UNIVERSITY OF CAMBRIDGE

3F1, Signals and Systems

PART II.1: Design of filters

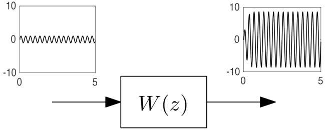
Fulvio Forni (f.forni@eng.cam.ac.uk)

October 24, 2018

Intro

Filter design for signal processing

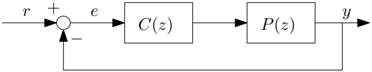
Design the filter $W(s)$ so that the filtered signal is...



Example: enhance/reduce the low frequencies of your music, band-pass filter to select a specific radio station

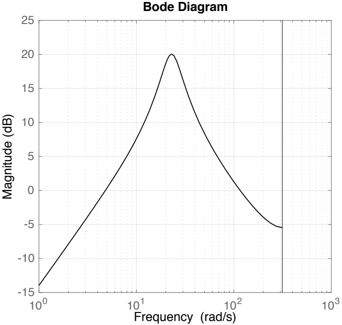
Filter design for control

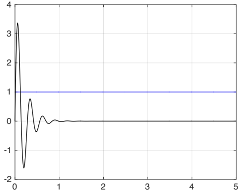
Design the controller $C(s)$ so that the closed loop $W_{r,e} = \frac{1}{1+PC} \ldots$

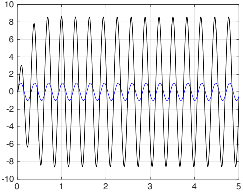


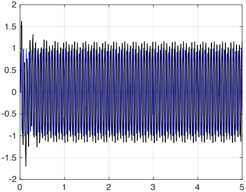
Example: shape the frequency response of the open loop PC to achieve performance/robustness of closed-loop tracking error $W_{r,e}$

$$G(z) = \frac{z-1}{z^2 - 1.85z + 0.9}$$



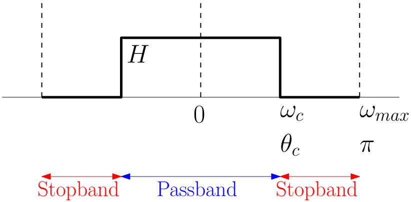






Module A

Ideal filters

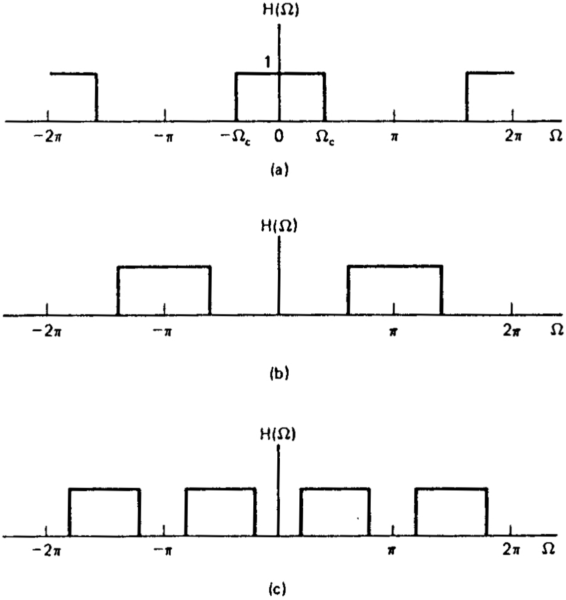


$$H_a(j\omega) = \begin{cases} 1 & |\omega| \leq \omega_c \\ 0 & |\omega| > \omega_c \end{cases} \qquad H_d(e^{j\theta}) = \begin{cases} 1 & |\theta| \leq \theta_c \\ 0 & |\theta| > \theta_c \end{cases}$$

Analog $\cos(\omega t)$ Digital $\cos(\theta k)$ $\theta = \omega T$

With sampling time T the max frequency is $\omega_{max} = \frac{\pi}{T}$.
Necessarily $\omega_c < \omega_{max}$. Normalized cutoff frequency $\theta_c = \omega_c T$.

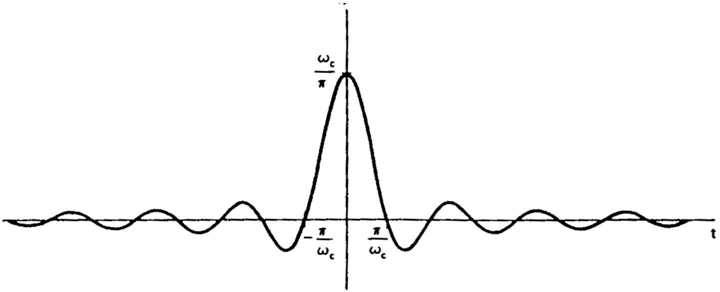
(a) Low pass, (b) high pass, (c) band pass



Ideal filter can't be implemented: non-causal impulse response $h_a(t) \neq 0$ and $h_d(k) \neq 0$ for $t, k < 0$.

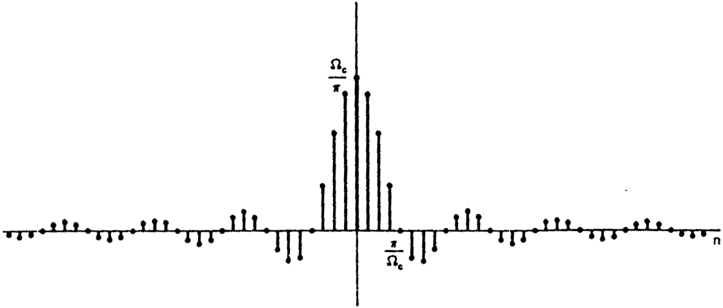
By inverse Fourier transform $\mathcal{F}^{-1}$

$$\begin{aligned} h_a(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} H_a(j\omega) e^{j\omega t} d\omega = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega t} d\omega \\ &= \frac{1}{2\pi} \frac{e^{j\omega_c t} - e^{-j\omega_c t}}{jt} = \frac{\sin(\omega_c t)}{\pi t} \end{aligned}$$



By inverse Fourier transform $\mathcal{F}^{-1}$

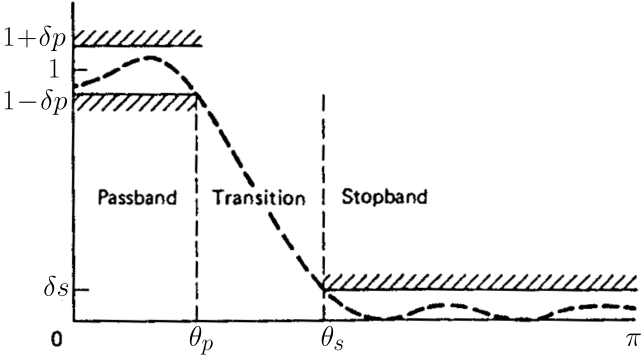
$$h_d(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\theta})e^{j\theta k}d\theta = \frac{1}{2\pi}\int_{-\theta_c}^{\theta_c} e^{j\theta k}d\theta = \frac{\sin(k\theta_c)}{\pi k}$$



Any realisable filter can only approximate it.

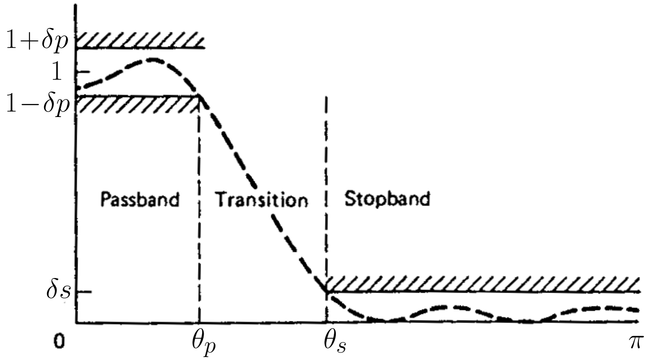
Module B

Realizable filters basic specs



A typical filter specification must specify maximum permissible deviations from the ideal:

- band edge frequencies or corner frequencies
- a maximum passband ripple
- a minimum stopband attenuation

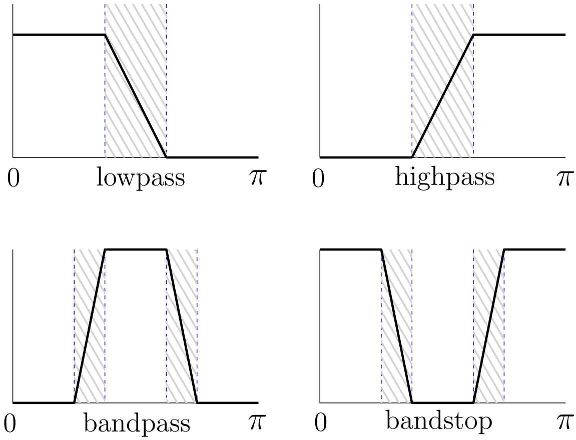


specs in dB:

- passband ripple $20\log_{10}(1+\delta p)$ dB
- peak-to-peak passband ripple $20\log_{10}(1+2\delta p)$ dB
- minimum stopband attenuation $= -20\log_{10}(\delta s)$ dB.

Example: $\delta p = 0.06$ gives peak-to-peak passband ripple $\simeq 1dB$; $\delta s = 0.01$ gives a minimum stopband attenuation $= 40dB$.

Nonideal filter classes



Module C

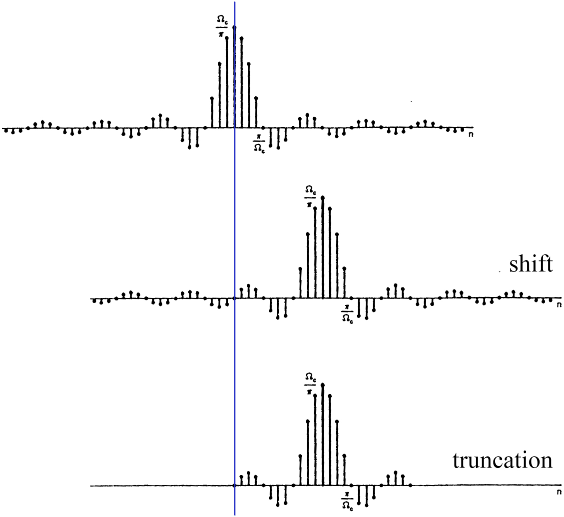
Finite impulse response (FIR) and Infinite impulse response (IIR) filters

FIR filters

h_k is the non-causal impulse response of the ideal filter

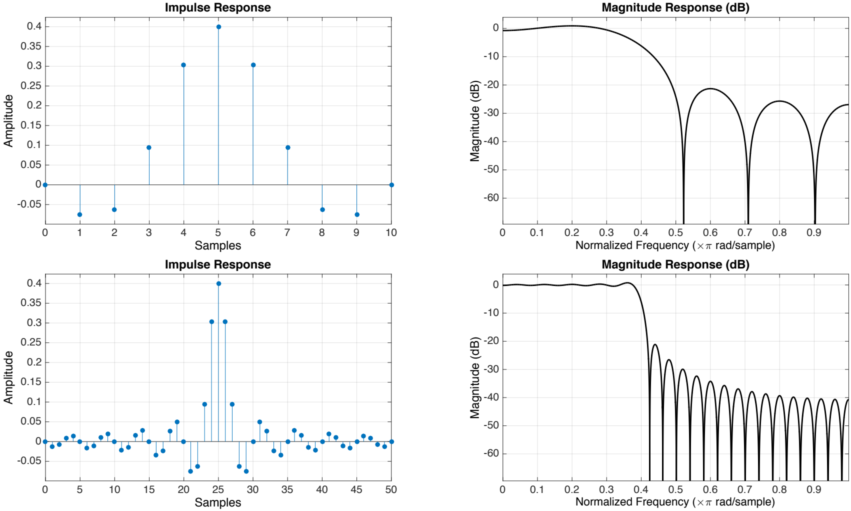
New (finite impulse response) filter G derived by shift and truncation at $N+1$ samples of the ideal response

Causality is recovered. Intuition: truncation of "small" samples has modest impact...



$$G(z) = \sum_{k=0}^{N} g_k z^{-k} \quad g_k = h_{k-N/2}$$

Truncation at N=10 and N=50 (11 and 51 samples)

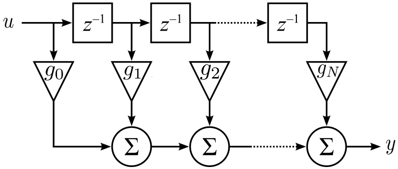


Longer horizon $N \rightarrow$ smaller samples are discarded $\rightarrow$ reduced distortion.

- FIR filters are simple

$$G(z) = \sum_{k=0}^{N} g_k z^{-k} \quad \rightarrow \quad y_n = \sum_{k=0}^{N} g_k u_{n-k}$$

- Feedforward or non-recursive



- Realized efficiently in hardware (FFT)
- Inherently stable: $G(z) = \frac{\sum_{k=0}^{N} g_k z^{N-k}}{z^N}$, all poles in 0.
- Truncation introduces distortion (can we do better?)

IIR filters

- Feedback: the output "goes back" to the filter:

$$y_k = -a_{n-1}y_{k-1} + \cdots - a_0 y_{k-n} + b_m u_{k-n+m-1} + \cdots + b_0 u_{k-n}$$

- They have finite polynomial representation:

$$A(z)Y(z) = B(z)U(z)$$

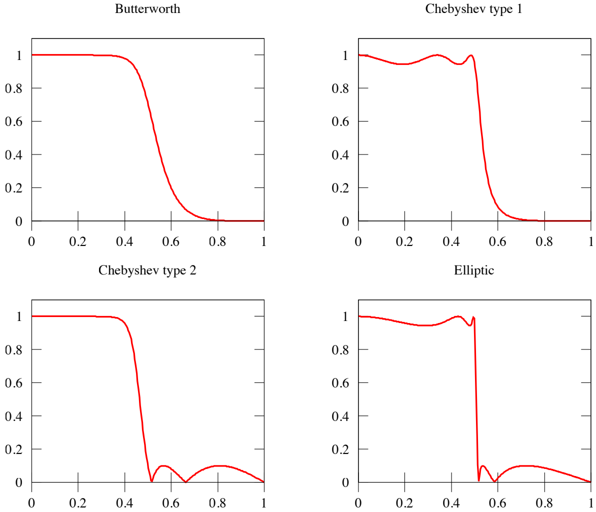
$$A(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_0$$
$$B(z) = b_m z^m + b_{m-1}z^{m-1} + \cdots + b_0$$

  but infinite impulse response:

$$G(z) = \frac{B(z)}{A(z)} = \sum_{k=0}^{\infty} g_k z^{-k}$$

- They typically meet a given set of specifications with a much lower filter order than FIR filters.

- Used for digital implementation of analog filters (discretization)



(do they preserve analog filter features? Stability?)

FIR filters

- Simple, stable, feedforward, fast implementation (FFT)
- Can have exactly linear phase $G(e^{j\theta}) = |G(e^{j\theta})|e^{-j\theta}$
- Design methods are generally linear (easy, efficient)
- Higher filter order than IIR filters to achieve a given level of performance.
- Often greater delay than for an equal performance IIR filter.

IIR filters

- Can be unstable, use feedback
- Design methods: (i) direct optimization of the transfer function, (ii) generation of a digital filter from an analogue prototype, k=0. Digital implementation of analog filters.
- Lower filter order than FIR filters to achieve a given level of performance.
- Shorter delay than for an equal performance FIR filter.

UNIVERSITY OF CAMBRIDGE

3F1, Signals and Systems

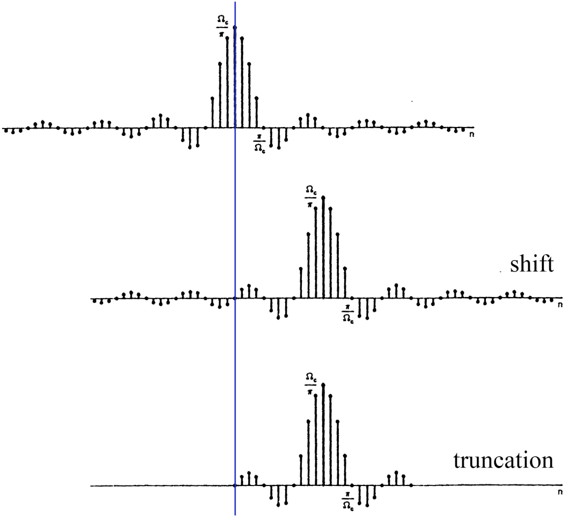
PART II.2: Design of FIR filters

Fulvio Forni (f.forni@eng.cam.ac.uk)

October 24, 2018

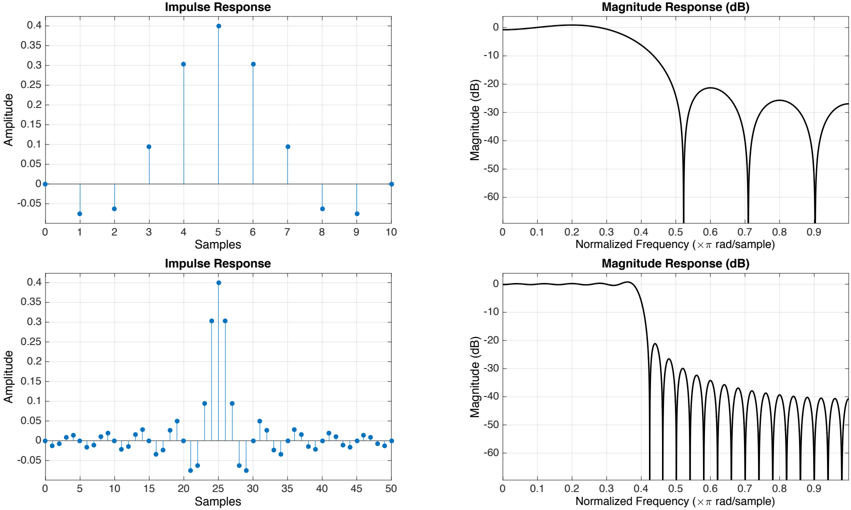
FIR filter derived by shift and truncation of the ideal response

Causality is recovered. Truncation of "small" samples has modest impact...



$$G(z) = \sum_{k=0}^{N} g_k z^{-k}$$

Truncation at N=10 and N=50 (11 and 51 samples)



Why frequency distortion?

Module A

Frequency distortion of truncation/windowing

Shifted desired impulse response

$$h_k$$

Truncated unit sample response

$$g_k = h_k \qquad 0 \leq k \leq N$$

Truncation = multiplication by a window

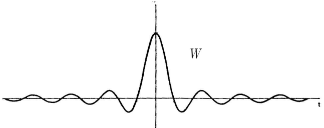
$$g_k = h_k w_k \qquad\qquad w_k = \begin{cases} 1 & 0 \leq k \leq N \\ 0 & \text{otherwise} \end{cases}$$

Using duality of multiplication/convolution

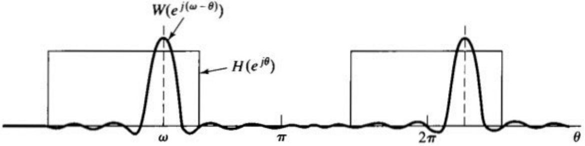
$$G(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta$$

$$W(e^{j\theta}) = \sum_{k=-\infty}^{\infty} w_k e^{-j\theta k} = \sum_{k=0}^{N} e^{-j\theta k} = \frac{1 - e^{-j\theta(N+1)}}{1 - e^{-j\theta}} = e^{\frac{-j\theta N}{2}} \frac{\sin\left(\frac{\theta(N+1)}{2}\right)}{\sin\left(\frac{\theta}{2}\right)}$$

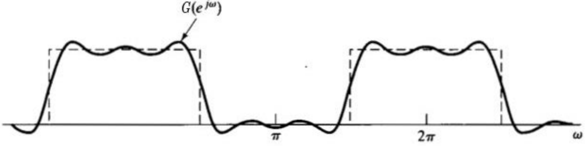
$H(e^{j\theta}) =$ 



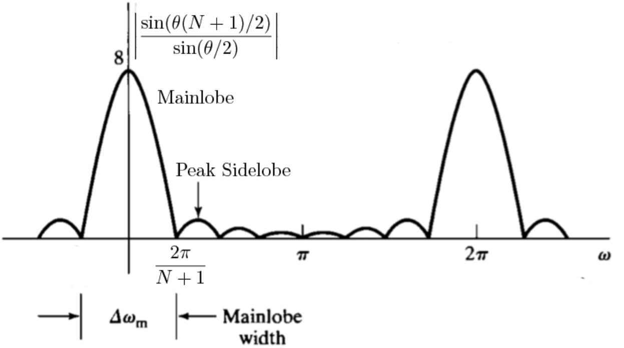
towards impulse as $N \to \infty$



$G = H * W$



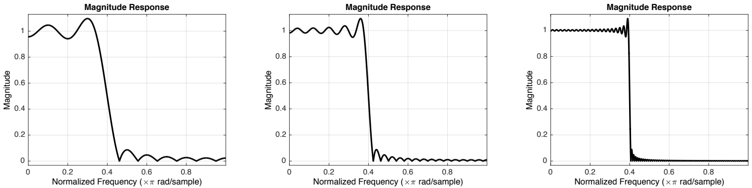
$|W(e^{j\theta})|$



- Transition band is related to mainlobe. It reduces as $N \to \infty$.
- Ripples of G are related to the area under sidelobes, which remains constant as N increases.

How to improve?

$|G(e^{j\theta})|$

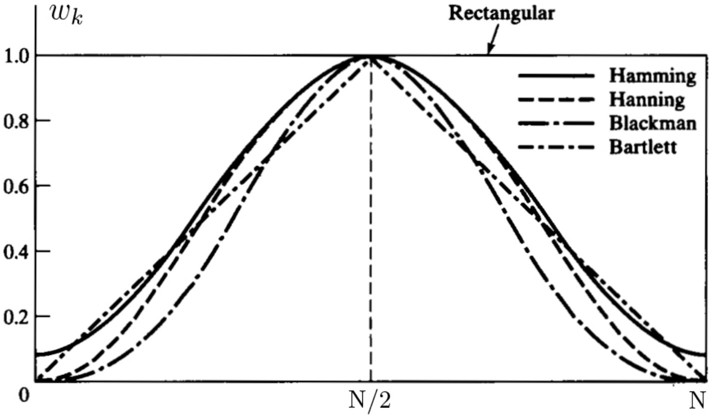


- Transition band is related to mainlobe. It reduces as $N \to \infty$.
- Ripples of G are related to the area under sidelobes, which remains constant as N increases.

How to improve?

Module B

Design by window method



$$g_k = h_k w_k$$

Reduce frequency distortion by adopting different windows

Rectangular

$$w_k = \begin{cases} 1 & 0 \leq k \leq N \\ 0 & \text{otherwise} \end{cases}$$

Bartlett (triangular)

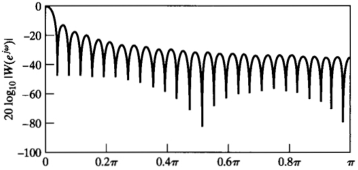
$$w_k = \begin{cases} 2k/N & 0 \leq k \leq N/2 \\ 2 - 2k/N & M/2 < k \leq N \\ 0 & \text{otherwise} \end{cases}$$

Hanning

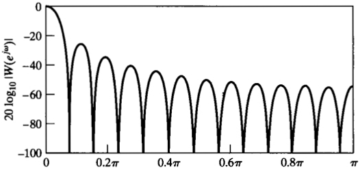
$$w_k = \begin{cases} 0.5 - 0.5\cos(2\pi k/N) & 0 \leq k \leq N \\ 0 & \text{otherwise} \end{cases}$$

Hamming

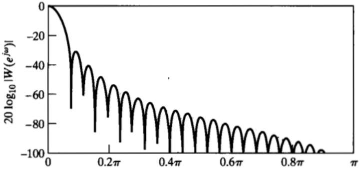
$$w_k = \begin{cases} 0.54 - 0.46\cos(2\pi k/N) & 0 \leq k \leq N \\ 0 & \text{otherwise} \end{cases}$$



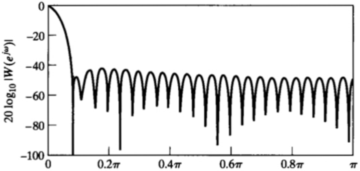
Rectangular



Triangular



Hanning

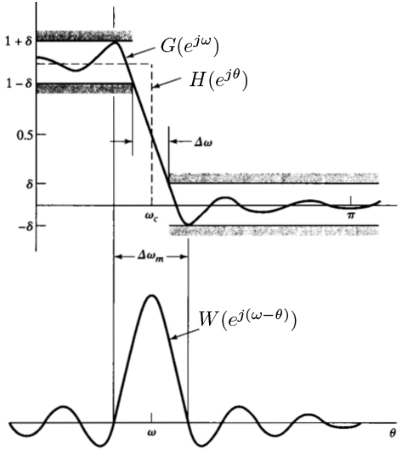


Hamming

$$G(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\theta})W(e^{j(\omega-\theta)})d\theta$$

COMPARISON OF COMMONLY USED WINDOWS

Type of Window	Peak Side-Lobe Amplitude (Relative)	Approximate Width of Main Lobe	Peak Approximation Error, $20\log_{10}\delta$ (dB)
Rectangular	−13	$4\pi/(N+1)$	−21
Bartlett	−25	$8\pi/N$	−25
Hanning	−31	$8\pi/N$	−44
Hamming	−41	$8\pi/N$	−53
Blackman	−57	$12\pi/N$	−74



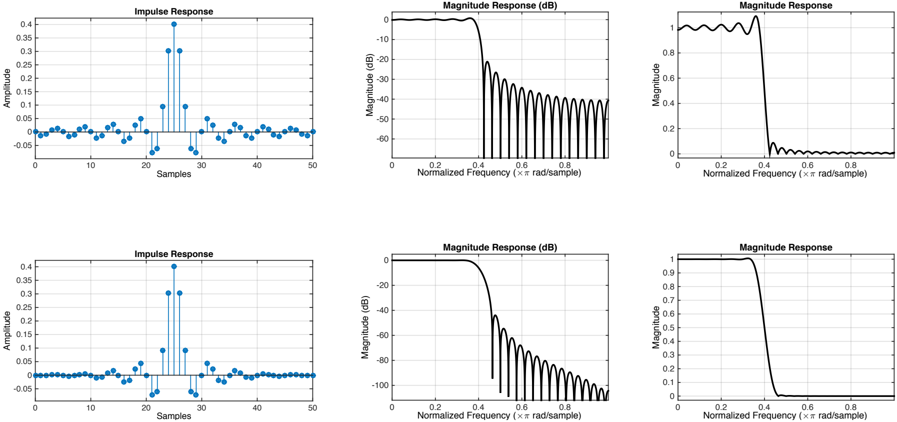
Shape and duration of the window to control the resulting filter properties.

- Smaller side lobes yield better approximations of the ideal response.
- Narrower transition bandwidth for increasing N. Symmetric at cutoff ω_c.
- Same δ for passband error and stopband approximation

The window method is conceptually simple and can quickly design filters to approximate a given target response. It does not explicitly impose amplitude response constraints, such as passband ripple or stopband attenuation, so it has to be used iteratively to produce designs which meet such specifications. Steps:

1. Select a suitable window function w_k.
2. Specify an ideal frequency response H.
3. Compute the coefficients of the ideal filter h_k.
4. Multiply the ideal coefficients by the window function to give the filter coefficients and delay to make causal.
5. Evaluate the frequency response of the resulting filter and iterate 1-5 if necessary.

Example: Rectangular vs Hanning, N=50

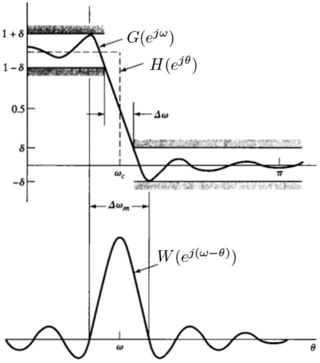


Example: Design of a low pass filter (Steps 1 and 2)

Passband $\omega_p = 0.2\pi$
Stopband $\omega_s = 0.3\pi$
Approximation error $\delta = 0.01$

COMPARISON OF COMMONLY USED WINDOWS

Type of Window	Peak Side-Lobe Amplitude (Relative)	Approximate Width of Main Lobe	Peak Approximation Error, $20\log_{10}\delta$ (dB)
Rectangular	−13	$4\pi/(N+1)$	−21
Bartlett	−25	$8\pi/N$	−25
Hanning	−31	$8\pi/N$	−44
Hamming	−41	$8\pi/N$	−53
Blackman	−57	$12\pi/N$	−74



Approximation error: -40 dB, Hanning window.
Cutoff frequency: $\omega_c = \frac{\omega_s+\omega_p}{2}\pi = 0.25\pi$.
Mainlobe width: $\omega_s - \omega_p = 0.1\pi \rightarrow N = 80$.

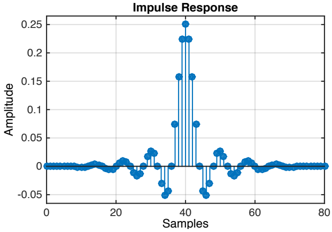
Example: Design of a low pass filter (Steps 3 and 4)

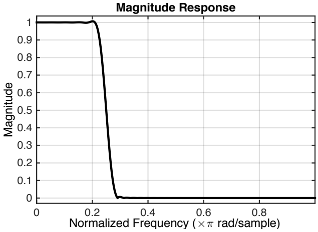
Ideal filter by inverse Fourier transform $\mathcal{F}^{-1}$

$$h_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\theta})e^{j\theta k}d\theta = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\theta k}d\theta = \frac{\sin(k\omega_c)}{\pi k} = \frac{\omega_c}{\pi}\mathrm{sinc}(k\omega_c)$$

Delay and product by window

$$g_k = h_{k-N/2}w_k$$





Note: h_k are computed by inverse Fourier transform of H. This is hard in general. Use of numerical algorithms (*lectures on Discrete Fourier Transform, Fast Fourier Transform...*).

Module C

Multi-band design

Design highpass, bandpass,...? **Composition of lowpass filters**

The ideal bandpass filter

$$
\begin{aligned}
H(e^{j\theta}) &= \begin{cases} 1 & \omega_1 \le \theta \le \omega_2 \\ 0 & \text{otherwise} \end{cases} \\
&= \mathrm{Lowpass}_{[0,\omega_2]}(e^{j\theta}) - \mathrm{Lowpass}_{[0,\omega_1]}(e^{j\theta})
\end{aligned}
$$

where

$$
\mathrm{Lowpass}_{[0,\omega_c]}(e^{j\theta}) = \begin{cases} 1 & 0 \le \theta \le \omega_c \\ 0 & \text{otherwise} \end{cases}
$$

Ideal impulse response by linearity and antitrasform:

$$
\begin{aligned}
h_k &= \mathcal{F}^{-1}(H) \\
&= \mathcal{F}^{-1}(\mathrm{Lowpass}_{[0,\omega_2]}) - \mathcal{F}^{-1}(\mathrm{Lowpass}_{[0,\omega_1]}) \\
&= \frac{\sin(k\omega_2)}{\pi k} - \frac{\sin(k\omega_1)}{\pi k}
\end{aligned}
$$

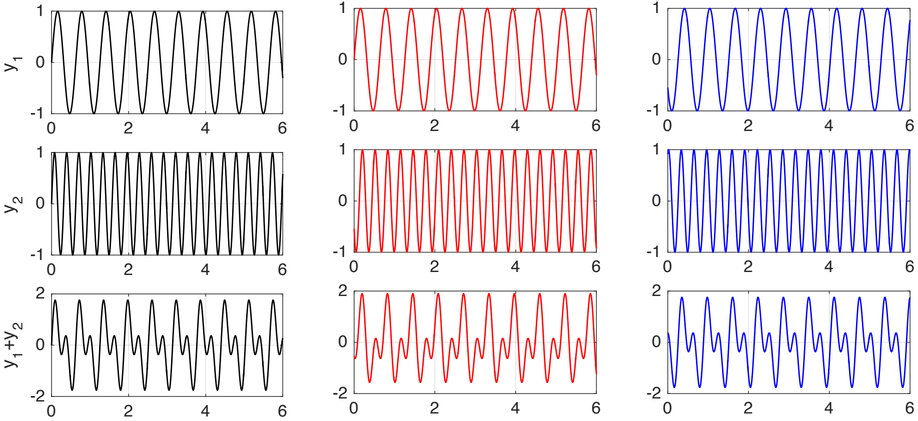
Final filter by delay and multiplication with desired window w_k:

$$
g_k = h_{k-N/2} w_k
$$

Module D

Linear Phase

Linear phase $G(e^{j\theta}) = |G(e^{j\theta})|e^{-j\theta\frac{N}{2}}$



No filter

Nonlinear phase filter

Linear phase filter

Linear phase $G(e^{j\theta}) = |G(e^{j\theta})|e^{-j\theta\frac{N}{2}}$ is achieved if $\mathbf{g_k} = \mathbf{g_{N-k}}$.

$$
\begin{aligned}
G(e^{j\theta}) &= \sum_{k=0}^{N} g_k e^{-j\theta k} \\
&= g_0 e^{-j\theta 0} + g_N e^{-j\theta N} + g_1 e^{-j\theta 1} + g_{N-1} e^{-j\theta(N-1)} + \dots \\
&= e^{-j\theta\frac{N}{2}} \Big(\underbrace{g_0 e^{j\theta\frac{N}{2}} + g_N e^{-j\theta\frac{N}{2}}}_{2g_0 \cos(\theta\frac{N}{2}) \text{ is real}} + \underbrace{g_1 e^{j\theta(\frac{N}{2}-1)} + g_{N-1} e^{-j\theta(\frac{N}{2}-1)}}_{2g_1 \cos(\theta(\frac{N}{2}-1)) \text{ is real}} + \dots \Big)
\end{aligned}
$$

Note: *window method gives linear phase filters*. The coefficients w_k of all window functions satisfies $w_k = w_{N-k}$. If the desired impulse response h_k is also symmetric $h_k = h_{N-k}$ then $g_k = h_k w_k$ is symmetric, that is, the resulting filter has linear phase.

Module E

Design by optimization

Given the ideal filter H and the weighting function W, find the optimal filter G of length N such that, given

$$E(\theta) = W(\theta)[H(\theta) - G(\theta)] \ ,$$

- G minimizes the least-squares error

$$\int_{-\pi}^{\pi} E^2(\theta) d\theta$$

- G minimizes the max error

$$\sup_{-\pi \leq \theta \leq \pi} |E(\theta)|$$

$\Rightarrow$ "Equiripple filters".

Module F

Matlab code

b = fir1(n,Wn)

b contains the coefficients of the order n Hamming-windowed filter. This is a lowpass, linear phase FIR filter with cutoff frequency Wn.

b = fir1(n,Wn,window)

Uses the window specified in column vector window for the design.

b = fir2(n,f,m)

b contains the coefficients of the order n FIR filter whose frequency-magnitude characteristics match f and m.
f is a vector of frequency points ranging from 0 to 1.
m is a vector containing the desired magnitude response.

firls

The firls function minimizes the integral of the square of the error between the desired frequency response and the actual frequency response.

firpm

The firpm function returns filters that are optimal in the sense that the maximum error between the desired frequency response and the actual frequency response is minimized.

filterbuilder

GUI-based filter design

fvtool

Open Filter Visualization Tool

More info: ▸ Matlab: filter design ▸ Matlab: filter analysis

3F1, Signals and Systems

PART II.3: Design of IIR filters

Fulvio Forni (`f.forni@eng.cam.ac.uk`)

October 24, 2018

Module A

Discretization by response matching

Impulse invariance

$G_c(s)$ Laplace transform continuous-time filter. The impulse response of the corresponding *impulse invariance* digital filter G(z) (with sampling T) is equal to the impulse response of the $G(s)$ sampled at $t = kT$.

$$G(z) = \mathcal{Z}\left(\mathcal{L}^{-1}\left(G_c(s)\right)_{t=kT}\right)$$

- For <u>bandlimited</u> filters the digital filter frequency response will closely approximate the continuous-time frequency response.
- Preserves stability ($\Re(\beta_q) < 0 \to |e^{\beta_q T}| < 1$).

Example:

$$G_c(s) = \frac{\alpha}{s-\beta} \overset{\mathcal{L}^{-1}}{\rightarrow} \alpha e^{\beta t} \xrightarrow{\text{sample}} \alpha e^{\beta T k} \overset{\mathcal{Z}}{\rightarrow} \frac{\alpha}{1 - e^{\beta T} z^{-1}} = G(z)$$

Step response invariance

Take a continuous time filter/system with Laplace transfer function $G_c(s)$. The corresponding *step response invariance* digital filter (with sampling period T) is a digital filter whose step response is equal to the step response of the continuous time filter sampled at $t = kT$.

$$G(z) = \frac{z-1}{z}\mathcal{Z}\left(\mathcal{L}^{-1}\left(\frac{G_c(s)}{s}\right)_{t=kT}\right)$$

$$G_c(s) \overset{\text{step}}{\to} \frac{G_c(s)}{s} \overset{\mathcal{L}^{-1}}{\to} y(t) \overset{\text{sample}}{\to} y(kT) \overset{\mathcal{Z}}{\to} Y(z) \overset{\frac{1}{\text{step}}}{\to} \frac{z-1}{z}Y(z) = G(z)$$

1. Find the step response of the continuous system
2. Sample at time $t = kT$ and take the z-transform
3. Multiply by $(z-1)/z$.

Ramp invariance

Take a continuous time filter/system with Laplace transfer function $G_c(s)$. The corresponding *ramp response invariance* digital filter (with sampling period T) is a digital filter whose ramp response is equal to the ramp response of the continuous time filter sampled at $t = kT$.

$$G(z) = \frac{(z-1)^2}{Tz}\mathcal{Z}\left(\mathcal{L}^{-1}\left(\frac{G_c(s)}{s^2}\right)_{t=kT}\right)$$

<u>Note</u>: any waveform invariance can be considered. The digital filter will preserve the properties of the continuous filter response to that particular waveform.

Module B

Discretization by algebraic transformations

Algebraic transformations are used to map filters that are not bandlimited (avoiding aliasing) at the cost of introducing distortion between the frequency response of the continuous-time filter and the frequency response of the digital filter.

$$H(z) = H_c(s)_{s=\psi(z)} \text{ where } \psi(\cdot) \text{ is given by}$$

Euler's method or Forward difference

$$s = \frac{z-1}{T} \qquad \text{(intuition } \dot{x} \simeq \frac{x(t+T) - x(t)}{T}\text{)}$$

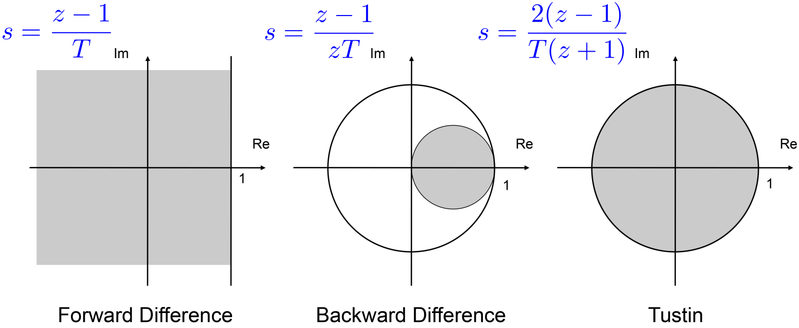
Backward difference

$$s = \frac{z-1}{zT} \qquad \text{(intuition } \dot{x} \simeq \frac{x(t) - x(t-T)}{T}\text{)}$$

Bilinear transformation or Tustin's transformation

$$s = \frac{2}{T}\frac{z-1}{z+1} \qquad \text{(or simply } \frac{z-1}{z+1}\text{, no normalization factor)}$$

Each of these transformations corresponds to a certain mapping between $s-$plane and $z-$plane. Below, the shaded region shows the set of points in the $z-$plane which corresponds to the left half of the s-plane (stable regions)



Backward difference and Tustin transformations applied to stable continuous systems result in stable discrete time systems (all the left plane poles get mapped into the unit disk). Not necessarily true for Euler's method.

Example: **first order low pass filter** $G_c(s) = \dfrac{1}{s+1}$. Sampling T.

Forward (possibly unstable)

$$G(z) = G_c\left(\frac{z-1}{T}\right) = \frac{T}{z+(T-1)} \qquad \text{pole } |T-1| > 1?$$

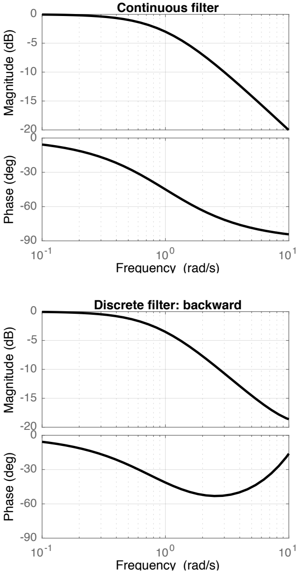
Backward (stable)

$$G(z) = G_c\left(\frac{z-1}{zT}\right) = \frac{\frac{T}{1+T}z}{z - \frac{1}{(1+T)}} \qquad \text{pole } \left|\frac{1}{1+T}\right| < 1$$

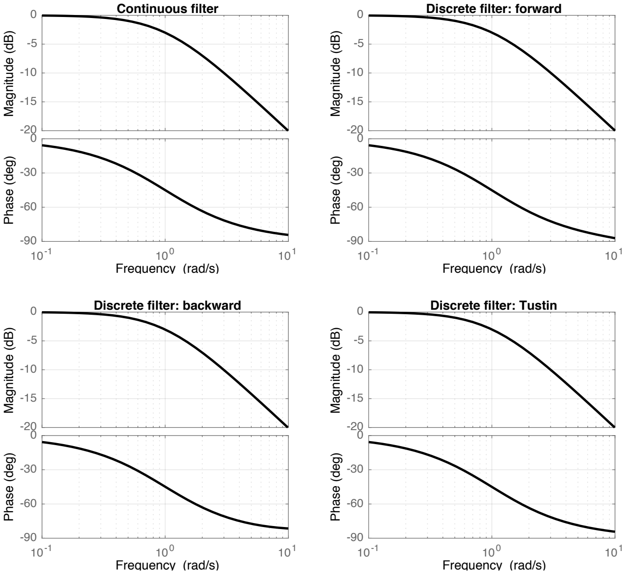
Tustin (stable)

$$G(z) = G_c\left(\frac{2}{T}\frac{z-1}{z+1}\right) = \frac{\frac{T}{T+2}(z+1)}{z+\frac{T-2}{T+2}} \qquad \text{pole } \left|\frac{T-2}{T+2}\right| < 1$$

Frequency response distortion of discretized filters.
Sampling $T = 0.25$ ($\omega_{max} = \frac{\pi}{T} = 4\pi$ rad/s).



The frequency response is recovered as $T \to 0$.
Sampling $T = 0.01$ ($\omega_{max} = \frac{\pi}{T} = 100\pi$ rad/s).

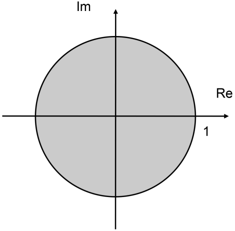


Module C

Bilinear transform in detail

Bilinear transform: stability is preserved

$$s = \psi(z) = \frac{z-1}{z+1}$$



Solve for z: $z = \psi^{-1}(s) = \dfrac{1+s}{1-s}$

For $s = \lambda + j\omega$:

$$|z|^2 = zz^* = \psi^{-1}(\lambda + j\omega)(\psi^{-1}(\lambda + j\omega))^* = \frac{(1+\lambda)^2 + \omega^2}{(1-\lambda)^2 + \omega^2}$$

If $\lambda = 0$ then

$$|z|^2 = \frac{1+\omega^2}{1+\omega^2} = 1 \ \rightarrow \ \text{unit circle}$$

If $\lambda < 0$ then $(1+\lambda)^2 < (1-\lambda)^2$ thus

$$|z|^2 = \frac{(1+\lambda)^2 + \omega^2}{(1-\lambda)^2 + \omega^2} < 1 \ \rightarrow \ \text{inside unit circle}$$

Bilinear transform: frequency warping

$$s = \psi(z) = \frac{z-1}{z+1}$$

Analog prototype filter: $G_c(s)$. Frequency response $G_c(j\omega)$.
Digital filter: $G(z) = G_c(\psi(z))$.

The normalized frequency response of the digital filter ($|\theta| \leq \pi$) is given by

$$G(e^{j\theta}) = G_c(\psi(e^{j\theta}))$$

where

$$\psi(e^{j\theta}) = \frac{e^{j\theta}-1}{e^{j\theta}+1} = \frac{e^{j\theta/2}-e^{-j\theta/2}}{e^{j\theta/2}+e^{-j\theta/2}} = \frac{j\sin(\theta/2)}{\cos\theta/2} = j\tan(\theta/2)$$

that is

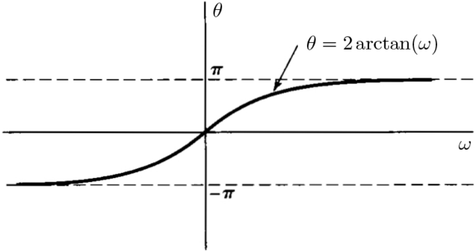
$$G(e^{j\theta}) = G_c(j\tan(\theta/2)) \qquad \text{frequency warping}$$

$$G(e^{j\theta}) = G_c(j\tan(\theta/2))$$

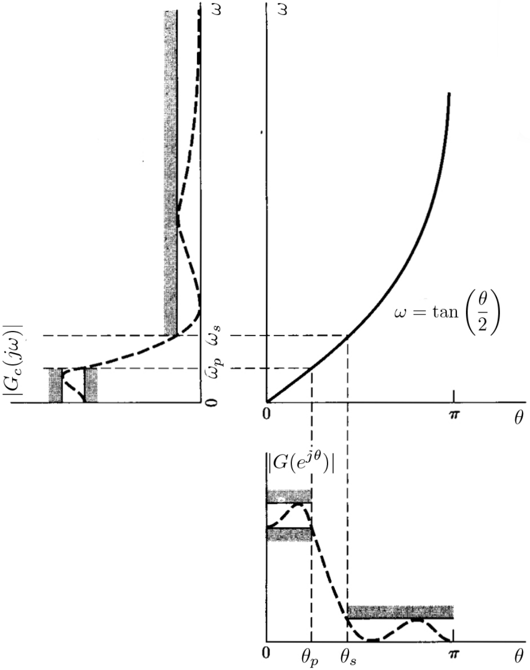
Inverse relation: the frequency response of the analog filter at ω is mapped into the frequency response of the digital filter at $\theta = 2\arctan(\omega)$.

$$\omega = \tan(\theta/2) \to \theta = 2\arctan(\omega) \qquad \text{frequency warping}$$

$$G_c(j\omega) = G\left(e^{j2\arctan(\omega)}\right)$$

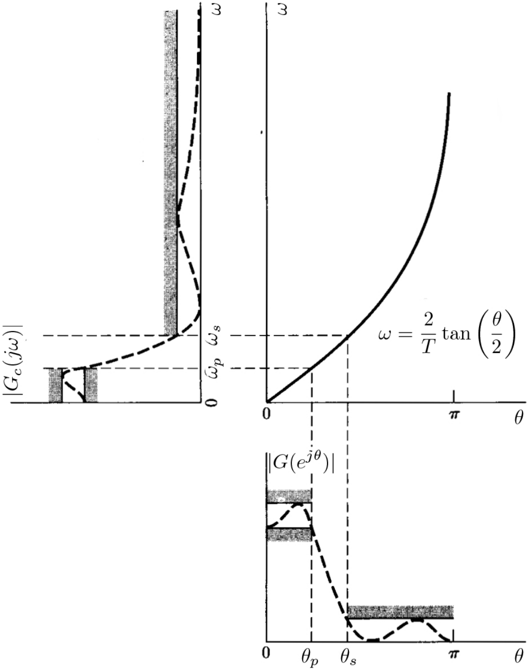


$$s = \frac{z-1}{z+1}$$



ω
ω_s
ω_p
0
$|G_c(j\omega)|$
$\omega = \tan\left(\frac{\theta}{2}\right)$
0
π
θ
$|G(e^{j\theta})|$
0
θ_p
θ_s
π
θ

$$s = \frac{2}{T}\frac{z-1}{z+1}$$



Example (low pass filter design)

Design a first order lowpass digital filter with -3dB frequency of 1kHz and a sampling frequency of 8kHz.

Consider the first order analogue lowpass filter

$$G_c(s) = \frac{1}{1 + \frac{s}{\omega_c}}$$

which has gain 1 (0dB) at $s = j0$ and gain 0.5 (-3dB) at $s = j\omega_c$ rad/s (cutoff frequency). Thus, the normalized digital cutoff frequency reads

$$\theta_c = (1000 \cdot 2\pi) \cdot T = \frac{1000 \cdot 2\pi}{8000} = \pi/4$$

The equivalent pre-warped analogue filter cutoff frequency:

$$\omega_c = \tan(\theta_c/2) = \tan(\pi/8) = 0.4142$$

Example (low pass filter design)

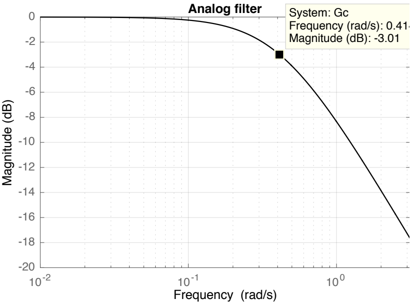
Apply now the bilinear transform $s = \psi(z) = \frac{z-1}{z+1}$.

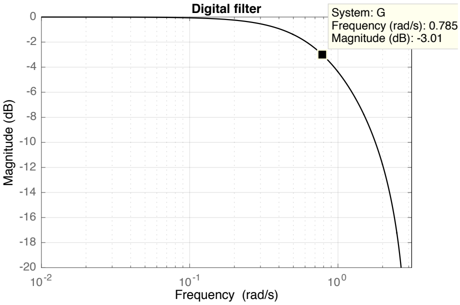
$$
\begin{aligned}
G(z) &= G_c(\psi(z)) = \frac{1}{1+\frac{\psi(z)}{\omega_c}} = \frac{1}{1+\frac{z-1}{(z+1)\omega_c}} \\
&= \frac{(z+1)\omega_c}{(z+1)\omega_c + z - 1} = \frac{(z+1)\omega_c}{(1+\omega_c)z + (\omega_c - 1)} \\
&= \frac{(z+1)\frac{\omega_c}{(1+\omega_c)}}{z + \frac{(\omega_c-1)}{(1+\omega_c)}} = \frac{0.2929(z+1)}{z - 0.4142}
\end{aligned}
$$

whose implementation reads

$$y_k = 0.4142 y_{k-1} + 0.2929(u_k + u_{k-1})$$

Example (low pass filter design)





Module D

Band transformations

Transformation between different filter types

Analogue prototypes are typically lowpass. Standard transformation can be used to convert lowpass fprototype into other types.

Assuming a lowpass prototype with cutoff at 1:

- Lowpass to Lowpass:
  set $s = \frac{\bar{s}}{\omega_c}$ to change the cutoff frequency to ω_c.
- Lowpass to Highpass:
  set $s = \frac{\omega_c}{\bar{s}}$ to get highpass with cutoff frequency at ω_c.
- Lowpass to Bandpass:
  set $s = \frac{\bar{s}^2 + \omega_l \omega_u}{\bar{s}(\omega_u - \omega_l)}$ to get bandpass with lower cutoff at ω_l and upper cutoff at ω_u.
- Lowpass to Bandstop:
  set $s = \frac{\bar{s}(\omega_u - \omega_l)}{\bar{s}^2 + \omega_l \omega_u}$ to get bandstop with lower cutoff at ω_l and upper cutoff at ω_u.

Transformation between different filter types

- Lowpass to Lowpass by $s = \frac{\bar{s}}{\omega_c}$

$$\frac{1}{s+1} \quad \rightarrow \quad \frac{1}{\frac{\bar{s}}{\omega_c}+1} = \frac{\omega_c}{\bar{s}+\omega_c}$$

after transformation cutoff at ω_c, $|G(j0)| = 1$, $|G(j\infty)| = 0$.

- Lowpass to Highpass by $s = \frac{\omega_c}{\bar{s}}$

$$\frac{1}{s+1} \quad \rightarrow \quad \frac{1}{\frac{\omega_c}{\bar{s}}+1} = \frac{\bar{s}}{\bar{s}+\omega_c}$$

after transformation cutoff at ω_c, $|G(j0)| = 0$, $|G(j\infty)| = 0$.

- try the others...

Module E

Matlab code

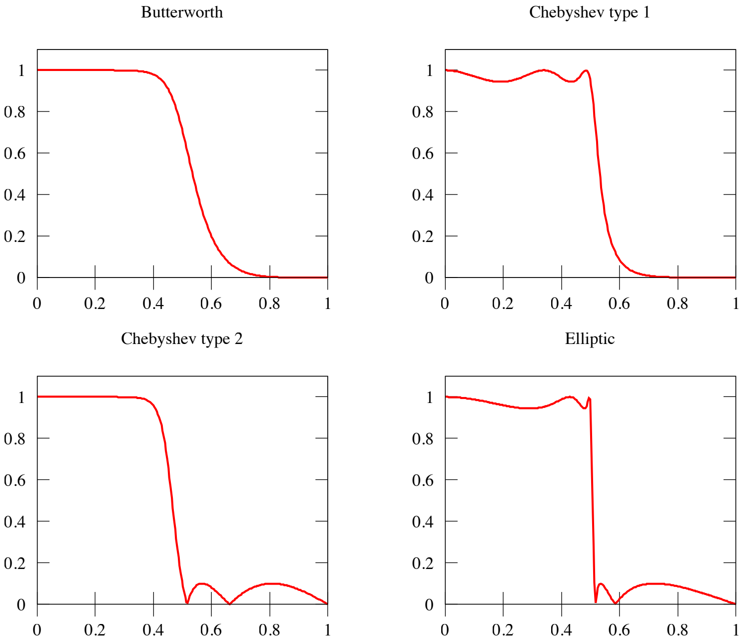
Useful Matlab code:

▸ Matlab: IIR filter design

▸ Matlab: analog filters

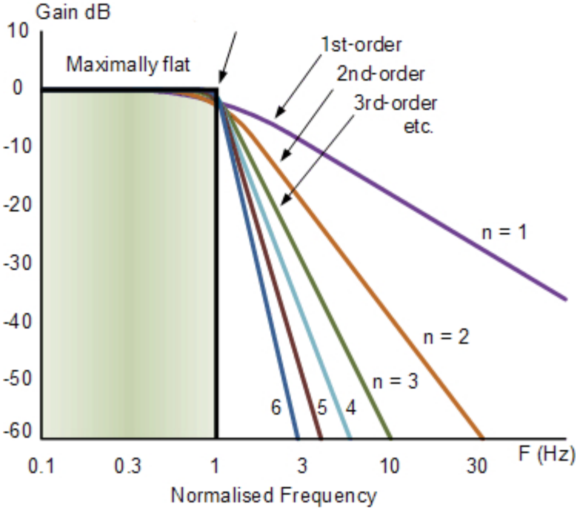
▸ Matlab: analog filters comparison

Appendix - Classical analog prototypes


Butterworth
Chebyshev type 1
Chebyshev type 2
Elliptic
0
0.2
0.4
0.6
0.8
1

Butterworth: Nth-order lowpass, $G_c(s)$ satisfies

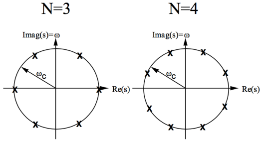
$$G_c(s)G_c(-s) = \frac{1}{1 + \left(\frac{s}{j\omega c}\right)^{2N}}$$



Butterworth: Nth-order lowpass, $G_c(s)$ satisfies

$$G_c(s)G_c(-s) = \frac{1}{1 + \left(\frac{s}{j\omega c}\right)^{2N}}$$

- Unit DC gain, $G_c(j0) = 1$.
- -3dB cutoff frequency at $s = j\omega_c$.
- $G_c(s)G_c(-s)$ poles satisfies $\left(\frac{s}{j\omega c}\right)^{2N} = -1$, i.e. $s = j\omega_c e^{\frac{j(2k+1)\pi}{2N}}$



- If p_i is a root of $G_c(s)$ then $-p_i$ is a root of $G_c(-s)$. Thus the poles of $G_c(s)$ are those roots lying in the left half plane (stability), so that

$$G_c(s) = \prod_{i=1}^{P} \frac{1}{s + p_i}$$

- Matlab routine `[B,A] = butter(N,Wn)` designs digital Butterworth filters (using bilinear transform) ‣ Matlab: butter .