Digital Signal Processing

Jiří Málek

Part I

FIR filter design (continued)

Frequency Sampling Filter Design:

- Method is based on equidistant sampling of the desired ideal frequency response $H_{ID}(e^{j\omega})$ in L points
- The samples

$$H[k] = H_{ID}(e^{j2\pi k/L}), \quad k = 0, \dots, L-1 \tag{1}$$

  ae obtained, which correspond to the L-point DFT of the impulse response $h[n]$ of the FIR filter of order $N = L - 1$.
- The frequency response of the designed filter $H(e^{j\omega})$ corresponds exactly to the response of the ideal filter $H_{ID}(e^{j\omega})$
- However, the method does not provide any control, how the frequency response $H(e^{j\omega})$ behaves *between samples* of $H[k]$
- Filters designed by this method are therefore considered inadequate
- Improved behavior around transition band can be obtained by inclusion of samples, which describe it
- MATLAB: `B=FIR2(N,F,M)`

Filter design based on optimality criteria

- Filter design using windows is simple and leads to sufficiently accurate filters
- However, it is suboptimal from two points of view
- **Tolerances** δ_p, δ_s **must equal** and cannot be changed independently
- In practice, the ripple δ_s is required much lower than δ_p is allowed
- Impossibility of independent design leads to the necessity, to select tolerances in the pass-band too strict, in order to comply with the tolerances in the stop-band (results in higher order of the resulting filter)
- **Distribution of ripples is uneven** for most of the windows and diminishes in the direction away from the transitional band

Advanced design methods allow more freedom in the selection of the magnitude response, allow independently change δ_p, δ_s and/or distribute ripples equally

Least square design of frequency selective FIR filter

- Method allowing independent selection of tolerances δ_p/δ_s
- It is based on minimization of a criterion given by

$$\epsilon^2 = \int_0^{\pi} \left(V(\omega)[A_d(\omega) - A(\omega)]\right)^2 d\omega, \tag{2}$$

  where $V(\omega)$ is the weight (importance) assigned to frequency ω within the design and $(A_d(\omega) - A(\omega))$ is the deviation of the given and desired response
- **Weight** $V(\omega)$ is a non-negative number determining, how much the difference $(A_d(\omega) - A(\omega))$ is undesired in the given band
- Usually, it is selected reciprocal to the tolerances δ, the lesser the tolerance the bigger the weight
- Transition bands have zero weight
- Ususally, the linear phase is desired, the resulting $h[n]$ is thus symmetric
- MATLAB: `B=FIRLS(N,F,A,W)`
- REMARK: This design method should not be mistaken with design of *optimal filter in the least square sense* (LSE). The LSE filter is completely data-based, it is not a frequency selective filter (high pass, low-pass etc.)

Equiripple design of FIR filters

- Designs filters with evenly distributed ripples and independent selection of tolerances δ_p/δ_s
- Complicated computation of coefficients via numerical solution to an optimization problem
- Minimizes criterion given by

$$\epsilon = \max_{\omega \in \mathcal{S}} |E(\omega)| \tag{3}$$

where

$$E(\omega) = \tilde{V}(\omega)[\tilde{A}_d(\omega) - G(\omega)] \tag{4}$$

$$\tilde{V}(\omega) = V(\omega)F(\omega), \tilde{A}_d = \frac{A_d(\omega)}{F(\omega)} \tag{5}$$

- Function $F(\omega)$ depends on a filter type, e.g., FIR type I: $F(\omega) = 1$
- $V(\omega)$ is a weight function, $A(\omega)$ - amplitude of the designed filter
- $A(\omega) = F(\omega)G(\omega)$
- $\mathcal{S}$ - A set of frequencies in the interval $< 0, \pi >$, which correspond to pass-/stop-bands of the designed filter
- $A_d(\omega)$ represents the desired amplitude characteristics

Equiripple design of FIR filters II

- Optimization of criterion (3) stems from the *alternation theorem*
- The numerical solution was originally proposed by Remez (1957) - *Remez exchange algorithm*
- Nowadays, the design is realized through an algorithm proposed by Parks a McCllelan (1982) - *Parks-McClellan algorithm*
- The order of the filter is selected via an ad-hoc empirical formula designed by Kaiser

$$N = \frac{-20\log_{10}\sqrt{\delta_p\delta_s} - 13}{2.32|\omega_p - \omega_s|} \tag{6}$$

- MATLAB: `[N,Fo,Ao,W] = FIRPMORD(F,A,DEV,Fs)` - Determines the order based on equation (6)
  `B=FIRPM(N,Fo,Ao,W)` - Computation of the impulse response
- EXAMPLE: Comparison of the discussed methods for FIR design (`FIRDesignMethodsComparison.m`)

Part II

IIR filter design

Infinite Impulse Response Filters (IIR)

- Impulse response $h[n]$ is infinite right-sided sequence
- Described by recursive difference equation

$$y[n] = \sum_{k=0}^{q} b[k]x[n-k] - \sum_{k=1}^{p} a[k]y[n-k] \tag{7}$$

- Can be unstable, the system function is given by

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b[k]z^{-k}}{1 + \sum_{k=1}^{p} a[k]z^{-k}}, \tag{8}$$

  where q is the number of zeros and p is the number of poles.
- IIR filters cannot be designed to have *(generalized) linear* phase response
- Compared to FIR, filters with the same order have narrower transition band

IIR filter design

- IIR design methods are known for a long time (from analog system theory)
- Design is performed via *discretization of analog filters*
- Advantage: well known algorithms with given properties
- Disadvantage: limited design flexibility (to basic filter types LP, HP, BP, BS)
- Multi-band filter design is difficult
- IIR filters can be designed in the digital domain, but the methods are not very popular (either too difficult or mediocre results)
- Analog and digital filter coincide for the lowest band of frequencies (up to Nyquist frequency)
- IIR design is focused on the magnitude response, the phase response is of a minor importance
- *Phase distortion* can be significant due to non-linear phase

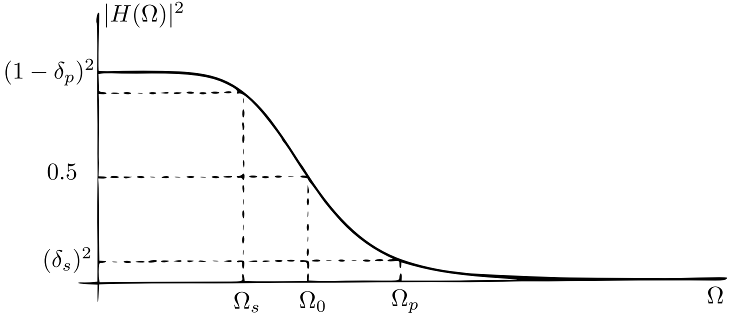
Butterworth filters I

- Butterworth filters are defined using squared magnited as

$$|H(\Omega)|^2 = \frac{1}{1 + (\frac{\Omega}{\Omega_0})^{2N}} \tag{9}$$

- **Butterworth filter properties:**
  - Magnitude response is monotone decreasing function of Ω
  - Maximum is located in $|H(\Omega)|_{\Omega=0} = 1$
  - Attenuation of square magnitude by 3dB (to 0.5) is located in Ω_0
  - Magnitude response is practically constant for low frequencies
- MATLAB:
  `[N, Wn] = BUTTORD(Wp, Ws, Rp, Rs)` - Filter order for given specifications
  `[B,A] = BUTTER(N,Wn)` - Filter parameters

Butterworth filters II



Squared magnitude of analog Butterworth filter (N=3)

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

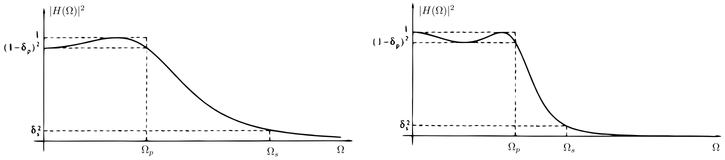
Chebyshev filters I

- Two filter types design based on Chebyshev polynomial
- Chebyshev filters exhibit monotone magnitude response either in the passband (type I) or in the stopband (type II)
- Ripples allow narrower transition band compared to Butterworth filter
- **Chebyshev filters type I** are defined using squared magnitude response as

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 T_N^2(\frac{\Omega}{\Omega_0})}, \tag{10}$$

  Ω_0, ϵ - free parameters, $T_N(\cdot)$ - Chebyshev polynomial of order N
- **Properties of Chebyshev I filters:**
  - Magnitude response exhibit ripples in the passband
  - Squared magnitude equals 1 (odd N) or $1/(1+\epsilon^2)$ (even N) for $\Omega = 0$
  - Maximum value 1 is achieved several times in the pass band (based on filter order N)
  - For $\Omega > \Omega_0$ is the magnitude response monotone decreasing

- MATLAB:
  `[N, Wp] = CHEB1ORD(Wp, Ws, Rp, Rs)` - Filter order satisfying the specifications
  `[B,A] = CHEBY1(N,R,Wp)` - Filter parameter computation



Squared magnitude response of analog Chebyshev type I filter (N=2/N=3)

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

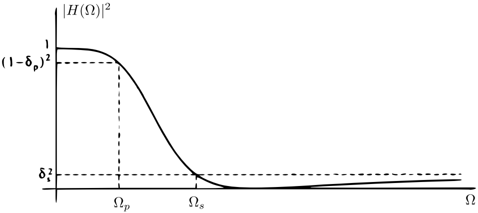
- **Chebyshev type II filters** are defined using the squared magnitude response as

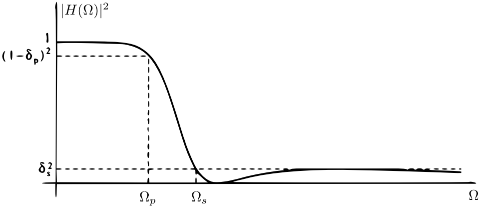
$$|H(\Omega)|^2 = 1 - \frac{1}{1 + \epsilon^2 T_N^2(\frac{\Omega}{\Omega_0})}, \quad (11)$$

Ω_0, ϵ - free parameters, $T_N(\cdot)$ - Chebyshev polynomial of order N

- **Properties of Chebyshev II filters:**
  - For $\Omega > \Omega_0$ exhibits the magnitude response ripples
  - Squared magnitude response for $\Omega = 0$ equals 1 for all N
  - For values $\Omega < \Omega_0$ is magnitude response monotone decreasing
- MATLAB:
  `[N, Ws] = CHEB2ORD(Wp, Ws, Rp, Rs)` - Filter order satisfying the specifications
  `[B,A] = CHEBY2(N,R,Wst)` - Filter parameter computation

Chebyshev filters IV





Squared magnitude response of analog Chebyshev type II filter (N=2/N=3)

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Elliptic filters

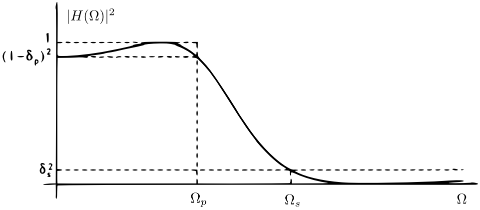
- **Elliptic filters** are defined using the squared magnitude response as

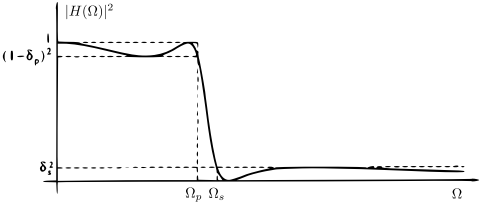
$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 R_N^2(\frac{\Omega}{\Omega_0})}, \tag{12}$$

Ω_0, ϵ - free parameters, $R_N(\cdot)$ - Chebyshev rational function of order N

- **Properties of Elliptic filters:**
  - Magnitude response is rippled in the passband and the in stopband
  - Ripples allow narrower transition band compared to other mentioned IIR filter types
- MATLAB:
  `[N, Wp] = ELLIPORD(Wp, Ws, Rp, Rs)` - Filter order satisfying the specifications
  `[B,A] = ELLIP(N,Rp,Rs,Wp)` - Filter parameter computation

Elliptic filters II





Squared magnitude response of analog elliptic filter (N=2/N=3)

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

Comparison of FIR and IIR

FIR:

- Usually designed with linear phase - minimum phase distortion
- More intuitive filter design
- Always stable

IIR:

- Lower latency + narrower transition band compared to similar order FIR filter
- Lesser computational burden

Phase distortion IIR/FIR:

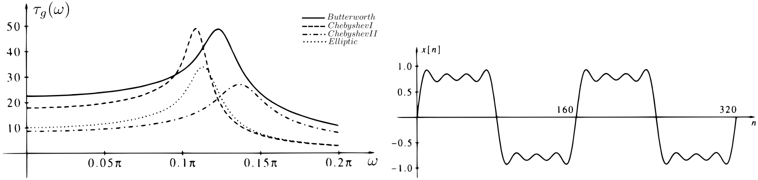
- IIR cannot be designed with linear phase
- Signasl passing through IIR are distorted (each frequency component is delayed differently) even if its spectrum is located in the filter passband
- EXAMPLE: Comparison - phase distortion FIR/IIR

Phase distortion for signals within the passband of FIR/IIR

- Let us have a filter specification: $\omega_p = 0.1\pi, \omega_s = 0.2\pi, \delta_p = \delta_s = 0.001$
- Let us design IIR of all basic types (with minimum order) satisfying the specifications
- Let us generate a test signal (first four components of a square wave)

$$x[n] = \sum_{m=1}^{4} \frac{1}{2m-1} \sin(0.0125\pi(2m-1)n) \qquad (13)$$

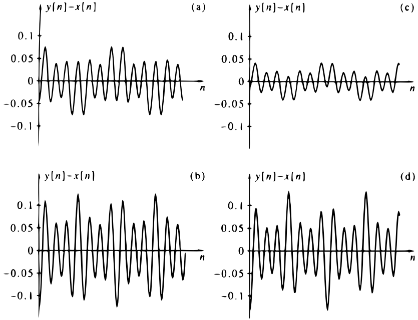
- Signal is band limited by the highest frequency 0.0875π, i.e., it is within the filter passband



(a) Designed filters: group delay, (b) Test signal $x[n]$

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

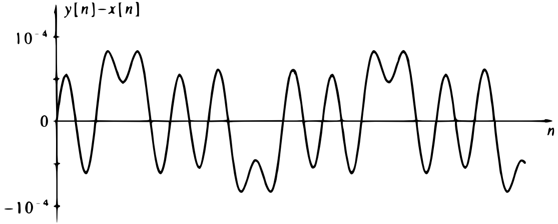
- If the signal is located within the passband of the filter, we aim for the lowest possible distortion
- Let us compare the difference between the output signal $y[n]$ and the input signal $x[n]$
- (Signal are shifted to compensate the varying group delay of the filters for the first harmonic $w = 0.0125\pi$)



$y[n] - x[n]$, (a) Butterworth, (b) Chebyshev I, (c) Chebyshev II, (d) Elliptic

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

- Allowed tolerance $\delta_p = 0.001$, phase distortion causes difference almost 100 times higher (in the time-domain)
- Let us repeat the experiment with FIR filter designed by windowing method and Kaiser window



$y[n] - x[n]$ for FIR designed by windowing method and Kaiser window

SOURCE: BOAZ PORAT, A Course in Digital Signal Processing

- Error of FIR is smaller than the desired tolerance
- FIR exhibits advantage against IIR: it eliminates the phase distortion
- FIR latency is higher compared to IIR ($\tau_{g,FIR} = 38$; $9 < \tau_{g,IIR} < 23$)

Part III

Matlab tools for the digital filter design

FIR filters:

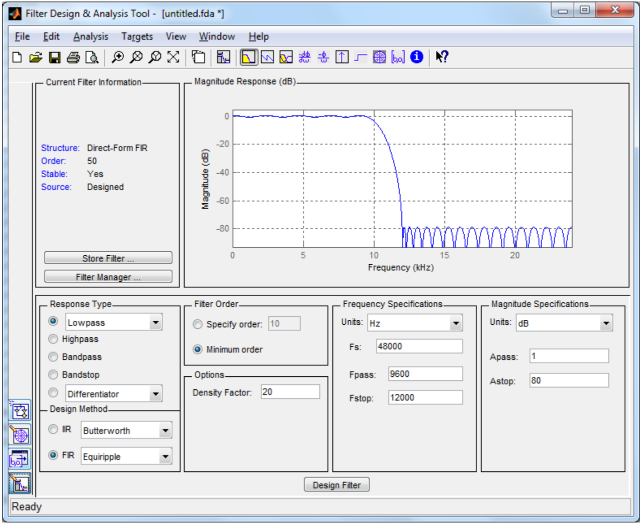
- `B=FIR1(n,Wn)` - Windowing method
- `B=FIR2(N,F,M)` - Frequency response sampling
- `B=FIRLS(N,F,A,W)` - Least squares filter design
- `[N,Fo,Ao,W] = FIRPMORD(F,A,DEV,Fs)` - Equiripple filter design - order
  `B=FIRPM(N,Fo,Ao,W)` - Equiripple filter design - coefficients

IIR filters:

- **Butterworth IIR filter - order**
  ```
  [N, Wn] = BUTTORD(Wp, Ws, Rp, Rs)
  ```
  Butterworth IIR filter - coefficients
  ```
  [B,A] = BUTTER(N,Wn)
  ```
- **Chebyshev type 1 IIR filter - order**
  ```
  [N, Wp] = CHEB1ORD(Wp, Ws, Rp, Rs)
  ```
  Chebyshev type 1 IIR filter - coefficients
  ```
  [B,A] = CHEBY1(N,R,Wp)
  ```
- **Chebyshev type 2 IIR filter - order**
  ```
  [N, Ws] = CHEB2ORD(Wp, Ws, Rp, Rs)
  ```
  Chebyshev type 2 IIR filter - coefficients
  ```
  [B,A] = CHEBY2(N,R,Wst)
  ```
- **Elliptic IIR filter - order**
  ```
  [N, Wp] = ELLIPORD(Wp, Ws, Rp, Rs)
  ```
  Elliptic IIR filter - coefficients
  ```
  [B,A] = ELLIP(N,Rp,Rs,Wp)
  ```

Matlab tools for the digital filter design III

`filterDesigner` (`fdatool` until Matlab 2016b)
- GUI for digital filter design (Signal Processing Toolbox)



- EXAMPLE: Filter design using `filterDesigner`
- Let us design Butterworth filter from previous example:
- $\omega_p = 0.1\pi, \omega_s = 0.2\pi, \delta_p = \delta_s = 0.001$

1. Select "**Lowpass**" in menu "**Response Type**"
2. Select "**IIR: Butterworth**" in menu "**Design method**"
3. Select "**Minimum order**" in menu "**Filter Order**"
4. Specify cut-off frequencies in menu "**Frequency specifications**"
   Select as units the normalized digital frequency and enter values *wpass* = 0.1 and *wstop* = 0.2, where 1 corresponds to the Nyquist frequency, i.e., $\omega = \pi$
5. Select tolerance in "**Magnitude specifications**"
   Select units dB and tolerance Apass = 0.009dB, Astop = 60dB

Matlab tools for the digital filter design V

- `filterDesigner` allows to **display various characteristics** of the designed filter, such as:
- Magnitude/phase response, group/phase delay, impulse response, zero-pole plot...
- `filterDesigner` allows **factorization of high order IIR** into a cascade of second-order filters (biquadratic filter)
- This is done due to numerical instability of high-order IIRs, if coefficient quantization is required
- To filter using the second-order cascade Matlab uses command `sosfilt`

Thank you for attention!