INFINITE IMPULSE RESPONSE DIGITAL FILTERS

Paolo Favaro

Chapter 5 of Textbook

(original material from John Thompson)

Contents

- Digital filter design
- Analogue prototype filters
- Bilinear transformation filter design

Why Digital Filters?

- Easily implement low cutoff frequencies
- Digital filters are programmable
- Digital filters do not suffer from aging or thermal variations
- Digital filters are now very cheap to implement

Infinite Impulse Response Filters

- Design of analogue RLC filters is much studied, well understood
- Take analogue designs and apply them to digital domain
- Digital designs are based on analogue prototype filters, eg Butterworth or Chebychev

Analogue Prototype Filters

- The ideal brick wall filter:

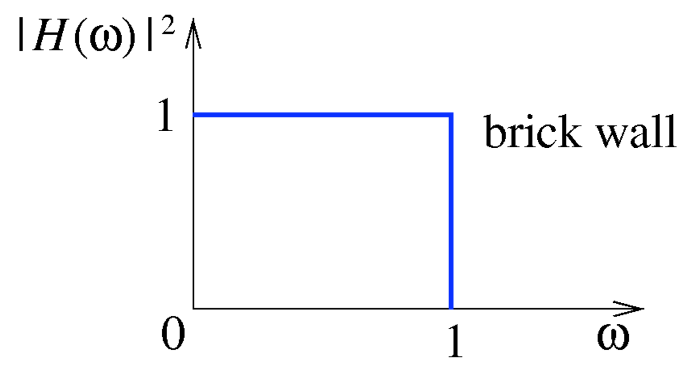


Fig5.1

- Model $H(\omega)$ as: $$|H(\omega)|^2 = \frac{1}{1+F(\omega^2)}$$
- Consider 2 candidate functions $F(\omega^2)$

Butterworth Polynomials

- The Butterworth function: $F(\omega^2) = \omega^{2n}$
- This function approximates $H(\omega)$ best around $\omega=0 \rightarrow F(\omega^2) \approx 0$ so $|H(\omega)| \approx 1$
- Approximation not so good for $\omega=1$, $F(\omega^2)=1$ so $|H(\omega)|^2 = 0.5$
- The function $F(\omega^2)$ increases with ω, so no ripple

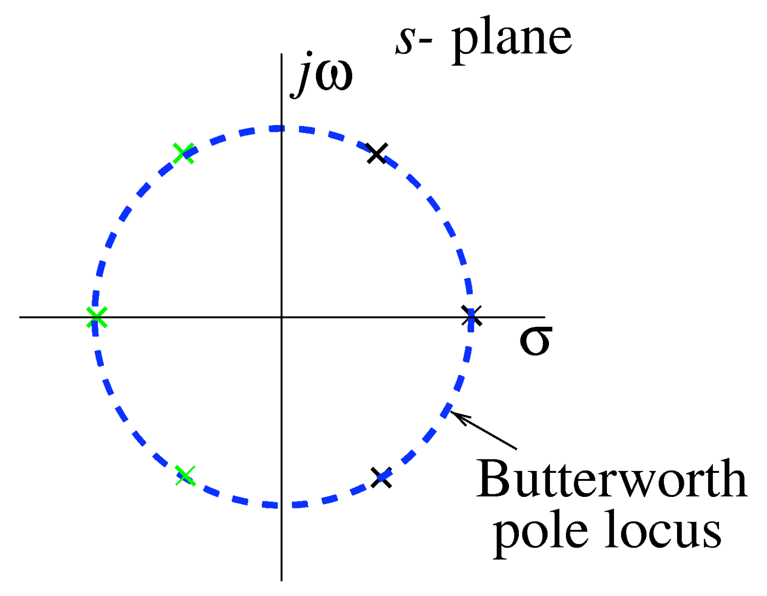
- To find the poles, write $p=j\omega$:

$$H(\omega)=\frac{1}{\sqrt{1+\omega^{2n}}}\Rightarrow H(p)=\frac{1}{\sqrt{1+p^{2n}}} \text{ for n even}$$

- So we have to solve $p^{2n}=-1$

Soln is $2n$-th roots of minus one: $\exp(j[2k-1]\pi/2n)$ $(k=1\ldots2n)$

Fig5.2



Butterworth: Amplitude Response

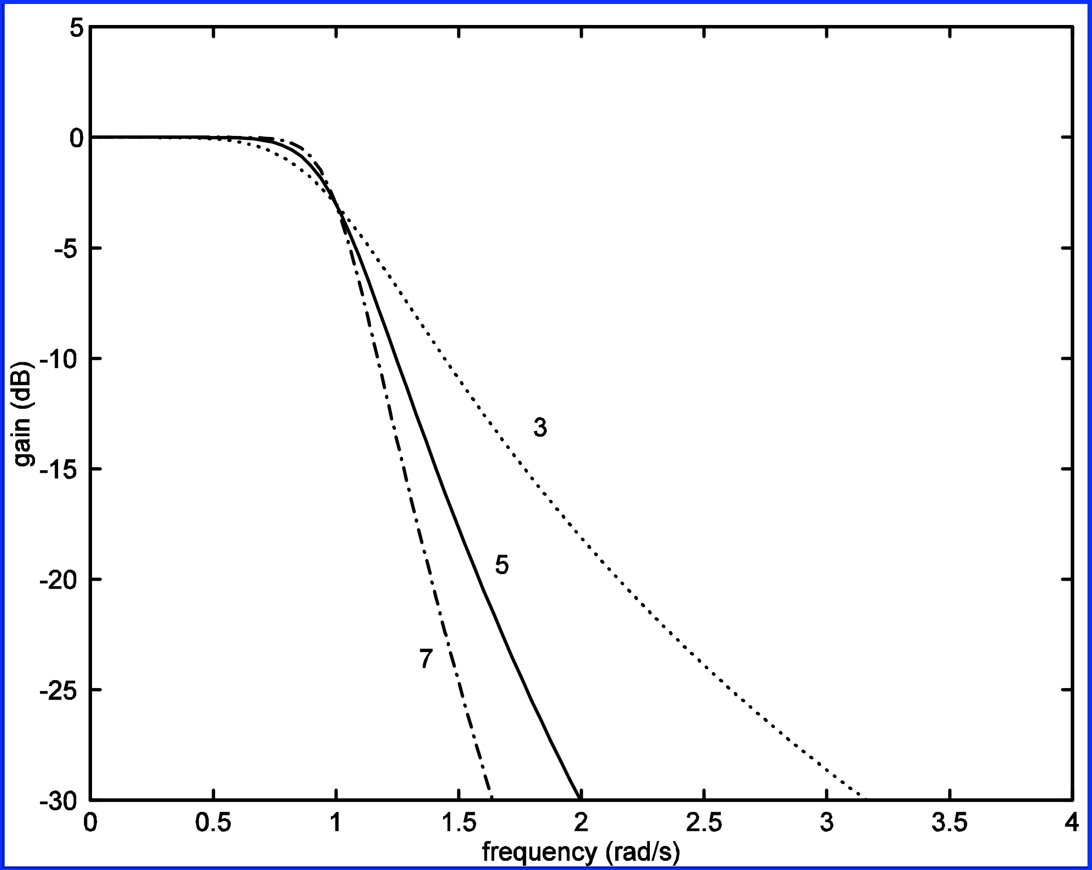


Fig5.3a

Butterworth: Phase Response

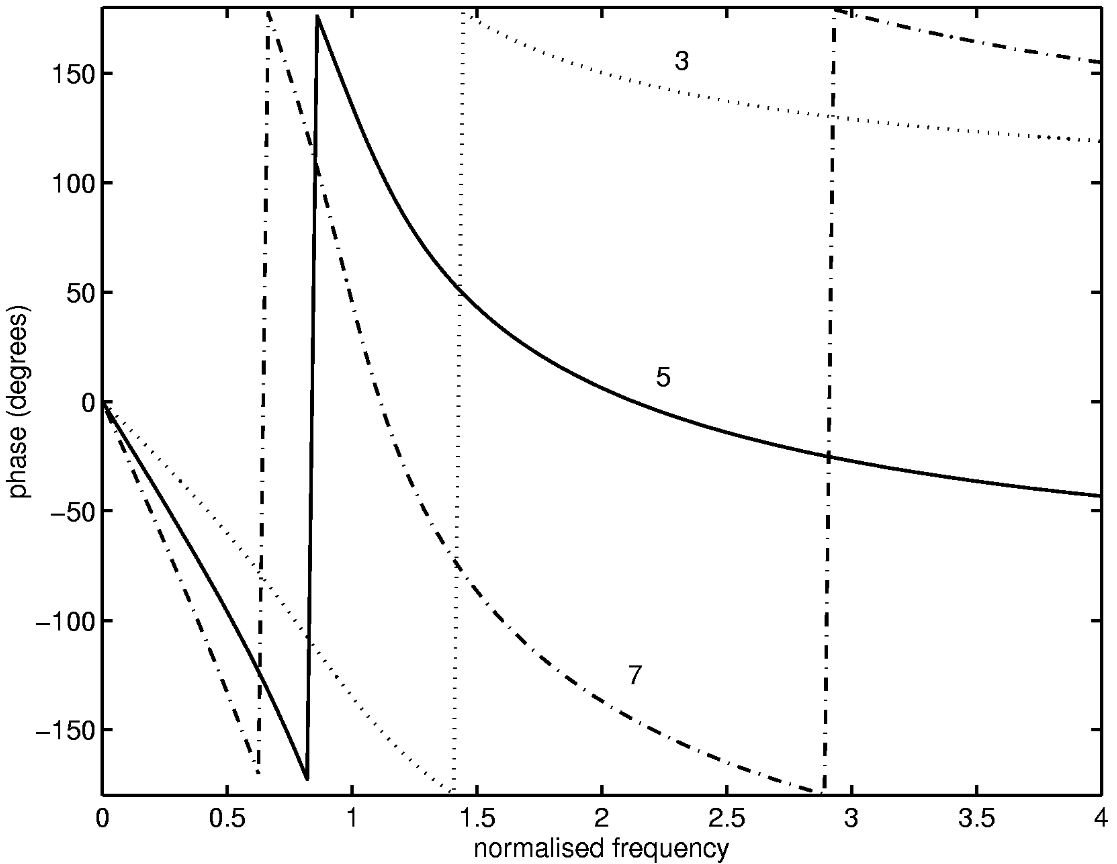


Fig5.3b

Chebychev Polynomials

- The Chebychev function:

$$F(\omega^2) = \varepsilon^2 C_n^2(\omega), \text{ where:}$$

$$C_n^2(\omega) = \cos(n \cos^{-1} \omega) \text{ when } |\omega| < 1$$

$$= \cosh(n \cosh^{-1} \omega) \text{ when } |\omega| > 1$$

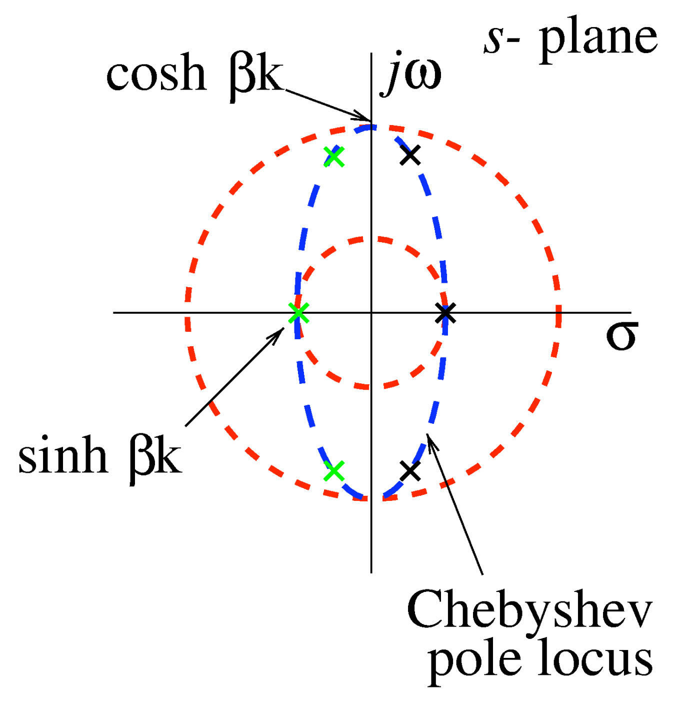
- Approximation good at $\omega = 0$ and *1*
  - Maximise roll-off rate in stopband
  - BUT now get ripple in the passband

- To find the poles, define parameter β_k

$$\beta_k = \frac{1}{n}\sinh^{-1}\frac{1}{\varepsilon}$$

Poles lie on an ellipse defined by β_k

Fig5.2



Chebychev: Amplitude Response

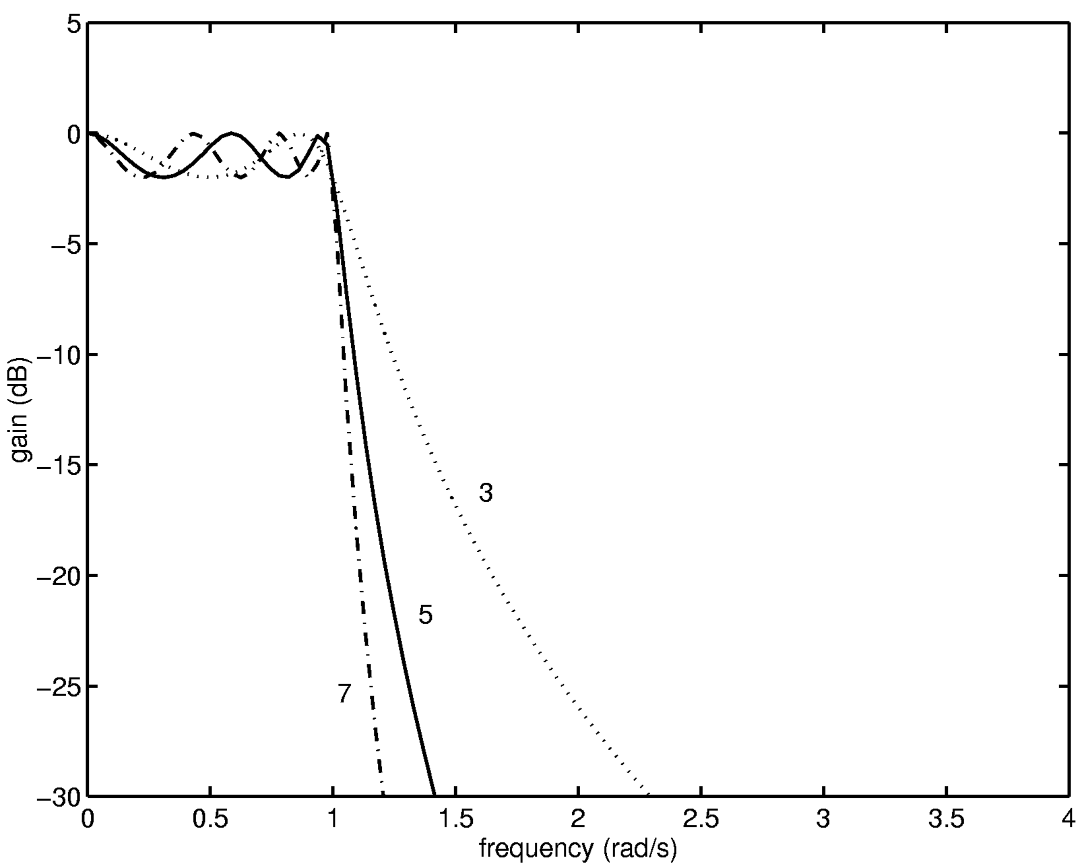


Fig5.4a

Chebychev: Phase Response

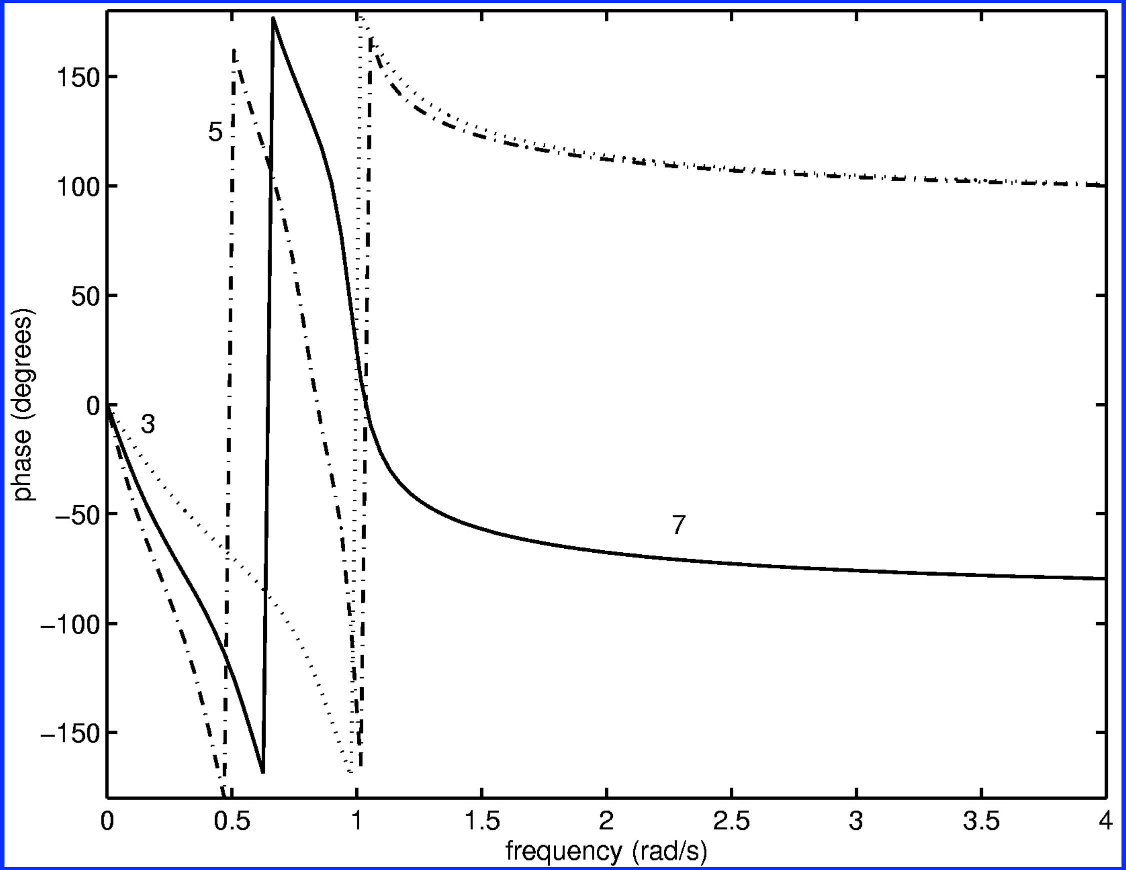


Fig5.4b

Digital IIR Filter Design

- No technique can perfectly map *H(s)* to *H(z)*
- Three main approaches:
  - ✦ Impulse Invariant
  - ✦ Matched *z*-transform: map poles/zeros
  - ✦ Bilinear transformation: map *s*-plane to *z*-plane

Bilinear Transformation

- Map vertical *s*-plane axis onto *z*-plane unit circle:

$$\omega_a = \frac{2}{\Delta t}\tan\left(\frac{\omega_d \Delta t}{2}\right)$$

- The total analogue frequency response is *compressed* into a finite digital one

$$s = j\omega_a = \frac{2(1 - z^{-1})}{\Delta t(1 + z^{-1})} \qquad z = e^{j\omega_d \Delta t}$$

Frequency Compression

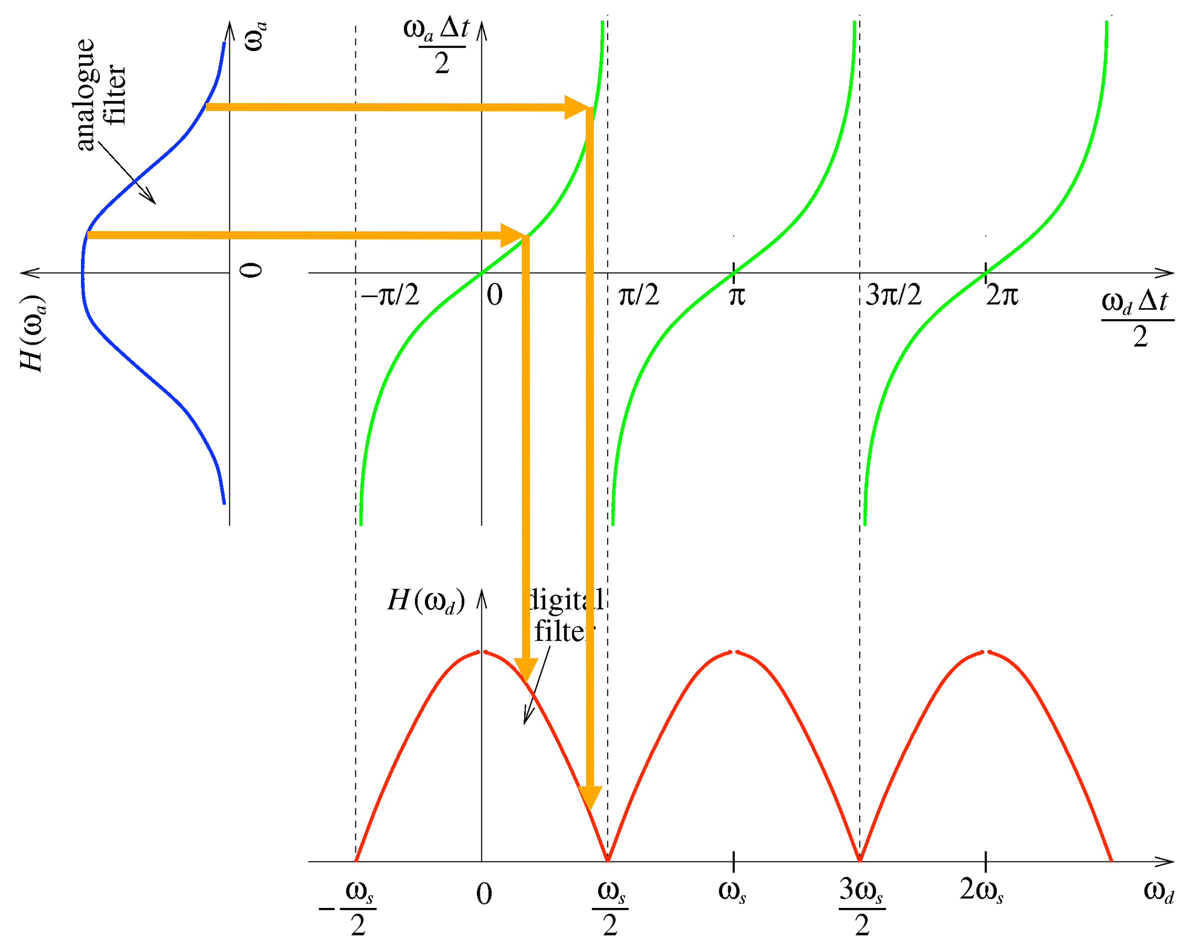


Fig5.9

Bilinear Pre-warping

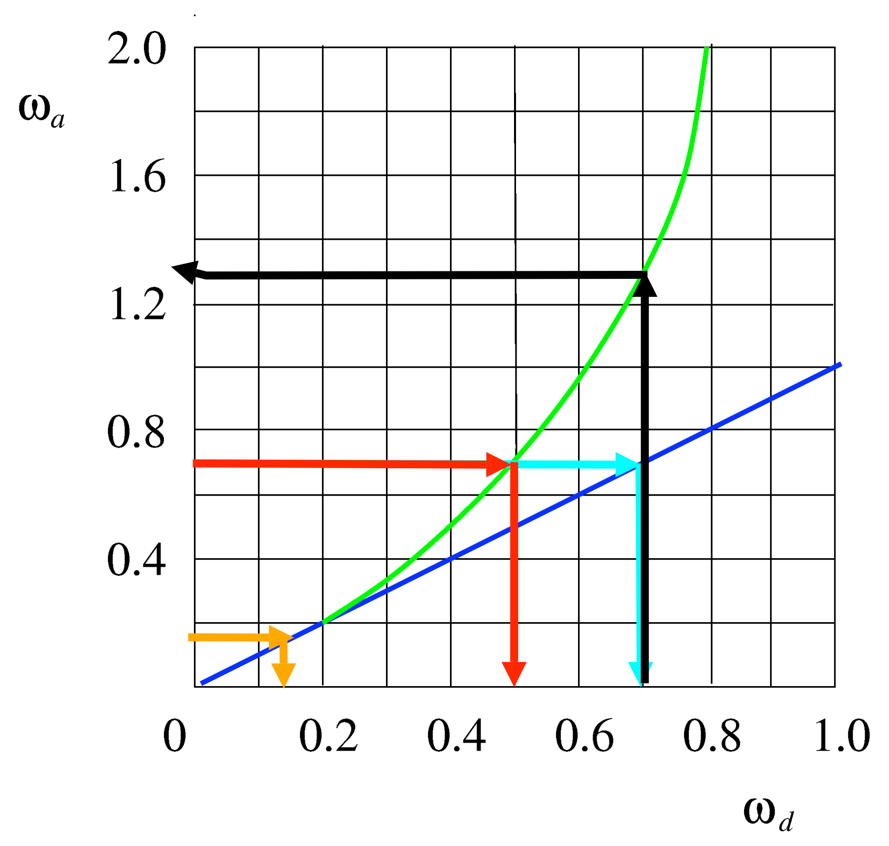


Fig5.10

Bilinear Design Example

Problem: Design a 2nd order Butterworth filter with cutoff ω_c=628 rad/s, sampling freq ω_s=5024 rad/s

- The normalised Butterworth filter is:

$$H(s) = \frac{1}{1+\sqrt{2}s+s^2}$$

- Calculate pre-warping frequency:

$$\omega_a = \left(\frac{2}{1/800}\right)\tan\left(\frac{2\pi 100}{2\times 800}\right) = 663 \text{ rad/s}$$

- De-normalise using : $s \mapsto s/\omega_a = s/663$

$$H(s) = \frac{1}{1 + (\sqrt{2}s/663) + (s/663)^2}$$

- Apply bilinear transformation:

$$s = \frac{2(1 - z^{-1})}{\Delta t(1 + z^{-1})}$$

- Applying bilinear transformation gives:

$$H(z) = \frac{1}{\left(\frac{2\times 800(1-z^{-1})}{663(1+z^{-1})}\right)^{2} + \sqrt{2}\left(\frac{2\times 800(1-z^{-1})}{663(1+z^{-1})}\right) + 1}$$

- Algebraic simplification leads to:

$$H(z) = \frac{0.098 + 0.195z^{-1} + 0.098z^{-2}}{1 - 0.942z^{-1} + 0.333z^{-2}}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{0.098 + 0.195z^{-1} + 0.098z^{-2}}{1 - 0.942z^{-1} + 0.333z^{-2}}$$

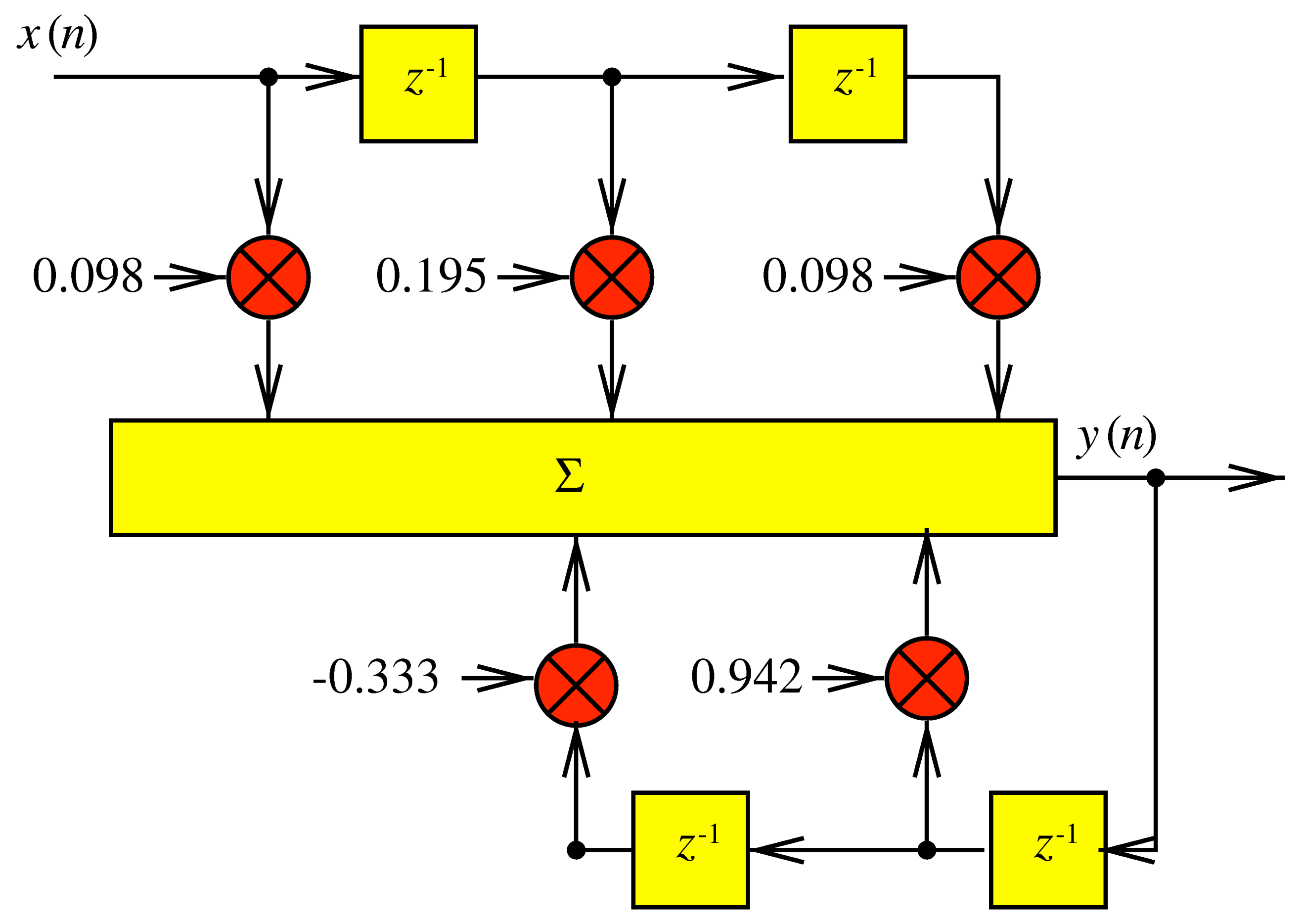
- Multiplying out:

$$Y(z)(1 - 0.942z^{-1} + 0.333z^{-2}) = X(z)(0.098 + 0.195z^{-1} + 0.098z^{-2})$$

- Finally can apply inverse *z*-transform to yield the *difference equation*:

$$\begin{aligned} y(n) = 0.098x(n) + 0.195x(n-1) + 0.098x(n-2) \\ + 0.942y(n-1) - 0.333y(n-2) \end{aligned}$$

IIR Filter Structure



Filter Frequency Response

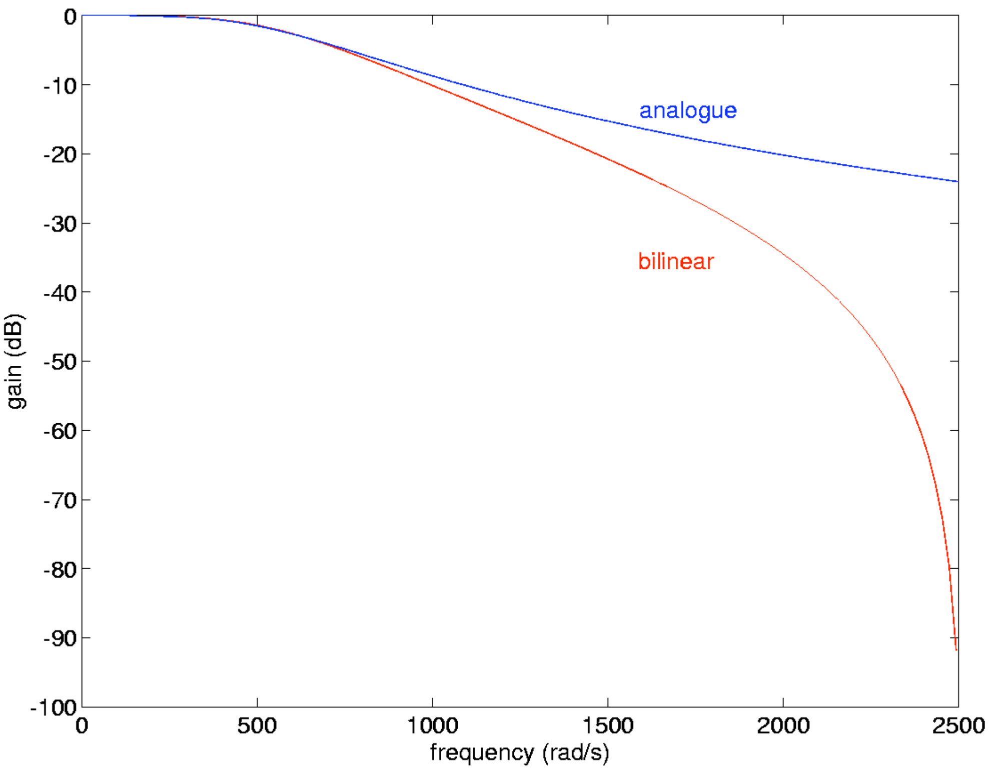


Fig5.12

Filter Phase Response

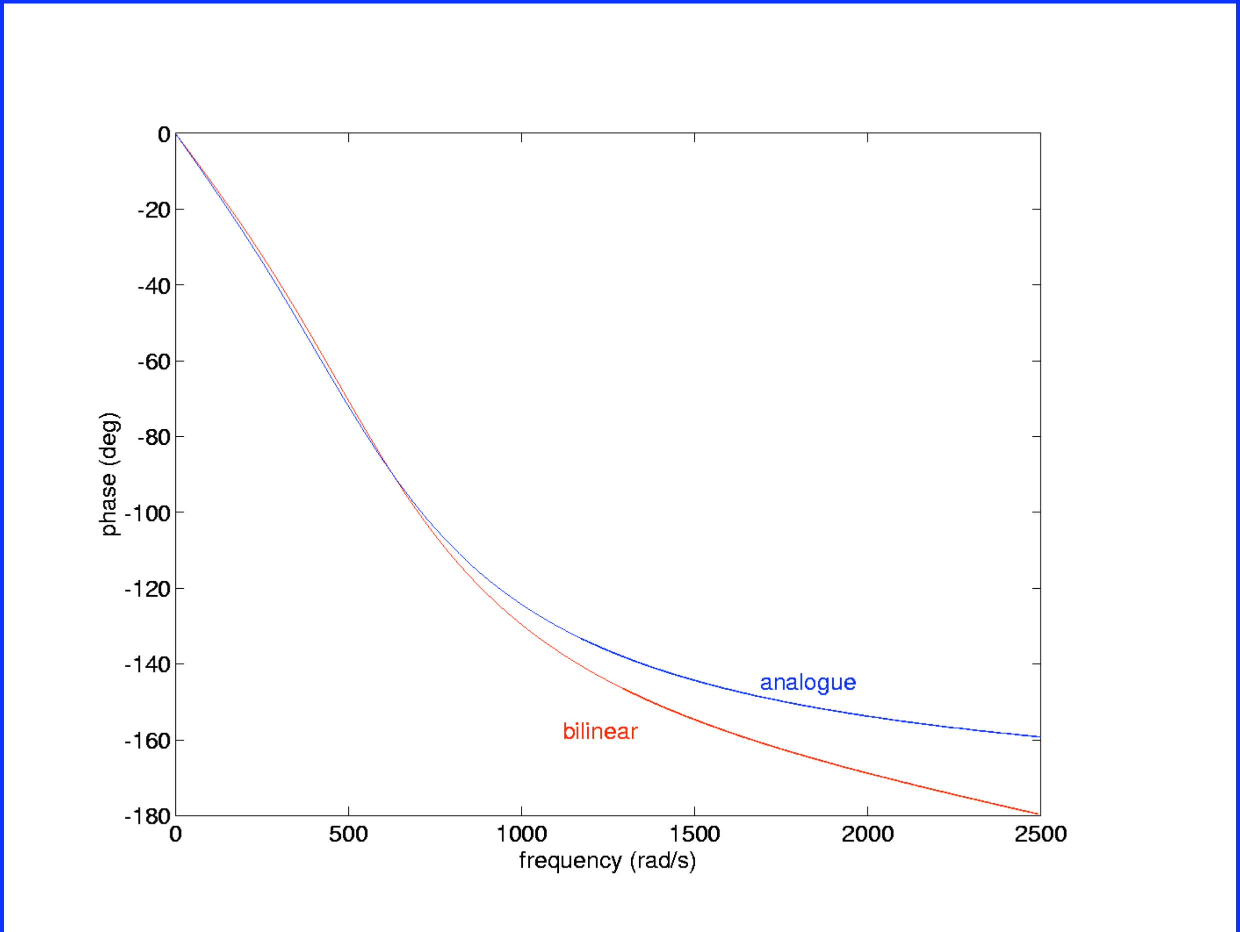


Fig5.13

Bilinear Design Summary

1. Calculate pre-warping analogue cutoff frequency
2. Denormalise filter transfer function using pre-warping cutoff
3. Apply bilinear transform and simplify
4. Use inverse *z*-transform to obtain difference equation