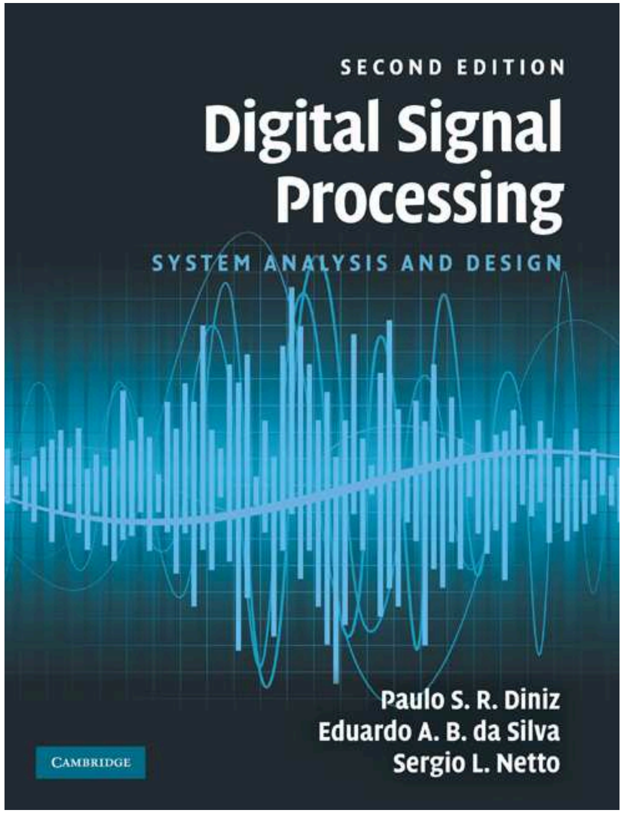
IIR Filter Approximations



Paulo S. R. Diniz

Eduardo A. B. da Silva

Sergio L. Netto

diniz,eduardo,sergioln@lps.ufrj.br

September 2010

Contents

Contents (cont.)

Introduction

- This chapter deals with the design methods in which a desired frequency response is approximated by a transfer function consisting of a ratio of polynomials. In general, this type of transfer function yields an impulse response of infinite duration. Therefore, the systems approximated in this chapter are commonly referred to as infinite-duration impulse-response (IIR) filters.

- In general, IIR filters are able to approximate a prescribed frequency response with fewer multiplications than FIR filters. For that matter, IIR filters can be more suitable for some practical applications, especially those ones involving real-time signal processing.

Introduction

- In Section 6.2, we study the classical methods of analog filter approximation, namely the Butterworth, Chebyshev, and elliptic approximations. These methods are the most widely used for approximations meeting prescribed magnitude specifications. They originated in the continuous-time domain and their use in the discrete-time domain requires an appropriate transformation.
- We then address, in Section 6.3, two approaches that transform a continuous-time transfer function into a discrete-time transfer function, the impulse-invariance and bilinear transformation methods.
- Section 6.4 deals with frequency transformation methods in the discrete-time domain. These methods allow the mapping of a given filter type to another, for example the transformation of a given lowpass into a desired bandpass filter.

Introduction

- In applications where magnitude and phase specifications are imposed, we can approximate the desired magnitude specifications by one of the classical transfer functions and design a phase equalizer to meet the phase specifications. As an alternative, we can carry out the design entirely in the digital domain, by using optimization methods to design transfer functions satisfying the magnitude and phase specifications simultaneously. Section 6.5 covers a procedure to approximate a given frequency response iteratively, employing a nonlinear optimization algorithm.
- In Section 6.6, we address the situations where an IIR digital filter must present an impulse response similar to a given discrete-time sequence. This problem is commonly known as time-domain approximation.
- Finally, we present some hands-on experiments with IIR filters in the Do-it-yourself section.

Analog filter approximations

- This section covers the classical approximations for normalized-lowpass analog filters.
- Normalized filters are derived from standard ones through a simple variable scaling. The original filter is then determined by reversing the frequency transformation previously applied.
- In this section, to avoid any source of confusion, a normalized analog frequency is always denoted by a primed variable such as Ω'.
- The other types of filters, such as the denormalized-lowpass, highpass, bandstop, and bandpass filters, are obtained from the normalized-lowpass prototype through frequency transformations, which are also addressed in this section.

Analog filter specification

- An important step in the design of an analog filter is the definition of the desired magnitude and/or phase specifications that should be satisfied by the filter frequency response.
- Usually, a classical analog filter is specified through a region of the $\Omega \times H(j\Omega)$ plane where its frequency response must be contained.
- This is illustrated in Figure 1 for a lowpass filter.
- In this figure, Ω_p and Ω_r denote the passband and stopband edge frequencies, respectively.
- The frequency region between Ω_p and Ω_r is the so-called transition band where no specification is provided.
- In addition, the maximum ripples in the passband and the stopband are denoted by δ_p and δ_r, respectively.

Analog filter specification

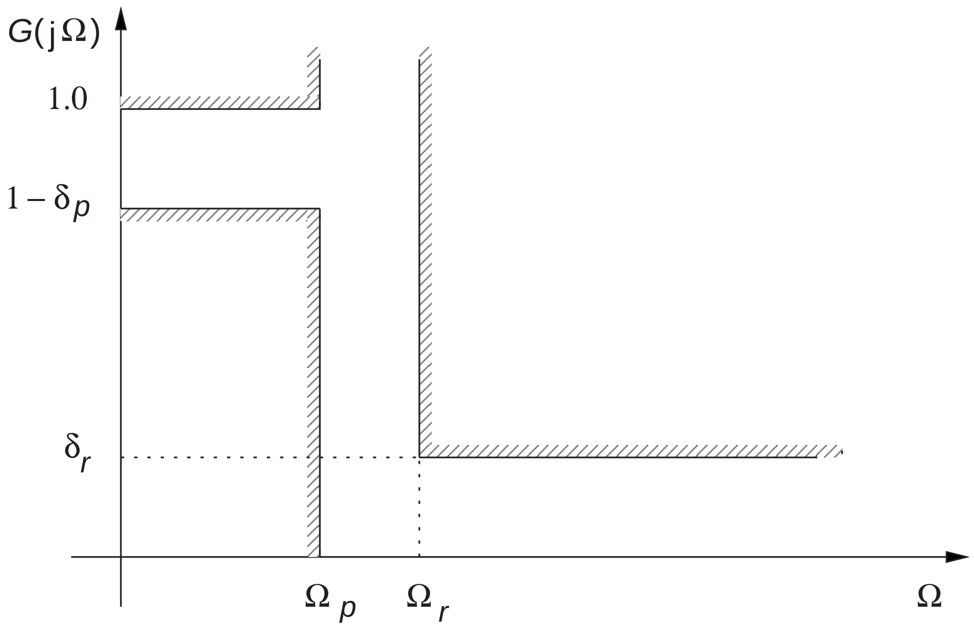


Figure 1: Typical gain specifications of a lowpass filter.

Analog filter specification

- Alternatively, the specifications can be given in decibels (dB), as shown in Figure 2a, in the case of gain specifications.
- Figure 2b shows the same filter specified in terms of attenuation instead of gain.
- The relationships between the parameters of these three representations are given in Table 1.
- For historical reasons, in this chapter, we work with the attenuation specifications in dB. Using the relationships given in Table 1, readers should be able to transform any other format into the set of parameters that characterize the attenuation in dB.

Analog filter specification

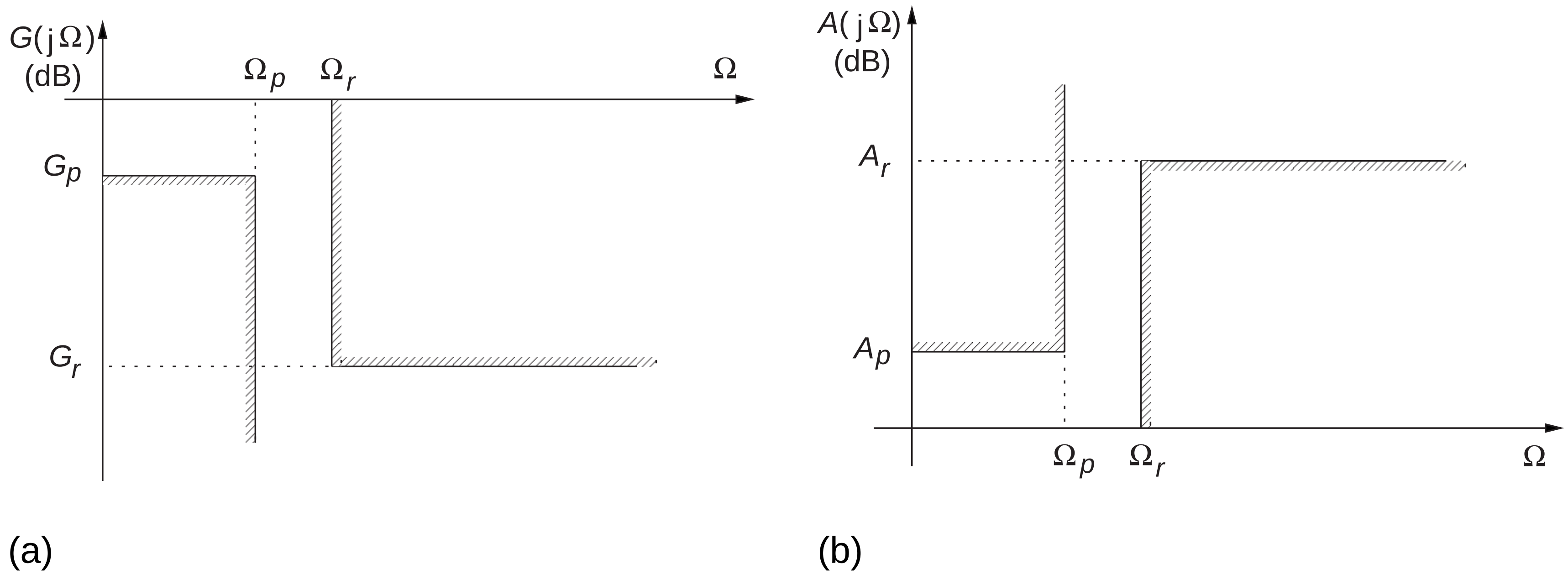


Figure 2: Typical specifications of a lowpass filter in dB: (a) gain; (b) attenuation.

Analog filter specification

Table 1: Relationships among the parameters for the gain, gain in dB, and attenuation in dB specification formats.

	Ripple	Gain [dB]	Attenuation [dB]
Passband	δ_p	$G_p = 20\log_{10}(1-\delta_p)$	$A_p = -G_p$
Stopband	δ_r	$G_r = 20\log_{10}\delta_r$	$A_r = -G_r$

Butterworth approximation

- Usually, the attenuation of an all-pole normalized-lowpass filter (that is, with $\Omega'_p = 1$), is expressed by an equation of the following type:

$$|A(j\Omega')|^2 = 1 + |E(j\Omega')|^2 \tag{1}$$

where $A(s')$ is the desired attenuation function and $E(s')$ is a polynomial which has low magnitude at low frequencies and large magnitude at high frequencies.

- The Butterworth approximation is characterized by a maximally flat magnitude response at $\Omega' = 0$.

- In order to achieve this property, we choose $E(j\Omega')$ as

$$E(j\Omega') = \epsilon\,(j\Omega')^n \tag{2}$$

where ϵ is a constant and n is the filter order.

Butterworth approximation

- Equation (1) then becomes

$$|A(j\Omega')|^2 = 1 + \epsilon^2 (\Omega')^{2n} \tag{3}$$

resulting in the fact that the first $(2n - 1)$ derivatives of the attenuation function at $\Omega' = 0$ are equal to zero, as desired in the Butterworth approximation.

- The parameter ϵ depends on the passband maximum attenuation A_p.

- In that manner, since

$$A_{\mathrm{dB}}(\Omega') = 20 \log_{10} |A(j\Omega')| = 10 \log_{10} \left[1 + \epsilon^2 (\Omega')^{2n}\right] \tag{4}$$

at $\Omega' = \Omega'_p = 1$, we must have that

$$A_p = A_{\mathrm{dB}}(1) = 10 \log_{10} \left(1 + \epsilon^2\right) \tag{5}$$

and then

$$\epsilon = \sqrt{10^{0.1A_p} - 1} \tag{6}$$

Butterworth approximation

- To determine the filter order required to meet the attenuation specification, A_r, in the stopband, at $\Omega' = \Omega'_r$ we must have that

$$A_r = A_{\text{dB}}(\Omega'_r) = 10\log_{10}\left[1 + \epsilon^2 \left(\Omega'_r\right)^{2n}\right] \tag{7}$$

- Therefore, n should be smallest integer such that

$$n \geq \frac{\log_{10}\left(\frac{10^{0.1A_r}-1}{\epsilon^2}\right)}{2\log_{10}\Omega'_r} \tag{8}$$

with ϵ as in equation (6).

- With n and ϵ available, one has to find the transfer function $A(s')$. We can factor $|A(j\Omega')|^2$ in equation (3) as

$$|A(j\Omega')|^2 = A(-j\Omega')A(j\Omega') = 1 + \epsilon^2{\Omega'}^{2n} = 1 + \epsilon^2[-(j\Omega')^2]^n \tag{9}$$

Butterworth approximation

- Using the analytical continuation for complex variables, that is, replacing $j\Omega'$ by s', we have that

$$A(s')A(-s') = 1 + \epsilon^2(-s'^2)^n \tag{10}$$

- In order to determine $A(s')$, we must then find the roots of $[1 + \epsilon^2(-s'^2)^n]$ and then choose which ones belong to $A(s')$ and which ones belong to $A(-s')$.

- The solutions of

$$1 + \epsilon^2(-s'^2)^n = 0 \tag{11}$$

are

$$s_i = \epsilon^{-\frac{1}{n}} e^{j\frac{\pi}{2}\left(\frac{2i+n+1}{n}\right)} \tag{12}$$

with $i = 1, 2, \ldots, 2n$.

Butterworth approximation

- These $2n$ roots are located at equally spaced positions on the circumference of radius $\epsilon^{-\frac{1}{n}}$ centered at the origin of the s plane.

- In order to obtain a stable filter, we choose the n roots p_i on the left-hand side of the s plane to belong to the polynomial $A(s')$.

- As a result, the normalized transfer function is obtained as

$$H'(s') = \frac{H'_0}{A(s')} = \frac{H'_0}{\prod_{i=1}^{n}(s' - p_i)} \tag{13}$$

where H'_0 is chosen so that $|H'(j0)| = 1$, and thus

$$H'_0 = \prod_{i=1}^{n}(-p_i) \tag{14}$$

Butterworth approximation

- An important characteristic of the Butterworth approximation is that its attenuation increases monotonically with frequency.
- In addition, it increases very slowly in the passband and quickly in the stopband.
- In the Butterworth approximation, if one wants to increase the attenuation one has to increase the filter order.
- However, if one sacrifices the monotonicity of the attenuation, a higher attenuation in the stopband can be obtained for the same filter order.
- A classic example of one such approximation is the Chebyshev approximation.

CAMBRIDGE UNIVERSITY PRESS

Chebyshev approximation

- The attenuation function of a normalized-lowpass Chebyshev filter is characterized by

$$|A(j\Omega')|^2 = 1 + \epsilon^2 C_n^2(\Omega') \tag{15}$$

where $C_n(\Omega')$ is a Chebyshev function of order n, which can be written in its trigonometric form as

$$C_n(\Omega') = \begin{cases} \cos(n \cos^{-1} \Omega'), & 0 \leq \Omega' \leq 1 \\ \cosh(n \cosh^{-1} \Omega'), & \Omega' > 1 \end{cases} \tag{16}$$

- These functions $C_n(\Omega')$ have the following properties

$$\begin{cases} 0 \leq C_n^2(\Omega') \leq 1, & 0 \leq \Omega' \leq 1 \\ C_n^2(\Omega') > 1, & \Omega' > 1 \end{cases} \tag{17}$$

Chebyshev approximation

- As a consequence, for the attenuation function defined in equation (15), the passband is placed in the frequency range $0 \le \Omega' \le \Omega'_p = 1$, the rejection band is in the range $\Omega' \ge \Omega'_r > 1$, as desired, and the parameter ϵ once again determines the maximum passband ripple.

- The Chebyshev functions defined above can also be expressed in polynomial form as

$$\begin{aligned} C_{n+1}(\Omega') + C_{n-1}(\Omega') &= \cos[(n+1)\cos^{-1}\Omega'] + \cos[(n-1)\cos^{-1}\Omega'] \\ &= 2\cos(\cos^{-1}\Omega')\cos(n\cos^{-1}\Omega') \\ &= 2\Omega' C_n(\Omega') \end{aligned} \tag{18}$$

with $C_0(\Omega') = 1$ and $C_1(\Omega') = \Omega'$.

Chebyshev approximation

- We can then generate higher order Chebyshev polynomials through the recursive relation above, that is

$$\left.\begin{aligned} C_2(\Omega') &= 2\Omega'^2 - 1 \\ C_3(\Omega') &= 4\Omega'^3 - 3\Omega' \\ &\vdots \\ C_{n+1}(\Omega') &= 2\Omega' C_n(\Omega') - C_{n-1}(\Omega') \end{aligned}\right\} \tag{19}$$

- Figure 3 depicts the Chebyshev functions for several values of n.

Chebyshev approximation

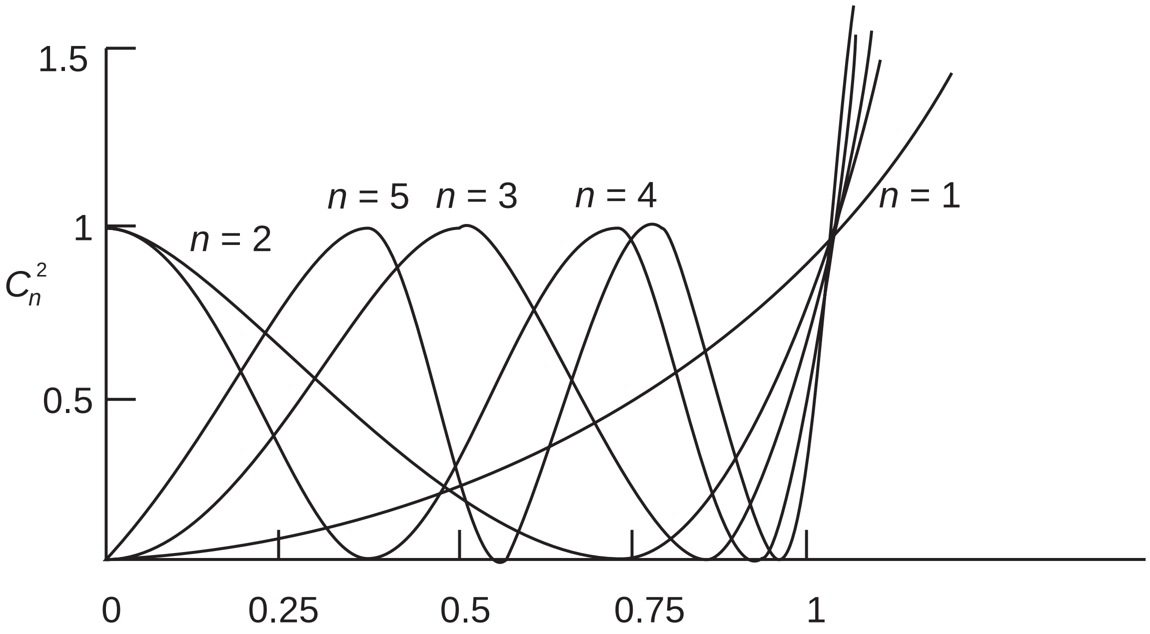


Figure 3: Chebyshev functions for $n = 1, 2, \ldots, 5$.

Chebyshev approximation

- Since $C_n(\Omega') = 1$ at $\Omega' = \Omega'_p$ =1, we have that

$$A_p = A_{\text{dB}}(1) = 10\log_{10}(1+\epsilon^2) \tag{20}$$

and then

$$\epsilon = \sqrt{10^{0.1A_p} - 1} \tag{21}$$

- From equations (15) and (16), when $\Omega' = \Omega'_r$, we find

$$A_r = A_{\text{dB}}(\Omega'_r) = 10\log_{10}\left[1 + \epsilon^2 \cosh^2\left(n\cosh^{-1}\Omega'_r\right)\right] \tag{22}$$

and thus the order of the normalized-lowpass Chebyshev filter that satisfies the required stopband attenuation is the smallest integer number that satisfies

$$n \geq \frac{\cosh^{-1}\sqrt{\dfrac{10^{0.1A_r} - 1}{\epsilon^2}}}{\cosh^{-1}\Omega'_r} \tag{23}$$

Chebyshev approximation

- Similarly to the Butterworth case (see equation (9)), we can now continue the approximation process by evaluating the zeros of $A(s')A(-s')$, with $s' = j\Omega'$.

- Since zero attenuation can never occur in the stopband, these zeros are in the passband region $0 \leq \Omega' \leq 1$, and thus, from equation (16), we have

$$\cos\left(n \cos^{-1} \frac{s'}{j}\right) = \pm\frac{j}{\epsilon} \tag{24}$$

- The above equation can be solved for s' by defining a complex variable p as

$$p = x_1 + jx_2 = \cos^{-1}\left(\frac{s'}{j}\right) \tag{25}$$

- Replacing this value of p in equation (24), we arrive at

$$\cos n(x_1 + jx_2) = (\cos nx_1 \cosh nx_2) - j(\sin nx_1 \sinh nx_2) = \pm\frac{j}{\epsilon} \tag{26}$$

Chebyshev approximation

- By equating the real parts of both sides of the above equation, we can deduce that

$$\cos nx_1 \cosh nx_2 = 0 \tag{27}$$

and considering that

$$\cosh nx_2 \geq 1, \qquad \text{for all } n, x_2 \tag{28}$$

we then have

$$\cos nx_1 = 0 \tag{29}$$

which yields the following $2n$ solutions:

$$x_{1i} = \frac{2i+1}{2n}\pi \tag{30}$$

for $i = 0, 1, \ldots, (2n-1)$.

Chebyshev approximation

- Now, by equating the imaginary parts of both sides of equation (26) and using the values of x_{1i} obtained in equation (30), it follows that

$$\sin nx_{1i} = \pm 1 \tag{31}$$

$$x_2 = \frac{1}{n}\sinh^{-1}\left(\frac{1}{\epsilon}\right) \tag{32}$$

- Since, from equations (32) and (25), the zeros of $A(s')A(-s')$ are given by

$$s'_i = \sigma'_i \pm j\Omega'_i = j\cos(x_{1i} + jx_2) = \sin x_{1i}\sinh x_2 + j\cos x_{1i}\cosh x_2 \tag{33}$$

for $i = 0, 1, \ldots, (2n-1)$, we have, from equations (25) and (30), that

$$\sigma_i = \pm\sin\left[\frac{\pi}{2}\left(\frac{2i+1}{n}\right)\right]\sinh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\epsilon}\right) \tag{34}$$

$$\Omega_i = \cos\left[\frac{\pi}{2}\left(\frac{2i+1}{n}\right)\right]\cosh\left(\frac{1}{n}\sinh^{-1}\frac{1}{\epsilon}\right) \tag{35}$$

Chebyshev approximation

- The calculated zeros belong to $A(s')A(-s')$. Analogously to the Butterworth case, we associate the n zeros, p_i, with negative real part to $A(s')$, in order to guarantee the filter stability.

- The above equations indicate that the zeros of a Chebyshev approximation are placed on an ellipse in the s plane, since equation (34) implies the following relation:

$$\left[\frac{\sigma_i}{\sinh(\frac{1}{n}\sinh^{-1}\frac{1}{\epsilon})}\right]^2 + \left[\frac{\Omega_i}{\cosh(\frac{1}{n}\sinh^{-1}\frac{1}{\epsilon})}\right]^2 = 1 \tag{36}$$

Chebyshev approximation

- The transfer function of the Chebyshev filter is then given by

$$H'(s') = \frac{H'_0}{A(s')} = \frac{H'_0}{\displaystyle\prod_{i=1}^{n}(s' - p_i)} \tag{37}$$

where H'_0 is chosen so that $A(s')$ satisfies equation (15), that is (see also Figure 3)

$$H'_0 = \begin{cases} \displaystyle\prod_{i=1}^{n}(-p_i), & \text{for } n \text{ odd} \\ 10^{-0.05A_p}\displaystyle\prod_{i=1}^{n}(-p_i), & \text{for } n \text{ even} \end{cases} \tag{38}$$

Chebyshev approximation

- It is interesting to note that in the Butterworth case the frequency response is monotone in both passband and stopband, and is maximally flat at $\Omega = 0$.
- In the case of the Chebyshev filters, the smooth passband characteristics are exchanged for steeper transition bands for the same filter orders.
- In fact, for a given prescribed specification, Chebyshev filters usually require lower-order transfer functions than Butterworth filters, owing to their equiripple behavior in the passband.

Elliptic approximation

- The two approximations discussed so far, namely the lowpass Butterworth and Chebyshev approximations, lead to transfer functions whose numerator is a constant and the denominator is a polynomial in s.
- These are called all-pole filters, because all their zeros are located at infinity.
- When going from the Butterworth to the Chebyshev filters, we have traded-off monotonicity and maximal flatness in the passband for higher attenuation in the stopband.
- At this point, it is natural to wonder whether we could also exchange the monotonicity in the stopband possessed by the Butterworth and Chebyshev filters for an even steeper transition band without increasing the filter order.
- This is indeed the case, as approximations with finite-frequency zeros can have transition bands with very steep slopes.

Elliptic approximation

- In practice, there are transfer function approximations with finite zeros which have equiripple characteristics in the passband and in the stopband, with the advantage that their coefficients can be computed using closed formulas.
- These filters are usually called elliptic filters, as their closed-form equations are derived based on elliptic functions, but they are also known as Cauer or Zolotarev filters.
- This section covers the lowpass elliptic filter approximation (the derivations are not detailed here, as they are beyond the scope of this book).
- In the following, we describe an algorithm to calculate the coefficients of elliptic filters which is based on the procedure described in the benchmark book by Antoniou.

Elliptic approximation

- Consider the following lowpass filter transfer function:

$$|H(j\Omega')| = \frac{1}{\sqrt{1 + R_n^2(\Omega')}} \tag{39}$$

where

$$R_n(\Omega') = \begin{cases} C_e \prod_{i=1}^{\frac{n}{2}} \left[\dfrac{\Omega'^2 - \left(\dfrac{\Omega_r'^2}{\Omega_i'^2} \right)^2}{\Omega'^2 - \Omega_i'^2} \right], & \text{for } n \text{ even} \\ C_o \Omega' \prod_{i=1}^{\frac{n-1}{2}} \left[\dfrac{\Omega'^2 - \left(\dfrac{\Omega_r'^2}{\Omega_i'^2} \right)^2}{\Omega'^2 - \Omega_i'^2} \right], & \text{for } n \text{ odd} \end{cases} \tag{40}$$

- The computation of $R_n(\Omega')$ requires the use of some elliptic functions.

Elliptic approximation

- All frequencies in equation (39) are normalized. The normalization procedure for the elliptic approximation is rather distinct from the one for the Butterworth and Chebyshev filters.
- Here, the frequency normalization factor is given by

$$\Omega_c = \sqrt{\Omega_p \Omega_r} \tag{41}$$

- In that manner, we have that

$$\Omega'_p = \frac{\Omega_p}{\Omega_c} = \sqrt{\frac{\Omega_p}{\Omega_r}} \tag{42}$$

$$\Omega'_r = \frac{\Omega_r}{\Omega_c} = \sqrt{\frac{\Omega_r}{\Omega_p}} \tag{43}$$

Elliptic approximation

- Defining

$$k = \frac{\Omega'_p}{\Omega'_r} = \frac{1}{\Omega'^2_r} \tag{44}$$

$$q_0 = \frac{1}{2}\left[\frac{1-(1-k^2)^{\frac{1}{4}}}{1+(1-k^2)^{\frac{1}{4}}}\right] \tag{45}$$

$$q = q_0 + 2q_0^5 + 15q_0^9 + 150q_0^{13} \tag{46}$$

$$\epsilon = \sqrt{\frac{10^{0.1A_p}-1}{10^{0.1A_r}-1}} \tag{47}$$

the specifications are satisfied if the filter order n is chosen through:

$$n \geq \frac{\log_{10}\dfrac{16}{\epsilon^2}}{\log_{10}\dfrac{1}{q}} \tag{48}$$

Elliptic approximation

- Having the filter order n, we can then determine the following parameters before proceeding with the computation of the filter coefficients:

$$\Theta = \frac{1}{2n} \ln \frac{10^{0.05A_p} + 1}{10^{0.05A_p} - 1} \tag{49}$$

$$\sigma = \left| \frac{2q^{\frac{1}{4}} \sum_{j=0}^{\infty} (-1)^j q^{j(j+1)} \sinh[(2j+1)\Theta]}{1 + 2 \sum_{j=1}^{\infty} (-1)^j q^{j^2} \cosh(2j\Theta)} \right| \tag{50}$$

$$W = \sqrt{(1 + k\sigma^2)\left(1 + \frac{\sigma^2}{k}\right)} \tag{51}$$

Elliptic approximation

- Also, for $i = 1, 2, \ldots, l$, where $l = \frac{n}{2}$ for n even and $l = \frac{n-1}{2}$ for n odd, we compute

$$\Omega_i' = \frac{2q^{\frac{1}{4}} \displaystyle\sum_{j=0}^{\infty} (-1)^j q^{j(j+1)} \sin \frac{(2j+1)\pi u}{n}}{1 + 2 \displaystyle\sum_{j=1}^{\infty} (-1)^j q^{j^2} \cos \frac{2j\pi u}{n}} \tag{52}$$

$$V_i = \sqrt{(1 - k\Omega_i'^2)\left(1 - \frac{\Omega_i'^2}{k}\right)} \tag{53}$$

where

$$\left.\begin{array}{ll} u = i, & \text{for } n \text{ odd} \\ u = i - \frac{1}{2}, & \text{for } n \text{ even} \end{array}\right\} \tag{54}$$

Elliptic approximation

- The infinite summations in equations (50) and (52) converge extremely quickly, and only two or three terms are sufficient to reach a very accurate result.
- The transfer function of a normalized-lowpass elliptic filter can be written as

$$H'(s') = \frac{H'_0}{(s'+\sigma)^m} \prod_{i=1}^{l} \frac{s'^2 + b_{2i}}{s'^2 + a_{1i}s' + a_{2i}} \tag{55}$$

where

$$\left.\begin{array}{ll} m = 0 \text{ and } l = \frac{n}{2}, & \text{for } n \text{ even} \\ m = 1 \text{ and } l = \frac{n-1}{2}, & \text{for } n \text{ odd} \end{array}\right\} \tag{56}$$

Elliptic approximation

- The coefficients of the above transfer function are calculated based on the parameters obtained from equations (44)–(53) as

$$b_{2i} = \frac{1}{\Omega_i'^2} \tag{57}$$

$$a_{2i} = \frac{(\sigma V_i)^2 + (\Omega_i' W)^2}{(1 + \sigma^2 \Omega_i'^2)^2} \tag{58}$$

$$a_{1i} = \frac{2\sigma V_i}{1 + \sigma^2 \Omega_i'^2} \tag{59}$$

$$H_0' = \begin{cases} \sigma \displaystyle\prod_{i=1}^{l} \frac{a_{2i}}{b_{2i}}, & \text{for } n \text{ odd} \\ 10^{-0.05 A_p} \sigma \displaystyle\prod_{i=1}^{l} \frac{a_{2i}}{b_{2i}}, & \text{for } n \text{ even} \end{cases} \tag{60}$$

Elliptic approximation

- Following this procedure, the resulting minimum stopband attenuation is slightly better than the specified value, being precisely given by

$$A_r = 10 \log_{10} \left(\frac{10^{0.1 A_p} - 1}{16 q^n} + 1 \right) \tag{61}$$

Frequency transformations

- The approximation methods presented so far are meant for designing normalized-lowpass filters.
- In this subsection, we address the issue of how a transfer function of a general lowpass, highpass, symmetric bandpass, or symmetric bandstop filter can be transformed into a normalized-lowpass transfer function, and vice versa.
- The procedure used here, the so-called frequency transformation technique, consists of replacing the variable s' in the normalized-lowpass filter by an appropriate function of s.
- In the following, we make a detailed analysis of the normalized-lowpass $\leftrightarrow$ bandpass transformation. The analyses of the other transformations are similar, and their expressions are summarized in Table 2.

Frequency transformations

- A normalized-lowpass transfer function $H'(s')$ can be transformed into a symmetric bandpass transfer function by applying the following substitution of variables

$$s' \leftrightarrow \frac{1}{a}\frac{s^2 + \Omega_0^2}{Bs} \tag{62}$$

where Ω_0 is the central frequency of the bandpass filter, B is the filter passband width, and a is a normalization parameter that depends upon the filter type.

Frequency transformations

- The parameters Ω_0, B, and a are determined as follows:

$$\Omega_0 = \sqrt{\Omega_{p_1}\Omega_{p_2}} \tag{63}$$

$$B = \Omega_{p_2} - \Omega_{p_1} \tag{64}$$

$$a = \frac{1}{\Omega_p'} = \begin{cases} 1, & \text{for any Butterworth or Chebyshev filter} \\ \sqrt{\dfrac{\Omega_r}{\Omega_p}}, & \text{for a lowpass elliptic filter} \\ \sqrt{\dfrac{\Omega_p}{\Omega_r}}, & \text{for a highpass elliptic filter} \\ \sqrt{\dfrac{\Omega_{r_2}-\Omega_{r_1}}{\Omega_{p_2}-\Omega_{p_1}}}, & \text{for a bandpass elliptic filter} \\ \sqrt{\dfrac{\Omega_{p_2}-\Omega_{p_1}}{\Omega_{r_2}-\Omega_{r_1}}}, & \text{for a bandstop elliptic filter} \end{cases} \tag{65}$$

Frequency transformations

- The value of a is different from unity for the elliptic filters because in this case the normalization is not $\Omega'_p = 1$, but $\sqrt{\Omega'_p \Omega'_r} = 1$ (see equations (41)–(6.43)).
- The frequency transformation in equation (62) has the following properties:
  - The frequency $s' = j0$ is transformed into $s = \pm j\Omega_0$.
  - Any complex frequency $s' = -j\Omega'$, corresponding to an attenuation of A_{dB} in the normalized-lowpass filter, is transformed into the two distinct frequencies

$$\Omega_1 = -\frac{1}{2}aB\Omega' + \sqrt{\frac{1}{4}a^2B^2\Omega'^2 + \Omega_0^2} \tag{66}$$

$$\overline{\Omega}_1 = -\frac{1}{2}aB\Omega' - \sqrt{\frac{1}{4}a^2B^2\Omega'^2 + \Omega_0^2} \tag{67}$$

where Ω_1 is a positive frequency and $\overline{\Omega}_1$ is a negative frequency, both corresponding to the attenuation A_{dB}.

Frequency transformations

- (cont.)
  - In addition, a complex frequency $s' = \mathrm{j}\Omega'$, which also corresponds to an attenuation of A_{dB}, is transformed into two frequencies with the same attenuation level, that is,

$$\Omega_2 = \frac{1}{2}aB\Omega' + \sqrt{\frac{1}{4}a^2B^2\Omega'^2 + \Omega_0^2} \tag{68}$$

$$\overline{\Omega}_2 = \frac{1}{2}aB\Omega' - \sqrt{\frac{1}{4}a^2B^2\Omega'^2 + \Omega_0^2} \tag{69}$$

and it can be seen that $\overline{\Omega}_1 = -\Omega_2$ and $\overline{\Omega}_2 = -\Omega_1$.

Frequency transformations

- (cont.)
  - The positive frequencies Ω_1 and Ω_2 are the ones we are interested in analyzing. They can be expressed in a single equation as follows:

$$\Omega_{1,2} = \mp\frac{1}{2}aB\Omega' + \sqrt{\frac{1}{4}a^2B^2\Omega'^2 + \Omega_0^2} \tag{70}$$

from which we get

$$\Omega_2 - \Omega_1 = aB\Omega' \tag{71}$$

$$\Omega_1\Omega_2 = \Omega_0^2 \tag{72}$$

  - These relationships indicate that, in this kind of transformation, for each frequency with attenuation A_{dB} there is another frequency geometrically symmetric with respect to the central frequency Ω_0 with the same attenuation.

Frequency transformations

- (cont.)
  - From the above the cutoff frequency of the normalized-lowpass filter Ω'_p is mapped into the frequencies

$$\Omega_{p_{1,2}} = \mp\frac{1}{2}aB\Omega'_p + \sqrt{\frac{1}{4}a^2B^2{\Omega'_p}^2 + \Omega_0^2} \quad (73)$$

such that

$$\Omega_{p_2} - \Omega_{p_1} = aB\Omega'_p \quad (74)$$

$$\Omega_{p_1}\Omega_{p_2} = \Omega_0^2 \quad (75)$$

Frequency transformations

- (cont.)
  - Similarly, the stopband edge frequency Ω'_r of the normalized-lowpass prototype is transformed into the frequencies

$$\Omega_{r_{1,2}} = \mp\frac{1}{2}aB\Omega'_r + \sqrt{\frac{1}{4}a^2B^2{\Omega'_r}^2 + \Omega_0^2} \tag{76}$$

such that

$$\Omega_{r_2} - \Omega_{r_1} = aB\Omega'_r \tag{77}$$

$$\Omega_{r_1}\Omega_{r_2} = \Omega_0^2 \tag{78}$$

Frequency transformations

- The above analysis leads to the conclusion that this normalized-lowpass $\leftrightarrow$ bandpass transformation works for bandpass filters which are geometrically symmetric with respect to the central frequency.

- However, bandpass filter specifications are not usually geometrically symmetric.

- We can generate geometrically symmetric bandpass specifications satisfying the minimum stopband attenuation requirements by the following procedure (see Figure 4):
  - (i) Compute $\Omega_0^2 = \Omega_{p_1}\Omega_{p_2}$.
  - (ii) Compute $\overline{\Omega}_{r_1} = \frac{\Omega_0^2}{\Omega_{r_2}}$, and if $\overline{\Omega}_{r_1} > \Omega_{r_1}$, replace Ω_{r_1} with $\overline{\Omega}_{r_1}$, as illustrated in Figure 4.
  - (iii) If $\overline{\Omega}_{r_1} \leq \Omega_{r_1}$, then compute $\overline{\Omega}_{r_2} = \frac{\Omega_0^2}{\Omega_{r_1}}$, and replace Ω_{r_2} with $\overline{\Omega}_{r_2}$.
  - (iv) If $A_{r_1} \neq A_{r_2}$, choose $A_r = \max\{A_{r_1}, A_{r_2}\}$.

Frequency transformations

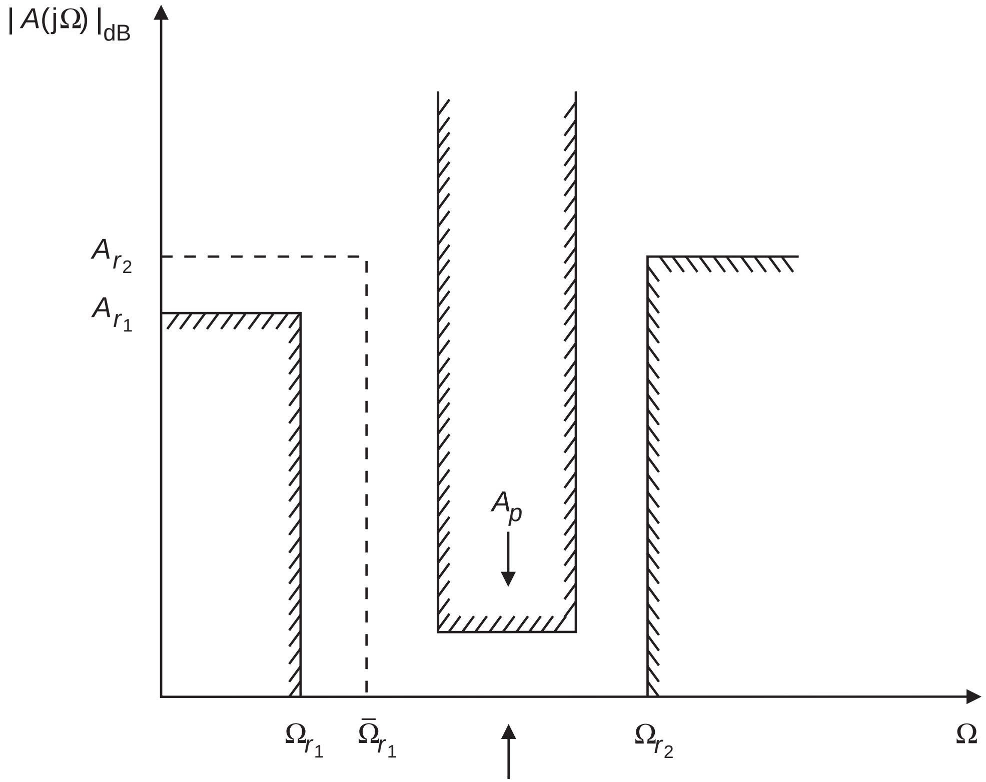


Figure 4: Nonsymmetric bandpass filter specifications.

Frequency transformations

- Once the geometrically symmetric bandpass filter specifications are available, we need to determine the normalized frequencies Ω'_p and Ω'_r, in order to have the corresponding normalized-lowpass filter completely specified.

- According to equations (74) and (77), they can be computed as follows:

$$\Omega'_p = \frac{1}{a} \tag{79}$$

$$\Omega'_r = \frac{1}{a}\frac{\Omega_{r_2} - \Omega_{r_1}}{\Omega_{p_2} - \Omega_{p_1}} \tag{80}$$

- It is worth noting that bandstop filter specifications must also be geometrically symmetric. In this case, however, in order to satisfy the minimum stopband attenuation requirements, the stopband edges must be preserved, while the passband edges should be modified analogously to the procedure described above.

- A summary of all types of transformations are shown in Table 2.

Frequency transformations

Table 2: Analog frequency transformations.

Transformation	Normalization	Denormalization
lowpass(Ω) $\leftrightarrow$ lowpass(Ω')	$\Omega'_p = \frac{1}{a}$; $\Omega'_r = \frac{1}{a}\frac{\Omega_r}{\Omega_p}$	$s' \leftrightarrow \frac{1}{a}\frac{s}{\Omega_p}$
highpass(Ω) $\leftrightarrow$ lowpass(Ω')	$\Omega'_p = \frac{1}{a}$; $\Omega'_r = \frac{1}{a}\frac{\Omega_p}{\Omega_r}$	$s' \leftrightarrow \frac{1}{a}\frac{\Omega_p}{s}$
bandpass(Ω) $\leftrightarrow$ lowpass(Ω')	$\Omega'_p = \frac{1}{a}$	$s' \leftrightarrow \frac{1}{a}\frac{s^2+\Omega_0^2}{Bs}$
	$\Omega'_r = \frac{1}{a}\frac{\Omega_{r_2}-\Omega_{r_1}}{\Omega_{p_2}-\Omega_{p_1}}$	
bandstop(Ω) $\leftrightarrow$ lowpass(Ω')	$\Omega'_p = \frac{1}{a}$	$s' \leftrightarrow \frac{1}{a}\frac{Bs}{s^2+\Omega_0^2}$
	$\Omega'_r = \frac{1}{a}\frac{\Omega_{p_2}-\Omega_{p_1}}{\Omega_{r_2}-\Omega_{r_1}}$	

Frequency transformations

- The general procedure to approximate a standard analog filter using frequency transformations can be summarized as follows:

- - (i) Determine the specifications for the lowpass, highpass, bandpass, or bandstop analog filter.
  - (ii) When designing a bandpass or bandstop filter, make sure the specifications are geometrically symmetric following the proper procedure described earlier in this subsection.
  - (iii) Determine the normalized-lowpass specifications equivalent to the desired filter, following the relationships seen in Table 2.
  - (iv) Perform the filter approximation using the Butterworth, Chebyshev, or elliptic methods.
  - (v) Denormalize the prototype using the frequency transformations given on the right-hand side of Table 2.

Frequency transformations

- Sometimes the approximation of analog filters can present poor numerical conditioning, especially when the desired filter has a narrow transition and/or passband.
- In this case, design techniques employing transformed variables are available, which can improve the numerical conditioning by separating the roots of the polynomials involved.

Example 6.1

- Design a bandpass filter satisfying the specification below using the Butterworth, Chebyshev, and elliptic approximation methods:

$$\left.\begin{aligned} A_{\mathrm{p}} &= 1.0 \text{ dB} \\ A_{\mathrm{r}} &= 40 \text{ dB} \\ \Omega_{\mathrm{r}_1} &= 1394\pi \text{ rad/s} \\ \Omega_{\mathrm{p}_1} &= 1510\pi \text{ rad/s} \\ \Omega_{\mathrm{p}_2} &= 1570\pi \text{ rad/s} \\ \Omega_{\mathrm{r}_2} &= 1704\pi \text{ rad/s} \end{aligned}\right\} \tag{81}$$

Example 6.1 - Solution

- Since $\Omega_{p_1}\Omega_{p_2} \neq \Omega_{r_1}\Omega_{r_2}$, the first step in the design is to determine the geometrically symmetric bandpass filter following the procedure described earlier in this subsection.

- In that manner, we get

$$\Omega_{r_2} = \bar{\Omega}_{r_2} = \frac{\Omega_0^2}{\Omega_{r_1}} = 1700.6456\pi \text{ rad/s} \tag{82}$$

Example 6.1 - Solution

- Finding the corresponding lowpass specifications based on the transformations in Table 2, we have

$$\Omega'_p = \frac{1}{a} \tag{83}$$

$$\Omega'_r = \frac{1}{a}\frac{\bar{\Omega}_{r_2} - \Omega_{r_1}}{\Omega_{p_2} - \Omega_{p_1}} = \frac{1}{a}5.1108 \tag{84}$$

where

$$a = \begin{cases} 1, & \text{for the Butterworth and Chebyshev filters} \\ 2.2607, & \text{for the elliptic filter} \end{cases} \tag{85}$$

Example 6.1 - Solution

- (a) Butterworth approximation: from the specifications above, we can compute ϵ from equation (6), and, having ϵ, the minimum filter order required to satisfy the specifications from equation (8):

$$\epsilon = 0.5088 \tag{86}$$

$$n = 4 \tag{87}$$

- From equation (12), the zeros of the normalized Butterworth polynomial when $n = 4$ are given by

$$\left.\begin{aligned} s'_{1,2} &= -1.0939 \pm j0.4531 \\ s'_{3,4} &= -0.4531 \pm j1.0939 \\ s'_{5,6} &= 1.0939 \pm j0.4531 \\ s'_{7,8} &= 0.4531 \pm j1.0939 \end{aligned}\right\} \tag{88}$$

Example 6.1 - Solution

- Selecting the ones with negative real part to be the poles of $H'(s')$, this normalized transfer function becomes

$$H'(s') = 1.9652 \frac{1}{s'^4 + 3.0940 s'^3 + 4.7863 s'^2 + 4.3373 s' + 1.9652} \tag{89}$$

- The design is completed by applying the lowpass to bandpass transformation in Table 2.

- The resulting bandpass transfer function is then given by

$$H(s) = H_0 \frac{s^4}{a_8 s^8 + a_7 s^7 + a_6 s^6 + a_5 s^5 + a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{90}$$

where the filter coefficients and poles are listed in Table 3.

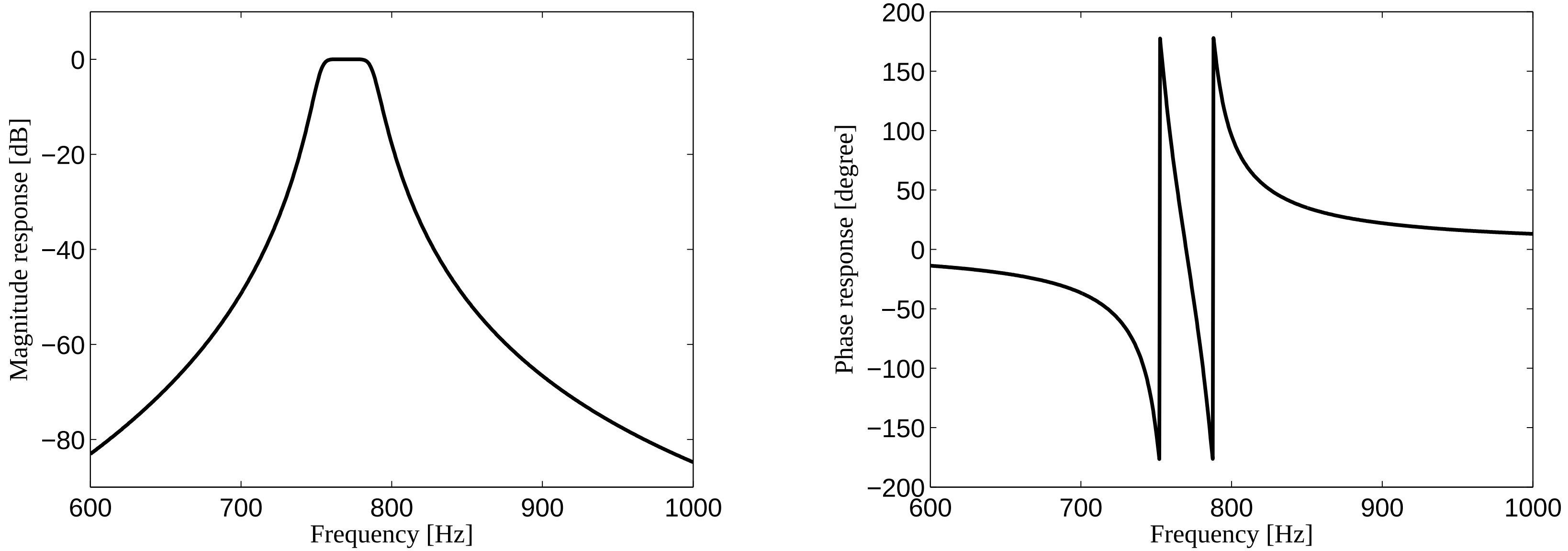
Example 6.1 - Solution

Table 3: Characteristics of the Butterworth bandpass filter. Gain constant: $H_0 = 2.4809 \times 10^9$.

Denominator coefficients	Filter poles
$a_0 = 2.9971 \times 10^{29}$	$p_1 = -41.7936 + j4734.9493$
$a_1 = 7.4704 \times 10^{24}$	$p_2 = -41.7936 - j4734.9493$
$a_2 = 5.1331 \times 10^{22}$	$p_3 = -102.1852 + j4793.5209$
$a_3 = 9.5851 \times 10^{17}$	$p_4 = -102.1852 - j4793.5209$
$a_4 = 3.2927 \times 10^{15}$	$p_5 = -104.0058 + j4878.9280$
$a_5 = 4.0966 \times 10^{10}$	$p_6 = -104.0058 - j4878.9280$
$a_6 = 9.3762 \times 10^{7}$	$p_7 = -43.6135 + j4941.1402$
$a_7 = 5.8320 \times 10^{2}$	$p_8 = -43.6135 - j4941.1402$
$a_8 = 1.0$	

Example 6.1 - Solution

- Figure 5 depicts the frequency response of the designed Butterworth bandpass filter.



(a) (b)

Figure 5: Bandpass Butterworth filter: (a) magnitude response; (b) phase response.

Example 6.1 - Solution

- (b) Chebyshev approximation: from the normalized specifications in equations (81) and (82), one can compute ϵ and n based on equations (21) and (23), respectively, resulting in

$$\epsilon = 0.5088 \tag{91}$$

$$n = 3 \tag{92}$$

- Then, from equations (24)–(35), we have that the poles of the normalized transfer function are:

$$\left.\begin{aligned} s'_{1,2} &= -0.2471 \mp j0.9660 \\ s'_3 &= -0.4942 - j0.0 \end{aligned}\right\} \tag{93}$$

Example 6.1 - Solution

- This implies that the normalized-lowpass filter has the following transfer function:

$$H'(s') = 0.4913 \frac{1}{s'^3 + 0.9883 s'^2 + 1.2384 s' + 0.4913} \tag{94}$$

- The denormalized design is obtained by applying the lowpass to bandpass transformation.

- The resulting transfer function is of the form

$$H(s) = H_0 \frac{s^3}{a_6 s^6 + a_5 s^5 + a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{95}$$

where all filter coefficients and poles are listed in Table 4.

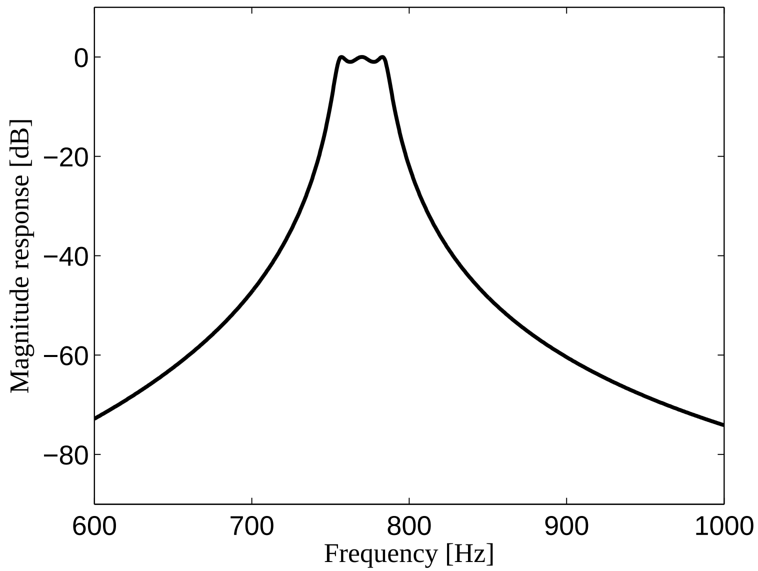
Example 6.1 - Solution

Table 4: Characteristics of the Chebyshev bandpass filter. Gain constant: $H_0 = 3.2905 \times 10^6$.

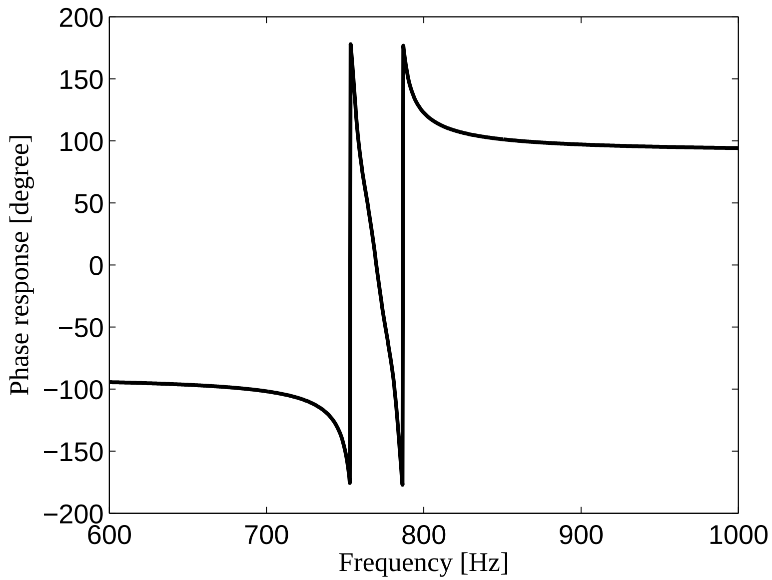
Denominator coefficients	Filter poles
$a_0 = 1.2809 \times 10^{22}$	$p_1 = -22.8490 + j4746.8921$
$a_1 = 1.0199 \times 10^{17}$	$p_2 = -22.8490 - j4746.8921$
$a_2 = 1.6434 \times 10^{15}$	$p_3 = -46.5745 + j4836.9104$
$a_3 = 8.7212 \times 10^{9}$	$p_4 = -46.5745 - j4836.9104$
$a_4 = 7.0238 \times 10^{7}$	$p_5 = -23.7255 + j4928.9785$
$a_5 = 1.8630 \times 10^{2}$	$p_6 = -23.7255 - j4928.9785$
$a_6 = 1.0$	

Example 6.1 - Solution

- Figure 6 depicts the frequency response of the resulting Chebyshev bandpass filter.



(a)



(b)

Figure 6: Bandpass Chebyshev filter: (a) magnitude response; (b) phase response.

Example 6.1 - Solution

- Elliptic approximation: from equation (6.85), for this elliptic approximation, we have that $a = 2.2607$, and then the normalized specifications are

$$\Omega'_p = 0.4423 \tag{96}$$

$$\Omega'_r = 2.2607 \tag{97}$$

- From equation (48), the minimum order required for the elliptic approximation to satisfy the specifications is $n = 3$. Therefore, from equations (55)–(60), the normalized-lowpass filter has the transfer function

$$H'(s') = 6.3627 \times 10^{-3} \frac{s'^2 + 6.7814}{s'^3 + 0.4362s'^2 + 0.2426s' + 0.0431} \tag{98}$$

Example 6.1 - Solution

- The denormalized-bandpass design is then obtained by applying the lowpass to bandpass transformation given in Table 2, with $a = 2.2607$. The resulting bandpass transfer function is given by

$$H(s) = H_0 \frac{b_5 s^5 + b_3 s^3 + b_1 s}{a_6 s^6 + a_5 s^5 + a_4 s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \tag{99}$$

where all filter coefficients, zeros, and poles are listed in Table 5.

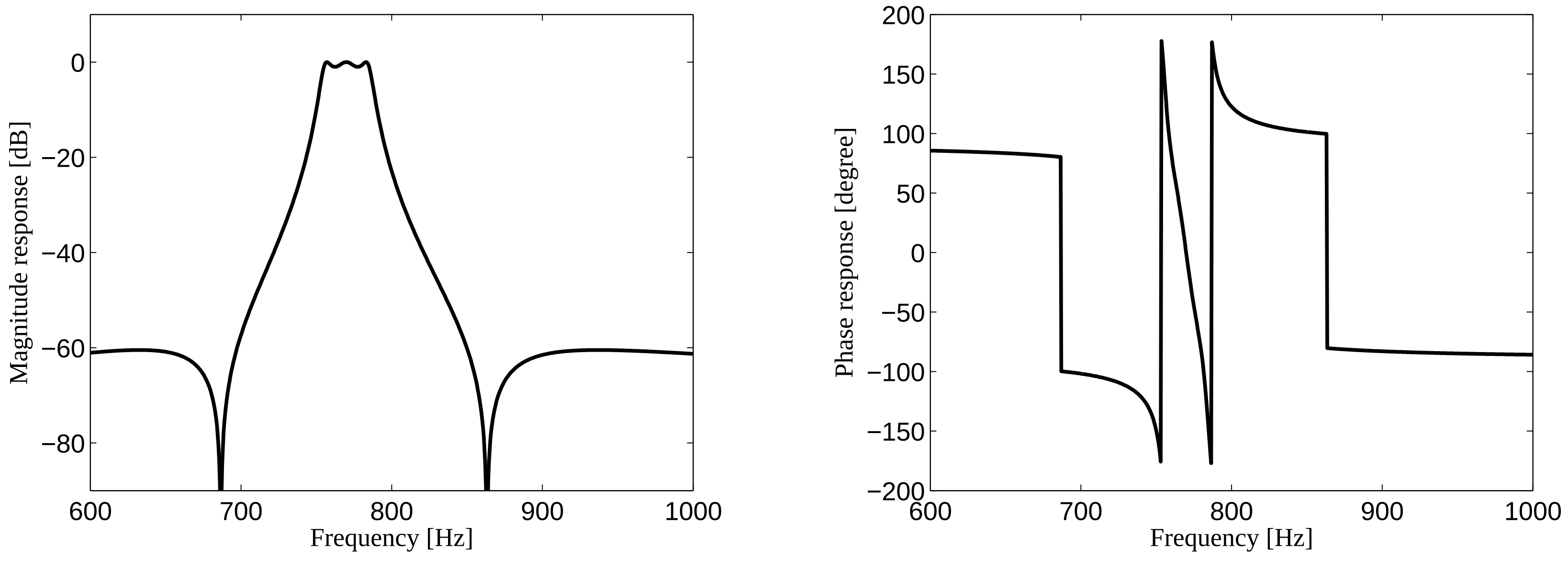
Example 6.1 - Solution

Table 5: Characteristics of the elliptic bandpass filter. Gain constant: $H_0 = 2.7113\,10^6$.

Numerator coefficients	Denominator coefficients
$b_0 = 0.0$	$a_0 = 1.2809 \times 10^{22}$
$b_1 = 5.4746 \times 10^{14}$	$a_1 = 1.0175 \times 10^{17}$
$b_2 = 0.0$	$a_2 = 1.6434 \times 10^{15}$
$b_3 = 4.8027 \times 10^{7}$	$a_3 = 8.7008 \times 10^{9}$
$b_4 = 0.0$	$a_4 = 7.0238 \times 10^{7}$
$b_5 = 1.0$	$a_5 = 1.8586 \times 10^{2}$
$b_6 = 0.0$	$a_6 = 1.0$
Filter zeros	**Filter poles**
$z_1 = +j4314.0061$	$p_1 = -22.4617 + j4746.6791$
$z_2 = -j4314.0061$	$p_2 = -22.4617 - j4746.6791$
$z_3 = +j5423.6991$	$p_3 = -47.1428 + j4836.9049$
$z_4 = -j5423.6991$	$p_4 = -47.1428 - j4836.9049$
$z_5 = 0.0$	$p_5 = -23.3254 + j4929.2035$
	$p_6 = -23.3254 - j4929.2035$

Example 6.1 - Solution

- Figure 7 depicts the frequency response of the resulting elliptic bandpass filter.



(a) (b)

Figure 7: Bandpass elliptic filter: (a) magnitude response; (b) phase response.

Continuous-time to discrete-time transformations

- As mentioned at the beginning of this chapter, a classical procedure for designing IIR digital filters is to design an analog prototype first and then transform it into a digital filter.
- In this section, we study two methods of carrying out this transformation, namely, the impulse-invariance method and the bilinear transformation method.

Impulse-invariance method

- The intuitive way to implement a digital filtering operation, having an analog prototype as starting point, is the straightforward digitalization of the convolution operation, as follows.

- The output $y_a(t)$ of an analog filter having impulse response $h_a(t)$ when excited by a signal $x_a(t)$ is

$$y_a(t) = \int_{-\infty}^{\infty} x_a(\tau) h_a(t-\tau) d\tau \tag{100}$$

- One possible way to implement this operation in the discrete-time domain is to divide the time axis into slices of size T, replacing the integral by a summation of the areas of rectangles of width T and height $x_a(mT)h_a(t-mT)$, for all integers m.

Impulse-invariance method

- Equation (100) then becomes

$$y_a(t) = \sum_{m=-\infty}^{\infty} x_a(mT)h_a(t - mT)T \tag{101}$$

- The sampled version of $y_a(t)$ is obtained by substituting t by nT, yielding

$$y_a(nT) = \sum_{m=-\infty}^{\infty} x_a(mT)h_a(nT - mT)T \tag{102}$$

- This is clearly equivalent to obtaining the samples $y_a(nT)$ of $y_a(t)$ by filtering the samples $x_a(nT)$ with a digital filter having impulse response $h(n) = h_a(nT)$.

- That is, the impulse response of the equivalent digital filter would be a sampled version of the impulse response of the analog filter, using the same sampling rate for the input and output signals.

Impulse-invariance method

- Roughly speaking, if the Nyquist criterion is met by the filter impulse response during the sampling operation, the discrete-time prototype has the same frequency response as the continuous-time one.
- In addition, a sampled version of a stable analog impulse response is clearly also stable. These are the main properties of this method of generating IIR filters, called the impulse-invariance method.
- In what follows, we analyze the above properties more precisely, in order to get a better understanding of the main strengths and limitations of this method.

CAMBRIDGE UNIVERSITY PRESS

Impulse-invariance method

- We can begin by investigating the properties of the digital filter with impulse response $h(n) = h_a(nT)$ in the frequency domain.

- From equation (102), the discrete-time Fourier transform of $h(n)$ is

$$H(e^{j\Omega T}) = \frac{1}{T} \sum_{l=-\infty}^{\infty} H_a(j\Omega + j\Omega_s l) \tag{103}$$

  where $H_a(s)$, $s = \sigma + j\Omega$, is the analog transfer function, and $\Omega_s = \frac{2\pi}{T}$ is the sampling frequency.

- That is, the digital frequency response is equal to the analog one replicated at intervals $l\Omega_s$.

- One important consequence of this fact is that if $H_a(j\Omega)$ has much energy for $\Omega > \Omega_s/2$, there will be aliasing, and therefore the digital frequency response will be a severely distorted version of the analog one.

Impulse-invariance method

- Another way of seeing this is that the digital frequency response is obtained by folding the analog frequency response, for $-\infty < \Omega < \infty$, on the unit circle of the $z = e^{sT}$ plane, with each interval $[\sigma + j(l - \frac{1}{2})\Omega_s, \sigma + j(l + \frac{1}{2})\Omega_s]$, for all integers l, corresponding to one full turn over the unit circle of the z plane.

- This limits the usefulness of the impulse-invariance method to the design of transfer functions whose magnitude responses decrease monotonically at high frequencies.

- For example, its use in the direct design of highpass, bandstop, or even elliptic lowpass and bandpass filters is strictly forbidden, and other methods should be considered for designing such filters.

Impulse-invariance method

- Stability of the digital filter can also be inferred from the stability of the analog prototype by analyzing equation (103).
- In fact, based on that equation, we can interpret the impulse-invariance method as a mapping from the s domain to the z domain such that each slice of the s plane given by the interval $[\sigma + \mathrm{j}(l - \frac{1}{2})\Omega_s, \sigma + \mathrm{j}(l + \frac{1}{2})\Omega_s]$, for all integers l, where $\sigma = \mathrm{Re}\{s\}$, is mapped into the same region of the z plane.
- Also, the left side of the s plane, that is, where $\sigma < 0$, is mapped into the interior of the unit circle, implying that if the analog transfer function is stable (all poles on the left side of the s plane), then the digital transfer function is also stable (all poles inside the unit circle of the z plane).

Impulse-invariance method

- In practice, the impulse-invariance transformation is not implemented through equation (103), as a simpler procedure can be deduced by expanding an Nth-order $H_a(s)$ as follows:

$$H_a(s) = \sum_{l=1}^{N} \frac{r_l}{s - p_l} \tag{104}$$

where it is assumed that $H_a(s)$ does not have multiple poles.

- The corresponding impulse response is given by

$$h_a(t) = \sum_{l=1}^{N} r_l e^{p_l t} u(t) \tag{105}$$

where $u(t)$ is the unit step function.

Impulse-invariance method

- If we now sample that impulse response, the resulting sequence is

$$h_d(n) = h_a(nT) = \sum_{l=1}^{N} r_l e^{p_l nT} u(nT) \tag{106}$$

and the corresponding discrete-time transfer function is given by

$$H_d(z) = \sum_{l=1}^{N} \frac{r_l z}{z - e^{p_l T}} \tag{107}$$

- This equation shows that a pole $s = p_l$ of the continuous-time filter corresponds to a pole of the discrete-time filter at $z = e^{p_l T}$.

- In that way, if p_l has negative real part, then $e^{p_l T}$ is inside the unit circle, generating a stable digital filter when we use the impulse-invariance method.

Impulse-invariance method

- In order to obtain the same passband gain for the continuous- and discrete-time filters, for any value of the sampling period T, we should use the following expression for $H_d(z)$:

$$H_d(z) = \sum_{l=1}^{N} T \frac{r_l z}{z - e^{p_l T}} \tag{108}$$

which corresponds to

$$h_d(n) = T h_a(nT) \tag{109}$$

- Thus, the overall impulse-invariance method consists of writing the analog transfer function $H_a(s)$ in the form of equation (104), determining the poles p_l and corresponding residues r_l, and generating $H_d(z)$ according to equation (108).

Example 6.2

- Transform the continuous-time lowpass transfer function given by

$$H(s) = \frac{1}{s^2 + s + 1} \tag{110}$$

into a discrete-time transfer function using the impulse-invariance transformation method with $\Omega_s = 10$ rad/s. Plot the corresponding analog and digital magnitude responses.

Example 6.2 - Solution

- A second-order lowpass transfer function can be written as

$$
\begin{aligned}
H(s) &= \frac{\Omega_0^2}{s^2 + \frac{\Omega_0}{Q} s + \Omega_0^2} \\
&= \frac{\Omega_0^2}{\sqrt{\frac{\Omega_0^2}{Q^2} - 4\Omega_0^2}} \left(\frac{1}{s + \frac{\Omega_0}{2Q} - \sqrt{\frac{\Omega_0^2}{4Q^2} - \Omega_0^2}} - \frac{1}{s + \frac{\Omega_0}{2Q} + \sqrt{\frac{\Omega_0^2}{4Q^2} - \Omega_0^2}} \right)
\end{aligned}
\tag{111}
$$

- Its poles are located at

$$
p_1 = p_2^* = -\frac{\Omega_0}{2Q} + j\sqrt{\Omega_0^2 - \frac{\Omega_0^2}{4Q^2}}
\tag{112}
$$

and the corresponding residues are given by

$$
r_1 = r_2^* = \frac{-j\Omega_0^2}{\sqrt{4\Omega_0^2 - \frac{\Omega_0^2}{Q^2}}}
\tag{113}
$$

- Applying the impulse-invariance method with $T = 2\pi/10$, the resulting discrete-time transfer function is given by

$$H(z) = \frac{2jTr_1 \sin(\mathrm{Im}\{p_1\}T)e^{\mathrm{Re}\{p_1\}T}z}{z^2 - 2\cos(\mathrm{Im}\{p_1\}T)e^{\mathrm{Re}\{p_1\}T}z + e^{2\mathrm{Re}\{p_1\}T}}$$
$$= \frac{0.274\,331\,03\,z}{z^2 - 1.249\,825\,52\,z + 0.533\,488\,09} \tag{114}$$

- The magnitude responses corresponding to the analog and digital transfer functions are depicted in Figure 8.

- As can be seen, the frequency responses are similar except for the limited stopband attenuation of the discrete-time filter which is due to the aliasing effect.

Example 6.2 - Solution

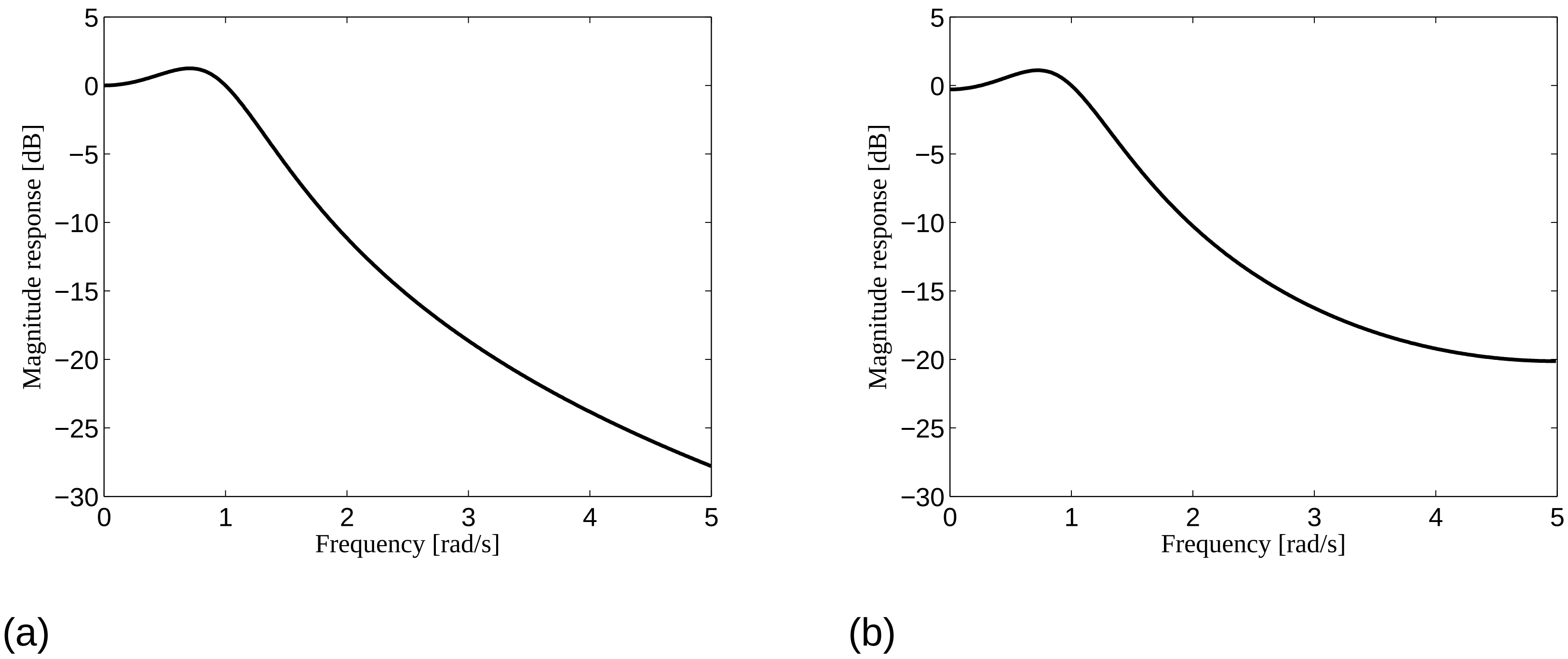


Figure 8: Magnitude responses obtained with the impulse-invariance method: (a) continuous-time filter; (b) discrete-time filter.

Impulse-invariance method

- We should again emphasize that the impulse-invariance method is suitable only for continuous-time prototypes with frequency responses that decrease monotonically at high frequencies, which limits its applicability a great deal.
- In the next section, we analyze the bilinear transformation method, which overcomes some of the limitations of the impulse-invariance method.

Bilinear transformation method

- The bilinear transformation method, like the impulse-invariance method, basically consists of mapping the left-hand side of the s plane into the interior of the unit circle of the z plane.

- The main difference between them is that in the bilinear transformation method the whole analog frequency range $-\infty < \Omega < \infty$ is squeezed into the unit circle $-\pi \leq \omega \leq \pi$, while in the impulse-invariance method the analog frequency response is folded around the unit circle indefinitely.

- The main advantage of the bilinear transformation method is that aliasing is avoided, thereby keeping the magnitude response characteristics of the continuous-time transfer function when generating the discrete-time transfer function.

- The bilinear mapping is derived by first considering the key points of the s plane and analyzing their corresponding points in the z plane after the transformation.

- Hence, the left-hand side of the s plane should be uniquely mapped into the interior of the unit circle of the z plane, and so on, as given in Table 6.

Bilinear transformation method

Table 6: Correspondence of key points of the s and z planes using the bilinear transformation method.

s plane	$\rightarrow$	z plane
$\sigma \pm j\Omega$	$\rightarrow$	$re^{\pm j\omega}$
$j0$	$\rightarrow$	1
$j\infty$	$\rightarrow$	-1
$\sigma > 0$	$\rightarrow$	$r > 1$
$\sigma = 0$	$\rightarrow$	$r = 1$
$\sigma < 0$	$\rightarrow$	$r < 1$
$j\Omega$	$\rightarrow$	$e^{j\omega}$
$-\infty < \Omega < \infty$	$\rightarrow$	$-\pi < \omega < \pi$

Bilinear transformation method

- In order to satisfy the second and third requirements of Table 6, the bilinear transformation must have the following form

$$s \rightarrow k \frac{f_1(z) - 1}{f_2(z) + 1} \tag{115}$$

where $f_1(1) = 1$ and $f_2(-1) = -1$.

- Sufficient conditions for the last three mapping requirements to be satisfied can be determined as follows:

$$s = \sigma + j\Omega = k \frac{(\text{Re}\{f_1(z)\} - 1) + j\text{Im}\{f_1(z)\}}{(\text{Re}\{f_2(z)\} + 1) + j\text{Im}\{f_2(z)\}} \tag{116}$$

Bilinear transformation method

- Equating the real parts of both sides of the above equation, we have that

$$\sigma = k\frac{(\mathrm{Re}\{f_1(z)\} - 1)(\mathrm{Re}\{f_2(z)\} + 1) + \mathrm{Im}\{f_1(z)\}\mathrm{Im}\{f_2(z)\}}{(\mathrm{Re}\{f_2(z)\} + 1)^2 + (\mathrm{Im}\{f_2(z)\})^2} \tag{117}$$

and since $\sigma = 0$ implies that $r = 1$, the following relation is valid:

$$\frac{\mathrm{Re}\{f_1(e^{j\omega})\} - 1}{\mathrm{Im}\{f_1(e^{j\omega})\}} = -\frac{\mathrm{Im}\{f_2(e^{j\omega})\}}{\mathrm{Re}\{f_2(e^{j\omega})\} + 1} \tag{118}$$

- The condition $\sigma < 0$ is equivalent to

$$\frac{\mathrm{Re}\{f_1(re^{j\omega})\} - 1}{\mathrm{Im}\{f_1(re^{j\omega})\}} < -\frac{\mathrm{Im}\{f_2(re^{j\omega})\}}{\mathrm{Re}\{f_2(re^{j\omega})\} + 1}, \quad r < 1 \tag{119}$$

- The last two lines of Table 6 show the correspondence between the analog frequency and the unit circle of the z plane.

Bilinear transformation method

- If we want the orders of the discrete-time and continuous-time systems to remain the same after the transformation, then $f_1(z)$ and $f_2(z)$ must be first-order polynomials.
- In addition, if we wish to satisfy the conditions imposed by equation (115), we must choose $f_1(z) = f_2(z) = z$.
- It is straightforward to verify that equation (118), as well as the inequality (119), are automatically satisfied with this choice for $f_1(z)$ and $f_2(z)$.

Bilinear transformation method

- The bilinear transformation is then given by

$$s \to k\frac{z-1}{z+1} \tag{120}$$

which, for $s = \mathrm{j}\Omega$ and $z = \mathrm{e}^{\mathrm{j}\omega}$, is equivalent to

$$\mathrm{j}\Omega \to k\frac{\mathrm{e}^{\mathrm{j}\omega}-1}{\mathrm{e}^{\mathrm{j}\omega}+1} = k\frac{\mathrm{e}^{\frac{\mathrm{j}\omega}{2}}-\mathrm{e}^{\frac{-\mathrm{j}\omega}{2}}}{\mathrm{e}^{\frac{\mathrm{j}\omega}{2}}+\mathrm{e}^{\frac{-\mathrm{j}\omega}{2}}} = \mathrm{j}k\frac{\sin\frac{\omega}{2}}{\cos\frac{\omega}{2}} = \mathrm{j}k\tan\frac{\omega}{2} \tag{121}$$

that is

$$\Omega \to k\tan\frac{\omega}{2} \tag{122}$$

Bilinear transformation method

- For small frequencies, $\tan\frac{\omega}{2} \approx \frac{\omega}{2}$. Hence, to keep the magnitude response of the digital filter approximately the same as the prototype analog filter at low frequencies, we should have for small frequencies $\Omega = \omega\frac{\Omega_s}{2\pi}$, and therefore we should choose $k = \frac{\Omega_s}{\pi} = \frac{2}{T}$.

- In conclusion, the bilinear transformation of a continuous-time transfer function into a discrete-time transfer function is implemented through the following mapping:

$$H(z) = H_a(s)\big|_{s=\frac{2}{T}\frac{z-1}{z+1}} \tag{123}$$

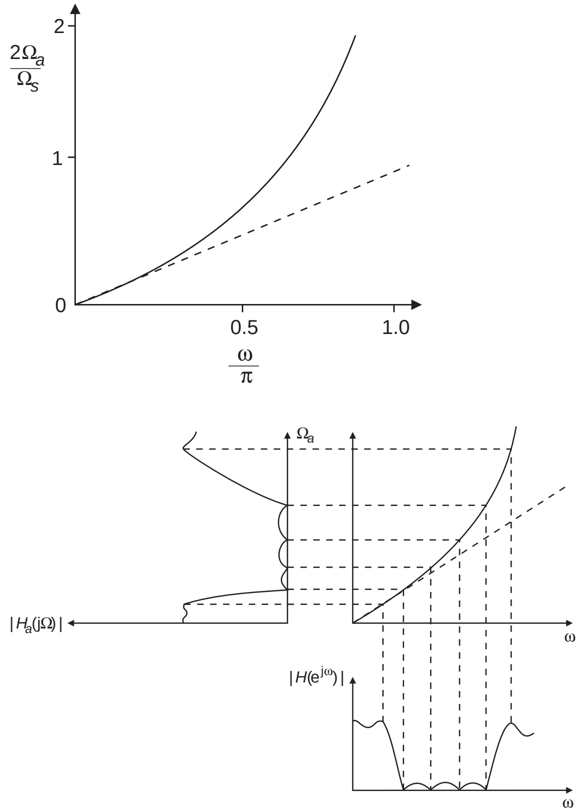
and therefore, the bilinear transformation maps analog frequencies into digital frequencies as follows:

$$\Omega \to \frac{2}{T}\tan\frac{\omega}{2} \tag{124}$$

Bilinear transformation method

- For high frequencies, this relationship is highly nonlinear, as seen in Figure 9a, corresponding to a large distortion in the frequency response of the digital filter when compared to the analog prototype.
- The distortion in the magnitude response, also known as the warping effect, can be visualized as in Figure 9b.

Bilinear transformation method



(a)

(b)

Figure 9: Bilinear transformation method: (a) relation between the analog and digital frequencies; (b) warping effect in the magnitude response of a bandstop filter.

Bilinear transformation method

- The warping effect caused by the bilinear transformation can be compensated for by prewarping the frequencies given at the specifications before the analog filter is actually designed.
- For example, suppose we wish to design a lowpass digital filter with cutoff frequency ω_p and stopband edge ω_r.
- The prewarped specifications Ω_{a_p} and Ω_{a_r} of the lowpass analog prototype are then given by

$$\Omega_{a_p} = \frac{2}{T} \tan \frac{\omega_p}{2} \tag{125}$$

$$\Omega_{a_r} = \frac{2}{T} \tan \frac{\omega_r}{2} \tag{126}$$

Bilinear transformation method

- Following the same line of thought, we may apply prewarping to as many frequencies of interest as specified for the digital filter.
- If these frequencies are given by ω_i, for $i = 1, 2, \ldots, n$, then the frequencies to be included in the analog filter specifications are

$$\Omega_{a_i} = \frac{2}{T} \tan \frac{\omega_i}{2} \tag{127}$$

for $i = 1, 2, \ldots, n$.

Bilinear transformation method

- Hence, the design procedure using the bilinear transformation method can be summarized as follows:
  - (i) Prewarp all the prescribed frequency specifications ω_i, obtaining Ω_{a_i}, for $i = 1, 2, \ldots, n$.
  - (ii) Generate $H_a(s)$, following the procedure given in Subsection 6.2.5, satisfying the specifications for the frequencies Ω_{a_i}.
  - (iii) Obtain $H_d(z)$, by replacing s with $\frac{2}{T}\frac{z-1}{z+1}$ in $H_a(s)$.

Bilinear transformation method

- With the bilinear transformation, we can design Butterworth, Chebyshev, and elliptic digital filters starting with a corresponding analog prototype.
- The bilinear transformation method always generates stable digital filters as long as the prototype analog filter is stable.
- Using the prewarping procedure, the method keeps the magnitude characteristics of the prototype but introduces distortions to the phase response.

Example 6.3

- Design a digital elliptic bandpass filter satisfying the following specifications:

$$\left.\begin{aligned} A_p &= 0.5 \text{ dB} \\ A_r &= 65 \text{ dB} \\ \Omega_{r_1} &= 850 \text{ rad/s} \\ \Omega_{p_1} &= 980 \text{ rad/s} \\ \Omega_{p_2} &= 1020 \text{ rad/s} \\ \Omega_{r_2} &= 1150 \text{ rad/s} \\ \Omega_s &= 10\,000 \text{ rad/s} \end{aligned}\right\} \tag{128}$$

Example 6.3 - Solution

- First, we have to normalize the frequencies above to the range of digital frequencies using the expression $\omega = \Omega \frac{2\pi}{\Omega_s}$.
- Since $\Omega_s = 10\,000$ rad/s, we have that

$$\left.\begin{aligned} \omega_{r_1} &= 0.5341 \text{ rad/sample} \\ \omega_{p_1} &= 0.6158 \text{ rad/sample} \\ \omega_{p_2} &= 0.6409 \text{ rad/sample} \\ \omega_{r_2} &= 0.7226 \text{ rad/sample} \end{aligned}\right\} \tag{129}$$

Example 6.3 - Solution

- Then, by applying equation (127), the prewarped frequencies become

$$\left.\begin{aligned}\Omega_{a_{r_1}} &= 870.7973 \text{ rad/s}\\ \Omega_{a_{p_1}} &= 1012.1848 \text{ rad/s}\\ \Omega_{a_{p_2}} &= 1056.4085 \text{ rad/s}\\ \Omega_{a_{r_2}} &= 1202.7928 \text{ rad/s}\end{aligned}\right\} \tag{130}$$

- By making $\Omega_{a_{r_1}} = 888.9982$ rad/s to obtain a geometrically symmetric filter, from Table 2, we have that

$$\Omega_0 = 1034.0603 \text{ rad/s} \tag{131}$$

$$B = 44.2237 \text{ rad/s} \tag{132}$$

$$a = 2.6638 \tag{133}$$

$$\Omega'_p = 0.3754 \tag{134}$$

$$\Omega'_r = 2.6638 \tag{135}$$

- From the filter specifications, the required order for the analog elliptic normalized-lowpass filter is $n = 3$, and the resulting normalized transfer function is

$$H'(s') = 4.0426 \times 10^{-3} \frac{s'^2 + 9.4372}{s'^3 + 0.4696s'^2 + 0.2162s' + 0.0382} \tag{136}$$

- The denormalized design is then obtained by applying the lowpass-to-bandpass transformation given in Table 2, with $a = 2.6638$.

- After applying the bilinear transformation the resulting digital bandpass transfer function becomes

$$H(z) = H_0 \frac{b_6 z^6 + b_5 z^5 + b_4 z^4 + b_3 z^3 + b_2 z^2 + b_1 z + b_0}{a_6 z^6 + a_5 z^5 + a_4 z^4 + a_3 z^3 + a_2 z^2 + a_1 z + a_0} \tag{137}$$

  where all filter coefficients, zeros, and poles are listed in Table 7.

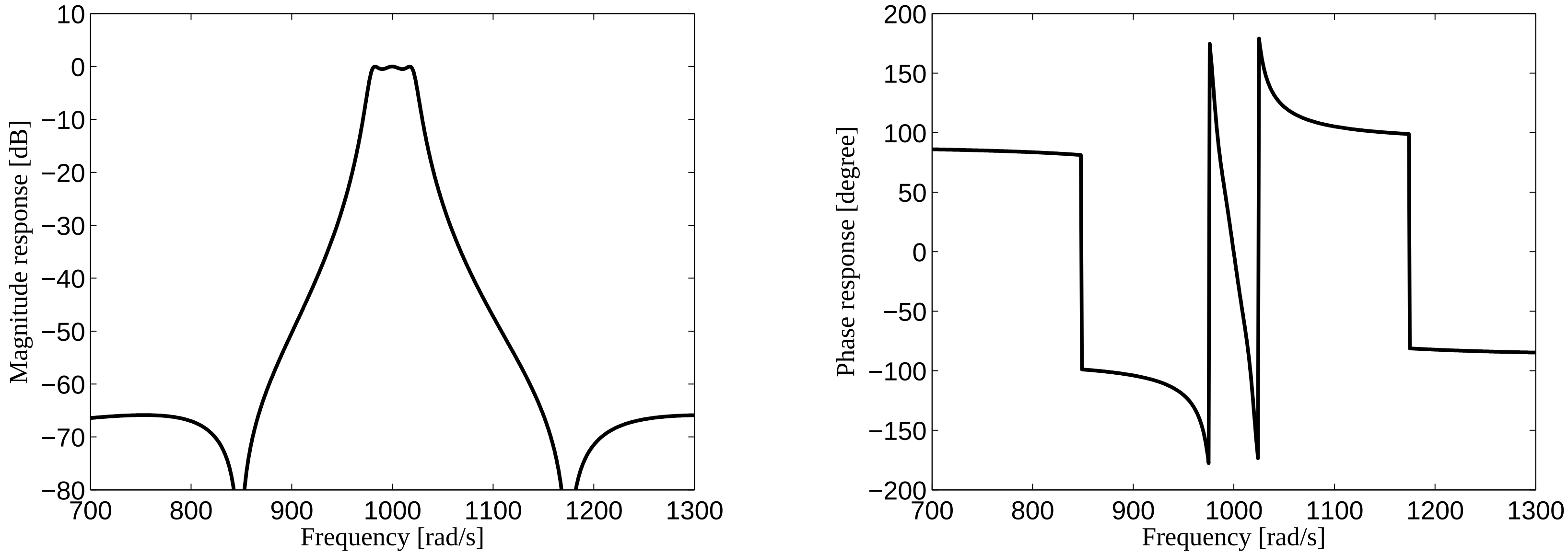
- Figure 10 depicts the frequency response of the resulting digital elliptic bandpass filter.

Example 6.3 - Solution

Table 7: Characteristics of the digital elliptic bandpass filter. Gain constant: $H_0 = 1.3461 \times 10^{-4}$.

Numerator coefficients	Denominator coefficients
$b_0 = -1.0$	$a_0 = 0.9691$
$b_1 = 3.2025$	$a_1 = -4.7285$
$b_2 = -3.5492$	$a_2 = 10.6285$
$b_3 = 0.0$	$a_3 = -13.7261$
$b_4 = 3.5492$	$a_4 = 10.7405$
$b_5 = -3.2025$	$a_5 = -4.8287$
$b_6 = 1.0$	$a_6 = 1.0$
Filter zeros	**Filter poles**
$z_1 = 0.7399 + j0.6727$	$p_1 = 0.7982 + j0.5958$
$z_2 = 0.7399 - j0.6727$	$p_2 = 0.7982 - j0.5958$
$z_3 = 0.8613 + j0.5081$	$p_3 = 0.8134 + j0.5751$
$z_4 = 0.8613 - j0.5081$	$p_4 = 0.8134 - j0.5751$
$z_5 = 1.0 + j0.0$	$p_5 = 0.8027 + j0.5830$
$z_6 = -1.0 + j0.0$	$p_6 = 0.8027 - j0.5830$

Example 6.3 - Solution



(a) (b)

Figure 10: Digital elliptic bandpass filter: (a) magnitude response; (b) phase response.

Bilinear transformation method

- We can then observe that, as we perform the mapping $s \longrightarrow z$, either opting for the impulse-invariance method or the bilinear transformation method, we are essentially folding the continuous-time frequency axis around the z-domain unit circle.
- Hence, any digital frequency response is periodic, and the interval $-\frac{\Omega_s}{2} \leq \Omega \leq \frac{\Omega_s}{2}$, or equivalently, $-\pi \leq \omega \leq \pi$, is the so-called fundamental period.
- We should bear in mind that in the expressions developed in this subsection the unfolded analog frequencies Ω are related to the digital frequencies ω, which are restricted to the interval $-\pi \leq \omega \leq \pi$.
- Therefore, all specifications should be normalized to the interval $[-\pi, \pi]$ using

$$\omega = \Omega \frac{2\pi}{\Omega_s} \tag{138}$$

- In the discussion that follows, we assume that $\Omega_s = 2\pi$ rad/s.

Frequency transformation in the discrete-time domain

- Usually, in the approximation of a continuous-time filter, we begin by designing a normalized lowpass filter and then, through a frequency transformation, the filter with the specified magnitude response is obtained.
- In the design of digital filters, we can also start by designing a digital lowpass filter and then apply a frequency transformation in the discrete-time domain.
- The procedure consists of replacing the variable z by an appropriate function $\mathrm{g}(z)$ to generate the desired magnitude response.

Frequency transformation in the discrete-time domain

- The function $\mathrm{g}(z)$ needs to meet some constraints to be a valid transformation, namely:
  - The function $\mathrm{g}(z)$ must be a ratio of polynomials, since the digital filter transfer function must remain a ratio of polynomials after the transformation.
  - The mapping $z \longrightarrow \mathrm{g}(z)$ must be such that the filter stability is maintained, that is, stable filters generate stable transformed filters and unstable filters generate unstable transformed filters. This is equivalent to saying that the transformation maps the interior of the unit circle onto the interior of the unit circle and the exterior of the unit circle onto the exterior of the unit circle.

Frequency transformation in the discrete-time domain

- It can be shown that a function $g(z)$ satisfying the above conditions is of the form

$$g(z) = \pm \left[\prod_{i=1}^{n} \frac{(z - \alpha_i)}{(1 - z\alpha_i^*)} \frac{(z - \alpha_i^*)}{(1 - z\alpha_i)} \right] \left(\prod_{i=n+1}^{m} \frac{z - \alpha_i}{1 - z\alpha_i} \right) \tag{139}$$

  where α_i^* is complex conjugate of α_i, and α_i is real for $n + 1 \leq i \leq m$.

- In the following subsections, we analyze special cases of $g(z)$ that generate lowpass-to-lowpass, lowpass-to-highpass, lowpass-to-bandpass, and lowpass-to-bandstop transformations.

Lowpass-to-lowpass transformation

- One necessary condition for a lowpass-to-lowpass transformation is that a magnitude response must keep its original values at $\omega = 0$ and $\omega = \pi$ after the transformation.

- Therefore, we must have

$$g(1) = 1 \tag{140}$$

$$g(-1) = -1 \tag{141}$$

Lowpass-to-lowpass transformation

- Another necessary condition is that the frequency response should only be warped between $\omega = 0$ and $\omega = \pi$, that is, a full turn around the unit circle in z must correspond to a full turn around the unit circle in $g(z)$.
- One possible $g(z)$ in the form of equation (139) that satisfies these conditions is

$$g(z) = \frac{z - \alpha}{1 - \alpha z} \tag{142}$$

where α is real such that $|\alpha| < 1$.

Lowpass-to-lowpass transformation

- Assuming that the passband edge frequency of the original lowpass filter is given by $\omega_{\rm p}$, and that we wish to transform the original filter into a lowpass filter with cutoff frequency at $\omega_{\rm p_1}$, that is, $g(e^{j\omega_{\rm p_1}}) = e^{j\omega_{\rm p}}$, the following relation must be valid:

$$e^{j\omega_{\rm p}} = \frac{e^{j\omega_{\rm p_1}} - \alpha}{1 - \alpha e^{j\omega_{\rm p_1}}} \tag{143}$$

and then

$$\alpha = \frac{e^{-j\left(\frac{\omega_{\rm p} - \omega_{\rm p_1}}{2}\right)} - e^{j\left(\frac{\omega_{\rm p} - \omega_{\rm p_1}}{2}\right)}}{e^{-j\left(\frac{\omega_{\rm p} + \omega_{\rm p_1}}{2}\right)} - e^{j\left(\frac{\omega_{\rm p} + \omega_{\rm p_1}}{2}\right)}} = \frac{\sin\left(\frac{\omega_{\rm p} - \omega_{\rm p_1}}{2}\right)}{\sin\left(\frac{\omega_{\rm p} + \omega_{\rm p_1}}{2}\right)} \tag{144}$$

- The desired transformation is then implemented by replacing z by $g(z)$ given in equation (142), with α calculated as indicated in equation (144).

Lowpass-to-highpass transformation

- If ω_{p_1} is the highpass filter band edge and ω_p is the lowpass filter cutoff frequency, the lowpass-to-highpass transformation function is given by

$$g(z) = -\frac{z+\alpha}{\alpha z+1} \tag{145}$$

where

$$\alpha = -\frac{\cos\left(\frac{\omega_p+\omega_{p_1}}{2}\right)}{\cos\left(\frac{\omega_p-\omega_{p_1}}{2}\right)} \tag{146}$$

Lowpass-to-bandpass transformation

- The lowpass-to-bandpass transformation is accomplished if the following mappings occur

$$g(1) = -1 \tag{147}$$

$$g(e^{-j\omega_{p_1}}) = e^{j\omega_p} \tag{148}$$

$$g(e^{j\omega_{p_2}}) = e^{j\omega_p} \tag{149}$$

$$g(-1) = -1 \tag{150}$$

where ω_{p_1} and ω_{p_2} are the band edges of the bandpass filter, and ω_p is the band edge of the lowpass filter.

- Since the bandpass filter has two passband edges, we need a second-order function $g(z)$ to accomplish the lowpass-to-bandpass transformation.

Lowpass-to-bandpass transformation

- After some manipulation, it can be inferred that the required transformation and its parameters are given by

$$g(z) = -\frac{z^2 + \alpha_1 z + \alpha_2}{\alpha_2 z^2 + \alpha_1 z + 1} \tag{151}$$

with

$$\alpha_1 = -\frac{2\alpha k}{k+1}; \quad \alpha_2 = \frac{k-1}{k+1} \tag{152}$$

where

$$\alpha = \frac{\cos\left(\frac{\omega_{p_2}+\omega_{p_1}}{2}\right)}{\cos\left(\frac{\omega_{p_2}-\omega_{p_1}}{2}\right)} \tag{153}$$

$$k = \cot\left(\frac{\omega_{p_2} - \omega_{p_1}}{2}\right)\tan\left(\frac{\omega_p}{2}\right) \tag{154}$$

Lowpass-to-bandstop transformation

- The lowpass-to-bandstop transformation function $g(z)$ is given by

$$g(z) = \frac{z^2 + \alpha_1 z + \alpha_2}{\alpha_2 z^2 + \alpha_1 z + 1} \tag{155}$$

with

$$\alpha_1 = -\frac{2\alpha}{k+1}; \quad \alpha_2 = \frac{1-k}{1+k} \tag{156}$$

where

$$\alpha = \frac{\cos\left(\frac{\omega_{p_2}+\omega_{p_1}}{2}\right)}{\cos\left(\frac{\omega_{p_2}-\omega_{p_1}}{2}\right)} \tag{157}$$

$$k = \tan\left(\frac{\omega_{p_2} - \omega_{p_1}}{2}\right) \tan\left(\frac{\omega_p}{2}\right) \tag{158}$$

Variable-cutoff filter design

- An interesting application for the frequency transformations is to design highpass and lowpass filters with variable cutoff frequency with the cutoff frequency being directly controlled by a single parameter α.
- This method can be best understood through an example, as given below.

Example 6.4

- Consider the lowpass notch filter

$$H(z) = 0.004 \frac{z^2 - \sqrt{2}z + 1}{z^2 - 1.8z + 0.96} \tag{159}$$

whose zeros are located at $z = \frac{\sqrt{2}}{2}(1 \pm \mathrm{j})$. Transform this filter into a highpass notch with a zero at frequency $\omega_{p_1} = \frac{\pi}{6}$ rad/sample.

- Plot the magnitude responses before and after the frequency transformation.

Example 6.4 - Solution

- Using the lowpass-to-highpass transformation given in equation (145), the highpass transfer function is of the form

$$H(z) = H_0 \frac{(\alpha^2+\sqrt{2}\alpha+1)(z^2+1)+(\sqrt{2}\alpha^2+4\alpha+\sqrt{2})z}{(0.96\alpha^2+1.8\alpha+1)z^2+(1.8\alpha^2+3.92\alpha+1.8)z+(\alpha^2+1.8\alpha+0.96)} \tag{160}$$

  with $H_0 = 0.004$.

- The parameter α can control the position of the zeros of the highpass notch filter.

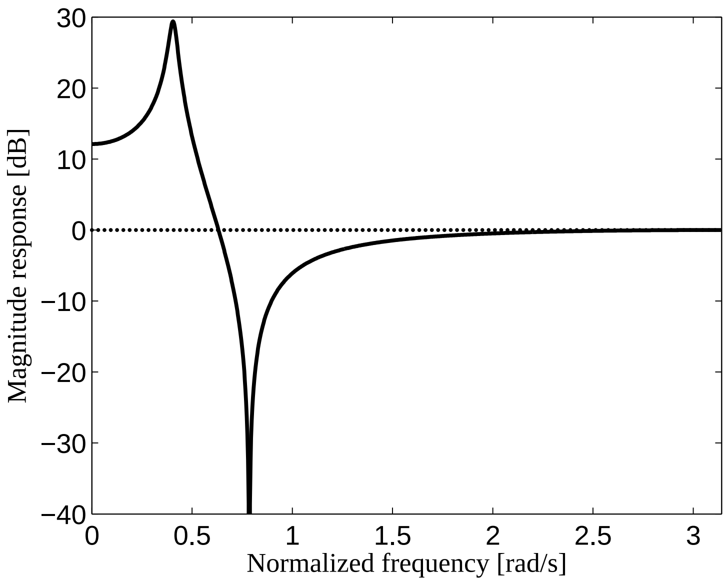
Example 6.4 - Solution

- For instance, in this example, as the original zero is at $\omega_{\mathrm{p}} = \frac{\pi}{4}$ rad/sample and the desired zero is at $\omega_{\mathrm{p}_1} = \frac{\pi}{6}$ rad/sample, the parameter α should be, as given in equation (146), equal to

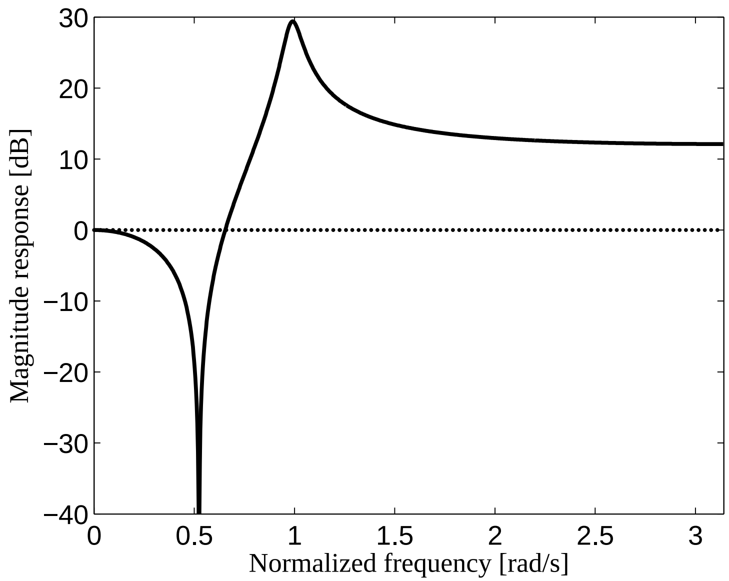
$$\alpha = -\frac{\cos\left(\frac{\frac{\pi}{4}+\frac{\pi}{6}}{2}\right)}{\cos\left(\frac{\frac{\pi}{4}-\frac{\pi}{6}}{2}\right)} = -0.8002 \tag{161}$$

- The magnitude responses corresponding to the lowpass and highpass transfer functions are seen in Figure 11. Notice how the new transfer function has indeed a zero at the desired position.

Example 6.4 - Solution



(a)



(b)

Figure 11: Magnitude responses of notch filters: (a) lowpass notch filter; (b) highpass notch filter.

Magnitude and phase approximation

- In this section, we discuss the approximation of IIR digital filters using optimization techniques aimed at the simultaneous approximation of the magnitude and phase responses.
- The same approach is useful in designing continuous-time filters and FIR filters.
- However, in the case of FIR filters more efficient approaches exist, as we have seen in Subsections 5.6.2 and 5.6.3.

Basic principles

- Assume that $\mathrm{H}(z)$ is the transfer function of an IIR digital filter. Then $\mathrm{H}(\mathrm{e}^{\mathrm{j}\omega})$ is a function of the filter coefficients, which are usually grouped into a single vector $\boldsymbol{\gamma}$, and of the independent variable $\theta = \omega$.

- The frequency response of a digital filter can be expressed as a function of the filter parameters $\boldsymbol{\gamma}$ and θ, that is $\mathrm{F}(\boldsymbol{\gamma}, \theta)$, and a desired frequency response is usually referred to as $\mathrm{f}(\theta)$.

- The complete specification of an optimization problem involves: definition of an objective function (also known as a cost function), determination of the form of the transfer function $\mathrm{H}(z)$ and its coefficients $\boldsymbol{\gamma}$, and the solution methods for the optimization problem.

Basic principles

- These three items are further discussed below.
- (a) Choosing the objective function: a widely used type of objective function in filter design is the weighted L_p norm, defined as

$$\|L(\boldsymbol{\gamma})\|_p = \left(\int_0^{\pi} W(\theta)|F(\boldsymbol{\gamma}, \theta) - f(\theta)|^p d\theta\right)^{\frac{1}{p}} \tag{162}$$

  where $W(\theta) > 0$ is the so-called weight function.
- Problems based on the L_p-norm minimization criteria with different values of p lead, in general, to different solutions.
- An appropriate choice for the value of p depends on the type of error which is acceptable for the given application.
- For example, when we wish to minimize the mean-square value of the error between the desired and the designed responses, we should choose $p = 2$.

Basic principles

- Another problem is the minimization of the maximum deviation between the desired specification and the designed filter by searching the space of parameters.
- This case, which is known as the Chebyshev or minimax criterion, corresponds to $p \rightarrow \infty$. This important result derived from the optimization theory can be stated more formally as:
- **Theorem:** For a given coefficient space P and a given angle space X_θ, there is a unique optimal minimax approximation $F(\boldsymbol{\gamma}_\infty^*, \theta)$ for $f(\theta)$. In addition, if the best L_p approximation for the function $f(\theta)$ is denoted by $F(\boldsymbol{\gamma}_p^*, \theta)$, then it can be demonstrated that

$$\lim_{p \rightarrow \infty} \boldsymbol{\gamma}_p^* = \boldsymbol{\gamma}_\infty^* \tag{163}$$

Basic principles

- This result shows that we can use any minimization program based on the L_p norm to find a minimax (or approximately minimax) solution, by progressively calculating the L_p optimal solution with, for instance, $p = 2, 4, 6$, and so on, indefinitely.
- Specifically, the minimax criterion for a continuous frequency function is best defined as

$$\|L(\boldsymbol{\gamma}^*)\|_\infty = \min_{\boldsymbol{\gamma} \in P} \{ \max_{\theta \in X_\theta} \{W(\theta)|F(\boldsymbol{\gamma}, \theta) - f(\theta)|\}\} \tag{164}$$

Basic principles

- In practice, due to several computational aspects, it is more convenient to use a simplified objective function given by

$$L_{2p}(\boldsymbol{\gamma}) = \sum_{k=1}^{K} W(\theta_k) \left(F(\boldsymbol{\gamma}, \theta_k) - f(\theta_k)\right)^{2p} \tag{165}$$

  where, by minimizing $L_{2p}(\boldsymbol{\gamma})$, we also minimize $\|L(\boldsymbol{\gamma})\|_{2p}$. In this case, the minimax solution is obtained by minimizing $L_{2p}(\boldsymbol{\gamma})$, for $p = 1, 2, 3$, and so on, indefinitely.

- The points θ_k are the angles chosen to sample the desired and the prototype frequency responses. These points, lying on the unit circle do not need to be equally spaced. In fact, we usually choose θ_k such that there are denser grids in the regions where the error function has more variations.

Basic principles

- The sort of filter designs described here can be applied to a large class of problems, in particular the design of filters with arbitrary magnitude response, phase equalizers, and filters with simultaneous specifications of magnitude and phase responses. The last two classes are illustrated below.

Basic principles

- (cont.)
  - Phase equalizer: the transfer function of a phase equalizer is (see Section 4.7.1)

$$H_l(z) = \prod_{i=1}^{M} \frac{a_{2i}z^2 + a_{1i}z + 1}{z^2 + a_{1i}z + a_{2i}} \tag{166}$$

Since its magnitude is 1, the objective function becomes

$$L_{2p}\tau(\boldsymbol{\gamma}, \tau_0) = \sum_{k=1}^{K} W(\theta_k)\left(\tau_l(\boldsymbol{\gamma}, \theta_k) - \tau_s(\theta_k) + \tau_0\right)^{2p} \tag{167}$$

where τ_s is the group delay of the original digital filter, $\tau_l(\boldsymbol{\gamma}, \theta_k)$ is the equalizer group delay and τ_0 is a constant delay, whose value minimizes $\sum_{k=1}^{K}(\tau_s(\theta_k) - \tau_0)^{2p}$.

Basic principles

- (cont.)
  - Simultaneous approximation of magnitude and group-delay responses: for this type of approximation, the objective function can be given by

$$
\begin{aligned}
L_{2p,2q}M, \tau(\boldsymbol{\gamma}, \tau_0) \;=\; & \delta \sum_{k=1}^{K} W_M(\theta_k) \left(M(\boldsymbol{\gamma}, \theta_k) - f(\theta_k) \right)^{2p} \\
& + (1-\delta) \sum_{r=1}^{R} W_\tau(\theta_k) \left(\tau(\boldsymbol{\gamma}, \theta_r) + \tau(\theta_r) - \tau_0 \right)^{2p}
\end{aligned}
\tag{168}
$$

    where $0 \leq \delta \leq 1$ and $\tau(\theta_r)$ is the group delay which we wish to equalize. Usually, in the simultaneous approximation of magnitude and group-delay responses, the numerator of $H(z)$ is forced to have zeros on the unit circle or in reciprocal pairs, such that the group delay is a function of the poles of $H(z)$ only. The task of the zeros would be to shape the magnitude response.

Basic principles

- (b) Choosing the form of the transfer function: one of the most convenient ways to describe an IIR $H(z)$ is the cascade form decomposition, because the filter stability can be easily tested and controlled.

- In this case, the coefficient vector $\boldsymbol{\gamma}$ is of the form

$$\boldsymbol{\gamma} = (\gamma'_{11}, \gamma'_{21}, m_{11}, m_{21}, \ldots, \gamma'_{1i}, \gamma'_{2i}, m_{1i}, m_{2i}, \ldots, H_0) \tag{169}$$

- Unfortunately, the expressions for the magnitude and group delay of $H(z)$, as a function of the coefficients of the second-order sections, are very complicated.

- The same comment holds for the expressions of the partial derivatives of $H(z)$ with respect to the coefficients, which are also required in the optimization algorithm.

- An alternative solution is to use the poles and zeros of the second-order sections represented in polar coordinates as parameters.

- In this case, the coefficient vector $\boldsymbol{\gamma}$ becomes

$$\boldsymbol{\gamma} = (r_{z1}, \phi_{z1}, r_{p1}, \phi_{p1}, \ldots, r_{zi}, \phi_{zi}, r_{pi}, \phi_{pi}, \ldots, k_0) \tag{170}$$

and the magnitude and group-delay responses are respectively expressed as

$$\begin{aligned} M(\boldsymbol{\gamma}, \omega) &= k_0 \prod_{i=1}^{m} \left\{ \frac{\left[1 - 2r_{zi}\cos(\omega - \phi_{zi}) + r_{zi}^2\right]^{\frac{1}{2}}}{\left[1 - 2r_{pi}\cos(\omega - \phi_{pi}) + r_{pi}^2\right]^{\frac{1}{2}}} \right\} \\ &\times \left\{ \frac{\left[1 - 2r_{zi}\cos(\omega + \phi_{zi}) + r_{zi}^2\right]^{\frac{1}{2}}}{\left[1 - 2r_{pi}\cos(\omega + \phi_{pi}) + r_{pi}^2\right]^{\frac{1}{2}}} \right\} \end{aligned} \tag{171}$$

$$\begin{aligned} \tau(\boldsymbol{\gamma}, \omega) &= \sum_{i=1}^{N} \left[\frac{1 - r_{pi}\cos(\omega - \phi_{pi})}{1 - 2r_{pi}\cos(\omega - \phi_{pi}) + r_{pi}^2} + \frac{1 - r_{pi}\cos(\omega + \phi_{pi})}{1 - 2r_{pi}\cos(\omega + \phi_{pi}) + r_{pi}^2} \right. \\ &\left. - \frac{1 - r_{zi}\cos(\omega - \phi_{zi})}{1 - 2r_{zi}\cos(\omega - \phi_{zi}) + r_{zi}^2} - \frac{1 - r_{zi}\cos(\omega + \phi_{zi})}{1 - 2r_{zi}\cos(\omega + \phi_{zi}) + r_{zi}^2} \right] \end{aligned} \tag{172}$$

Basic principles

- In an optimization problem such as this, the first- and second-order derivatives should be determined using closed-form formulas to speed up the convergence process.
- In fact, the use of numerical approximation to calculate such derivatives would make the optimization procedure too complex.

Basic principles

- The partial derivatives of the magnitude and group delay with respect to the radii and angles of the poles and zeros, that are required in the optimization processes, are:

$$\frac{\partial M}{\partial r_{zi}} = M(\boldsymbol{\gamma},\omega)\left[\frac{r_{zi}-\cos(\omega-\phi_{zi})}{1-2r_{zi}\cos(\omega-\phi_{zi})+r_{zi}^2}+\frac{r_{zi}-\cos(\omega+\phi_{zi})}{1-2r_{zi}\cos(\omega+\phi_{zi})+r_{zi}^2}\right] \tag{173}$$

$$\frac{\partial M}{\partial \phi_{zi}} = -M(\boldsymbol{\gamma},\omega)\left[\frac{r_{zi}\sin(\omega-\phi_{zi})}{1-2r_{zi}\cos(\omega+\phi_{zi})+r_{zi}^2}-\frac{r_{zi}\sin(\omega+\phi_{zi})}{1-2r_{zi}\cos(\omega+\phi_{zi})+r_{zi}^2}\right] \tag{174}$$

$$\frac{\partial \tau}{\partial r_{pi}} = \left\{\frac{(1+r_{pi}^2)\cos(\omega-\phi_{pi})-2r_{pi}}{\left[1-2r_{pi}\cos(\omega-\phi_{pi})+r_{pi}^2\right]^2}+\frac{(1+r_{pi}^2)\cos(\omega+\phi_{pi})-2r_{pi}}{\left[1-2r_{pi}\cos(\omega+\phi_{pi})+r_{pi}^2\right]^2}\right\} \tag{175}$$

$$\frac{\partial \tau}{\partial \phi_{pi}} = \left\{\frac{r_{pi}(1-r_{pi}^2)\sin(\omega-\phi_{pi})}{\left[1-2r_{pi}\cos(\omega-\phi_{pi})+r_{pi}^2\right]^2}-\frac{r_{pi}(1-r_{pi}^2)\sin(\omega+\phi_{pi})}{\left[1-2r_{pi}\cos(\omega+\phi_{pi})+r_{pi}^2\right]^2}\right\} \tag{176}$$

Basic principles

- We also need $\frac{\partial M}{\partial r_{pi}}$, $\frac{\partial M}{\partial \phi_{pi}}$, $\frac{\partial \tau}{\partial r_{zi}}$, and $\frac{\partial \tau}{\partial \phi_{zi}}$, which are similar to the expressions above.

- These derivatives are part of the expressions of the partial derivatives of the objective function with respect to the filter poles and zeros which are the derivatives used in the optimization algorithms employed.

- These derivatives are:

$$\frac{\partial L_{2p} M(\boldsymbol{\gamma})}{\partial r_{zi}} = \sum_{k=1}^{K} 2p W_M(\theta_k) \frac{\partial M}{\partial r_{zi}} \left(M(\boldsymbol{\gamma}, \theta_k) - f(\theta_k) \right)^{2p-1} \quad (177)$$

$$\frac{\partial L_{2p} \tau(\boldsymbol{\gamma})}{\partial r_{zi}} = \sum_{k=1}^{K} 2p W_\tau(\theta_k) \frac{\partial \tau}{\partial r_{zi}} \left(\tau(\boldsymbol{\gamma}, \theta_k) - f(\theta_k) \right)^{2p-1} \quad (178)$$

Basic principles

- Analogously, we need the expressions for $\frac{\partial L_{2p}M(\boldsymbol{\gamma})}{\partial \phi_{zi}}$, $\frac{\partial L_{2p}M(\boldsymbol{\gamma})}{\partial r_{pi}}$, $\frac{\partial L_{2p}M(\boldsymbol{\gamma})}{\partial \phi_{pi}}$, $\frac{\partial L_{2p}\tau(\boldsymbol{\gamma})}{\partial \phi_{zi}}$, $\frac{\partial L_{2p}\tau(\boldsymbol{\gamma})}{\partial r_{pi}}$, and $\frac{\partial L_{2p}\tau(\boldsymbol{\gamma})}{\partial \phi_{pi}}$, which are similar to the expressions given above.

- It is important to note that we are interested only in generating stable filters. Since we are performing a search for the minimum of the error function on the parameter space Γ, the region in which the optimal parameter should be searched is a restricted subspace $\Gamma_s = \{\boldsymbol{\gamma} \mid r_{pi} < 1, \forall i\}$.

Basic principles

- (c) Choosing the optimization procedure: there are several optimization methods suitable for solving the problem of filter approximation. Choosing the best method depends heavily on the designer's experience in dealing with this problem and on the available computer resources.

- The optimization algorithms used are such that they will converge only if the error function has a local minimum in the interior of the subspace Γ_s and not on the boundaries of Γ_s.

- In the present case, this is not a cause for concern because the magnitude and group delay of digital filters become large when a pole approaches the unit circle, and, as a consequence, there is no local minimum corresponding to poles on the unit circle.

Basic principles

- In this manner, if we start the search from the interior of Γ_s, that is, with all the poles strictly inside the unit circle, and constrain our search to the subspace Γ_s, a local minimum not located at the boundary of Γ_s will be reached for certain.
- Due to the importance of this step for setting up a procedure for designing IIR digital filters, for the sake of completeness and clarity of presentation, its discussion is left to the next subsection, which is devoted exclusively to it.

Multi-variable function minimization method

- An n-variable function $F(\mathbf{x})$ can be approximated by a quadratic function in a small region around a given operating point.
- For instance, in a region close to a point $\mathbf{x}_k$, we can write that

$$F(\mathbf{x}_k + \boldsymbol{\delta}_k) \approx F(\mathbf{x}_k) + \mathbf{g}^T(\mathbf{x}_k)\boldsymbol{\delta}_k + \frac{1}{2}\boldsymbol{\delta}_k^T \mathbf{H}(\mathbf{x}_k)\boldsymbol{\delta}_k \tag{179}$$

where

$$\mathbf{g}^T(\mathbf{x}_k) = \left[\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \ldots, \frac{\partial F}{\partial x_n}\right] \tag{180}$$

is the gradient vector of $F(\mathbf{x})$ at the operating point $\mathbf{x}_k$.

Multi-variable function minimization method

- The Hessian matrix $\mathbf{H}(\mathbf{x}_k)$ of $F(\mathbf{x})$ defined as

$$\mathbf{H}(\mathbf{x}_k) = \begin{bmatrix} \frac{\partial^2 F}{\partial x_1^2} & \frac{\partial^2 F}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 F}{\partial x_1 \partial x_n} \\ \frac{\partial^2 F}{\partial x_2 \partial x_1} & \frac{\partial^2 F}{\partial x_2^2} & \cdots & \frac{\partial^2 F}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F}{\partial x_n \partial x_1} & \frac{\partial^2 F}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 F}{\partial x_n^2} \end{bmatrix} \tag{181}$$

- Clearly, if $F(\mathbf{x})$ is a quadratic function, the right-hand side of equation (179) is minimized when

$$\delta_k = -\mathbf{H}^{-1}(\mathbf{x}_k)\mathbf{g}(\mathbf{x}_k) \tag{182}$$

Multi-variable function minimization method

- If, however, the function $F(\mathbf{x})$ is not quadratic and the operating point is far away from a local minimum, we can devise an algorithm which iteratively searches the minimum in the direction of δ_k as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \delta_k = \mathbf{x}_k - \alpha_k \mathbf{H}^{-1}(\mathbf{x}_k)\mathbf{g}(\mathbf{x}_k) \tag{183}$$

  where the convergence factor, α_k, is a scalar that minimizes, in the kth iteration, $F(\mathbf{x}_k + \delta_k)$ in the direction of δ_k.

- There are several procedures for determining the value of α_k, which can be considered as two classes: exact and inexact line searches.

Multi-variable function minimization method

- As a general rule of thumb, an inexact line search should be used when the operating point is far from a local minimum, because in these conditions it is appropriate to trade accuracy for faster results.
- However, when the parameters approach a minimum, accuracy becomes an important issue, and an exact line search is the best choice.
- The minimization procedure described above is widely known as the Newton method. The main drawbacks related to this method are the need for computation of the second-order derivatives of the objective function $F(\mathbf{x})$ with respect to the parameters in $\mathbf{x}$ and the necessity of inverting the Hessian matrix.

Multi-variable function minimization method

- Due to these two reasons, the most widely used methods for the solution of the simultaneous approximation of magnitude and phase are the so-called quasi-Newton methods.

- These methods are characterized by an attempt to build the inverse of the Hessian matrix, or an approximation of it, using the data obtained during the optimization process.

- The updated approximation of the Hessian inverse is used in each step of the algorithm in order to define the next direction in which to search for the minimum of the objective function.

- A general structure of an optimization algorithm suitable for designing digital filters is given below, where $\mathbf{P}_k$ is used as an estimate of the Hessian inverse.

Multi-variable function minimization method

- Algorithm:
  - (i) Algorithm initialization:

    Set the iteration counter as $k = 0$.

    Choose the initial vector $\mathbf{x}_0$ corresponding to a stable filter.

    Use the identity as the first estimate of the Hessian inverse, that is, $\mathbf{P}_k = \mathbf{I}$.

    Compute $F_0 = F(\mathbf{x}_0)$.
  - (ii) Convergence check:

    Check if convergence was achieved by using an appropriate criterion. For example, a criterion would be to verify if $F_k < \epsilon$ where ϵ is a pre-defined error threshold. An alternative criterion is to verify that $\| \mathbf{x}_k - \mathbf{x}_{k-1} \|^2 < \epsilon'$.

    If the algorithm has converged, go to step (iv), otherwise go on to step (iii).

Multi-variable function minimization method

- (cont.):
  - (iii) Algorithm iteration:

    Compute $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k)$.

    Set $\mathbf{s}_k = -\mathbf{P}_k \mathbf{g}_k$.

    Compute α_k that minimizes $F(\mathbf{x})$ in the direction of $\mathbf{s}_k$.

    Set $\delta_k = \alpha_k \mathbf{s}_k$.

    Upgrade the coefficient vector, $\mathbf{x}_{k+1} = \mathbf{x}_k + \delta_k$.

    Compute $F_{k+1} = F(\mathbf{x}_{k+1})$.

    Update $\mathbf{P}_k$, generating $\mathbf{P}_{k+1}$ (see discussion below).

    Increment k and return to step (ii).
  - (iv) Data output:

    Display $\mathbf{x}^* = \mathbf{x}_k$ and $F^* = F(\mathbf{x}^*)$.

Multi-variable function minimization method

- We should note that the way that the estimate $\mathbf{P}_k$ of the Hessian inverse is updated was omitted from the above algorithm. In fact, what distinguishes the different quasi-Newton methods is solely the way that $\mathbf{P}_k$ is updated.

- The most widely known quasi-Newton method is the Davidson-Fletcher-Powell method. Such an algorithm updates $\mathbf{P}_k$ in the form

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\delta}_k^T \Delta \mathbf{g}_k} - \frac{\mathbf{P}_k \Delta \mathbf{g}_k \Delta \mathbf{g}_k^T \mathbf{P}_k}{\Delta \mathbf{g}_k^T \mathbf{P}_k \Delta \mathbf{g}_k} \tag{184}$$

where $\Delta \mathbf{g}_k = \mathbf{g}_k - \mathbf{g}_{k-1}$.

Multi-variable function minimization method

- However, our experience has shown that the Broyden-Fletcher-Goldfarb-Shannon (BFGS) method is more efficient. This algorithm updates $\mathbf{P}_k$ in the form

$$\mathbf{P}_{k+1} = \mathbf{P}_k + \left(1 + \frac{\Delta\mathbf{g}_k^T \mathbf{P}_k \Delta\mathbf{g}_k}{\Delta\mathbf{g}_k^T \boldsymbol{\delta}_k}\right) \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T}{\Delta\mathbf{g}_k^T \boldsymbol{\delta}_k} - \frac{\boldsymbol{\delta}_k \Delta\mathbf{g}_k^T \mathbf{P}_k + \mathbf{P}_k \Delta\mathbf{g}_k \boldsymbol{\delta}_k^T}{\Delta\mathbf{g}_k^T \boldsymbol{\delta}_k} \tag{185}$$

  with $\Delta\mathbf{g}_k$ as before.

- It is important to notice that, in general, filter designers do not need to implement an optimization routine as they can employ optimization routines already available in a number of computer packages.

- What the designers are required to do is to express the objective function and optimization problem in a way that can be input to the chosen optimization routine.

Example 6.5

- Design a bandpass filter satisfying the specifications below:

$$\left.\begin{array}{l} M(\omega) = 1, \quad \text{for } 0.2\pi < \omega < 0.5\pi \\ M(\omega) = 0, \quad \text{for } 0 < \omega < 0.1\pi \text{ and } 0.6\pi < \omega < \pi \\ \tau(\omega) = L, \quad \text{for } 0.2\pi < \omega < 0.5\pi \end{array}\right\} \tag{186}$$

where L is constant.

Example 6.5 - Solution

- Since it is a simultaneous magnitude and phase approximation, the objective function is given by equation (168), with the expressions for magnitude and group delay, and their derivatives given in equations (171)–(178).
- We can start the design with an eighth-order transfer function with the characteristics given in Table 8.
- This initial filter is designed with the objective of approximating the desired magnitude specifications, and its average delay in the passband is used to estimate an initial value for L.

Example 6.5 - Solution

Table 8: Characteristics of the initial bandpass filter. Gain constant: $H_0 = 0.0588$.

Filter zeros $(r_{z_i}; \phi_{z_i}$ [rad]$)$	Filter poles $(r_{p_i}; \phi_{p_i}$ [rad]$)$
$r_{z_1} = 1.0; \quad \phi_{z_1} = 0.1740$	$r_{p_1} = 0.8182; \phi_{p_1} = 0.3030$
$r_{z_2} = 1.0; \quad \phi_{z_2} = -0.1740$	$r_{p_2} = 0.8182; \phi_{p_2} = -0.3030$
$r_{z_3} = 0.7927; \phi_{z_3} = 0.5622$	$r_{p_3} = 0.8391; \phi_{p_3} = 0.4837$
$r_{z_4} = 0.7927; \phi_{z_4} = -0.5622$	$r_{p_4} = 0.8391; \phi_{p_4} = -0.4837$
$r_{z_5} = 1.0; \quad \phi_{z_5} = 0.9022$	$r_{p_5} = 0.8346; \phi_{p_5} = 0.6398$
$r_{z_6} = 1.0; \quad \phi_{z_6} = -0.9022$	$r_{p_6} = 0.8346; \phi_{p_6} = -0.6398$
$r_{z_7} = 1.0; \quad \phi_{z_7} = 2.6605$	$r_{p_7} = 0.8176; \phi_{p_7} = 0.8053$
$r_{z_8} = 1.0; \quad \phi_{z_8} = -2.6605$	$r_{p_8} = 0.8176; \phi_{p_8} = -0.8053$

Example 6.5 - Solution

- In order to solve this optimization problem, we used a quasi-Newton program based on the BFGS method.
- Keeping the order of the starting filter at $n = 8$, we ran 100 iterations without obtaining noticeable improvements.
- We then increased the numerator and denominator orders by two, that is, we made $n = 10$, and after a few iterations the solution described in Table 9 was achieved.

Example 6.5 - Solution

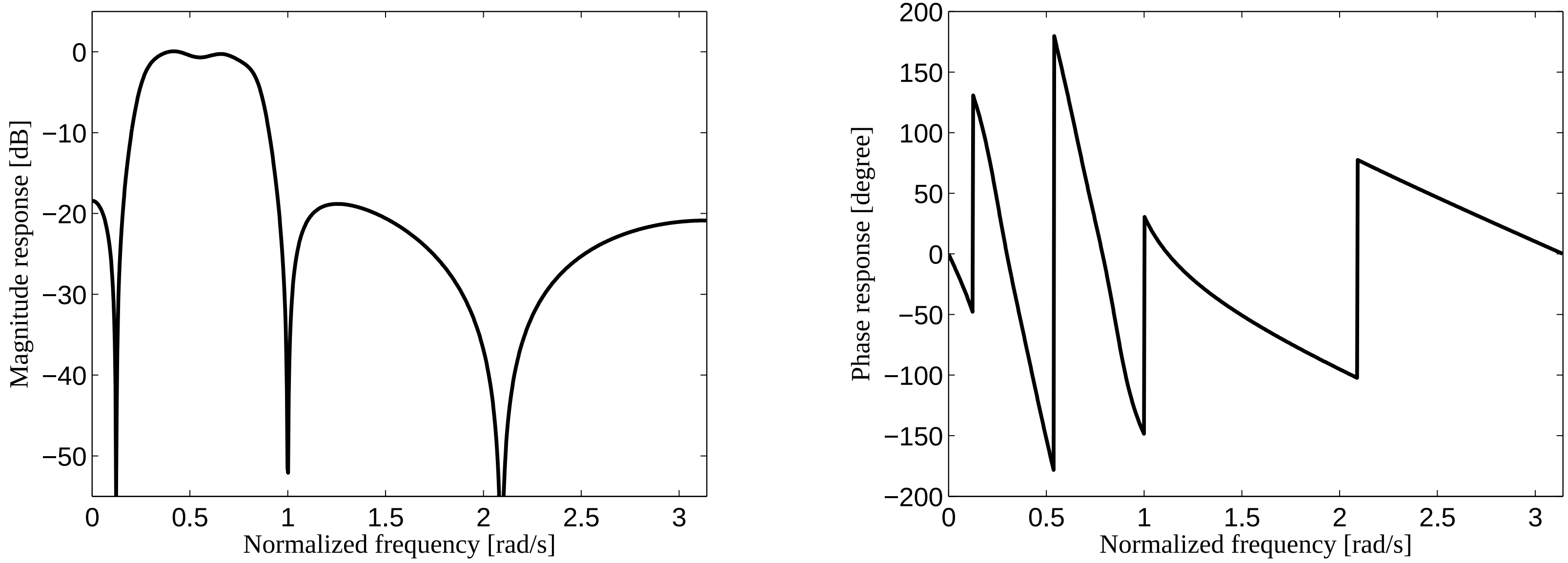
Table 9: Characteristics of the resulting bandpass filter. Gain constant: $H_0 = 0.058\,772\,50$.

Filter zeros $(r_{z_i}; \phi_{z_i}\,[\mathrm{rad}])$		Filter poles $(r_{p_i}; \phi_{p_i}\,[\mathrm{rad}])$	
$r_{z_1} = 1.0;$	$\phi_{z_1} = 0.1232$	$r_{p_1} = 0.0;$	$\phi_{p_1} = 0.0$
$r_{z_2} = 1.0;$	$\phi_{z_2} = -0.1232$	$r_{p_2} = 0.0;$	$\phi_{p_2} = 0.0$
$r_{z_3} = 0.7748;$	$\phi_{z_3} = 0.5545$	$r_{p_3} = 0.9072;$	$\phi_{p_3} = 0.2443$
$r_{z_4} = 0.7748;$	$\phi_{z_4} = -0.5545$	$r_{p_4} = 0.9072;$	$\phi_{p_4} = -0.2443$
$r_{z_5} = 1.2907;$	$\phi_{z_5} = 0.5545$	$r_{p_5} = 0.8654;$	$\phi_{p_5} = 0.4335$
$r_{z_6} = 1.2907;$	$\phi_{z_6} = -0.5545$	$r_{p_6} = 0.8654;$	$\phi_{p_6} = -0.4335$
$r_{z_7} = 1.0;$	$\phi_{z_7} = 1.0006$	$r_{p_7} = 0.8740;$	$\phi_{p_7} = 0.6583$
$r_{z_8} = 1.0;$	$\phi_{z_8} = -1.0006$	$r_{p_8} = 0.8740;$	$\phi_{p_8} = -0.6583$
$r_{z_9} = 1.0;$	$\phi_{z_9} = 2.0920$	$r_{p_9} = 0.9152;$	$\phi_{p_9} = 0.8604$
$r_{z_{10}} = 1.0;$	$\phi_{z_{10}} = -2.0920$	$r_{p_{10}} = 0.9152;$	$\phi_{p_{10}} = -0.8604$

Example 6.5 - Solution

- Figure 12 illustrates the resulting frequency response.
- The attenuation at the first stopband edges are 18.09 dB and 18.71 dB, respectively.
- The attenuation at the second stopband edges are 18.06 dB and 19.12 dB, respectively.
- The passband attenuation at the edges are 0.69 dB and 0.71 dB, the two passband peaks have gains of 0.50 dB and 0.41 dB, whereas the attenuation at the passband minimum point is 0.14 dB.
- The group-delay values at the beginning of the passband, at the passband minimum, and at the end of the passband are 14.02 s, 12.09 s, and 14.38 s, respectively.

Example 6.5 - Solution



(a) (b)

Figure 12: Optimized bandpass filter: (a) magnitude response; (b) phase response.

Time-domain approximation

- In some applications, time-domain specifications are given to the filter designer. In these cases the objective is to design a transfer function $H(z)$ such that the corresponding impulse response h_n is as close as possible to a given sequence g_n, for $n = 0, 1, \ldots, (K-1)$, where

$$H(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} = h_0 + h_1 z^{-1} + h_2 z^{-2} + \cdots \quad (187)$$

- Since $H(z)$ has $(M+N+1)$ coefficients, if $K = (M+N+1)$ there is at least one transfer function available which satisfies the specifications. This solution can be obtained through optimization, as follows.

Time-domain approximation

- By equating

$$H(z) = g_0 + g_1 z^{-1} + \cdots + g_{M+N} z^{-(M+N)} + \cdots \tag{188}$$

and considering the z-transform products as convolutions in the time domain, we can write, from equations (187) and (188), that

$$\sum_{n=0}^{N} a_n g_{i-n} = \begin{cases} b_i, & \text{for } i = 0, 1, \ldots, M \\ 0, & \text{for } i > M \end{cases} \tag{189}$$

Time-domain approximation

- Now, assuming that $g_n = 0$, for all $n < 0$, this equation can be rewritten in matrix form as

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_M \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \cdots & 0 \\ g_2 & g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_M & g_{M-1} & g_{M-2} & \cdots & g_{M-N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{M+N} & g_{M+N-1} & g_{M+N-2} & \cdots & g_M \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ a_N \end{bmatrix} \tag{190}$$

Time-domain approximation

- Which can be partitioned as

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_M \end{bmatrix} = \begin{bmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \cdots & 0 \\ g_2 & g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_M & g_{M-1} & g_{M-2} & \cdots & g_{M-N} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \tag{191}$$

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} g_{M+1} & \cdots & g_{M-N+1} \\ \vdots & \ddots & \vdots \\ g_{K-1} & \cdots & g_{K-N-1} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} \tag{192}$$

Time-domain approximation

- Or, equivalently,

$$\begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} & \mathbf{G}_1 & \\ \mathbf{g}_2 & & \mathbf{G}_3 \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{a} \end{bmatrix} \tag{193}$$

where $\mathbf{g}_2$ is a column vector and $\mathbf{G}_3$ is an $N \times N$ matrix.

- If $\mathbf{G}_3$ is nonsingular, the coefficients **a** are given by

$$\mathbf{a} = -\mathbf{G}_3^{-1}\mathbf{g}_2 \tag{194}$$

- If $\mathbf{G}_3$ is singular of rank $R < N$, there are infinite solutions, one of which is obtained by forcing the first $(N - R)$ entries of **a** to be null.

Time-domain approximation

- With **a** available, **b** can be computed as

$$\mathbf{b} = \mathbf{G}_1 \begin{bmatrix} 1 \\ \mathbf{a} \end{bmatrix} \tag{195}$$

- The main differences between the filters designed with different values of M and N, while keeping $K = (M + N + 1)$ constant, are the values of h_k, for $k > K$.

Time-domain approximation - Approximate approach

- A solution that is in general satisfactory is obtained by replacing the null vector in equation (193) by a vector $\hat{\mathbf{\epsilon}}$ whose magnitude should be minimized.
- In that manner, equation (193) becomes

$$\begin{bmatrix} \mathbf{b} \\ \hat{\mathbf{\epsilon}} \end{bmatrix} = \begin{bmatrix} & \mathbf{G}_1 \\ \mathbf{g}_2 & \mathbf{G}_3 \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{a} \end{bmatrix} \tag{196}$$

- Given the prescribed g_n and the values of N and M, we then have to find a vector **a** such that $\left(\hat{\mathbf{\epsilon}}^{\mathsf{T}}\hat{\mathbf{\epsilon}}\right)$ is minimized, with

$$\hat{\mathbf{\epsilon}} = \mathbf{g}_2 + \mathbf{G}_3\mathbf{a} \tag{197}$$

Time-domain approximation - Approximate approach

- The value of **a** which minimizes $(\hat{\epsilon}^{\mathsf{T}}\hat{\epsilon})$ is the normal equation solution,

$$\mathbf{G}_3^{\mathsf{T}}\mathbf{G}_3\mathbf{a} = -\mathbf{G}_3^{\mathsf{T}}\mathbf{g}_2 \tag{198}$$

- If the rank of $\mathbf{G}_3$ is N, then the rank of $\mathbf{G}_3^{\mathsf{T}}\mathbf{G}_3$ is also N, and, therefore, the solution is unique, being given by

$$\mathbf{a} = -[\mathbf{G}_3^{\mathsf{T}}\mathbf{G}_3]^{-1}\mathbf{G}_3^{\mathsf{T}}\mathbf{g}_2 \tag{199}$$

- On the other hand, if the rank of $\mathbf{G}_3$ is $R < N$, we should force $a_i = 0$, for $i = 0, 1, \ldots, (R-1)$, as before, and redefine the problem.

Time-domain approximation - Approximate approach

- It is important to point out that the procedure described above does not lead to a minimum squared error in the specified samples.
- In fact, the squared error is given by

$$\mathbf{e}^{\mathsf{T}}\mathbf{e} = \sum_{n=0}^{K} (g_n - h_n)^2 \qquad (200)$$

where g_n and h_n are the desired and obtained impulse responses, respectively.

- In order to obtain **b** and **a** which minimize $\mathbf{e}^{\mathsf{T}}\mathbf{e}$, we need an iterative process. The time-domain approximation can also be formulated as a system identification problem.

Example 6.6

- Design a digital filter characterized by $M = 3$ and $N = 4$ such that its impulse response approximates the following sequence:

$$g_n = \frac{1}{3}\left[\frac{1}{4^{n+1}} + e^{-n-1} + \frac{1}{(n+2)}\right]u(n) \qquad (201)$$

for $n = 0, 1, \ldots, 7$.

Example 6.6 - Solution

- Using $M = 3$ and $N = 4$, one gets

$$\mathbf{G}_1 = \begin{bmatrix} g_0 & 0 & 0 & 0 & 0 \\ g_1 & g_0 & 0 & 0 & 0 \\ g_2 & g_1 & g_0 & 0 & 0 \\ g_3 & g_2 & g_1 & g_0 & 0 \end{bmatrix} \qquad (202)$$

$$\mathbf{g}_2 = \begin{bmatrix} g_4 & g_5 & g_6 & g_7 \end{bmatrix}^{\mathrm{T}} \qquad (203)$$

$$\mathbf{G}_3 = \begin{bmatrix} g_3 & g_2 & g_1 & g_0 \\ g_4 & g_3 & g_2 & g_1 \\ g_5 & g_4 & g_3 & g_2 \\ g_6 & g_5 & g_4 & g_3 \end{bmatrix} \qquad (204)$$

Example 6.6 - Solution

- As $\mathbf{G}_3$ is nonsingular, we can use equations (197) and (195) to determine

$$H(z) = \frac{0.3726z^3 - 0.6446z^2 + 0.3312z - 0.0466}{z^4 - 2.2050z^3 + 1.6545z^2 - 0.4877z + 0.0473} \tag{205}$$

  which has the exact desired impulse response for $n = 0, 1, \ldots, 7$.

- The impulse response corresponding to the transfer function above is depicted in Figure 13, together with the prescribed impulse response.

- The responses are the same in the first few iterations, and they become distinct for $n > 7$, as expected, because we have only eight coefficients to adjust.

Example 6.6 - Solution

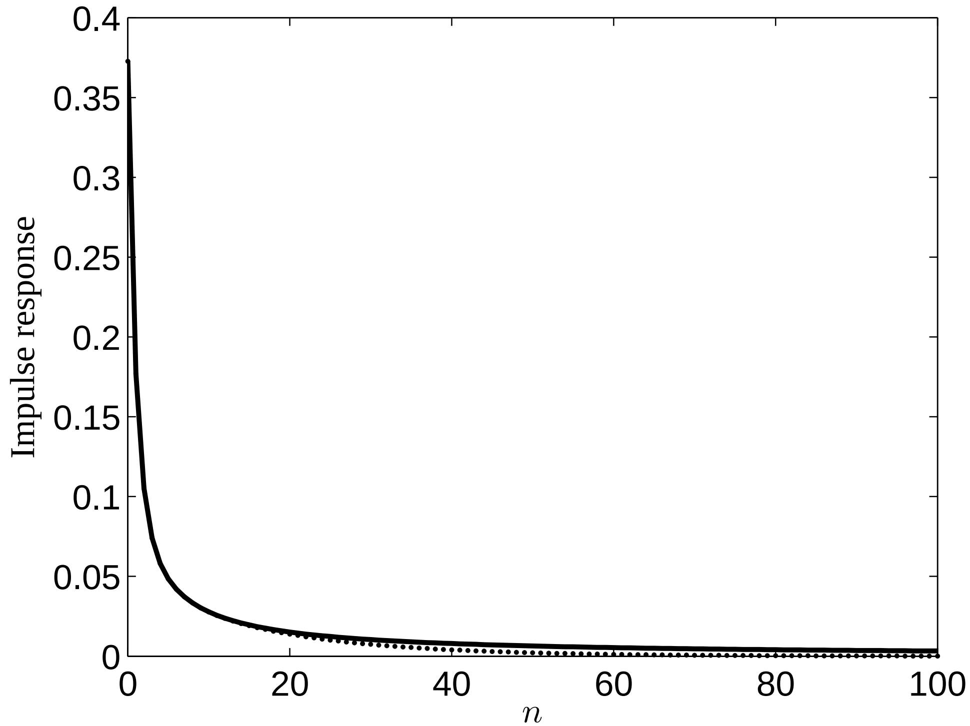


Figure 13: Impulse responses: desired (solid line) and obtained (dotted line).

Do-it-yourself: IIR filter approximations

- **Experiment 6.1:** The elliptic bandstop filter specified in Example 6.3 can be readily designed in MATLAB as follows:

```
Ap = 0.5; Ar = 65;
wr1 = 850/5000; wr2 = 1150/5000;
wp1 = 980/5000; wp2 = 1020/5000;
wp = [wp1 wp2]; wr = [wr1 wr2];
[n,wn] = ellipord(wp,wr,Ap,Ar);
[b,a] = ellip(n,Ap,Ar,wp);
```

- In this script, commands `ellipord` and `ellip` require a frequency normalization such that the $\bar{\Omega}_s = 2$, thus explaining all divisions by $\frac{\Omega_s}{2} = 5000$.

- Similar Butterworth or Chebyshev filters can be designed using `butterord`-`butter` or `chebyord`-`cheby` commands, respectively.

Do-it-yourself: IIR filter approximations

- The group delay response, determined with the `grpdelay` command, for the elliptic filter is seen in Figure 14.

- This figure indicates that two similar frequencies within the filter passband can suffer quite different delays.

- For instance, frequencies $f_1 = 980$ rad/s and $f_2 = 990$ rad/s are delayed in approximately 300 and 150 samples, respectively, corresponding in this case to a difference of about

$$\Delta t = \frac{300 - 150}{F_s} = \frac{150}{\frac{\Omega_s}{2\pi}} = 94 \text{ ms}$$

Do-it-yourself: IIR filter approximations

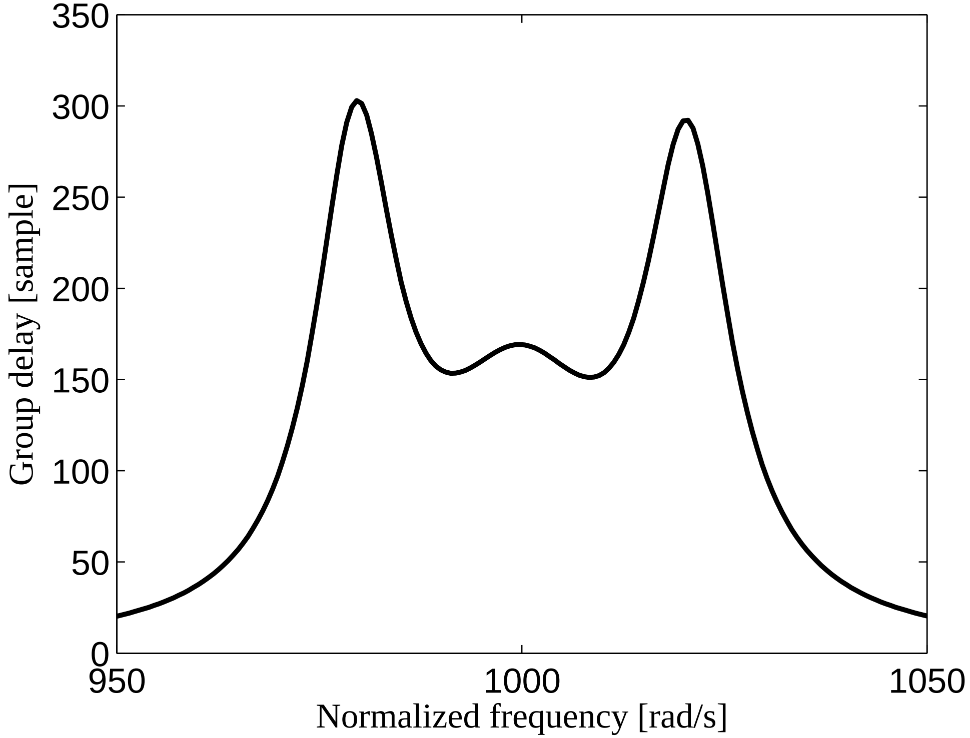


Figure 14: Group delay response in passband of elliptic filter.

Do-it-yourself: IIR filter approximations

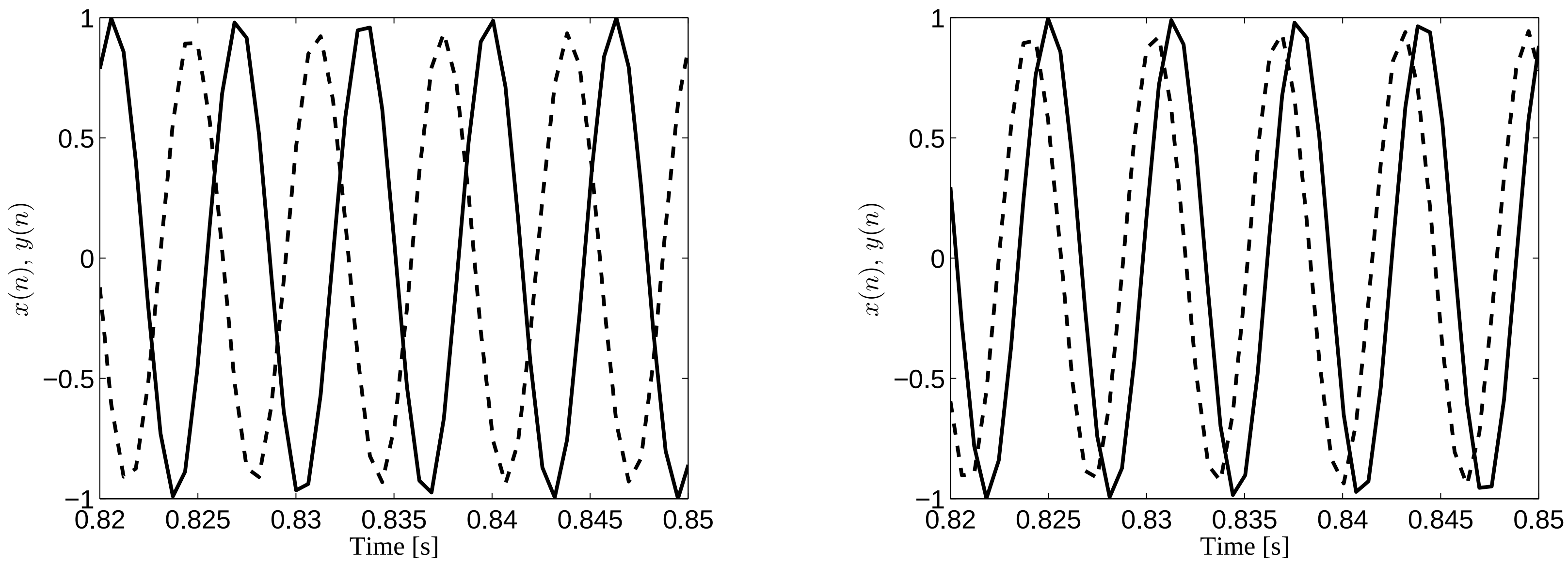
- Figure 15 compares the input and output signals for each frequency f_1 and f_2 as determined by the following script:

```
Fs = 10000/(2*pi); Ts = 1/Fs; time = 0:Ts:(1-Ts);
f1 = 980; f2 = 990;
x1 = cos(f1.*time); y1 = filter(b,a,x1);
x2 = cos(f2.*time); y2 = filter(b,a,x2);
```

- When the input signal presents a richer spectral component, this delay difference may cause severe distortion on the output signal.

- In such cases, a delay equalizer must be employed or the designer should opt for an FIR filter with perfectly linear phase.

Do-it-yourself: IIR filter approximations



(a) (b)

Figure 15: Input (solid line) and output (dashed line) signals for elliptic bandpass filter: (a) $f_1 = 980$ rad/s; (b) $f_2 = 990$ rad/s.

Do-it-yourself: IIR filter approximations

- **Experiment 6.2:** Consider the analog transfer function of the normalized-lowpass Chebyshev filter in Example 6.1, repeated here for convenience

$$H_a(s) = 0.4913 \frac{1}{s^3 + 0.9883s^2 + 1.2384s + 0.4913} \tag{206}$$

- The corresponding discrete-time transfer function $H(z)$ obtained with the bilinear transformation method with $F_s = 2$ Hz can be determined in MATLAB using the command lines

```
b = [0.4913]; a = [1 0.9883 1.2384 0.4913]; Fs = 2;
[bd,ad] = bilinear(b,a,Fs);
```

where `bd` and `ad` receive the numerator and denominator coefficients, respectively, of $H(z)$ such that

$$H(z) = \frac{0.0058(z^3 + 3z^2 + 3z + 1)}{z^3 - 2.3621z^2 + 2.0257z - 0.6175} \tag{207}$$

Do-it-yourself: IIR filter approximations

- The magnitude responses of $H_a(s)$ and $H(z)$ are shown in Figure 16.

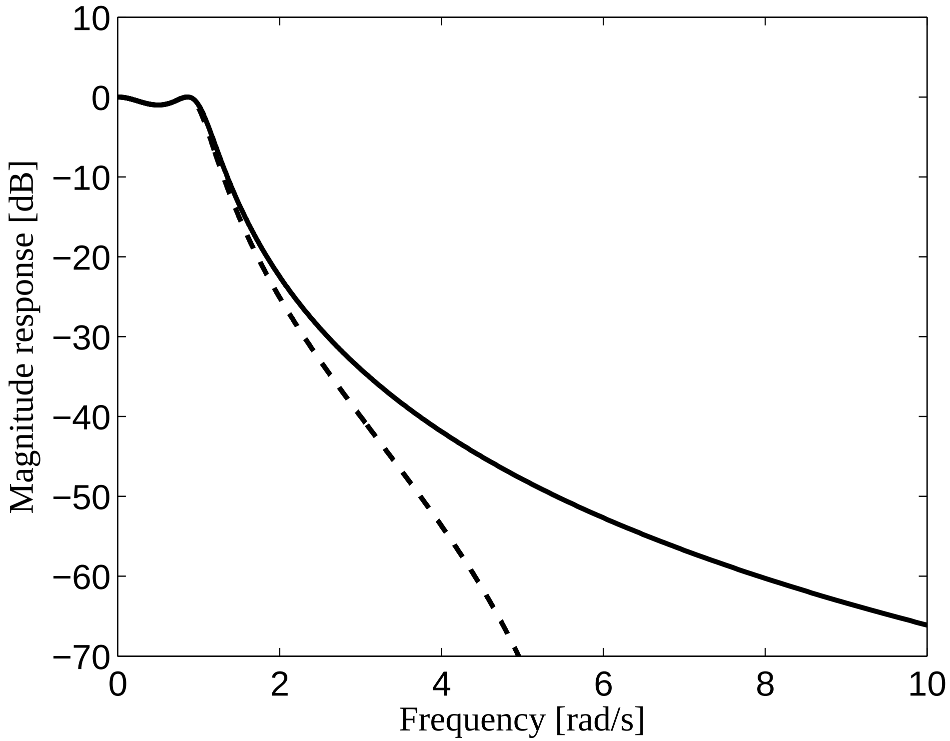


Figure 16: Analog (solid line) and digital (dashed line) magnitude responses related by bilinear transformation method with $F_s = 2$ Hz.

Do-it-yourself: IIR filter approximations

- Interesting enough, the bilinear transformation can be implemented as a coefficient mapping between the two transfer functions.
- If we write

$$H_a(s) = \frac{\hat{b}_N s^n + \hat{b}^{N-1} s^{N-1} + \cdots + \hat{b}_1 s + \hat{b}_0}{\hat{a}_N s^n + \hat{a}^{N-1} s^{N-1} + \cdots + \hat{a}_1 s + \hat{a}_0} \tag{208}$$

$$H(z) = \frac{b_N z^n + b^{N-1} z^{N-1} + \cdots + b_1 z + b_0}{a_N z^n + a^{N-1} z^{N-1} + \cdots + a_1 z + a_0} \tag{209}$$

and define the coefficient vectors

$$\hat{\mathbf{a}} = [\hat{a}_N \ \ \hat{a}_{N-1} \ \ \ldots \ \ \hat{a}_0]^T; \ \hat{\mathbf{b}} = \left[\hat{b}_N \ \ \hat{b}_{N-1} \ \ \ldots \ \ \hat{b}_0\right]^T \tag{210}$$

$$\mathbf{a} = [a_N \ \ a_{N-1} \ \ \ldots \ \ a_0]^T; \ \mathbf{b} = [b_N \ \ b_{N-1} \ \ \ldots \ \ b_0]^T \tag{211}$$

Do-it-yourself: IIR filter approximations

- One may then write that:

$$\mathbf{a} = \mathbf{P}_{N+1} \Delta_{N+1} \hat{\mathbf{a}} \tag{212}$$

$$\mathbf{b} = \mathbf{P}_{N+1} \Delta_{N+1} \hat{\mathbf{b}} \tag{213}$$

where

$$\Delta_{N+1} = \text{diag}\left[\left(\frac{2}{T}\right)^{N}, \left(\frac{2}{T}\right)^{N-1}, \ldots, \left(\frac{2}{T}\right), 1 \right] \tag{214}$$

Do-it-yourself: IIR filter approximations

- And $\mathbf{P}_{N+1}$ is an $(N+1) \times (N+1)$ Pascal matrix with the following properties:
  - All elements in the first row are equal to 1.
  - Elements of first column are determined, for $i = 1, 2, \ldots, (N+1)$, as

$$P_{i,1} = (-1)^{i-1} \frac{N!}{(N-i+1)!(i-1)!} \tag{215}$$

  - The remaining elements, for $i, j = 2, 3, \ldots, (N+1)$, are given by

$$P_{i,j} = P_{i-1,j} + P_{i-1,j-1} + P_{i,j-1} \tag{216}$$

Do-it-yourself: IIR filter approximations

- In this experiment, since $Fs = 2$ and $N = 3$ we get

$$\mathbf{P}_{N+1} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -3 & -1 & 1 & 3 \\ 3 & -1 & -1 & 3 \\ -1 & 1 & -1 & 1 \end{bmatrix}; \; \Delta_{N+1} = \begin{bmatrix} 4^3 & 0 & 0 & 0 \\ 0 & 4^2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{217}$$

Do-it-yourself: IIR filter approximations

- Such that

$$\begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 64 & 16 & 4 & 1 \\ -192 & -16 & 4 & 3 \\ 192 & -16 & -4 & 3 \\ -64 & 16 & -4 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0.9883 \\ 1.2384 \\ 0.4913 \end{bmatrix} = \begin{bmatrix} 85.2577 \\ -201.3853 \\ 172.7075 \\ -52.6495 \end{bmatrix} \quad (218)$$

$$\begin{bmatrix} b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} 64 & 16 & 4 & 1 \\ -192 & -16 & 4 & 3 \\ 192 & -16 & -4 & 3 \\ -64 & 16 & -4 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.4913 \end{bmatrix} = \begin{bmatrix} 0.4913 \\ 1.4739 \\ 1.4739 \\ 0.4913 \end{bmatrix} \quad (219)$$

which correspond to the same discrete-time transfer function as before, after proper normalization forcing $a_N = 1$.