Digital Signal Processing

Infinite Impulse Response Filters

Moslem Amiri, Václav Přenosil
Masaryk University

Understanding Digital Signal Processing, Third Edition, Richard Lyons
(0-13-261480-4) © Pearson Education, 2011.

Introduction

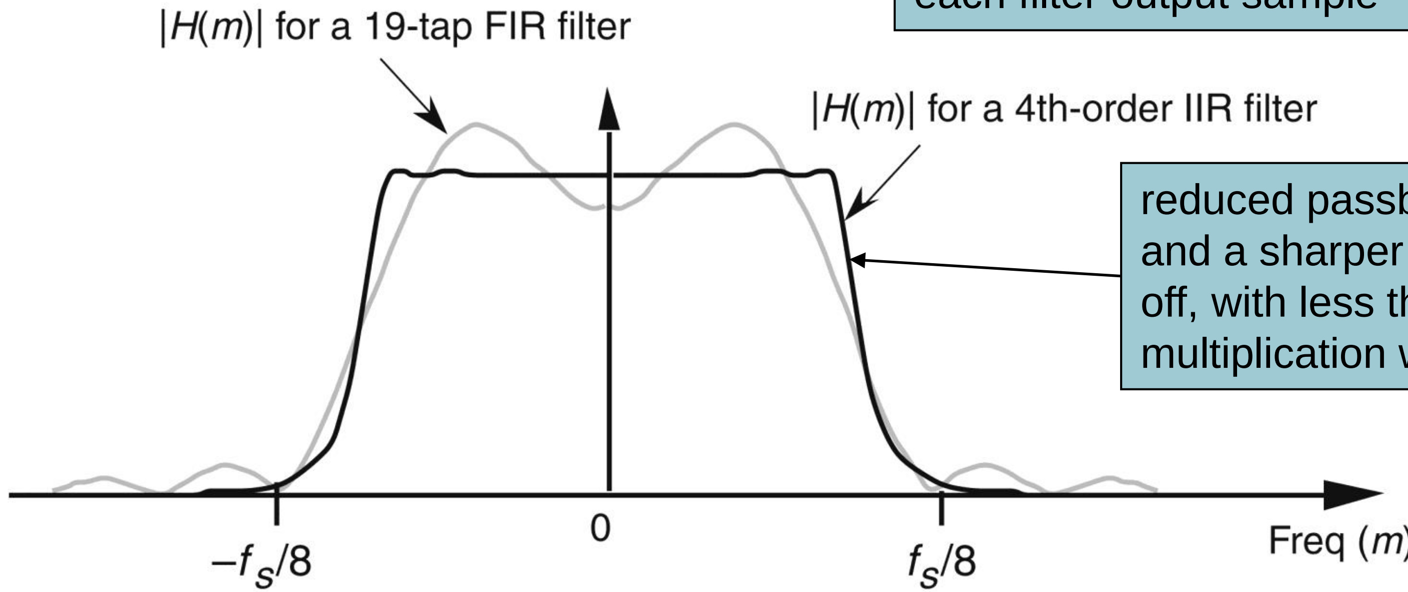
- Infinite impulse response (IIR) digital filters
  - Are fundamentally different from FIR filters
    - FIR filter output samples depend only on past input samples
    - Each IIR filter output sample depends on previous input samples *and* previous filter output samples
      - IIR filters have *memory* of past outputs (require feedback)
  - As in all feedback systems, perturbations at IIR filter input could cause filter output to become unstable and oscillate indefinitely
    - *Infinite impulse response*
  - IIR filters have more complicated structures (block diagrams), are harder to design and analyze, and do not have linear phase responses

Introduction

- Why use an IIR filter?
  - Because they are very efficient
    - IIR filters require far fewer multiplications per filter output sample to achieve a given frequency magnitude response
    - From a hardware standpoint, IIR filters can be very fast, allowing us to build real-time IIR filters that operate over much higher sample rates than FIR filters

Introduction

where the 19-tap FIR filter requires 19 multiplications per filter output sample, the 4th-order IIR filter requires only 9 multiplications for each filter output sample



reduced passband ripple and a sharper filter roll-off, with less than half the multiplication workload

Figure 6–1 Comparison of the frequency magnitude responses of a 19-tap lowpass FIR filter and a 4th-order lowpass IIR filter.

Introduction

- IIR vs. FIR
  - An FIR filter's frequency response with very steep transition regions requires a very long impulse response
  - The maximum number of FIR filter taps we can have (length of impulse response) depends on how fast our hardware can perform the required number of multiplications and additions to get a filter output before the next input sample arrives
  - IIR filters can be designed to have impulse responses longer than their number of taps
    - Thus, IIR filters can give much better filtering for a given number of multiplications per output sample

An Introduction to IIR Filters

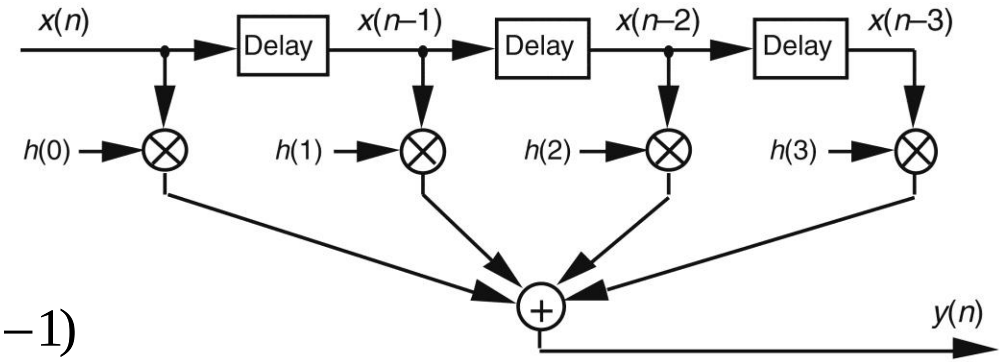
- If IIR filter's input suddenly becomes all zeros, its output could remain nonzero forever
  - This is because of feedback structure of their delay units, multipliers, and adders

An Introduction to IIR Filters

difference equation

$$y(n) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + h(3)x(n-3)$$

(a)



(b)

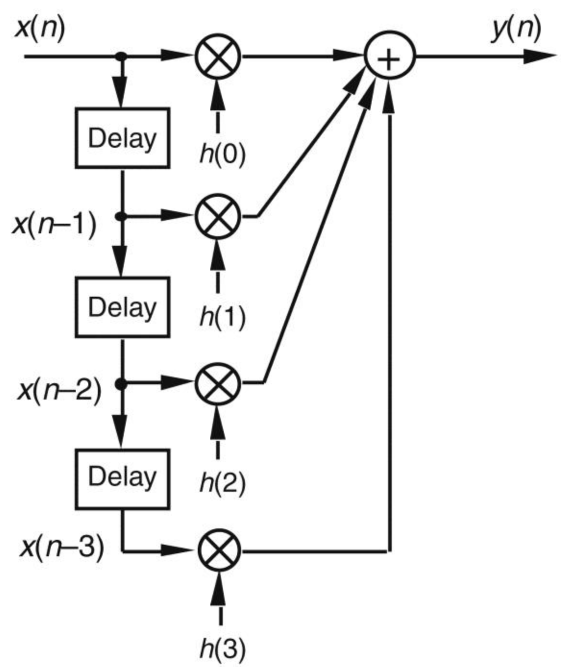


Figure 6–2 FIR digital filter structures: (a) traditional FIR filter structure; (b) re-arranged, but equivalent, FIR filter structure.

An Introduction to IIR Filters

$$y(n) = b(0)x(n) + b(1)x(n-1) + b(2)x(n-2) + b(3)x(n-3)$$
$$+ a(1)y(n-1) + a(2)y(n-2) + a(3)y(n-3)$$

difference equation describing this IIR filter

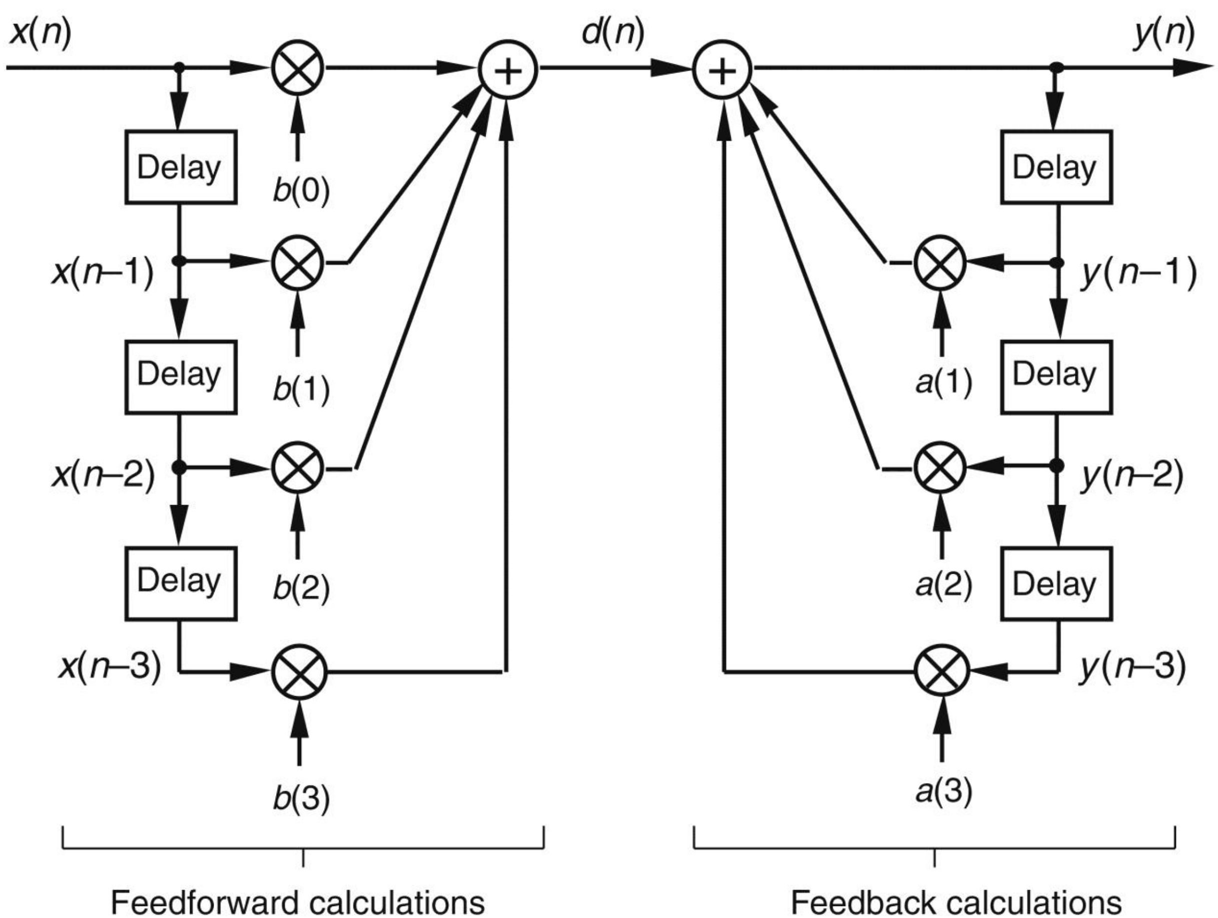


Figure 6–3 IIR digital filter structure showing feedforward and feedback calculations.

An Introduction to IIR Filters

- How determine *a*(*k*) and *b*(*k*) IIR filter coefficients
  - Window method of lowpass FIR filter design
    - Define frequency response of desired FIR filter → take inverse Fourier transform → shift that transform result → we get filter's time-domain impulse response
    - Due to the nature of transversal FIR filters, the desired *h*(*k*) filter coefficients turn out to be exactly equal to the impulse response sequence
  - Following that same procedure with IIR filters
    - Desired frequency response of IIR filter → inverse Fourier transform → time-domain impulse response
    - But there's no direct method for computing IIR filter's *a*(*k*) and *b*(*k*) coefficients from impulse response
    - FIR filter design techniques cannot be used here

An Introduction to IIR Filters

- Standard IIR filter design techniques
  - Fall into three basic classes: impulse invariance, bilinear transform, and optimization methods
  - These design methods use a discrete sequence, mathematical transformation process known as the *z*-transform whose origin is Laplace transform used in analyzing of continuous systems

The Laplace Transform

- Laplace transform
  - A mathematical method of solving linear differential equations
  - Process of using Laplace transform
    - A time-domain differential equation is written that describes input/output relationship of a physical system
    - The differential equation is Laplace transformed, converting it to an algebraic equation
    - Standard algebraic techniques are used to determine desired output function’s equation in Laplace domain
    - The desired Laplace output equation is, then, inverse Laplace transformed to yield the desired time-domain output function’s equation

The Laplace Transform

- Laplace transform of a continuous time-domain function $f(t)$

$$F(s)=\int_0^\infty f(t)e^{-st}dt$$

  - Variable s is the complex number $s = \sigma + j\omega$
  - $f(t)$ is defined only for positive time ($t > 0$)
    - Systems where system conditions for negative time ($t < 0$) are not needed (*one-sided*) are referred to as *causal systems*
  - Causal systems may have initial conditions at $t = 0$ that must be taken into account, but we don't need to know what the system was doing prior to $t = 0$

The Laplace Transform

- $s = \sigma + j\omega$
  - σ is a real number
  - ω is frequency in radians/second
  - e^{-st} is dimensionless → s has dimension of 1/time, or frequency
    - Laplace variable s is often called a complex frequency

The Laplace Transform

- Laplace transform

$$F(s) = \int_0^\infty f(t) e^{-st} dt$$

$$e^{-st} = e^{-(\sigma + j\omega)t} = e^{-\sigma t} e^{-j\omega t} = \frac{e^{-j\omega t}}{e^{\sigma t}} = \frac{\cos(\omega t)}{e^{\sigma t}} - j\frac{\sin(\omega t)}{e^{\sigma t}}$$

  - $e^{-j\omega t}$ is a unity-magnitude phasor rotating clockwise around origin of a complex plane at a frequency of ω radians/second
  - $e^{\sigma t}$ is a real number whose value is one at $t = 0$
    - As t increases, $e^{\sigma t}$ gets larger (when σ is positive), and complex e^{-st} phasor's magnitude gets smaller as phasor rotates on complex plane
    - The tip of that phasor traces out a curve spiraling in toward origin of complex plane

The Laplace Transform

real part of $F(s)$, for a particular value of s, is correlation of $f(t)$ with a cosine wave of frequency ω and a damping factor of σ, and imaginary part of $F(s)$ is correlation of $f(t)$ with a sinewave of frequency ω and a damping factor of σ

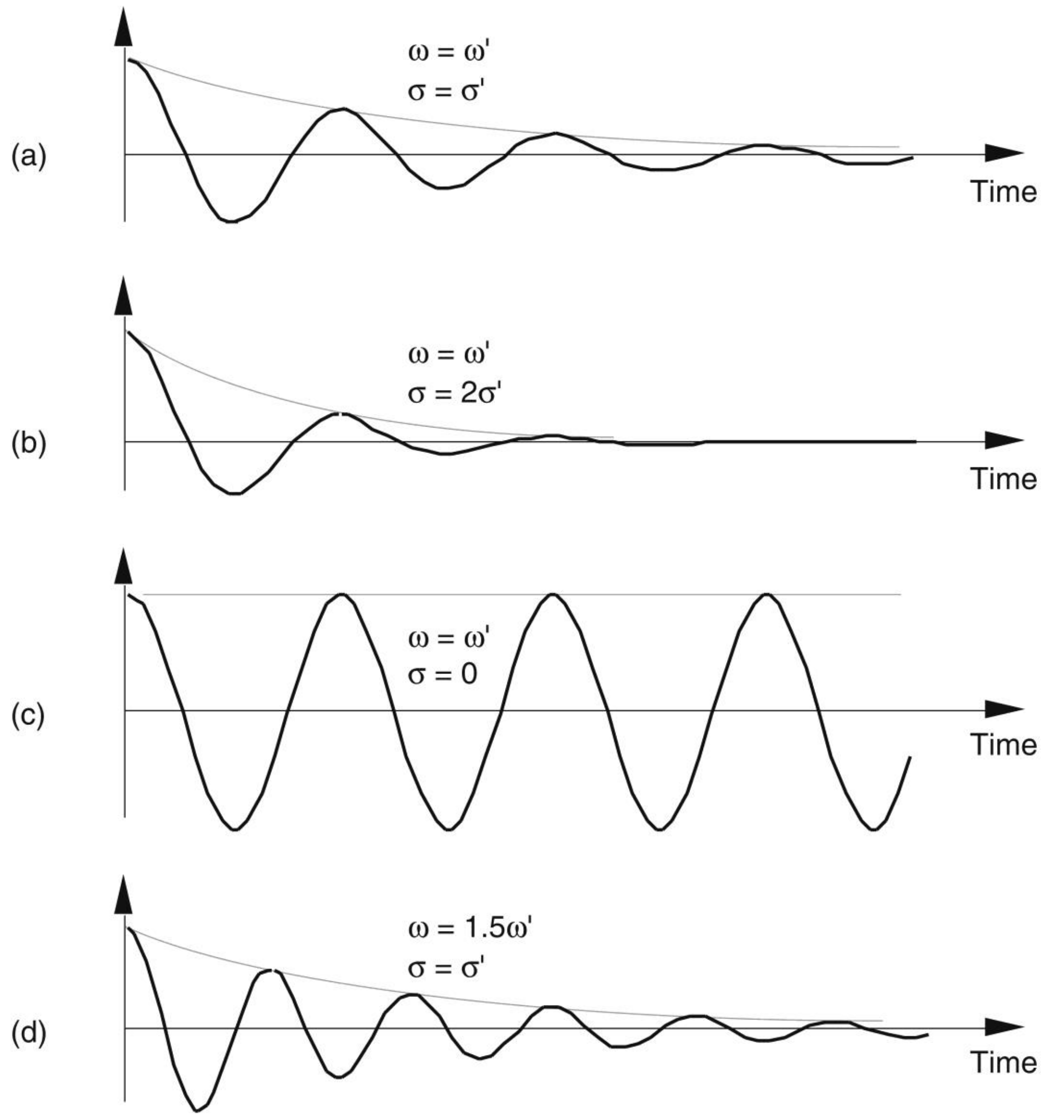


Figure 6–4 Real part (cosine) of various e^{-st} functions, where $s = \sigma + j\omega$, to be correlated with $f(t)$.

The Laplace Transform

- *s*-plane
  - If we associate each of different values of complex *s* variable with a point on a complex plane, called *s*-plane, we could plot real part of $F(s)$ correlation as a surface above (or below) that *s*-plane and generate a second plot of imaginary part of $F(s)$ correlation as a surface above (or below) *s*-plane
  - We can also graph magnitude $|F(s)|$ as a function of *s*

The Laplace Transform

$$a_2 \frac{d^2 y(t)}{dt^2} + a_1 \frac{dy(t)}{dt} + a_0 y(t) = b_1 \frac{dx(t)}{dt} + b_0 x(t)$$

Figure 6–5 System described by Eq. (6–6). The system's input and output are the continuous-time functions $x(t)$ and $y(t)$ respectively.

We'll use Laplace transform toward our goal of figuring out what the $y(t)$ output will be for any given $x(t)$ input

The Laplace Transform

$$\frac{d(e^{st})}{dt} = se^{st}, \frac{d^2(e^{st})}{dt^2} = s^2 e^{st}, \frac{d^3(e^{st})}{dt^3} = s^3 e^{st}, \ldots, \frac{d^n(e^{st})}{dt^n} = s^n e^{st}$$

$$a_2 \frac{d^2 y(t)}{dt^2} + a_1 \frac{dy(t)}{dt} + a_0 y(t) = b_1 \frac{dx(t)}{dt} + b_0 x(t)$$

$$\xrightarrow{\text{If we let } x(t) \text{ and } y(t) \text{ be functions of } e^{st},\ x(e^{st}) \text{ and } y(e^{st})}$$

$$a_2 s^2 y(e^{st}) + a_1 s y(e^{st}) + a_0 y(e^{st}) = b_1 s x(e^{st}) + b_0 x(e^{st})$$

$$\text{or } (a_2 s^2 + a_1 s + a_0) y(e^{st}) = (b_1 s + b_0) x(e^{st})$$

$$\text{transfer function } H(s) = \frac{Y(s)}{X(s)} = \frac{y(e^{st})}{x(e^{st})} = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0}$$

$$Y(s) = X(s) \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0} = X(s) H(s)$$

The Laplace Transform

$$\text{transfer function } H(s) = \frac{Y(s)}{X(s)} = \frac{y(e^{st})}{x(e^{st})} = \frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0}$$

$$Y(s) = X(s)\frac{b_1 s + b_0}{a_2 s^2 + a_1 s + a_0} = X(s)H(s)$$

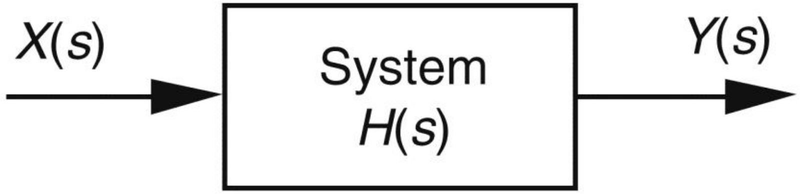


Figure 6–6 Linear system described by Eqs. (6–10) and (6–11). The system's input is the Laplace function *X*(*s*), its output is the Laplace function *Y*(*s*), and the system transfer function is *H*(*s*).

The Laplace Transform

- Laplace analysis technique is based on system's $x(t)$ input being some function of e^{st}, or $x(e^{st})$
  - All practical $x(t)$ input functions can be represented with complex exponentials, e.g.,
    - A constant, $c = ce^{0t}$
    - Sinusoids, $\sin(\omega t) = (e^{j\omega t} - e^{-j\omega t})/2j$ or $\cos(\omega t) = (e^{j\omega t} + e^{-j\omega t})/2$
    - A monotonic exponential, e^{at}
    - An exponentially varying sinusoid, $e^{-at}\cos(\omega t)$

The Laplace Transform

- System's transfer function $H(s)$
  - If we know $H(s)$, we can take Laplace transform of any $x(t)$ input to determine $X(s)$, multiply that $X(s)$ by $H(s)$ to get $Y(s)$, and then inverse Laplace transform $Y(s)$ to yield time-domain expression for the output $y(t)$
    - Not needed because it's $H(s)$ in which we're interested
  - Being able to express $H(s)$ mathematically or graph the surface $|H(s)|$ as a function of s will tell us two important properties we need to know about system:
    - Is system stable
    - And if so, what is its frequency response

The Laplace Transform

- Poles and zeros on *s*-plane and stability
  - A system is stable if, given any bounded input, the output will always be bounded
  - Instability would result in a filter output that's not at all representative of filter input—a situation we'd like to avoid if we can
  - Example

$$\text{System's Laplace transfer function} = H_1(s) = \frac{Y(s)}{X(s)} = \frac{b_0}{a_1 s + a_0} = \frac{b_0 / a_1}{s + a_0 / a_1}$$

- If $s = -a_0/a_1$, denominator equals zero and $H_1(s)$ would have an infinite magnitude
- This $s = -a_0/a_1$ point on *s*-plane is called a *pole*

The Laplace Transform

If the system described by H_1 were at rest and we disturbed it with an impulse like $x(t)$ input at time $t = 0$, its continuous time-domain $y(t)$ output would be the damped exponential curve shown in (b)

the pole is located exactly on negative portion of real σ axis

distance of pole from $\sigma = 0$ axis, a_0/a_1 for our $H_1(s)$, gives the decay rate of $y(t)$ impulse response

$H_1(s)$ is stable because its $y(t)$ output approaches zero as time passes

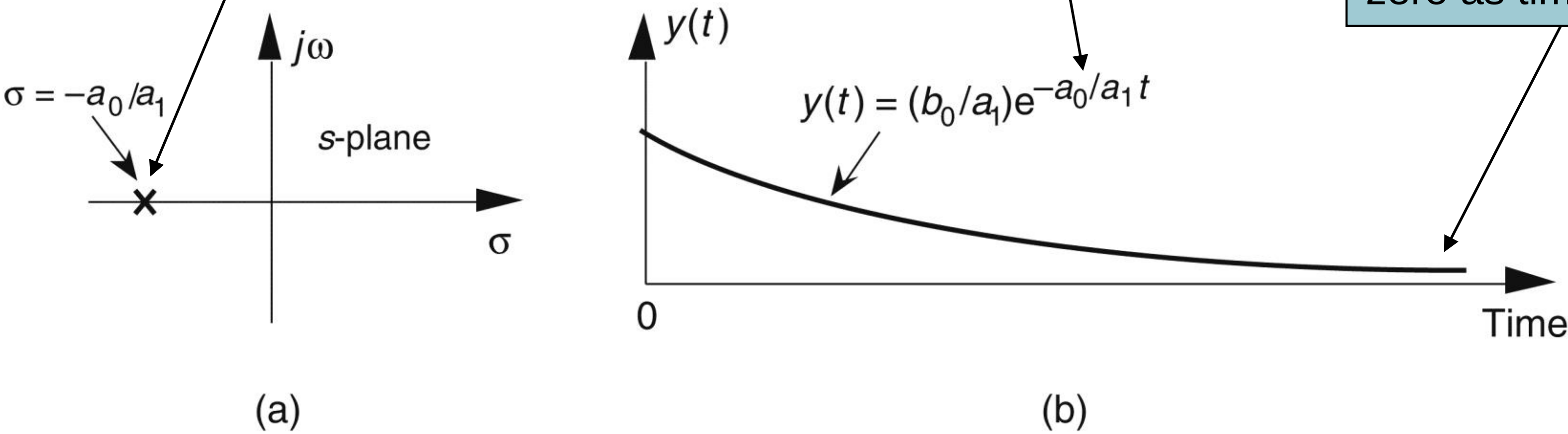


Figure 6–7 Descriptions of $H_1(s)$: (a) pole located at $s = \sigma + j\omega = -a_0/a_1 + j0$ on the s-plane; (b) time-domain $y(t)$ impulse response of the system.

The Laplace Transform

intersection of vertical σ = 0 plane ($j\omega$ axis plane) and $|H_1(s)|$ surface, gives us frequency magnitude response $|H_1(\omega)|$ of system

Laplace transform is a more general case of Fourier transform because if σ = 0, then $s = j\omega$

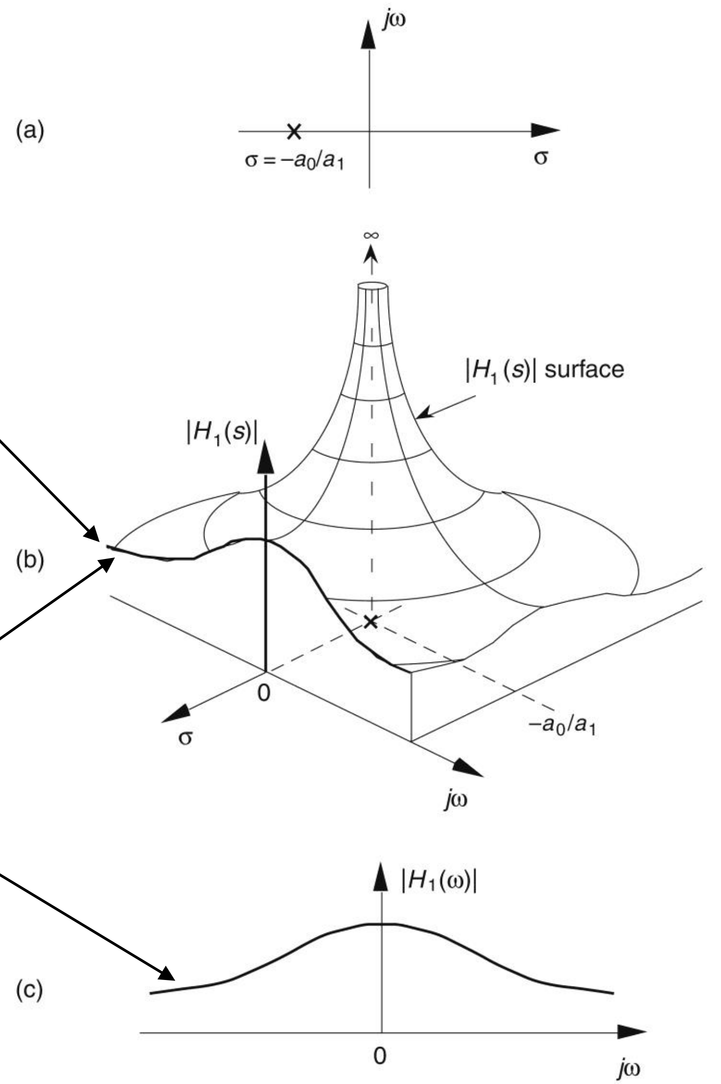


Figure 6-8 Further depictions of $H_1(s)$: (a) pole located at $\sigma = -a_0/a_1$ on the *s*-plane; (b) $|H_1(s)|$ surface; (c) curve showing the intersection of the $|H_1(s)|$ surface and the vertical σ = 0 plane. This is the conventional depiction of the $|H_1(\omega)|$ frequency magnitude response.

The Laplace Transform

- Example: 2nd-order transfer function $H_2(s)$

$$H_2(s) = \frac{Y(s)}{X(s)} = \frac{b_1 s}{a_2 s^2 + a_1 s + a_0} = \frac{(b_1/a_2)s}{s^2 + (a_1/a_2)s + a_0/a_2}$$

$$H_2(s) = \frac{As}{(s+p)(s+p^*)}$$

where $A = b_1/a_2,\ p = p_{\text{real}} + jp_{\text{imag}},\ p^* = p_{\text{real}} - jp_{\text{imag}}$

$$f(s) = as^2 + bs + c = \left(s + \frac{b}{2a} + \sqrt{\frac{b^2 - 4ac}{4a^2}}\right)\cdot\left(s + \frac{b}{2a} - \sqrt{\frac{b^2 - 4ac}{4a^2}}\right)$$

  - *Order* of transfer function is the largest exponential order of *s* in either numerator or denominator
  - If *s* is equal to $-p$ or $-p^*$, one of polynomial roots in the denominator will equal zero, and $H_2(s)$ will have an infinite magnitude → two complex poles

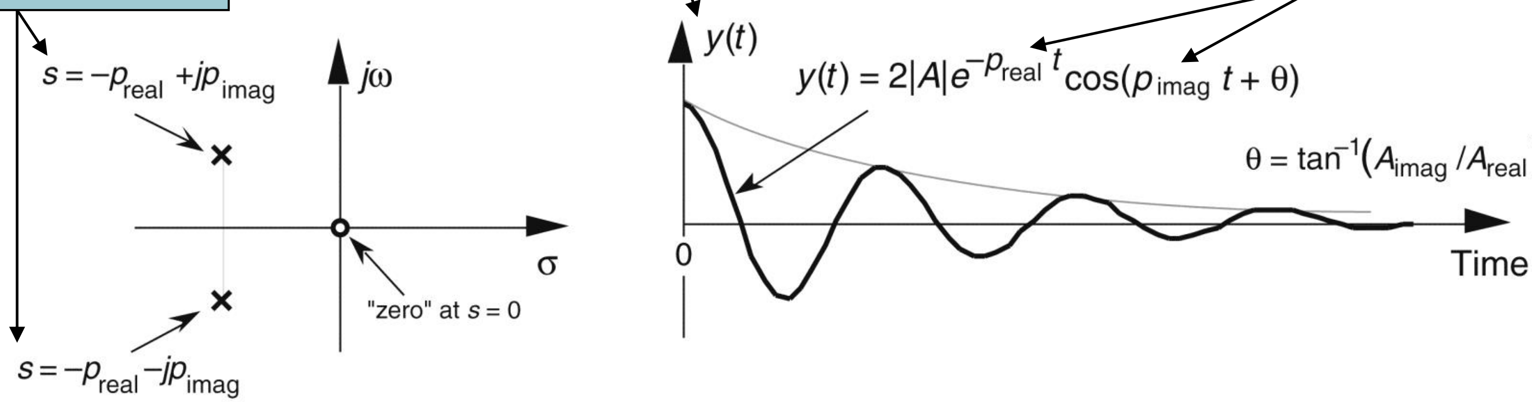
The Laplace Transform

If H_2 system were at rest and we disturbed it with an impulse-like $x(t)$ input at $t = 0$, its continuous time-domain $y(t)$ output would be the damped sinusoidal curve shown in (b)

the two complex poles are located off the negative portion of real σ axis

$H_2(s)$ is stable because its oscillating $y(t)$ output approaches zero as time increases

distance of poles from σ = 0 axis ($-p_{real}$) gives decay rate of sinusoidal $y(t)$ impulse response. Likewise, distance of poles from $j\omega = 0$ axis ($\pm p_{imag}$) gives frequency of sinusoidal $y(t)$ impulse response



(a) (b)

Figure 6-9 Descriptions of $H_2(s)$: (a) poles located at $s = p_{real} \pm jp_{imag}$ on the s-plane; (b) time-domain $y(t)$ impulse response of the system.

The Laplace Transform

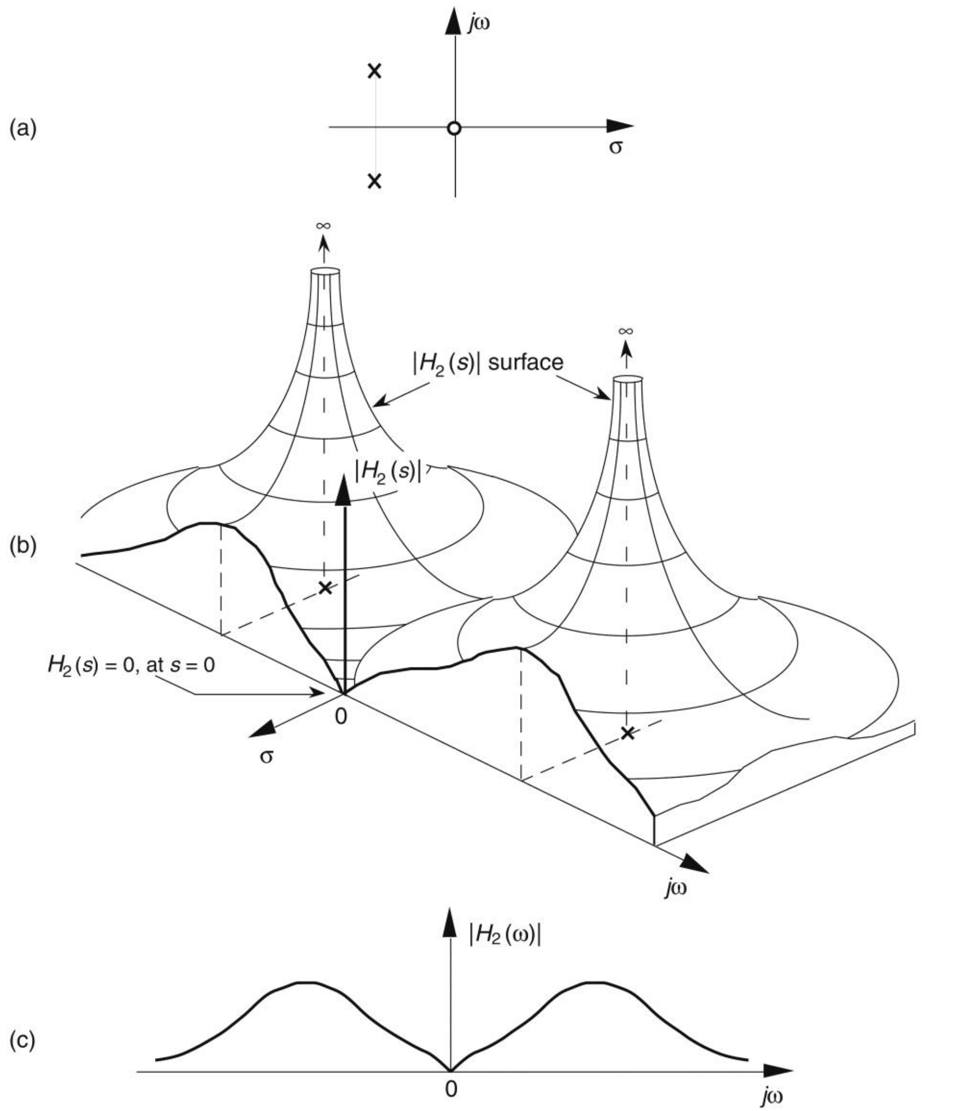


Figure 6–10 Further depictions of $H_2(s)$: (a) poles and zero locations on the s-plane; (b) $|H_2(s)|$ surface; (c) $|H_2(\omega)|$ frequency magnitude response curve.

The Laplace Transform

stable systems: when disturbed from their at-rest condition, they respond and, at some later time, return to that initial condition

1/*s* transfer function

instability: the poles lie to the right of *j*ω axis. When disturbed from their initial at-rest condition by an impulse input, their outputs grow without bound

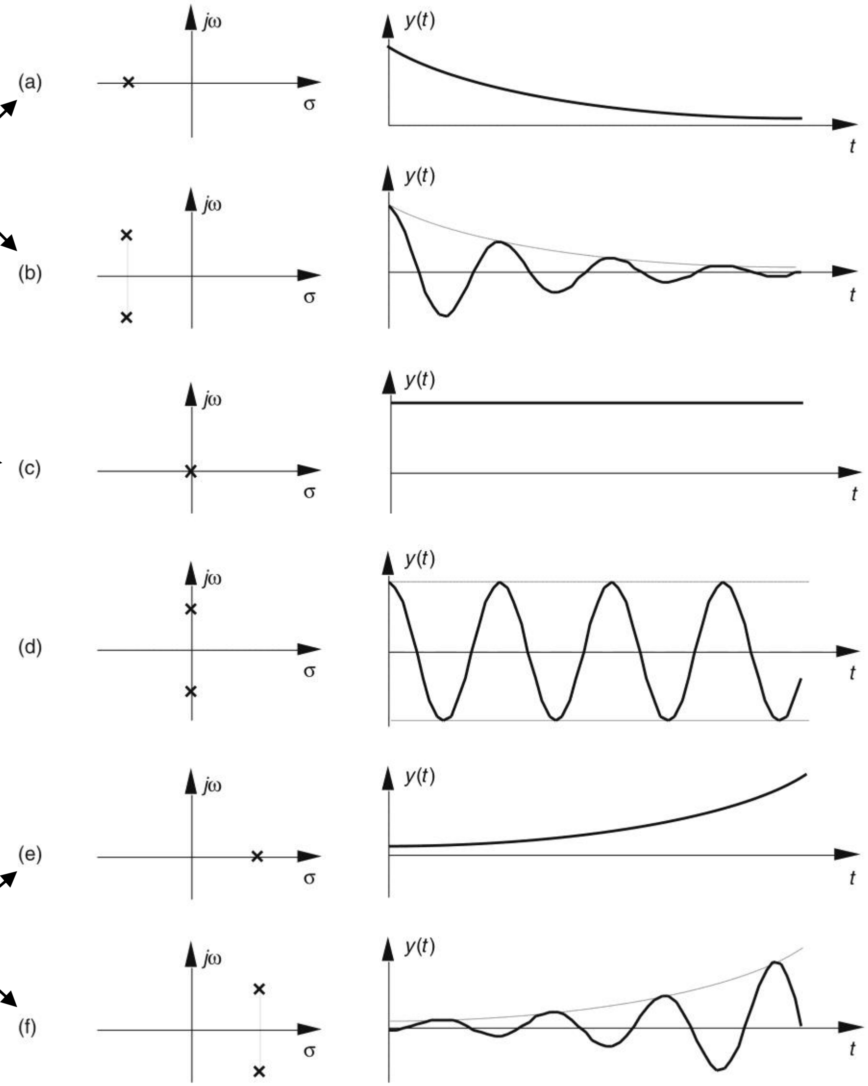


Figure 6-11 Various $H(s)$ pole locations and their time-domain impulse responses: (a) single pole at $\sigma < 0$; (b) conjugate poles at $\sigma < 0$; (c) single pole located at $\sigma = 0$; (d) conjugate poles located at $\sigma = 0$; (e) single pole at $\sigma > 0$; (f) conjugate poles at $\sigma > 0$.

conditional stability: if an $H(s)$ transfer function has conjugate poles located exactly on *j*ω axis ($\sigma = 0$), the system will go into oscillation when disturbed from its initial condition

The Laplace Transform

for a system to be stable, all of its transfer-function poles must lie on the left half of *s*-plane

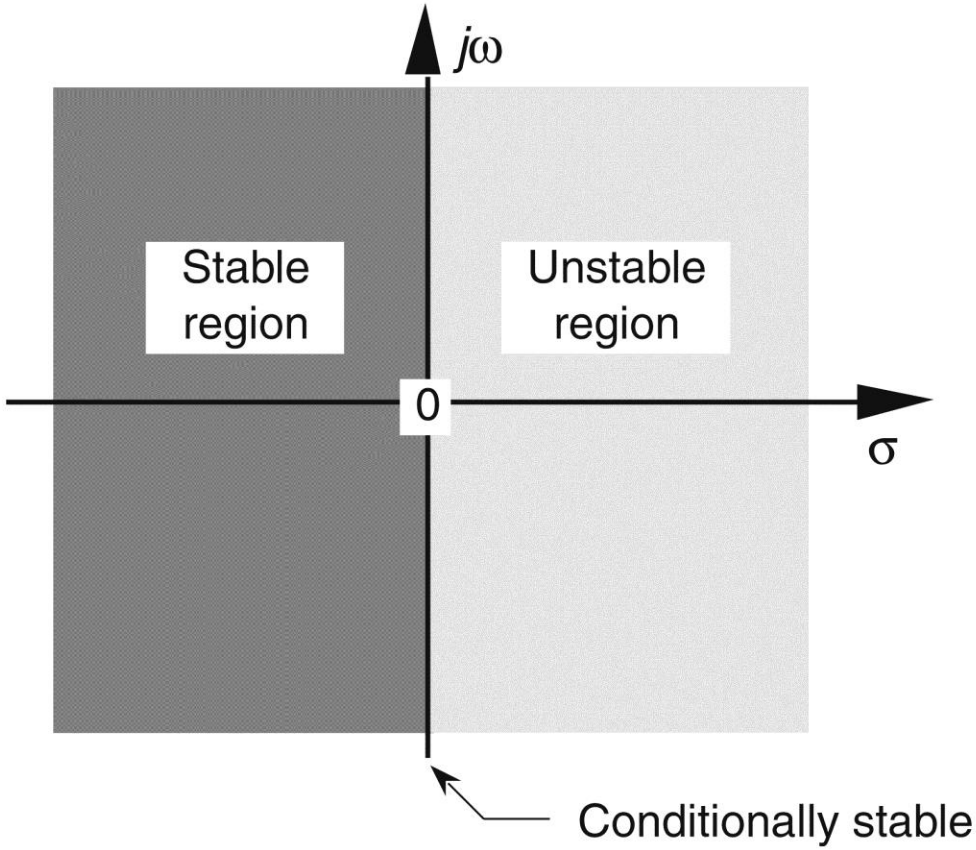


Figure 6–12 The Laplace *s*–*p*lane showing the regions of stability and instability for pole locations for linear continuous systems.

The *z*-Transform

- *z*-transform
  - Discrete-time cousin of continuous Laplace transform
    - While Laplace transform is used to simplify analysis of continuous differential equations, *z*-transform facilitates analysis of discrete difference equations
  - *z*-transform is performed on a discrete *h*(*n*) sequence, converting that sequence into a continuous function *H*(*z*) of the continuous complex variable *z*

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) z^{-n}$$

The *z*-Transform

- *z*-transform

$$H(z) = \sum_{n=-\infty}^{\infty} h(n) z^{-n}$$

$$\xrightarrow{z=re^{j\omega}} H(z) = H(re^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n)(re^{j\omega})^{-n} = \sum_{n=-\infty}^{\infty} h(n) r^{-n} (e^{-j\omega n})$$

  - This Equation can be interpreted as Fourier transform of product of original sequence $h(n)$ and exponential sequence r^{-n}
  - When $r = 1$, it simplifies to Fourier transform
  - Thus on *z*-plane, the contour of $H(z)$ surface for those values where $|z| = 1$ is Fourier transform of $h(n)$
  - If $h(n)$ represents a filter impulse response sequence, evaluating $H(z)$ transfer function for $|z| = 1$ yields frequency response of filter

The *z*-Transform

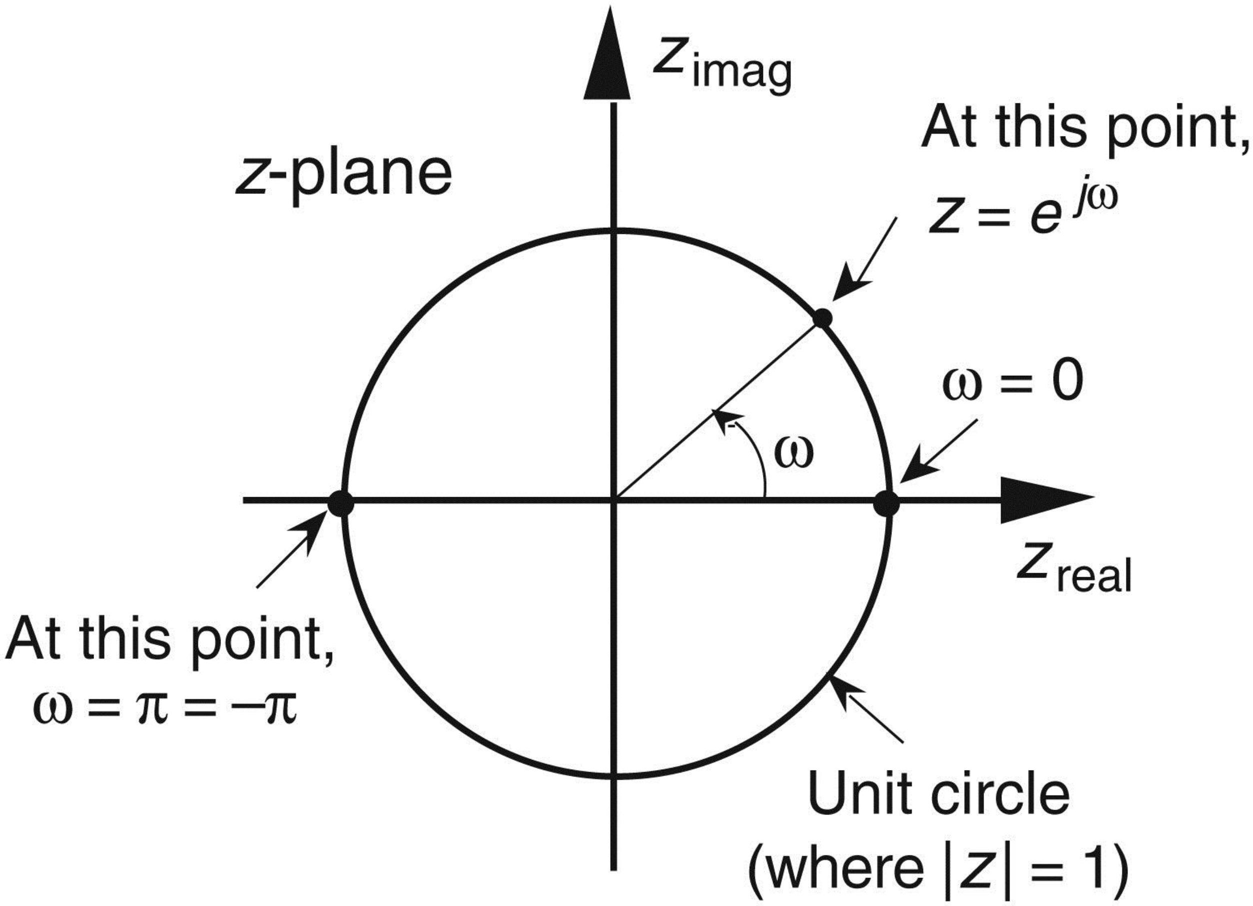


Figure 6–13 Unit circle on the complex *z*–plane.

The *z*-Transform

*j*ω frequency axis on continuous Laplace *s*-plane is linear and ranges from − ∞ to + ∞ radians/second. The ω frequency axis on complex *z*-plane, however, spans only the range from −π to +π radians → *z*-plane frequency axis is equivalent to coiling *s*-plane's jω axis about the unit circle on *z*-plane

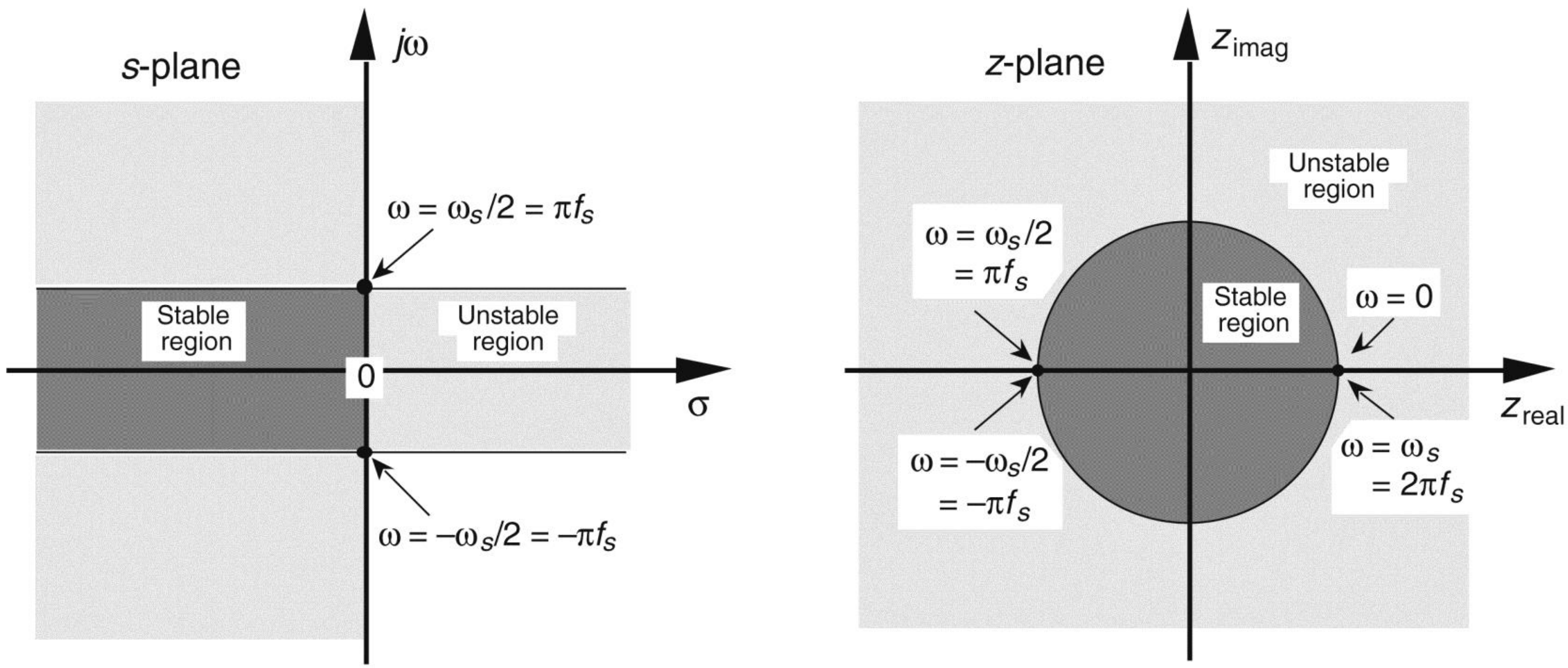


Figure 6-14 Mapping of the Laplace *s*-plane to the *z*-plane. All frequency values are in radians/second.

The *z*-Transform

- Poles, zeros, and digital filter stability
  - Region of filter stability is mapped to the inside of unit circle on *z*-plane
  - Given $H(z)$ transfer function of a digital filter, we can examine that function's pole locations to determine filter stability
  - If all poles are located inside unit circle, filter will be stable

The *z*-Transform

- Example
  - If a causal filter's $H(z)$ transfer function has a single pole at location p on z-plane, its transfer function can be represented by

$$H(z) = \frac{1}{1 - pz^{-1}}$$

  - Filter's time-domain impulse response sequence

$$h(n) = p^n \cdot u(n)$$

    - $u(n)$ represents a unit step (all ones) sequence beginning at time $n = 0$
    - When $|p| < 1$, $h(n)$ impulse response sequence is unconditionally bounded in amplitude
      - $|p| < 1$ means that pole must lie inside z-plane's unit circle

The z-Transform

point $z = -1$ corresponds to π radians (or πf_s radians/second = $f_s/2$ Hz) → $\omega_o = \pi/4$ means that $f_o = f_s/8$ and our $y(n)$ will have eight samples per cycle of f_o

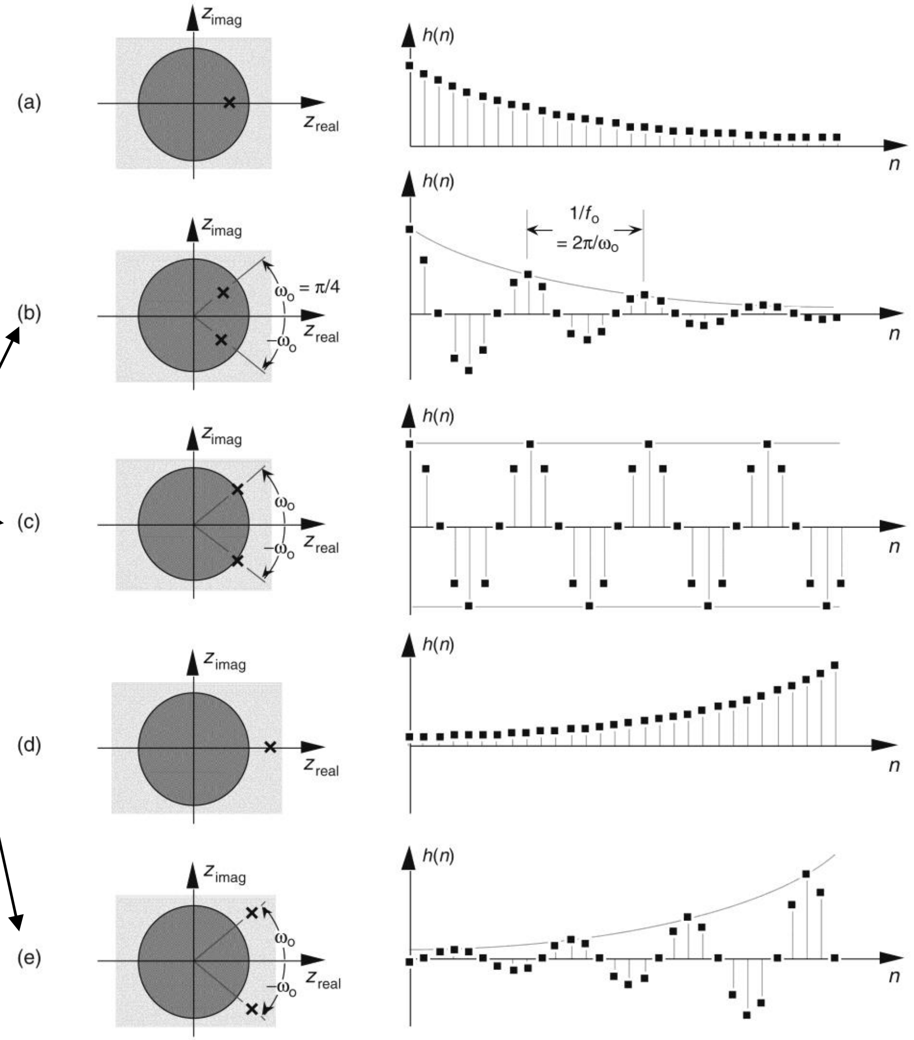


Figure 6-15 Various $H(z)$ pole locations and their discrete time-domain impulse responses: (a) single pole inside the unit circle; (b) conjugate poles located inside the unit circle; (c) conjugate poles located on the unit circle; (d) single pole outside the unit circle; (e) conjugate poles located outside the unit circle.

Using *z*-Transform to Analyze IIR Filters

- Representing delay operation
  - Assume we have a sequence $x(n)$ whose z-transform is $X(z)$ and a sequence $y(n) = x(n-1)$ whose z-transform is $Y(z)$

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n) z^{-n} = \sum_{n=-\infty}^{\infty} x(n-1) z^{-n}$$

$$\xrightarrow{\text{if we let } k=n-1} Y(z) = \sum_{k=-\infty}^{\infty} x(k) z^{-(k+1)} = \sum_{k=-\infty}^{\infty} x(k) z^{-k} z^{-1}$$

$$= z^{-1} \sum_{k=-\infty}^{\infty} x(k) z^{(-k)} = z^{-1}[X(z)]$$

  - Thus, effect of a single unit of time delay is to multiply z-transform of undelayed sequence by z^{-1}

Using *z*-Transform to Analyze IIR Filters

Because a delay of one sample is equivalent to factor z^{-1}, the unit time delay symbol is usually indicated by z^{-1} operator

$X(z)z^{-k}$ is *z*-transform of $x(n)$ delayed by k samples. So a transfer function of z^{-k} is equivalent to a delay of kt_s seconds from the instant when $t = 0$, where $t_s = 1/f_s$

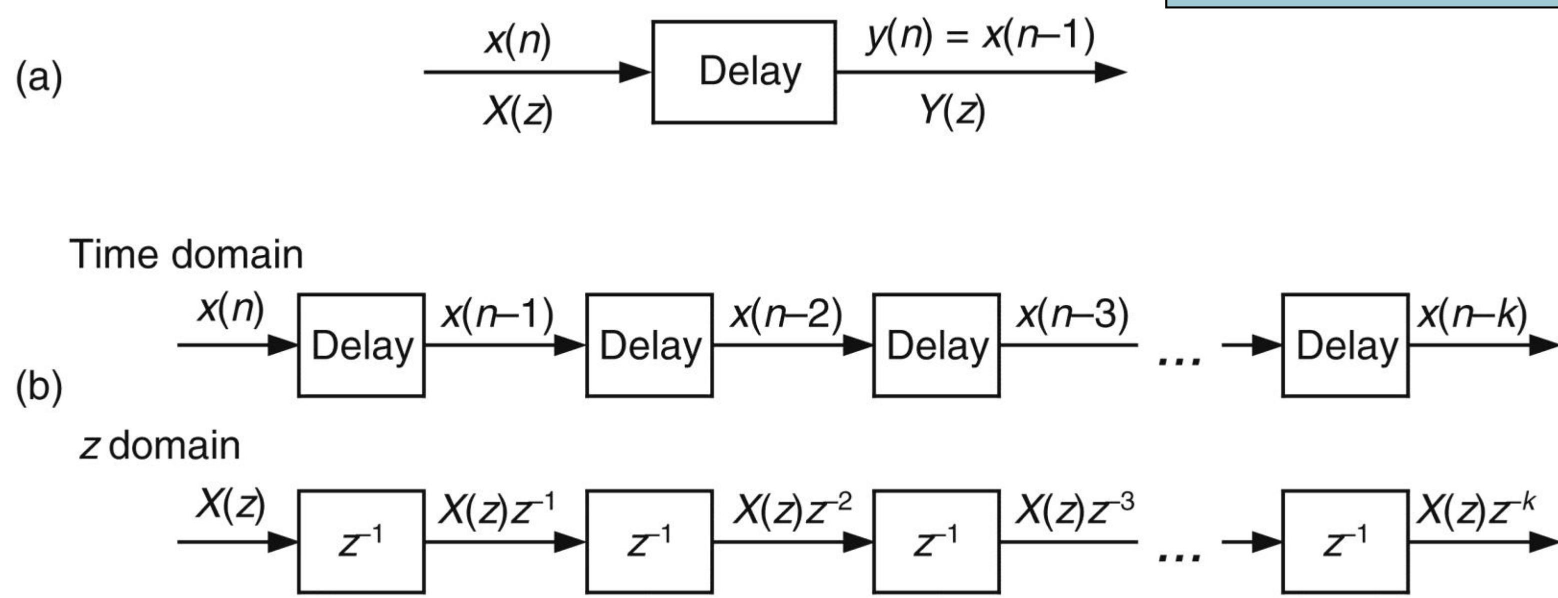


Figure 6–16 Time- and *z*-domain delay element relationships: (a) single delay; (b) multiple delays.

Using *z*-Transform to Analyze IIR Filters

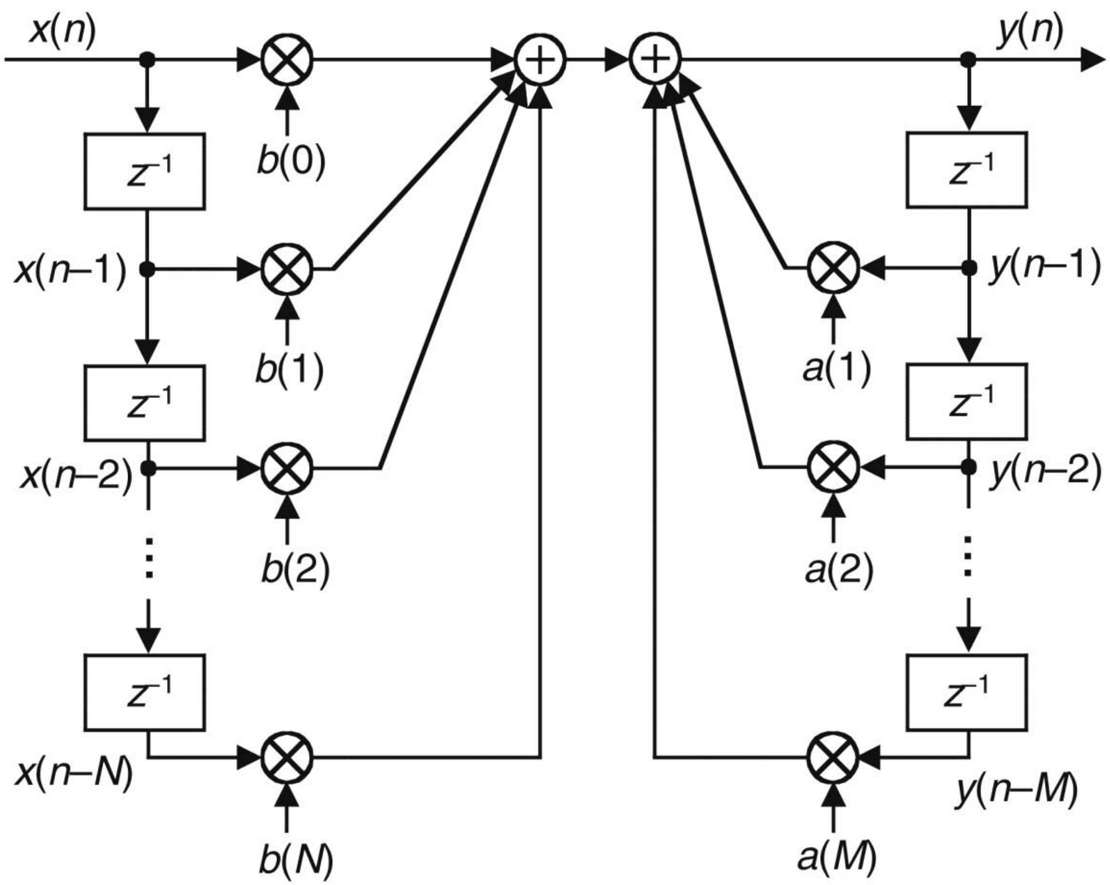


Figure 6–17 General (Direct Form I) structure of an *M*th-order IIR filter, having *N* feedforward stages and *M* feedback stages, with the z^{-1} operator indicating a unit time delay.

Using *z*-Transform to Analyze IIR Filters

- Fig. 6-17
  - Is a general *M*th-order IIR filter
  - This IIR filter structure is called *Direct Form I*

$$y(n) = b(0)x(n) + b(1)x(n-1) + b(2)x(n-2) + \ldots + b(N)x(n-N)$$
$$+ a(1)y(n-1) + a(2)y(n-2) + \ldots + a(M)y(n-M)$$

$$Y(z) = b(0)X(z) + b(1)X(z)z^{-1} + b(2)X(z)z^{-2} + \ldots + b(N)X(z)z^{-N}$$
$$+ a(1)Y(z)z^{-1} + a(2)Y(z)z^{-2} + \ldots + a(M)Y(z)z^{-M}$$

$$Y(z) = X(z)\sum_{k=0}^{N} b(k)z^{-k} + Y(z)\sum_{k=1}^{M} a(k)z^{-k}$$

order of *H(z)* and order of filter = the largest exponential order of *z* in either numerator or denominator

$$Y(z)\left[1 - \sum_{k=1}^{M} a(k)z^{-k}\right] = X(z)\sum_{k=0}^{N} b(k)z^{-k} \xrightarrow{\text{transfer function}} H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b(k)z^{-k}}{1 - \sum_{k=1}^{M} a(k)z^{-k}}$$

Using *z*-Transform to Analyze IIR Filters

- Two things to know about an IIR filter: its frequency response and stability
  - We can evaluate denominator of *H(z)* to determine positions of filter's poles on *z*-plane indicating filter's stability
  - From *H(z)* we develop an expression for IIR filter's frequency response
    - *H*(*z*) is a complex-valued surface above, or below, the *z*-plane
    - Intersection of *H*(*z*) surface and perimeter of a cylinder representing $z = e^{j\omega}$ unit circle is the filter's complex frequency response

Using *z*-Transform to Analyze IIR Filters

$$H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{k=0}^{N}b(k)z^{-k}}{1-\sum_{k=1}^{M}a(k)z^{-k}}\rightarrow H(\omega)=H(z)\Big|_{z=e^{j\omega}}=\frac{\sum_{k=0}^{N}b(k)e^{-jk\omega}}{1-\sum_{k=1}^{M}a(k)e^{-jk\omega}}$$

$$\xrightarrow{e^{-j\omega}=\cos(\omega)-j\sin(\omega)}H(\omega)=\frac{\sum_{k=0}^{N}b(k)\cos(k\omega)-j\sum_{k=0}^{N}b(k)\sin(k\omega)}{1-\sum_{k=1}^{M}a(k)\cos(k\omega)+j\sum_{k=1}^{M}a(k)\sin(k\omega)}$$

Using *z*-Transform to Analyze IIR Filters

(a)

(b)

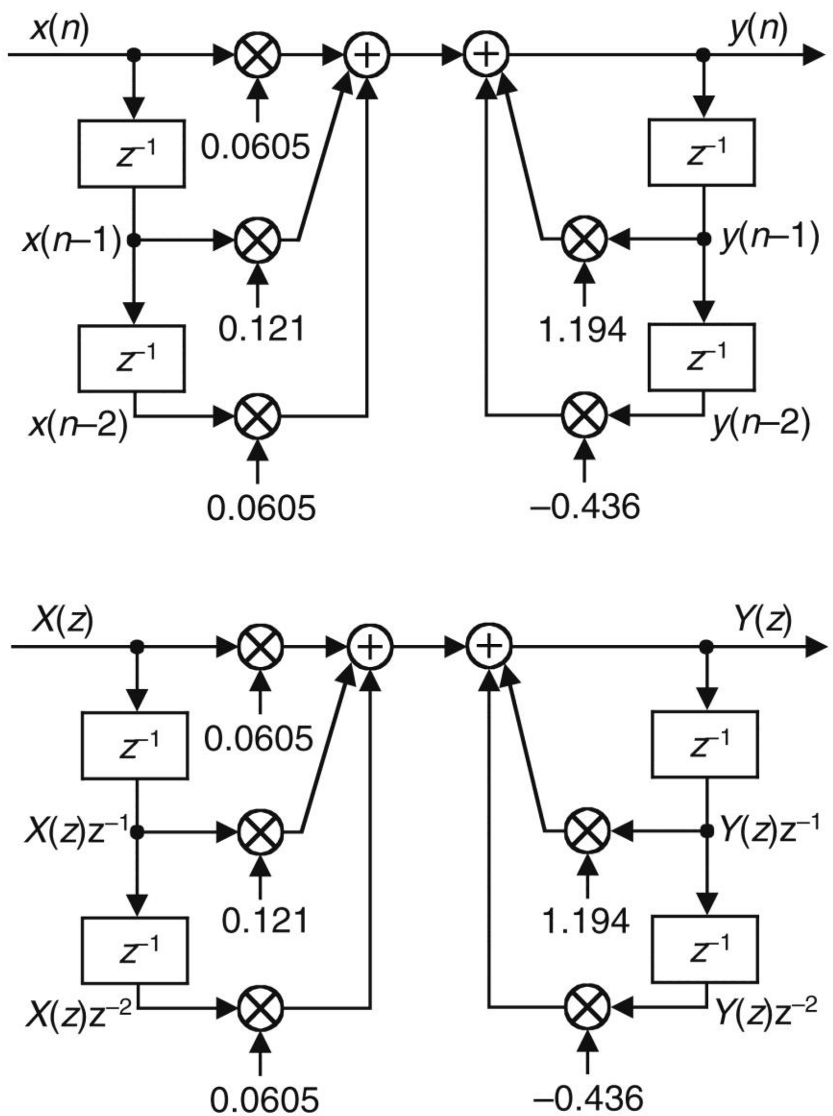


Figure 6–18 Second-order lowpass IIR filter example.

Using *z*-Transform to Analyze IIR Filters

- Fig. 6-18
  - (a) is a 2nd-order lowpass IIR filter whose positive cutoff frequency is ω = π/5 (f_s/10 Hz)

$$y(n) = 0.0605 \cdot x(n) + 0.121 \cdot x(n-1) + 0.0605 \cdot x(n-2)$$

$$+1.194 \cdot y(n-1) - 0.436 \cdot y(n-2)$$

$$Y(z) = 0.0605 \cdot X(z) + 0.121 \cdot X(z) z^{-1} + 0.0605 \cdot X(z) z^{-2}$$

$$+1.194 \cdot Y(z) z^{-1} - 0.436 \cdot Y(z) z^{-2}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{0.0605 \cdot z^{0} + 0.121 \cdot z^{-1} + 0.0605 \cdot z^{-2}}{1 - 1.194 \cdot z^{-1} + 0.436 \cdot z^{-2}}$$

$$H(\omega) = \frac{0.0605 \cdot e^{-j0\omega} + 0.121 \cdot e^{-j1\omega} + 0.0605 \cdot e^{-j2\omega}}{1 - 1.194 \cdot e^{-j1\omega} + 0.436 \cdot e^{-j2\omega}}$$

$$H(\omega) = \frac{0.0605 + 0.121 \cdot \cos(1\omega) + 0.0605 \cdot \cos(2\omega) - j[0.121 \cdot \sin(1\omega) + 0.0605 \cdot \sin(2\omega)]}{1 - 1.194 \cdot \cos(1\omega) + 0.436 \cdot \cos(2\omega) + j[1.194 \cdot \sin(1\omega) - 0.436 \cdot \sin(2\omega)]}$$

Using *z*-Transform to Analyze IIR Filters

although both filters require the same computational workload, five multiplications per filter output sample, lowpass IIR filter has the superior frequency magnitude response

phase nonlinearity is inherent in IIR filters

knowing that the filter's phase response is nonlinear, we should expect the impulse response to be asymmetrical

infinite impulse response: if we used infinite-precision arithmetic in our filter implementation, $h(k)$ impulse response would be infinite in duration

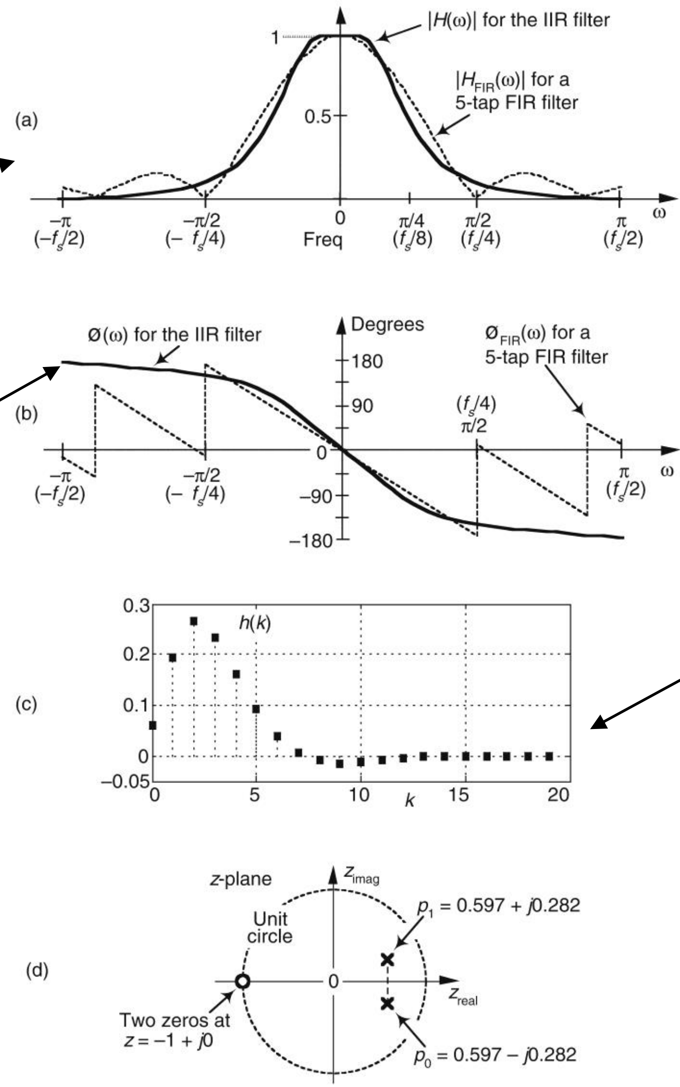


Figure 6-19 Performances of the example IIR filter (solid curves) in Figure 6-18 and a 5-tap FIR filter (dashed curves): (a) magnitude responses; (b) phase responses; (c) IIR filter impulse response; (d) IIR filter poles and zeros.

Using *z*-Transform to Analyze IIR Filters

- To determine our IIR filter's stability

$$H(z)=\frac{Y(z)}{X(z)}=\frac{0.0605\cdot z^{0}+0.121\cdot z^{-1}+0.0605\cdot z^{-2}}{1-1.194\cdot z^{-1}+0.436\cdot z^{-2}}$$

$$\xrightarrow{\text{multiply } H(z) \text{ by } z^2/z^2} H(z)=\frac{0.0605\cdot z^{2}+0.121\cdot z+0.0605}{z^{2}-1.194\cdot z+0.436}$$

$$H(z)=\frac{(z-z_0)(z-z_1)}{(z-p_0)(z-p_1)}=\frac{(z+1)(z+1)}{(z-0.597+j0.282)(z-0.597-j0.282)}$$

  - So when $z = p_0 = 0.597 - j0.282$, or when $z = p_1 = 0.597 + j0.282$, filter's $H(z)$ transfer function's denominator is zero and $|H(z)|$ is infinite
    - Because those pole locations are inside the unit circle (their magnitudes are less than one), our example IIR filter is unconditionally stable

Using *z*-Transform to Analyze IIR Filters

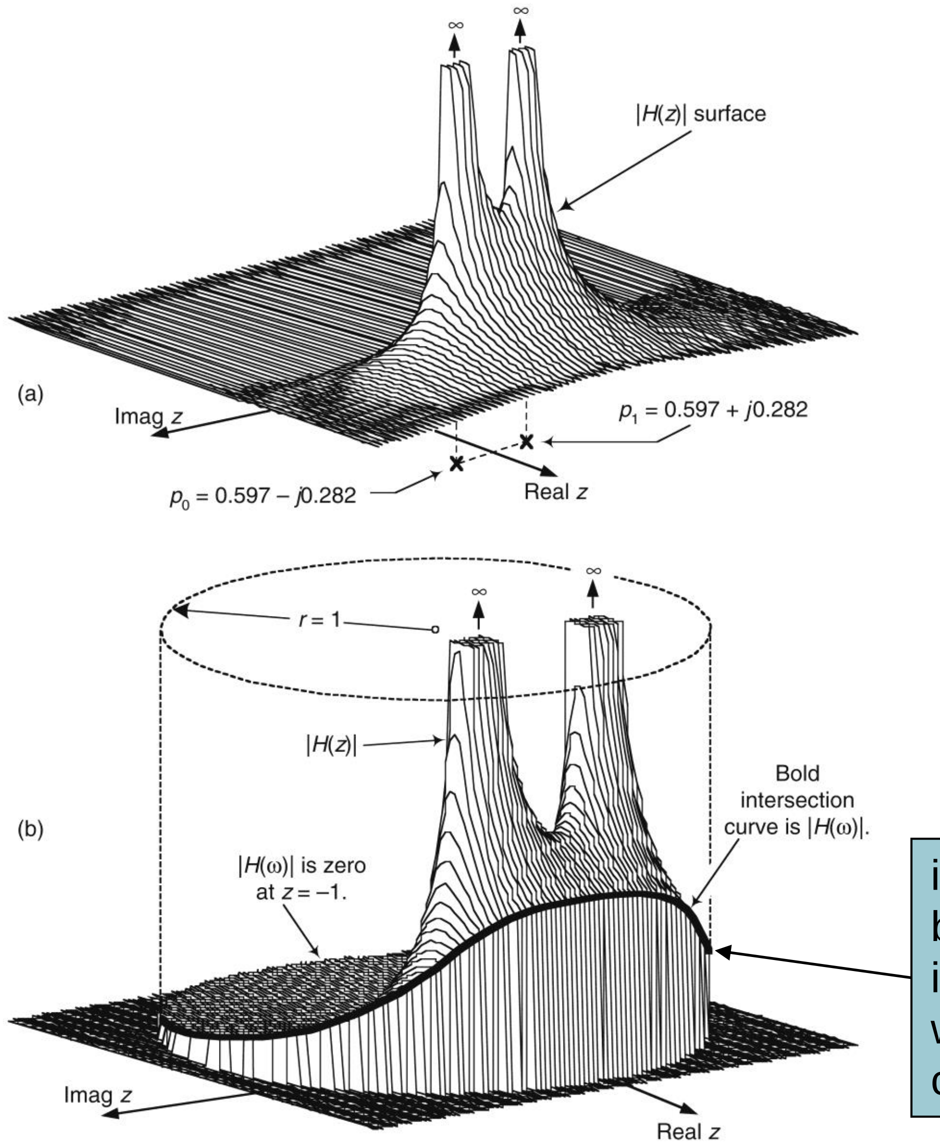


Figure 6–20 IIR filter's $|H(z)|$ surface: (a) pole locations; (b) frequency magnitude response.

if we were to unwrap the bold $|H(\omega)|$ curve and lay it on a flat surface, we would have the $|H(\omega)|$ curve in Fig. 6-19(a)

Using Poles and Zeros to Analyze IIR Filters

- IIR filter transfer function algebra
  - Several ways to write $H(z) = Y(z)/X(z)$ z-domain transfer function

useful because we can replace z with $e^{j\omega}$ to obtain an expression for frequency response of filter

$$H(z) = \frac{b(0) + b(1)z^{-1} + b(2)z^{-2} + b(3)z^{-3} + b(4)z^{-4}}{1 + a(1)z^{-1} + a(2)z^{-2} + a(3)z^{-3} + a(4)z^{-4}}$$

$$\xrightarrow[\text{(polynomial form)}]{\text{multiplying by } z^4/z^4} H(z) = \frac{b(0)z^4 + b(1)z^3 + b(2)z^2 + b(3)z + b(4)}{z^4 + a(1)z^3 + a(2)z^2 + a(3)z + a(4)}$$

$$\xrightarrow{\text{factored form}} H(z) = \frac{(z - z_0)(z - z_1)(z - z_2)(z - z_3)}{(z - p_0)(z - p_1)(z - p_2)(z - p_3)}$$

necessary so we can factor (find roots of) polynomials to obtain values (locations) of numerator zeros and denominator poles

Using Poles and Zeros to Analyze IIR Filters

- Using poles/zeros to obtain transfer functions
  - We can analyze an IIR filter's frequency-domain performance based solely on poles and zeros
  - Given z_k zeros and p_k poles, we can write the factored form of filter's transfer function as

$$H(z) = \frac{G_1(z-z_0)(z-z_1)(z-z_2)(z-z_3)(z-z_4)\ldots}{G_2(z-p_0)(z-p_1)(z-p_2)(z-p_3)(z-p_4)\ldots}$$

$$= \frac{G(z-z_0)(z-z_1)(z-z_2)(z-z_3)(z-z_4)\ldots}{(z-p_0)(z-p_1)(z-p_2)(z-p_3)(z-p_4)\ldots}$$

    - $G = G_1/G_2$ is an arbitrary gain constant
    - Filter zeros are associated with decreased frequency magnitude response, and filter poles are associated with increased frequency magnitude response

Using Poles and Zeros to Analyze IIR Filters

if a filter has no *z*-plane zeros, and one pole at p_0 = 0.8, we can write its transfer function as $H_1(z)=G/(z-0.8)$

if a filter has a zero at z_0 = 1, and a pole at p_0 = −0.8, we write its transfer function as

$$H_2(z) = \frac{G(z-1)}{z-(-0.8)} = \frac{Gz-G}{z+0.8}$$

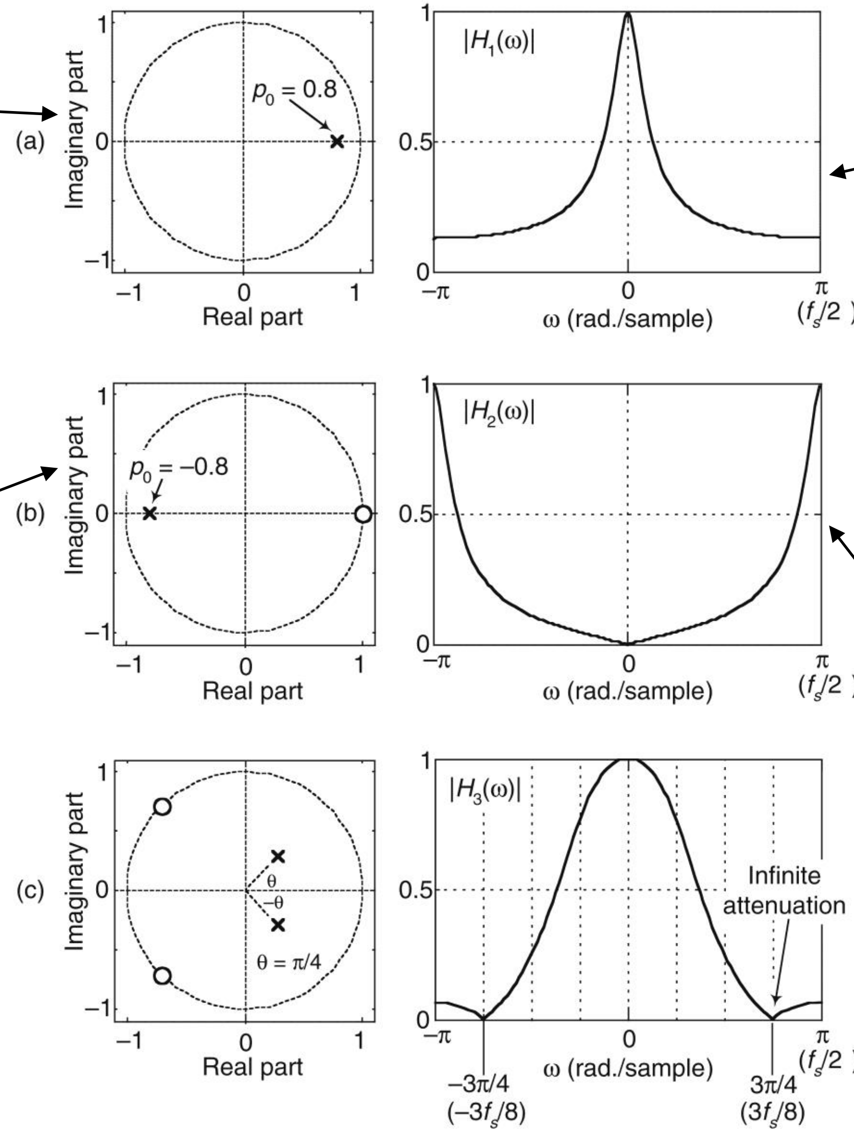


Figure 6-21 IIR filter poles/zeros and normalized frequency magnitude responses.

$|H_1(\omega)|$ is normalized.
P_0 is closest to ω = 0 radians/sample (*z* = 1) on unit circle → lowpass filter.
$|p_0| < 1$ → filter is unconditionally stable

pole is closest to ω = π radians/sample (*z* = −1) on unit circle → highpass filter.
the zero located at *z* = 1 causes filter to have infinite attenuation at ω = 0 radians/sample (zero Hz)

Using Poles and Zeros to Analyze IIR Filters

- Fig. 6-21(c)
  - Consider a filter having two complex conjugate zeros at −0.707 ± *j*0.707, as well as two complex conjugate poles at 0.283 ± *j*0.283

$$H_3(z) = \frac{G[z-(-0.707+j0.707)]\cdot[z-(-0.707-j0.707)]}{[z-(0.283+j0.283)]\cdot[z-(0.283-j0.283)]}$$

$$= \frac{G(z+0.707-j0.707)\cdot(z+0.707+j0.707)}{(z-0.283-j0.283)\cdot(z-0.283+j0.283)}$$

    - The two poles on the right side of *z*-plane make this a lowpass filter having a wider passband than $H_1(z)$
    - Two zeros are on unit circle at angles of ω = ±3π/4 radians, causing filter to have infinite attenuation at frequencies ω = ±3π/4 radians/sample (±3f_s/8 Hz)

Using Poles and Zeros to Analyze IIR Filters

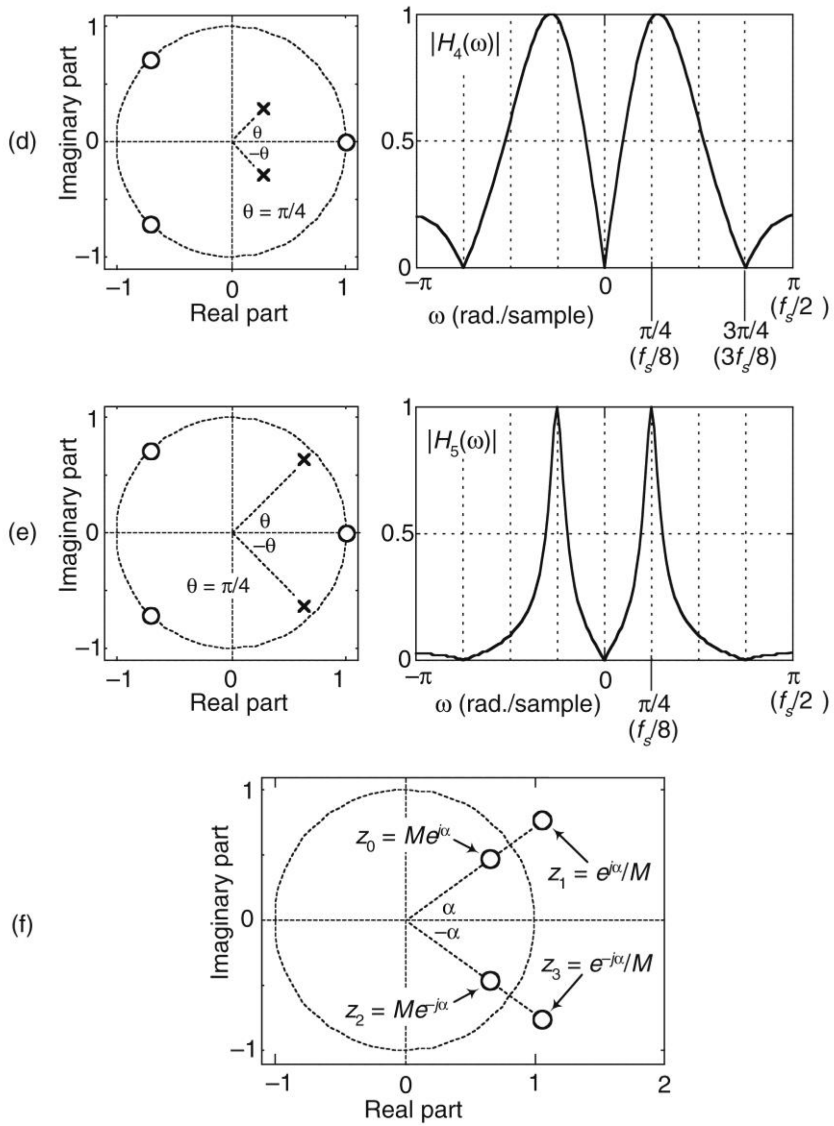


Figure 6–21 (continued) IIR filter poles/zeros and normalized frequency magnitude responses.

Using Poles and Zeros to Analyze IIR Filters

- Fig. 6-21(d)
  - If we add a z-plane zero at $z = 1$ to $H_3(z)$

$$H_4(z) = \frac{G(z-1)\cdot(z+0.707-j0.707)\cdot(z+0.707+j0.707)}{(z-0.283-j0.283)\cdot(z-0.283+j0.283)}$$

  - The zero at $z = 1$ yields infinite attenuation at $\omega = 0$ radians/sample (zero Hz), creating a bandpass filter
  - Because p_0 and p_1 poles are oriented at angles of $\theta = \pm\pi/4$ radians, filter's passbands are centered in the vicinity of frequencies $\omega = \pm\pi/4$ radians/sample ($\pm f_s/8$ Hz)

Using Poles and Zeros to Analyze IIR Filters

- Fig. 6-21(e)
  - If we increase magnitude of $H_4(z)$ filter's poles, making them equal to $0.636 \pm j0.636$, we position the conjugate poles much closer to unit circle

$$H_5(z) = \frac{G(z-1)\cdot(z+0.707-j0.707)\cdot(z+0.707+j0.707)}{(z-0.636-j0.636)\cdot(z-0.636+j0.636)}$$

  - Poles near unit circle now have a much more profound effect on filter's magnitude response
    - The poles' infinite gains cause $H_5(z)$ passbands to be very narrow (sharp)
  - When a pole is close to unit circle, center frequency of its associated passband can be accurately estimated to be equal to pole's angle

Using Poles and Zeros to Analyze IIR Filters

- Fig. 6-21(f)
  - Consider an FIR filter—a digital filter whose $H(z)$ transfer function denominator is unity
    - For an FIR filter to have linear phase, each z-plane zero located at $z = z_0 = Me^{j\alpha}$, where $M \neq 1$, must be accompanied by a zero having an angle of $-\alpha$ and a magnitude of $1/M$
      - z_0 is accompanied by z_3
    - If FIR filter's transfer function polynomial has real-valued b_k coefficients, then a z_0 zero not on the z-plane's real axis will be accompanied by a complex conjugate zero at $z = z_2$
    - Likewise, for FIR filter to have linear phase, z_2 zero must be accompanied by z_1 zero
      - z_1 and z_3 zeros are complex conjugates of each other

Using Poles and Zeros to Analyze IIR Filters

- *z*-plane pole/zero properties
  - Filter poles are associated with increased frequency magnitude response (gain)
  - Filter zeros are associated with decreased frequency magnitude response (attenuation)
  - To be unconditionally stable, all filter poles must reside inside the unit circle
  - Filter zeros do not affect filter stability
  - The closer a pole (zero) is to unit circle, the stronger will be its effect on filter’s gain (attenuation) at the frequency associated with the pole’s (zero’s) angle

Using Poles and Zeros to Analyze IIR Filters

- *z*-plane pole/zero properties
  - A pole (zero) located on unit circle produces infinite filter gain (attenuation)
  - If a pole is at the same *z*-plane location as a zero, they cancel each other
  - Poles or zeros located at origin of *z*-plane do not affect frequency response of filter
  - Filters whose transfer function denominator (numerator) polynomial has real-valued coefficients have poles (zeros) located on real *z*-plane axis, or pairs of poles (zeros) that are complex conjugates of each other

Using Poles and Zeros to Analyze IIR Filters

- *z*-plane pole/zero properties
  - For an FIR filter (transfer function denominator is unity) to have linear phase, any zero on *z*-plane located at $z_0 = Me^{j\alpha}$, where z_0 is not on unit circle and α is not zero, must be accompanied by a reciprocal zero whose location is $1/z_0 = e^{-j\alpha}/M$
  - If an FIR filter has real-valued coefficients, is linear phase, and has a *z*-plane zero not located on real *z*-plane axis or on unit circle, that *z*-plane zero is a member of a “gang of four” zeros
    - If we know *z*-plane location of one of those four zeros, then we know location of the other three zeros

Alternate IIR Filter Structures

- Direct Form I, Direct Form II, and transposed structures
  - Direct Form I structure of an IIR filter can be converted to several alternate forms

Alternate IIR Filter Structures

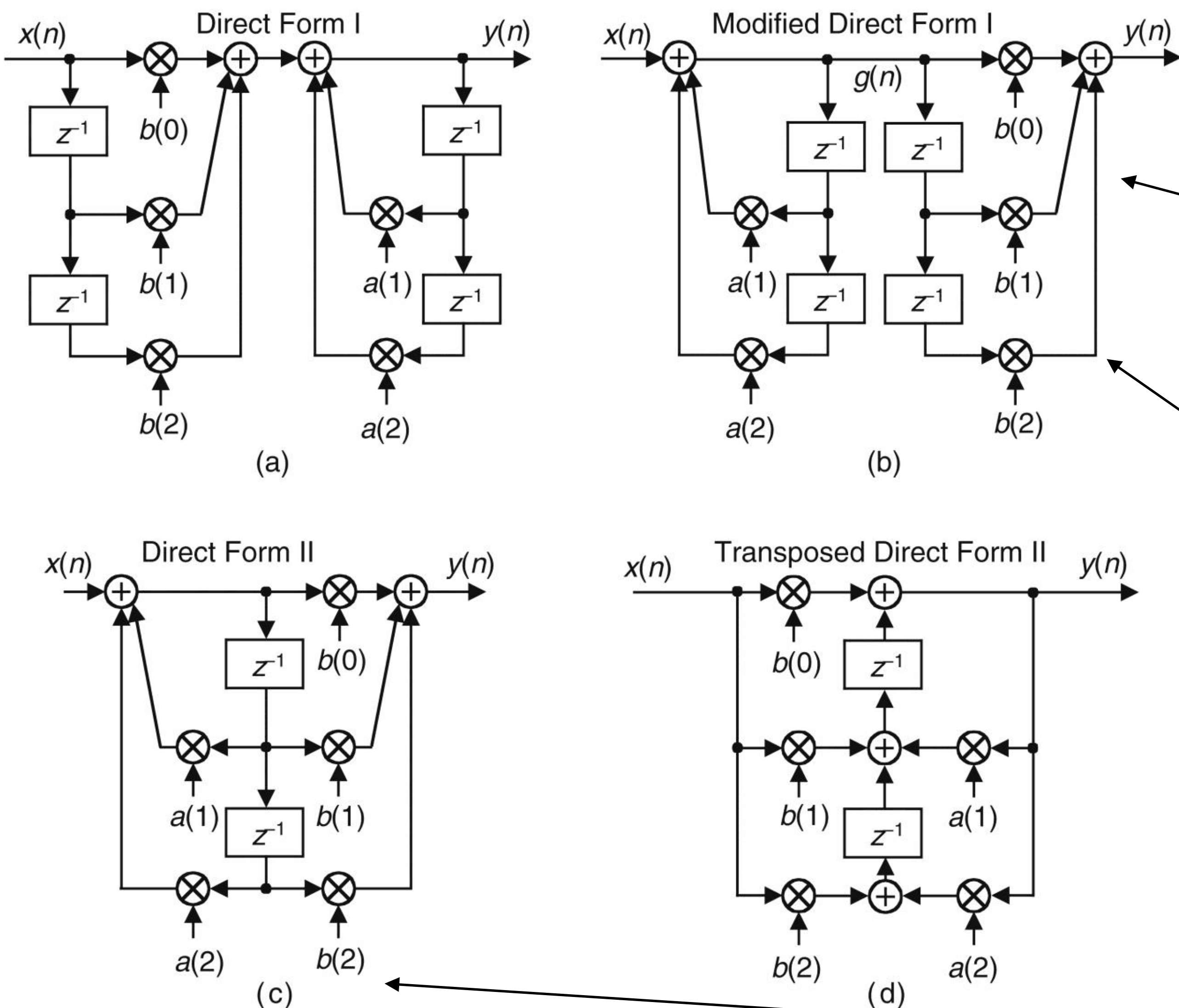


Figure 6-22 Rearranged 2nd-order IIR filter structures: (a) Direct Form I; (b) modified Direct Form I; (c) Direct Form II; (d) transposed Direct Form II.

thinking of feedforward and feedback portions as two separate filter stages, because both stages are linear and time invariant, we can swap them with no change in *y*(*n*) output

because sequence *g*(*n*) is shifted down along both delay lines in (b), we can eliminate one of delay paths and arrive at filter structure shown in (c), where only half the delay storage registers are required compared to Direct Form I structure

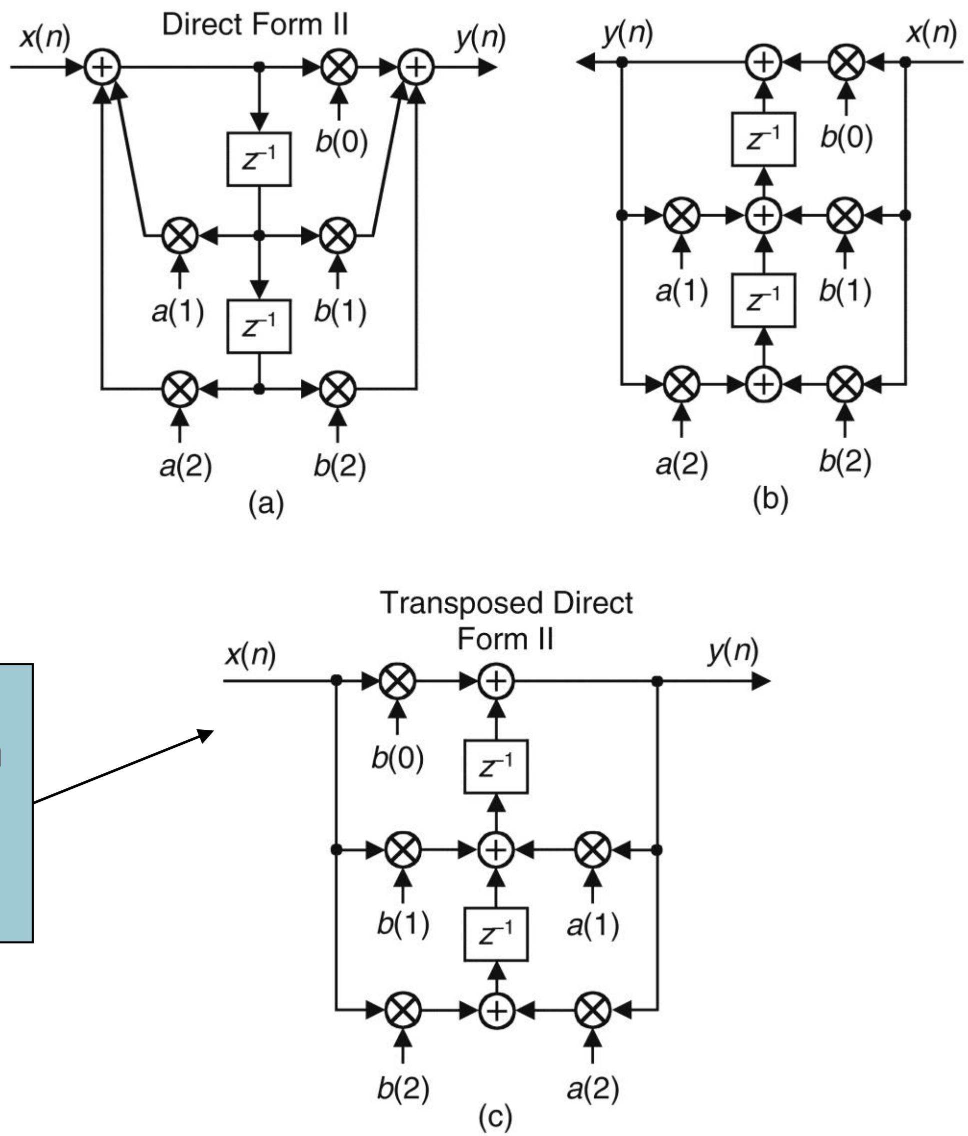
Alternate IIR Filter Structures

- Transposition theorem
  - There is a process in DSP that allows us to change structure of an LTI digital network without changing network’s transfer function (its frequency response)
    - That network conversion process follows *transposition theorem*
  - A transposed version of some digital network might be easier to implement, or may exhibit more accurate processing, than the original network

Alternate IIR Filter Structures

- Steps to transpose a digital filter (starting with Direct Form II)
  - 1. reverse direction of all signal-flow arrows
  - 2. convert all adders to signal nodes
  - 3. convert all signal nodes to adders
  - 4. swap $x(n)$ input and $y(n)$ output labels

Alternate IIR Filter Structures



By convention, we flip the network in (b) from left to right so that $x(n)$ input is on the left as shown in (c)

Figure 6–23 Converting a Direct Form II filter to its transposed form.

Alternate IIR Filter Structures

- Fig. 6-23
  - Transposed filter contains the same number of delay elements, multipliers, and addition operations as the original filter, and both filters have the same transfer function given by

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b(0) + b(1)z^{-1} + b(2)z^{-2}}{1 - a(1)z^{-1} - a(2)z^{-2}}$$

  - When implemented using infinite-precision arithmetic, Direct Form II and transposed Direct Form II filters have identical frequency response properties

Alternate IIR Filter Structures

- Fig. 6-23
  - Transposed Direct Form II structure is less susceptible to errors that can occur when finite-precision binary arithmetic is used to represent data values and filter coefficients within a filter implementation
    - Direct Form II filters implement (possibly high-gain) feedback pole computations before feedforward zeros computations → large intermediate data values which must be truncated
    - Transposed Direct Form II filters implement zeros computations first followed by pole computations
    - Direct Form I filter has the most resistance to coefficient quantization and stability problems

Impulse Invariance IIR Filter Design Method

- Impulse invariance method
  - Is based upon the notion that we can design a discrete filter whose time-domain impulse response is a sampled version of impulse response of a continuous analog filter
  - If that analog filter (called *prototype filter*) has some desired frequency response, then our IIR filter will yield a discrete approximation of that desired response

Impulse Invariance IIR Filter Design Method

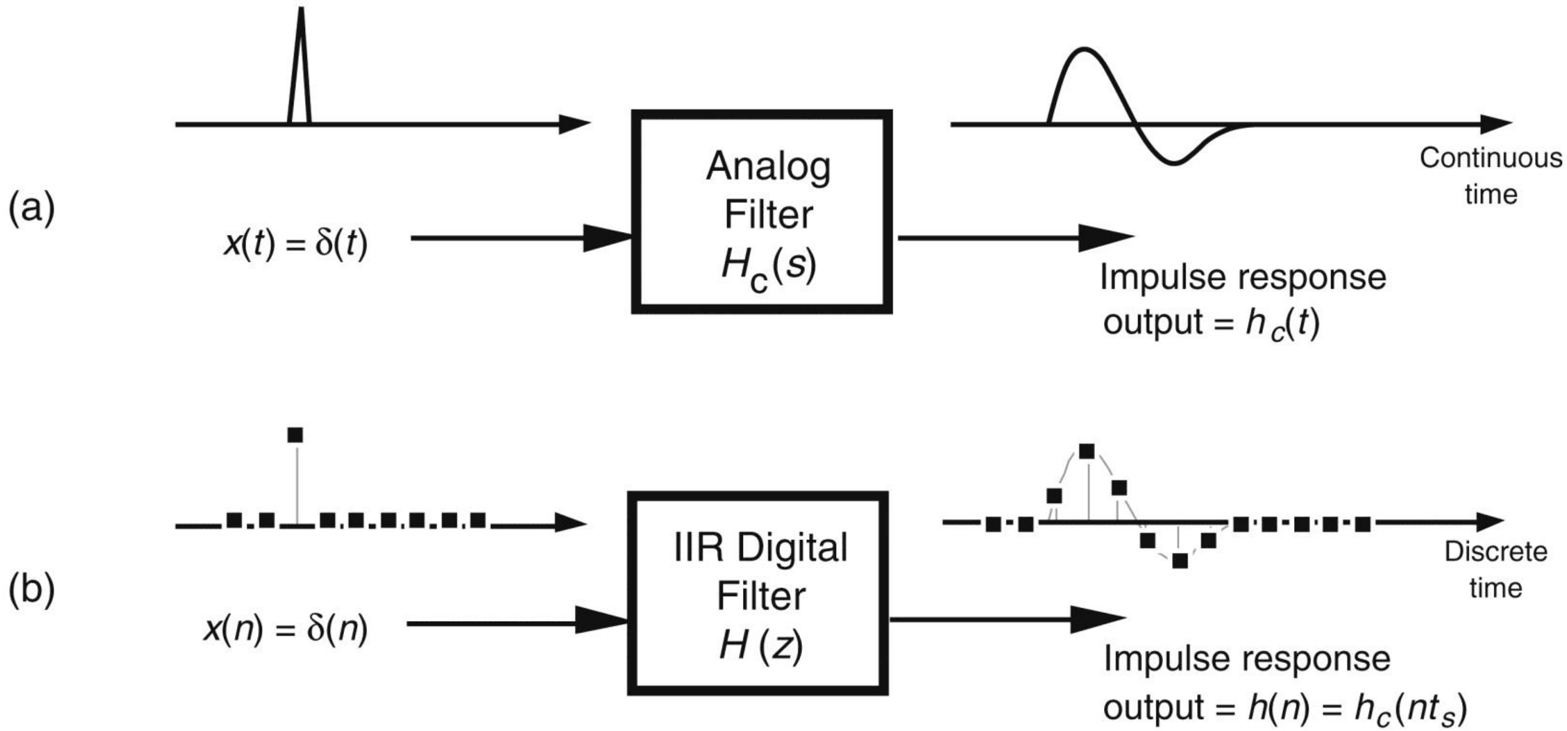
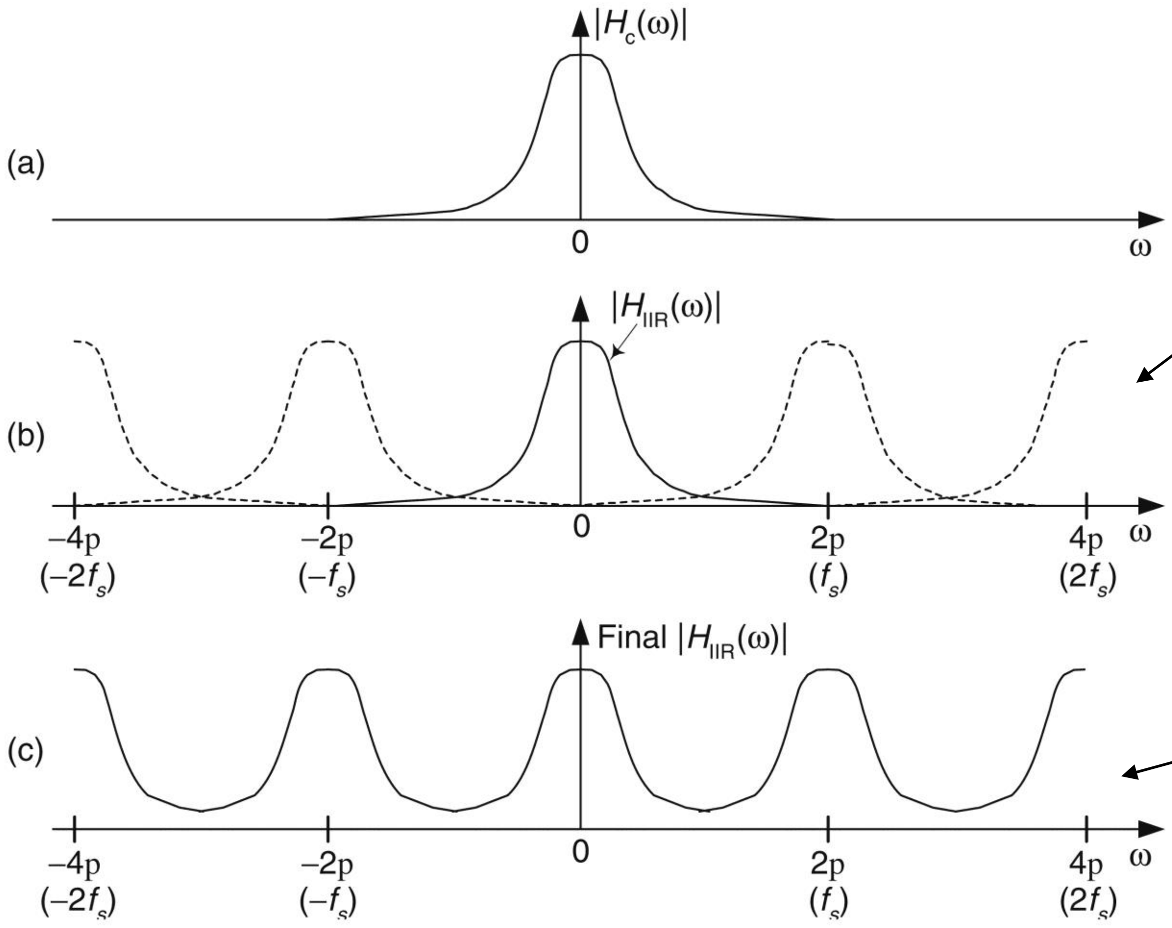


Figure 6–31 Impulse invariance design equivalence of (a) analog filter continuous impulse response; (b) digital filter discrete impulse response.

Impulse Invariance IIR Filter Design Method

- Impulse invariance method
  - Our goal is to design a digital filter whose impulse response is a sampled version of analog filter's continuous impulse response
  - We can map each pole on *s*-plane for analog filter's $H_c(s)$ transfer function to a pole on *z*-plane for discrete IIR filter's $H(z)$ transfer function
  - Impulse invariance method yields useful IIR filters as long as sampling rate is high relative to bandwidth of signal to be filtered
    - IIR filters designed using impulse invariance method are susceptible to aliasing problems because practical analog filters cannot be perfectly band-limited

Impulse Invariance IIR Filter Design Method



we prefer to make f_s as large as possible to minimize the overlap between primary frequency response curve and its replicated images spaced at multiples of $\pm f_s$ Hz

Due to aliasing behavior of impulse invariance design method, this filter design process should never be used to design highpass digital filters

Figure 6–32 Aliasing in the impulse invariance design method: (a) prototype analog filter magnitude response; (b) replicated magnitude responses where $H_{IIR}(\omega)$ is the discrete Fourier transform of $h(n) = h_c(nt_s)$; (c) potential resultant IIR filter magnitude response with aliasing effects.

Impulse Invariance IIR Filter Design Method

- Two methods for designing IIR filters using impulse invariance
  - Method 1
    - Requires that an inverse Laplace transform as well as a *z*-transform be performed
  - Method 2
    - Uses a direct substitution process to avoid inverse Laplace and *z*-transformations at the expense of needing partial fraction expansion algebra necessary to handle polynomials

Impulse Invariance IIR Filter Design Method

- Method 1
  - Step 1: Design a prototype analog filter with desired frequency response
    - In a lowpass filter design, for example, filter type (Chebyshev, Butterworth, elliptic), filter order (number of poles), and cutoff frequency are parameters to be defined in this step
    - Result of this step is a continuous Laplace transfer function $H_c(s)$ expressed as ratio of two polynomials, such as

$$H_c(s) = \frac{b(N)s^N + b(N-1)s^{N-1} + \ldots + b(1)s + b(0)}{a(M)s^M + a(M-1)s^{M-1} + \ldots + a(1)s + a(0)} = \frac{\sum_{k=0}^{N} b(k)s^k}{\sum_{k=0}^{M} a(k)s^k}$$

    - $N < M$, and $a(k)$ and $b(k)$ are constants

Impulse Invariance IIR Filter Design Method

- Method 1
  - Step 2: Determine analog filter's continuous time-domain impulse response $h_c(t)$ from $H_c(s)$ Laplace transfer function
    - Can be done using Laplace tables as opposed to actually evaluating an inverse Laplace transform equation
  - Step 3: Determine digital filter's f_s, and calculate $t_s = 1/f_s$
    - f_s is chosen based on absolute frequency, in Hz, of prototype analog filter
    - Because of aliasing problems associated with this impulse invariance design method, f_s should be made as large as is practical

Impulse Invariance IIR Filter Design Method

- Method 1
  - Step 4: Find z-transform of continuous $h_c(t)$ to obtain IIR filter's z-domain transfer function $H(z)$ in form of a ratio of polynomials in z
  - Step 5: Substitute the value t_s for the continuous variable t in $H(z)$ transfer function obtained in Step 4
    - In performing this step, we are ensuring that IIR filter's discrete $h(n)$ impulse response is a sampled version of continuous filter's $h_c(t)$ impulse response so that $h(n) = h_c(nt_s)$, for $0 \leq n \leq \infty$

Impulse Invariance IIR Filter Design Method

- Method 1
  - Step 6: $H(z)$ from Step 5 will now be of the general form

$$H(z) = \frac{b(N)z^{-N} + b(N-1)z^{-(N-1)} + \ldots + b(1)z^{-1} + b(0)}{a(M)z^{-M} + a(M-1)z^{-(M-1)} + \ldots + a(1)z^{-1} + a(0)} = \frac{\sum_{k=0}^{N} b(k)z^{-k}}{1 - \sum_{k=1}^{M} a(k)z^{-k}}$$

    - Because process of sampling continuous impulse response results in a digital filter frequency response that's scaled by a factor of $1/t_s$, we include t_s factor in this equation

$$H(z) = \frac{Y(z)}{X(z)} = \frac{t_s \sum_{k=0}^{N} b(k)z^{-k}}{1 - \sum_{k=1}^{M} a(k)z^{-k}}$$

    - Incorporating t_s makes IIR filter time-response scaling independent of sampling rate, and discrete filter will have the same gain as prototype analog filter

Impulse Invariance IIR Filter Design Method

- Method 1
  - Step 7: By inspection, we can express filter's time-domain difference equation as

these time-domain expressions apply only to the filter structure in Fig. 6-18. The *a*(*k*) and *b*(*k*), or $t_s \cdot b(k)$, coefficients, however, can be applied to the improved IIR structure shown in Fig. 6-22 to complete our design

$$H(z)=\frac{\sum_{k=0}^{N}b(k)z^{-k}}{1-\sum_{k=1}^{M}a(k)z^{-k}}\rightarrow$$

$$y(n)=b(0)x(n)+b(1)x(n-1)+b(2)x(n-2)+\ldots+b(N)x(n-N)$$
$$+a(1)y(n-1)+a(2)y(n-2)+\ldots+a(M)y(n-M)$$

$$H(z)=\frac{t_s\sum_{k=0}^{N}b(k)z^{-k}}{1-\sum_{k=1}^{M}a(k)z^{-k}}\rightarrow$$

$$y(n)=t_s\cdot[b(0)x(n)+b(1)x(n-1)+b(2)x(n-2)+\ldots+b(N)x(n-N)]$$
$$+a(1)y(n-1)+a(2)y(n-2)+\ldots+a(M)y(n-M)$$

Impulse Invariance IIR Filter Design Method

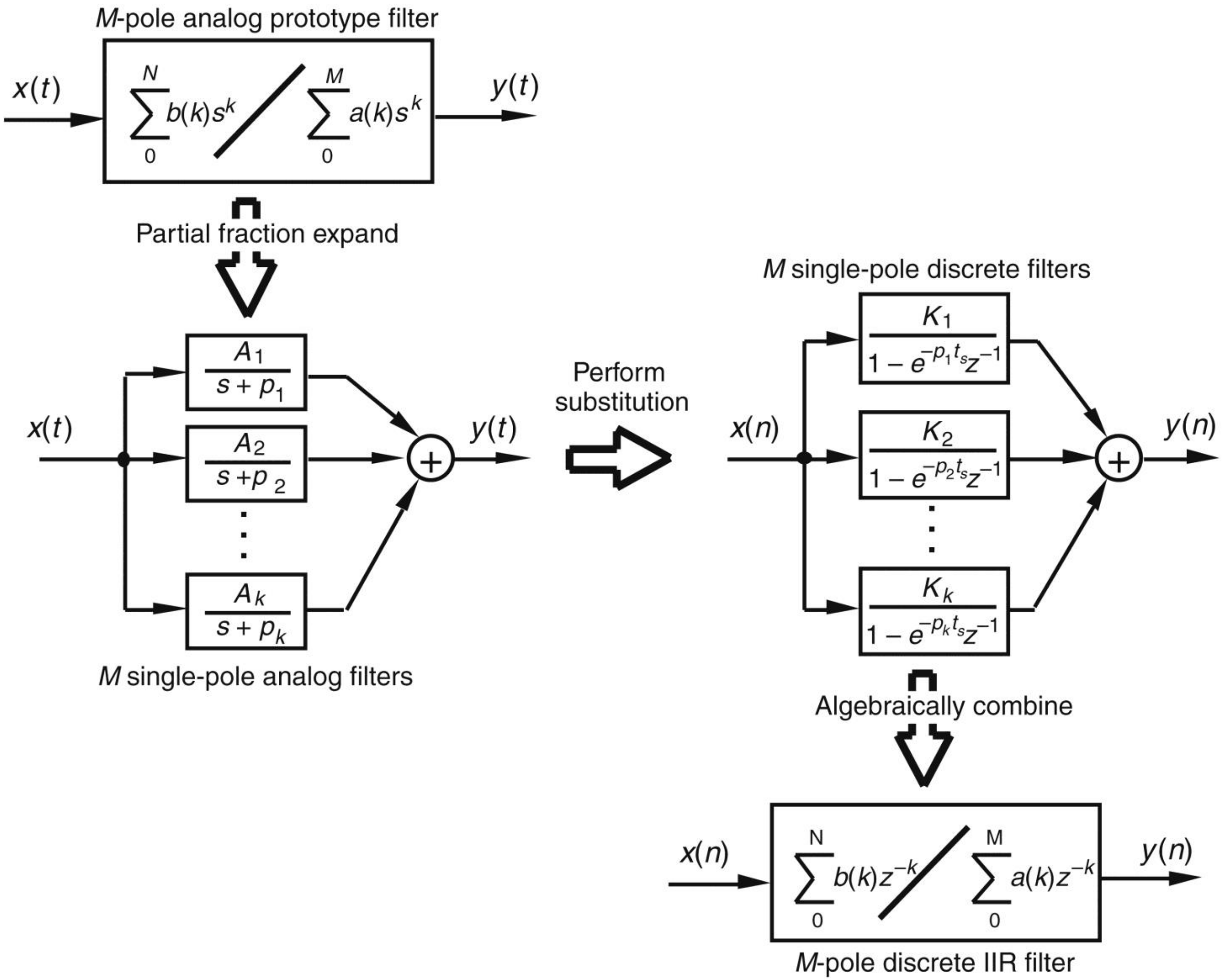


Figure 6–33 Mathematical flow of the impulse invariance design Method 2.

Impulse Invariance IIR Filter Design Method

- Method 2
  - Step 1: Obtain Laplace transfer function $H_c(s)$ for prototype analog filter
    - Same as Method 1, Step 1

$$H_c(s) = \frac{b(N)s^N + b(N-1)s^{N-1} + \ldots + b(1)s + b(0)}{a(M)s^M + a(M-1)s^{M-1} + \ldots + a(1)s + a(0)} = \frac{\sum_{k=0}^{N} b(k)s^k}{\sum_{k=0}^{M} a(k)s^k}$$

  - Step 2: Select an appropriate sampling frequency f_s and calculate sample period as $t_s = 1/f_s$
    - Same as Method 1, Step 3

Impulse Invariance IIR Filter Design Method

- Method 2
  - Step 3: Express analog filter's Laplace transfer function $H_c(s)$ as sum of single-pole filters
    - This requires us to use partial fraction expansion methods

$$H_c(s) = \frac{b(N)s^N + b(N-1)s^{N-1} + \ldots + b(1)s + b(0)}{a(M)s^M + a(M-1)s^{M-1} + \ldots + a(1)s + a(0)}$$

$$= \sum_{k=1}^{M} \frac{A_k}{s + p_k} = \frac{A_1}{s + p_1} + \frac{A_2}{s + p_2} + \ldots + \frac{A_M}{s + p_M}$$

where $M > N$, individual A_k factors are constants, and the kth pole is located at $-p_k$ on s-plane

- $H_k(s)$ = kth single-pole analog filter

$$H_k(s) = \frac{A_k}{s + p_k}$$

Impulse Invariance IIR Filter Design Method

- Method 2
  - Step 4: Substitute $1-e^{-p_k t_s} z^{-1}$ for $s+p_k$ in $H_c(s)$ equation in Step 3
    - This mapping of each $H_k(s)$ pole, located at $s = -p_k$ on s-plane, to $z = e^{-p_k t_s}$ location on z-plane is how we approximate the impulse response of each single-pole analog filter by a single-pole digital filter
    - So, the kth analog single-pole filter $H_k(s)$ is approximated by a single-pole digital filter whose z-domain transfer function is

$$H_k(z) = \frac{A_k}{1-e^{-p_k t_s} z^{-1}}$$

$$\xrightarrow{\text{final combined } H(z)} H(z) = \sum_{k=1}^{M} H_k(z) = \sum_{k=1}^{M} \frac{A_k}{1-e^{-p_k t_s} z^{-1}}$$

Impulse Invariance IIR Filter Design Method

- Method 2
  - Step 5: Calculate z-domain transfer function of the sum of M single-pole digital filters in the form of a ratio of two polynomials in z
    - Because $H(z)$ in Step 4 will be a series of fractions, we'll have to combine those fractions over a common denominator to get a single ratio of polynomials in form of

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b(k) z^{-k}}{1 - \sum_{k=1}^{M} a(k) z^{-k}}$$

Impulse Invariance IIR Filter Design Method

- Method 2
  - Step 6: As in Method 1, Step 6, by inspection, we express filter's time-domain equation in general form of

$$H(z) = \frac{\sum_{k=0}^{N} b(k) z^{-k}}{1 - \sum_{k=1}^{M} a(k) z^{-k}} \rightarrow$$

$$y(n) = b(0)x(n) + b(1)x(n-1) + b(2)x(n-2) + \ldots + b(N)x(n-N)$$
$$+ a(1)y(n-1) + a(2)y(n-2) + \ldots + a(M)y(n-M)$$

$$H(z) = \frac{t_s \sum_{k=0}^{N} b(k) z^{-k}}{1 - \sum_{k=1}^{M} a(k) z^{-k}} \rightarrow$$

$$y(n) = t_s \cdot [b(0)x(n) + b(1)x(n-1) + b(2)x(n-2) + \ldots + b(N)x(n-N)]$$
$$+ a(1)y(n-1) + a(2)y(n-2) + \ldots + a(M)y(n-M)$$

Finally, we implement the improved IIR structure shown in Fig. 6-22 using $a(k)$ and $b(k)$ coefficients or $a(k)$ and $t_s \cdot b(k)$ coefficients from these time-domain expressions

Impulse Invariance IIR Filter Design Method

- Impulse invariance design Method 1 example
  - Design an IIR filter that approximates a 2nd-order Chebyshev prototype analog lowpass filter whose passband ripple is 1 dB
  - f_s = 100 Hz (t_s = 0.01)
  - Filter's 1 dB cutoff frequency = 20 Hz

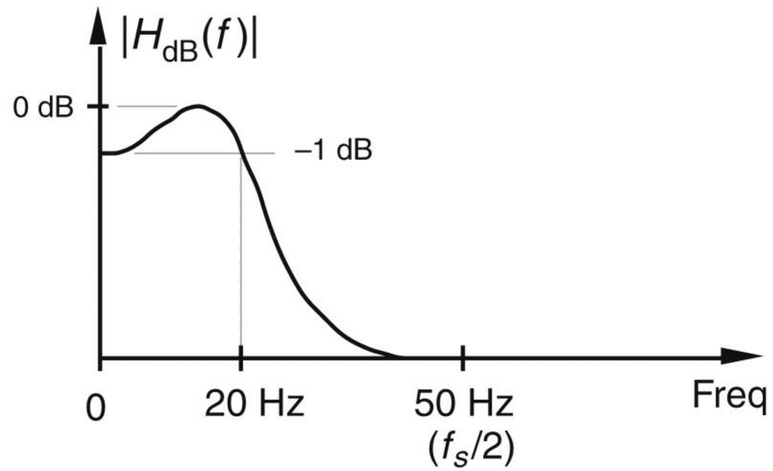


Figure 6–34 Frequency magnitude response of the example prototype analog filter.

Impulse Invariance IIR Filter Design Method

- Method 1 example
  - Given above filter requirements, assume that analog prototype filter design effort results in

$$H_c(s) = \frac{17410.145}{s^2 + 137.94536\,s + 17410.145}$$

    - It's this transfer function that we intend to approximate with our discrete IIR filter

$X(s)$, Laplace transform of $x(t)$: $\qquad x(t)$:

$$\frac{A\omega}{(s+\alpha)^2 + \omega^2} \leftrightarrow A e^{-\alpha t} \cdot \sin(\omega t)$$

$$\frac{A\omega}{(s+\alpha)^2 + \omega^2} = \frac{A\omega}{s^2 + 2\alpha s + \alpha^2 + \omega^2}$$

Impulse Invariance IIR Filter Design Method

$$H_c(s) = \frac{17410.145}{s^2 + 137.94536\,s + 17410.145} = \frac{A\omega}{s^2 + 2\alpha s + \alpha^2 + \omega^2}$$

$$\alpha = \frac{137.94536}{2} = 68.972680$$

$$\alpha^2 + \omega^2 = 17410.145 \rightarrow \omega = \sqrt{17410.145 - \alpha^2} = 112.485173$$

$$A = \frac{17410.145}{\omega} = 154.77724$$

$$H_c(s) = \frac{(154.77724)(112.485173)}{(s + 68.972680)^2 + (112.485173)^2}$$

time-domain impulse response of prototype analog filter

$$\longrightarrow h_c(t) = Ae^{-\alpha t} \cdot \sin(\omega t)$$

$$= 154.77724e^{-68.972680t} \cdot \sin(112.485173t)$$

Impulse Invariance IIR Filter Design Method

$x(t):$ $\quad X(z)$, z-transform of $x(t):$

$$Ce^{-\alpha t}\cdot\sin(\omega t) \leftrightarrow \frac{Ce^{-\alpha t}\cdot\sin(\omega t)z^{-1}}{1-2[e^{-\alpha t}\cdot\cos(\omega t)]z^{-1}+e^{-2\alpha t}z^{-2}}$$

$$h_c(t)=154.77724e^{-68.972680t}\cdot\sin(112.485173t)\rightarrow$$

$$H(z)=\frac{154.77724e^{-68.972680t}\cdot\sin(112.485173t)z^{-1}}{1-2[e^{-68.972680t}\cdot\cos(112.485173t)]z^{-1}+e^{-2\times 68.972680t}z^{-2}}$$

substitute $t_s=0.01$ for continuous variable t

$$\xrightarrow{} H(z)=\frac{154.77724e^{-68.972680\times 0.01}\cdot\sin(112.485173\times 0.01)z^{-1}}{1-2[e^{-68.972680\times 0.01}\cdot\cos(112.485173\times 0.01)]z^{-1}+e^{-2\times 68.972680\times 0.01}z^{-2}}$$

$$=\frac{154.77724e^{-0.68972680}\cdot\sin(1.12485173)z^{-1}}{1-2[e^{-0.68972680}\cdot\cos(1.12485173)]z^{-1}+e^{-2\times 0.68972680}z^{-2}}$$

$$=\frac{Y(z)}{X(z)}=\frac{70.059517z^{-1}}{1-0.43278805z^{-1}+0.25171605z^{-2}}$$

Impulse Invariance IIR Filter Design Method

$$H(z)=\frac{Y(z)}{X(z)}=\frac{70.059517z^{-1}}{1-0.43278805z^{-1}+0.25171605z^{-2}}$$

$$Y(z)\cdot(1-0.43278805z^{-1}+0.25171605z^{-2})=X(z)\cdot(70.059517z^{-1})$$

$$Y(z)=70.059517\cdot X(z)z^{-1}+0.43278805\cdot Y(z)z^{-1}-0.25171605\cdot Y(z)z^{-2}$$

$$\xrightarrow{\text{Get time-domain expression for IIR filter. Multiply } x(n\text{-}1) \text{ coefficient by } t_s=0.01 \text{ for scaling}}$$

$$y(n)=0.01\cdot 70.059517\cdot x(n-1)+0.43278805\cdot y(n-1)-0.25171605\cdot y(n-2)$$

$$y(n)=0.70059517\cdot x(n-1)+0.43278805\cdot y(n-1)-0.25171605\cdot y(n-2)$$

these coefficients are what we use in implementing the improved IIR structure shown in Fig. 6-22 to approximate the original 2nd-order Chebyshev analog lowpass filter

Impulse Invariance IIR Filter Design Method

- Impulse invariance design Method 2 example
  - Given original prototype filter's $H_c(s)$ as

$$H_c(s) = \frac{17410.145}{s^2 + 137.94536\,s + 17410.145}$$

and $t_s = 0.01$

$$\xrightarrow[\ ]{\substack{b\ =137.94536\\ c\ =17410.145}} H_c(s) = \frac{c}{s^2 + b\,s + c}$$

If we substitute the values for *b* and *c*, the quantity under radical sign is negative → factors in denominator are complex

$$H_c(s) = \frac{c}{\left(s + \frac{b}{2} + \sqrt{\frac{b^2 - 4c}{4}}\right) \cdot \left(s + \frac{b}{2} - \sqrt{\frac{b^2 - 4c}{4}}\right)}$$

$$\xrightarrow{j=\sqrt{-1}\ \text{and}\ R=\sqrt{|(b^2-4c)/4|}} H_c(s) = \frac{c}{(s + b/2 + jR)(s + b/2 - jR)}$$

Impulse Invariance IIR Filter Design Method

$$H_c(s) = \frac{c}{(s+b/2+jR)(s+b/2-jR)} = \frac{K_1}{(s+b/2+jR)} + \frac{K_2}{(s+b/2-jR)}$$

$$K_1 = \left[\frac{c}{(s+b/2+jR)(s+b/2-jR)} \times (s+b/2+jR)\right]_{s=-b/2-jR}$$

$$= \frac{c}{-b/2-jR+b/2-jR} = \frac{c}{-2jR} = \frac{jc}{2R}$$

$$K_2 = \left[\frac{c}{(s+b/2+jR)(s+b/2-jR)} \times (s+b/2-jR)\right]_{s=-b/2+jR}$$

$$= \frac{c}{-b/2+jR+b/2+jR} = \frac{c}{2jR} = \frac{-jc}{2R}$$

$$H_c(s) = \frac{jc/2R}{(s+b/2+jR)} + \frac{-jc/2R}{(s+b/2-jR)}$$

Impulse Invariance IIR Filter Design Method

$$H_c(s) = \frac{jc/2R}{(s+b/2+jR)} + \frac{-jc/2R}{(s+b/2-jR)}$$

map poles $p_1 = -b/2 - jR$ and $p_2 = -b/2 + jR$ from s-plane to z-plane:
substitute $1 - e^{-p_k t_s} z^{-1}$ for $s + p_k$ terms

$$\longrightarrow$$

$$H(z) = \frac{jc/2R}{1-e^{-(b/2+jR)t_s} z^{-1}} + \frac{-jc/2R}{1-e^{-(b/2-jR)t_s} z^{-1}}$$

$$H(z) = \frac{(jc/2R)(1-e^{-(b/2-jR)t_s} z^{-1}) - (jc/2R)(1-e^{-(b/2+jR)t_s} z^{-1})}{(1-e^{-(b/2+jR)t_s} z^{-1})(1-e^{-(b/2-jR)t_s} z^{-1})}$$

$$H(z) = \frac{(jc/2R)(1-e^{-(b/2-jR)t_s} z^{-1} - 1 + e^{-(b/2+jR)t_s} z^{-1})}{1-e^{-(b/2-jR)t_s} z^{-1} - e^{-(b/2+jR)t_s} z^{-1} + e^{-bt_s} z^{-2}}$$

$$H(z) = \frac{(jc/2R)(e^{-(b/2+jR)t_s} - e^{-(b/2-jR)t_s})z^{-1}}{1-(e^{-(b/2-jR)t_s} + e^{-(b/2+jR)t_s})z^{-1} + e^{-bt_s} z^{-2}}$$

$$H(z) = \frac{(jc/2R)e^{-bt_s/2}(e^{-jRt_s} - e^{jRt_s})z^{-1}}{1-e^{-bt_s/2}(e^{jRt_s} + e^{-jRt_s})z^{-1} + e^{-bt_s} z^{-2}}$$

Impulse Invariance IIR Filter Design Method

$$H(z)=\frac{(jc/2R)e^{-bt_s/2}(e^{-jRt_s}-e^{jRt_s})z^{-1}}{1-e^{-bt_s/2}(e^{jRt_s}+e^{-jRt_s})z^{-1}+e^{-bt_s}z^{-2}}$$

$$\xrightarrow{\text{Euler: } \sin(\phi)=(e^{j\phi}-e^{-j\phi})/2j \text{ and } \cos(\phi)=(e^{j\phi}+e^{-j\phi})/2}$$

$$H(z)=\frac{(jc/2R)e^{-bt_s/2}[-2j\sin(Rt_s)]z^{-1}}{1-e^{-bt_s/2}[2\cos(Rt_s)]z^{-1}+e^{-bt_s}z^{-2}}$$

$$=\frac{(c/R)e^{-bt_s/2}[\sin(Rt_s)]z^{-1}}{1-e^{-bt_s/2}[2\cos(Rt_s)]z^{-1}+e^{-bt_s}z^{-2}}$$

$$\xrightarrow{c=17410.145,\ b=137.94536,\ R=112.48517, \text{ and } t_s=0.01}$$

$$H(z)=\frac{(154.77724)(0.50171312)(0.902203655)z^{-1}}{1-(0.50171312)(0.86262058)z^{-1}+0.25171605z^{-2}}$$

$$=\frac{70.059517z^{-1}}{1-0.43278805z^{-1}+0.25171605z^{-2}}$$

Impulse Invariance IIR Filter Design Method

$$H(z) = \frac{Y(z)}{X(z)} = \frac{70.059517z^{-1}}{1 - 0.43278805z^{-1} + 0.25171605z^{-2}}$$

$$Y(z)\cdot(1 - 0.43278805z^{-1} + 0.25171605z^{-2}) = X(z)\cdot(70.059517z^{-1})$$

$$Y(z) = 70.059517\cdot X(z)z^{-1} + 0.43278805\cdot Y(z)z^{-1} - 0.25171605\cdot Y(z)z^{-2}$$

$$y(n) = 70.059517\cdot x(n-1) + 0.43278805\cdot y(n-1) - 0.25171605\cdot y(n-2)$$

$$\xrightarrow{\text{multiply the } x(n-1) \text{ coefficient by } t_s}$$

$$y(n) = 0.01\cdot 70.059517\cdot x(n-1) + 0.43278805\cdot y(n-1) - 0.25171605\cdot y(n-2)$$

$$y(n) = 0.70059517\cdot x(n-1) + 0.43278805\cdot y(n-1) - 0.25171605\cdot y(n-2)$$

Impulse Invariance IIR Filter Design Method

$$H_c(s) = \frac{jc/2R}{(s+b/2+jR)} + \frac{-jc/2R}{(s+b/2-jR)}$$

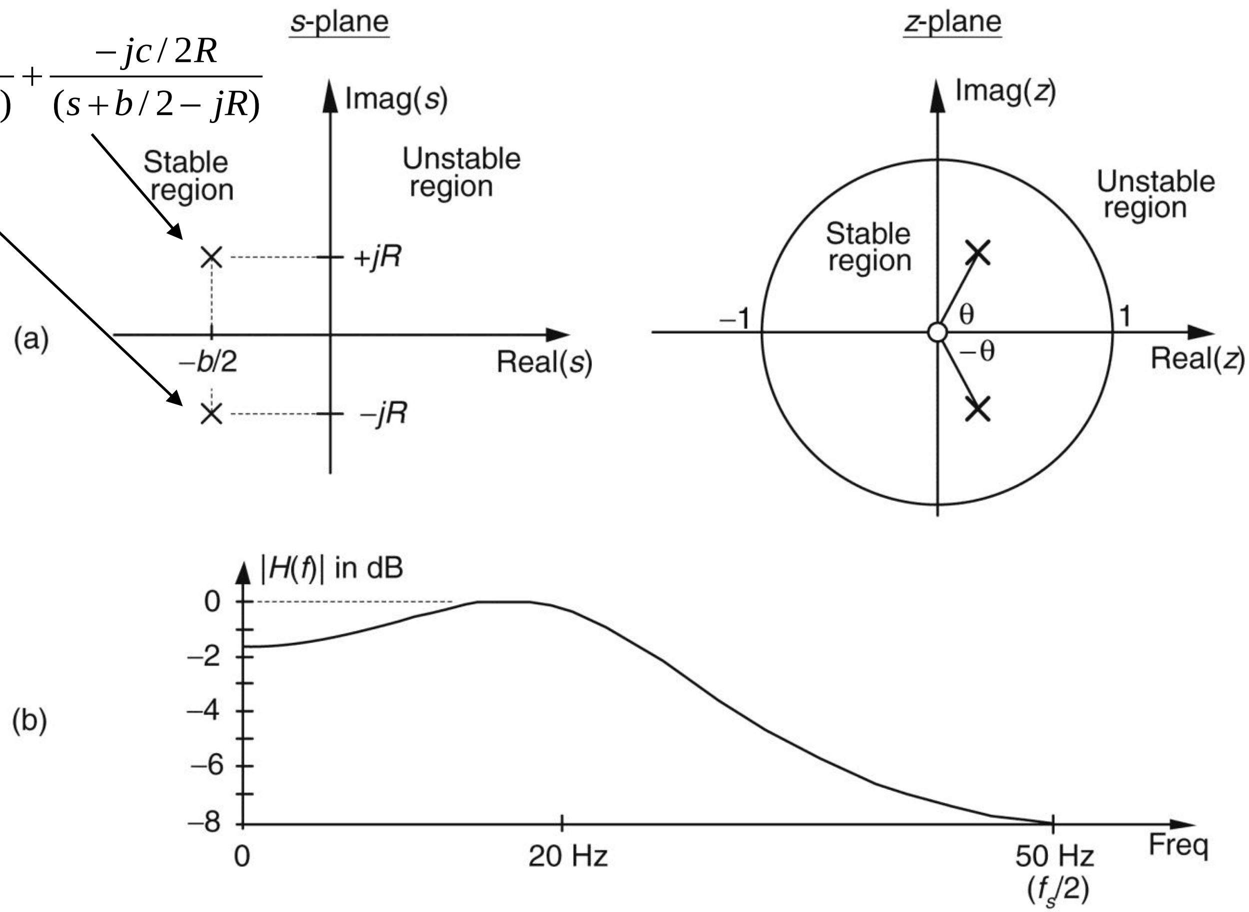


Figure 6–35 Impulse invariance design example filter characteristics: (a) s-plane pole locations of prototype analog filter and z-plane pole locations of discrete IIR filter; (b) frequency magnitude response of the discrete IIR filter.

Impulse Invariance IIR Filter Design Method

- IIR filter's *z*-plane pole locations

$$H(z) = \frac{jc/2R}{1 - e^{-(b/2 + jR)t_s} z^{-1}} + \frac{-jc/2R}{1 - e^{-(b/2 - jR)t_s} z^{-1}}$$

$$H(z) \cdot \frac{z}{z} = \frac{z(jc/2R)}{z(1 - e^{-(b/2 + jR)t_s} z^{-1})} + \frac{z(-jc/2R)}{z(1 - e^{-(b/2 - jR)t_s} z^{-1})}$$

$$= \frac{(jc/2R)z}{z - e^{-(b/2 + jR)t_s}} + \frac{(-jc/2R)z}{z - e^{-(b/2 - jR)t_s}}$$

$$\xrightarrow{\text{lower z-plane pole}} z = e^{-(b/2 + jR)t_s} = e^{-bt_s/2} e^{-jRt_s} = e^{-bt_s/2} \angle -Rt_s \text{ radians}$$

$$0.5017 \angle -64.45^\circ$$

$$\xrightarrow{\text{upper z-plane pole}} = \text{conjugate pole}: 0.5017 \angle +64.45^\circ$$

Impulse Invariance IIR Filter Design Method

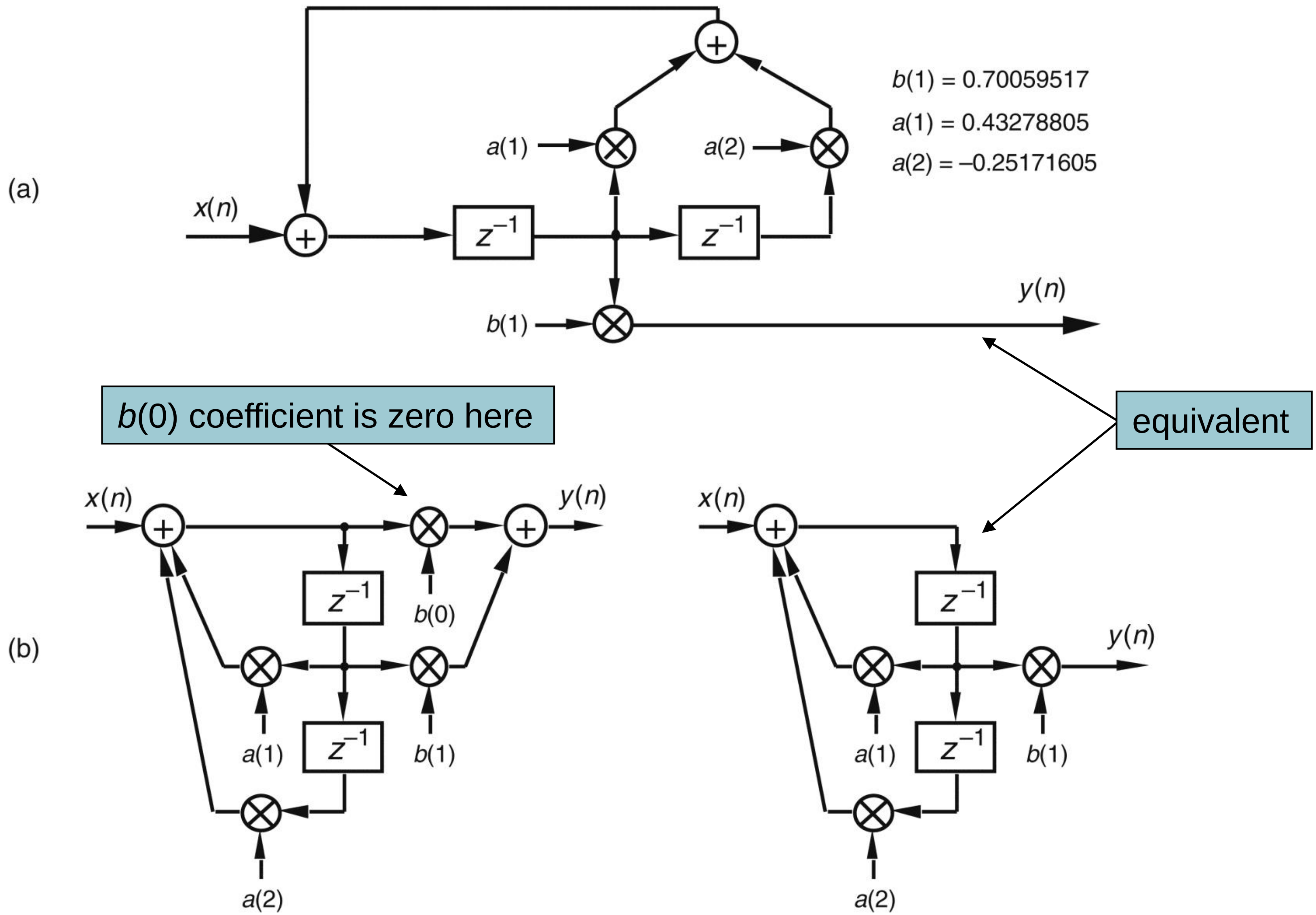


Figure 6–36 Implementations of the impulse invariance design example filter.

Impulse Invariance IIR Filter Design Method

- In general, Method 2 is more popular for two reasons
  - (1) inverse Laplace and *z*-transformations can be very difficult for higher-order filters
  - (2) unlike Method 1, Method 2 can be coded in a software routine or a computer spreadsheet

Impulse Invariance IIR Filter Design Method

- Fig. 6-35(b)
  - Roll-off is not particularly steep
    - Attenuation slope is so gradual that it doesn't appear to be of much use as a lowpass filter
    - Any frequency representation (be it a digital filter magnitude response or a signal spectrum) that has nonzero values at $+f_s/2$, most probably has aliasing problem
  - We also see that passband ripple is greater than the desired value of 1 dB in Fig. 6-34
  - It's not the low order of filter that contributes to its poor performance, but the sampling rate used

Impulse Invariance IIR Filter Design Method

Increasing sampling rate to 400 Hz results in the much improved frequency response

Sampling rate changes do not affect filter order or implementation structure; it only changes t_s in our design equations, resulting in a different set of filter coefficients

the smaller we make t_s (larger f_s), the better the resulting filter because the replicated spectral overlap indicated in Fig. 6-32(b) is reduced due to the larger f_s

impulse invariance IIR filter design techniques are most appropriate for narrowband filters, that is, lowpass filters whose cutoff frequencies are much smaller than the sampling rate

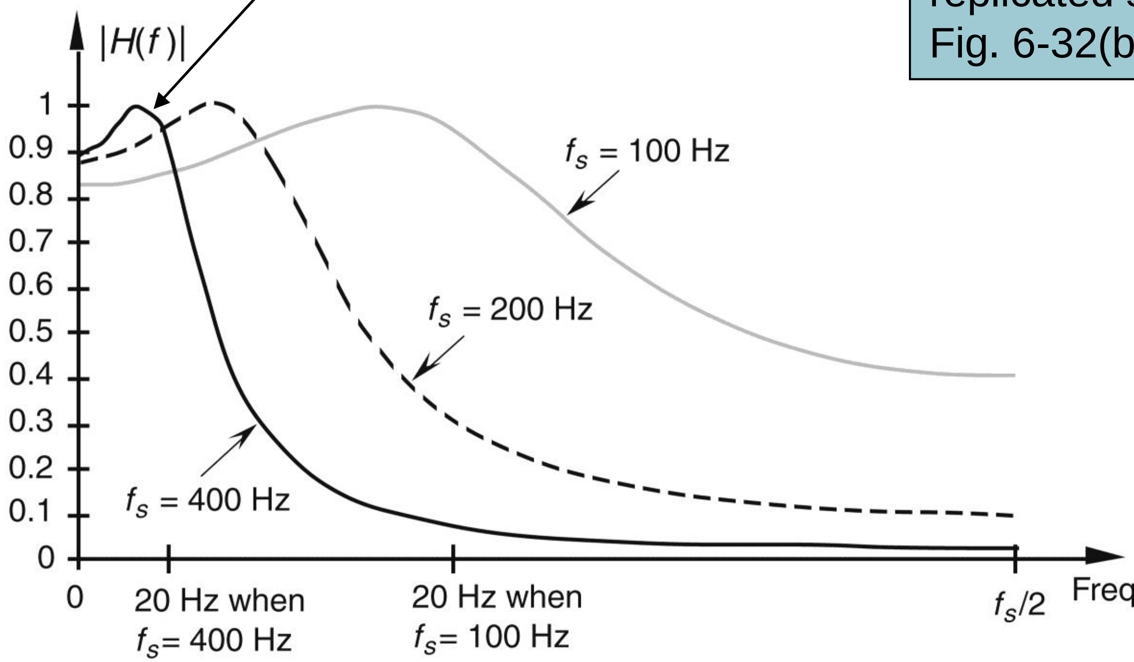


Figure 6–37 IIR filter frequency magnitude response, on a linear scale, at three separate sampling rates. Notice how the filter's absolute cutoff frequency of 20 Hz shifts relative to the different f_s sampling rates.

Bilinear Transform IIR Filter Design Method

- Bilinear transform method
  - A popular analytical IIR filter design technique
  - Like impulse invariance method, this design technique approximates a prototype analog filter defined by continuous Laplace transfer function $H_c(s)$ with a discrete filter whose transfer function is $H(z)$

Bilinear Transform IIR Filter Design Method

- Bilinear transform method has great utility
  - It allows to substitute a function of z for s in $H_c(s)$ to get $H(z)$, eliminating the need for Laplace and z-transformations as well as any necessity for partial fraction expansion algebra
  - It maps the entire s-plane to z-plane, enabling us to completely avoid frequency-domain aliasing problems we had with impulse invariance design method
  - It induces a nonlinear distortion of $H(z)$'s frequency axis, relative to the original prototype analog filter's frequency axis, that sharpens the final roll-off of digital lowpass filters

Bilinear Transform IIR Filter Design Method

- Bilinear transform method
  - If transfer function of a prototype analog filter is $H_c(s)$, we obtain discrete IIR filter z-domain transfer function $H(z)$ by substituting the following for s in $H_c(s)$

$$s = \frac{2}{t_s}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)$$

where t_s is discrete filter's sampling period ($1/f_s$)

Bilinear Transform IIR Filter Design Method

- Bilinear transform's *s*- to *z*-plane mapping
  - Any pole on the left side of *s*-plane will map to the inside of unit circle in *z*-plane

$$s = \frac{2}{t_s}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \rightarrow z = \frac{1 + (t_s/2)s}{1 - (t_s/2)s}$$

$$\xrightarrow{s=\sigma + j\omega_a} z = \frac{1 + \sigma t_s/2 + j\omega_a t_s/2}{1 - \sigma t_s/2 - j\omega_a t_s/2} = \frac{(1 + \sigma t_s/2) + j\omega_a t_s/2}{(1 - \sigma t_s/2) - j\omega_a t_s/2}$$

analog

$$|z| = \frac{\text{Mag}_{\text{numerator}}}{\text{Mag}_{\text{denominator}}} = \sqrt{\frac{(1 + \sigma t_s/2)^2 + (\omega_a t_s/2)^2}{(1 - \sigma t_s/2)^2 + (\omega_a t_s/2)^2}}$$

- If σ < 0, |*z*| will be less than 1
- If σ > 0, |*z*| will be greater than 1
- This confirms that when using bilinear transform, any pole located on the left side of *s*-plane (σ < 0) will map to a *z*-plane location inside unit circle

Bilinear Transform IIR Filter Design Method

- Bilinear transform's *s*- to *z*-plane mapping
  - $j\omega_a$ axis of *s*-plane maps to perimeter of unit circle in *z*-plane

$$z=\frac{(1+\sigma t_s/2)+j\omega_a t_s/2}{(1-\sigma t_s/2)-j\omega_a t_s/2}\xrightarrow{\sigma=0} z=\frac{1+j\omega_a t_s/2}{1-j\omega_a t_s/2}$$

$$|z|_{\sigma=0}=\frac{\text{Mag}_{\text{numerator}}}{\text{Mag}_{\text{denominator}}}=\sqrt{\frac{(1)^2+(\omega_a t_s/2)^2}{(1)^2+(\omega_a t_s/2)^2}}=1$$

Bilinear Transform IIR Filter Design Method

- Bilinear transform's *s*- to *z*-plane mapping
  - Frequency mapping from $j\omega_a$ axis of *s*-plane to perimeter of unit circle in *z*-plane is not linear

$$s = \frac{2}{t_s}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) \xrightarrow{z \text{ on unit circle } = 1e^{j\omega_d}} s = \frac{2}{t_s}\left(\frac{1-e^{-j\omega_d}}{1+e^{-j\omega_d}}\right)$$

$$s = \sigma + j\omega_a = \frac{2}{t_s}\cdot\frac{e^{-j\omega_d/2}(e^{j\omega_d/2} - e^{-j\omega_d/2})}{e^{-j\omega_d/2}(e^{j\omega_d/2} + e^{-j\omega_d/2})}$$

$$\xrightarrow[\text{and } \cos(\phi)=(e^{j\phi}+e^{-j\phi})/2]{\text{Euler: } \sin(\phi)=(e^{j\phi}-e^{-j\phi})/2j} s = \sigma + j\omega_a = \frac{2}{t_s}\cdot\frac{e^{-j\omega_d/2}[2j\sin(\omega_d/2)]}{e^{-j\omega_d/2}[2\cos(\omega_d/2)]}$$

$$s = \sigma + j\omega_a = \frac{2}{t_s}\cdot\frac{2e^{-j\omega_d/2}}{2e^{-j\omega_d/2}}\cdot\frac{j\sin(\omega_d/2)}{\cos(\omega_d/2)} = \frac{j2}{t_s}\tan(\omega_d/2)$$

Bilinear Transform IIR Filter Design Method

- Bilinear transform's *s*- to *z*-plane mapping

$$s = \sigma + j\omega_a = \frac{j2}{t_s}\tan(\omega_d/2) \rightarrow \begin{cases} \sigma = 0 \\ \omega_a = \dfrac{2}{t_s}\tan(\omega_d/2) \quad -\infty \le \omega_a \le \infty \end{cases}$$

$$\omega_d = 2\tan^{-1}\left(\frac{\omega_a t_s}{2}\right) \quad -\pi \le \omega_d \le \pi$$

  - Range of ω_d is $\pm\pi$, and dimensions of digital frequency ω_d are radians/sample
  - Range of ω_a is $\pm\infty$, and dimensions of analog frequency ω_a are radians/second

Bilinear Transform IIR Filter Design Method

$$\omega_d = 2\tan^{-1}\left(\frac{\omega_a t_s}{2}\right) \qquad -\pi \le \omega_d \le \pi$$

frequency warping due to bilinear transform

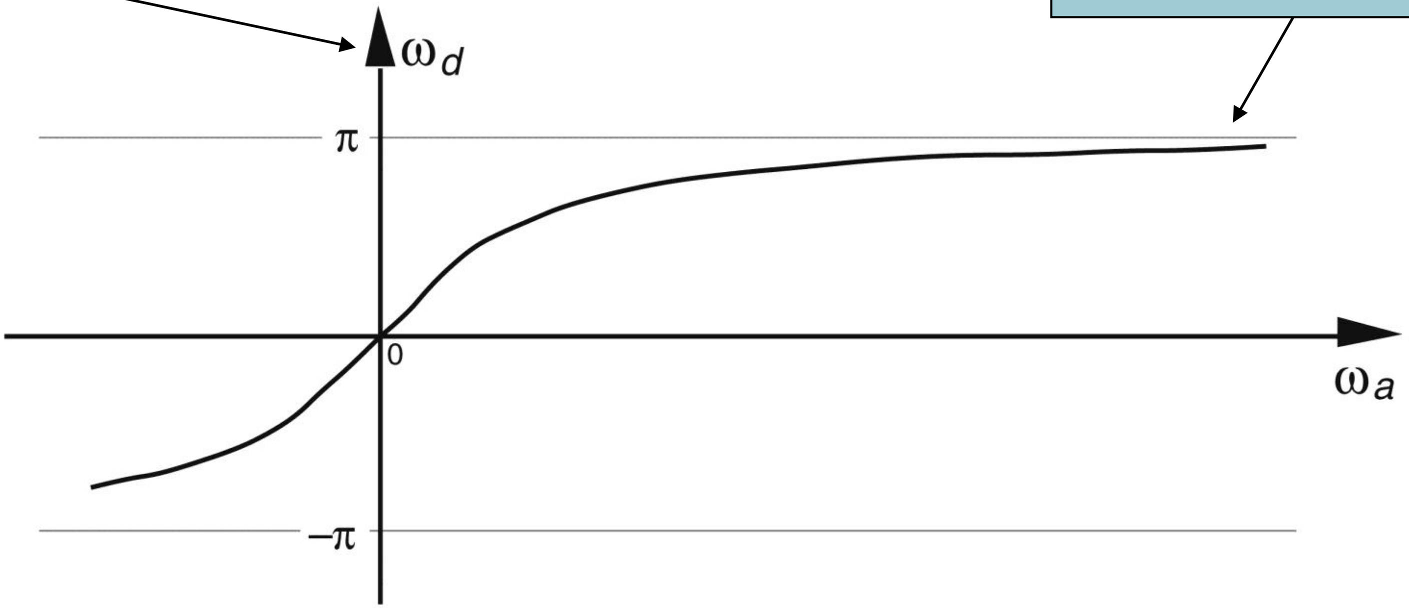


Figure 6–38 Nonlinear relationship between the *z*-domain frequency ω_d and the *s*-domain frequency ω_a.

no matter how large *s*-plane's analog ω_a becomes, *z*-plane's ω_d will never be greater than π radians/sample (f_s/2 Hz)

Bilinear Transform IIR Filter Design Method

bilinear transform maps *s*-plane's entire $j\omega_a$ axis onto the unit circle in *z*-plane

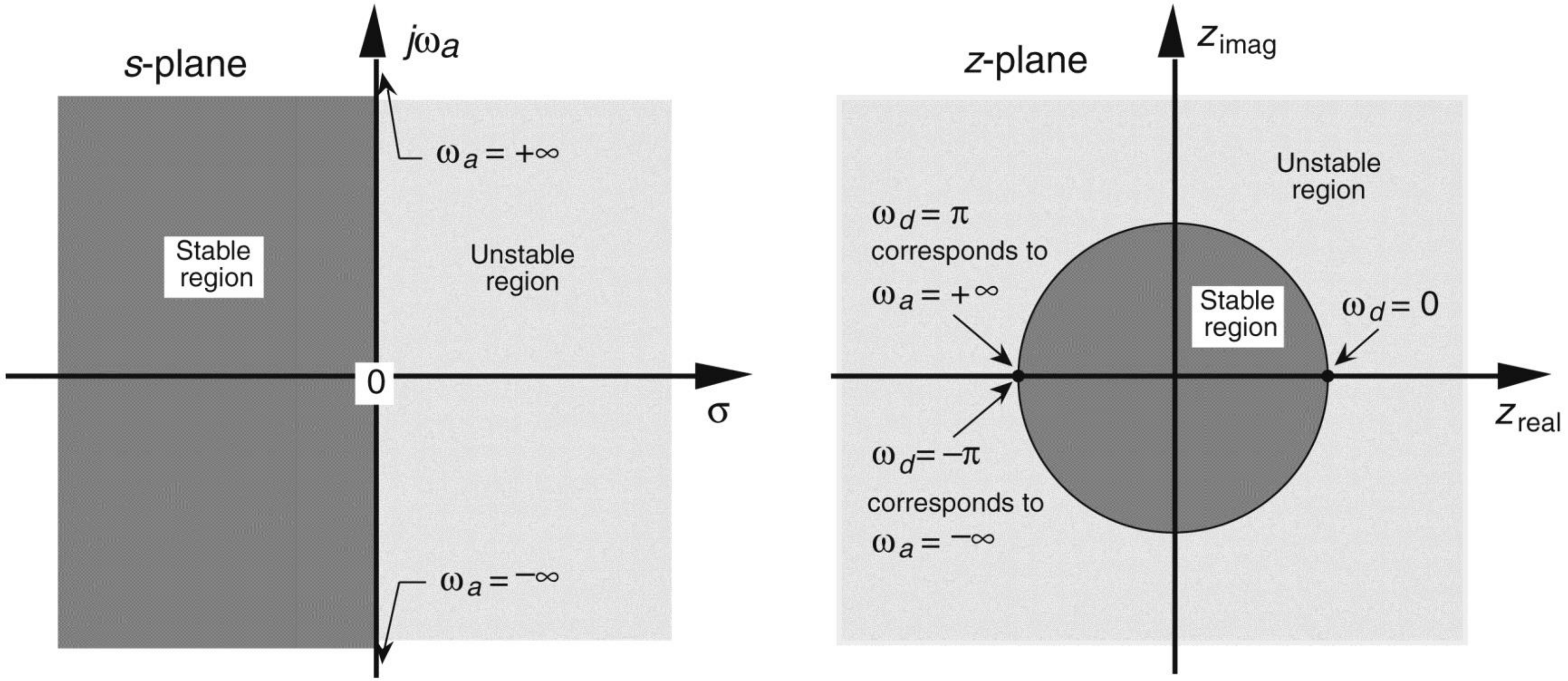


Figure 6–39 Bilinear transform mapping of the *s*-plane to the *z*-plane.

Bilinear Transform IIR Filter Design Method

- Frequency warping
  - $\omega_d = \pi$ radians/sample corresponds to $f_d = f_s/2$ Hz → $\omega_d = 2\pi(f_d / f_s)$

$$\omega_d = 2\tan^{-1}\left(\frac{\omega_a t_s}{2}\right) \xrightarrow[\omega_a = 2\pi f_a]{\omega_d = 2\pi(f_d / f_s)} f_d = \frac{f_s \tan^{-1}(\pi f_a / f_s)}{\pi} \text{ Hz}$$

$$\omega_a = \frac{2}{t_s}\tan\left(\frac{\omega_d}{2}\right) \xrightarrow[\omega_a = 2\pi f_a]{\omega_d = 2\pi(f_d / f_s)} f_a = \frac{f_s \tan(\pi f_d / f_s)}{\pi} \text{ Hz}$$

Bilinear Transform IIR Filter Design Method

$$f_d = \frac{f_s \tan^{-1}(\pi f_a / f_s)}{\pi} \text{ Hz}$$

f_d frequency warping (compression) becomes more severe as f_d approaches $f_s/2$

no IIR filter response aliasing can occur with bilinear transform design method

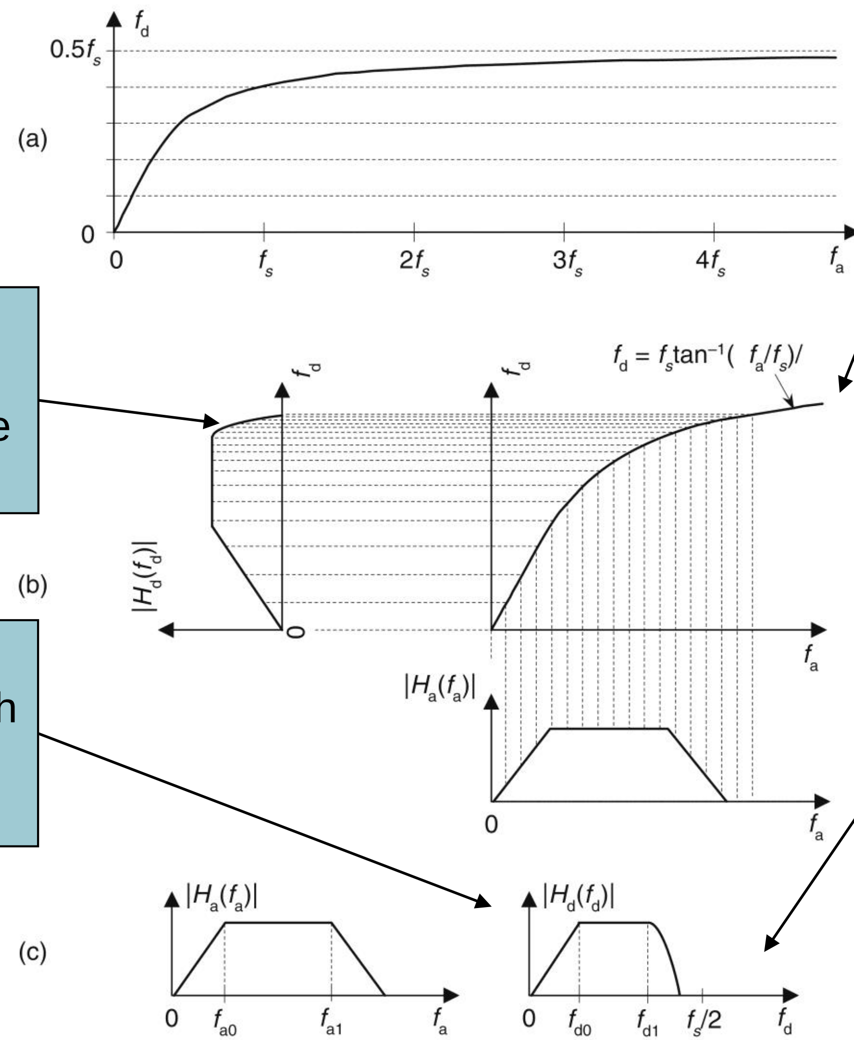


Figure 6-40 Nonlinear relationship between the f_d and f_a frequencies: (a) frequency warping curve; (b) s-domain frequency response transformation to a z-domain frequency response; (c) example $|H_a(f_a)|$ and transformed $|H_d(f_d)|$.

if a bilinear-transform-designed digital bandpass filter is desired to have an upper cutoff frequency of f_{d1} Hz, then the original prototype analog bandpass filter must be designed (prewarped) to have an upper cutoff frequency of f_{a1} Hz using:

$$f_a = \frac{f_s \tan(\pi f_d / f_s)}{\pi} \text{ Hz}$$

Bilinear Transform IIR Filter Design Method

- Steps to perform an IIR filter design using bilinear transform method
  - Step 1: Obtain Laplace transfer function $H_c(s)$ for the prototype analog filter
  - Step 2: Determine digital filter's equivalent f_s and establish $t_s = 1/f_s$
  - Step 3: In Laplace $H_c(s)$ transfer function, substitute the expression

    $$\frac{2}{t_s}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)$$

    for the variable s to get IIR filter's $H(z)$ transfer function

Bilinear Transform IIR Filter Design Method

- Step 4: Multiply numerator and denominator of $H(z)$ by appropriate power of $(1 + z^{-1})$ and collect terms of like powers of z in the form

$$H(z) = \frac{\sum_{k=0}^{N} b(k) z^{-k}}{1 - \sum_{k=1}^{M} a(k) z^{-k}}$$

- Step 5: By inspection, express IIR filter's time-domain equation in the general form of

$$\begin{aligned} y(n) = {} & b(0)x(n) + b(1)x(n-1) + b(2)x(n-2) + \ldots + b(N)x(n-N) \\ & + a(1)y(n-1) + a(2)y(n-2) + \ldots + a(M)y(n-M) \end{aligned}$$

  - Although this expression only applies to filter structure in Fig. 6-18, we can apply $a(k)$ and $b(k)$ coefficients to improved IIR structure shown in Fig. 6-22

Bilinear Transform IIR Filter Design Method

- Bilinear transform design example
  - Design an IIR filter that approximates 2nd-order Chebyshev prototype analog lowpass filter whose passband ripple is 1 dB
    - f_s = 100 Hz (t_s = 0.01)
    - Filter's 1 dB cutoff frequency = 20 Hz
    - Original prototype filter's Laplace transfer function is given as

$$H_c(s) = \frac{17410.145}{s^2 + 137.94536\, s + 17410.145}$$

Bilinear Transform IIR Filter Design Method

$$H_c(s) = \frac{17410.145}{s^2 + 137.94536\,s + 17410.145} \xrightarrow[c\,=\,17410.145]{b\,=\,137.94536} H_c(s) = \frac{c}{s^2 + b\,s + c}$$

$$\xrightarrow{s=\frac{2}{t_s}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)} H(z) = \frac{c}{\left(\frac{2}{t_s}\cdot\frac{1-z^{-1}}{1+z^{-1}}\right)^2 + b\,\frac{2}{t_s}\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + c}$$

$$\xrightarrow[a=2/t_s]{\text{to simplify:}} H(z) = \frac{c}{a^2\left(\frac{1-z^{-1}}{1+z^{-1}}\right)^2 + ab\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + c}$$

$$\xrightarrow[\text{denominator by }(1+z^{-1})^2]{\text{multiply numerator and}} H(z) = \frac{c(1+z^{-1})^2}{a^2(1-z^{-1})^2 + ab(1+z^{-1})(1-z^{-1}) + c(1+z^{-1})^2}$$

$$\xrightarrow{\text{collecting like powers of z}} H(z) = \frac{c(1+2z^{-1}+z^{-2})}{(a^2+ab+c)+(2c-2a^2)z^{-1}+(a^2+c-ab)z^{-2}}$$

Bilinear Transform IIR Filter Design Method

$$\xrightarrow[\text{divide numerator and denominator by } (a^2+ab+c)]{\text{to get a constant term of one in denominator,}} H(z) = \frac{\dfrac{c}{(a^2+ab+c)}(1+2z^{-1}+z^{-2})}{1+\dfrac{(2c-2a^2)}{(a^2+ab+c)}z^{-1}+\dfrac{(a^2+c-ab)}{(a^2+ab+c)}z^{-2}}$$

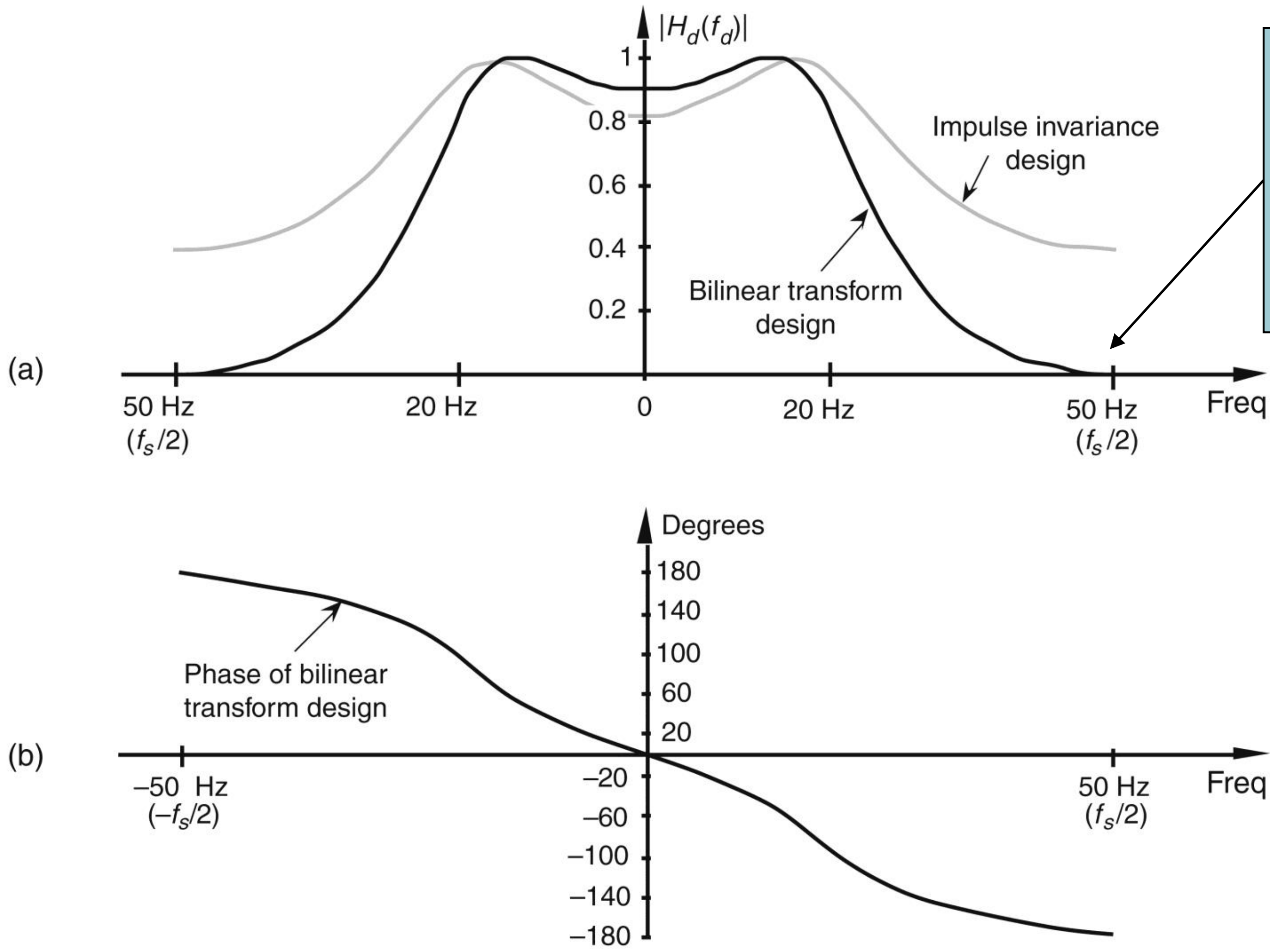
$$\xrightarrow[\substack{b=137.94536 \\ c=17410.145}]{a=2/t_s=200} H(z) = \frac{0.20482712(1+2z^{-1}+z^{-2})}{1-0.53153089z^{-1}+0.35083938z^{-2}}$$

$$H(z) = \frac{0.20482712+0.40965424z^{-1}+0.20482712z^{-2}}{1-0.53153089z^{-1}+0.35083938z^{-2}}$$

$$\xrightarrow{\text{time-domain expression for IIR filter}} y(n) = 0.20482712\cdot x(n)+0.40965424\cdot x(n-1)+0.20482712\cdot x(n-2)$$

$$+0.53153089\cdot y(n-1)-0.35083938\cdot y(n-2)$$

Bilinear Transform IIR Filter Design Method



bilinear-transform-designed filter's magnitude response approaches zero at folding frequency of $f_s/2$ = 50 Hz

Figure 6–41 Comparison of the bilinear transform and impulse invariance design IIR filters: (a) frequency magnitude responses; (b) phase of the bilinear transform IIR filter.

Bilinear Transform IIR Filter Design Method

bilinear transform design method gives a much sharper roll-off for our lowpass filter for two reasons: 1) frequency warping of bilinear transform method compresses (sharpens) roll-off portion of a lowpass filter; 2) the price we pay in terms of additional complexity of implementation of our IIR filter: our new filter requires five multiplications per filter output sample where impulse invariance design filter in Fig. 6-36(a) required only three multiplications

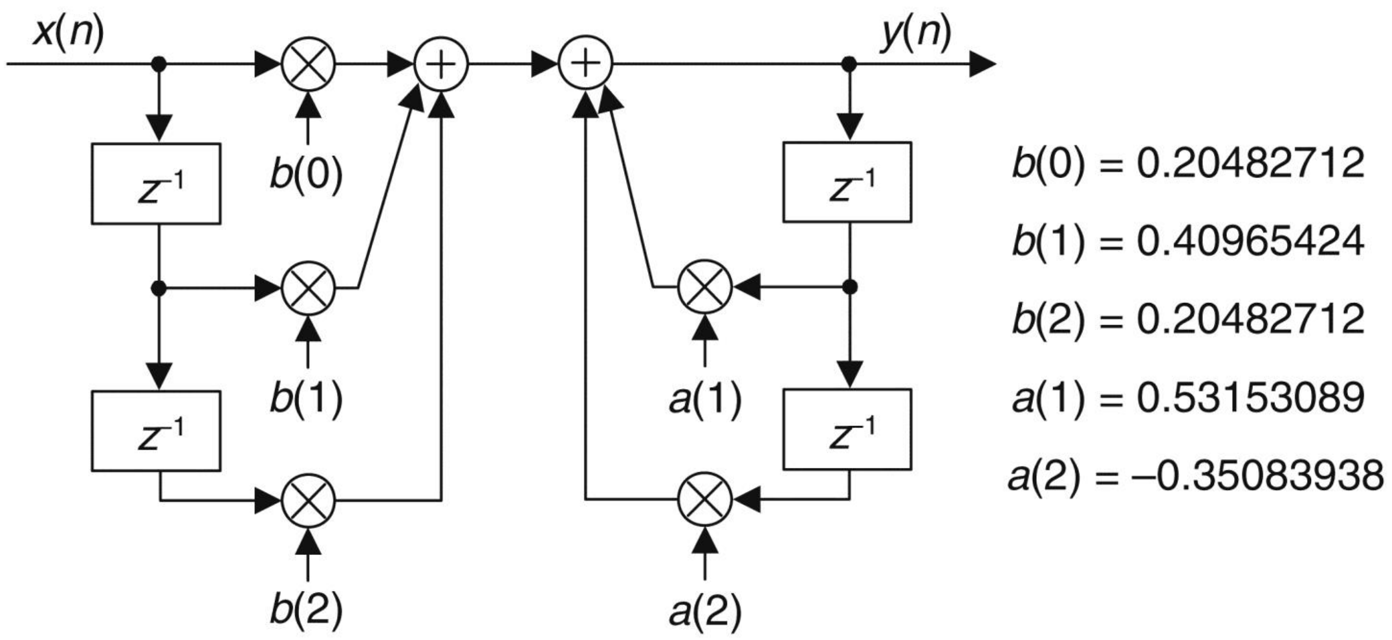


Figure 6–42 Implementation of the bilinear transform design example filter.

Bilinear Transform IIR Filter Design Method

- Prewarping
  - If cutoff frequency is a large percentage of f_s, resultant $|H_d(f_d)|$ cutoff frequency will be below the desired value
    - To avoid this, we *prewarp* prototype analog filter's cutoff frequency requirement before the analog $H_c(s)$ transfer function is derived in Step 1
      - In that way, they compensate for the bilinear transform's frequency warping before it happens
    - To determine prewarped analog filter lowpass cutoff frequency that we want mapped to the desired IIR lowpass cutoff frequency, use

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

we plug desired IIR cutoff frequency ω_d in here to calculate ω_a cutoff frequency used to derive prototype analog filter's $H_c(s)$ transfer function

Optimized IIR Filter Design Method

- Optimization methods
  - Developed for situation when the desired IIR filter frequency response is not of standard lowpass, bandpass, or highpass form
    - Closed-form expressions for filter's z-transform do not exist → no explicit equations to work with
    - Designer should describe, in some way, the desired IIR filter frequency response
    - The algorithm, then, assumes a filter transfer function $H(z)$ as a ratio of polynomials in z and starts to calculate filter's frequency response
    - Based on some error criteria, the algorithm iteratively adjusts filter's coefficients to minimize the error between desired and actual filter frequency response

Optimized IIR Filter Design Method

- Optimized IIR filter design routines
  - Are used to design the simpler lowpass, bandpass, or highpass forms even though analytical techniques exist
  - They only require the designer to specify a few key amplitude and frequency values, and the desired order of IIR filter (the number of feedback taps), and the software computes the final feedforward and feedback coefficients

Optimized IIR Filter Design Method

In specifying a lowpass IIR filter, a software design routine might require us to specify the values for δ_p, δ_s, f_1, and f_2

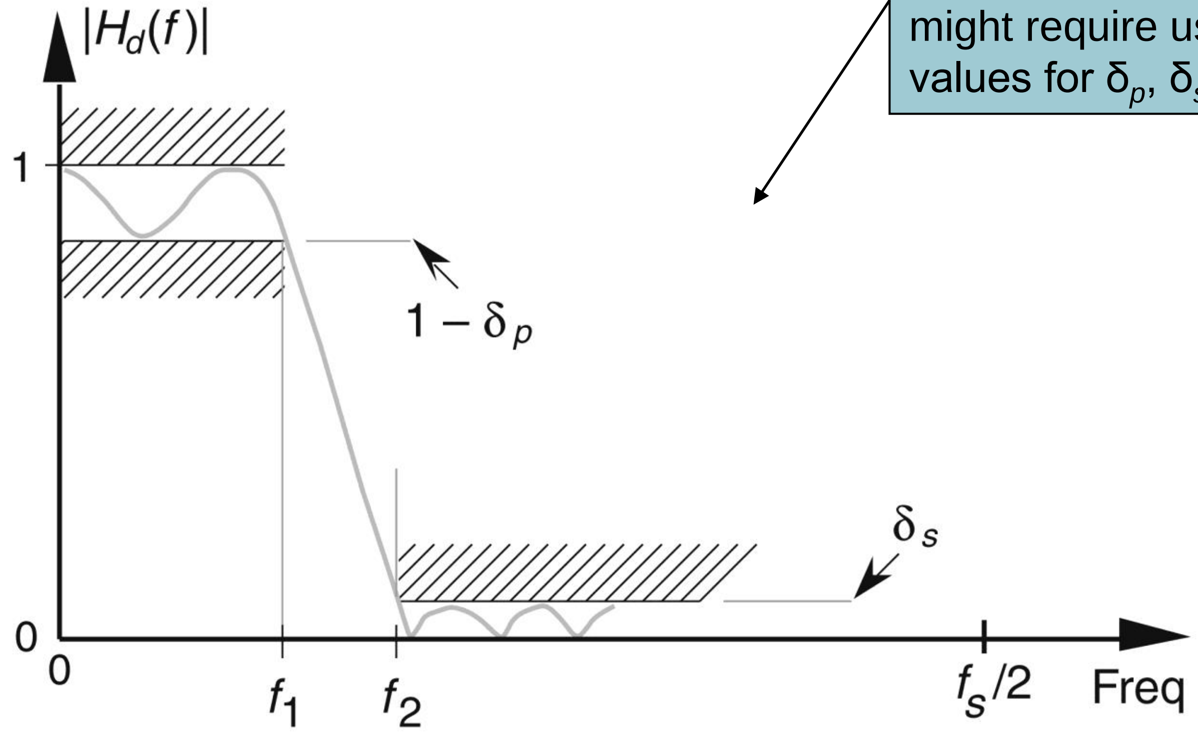


Figure 6-43 Example lowpass IIR filter design parameters required for the optimized IIR filter design method.

Comparison of IIR and FIR Filters

Characteristic	IIR	FIR (nonrecursive)
Number of necessary multiplications	Least	Most
Sensitivity to filter coefficient quantization	Can be high	Very low
Probability of overflow errors	Can be high	Very low
Stability	Must be designed in	Guaranteed
Linear phase	No	Guaranteed
Can simulate prototype analog filters	Yes	No
Required coefficient memory	Least	Most
Hardware filter control complexity	Moderate	Simple
Availability of design software	Good	Very good
Ease of design, or complexity of design software	Moderately complicated	Simple
Difficulty of quantization noise analysis	Most complicated	Least complicated
Supports adaptive filtering	With difficulty	Yes

So long as FIR coefficients are symmetrical (or antisymmetrical)