COM303: Digital Signal Processing

Lecture 11: z-transform, filter structures and design examples

Overview

- the z-transform
- filtering algorithms and filter structures
- intuitive filter design
- two more ideal filters

Overview

- the z-transform
- filtering algorithms and filter structures
- intuitive filter design
- two more ideal filters

Overview

- the z-transform
- filtering algorithms and filter structures
- intuitive filter design
- two more ideal filters

Overview

- the z-transform
- filtering algorithms and filter structures
- intuitive filter design
- two more ideal filters

the z-transform

Overview:

- Constant-Coefficient Difference Equations
- The z-transform
- System transfer function
- Region of convergence and system stability

Overview:

- Constant-Coefficient Difference Equations
- The z-transform
- System transfer function
- Region of convergence and system stability

Overview:

- Constant-Coefficient Difference Equations
- The z-transform
- System transfer function
- Region of convergence and system stability

Overview:

- Constant-Coefficient Difference Equations
- The z-transform
- System transfer function
- Region of convergence and system stability

The most general LTI systems

- ideal filters cannot be implemented
- what is the most general, realizable LTI transformation?
  - linearity: only sums and multiplications
  - time-invariance: only multiplications by constants
  - realizability: only finite number of past and future samples

The most general LTI systems

- ideal filters cannot be implemented
- what is the most general, realizable LTI transformation?
  - linearity: only sums and multiplications
  - time-invariance: only multiplications by constants
  - realizability: only finite number of past and future samples

The most general LTI systems

- ideal filters cannot be implemented
- what is the most general, realizable LTI transformation?
  - linearity: only sums and multiplications
  - time-invariance: only multiplications by constants
  - realizability: only finite number of past and future samples

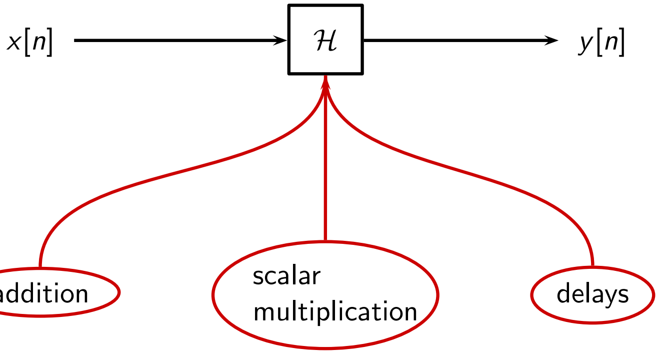
The most general LTI systems

- ideal filters cannot be implemented
- what is the most general, realizable LTI transformation?
  - linearity: only sums and multiplications
  - time-invariance: only multiplications by constants
  - realizability: only finite number of past and future samples

The most general LTI systems

- ideal filters cannot be implemented
- what is the most general, realizable LTI transformation?
  - linearity: only sums and multiplications
  - time-invariance: only multiplications by constants
  - realizability: only finite number of past and future samples

Linear, time-invariant systems



Constant-Coefficient Difference Equation

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]$$

- uses M input and N output values
- how do we compute the frequency response?
- we need a new tool!

Constant-Coefficient Difference Equation

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]$$

- uses M input and N output values
- how do we compute the frequency response?
- we need a new tool!

Constant-Coefficient Difference Equation

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]$$

- uses M input and N output values
- how do we compute the frequency response?
- we need a new tool!

Constant-Coefficient Difference Equation

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]$$

- uses M input and N output values
- how do we compute the frequency response?
- we need a new tool!

The z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}, \quad z \in \mathbb{C}$$

- for us mostly a *formal operator*...
- ...but also as the extension of the DTFT to the whole complex plane:

$$X(z)\big|_{z=e^{j\omega}} = \text{DTFT}\{x[n]\}$$

- (and now the notation $X(e^{j\omega})$ should make more sense)

The z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad z \in \mathbb{C}$$

- for us mostly a *formal operator*...
- ...but also as the extension of the DTFT to the whole complex plane:

$$X(z)\big|_{z=e^{j\omega}} = \mathrm{DTFT}\left\{x[n]\right\}$$

- (and now the notation $X(e^{j\omega})$ should make more sense)

The z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad z \in \mathbb{C}$$

- for us mostly a *formal operator*...
- ...but also as the extension of the DTFT to the whole complex plane:

$$X(z)|_{z=e^{j\omega}} = \mathrm{DTFT}\left\{x[n]\right\}$$

- (and now the notation $X(e^{j\omega})$ should make more sense)

The z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}, \quad z \in \mathbb{C}$$

- for us mostly a *formal operator*...
- ...but also as the extension of the DTFT to the whole complex plane:

$$X(z)|_{z=e^{j\omega}} = \text{DTFT}\{x[n]\}$$

- (and now the notation $X(e^{j\omega})$ should make more sense)

Key properties

linearity:

$$\mathcal{Z}\{\alpha x[n] + \beta y[n]\} = \alpha X(z) + \beta Y(z)$$

time shift:

$$\mathcal{Z}\{x[n - N]\} = z^{-N} X(z)$$

Key properties

linearity:

$$\mathcal{Z}\{\alpha x[n] + \beta y[n]\} = \alpha X(z) + \beta Y(z)$$

time shift:

$$\mathcal{Z}\{x[n-N]\} = z^{-N} X(z)$$

Applying the z-transform to CCDE's

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]$$

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

IMPORTANT: this assumes $X(z)$ and $Y(z)$ exist!

Applying the z-transform to CCDE's

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]$$

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

IMPORTANT: this assumes $X(z)$ and $Y(z)$ exist!

Rational Transfer Function

$$Y(z) = H(z)X(z)$$

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

From CCDE to Frequency Response

- z-transform of CCDE:

$$Y(z) = H(z)X(z)$$

- convolution theorem:

$$y[n] = h[n] * x[n] \quad \Rightarrow \quad Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

- $H(z)$ is the z-transform of the impulse response

Result #1: by setting $z = e^{j\omega}$ in $H(z)$ we get the frequency response!

From CCDE to Frequency Response

- z-transform of CCDE:

$$Y(z) = H(z)X(z)$$

- convolution theorem:

$$y[n] = h[n] * x[n] \quad \Rightarrow \quad Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

- $H(z)$ is the z-transform of the impulse response

Result #1: by setting $z = e^{j\omega}$ in $H(z)$ we get the frequency response!

From CCDE to Frequency Response

- z-transform of CCDE:

$$Y(z) = H(z)X(z)$$

- convolution theorem:

$$y[n] = h[n] * x[n] \quad \Rightarrow \quad Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

- $H(z)$ is the z-transform of the impulse response

Result #1: by setting $z = e^{j\omega}$ in $H(z)$ we get the frequency response!

From CCDE to Frequency Response

- z-transform of CCDE:

$$Y(z) = H(z)X(z)$$

- convolution theorem:

$$y[n] = h[n] * x[n] \quad \Rightarrow \quad Y(e^{j\omega}) = H(e^{j\omega})X(e^{j\omega})$$

- $H(z)$ is the z-transform of the impulse response

Result #1: by setting $z = e^{j\omega}$ in $H(z)$ we get the frequency response!

Leaky Integrator revisited

$$y[n] = (1-\lambda)x[n] + \lambda y[n-1]$$

$$Y(z) = (1-\lambda)X(z) + \lambda z^{-1} Y(z)$$

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

$$H(e^{j\omega}) = \frac{(1-\lambda)}{1-\lambda e^{-j\omega}}$$

Leaky Integrator revisited

$$y[n] = (1-\lambda)x[n] + \lambda y[n-1]$$

$$Y(z) = (1-\lambda)X(z) + \lambda z^{-1}Y(z)$$

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

$$H(e^{j\omega}) = \frac{(1-\lambda)}{1-\lambda e^{-j\omega}}$$

Leaky Integrator revisited

$$y[n] = (1-\lambda)x[n] + \lambda y[n-1]$$

$$Y(z) = (1-\lambda)X(z) + \lambda z^{-1}Y(z)$$

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

$$H(e^{j\omega}) = \frac{(1-\lambda)}{1-\lambda e^{-j\omega}}$$

Leaky Integrator revisited

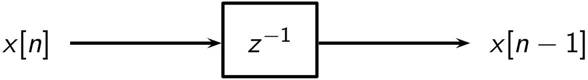
$$y[n] = (1-\lambda)x[n] + \lambda y[n-1]$$

$$Y(z) = (1-\lambda)X(z) + \lambda z^{-1}Y(z)$$

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

$$H(e^{j\omega}) = \frac{(1-\lambda)}{1-\lambda e^{-j\omega}}$$

BTW, remember the delay block?



$$Y(z) = z^{-1} X(z)$$

now the notation should make more sense!

Existence and region of convergence (ROC)

$$|X(z)| < \infty \quad \text{for } z \in \text{ROC} \subseteq \mathbb{C}$$

How can we determine the ROC?

- ROC depends on the values of $x[n]$
- for rational transfer function we can use indirect methods
- we don't care about convergence in zero and infinity

Existence and region of convergence (ROC)

$$|X(z)| < \infty \quad \text{for } z \in \text{ROC} \subseteq \mathbb{C}$$

How can we determine the ROC?

- ROC depends on the values of $x[n]$
- for rational transfer function we can use indirect methods
- we don't care about convergence in zero and infinity

Region of convergence (ROC)

Obvious fact:

for finite-support signals, the z-transform converges everywhere (except in 0 and/or ∞)

$$X(z) = \sum_{n=-M}^{N} x[n]z^{-n} = z^{-N} P_{(M+N-1)}(z)$$

Region of convergence (ROC)

Theorem:

the z-transform is a power series, so convergence is always absolute

$$|X(z)| < \infty \Longleftrightarrow \sum_{n=-\infty}^{\infty} |x[n]z^{-n}| < \infty$$

Region of convergence (ROC)

Corollary #1:

the region of convergence has circular symmetry: set $z = ae^{j\theta}$:

$$\left| \sum_{n=-\infty}^{\infty} x[n] z^{-n} \right| < \infty \iff \sum_{n=-\infty}^{\infty} |x[n]| \, |a^{-n}| < \infty$$

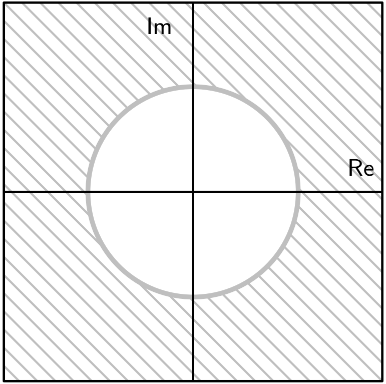
Region of convergence (ROC)

Corollary #2:

for causal signals, the ROC extends from a circle to infinity:
assume $H(z_0)$ exists and $|z_1| > |z_0|$:

$$|X(z_1)| = \left| \sum_{n=0}^{\infty} \frac{x[n]}{z_1^n} \right| \leq \sum_{n=0}^{\infty} \frac{|x[n]|}{|z_1^n|} \leq \sum_{n=0}^{\infty} \frac{|x[n]|}{|z_0^n|} \leq \infty$$

ROC shape for causal sequences



Region of convergence (ROC)

so where are the convergence problems?

in general, difficult question; but we're only interested in rational transfer functions!

Region of convergence (ROC)

so where are the convergence problems?

in general, difficult question; but we're only interested in rational transfer functions!

ROC for causal systems

Consider the transfer function for an LTI system:

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_{M-1} z^{-(M-1)}}{a_0 + a_1 z^{-1} + \ldots + a_{N-1} z^{-(N-1)}}$$

It can always be factored as:

$$H(z) = C \frac{\prod_{n=1}^{M-1} (1 - z_n z^{-1})}{\prod_{n=1}^{N-1} (1 - p_n z^{-1})}$$

ROC for causal systems

- z_n's: *zeros* of the transfer function
- p_n's: *poles* of the transfer function
- only trouble spots for ROC are the poles

ROC for causal systems

- z_n's: *zeros* of the transfer function
- p_n's: *poles* of the transfer function
- only trouble spots for ROC are the poles

ROC for causal systems

- z_n's: *zeros* of the transfer function
- p_n's: *poles* of the transfer function
- only trouble spots for ROC are the poles

ROC for causal systems

We know:

- ROC extends outwards
- ROC cannot include poles

ROC extends outwards from a circle touching the largest-magnitude pole

ROC for causal systems

We know:

- ROC extends outwards
- ROC cannot include poles

ROC extends outwards from a circle touching the largest-magnitude pole

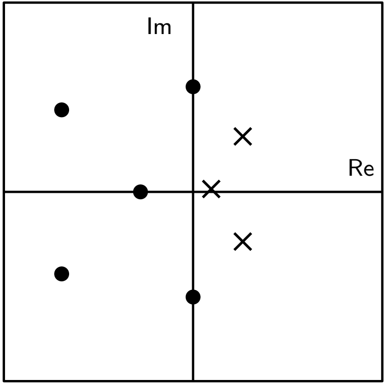
ROC for causal systems

We know:

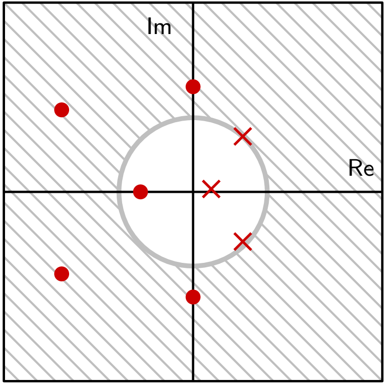
- ROC extends outwards
- ROC cannot include poles

ROC extends outwards from a circle touching the largest-magnitude pole

ROC for causal systems



ROC for causal systems



System stability criterion

Consider a filter with impulse response $h[n]$

- BIBO stability $\iff \sum_{n=-\infty}^{\infty} |h[n]| < \infty$
- $1 \in \mathrm{ROC} \iff \sum_{n=-\infty}^{\infty} |h[n]| < \infty$ because $H(z)$ converges absolutely in $z = 1$

Result #2: system is stable if and only if ROC includes the unit circle!

System stability criterion

Consider a filter with impulse response $h[n]$

- BIBO stability $\iff \sum_{n=-\infty}^{\infty} |h[n]| < \infty$
- $1 \in \text{ROC} \iff \sum_{n=-\infty}^{\infty} |h[n]| < \infty$ because $H(z)$ converges absolutely in $z = 1$

Result #2: system is stable if and only if ROC includes the unit circle!

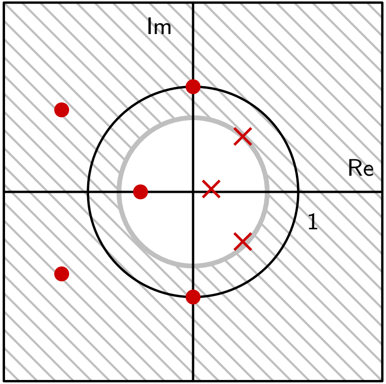
System stability criterion

Consider a filter with impulse response $h[n]$

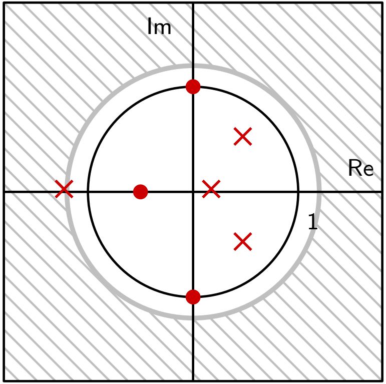
- BIBO stability $\iff \displaystyle\sum_{n=-\infty}^{\infty} |h[n]| < \infty$
- $1 \in \text{ROC} \iff \displaystyle\sum_{n=-\infty}^{\infty} |h[n]| < \infty$ because $H(z)$ converges absolutely in $z = 1$

Result #2: system is stable if and only if ROC includes the unit circle!

Stable system



Unstable system



Common confusion...

$$y[n] = 2y[n-1] + x[n] \qquad \text{(unstable!)}$$

apply z-transform as a formal operator:

$$H(z) = Y(z)/X(z) = \frac{1}{1 - 2z^{-1}}$$

$H(1) = -1 < \infty$, so system is stable?

Common confusion...

$$y[n] = 2y[n-1] + x[n] \qquad \text{(unstable!)}$$

apply z-transform as a formal operator:

$$H(z) = Y(z)/X(z) = \frac{1}{1 - 2z^{-1}}$$

$H(1) = -1 < \infty$, so system is stable?

Common confusion...

$$y[n] = 2y[n-1] + x[n] \qquad \text{(unstable!)}$$

apply z-transform as a formal operator:

$$H(z) = Y(z)/X(z) = \frac{1}{1 - 2z^{-1}}$$

$H(1) = -1 < \infty$, so system is stable?

Common confusion clarified

ROC depends on $h[n]$, NOT on *formal* value of $H(z)$:

- $h[n] = 2^n \, u[n]$
- to apply the z-transform operator we *assume* to be in the ROC
- the region of convergence is $|z| > 2$ because

$$\sum_{n=0}^{\infty} a^n \, z^{-n} = \lim_{N \to \infty} \frac{1 - (a/z)^N}{1 - (a/z)} = \begin{cases} \dfrac{1}{1 - az^{-1}} & \text{if } |z| > |a| \\ \infty & \text{otherwise} \end{cases}$$

In other words...

the function $\dfrac{1}{1 - az^{-1}}$ is defined for all $z \in \mathbb{C} \setminus \{2\}$

BUT

it is the z-transform of $a^n u[n]$ *only* for $|z| > |a|$

For a generic rational transfer function

$$\sum_{k=0}^{N-1} a_k y[n-k] = \sum_{k=0}^{M-1} b_k x[n-k]$$

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

$$Y(z) = H(z)X(z)$$

$$H(z) = \frac{\sum_{k=0}^{M-1} b_k z^{-k}}{\sum_{k=0}^{N-1} a_k z^{-k}} = \mathcal{Z}\{h[n]\} \qquad \leftarrow \text{this must exist!}$$

Rational Transfer Function

for which values of z does $H(z)$ exist?

- option 1: compute $h[n]$ explicitly and find ROC for the power series $\sum h[n]z^{-n}$
- option 2: derive ROC indirectly:
  - ROC is circular symmetric
  - ROC extends outwards for causal sequences
  - ROC cannot include poles

Estimating the frequency response from the pole-zero plot

The "circus tent" method:

- magnitude of z transform is like a rubber sheet over the complex plane
- zeros glue the sheet to the ground
- poles are like … poles, pushing it up
- frequency response (in magnitude) is sheet profile around the unit circle

Estimating the frequency response from the pole-zero plot

The "circus tent" method:

- magnitude of z transform is like a rubber sheet over the complex plane
- zeros glue the sheet to the ground
- poles are like … poles, pushing it up
- frequency response (in magnitude) is sheet profile around the unit circle

Estimating the frequency response from the pole-zero plot

The "circus tent" method:

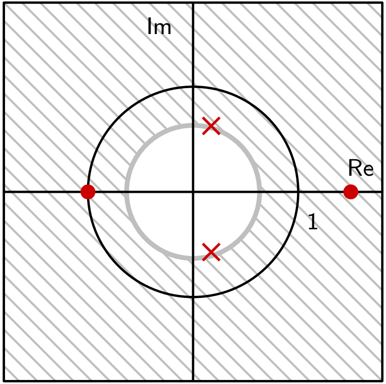
- magnitude of z transform is like a rubber sheet over the complex plane
- zeros glue the sheet to the ground
- poles are like … poles, pushing it up
- frequency response (in magnitude) is sheet profile around the unit circle

Estimating the frequency response from the pole-zero plot

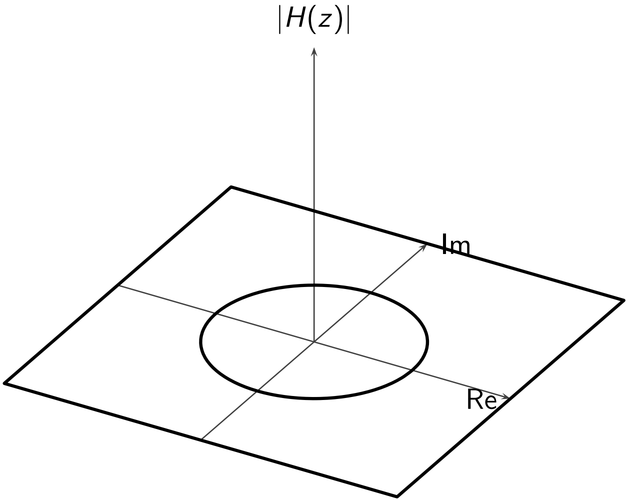
The "circus tent" method:

- magnitude of z transform is like a rubber sheet over the complex plane
- zeros glue the sheet to the ground
- poles are like … poles, pushing it up
- frequency response (in magnitude) is sheet profile around the unit circle

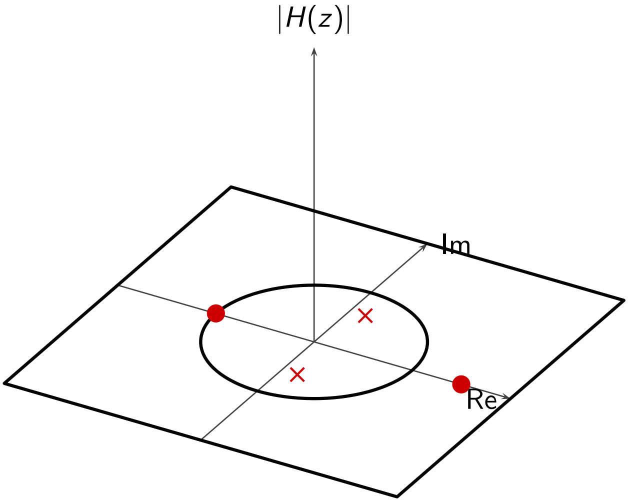
Estimating the frequency response



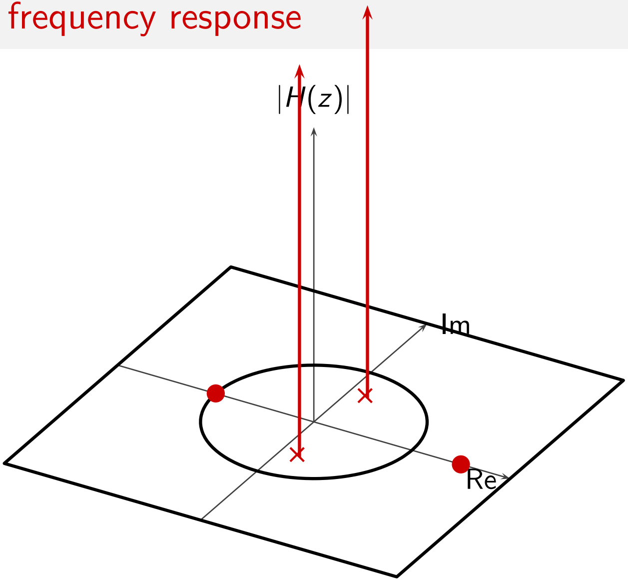
Estimating the frequency response



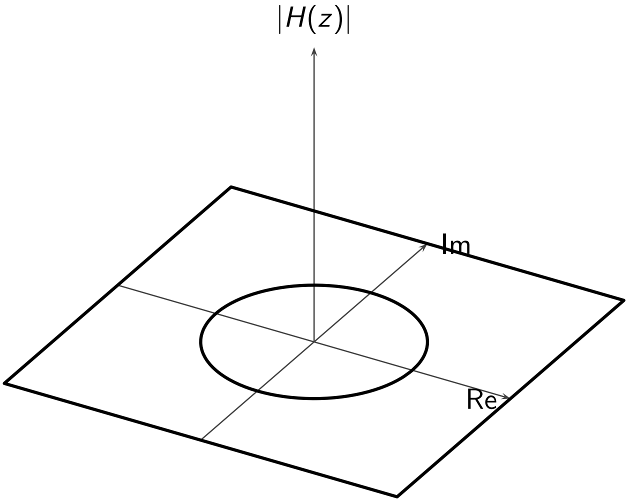
Estimating the frequency response



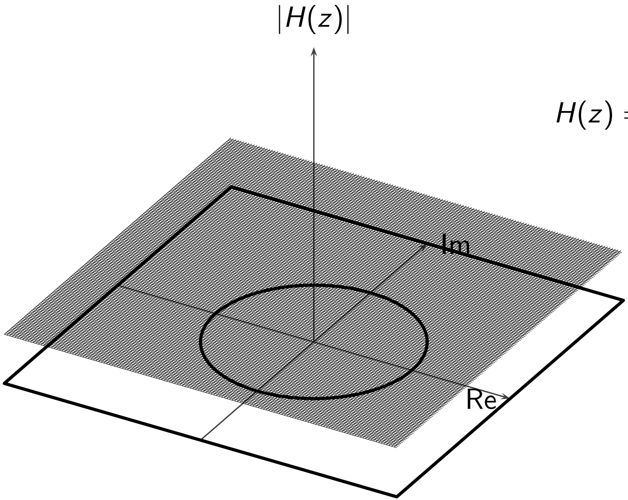
Estimating the frequency response



Estimating the frequency response

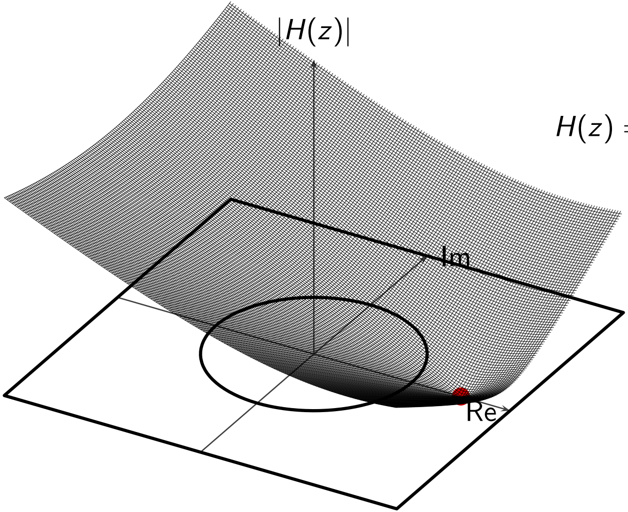


Estimating the frequency response



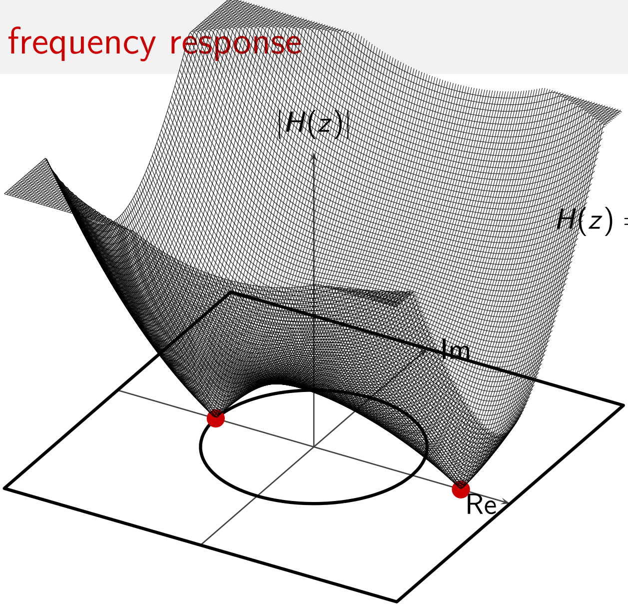
$H(z) = 1$

Estimating the frequency response



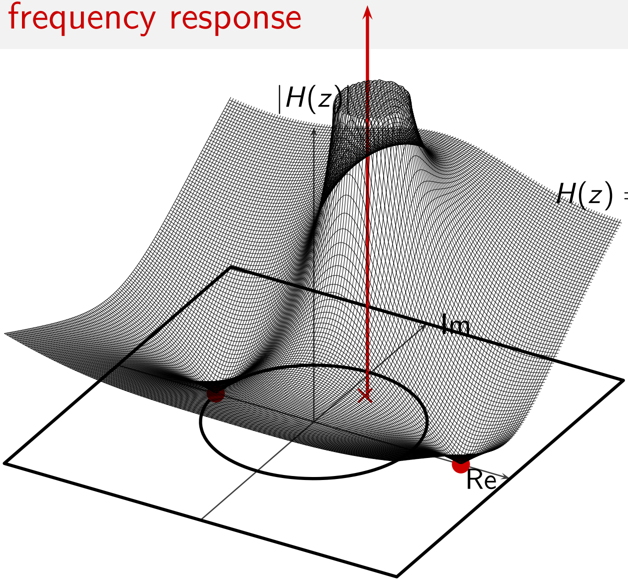
$$H(z) = (1 - z_0\, z^{-1})$$

Estimating the frequency response



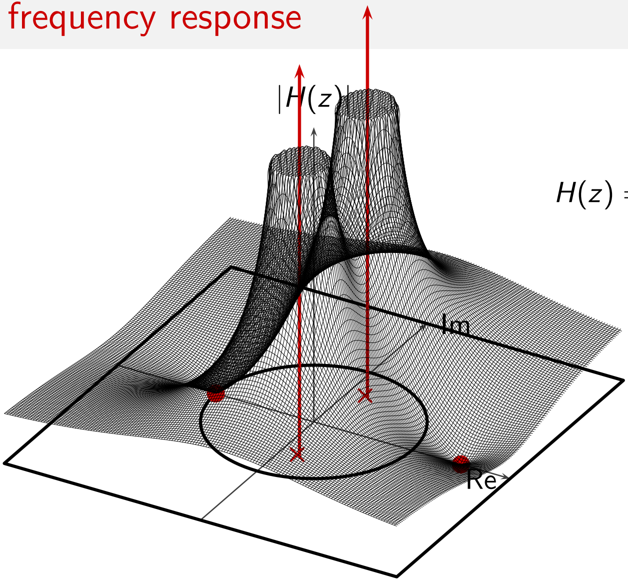
$$H(z) = (1 - z_0\, z^{-1})(1 + z_1\, z^{-1})$$

Estimating the frequency response



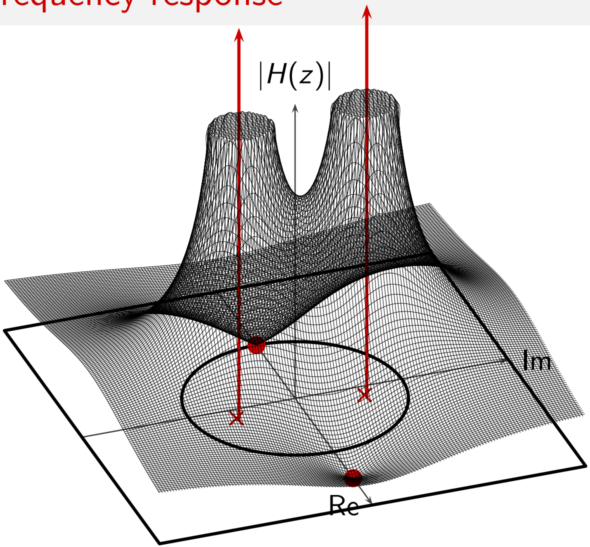
$$H(z) = \frac{(1-z_0\, z^{-1})(1+z_1\, z^{-1})}{(1-p_0\, z^{-1})}$$

Estimating the frequency response

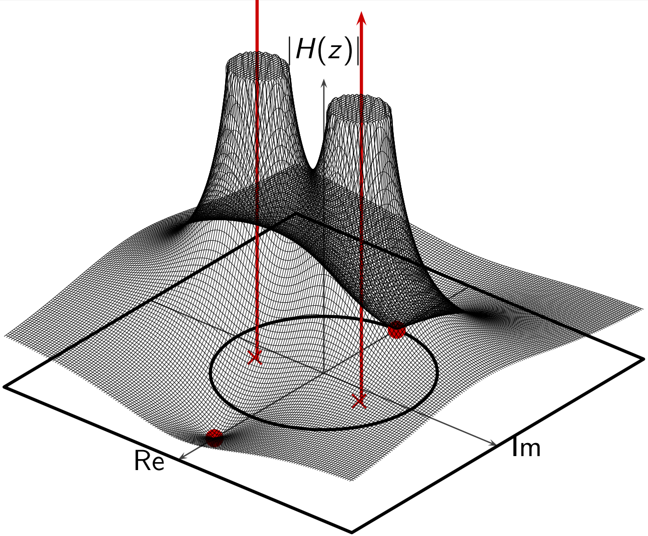


$$H(z) = \frac{(1 - z_0\, z^{-1})(1 + z_1\, z^{-1})}{(1 - p_0\, z^{-1})(1 - p_0^*\, z^{-1})}$$

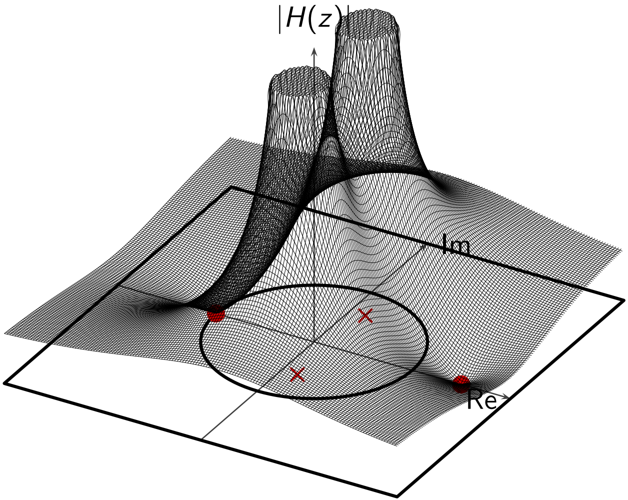
Estimating the frequency response



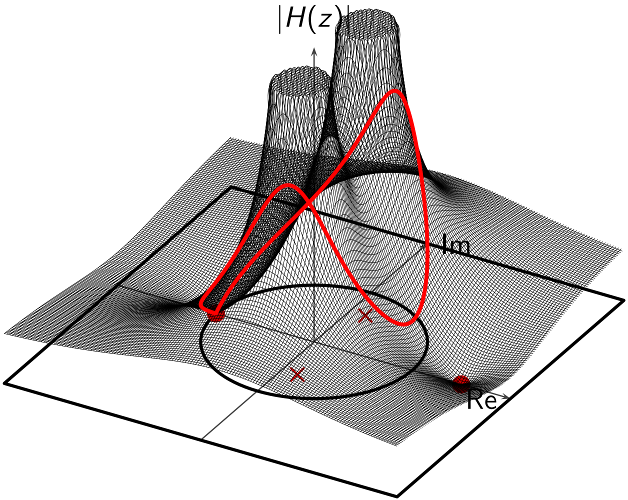
Estimating the frequency response

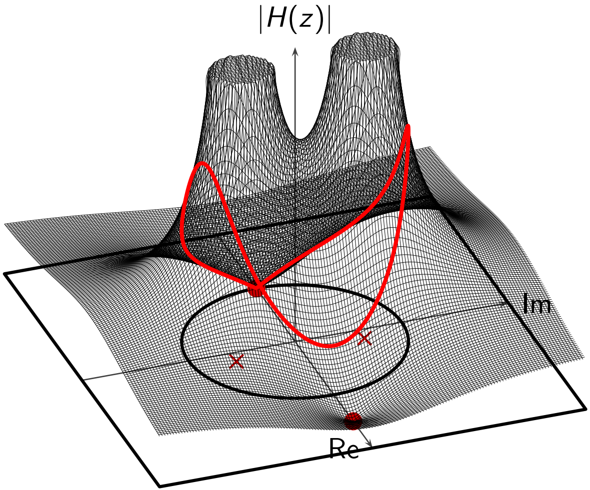


Estimating the frequency response

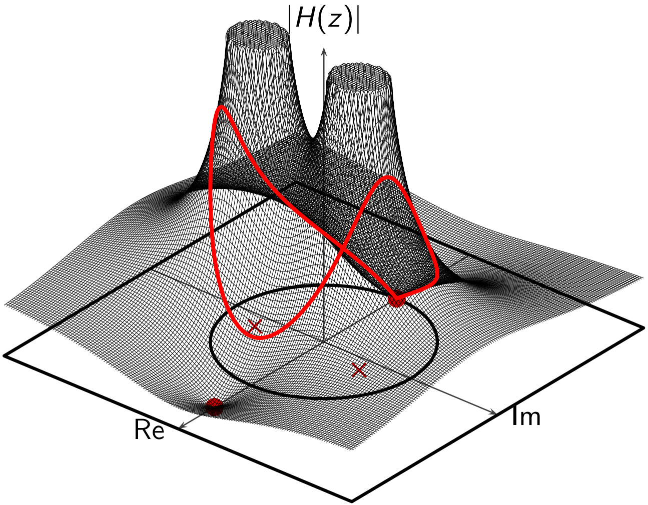


Estimating the frequency response

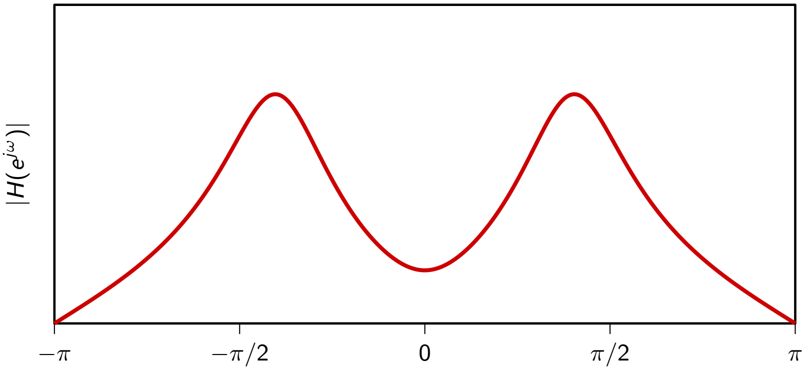



|H(z)|
Im
Re

Estimating the frequency response



Estimating the frequency response



filtering algorithms and structures

Overview:

- Algorithms for CCDE's
- Block diagram
- Real-time processing

An old friend

```
class Leaky:
    def __init__(self, lmb):
        self.lmb=lmb
        self.y=0

    def compute(self, x):
        self.y = self.lmb * self.y + (1 - self.lmb) * x
        return self.y
```

Testing the code

```
>>> from leaky import Leaky
>>> li = Leaky(0.95)
>>> for v in [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]:
>>>     print li.compute(v)

[0.0, 0.0, 0.0, 0.0, 0.0500000000000000, 0.0475000000000000,
 0.0451250000000000, 0.0428687500000000, 0.0407253125000000,
 0.0386890468750000, 0.0367545945312500]
>>>
```

Key points

- we need a "memory cell" to store previous output
- we need to initialize the storage before first use
- we need 2 multiplications and one addition per output sample

Another old friend

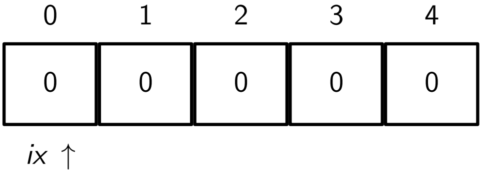
$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k]$$

Another old friend

```
class MA:
    def __init__(self, M):
        self.M = M
        self.buf = [0] * M
        self.ix = 0

    def compute(self, x):
        self.buf[self.ix] = x
        self.ix = (self.ix + 1) % self.M
        res = 0.0
        for v in self.buf:
            res += v
        return res / self.M
```

The circular buffer



The circular buffer

0	1	2	3	4
x_0	0	0	0	0
	$ix \uparrow$			

The circular buffer

0	1	2	3	4
x_0	x_1	0	0	0

$ix \uparrow$

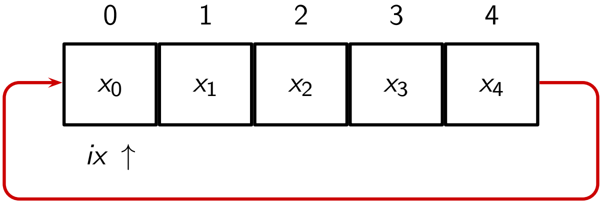
The circular buffer

0	1	2	3	4
x_0	x_1	x_2	0	0
			$ix \uparrow$	

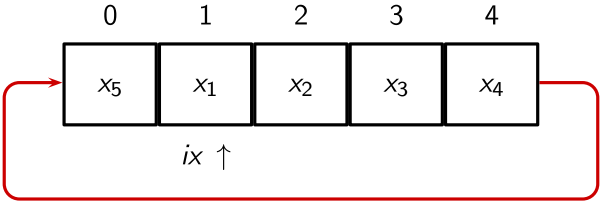
The circular buffer

0	1	2	3	4
x_0	x_1	x_2	x_3	0
				ix $\uparrow$

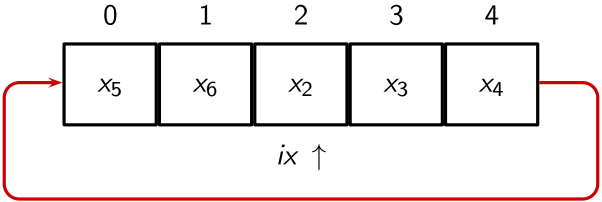
The circular buffer



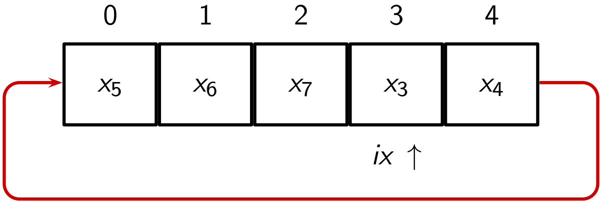
The circular buffer



The circular buffer



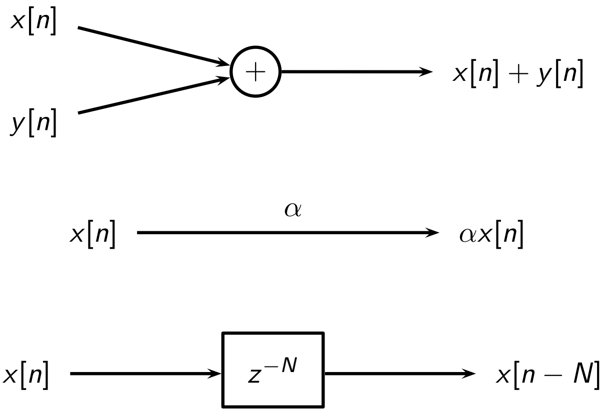
The circular buffer



Key points

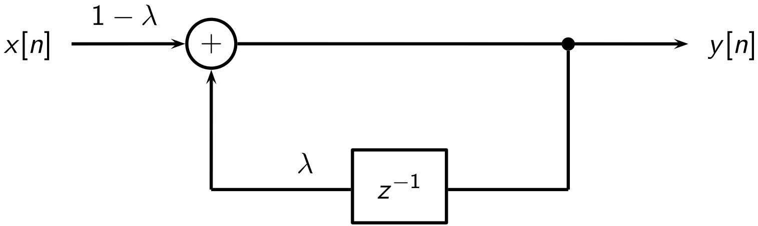
- we now need M memory cells to store previous input values
- we need to initialize the storage before first use
- we need 1 division and M additions per output sample

We can abstract from the implementation



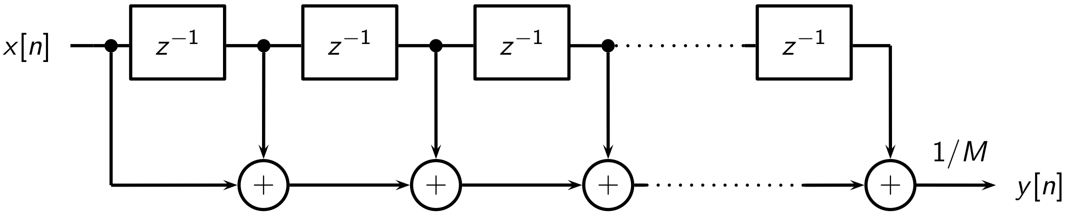
Leaky Integrator

$$y[n] = \lambda y[n-1] + (1-\lambda)x[n]$$



Moving Average

$$y[n] = \frac{1}{M} \sum_{k=0}^{M-1} x[n-k]$$



The second-order section

$$y[n] + a_1 y[n-1] + a_2 y[n-2] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

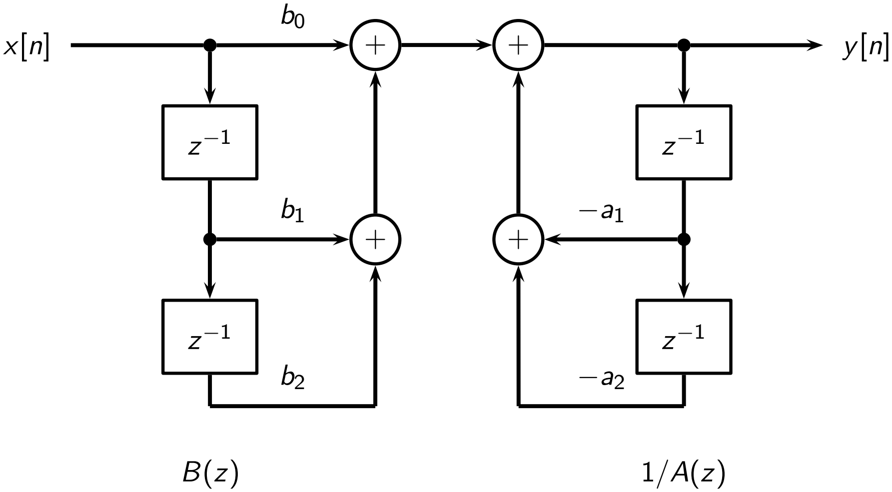
$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{B(z)}{A(z)}$$

The second-order section

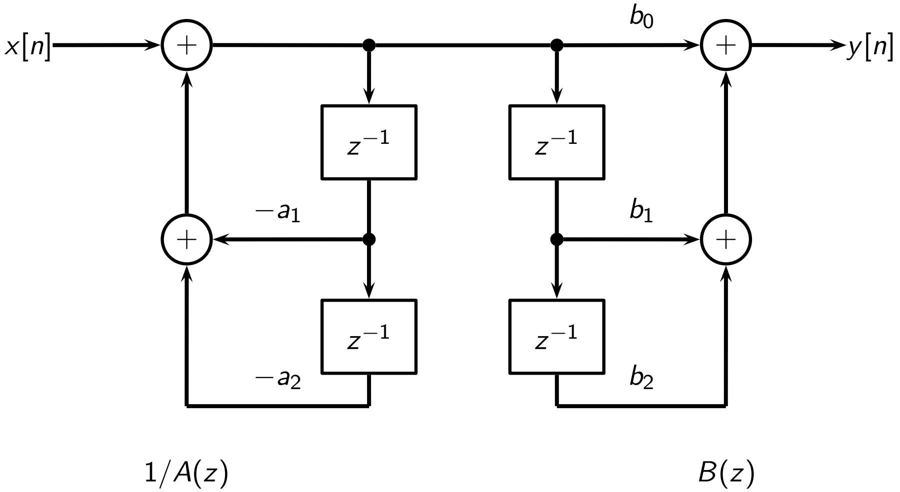
$$y[n] + a_1 y[n-1] + a_2 y[n-2] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{B(z)}{A(z)}$$

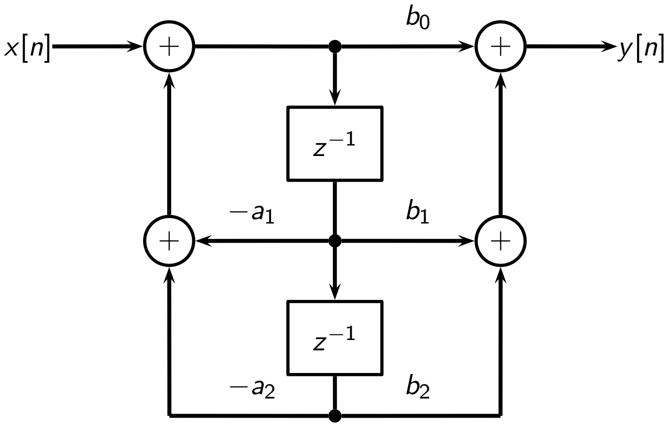
Second-order section, direct form I



Second-order section, direct form I, inverted order



Second-order section, direct form II



intuitive filter design

Overview:

- General problem
- “Intuitive” IIR design

Simple, useful filters

- many signal processing problems can be solved using simple filters
- we have seen simple lowpass filters already (Moving Average, Leaky Integrator)
- simple (low order) transfer functions allow for intuitive design and tuning

Simple lowpass

- let only low frequencies pass
- used to remove high frequency components (e.g. noise)
- useful in audio, communication, control systems
- we know a simple answer: leaky integrator

Simple lowpass

- let only low frequencies pass
- used to remove high frequency components (e.g. noise)
- useful in audio, communication, control systems
- we know a simple answer: leaky integrator

Simple lowpass

- let only low frequencies pass
- used to remove high frequency components (e.g. noise)
- useful in audio, communication, control systems
- we know a simple answer: leaky integrator

Simple lowpass

- let only low frequencies pass
- used to remove high frequency components (e.g. noise)
- useful in audio, communication, control systems
- we know a simple answer: leaky integrator

Leaky Integrator

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

Leaky Integrator

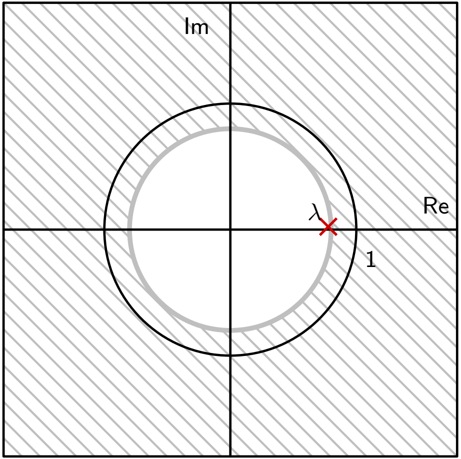
$$H(z) = \frac{(1-\lambda)}{1-\lambda z^{-1}}$$

$$y[n] = (1-\lambda)x[n] + \lambda y[n-1]$$

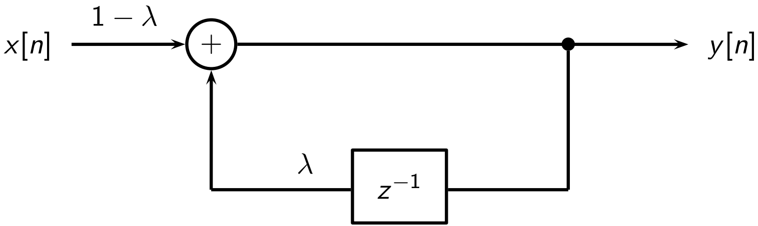
Leaky Integrator

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

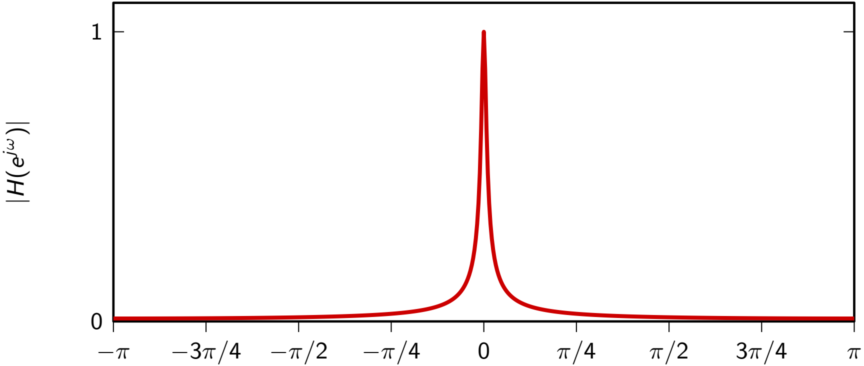
$$y[n] = (1-\lambda)x[n] + \lambda y[n-1]$$



Leaky Integrator, filter structure



Leaky Integrator, $\lambda = 0.98$



Resonator

- a resonator is a narrow bandpass filter
- used to detect the presence of a sinusoid of a given frequency
- useful in communication systems and telephony (DTMF)
- idea: shift the passband of the Leaky Integrator!

Resonator

- a resonator is a narrow bandpass filter
- used to detect the presence of a sinusoid of a given frequency
- useful in communication systems and telephony (DTMF)
- idea: shift the passband of the Leaky Integrator!

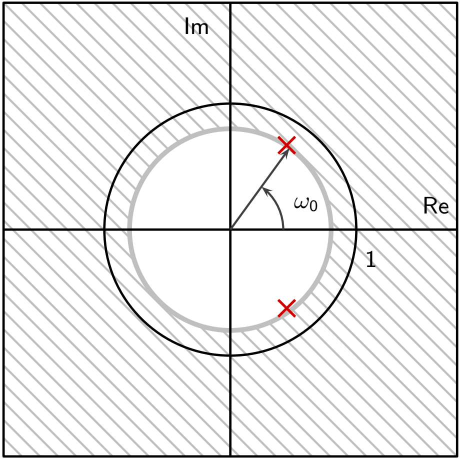
Resonator

- a resonator is a narrow bandpass filter
- used to detect the presence of a sinusoid of a given frequency
- useful in communication systems and telephony (DTMF)
- idea: shift the passband of the Leaky Integrator!

Resonator

- a resonator is a narrow bandpass filter
- used to detect the presence of a sinusoid of a given frequency
- useful in communication systems and telephony (DTMF)
- idea: shift the passband of the Leaky Integrator!

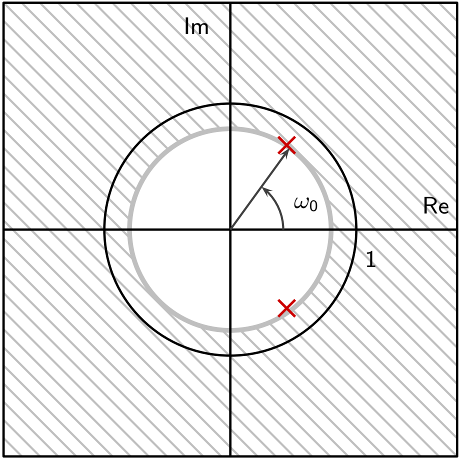
Resonator



Resonator

$$H(z) = \frac{G_0}{(1 - pz^{-1})(1 - p^* z^{-1})}$$

$$p = \lambda e^{j\omega_0}$$

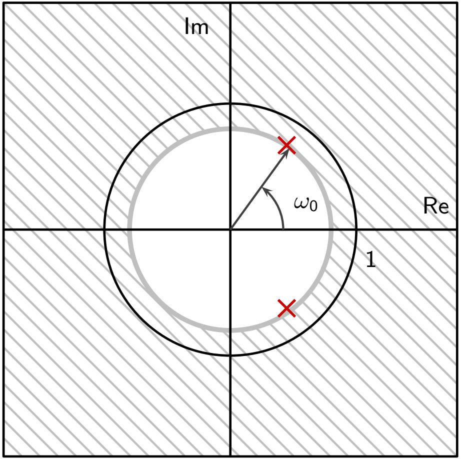


Resonator

$$H(z) = \frac{G_0}{(1 - pz^{-1})(1 - p^* z^{-1})}$$

$$p = \lambda e^{j\omega_0}$$

$$y[n] = G_0 x[n] - a_1 y[n-1] - a_2 y[n-2]$$



Resonator

$$H(z) = \frac{G_0}{(1 - pz^{-1})(1 - p^* z^{-1})}, \qquad p = \lambda e^{j\omega_0}$$

$$= \frac{G_0}{1 - 2\Re\{p\}\, z^{-1} + |p|^2\, z^{-2}}$$

$$= \frac{G_0}{1 - 2\lambda \cos\omega_0\, z^{-1} + |\lambda|^2\, z^{-2}}$$

$$a_1 = 2\lambda \cos\omega_0$$

$$a_2 = -|\lambda|^2$$

Resonator

$$
\begin{aligned}
H(z) &= \frac{G_0}{(1 - pz^{-1})(1 - p^* z^{-1})}, \qquad p = \lambda e^{j\omega_0} \\
&= \frac{G_0}{1 - 2\Re\{p\}\, z^{-1} + |p|^2\, z^{-2}} \\
&= \frac{G_0}{1 - 2\lambda \cos\omega_0\, z^{-1} + |\lambda|^2\, z^{-2}}
\end{aligned}
$$

$$
\begin{aligned}
a_1 &= 2\lambda \cos\omega_0 \\
a_2 &= -|\lambda|^2
\end{aligned}
$$

Resonator

$$H(z) = \frac{G_0}{(1 - pz^{-1})(1 - p^* z^{-1})}, \qquad p = \lambda e^{j\omega_0}$$

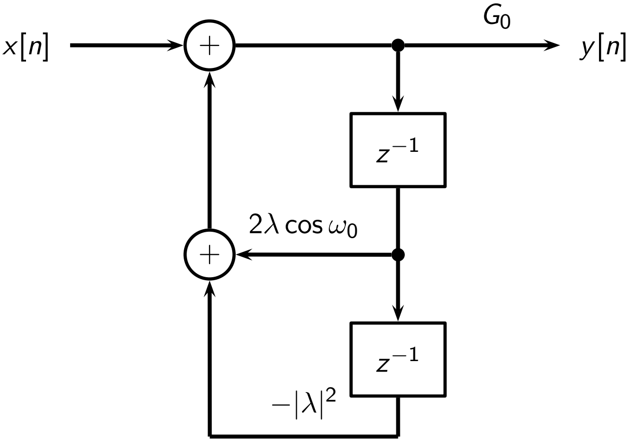
$$= \frac{G_0}{1 - 2\Re\{p\}\, z^{-1} + |p|^2\, z^{-2}}$$

$$= \frac{G_0}{1 - 2\lambda \cos\omega_0\, z^{-1} + |\lambda|^2\, z^{-2}}$$

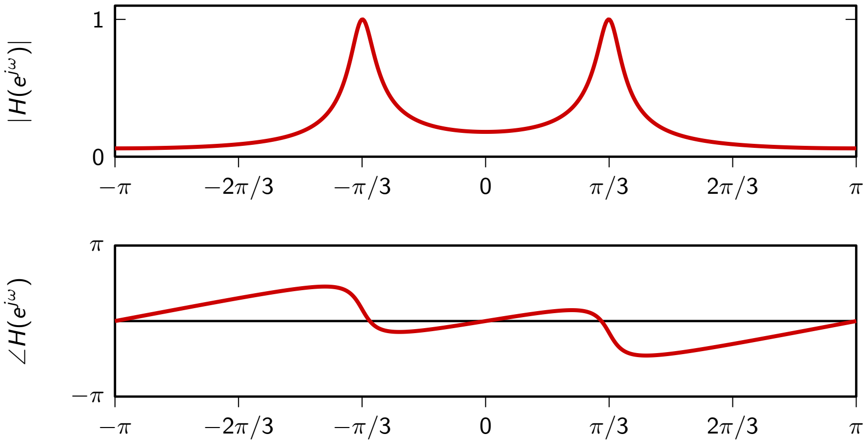
$$a_1 = 2\lambda \cos\omega_0$$

$$a_2 = -|\lambda|^2$$

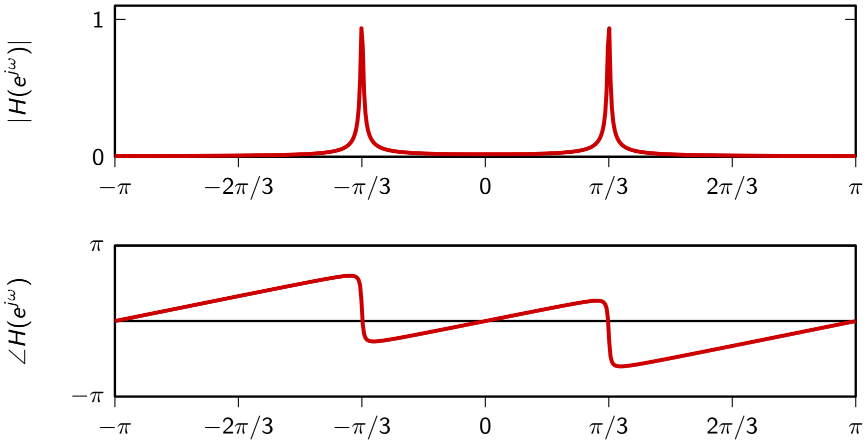
Resonator, filter structure



Resonator, $\lambda = 0.9, \omega_0 = \pi/3$



Resonator, $\lambda = 0.99, \omega_0 = \pi/3$



DC removal

- a DC-balanced signal has zero sum: $\lim_{N\to\infty} \sum_{n=-N}^{N} x[n] = 0$

  i.e. there is no Direct Current component
- its DTFT value at zero is zero
- we want to remove the DC bias from a non zero-centered signal
- we want to kill the frequency component at $\omega = 0$

DC removal

- a DC-balanced signal has zero sum: $\lim_{N\to\infty} \sum_{n=-N}^{N} x[n] = 0$

  i.e. there is no Direct Current component
- its DTFT value at zero is zero
- we want to remove the DC bias from a non zero-centered signal
- we want to kill the frequency component at $\omega = 0$

DC removal

- a DC-balanced signal has zero sum: $\lim_{N\to\infty} \sum_{n=-N}^{N} x[n] = 0$

  i.e. there is no Direct Current component
- its DTFT value at zero is zero
- we want to remove the DC bias from a non zero-centered signal
- we want to kill the frequency component at $\omega = 0$

DC removal

- a DC-balanced signal has zero sum: $\lim_{N \to \infty} \sum_{n=-N}^{N} x[n] = 0$

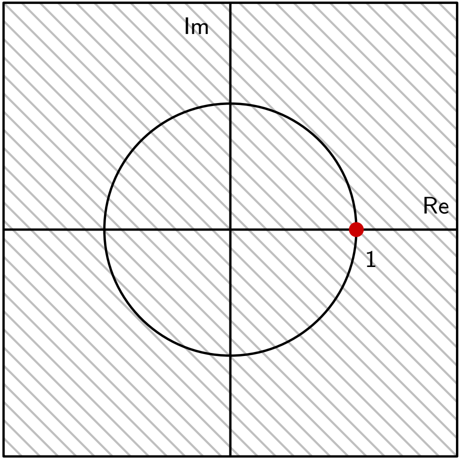
  i.e. there is no Direct Current component

- its DTFT value at zero is zero

- we want to remove the DC bias from a non zero-centered signal

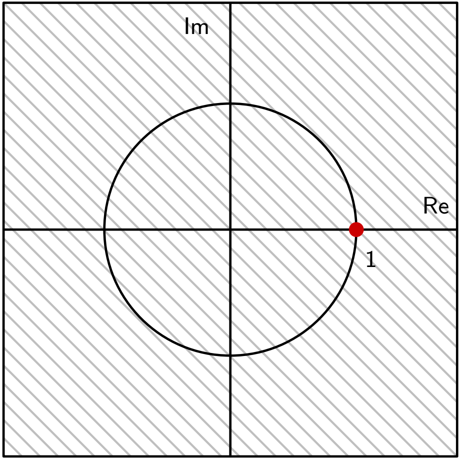
- we want to kill the frequency component at $\omega = 0$

DC removal



DC removal

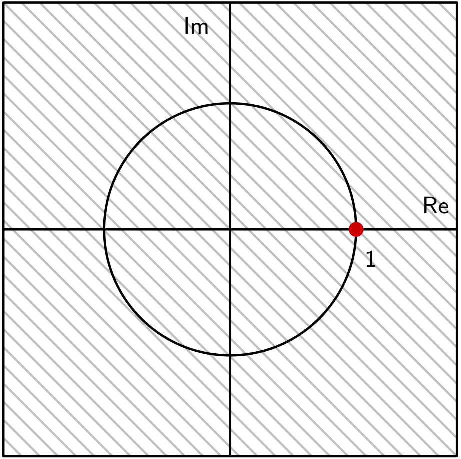
$$H(z) = 1 - z^{-1}$$



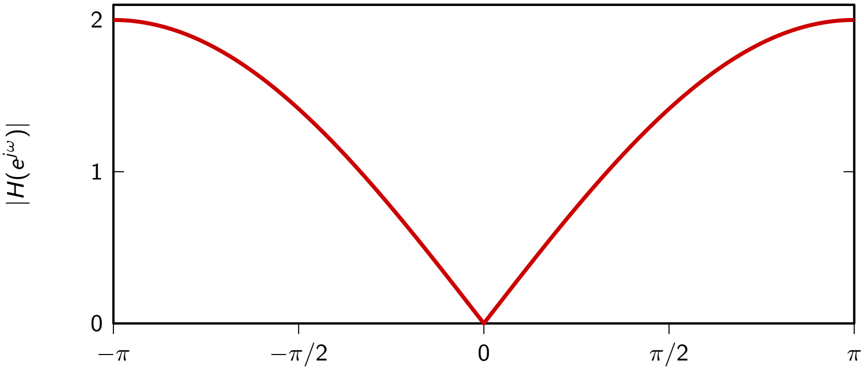
DC removal

$$H(z) = 1 - z^{-1}$$

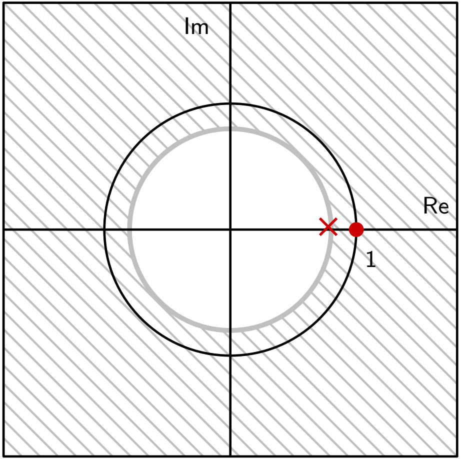
$$y[n] = x[n] - x[n-1]$$



DC notch

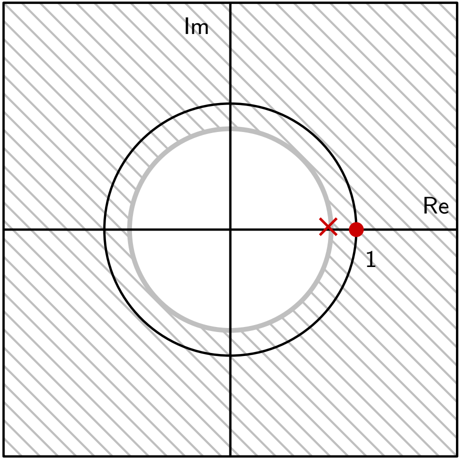


DC removal, improved



DC removal, improved

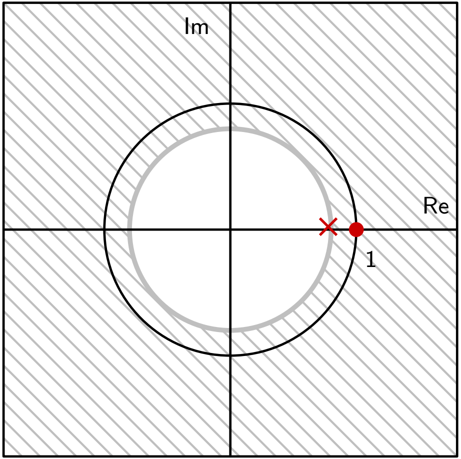
$$H(z) = \frac{1 - z^{-1}}{1 - \lambda z^{-1}}$$



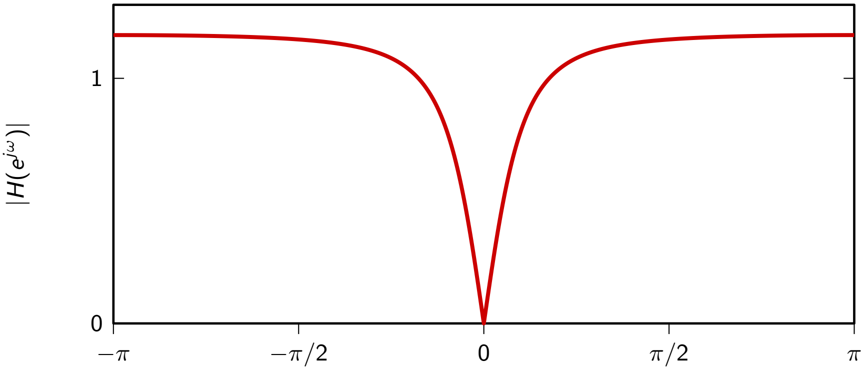
DC removal, improved

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

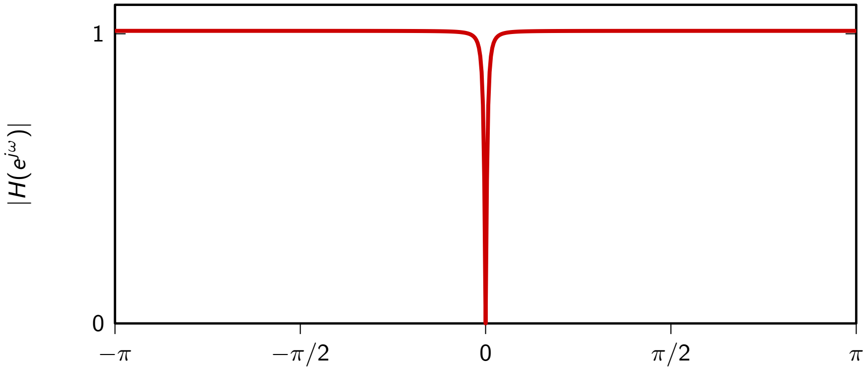
$$y[n] = \lambda y[n-1] + x[n] - x[n-1]$$



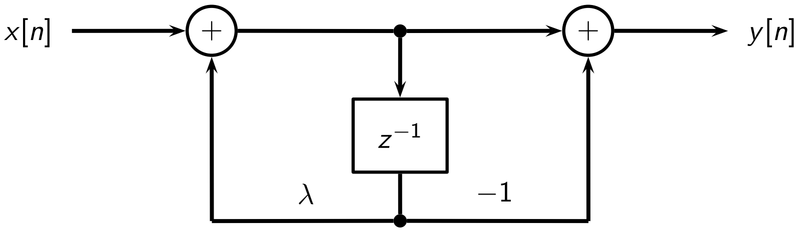
DC notch, $\lambda = 0.7$



DC notch, $\lambda = 0.98$



DC notch, filter structure



Hum removal

- similar to DC removal but we want to remove a specific nonzero frequency
- very useful for musicians: amplifiers for electric guitars pick up the hum from the electric mains (50Hz in Europe and 60Hz in North America)
- we need to tune the hum removal according to country

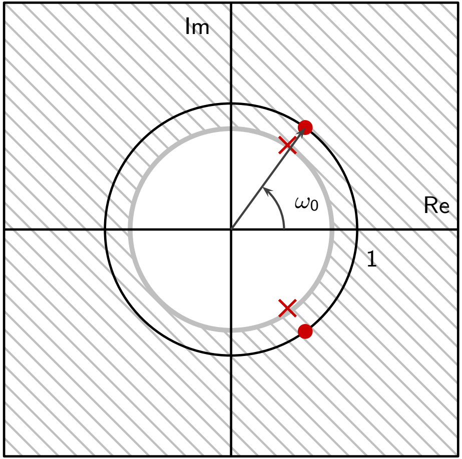
Hum removal

- similar to DC removal but we want to remove a specific nonzero frequency
- very useful for musicians: amplifiers for electric guitars pick up the hum from the electric mains (50Hz in Europe and 60Hz in North America)
- we need to tune the hum removal according to country

Hum removal

- similar to DC removal but we want to remove a specific nonzero frequency
- very useful for musicians: amplifiers for electric guitars pick up the hum from the electric mains (50Hz in Europe and 60Hz in North America)
- we need to tune the hum removal according to country

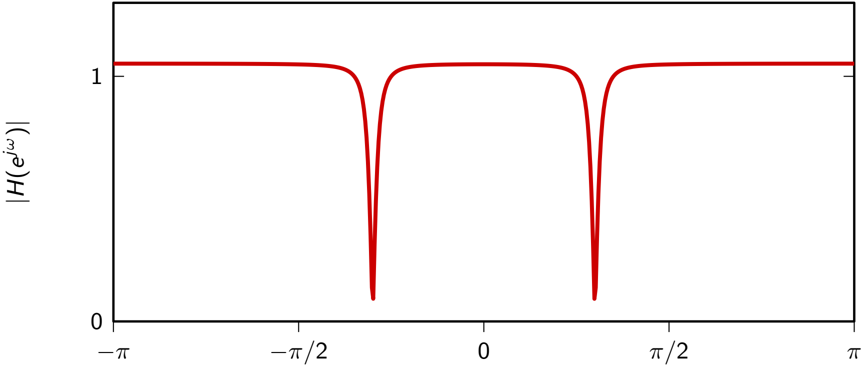
Hum removal



Hum removal

$$H(z) = \frac{(1 - e^{j\omega_0} z^{-1})(1 - e^{-j\omega_0} z^{-1})}{(1 - \lambda e^{j\omega_0} z^{-1})(1 - \lambda e^{-j\omega_0} z^{-1})}$$

Hum removal, $\lambda = 0.95$



Hum removal, filter structure