-kn} \tag{3.2}$$

with $W_N := e^{-j2\pi/N}$, N: number of DFT points

3.1.1 Linear and circular convolution

Linear convolution of two sequences $v_1(n)$ and $v_2(n)$, $n \in \mathbb{Z}$:

$$y_l(n) = v_1(n) * v_2(n) = v_2(n) * v_1(n)$$
$$= \sum_{k=-\infty}^{\infty} v_1(k)\, v_2(n-k) = \sum_{k=-\infty}^{\infty} v_2(k)\, v_1(n-k) \tag{3.3}$$

Circular convolution of two *periodic* sequences $v_1(n)$ and $v_2(n)$, $n = \{0, \ldots, N_{1,2} - 1\}$ with the *same* period $N_1 = N_2 = N$ and $n_0 \in \mathbb{Z}$:

$$y_c(n) = v_1(n) \circledast v_2(n) = v_2(n) \circledast v_1(n)$$
$$= \sum_{k=n_0}^{n_0+N-1} v_1(k)\, v_2(n-k) = \sum_{k=n_0}^{n_0+N-1} v_2(k)\, v_1(n-k) \tag{3.4}$$

We also use the symbol $\textcircled{N}$ instead of $\circledast$.

DFT and circular convolution

Inverse transform of a finite-length sequence $v(n)$, $n, k = 0, \ldots, N-1$:

$$v(n) \circ\!\!-\!\!\bullet\, V_N(k) \bullet\!\!-\!\!\circ\, v(n) = v(n + \lambda N) \tag{3.5}$$

$\Rightarrow$ DFT of a finite-length sequence and its periodic extension is identical!
Circular convolution property ($n, k = 0, \ldots, N-1$)
($v_1(n)$ and $v_2(n)$ denote *finite-length* sequences):

$$\boxed{y(n) = v_1(n) \,\textcircled{N}\, v_2(n) \circ\!\!-\!\!\bullet\, Y(k) = V_{1_N}(k) \cdot V_{2_N}(k)} \tag{3.6}$$

Proof:

$$\text{IDFT of } Y(k): \quad y(n) = \frac{1}{N} \sum_{k=0}^{N-1} Y(k) W_N^{-kn}$$
$$= \frac{1}{N} \sum_{k=0}^{N-1} V_{1_N}(k) V_{2_N}(k) W_N^{-kn}$$

Substitution of the DFT definition in (3.1) for $v_1(n)$ and $v_2(n)$:

$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} \left[\sum_{m=0}^{N-1} v_1(m) W_N^{km} \right] \left[\sum_{l=0}^{N-1} v_2(l) W_N^{kl} \right] W_N^{-kn}$$
$$= \frac{1}{N} \sum_{m=0}^{N-1} v_1(m) \sum_{l=0}^{N-1} v_2(l) \left[\sum_{k=0}^{N-1} W_N^{-k(n-m-l)} \right] \tag{3.7}$$

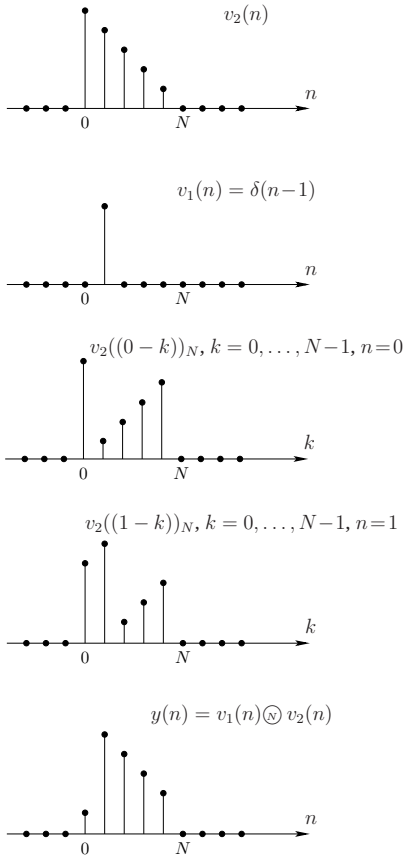
Term in brackets: Summation over the unit circle

$$\sum_{k=0}^{N-1} e^{j2\pi k(n-m-l)/N} = \begin{cases} N & \text{for} \quad l = n-m+\lambda N,\ \lambda \in \mathbb{Z} \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

Substituting (3.8) into (3.7) yields the desired relation

$$y(n) = \sum_{k=0}^{N-1} v_1(k) \underbrace{\sum_{\lambda=-\infty}^{\infty} v_2(n-k+\lambda N)}_{=:\, v_2((n-k))_N \text{ (periodic extension)}}$$
$$= \sum_{k=0}^{N-1} v_1(k) v_2((n-k))_N \tag{3.9}$$
$$= v_1(n) \,\textcircled{N}\, v_2(n)$$

Example: Circular convolution $y(n) = v_1(n) \textcircled{N} v_2(n)$:



3.1.2 Use of the DFT in linear filtering

- Filtering operation can also be carried out in the frequency domain using the DFT $\Rightarrow$ attractive since fast algorithms (fast Fourier transforms) exist
- DFT only realizes circular convolution, however, the desired operation for linear filtering is linear convolution. How can this be achieved by means of the DFT?

Given: Finite-length sequences $v_1(n)$ with length N_1 and $v_2(n)$ with length N_2

- Linear convolution:

$$y(n) = \sum_{k=0}^{N_1-1} v_1(k)\, v_2(n-k)$$

  Length of the convolution result $y(n)$: $N_1 + N_2 - 1$
- Frequency domain equivalent: $Y(e^{j\omega}) = V_1(e^{j\omega})\, V_2(e^{j\omega})$
- In order to represent the sequence $y(n)$ uniquely in the frequency domain by samples of its spectrum $Y(e^{j\omega})$, the number of samples must be equal or exceed $N_1 + N_2 - 1$ $\Rightarrow$ DFT of size $N \geq N_1 + N_2 - 1$ is required.
- Then, the DFT of the linear convolution $y(n) = v_1(n) * v_2(n)$ is $Y(k) = V_1(k) \cdot V_2(k)$, $k = 0, \ldots, N-1$.

This result can be summarized as follows:
The circular convolution of two sequences $v_1(n)$ with length N_1 and $v_2(n)$ with length N_2 leads to the same result as the linear convolution $v_1(n) * v_2(n)$ when the lengths of $v_1(n)$ and $v_2(n)$ are increased to $N = N_1 + N_2 - 1$ points by *zero padding*.

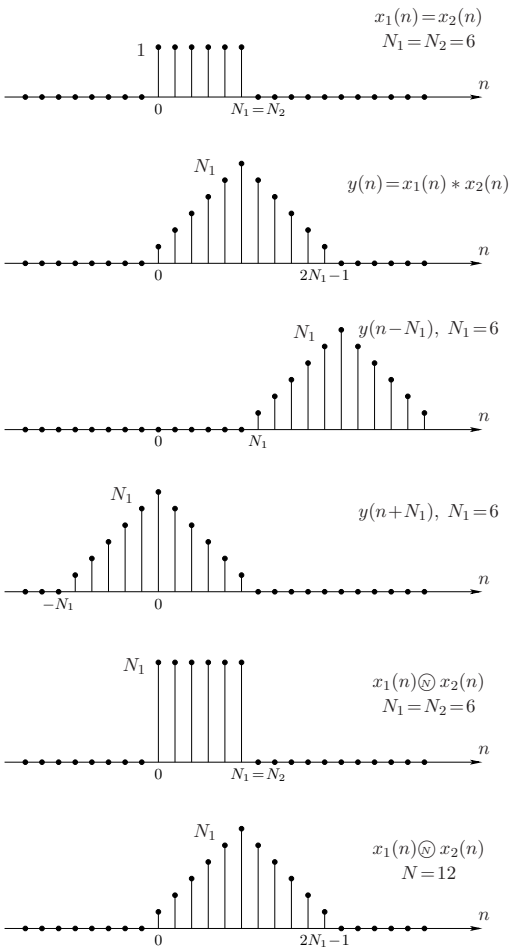
Other interpretation: Circular convolution as linear convolution with aliasing
IDFT leads to periodic sequence in the time-domain:

$$y_p(n) = \begin{cases} \sum\limits_{\lambda=-\infty}^{\infty} y(n - \lambda N) & n = 0, \ldots, N-1, \\ 0 & \text{otherwise} \end{cases}$$

with $Y(k) = \text{DFT}_N\{y(n)\} = \text{DFT}_N\{y_p(n)\}$

$\Rightarrow$ For $N < N_1 + N_2 - 1$: Circular convolution equivalent to linear convolution followed by *time domain aliasing*

Example:



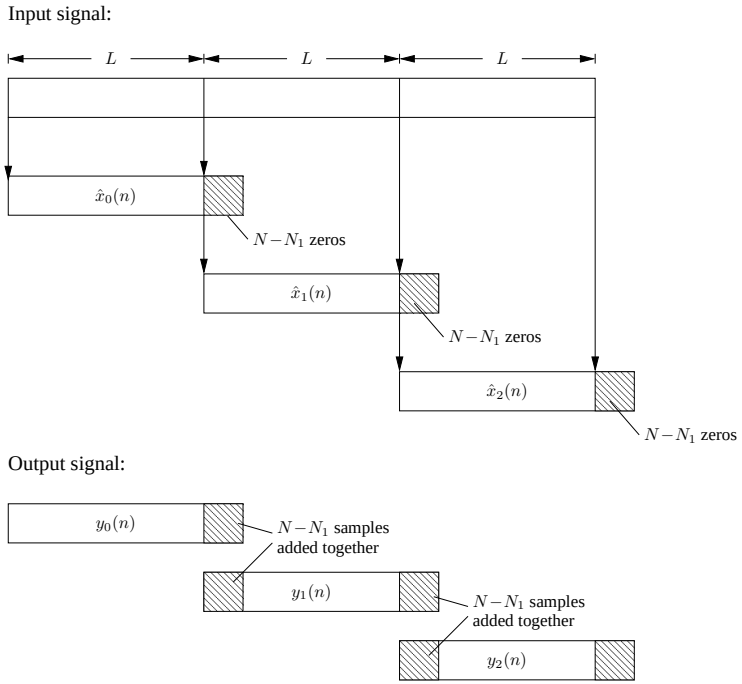
3.1.3 Filtering of long data sequences

Filtering of a long input signal $v(n)$ with the finite impulse response $h(n)$ of length N_2

Overlap-add method

1. Input signal is segmented into separate blocks:
   $v_\nu(n) = v(n + \nu N_1),\ n \in \{0, 1, \ldots, N_1{-}1\},\ \nu \in \mathbb{Z}$.
2. Zero-padding for the signal blocks $v_\nu(n) \rightarrow \tilde{v}_\nu(n)$ and the impulse response $h(n) \rightarrow \tilde{h}(n)$ to the length $N = N_1 + N_2 - 1$.
   Input signal can be reconstructed according to
   $$v(n) = \sum_{\nu=-\infty}^{\infty} \tilde{v}_\nu(n - \nu N_1)$$
   since $\tilde{v}_\nu(n) = 0$ for $n = N_1{+}1, \ldots, N$.
3. The two N-point DFTs are multiplied together to form
   $$Y_\nu(k) = \tilde{V}_\nu(k) \cdot \tilde{H}(k), \quad k = 0, \ldots, N{-}1.$$
4. The N-point IDFT yields data blocks that are free from aliasing due to the zero-padding in step 2.
5. Since each input data block $v_\nu(n)$ is terminated with $N{-}N_1$ zeros the last $N - N_1$ points from each output block $y_\nu(n)$ must be overlapped and added to the first $N{-}N_1$ points of the succeeding block (linearity property of convolution):
   $$y(n) = \sum_{\nu=-\infty}^{\infty} y_\nu(n - \nu N_1)$$
   $\Rightarrow$ *Overlap-add* method

Linear FIR (finite impulse response) filtering by the overlap-add method:

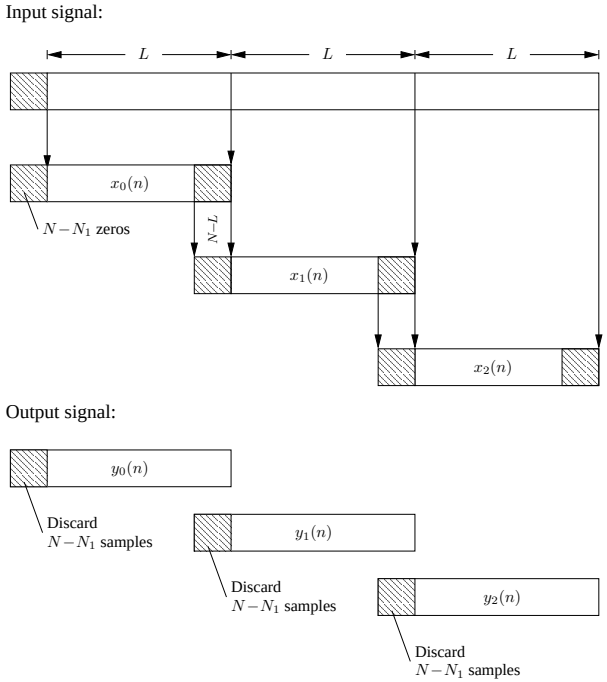


Overlap-save method

1. Input signal is segmented into by $N-N_1$ samples overlapping blocks:
   $v_\nu(n) = v(n+\nu N_1),\ n \in \{0,1,\ldots,N-1\},\ \nu \in \mathbb{Z}$.
2. Zero-padding of the filter impulse response $h(n) \rightarrow \tilde{h}(n)$ to the length $N = N_1 + N_2 - 1$.
3. The two N-point DFTs are multiplied together to form $Y_\nu(k) = V_\nu(k) \cdot \tilde{H}(k), k = 0,\ldots,N-1$.

4. Since the input signal block is of length N the first $N-N_1$ points are corrupted by aliasing and must be discarded. The last $N_2 = N-N_1-1$ samples in $y_\nu(n)$ are exactly the same as the result from linear convolution.
5. In order to avoid the loss of samples due to aliasing the last $N-N_1$ samples are saved and appended at the beginning of the next block. The processing is started by setting the first $N-N_1$ samples of the first block to zero.

Linear FIR filtering by the overlap-save method:



Nore computationally efficient than linear convolution? Yes, in combination with very efficient algorithms for DFT computation.

3.1.4 Frequency analysis of stationary signals

Leakage effect

Spectral analysis of an analog signal $v(t)$:

- Antialiasing lowpass filtering and sampling with $\Omega_s \geq 2\Omega_b$, Ω_b denoting the cut-off frequency of the signal
- For practical purposes (delay, complexity): Limitation of the signal duration to the time interval $T_0 = L\,T$ (L: number of samples under consideration, T: sampling interval)

Limitation to a signal duration of T_0 can be modeled as multiplication of the sampled input signal $v(n)$ with a rectangular *window* $w(n)$

$$\hat{v}(n) = v(n)\,w(n) \quad \text{with} \quad w(n) = \begin{cases} 1 & \text{for } 0 \leq n \leq L-1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.10}$$

Suppose that the input sequence just consists of a single sinusoid, that is $v(n) = \cos(\omega_0 n)$. The Fourier transform is

$$V(e^{j\omega}) = \pi(\delta(\omega - \omega_0) + \delta(\omega + \omega_0)). \tag{3.11}$$

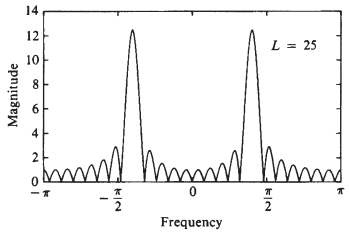
For the window $w(n)$ the Fourier transform can be obtained as

$$W(e^{j\omega}) = \sum_{n=0}^{L-1} e^{-j\omega n} = \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}} = e^{-j\omega(L-1)/2}\frac{\sin(\omega L/2)}{\sin(\omega/2)}. \tag{3.12}$$

We finally have

$$\begin{aligned} \hat{V}(e^{j\omega}) &= \frac{1}{2\pi}\left[V(e^{j\omega}) \circledast W(e^{j\omega})\right] \\ &= \frac{1}{2}\left[W(e^{j(\omega-\omega_0)}) + W(e^{j(\omega+\omega_0)})\right] \end{aligned} \tag{3.13}$$

Nagnitude frequency response $|\hat{V}(e^{j\omega})|$ for $L = 25$:



(from [Proakis, Nanolakis, 1996])

Windowed spectrum $\hat{V}(e^{j\omega})$ is not localized to one frequency, instead it is spread out over the whole frequency range
$\Rightarrow$ *spectral leaking*

First zero crossing of $W(e^{j\omega})$ at $\omega_z = \pm 2\pi/L$:

- The larger the number of sampling points L (and thus also the width of the rectangular window) the smaller becomes ω_z (and thus also the main lobe of the frequency response).
- $\Rightarrow$ Decreasing the frequency resolution leads to an increase of the time resolution and vice versa (duality of time and frequency domain).

In practice we use the DFT in order to obtain a sampled representation of the spectrum according to $\hat{V}(e^{j\omega_k})$, $k = 0, \ldots, N-1$.

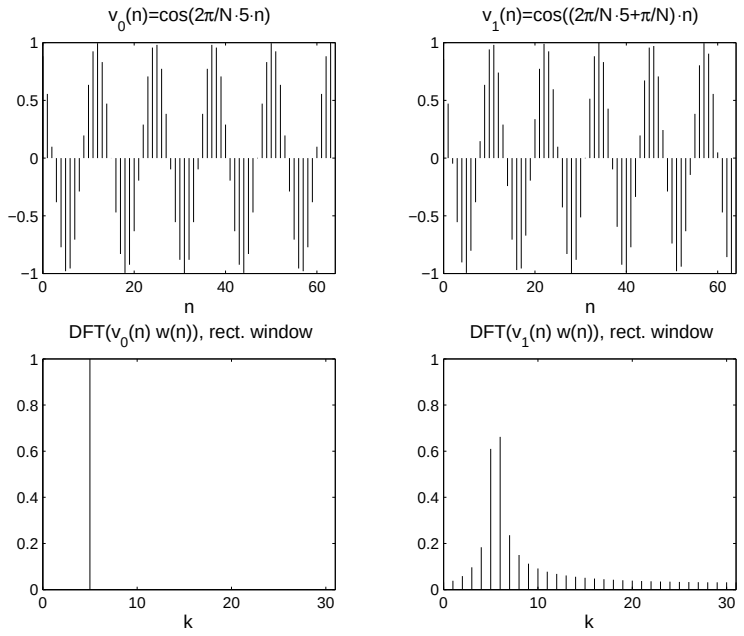
Special case: If

$$N = L \quad \text{and} \quad \omega_0 = \frac{2\pi}{N}\,\nu, \quad \nu = 0, \ldots, N-1$$

then the Fourier transform is exactly zero at the sampled frequencies ω_k, except for $k = \nu$.

Example: ($N = 64$, $n = 0, \ldots, N-1$, rectangular window $w(n)$)

$$v_0(n) = \cos\left[5\,\frac{2\pi}{N}\,n\right],\; v_1(n) = \cos\left[\left(5\,\frac{2\pi}{N} + \frac{\pi}{N}\right)\,n\right]$$



- Left hand side: $\hat{V}_0(e^{j\omega_k}) = V_0(e^{j\omega_k}) \circledast W(e^{j\omega_k}) = 0$ for $\omega_k = k2\pi/N$ except for $k = 5$, since ω_0 is exactly an integer multiple of $2\pi/N$
  $\Rightarrow$ periodic repetition of $v_0(n)$ leads to a pure cosine sequence
- Right hand side: slight increase of $\omega_0 \neq \nu 2\pi/N$ for $\nu \in \mathbb{Z}$ $\Rightarrow \hat{V}_1(e^{j\omega_k}) \neq 0$ for $\omega_k = k2\pi/N$, periodic repetition is not a cosine sequence anymore

Windowing and different window functions

Windowing not only distorts the spectral estimate due to leakage effects, it also reduces the spectral resolution.
Consider a sequence of two frequency components
$v(n) = \cos(\omega_1 n) + \cos(\omega_2 n)$ with the Fourier transform

$$V(e^{j\omega}) = \frac{1}{2}\Big[W(e^{j(\omega-\omega_1)}) + W(e^{j(\omega-\omega_2)}) + \\ + W(e^{j(\omega+\omega_1)}) + W(e^{j(\omega+\omega_2)})\Big]$$

where $W(e^{j\omega})$ is the Fourier transform of the rectangular window from (3.12).

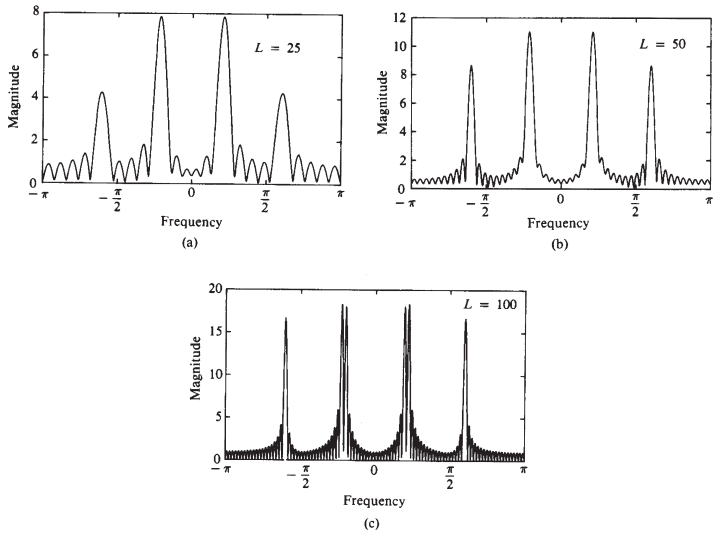
- $2\pi/L < |\omega_1 - \omega_2|$: Two maxima, main lobes for both window spectra $W(e^{j(\omega-\omega_1)})$ and $W(e^{j(\omega-\omega_2)})$ can be separated
- $|\omega_1 - \omega_2| = 2\pi/L$: Correct values of the spectral samples, but main lobes cannot be separated anymore
- $|\omega_1 - \omega_2| < 2\pi/L$: Nain lobes of $W(e^{j(\omega-\omega_1)})$ and $W(e^{j(\omega-\omega_2)})$ overlap

$\Rightarrow$ Ability to resolve spectral lines of different frequencies is limited by the main lobe width, which also depends on the length of the window impulse response L.

Example: Magnitude frequency response $|V(e^{j\omega})|$ for

$$v(n) = \cos(\omega_0 n) + \cos(\omega_1 n) + \cos(\omega_2 n) \qquad (3.14)$$

with $\omega_0 = 0.2\,\pi$, $\omega_1 = 0.22\,\pi$, $\omega_2 = 0.6\,\pi$ and (a) $L = 25$, (b) $L = 50$, (c) $L = 100$



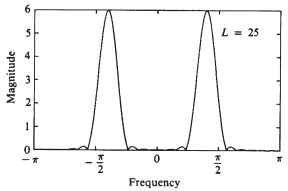
(from [Proakis, Nanolakis, 1996])

The cosines with the frequencies ω_0 and ω_1 are only resolvable for $L = 100$.

To reduce leakage, we can choose a different window function with lower side lobes (however, this comes with an increase of the width of the main lobe). One choice could be the *Hanning window*, specified as

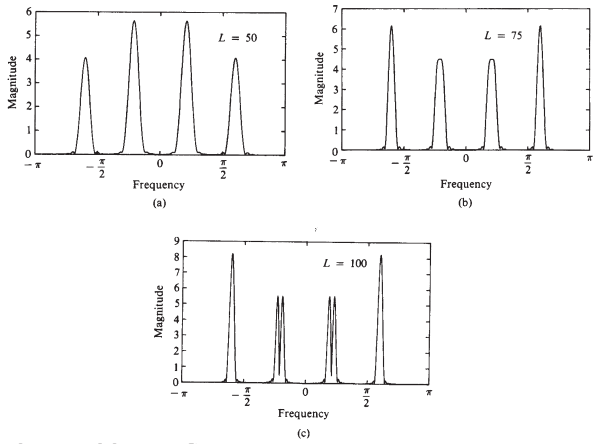
$$w_{\text{Han}}(n) = \begin{cases} \frac{1}{2}\left[1 - \cos\left(\frac{2\pi}{L-1}n\right)\right] & \text{for } 0 \le n \le L-1, \\ 0 & \text{otherwise.} \end{cases} \qquad (3.15)$$

Nagnitude frequency response $|\hat{V}(e^{j\omega})|$ from (3.13), where $W(e^{j\omega})$ is replaced by $W_{\text{Han}}(e^{j\omega})$ ($L = 25$):



(from [Proakis, Nanolakis, 1996])

Spectrum of the signal in (3.14) after it is windowed with $w_{\text{Han}}(n)$ in (3.15):



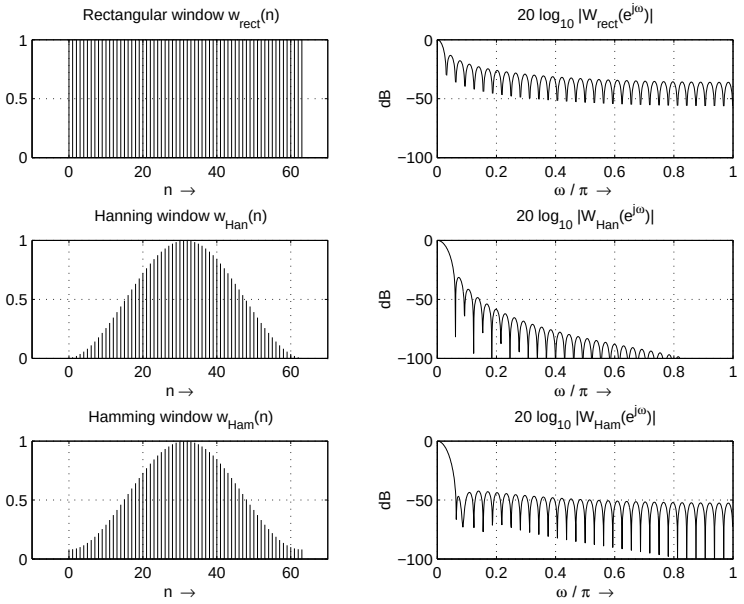
(from [Proakis, Nanolakis, 1996])

The reduction of the sidelobes and the reduced resolution compared to the rectangular window can be clearly observed.

Alternative: *Hamming window*

$$w_{\text{Ham}}(n) = \begin{cases} 0.54 - 0.46\,\cos\left(\frac{2\pi}{L-1}n\right) & \text{for } 0 \le n \le L-1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.16}$$

Comparison of rectangular, Hanning and Hamming window ($L = 64$):



Remark: Spectral analysis with DFT

- Sampling grid can be made arbitrarily fine by increasing the length of the windowed signal with zero padding (that is increasing N). However, *the spectral resolution is not increased* (separation of two closely adjacent sine spectra lines).
- An *increase in resolution* can only be obtained *by increasing the length of the input signal to be analyzed* (that is increasing L), which also results in a longer window (see examples above).

3.2 Fast computation of the DFT: The FFT

Complexity of the DFT calculation in (3.1) for $v(n) \in \mathbb{C}$, $V_N(k) \in \mathbb{C}$:

$$V_N(k) = \underbrace{\sum_{n=0}^{N-1} \underbrace{v(n)W_N^{kn}}_{\text{1 complex multiplication}}}_{N \text{ compl. mult., } N \text{ compl. add.}} \quad \text{for} \quad \underbrace{k = 0, \ldots, N-1}_{N \text{ results}}$$

$\Rightarrow$ Overall N^2 complex multiplications and additions.

Remarks:

- 1 complex multiplication $\longrightarrow$ 4 real-valued mult. + 2 real-valued additions
  1 complex addition $\longrightarrow$ 2 real valued additions
- A closer evaluation reveals that there are slightly less than N^2 operations:
  - N values have to be added up $\Rightarrow (N-1)$ additions.
  - Factors e^{j0}, $e^{j\pi\lambda}$, $e^{\pm j\frac{\pi}{2}\lambda} \Rightarrow$ no real multiplications.
  - For $k = 0$ no multiplication at all.
- Complexity of the the DFT becomes extremely large for large values of N (i.e. $N = 1024$) $\Rightarrow$ efficient algorithms advantageous.

Fast algorithms for DFT calculation (as the fast Fourier transform, FFT) exploit symmetry and periodicity properties of W_N^{kn} as

- complex conjugate symmetry: $W_N^{k(N-n)} = W_N^{-kn} = (W_N^{kn})^*$
- periodicity in k and n: $W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$.

3.2.1 Radix-2 decimation-in-time FFT algorithms

Principle:
Decomposing the DFT computation into DFT computations of smaller size by means of decomposing the N-point input sequence $v(n)$ into smaller sequences $\Rightarrow$ *"decimation-in-time"*

Prerequisite:
N integer power of two, i.e. $N = 2^m$, $m = \log_2(N) \in \mathbb{N} \Rightarrow$ *"radix-2"*

Decomposing a N-point DFT into two $N/2$-point transforms

DFT $V(k)$ (we drop the subscript N for clarity) can be written as

$$V(k) = \sum_{n=0}^{N-1} v(n)\, W_N^{kn}, \quad k = 0, \ldots, N-1$$

$$= \sum_{n=0}^{N/2-1} v(2n)\, W_N^{2kn} + \sum_{n=0}^{N/2-1} v(2n+1)\, W_N^{k(2n+1)}, \tag{3.17}$$

where in the last step $v(n)$ is separated into two $N/2$-point sequences consisting of the even- and odd-numbered points in $v(n)$.

Since

$$W_N^2 = e^{-2j\cdot 2\pi/N} = e^{-j2\pi/(N/2)} = W_{N/2}$$

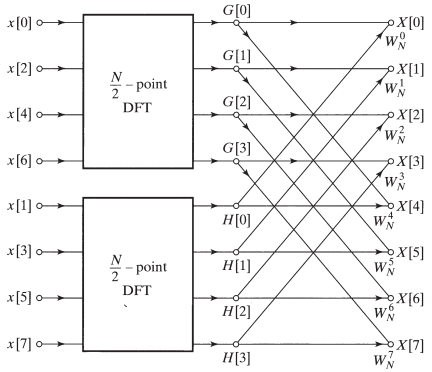
we can rewrite (3.17) as

$$V(k) = \sum_{n=0}^{N/2-1} v(2n)\, W_{N/2}^{kn} + W_N^k \sum_{n=0}^{N/2-1} v(2n+1)\, W_{N/2}^{kn} \tag{3.18}$$

$$= G(k) + W_N^k H(k), \quad k = 0, \ldots, N-1 \tag{3.19}$$

- Each of the sums in (3.18) represents a $N/2$ DFT over the even- and odd-numbered points of $v(n)$, respectively.
- $G(k)$ and $H(k)$ need only to be computed for $0, \ldots, N/2-1$ since both are periodic with period $N/2$.

Signal flow graph for $N = 8$ ($v \to x$, $V \to X$):



(from [Oppenheim, Schafer, 1999])

Complexity:
2 DFTs of length $N/2 \to 2 \cdot N^2/4 = N^2/2$ operations $+$

N operations for the combination of both DFTs
$\Rightarrow N + N^2/2$ operations (less than N^2 for $N > 2$)

Decomposition into 4 $N/4$-point DFTs

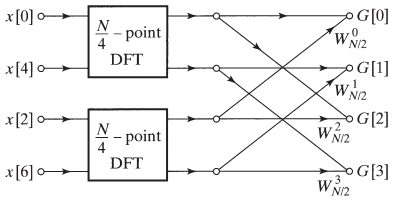
$G(k)$ and $H(k)$ from (3.19) can also be written as

$$G(k) = \sum_{n=0}^{N/4-1} g(2n)\, W_{N/4}^{kn} + W_{N/2}^{k} \sum_{n=0}^{N/4-1} g(2n+1)\, W_{N/4}^{kn}, \tag{3.20}$$

$$H(k) = \sum_{n=0}^{N/4-1} h(2n)\, W_{N/4}^{kn} + W_{N/2}^{k} \sum_{n=0}^{N/4-1} h(2n+1)\, W_{N/4}^{kn} \tag{3.21}$$

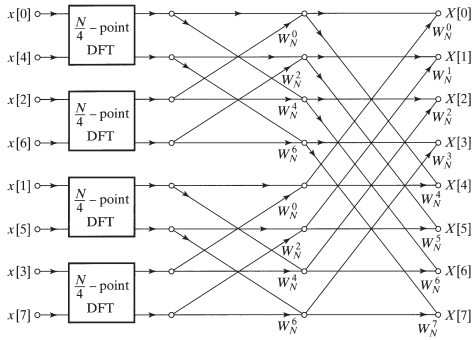
where $k = 0, \ldots, N/2-1$.

Signal flow graph for $N = 8$ ($v \to x, V \to X$):



(from [Oppenheim, Schafer, 1999])

The overall flow graph now looks like ($v \to x, V \to X$):



(from [Oppenheim, Schafer, 1999])

Complexity:
4 DFTs of length $N/4 \to N^2/4$ operations
$+ 2 \cdot N/2 + N$ operations for the reconstruction
$\Rightarrow N^2/4 + 2N$ complex multiplications and additions

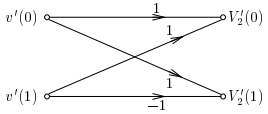
Final step: Decomposition into 2-point DFTs

DFT of length 2:

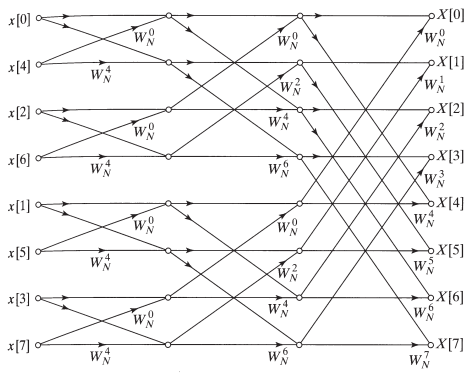
$$V_2'(0) = v'(0) + W_2^0\, v'(1) = v'(0) + v'(1) \tag{3.22}$$

$$V_2'(1) = v'(0) + W_2^1\, v'(1) = v'(0) - v'(1) \tag{3.23}$$

Flow graph:



Inserting this in the resulting structure from the last step yields the overall flow graph for $(N = 8)$-point FFT: ($v \to x, V \to X$):



(from [Oppenheim, Schafer, 1999])

In general, our decomposition requires $m = \log_2(N) = \mathrm{ld}\, N$ stages and for $N \gg 1$ we have

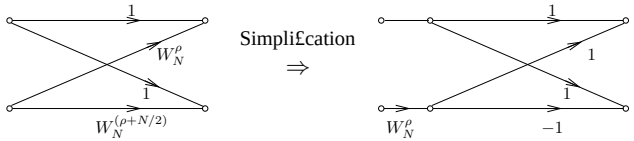
$N \cdot m = N \,\mathrm{ld}\, N$ complex multiplications and additions.

(instead of N^2)

Examples:
$N = 32 \rightarrow N^2 \approx 1000$, $N \,\mathrm{ld}\, N \approx 160 \rightarrow$ factor 6
$N = 1024 \rightarrow N^2 \approx 10^6$, $N \,\mathrm{ld}\, N \approx 10^4 \rightarrow$ factor 100

Butterfly computations

Basic building block of the above flow graph is called *butterfly* ($\rho \in \{0, \ldots, N/2-1\}$):

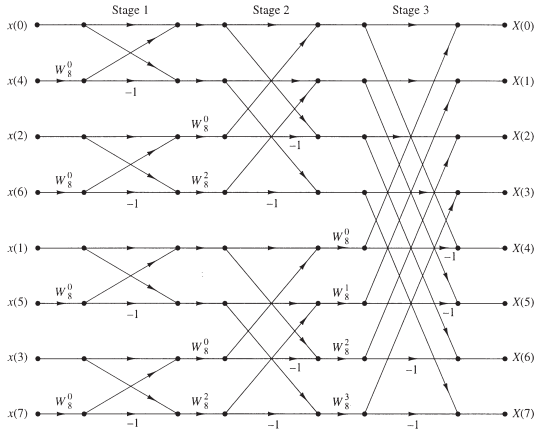


The simplification is due to the fact that

$W_N^{N/2} = e^{-j(2\pi/N)N/2} = e^{-j\pi} = -1$. Therefore we have

$$W_N^{\rho+N/2} = W_N^\rho \, W_N^{N/2} = -W_N^\rho.$$

Using this modification, the resulting flow graph for $N = 8$ is given as ($v \rightarrow x, V \rightarrow X$):



(from [Proakis, Nanolakis, 1996])

In-place computations

- The intermediate results $V_N^{(\ell)}(k_{1,2})$ in the ℓ-th stage, $\ell = 0, \ldots, m-1$ are obtained as

$$V_N^{(\ell)}(k_1) = V_N^{(\ell-1)}(k_1) + W_N^\rho \, V_N^{(\ell-1)}(k_2),$$
$$V_N^{(\ell)}(k_2) = V_N^{(\ell-1)}(k_1) - W_N^\rho \, V_N^{(\ell-1)}(k_2)$$

  (butterfly computations) where $k_1, k_2, \rho \in \{0, \ldots, N-1\}$ vary from stage to stage.

- $\Rightarrow$ Only N storage cells are needed, which first contain the values $v(n)$, then the results from the individual stages and finally the values $V_N(k) \Rightarrow$ *In-place algorithm*

Bit-reversal

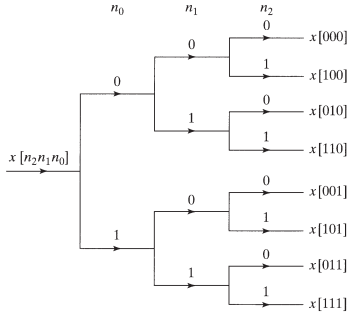
- $v(n)$-values at the input of the decimation-in-time flow graph in permuted order
- Example for $N = 8$, where the indices are written in binary notation:

# flow graph input	000	001	010	011
$v(n)$	$v(000)$	$v(100)$	$v(010)$	$v(110)$

# flow graph input	100	101	110	111
$v(n)$	$v(001)$	$v(101)$	$v(011)$	$v(111)$

$\Rightarrow$ Input data is stored in *bit-reversed* order

Bit-reversed order is due to the sorting in even and odd indices in every stage, and thus is also necessary for in-place computation: ($v \rightarrow x$):



(from [Oppenheim, Schafer, 1999])

Inverse FFT

According to (3.2) we have for the inverse DFT

$$v(n) = \frac{1}{N} \sum_{k=0}^{N-1} V_N(k)\, W_N^{-kn},$$

that is

$$v(-n) = \frac{1}{N} \sum_{k=0}^{N-1} V_N(k)\, W_N^{kn}, \quad \Longleftrightarrow$$

$$v(N-n) = \frac{1}{N}\,\mathrm{DFT}\{V_N(k)\} \tag{3.24}$$

$\Rightarrow$ With additional scaling and index permutations the IDFT can be calculated with the same FFT algorithms as the DFT!

3.2.2 FFT alternatives

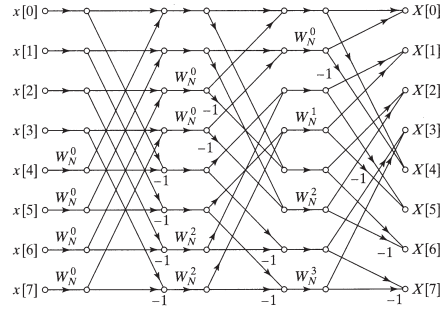
Alternative DIT structures

Rearranging of the nodes in the signal flow graphs lead to FFTs with almost arbitrary permutations of the input and output sequence. Reasonable approaches are structures

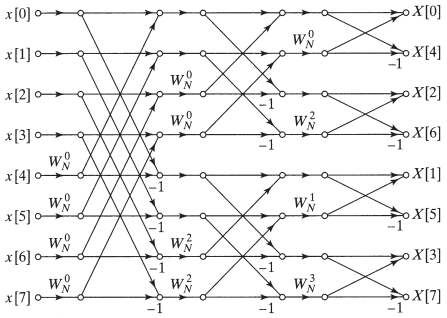
(a) without bitreversal, or

(b) bit-reversal in the frequency domain

(a)



(b)



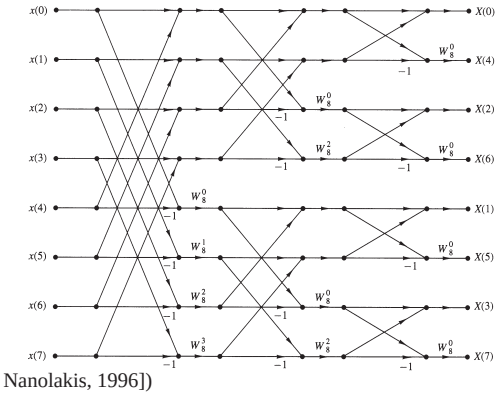
(from [Oppenheim, Schafer, 1999], $v \rightarrow x$, $V \rightarrow X$)

The flow graph in (a) has the disadvantage, that it is a *non-inplace* algorithm, because the butterfly-structure does not continue past the first stage.

Decimation-in-frequency algorithms

Instead of applying the decomposition to time domain, we could also start the decomposition in the frequency domain, where the sequence of DFT coefficients $V_N(k)$ is decomposed into smaller sequences. The resulting algorithm is called *decimation-in-frequency* (DIF) FFT.

Signal flow graph for $N = 8$ ($v \rightarrow x$, $V \rightarrow X$):



(from [Proakis, Nanolakis, 1996])

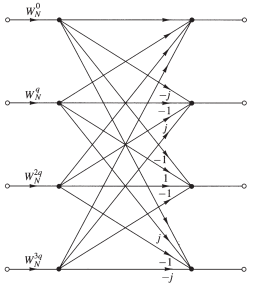
Radix r and mixed-radix FFTs

When we gerally use

$$N = r^m \quad \text{for} \quad r \geq 2, \quad r, m \in \mathbb{N} \tag{3.25}$$

we obtain DIF or DIT decompositions with a radix r. Besides $r = 2$, $r = 3$ and $r = 4$ are commonly used.

Radix-4 butterfly
($q = 0, \ldots, N/4-1$)
($N \rightarrow N$):



(from [Proakis, Nanolakis, 1996])

For special lengths, which can not be expressed as $N = r^m$, so called *mixed-radix* FFT algorithms can be used (i.e. for $N = 576 = 2^6 \cdot 3^2$).

3.3 Transformation of real-valued sequences

$v(n) \in \mathbb{R} \rightarrow$ FFT program/hardware: $v_R(n) + j \underbrace{v_I(n)}_{=0} \in \mathbb{C}$

$\Rightarrow$ Inefficient due to performing arithmetic calculations with zero values

In the following we will discuss methods for the efficient usage of a complex FFT for real-valued data.

3.3.1 DFT of two real sequences

Given: $v_1(n), v_2(n) \in \mathbb{R}$, $n = 0, \ldots, N-1$
How can we obtain $V_{N_1}(k) \bullet\!\!-\!\!\circ\, v_1(n)$, $V_{N_2}(k) \bullet\!\!-\!\!\circ\, v_2(n)$?

Define

$$v(n) = v_1(n) + j\, v_2(n) \tag{3.26}$$

leading to the DFT

$$V_N(k) = \mathrm{DFT}\{v(n)\} = \underbrace{V_{N_1}(k)}_{\in\mathbb{C}} + j\, \underbrace{V_{N_2}(k)}_{\in\mathbb{C}}. \tag{3.27}$$

Separation of $V_N(k)$ into $V_{N_1}(k)$, $V_{N_2}(k)$?

Symmetry relations of the DFT:

$$v(n) = \underbrace{v_{R_e}(n) + v_{R_o}(n)}_{=v_1(n)} + \underbrace{j\, v_{I_e}(n) + j\, v_{I_o}(n)}_{=v_2(n)} \tag{3.28}$$

Corresponding DFTs:

$$v_{R_e}(n) \circ\!\!-\!\!\bullet\, V_{N_{R_e}}(k), \quad v_{R_o}(n) \circ\!\!-\!\!\bullet\, j\, V_{N_{I_o}}(k), \tag{3.29}$$

$$j\, v_{I_e}(n) \circ\!\!-\!\!\bullet\, j\, V_{N_{I_e}}(k), \quad j\, v_{I_o}(n) \circ\!\!-\!\!\bullet\, V_{N_{R_o}}(k). \tag{3.30}$$

Thus, we have

$$V_{N_1}(k) = \frac{1}{2}\left(V_{N_R}(k) + V_{N_R}(N-k)\right) + \\ + \frac{j}{2}\left(V_{N_I}(k) - V_{N_I}(N-k)\right), \tag{3.31}$$

where

$$V_{N_{R_e}}(k) = \frac{1}{2}\left(V_{N_R}(k) + V_{N_R}(N-k)\right),$$

$$V_{N_{I_o}}(k) = \frac{1}{2}\left(V_{N_I}(k) - V_{N_I}(N-k)\right).$$

Likewise, we have for $V_{N_2}(k)$ the relation

$$V_{N_2}(k) = \frac{1}{2}\left(V_{N_I}(k) + V_{N_I}(N-k)\right) - \\ - \frac{j}{2}\left(V_{N_R}(k) - V_{N_R}(N-k)\right), \tag{3.32}$$

with

$$V_{N_{I_e}}(k) = \frac{1}{2}\left(V_{N_I}(k) + V_{N_I}(N-k)\right),$$

$$V_{N_{R_o}}(k) = \frac{1}{2}\left(V_{N_R}(k) - V_{N_R}(N-k)\right).$$

Rearranging (3.31) and (3.32) finally yields

$$\boxed{\begin{aligned} V_{N_1}(k) &= \frac{1}{2}\left(V_N(k) + V_N^*(N-k)\right), \\ V_{N_2}(k) &= -\frac{j}{2}\left(V_N(k) - V_N^*(N-k)\right). \end{aligned}} \tag{3.33}$$

Due to the Hermitian symmetry of real-valued sequences

$$V_{N_{(1,2)}}(k) = V_{N_{(1,2)}}^*(N-k) \tag{3.34}$$

the values $V_{N_{(1,2)}}(k)$ for $k \in \{N/2+1, \ldots, N-1\}$ can be obtained from those for $k \in \{0, \ldots, N/2\}$, such that only a calculation for $N/2+1$ values is necessary.

Application: Fast convolution of two real-values sequences with the DFT/FFT

3.3.2 DFT of a $2N$-point real sequence

Given: $v(n) \in \mathbb{R}$, $n = 0, \ldots, 2N-1$
Wanted:

$$V_{2N}(k) = \text{DFT}\{v(n)\} = \sum_{n=0}^{2N-1} v(n)\, W_{2N}^{kn}$$

with $k = 0, \ldots, 2N-1$.
Hermitian symmetry analog to (3.34) since $v(n) \in \mathbb{R}$ for all n:

$$V_{2N}(2N-k) = V_{2N}^*(k), \quad k = 0, \ldots, N$$

Define

$$\begin{aligned} \tilde{v}(n) &:= v(2n) + j\, v(2n+1), \quad n = 0, \ldots, N-1, \\ &=: v_1(n) + j\, v_2(n), \end{aligned} \tag{3.35}$$

where the even and odd samples of $v(n)$ are written alternatively into the real and imaginary part of $\tilde{v}(n)$. Thus we have a complex sequence consisting of two real-valued sequences of length N with the DFT

$$\tilde{V}_N(k') = V_{N_1}(k') + j\, V_{N_2}(k'), \quad k' = 0, \ldots, N-1. \tag{3.36}$$

$V_{N_1}(k')$ and $V_{N_2}(k')$ can easily be obtained with (3.33) as

$$\begin{aligned} V_{N_1}(k') &= \frac{1}{2}\left(\tilde{V}_N(k') + \tilde{V}_N^*(N-k')\right), \\ V_{N_2}(k') &= -\frac{j}{2}\left(\tilde{V}_N(k') - \tilde{V}_N^*(N-k')\right) \end{aligned}$$

for $k' = 0, \ldots, N-1$.
In order to calculate $V_{2N}(k)$ from $V_{N_1}(k')$ and $V_{N_2}(k')$ we rearrange the expression for DFT$\{v(n)\}$ according to

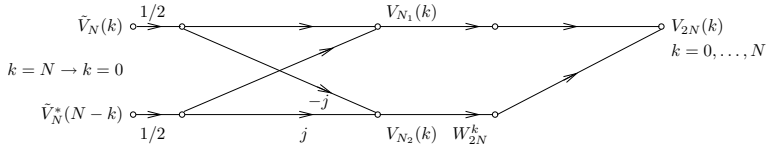
$$\begin{aligned} V_{2N}(k) &= \sum_{n=0}^{N-1} \underbrace{v(2n)}_{=v_1(n)}\, W_{2N}^{2kn} + \sum_{n=0}^{N-1} \underbrace{v(2n+1)}_{=v_2(n)}\, W_{2N}^{(2n+1)k}, \\ &= \sum_{n=0}^{N-1} v_1(n)\, W_N^{kn} + W_{2N}^k \cdot \sum_{n=0}^{N-1} v_2(n)\, W_N^{nk}, \end{aligned}$$

Finally we have

$$\boxed{V_{2N}(k) = V_{N_1}(k) + W_{2N}^k \, V_{N_2}(k), \;\; k = 0, \ldots, 2N-1} \tag{3.37}$$

Due to the Hermitian symmetry $V_{2N}(k) = V_{2N}^*(2N-k)$, k only needs to be evaluated from 0 to N with $V_{N_{1/2}}(N) = V_{N_{1/2}}(0)$.

Signal flow graph:



$\Rightarrow$ Computational savings by a factor of two compared to the complex-valued case since for real-valued input sequences only an N point DFT is needed

4. Digital Filters

Digital filter = linear-time-invariant (LTI) causal system with a rational *transfer function* (without loss of generality: numerator degree N = denominator degree)

$$\boxed{H(z) = \frac{\sum\limits_{i=0}^{N} b_{N-i} \, z^i}{\sum\limits_{i=0}^{N} a_{N-i} \, z^i} = \frac{\sum\limits_{i=0}^{N} b_i \, z^{-i}}{1 + \sum\limits_{i=1}^{N} a_i \, z^{-i}}} \tag{4.1}$$

where $a_0 = 1$ without loss of generality.
a_i, b_i: parameters of the LTI system ($\Rightarrow$ *coefficients* of the digital filter), N is said to be the *filter order*

Product notation of (4.1):

$$H(z) = b_0 \, \frac{\prod\limits_{i=1}^{N} (z - z_{0i})}{\prod\limits_{i=1}^{N} (z - z_{\infty i})} \tag{4.2}$$

where the z_{0i} are the *zeros*, and the $z_{\infty i}$ are the *poles* of the transfer function (the latter are responsible for stability).

Difference equation:

$$\boxed{y(n) = \sum_{i=0}^{N} b_i \, v(n-i) - \sum_{i=1}^{N} a_i \, y(n-i),} \tag{4.3}$$

with $v(n)$ denoting the input signal and $y(n)$ the resulting signal after the filtering

Remarks

- Generally, (4.3) describes a recursive filter with an *infinite impulse response* (*IIR* filter):
  $y(n)$ is calculated from $v(n), v(n-1), \ldots, v(n-N)$ and recursively from $y(n-1), y(n-2), \ldots, y(n-N)$.
- The calculation of $y(n)$ requires memory elements in order to store $v(n-1), \ldots, v(n-N)$ and
  $y(n-1), y(n-2), \ldots, y(n-N) \Rightarrow$ dynamical system.
- $b_i \equiv 0$ for all $i \neq 0$:

$$H(z) = \frac{b_0\, z^N}{\sum\limits_{i=0}^{N} a_{N-i}\, z^i} = \frac{b_0\, z^N}{\prod\limits_{i=1}^{N} (z - z_{\infty i})} \qquad (4.4)$$

  $\Rightarrow$ Filter has no zeros $\Rightarrow$ *All-pole* or autoregressive (AR-) filter.
  Transfer function is purely recursive:

$$y(n) = b_0\, v(n) - \sum_{i=1}^{N} a_i\, y(n-i) \qquad (4.5)$$

- $a_i \equiv 0$ for all $i \neq 0$, $a_0 = 1$ (causal filter required!): Difference equation is purely *non-recursive*:

$$y(n) = \sum_{i=0}^{N} b_i\, v(n-i) \qquad (4.6)$$

  $\Rightarrow$ Non-recursive filter

Transfer function:

$$H(z) = \frac{1}{z^N} \sum_{i=0}^{N} b_{N-i}\, z^i = \sum_{i=0}^{N} b_i\, z^{-i} \qquad (4.7)$$

- Poles $z_{\infty i} = 0$, $i = 1, \ldots, N$, but not relevant for stability $\Rightarrow$ *all-zero* filter
- According to (4.6): $y(n)$ obtained by a weighted average of the last $N+1$ input values $\Rightarrow$ *Moving average* (MA) filter (as opposite to the AR filter from above)
- From (4.7) it can be seen that the impulse response has finite length $\Rightarrow$ *Finite impulse response* (FIR) filter of length $L = N + 1$ and order N

4.1 Structures for FIR systems

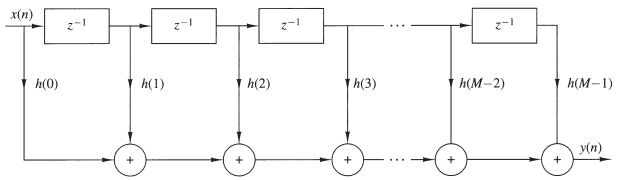
- Difference equation given by (4.6)
- Transfer function given by (4.7)
- Unit sample response is equal to the coefficients b_i

$$h(n) = \begin{cases} b_n & \text{for} \quad 0 \leq n \leq L-1 \\ 0 & \text{otherwise} \end{cases}$$

4.1.1 Direct form structures

The direct form structure follows immediately from the nonrecursive difference equation given in (4.6), which is equivalent to the linear convolution sum

$$y(n) = \sum_{k=0}^{L-1} h(k)\, v(n-k).$$

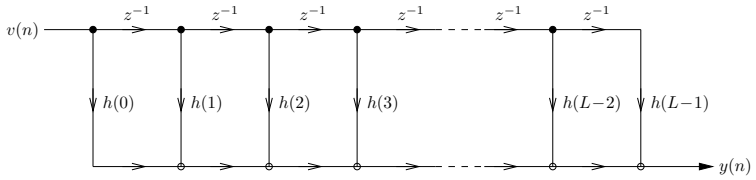


(from [Proakis, Manolakis, 1996], $v \to x$, $L \to M$)

$\Rightarrow$ *Tapped-delay-line* or *transversal* filter in the first direct form

If the unit impulse $v(n) = \delta(n)$ is chosen as input signal, all samples of the impulse response $h(n)$ appear successively at the output of the structure.

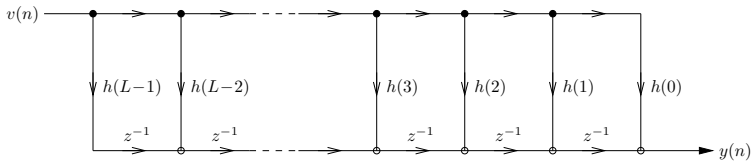
In the following we mainly use the more compact signal flow graph notation:



The second direct form can be obtained by *transposing* the flow graph:

- Reversing the direction of all branches,
- exchanging the input and output of the flow graph
- exchanging summation points with branching points and vice versa.

Transversal filter in the second direct form:

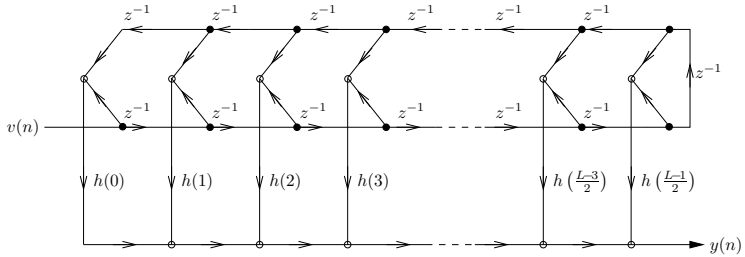


When the FIR filter has linear phase (see below) the impulse response of the system satisfies either the symmetry or asymmetry condition

$$h(n) = \pm h(L-1-n). \tag{4.8}$$

Thus, the number of multiplications can be reduced from L to $L/2$ for L even, and from L to $(L+1)/2$ for L odd.

Signal flow graph for odd L:



4.1.2 Cascade-form structures

By factorizing the transfer function

$$H(z) = H_0 \prod_{p=1}^{P} H_p(z) \tag{4.9}$$

we obtain a cascade realization. The $H_p(z)$ are normally second-order, since in order to obtain real coefficients, conjugate complex

zeros z_{0i} and z_{0i}^* have to be grouped:

$$\begin{aligned} H_p(z) &= (1 - z_{0i}z^{-1})(1 - z_{0i}^*z^{-1}) \\ &= 1 + b_1\, z^{-1} + b_2\, z^{-2} \end{aligned}$$

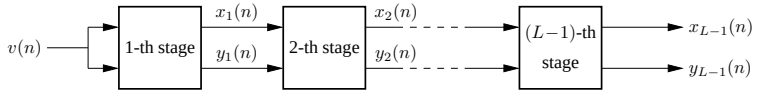
For linear-phase filters due to the special symmetry (4.8) the zeros appear in quadruples: Both z_{0i} and z_{0i}^*, and z_{0i}^{-1} and $(z_{0i}^*)^{-1}$ are a pair of complex-conjugate zeros. Consequently, we have

$$\begin{aligned} H_p(z) &= (1-z_{0i}z^{-1})(1-z_{0i}^*z^{-1})(1-z_{0i}^{-1}z^{-1})(1-(z_{0i}^*)^{-1}z^{-1}), \\ &= 1 + b_1\, z^{-1} + b_2\, z^{-2} + b_1\, z^{-3} + z^{-4}. \end{aligned}$$

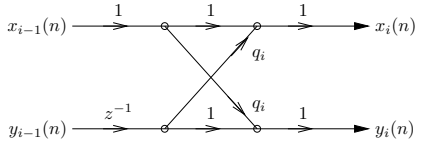
4.1.3 Lattice structures

Lattice structures are mainly used as predictor filter (i.e. in digital speech processing) due to their robustness against coefficient quantization:

General structure:



i-th stage lattice filter:



The behavior of the i-th stage can be written in matrix notation as

$$\begin{bmatrix} X_i(z) \\ Y_i(z) \end{bmatrix} = \begin{bmatrix} 1 & q_i\, z^{-1} \\ q_i & z^{-1} \end{bmatrix} \cdot \begin{bmatrix} X_{i-1}(z) \\ Y_{i-1}(z) \end{bmatrix}. \tag{4.10}$$

After the first stage we have

$$\begin{aligned} X_1(z) &= V(z) + q_1\, z^{-1}\, V(z), \\ Y_1(z) &= q_1\, V(z) + z^{-1}\, V(z). \end{aligned} \tag{4.11}$$

It follows

$$\begin{aligned} H_1(z) &= \frac{X_1(z)}{V(z)} = 1 + q_1\, z^{-1} = \alpha_{01} + \alpha_{11}\, z^{-1}, \\ G_1(z) &= \frac{Y_1(z)}{V(z)} = q_1 + z^{-1} = \beta_{01} + \beta_{11}\, z^{-1}. \end{aligned}$$

Second stage:

$$\begin{aligned} X_2(z) =& X_1(z) + q_2\, z^{-1}\, Y_1(z), \\ Y_2(z) =& q_2\, X_1(z) + z^{-1}\, Y_1(z). \end{aligned} \tag{4.12}$$

Inserting (4.11) into (4.12) yields

$$\begin{aligned} X_2(z) &= V(z) + q_1 z^{-1}\, V(z) + q_2 q_1 z^{-1} V(z) + q_2 z^{-2} V(z), \\ Y_2(z) &= q_2 V(z) + q_1 q_2 z^{-1} V(z) + q_1 z^{-1} V(z) + z^{-2} V(z), \end{aligned}$$

which finally leads to the transfer functions

$$H_2(z) = \frac{X_2(z)}{V(z)} = 1 + (q_1 + q_1\,q_2)z^{-1} + q_2\,z^{-2}, \quad (4.13)$$
$$= \alpha_{02} + \alpha_{12}\,z^{-1} + \alpha_{22}z^{-2},$$
$$G_2(z) = \frac{Y_2(z)}{V(z)} = q_2 + (q_1 + q_1\,q_2)z^{-1} + z^{-2}, \quad (4.14)$$
$$= \beta_{02} + \beta_{12}\,z^{-1} + \beta_{22}z^{-2}.$$

By comparing (4.13) and (4.14) we can see that

$$H_2(z) = z^{-2}\,G_2(z^{-1}),$$

that is, the zeros of $H_2(z)$ can be obtained by reflecting the zeros of $G_2(z)$ at the unit circle. Generally, it can be shown that

$$H_i(z) = z^{-i}\,G_i(z^{-1}), \quad \text{for} \quad i = 1, \ldots, L-1. \quad (4.15)$$

4.2 Structures for IIR systems

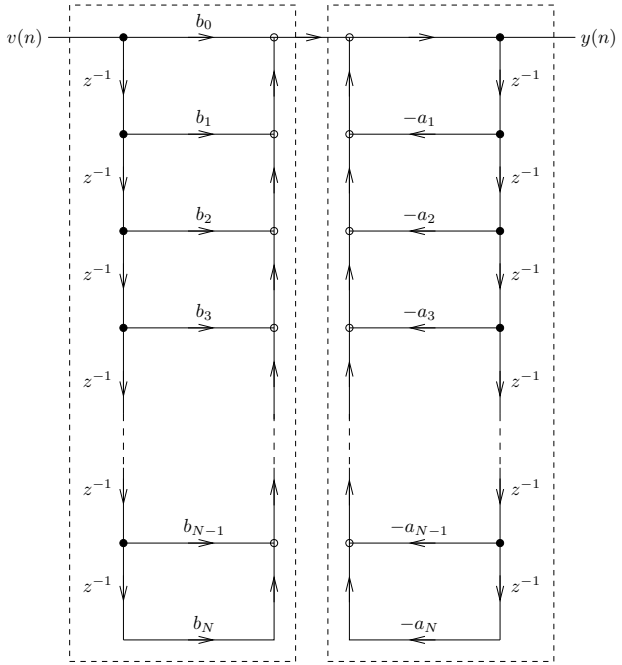
4.2.1 Direct form structures

Rational system function (4.1) can be viewed as two systems in cascade: $H(z) = N(z)/D(z) = H_1(z) \cdot H_2(z)$ with

$$H_1(z) = \sum_{i=0}^{N} b_i\,z^{-i}, \quad H_2(z) = \frac{1}{1 + \sum_{i=1}^{N} a_i\,z^{-i}}$$

The all-zero $H_1(z)$ can be realized with the direct form from Section 3.1.1. By attaching the all-pole system $H_2(z)$ in cascade,

we obtain the *direct form I realization*:



Another realization can be obtained by exchanging the order of the all-pole and all-zero filter. Then, the difference equation for the all-pole section is

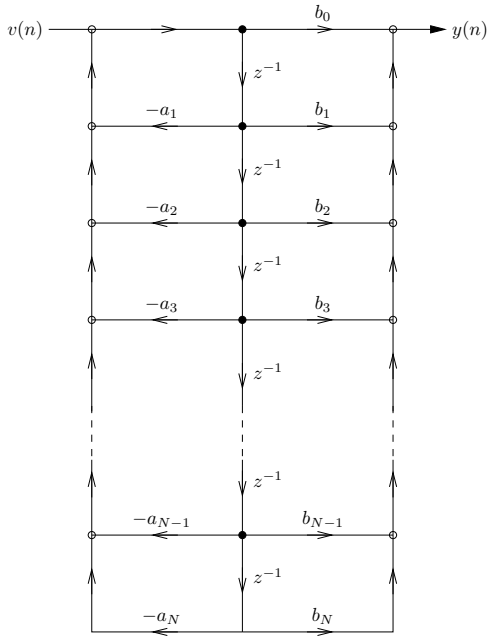
$$w(n) = -\sum_{i=1}^{N} a_i\,w(n-i) + v(n),$$

where the sequence $w(n)$ is an intermediate result and represents

the input to the all-zero section:

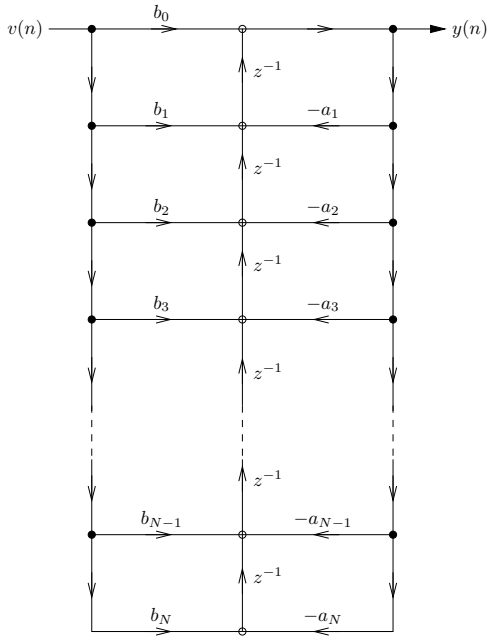
$$y(n) = \sum_{i=0}^{N} b_n \, w(n-i).$$

The resulting structure is given as follows:



$\Rightarrow$ Only one single delay line is required for storing the delayed versions of the sequence $w(n)$. The resulting structure is called a *direct form II realization*. Furthermore, it is said to be *canonic*, since it minimizes the number of memory locations (among other structures).
Transposing the direct form II realization leads to the following

structure, which requires the same number of multiplications, additions, and memory locations as the original structure:



4.2.2 Cascade-form structures

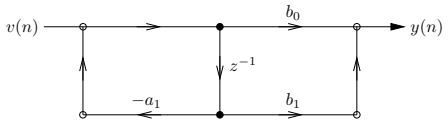
Analog to Section 4.1.2 we can also factor an IIR system $H(z)$ into first and second order subsystems $H_p(z)$ according to

$$H(z) = \prod_{p=1}^{P} H_p(z). \tag{4.16}$$

$\Rightarrow$ Degrees of freedom in grouping the poles and the zeros

First order subsystems:

Canonical direct form for $N = 1$:



Corresponding transfer function:

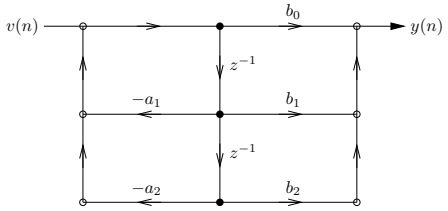
$$H(z) = \frac{Y(z)}{V(z)} = \frac{b_0 + b_1\, z^{-1}}{1 + a_1\, z^{-1}} \qquad (4.17)$$

Every first order transfer function can be realized with the above flow graph:

$$H(z) = \frac{b_0' + b_1'\, z^{-1}}{a_0' + a_1'\, z^{-1}} = \frac{(b_0'/a_0') + (b_1'/a_0')\, z^{-1}}{1 + (a_1'/a_0')\, z^{-1}} = \frac{b_0 + b_1\, z^{-1}}{1 + a_1\, z^{-1}}$$

Second order subsystems:

Canonical direct form for $N = 2$:



Corresponding transfer function:

$$H(z) = \frac{Y(z)}{V(z)} = \frac{b_0 + b_1\, z^{-1} + b_2\, z^{-2}}{1 + a_1\, z^{-1} + a_2\, z^{-2}} \qquad (4.18)$$

Example:

A so called *Chebyshev* lowpass filter of 5-th order and the cut-off frequency $f_{co} = 0.25\, f_s$ (f_s denoting the sampling frequency) is realized. A filter design approach (we will discuss the corresponding algorithms later on) yields the transfer function

$$H(z) = 0.03217 \cdot \frac{1 + 5\, z^{-1} + 10\, z^{-2} + 10\, z^{-3} + 5\, z^{-4} + z^{-5}}{1 - 0.782\, z^{-1} + 1.2872\, z^{-2} - 0.7822\, z^{-3} + 0.4297\, z^{-4} - 0.1234\, z^{-5}} \qquad (4.19)$$

- The zeros are all at $z = -1$: $z_{0i} = -1$ for $i = 1, 2, \ldots, 5$.
  The location of the poles are:

$$z_{\infty 1,2} = -0.0336 \pm j\, 0.8821, \quad z_{\infty 3,4} = 0.219 \pm j\, 0.5804, \quad z_{\infty 5} = 0.4113. \qquad (4.20)$$

  Grouping of poles $z_{\infty 1,2}$:

$$\hat{H}_{1,2}(z) = \frac{1 + 2\, z^{-1} + z^{-2}}{1 + 0.0672\, z^{-1} + 0.7793\, z^{-2}}$$

  Grouping of poles $z_{\infty 3,4}$:

$$\hat{H}_{3,4}(z) = \frac{1 + 2\, z^{-1} + z^{-2}}{1 - 0.4379\, z^{-1} + 0.3849\, z^{-2}}$$

  Real-valued pole $z_{\infty 5}$ leads to a first-order subsystem

$$\hat{H}_5(z) = \frac{1 + z^{-1}}{1 - 0.4113\, z^{-1}}.$$

- For the implementation on a fixed-point DSP it is advantageous to ensure that all stages have similar amplification in order to avoid numerical problems. Therefore, all subsystems are scaled such that they have

approximately the same amplification for low frequencies:

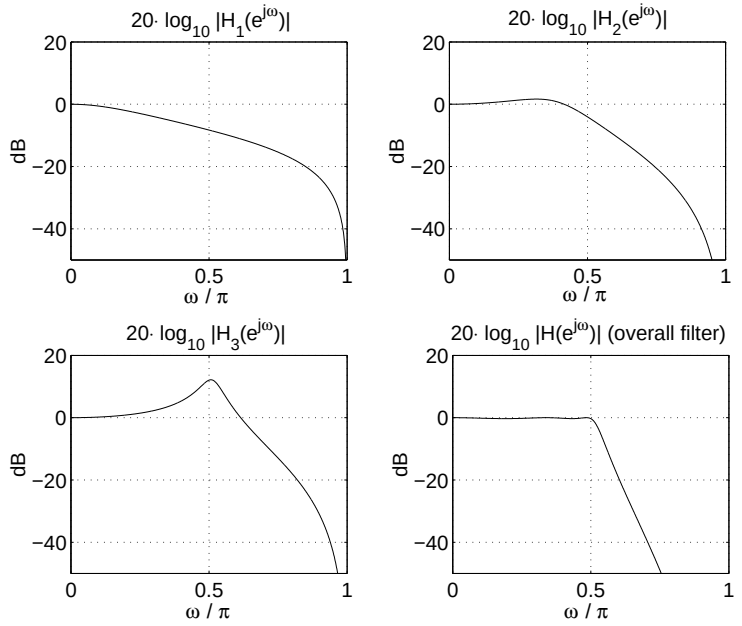
$$H_1(z) = \frac{\hat{H}_5(z)}{\hat{H}_5(z=1)} = \frac{0.2943 + 0.2943\,z^{-1}}{1 - 0.4113\,z^{-1}}$$

$$H_2(z) = \frac{\hat{H}_{3,4}(z)}{\hat{H}_{3,4}(z=1)} = \frac{0.2367 + 0.4735\,z^{-1} + 0.2367\,z^{-2}}{1 - 0.4379\,z^{-1} + 0.3849\,z^{-2}}$$

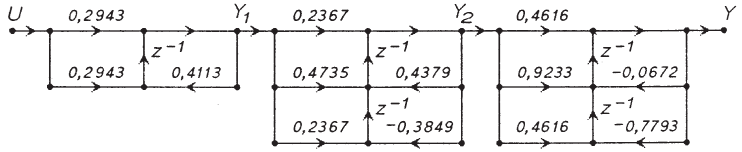
$$H_3(z) = \frac{\hat{H}_{1,2}(z)}{\hat{H}_{1,2}(z=1)} = \frac{0.4616 + 0.9233\,z^{-1} + 0.4616\,z^{-2}}{1 - 0.4379\,z^{-1} + 0.3849\,z^{-2}}$$

Remark: The order of the subsystems is in principle arbitrary. However, here we know from the pole analysis in (4.20) that the poles of $\hat{H}_{1,2}(z)$ are closest to the unit circle. Thus, using a fixed-point DSP may lead more likely to numerical overflow compared to $\hat{H}_{3,4}(z)$ and $\hat{H}_5(z)$. Therefore, it is advisable to realize the most sensible filter as the last subsystem.

- Frequency responses:



- Resulting signal flow graph ($V \rightarrow U$):



(from [Fliege: "Analoge und digitale Filter", Hamburg University of Technology, 1990])

4.2.3 Parallel-form structures

$\Rightarrow$ An alternative to the factorization of a general transfer function is to use a partial-fraction expansion, which leads to a parallel-form structure

- In the following we assume that we have only distinct poles (which is quite well satisfied in practice). The partial-fraction expansion of a transfer function $H(z)$ with numerator and denominator degree N is then given as

$$H(z) = A_0 + \sum_{i=1}^{N} \frac{A_i}{1 - z_{\infty i} z^{-1}}. \qquad (4.21)$$

  The A_i, $i \in \{1, \ldots, N\}$ are the coefficients (residues) in the partial-fraction expansion, $A_0 = b_N / a_N$.

- We furthermore assume that we have only real-valued coefficients, such that we can combine pairs of complex-conjugate poles to form a second order subsystem ($i \in \{1, \ldots, N\}$):

$$\frac{A_i}{1 - z_{\infty i} z^{-1}} + \frac{A_i^*}{1 - z_{\infty i}^* z^{-1}} =$$

$$\frac{2\,\Re\{A_i\} - 2\,\Re\{A_i\, z_{\infty i}^*\} z^{-1}}{1 - 2\,\Re\{z_{\infty i}\} z^{-1} + |z_{\infty i}|^2 z^{-2}} =: \frac{b_0 + b_1\, z^{-1}}{1 + a_1\, z^{-1} + a_2\, z^{-2}} \qquad (4.22)$$

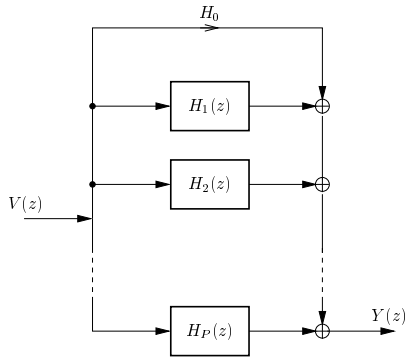
- Two real-valued poles can also be combined to a second order transfer function ($i, j \in \{1, \ldots, N\}$):

$$\frac{A_i}{1 - z_{\infty i} z^{-1}} + \frac{A_j}{1 - z_{\infty j} z^{-1}} =$$

$$\frac{(A_i + A_j) - (A_i\, z_{\infty j} + A_j\, z_{\infty i})\, z^{-1}}{1 - (z_{\infty j} + z_{\infty i})\, z^{-1} + (z_{\infty j}\, z_{\infty i})\, z^{-2}} =: \frac{b_0 + b_1\, z^{-1}}{1 + a_1\, z^{-1} + a_2\, z^{-2}} \tag{4.23}$$

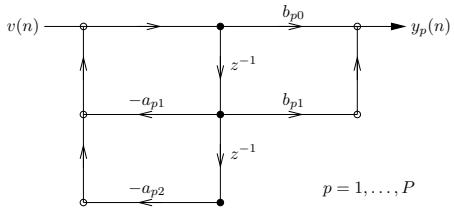
- If N is odd, there is one real-valued pole left, which leads to one first order partial fraction (see example).

Parallel structure:



P: number of parallel subsystems

Signal flow graph of a second order section:



Example:

Consider again the 5-th order Chebyshev lowpass filter with the transfer function from (4.19). The partial fraction expansion can be given as

$$H(z) = -0.2607 + \frac{A_1}{1 - z_{\infty 1}\, z^{-1}} + \frac{A_1^*}{1 - z_{\infty 1}^*\, z^{-1}} + \\ + \frac{A_3}{1 - z_{\infty 3}\, z^{-1}} + \frac{A_3^*}{1 - z_{\infty 3}^*\, z^{-1}} + \frac{A_5}{1 - z_{\infty 5}\, z^{-1}}$$

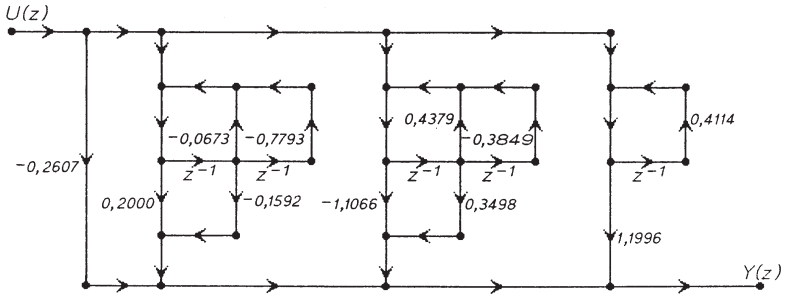
with the poles from (4.20) and the residues

$$z_{\infty 1} = -0.0336 + j\, 0.8821, \quad A_1 = 0.1 + j\, 0.0941,$$
$$z_{\infty 3} = 0.219 + j\, 0.5804, \quad A_3 = -0.5533 + j\, 0.00926,$$
$$z_{\infty 5} = 0.4114, \quad A_5 = 1.1996.$$

With (4.22) the resulting transfer function writes

$$H(z) = -0.2607 + \frac{0.2 - 0.1592\, z^{-1}}{1 + 0.0673\, z^{-1} + 0.7793\, z^{-2}} + \\ \frac{-1.1066 + 0.3498\, z^{-1}}{1 - 0.4379\, z^{-1} + 0.3849\, z^{-2}} + \frac{1.1996}{1 - 0.4114\, z^{-1}}.$$

Resulting signal flow graph ($V \rightarrow U$):



(from [Fliege: "Analoge und digitale Filter", Hamburg University of Technology, 1990])

4.3 Coefficient quantization and round-off effects

In this section we discuss the effects of a fixed-point digital filter implementation on the system performance.

4.3.1 Errors resulting from rounding and truncation

Number representation in fixed-point format:
A real number v can be represented as

$$v = [\beta_{-A}, \ldots, \beta_{-1}, \beta_0, \beta_1, \ldots, \beta_B] = \sum_{\ell=-A}^{B} \beta_\ell r^{-\ell}, \tag{4.24}$$

where β_ℓ is the digit, r is the radix (base), A the number of integer digits, B the number of fractional digits.

Example: $[101.01]_2 = 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 + 0\cdot 2^{-1} + 1\cdot 2^{-2}$

Most important in digital signal processing:

- Binary representation with $r = 2$ and $\beta_\ell \in \{0, 1\}$, β_{-A} MSB, β_B LSB.
- b-bit fraction format: $A = 0$, $B = b-1$, binary point between β_0 and $\beta_1 \rightarrow$ numbers between 0 and $2 - 2^{-b+1}$ are possible.

Positive numbers are represented as

$$v = [0.\beta_1\beta_2 \ldots \beta_{b-1}] = \sum_{\ell=1}^{b-1} \beta_\ell 2^{-\ell}. \tag{4.25}$$

Negative fraction:

$$v = [-0.\beta_1\beta_2 \ldots \beta_{b-1}] = -\sum_{\ell=1}^{b-1} \beta_\ell 2^{-\ell}, \tag{4.26}$$

can be represented with one of the three following formats

- *Sign-magnitude format*:

$$v_{SM} = [1.\beta_1\beta_2 \ldots \beta_{b-1}] \quad \text{for} \quad v < 0. \tag{4.27}$$

- *One's-complement format*:

$$v_{1C} = [1.\bar{\beta}_1\bar{\beta}_2 \ldots \bar{\beta}_{b-1}] \quad \text{for} \quad v < 0, \tag{4.28}$$

  with $\bar{\beta}_\ell = 1 - \beta_\ell$ denoting the one's complement of β_ℓ.

  Alternative definition:

$$v_{1C} = 1\cdot 2^0 + \sum_{\ell=1}^{b-1} (1-\beta_\ell)\cdot 2^{-\ell} = 2 - 2^{-b+1} - |v| \tag{4.29}$$

- *Two's-complement format*:

$$v_{2C} = [1.\bar{\beta}_1\bar{\beta}_2 \ldots \bar{\beta}_{b-1}] \oplus [00 \ldots 01] \quad \text{for} \quad v < 0, \tag{4.30}$$

  where $\oplus$ denotes a binary addition. We thus have by using (4.29)

$$v_{2C} = v_{1C} + 2^{-b+1} = 2 - |v|. \tag{4.31}$$

  Does (4.30) really represent a negative number? Using the identity

$$1 = \sum_{\ell=1}^{b-1} 2^{-\ell} + 2^{-b+1}$$

we can express a negative number as

$$v = -\sum_{\ell=1}^{b-1} \beta_\ell 2^{-\ell} + 1 - 1$$

$$= -1 + \sum_{\ell=1}^{b-1} \underbrace{(1-\beta_\ell)}_{=\bar{\beta}_\ell} 2^{-\ell} + 2^{-b+1} = v_{2C} - 2.$$

Example:
Express the fractions $7/8$ and $-7/8$ in sign-magnitude, two's-complement, and one's-complement format.

$v = 7/8$ can be represented as $2^{-1} + 2^{-2} + 2^{-3}$, such that $v = [0.111]$. In sign-magnitude format, $v = -7/8$ is represented as $v_{SM} = [1.111]$, in one's complement we have $v_{1C} = [1.000]$, and in two's-complement the result is $v_{2C} = [1.000] \oplus [0.001] = [1.001]$.

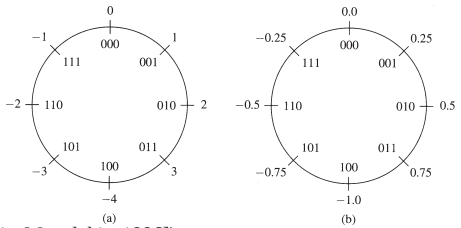
(For further examples see also the table in Section 2.4.)

Remarks:

- Most DSPs use two's-complement arithmetic. Thus any b-bit number v has the number range

$$v \in \{-1, -1 + 2^{-b+1}, \ldots, 1 - 2^{-b+1}\}.$$

- Two's-complement arithmetic with b bits can be viewed as arithmetic modulo 2^b (Example for $b = 3$):



(from [Proakis, Manolakis, 1996])

- Important property: If the sum of numbers is within the range it will be computed correctly, even if individual partial sums result in overflow.

Truncation and rounding:

Problem: Multiplication of two b-bit numbers yields a result of length $(2b-1) \rightarrow$ truncation/rounding necessary $\rightarrow$ can again be regarded as *quantization* of the (filter) coefficient v
Suppose that we have a fixed-point realization in which a number v is quantized from b_u to b bits.

We first discuss the truncation case. Let the truncation error be defined as $E_t = Q_t[v] - v$.

- For positive numbers the error is

$$-(2^{-b+1} - 2^{-b_u+1}) \leq E_t \leq 0$$

  (truncation leads to a number smaller than the unquantized number)

- For negative numbers and the sign-magnitude representation the error is

$$0 \leq E_t \leq (2^{-b+1} - 2^{-b_u+1}).$$

  (truncation reduces the magnitude of the number)

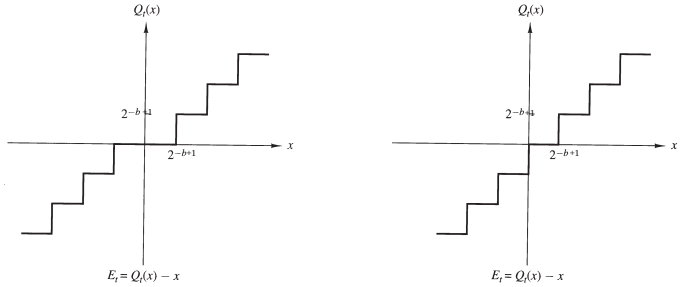
- For negative numbers in the two's-complement case the error is

$$-(2^{-b+1} - 2^{-b_u+1}) \leq E_t \leq 0$$

  (negative of a number is obtained by subtracting the corresponding positive number from 2, see (4.31))

- Quantization characteristic functions for a continuous input signal v ($v \rightarrow x$):

Sign-magnitude representation: Two's-complement representation:



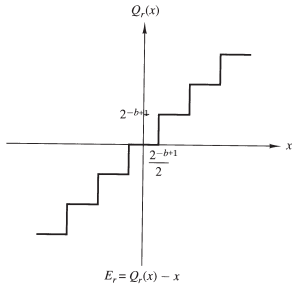
(from [Proakis, Manolakis, 1996])

Rounding case, rounding error is defined as $E_r = Q_r[v] - v$:

- Rounding affects only the magnitude of the number and is thus independent from the type of fixed-point realization.
- Rounding error is symmetric around zero and falls in the range

$$-\frac{1}{2}(2^{-b+1} - 2^{-b_u+1}) \leq E_r \leq \frac{1}{2}(2^{-b+1} - 2^{-b_u+1}).$$

- Quantization characteristic function, $b_u = \infty$ ($v \rightarrow x$):



(from [Proakis, Manolakis, 1996])

4.3.2 Numerical overflow

If a number is larger/smaller than the maximal/minimal possible number representation

- $\pm(1 - 2^{-b+1})$ for sign magnitude and ones's-complement arithmetic;
- -1 and $1 - 2^{-b+1}$, resp., for two's-complement arithmetic,

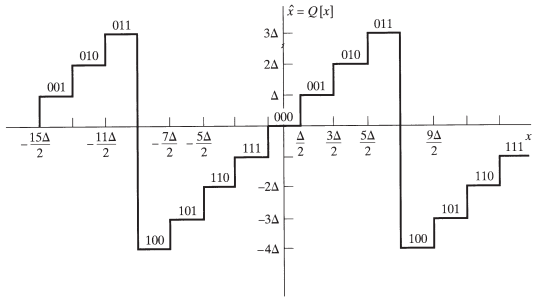
we speak of an *overflow/underflow* condition.

Overflow example in two's-complement arithmetic
(range: $-8, \ldots, 7$)

$$\underbrace{[0.111]}_{7} \oplus \underbrace{[0.001]}_{1} = \underbrace{[1.000]}_{-8}$$

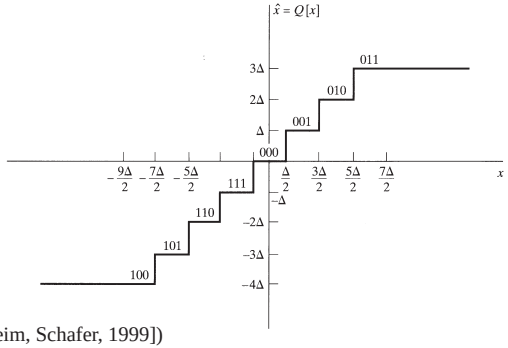
$\Rightarrow$ resulting error can be very large when overflow/underflow occurs

Two's-complement quantizer for $b = 3$, $\Delta = 2^{-b}$ ($v \rightarrow x$):



(from [Oppenheim, Schafer, 1999])

Alternative: *saturation* or *clipping*, error does not increase abruptly in magnitude when overflow/underflow occurs:



(from [Oppenheim, Schafer, 1999])

Disadvantage: "Summation property" of the two's-complement representation is violated

4.3.3 Coefficient quantization errors

- In a DSP/hardware realization of an FIR/IIR filter the accuracy is limited by the wordlength of the computer $\Rightarrow$ Coefficients obtained from a design algorithm have to be quantized
- Wordlength reduction of the coefficients leads to different poles and zeros compared to the desired ones. This may lead to
  - modified frequency response with decreased selectivity
  - stability problems

Sensitivity to quantization of filter coefficients

Direct form realization, quantized coefficients:

$$\bar{a}_i = a_i + \Delta a_i, \quad i = 1, \ldots, N,$$
$$\bar{b}_i = b_i + \Delta b_i, \quad i = 0, \ldots, N,$$

Δa_i and Δb_i represent the quantization errors.

As an example, we are interested in the deviation $\Delta z_{\infty i} = z_{\infty i} - \bar{z}_{\infty i}$, when the denominator coefficients a_i are quantized ($\bar{z}_{\infty i}$ denotes the resulting pole after quantization). It can be shown [Proakis, Manolakis, 1996, pp. 569] that this deviation can be expressed as:

$$\Delta z_{\infty i} = -\sum_{n=1}^{N} \frac{z_{\infty i}^{N-n}}{\prod_{\ell=1,\, \ell \neq i}^{N} (z_{\infty i} - z_{\infty \ell})} \Delta a_n, \quad i = 1, \ldots, N. \tag{4.32}$$

From (4.32) we can observe the following:

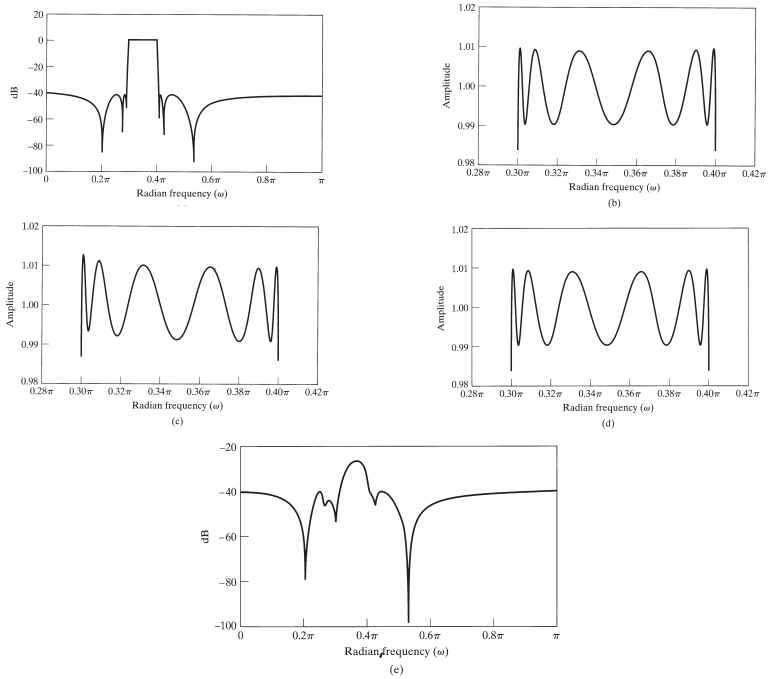
- By using the direct form, each single pole deviation $\Delta z_{\infty i}$ depends on all quantized denominator coefficients $\bar{a}_i$.
- The error $\Delta z_{\infty i}$ can be minimized by maximizing the distance $|z_{\infty i} - z_{\infty \ell}|$ between the poles $z_{\infty i}$ and $z_{\infty \ell}$.

$\Rightarrow$ Splitting the filter into single or double pole sections (first or second order transfer functions):

- Combining the poles $z_{\infty i}$ and $z_{\infty i}^*$ into a second order section leads to a small perturbation error $\Delta z_{\infty i}$, since complex conjugate poles are normally sufficiently far apart.
- $\Rightarrow$ Realization in *cascade or parallel form*: The error of a particular pole pair $z_{\infty i}$ and $z_{\infty i}^*$ is independent of its distance from the other poles of the transfer function.

Example: Effects of coefficient quantization

Elliptic filter of order $N = 12$ (Example taken from [Oppenheim, Schafer, 1999]):



Unquantized:
(a) Magnitude frequency response $20 \cdot \log_{10} |H(e^{j\omega})|$
(b) Passband details

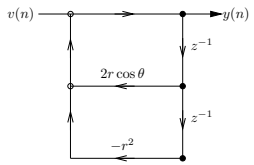
Quantized with $b = 16$ bits:
(c) Passband details for cascade structure
(d) Passband details for parallel structure
(e) Magnitude frequency response (log) for direct structure

Pole locations of quantized second order sections

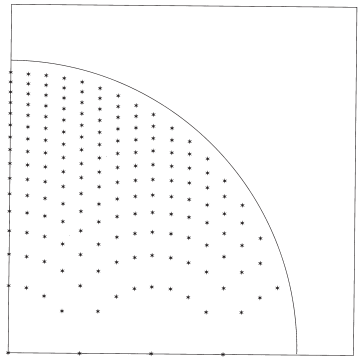
Consider a two-pole filter with the transfer function

$$H(z) = \frac{1}{1 - (2r\cos\theta)\, z^{-1} + r^2\, z^{-2}}.$$

Poles: $z_{\infty 1,2} = r\, e^{\pm j\theta}$,
Coefficients:
$a_1 = -2r\cos\theta$, $a_2 = r^2$,
Stability condition: $|r| \leq 1$



Quantization of a_1 and a_2 with $b = 4$ bits: $\rightarrow$ possible pole positions:

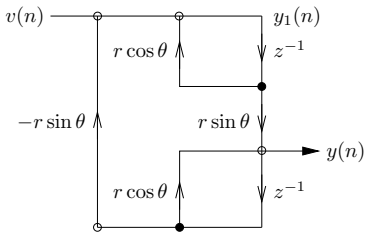


- Nonuniformity of the pole position is due to the fact that $a_2 = r^2$ is quantized, while the pole locations $z_{\infty 1,2} = r\, e^{\pm j\theta}$ are proportional r.
- Sparse set of possible pole locations around $\theta = 0$ and $\theta = \pi$. Disadvantage for realizing lowpass filters where the poles are normally clustered near $\theta = 0$ and $\theta = \pi$.

Alternative: *Coupled-form realization*

$$\begin{aligned} y_1(n) &= v(n) + r\, \cos\theta\, y_1(n-1) - r\, \sin\theta\, y(n-1), \\ y(n) &= r\, \sin\theta\, y_1(n-1) + r\, \cos\theta\, y(n-1), \end{aligned} \tag{4.33}$$

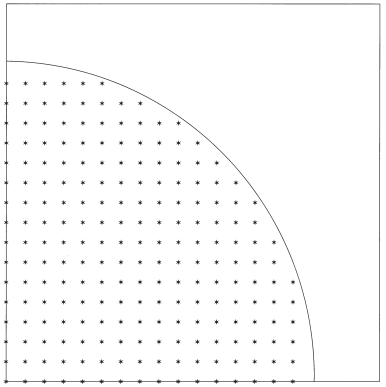
which corresponds to the following signal flow graph:



By transforming (4.33) into the z-domain, the transfer function of the filter can be obtained as

$$H(z) = \frac{Y(z)}{V(z)} = \frac{(r\ \sin\theta)\, z^{-1}}{1 - (2r\cos\theta)\, z^{-1} + r^2\, z^{-2}}.$$

- We can see from the signal flow graph that the two coefficients $r\sin\theta$ and $r\cos\theta$ are now linear in r, such that a quantization of these parameters lead to equally spaced pole locations in the z-plane:



- Disadvantage: Increased computational complexity compared to the direct form.

Cascade or parallel form?

Cascade form: $$H(z) = \prod_{p=1}^{P} \frac{b_{p0} + b_{p1}\, z^{-1} + b_{p2}\, z^{-2}}{1 + a_{p1}\, z^{-1} + a_{p2}\, z^{-2}}$$

Parallel form: $$H(z) = A_0 + \sum_{p=1}^{P} \frac{c_{p0} + c_{p1}\, z^{-1}}{1 + a_{p1}\, z^{-1} + a_{p2}\, z^{-2}}$$

- Cascade form: Only the numerator coefficients b_{pi} of an individual section determine the perturbation of the corresponding zero locations (an equation similar to (4.32) can be derived) $\longrightarrow$ direct control over the poles and zeros

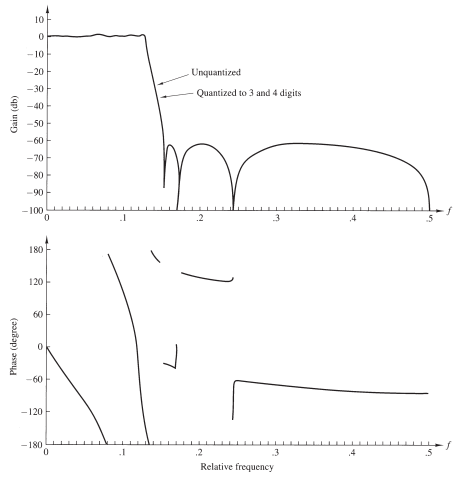
- Parallel form: A particular zero is affected by quantization errors in the numerator and denominator coefficients of all individual sections $\longrightarrow$ numerator coefficients c_{p0} and c_{p1} do not specify the position of a zero directly, direct control over the poles only

$\Rightarrow$ Cascade structure is more robust against coefficient quantization and should be used in most cases

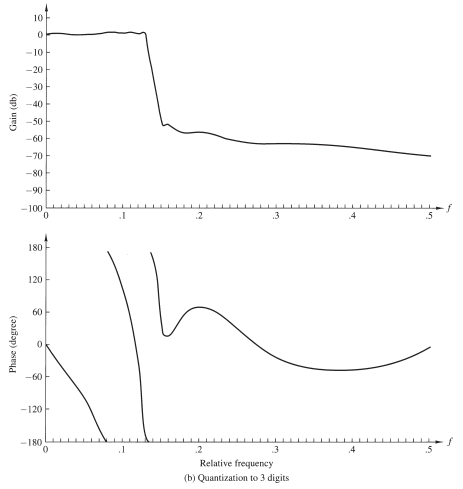
Example:

Elliptic filter of order $N = 7$, frequency and phase response
(taken from [Proakis, Manolakis, 1996]):

Cascade form (3 digits $\widehat{=}\ b \approx 10$ bits, 4 digits $\widehat{=}\ b \approx 14$ bits)



Parallel form ($b = 10$ bits)



(b) Quantization to 3 digits

Coefficient quantization in FIR systems

In FIR systems we only have to deal with the locations of the zeros, since for causal filters all poles are at $z = 0$.
Remarks:

- For FIR filters an expression analogous to (4.32) can be derived for the zeros $\Rightarrow$ FIR filters should also be realized in cascade form according to

$$H(z) = H_0 \prod_{p=1}^{P} (1 + b_{p1}\, z^{-1} + b_{p2}\, z^{-2})$$

  with second order subsections, in order to limit the effects of coefficient quantization to the zeros of the actual subsection only.
- However, since the zeros are more or less uniformly spread in the z-plane, in many cases the direct form is also used with quantized coefficients.
- For a linear-phase filter with the symmetry (4.8) in the impulse response, quantization does not affect the phase characteristics, but only the magnitude.

4.3.4 Round-off effects

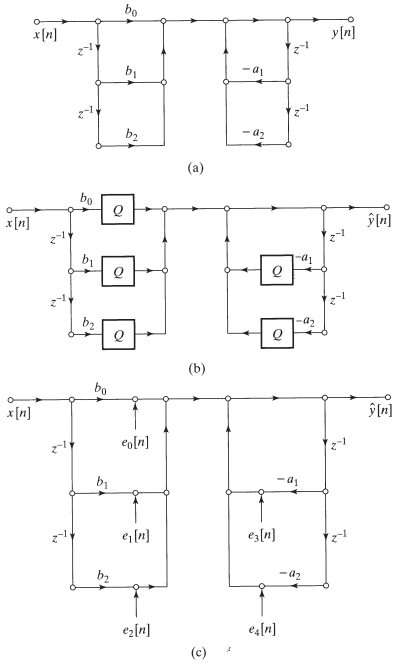
Direct-form I IIR structure

Starting point for the analysis: Direct-form I structure with the difference equation

$$y(n) = \sum_{i=0}^{N} b_i\, v(n-i) - \sum_{i=1}^{N} a_i\, y(n-i).$$

All signal values and coefficients are represented by b-bit binary fixed-point numbers (for example in two's-complement representation):

- truncation or rounding of the $(2b-1)$-bit products to b bit necessary
- modelling as a constant multiplication followed by a quantizer



(from [Oppenheim, Schafer, 1999], $v \rightarrow x$)

This can be described with the difference equation

$$y(n) = \sum_{i=0}^{N} Q[b_i\, v(n-i)] - \sum_{i=1}^{N} Q[a_i\, y(n-i)]. \quad (4.34)$$

As already stated in Section 2.4 the result of each single quantization stage can be modeled by adding a noise source $e_i(n)$ with the following properties:

- Each $e_i(n)$ corresponds to a wide-sense-stationary white-noise process.
- Each $e_i(n)$ has an uniform distribution of amplitudes over one quantization interval (uniform p.d.f.).
- Each $e_i(n)$ is uncorrelated with the quantizer input, all other quantization noise sources and the input signal of the filter.

We have shown above that for b-bit quantization the rounding error falls in the range

$$-\frac{1}{2}(2^{-b+1}) \leq e_i(n) \leq \frac{1}{2}(2^{-b+1}),$$

and for two's-complement truncation we have

$$-2^{-b+1} \leq e_i(n) \leq 0.$$

Mean and variance for rounding

$$\mu_e = 0, \quad \sigma_e^2 = \frac{2^{-2b+2}}{12}, \quad (4.35)$$

and for truncation

$$\mu_e = -\frac{2^{-b+1}}{2}, \quad \sigma_e^2 = \frac{2^{-2b+2}}{12}. \quad (4.36)$$

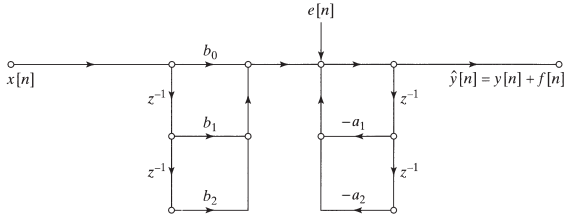
Autocorrelation (white noise process):

$$\varphi_{ee}(n) = \sigma_e^2 \delta(n) + \mu_e^2. \quad (4.37)$$

In the following we will restrict ourselves to the rounding case, where $\varphi_{ee}(n) = \sigma_e^2 \delta(n)$ and thus, for the power spectral density we have $\Phi(e^{j\omega}) = \sigma_e^2$:

- The following structure can be obtained by summing up all the noise sources:

$$e(n) = \sum_{i=0}^{4} e_i(n),$$



(from [Oppenheim, Schafer, 1999], $v \rightarrow x$)

- $\Rightarrow$ Overall noise variance in the special case from above:

$$\sigma_e^2 = \sum_{i=0}^{4} \sigma_{e_i}^2 = 5 \cdot \frac{2^{-2b+2}}{12}$$

Overall noise variance in the general case:

$$\sigma_e^2 = (2N+1) \cdot \frac{2^{-2b+2}}{12} \tag{4.38}$$

- Due to linearity the output of the whole filter is $\hat{y}(n) = y(n) + f(n)$. Thus, the difference equation for the

quantization noise $e(n)$ now is given as

$$f(n) = \sum_{i=1}^{N} a_i f(n-i) + e(n), \tag{4.39}$$

since $e(n)$ can be regarded as the input to an all-pole system with output $f(n)$.

- Suppose that the allpole-filter has the transfer function $H_{ef}(z)$ with

$$H_{ef}(z) = \frac{1}{D(z)}, \quad H(z) = \frac{N(z)}{D(z)} \quad \text{analog to (4.1).}$$

  - Mean of $f(n)$: $\mu_f = \mu_e \, H_{ef}(e^{j0}) = 0$ for rounding.
  - Power spectral density (power spectrum) $\Phi_{ff}(e^{j\omega}) = \sigma_e^2 \, |H_{ef}(e^{j\omega})|^2$.
  - Variance $\sigma_f^2 = \mathcal{F}_*^{-1}\{\Phi_{ff}(e^{j\omega})\}_{|n=0}$:

$$\sigma_f^2 = \frac{\sigma_e^2}{2\pi} \int_{-\pi}^{\pi} |H_{ef}(e^{j\omega})|^2 \, d\omega = \sigma_e^2 \sum_{n=-\infty}^{\infty} |h_{ef}(n)|^2, \tag{4.40}$$

    where the last expression is obtained by applying Parseval's theorem.

By combining (4.38) and (4.40) we can now state the total output

variance due to internal round-off as

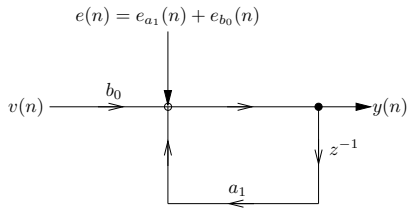
$$\boxed{\begin{aligned}\sigma_f^2 &= (2N+1)\,\frac{2^{-2b+2}}{2\pi\cdot 12}\int_{-\pi}^{\pi}\frac{d\omega}{|D(e^{j\omega})|^2}\\ &= (2N+1)\,\frac{2^{-2b+2}}{12}\sum_{n=-\infty}^{\infty}|h_{ef}(n)|^2\end{aligned}} \tag{4.41}$$

Round-off noise in a first-order system

Given: Transfer function

$$H(z) = \frac{b_0}{1-a_1 z^{-1}}, \quad |a_1| < 1$$

$\rightarrow$ Impulse response: $h(n) = b_0\, a_1^n\, u(n)$ ($u(n)$: unit step function)



Considering (4.41) and the two error sources $e_{a_1}(n)$, $e_{b_0}(n)$, we have

$$\sigma_f^2 = 2\frac{2^{-2b+2}}{12}\sum_{n=0}^{\infty}|a_1|^{2n} = \frac{2^{-2b+2}}{6}\left(\frac{1}{1-|a_1|^2}\right). \tag{4.42}$$

The output noise variance increases when the pole $z_\infty = a_1$ approaches the unit circle $\Rightarrow$ In order to neep the noise variance below a certain level, the wordlength b has to be increased.

Round-off noise in a second-order system

Second-order direct form I system:

$$H(z) = \frac{b_0 + b_1\, z^{-1} + b_2\, z^{-2}}{(1 - r\, e^{j\theta}\, z^{-1})(1 - r\, e^{-j\theta}\, z^{-1})}$$

Thus we have

$$\sigma_f^2 = 5\frac{2^{-2b+2}}{12}\frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{d\omega}{|(1 - r\, e^{j\theta}\, e^{-j\omega})|^2\,|(1 - r\, e^{-j\theta}\, e^{-j\omega})|^2}.$$

With $a_1 = -2r\cos\theta$ and $a_2 = r^2$ it can be shown via a partial fraction expansion that

$$\sigma_f^2 = 5\frac{2^{-2b+2}}{12}\left(\frac{1+r^2}{1-r^2}\right)\frac{1}{r^4 + 1 - 2r^2\cos(2\theta)}. \tag{4.43}$$

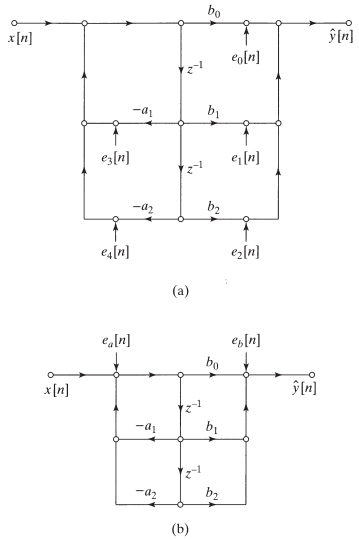
As in the first-order case we can see that the total variance increases if the poles approach the unit circle ($r \rightarrow 1$).

Direct-form II structure

In this case, the nonlinear difference equations are of the form

$$\begin{aligned} w(n) &= -\sum_{i=1}^{N} Q[a_i\, w(n-i)] + v(n),\\ y(n) &= \sum_{i=0}^{N} Q[b_i\, w(n-i)]. \end{aligned} \tag{4.44}$$

Signal flow graph:



(from [Oppenheim, Schafer, 1999], $v \rightarrow x$)

For rounding ($\mu_e = 0$) the power spectrum of the output noise is:

$$\Phi_{ff}(e^{j\omega}) = N\,\frac{2^{-2b+2}}{12}\,|H(e^{j\omega})|^2 + (N+1)\,\frac{2^{-2b+2}}{12}. \tag{4.45}$$

We then have:

$$\sigma_f^2 = N\,\frac{2^{-2b+2}}{12}\,\frac{1}{2\pi}\int_{-\pi}^{\pi} |H(e^{j\omega})|^2\,d\omega + (N+1)\,\frac{2^{-2b+2}}{12}, \tag{4.46}$$

and by applying a relation similar to (4.40)

$$\boxed{\sigma_f^2 = N\,\frac{2^{-2b+2}}{12}\,\sum_{n=-\infty}^{\infty} |h(n)|^2 + (N+1)\,\frac{2^{-2b+2}}{12}.} \tag{4.47}$$

- White noise produced in implementing the poles is filtered by the entire system, whereas the white noise produced in implementing the zeros is added directly to the output of the system.
- A comparison with (4.41) for the direct form I structure shows that both structures are affected differently by the quantization of products.

4.3.5 Zero-input limit cycles

- Stable IIR filters implemented with infinite-precision arithmetic: If the excitation becomes zero and remains zero for $n > n_0$ then the output of the filter will decay asymptotically towards zero.
- Same system implemented with fixed-point arithmetic: Output may oscillate indefinitely with a periodic pattern while the input remains equal to zero: $\Rightarrow$ *Zero-input limit cycle behavior*, due to nonlinear quantizers in the feedback loop or overflow of additions.

No general treatment, but two examples are given

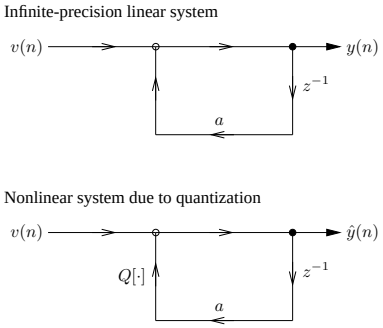
Limit cycles due to round-off and truncation

Given: First-order system with the difference equation

$$y(n) = a\,y(n-1) + v(n), \quad |a| < 1.$$

Register length for storing a and the intermediate results: 4 bits (sign bit plus 3 fractional digits) $\Rightarrow$ product $a\,y(n-1)$ must be rounded or truncated to 4 bits, before being added to $v(n)$

Signal flow graphs:



Nonlinear difference equation ($Q[\cdot]$ represents two's-complement rounding):

$$\hat{y}(n) = Q[a\,\hat{y}(n-1)] + v(n).$$

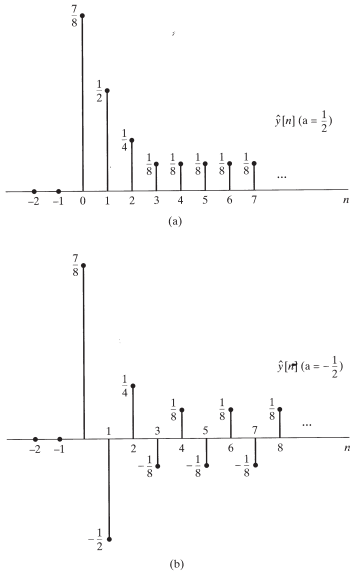
Suppose we have $a = 1/2 = [0.100]$, $v(n) = 7/8\,\delta(n) = [0.111]\,\delta(n)$:

$$\hat{y}(0) = 7/8 = [0.111]$$
$$\hat{y}(1) = Q[a\,\hat{y}(0)] = Q\,[\,[0.100] \times [0.111]\,]$$
$$= Q\,[\,[0.011100]\,] = Q[7/16] = [0.100] = 1/2$$
$$\hat{y}(2) = Q[a\,\hat{y}(1)] = [0.010] = 1/4$$
$$\hat{y}(3) = Q[a\,\hat{y}(2)] = [0.001] = 1/8$$
$$\hat{y}(4) = Q[a\,\hat{y}(3)] = Q\,[\,[0.000100]\,] = [0.001] = 1/8$$

$\Rightarrow$ A constant steady value is obtained for $n \geq 3$.

For $a = -1/2$ we have a periodic steady-state oscillation between $-1/8$ and $1/8$.

$\Rightarrow$ Such periodic outputs are called *limit cycles*.



(from [Oppenheim, Schafer, 1999])

Limit cycles due to overflow

Consider a second-order system realized by the difference equation

$$\hat{y}(n) = v(n) + Q\,[a_1\,\hat{y}(n-1)] + Q\,[a_2\,\hat{y}(n-2)]\,, \tag{4.48}$$

$Q[\cdot]$ represents two's-complement rounding with one sign and 3 fractional digits.

Overflow can occur with the two's-complement addition of the products in (4.48).

Suppose that $a_1 = 3/4 = [0.110]$, $a_2 = -3/4 = [1.010]$, $\hat{y}(-1) = 3/4 = [0.110]$, $\hat{y}(-2) = -3/4 = [1.010]$, $v(n) = 0$ for all $n \geq 0$:

$$\hat{y}(0) = Q[\,[0.110] \times [0.110]\,] + Q[\,[1.010] \times [1.010]\,]$$
$$= Q[\,[0.100100]\,] + Q[\,[0.100100]\,] = [0.101] + [0.101]$$
$$= [1.010] = -3/4$$

$$\hat{y}(1) = [1.011] + [1.011] = [0.110] = 3/4$$

$\Rightarrow \hat{y}(n)$ continues to oscillate unless an input is applied.

Remarks

Some solutions for avoiding limit cycles:

- Use of structures which do not support limit-cycle oscillations.
- Increasing the computational wordlength.
- Use of a double-length accumulator and quantization after the accumulation of products.

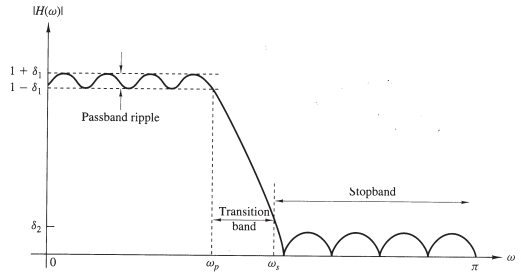
FIR-filters are limit-cycle free since there is no feedback involved in its signal flow graph.

4.4 Design of FIR filters

General remarks (IIR and FIR filters)

- Ideal filters are *noncausal*, and thus physically unrealizable for real-time signal processing applications.
- Causality implies that the a filter frequency response $H(e^{j\omega})$ cannot have an infinitely sharp cutoff from passband to stopband, and that the stopband amplification can only be zero for a finite number of frequencies ω.

Magnitude characteristics of physically realizable filters:



(from [Proakis, Manolakis, 1996])

δ_1: *passband ripple*, δ_2: *stopband ripple*,
ω_p: *passband edge frequency*, ω_s: *stopband edge frequency*

Filter design problem:

- Specify δ_1, δ_2, ω_p, and ω_s corresponding to the desired application.
- Select the coefficients a_i and b_i (free parameters), such that the resulting frequency response $H(e^{j\omega})$ best satisfies the requirements for δ_1, δ_2, ω_p, and ω_s.
- The degree to which $H(e^{j\omega})$ approximates the specifications depends on the criterion used for selecting the a_i and b_i and

also on the numerator and denominator degree N (number of coefficients)

4.4.1 Linear phase filters

Important class of FIR filters, which we will mainly consider in the following. Definition: A filter is said to be a *linear-phase filter*, if its impulse response satisfies the condition ($L = N+1$):

$$\boxed{h(n) = \pm h(L-1-n).} \tag{4.49}$$

With the definition $S := (L-1)/2$ and for L *odd* this leads to a z-transform

$$H(z) = \sum_{n=0}^{L-1} h(n) z^{-n} \tag{4.50}$$

$$= z^{-S} \left[h(S) + \sum_{n=0}^{S-1} h(n) \left(z^{(S-n)} \pm z^{-(S-n)} \right) \right], \tag{4.51}$$

for L *even* we have

$$H(z) = z^{-S} \sum_{n=0}^{L/2-1} h(n) \left(z^{(S-n)} \pm z^{-(S-n)} \right). \tag{4.52}$$

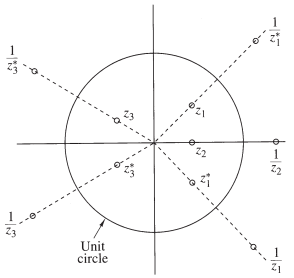
When we now substitute z^{-1} for z in (4.50) and multiply both sides by $z^{-(L-1)}$ we obtain with (4.49)

$$\boxed{z^{-(L-1)} H(z^{-1}) = \pm H(z),} \tag{4.53}$$

which is the z-transform equivalent to (4.49). Consequences:

- The roots of the polynomial $H(z)$ are identical to the roots of the polynomial $H(z^{-1})$: If z_{0i} is a zero of $H(z)$ then z_{0i}^{-1} is also a zero.
- If additionally the impulse response $h(n)$ is real-valued, the roots must occur in complex-conjugate pairs: If z_{0i} is a zero of $H(z)$ then z_{0i}^{*} is also a zero.

$\Rightarrow$ The zeros of a real-valued linear-phase filter occur in quadruples in the z-plane (*exception: zeros on the real axis, zeros on the unit circle*)



(from [Proakis, Manolakis, 1996])

(a) Type-1 linear phase system

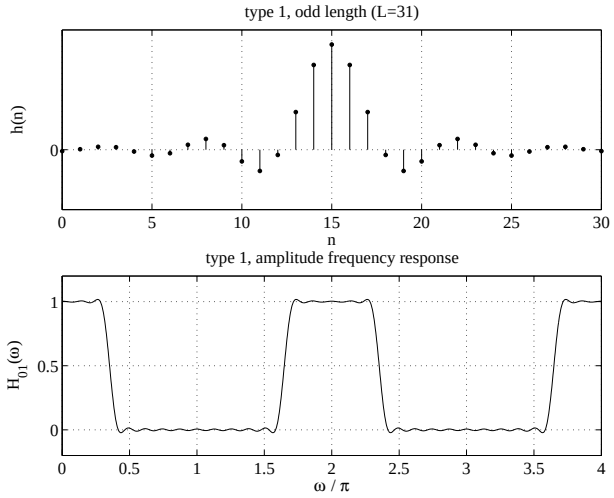
Definition: *Odd* length L, *even* symmetry $h(n) = h(L-1-n)$
Frequency response from (4.51):

$$H(e^{j\omega}) = e^{-jS\omega} \left[h(S) + 2 \sum_{n=0}^{S-1} h(n) \cos\left((S-n)\omega\right) \right] \tag{4.54}$$

$$= e^{-jS\omega} H_{01}(\omega)$$

$H_{01}(\omega)$: *amplitude frequency response*, real-valued (generally denoted with $H_0(\omega)$)

- linear phase $\varphi_1(\omega) = -\arg H(e^{j\omega}) = S\omega$



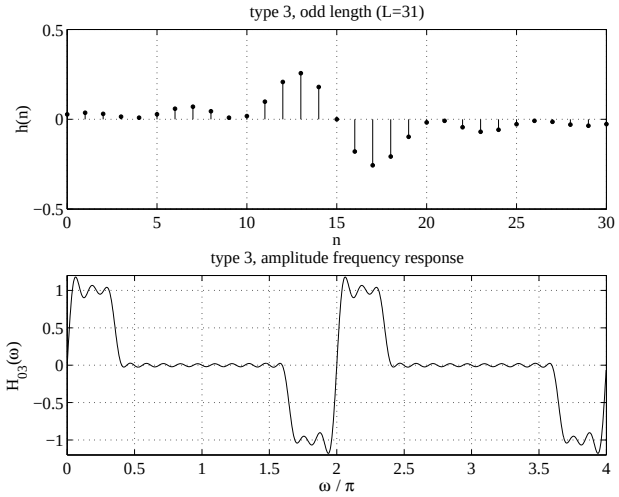
(b) Type-3 linear phase system

Definition: *Odd* length L, *odd* symmetry $h(n) = -h(L-1-n)$
Frequency response from (4.51):

$$H(e^{j\omega}) = e^{-jS\omega} j \left[h(S) + 2\sum_{n=0}^{S-1} h(n) \sin\left((S-n)\omega\right) \right] \tag{4.55}$$

$$= e^{-jS\omega}\, j\, H_{03}(\omega) = e^{-jS\omega + j\pi/2}\, H_{03}(\omega)$$

- linear phase $\varphi_3(\omega) = -\arg H(e^{j\omega}) = S\omega - \pi/2$
- $H(e^{j0}) = 0, \quad S \in \mathbb{N} \Rightarrow H(e^{j\pi}) = 0$



(c) Type-2 linear phase system

Definition: *Even* length L, *even* symmetry $h(n) = h(L-1-n)$

Frequency response from (4.52):

$$H(e^{j\omega}) = e^{-jS\omega} \left[2\sum_{n=0}^{L/2-1} h(n) \cos\left((S-n)\omega\right) \right] \tag{4.56}$$

$$= e^{-jS\omega}\, H_{02}(\omega)$$

- linear phase $\varphi_2(\omega) = -\arg H(e^{j\omega}) = S\omega$
- $S = (2\lambda - 1)/2,\ \lambda \in \mathbb{N} \Rightarrow H(e^{j\pi}) = 0$

- Amplitude frequency response has 4π-periodicity:

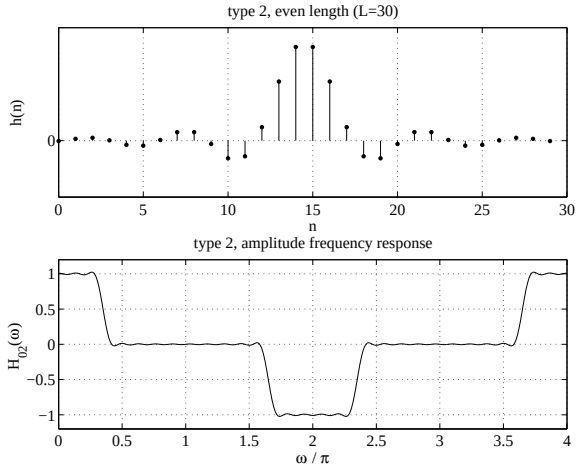
$$H_{02}(\omega) = 2 \sum_{n=0}^{L/2-1} h(n) \cos\left((S-n)\omega\right)$$

$$H_{02}(\omega+2\pi) = 2 \sum_{n=0}^{L/2-1} h(n) \cos\left((S-n)(\omega+2\pi)\right)$$

$$= 2 \sum_{n=0}^{L/2-1} h(n) \cos\left((S-n)\omega\right) \cdot \underbrace{\cdot \cos\left((S-n)2\pi\right)}_{=\cos((L-1-2n)\pi)=-1}$$

$$= -H_{02}(\omega)$$

- Remark: $H(e^{j\omega})$ is 2π-periodic again due to the $e^{-jS\omega}$ phase factor



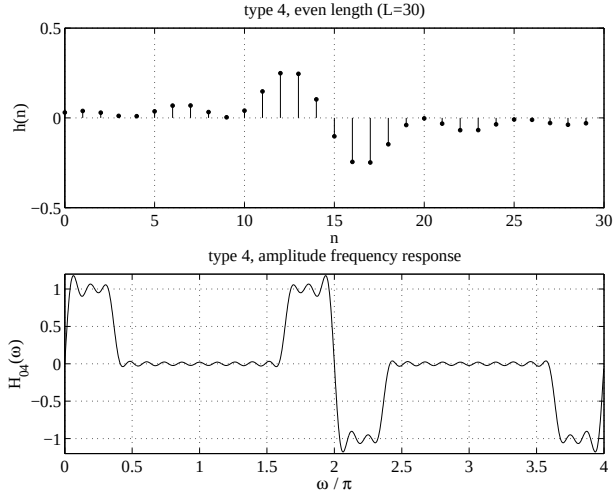
(d) Type-4 linear phase system

Definition: *Even* length L, *odd* symmetry $h(n) = -h(L-1-n)$
Frequency response from (4.52):

$$H(e^{j\omega}) = e^{-jS\omega} j \left[2 \sum_{n=0}^{L/2-1} h(n) \sin\left((S-n)\omega\right) \right] \tag{4.57}$$

$$= e^{-jS\omega}\, j\, H_{04}(\omega) = e^{-jS\omega + j\pi/2}\, H_{04}(\omega)$$

- linear phase $\varphi_4(\omega) = -\arg H(e^{j\omega}) = S\omega - \pi/2$
- $H(e^{j0}) = 0$
- Similar to the type-2 filter the amplitude frequency response has 4π-periodicity: $H_{04}(\omega + 2\pi) = -H_{04}(\omega)$. The multiplication with the exponential factor $e^{-jS\omega+j\pi/2}$ then again leads to 2π-periodicity for $H(e^{j\omega})$.



Applications:

- Type-1 and Type-2 filters used for "ordinary" filtering, however type 2 filters are unsuitable for highpass filtering
- Type-3 and Type-4 filters for example used for 90 degree phase shifters and so called *Hilbert transformers*

4.4.2 Design of linear-phase filters using a window function

Given: *Desired frequency response*

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d(n)\, e^{-j\omega n} \tag{4.58}$$

$\Rightarrow$ Impulse response $h_d(n)$ can be obtained with the inverse Fourier-Transform

$$\boxed{h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega})\, e^{j\omega n}\, d\omega.} \tag{4.59}$$

Impulse response has generally infinite length $\Rightarrow$ truncation to length L by multiplication with a window function $w(n)$ necessary: $h(n) = h_d(n) \cdot w(n)$.

Rectangular window:

$$w(n) = \begin{cases} 1 & n = 0, \ldots, L-1 \\ 0 & \text{otherwise} \end{cases} \Rightarrow h(n) = \begin{cases} h_d(n) & n = 0, \ldots, L-1 \\ 0 & \text{otherwise} \end{cases}$$

Frequency response of the rectangular window: see (3.12)

Suppose, we want to design a linear-phase lowpass filter of length

L with the desired frequency response

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\omega(L-1)/2} & \text{for} \quad 0 \le |\omega| < \omega_c, \\ 0 & \text{otherwise}, \end{cases} \tag{4.60}$$

where ω_c is denoting the cut-off frequency.
Corresponding unit sample response:

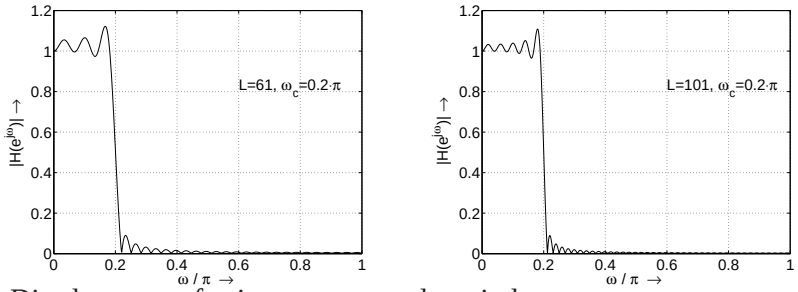
$$\begin{aligned} h_d(n) &= \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega(n-(L-1)/2)}\, d\omega. \\ &= \frac{\sin\left[\omega_c\left(n - \frac{L-1}{2}\right)\right]}{\pi\left(n - \frac{L-1}{2}\right)}, \quad n \neq \frac{L-1}{2} \end{aligned} \tag{4.61}$$

Multiplication with a rectangular window of length L leads to

$$h(n) = \frac{\sin\left[\omega_c\left(n - \frac{L-1}{2}\right)\right]}{\pi\left(n - \frac{L-1}{2}\right)}, \quad n \neq \frac{L-1}{2}, \quad n = 0, \ldots, L-1$$

For L odd: $\quad h\left(\frac{L-1}{2}\right) = \frac{\omega_c}{\pi}$

Example for $\omega_c = 0.2\pi$, $L = 61$ and $L = 101$:



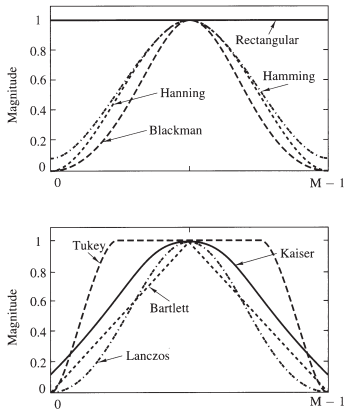
Disadvantage of using an rectangular window:

Large sidelobes lead to in undesirable ringing effects (overshoot at the boundary between pass- and stopband) in the frequency response of the resulting FIR filter

$\Rightarrow$ *Gibbs phenomenon*:

- Result of approximating a discontinuity in the frequency response with a finite number of filter coefficients and a mean square error criterion
- Eq. (4.58) can be interpreted as a Fourier series representation for $H_d(e^{j\omega})$ with the Fourier coefficients $h_d(n) \longrightarrow$ Gibbs phenomenon resulting from a Fourier series approximation
- *Squared integral error* $E = \int_{-\pi}^{\pi} (H_d(e^{j\omega}) - H(e^{j\omega}))^2 \, d\omega$ approaches zero with increasing length of $h(n)$. However, the maximum value of the error $|H_d(e^{j\omega}) - H(e^{j\omega})|$ approaches a constant value.

$\Rightarrow$ Use of **other appropriate window functions** with lower sidelobes in their frequency responses



($n \longrightarrow n$, $L \longrightarrow M$, I_0: Bessel function of the first kind of order zero)

Name of window	Time-domain sequence, $h(n)$, $0 \le n \le M-1$
Bartlett (triangular)	$1 - \dfrac{2\left\lvert n - \frac{M-1}{2}\right\rvert}{M-1}$
Blackman	$0.42 - 0.5\cos\dfrac{2\pi n}{M-1} + 0.08\cos\dfrac{4\pi n}{M-1}$
Hamming	$0.54 - 0.46\cos\dfrac{2\pi n}{M-1}$
Hanning	$\dfrac{1}{2}\left(1 - \cos\dfrac{2\pi n}{M-1}\right)$
Kaiser	$\dfrac{I_0\left[\alpha\sqrt{\left(\frac{M-1}{2}\right)^2 - \left(n - \frac{M-1}{2}\right)^2}\right]}{I_0\left[\alpha\left(\frac{M-1}{2}\right)\right]}$
Lanczos	$\left\{\dfrac{\sin\left[2\pi\left(n - \frac{M-1}{2}\right)\Big/(M-1)\right]}{2\pi\left(n - \frac{M-1}{2}\right)\Big/\left(\frac{M-1}{2}\right)}\right\}^L \quad L > 0$
	$1, \left\lvert n - \dfrac{M-1}{2}\right\rvert \le \alpha\dfrac{M-1}{2} \quad 0 < \alpha < 1$
Tukey	$\dfrac{1}{2}\left[1 + \cos\left(\dfrac{n - (1+a)(M-1)/2}{(1-\alpha)(M-1)/2}\pi\right)\right]$
	$\alpha(M-1)/2 \le \left\lvert n - \dfrac{M-1}{2}\right\rvert \le \dfrac{M-1}{2}$

(from [Proakis, Manolakis, 1996])

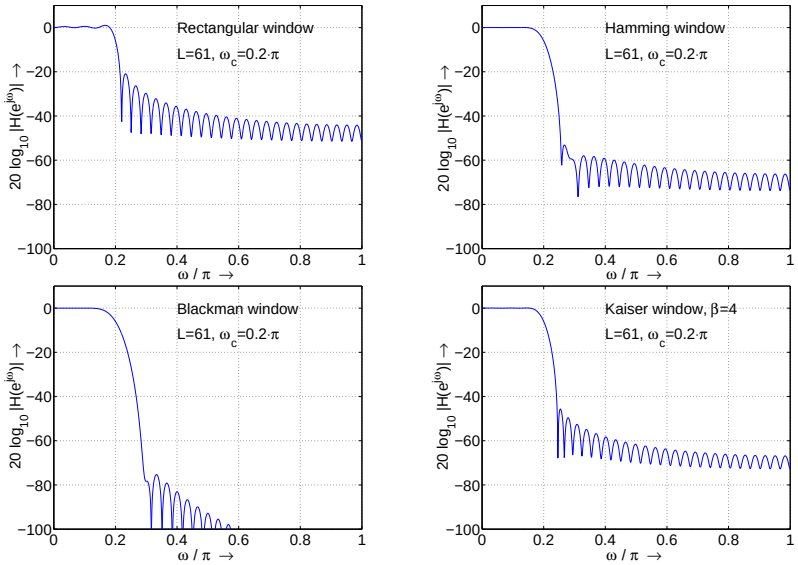
Frequency-domain characteristics of some window functions (taken from [Proakis, Manolakis, 1996]):

Type of window	Approximate transition width of main lobe	Peak sidelobe [dB]
Rectangular	$4\pi/L$	-13
Bartlett	$8\pi/L$	-27
Hanning	$8\pi/L$	-32
Hamming	$8\pi/L$	-43
Blackman	$12\pi/L$	-58

Parameter α in the Kaiser window allows to adjust the width of the main lobe, and thus also to adjust the compromise between overshoot reduction and increased transition bandwidth in the

resulting FIR filter.

Magnitude frequency response $20\log_{10}|H(e^{j\omega})|$ of the resulting linear-phase FIR filter, when different window functions are used to truncate the infinite-length impulse response from (4.61) (desired frequency response $H_d(e^{j\omega})$ in (4.60)):



MATLAB-command for windowed linear-phase FIR design: `fir1`

4.4.3 Frequency sampling design

Desired frequency response $H_d(e^{j\omega})$ is specified as a set of equally spaced frequencies:

$$\omega_k = \frac{2\pi}{L}(k+\alpha), \quad k = 0, 1, \ldots, L-1, \quad \alpha \in \left\{0, \frac{1}{2}\right\}. \tag{4.62}$$

Frequency response of an FIR filter (requirement):

$$H_d(e^{j\omega}) \stackrel{!}{=} \sum_{n=0}^{L-1} h(n) e^{-j\omega n}$$

With (4.62) we obtain for $k = 0, \ldots, L-1$:

$$H_d(e^{j\frac{2\pi}{L}(k+\alpha)}) = \sum_{n=0}^{L-1} h(n) e^{-j2\pi(k+\alpha)n/L}. \tag{4.63}$$

Multiplication of (4.63) with $e^{j2\pi k\ell/L}$, $\ell = 0, \ldots, L-1$ and summation over $k = 0, \ldots, L-1$ yields to

$$\begin{aligned}
\sum_{k=0}^{L-1} e^{j2\pi k\ell/L} H_d(e^{j\frac{2\pi}{L}(k+\alpha)}) &= \\
&= \sum_{k=0}^{L-1}\sum_{n=0}^{L-1} h(n) e^{-j2\pi(k+\alpha)n/L}\, e^{j2\pi k\ell/L} \\
&= \sum_{n=0}^{L-1} h(n) e^{-j2\pi\alpha n/L} \sum_{k=0}^{L-1} e^{-j2\pi(n-\ell)k/L} \\
&= L\, h(\ell)\, e^{-j2\pi\alpha\ell/L}.
\end{aligned}$$

Thus, the impulse response $h(n)$ can be obtained from the sampled desired frequency response as ($n = 0, \ldots, L-1$):

$$\boxed{h(n) = \frac{1}{L}\sum_{k=0}^{L-1} H_d(e^{j\frac{2\pi}{L}(k+\alpha)})\, e^{j2\pi(k+\alpha)n/L}} \tag{4.64}$$

Remarks:

- For $\alpha = 0$ (4.64) is identical to the IDFT $\Rightarrow$ fast evaluation with IFFT
- In general $H_d(e^{j\frac{2\pi}{L}(k+\alpha)})$ has to be specified in amplitude and phase for every k.
  Exception: Type 1/3 linear-phase filters, where $H_d(\cdot)$ can be real-valued with an additional shift of the obtained impulse response (see below).

If $h(n)$ is a real-valued sequence, the frequency response and thus also its sampled version have the *Hermitian symmetry*

$$H_d(e^{j\frac{2\pi}{L}(k+\alpha)}) = H_d^*(e^{j\frac{2\pi}{L}(L-k-\alpha)}).$$

$\Rightarrow$ The number of frequency specifications can be reduced, and (4.62) becomes

$$\omega_k = \frac{2\pi}{L}(k+\alpha), \quad \begin{cases} k = 0, 1, \ldots, \frac{L-1}{2} & L \text{ odd}, \\ k = 0, 1, \ldots, \frac{L}{2} - 1 & L \text{ even}, \\ \alpha \in \left\{0, \frac{1}{2}\right\}. \end{cases} \tag{4.65}$$

Linear-phase FIR filters:

- Symmetry in $h(n)$ can be additionally exploited such that (4.64) only has to be evaluated for $n = 0, \ldots, \frac{L}{2} - 1$ for L even, and $n = 0, \ldots, \frac{L-1}{2}$ for L odd, resp.
- Linear-phase property may be included by specifying real-valued frequency response samples $H_d(e^{j\frac{2\pi}{L}(k+\alpha)}) \rightarrow$
  Application of (4.64) leads to a *zero-phase response* which has to be shifted to the right by $\frac{L-1}{2}$ samples.

Example:

Determine the coefficients of a type 2 linear-phase filter with length $L = 32$, and the desired sampled frequency response

$$H_d(e^{j\frac{2\pi}{L}(k+\alpha)}) = e^{-j\frac{L-1}{2}\frac{2\pi}{L}(k+\alpha)} \cdot \begin{cases} 1 & k = 0, 1, \ldots, 5, \\ T_1 & k = 6, \\ 0 & k = 7, 8, \ldots, 15. \end{cases}$$

The parameter T_1 is responsible for the transition band behavior and is obtained via numerical optimization in order to reduce the magnitude of the sidelobes. The corresponding values are tabulated in the literature ([Rabiner, Gold, McGonegal, 1970], [Proakis, Manolakis, 1996]).

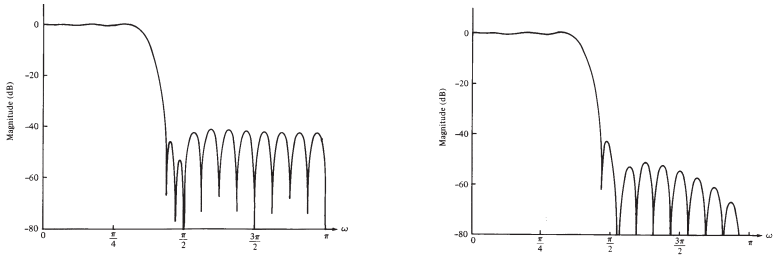
For $L = 32$ we obtain $T_1 = 0.3789795$ for $\alpha = 0$, and $T_1 = 0.3570496$ for $\alpha = 1/2$.

Coefficients of the impulse response $h(n)$:

M = 32	M = 32
ALPHA = 0.	ALPHA = 0.5
T1 = 0.3789795E+00	T1 = 0.3570496E+00
h(0) = -0.7141978E-02	h(0) = -0.4089120E-02
h(1) = -0.3070801E-02	h(1) = -0.9973779E-02
h(2) = 0.5891327E-02	h(2) = -0.7379891E-02
h(3) = 0.1349923E-01	h(3) = 0.5949799E-02
h(4) = 0.8087033E-02	h(4) = 0.1727056E-01
h(5) = -0.1107258E-01	h(5) = 0.7878412E-02
h(6) = -0.2420687E-01	h(6) = -0.1798590E-01
h(7) = -0.9446550E-02	h(7) = -0.2670584E-01
h(8) = 0.2544464E-01	h(8) = 0.3778549E-02
h(9) = 0.3985050E-01	h(9) = 0.4191022E-01
h(10) = 0.2753036E-02	h(10) = 0.2839344E-01
h(11) = -0.5913959E-01	h(11) = -0.4163144E-01
h(12) = -0.6841660E-01	h(12) = -0.8254962E-01
h(13) = 0.3175741E-01	h(13) = 0.2802212E-02
h(14) = 0.2080981E+00	h(14) = 0.2013655E+00
h(15) = 0.3471138E+00	h(15) = 0.3717532E+00

Magnitude frequency responses ($20 \log_{10} |H(e^{j\omega})|$):

$L = 32$, $\alpha = 0$: $L = 32$, $\alpha = 0.5$:



(from [Proakis, Manolakis, 1996])

MATLAB-command for the frequency sampling design of linear-phase FIR filters: `fir2`

4.4.4 Optimum equiripple design (Chebyshev approximation)

- Windowing design techniques (section 4.4.2) try to reduce the difference between the desired and the actual frequency response (error function) by choosing suitable windows
- How far can the maximum error be reduced?
  $\Rightarrow$ Theory of *Chebyshev approximation* answers this question and provides us with algorithms to find the coefficients of linear-phase FIR filters, where the maximum frequency response error is minimized
- *Chebyshev approximation*: Approximation that minimizes the maximum errors over a set of frequencies
- Resulting filters exhibit an equiripple behavior in their frequency responses $\Rightarrow$ *equiripple filters*

Linear-phase filters revisited

As we have shown in section 4.4.1, every linear-phase filter has a frequency response of the form

$$H(e^{j\omega}) = (j)^m \cdot A(\omega) \cdot e^{-j\omega(L-1)/2}, \quad m \in \{0, 1\}, \quad (4.66)$$

where $A(\omega)$ is a real-valued positive or negative function (amplitude frequency response) (cmp. (4.54), (4.55), (4.56), (4.57)).

It can be shown that for all types of linear-phase symmetry $A(\omega)$ can always be written as a weighted sum of cosines. For example for type 1 linear-phase filters we have

$$A(\omega) = \sum_{n=0}^{(L-1)/2} a_n \cos(n\omega) \quad (4.67)$$

$$\text{with } a_0 = h\left(\frac{L-1}{2}\right), \; a_n = 2\,h\left(-n + \frac{L-1}{2}\right). \quad (4.68)$$

Remaining linear-phase filters ($\omega \rightarrow 2\pi f$):

	Symmetry	
	Even $h(n) = h(N-1-n)$, $(m=0)$	Odd $h(n) = -h(N-1-n)$, $(m=1)$
Odd Length (N)	$A(f) = \sum_{k=0}^{(N-1)/2} a_k \cos 2\pi kf$ $a_0 = h((N-1)/2)$ $a_k = 2h(-k+(N-1)/2)$ $k = 1, \ldots, (N-1)/2$	$A(f) = \sin 2\pi f \sum_{k=0}^{(N-3)/2} c_k \cos 2\pi kf$ $c_0 - \frac{1}{2}c(2) = 2h((N-3)/2)$ $c((N-5)/2) = 4h(1)$ $c((N-3)/2) = 4h(0)$ $c(k-1) - c(k+1) = 2h(-k+(N-1)/2)$ $k = 2, \ldots, (N-5)/2$
Even Length (N)	$A(f) = \cos \pi f \sum_{k=0}^{(N-2)/2} b_k \cos 2\pi kf$ $b_0 + \frac{1}{2}b(1) = 2h((N-3)/2$ $b((N-3)/2) = 4h(0)$ $b(k-1) + b(k) = 4h(-k+(N-1)/2)$ $k = 2, \ldots, (N-3)/2$	$A(f) = \sin \pi f \sum_{k=0}^{(N-2)/2} d_k \cos 2\pi kf$ $d_0 - \frac{1}{2}d(1) = 2h((N-3)/2)$ $d((N-3)/2) = 4h(0)$ $d(k-1) - d(k) = 4h(-k+(N-1)/2)$ $k = 2, \ldots, (N-3)/2$

(from [Parks, Burrus: Digital Filter Design, 1987])

Problem definition

Acceptable frequency response for the resulting FIR filter:

- Linear phase,
- transition bandwidth $\Delta\omega$ between pass- and stopband,
- passband deviation $\pm\delta_1$ from unity,
- stopband deviation $\pm\delta_2$ from zero.

(Multiple bands are possible as well.)

We will restrict ourselves in the following to lowpass type 1 linear-phase filters.

Approximation problem: Given

- a compact subset $\mathcal{F}$ of $[0, \pi]$ in the frequency domain (consisting of pass- and stopband in the lowpass filter case),
- a desired real-valued frequency response $D(\omega)$, defined on $\mathcal{F}$,
- a positive weight function $W(\omega)$, defined on $\mathcal{F}$, and
- the form of $A(\omega)$, here (type-1 linear-phase) $A(\omega) = \sum_{n=0}^{r-1} a_n \cos(n\omega)$.

Goal: Minimization of the error

$$E_{\max} = \max_{\omega \in \mathcal{F}} W(\omega) \cdot |D(\omega) - A(\omega)| \qquad (4.69)$$

over a_n by the choice of $A(\omega)$.

Alternation theorem (without proof):
If $A(\omega)$ is a linear combination of r cosine functions,

$$A(\omega) = \sum_{n=0}^{r-1} a_n \cos(n\omega),$$

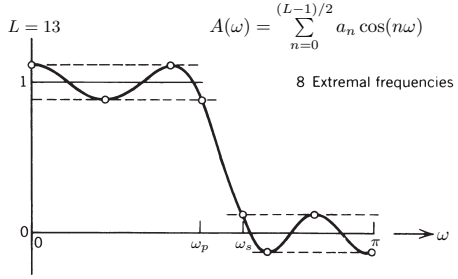
then a necessary and sufficient condition that $A(\omega)$ be the unique, best weighted Chebyshev approximation to a given continuous function $D(\omega)$ on $\mathcal{F}$ is that the weighted error function $E(\omega) = W(\omega) \cdot (D(\omega) - A(\omega))$ exhibit *at least* $r + 1$ extremal frequencies in $\mathcal{F}$. These frequencies are points for which $\omega_1 < \cdots < \omega_r < \omega_{r+1}$,

$$E(\omega_m) = -E(\omega_{m+1}), \quad m = 1, 2, \ldots, r,$$

and

$$|E(\omega_i)| = \max_{\omega \in \mathcal{F}} E(\omega), \quad i = 1, \ldots, r + 1.$$

- Consequences from the alternation theorem: Best Chebyshev approximation must have an equiripple error function $E(\omega)$, and is *unique*.
- Example: Amplitude frequency response of an optimum type 1 linear-phase filter with $L = 13 \rightarrow r = 7$



(from [Parks, Burrus: Digital Filter Design, 1987])

- If the r+1 extremal frequencies were known, we could use the frequency-sampling design from above to specify the desired

values $1 \pm \delta_1$ at the extremal frequencies in the passband, and $\pm\delta_2$ in the stopband, respectively.

How to find the set of extremal frequencies?

Remez exchange algorithm (Parks, McClellan, 1972)

- It can be shown that the error function

$$E(\omega) = D(\omega) - \sum_{n=0}^{r-1} a_n \cos(n\omega) \tag{4.70}$$

can always be forced to take on some values $\pm\delta$ for any given set of $r+1$ frequency points ω_i, $i = 1, \ldots, r+1$.
Simplification: Restriction to $W(\omega) = 1$, leading to $\delta_1 = \delta_2 = \delta$.
This can be written as a set of linear equations according to

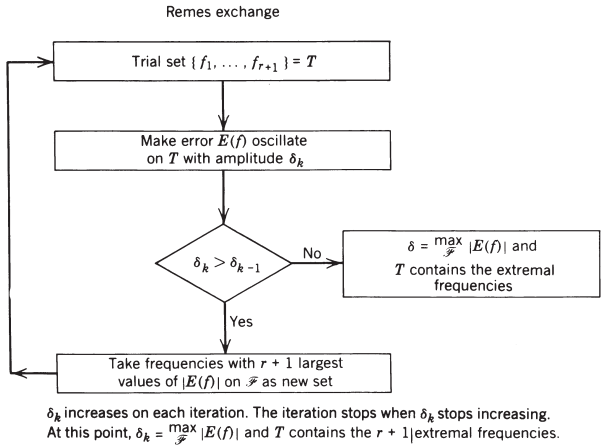
$$D(\omega_i) = \sum_{n=0}^{r-1} a_n \cos(n\omega_i) + (-1)^i \delta, \quad i = 1, \ldots, r+1, \tag{4.71}$$

from which we obtain a unique solution for the coefficients a_n, $n = 0, \ldots, r-1$ and the error magnitude δ.

- In the Remez exchange algorithm $\{\mathcal{F}\}$ is usually chosen as an equally spaced grid of frequency points with the number of frequency points being approximately $10 \cdot L$. The algorithm is initialized with a trial set of arbitrarily chosen frequencies $T = \{\omega_1, \omega_2, \ldots, \omega_{r+1}\}$.
- The Remez algorithm now consists of the following basic computations:
  1. Solve the linear equations in (4.71), yielding an error magnitude δ_k in the k-th iteration.

  2. Interpolate to find the frequency response on the entire grid of frequencies.
  3. Search over the entire grid of frequencies for a larger error magnitude than δ_k obtained in step 1.
  4. Stop, if no larger magnitude error has been found. Otherwise, take the $r+1$ frequencies, where the error attains its maximum magnitude as a new trial set of extremal frequencies and go to step 1.

Flowchart ($\omega \rightarrow f$):



(from [Parks, Burrus: Digital Filter Design, 1987])

Example:

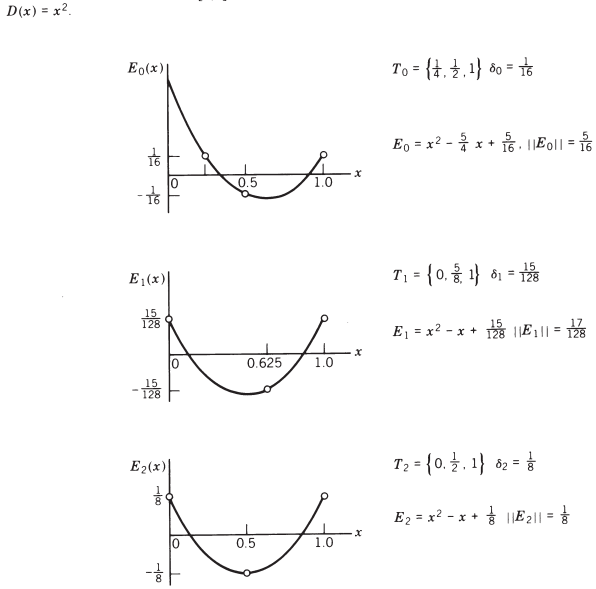
Choose the two coefficients d_0 and d_1 such that they minimize the Chebyshev error

$$\max_{x \in [0,1]} |x^2 - (d_0 + d_1 x_1)|$$

(approximation of a parabola by a straight line) $\rightarrow$ three extremal points $\rightarrow$

resulting linear equations to be solved:

$$x_i^2 = d_0 + d_1\, x_i + (-1)^i\, \delta, \quad i = 0, 1, 2 \tag{4.72}$$



(from [Parks, Burrus: Digital Filter Design, 1987])

1. Arbitrarily chosen trial set: $T_0 = [0.25, 0.5, 1.0]$
   Matrix version of the linear equations in (4.72):

$$\begin{bmatrix} 1 & 0.25 & 1 \\ 1 & 0.5 & -1 \\ 1 & 1.0 & 1 \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ \delta_0 \end{bmatrix} = \begin{bmatrix} 0.0625 \\ 0.25 \\ 1.0 \end{bmatrix}$$

   $\rightarrow \delta_0 = 0.0625$
2. Next trial set chosen as those three points, where the error

$$E(x) = x^2 - (d_0 + d_1 x)$$

   achieves its maximum magnitude: $\rightarrow T_1 = [0.0, 0.625, 1.0]$
   Linear equations to solve:

$$\begin{bmatrix} 1 & 0.0 & 1 \\ 1 & 0.625 & -1 \\ 1 & 1.0 & 1 \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ \delta_0 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.390625 \\ 1.0 \end{bmatrix}$$

   $\rightarrow \delta_1 = 0.1171875$
3. Next trial set: $T_2 = [0.0, 0.5, 1.0]$
   Linear equations to solve:

$$\begin{bmatrix} 1 & 0.0 & 1 \\ 1 & 0.5 & -1 \\ 1 & 1.0 & 1 \end{bmatrix} \begin{bmatrix} d_0 \\ d_1 \\ \delta_0 \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.25 \\ 1.0 \end{bmatrix}$$

   $\rightarrow \delta_1 = 0.125 \mathrel{\widehat{=}}$ maximum error $\rightarrow T_2$ is the extremal point set

After the extremal points ω_i are found, the coefficients a_n from Step 1 of the above algorithm are used to obtain the filter coefficients with (4.68).

MATLAB-command for optimum equiripple design: `remez`

Estimation of the filter length

Given the stop- / passband ripple δ_1, δ_2, and the transition bandwidth $\Delta\omega = \omega_s - \omega_p$ the necessary filter order N can be estimated as (Kaiser, 1974)

$$\boxed{N = \frac{-10 \log_{10}(\delta_1\, \delta_2) - 13}{2.324 \Delta\omega}.} \tag{4.73}$$

MATLAB-command for estimating the filter order: `remezord`

Design example

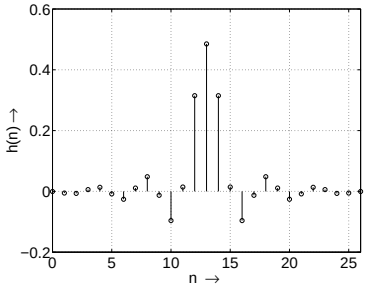
Design a linear-phase lowpass filter with the specifications

$$\delta_1 = 0.01, \quad \delta_2 = 0.001, \quad \omega_p = 0.4\pi, \quad \omega_s = 0.6\pi.$$

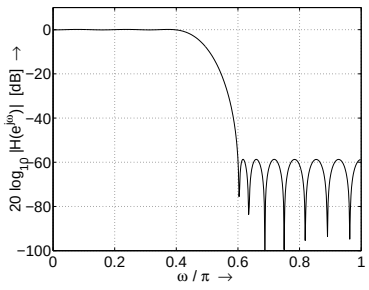
$\rightarrow$ weighting $\delta_2/\delta_1 = 10$ in the stopband.

Inserting these values into (4.73) leads to $N \approx 25.34$ and rounding up (to be on the save side) yields a filter length $L = N+1 = 27$.

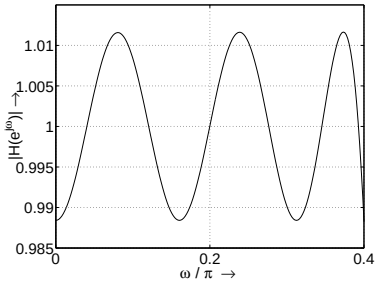
Impulse response



Magnitude frequency response



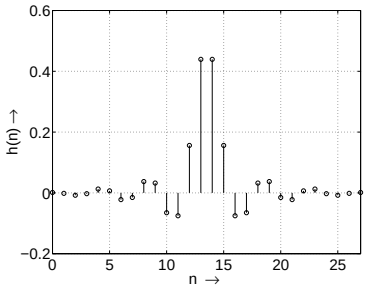
Passband detail



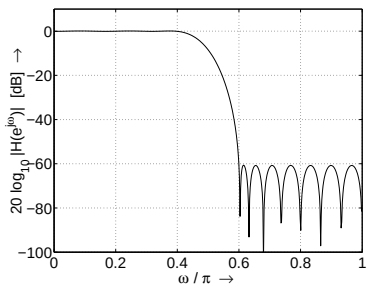
In the passband the specifications are not satisfied $\rightarrow$ Increasing

the filter-length by one, $L = 28$:

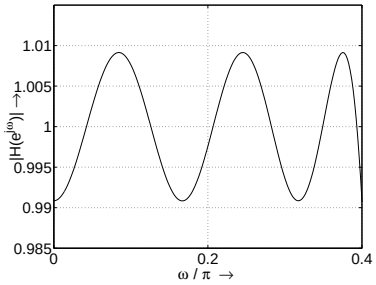
Impulse response



Magnitude frequency response



Passband detail



4.5 Design of IIR filters

- In the following only design algorithms are discussed which convert an analog into a digital filter, however, there are also numerous algorithms for directly designing an IIR filter in the z-domain (frequency sampling method, least-squares design).
- Why starting point analog filter? Analog filter design is a well developed field (lots of existing design catalogs). Problem can be defined in the z-domain, transformed into the s-domain and solved there, and finally transformed back into the z-domain.

- Analog filter: Transfer function

$$H_a(s) = \frac{N(s)}{D(s)} = \frac{\sum_{n=0}^{M} \beta_n \, s^n}{\sum_{n=0}^{N} \alpha_n \, s^n} \tag{4.74}$$

with the filter coefficients α_n, β_n, and the filter order N. Furthermore,

$$H_a(s) = \int_{-\infty}^{\infty} h(t) \, e^{-st} \, dt \tag{4.75}$$

(Laplace transform).

- Note that linear-phase designs are not possible for causal and stable IIR filters, since the condition

$$H(z) = \pm z^{-N} H(z^{-1})$$

has to be satisfied (compare (4.53)) $\longrightarrow$ mirror-image pole outside the unit-circle for every pole inside the unit-circle $\longrightarrow$ unstable filter.

4.5.1 Filter design by impulse invariance

Goal: Design an IIR filter with an impulse response $h(n)$ being the sampled version of the impulse response $h_a(t)$ of the analog filter:

$$h(n) = h_a(nT), \quad n = 0, 1, 2, \ldots,$$

where T is the sampling interval.

Frequency response (ideal sampling assumed, compare (2.4)):

$$H(j\Omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} H_a\left(j\Omega - j\frac{2\pi n}{T}\right) \tag{4.76}$$

- T should be selected sufficiently small to avoid aliasing.
- Method is not suitable to design highpass filters due to the large amount of possible aliasing.

Suppose that the poles of the analog filter are distinct. Then the partial-fraction expansion of $H_a(s)$ writes

$$H_a(s) = \sum_{i=1}^{N} \frac{A_i}{s - s_{\infty i}}, \tag{4.77}$$

the A_i are the coefficients of the partial-fraction expansion, and the $s_{\infty i}$ denote the poles of the analog filter. Inverse Laplace transform of (4.77) yields

$$h_a(t) = \sum_{i=1}^{N} A_i \, e^{s_{\infty i} t}, \quad t \geq 0.$$

Periodical sampling:

$$h(n) = h_a(nT) = \sum_{i=1}^{N} A_i \, e^{s_{\infty i} nT}.$$

Transfer function $H(z) \bullet\!\!-\!\!\circ\, h(n)$:

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-n} = \sum_{n=0}^{\infty} \left(\sum_{i=1}^{N} A_i \, e^{s_{\infty i} nT} \right) z^{-n}.$$

We then have

$$H(z) = \sum_{i=1}^{N} A_i \sum_{n=0}^{\infty} \left(e^{s_{\infty i} T} z^{-1} \right)^n = \sum_{i=1}^{N} \frac{A_i}{1 - e^{s_{\infty i} T} z^{-1}}.$$

Thus, given an analog filter $H_a(s)$ with poles $s_{\infty i}$, the transfer function of the corresponding digital filter using the impulse invariant transform is

$$\boxed{H(z) = \sum_{i=1}^{N} \frac{A_i}{1 - e^{s_{\infty i} T} z^{-1}}} \qquad (4.78)$$

with poles at $z_{\infty i} = e^{s_{\infty i} T}$, $i = 1, \ldots, N$. Note: (4.78) holds only for *distinct poles*, generalization to multiple-order poles possible.

Example:

Convert the analog filter with transfer function

$$H_a(s) = \frac{s + 0.1}{(s + 0.1)^2 + 9}$$

into a digital filter using the impulse invariant method.

Poles of $H_a(s)$: $s_{\infty 0,1} = -0.1 \pm j3$

Partial-fraction expansion:

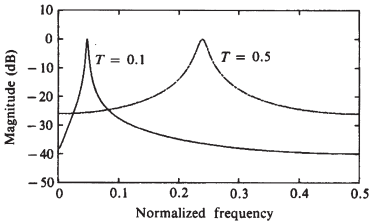
$$H_a(s) = \frac{0.5}{s + 0.1 - j3} + \frac{0.5}{s + 0.1 + j3}$$

From (4.78) we then finally have

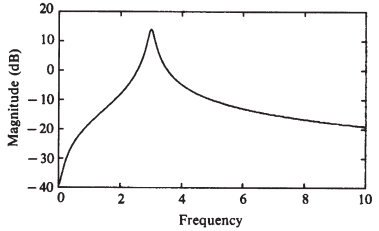
$$\begin{aligned} H(z) &= \frac{0.5}{1 - e^{-(0.1 - j3)T} z^{-1}} + \frac{0.5}{1 - e^{-(0.1 + j3)T} z^{-1}} \\ &= \frac{1 - (e^{-0.1T} \cos(3T))\, z^{-1}}{1 - (2\, e^{-0.1T} \cos(3T))\, z^{-1} + e^{-0.2T} z^{-2}}. \end{aligned}$$

Magnitude frequency responses:

Digital filter: $20 \log_{10} |H(e^{j\omega})|$



Analog filter: $20 \log_{10} |H_a(j\Omega)|$



(from [Proakis, Manolakis, 1996])

4.5.2 Bilinear transform

Algebraic transformation between the variables s and z, mapping of the entire $j\Omega$-axis of the s-plane to one revolution of the unit circle in the z-plane.

Definition:

$$\boxed{s = \frac{2}{T} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right),} \qquad (4.79)$$

T denoting the sampling interval.

The transfer function of the corresponding digital filter can be obtained from the transfer function of the analog filter $H_a(s)$

according to

$$H(z) := H_a\left[\frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)\right] = H_a(s).$$

Properties

- Solving (4.79) for z yields

$$z = \frac{1+(T/2)s}{1-(T/2)s}, \tag{4.80}$$

  and by substituting $s = \sigma + j\Omega$ we obtain

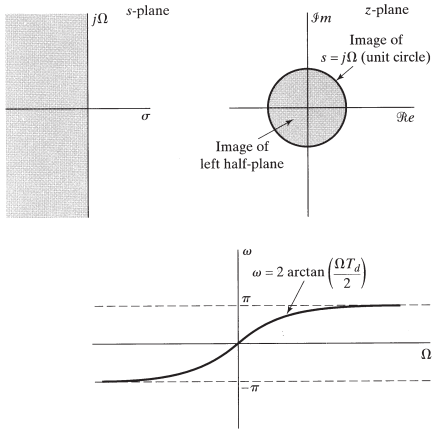
$$z = \frac{1+\sigma T/2 + j\Omega T/2}{1-\sigma T/2 - j\Omega T/2}.$$

  $\sigma < 0 \rightarrow |z| < 1, \sigma > 0 \rightarrow |z| > 1$ for all Ω
  $\Rightarrow$ causal, stable continuous-time filters map into causal stable discrete-time filters
- By inserting $s = j\Omega$ into (4.80), it can be seen that $|z| = 1$ for all values of s on the $j\Omega$-axis $\Rightarrow$ $j\Omega$-axis maps onto the unit circle.
- Relationship between ω and Ω: From (4.79) we obtain with $s = j\Omega$ and $z = e^{j\omega}$

$$\begin{aligned} j\Omega &= \frac{2}{T}\left(\frac{1-e^{-j\omega}}{1+e^{-j\omega}}\right), \\ &= \frac{2}{T}\left(\frac{j\sin(\omega/2)}{\cos(\omega/2)}\right) = \frac{2j}{T}\tan(\omega/2) \end{aligned}$$

$\Rightarrow$ Nonlinear mapping between ω and Ω (warping of the frequency axis) according to

$$\Omega = \frac{2}{T}\tan(\omega/2), \quad \omega = 2\text{arctan}(\Omega T/2). \tag{4.81}$$



(from [Oppenheim, Schafer, 1999])

Remarks:

- Design of a digital filter often begins with frequency specifications in the digital domain, which are converted to the analog domain by (4.81). The analog filter is then designed considering these specifications (i.e. using the classical approaches from the following section) and converted back into the digital domain using the bilinear transform.
- When using this procedure, the parameter T cancels out and can thus be set to an arbitrary value ($T = 1$).
- Example:
  Design a digital single-pole lowpass filter with a -3 dB frequency (cutoff frequency) of $\omega_c = 0.2\pi$, using the bilinear transform applied to the

analog filter with the transfer function

$$H_a(s) = \frac{\Omega_c}{s + \Omega_c},$$

Ω_c denoting the analog cutoff frequency.
Ω_c is obtained from ω_c using (4.81)

$$\Omega_c = \frac{2}{T}\tan(\omega_c/2) = \frac{0.65}{T}.$$

The analog filter now has the transfer function

$$H_a(s) = \frac{0.65/T}{s + 0.65/T},$$

which is transformed back into the digital domain using the bilinear transform in (4.79), leading to the transfer function of our desired digital filter:

$$H(z) = \frac{0.245\,(1 + z^{-1})}{1 - 0.509\,z^{-1}}.$$

Note that the parameter T has been divided out. The frequency response is

$$H(e^{j\omega}) = \frac{0.245\,(1 + e^{-j\omega})}{1 - 0.509\,e^{-j\omega}},$$

especially we have $H(e^{j0}) = 1$, and $|H(e^{j0.2\pi})| = 0.707$, which is the desired response.

4.5.3 Characteristics of commonly used analog filters

- Design of a digital filter can be reduced to design an appropriate analog filter and then performing the conversion from $H(s)$ to $H(z)$.
- In the following we briefly discuss the characteristics of commonly used analog (lowpass) filters.

Butterworth filters

Lowpass Butterworth filters are allpole-filters characterized by the squared magnitude frequency response

$$|H(\Omega)|^2 = \frac{1}{1 + (\Omega/\Omega_c)^{2N}}, \tag{4.82}$$

N is the order of the filter, Ω_c is the $-3\,$dB frequency (cutoff frequency).
Since $H(s) \cdot H(-s)_{|s=j\Omega} = |H(j\Omega)|^2$ we have from (4.82) by analytic continuation into the whole s-plane

$$H(s) \cdot H(-s) = \frac{1}{1 + (-s^2/\Omega_c^2)^N}.$$

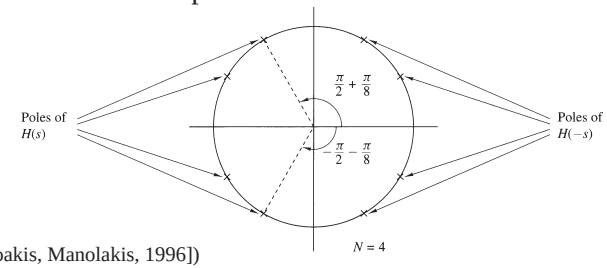
$\rightarrow$ Poles of $H(s)\,H(-s)$:

$$\frac{-s^2}{\Omega_c^2} = (-1)^{1/N} = e^{j(2n+1)\pi/N}$$

$$\rightarrow \quad s_{\infty,n} = \Omega_c\, e^{j\pi/2}\, e^{j(2n+1)\pi/(2N)}, \quad n = 0, \ldots, 2N-1 \tag{4.83}$$

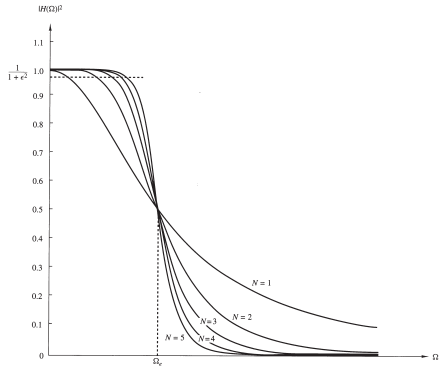
- From (4.83): The $2N$ poles of $H(s)\,H(-s)$ occur on a circle of radius Ω_c at equally spaced points in the s-plane.
- The N poles for $n = 0, \ldots, N-1$ in (4.83) are located in the left half of the s-plane and belong to $H(s)$.
- The N remaining poles lie in the right half of the s-plane and belong to $H(-s)$ (stability!).
- Furthermore, a Butterworth filter has N zeros at $\Omega \rightarrow \infty$.

Pole locations in the s-plane:



(from [Proakis, Manolakis, 1996])

Frequency responses ($\omega \rightarrow \Omega$, $|H(\Omega_p)|^2 = 1/(1+\epsilon^2)$)



(from [Proakis, Manolakis, 1996])

Estimation of the required filter order N:

At the stopband edge frequency Ω_s (4.82) can be written as

$$\frac{1}{1+(\Omega_s/\Omega_c)^{2N}} = \delta_2^2,$$

which leads to

$$N = \frac{\log((1/\delta_2^2)-1)}{2\log(\Omega_s/\Omega_c)}. \qquad (4.84)$$

MATLAB commands: `buttord` for order estimation, `butter` for the design of the corresponding digital filter obtained via bilinear transform.

Example:

Determine the order and the poles of a lowpass Butterworth filter that has a -3 dB bandwidth of 500 Hz and an attenuation of 40 dB at 1000 Hz.

- -3 dB frequency $\Omega_c = 2\,\pi \cdot f_c = 1000\,\pi$,
- stopband frequency $\Omega_s = 2\,\pi \cdot f_s = 2000\,\pi$,
- attenuation of 40 dB $\rightarrow \delta_2 = 0.01$.

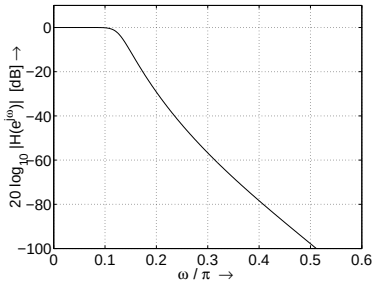
From (4.84) we then obtain

$$N = \frac{\log_{10}(10^4-1)}{2\log_{10} 2} = 6.64$$

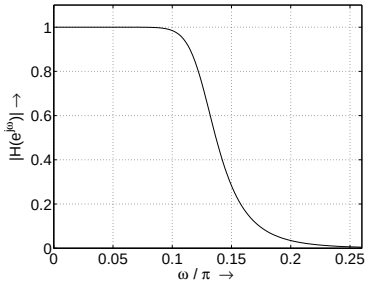
In order to be on the safe side we choose $N = 7$.

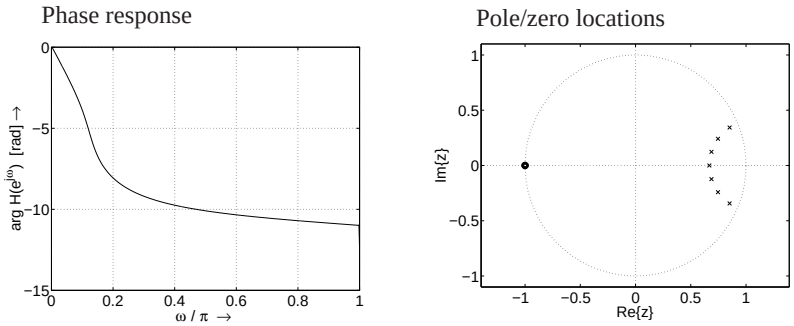
Properties of the resulting digital filter designed with `butter` for $N = 7$, $f_{\text{samp}} = 8000$ Hz, and the above parameters (continuous-time filter transformed by bilinear transform into the discrete-time domain):

Magnitude frequency response



Transition band





Chebyshev filters

Two types of Chebyshev filters:

- Type 1 filters are all-pole filters with equiripple behavior in the passband and monotonic characteristic (similar to a Butterworth filter) in the stopband.
- Type 2 filters have poles and zeros (for finite s), and equiripple behavoir in the stopband, but a monotonic characteristic in the passband.

Type 1 Chebyshev filter:

Squared magnitude frequency response:

$$|H(\Omega)|^2 = \frac{1}{1+\epsilon^2\, T_N^2(\Omega/\Omega_p)}, \tag{4.85}$$

where ϵ is a parameter related to the passband ripple, and $T_N(x)$ is the N-th order Chebyshev polynomial defined as

$$T_N(x) = \begin{cases} \cos(N\cos^{-1}(x)) & \text{for } |x| \le 1, \\ \cosh(N\cosh^{-1}(x)) & \text{for } |x| > 1. \end{cases} \tag{4.86}$$

The Chebyshev polynomials can be obtained by the recursive equation

$$T_{N+1}(x) = 2\,x\,T_N(x) - T_{N-1}(x), \quad N = 1, 2, \ldots$$

Examples:

- $T_0(x) = 1, T_1(x) = \cos(\cos^{-1}(x)) = x$
- $T_2(x) = \cos(2\cos^{-1}(x)) = 2\cos^2(\cos^{-1}x) - 1$
  $= 2\,x^2 - 1$
- $T_3(x) = \cos(3\cos^{-1}(x))$
  $= 4\cos^3(\cos^{-1}x) - 3\cos(\cos^{-1}(x)) = 4\,x^3 - 3\,x$
- ⋮

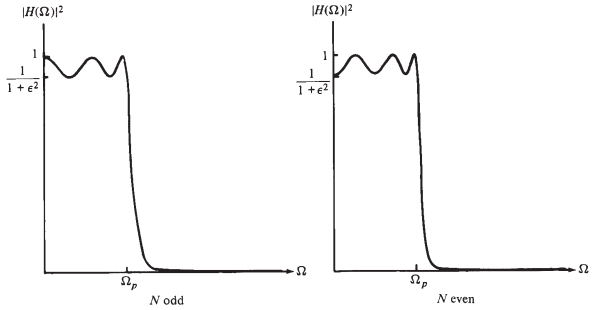
$\Rightarrow T_N(x)$ represents a polynom of degree N in x.
$\Rightarrow$ Chebyshev behavior (minimizing the maximal error) in the passband (or in the stopband for type 2 filters).

The filter parameter ϵ in (4.85) is related to the passband ripple:
For N odd, $T_N(0) = 0 \rightarrow |H(0)|^2 = 1$,
for N even, $T_N(0) = 1 \rightarrow |H(0)|^2 = \frac{1}{1+\epsilon^2}$
At the passband edge frequency $\Omega = \Omega_p$ we have $T_N(1) = 1$, such that

$$\frac{1}{\sqrt{1+\epsilon^2}} = 1-\delta_1 \quad \Longleftrightarrow \quad \epsilon = \sqrt{\frac{1}{(1-\delta_1)^2} - 1}, \tag{4.87}$$

which establishes a relation between the passband ripple δ_1 and the parameter ϵ.

Typical squared magnitude frequency response for a Chebyshev type 1 filter:



(from [Proakis, Manolakis, 1996])

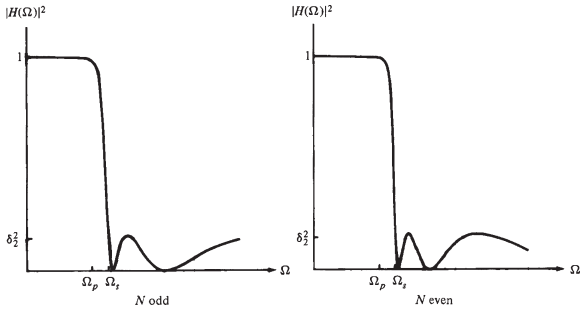
Type 2 Chebyshev filter:

Squared magnitude frequency response:

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 \left[T_N^2(\Omega_s/\Omega_p)/T_N^2(\Omega_s/\Omega)\right]} \qquad (4.88)$$

$\Rightarrow$ contains zeros at $s < \infty$ and poles

Typical squared magnitude frequency response for a Chebyshev type 2 filter:



(from [Proakis, Manolakis, 1996])

Estimation of the filter order:

Chebyshev filter only depend on the parameters N, ϵ, δ_2, and the ratio Ω_s/Ω_p. Using these values, it can be shown that the required order can be estimated as

$$N = \frac{\log\left[\left(\sqrt{1-\delta_2^2} + \sqrt{1-\delta_2^2(1+\epsilon^2)}\right)/(\epsilon\,\delta_2)\right]}{\log\left[\Omega_s/\Omega_p + \sqrt{(\Omega_s/\Omega_p)^2 - 1}\right]}. \qquad (4.89)$$

MATLAB commands:

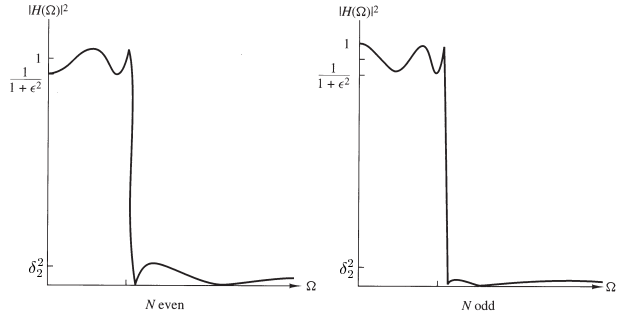
- Order estimation: `cheb1ord` for type 1, `cheb2ord` for type 2.
- Design of the corresponding digital filter, obtained from the analog version by bilinear transform: `cheby1` for type 1, `cheby2` for type 2.

Elliptic (Cauer) filters

- Elliptic filters have equiripple (Chebyshev) behavior in both pass- and stopband.
- Transfer function contains both poles and zeros, where the zeros are located on the $j\Omega$-axis.
- Squared magnitude frequency response

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2\, U_N(\Omega/\Omega_p)}, \qquad (4.90)$$

  where $U_N(x)$ denotes the Jacobian elliptic function of order N, and the parameter ϵ controls the passband ripple.
- Characteristic squared magnitude frequency response for a elliptic filter:



(from [Proakis, Manolakis, 1996])

- Filter design is optimal in pass- and stopband in the equiripple sense: However, other types of filters may be preferred due to their better phase response characteristics (i.e. approximately linear-phase), for example the Butterworth filter.

Estimation of the filter order:

Required order to achieve the specifications with the parameters δ_1, δ_2 and Ω_p/Ω_s, ($1-\delta_1 = 1/\sqrt{1+\epsilon^2}$, $1-\delta_2 = 1/\sqrt{1+\delta^2}$):

$$N = \frac{K(\Omega_p/\Omega_s)\, K(\sqrt{1-(\epsilon/\delta)^2})}{K(\epsilon/\delta)\, K(\sqrt{1-(\Omega_p/\Omega_s)^2})} \qquad (4.91)$$

where $K(x)$ denotes the complete elliptic integral of the first kind (tabulated)

$$K(x) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1-x^2\sin^2\theta}}.$$

MATLAB commands: `ellipord` for order estimation, `ellip` for the design of the corresponding digital filter obtained via bilinear transform.

5. Multirate Digital Signal Processing

- In many practical signal processing applications different sampling rates are present, corresponding to different bandwidths of the individual signals $\rightarrow$ *multirate systems*.

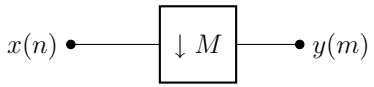
- Often, a signal has to be converted from one rate to another. This process is called *sampling rate conversion*.
  - Sampling rate conversion can be carried out by analog means, that is D/A conversion followed by A/D conversion using a different sampling rate $\rightarrow$ D/A converter introduces signal distortion, and the A/D converter leads to quantization effects.
  - Sampling rate conversion can also be carried out completely in the digital domain: Less signal distortions, more elegant and efficient approach.

- $\Rightarrow$ Topic of this chapter is multirate signal processing and sampling rate conversion in the digital domain.

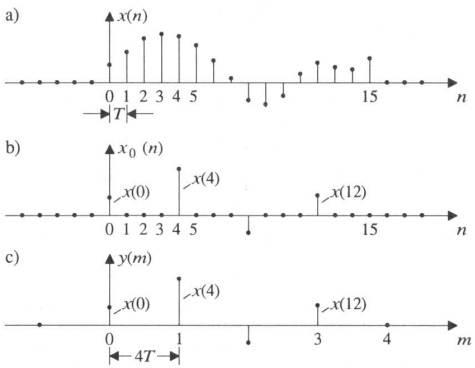
5.1 Basic multirate operations

5.1.1 Sampling rate reduction

Reduction of the sampling rate (*downsampling*) by factor M: Only every M-th value of the signal $x(n)$ is used for further processing, i.e. $y(m) = x(m \cdot M)$



Example: Sampling rate reduction by factor 4



(from [Fliege: Multiraten-Signalverarbeitung, 1993])

In the z-domain we have

$$X_0(z) = X_0(z^M) = \sum_{m=-\infty}^{\infty} x(mM) z^{-mM}$$

$$= \sum_{m=-\infty}^{\infty} y(m)(z^M)^{-m}$$

$$= Y(z^M) = Y(z') \bullet\!\!-\!\!\circ\, y(m) \qquad (5.1)$$

Frequency response after downsampling

Starting point: orthogonality of the complex exponential sequence

$$\frac{1}{M}\sum_{k=0}^{M-1} e^{j2\pi km/M} = \begin{cases} 1 & \text{for } m = \lambda M, \quad \lambda \in \mathbb{Z}, \\ 0 & \text{otherwise.} \end{cases} \qquad (5.2)$$

With $x_0(mM) = x(mM)$ it follows

$$x_0(m) = x(m)\,\frac{1}{M}\sum_{k=0}^{M-1} W_M^{-km}, \quad W_M := e^{-j2\pi/M} \qquad (5.3)$$

With (5.3) the z-transform $X_0(z)$ can be obtained as

$$X_0(z) = \sum_{m=-\infty}^{\infty} x_0(m) z^{-m}$$

$$= \frac{1}{M}\sum_{k=0}^{M-1}\sum_{m=-\infty}^{\infty} x(m)(W_M^k z)^{-m}. \qquad (5.4)$$

By replacing $Y(z^M) = X_0(z)$ in (5.4) we have for the z-transform of the downsampled sequence $y(m)$

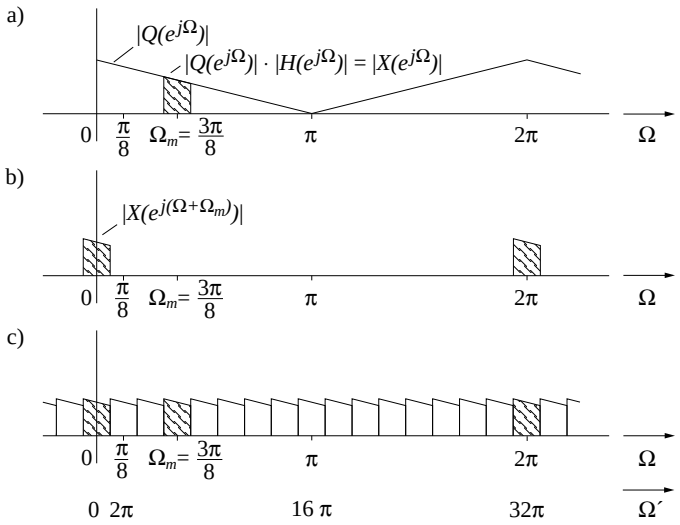
$$\boxed{Y(z^M) = \frac{1}{M}\sum_{k=0}^{M-1} X(zW_M^k).} \qquad (5.5)$$

With $z = e^{j\omega}$ and $\omega' = \omega M$ the corresponding frequency response can be derived from (5.5):

$$\boxed{Y(e^{j\omega'}) = \frac{1}{M}\sum_{k=0}^{M-1} X(e^{j(\omega'-k2\pi)/M})} \qquad (5.6)$$

$\Rightarrow$ Downsampling by factor M leads to a periodic repetition of the spectrum $X(e^{j\omega})$ at intervals of $2\pi/M$ (related to the high sampling frequency).

Example: Sampling rate reduction of a bandpass signal by $M = 16$ ($\Omega \to \omega$)



(from [Vary, Heute, Hess: Digitale Sprachsignalverarbeitung, 1998])

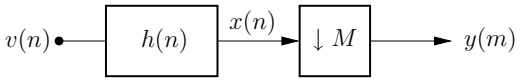
(a) Bandpass spectrum $X(e^{j\omega})$ is obtained by filtering.
(b) Shift to the baseband, followed by decimation with $M = 16$.
(c) Magnitude frequency response $|X(e^{j\omega'})|$ at the lower sampling rate.

Remark: Shifted versions of $X(e^{j\omega})$ are weighted with the factor $1/M$ according to (5.6).

Decimation and aliasing

If the sampling theorem is violated in the lower clock rate, we obtain *spectral overlapping* between the repeated spectra $\Rightarrow$ *Aliasing*

How to avoid aliasing? Bandlimitation of the input signal $v(n)$ prior to the sampling rate reduction with an *antialiasing filter* $h(n)$ (lowpass filter).



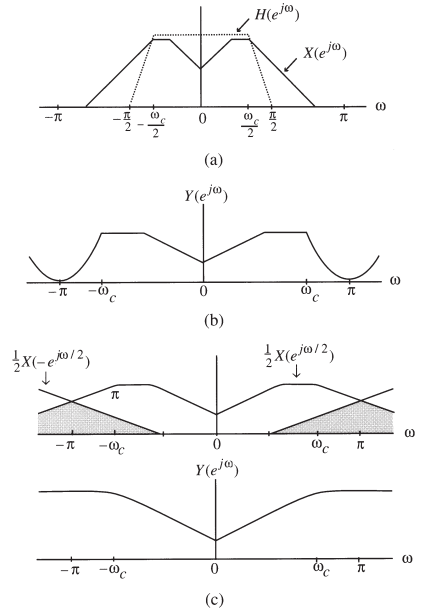
$\Rightarrow$ Antialiasing filtering followed by downsampling is often called *decimation*.

Specification for the desired magnitude frequency response of the lowpass antialiasing (or decimation) filter:

$$|H_d(e^{j\omega})| = \begin{cases} 1 & \text{for} \quad |\omega| \le \omega_c/M, \\ 0 & \text{for} \quad \pi/M \le |\omega| \le \pi, \end{cases} \tag{5.7}$$

where $\omega_c < \pi$ denotes the highest frequency that needs to be preserved in the *decimated* signal.

Downsampling in the frequency domain, illustration for $M = 2$: (a) input and filter spectra, (b) output of the decimator, (c) no filtering, only downsampling ($V \to X$):

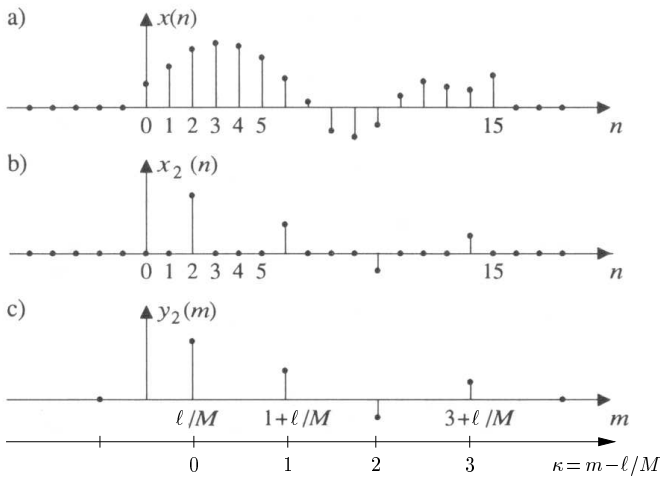


(from [Mitra, 2000])

More general approach: Sampling rate reduction with phase offset

Up to now we have always used $y(0) = x(0)$, now we introduce an additional phase offset ℓ into the decimation process.

Example for $\ell = 2$



(from [Fliege: Multiraten-Signalverarbeitung, 1993])

Note that $y_2(m)$ in (c) is a *formal* description for the output signal of the downsampler with non-integer sample indices. The real output signal $y_2(\kappa)$ is obtained by assuming integer sample locations.

Derivation of the Fourier transform of the output signal $y(m)$:

Orthogonality relation of the complex exponential sequence:

$$\frac{1}{M}\sum_{k=0}^{M-1} e^{j2\pi k(m-\ell)/M} = \begin{cases} 1 & \text{for } m = \lambda M + \ell, \quad \lambda \in \mathbb{Z}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

Using (5.8) we have

$$x_\ell(m) = x(m)\frac{1}{M}\sum_{k=0}^{M-1} W_M^{-k(m-\ell)}, \tag{5.9}$$

and transforming (5.9) into the z-domain yields

$$\begin{aligned} X_\ell(z) &= \frac{1}{M}\sum_{k=0}^{M-1}\sum_{m=-\infty}^{\infty} x(m)(W_M^k z)^{-m} W_M^{k\ell} \\ &= \frac{1}{M}\sum_{k=0}^{M-1} X(zW_M^k)\, W_M^{k\ell}. \end{aligned} \tag{5.10}$$

The frequency response can be obtained from (5.10) by substituting $z = e^{j\omega}$ and $\omega' = M\omega$ as

$$Y_\ell(e^{j\omega'}) = \frac{1}{M}\sum_{k=0}^{M-1} X(e^{j(\omega'-2\pi k)/M})\, W_M^{k\ell}, \tag{5.11}$$

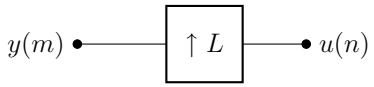
$$Y_\ell(e^{jM\omega}) = \frac{1}{M}\sum_{k=0}^{M-1} X(e^{j\omega - j2\pi k/M})\, W_M^{k\ell}. \tag{5.12}$$

$\Rightarrow$ We can see that each repeated spectrum is weighted with a complex exponential (rotation) factor.

5.1.2 Sampling rate increase

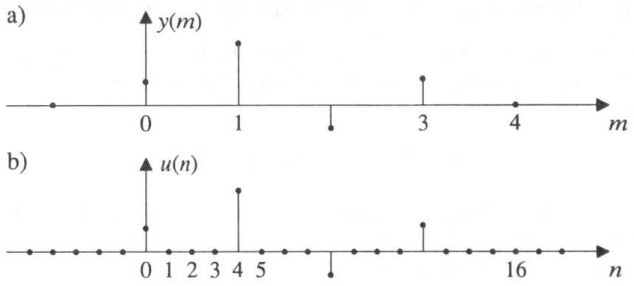
Increase of the sampling rate by factor L (*upsampling*): Insertion of $L-1$ zero samples between all samples of $y(m)$

$$u(n) = \begin{cases} y(n/L) & \text{for } n = \lambda L, \quad \lambda \in \mathbb{Z}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.13}$$



Notation: Since the upsampling factor is named with L in conformance with the majority of the technical literature in the following we will denote the *length of an FIR filter* with L_F.

Example: Sampling rate increase by factor 4



(from [Fliege: Multiraten-Signalverarbeitung, 1993])

In the z-domain the input/output relation is

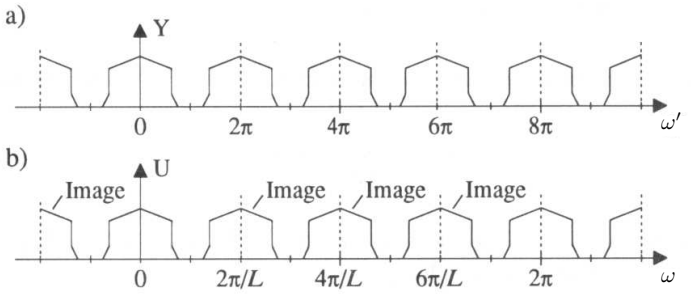
$$\boxed{U(z) = Y(z^L).} \tag{5.14}$$

Frequency response after upsampling

From (5.14) we obtain with $z = e^{j\omega}$

$$\boxed{U(e^{j\omega}) = Y(e^{jL\omega}).} \tag{5.15}$$

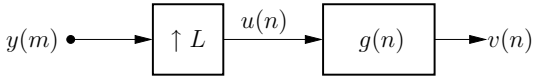
$\Rightarrow$ The frequency response of $y(m)$ does not change by upsampling, however the frequency axis is scaled differently. The new sampling frequency is now (in terms of ω' for the lower sampling rate) equal to $L \cdot 2\pi$.



(from [Fliege: Multiraten-Signalverarbeitung, 1993])

Interpolation

The inserted zero values are interpolated with suitable values, which corresponds to the suppression of the $L-1$ *imaging spectra* in the frequency domain by a suitable lowpass interpolation filter.



$g(n)$: Interpolation or *antiimaging* lowpass filter

Specifications for the interpolation filter:

Suppose $y(m)$ is obtained by sampling a bandlimited continuous-time signal $y_a(t)$ at the Nyquist rate (such that the sampling theorem is just satisfied). The Fourier transform $Y(e^{j\omega})$ can thus be written with (2.4) and $\Omega = \omega/T_0$ as

$$Y(e^{j\omega}) = \frac{1}{T_0} \sum_{k=-\infty}^{\infty} Y_a\left(\frac{j(\omega - 2\pi k)}{T_0}\right),$$

where T_0 denotes the sampling period. If we instead sample $y_a(t)$ at a much higher rate $T = T_0/L$ we have

$$V(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} Y_a\left(\frac{j(\omega - 2\pi k)}{T}\right), \qquad (5.16)$$

$$= \frac{L}{T_0}\sum_{k=-\infty}^{\infty} Y_a\left(\frac{j(\omega - 2\pi k)}{(T_0/L)}\right).$$

On the other hand by upsampling of $y(m)$ with factor L we obtain the Fourier transform of the upsampled sequence $u(n)$ analog to (5.15) as

$$U(e^{j\omega}) = Y(e^{j\omega L}).$$

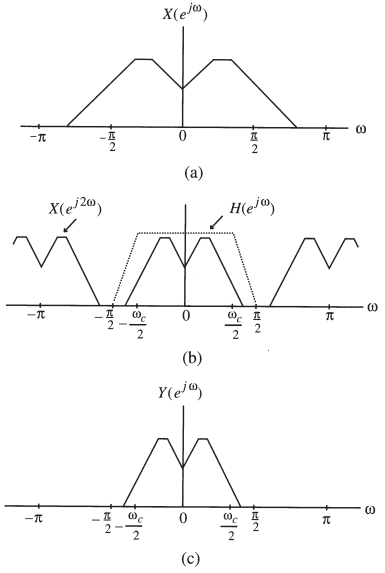
$\Rightarrow$ If $u(n)$ is passed through an ideal lowpass filter with cutoff frequency at π/L and a gain of L, the output of the filter will be precisely $v(n) = \mathcal{F}^{-1}\{V(e^{j\omega})\}$ in (5.16).

Therefore, we can now state our specifications for the lowpass interpolation filter:

$$|G_d(e^{j\omega})| = \begin{cases} L & \text{for} \quad |\omega| \le \omega_c/L, \\ 0 & \text{for} \quad \pi/L \le |\omega| \le \pi, \end{cases} \qquad (5.17)$$

where ω_c denotes the highest frequency that needs to be preserved in the interpolated signal (related to the lower sampling frequency).

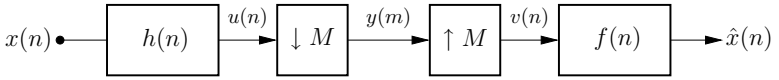
Upsampling in the frequency domain, illustration for $L = 2$: (a) input spectrum, (b) output of the upsampler, (c) output after interpolation with the filter $h(n)$:



(from [Mitra, 2000])

5.1.3 Example: Decimation and interpolation

Consider the following structure:



Input-output relation?

Relation between $Y(z)$ and $U(z)$ (see (5.5)), where z is replaced

by $z^{1/M}$:

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} U(z^{1/M} W_M^k),$$

which by using $U(z) = H(z)X(z)$ leads to

$$Y(z) = \frac{1}{M} \sum_{k=0}^{M-1} H(z^{1/M} W_M^k)\, X(z^{1/M} W_M^k). \qquad (5.18)$$

With $V(z) = Y(z^M)$ it follows

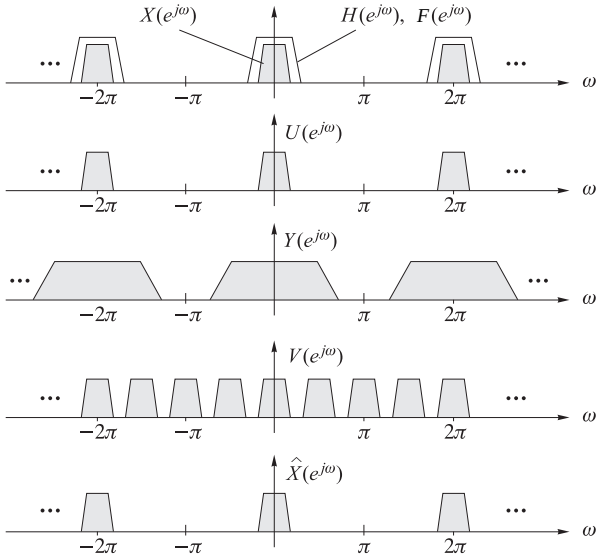
$$V(z) = \frac{1}{M} \sum_{k=0}^{M-1} H(z W_M^k)\, X(z W_M^k), \qquad (5.19)$$

and we finally have

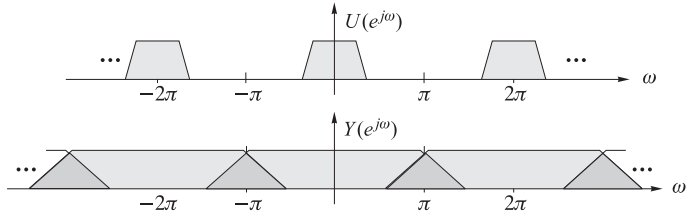
$$\boxed{\hat{X}(z) = F(z)Y(z^M) = \frac{1}{M} \sum_{k=0}^{M-1} F(z)\, H(z W_M^k)\, X(z W_M^k).} \qquad (5.20)$$

Example:

$M = 4$, no aliasing:



with aliasing:



(from [Mertins: Signal Analysis, 1999])

5.1.4 Polyphase decomposition

- A *polyphase decomposition* of a sequence $x(n)$ leads to M subsequences $x_\ell(m)$, $\ell = 0, \ldots, M-1$, which contain

only every M-th value of $x(n)$. Example for $M = 2$: Decomposition into an even and odd subsequence.

- Important tool for the derivation of efficient multirate filtering structures later on.

Three different decomposition types:

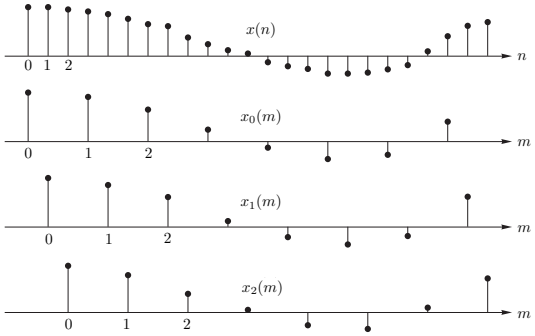
- *Type-1 polyphase components*:
  Decomposition of $x(n)$ into $x_\ell(m)$, $\ell = 0, 1, \ldots, M-1$ with

$$x_\ell(m) = x(mM + \ell), \quad n = mM + \ell. \tag{5.21}$$

  With $x_\ell(m) \circ\!\!-\!\!\bullet\, X_\ell(z)$ the z-transform $X(z)$ can be obtained as

$$\boxed{X(z) = \sum_{\ell=0}^{M-1} z^{-\ell} X_\ell(z^M)} \tag{5.22}$$

  Example for $M = 3$:



$$x_0(0) = x(0), \quad x_0(1) = x(3), \quad \ldots$$
$$x_1(0) = x(1), \quad x_1(1) = x(4), \quad \ldots$$
$$x_2(0) = x(2), \quad x_2(1) = x(5), \quad \ldots$$

- *Type-2 polyphase components*:

$$\boxed{X(z) = \sum_{\ell=0}^{M-1} z^{-(M-1-\ell)} X'_\ell(z^M)} \tag{5.23}$$

  with $\quad X'_\ell(z) \bullet\!\!-\!\!\circ\, x'_\ell(n) = x(nM + M - 1 - \ell)$ (5.24)

  Example for $M = 3$:

$$x'_0(0) = x(2), \quad x'_0(1) = x(5), \quad \ldots$$
$$x'_1(0) = x(1), \quad x'_1(1) = x(4), \quad \ldots$$
$$x'_2(0) = x(0), \quad x'_2(1) = x(3), \quad \ldots$$

- *Type-3 polyphase components*:

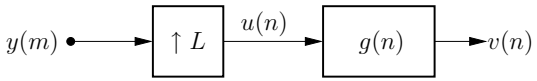
$$\boxed{X(z) = \sum_{\ell=0}^{M-1} z^{\ell} \bar{X}_\ell(z^M)} \tag{5.25}$$

  with $\quad \bar{X}_\ell(z) \bullet\!\!-\!\!\circ\, \bar{x}_\ell(n) = x(nM - \ell)$ (5.26)

5.1.5 Nyquist-Filters

Nyquist- or L-band filters:

- Used as interpolator filters since they preserve the nonzero samples at the output of the upsampler also at the interpolator output.
- Computationally more efficient since they contain zero coefficients.
- Preferred in interpolator and decimator designs.



Using (5.14) the input-output relation of the interpolator can be stated as $V(z) = G(z)\,Y(z^L)$.

The filter $G(z)$ can be written in polyphase notation according to

$$G(z) = G_0(z^L) + z^{-1}\,G_1(z^L) + \cdots + z^{-(L-1)}\,G_{L-1}(z^L),$$

where the $G_\ell(z)$, $\ell = 0, \ldots, L-1$ denote the type 1 polyphase components of the filter $G(z)$.

Suppose now that the m-th polyphase component of $G(z)$ is a constant, i.e. $G_m(z) = \alpha$. Then the interpolator output $V(z)$ can be expressed as

$$V(z) = \alpha z^{-m} Y(z^L) + \sum_{\ell=0,\ \ell\neq m}^{L-1} z^{-\ell}\,G_\ell(z^L) Y(z^L). \tag{5.27}$$

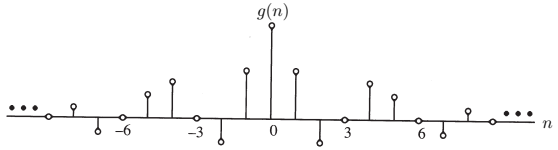
$\Rightarrow v(Ln+m) = \alpha y(n)$; the input samples appear at the output of the system without any distortion for all n. All in-between $(L-1)$ samples are determined by interpolation.

Properties:

- Impulse response of a zero-phase L-th band filter:

$$g(Ln) = \begin{cases} \alpha & \text{for} \quad n = 0, \\ 0 & \text{otherwise.} \end{cases} \tag{5.28}$$

  $\Rightarrow$ every L-th coefficient is zero (except for $n = 0$) $\rightarrow$ computationally attractive

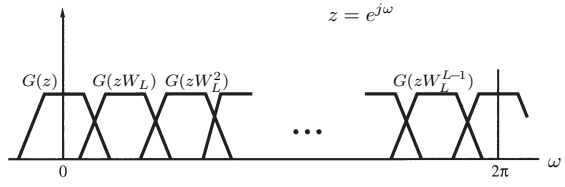


(from [Mitra, 2000])

- It can be shown for $\alpha = 1/L$ that for a zero-phase L-th band filter

$$\boxed{\sum_{\ell=0}^{L-1} G(zW_L^\ell) = L\alpha = 1.} \tag{5.29}$$

  $\Rightarrow$ The sum of all L uniformly shifted versions of $G(e^{j\omega})$ add up to a constant.



(from [Mitra, 2000])

Half-band filters

Special case of L-band filters for $L = 2$:

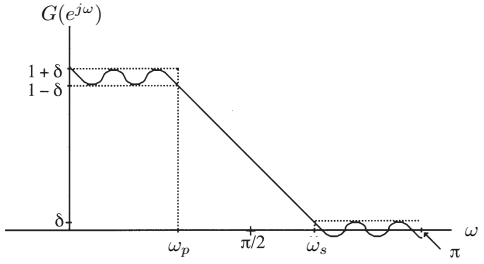
- Transfer function $G(z) = \alpha + z^{-1}G_1(z^2)$
- For $\alpha = 1/2$ we have from (5.29) for the zero-phase filter $g(n)$

$$G(z) + G(-z) = 1. \tag{5.30}$$

- If $g(n)$ is real-valued then $G(-e^{j\omega}) = G(e^{j(\pi-\omega)})$ and by using (5.30) it follows

$$\boxed{G(e^{j\omega}) + G(e^{j(\pi-\omega)}) = 1.} \tag{5.31}$$

$\Rightarrow G(e^{j\omega})$ exhibits a symmetry with respect to the half-band frequency $\pi/2 \rightarrow$ *halfband filter*.

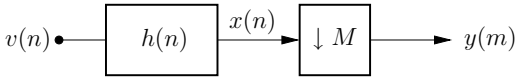


(from [Mitra, 2000])

- FIR linear-phase halfband filter: Length is restricted to $L_F = 4\lambda - 1, \lambda \in \mathbb{N}$

5.2 Structures for decimation and interpolation

5.2.1 FIR direct form realization for decimation



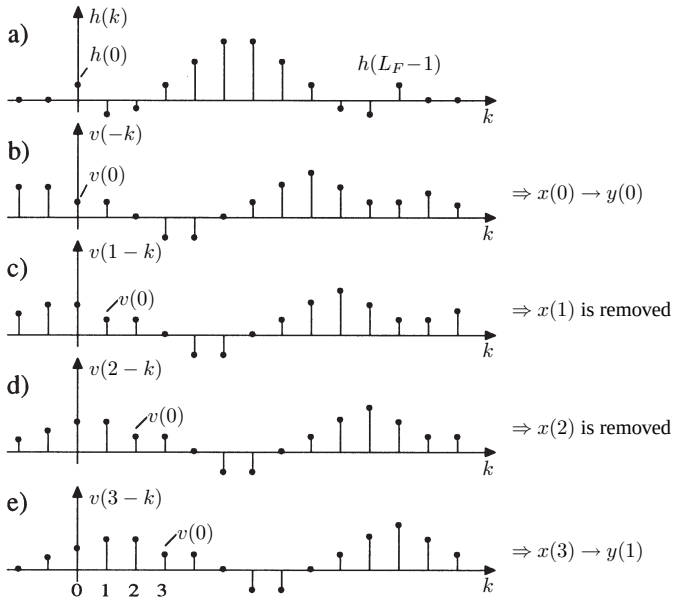
The convolution with the length L_F FIR filter $h(n)$ can be described as

$$x(n) = \sum_{k=0}^{L_F-1} h(k) \cdot v(n-k),$$

and the downsampling with $y(m) = x(m\,M)$. Combining both equations we can write the decimation operation according to

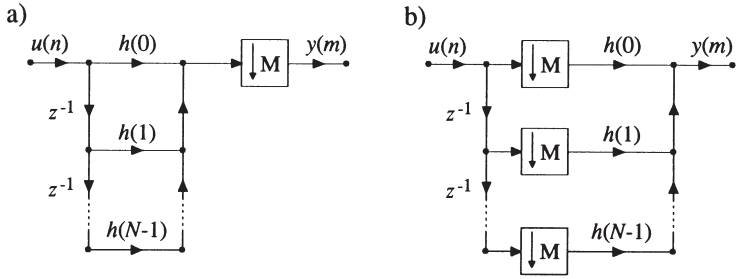
$$y(m) = \sum_{k=0}^{L_F-1} h(k) \cdot v(m\,M - k). \tag{5.32}$$

Visualization ($M = 3$):



$\Rightarrow$ Multiplication of $h(n)$ with $v(1-n)$ and $v(2-n)$ leads to the results $x(1)$ and $x(2)$ which are discarded in the decimation process $\rightarrow$ these computations are not necessary.

More efficient implementation ($v(n) \rightarrow u(n)$, $L_F \rightarrow N$):

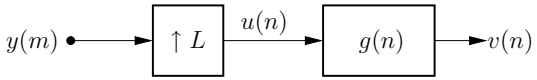


(from [Fliege: Multiraten-Signalverarbeitung, 1993])

(a) Antialiasing FIR filter in first direct form followed by downsampling.
(b) Efficient structure obtained from shifting the downsampler before the multipliers:

- Multiplications and additions are now performed at the lower sampling rate.
- Additional reductions can be obtained by exploiting the symmetry of $h(n)$ (linear-phase).

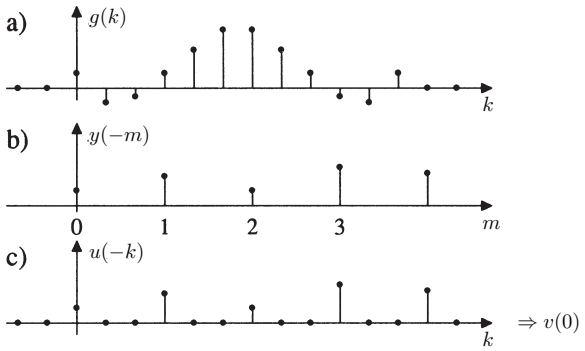
5.2.2 FIR direct form realization for interpolation



The output $v(n)$ of the interpolation filter can be obtained as

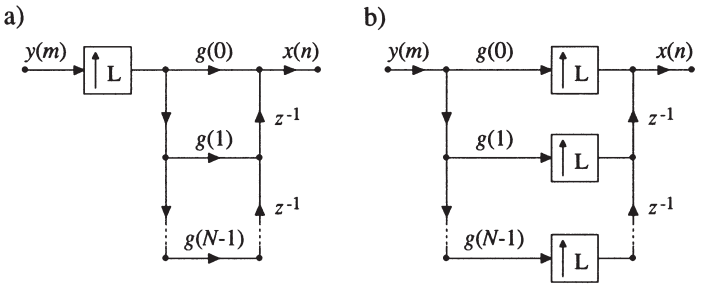
$$v(n) = \sum_{k=0}^{L_F-1} g(k) \cdot u(n-k),$$

which is depicted in the following:



$\Rightarrow$ The output sample $v(0)$ is obtained by multiplication of $g(n)$ with $u(-n)$, where a lot of zero multiplications are involved, which are inserted by the upsampling operation.

More efficient implementation ($v(n) \rightarrow x(n)$, $L_F \rightarrow N$):



(a) Upsampling followed by interpolation FIR filter in second direct form
(b) Efficient structure obtained from shifting the upsampler behind the multipliers:

- Multiplications are now performed at the lower sampling

rate, however the output delay chain still runs in the higher sampling rate.

- Zero multiplications are avoided.
- Additional reductions can be obtained by exploiting the symmetry of $h(n)$ (linear-phase).

5.3 Decimation and interpolation with polyphase filters

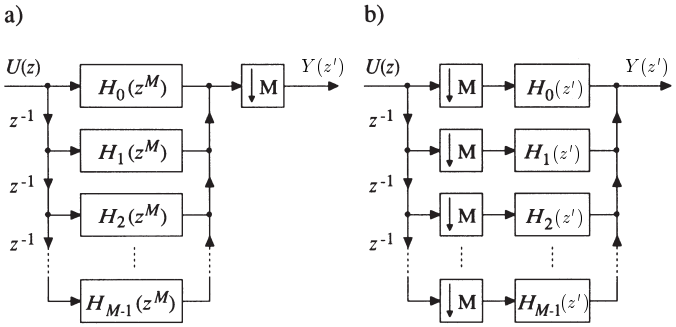
5.3.1 Decimation

- From Section 5.1.4 we know that a sequence can be decomposed into polyphase components. Here type-1 polyphase components (5.21) are considered in the following.
- Type-1 polyphase decomposition of the decimation filter $h(n)$: The z-transform $H(z)$ can be written according to (5.22) as

$$H(z) = \sum_{\ell=0}^{M-1} z^{-\ell} H_\ell(z^M), \qquad (5.33)$$

M denoting the downsampling factor and $H_\ell(z') \bullet\!\!-\!\!\circ\, h_\ell(m)$ the z-transform of the type-1 polyphase components $h_\ell(m)$, $\ell = 0, \ldots, M-1$.

Resulting decimator structure ($V(z) \rightarrow U(z)$):

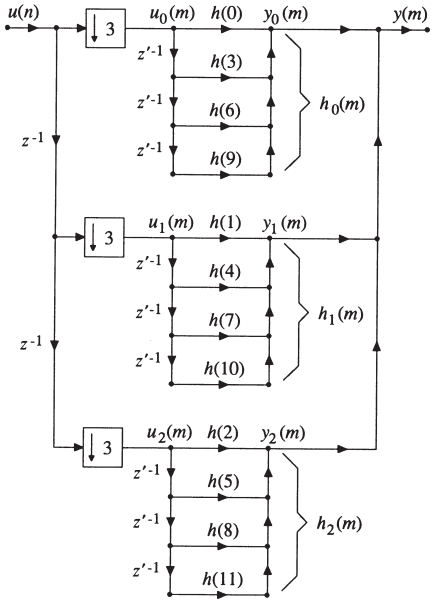


(from [Fliege: Multiraten-Signalverarbeitung, 1993])

(a): Decimator with decimation filter in polyphase representation
(b): Efficient version of (a) with M times reduced complexity

Remark: The structure in (b) has the same complexity as the direct form structure from Section 5.2.1, therefore no further advantage. However, the polyphase structures are important for digital filter banks which will be discussed later on.

Structure (b) in time domain ($v(n) \rightarrow u(n)$):



(from [Fliege: Multiraten-Signalverarbeitung, 1993])

5.3.2 Interpolation

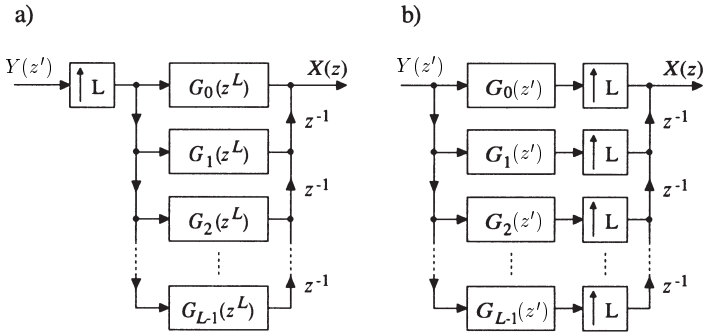
Transfer function of the interpolation filter can be written analog

to (5.33) for the decimation filter as

$$G(z) = \sum_{\ell=0}^{L-1} z^{-\ell} G_\ell(z^L),$$

L denoting the upsampling factor, and $g_\ell(m)$ the type-1 polyphase components of $g(n)$ with $g_\ell(m) \circ\!\!-\!\!\bullet\, G_\ell(z')$.

Resulting interpolator structure ($V(z) \rightarrow X(z)$):



(from [Fliege: Multiraten-Signalverarbeitung, 1993])

(a): Interpolator with interpolation filter in polyphase representation
(b): Efficient version of (a) with L times reduced complexity

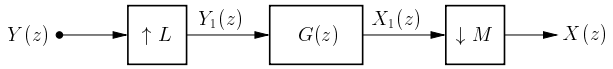
As in the decimator case the computational complexity of the efficient structure in (b) is the same as for the direct form interpolator structure from Section 5.2.2.

5.4 Noninteger sampling rate conversion

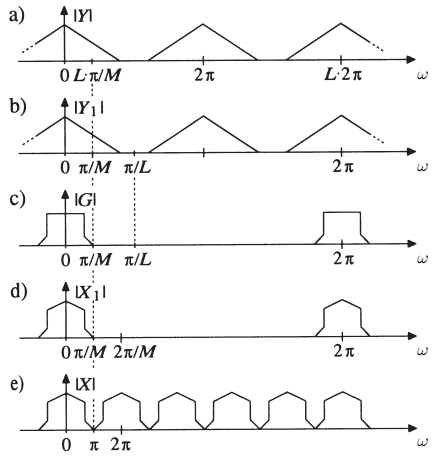
Notation: For simplicity a delay by one sample will be generally denoted with

z^{-1} for every sampling rate in a multirate system in the following (instead of introducing a special z for each sampling rate as in the sections before).

- In practice often there are applications where data has to be converted between different sampling rates with a rational ratio.
- Noninteger (synchronous) sampling rate conversion by factor L/M: Interpolation by factor L, followed by a decimation by factor M; decimation and interpolation filter can be combined:



- Magnitude frequency responses:



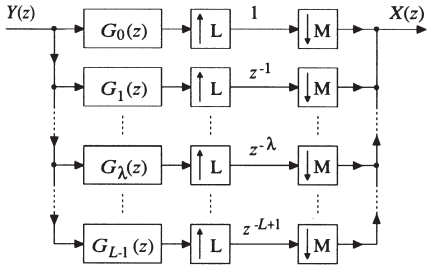
(from [Fliege: Multiraten-Signalverarbeitung, 1993])

Efficient conversion structure

In the following derivation of the conversion structure we assume

a ratio $L/M < 1$. However, a ratio $L/M > 1$ can also be used with the *dual* structures.

1. Implementation of the filter $G(z)$ in polyphase structure, shifting of all subsamplers into the polyphase branches:

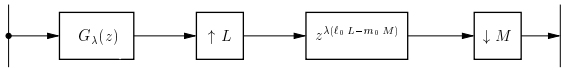


(from [Fliege: Multiraten-Signalverarbeitung, 1993])

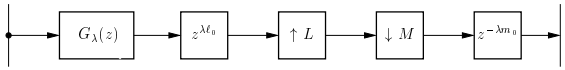
2. Application of the following structural simplifications:
   (a) It is known that if L and M are coprime (that is they have no common divider except one) we can find $\ell_0, m_0 \in \mathbb{N}$ such that

$$\ell_0 \, L - m_0 \, M = -1 \quad \text{(diophantic equation)} \qquad (5.34)$$

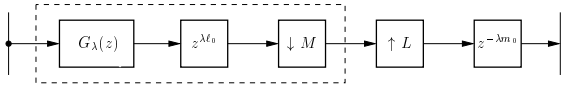
   $\Rightarrow$ delay $z^{-\lambda}$ in one branch of the polyphase structure can be replaced with the delay $z^{\lambda(\ell_0 \, L - m_0 \, M)}$



   (b) The factor $z^{\lambda \ell_0 L}$ can be shifted before the upsampler, and the factor $z^{-\lambda m_0 M}$ behind the downsampler:

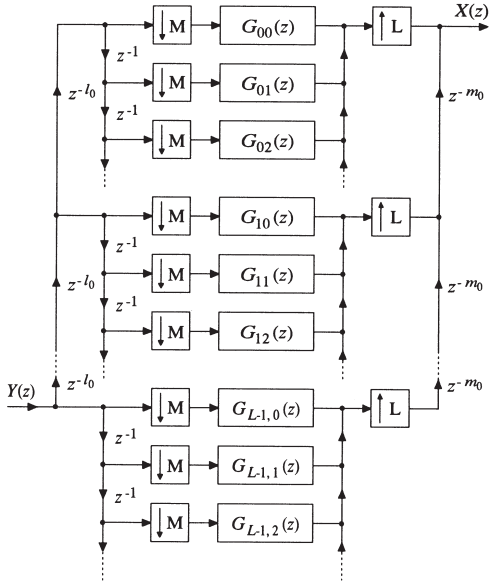


   (c) Finally, if M and L are coprime, it can be shown that up- and downsampler may be exchanged in their order:



   (d) In every branch we now have a decimator (marked with the dashed box), which can again be efficiently realized using the polyphase structure from Section 5.3.1. Thus, each type-1 polyphase component $g_\lambda(n)$ is itself decomposed again in M polyphase components $g_{\lambda\mu}(n) \circ\!\!-\!\!\bullet\, G_{\lambda\mu}(z)$, $\lambda = 0, \ldots, L-1$, $\mu = 0, \ldots, M-1$.

Resulting structure:



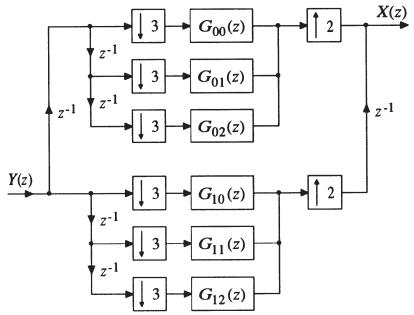
(from [Fliege: Multiraten-Signalverarbeitung, 1993])

- Delays $z^{-\lambda m_0}$ are realized with the output delay chain.

- The terms $z^{\lambda\ell_0}$ are noncausal elements: In order to obtain a causal representation, we have to insert the extra delay block $z^{-(L-1)\ell_0}$ at the input of the whole system, which cancels out the "negative" delays $z^{\lambda\ell_0}$.
- Polyphase filters are calculated with the lowest possible sampling rate.
- $L/M > 1$ is realizable using the dual structure (exchange: input $\leftrightarrow$ output, downsamplers $\leftrightarrow$ upsamplers, summation points $\leftrightarrow$ branching points, reverse all branching directions)

Example for $L = 2$ and $M = 3$:

Application: Sampling rate conversion for digital audio signals from 48 kHz to 32 kHz sampling rate



(from [Fliege: Multiraten-Signalverarbeitung, 1993])

Polyphase filters are calculated with 16 kHz sampling rate compared to 96 kHz sampling rate in the original structure.

Rate conversion from 32 kHz to 48 kHz: Exercise!

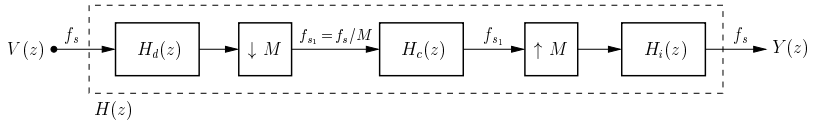
5.5 Efficient multirate filtering

In the following we only consider lowpass filtering, however, the

presented methods can easily be extended to band- or highpass filters.

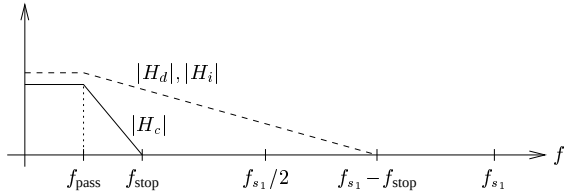
5.5.1 Filtering with lower sampling rate

If the stopband edge frequency of a lowpass filter is substantially smaller than half of the sampling frequency, it may be advisable to perform the filtering at a lower sampling rate:



$H_d(z)$ is the (input) decimation filter, the actual filtering is carried out with the core filter $H_c(z)$ in the M-times lower sampling rate with sampling frequency $f_{s_1} = f_s/M$, and after upsampling the output signal is interpolated with $H_i(z)$ $\Rightarrow$ *Single-stage* implementation

Stopband- and passband edge frequencies of the decimation and interpolation filters have to be adjusted to the filter specifications of the core filter:



- Stop- and passband edge frequencies f_{stop} and f_{pass} of the core filter $H_c(z)$ are identical with those for the overall filter $H(z) = Y(z)/V(z)$.

- Stopband edge frequency for the decimation filter then has to be chosen less or equal than $(f_{s_1} - f_{\text{stop}})$.
- The interpolation filter can be chosen identical to the decimation filter, since then it is guaranteed that all imaging components are in the stopband region of the interpolation filter.
- Transition bandwidth for $H(z)$ is M-times smaller than for $H_c(z) \Rightarrow$ design with a fairly small number of coefficients for $H_c(z)$ possible (compared to a direct design of $H(z)$).
- Stopband ripple δ_2 for the overall filter $H(z)$:

$$\delta_2 = \begin{cases} \delta_{2,c}(1+\delta_{1,i})(1+\delta_{1,d}) \approx \delta_{2,c}, & f_{\text{stop}} \le f \le (f_{s_1} - f_{\text{stop}}), \\ \delta_{2,c}\,\delta_{2,d}\,\delta_{2,i}, & (f_{s_1} - f_{\text{stop}}) < f \le f_s \end{cases} \tag{5.35}$$

  where the approximation for $f_{\text{stop}} \le f \le (f_{s_1} - f_{\text{stop}})$ holds for small decimation and interpolation filter passband ripples $\delta_{1,d}$ and $\delta_{1,i}$.
- Passband ripple δ_1 for $H(z)$:

$$1 + \delta_1 = (1+\delta_{1,c})\,(1+\delta_{1,d})\,(1+\delta_{1,i}), \tag{5.36}$$

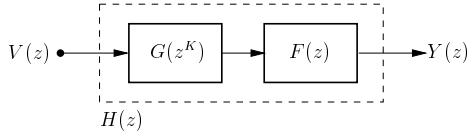
$$\overset{\text{approximation}}{\Longrightarrow} \qquad \delta_1 \approx \delta_{1,c} + \delta_{1,d} + \delta_{1,i}, \tag{5.37}$$

  where the last approximation is valid for small passband ripples $\delta_{1,c}$, $\delta_{1,d}$, and $\delta_{1,i}$.
- Complexity savings (#multiplications and #additions) can be obtained by roughly a factor of 100. An even higher gain can be achieved by multistage implementations.

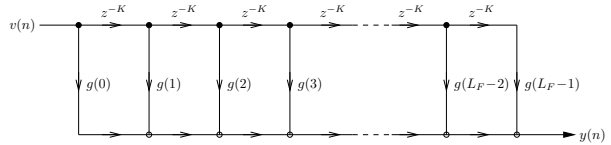
5.5.2 Interpolating FIR (IFIR) filters

Alternative to multirate filters with decimation and interpolation, also suitable for very narrowband filters.

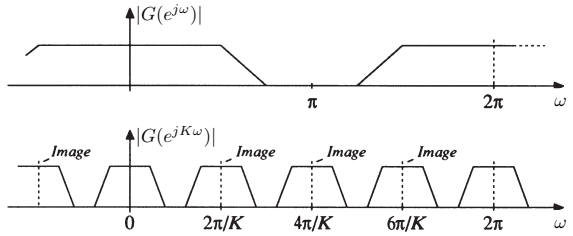
Principle:



- No real multirate filter since both filters are calculated with the same (input) sampling rate. Multirate technique is applied to the coefficients of the impulse response $h(n)$.
- Realization of $G(z^K)$ in the first direct structure:



  $G(z^K)$ is a function where all z^{-1} are replaced by z^{-K}, which is equivalent to inserting $K-1$ zeros between the coefficients of $G(z)$.
- $G(e^{j\omega}) \rightarrow G(e^{jK\omega})$: Frequency response $G(e^{j\omega})$ is "compressed" by factor K, $K-1$ imaging spectra are present:



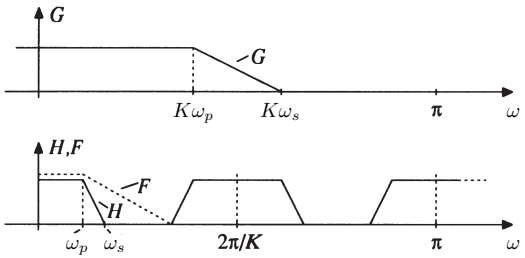
(from [Fliege: Multiraten-Signalverarbeitung, 1993])

Furthermore, the transition bandwidth and the width of the passband for $G(e^{jK\omega})$ are K-times smaller than for the original filter $G(e^{j\omega})$ with the same number of filter coefficients.

- The filter $F(z)$ removes the imaging spectra, and $H(z) = G(z^K) \cdot F(z)$ only consists of the baseband part of $G(e^{jK\omega})$.

Design: Starting point for the design: Passband and stopband edge frequencies ω_s, ω_p for the overall filter $H(z) \longrightarrow$ search for a suitable factor K leading to a less complex interpolation filter $f(n)$.

Filter specifications (H, F, G are all *magnitude frequency responses*)



(from [Fliege: Multiraten-Signalverarbeitung, 1993])

- Requirements for passband and stopband edge frequency of the prototype $G(z)$:

$$\omega_{p,G} = K \cdot \omega_p, \quad \omega_{s,G} = K \cdot \omega_s. \tag{5.38}$$

- Requirements for passband and stopband edge frequency of

the interpolation filter $F(z)$:

$$\omega_{p,F} = \omega_p, \quad \omega_{s,F} = \frac{2\pi}{K} - \omega_s. \tag{5.39}$$

- Passband ripple δ_1 for $H(z)$:

$$1 + \delta_1 = (1 + \delta_{1,G})\,(1 + \delta_{1,F}). \tag{5.40}$$

Small passband ripples $\delta_{1,G}$ for $G(z)$ and $\delta_{1,F}$ for $F(z)$, resp., lead to the simplification

$$\delta_1 \approx \delta_{1,G} + \delta_{1,F}. \tag{5.41}$$

- Stopband ripple δ_2 for $H(z)$:

$$\delta_2 = \begin{cases} \delta_{2,G}\,(1 + \delta_{1,F}) & \text{for} \quad \omega_s \le \omega \le \omega_{s,F}, \\ \delta_{2,F}\,(1 + \delta_{1,G}) & \text{for} \quad \omega_{s,F} < \omega \le \pi. \end{cases} \tag{5.42}$$

For small passband ripples $\delta_{1,G}$, $\delta_{1,F}$ we have approximately

$$\delta_2 \approx \delta_{2,F} = \delta_{2,G}. \tag{5.43}$$

Example:

Design a lowpass IFIR filter with the following specifications:

$\omega_d = 0.05\pi, \quad \omega_s = 0.1\pi,$

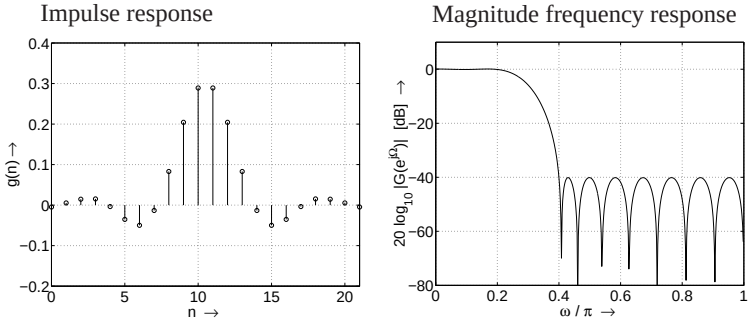
$20\log_{10}(1 + \delta_1) = 0.2\,\text{dB} \rightarrow \delta_1 \approx 0.023, \;\; 20\log_{10}(|\delta_2|) = -40\,\text{dB}$

1. We select a factor $K = 4$: Prototype $G(z)$ has the parameters

$$\omega_{p,G} = 0.2\pi, \quad \omega_{s,G} = 0.4\pi,$$
$$\delta_{1_G} \approx 0.0116 \rightarrow 20\log_{10}(1+\delta_{1,G}) \approx 0.1\,\text{dB},$$
$$20\log_{10}(|\delta_{2,G}|) = -40\,\text{dB},$$

ripple is equally distributed between $G(z)$ and $F(z)$, see (5.41).

2. We use an linear-phase FIR Chebyshev design and insert these values for $G(z)$ into (4.73), yielding a filter order $N = 19$. However, several test designs show that the above specifications are only met for $N = 21$, leading to a filter length of $L_F = N+1 = 22$.

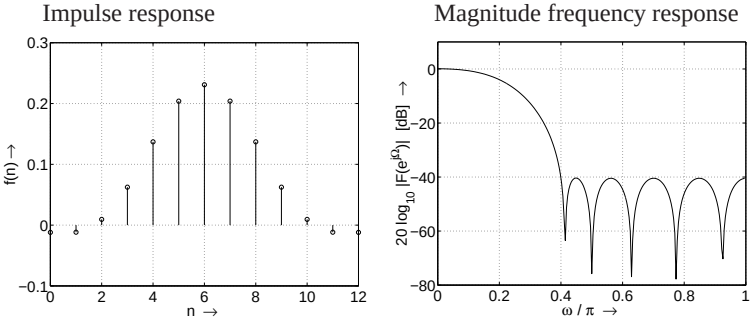


3. Specifications for the interpolation filter $F(z)$:

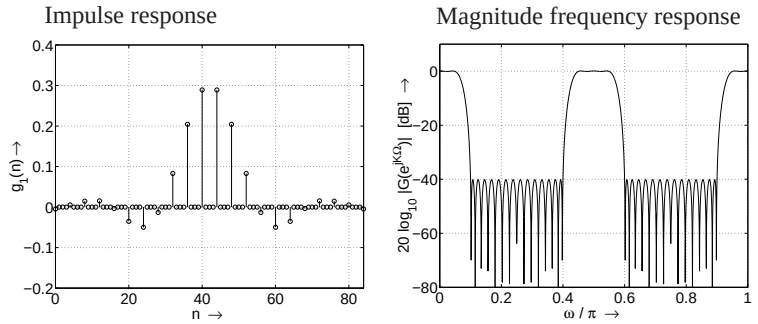
$$\omega_{p,F} = \omega_p = 0.05\pi, \quad \omega_{s,F} = \frac{2\pi}{K} - \omega_s = 0.4\pi,$$

$$20\log_{10}(1+\delta_{1,F}) \approx 0.1\,\text{dB}, \quad 20\log_{10}(|\delta_{2,F}|) = -40\,\text{dB}.$$

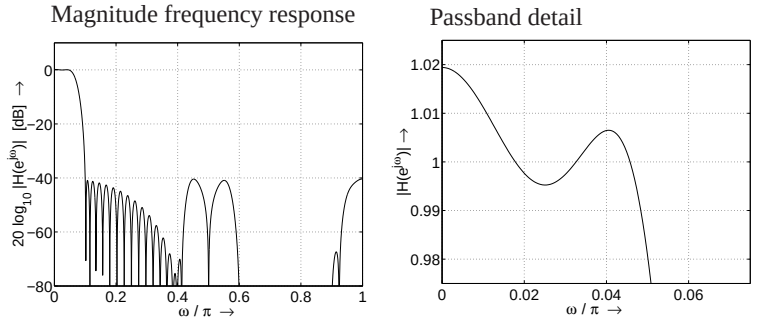
$\Rightarrow$ resulting filter order ((4.73)) $N = 12$:



4. Upsampling of the impulse response $g(n)$ by factor $K = 4 \rightarrow g_1(n) \circ\!\!-\!\!\bullet\, G(e^{jK\omega})$:



5. Final IFIR filter $h(n) = g_1(n) * f(n)$:



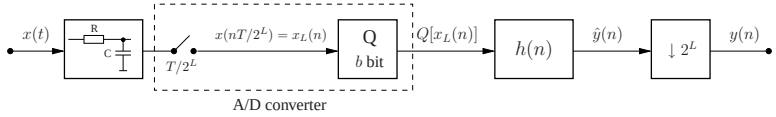
6. $H(z)$ in the IFIR realization has an overall length of 35 coefficients. On the other hand we obtain from (4.73) an estimated length of 65 coefficients for the direct form implementation.

5.6 Application: Oversampled A/D and D/A converter

5.6.1 Oversampled A/D converter

Extension of the A/D and D/A conversion concepts from Section 2.

Structure:



- Continuous-time input signal is band-limited by the analog lowpass such that the frequency ω_u represents the maximum frequency in the interesting frequency range for the input signal:

  Sampling with a sampling frequency $2^L \cdot \omega_s \geq 2 \cdot \omega_u$, $L \in \{0, 1, 2, \dots\}$, after the analog filtering. Here ω_s denotes the lowest possible sampling frequency in order not to violate the sampling theorem: $\omega_s = 2 \cdot \omega_u = 2\pi/T$.

- A/D converter here is idealized as concatenation of sampler and quantizer.

- After A/D conversion a lowpass filtering is carried out where the lowpass has the idealized frequency response

$$|H(e^{j\omega})| = \begin{cases} 1 & \text{for } |\omega| < \pi/2^L, \\ 0 & \text{otherwise.} \end{cases}$$

  The resulting bandlimited signal can then be downsampled by factor 2^L.

Quantization noise variance of a b-bit midtreat quantizer according to (2.27), where the range R of the quantizer is chosen as $R = 2$:

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

As an alternative σ_e^2 can also be obtained via the power spectral density (power spectrum) $\Phi_{ee}(e^{j\omega})$ as

$$\sigma_e^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{ee}(e^{j\omega})\, d\omega.$$

The filtering with the lowpass $h(n)$ now reduces the quantization noise variance by factor 2^L, since

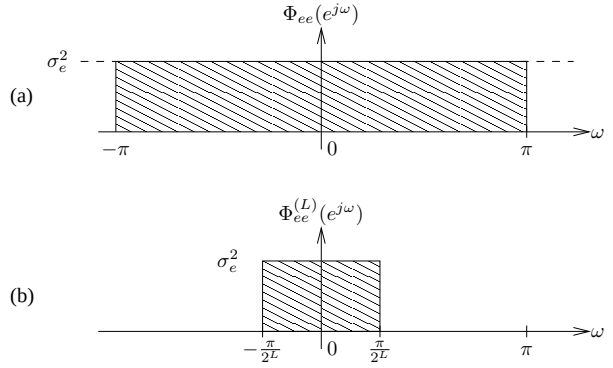
$$\sigma_{e_{(L,b)}}^2 = \frac{1}{2\pi} \int_{-\pi/2^L}^{\pi/2^L} \Phi_{ee}^{(L)}(e^{j\omega})\, d\omega = \frac{2^{-2b+2}}{12 \cdot 2^L}. \qquad (5.44)$$

$\Rightarrow$ Reduction of the noise variance due to oversampling by factor 2^L:

$$\text{Gain} = 10\, \log_{10}\left(\frac{\sigma_e^2}{\sigma_{e_{(L,b)}}^2}\right) = 10\, \log_{10}(2^L) = L \cdot 3.01\,\text{dB}. \qquad (5.45)$$

$\Rightarrow$ An increase of the *oversampling factor* 2^L by factor 2 leads to an increase in quantizer resolution by half a bit (compare (2.29)) and to an increase of the signal-to-noise ratio (SNR) in (2.29) by 3 dB!

Visualization:



Gain in (5.45) can also be used for reducing the quantizer wordlength while keeping the SNR constant:

- Reduction of the hardware complexity of the core A/D converter
- Extreme case: Quantizer wordlength is reduced to $b = 1$ bit $\rightarrow$ only a simple comparator is necessary

Requirements:

$$\frac{\sigma^2_{e_{(L,b_L)}}}{\sigma^2_{e_{(0,b_0)}}} \stackrel{!}{=} 1 = \frac{2^{-2b_L+2}}{2^L \cdot 2^{-2b_0+2}} \quad \rightarrow \quad b_L = b_0 - \frac{L}{2}, \quad (5.46)$$

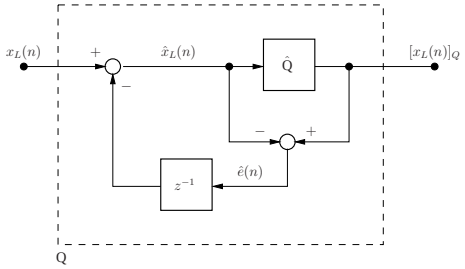
with b_L denoting the quantizer wordlength for a given L leading to the same quantization noise variance as b_0 for $L = 0$, $b_L \leq b_0$.

Example:
Given a $b_0 = 16$ bit A/D converter, where the analog input signal is sampled at the Nyquist rate. Choose the parameter L in the oversampled A/D converter from above such that the same quantization noise variance for $b_L = 1$ bit is achieved.

From (5.46) we obtain $L = 30$, leading to an oversampling factor of $2^L \approx 10^9$.

Improvement: Shaping of the quantization noise

The quantizer in the above block is now replaced by the following structure:



Analysis:

$$\begin{aligned} y(n) = [x_L(n)]_Q &= \hat{Q}[x_L(n) - \hat{e}(n-1)], \\ &= x_L(n) - \hat{e}(n-1) + \hat{e}(n), \\ &= x_L(n) + \hat{e}(n) * (1 - \delta(n-1)) \quad (5.47) \end{aligned}$$

$$\circ\!\!-\!\!\bullet \quad Y(z) = X_L(z) + \hat{E}(z)(1 - z^{-1}). \quad (5.48)$$

Therefore, the z-transform of the overall quantization error sequence can be expressed as

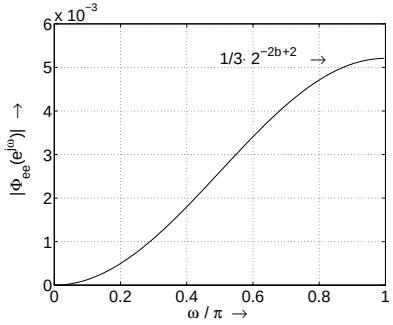
$$E(z) = \hat{E}(z)\left(1 - z^{-1}\right),$$

which leads to the quantization noise power spectrum

$$\Phi_{ee}(e^{j\omega}) = \Phi_{\hat{e}\hat{e}}(e^{j\omega})|1 - e^{-j\omega}|^2.$$

With $\Phi_{\hat{e}\hat{e}}(e^{j\omega}) = \frac{2^{-2b+2}}{12}$ (noise power spectrum of a b-bit midtreat quantizer with range $R = 2$) we have

$$\Phi_{ee}(e^{j\omega}) = \frac{2^{-2b+2}}{6}(1 - \cos(\omega)). \quad (5.49)$$



$\Rightarrow$ Quantization noise power spectrum now has highpass character $\rightarrow$ *noiseshaping*

The noise variance after lowpass filtering with $h(n)$ in the above oversampled A/D converter structure is now given with (5.49) as

$$\sigma^2_{e_{(L,b)}} = \frac{1}{2\pi} \int\limits_{-\pi/2^L}^{\pi/2^L} \Phi_{ee}(e^{j\omega})\, d\omega,$$

$$= \frac{2^{-2b+2}}{12} \left(2^{-L+1} - \frac{2}{\pi} \sin(2^{-L}\pi) \right). \qquad (5.50)$$

Reduction of the noise variance due to oversampling by factor 2^L:

$$\text{Gain} = -10\ \log_{10} \left(2^{-L+1} - \frac{2}{\pi} \sin(2^{-L}\pi) \right). \qquad (5.51)$$

For $L = 4$: Gain $\approx$ 31 dB (compared to 12 dB without noise shaping, see (5.45)).

Reducing the quantizer wordlength for constant SNR:

$$\frac{\sigma^2_{e_{(0,b_0)}}}{\sigma^2_{e_{(L,b_L)}}} \stackrel{!}{=} 1 \ \rightarrow\ b_L = b_0 + \frac{1}{2} \log_2 \left(2^{-L+1} - \frac{2}{\pi}\ \sin(2^{-L}\pi) \right). \qquad (5.52)$$

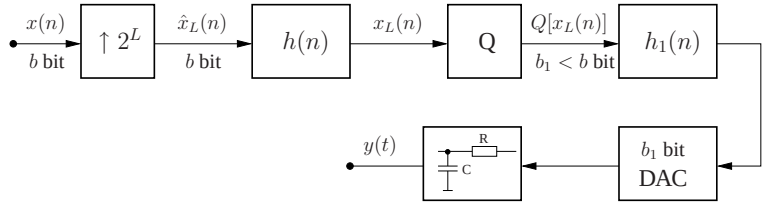
Example:

The above example is again carried out for the noiseshaping case: For $b_0 = 16$ bit and $b_L = 1$ bit we obtain from (5.52) via a computer search (fix-point iteration) $L \approx 10.47$ $\rightarrow$ When the input signal is sampled with $f_s = 44.1$ kHz and $b_0 = 16$ the new sampling frequency in the oversampled A/D converter would be $f_{s\text{over}} = 2^{11} \cdot f_s \approx 90$ MHz for $b_L = 1$.

$\Rightarrow$ Improved noiseshaping by other techniques, where the quantization noise power spectrum shows even stronger highpass character (sigma-delta modulation, more selective shaping filters).

5.6.2 Oversampled D/A converter

Structure:



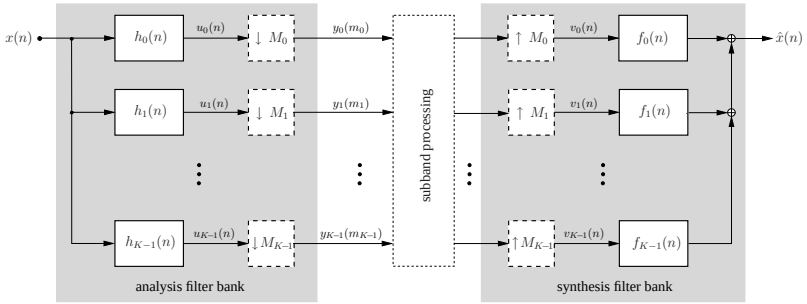
1. Input signal sampled with b bits is upsampled by factor 2^L and then interpolated with $h(n)$.
2. The resulting signal $x_L(n)$ is requantized to a wordlength of $b_1 < b$ bits, leading to a worse SNR due to higher quantization noise.
3. Filtering by $h_1(n)$ removes the quantization noise in the unused spectral regions, which increases the SNR again.
4. The b_1 bit DAC in combination with a simple analog lowpass converts the signal back into the analog domain.

Reason for performing a requantization: Use of a cheap low resolution D/A converter is possible, often with $b_1 = 1$ bit. In combination with a noiseshaping approach the same SNR is obtained as when a b bit converter would be used directly on the input signal (but with higher costs).

$\Rightarrow$ Favored converter principle in CD players ($\rightarrow$ *"bitstream" conversion* for $b_1 = 1$ bit)

5.7 Digital filter banks

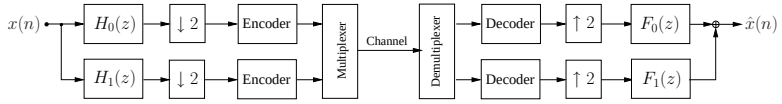
- A *digital filter bank* consists of a set of filters (normally lowpass, bandpass and highpass filters) arranged in a parallel bank.
- The filters split the input signal $x(n)$ into a number of subband signals $y_k(n)$, $k = 0, \ldots, K-1$ (*analysis filter bank*).
- Subband signals are processed and finally combined in a *synthesis filter bank* leading to the output signal $\hat{x}(n)$.
- If the bandwidth of the subband signal is smaller than the bandwidth of the original signal, they can be downsampled before processing → processing is carried out more efficiently.



Subband processing: Quantization (wordlength reduction) → coding, (adaptive) filtering → equalization, ...

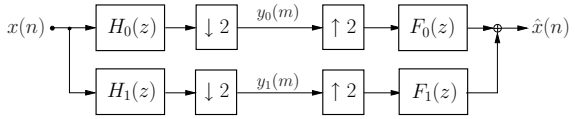
5.7.1 Two-channel filter banks: Structure and analysis

Basic two-channel subband coder:

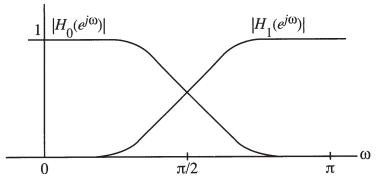


Only the errors in the above system related to the filter bank are

investigated in the following → simplified structure:

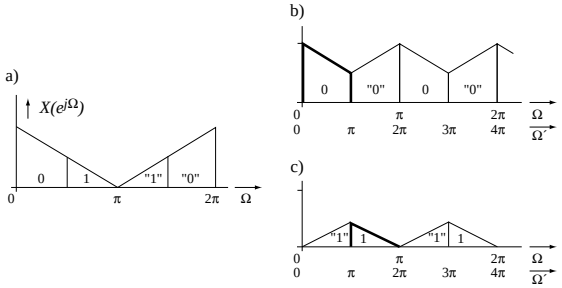


⇒ *Critically subsampled* filter bank: The number of subband equals the subsampling factor in every subband.
Typical frequency responses for the analysis filters:



(from [Mitra, 2000])

Frequency responses of the subband signals (for *ideal* filters) ($\Omega \rightarrow \omega$) :



(From [Vary, Heute, Hess: Digitale Sprachsignalverarbeitung, 1998])

(a): Magnitude frequency response of the input signal, (b) magnitude frequency response of the lowpass, and (c) highpass branch after analysis filtering and downsampling

⇒ Highpass subband: Baseband spectra in *frequency reversed order*

Analysis

How do we have to design the filters such that

$$\hat{x}(n) = x(n - D) \quad \text{holds?}$$

(D denotes the delay of the overall analysis-synthesis system in samples)
We have the following relations:

$$Y_0(z^2) = \frac{1}{2}\left[H_0(z)\,X(z) + H_0(-z)X(-z)\right] \qquad (5.53)$$

$$Y_1(z^2) = \frac{1}{2}\left[H_1(z)\,X(z) + H_1(-z)X(-z)\right] \qquad (5.54)$$

Proof:
These relations can be obtained from (5.18)

$$Y(z) = \frac{1}{M}\sum_{k=0}^{M-1} H(z^{1/M}W_M^k)\,X(z^{1/M}W_M^k)$$

for $M = 2$. With

$$W_2^k = e^{-j2\pi k/2} = e^{-j\pi k} = \begin{cases} 1 & \text{for } n \text{ even,} \\ -1 & \text{for } n \text{ odd,} \end{cases}$$

we have

$$\begin{aligned} Y(z) &= \frac{1}{2}\sum_{k=0}^{1} H((-1)^k z^{1/2})\,X((-1)^k z^{1/2}) \\ &= \frac{1}{2}\left[H(+z^{1/2})\,X(+z^{1/2}) + H(-z^{1/2})\,X(-z^{1/2})\right]. \end{aligned} \qquad (5.55)$$

Replacing z by z^2 then leads to

$$Y(z^2) = \frac{1}{2}\left[H(z)\,X(z) + H(-z)\,X(-z)\right],$$

where (5.53), (5.54) can be obtained by replacing H with H_0 and H_1, resp. □
The connection between the subband signals and the reconstructed signal is

$$\hat{X}(z) = \left[Y_0(z^2)\,F_0(z) + Y_1(z^2)\,F_1(z)\right], \qquad (5.56)$$

and finally by combining (5.56), (5.53), and (5.54) the input-output relation for the two-channel filter bank writes

$$\boxed{\begin{aligned} \hat{X}(z) = &\frac{1}{2}\left[H_0(z)\,F_0(z) + H_1(z)\,F_1(z)\right]\,X(z) + \\ &+ \frac{1}{2}\left[H_0(-z)\,F_0(z) + H_1(-z)\,F_1(z)\right]\,X(-z). \end{aligned}} \qquad (5.57)$$

(5.57) consists of two parts:

$$S(z) = \left[H_0(z)\,F_0(z) + H_1(z)\,F_1(z)\right], \qquad (5.58)$$

$$G(z) = \left[H_0(-z)\,F_0(z) + H_1(-z)\,F_1(z)\right] \qquad (5.59)$$

- $S(z)$: Transfer function for the input signal $X(z)$ through the filter bank, desirable is

$$\boxed{S(z) \stackrel{!}{=} 2z^{-D}.} \qquad (5.60)$$

- $G(z)$: Transfer function for the aliasing component $X(-z)$, desirable is (no aliasing!)

$$\boxed{G(z) \stackrel{!}{=} 0.} \tag{5.61}$$

Two cases have now to be distinguished:

1. If $G(z) = 0$, but $S(z) \neq c\,z^{-D}$, then the reconstructed signal $\hat{x}(n)$ is *free of aliasing*, however, *linear distortions* are present.
2. If $G(z) = 0$ and $S(z) = c\,z^{-D}$, then we have a *perfect reconstruction* (PR) system, except of a scaling factor $c/2$ and an additional delay of D samples.

5.7.2 Two-channel quadrature-mirror filter (QMF) banks

(Crosier, Esteban, Galand, 1976)
Quadrature-mirror filter (QMF) banks allow the *cancelation of all aliasing components*, but generally lead to *linear distortions* (i.e. phase and amplitude distortions)

Starting point: Given (lowpass) prototype $H_0(z)$, all other filters are chosen as

$$\boxed{F_0(z) = H_0(z), \quad H_1(z) = H_0(-z), \quad F_1(z) = -H_1(z)} \tag{5.62}$$

Thus we have from (5.59) for the aliasing transfer function

$$\begin{aligned} G(z) &= H_0(-z)\,F_0(z) + H_1(-z)\,F_1(z) \\ &= H_0(-z)\,H_0(z) + H_0(z)(-H_0(-z)) = 0 \end{aligned} \tag{5.63}$$

$\Rightarrow$ *Cancelation* of all aliasing components
For the linear distortion transfer function $S(z)$ one obtains by inserting (5.62) into (5.58)

$$S(z) = H_0^2(z) - H_0^2(-z),$$

that is, the prototype $H_0(z)$ has to be designed such that

$$S(z) = H_0^2(z) - H_0^2(-z) \stackrel{!}{\approx} 2z^{-D} \tag{5.64}$$

is satisfied as good as possible $\rightarrow$ requirement for an *ideal (constant)* overall frequency response for the whole analysis-synthesis system

- Unfortunately, (5.64) can not be satisfied exactly with FIR filters, but it can be approximated with an arbitrarily small error
  $\Rightarrow$ Linear distortions can be kept small
- Exception: (5.64) is satisfied exactly by using the prototype

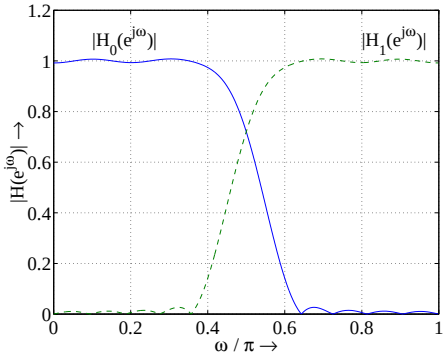
$$H_0(z) = \frac{1}{\sqrt{2}}(1 + z^{-1})$$

(*Haar filter*):

$$\frac{1}{2}(1 + 2z^{-1} + z^{-2}) - \frac{1}{2}(1 - 2z^{-1} + z^{-2}) = 2z^{-1}$$

- The magnitude frequency responses of highpass and lowpass filter have for real-valued filter coefficients the mirror image property (therefore the name QMF):

$$|H_1(e^{j(\frac{\pi}{2}-\omega)})| = |H_0(e^{j(\frac{\pi}{2}+\omega)})| \tag{5.65}$$



Design of QMF banks

- Usually the design is carried out by minimization of an error measure

$$E = E_r + \alpha E_s \stackrel{!}{=} \text{min.} \tag{5.66}$$

E_r refers to the linear distortion error energy

$$E_r = 2 \int_{\omega=0}^{\pi} \left(|H_0(e^{j\omega})|^2 + |H_0(e^{j(\omega-\pi)})|^2 - 1 \right) \, d\omega, \tag{5.67}$$

and E_s to the energy in the stopband region of the filter

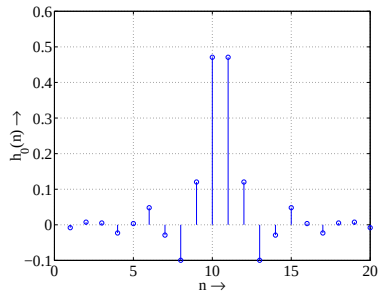
$$E_s = \int_{\omega=\omega_s}^{\pi} |H_0(e^{j\omega})|^2 \, d\omega, \tag{5.68}$$

with the stopband edge $\omega_s = (\pi + \Delta\omega)/2$. $\Delta\omega$ denotes the width of the transition band, which is symmetrically centered around $\omega = \pi/2$.

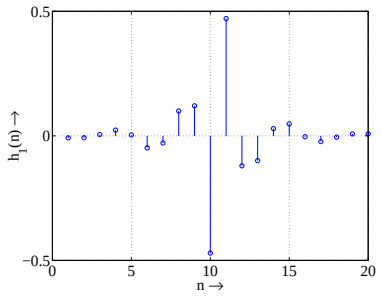
- Minimization of (5.66) can be carried out via a numerical minimization approach for a given $\Delta\omega$ and given prototype length L_F.
- Catalogs with optimized filter coefficients for $h_0(n)$ may for example be found in [Akansu, Haddad: Multirate signal decomposition, 1992].
- Once a good prototype $H_0(z)$ has been found, the remaining filters can be obtained from (5.62).

Example: ($L_F = 20$, $\Delta\omega = 0.2\pi$)

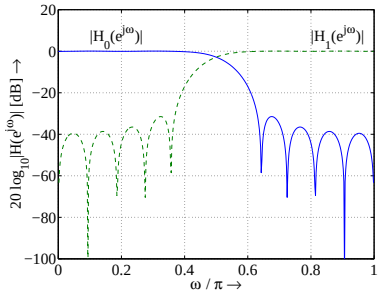
Impulse response $h_0(n)$



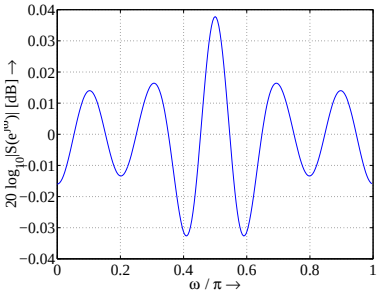
Impulse response $h_1(n)$



Frequency responses for H_0, H_1



Frequency response of the QMF bank

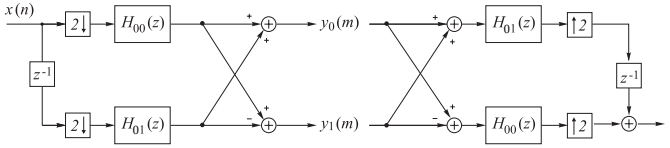


Efficient realization using polyphase components

From (5.62) we know that $H_1(z) = H_0(-z)$ holds for the analysis highpass filter. Then, the type-1 polyphase components $H_{0\ell}(z)$ and $H_{1\ell}(z)$, $\ell \in \{0, 1\}$, are related according to

$$H_{10}(z) = H_{00}(z) \quad \text{and} \quad H_{11}(z) = -H_{01}(z). \tag{5.69}$$

This leads to an efficient realization of the analysis and synthesis filter bank, where the number of multiplications and additions can be reduced by factor four:



(From [Mertins: Signal analysis, 1999])

5.7.3 Two-channel perfect reconstruction filter banks

In order to obtain a perfect reconstruction filter bank the analysis and synthesis filters are chosen as follows ($\ell \in \{0, 1, 2, \dots\}$):

$$\boxed{F_0(z) = z^{-\ell}\, H_1(-z), \quad F_1(z) = -z^{-\ell}\, H_0(-z)} \tag{5.70}$$

Aliasing transfer function: Inserting (5.70) into (5.59) yields

$$\begin{aligned} G(z) &= H_0(-z)\, F_0(z) + H_1(-z)\, F_1(z) \\ &= H_0(-z)\, z^{-\ell}\, H_1(-z) + H_1(-z) \left(-z^{-\ell}\, H_0(-z)\right) \\ &= 0 \quad \Rightarrow \textit{ No aliasing components} \text{ in the reconstructed signal!} \end{aligned}$$

Transfer function for the input signal: Inserting (5.70) into (5.58)

yields

$$\begin{aligned} S(z) &= H_0(z)\, F_0(z) + H_1(z)\, F_1(z) \\ &= H_0(z)\, F_0(z) + (-1)^{\ell+1} H_0(-z) F_0(-z) \end{aligned} \tag{5.71}$$

Condition for a linear distortion free system: $S(z) \stackrel{!}{=} 2z^{-D}$
With the abbreviation

$$T(z) := F_0(z)\, H_0(z) \tag{5.72}$$

the PR condition in (5.71) becomes

$$T(z) + (-1)^{\ell+1}\, T(-z) = 2\, z^{-D}. \tag{5.73}$$

Interpretation:

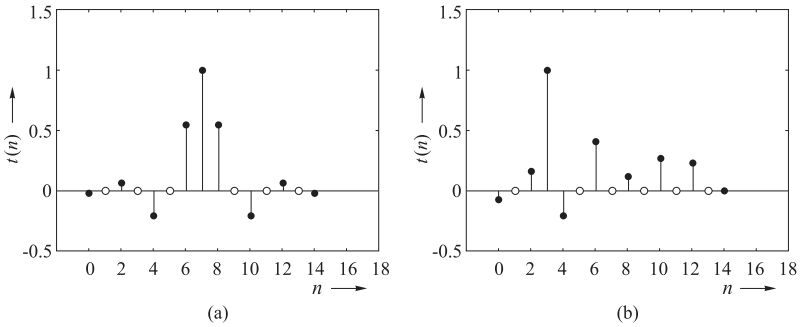
- $[T(z) + T(-z)]$ refers to the z-transform of a sequence whose *odd* coefficients are zero.
- $[T(z) - T(-z)]$: All coefficients with an *even* index are zero.
- The PR condition in (5.71) now states that the corresponding sequences with z-transforms $[T(z) + T(-z)]$ or $[T(z) - T(-z)]$ are allowed to have only one non-zero coefficient. Hence, for $t(n) \circ\!\!-\!\!\bullet\, T(z)$ holds

$$t(n) = \begin{cases} 1 & \text{for} \quad n = D, \\ 0 & \text{for} \quad n = D + 2\lambda, \quad \lambda \neq 0, \\ \text{arbitrary} & \text{otherwise.} \end{cases} \tag{5.74}$$

$\Rightarrow$ Half-band filter (Nyquist(2) filter), see Section 5.1.5.

A half-band filter has $4\lambda - 1$ coefficients ($\lambda \in \mathbb{N}$).

Example:
(a): linear-phase half-band filter
(b): half-band filter with lower system delay



(From [Mertins: Signal analysis, 1999])

Filter design via spectral factorization

A given half-band filter $T(z)$ can be factorized according to

$$T(z) = F_0(z)\, H_0(z),$$

i.e. one looks for the zeros of $T(z)$, and distributes them among the polynomials $F_0(z)$ and $H_0(z)$.

The missing filters $F_1(z)$ and $H_1(z)$ can then be obtained from (5.70).
$\Rightarrow$ *General approach* for the PR design of two-channel banks

Example:
A half-band filter is given as

$$\{t(n)\} = \{-1, 0, 9, 16, 9, 0, -1\}.$$

The zeros are $\underbrace{\{3.7321, -1.0, -1.0, 0.2679\}}_{\text{zeros of } H_0(z)}$, $\underbrace{\{-1.0, -1.0\}}_{\text{zeros of } F_0(z)}$,
such that (linear-phase filter).

$$H_0(z) = \alpha(-1 + 2z^{-1} + 6z^{-2} + 2z^{-3} - z^{-4}),$$
$$F_0(z) = \beta(1 + 2z^{-1} + z^{-2})$$

Orthogonal filter banks

- In the above example: Still two filters $F_0(z)$ and $H_0(z)$ to design in order to construct the whole PR filter bank.
- In an *orthogonal* two-channel filter bank (Smith, Barnwell, 1984), (Mintzer, 1985) it suffices to design the lowpass analysis filter $H_0(z)$ for a PR system.

For an orthogonal two-channel bank the filter $H_1(z)$ is chosen as

$$H_1(z) = z^{-(L_F-1)}\, H_0(-z^{-1}), \qquad (5.75)$$

L_F denoting the length of $h_0(n)$. In the following we will only consider *even lengths* L_F. Then, using (5.70) with $\ell = 0$, the remaining synthesis filter can be obtained as

$$\hat{F}_0(z) = (-z)^{-(L_F-1)}\, H_0(z^{-1}),\ \hat{F}_1(z) = -H_0(-z).$$

Note that $(-z)^{-(L_F-1)} = (-1)^{-(L_F-1)}\, z^{-(L_F-1)} = (-1)\, z^{-(L_F-1)}$ since (L_F-1) odd.
The factor (-1) in $F_0(z)$, $F_1(z)$ can be regarded as a common factor multiplied to the output of the synthesis bank and can thus be removed for simplicity reasons:

$$F_0(z) = z^{-(L_F-1)}\, H_0(z^{-1}),\ F_1(z) = H_0(-z). \qquad (5.76)$$

Removing the factor (-1) does not change anything at the aliasing cancelation property: Inserting $F_0(z)$ and $F_1(z)$ into (5.59) still yields $G(z) = 0$.

In order to obtain the condition for a analysis-synthesis system free of linear distortions we now insert (5.75) and (5.76) into (5.58), leading to

$$S(z) = z^{-(L_F-1)}\,(H_0(z)\,H_0(z^{-1}) + H_0(-z)\,H_0(-z^{-1})),$$
$$\stackrel{!}{=} 2\,z^{-D}.$$

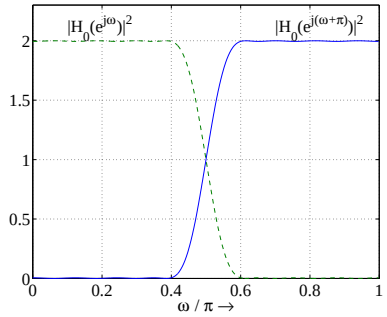
Hence, the PR condition is

$$H_0(z)\,H_0(z^{-1}) + H_0(-z)\,H_0(-z^{-1}) \stackrel{!}{=} 2. \qquad (5.77)$$

With $z = e^{j\omega}$ (5.77) can be written for real-valued $h_0(n)$ according to

$$\boxed{|H_0(e^{j\omega})|^2 + |H_0(e^{j(\omega+\pi)})|^2 = 2} \qquad (5.78)$$

$\Rightarrow$ *power-complementary* property for $H_0(e^{j\omega})$.



$\rightarrow$ It can be easily shown that the power-complementary property also holds for $h_1(n)$, $f_0(n)$, and $f_1(n)$.

Design of orthogonal filter banks

1. Search a prototype $H_0(z)$ which satisfies the power-complementary property (5.78) or (for the interesting real-valued filters) the corresponding equation (5.77) in the z-domain.

   With the abbreviation $T_Z(z) = H_0(z)\,H_0(z^{-1})$

   $$T_Z(z) + T_Z(-z) = 2 \qquad (5.79)$$

   is satisfied, where $T_Z(z)$ denotes a *zero-phase* half-band filter.

   Notation: In the following zero-phase filters and amplitude responses are denoted with the subscript "Z" instead of "0" to avoid confusion with the lowpass analysis filter $h_0(n)$.

   Valid half-band filter: $T_Z(z)$ is a valid half-band filter if it can be factorized into $H_0(z)$ and $H_0(z^{-1})$.

   $\Rightarrow$ Design goal: Find a valid half-band filter and factorize it into $H_0(z)$ and $H_0(z^{-1})$.
2. When a suitable filter $H_0(z)$ has been found, the remaining filter can be obtained from

   $$\boxed{\begin{aligned} H_1(z) &= z^{-(L_F-1)}\,H_0(-z^{-1}),\\ F_0(z) &= z^{-(L_F-1)}\,H_0(z^{-1}),\\ F_1(z) &= z^{-(L_F-1)}\,H_1(z^{-1}) = H_0(-z). \end{aligned}} \qquad (5.80)$$

   $\Rightarrow$ special case of the conditions in (5.70).

How to design a valid half-band filter?
Spectral factorization may be problematic:
With $T_Z(z) = H_0(z)\, H_0(z^{-1})$ and $z = e^{j\omega}$:

$$T_Z(e^{j\omega}) = H_0(e^{j\omega})\, H_0(e^{-j\omega}) = |H_0(e^{j\omega})|^2 \overset{!}{\geq} 0 \quad \text{for all } \omega. \tag{5.81}$$

Design approach due to Smith and Barnwell, 1986
Starting point is an arbitrary linear-phase half-band filter (for example designed with the Remez algorithm from Section 4.4.4):

$$A(e^{j\omega}) = A_Z(\omega) e^{-j\omega(L_F-1)}$$

$A_Z(\omega)$: Real-valued amplitude frequency response of the half-band filter $a(n)$, $A_Z(\omega) \bullet\!\!-\!\!\circ\, a_Z(n)$, L_F: length of $h_0(n)$.

If the value

$$\delta = \min_{\omega\in[0,2\pi]} A_Z(\omega) < 0, \tag{5.82}$$

a non-negative amplitude frequency response can be generated with

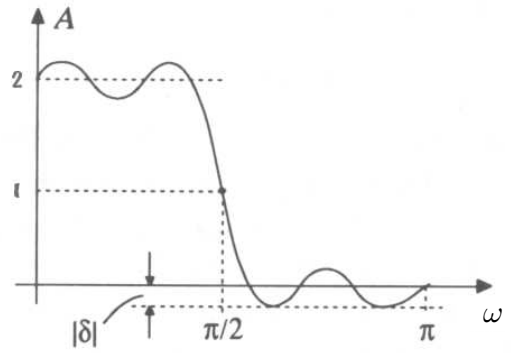
$$A_{Z+}(\omega) = A_Z(\omega) + |\delta|. \tag{5.83}$$

In time-domain this corresponds with

$$A_{Z+}(\omega) \bullet\!\!-\!\!\circ\, a_{Z+}(n) = \begin{cases} a_Z(n) & \text{for} \quad n \neq 0, \\ a_Z(n) + |\delta| & \text{for} \quad n = 0. \end{cases} \tag{5.84}$$

($a_Z(n)$ denotes the *zero-phase* impulse response with the center of symmetry located at $n = 0$).

Visualization:



(From [Fliege: Multiraten-Signalverarbeitung, 1993])

$\Rightarrow$ Single zeros on the unit-circle are converted to double zeros $\rightarrow$ factorization into two filters is now possible

A valid zero-phase half-band filter $T_Z(z)$ can finally be obtained by scaling of the resulting transfer function such that $T_Z(e^{j\pi/2}) = 1$ (note that $T_Z(z) + T_Z(-z) \overset{!}{=} 2$ has to hold), leading to the expression

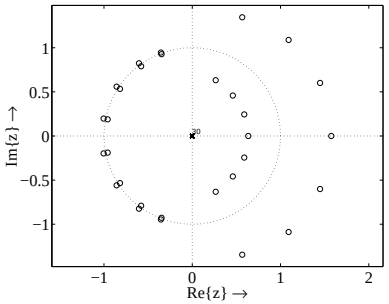
$$T_Z(z) = \frac{1}{1+|\delta|}\, A_{Z+}(z). \tag{5.85}$$

Remark:
In practice for double zeros on the unit circle the separation process is numerically very sensitive. As a solution, the parameter $|\delta|$ in (5.82) and (5.83) can be enlarged by a small value $\epsilon \rightarrow$ zeros move pairwise off the unit-circle where due to the linear-phase property they are mirrored at the unit-circle

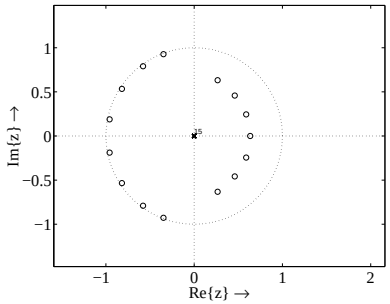
Example

Parameter for the design of $T_Z(z)$: $L_F = 16$, $\omega_s = 0.6\pi$, $\Delta\omega = 0.2\pi$, $\epsilon = 10^{-4}$

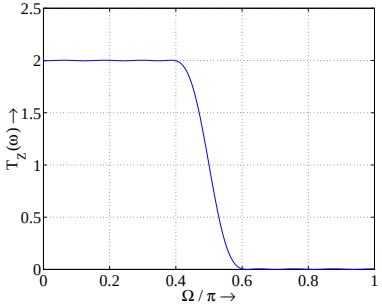
Pole-zero pattern $T_Z(z)$



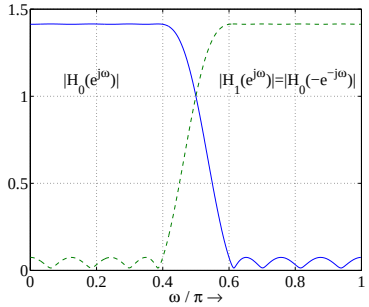
Pole-zero pattern $H_0(z)$



Amplitude frequency response $T_Z(\omega)$

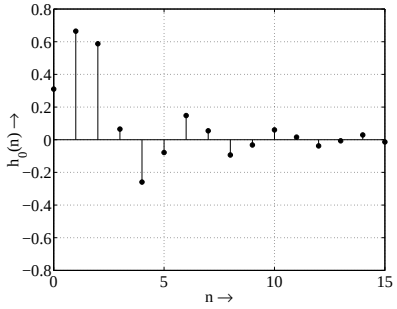


Frequency responses analysis filters

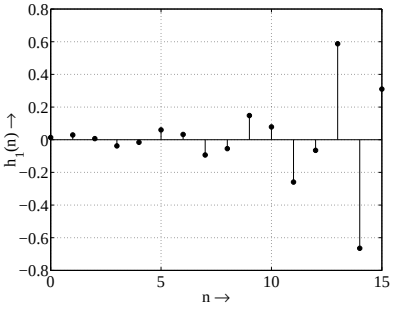


After $h_0(n)$ has been constructed, the remaining impulse responses $h_1(n)$, $f_0(n)$, $f_1(n)$ are obtained with (5.80).

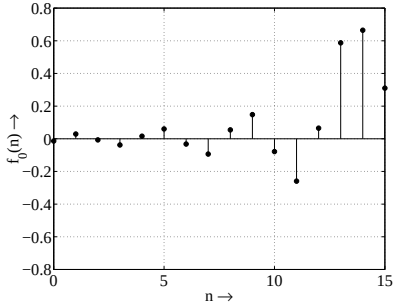
Impulse response $h_0(n)$



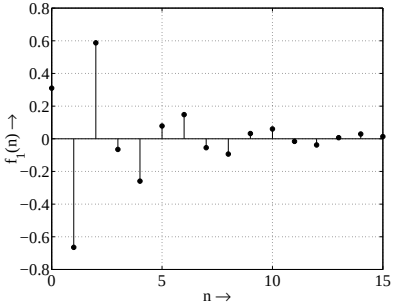
Impulse response $h_1(n)$



Impulse response $f_0(n)$

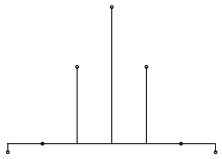


Impulse response $f_1(n)$



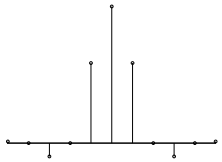
- Filter in orthogonal two-channel filter banks have an even number of coefficients since $T_Z(z) = H_0(z)\, H_0(z^{-1})$

  Example:



  $T_Z(z)$ with 7 coefficients can be factorized into two filters of length 4.

  The next feasible length is 11 coefficients

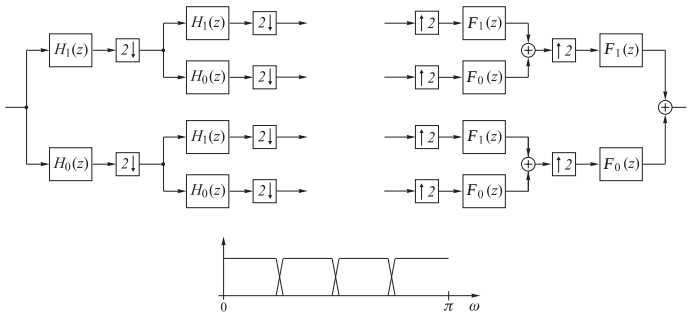


which leads to two filters of length 6.

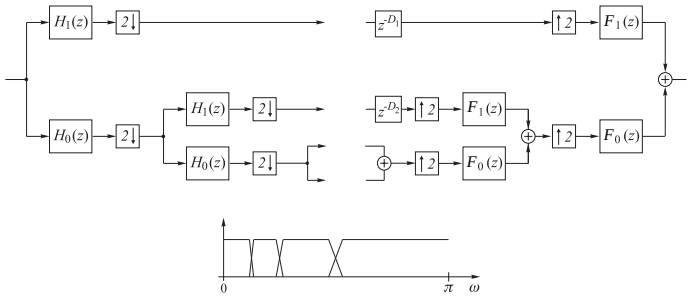
- Filter in orthogonal two-channel filter banks can not be made linear-phase (except two trivial cases) (Vaidyanathan, 1985).

5.8 Tree filter banks

K channel filter banks can be developed by iterating the two-channel systems from above.



(a)



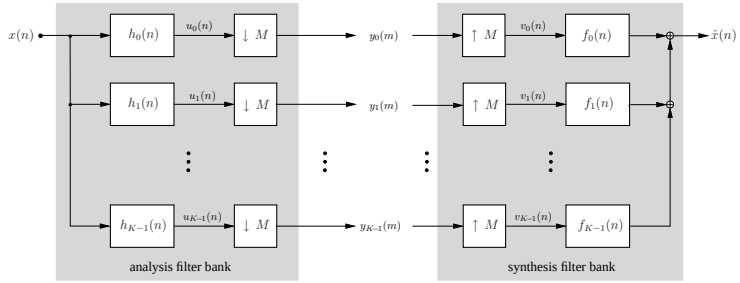
(From [Mertins: Signal analysis, 1999]) (b)

(a): Fully iterated tree filter bank, (b): Octave filter bank

If the two band filter bank is of the perfect reconstruction type, the generated multiband structure also exhibits the perfect reconstruction property.

5.9 K-band filter banks

In the following we briefly consider K-band systems with equal subband widths and equal subsampling factors in every subband.



- If $K = M$ the analysis-synthesis system is called *critically subsampled*, for $K > M$ we speak of an *oversampled* system.
- The case $K < M$ (*undersampled* system) is infeasible since the sampling theorem is violated in the subbands also for ideal (brickwall) filters $\rightarrow$ no reasonable reconstruction is possible.

Subset of general K-band filter banks:

$\rightarrow$ **Modulated filter banks**

All K analysis and synthesis filters are obtained from one single analysis and synthesis prototype, resp.

Advantages:

- More efficient implementation compared to general K-band systems.
- Less design complexity since only the prototype filters have to be designed.
- (Less storage required for the filter coefficients.)

Two important types of modulated filter banks: DFT and cosine-modulated filter banks
In the following only DFT filter banks are briefly considered.

DFT filter banks

- Analysis and synthesis filters ($k = 0, 1, \ldots, K-1$):

$$h_k(n) = \underbrace{p(n)}_{\text{analysis prototype}} \cdot \underbrace{e^{j\frac{2\pi}{K}k(n-\frac{D}{2})}}_{\text{modulation and phase shifting}} \tag{5.86}$$

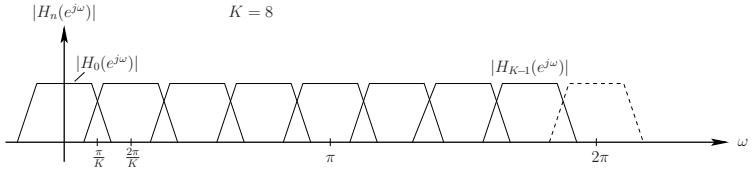
$$f_k(n) = \underbrace{q(n)}_{\text{synthesis prototype}} \cdot \underbrace{e^{j\frac{2\pi}{K}k(n-\frac{D}{2})}}_{\text{modulation and phase shifting}} \tag{5.87}$$

  D denotes the overall analysis-synthesis system delay, for linear phase filters we have $D = L_F - 1$.

- Frequency responses:

$$H_k(e^{j\omega}) = \underbrace{P(e^{j(\omega-\frac{2\pi}{K}k)})}_{\text{frequency shift by } k\,2\pi/K} \cdot \underbrace{e^{-j\frac{2\pi}{K}k\frac{D}{2}}}_{\text{phase factor}} \tag{5.88}$$

$$F_k(e^{j\omega}) = \underbrace{Q(e^{j(\omega-\frac{2\pi}{K}k)})}_{\text{frequency shift by } k\,2\pi/K} \cdot \underbrace{e^{-j\frac{2\pi}{K}k\frac{D}{2}}}_{\text{phase factor}} \tag{5.89}$$



- z-transforms:

$$H_k(z) = P(zW_K^k)W_K^{k\frac{D}{2}}, \quad F_k(z) = Q(zW_K^k)W_K^{n\frac{D}{2}} \tag{5.90}$$

- Perfect reconstruction in the critical subsampled case is only possible for filter lengths $L_F = K \rightarrow$ not very good frequency selectivity. Therefore, the DFT filter bank is mainly used with $M < K$.

Why the name *DFT* filter bank?

Type 1 polyphase components of the analysis prototype,

$$P(z) = \sum_{\ell=0}^{K-1} z^{-\ell}\, P_\ell(z^K). \tag{5.91}$$

Inserting (5.91) into (5.90) then yields

$$H_k(z) = W_K^{k\frac{D}{2}} \sum_{\ell=0}^{K-1} z^{-\ell}\, P_\ell(z^K)\, W_K^{-k\ell}. \tag{5.92}$$

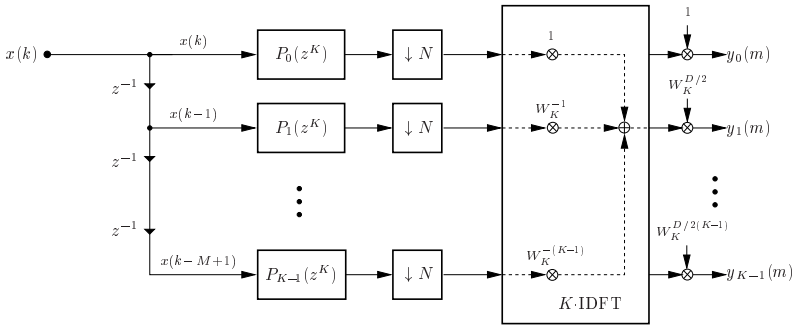
Analogy to the IDFT: $x(n) = \frac{1}{K}\sum_{k=0}^{K-1} X(k)\, W_K^{-kn}$

$\Rightarrow$ The subband filter of an DFT filter bank can be calculated with the IDFT (and thus also with the fast IFFT), where the input

signals of the IDFT are obtained by filtering with the delayed polyphase components

The n-th output of the IDFT has finally to be multiplied with the rotation factors $W_K^{n\frac{D}{2}}$.

Structure of the analysis filter bank:



(dual structure for the synthesis)

If $K = cM$, $c \in \mathbb{N}$, then besides the IDFT also the polyphase filtering can be calculated in the lower sampling rate.

6. Spectral Estimation

In this chapter we consider the problem of estimating the power spectral density (power spectrum) of a wide-sense stationary (WSS) random process.

Applications for power spectrum estimation:
Signal detection and tracking, frequency estimation (e.g. for sonar or radar signals), harmonic analysis and prediction, beamforming and direction finding,...

Problems in power spectrum estimation:

- The amount of data is generally limited, for example, a random process may be stationary only for a short time (e.g. speech signal). On the other hand, as we have seen in Section 3.1.4 for the frequency analysis of non-random signals using the DFT, the longer the input data vector, the higher the frequency resolution of the spectral analysis.
- Often, only one representation of a stochastic process may be available. Therefore, an ergodic process is often assumed.
- Additionally, the input data may be corrupted by noise.

Estimation approaches can be classified into two classes:

1. Classical or *nonparametric methods:* Estimation of the autocorrelation sequence, where the power spectrum is then obtained by Fourier transform of the autocorrelation sequence.
2. Nonclassical or *parametric* approaches, which use a model for the random process in order to estimate the power spectrum.

6.1 Periodogram-based nonparametric methods

6.1.1 The periodogram

Power spectrum $\Phi_{vv}(e^{j\omega})$ of a WSS random process $v(n)$ is the

Fourier transform of the autocorrelation function $\varphi_{vv}(\kappa)$:

$$\Phi_{vv}(e^{j\omega}) = \sum_{\kappa=-\infty}^{\infty} \varphi_{vv}(\kappa)\, e^{-j\kappa\omega}.$$

$\Rightarrow$ Spectral estimation is also an autocorrelation estimation problem

From Signals and Systems we know that for a stationary random process $v(n)$ which is ergodic in the first and second moments the autocorrelation sequence can be theoretically obtained from the time-average

$$\varphi_{vv}(\kappa) = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{k=-N}^{N} v(k+\kappa)\, v^*(k). \qquad (6.1)$$

If $v(n)$ is only measured over a finite interval of N samples, $n = 0, \ldots, N-1$, the autocorrelation sequence is estimated as

$$\hat{\varphi}_{vv}(\kappa) = \frac{1}{N} \sum_{k=0}^{N-1-\kappa} v(k+\kappa)\, v^*(k) \qquad (6.2)$$

with the values of $\hat{\varphi}_{vv}(\kappa)$ for $\kappa < 0$ defined via the symmetry relation $\hat{\varphi}_{vv}(-\kappa) = \hat{\varphi}^*_{vv}(\kappa)$, and with $\hat{\varphi}_{vv}(\kappa) = 0$ for $|\kappa| \geq N$.

The discrete Fourier transform of $\hat{\varphi}_{vv}(\kappa)$ from (6.2) is called *periodogram* (Schuster, 1898) and leads to an estimate of the power spectrum according to

$$\hat{\Phi}^{(\text{per})}_{vv}(e^{j\omega}) = \sum_{\kappa=-N+1}^{N-1} \hat{\varphi}_{vv}(\kappa) e^{-j\kappa\omega}. \qquad (6.3)$$

With the rectangular window

$$w_r(n) = \begin{cases} 1 & n = 0, \ldots, N-1, \\ 0 & \text{otherwise,} \end{cases}$$

we can describe the input sequence being analyzed also as

$$v_N(n) = v(n) \cdot w_r(n). \qquad (6.4)$$

Then, the estimated autocorrelation sequence may be written as

$$\hat{\varphi}_{vv}(\kappa) = \frac{1}{N} \sum_{k=-\infty}^{\infty} v_N(k+\kappa)\, v_N^*(k) = \frac{1}{N}\, v_N(\kappa) * v_N^*(-\kappa). \qquad (6.5)$$

Fourier transform of the rightmost equation in (6.5) finally yields the following expression for the periodogram ($V_N(e^{j\omega}) \bullet\!\!-\!\!\circ\, v_N(n)$):

$$\boxed{\hat{\Phi}^{(\text{per})}_{vv}(e^{j\omega}) = \frac{1}{N}\, V_N(e^{j\omega})\, V_N^*(e^{j\omega}) = \frac{1}{N}\, |V_N(e^{j\omega})|^2.} \qquad (6.6)$$

MATLAB-command: `periodogram`

The sampled version $\hat{\Phi}^{(\text{per})}_{vv}(e^{jk2\pi/M}) = \hat{\Phi}^{(\text{per})}_{vv}(k)$ can be obtained from the M_1-point DFT $V_N(k)$ of $v_N(n)$:

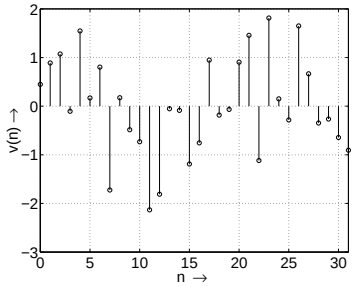
$$\hat{\Phi}^{(\text{per})}_{vv}(k) = \frac{1}{N}\, |V_N(k)|^2, \quad k = 0, \ldots, M_1 - 1. \qquad (6.7)$$

Thus, the periodogram can be calculated very efficiently using the FFT.

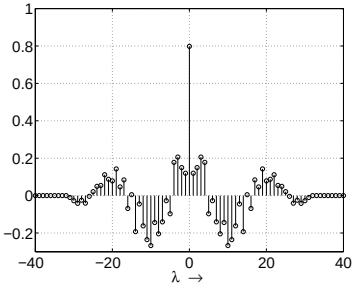
Example: Periodogram of white noise

If $v(n)$ is white noise with a variance of σ_v^2, then $\varphi_{vv}(\kappa) = \sigma_v^2\,\delta(\kappa)$ with a constant power spectrum $\Phi_{vv}(e^{j\omega}) = \sigma_v^2$.

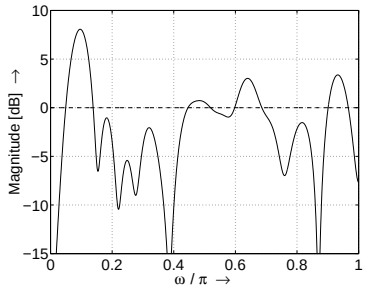
Sample realization for $N = 32$



Autocorrelation sequence $\hat{\varphi}_{vv}(\kappa)$



Periodogram $\hat{\Phi}_{vv}^{(\mathrm{per})}(e^{j\omega})$ (solid), power spectrum $\Phi_{vv}(e^{j\omega})$ (dashed)



Definition: Bias and consistency

Desirable:

- Convergence of the periodogram to the exact power spectrum in the mean-square sense

$$\lim_{N\to\infty} E\left\{\left(\hat{\Phi}_{vv}^{(\mathrm{per})}(e^{j\omega}) - \Phi_{vv}(e^{j\omega})\right)^2\right\} \stackrel{!}{=} 0. \qquad (6.8)$$

- In order to achieve this it is necessary that the periodogram is *asymptotically unbiased*, which means that for $N \to \infty$ the expectation value of the estimated power spectrum is equal to the true power spectrum:

$$\lim_{N\to\infty} E\left\{\hat{\Phi}_{vv}^{(\mathrm{per})}(e^{j\omega})\right\} \stackrel{!}{=} \Phi_{vv}(e^{j\omega}). \qquad (6.9)$$

  On the other hand for a *biased* estimator there would be a difference between the expectation value and the true result.

- Furthermore, the estimation variance should go to zero as the data length N goes to infinity:

$$\lim_{N\to\infty} \mathrm{Var}\left\{\hat{\Phi}_{vv}^{(\mathrm{per})}(e^{j\omega})\right\} \stackrel{!}{=} 0. \qquad (6.10)$$

- If (6.9) and (6.10) are satisfied we say that the periodogram $\hat{\Phi}_{vv}^{(\mathrm{per})}(e^{j\omega})$ is a *consistent* estimate of the power spectrum.

  Note that there are different definitions of consistency in the literature.

Bias of the periodogram

First step: Calculation of the expected value of the autocorrelation $\hat{\varphi}_{vv}(\kappa)$. From (6.2) we have

$$E\{\hat{\varphi}_{vv}(\kappa)\} = \frac{1}{N}\sum_{k=0}^{N-1-\kappa} E\{v(k+\kappa)\,v^*(k)\}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1-\kappa} \varphi_{vv}(\kappa) = \frac{N-\kappa}{N}\,\varphi_{vv}(\kappa) \quad (6.11)$$

for $\kappa = 0, \ldots, N-1$, and for $\kappa \geq N$ it follows $E\{\hat{\varphi}_{vv}(\kappa)\} = 0$.

By using the symmetry relation $\hat{\varphi}_{vv}(-\kappa) = \hat{\varphi}_{vv}^*(\kappa)$ (6.11) can be written as $E\{\hat{\varphi}_{vv}(\kappa)\} = w_B(\kappa)\,\varphi_{vv}(\kappa)$ with the Bartlett

(triangular) window

$$w_B(\kappa) = \begin{cases} \frac{N-|\kappa|}{N} & \text{for } |\kappa| \leq N, \\ 0 & \text{for } |\kappa| > N. \end{cases} \tag{6.12}$$

The expected value of the periodogram can now be obtained as

$$\begin{aligned} E\left\{\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})\right\} &= \sum_{\kappa=-N+1}^{N-1} E\{\hat{\varphi}_{vv}(\kappa)\} e^{-j\kappa\omega}, \\ &= \sum_{\kappa=-\infty}^{\infty} w_B(\kappa)\, \varphi_{vv}(\kappa)\, e^{-j\kappa\omega}, \end{aligned}$$

which finally yields

$$\boxed{E\left\{\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})\right\} = \frac{1}{2\pi}\, \Phi_{vv}(e^{j\omega}) \circledast W_B(e^{j\omega})} \tag{6.13}$$

with $W_B(e^{j\omega})$ denoting the Fourier transform of the Bartlett window

$$W_B(e^{j\omega}) = \frac{1}{N}\left(\frac{\sin(N\omega/2)}{\sin(\omega/2)}\right)^2.$$

$\Rightarrow$ Periodogram is a *biased estimate*, since the expected value is the convolution between the true power spectrum and the Fourier transform of the Bartlett window.

Since $W_B(e^{j\omega})$ converges to an impulse for $N \to \infty$ the periodogram is *asymptotically unbiased*:

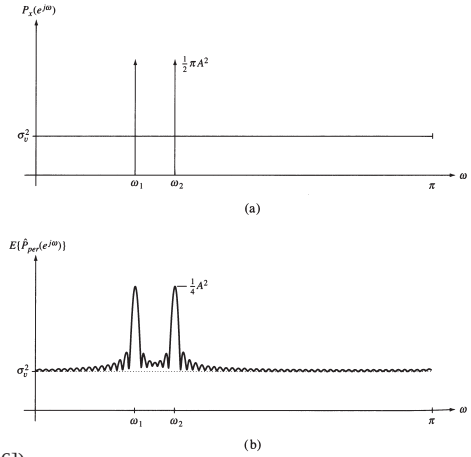
$$\boxed{\lim_{N\to\infty} E\left\{\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})\right\} = \Phi_{vv}(e^{j\omega})} \tag{6.14}$$

Spectral resolution

We know from the discussion in Section 3.1.4 that the convolution with the frequency response of a window may lead to

- spectral smoothing,
- spectral leaking,
- the loss of the ability to resolve two nearby spectral lines.

Similarly, this also holds for the convolution between the power spectrum and the Bartlett window frequency response in (6.13).
Example: (a) Power spectrum of two sinusoids in white noise, (b) expected value of the periodogram



(from [Hayes, 1996])

- Width of the main lobe of $W_B(e^{j\omega})$ increases as the data record length decreases.
- $\Rightarrow$ For a given length N there is a limit on how closely two sinusoids or narrowband processes may be located before they no longer can be resolved.

- One way to define this frequency resolution limit is to set $\Delta\omega$ equal to the width of the main lobe of the Bartlett window at its -6 dB point:

$$\boxed{\Delta\omega = 0.89\,\frac{2\pi}{N},} \tag{6.15}$$

which is also the *frequency resolution of the periodogram*.

Variance of the periodogram

White Gaussian random processes:
It can be shown that for a white Gaussian random process $v(n)$ the variance of the periodogram is equal to the square of the power spectrum $\Phi_{vv}(e^{j\omega})$ (see [Hayes, 1996]):

$$\text{Var}\left\{\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})\right\} = \Phi_{vv}^2(e^{j\omega}). \tag{6.16}$$

Non-white Gaussian random processes:

For non-white Gaussian processes, which are more important in practice, we derive an approximation for the variance of the periodogram in the following.

A random process $v(n)$ with power spectrum $\Phi_{vv}(e^{j\omega})$ may be generated by filtering white noise $x(n)$ with variance $\sigma_x^2 = 1$ with a linear filter $h(n) \circ\!\!-\!\!\bullet\, H(e^{j\omega})$ and

$$|H(e^{j\omega})|^2 = \Phi_{vv}(e^{j\omega}). \tag{6.17}$$

The sequences $v_N(n)$ and $x_N(n)$ are now formed by windowing

analog to (6.4). The periodograms of these processes are

$$\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega}) = \frac{1}{N}|V_N(e^{j\omega})|^2, \quad \hat{\Phi}_{xx}^{(\text{per})}(e^{j\omega}) = \frac{1}{N}|X_N(e^{j\omega})|^2. \tag{6.18}$$

If N is large compared to the length of $h(n)$, $v_N(n)$ can be described as $v_N(n) \approx h(n) * x_N(n)$, since the transition effects can be neglected. Thus, the magnitude square frequency response $|V_N(e^{j\omega})|^2$ of $v_N(n)$ can be with (6.17) expressed as

$$|V_N(e^{j\omega})|^2 \approx |H(e^{j\omega})|^2\,|X_N(e^{j\omega})|^2 = \Phi_{vv}(e^{j\omega})\,|X_N(e^{j\omega})|^2. \tag{6.19}$$

Inserting (6.18) into (6.19) yields

$$\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega}) \approx \Phi_{vv}(e^{j\omega})\,\hat{\Phi}_{xx}^{(\text{per})}(e^{j\omega}).$$

Applying the variances on both sides results in

$$\text{Var}\left\{\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})\right\} \approx \Phi_{vv}^2(e^{j\omega})\,\text{Var}\left\{\hat{\Phi}_{xx}^{(\text{per})}(e^{j\omega})\right\},$$

and, since $\text{Var}\{\hat{\Phi}_{xx}^{(\text{per})}(e^{j\omega})\} = 1$ according to (6.16), the variance for large N can be obtained as

$$\boxed{\text{Var}\left\{\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})\right\} \approx \Phi_{vv}^2(e^{j\omega}).} \tag{6.20}$$

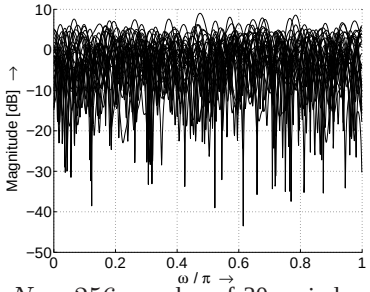
$\Rightarrow$ Periodogram is not a consistent estimator

Example:
For a white Gaussian noise process $v(n)$ with $\sigma_v^2 = 1$ and $\Phi_{vv}(e^{j\omega}) = 1$ it follows from (6.13) and (6.16), resp., that

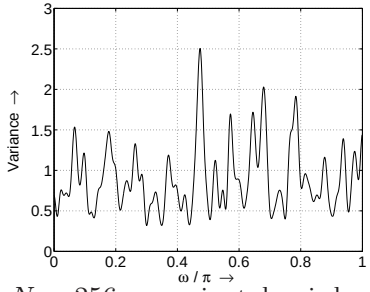
$$E\left\{\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})\right\} = 1 \quad \text{and} \quad \text{Var}\left\{\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})\right\} = 1.$$

Thus, although the periodogram for white Gaussian noise is unbiased, the variance is independent of the data record length N.

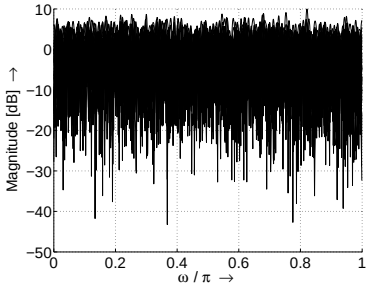
$N = 64$, overlay of 30 periodograms $\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})$



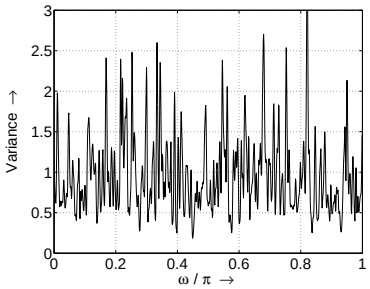
$N = 64$, approximated periodogram variance



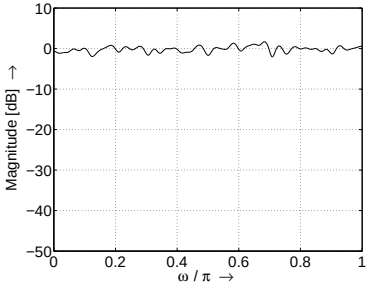
$N = 256$, overlay of 30 periodograms $\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})$



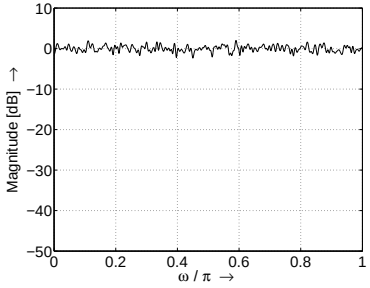
$N = 256$, approximated periodogram variance



$N = 64$, periodogram average



$N = 256$, periodogram average



6.1.2 Bartlett's method: Periodogram averaging

In contrast to the periodogram, Bartlett's method (1948) provides a *consistent estimate* of the power spectrum.

The motivation for this method comes from the fact that by averaging a set of uncorrelated measurements for a random variable v one obtains a consistent estimate of the mean $E\{v\}$. Since the periodogram is asymptotically unbiased

$$\lim_{N\to\infty} E\left\{\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})\right\} = \Phi_{vv}(e^{j\omega}),$$

it obviously suffices to find a consistent estimate of the periodogram $E\{\hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega})\}$ in order to find a consistent estimate for the true power spectrum $\Phi_{vv}(e^{j\omega})$.

$\Rightarrow$ Estimation of the power spectrum by periodogram averaging!

Let $v_i(n)$ for $i = 0, \ldots, K-1$ denote K uncorrelated realizations of a random process $v(n)$ for $n = 0, \ldots, L-1$. The periodogram of each single realization is obtained from (6.6) as

$$\hat{\Phi}_{v_i v_i}^{(\text{per})}(e^{j\omega}) = \frac{1}{L}\left|\sum_{n=0}^{L-1} v_i(n) e^{-jn\omega}\right|^2 . \qquad (6.21)$$

The average of these periodograms is

$$\hat{\Phi}_{vv}(e^{j\omega}) = \frac{1}{K}\sum_{i=0}^{K-1} \hat{\Phi}_{v_i v_i}^{(\text{per})}(e^{j\omega}). \qquad (6.22)$$

For the expected value of $\hat{\Phi}_{vv}(e^{j\omega})$ we have with (6.22) and

(6.13)

$$E\left\{\hat{\Phi}_{vv}(e^{j\omega})\right\} = E\left\{\hat{\Phi}^{(\text{per})}_{v_i v_i}(e^{j\omega})\right\}$$
$$= \frac{1}{2\pi}\,\Phi_{vv}(e^{j\omega}) \circledast W_B(e^{j\omega}). \qquad (6.23)$$

As with the periodogram, the estimate $\hat{\Phi}_{vv}(e^{j\omega})$ is asymptotically unbiased, i.e. for $L \to \infty$.

For uncorrelated data records $v_i(n)$ the variance Var$\{\hat{\Phi}_{vv}(e^{j\omega})\}$ can be obtained in the same way from (6.22) and (6.20) as

$$\text{Var}\left\{\hat{\Phi}_{vv}(e^{j\omega})\right\} = \frac{1}{K}\,\text{Var}\left\{\hat{\Phi}^{(\text{per})}_{v_i v_i}(e^{j\omega})\right\} \approx \frac{1}{K}\,\Phi^2_{vv}(e^{j\omega}). \qquad (6.24)$$

We can observe that the variance goes to zero if K goes to infinity $\Rightarrow \hat{\Phi}_{vv}(e^{j\omega})$ is a *consistent estimate* of the power spectrum if both L and K tend to infinity.

In practice:
Uncorrelated realizations of a random process are generally not available, instead we have only one single realization of length N.
Alternative: $v(n)$ of length N is divided into K nonoverlapping sequences of length L with $N = L \cdot K$, that is $v_i(n) = v(n + iL)$, $n = 0, \ldots, L-1$, $i = 0, \ldots, K-1$.

Thus, the Bartlett estimate is

$$\boxed{\hat{\Phi}^{(B)}_{vv}(e^{j\omega}) = \frac{1}{N}\sum_{i=0}^{K-1}\left|\sum_{n=0}^{L-1} v(n+iL)\,e^{-jn\omega}\right|^2.} \qquad (6.25)$$

Properties

From (6.23) the expected value is

$$\boxed{E\left\{\hat{\Phi}^{(B)}_{vv}(e^{j\omega})\right\} = \frac{1}{2\pi}\,\Phi_{vv}(e^{j\omega}) \circledast W_B(e^{j\omega})} \qquad (6.26)$$

As the periodogram the Bartlett estimate is *asymptotically unbiased*.

The *spectral resolution* of the Bartlett estimate can be obtained from the resolution of the periodogram in (6.15). Since we now use sequences of length L the resolution becomes

$$\boxed{\Delta\omega = 0.89\,\frac{2\pi}{L} = 0.89\,K\,\frac{2\pi}{N},} \qquad (6.27)$$

which is K times larger (worse!) than that of the periodogram.

Variance: Assuming that the data sequences $v_i(n)$ are approximately uncorrelated (this is generally not the case!) the variance of the Bartlett estimate is for large N

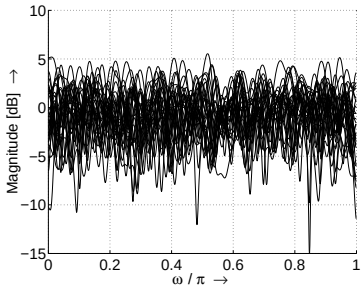
$$\boxed{\text{Var}\left\{\hat{\Phi}^{(B)}_{vv}(e^{j\omega})\right\} \approx \frac{1}{K}\,\Phi^2_{vv}(e^{j\omega}).} \qquad (6.28)$$

- $\Phi^{(B)}_{vv}(e^{j\omega})$ is a consistent estimate for $K, L \to \infty$.
- The Bartlett estimate allows to trade spectral resolution for a reduction in variance by adjusting the parameters L and K.

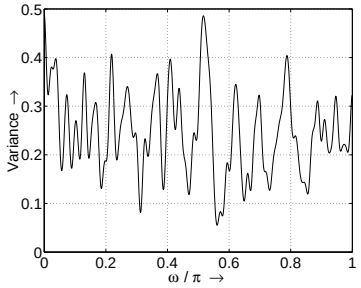
Examples:

- The power spectrum of a white noise Gaussian process with $\sigma_v^2 = 1$ of length $N = 256$ is estimated with Bartlett's method.

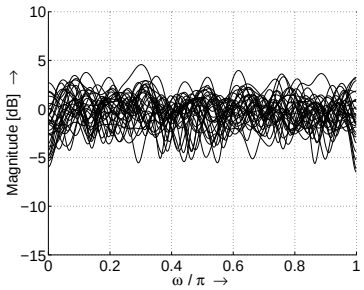
$K = 4$, $L = 64$, overlay of 30 Bartlett estimates $\hat{\Phi}_{vv}^{(B)}(e^{j\omega})$



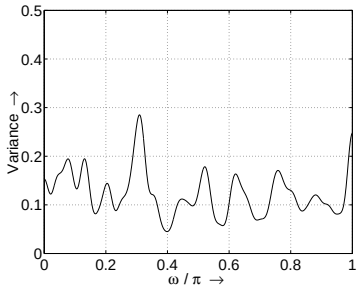
$K = 4$, $L = 64$, approximated variance of Bartlett estimates



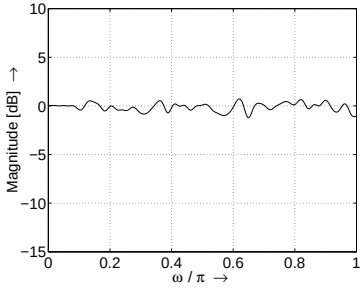
$K = 8$, $L = 32$, overlay of 30 Bartlett estimates $\hat{\Phi}_{vv}^{(B)}(e^{j\omega})$



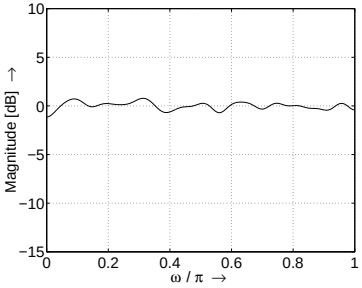
$K = 8$, $L = 32$, approximated variance of Bartlett estimates



$K = 4$, $L = 64$, Bartlett estimate average



$K = 8$, $L = 32$, Bartlett estimate average



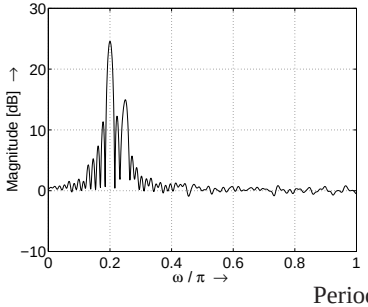
- Here, the input signal consists of two sinusoids in white Gaussian noise

$\eta(n)$ of variance $\sigma_\eta^2 = 1$,

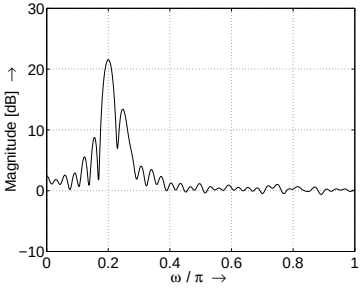
$$v(n) = 3 \cdot \sin(n\omega_1) + \sin(n\omega_2) + \eta(n) \tag{6.29}$$

with $\omega_1 = 0.2\pi$, $\omega_2 = 0.25\pi$, and length $N = 512$ samples.
The following figures show the average power spectrum estimate over 30 realizations, and demonstrate the reduced spectral resolution of the Bartlett estimate compared to the periodogram.

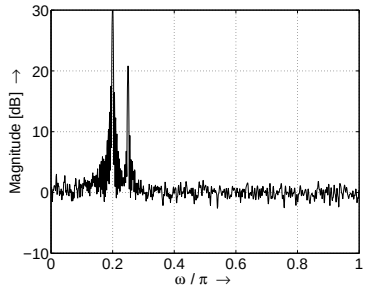
$K = 4$, $L = 128$, Bartlett estimate



$K = 8$, $L = 64$, Bartlett estimate



Periodogram



6.1.3 Welch's method: Averaging modified periodograms

In 1967, Welch proposed two modifications to Bartlett's method:

1. The data segments $v_i(n)$ of length L are allowed to overlap, where D denotes the offset between successive sequences:

$$v_i(n) = v(n+iD),\ n = 0, \ldots, L-1,\ i = 0, \ldots, K-1. \tag{6.30}$$

The amount of overlap between $v_i(n)$ and $v_{i+1}(n)$ is $L-D$ samples, and if K sequences cover the entire N data points we have $N = L + D(K-1)$. If $D = L$ the segments do not overlap as in Bartlett's method with $K = N/L$.

$\Rightarrow$ By allowing the sequences to overlap it is possible to increase the number and/or length of the sequences that are averaged. Reduction of the variance (for larger K) can thus be traded in for a reduction in resolution (for smaller L) and vice versa.

2. The second modification is to window the data segments prior to computing the periodogram. This leads to a so called *modified periodogram*

$$\hat{\Phi}_{v_i v_i}^{(\text{mod})}(e^{j\omega}) = \frac{1}{LU}\left|\sum_{n=0}^{L-1} v_i(n)\, w(n) e^{-jn\omega}\right|^2 \qquad (6.31)$$

with a general window $w(n)$ of length L, and U denoting a normalization factor for the power in the window function according to

$$U = \frac{1}{L}\sum_{n=0}^{L-1} |w(n)|^2. \qquad (6.32)$$

Welch's method may explicitly be written as

$$\hat{\Phi}_{vv}^{(W)}(e^{j\omega}) = \frac{1}{KLU}\sum_{i=0}^{K-1}\left|\sum_{n=0}^{L-1} v(n+iD)\, w(n) e^{-jn\omega}\right|^2. \qquad (6.33)$$

MATLAB-command: `pwelch`

Properties

- It can be shown that the expected value of Welch's estimate is

$$E\left\{\hat{\Phi}_{vv}^{(W)}(e^{j\omega})\right\} = \frac{1}{2\pi LU}\,\Phi_{vv}(e^{j\omega}) \circledast |W(e^{j\omega})|^2, \qquad (6.34)$$

  where $W(e^{j\omega})$ denotes the Fourier transform of the general L-point window sequence $w(n)$. Thus, Welch's method is an *asymptotically unbiased* estimate of the power spectrum.

- The *spectral resolution* of Welch's estimate depends on the used window sequence and is specified as the 3 dB width $\Delta\omega_{3\,\text{dB}}$ of the main lobe of the spectral window. $\Delta\omega_{3\,\text{dB}}$ is specified for some commonly used windows in the following table.

Type of window	Sidelobe level [dB]	3 dB bandwidth $\Delta\omega_{3\,\text{dB}}$
Rectangular	-13	$0.89\,\frac{2\pi}{L}$
Bartlett	-27	$1.28\,\frac{2\pi}{L}$
Hanning	-32	$1.44\,\frac{2\pi}{L}$
Hamming	-43	$1.30\,\frac{2\pi}{L}$
Blackman	-58	$1.68\,\frac{2\pi}{L}$

Remark: In (6.15) we stated the frequency resolution of the peridogram as the 6 dB main lobe width of the Bartlett window. Since $W_B(e^{j\omega}) = |W_R(e^{j\omega})|^2$ this is equivalent to the 3 dB bandwidth of the frequency response $W_R(e^{j\omega})$ of the rectangular window.

- The variance of Welch's estimate highly depends on the

amount of overlapping. For a Bartlett window and a 50% overlap the variance is approximately

$$\text{Var}\left\{\hat{\Phi}_{vv}^{(W)}(e^{j\omega})\right\} \approx \frac{9}{8K}\,\Phi_{vv}^{2}(e^{j\omega}) \qquad (6.35)$$

($\longrightarrow$ consistent estimate). A comparison with (6.28) shows that the variance for Welch's method seems to be larger than for Bartlett's method. However, for fixed amount of data N and a fixed resolution L here twice as many sections are averaged compared to Bartlett's method. With $K = 2N/L$ (50% overlap) (6.35) becomes

$$\boxed{\text{Var}\left\{\hat{\Phi}_{vv}^{(W)}(e^{j\omega})\right\} \approx \frac{9L}{16N}\,\Phi_{vv}^{2}(e^{j\omega}).} \qquad (6.36)$$

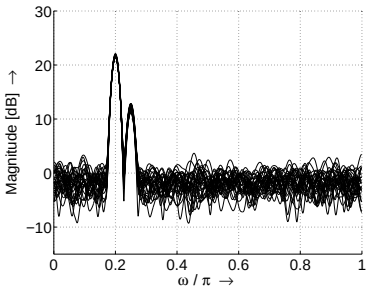
A comparison with (6.28) and $K = N/L$ for the Bartlett estimate we have

$$\text{Var}\left\{\hat{\Phi}_{vv}^{(W)}(e^{j\omega})\right\} \approx \frac{9}{16}\,\text{Var}\left\{\hat{\Phi}_{vv}^{(B)}(e^{j\omega})\right\}. \qquad (6.37)$$

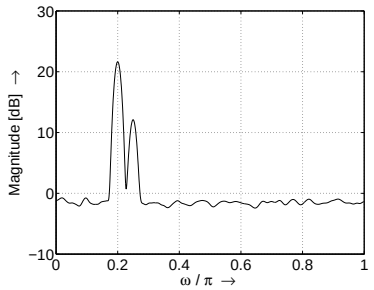
Increasing the amount of overlap yields higher computational complexity and also the correlation between the subsequences $v_i(n) \longrightarrow$ amount of overlap is typically chosen as 50% or 75%.

Example:
As an input signal we again use (6.29) which contains two sinusoids in white Gaussian noise $\eta(n)$ of variance $\sigma_\eta^2 = 1$, with $\omega_1 = 0.2\pi$, $\omega_2 = 0.25\pi$, and a signal length of $N = 512$ samples. The section length is chosen as $L = 128$, the amount of overlapping is 50%, and for the window we use a Hamming window.

Overlay plot of 30 Welch estimates



Welch estimate ensemble average



Compared to the Bartlett estimate for the same example above the use of the Hamming window reduces the spectral leakage in the estimated power spectrum.

Since the number of sections (7) are about the same to those in the above example for the Bartlett estimate with $K = 8$, $L = 64$ (8 sections) both variances are also approximately the same.

6.1.4 Blackman-Tukey method: Periodogram smoothing

Recall that the periodogram is obtained by a Fourier transform from the estimated autocorrelation sequence. However, for any finite data record of length N the variance of $\hat{\varphi}_{vv}(\kappa)$ will be large for values of κ, which are close to N. For example for lag $\kappa = N - 1$ we have from (6.2)

$$\hat{\varphi}_{vv}(N-1) = \frac{1}{N}\,v(N-1)\,v(0).$$

Two approaches for reducing the variance of $\hat{\varphi}_{vv}(\kappa)$ and thus also the variance of the peridogram:

1. Averaging periodograms and modified periodograms, resp., as utilized in the methods of Bartlett and Welch.
2. Periodogram smoothing $\longrightarrow$ Blackman-Tukey method (1958)

Blackman-Tukey method: Variance of the autocorrelation function is reduced by applying a window to $\hat{\varphi}_{vv}(\kappa)$ to decrease the contribution of the undesired estimates to the periodogram.

The Blackman-Tukey estimate is given as

$$\hat{\Phi}_{vv}^{(BT)}(e^{j\omega}) = \sum_{\kappa=-M}^{M} \hat{\varphi}_{vv}(\kappa)\, w(\kappa)\, e^{-j\kappa\omega}, \tag{6.38}$$

where $w(\kappa)$ is a *lag window* being applied to the autocorrelation estimate and extending from $-M$ to M for $M < N-1$.
$\Rightarrow$ Estimates of $\varphi_{vv}(\kappa)$ having the largest variance are set to zero by the lag window $\rightarrow$ the power spectrum estimate will have a smaller variance.

The Blackman-Tukey power spectrum estimate from (6.38) may also be written as

$$\hat{\Phi}_{vv}^{(BT)}(e^{j\omega}) = \frac{1}{2\pi}\, \hat{\Phi}_{vv}^{(\text{per})}(e^{j\omega}) \circledast W(e^{j\omega}), \tag{6.39}$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{\Phi}_{vv}^{(\text{per})}(e^{ju})\, W(e^{j(\omega-u)})\, du \tag{6.40}$$

with $W(e^{j\omega})$ denoting the Fourier transform of the lag window.
$\Rightarrow$ Blackman-Tukey estimate smooths the periodogram by convolution with $W(e^{j\omega})$.

Choice of a suitable window:

- $w(\kappa)$ should be conjugate symmetric, such that $W(e^{j\omega})$ (and also the power spectrum) is real-valued.

- $W(e^{j\omega}) \geq 0$, such that $\hat{\Phi}_{vv}^{(BT)}(e^{j\omega})$ is nonnegative for all ω. Note that some of the window functions we have introduced do not satisfy this condition, for example, the Hamming and Hanning windows.

Properties

- The expected value of the Blackman-Tukey estimate can be derived for $M \ll N$ as

$$E\left\{\hat{\Phi}_{vv}^{(BT)}(e^{j\omega})\right\} = \frac{1}{2\pi}\, \Phi_{vv}(e^{j\omega}) \circledast W(e^{j\omega}) \tag{6.41}$$

  where $W(e^{j\omega})$ is the Fourier transform of the lag window.
- The spectral resolution of the Blackman-Tukey estimate depends on the used window.
- It can be shown that the variance can be approximated as

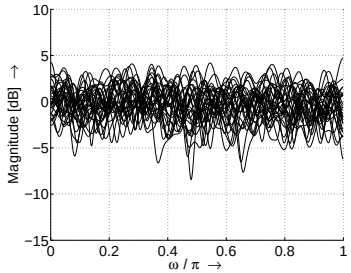
$$\text{Var}\left\{\hat{\Phi}_{vv}^{(BT)}(e^{j\omega})\right\} \approx \Phi_{vv}^{2}(e^{j\omega}) \frac{1}{N} \sum_{\kappa=-M}^{M} w^{2}(\kappa). \tag{6.42}$$

- From (6.41) and (6.42) we again see the trade-off between bias and variance: For a small bias, M should be large in order to minimize the width of the main lobe of $W(e^{j\omega})$, whereas M should be small to minimize the sum term in (6.42). As a general rule of thumb, M is often chosen as $M = N/5$.

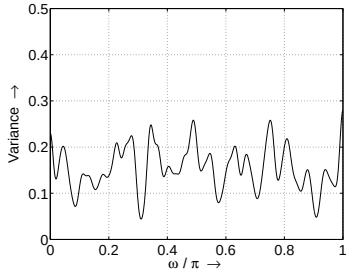
Examples:

- The power spectrum of a white noise Gaussian process with $\sigma_v^2 = 1$ of length $N = 256$ is estimated with the Blackman-Tukey method, where a Bartlett window with $M = 51$ is used.

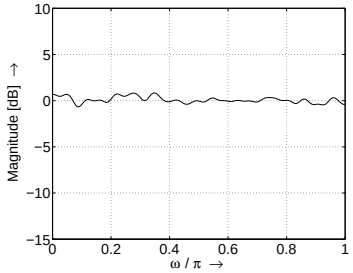
Overlay of 30 Blackman-Tukey estimates $\hat{\Phi}_{vv}^{(BT)}(e^{j\omega})$



Approximated variance of the Blackman-Tukey estimates

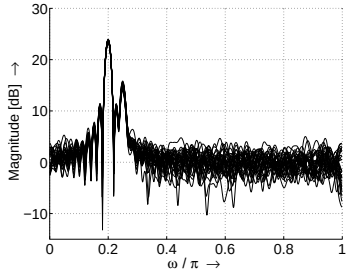


Blackman-Tukey estimate ensemble average

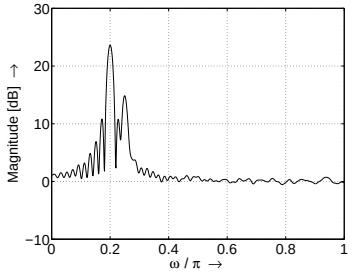


- In a second example we use two sinusoids in white Gaussian noise ((6.29), $\sigma_n^2 = 1$, $\omega_1 = 0.2\pi$, $\omega_2 = 0.25\pi$) for $N = 512$ samples. The window is a Bartlett window with $M = 102$.

Overlay plot of 30 Blackman-Tukey estimates



Blackman-Tukey estimate ensemble average



6.1.5 Performance comparisons

Performance of the discussed estimators is assessed in terms of two criteria:

1. *Variability* $\mathcal{V}$ of the estimate,

$$\mathcal{V} = \frac{\mathrm{Var}\left\{\hat{\Phi}_{vv}(e^{j\omega})\right\}}{\mathrm{E}^2\left\{\hat{\Phi}_{vv}(e^{j\omega})\right\}}, \tag{6.43}$$

   which can be regarded as a normalized variance.

2. Overall *figure of merit*, which is defined as the product of the variability and the spectral resolution $\Delta\omega$,

$$\mathcal{M} = \mathcal{V}\,\Delta\omega. \tag{6.44}$$

Results for the periodogram-based spectrum estimation techniques:

	Variability $\mathcal{V}$	Resolution $\Delta\omega$	Figure of merit $\mathcal{M}$
Periodogram	1	$0.89\,\frac{2\pi}{N}$	$0.89\,\frac{2\pi}{N}$
Bartlett	$\frac{1}{K}$	$0.89\,K\,\frac{2\pi}{N}$	$0.89\,\frac{2\pi}{N}$
Welch (50% overlap, Bartlett window)	$\frac{9}{8\,K}$	$1.28\,\frac{2\pi}{L}$	$0.72\,\frac{2\pi}{N}$
Blackman-Tukey (Bartlett window of length $2M$, $1 \ll M \ll N$)	$\frac{2\,M}{3\,N}$	$0.64\,\frac{2\pi}{M}$	$0.43\,\frac{2\pi}{N}$

- Each technique has a figure of merit being approximately the same, figures of merit are inversely proportional to the length N of the data sequence.
- $\Rightarrow$ Overall performance is fundamentally limited by the amount of data being available!

6.2 Parametric methods for power spectrum estimation

Disadvantages of the periodogram-based (nonparametric) methods:

- Long data records are required for sufficient performance, windowing of the autocorrelation sequence limits spectral resolution.
- Spectral leakage effects due to windowing.
- A-priori information about the generating process is not exploited.

Disadvantages are removed by using *parametric* estimation approaches, where an appropriate model for the input process is applied.

Parametric methods are based on modeling the data sequence as the output of a linear system with the transfer function (IIR filter!)

$$H(z) = \frac{\sum_{i=0}^{q} b_i \, z^{-i}}{1 + \sum_{i=1}^{p} a_i \, z^{-i}}, \qquad (6.45)$$

where the a_i and b_i are the model parameters. The corresponding

difference equation is

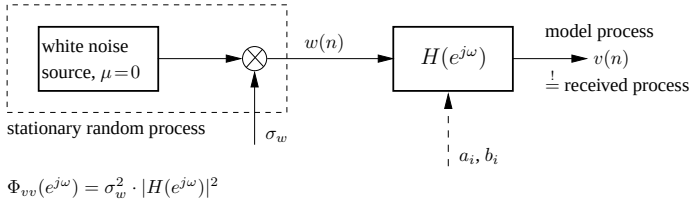
$$v(n) = \sum_{i=0}^{q} b_i \, w(n-i) - \sum_{i=1}^{p} a_i \, v(n-i), \qquad (6.46)$$

where $v(n)$ denotes the output and $w(n)$ the input sequence

If $w(n)$ represents a stationary random process, then $v(n)$ is also stationary and the power spectrum can be given as (Wiener-Lee relation)

$$\Phi_{vv}(e^{j\omega}) = |H(e^{j\omega})|^2 \, \Phi_{ww}(e^{j\omega}). \qquad (6.47)$$

Parametric power spectrum estimation:



In order to estimate the power spectrum $\Phi_{vv}(e^{j\omega})$ we assume in our model that $\Phi_{ww}(e^{j\omega})$ comes from a zero-mean white noise process with variance σ_w^2. By inserting $\Phi_{ww}(e^{j\omega}) = \sigma_w^2$ into (6.47) the power spectrum of the observed data is

$$\Phi_{vv}(e^{j\omega}) = \sigma_w^2 \, |H(e^{j\omega})|^2 = \sigma_w^2 \, \frac{|B(e^{j\omega})|^2}{|A(e^{j\omega})|^2}. \qquad (6.48)$$

Goal: Make the model process $v(n)$ as similar as possible to the unknown received process in the mean-square error sense by adjusting the parameters a_i, b_i, and σ_w $\Rightarrow$ the power spectrum $\Phi_{vv}(e^{j\omega})$ can then be obtained via (6.48).

In the following we distinguish among three specific cases for $H(z)$ leading to three different models:

Autoregressive (AR) process

The linear filter $H(z) = 1/A(z)$ is an *all-pole* filter, leading to $b_0 = 1$, $b_i = 0$ for $n > 0$ in the difference equation (6.46):

$$v(n) = w(n) - \sum_{i=1}^{p} a_i \, v(n-i). \tag{6.49}$$

Moving average (MA) process

Here, $H(z) = B(z)$ is an *all-zero* (FIR!) filter, and the difference equation becomes with $a_n = 0$ for $n \geq 0$

$$v(n) = \sum_{i=0}^{q} b_i \, w(n-i). \tag{6.50}$$

Autoregressive, moving average (ARMA) process

In this case the filter $H(z) = B(z)/A(z)$ has both finite poles and zeros in the z-plane and the corresponding difference equation is given by (6.46).

Remarks:

- The AR model is most widely used, since the AR model is suitable of modeling spectra with narrow peaks (resonances) by proper placement of the poles close to the unit circle.
- MA model requires more coefficients for modeling a narrow spectrum, therefore it is rarely used as a model for spectrum estimation.
- By combining poles and zeros the ARMA model has a more efficient spectrum representation as the MA model concerning the number of model parameters.

6.2.1 Relationship between the model parameters and the autocorrelation sequence

In the following it is shown that the model parameters a_i, b_i can be obtained from the autocorrelation sequence of the observed process $v(n)$. These values are then inserted into (6.45) yielding $H(e^{j\omega})$, which is then inserted into (6.48) leading to the power spectrum $\Phi_{vv}(e^{j\omega})$ of our observed process.

In a first step the difference equation (6.46) is multiplied by $v^*(n-\kappa)$ and the expected value is taken on both sides

$$\begin{aligned} E\{v(n)\, v^*(n-\kappa)\} = \sum_{i=0}^{q} b_i \, E\{w(n-i)\, v^*(n-\kappa)\} - \\ - \sum_{i=1}^{p} a_i \, E\{v(n-i)\, v^*(n-\kappa)\}, \end{aligned}$$

which leads to

$$\varphi_{vv}(\kappa) = \sum_{i=0}^{q} b_i \, \varphi_{wv}(\kappa - i) - \sum_{i=1}^{p} a_i \, \varphi_{vv}(\kappa - i). \tag{6.51}$$

The crosscorrelation sequence $\varphi_{wv}(\kappa)$ depends on the filter impulse response:

$$\begin{aligned} \varphi_{wv}(\kappa) &= E\{v^*(n)\, w(n+\kappa)\}, \\ &= E\left\{ \sum_{k=0}^{\infty} h(k)\, w^*(n-k)\, w(n+\kappa) \right\} = \sigma_w^2 \, h(-\kappa). \end{aligned} \tag{6.52}$$

In the last step we have used our prerequisite from above that the process $w(n)$ is assumed to be a zero-mean white random process with $E\{w(n-k)w^*(n+\kappa)\} = \delta(\kappa+k)\sigma_w^2$ and known variance σ_w^2. Thus we have from (6.52)

$$\varphi_{wv}(\kappa) = \begin{cases} 0 & \text{for} \quad \kappa > 0, \\ \sigma_w^2\, h(-\kappa) & \text{for} \quad \kappa \leq 0. \end{cases} \tag{6.53}$$

By combining (6.51) and (6.53) we obtain the desired relationship for the general ARMA case:

$$\varphi_{vv}(\kappa) = \begin{cases} -\sum\limits_{i=1}^{p} a_i\, \varphi_{vv}(\kappa - i) & \text{for } \kappa > q, \\ \sigma_w^2 \sum\limits_{i=0}^{q-\kappa} h(i)\, b_{i+\kappa} - \sum\limits_{i=1}^{p} a_i\, \varphi_{vv}(\kappa - i) & \text{for } 0 \leq \kappa \leq q \\ \varphi_{vv}^*(-\kappa) & \text{for } \kappa < 0. \end{cases} \tag{6.54}$$

$\rightarrow$ nonlinear relationship between the parameters $\varphi_{vv}(\kappa)$ and a_i, b_i

In the following we only consider the AR model case, where (6.54) simplifies to

$$\boxed{\varphi_{vv}(\kappa) = \begin{cases} -\sum\limits_{i=1}^{p} a_i\, \varphi_{vv}(\kappa - i) & \text{for } \kappa > 0, \\ \sigma_w^2 - \sum\limits_{i=1}^{p} a_i\, \varphi_{vv}(\kappa - i) & \text{for } \kappa = 0 \\ \varphi_{vv}^*(-\kappa) & \text{for } \kappa < 0. \end{cases}} \tag{6.55}$$

These equations are also called *Yule-Walker equations* and denote a system of linear equations for the parameters a_i. Equation (6.55)

may also be expressed in matrix notation according to

$$\begin{bmatrix} \varphi_{vv}(0) & \varphi_{vv}^*(1) & \dots & \varphi_{vv}^*(p-1) \\ \varphi_{vv}(1) & \varphi_{vv}(0) & \dots & \varphi_{vv}^*(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_{vv}(p-1) & \varphi_{vv}(p-2) & \dots & \varphi_{vv}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} \varphi_{vv}(1) \\ \varphi_{vv}(2) \\ \vdots \\ \varphi_{vv}(p) \end{bmatrix} \tag{6.56}$$

which is in short-hand notation $\mathbf{R}\,\mathbf{a} = \mathbf{r}$. Once the a_i have been obtained by solving for the coefficient vector $\mathbf{a}$, the variance σ_w^2 can be calculated from

$$\sigma_w^2 = \varphi_{vv}(0) + \sum_{i=1}^{p} a_i \varphi_{vv}(-i). \tag{6.57}$$

Since the matrix $\mathbf{R}$ has a special structure (*Toeplitz* structure) there exist efficient algorithms for solving the system of linear equations in (6.55) and (6.56), respectively (Levison-Durbin algorithm (1947, 1959), Schür recursion (1917)).

6.2.2 Yule-Walker method

In the Yule-Walker method (also called *autocorrelation method*) we simply estimate the autocorrelation sequence from the observed data $v(n)$, where the autocorrelation estimate in (6.2) is used:

$$\hat{\varphi}_{vv}(\kappa) = \frac{1}{N} \sum_{n=0}^{N-1-\kappa} v(n+\kappa)\, v^*(n), \quad \kappa = 0, \dots, p. \tag{6.58}$$

In the matrix version of the Yule-Walker equations (6.56) we replace $\varphi_{vv}(\kappa)$ with $\hat{\varphi}_{vv}(\kappa)$. The resulting linear equation system is solved for the parameter vector $\mathbf{a}$, which now contains the estimated AR parameters $\hat{a}_i$, $i = 1, \dots, p$. Finally, we

obtain σ_w^2 via (6.57) from the $\hat{a}_i$ and the estimated autocorrelation sequence $\hat{\varphi}_{vv}(\kappa)$.

The corresponding power spectrum estimate can now be stated from (6.48) as

$$\hat{\Phi}_{vv}^{(AR)}(e^{j\omega}) = \frac{\sigma_w^2}{\left|1 + \sum_{k=1}^{p} \hat{a}_k \, e^{-jk\omega}\right|^2}. \tag{6.59}$$

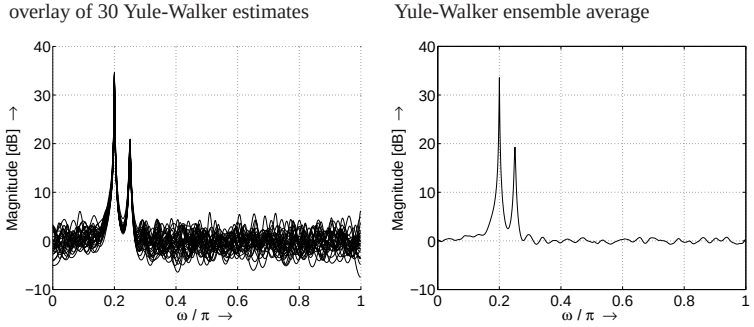
MATLAB-command: `paryule`

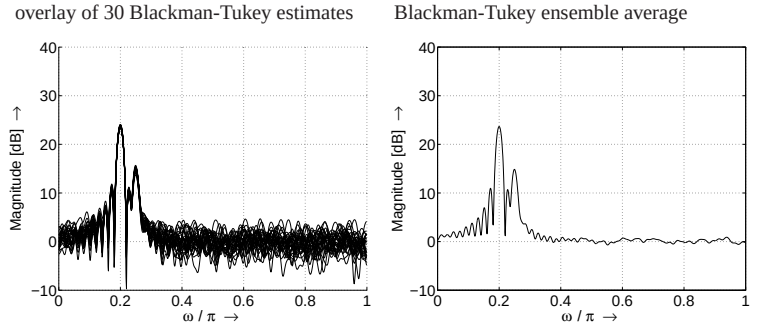
6.2.3 Examples and comparisons to nonparametric methods

In the following we compute power spectrum estimates obtained with the Yule-Walker method and for comparison purposes also with the Blackman-Tukey method (Bartlett window of length L_B).

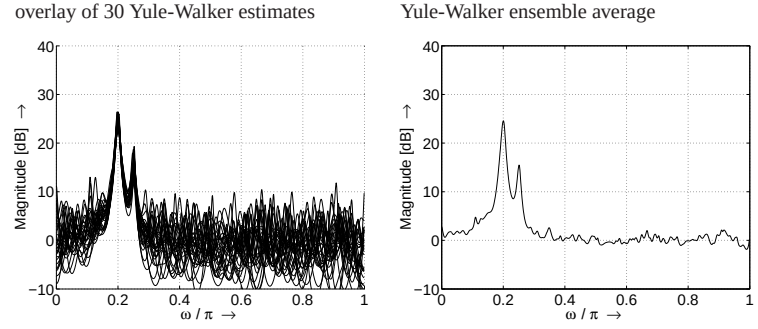
The input process consists of two sinusoids in white Gaussian noise $\eta(n)$ of variance $\sigma_\eta^2 = 1$ according to (6.29) with $\omega_1 = 0.2\pi$, $\omega_2 = 0.25\pi$. The model order of the Yule-Walker method is chosen as $p = 50$.

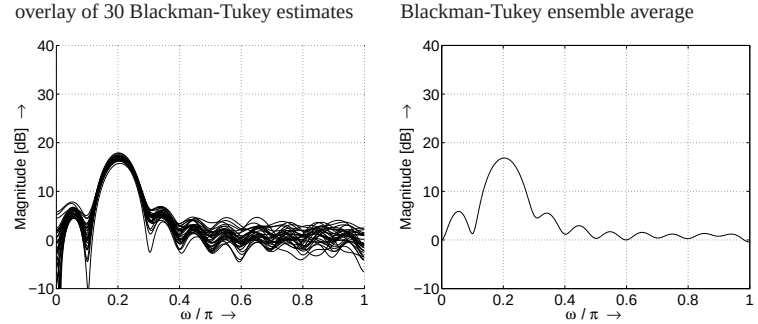
- Length of observed process $N = 512$ (Blackman-Tukey: $L_B = 205$):





- Length of observed process $N = 100$ (Blackman-Tukey: $L_B = 41$):





- We can see that only for the longer data sequence with $N = 512$ the resolution of the estimates are comparable. Clearly, for $N = 100$ the estimate based on an AR-model provides much better frequency resolution for the sinusoidal components than the Blackman-Tukey method.

Remark: Use of a certain model generally requires *a-priori knowledge* about the process. In case of a *model mismatch* (e.g. MA process and AR model) using a nonparametric approach may lead to a more accurate estimate.

Example:

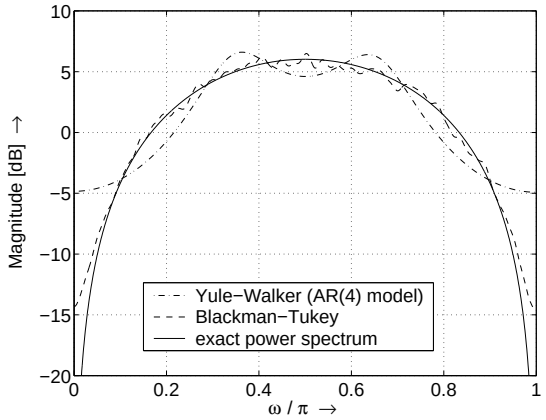
Consider the MA process (length $N = 512$)

$$v(n) = w(n) - w(n-2),$$

where $w(n)$ is again a white-noise zero-mean process with variance $\sigma_w^2 = 1$. The power spectrum of $v(n)$ is

$$\Phi_{vv}(e^{j\omega}) = 2 - 2\,\cos(2\omega).$$

Ensemble average over 30 power spectrum estimates for the Yule-Walker method (AR model of order $p = 4$) and the Blackman-Tukey method (Bartlett window, $L_B = 205$):



→ Blackman-Tukey estimate, which makes no assumption about the process, yields a better estimate of the power spectrum compared to the model-based Yule-Walker approach.