Basics of Digital Filters

Elena Punskaya
www-sigproc.eng.cam.ac.uk/~op205

Some material adapted from courses by
Prof. Simon Godsill, Dr. Arnaud Doucet,
Dr. Malcolm Macleod and Prof. Peter Rayner

What is a Digital Filter?

Digital Filter: numerical procedure or algorithm that transforms a given sequence of numbers into a second sequence that has some more desirable properties.

$$x_n \xrightarrow{\text{Input sequence}} \boxed{\textbf{Digital Filter}} \xrightarrow{\text{Output sequence}} y_n$$

Desired features

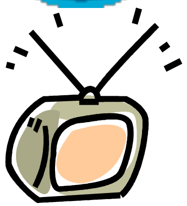
Desired features depend on the application, for example

Input Signal		Output
generated by sensing device (microphone)	→	having less noise or interferences
speech	→	with reduced redundancy for more efficient transmission

Examples of filtering operations

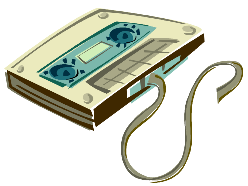
Noise suppression

- received radio signals



- signals received by image sensors (TV, infrared imaging devices)

- electrical signals measured from human body (brain heart, neurological signals)



- signals recorded on analog media such as analog magnetic tapes

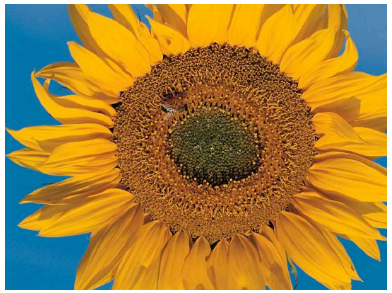
Examples of filtering operations

Enhancement of selected frequency ranges

- treble and bass control or graphic equalizers

increase sound level and high and low level frequencies to compensate for the lower sensitivity of the ear at those frequencies or for special sound effects

- enhancement of edges in images

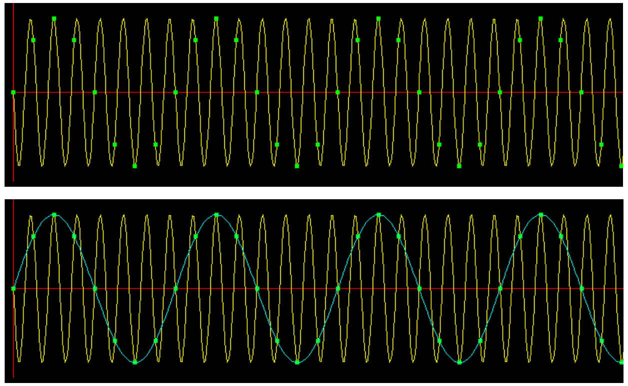


improve recognition of object (by human or computer)

edge – a sharp transition in the image brightness, sharp transitions in a signal (from Fourier theory) appear as high-frequency components which can be amplified

Examples of filtering operations

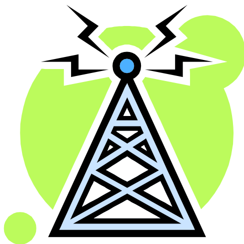
Bandwidth limiting



- means of aliasing prevention in sampling

- communication

  radio or TV signal transmitted over specific channel has to have a limited bandwidth to prevent interference with neighbouring channels

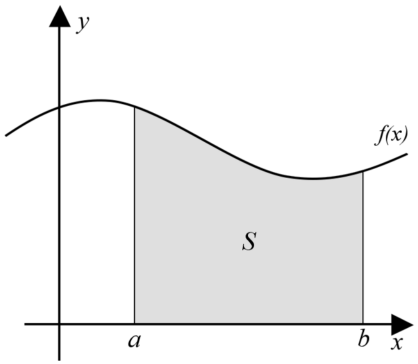


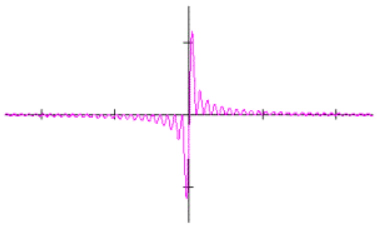
frequency components outside the permitted band are attenuated below a specific power level

Examples of filtering operations

Specific operations

$$\frac{dy}{dx} = 3x^2$$





- differentiation
- integration
- Hilbert transform

These operations can be approximated by digital filters operating on the sampled input signal

Linear time-invariant (LTI) digital filters.

We limit ourselves to LTI digital filters only.

Time-invariant:

$$x_n \longrightarrow \boxed{\textbf{LTI}} \longrightarrow y_n \quad \Longrightarrow \quad x_{n-k} \longrightarrow \boxed{\textbf{LTI}} \longrightarrow y_{n-k}$$

Linear: defined by the principle of <u>linear superposition</u>

$$x_{n1} \longrightarrow \boxed{\textbf{LTI}} \longrightarrow y_{n1} \qquad x_{n2} \longrightarrow \boxed{\textbf{LTI}} \longrightarrow y_{n2}$$

scaling & additivity properties

$$\Longrightarrow \quad a_1 x_{n1} + a_2 x_{n2} \longrightarrow \boxed{\textbf{LTI}} \longrightarrow a_1 y_{n1} + a_2 y_{n2}$$

If linear system's parameters are constant

$\Longrightarrow$ it is **Linear Time Invariant**

Analysis

We analyse DSP algorithms by determining:

- their *time-domain* characteristics
  - linear difference equations
  - filter’s unit-sample (impulse) response
- their *frequency-domain* characteristics
  - more general, Z-transform domain
    - system transfer function
    - poles and zeros diagram in the z-plane
  - Fourier domain
    - frequency response
    - spectrum of the signal

First method in time domain: Linear difference equations

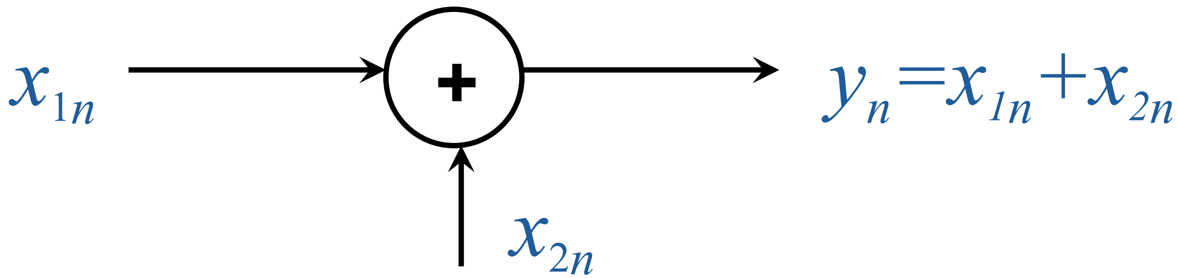
The linear time-invariant digital filter can then be described by the **linear difference equation**:

$$
\begin{aligned}
y_n &= -a_1 y_{n-1} - a_2 y_{n-2} - \ldots - a_N y_{n-N} + b_0 x_n + \ldots + b_M x_{n-M} \\
&= -\sum_{k=1}^{N} a_k y_{n-k} + \sum_{k=0}^{M} b_k x_{n-k}
\end{aligned}
$$

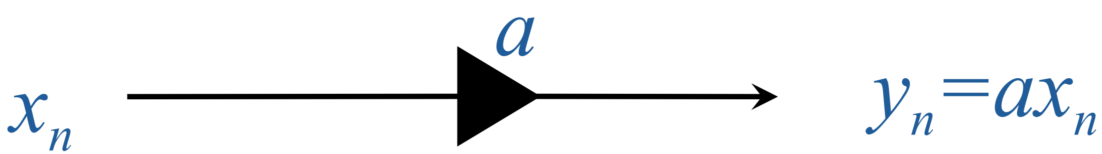
where $\{a_k\}$ and $\{b_k\}$ real

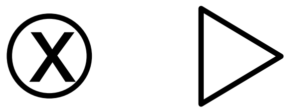
The *order* of the filter is the larger of M or N

Adders:



Multipliers:





alternative for multiplier

Simple components implemented in the arithmetic logic unit of the computer

Elements of a Digital Filter – Delays

Positive delay ("delay"):

stores the current value for one sample interval

$$x_n \xrightarrow{\;Z^{-1}\;} y_n = x_{n-1}$$

alternative for delay

Negative delay ("advance"):

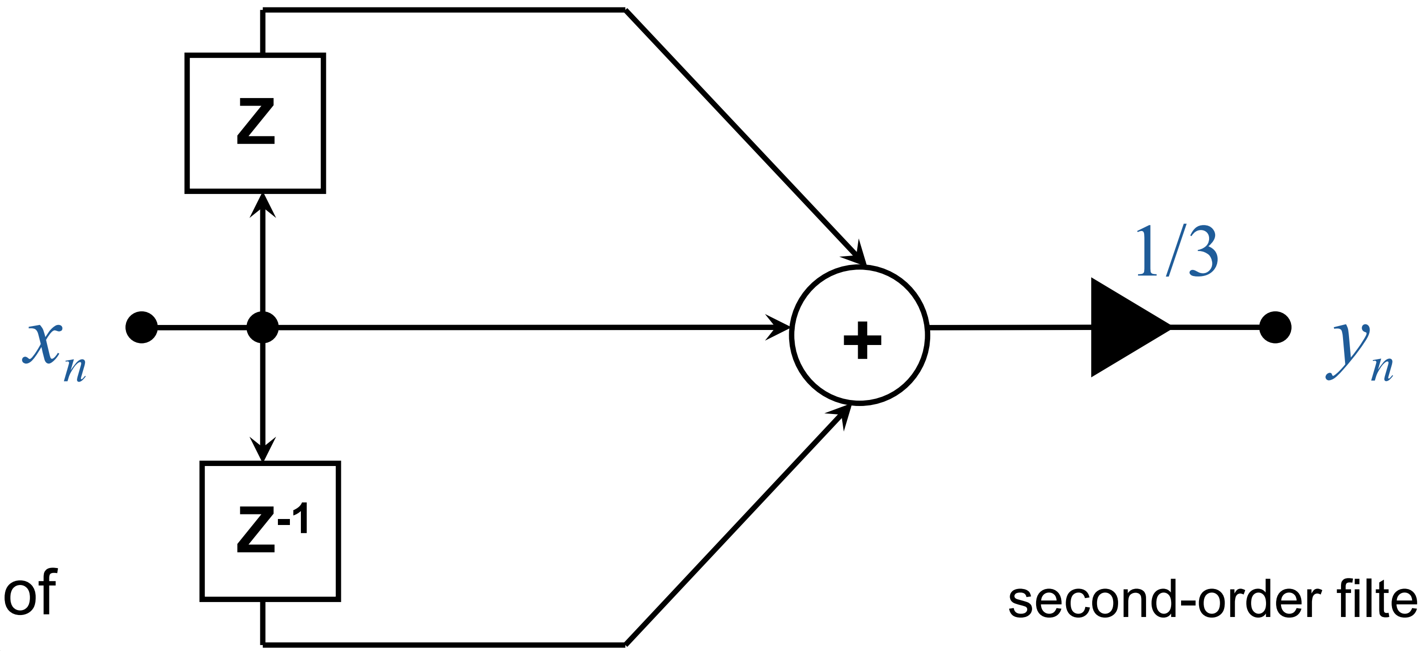
allows to look ahead, e.g. image processing

$$x_n \xrightarrow{\;Z\;} y_n = x_{n+1}$$

Components that allow access to future and past values in the sequence

Example: three-sample averager

Three-sample averager: $y_n = (x_{n+1} + x_n + x_{n-1})/3$



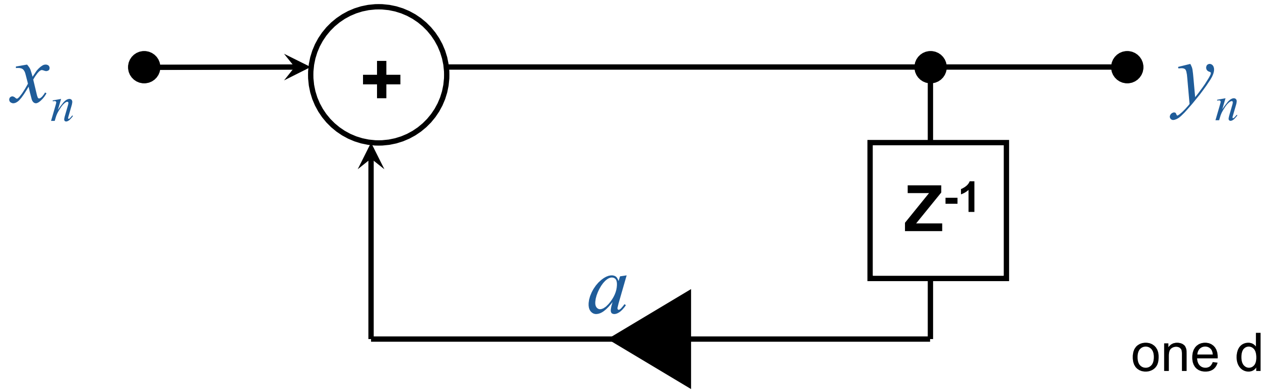
second-order filter

Order: number of delays & advances needed to implement

Nonrecursive filter – output is a function of only the input sequence

Example: first-order recursive filter

First-order recursive filter: $y_n = a y_{n-1} + x_n$



one delay

first-order

Example of feedback implemented in a digital filter

Recursive filter – output is also a function of the previous output

Full set of possible linear operations

The operations shown in the slides above are the **full set** of possible linear operations:

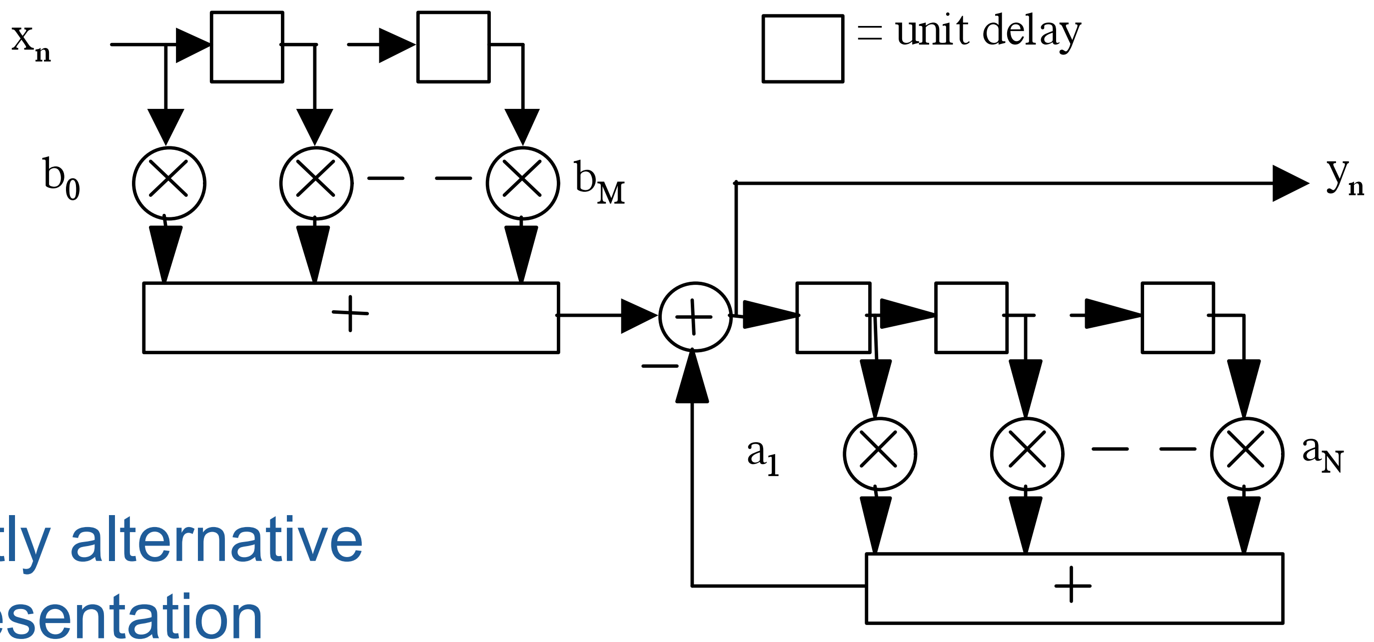
- constant delays (by any number of samples)
- addition or subtraction of signal paths
- multiplication (scaling) of signal paths by constants - (including -1)

Any other operations make the system non-linear.

Linear difference equations and digital filter structure

- Useful for implementing digital filter structures

$$y_n = -\sum_{k=1}^{N} a_k y_{n-k} + \sum_{k=0}^{M} b_k x_{n-k}$$



slightly alternative representation

Second method in time domain: unit-sample response

• Input signal is resolved into a weighted sum of elementary signal components, i.e. sum of unit samples or impulses

$$\boxed{x_n = \sum_{k=-\infty}^{\infty} x_k \,\delta_{n-k}}$$

• The response of the system to the unit sample sequence is determined

$$\delta_n \longrightarrow \boxed{\textbf{LTI}} \longrightarrow h_n$$

• Taking into account properties of the LTI system, the response of the system to x_n is the corresponding sum of weighted outputs

$$x_n = \sum_{k=-\infty}^{\infty} x_k \,\delta_{n-k} \longrightarrow \boxed{\textbf{LTI}} \longrightarrow y_n = \sum_{k=-\infty}^{\infty} x_k \,h_{n-k}$$

Linear convolution

Linear convolution

$$y_n = \sum_{k=-\infty}^{\infty} x_k h_{n-k}$$

gives the response of the LTI system as a function of the input signal and the unit sample (impulse) response

LTI is completely characterized by h_n

Causal LTI system

Causal system: output at time n depends only on present and past inputs but not on future

Impulse response:

$$\boxed{h_n = 0 \qquad \text{for} \quad n < 0}$$

Thus

$$\boxed{Y_n = \sum_{k=-\infty}^{\infty} x_k h_{n-k} = \sum_{k=0}^{\infty} h_k x_{n-k}}$$

Essentials of the z-transform

For a discrete time signal

For convenience we will use this notation ($x(n)$) for discrete signal as well as x_n from now on

$$x(n) \xleftrightarrow{z} X(z)$$

$$\boxed{X(z) = \sum_{n=-\infty}^{\infty} x(n)\, z^{-n}}$$

where z is a complex variable

For **causal** signals $X(z)$ is well defined for $|z| > r$

Essentials of the z-transform - convolution

Convolution

$$x(n) = \sum_{k=-\infty}^{\infty} x_1(k)\, x_2(n-k)$$

$$= x_1(n) * x_2(n)$$

admits a *z*-transform satisfying

$$\boxed{X(z) = X_1(z)\, X_2(z)}$$

where

$$x_1(n) \overset{z}{\longleftrightarrow} X_1(z)$$

$$x_2(n) \overset{z}{\longleftrightarrow} X_2(z)$$

Indeed,

$$X(z) = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x_1(k)\, x_2(n-k) \underbrace{z^{-n}}_{z^{-(n-k)}z^{-k}} = X_1(z)\, X_2(z)$$

Transfer function of LTI

Linear difference equation

$$\boxed{y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)}$$

Now, take z-transforms term by term

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

Rearranging, transfer function of the filter is

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

FIR and IIR filters

Transfer function of the filter is

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

Finite Impulse Response (FIR) Filters:

N = 0, no feedback

Infinite Impulse Response (IIR) Filters

Poles and Zeros

The roots of the numerator polynomial in *H(z)* are known as the **zeros,** and the roots of the denominator polynomial as **poles.** In particular, factorize *H(z)* top and bottom:

$$H(z) = b_0 \frac{\Pi_{q=1}^{M}(1 - c_q z^{-1})}{\Pi_{q=1}^{N}(1 - d_q z^{-1})}$$

Then, $\{c_q\}$ are the zeros and $\{d_q\}$ are the poles of the system.

Clearly all the poles $\{d_q\}$ must lie within the unit circle for filter stability, as for any discrete time system.

Linear Digital Filter in Matlab

Matlab has a filter command for implementation of linear digital filters.

Matlab transfer function: $$H(z) = \frac{\sum_{k=0}^{n_b+1} b_k z^{-k}}{a_0 + \sum_{k=1}^{n_a+1} a_k z^{-k}} \left(= \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \right)$$

The format is

```
y = filter( b, a, x);
```

where

$b = [b_0\ b_1\ b_2\ \ldots\ b_M\ ];$ $\qquad a = [\ 1\ \ a_1\ a_2\ a_3\ \ \ldots\ \ a_N\ ];$

So to compute the first P+1 samples of the filter's impulse response,

```
y = filter( b, a, [1 zeros(1,P)]);
```

Or step response,

```
y = filter( b, a, [ones(1,P)]);
```

z-transform and DTFT

DTFT

$$X(\omega) = X(z)\big|_{z=e^{j\omega}}$$

$$= \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n}$$

Similarly, to z-transforms a convolution theorem holds:

$$x_1(n) * x_2(n) \overset{F}{\longleftrightarrow} X(\omega) = X_1(\omega)\, X_2(\omega)$$

Frequency Response of LTI

LTI system with impulse response h_k

$$x_n \longrightarrow \boxed{\textbf{LTI}} \longrightarrow y_n = \sum_{k=0}^{\infty} h_k x_{n-k}$$

convolution

Fourier Transform

$$Y(\omega) = \underbrace{H(\omega)}_{\text{Fourier transform of } h_k} X(\omega)$$

$$\boxed{H(\omega) = H(z)\big|_{z=e^{j\omega}}}$$ - frequency response of LTI

Frequency Response of LTI

Indeed, let us excite the system with the complex exponential

$$x(n) = Ae^{j\omega_0 n} \qquad \text{where} \quad |\omega_0| < \pi$$

The response of the system to complex exponential

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{\infty} h(k)\, x(n-k) \\ &= \sum_{k=-\infty}^{\infty} h(k)\, Ae^{j\omega_0 (n-k)} = Ae^{j\omega_0 n} \overbrace{\sum_{k=-\infty}^{\infty} h(k)\, e^{-j\omega_0 k}}^{\text{Fourier transform of } h_k} \\ &= AH(\omega_0)\, e^{j\omega_0 n}. \end{aligned}$$

is also in the form of complex exponential but altered by the multiplicative factor $H(\omega_0)$

Example

Assume

$$h\left(n\right) = u\left(n\right) - u\left(n-3\right)$$

where $u\left(n\right) = 1$ for $n \geq 0$

0 otherwise

Frequency response

$$\begin{aligned} H\left(\omega\right) &= \sum_{n=-\infty}^{\infty} h\left(k\right) e^{-j\omega} \\ &= \sum_{n=0}^{2} e^{-jn\omega} = 1 + e^{-j\omega} + e^{-j2\omega} = e^{-j\omega}\left(1 + 2\cos\omega\right) \end{aligned}$$

Frequency Response of LTI

By knowing $H(\omega_0)$ we can determine the response of the system to any sinusoidal input signal, hence it specifies the response of the system in the frequency domain

$$\boxed{H(\omega) = |H(\omega)|\, e^{j\angle H(\omega)}}$$

$|H(\omega)|$ is called magnitude response of a system

$\angle H(\omega)$ is called phase response of a system

Frequency response of LTI

Transfer function:

$$H(z) = b_0 \frac{\Pi_{q=1}^{M}(1 - c_q z^{-1})}{\Pi_{q=1}^{N}(1 - d_q z^{-1})}$$

Frequency response:

$$H(e^{j\omega}) = b_0 \frac{\Pi_{q=1}^{M}(1 - c_q e^{-j\omega})}{\Pi_{q=1}^{N}(1 - d_q e^{-j\omega})} = b_0 \frac{e^{-jM\omega}\Pi_{q=1}^{M}(e^{j\omega} - c_q)}{e^{-jN\omega}\Pi_{q=1}^{N}(e^{j\omega} - d_q)}$$

The complex modulus:

$$|H(e^{j\omega})| = b_0 \frac{\Pi_{q=1}^{M}|e^{j\omega} - c_q|}{\Pi_{q=1}^{N}|e^{j\omega} - d_q|}$$

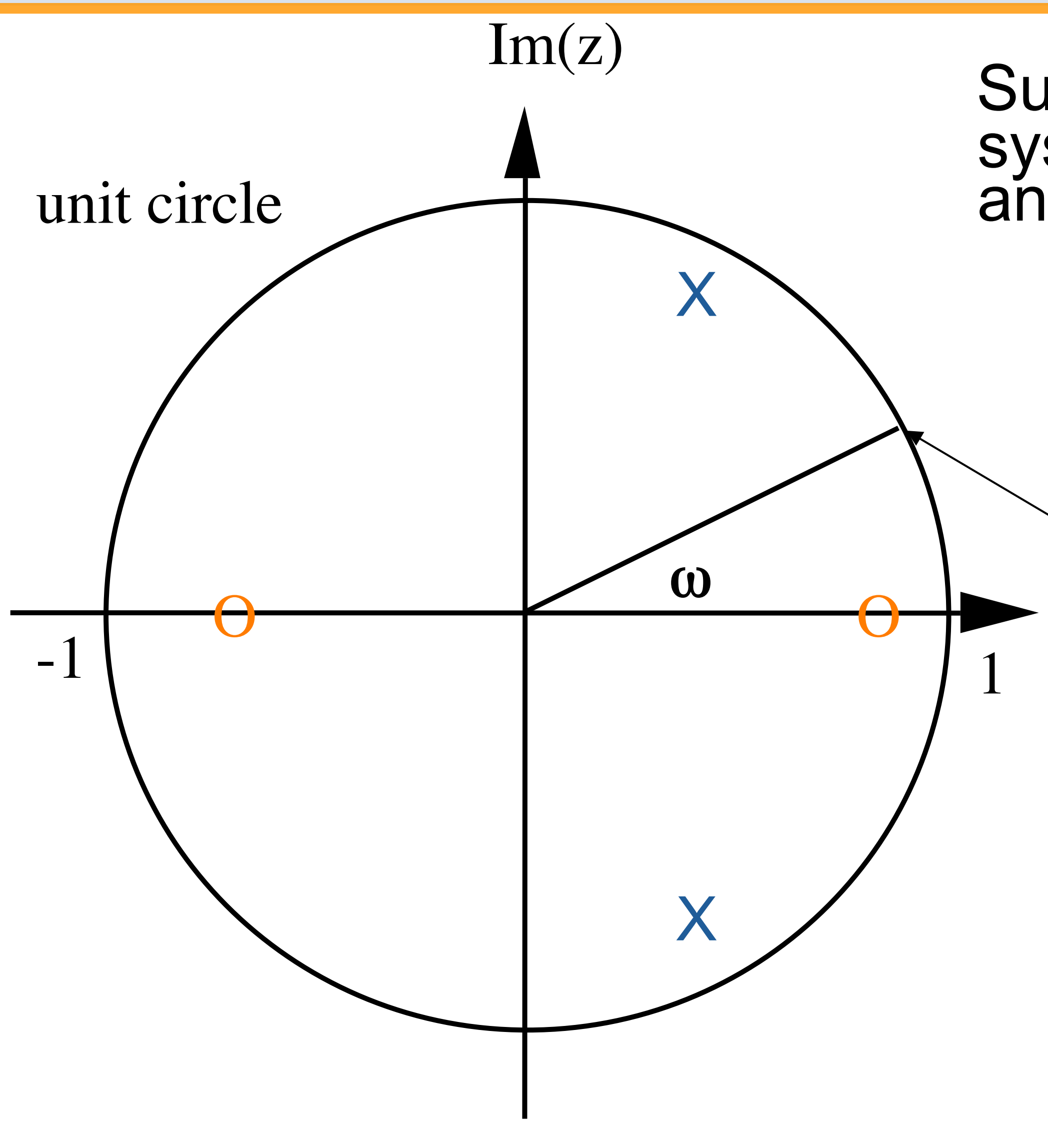
$$= b_0 \frac{\prod_{q=1}^{M} \text{Distance from } e^{j\omega} \text{ to } c_q}{\prod_{q=1}^{N} \text{Distance from } e^{j\omega} \text{ to } d_q}$$

and argument:

$$\angle(H(e^{j\omega})) = \underbrace{\omega(N - M)}_{\text{linear phase term}} + \underbrace{\sum_{q=1}^{M} \angle(e^{j\omega} - c_q)} - \underbrace{\sum_{q=1}^{N} \angle(e^{j\omega} - d_q)}$$

sum of the angles from the zeros/poles to unit circle

Frequency response of LTI

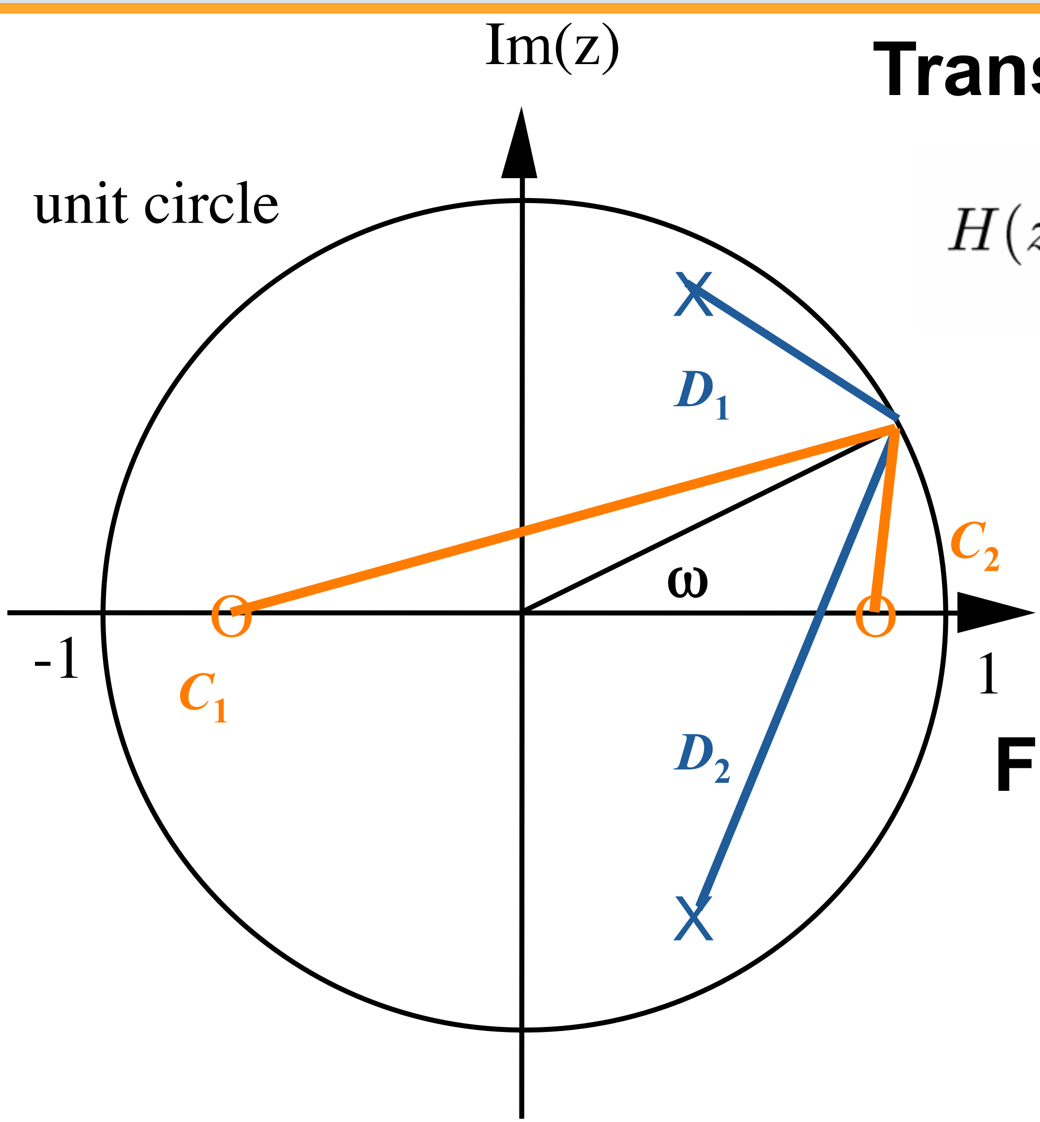


Suppose we have a system with 2 poles (o) and 2 zeros (x)

We are going around the unit circle with

$z = e^{j\omega}$

Frequency response of LTI



Transfer function:

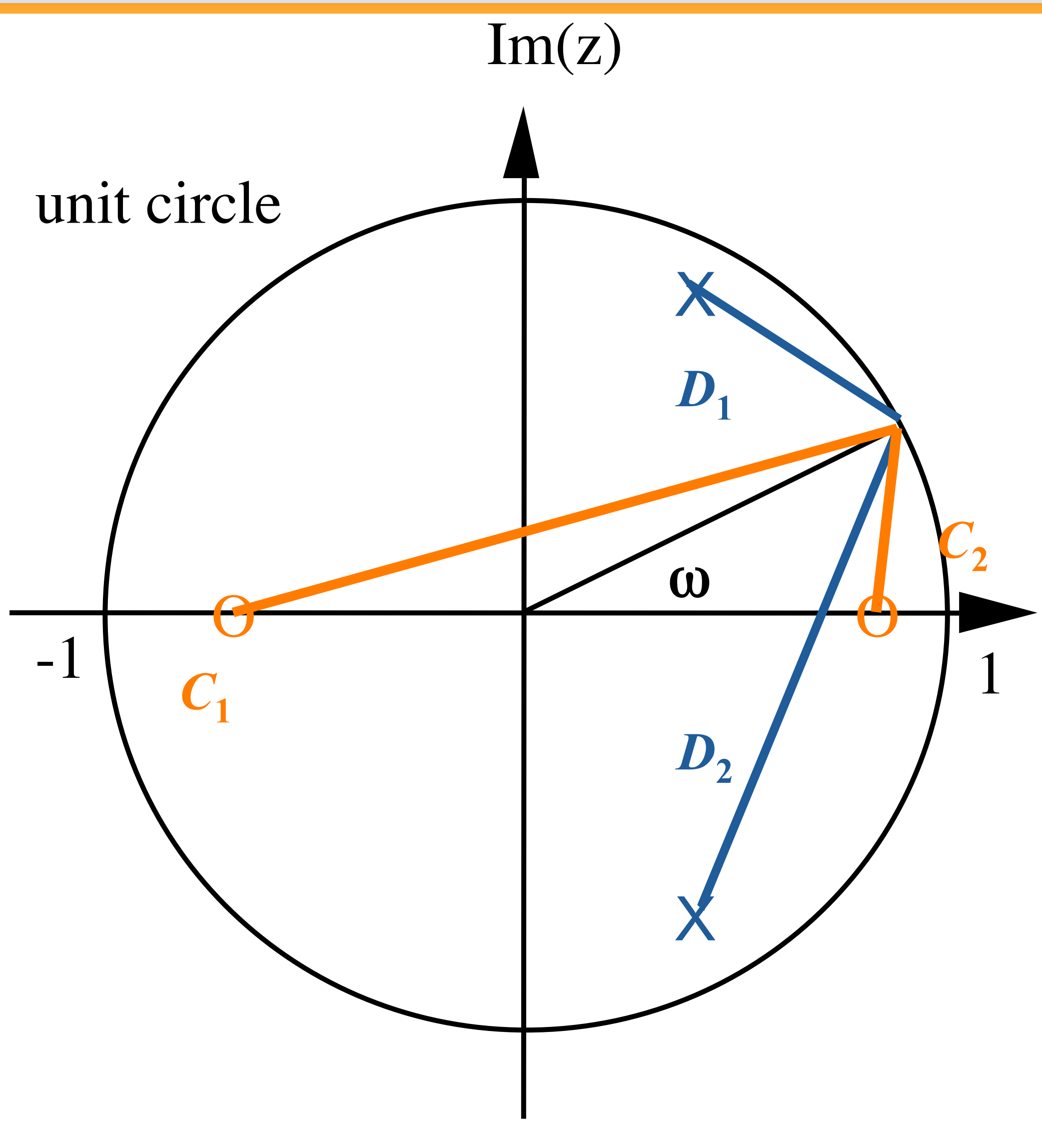
$$H(z) = b_0 \frac{\Pi_{q=1}^{M}(1 - c_q z^{-1})}{\Pi_{q=1}^{N}(1 - d_q z^{-1})}$$

Frequency response:

$$H(\omega) = H(z)|_{z=e^{j\omega}}$$

$$= b_0 \frac{C_1 C_2}{D_1 D_2}$$

Frequency response of LTI



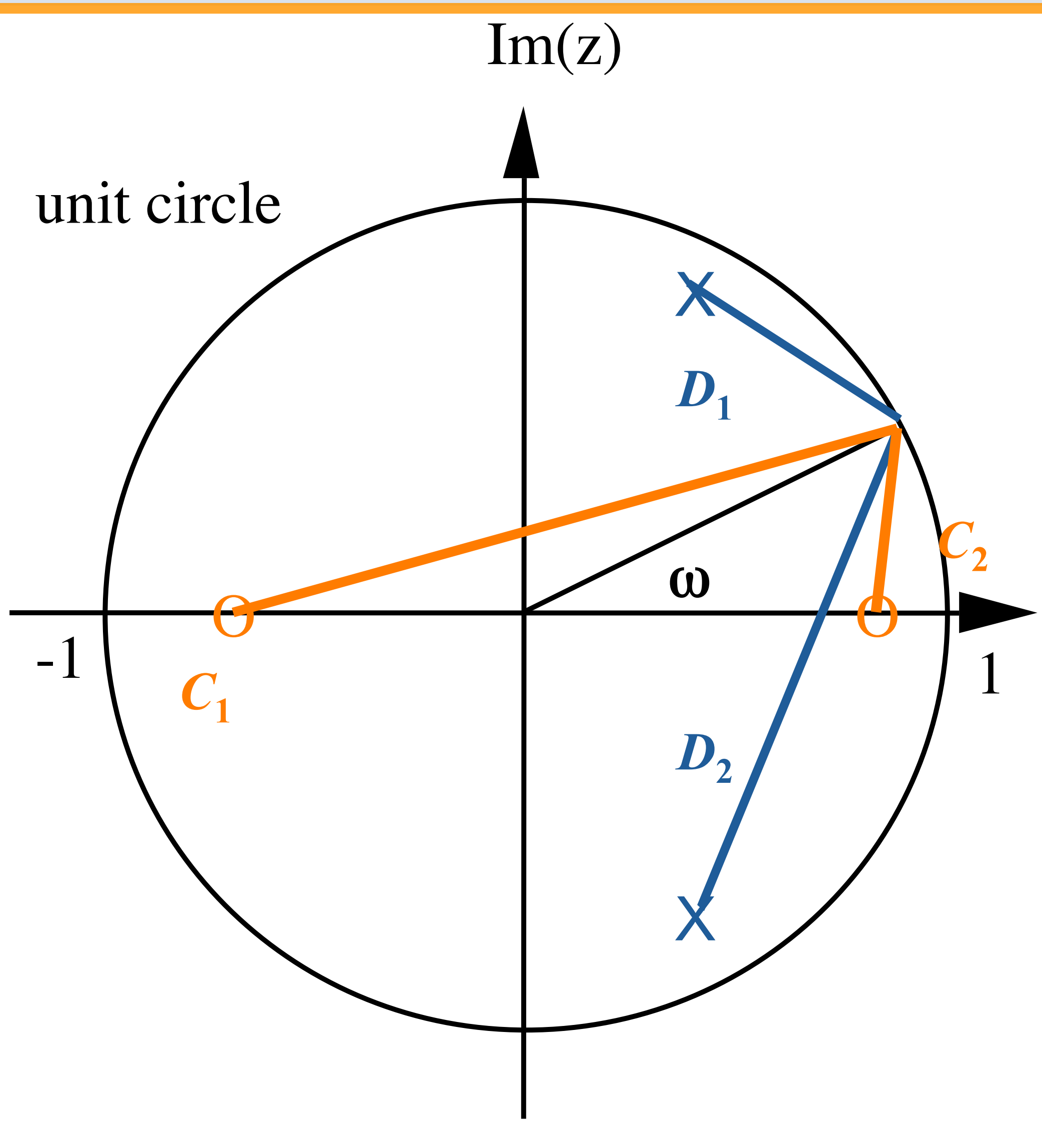
$$H(\omega) = H(z)|_{z=e^{j\omega}}$$

$$= b_0 \frac{C_1 C_2}{D_1 D_2}$$

The **magnitude** of the frequency response is given by b_0 times the **product of the distances from the zeros** to $z=e^{j\omega}$ **divided by the product of the distances from the poles to** $z=e^{j\omega}$

The **phase** response is given by the **sum of the angles from the zeros** to $z=e^{j\omega}$ **minus the sum of the angles from the poles** to $z=e^{j\omega}$ plus a linear phase term *(M-N) ω*

Frequency response of LTI



Thus when $z = e^{j\omega}$ 'is close to' a pole, the magnitude of the response rises (resonance).

When $z = e^{j\omega}$ 'is close to' a zero, the magnitude falls (a null).

The phase response – more difficult to get "intuition".

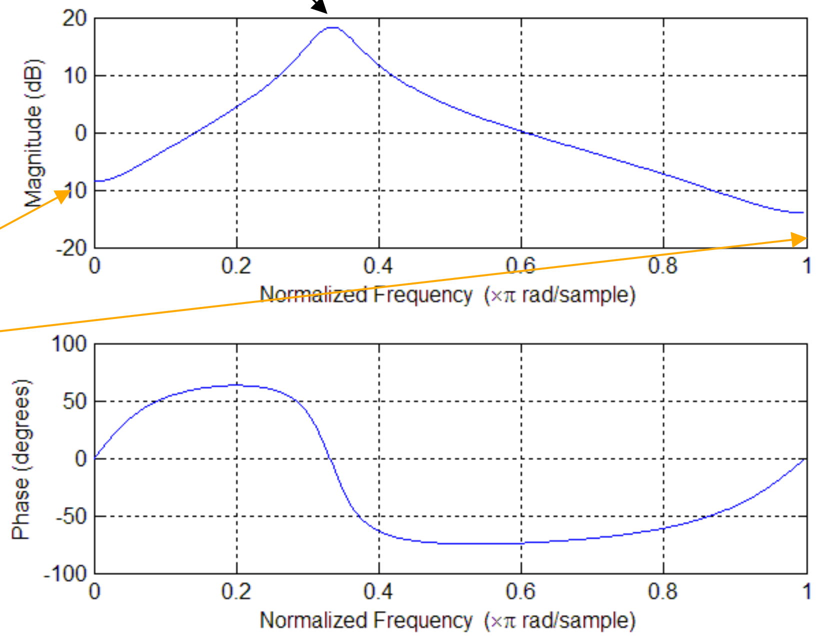
Frequency response of filter in Matlab

To evaluate the frequency response at n points equally spaced in the normalised frequency range ω=0 to ω= π, Matlab's function **`freqz`** is used:

```
freqz(b,a,n);
```

```
b=[1 -0.1 -0.56];
a=[1 -0.9 0.81];
freqz(b,a)
```

Peak close to pole frequency

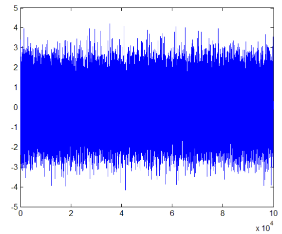


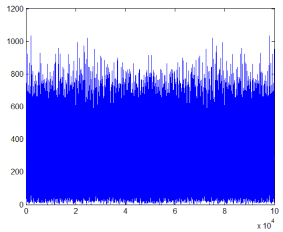
Troughs at zero frequencies

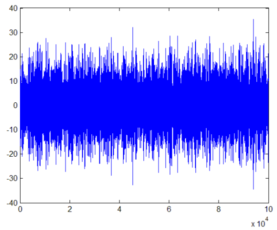
Filtering example

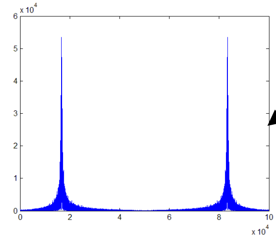
Generate a Gaussian random noise sequence:

```
x=randn(100000,1);
figure(1), plot(x)
figure(2), plot(abs(fft(x)))

a = [1 -0.99 0.9801];
b = [1, -0.1, -0.56];

y=filter(b,a,x);
figure(3), plot(y)
Figure(4), plot(abs(fft(y)))
```

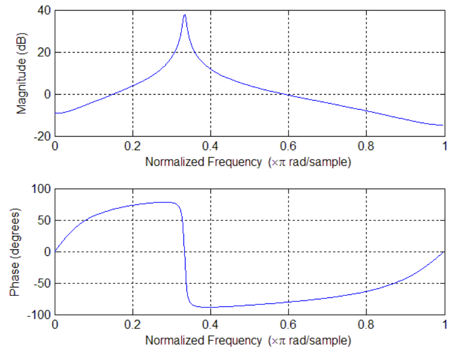








```
freqz(b,a);
```



Selective amplification of one frequency

Use `soundsc(x,44100)`
and `soundsc(y,44100)`
and hear the difference!

37

Filter specification

Before a filter is designed and implemented we need to specify its performance requirements.

There are four basic filter types:

- Low-pass
- High-pass
- Band-pass
- Band-stop

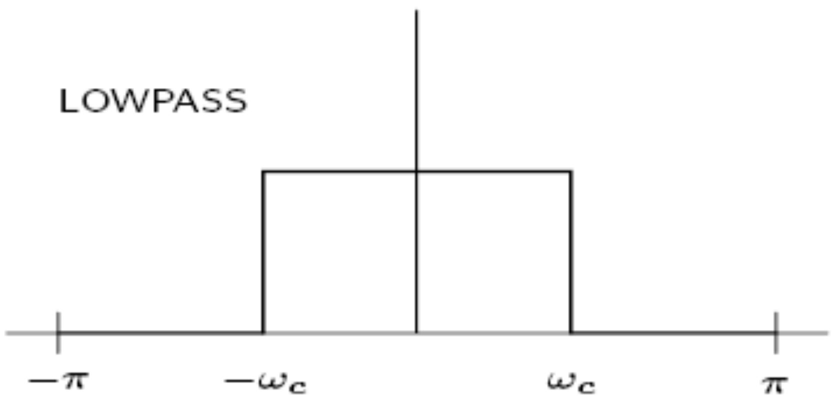
Frequency band where signal is passed is passband

Frequency band where signal is removed is stopband

Ideal Low-pass Filter

- **Low-pass:** designed to pass low frequencies from zero to a certain cut-off frequency and to block high frequencies

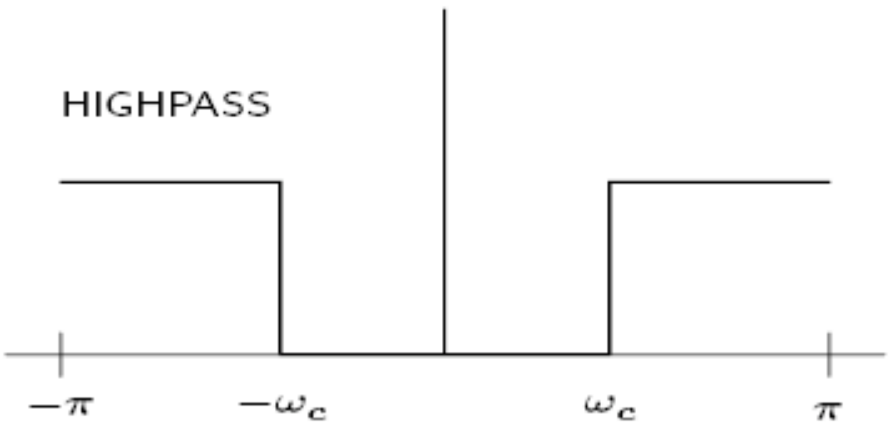
Ideal Frequency Response



Ideal High-pass Filter

- **High-pass:** designed to pass high frequencies from a certain cut-off frequency to π and to block low frequencies

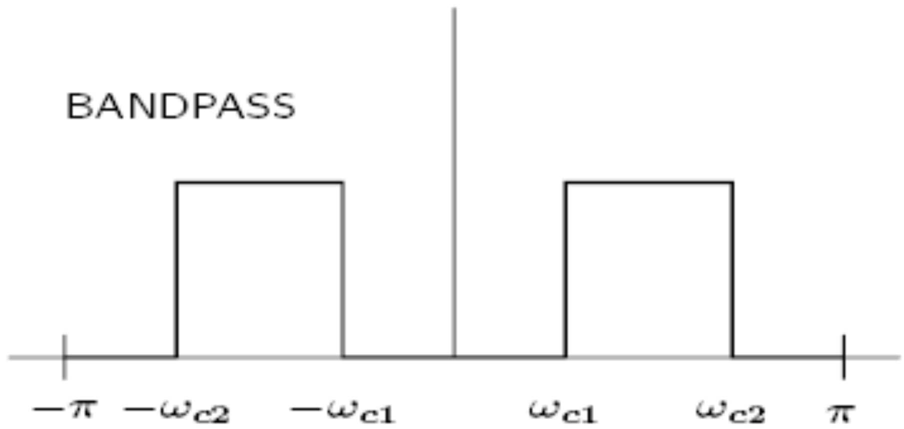
Ideal Frequency Response



Ideal Band-pass Filter

- **Band-pass:** designed to pass a certain frequency range which does not include zero and to block other frequencies

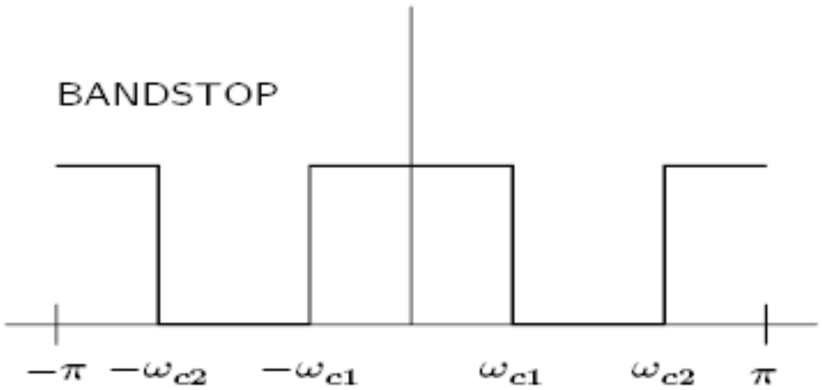
Ideal Frequency Response



Ideal Band-stop Filter

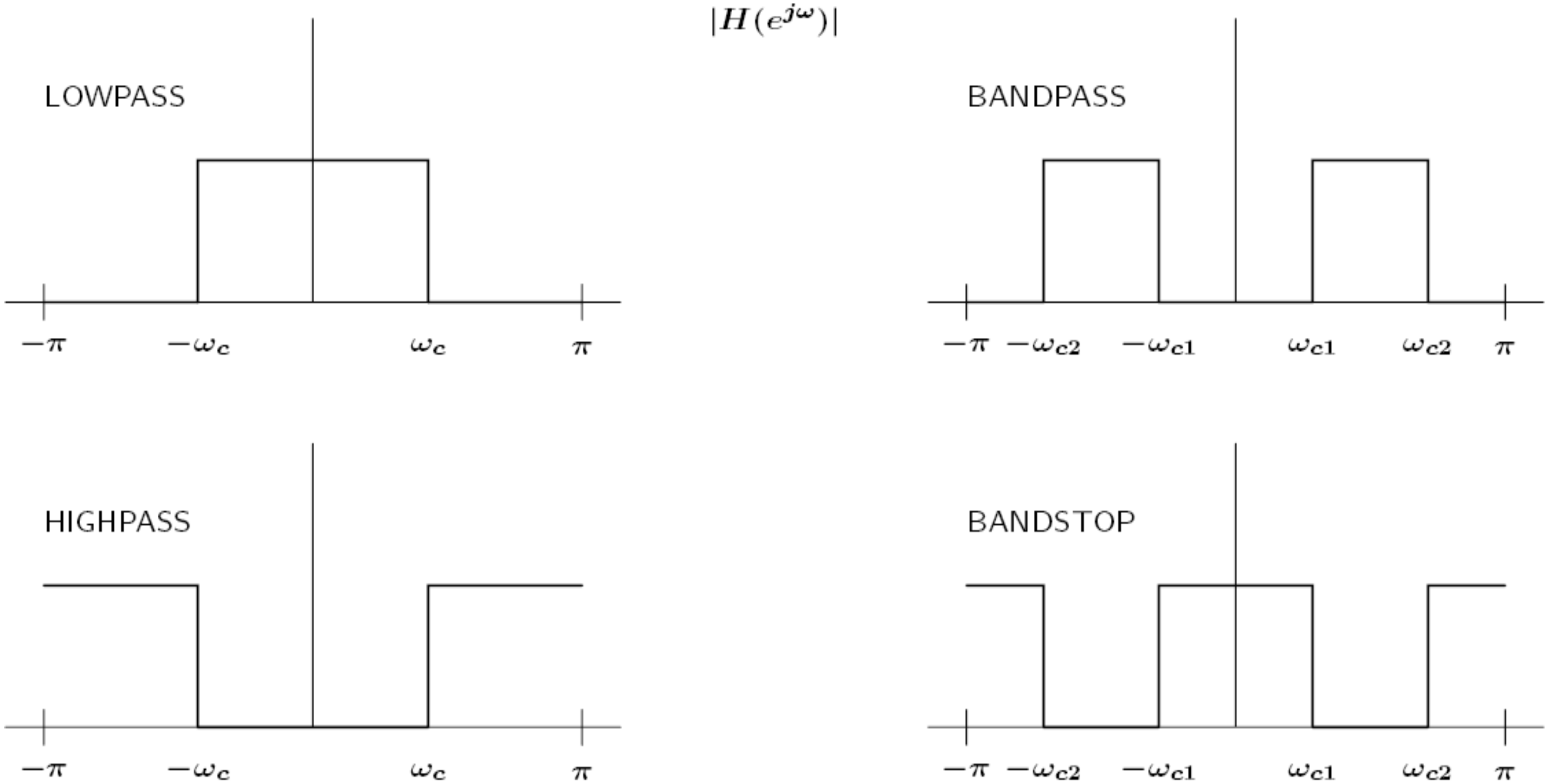
- **Band-stop:** designed to block a certain frequency range which does not include zero and to pass other frequencies

Ideal Frequency Response



Ideal Filters – Magnitude Response

Ideal Filters are usually such that they admit a gain of 1 in a given *passband* (where signal is passed) and 0 in their *stopband* (where signal is removed).



Ideal Filters – Phase Response

Another important characteristics is related to the phase

Ideal filter: admits a **linear phase response**

$$H(\omega) = |H(\omega)|\, e^{j\phi(\omega)}$$

$$\text{where } \phi(\omega) = -\omega n_0$$

Indeed,

$$Y(\omega) = |H(\omega)|\, \underbrace{e^{-j\omega n_0} X(\omega)}_{}$$

Fourier Transform of $\{x(n - n_0)\}$

delay

Ideal Filters – Linear Phase Response

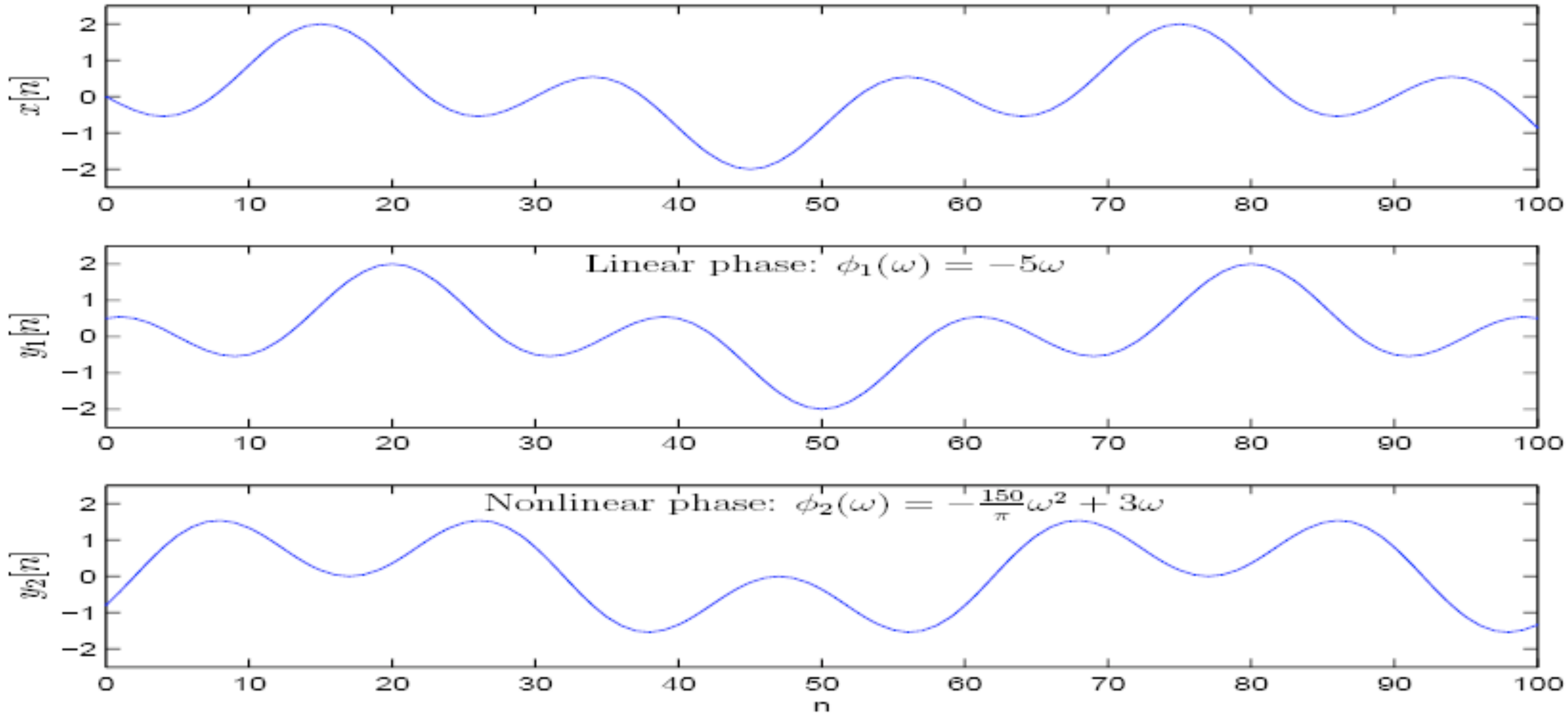
$$Y(\omega) = |H(\omega)|\, e^{-j\omega n_0} X(\omega)$$

$$\Rightarrow \quad \{x(n - n_0)\}$$

Important property: all frequencies suffer from the same delays

In some applications it is critical for this property to hold (at least approximately)

Linear / non-linear phase response



- Input $x(n) = \sin\left(\frac{\pi}{30}n\right) - \sin\left(\frac{\pi}{10}n\right)$
- Output: unit gain, linear phase $\phi_1(\omega) = -5\omega$ and nonlinear phase $\phi_2(\omega) = -\frac{150}{\pi}\omega^2 + 3\omega$

phase distortion

Ideal Filters – Linear Phase Response

$$H(\omega) = |H(\omega)|\, e^{j\phi(\omega)}$$

The derivative of the phase/signal respect with respect to ω is known as **group delay** of the filter - effectively *time delay of the frequency*

$$\tau_g(\omega) = -\frac{d\phi(\omega)}{d\omega}$$

when the phase admits a **linear phase response**

$$\phi(\omega) = -\omega n_0$$

$\Rightarrow$ **group delay is constant**

Designing Ideal Filters is Impossible

- In practice, a filter $\{h(n)\}$ needs to be causal to be realized;

i.e. $\boxed{h(n) = 0 \text{ for all } n < 0.}$

- The Paley-Wiener theorem shows that any causal filter is such that
  - $H(\omega)$ **cannot be** 0 except at a finite number of frequencies,
  - $|H(\omega)|$ **cannot be constant** in a frequency band
  - The transition from passband to stopband **cannot be infinitely** sharp
  - The magnitude $|H(\omega)|$ and the phase $\angle H(\omega)$ **cannot be independent** of each other.

$\Rightarrow$ **Bad news**: Ideal filters CANNOT BE IMPLEMENTED

Necessary to perform some approximations

Ideal Low- pass Filter Example

Let us consider for example a simple ideal lowpass filter defined by

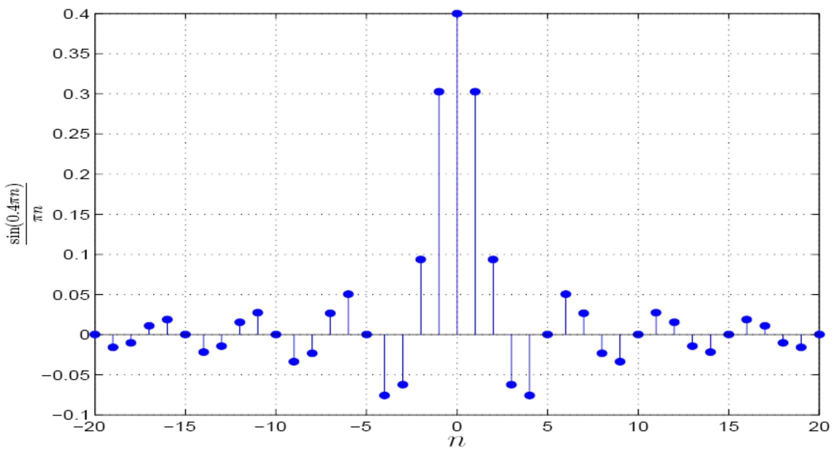
$$H_d(\omega) = \begin{cases} 1 \text{ if } |\omega| \leq \omega_c \\ 0 \text{ if } \omega_c < |\omega| < \pi. \end{cases}$$

It can be shown easily that the impulse response is given by

$$h_d(n) = \frac{\omega_c}{\pi} \frac{\sin \omega_c n}{\omega_c n}.$$

Desired impulse response has a sinc shape which is non-causal and infinite in duration.

clearly cannot be implemented in practice

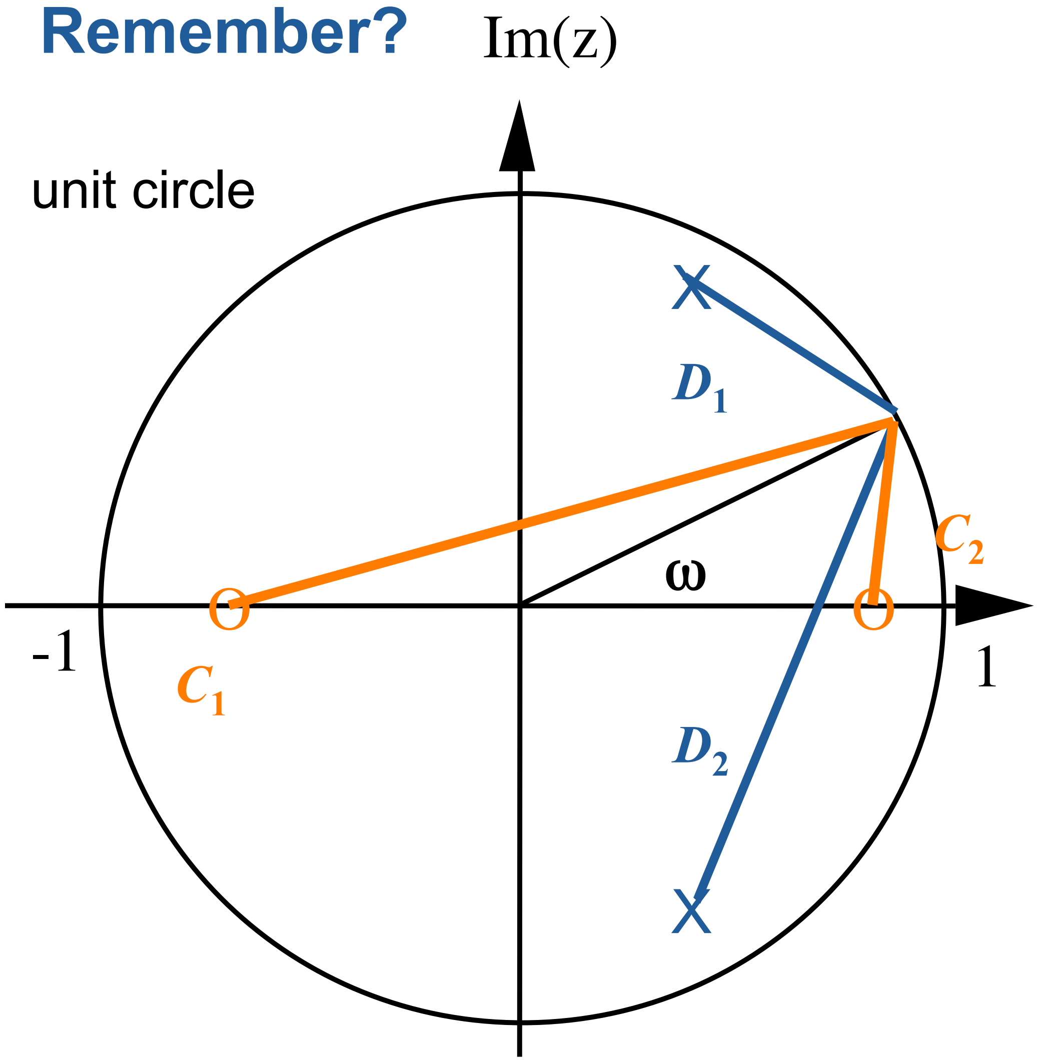


Filter Design

Let's have a go …

Frequency response of LTI

Remember?



Thus when $z = e^{j\omega}$ 'is close to' a pole, the magnitude of the response rises (resonance).

When $z = e^{j\omega}$ 'is close to' a zero, the magnitude falls (a null).

The phase response – more difficult to get “intuition”.

Approximate filter design – rough guidelines

A few rough guidelines to start with:

- Generally, as we approach a pole, the magnitude increases and as we approach a zero, the magnitude dips.

$\Rightarrow$ Possible to derive heuristically filters to fulfill *approximately* magnitude specifications.

- One needs to put the poles within the unit circle to ensure stability

Approximate Low-pass filter

Rule 1: The closer to a pole, the higher the magnitude of the response

- Poles placed near unit circle at points corresponding to $\underbrace{\text{low frequencies}}_{(\text{i.e. } \omega = 0)}$

Rule 2: The closer to a zero, the lower the magnitude of the response

- Zeros placed near or on the unit circle at points corresponding to $\underbrace{\text{high}}_{(\text{i.e. } \omega = \pi)}$ frequencies

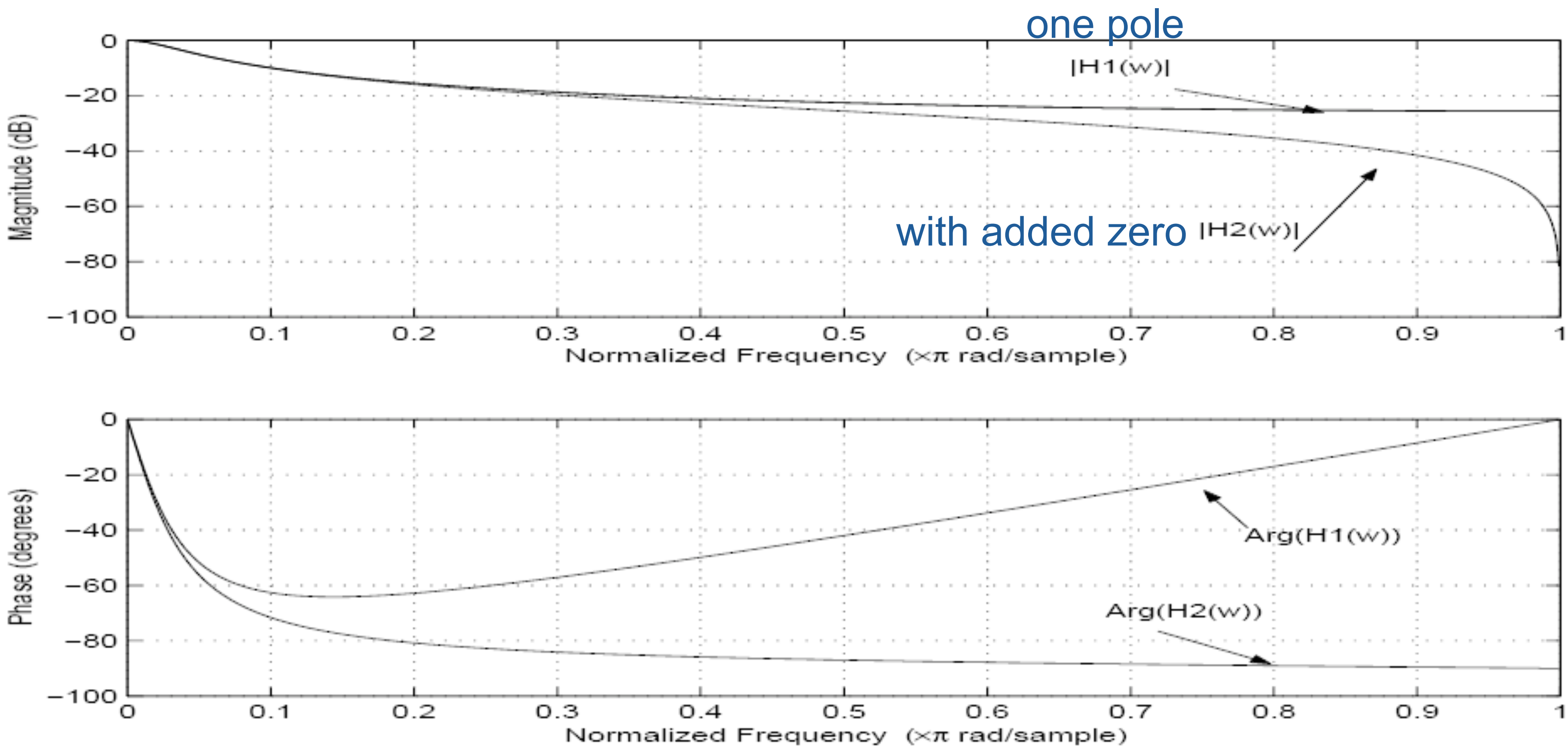
Example. Consider one pole

$$H_1(z) = \frac{1-\alpha}{1-\alpha z^{-1}}. \qquad \alpha \text{ close to } 1$$

Add a zero to further attenuate the response at high frequencies

$$H_2(z) = \frac{(1-\alpha)}{2} \frac{1+z^{-1}}{1-\alpha z^{-1}}.$$

Approximate Low-pass filter



Approximate High-pass filter

- Zeros placed near unit circle at points corresponding to low frequencies (i.e. $\omega = 0$)

- Poles placed near or on the unit circle at points corresponding to high frequencies (i.e. $\omega = \pi$).

- Lowpass Filters $\rightarrow$ Highpass: for poles and zeros $\alpha_k \rightarrow -\operatorname{Re}(\alpha_k) + j \operatorname{Im}(\alpha_k)$.

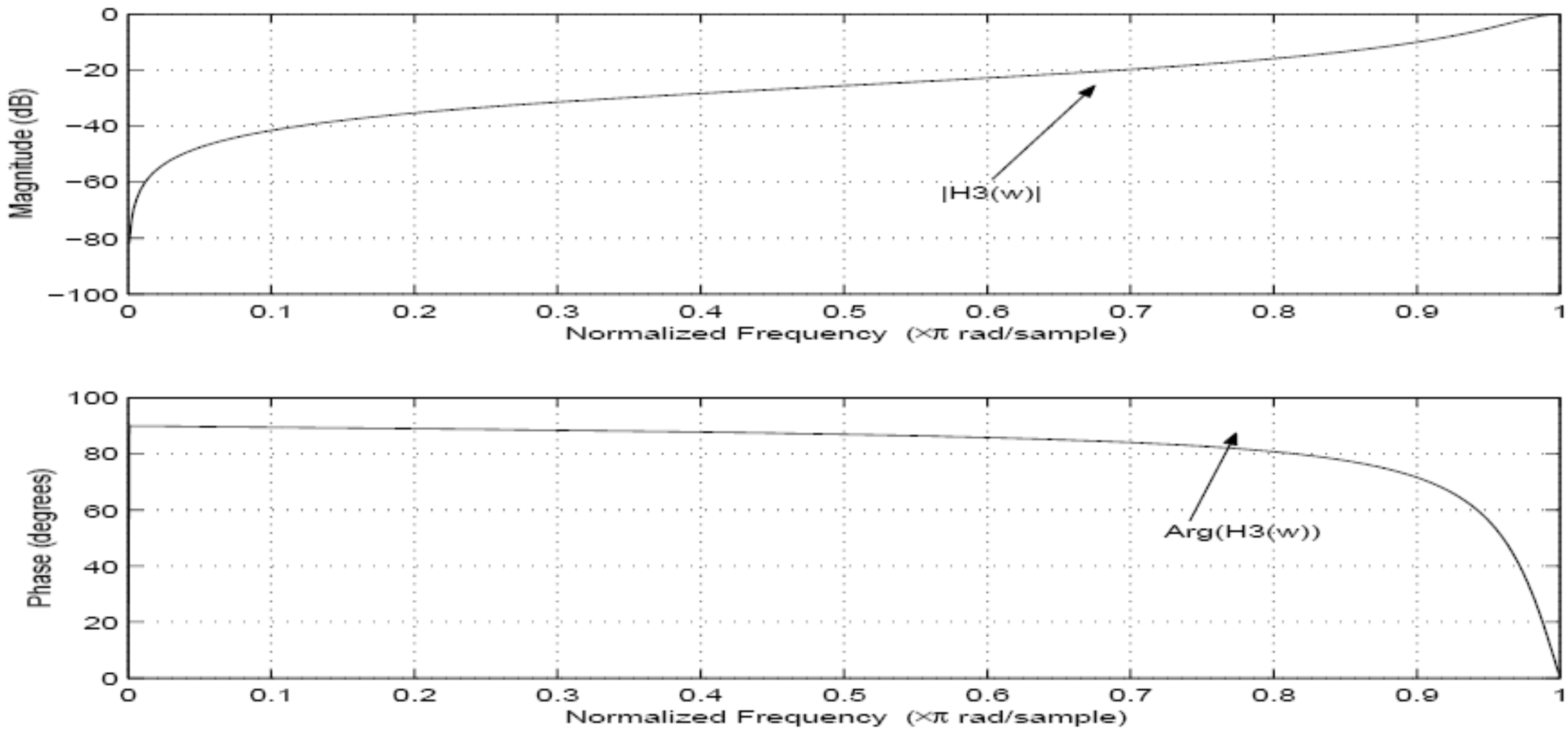
$$H_3(z) = \frac{(1-\alpha)}{2} \frac{1 - z^{-1}}{1 + \alpha z^{-1}}.$$

We had for Low-pass

$$H_2(z) = \frac{(1-\alpha)}{2} \frac{1 + z^{-1}}{1 - \alpha z^{-1}}$$

one can simply reflect the poles-zeros locations of the lowpass filter about the imaginary axis

Approximate High-pass filter



Alternative way: Low-pass to High-pass

Alternative way for Lowpass $\rightarrow$ Highpass: Translate lowpass filter by π

$$H_{HP}(\omega) = H_{LP}(\omega - \pi) \Rightarrow \underbrace{h_3(n) = \left(e^{-j\pi}\right)^n h_2(n) = (-1)^n h_2(n)}_{\text{in time domain}} .$$

If the lowpass filter was described by

$$y_n = -\sum_{k=1}^{N} a_k y_{n-k} + \sum_{k=0}^{M} b_k x_{n-k}$$

$$H_{HP}(\omega) = H_{LP}(\omega - \pi) = \frac{\sum_{k=0}^{M} b_k e^{-jk(\omega-\pi)}}{1 + \sum_{k=1}^{N} a_k e^{-jk(\omega-\pi)}} = \frac{\sum_{k=0}^{M} (-1)^k b_k e^{-jk\omega}}{1 + \sum_{k=1}^{N} (-1)^k a_k e^{-jk\omega}}$$

$$\Rightarrow \boxed{y(n) = -\sum_{k=1}^{N} (-1)^k a_k y(n-k) + \sum_{k=1}^{M} (-1)^k b_k x(n-k) .}$$

Example: $y(n) = \alpha y(n-1) + x(n) \Rightarrow y(n) = -\alpha y(n-1) + x(n)$

Bandpass and Resonator

Resonator.

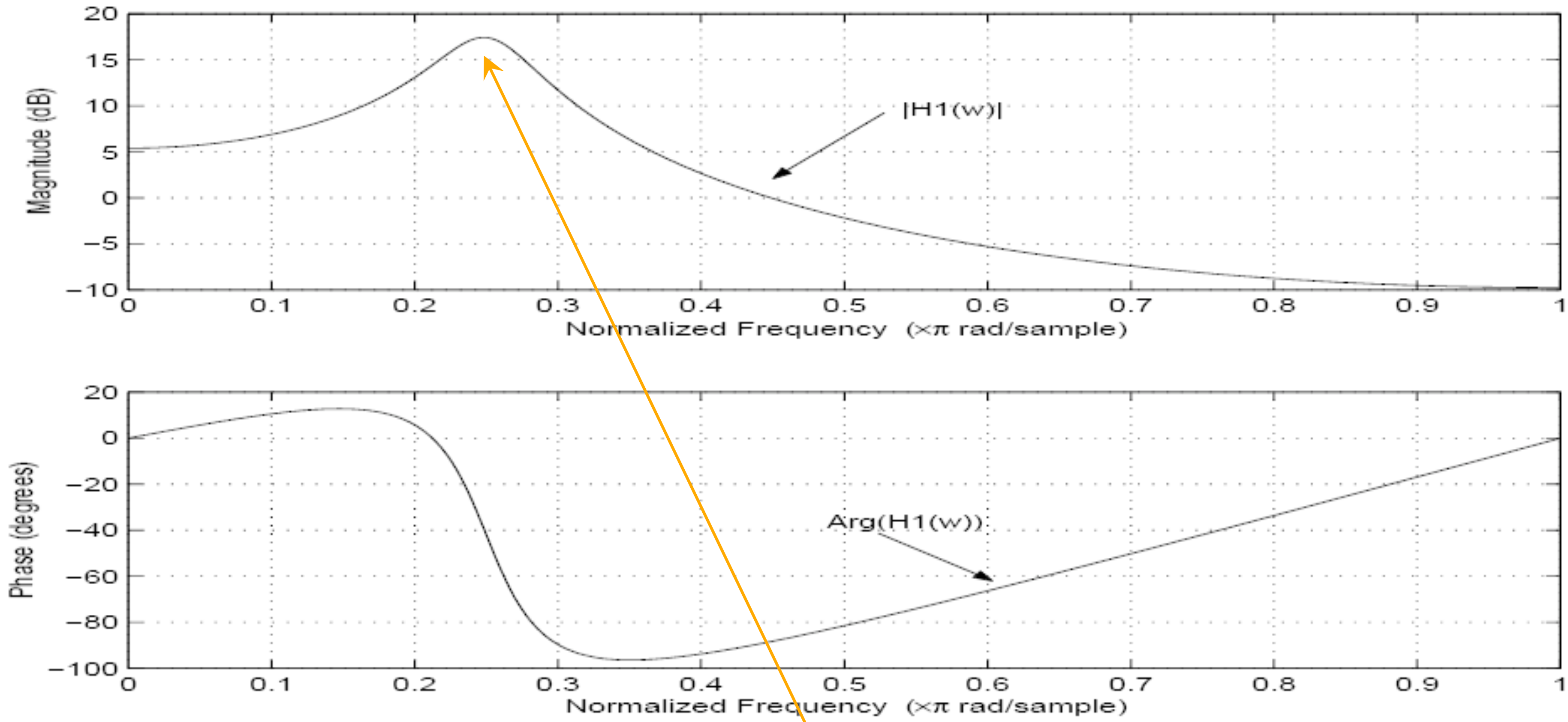
Resonator one or more pairs of complex conjugate poles near the unit circle, in the vicinity of the frequency band one wants to emphasize

$$H_1(z) = \frac{1}{\left(1 - re^{j\omega_0} z^{-1}\right)\left(1 - re^{-j\omega_0} z^{-1}\right)}$$

a filter which has its centre of the passband at ω_0

where r is close to 1 ($|r| < 1$). (e.g. $r = 0.9$ and $\omega_0 = \pi/4$.)

Resonator



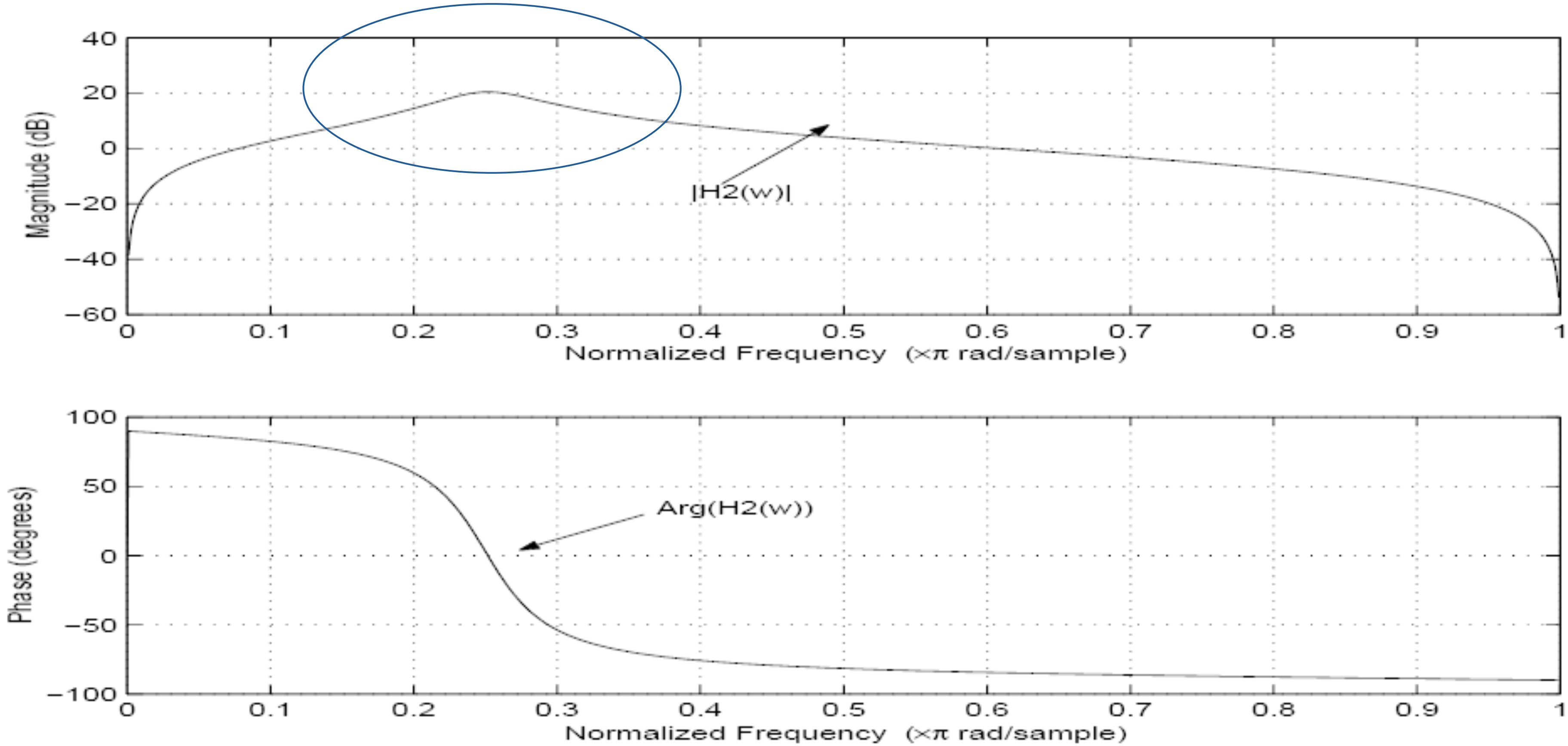
This filter is actually more a digital resonator than an bandpass filter; see its frequency response for r = 0.9 and $\omega_0 = \pi/4$ - it has a large magnitude response around ω_0

Band-pass Filter

Bandpass Filter. To obtain a standard bandpass filter, one can add zeros at $\omega = 0$ and $\omega = \pi$ to obtain

$$H_2(z) = \frac{\left(1 - z^{-1}\right)\left(1 + z^{-1}\right)}{\left(1 - re^{j\omega_0} z^{-1}\right)\left(1 - re^{-j\omega_0} z^{-1}\right)}.$$

Band-pass Filter



frequency response for r = 0.9 and $\omega_0 = \pi/4$

Notch Filter

a filter that contains one or several deeps/ notches in its frequency response

Notch filter. Interested in eliminating frequency ω_0

a pair of zeros at the locations $e^{\pm j\omega_0}$

$$H_1(z) = b_0 \left(1 - e^{j\omega_0} z^{-1}\right)\left(1 - e^{-j\omega_0} z^{-1}\right).$$

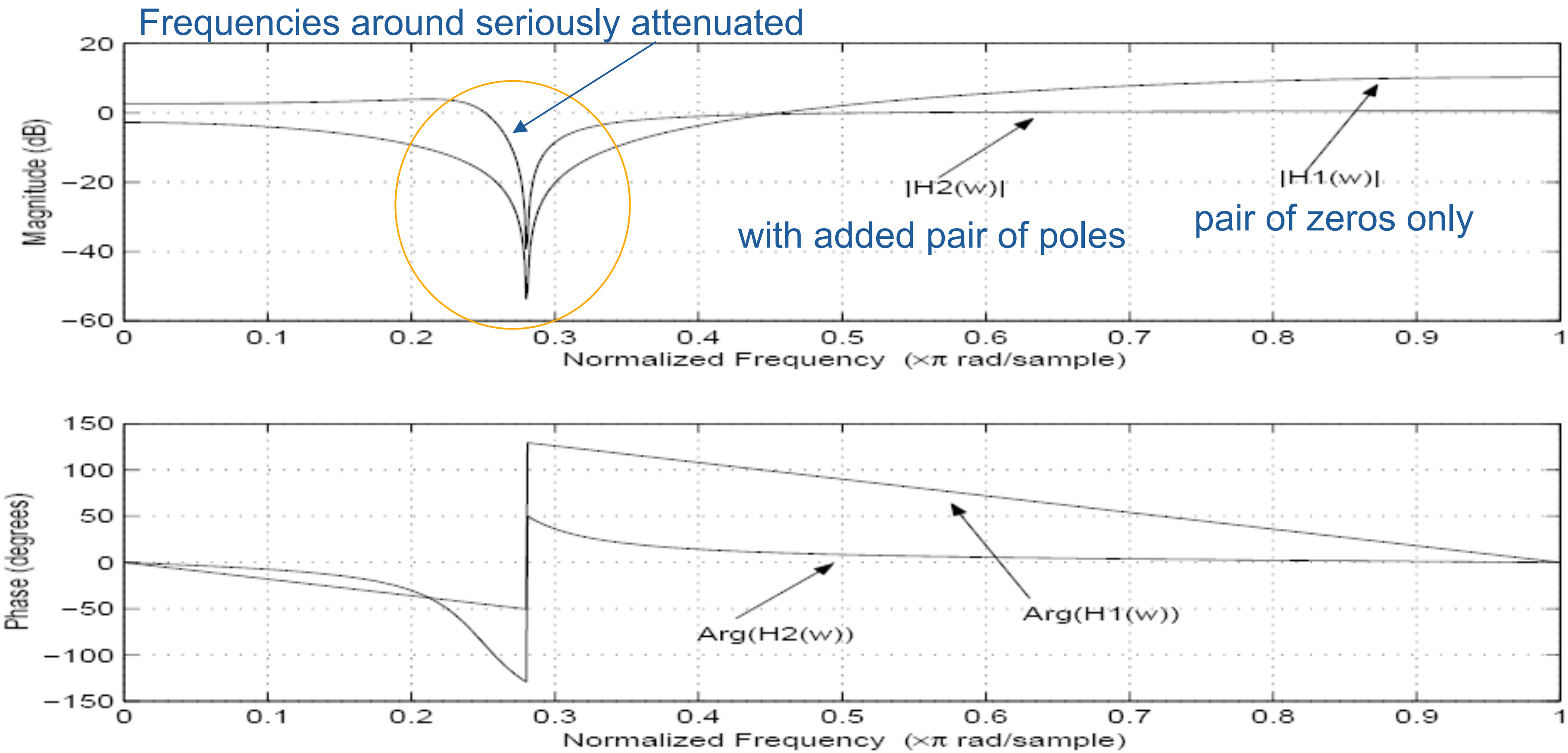
to eliminate ω_0

- Relatively large bandwidth. ⇒ *Frequencies around the desired null are also seriously attenuated*

- To compensate, add a pair of poles located at $re^{\pm j\omega_0}$ (r close to 1)

$$H_2(z) = b_0 \frac{\left(1 - e^{j\omega_0} z^{-1}\right)\left(1 - e^{-j\omega_0} z^{-1}\right)}{\left(1 - re^{j\omega_0} z^{-1}\right)\left(1 - re^{-j\omega_0} z^{-1}\right)}.$$

Notch Filter



For $r = 0.9$ and $\omega_0 = \pi/4$.

All pass Filter

All pass filter. Define a system admitting a constant magnitude response; i.e.

$$|H(\omega)| = 1 \text{ for all } \omega.$$

A trivial example is $H(z) = z^{-k}$

$\Rightarrow$ Allow to correct the phase response of systems.

Used as phase equalizer

$$\begin{aligned} H(z) &= \frac{a_N + a_{N-1}z^{-1} + \ldots + a_1 z^{-N+1} + z^{-N}}{1 + a_1 z^{-1} + \ldots + a_N z^{-N}} \\ &= \frac{z^{-N} + \sum_{k=1}^{N} a_k z^{-N+k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} = z^{-N}\frac{A(z^{-1})}{A(z)} \end{aligned}$$

$$A(z) = 1 + \sum_{k=1}^{N} a_k z^{-k}$$

One can check that $|H(\omega)| = \sqrt{e^{-j\omega N}\frac{A(e^{-j\omega})}{A(e^{j\omega})}e^{j\omega N}\frac{A^*(e^{-j\omega})}{A^*(e^{j\omega})}} = 1.$

if z_0 is a pole of $H(z)$ then $1/z_0$ is a zero of $H(z)$

Inverse Filter

- Assume one has

$$y(n) = h(n) * x(n) \Leftrightarrow Y(z) = H(z)X(z)$$

and is interesting in recovering $\{x(n)\}$ from $\{y(n)\}$: applications to telecommunications

- Filter $\{y(n)\}$ by the inverse filter

$$H_I(z) = \frac{1}{H(z)}.$$

The zeros become poles and the poles become zeros

- If $H(z)$ is a rational function

$$H(z) = b_0 \frac{\prod_{k=1}^{M}(1 - z_k z^{-1})}{\prod_{k=1}^{N}(1 - p_k z^{-1})} \Rightarrow H_I(z) = b_0^{-1} \frac{\prod_{k=1}^{N}(1 - p_k z^{-1})}{\prod_{k=1}^{M}(1 - z_k z^{-1})}.$$

- Only stable if $\{z_k\}$ within the unit circle.
- In practice almost never used... not robust to noise.

Digital Filter Design Considerations

Four steps:

1\. Specification of the filter’s response (very important! - senior engineers)

3\. Design the transfer function of the filter (main goal: meet spec with minimum complexity, often = minimum order)

5\. Verification of the filter’s performance

- analytic means
- simulations
- testing with real data if possible

4\. Implementation by hardware / software (or both)

Approximate filter design

Given precise specification what would you do??