Lecture 28: Design of IIR Filters

Foundations of Digital Signal Processing

Outline

- **Designing IIR Filters with Discrete Differentiation**
- Designing IIR Filters with Impulse Invariance
- Designing IIR Filters with the Bilinear Transform
- Related Analog Filters

IIR Filter Design from Derivatives

- **Designing IIR Filters**
  - No easy ways to design digital IIR filters
  - So let us start from analog filters

IIR Filter Design from Derivatives

- **Designing IIR Filters**
  - No easy ways to design digital IIR filters
  - So let us start from analog filters

- **Option 1: Preserve the difference equation!**

IIR Filter Design from Derivatives

- **Question: What is a derivative in discrete-time?**
  - In continuous-time

$$\frac{dx(t)}{dt} \rightarrow sX(s)$$

  - In discrete-time

IIR Filter Design from Derivatives

- **Question: What is a derivative in discrete-time?**
  - In continuous-time

$$\frac{dx(t)}{dt} \rightarrow sX(s)$$

  - In discrete-time

$$\frac{dx(t)}{dt} = \lim_{\Delta T \rightarrow 0} \frac{x(t) - x(t - \Delta T)}{\Delta T}$$

T=1

$$\left.\frac{dx(t)}{dt}\right|_{t=nT} = \frac{x(nT) - x(nT - T)}{T} = x[n] - x[n-1]$$

IIR Filter Design from Derivatives

- **Question: What is a derivative in discrete-time?**
  - In continuous-time

$$\frac{dx(t)}{dt} \to sX(s)$$

  - In discrete-time

$$\frac{dx(t)}{dt} = \lim_{\Delta T \to 0} \frac{x(t) - x(t - \Delta T)}{\Delta T}$$

$$\left.\frac{dx(t)}{dt}\right|_{t=nT} = \frac{x(nT) - x(nT - T)}{T} = \frac{1}{T}(x[n] - x[n-1])$$

IIR Filter Design from Derivatives

- **Question: What is a derivative in discrete-time?**
  - In continuous-time

$$\frac{dx(t)}{dt} \rightarrow sX(s)$$

  - In discrete-time

$$\frac{dx(t)}{dt} = \lim_{\Delta T \rightarrow 0} \frac{x(t) - x(t - \Delta T)}{\Delta T}$$

$$\left.\frac{dx(t)}{dt}\right|_{t=nT} = \frac{x(nT) - x(nT - T)}{T} = \frac{1}{T}(x[n] - x[n-1])$$

$$\left.\frac{dx(t)}{dt}\right|_{t=nT} \rightarrow \frac{1}{T}(1 - z^{-1})X(z)$$

IIR Filter Design from Derivatives

- **Question: What is a second-derivative in discrete-time?**
  - In continuous-time

$$\frac{d^2x(t)}{dt^2} \rightarrow s^2X(s)$$

  - In discrete-time

$$\frac{d^2x(t)}{dt^2} = \frac{dx(t)}{dt}\left[\frac{dx(t)}{dt}\right]$$

$$\left.\frac{dx(t)}{dt}\right|_{t=nT} = \frac{x(nT) - x(nT - T)}{T}$$

IIR Filter Design from Derivatives

- **Question: What is a second-derivative in discrete-time?**
  - In continuous-time

$$\frac{d^2x(t)}{dt^2} \rightarrow s^2X(s)$$

  - In discrete-time

$$\frac{d^2x(t)}{dt^2} = \frac{dx(t)}{dt}\left[\frac{dx(t)}{dt}\right]$$

$$\left.\frac{d^2x(t)}{dt^2}\right|_{t=nT} = \frac{[x(nT) - x(nT-T)]/T - [x(nT-T) - x(nT-2T)]/T}{T}$$

IIR Filter Design from Derivatives

- **Question: What is a second-derivative in discrete-time?**
  - In continuous-time

$$\frac{d^2x(t)}{dt^2} \rightarrow s^2X(s)$$

  - In discrete-time

$$\frac{d^2x(t)}{dt^2} = \frac{dx(t)}{dt}\left[\frac{dx(t)}{dt}\right]$$

$$\left.\frac{d^2x(t)}{dt^2}\right|_{t=nT} = \frac{x(nT) - 2x(nT-T) + x(nT-2T)}{T^2} \rightarrow \frac{x[n] - 2x[n-1] + x[n-2]}{T^2}$$

IIR Filter Design from Derivatives

- **Question: What is a second-derivative in discrete-time?**
  - In continuous-time

$$\frac{d^2x(t)}{dt^2} \to s^2X(s)$$

  - In discrete-time

$$\frac{d^2x(t)}{dt^2} = \frac{dx(t)}{dt}\left[\frac{dx(t)}{dt}\right]$$

$$\left.\frac{d^2x(t)}{dt^2}\right|_{t=nT} = \frac{x(nT) - 2x(nT-T) + x(nT-2T)}{T^2} \to \frac{x[n] - 2x[n-1] + x[n-2]}{T^2}$$

$$\left.\frac{dx(t)}{dt}\right|_{t=nT} \to \frac{1}{T^2}(1 - 2z^{-1} + z^{-2})X(z) = \frac{1}{T^2}(1 - z^{-1})^2X(z)$$

IIR Filter Design from Derivatives

- **Question: What is a derivative in discrete-time?**
  - Translate continuous-time to discrete-time

$$\frac{d^k x(t)}{dt^k} \to s^k X(s)$$

$$\left.\frac{d^k x(t)}{dt^k}\right|_{t=nT} \to \frac{1}{T}(1 - z^{-1})^k X(z)$$

IIR Filter Design from Derivatives

- **Question: What is a derivative in discrete-time?**
  - Translate continuous-time to discrete-time

$$\frac{d^k x(t)}{dt^k} \rightarrow s^k X(s)$$

$$\left.\frac{d^k x(t)}{dt^k}\right|_{t=nT} \rightarrow \frac{1}{T}(1 - z^{-1})^k X(z)$$

$$\boxed{s \rightarrow \frac{1}{T}(1 - z^{-1})}$$

IIR Filter Design from Derivatives

- **Example:** $s \rightarrow \frac{1}{T}(1 - z^{-1})$
  - Use the derivative conversion to transform the following bi-quad filter into the discrete-time domain.

$$H(s) = \frac{1}{(s + 0.1)^2 + 9}$$

IIR Filter Design from Derivatives

- **Example:** $s \rightarrow \frac{1}{T}(1 - z^{-1})$
  - Use the derivative conversion to transform the following bi-quad filter into the discrete-time domain.

$$H(s) = \frac{1}{(s + 0.1)^2 + 9}$$

$$H(z) = \frac{1}{\left(\frac{1}{T}(1 - z^{-1}) + 0.1\right)^2 + 9} = \frac{T^2}{T^2\left[\left(\frac{1}{T}(1 - z^{-1}) + 0.1\right)^2 + 9\right]}$$

$$= \frac{T^2}{\left((1 - z^{-1}) + 0.1T\right)^2 + 9T^2} = \frac{T^2}{\left((1 + 0.1T) - z^{-1}\right)^2 + 9T^2}$$

IIR Filter Design from Derivatives

- **Example:** $s \to \frac{1}{T}(1 - z^{-1})$
  - Use the derivative conversion to transform the following bi-quad filter into the discrete-time domain.

$$H(s) = \frac{1}{(s + 0.1)^2 + 9}$$

$$H(z) = \frac{T^2}{\left((1 + 0.1T) - z^{-1}\right)^2 + 9T^2}$$

Finding poles

$$\left((1 + 0.1T) - z^{-1}\right)^2 + 9T^2 = 0$$

$$\left((1 + 0.1T) - z^{-1}\right)^2 = -9T^2$$

$$(1 + 0.1T) - z^{-1} = \pm 3Tj$$

$$z^{-1} = (1 + 0.1T) \mp 3Tj$$

$$z = \frac{1}{1 + (0.1 \mp 3j)T}$$

IIR Filter Design from Derivatives

- **Example:** $s \rightarrow \frac{1}{T}(1 - z^{-1})$
  - Use the derivative conversion to transform the following bi-quad filter into the discrete-time domain.

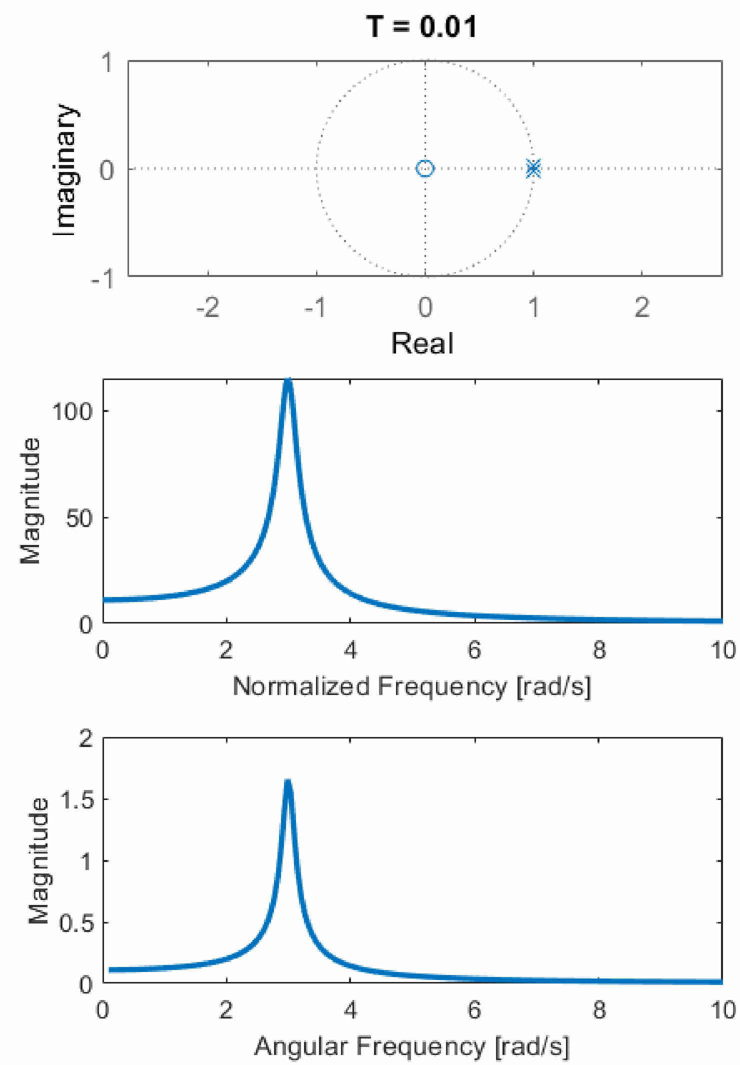
$$z = \frac{1}{1 + (0.1 + 3j)T}$$

Poles

IIR Filter Design from Derivatives

- **Example:** $s \rightarrow \frac{1}{T}(1 - z^{-1})$

$$z = \frac{1}{1 + (0.1 + 3j)T}$$

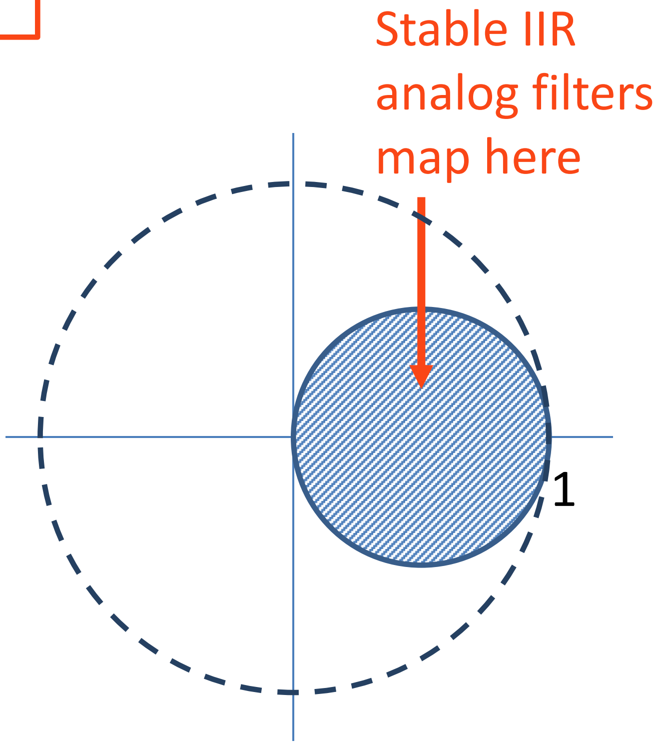


IIR Filter Design from Derivatives

- **Question: What is a derivative in discrete-time?**
  - Translate continuous-time to discrete-time

$$\boxed{s \rightarrow \frac{1}{T}\left(1 - z^{-1}\right)}$$

- **Pros:**
  - Relatively simple
- **Cons:**
  - Very limiting
  - Stable continuous-time poles can only be mapped to low frequencies



Lecture 19: Design of IIR Filters

Foundations of Digital Signal Processing

Outline

- Designing IIR Filters with Discrete Differentiation
- **Designing IIR Filters with Impulse Invariance**
- Designing IIR Filters with the Bilinear Transform
- Related Analog Filters

IIR Filter Design by Impulse Invariance

- **Designing IIR Filters**
  - No easy ways to design digital IIR filters
  - So let us start from analog filters

- **Option 2: Preserve the impulse response!**

IIR Filter Design by Impulse Invariance

- **Question:** **How else can I represent my transfer function?**

Partial-fraction decomposition

$$H(s) = \sum_{k=1}^{K} \frac{c_k}{s - p_k}$$

IIR Filter Design by Impulse Invariance

- **Question: How else can I represent my transfer function?**

$$H(s) = \sum_{k=1}^{K} \frac{c_k}{s - p_k}$$

$$h(t) = \sum_{k=1}^{K} c_k e^{p_k t}$$

IIR Filter Design by Impulse Invariance

- **Question: How else can I represent my transfer function?**

$$H(s) = \sum_{k=1}^{K} \frac{c_k}{s - p_k}$$

$$h(t) = \sum_{k=1}^{K} c_k e^{p_k t}$$

$$h(nT) = h[n] = \sum_{k=1}^{K} c_k e^{p_k nT} = \sum_{k=1}^{K} c_k [e^{p_k T}]^n$$

IIR Filter Design by Impulse Invariance

- **Question: How else can I represent my transfer function?**

$$H(s) = \sum_{k=1}^{K} \frac{c_k}{s - p_k} = \sum_{k=1}^{K} c_k e^{p_k t}$$

$$h(t) = \sum_{k=1}^{K} c_k e^{p_k t}$$

$$h(nT) = h[n] = \sum_{k=1}^{K} c_k e^{p_k nT} = \sum_{k=1}^{K} c_k [e^{p_k T}]^n$$

$$H(z) = \sum_{k=1}^{K} \frac{c_k}{1 - e^{p_k T} z^{-1}}$$

IIR Filter Design from Derivatives

- **Example:** $H(z) = \sum_{k=1}^{K} \frac{c_k}{1 - e^{p_k T} z^{-1}}$
  - Use impulse invariance to transform the following biquad filter into the discrete-time domain.

$$H(s) = \frac{1}{(s + 0.1)^2 + 9}$$

IIR Filter Design from Derivatives

- **Example:** $H(z) = \sum_{k=1}^{K} \frac{c_k}{1-e^{p_k T} z^{-1}}$
  - Use impulse invariance to transform the following biquad filter into the discrete-time domain.

$$H(s) = \frac{1}{(s+0.1)^2 + 9}$$

Poles:

$$(s+0.1)^2 + 9 = 0$$

$$s = \pm 3j - 0.1$$

IIR Filter Design from Derivatives

- **Example:** $H(z) = \sum_{k=1}^{K} \frac{c_k}{1-e^{p_k T} z^{-1}}$
  - Use impulse invariance to transform the following biquad filter into the discrete-time domain.

$$H(s) = \frac{1}{(s+0.1)^2+9} = \frac{1/2}{s+3j+0.1} + \frac{1/2}{s-3j+0.1}$$

Poles:

$$(s+0.1)^2+9=0$$

$$s = \pm 3j - 0.1$$

IIR Filter Design from Derivatives

- **Example:** $H(z) = \sum_{k=1}^{K} \frac{c_k}{1-e^{p_k T} z^{-1}}$
  - Use impulse invariance to transform the following biquad filter into the discrete-time domain.

$$H(s) = \frac{1}{(s+0.1)^2+9} = \frac{1/2}{s+3j+0.1} + \frac{1/2}{s-3j+0.1}$$

$$p_k = \pm 3j - 0.1$$

$$H(s) \rightarrow h(t) \rightarrow h(nT) \rightarrow H(z)$$

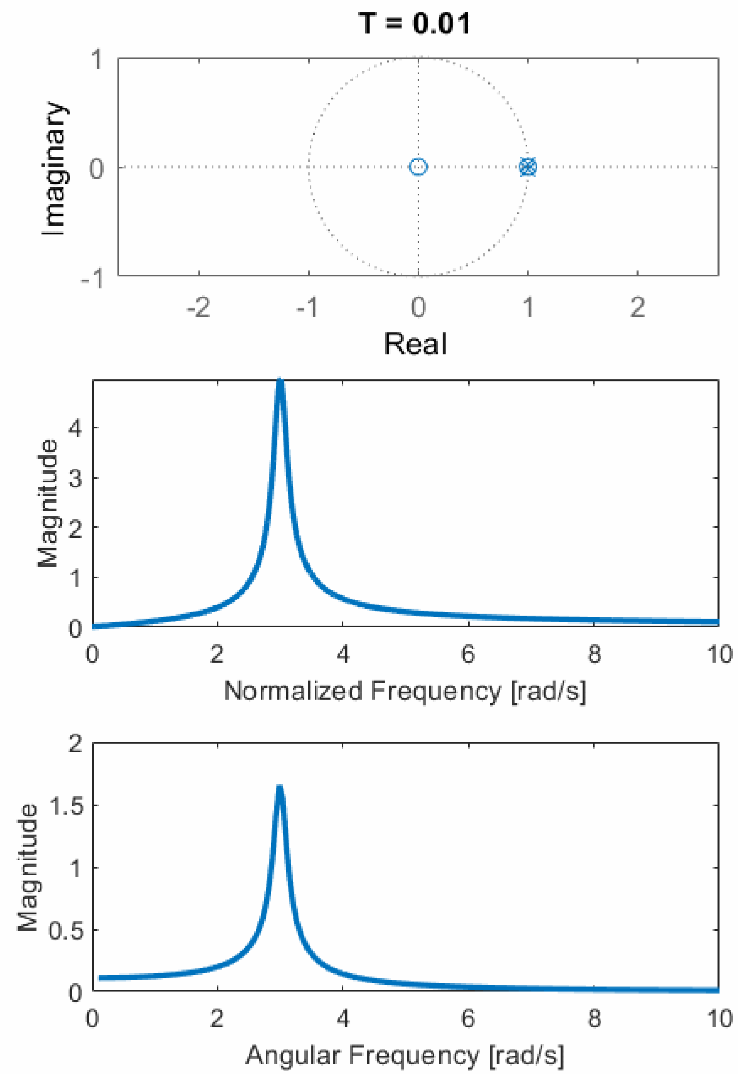
$$H(z) = \frac{1/2}{1-e^{(-3j-0.1)T} z^{-1}} + \frac{1/2}{1-e^{(3j-0.1)T} z^{-1}}$$

p_k

IIR Filter Design from Derivatives

- **Example:** $H(z) = \sum_{k=1}^{K} \frac{c_k}{1-e^{p_k T} z^{-1}}$

$$H(z) = \frac{1/2}{1 - e^{(-3j-0.1)T} z^{-1}} + \frac{1/2}{1 - e^{(3j-0.1)T} z^{-1}}$$



Lecture 19: Design of IIR Filters

Foundations of Digital Signal Processing

Outline

- Designing IIR Filters with Discrete Differentiation
- Designing IIR Filters with Impulse Invariance
- **Designing IIR Filters with the Bilinear Transform**
- Related Analog Filters

IIR Filter Design by Bilinear Transform

- **Designing IIR Filters**
  - No easy ways to design digital IIR filters
  - So let us start from analog filters

- **Option 3: Preserve the definition of z!**

IIR Filter Design by Bilinear Transform

- **Question: How are z and s related?**
  - From continuous-time to discrete-time

$$s = j\Omega$$

$$e^{st} = e^{snT} = z^n$$

$$z = e^{sT}$$

Taylor Series Expansion / Approximation

  - Building an approximation ($e^x \approx 1 + x$)

$$z = \frac{e^{\frac{sT}{2}}}{e^{-\frac{sT}{2}}} \approx \frac{1 + sT/2}{1 - sT/2}$$

IIR Filter Design by Bilinear Transform

- **Question: How are z and s related?**
  - From continuous-time to discrete-time

$$s = j\Omega$$

$$e^{st} = e^{snT} = z^n$$

$$z = e^{sT} \qquad\qquad s = \frac{1}{T}\ln(z)$$

Bilinear
Expansion / Approximation

  - Building an approximation

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

IIR Filter Design by Bilinear Transform

■ The Bilinear Transform

Continuous-time to discrete-time

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

Discrete-time to continuous-time

$$z \to \frac{1 + sT/2}{1 - sT/2}$$

IIR Filter Design by Bilinear Transform

- **Example:** $s \rightarrow \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}, \quad z \rightarrow \frac{1+sT/2}{1-sT/2}$
  - Use the bilinear transform to transform the following biquad filter into the discrete-time domain.

$$H(s) = \frac{1}{(s+0.1)^2+9}$$

IIR Filter Design by Bilinear Transform

- **Example:** $s \to \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}, \quad z \to \frac{1+sT/2}{1-sT/2}$
  - Use the bilinear transform to transform the following biquad filter into the discrete-time domain.

$$H(s) = \frac{1}{(s+0.1)^2 + 9}$$

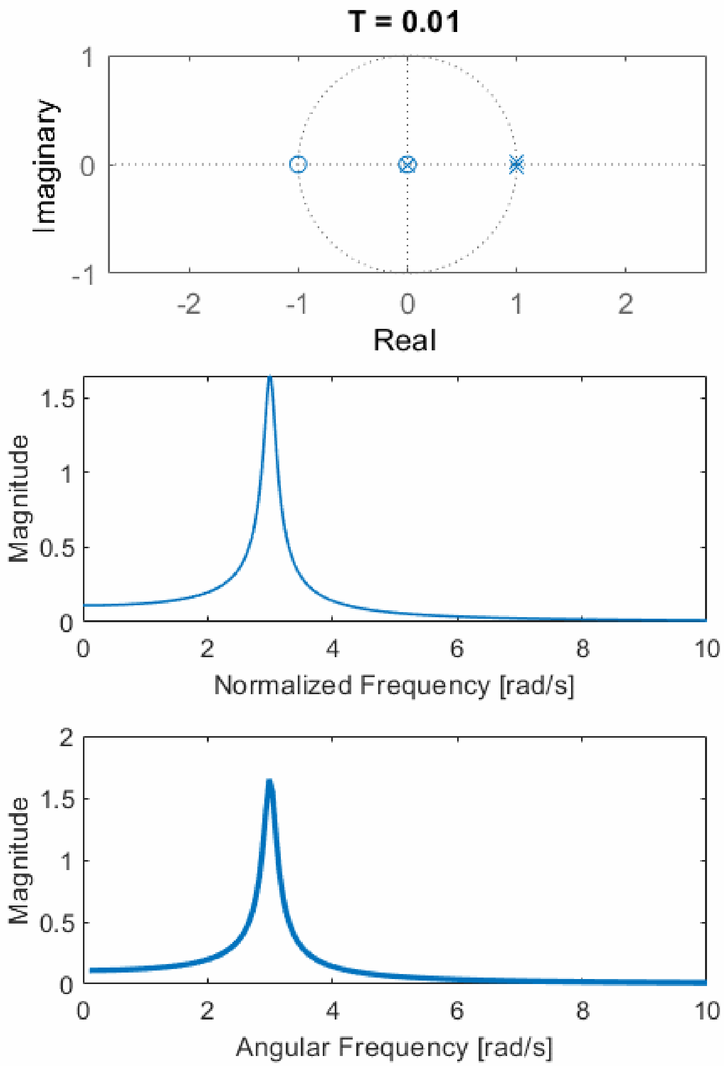
$$H(z) = \frac{1}{\left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}} + 0.1\right)^2 + 9}$$

$$= \frac{(1+z^{-1})^2}{\left(\frac{2}{T}(1-z^{-1}) + 0.1\right)^2 + 9(1+z^{-1})^2}$$

IIR Filter Design by Bilinear Transform

- **Example:** $s \rightarrow \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}, \quad z \rightarrow \frac{1+sT/2}{1-sT/2}$

$$H(z) = \frac{(1+z^{-1})^2}{\left(\frac{2}{T}(1-z^{-1})+0.1\right)^2 + 9(1+z^{-1})^2}$$



Lecture 19: Design of IIR Filters

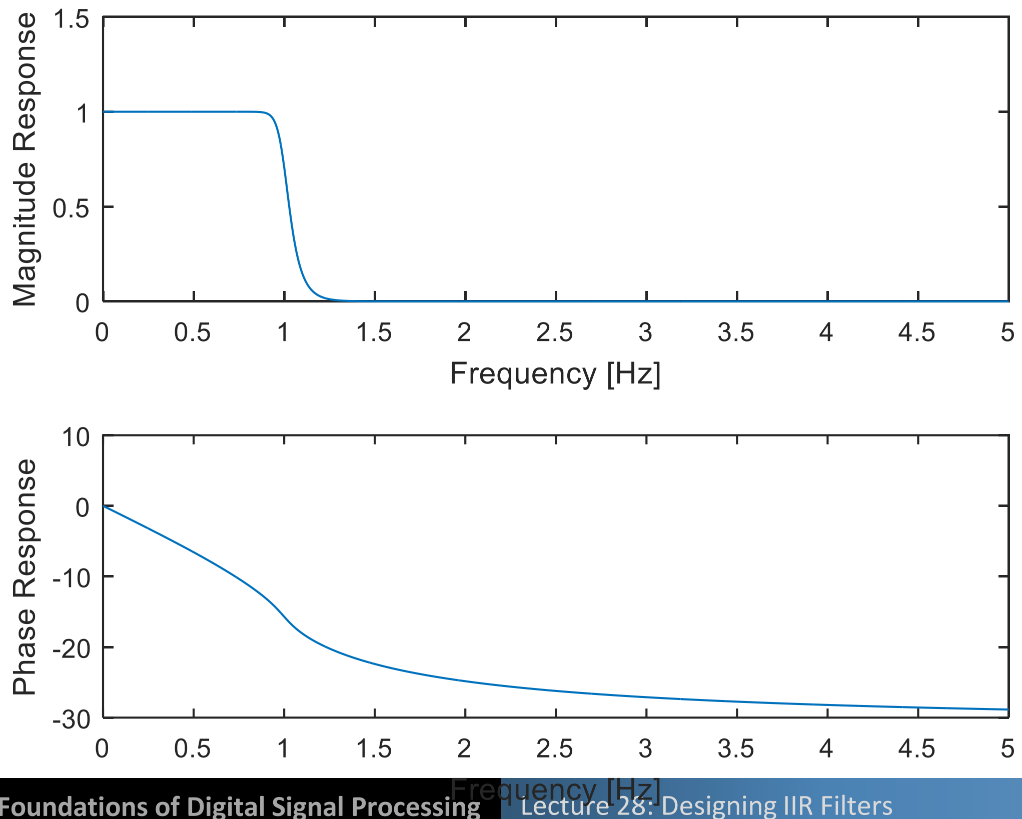
Foundations of Digital Signal Processing

Outline

- Designing IIR Filters with Discrete Differentiation
- Designing IIR Filters with Impulse Invariance
- Designing IIR Filters with the Bilinear Transform
- **Related Analog Filters**

Multi-pole Filters

- **Butterworth:** Maximally flat passband
- **Chebyshev:** Faster cutoff with passband ripple
- **Elliptic:** Fastest cutoff with passband and stopband ripple



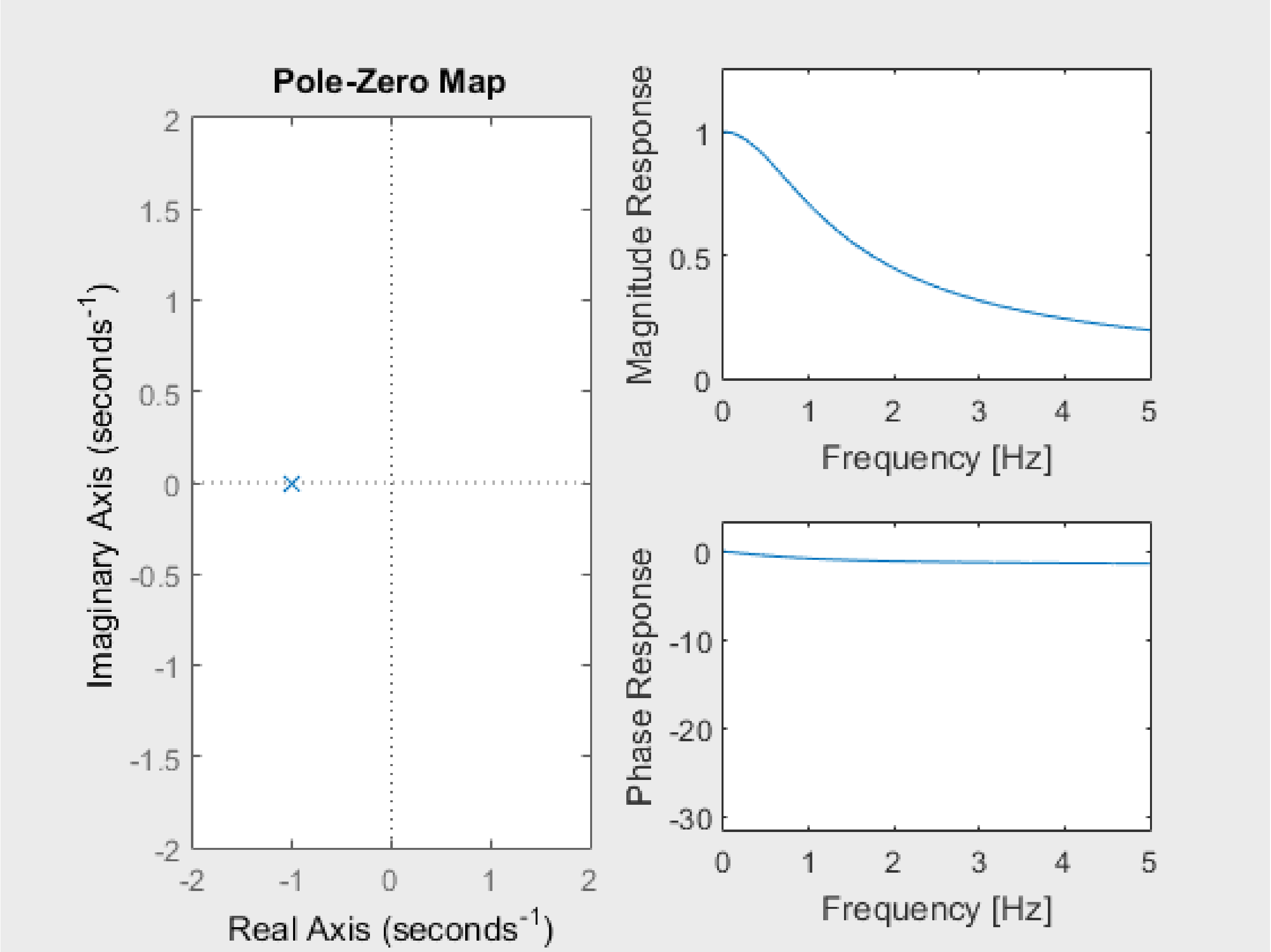
Butterworth Filter

- **Butterworth Filter of order N**

$$|H(j\omega)| = \frac{1}{\sum_{k=1}^{N}(s - s_k)} \qquad s_k = e^{\frac{j(2k+N-1)\pi}{2N}}$$

Butterworth Filter

- **Butterworth Filter of order N**
  - N equally spaced poles on a circle on the left-hand-side of the s-plane



Butterworth Filter

- **Properties of the Butterworth Filter**
  - It is maximally flat at $\omega = 0$
  - It has a cutoff frequency $|H(\omega)| = \frac{1}{\sqrt{2}}$ at $\omega = \omega_c$
  - For large n, it becomes an ideal filter

Chebyshev Filter

- **Chebyshev Filter of order N**

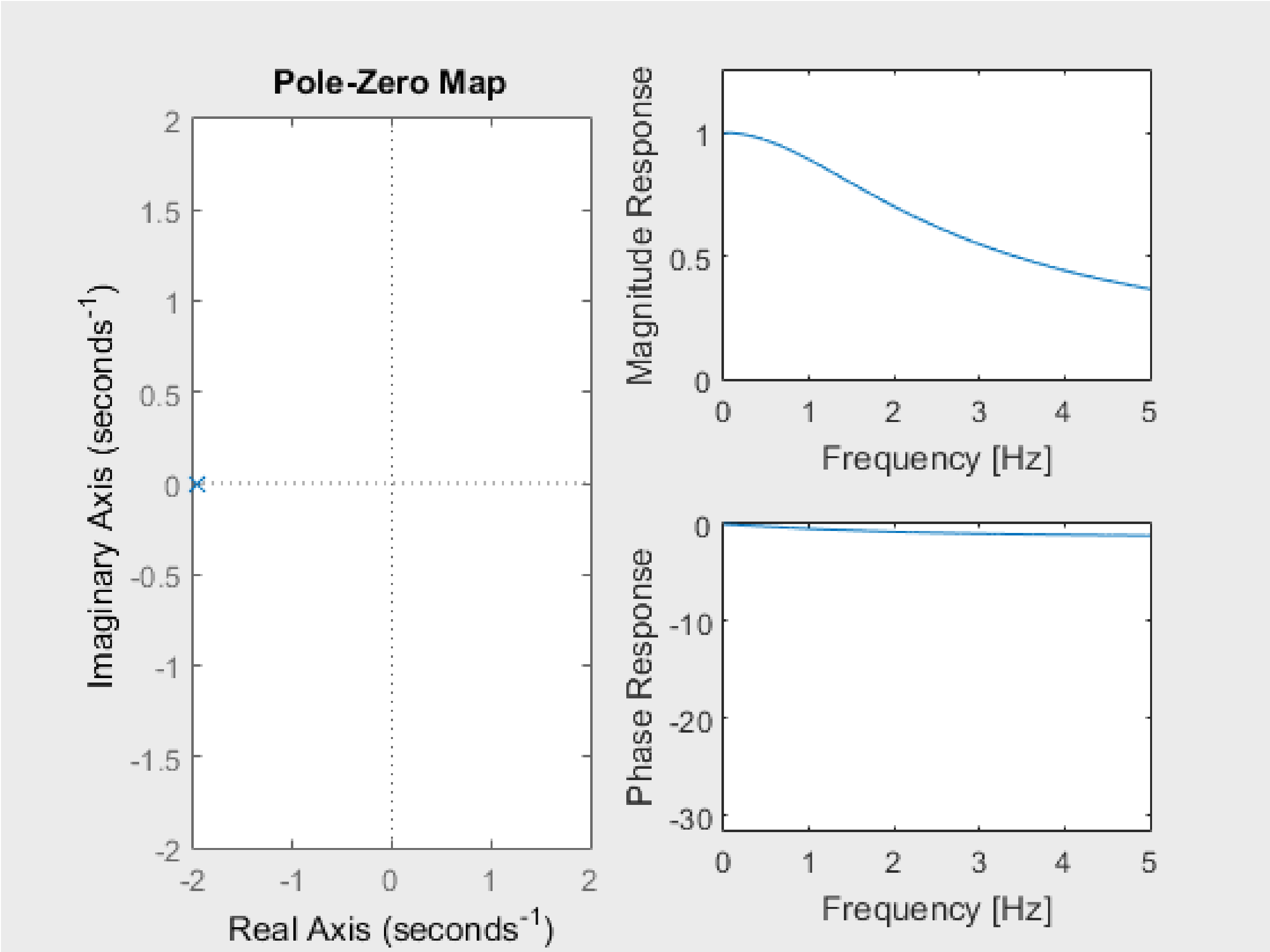
$$|H(j\omega)| = \frac{1}{\sqrt{1+\epsilon^2 C_n^2(\omega)}}$$

$C_n^2(\omega)$ is an nth-order Chebyshev Polynomial

ϵ^2 controls ripple

Chebyshev Filter

- **Chebyshev Filter of order N**



Chebyshev Filter

- **Properties of the Chebyshev Filter**
  - It has ripples in the passband and is smooth in the stopband.
  - The ratio between the maximum and minimum ripples in the passband is

$$(1 + \epsilon^2) - 1/2$$

  - If ϵ is reduced (i.e., the ripple size is reduced), then the stopband attenuation is reduced.
  - It has a sharper cut-off than a Butterworth filter, but at the expense of passband rippling

Elliptic Filter

- **Elliptic Filter of order N**

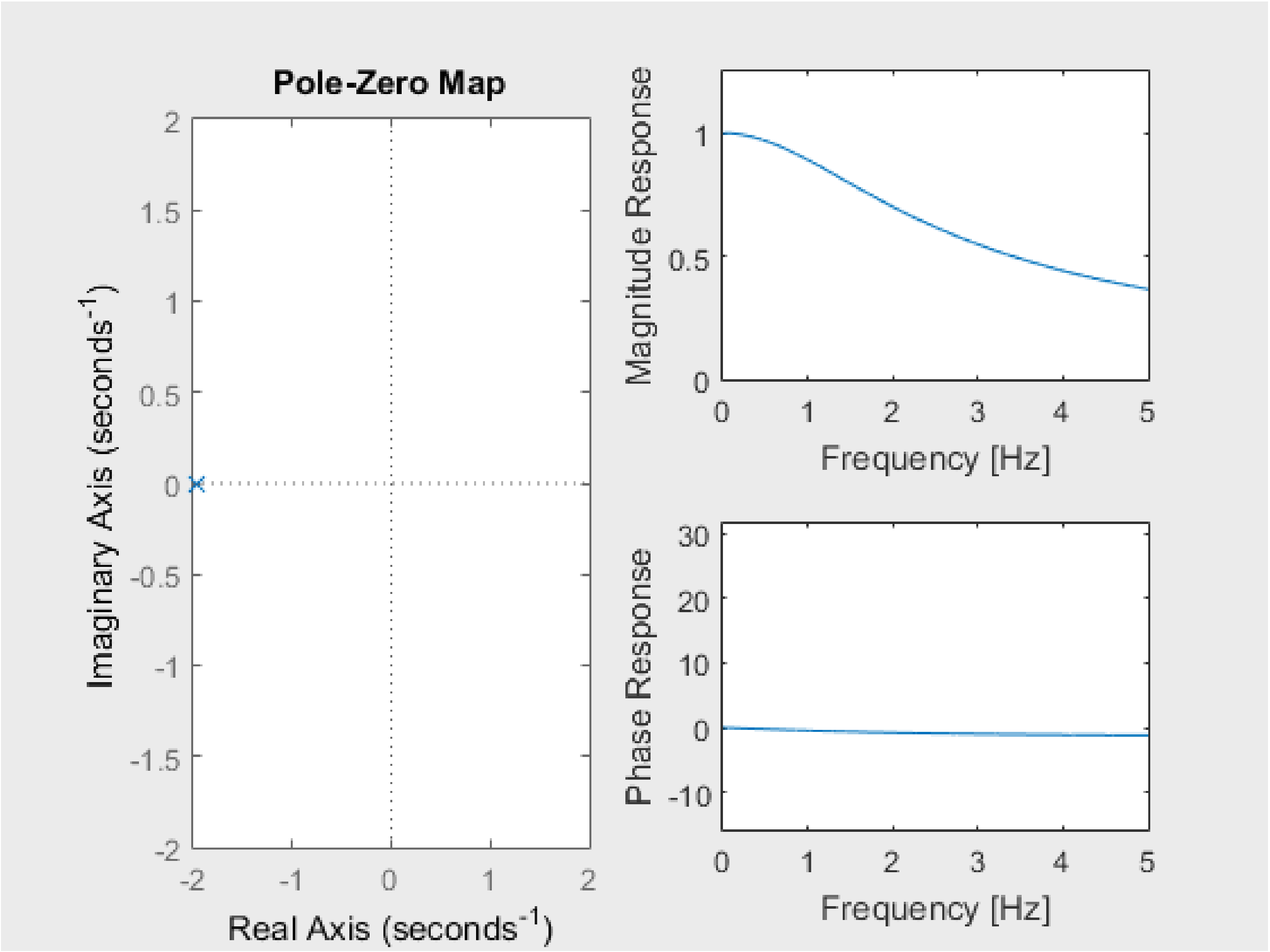
$$|H(j\omega)| = \frac{1}{\sqrt{1 + \epsilon^2 R_n^2(\omega)}}$$

$R_n^2(\omega)$ is an nth-order elliptic function

ϵ^2 controls ripple

Elliptic Filter

- **Elliptic Filter of order N**



Elliptic Filter

- **Properties of the Elliptic Filter**
  - It has ripples in the passband and the stopband
  - The ratio between the maximum and minimum ripples is larger than the Chebyshev filter, but it has an even quicker transition from passband to stopband
  - It has poles and zeros, but they are much more difficult to compute compared with the Butterworth and Chebyshev filters