So far

- We have introduced the $\mathcal{Z}$ transform
  - The digital equivalent of the Laplace transform
  - It facilitates the solving of difference equations
  - It allows to easily evaluate the system's response
  - It is critical in designing linear filters
- The Discrete-time Fourier Transform (DTFT)
  - $z = e^{j\omega}$ in the $\mathcal{Z}$ transform
  - Mapping into frequency space
- The Discrete Fourier Transform (DFT)
  - The sampling of the DTFT in the frequency domain
  - FFT: Fast Fourier Transform
    - Algorithm for the efficient computation of DFTs
- Sampling principle
  - The signal's bandwidth F_0 must be less than the Nyquist frequency $F_n = F_s/2$ in order to avoid *aliasing*

Digital Signal Processing 3

Digital Filters

- An LTI system to frequency select or discriminate
- Two classes
  - Finite-duration impulse response (FIR) Filters
  - Infinite-duration impulse response (IIR) Filters

FIR filter

- The filter's unit impulse response is of finite duration
  - Its response settles to zero in a finite time
  - There is no "feedback"
- Difference equation

$$y(n) = \sum_{m=0}^{M} b_m \, x(n-m)$$

- Also referred to as *recursive* or *moving average* filters.

IIR filter

- The filter's unit impulse response is of infinite duration
- Difference equation

$$\sum_{k=0}^{N} a_k \; y(n-k) = x(n)$$

- Output is recursively computed from previous computed values → infinite duration response

FIR filter example: Moving Average (MA)

In general

$$y(n) = \sum_{i=0}^{N} b_i \ x(n-i), \qquad b_i = \frac{1}{N+1}$$

For $N = 1$

$$y(n) = \frac{1}{2}\,x(n) + \frac{1}{2}\,x(n-1)$$

To the $\mathcal{Z}$ domain

Recall: when an LTI system is represented by the difference equation

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

Then

$$H(z) = \frac{\sum_{l=0}^{M} b_l \; z^{-l}}{1 + \sum_{k=1}^{N} a_k \; z^{-k}}$$

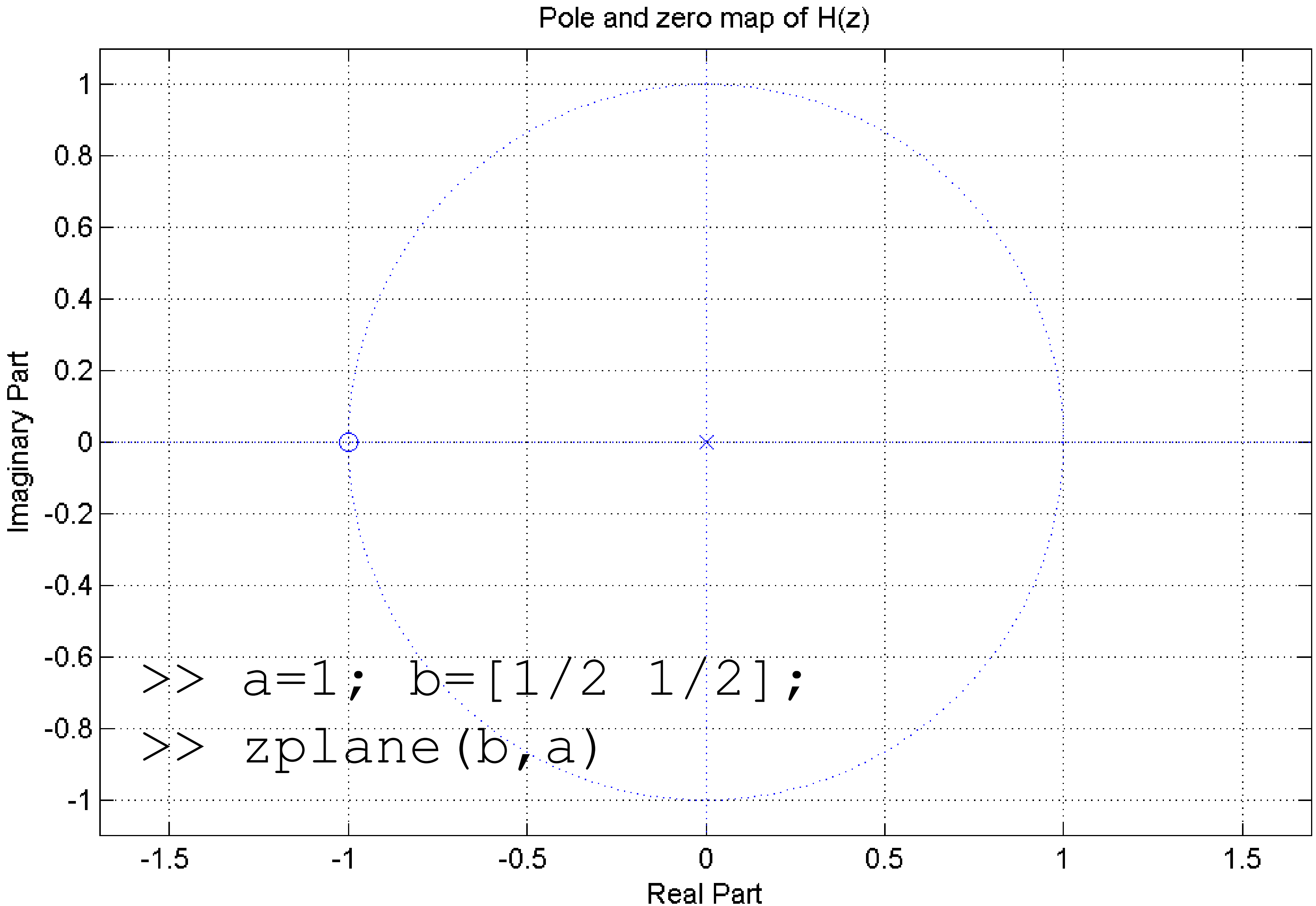
System function $H(z)$

Coefficients *a* and *b* are

$$a_0 = 1, b_0 = b_1 = 1/2$$

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

With a pole at the origin, and a zero at -1.


Pole and zero map of H(z)
Imaginary Part
Real Part
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1.5
-1
-0.5
0
0.5
1
1.5
>> a=1; b=[1/2 1/2];
>> zplane(b,a)

MA_example.m

Recall: difference equation and the `filter` command

- In general, a difference equation is of the form

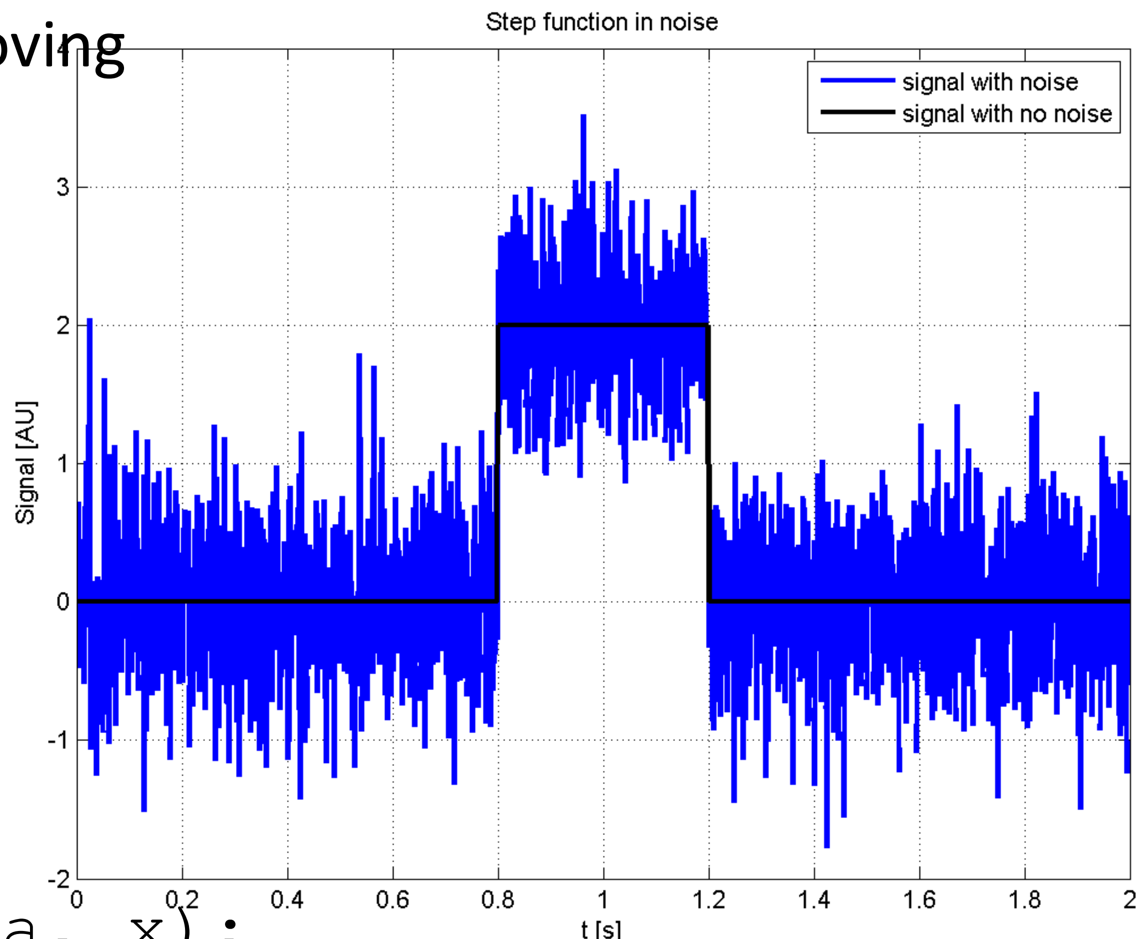
$$y(n) + \sum_{k=1}^{N} a_k \, y(n-k) = \sum_{l=0}^{M} b_l \, x(n-l)$$

- The MATLAB `filter` command solves the difference equations numerically
  - Given the input sequence $x(n)$, the output sequence $y(n)$ is computed using

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

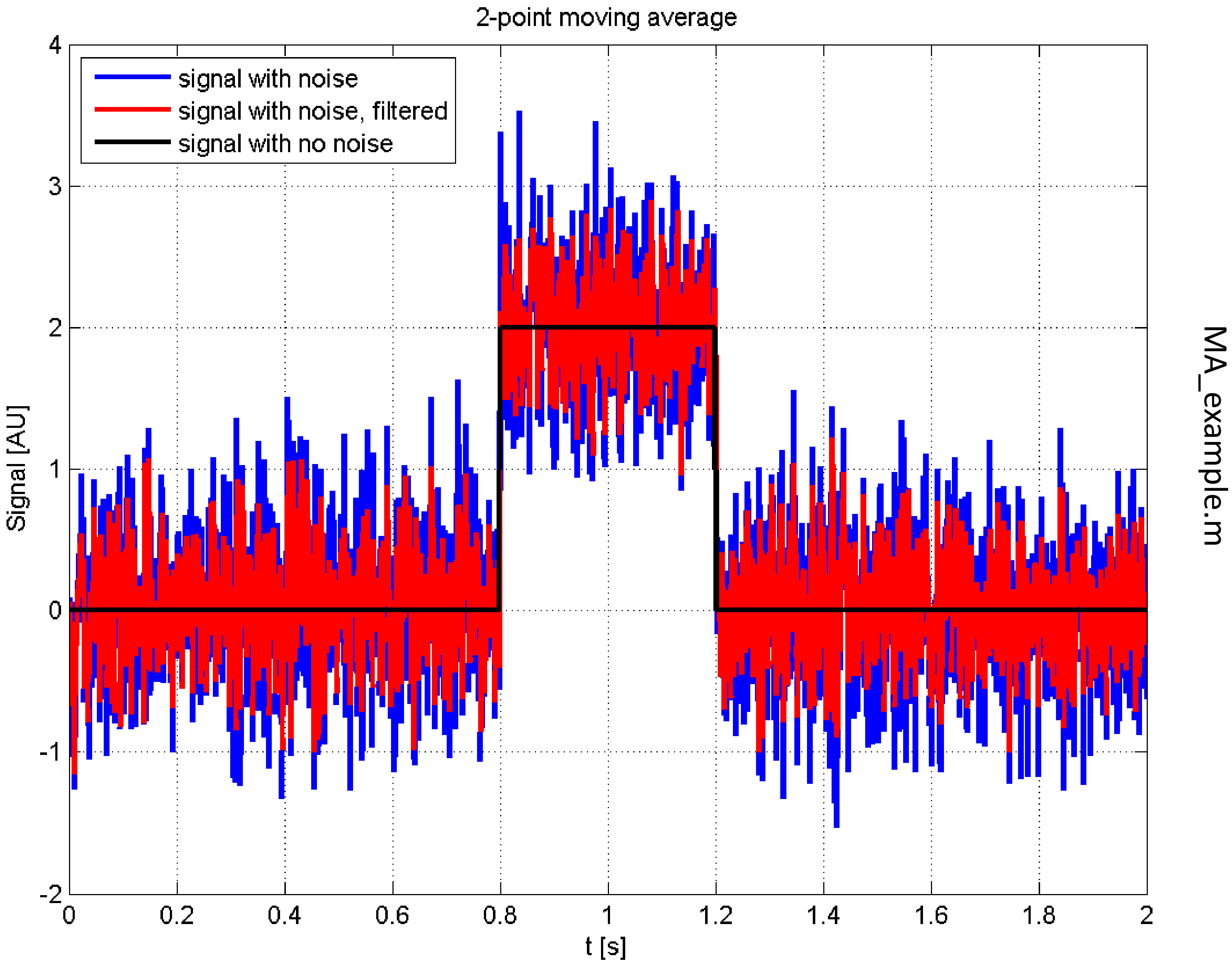
Let's apply the filter to a data stream

- Step function imbedded in noise is shown to the right.
- Let's apply the N=2 moving average filter

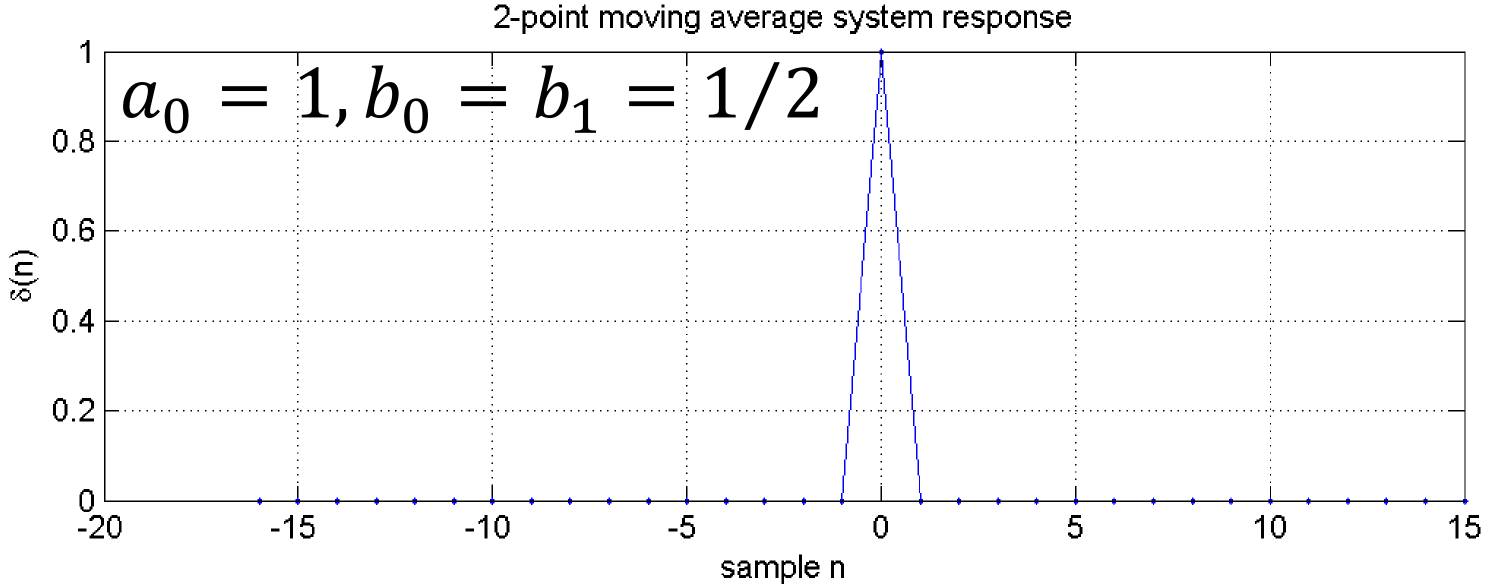



```
>> a=1;
>> b=[1/2 1/2];
>> y = filter(b, a, x);
```

MA_example.m

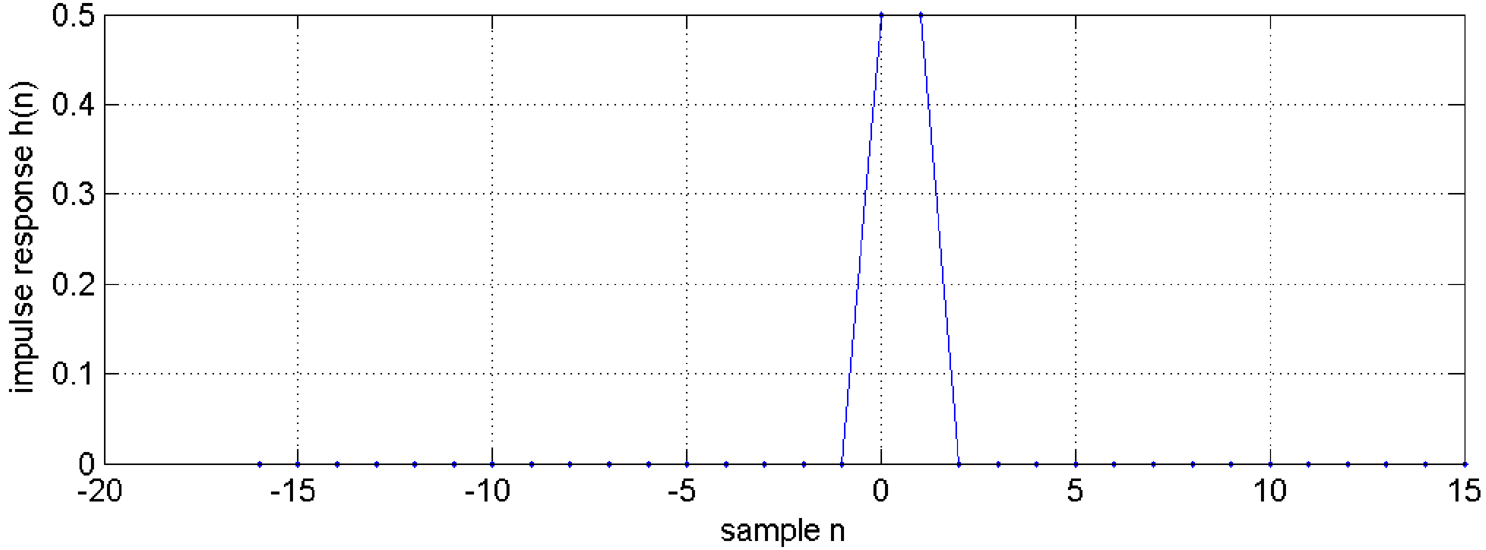

2-point moving average
signal with noise
signal with noise, filtered
signal with no noise
Signal [AU]
t [s]
MA_example.m
4
3
2
1
0
-1
-2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

The impulse response function h



MA_example.m

```
>> h = filter(b, a, delta)
```



The impulse response function h

From the `filter` command

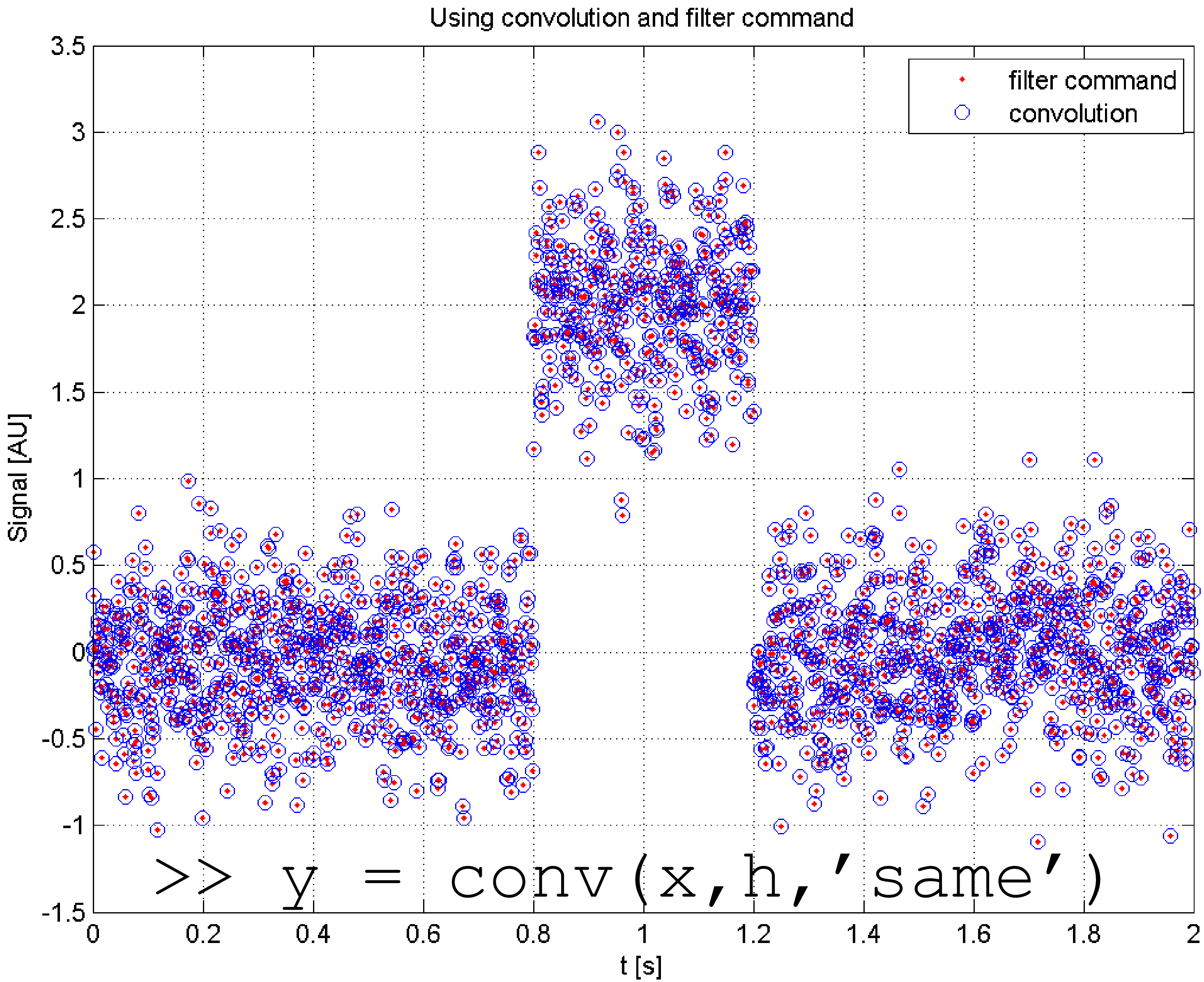
$$h(0) = h(1) = \frac{1}{2}$$

The system's response

$$y(n) = \sum_{i=0}^{N} b_i \; x(n-i) = \sum_{i=0}^{N} b_i \; \delta(n-i) \text{ with } b_i = \frac{1}{N+1}$$

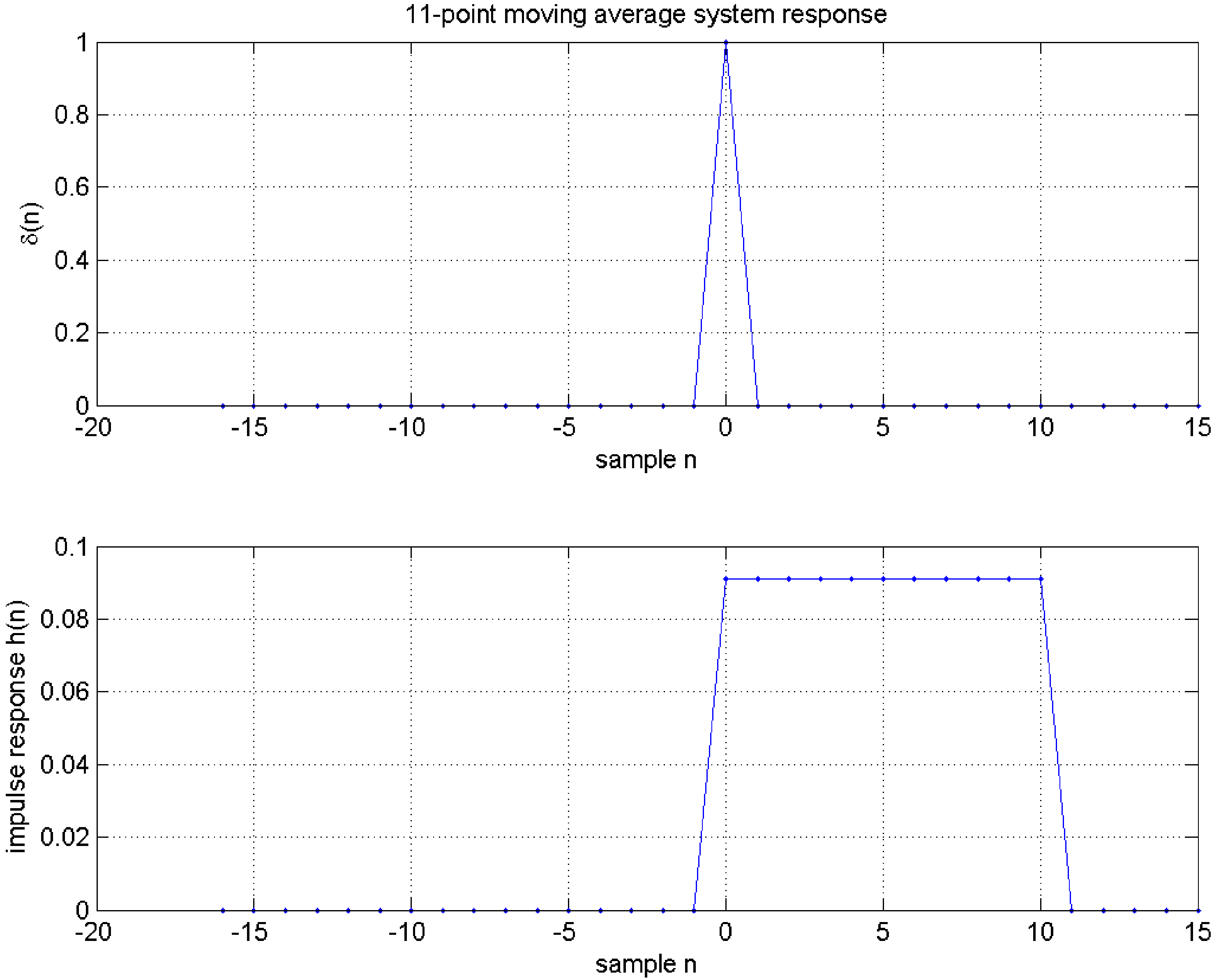
$$h(n) = b_n$$

Comparing the `filter` output with convolution



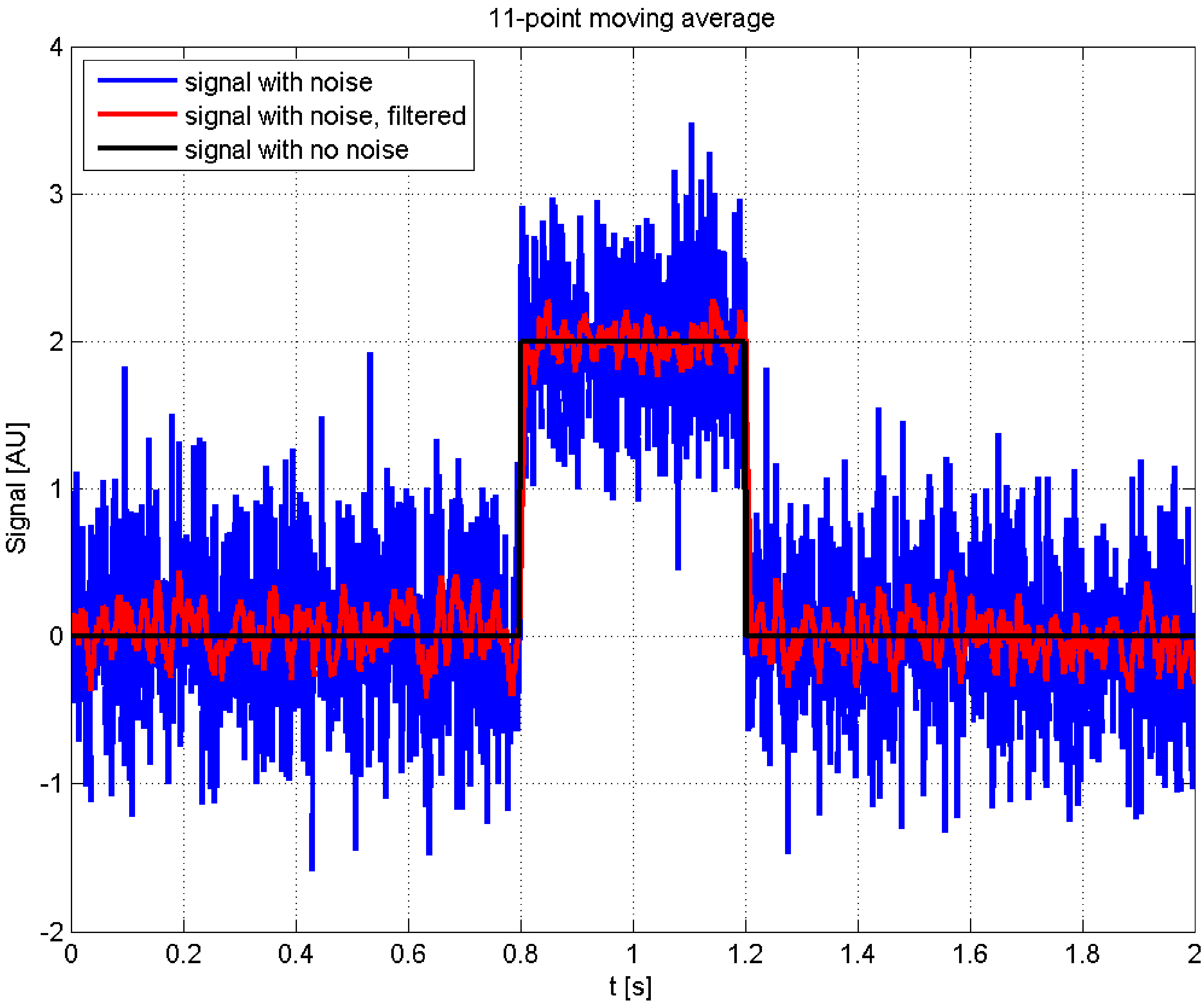
MA_example.m

Increasing filter order



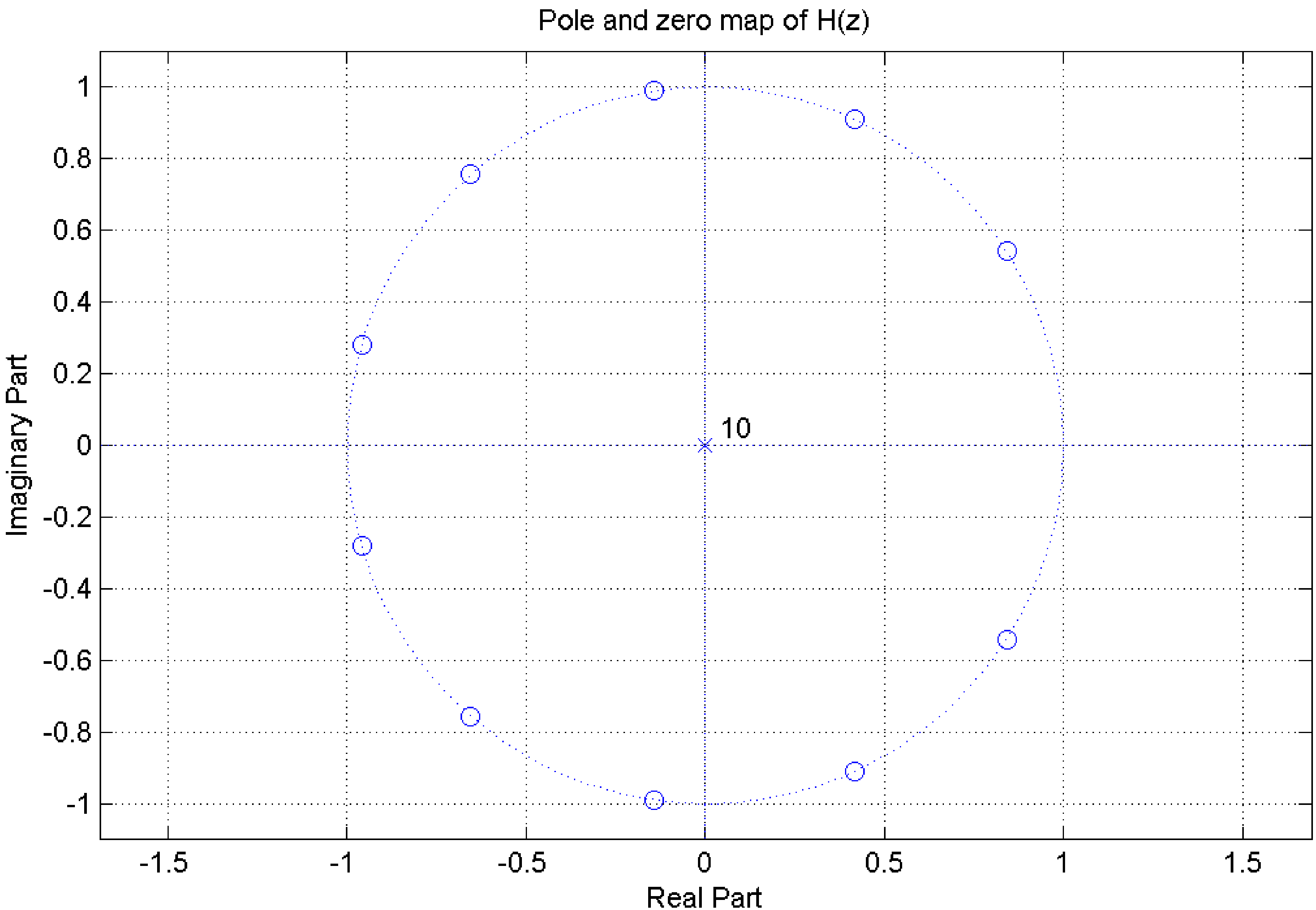
MA_exampleB.m

Increasing filter order



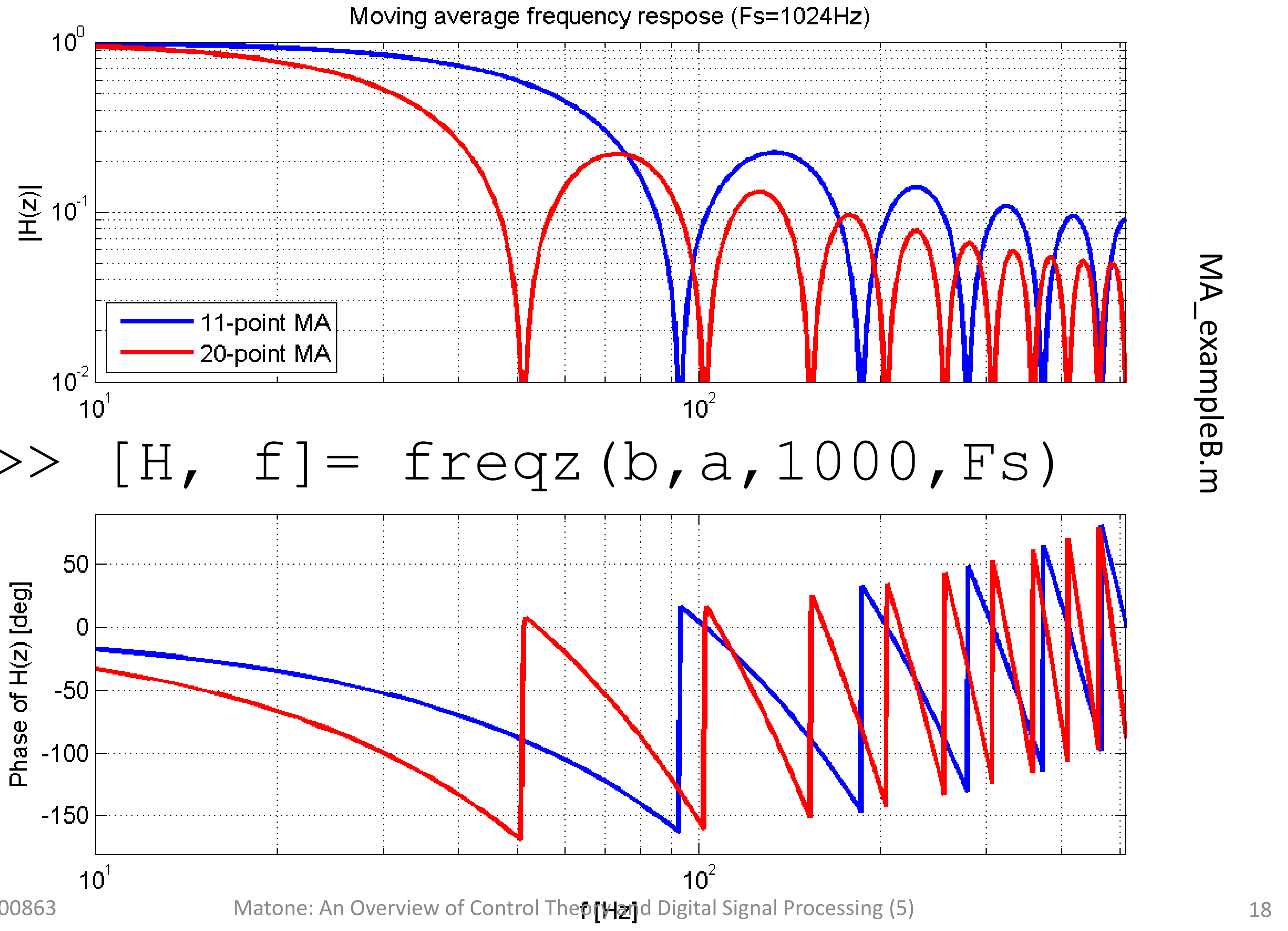
MA_exampleB.m

Increasing filter order



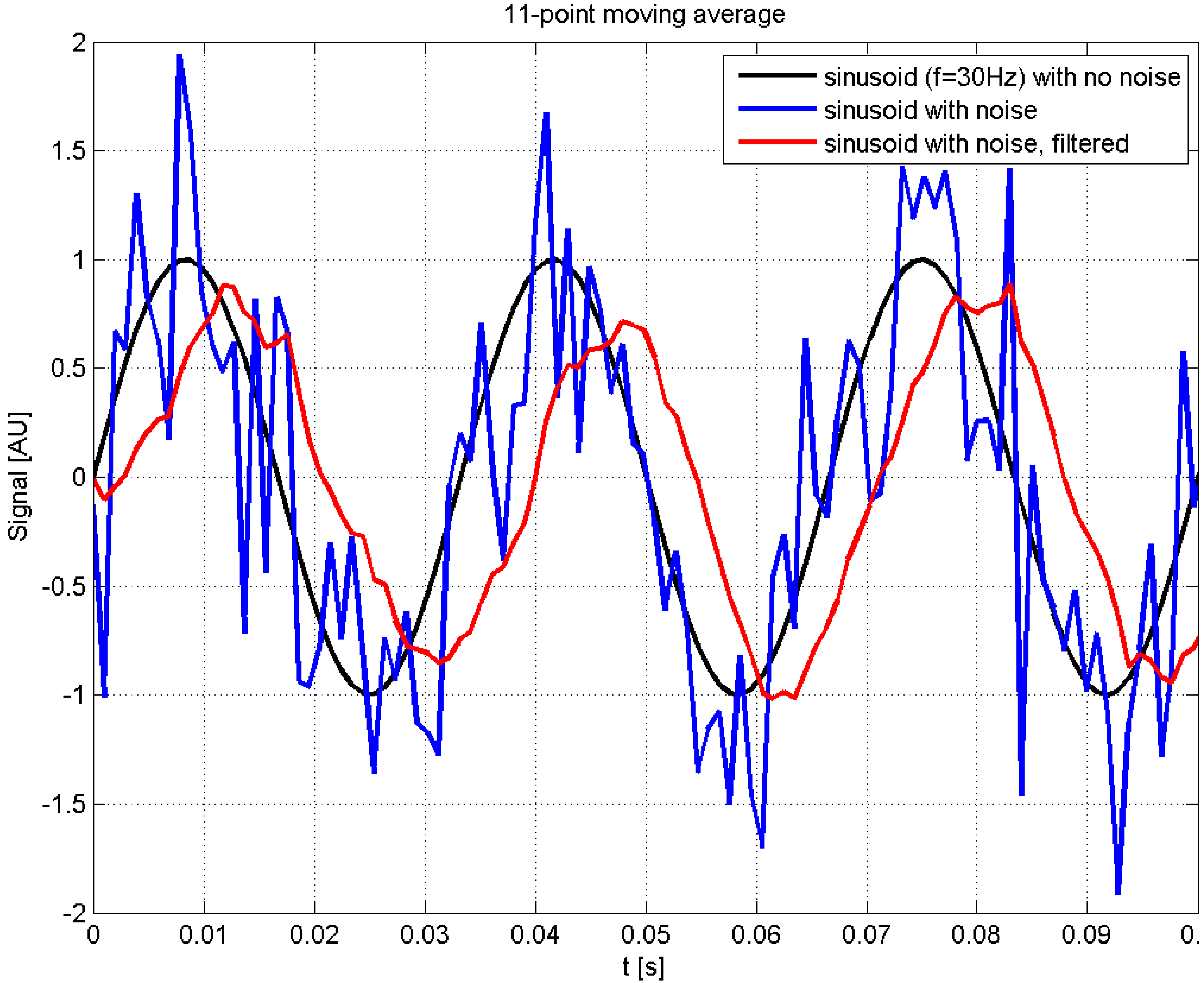
MA_exampleB.m

Frequency response of moving average filter




```
>> [H, f]= freqz(b,a,1000,Fs)
```

Suppression but with phase delay



MA_exampleC.m

Analog-to-digital filter transformation

1. First, we design an analog filter that satisfies the specifications.
2. Then we transform it into the digital domain.

Many transformations are available

- *Impulse invariance*
  - Designed to preserve the shape of the impulse response from analog to digital
- *Finite difference approximation*
  - Specifically designed to convert a differential equation representation to a difference equation representation
- *Step invariance*
  - Designed to preserve the shape of the step response

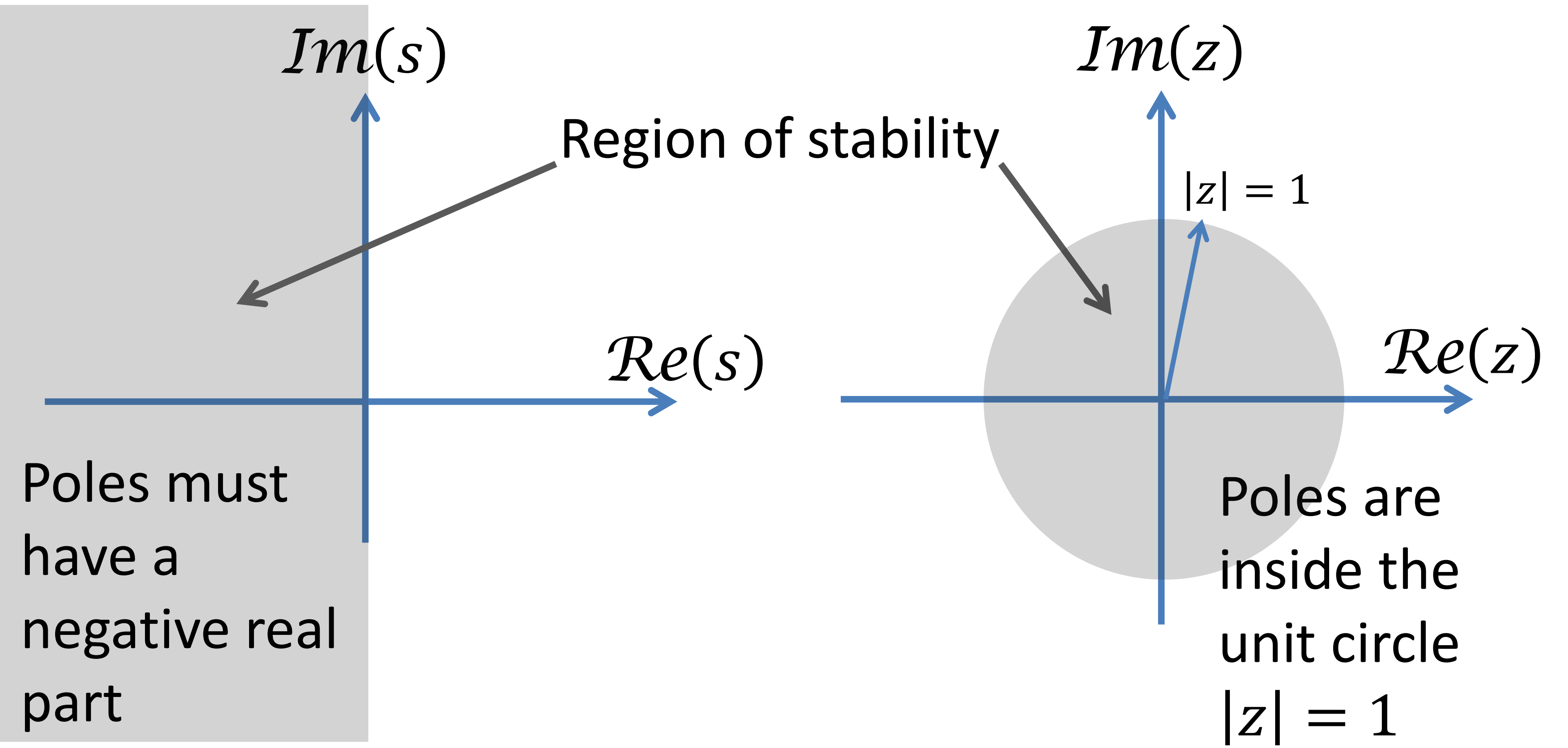
<u>Bilinear transformation</u>

- Most popular technique
- Preserves the system's function representation from analog to digital

Filter stability in the analog and digital domain

Analog domain (s-plane)

Digital domain (z-plane)

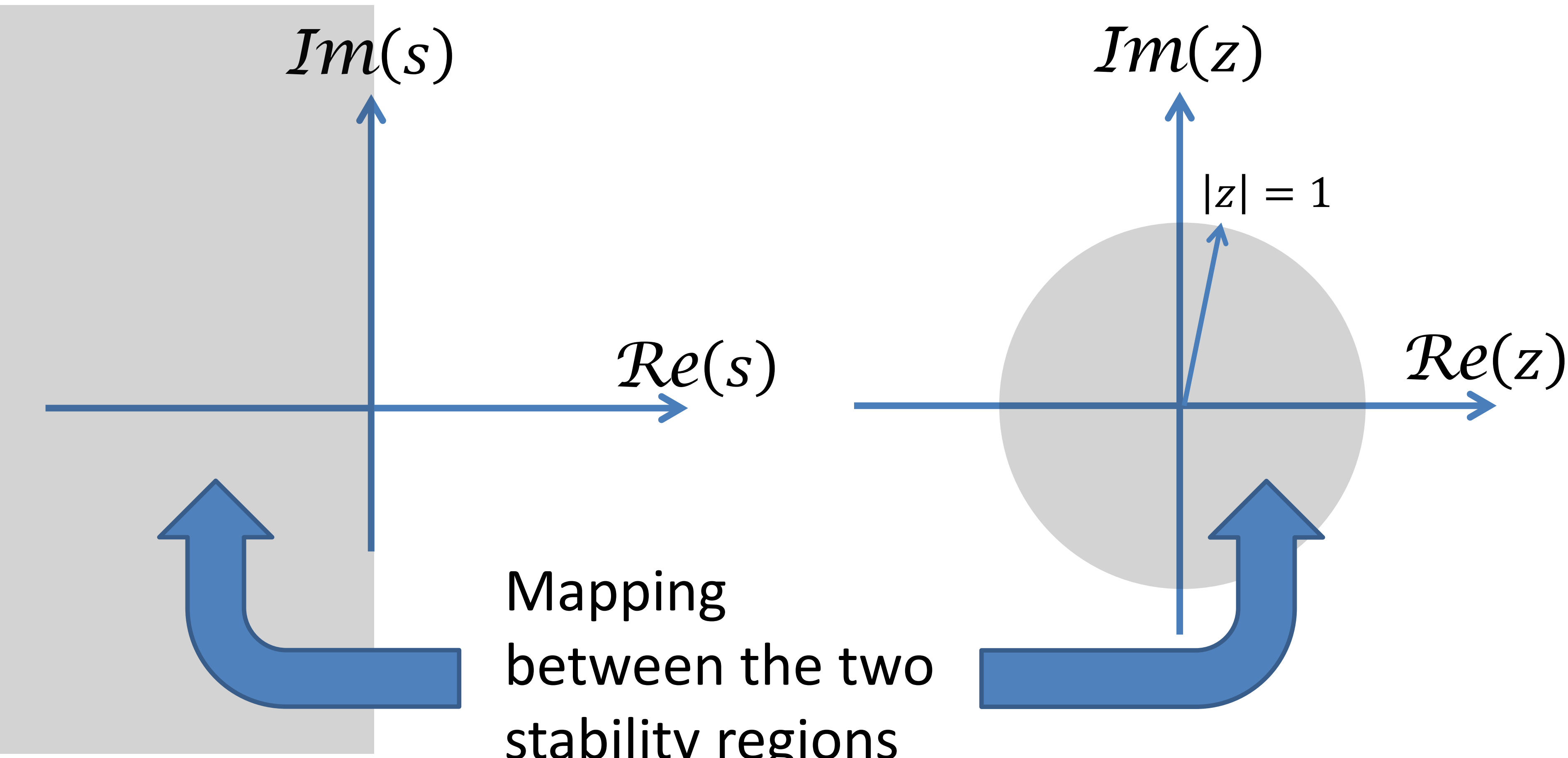


21

Bilinear transformation

Analog domain (s-plane)

Digital domain (z-plane)

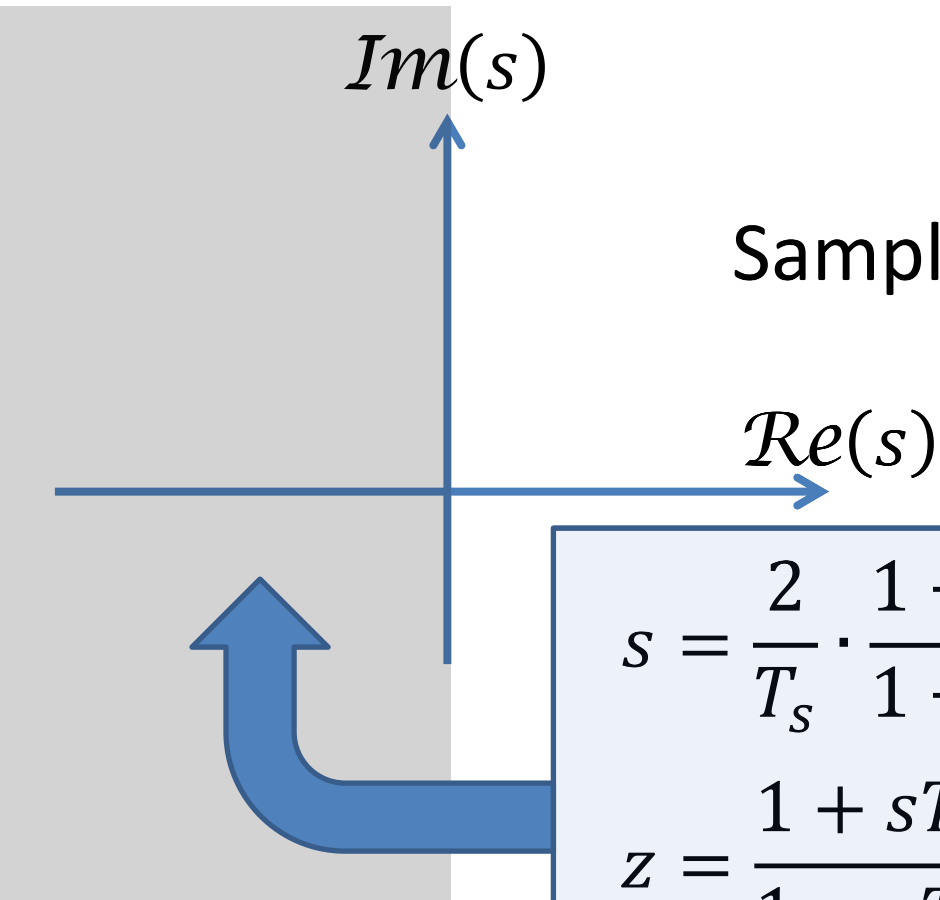


22

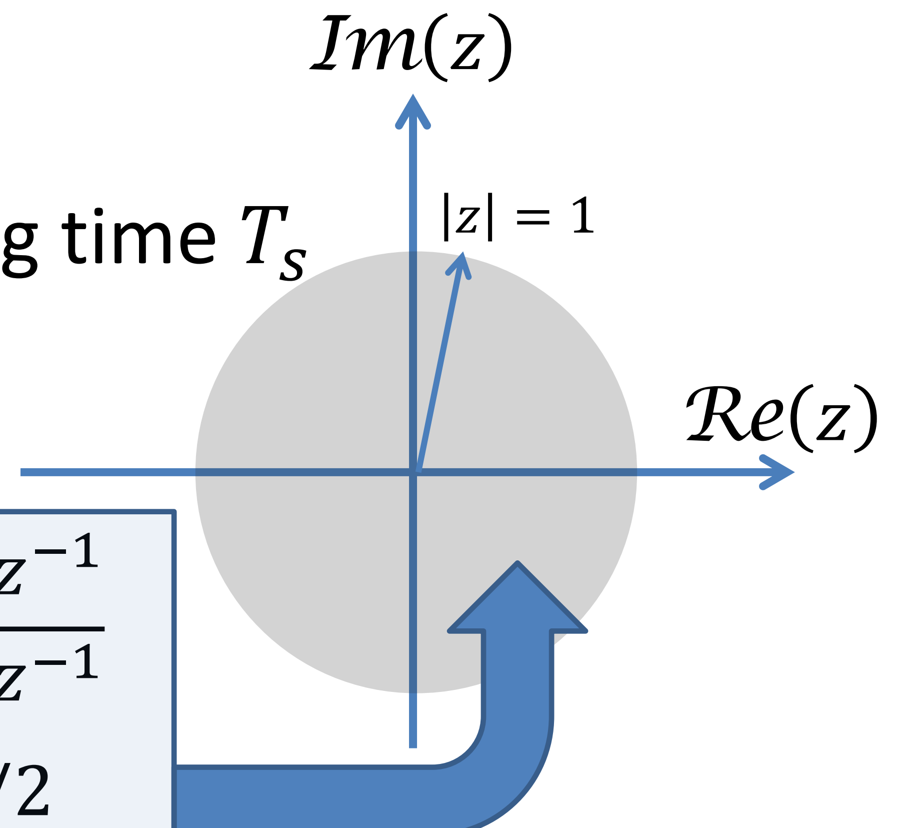
Bilinear transformation

Analog domain (s-plane)

Digital domain (z-plane)



Sampling time T_s



$$s = \frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}$$

$$z = \frac{1 + sT_s/2}{1 - sT_s/2}$$

23

Transformation example

Transform

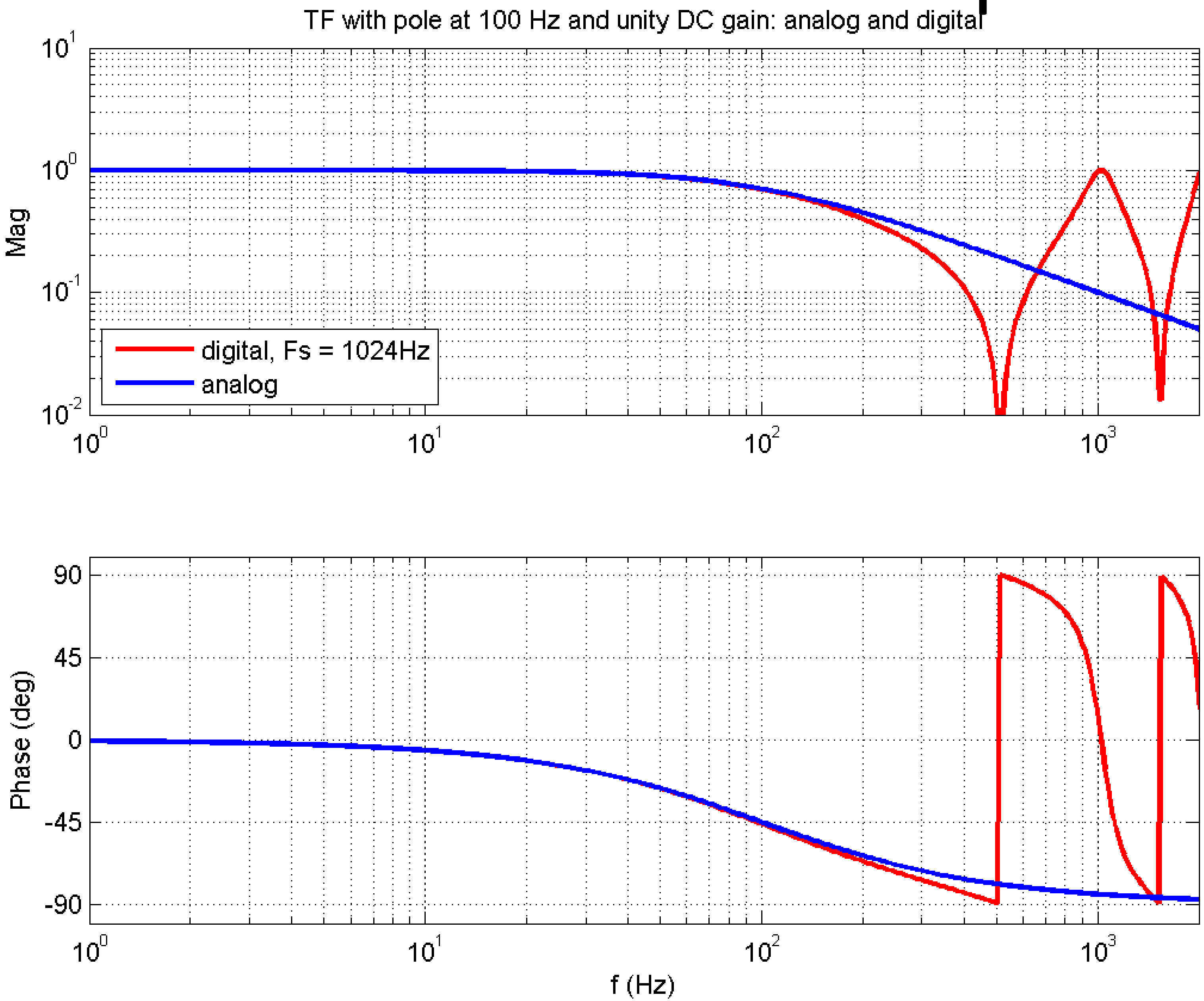
$$H_a(s) = \frac{s+1}{s^2 + 5s + 6}$$

into a digital filter with sampling $T_s = 1\ s$.

Sol.

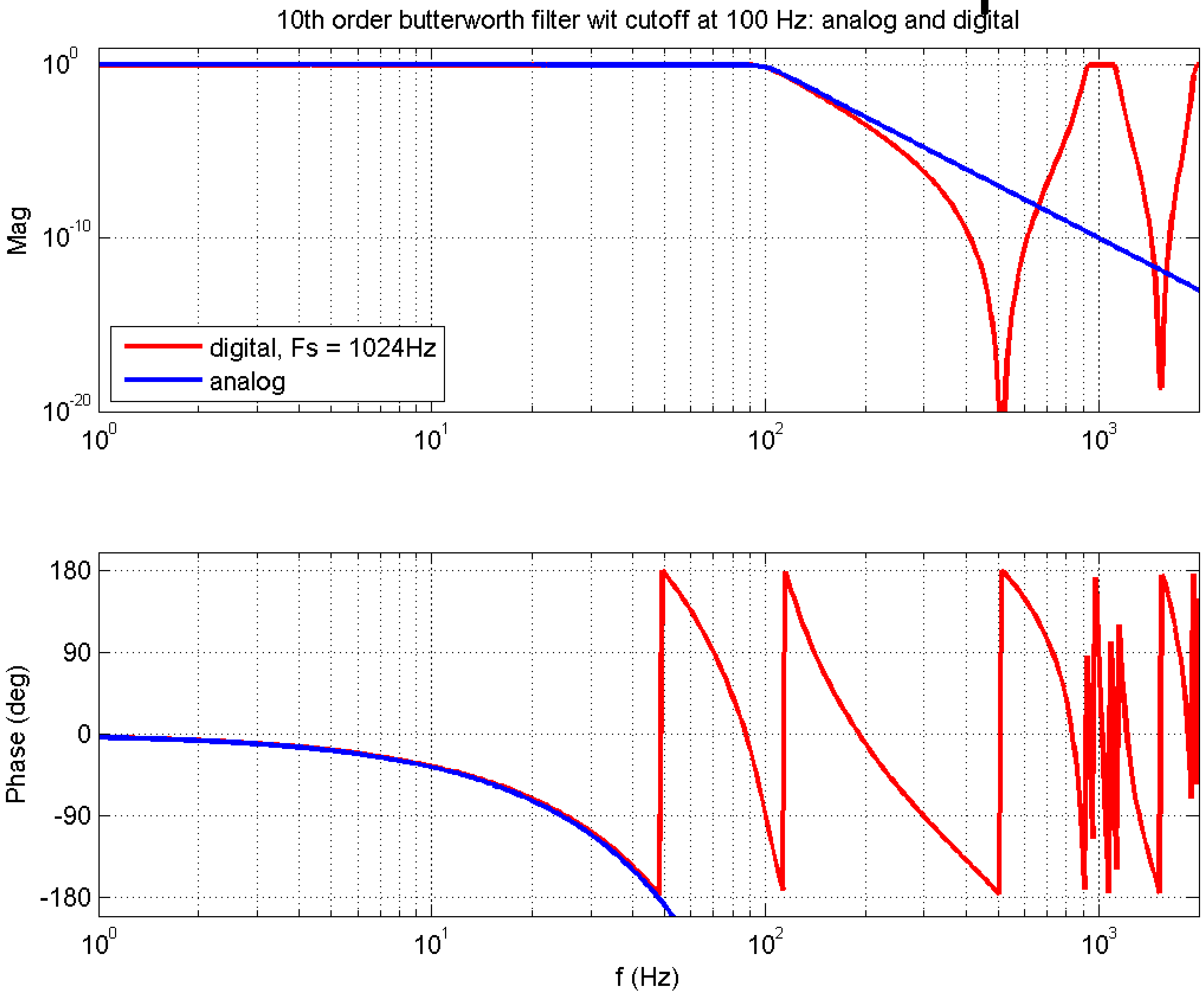
$$H(z) = H_a\left(\frac{2}{0.1} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right) = \frac{3 + 2z^{-1} - z^{-2}}{20 + 4z^{-1}}$$

Transformation example



bilinearexample.m

Transformation example



bilinearexample3.m

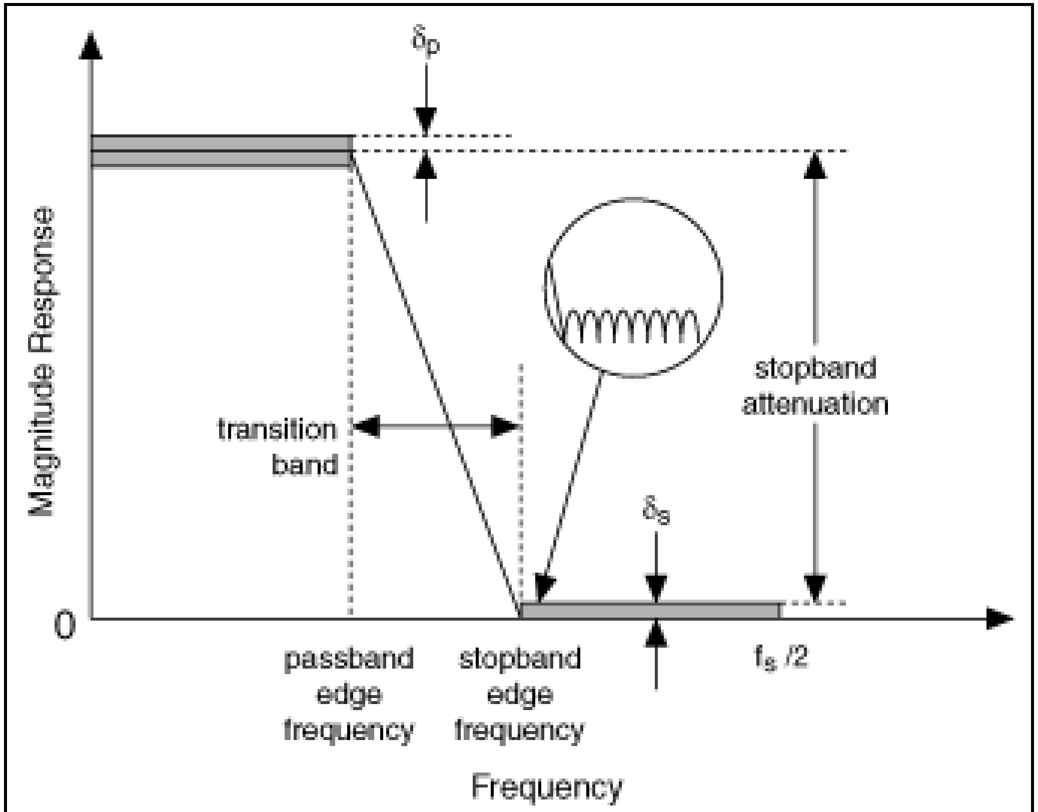
Filter Design

- Filter specifications
  - Constraints on the suppression factor
  - Constraints on the phase response
  - Constraints on the impulse response
  - Constraints on the step response
  - FIR or IIR
  - Filter order
- Typical filters
  - Low pass, High pass, Band pass and Band stop

FIR or IIR?

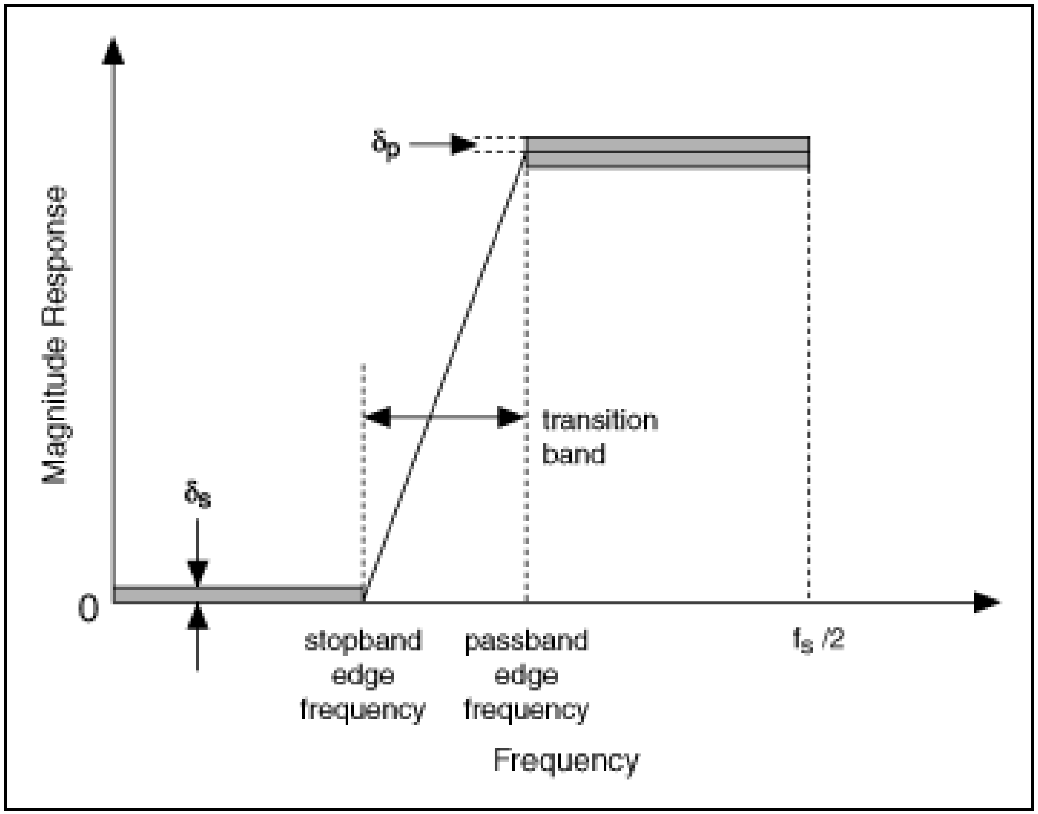
- Advantages of FIR filters over IIR
  - Can be designed to have a “linear phase”. This would “delay” the input signal but would not distort it
  - Simple to implement
  - Always stable
- Disadvantages
  - IIR filters are better in approximating analog systems
  - For a given magnitude response specification, IIR filters often require much less computation than an equivalent FIR

Low pass (LP) filter specifications



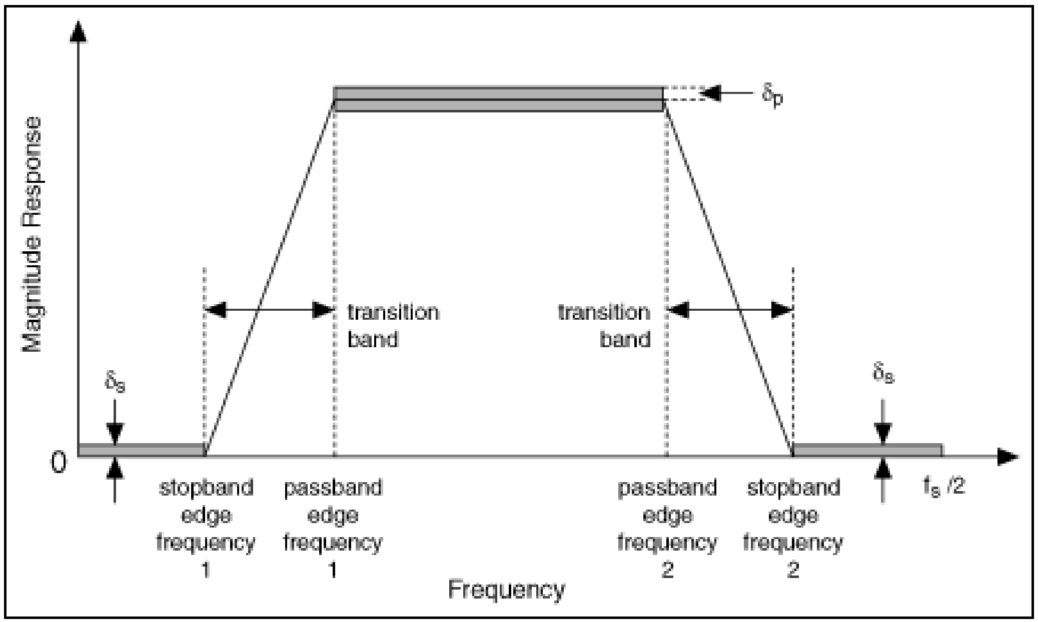
- LP filter
  - low frequencies pass, high frequencies are attenuated.
- Include
  - target magnitude response
  - phase response, and
  - the allowable deviation for each
- Transition band
  - frequency range from the passband edge frequency to the stopband edge frequency
- Ripples
  - The filter passband and stopband can contain oscillations, referred to as ripples. Peak-to-peak value, usually expressed in dB.

High pass (HP) filter specifications



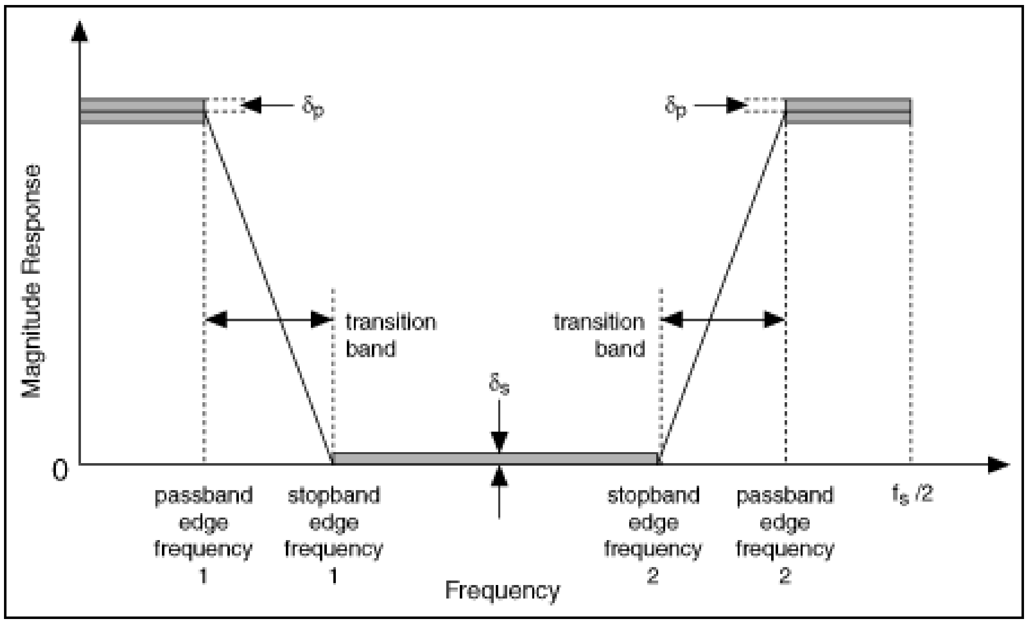
- HP filter
  - High frequencies pass, low frequencies are attenuated.

Band pass (BP) filter specifications



- BP filter
  - a certain band of frequencies pass while lower and higher frequencies are attenuated.

Band stop (BS) filter specifications



- BS filter
  - attenuates a certain band of frequencies and passes all frequencies not within the band.

A few types of IIR filters

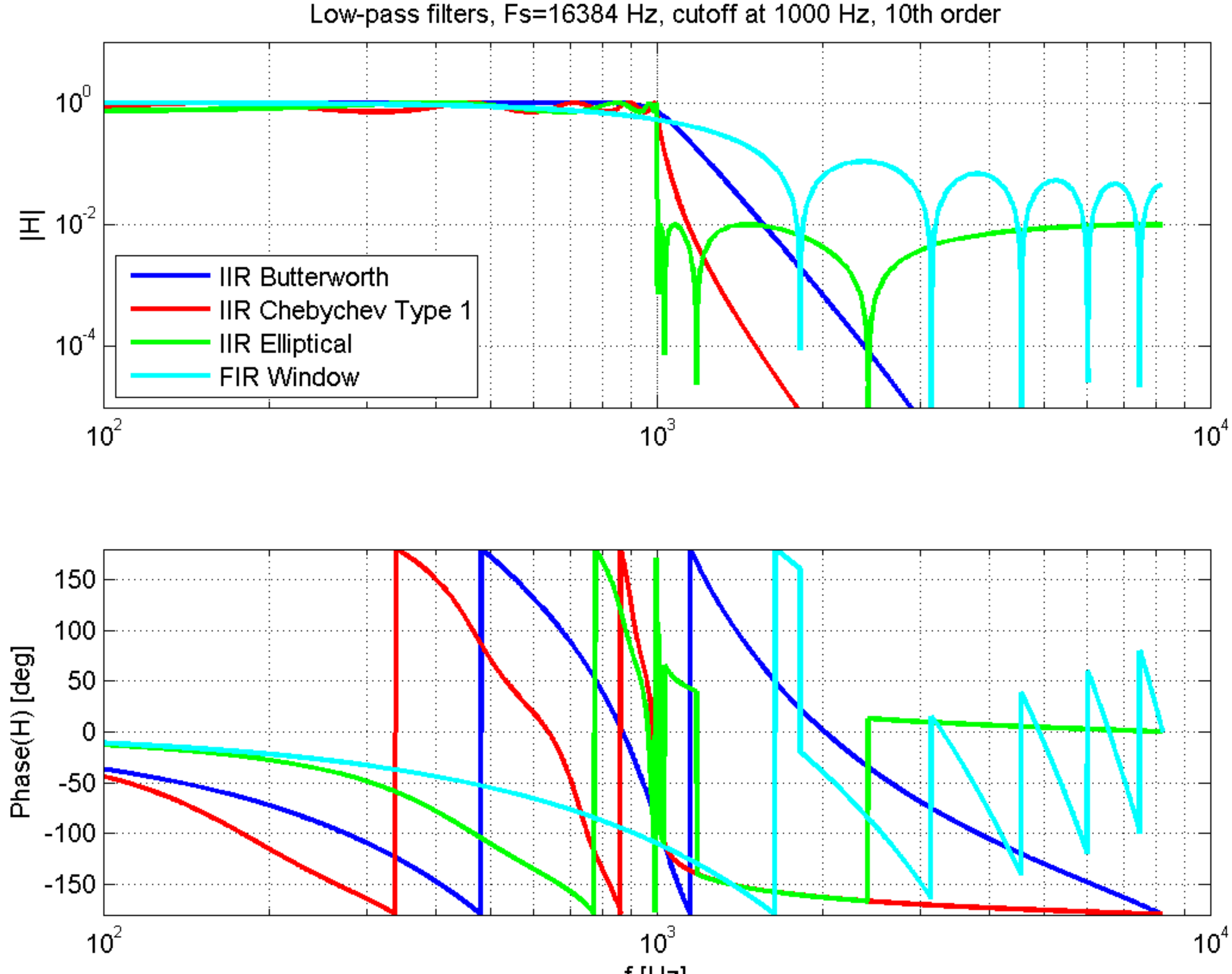
- Butterworth
  - Designed to have as flat a frequency response as possible in the passband
- Chebyshev Type 1
  - Steeper roll-off but more pass band ripple
- Chebyshev Type 2
  - Steeper roll-off but more stop band ripple
- Elliptic
  - Fastest transition

Comparison

Sampling frequency set to 16384 Hz

Filter order set to 10, cutoff set at 1 kHz

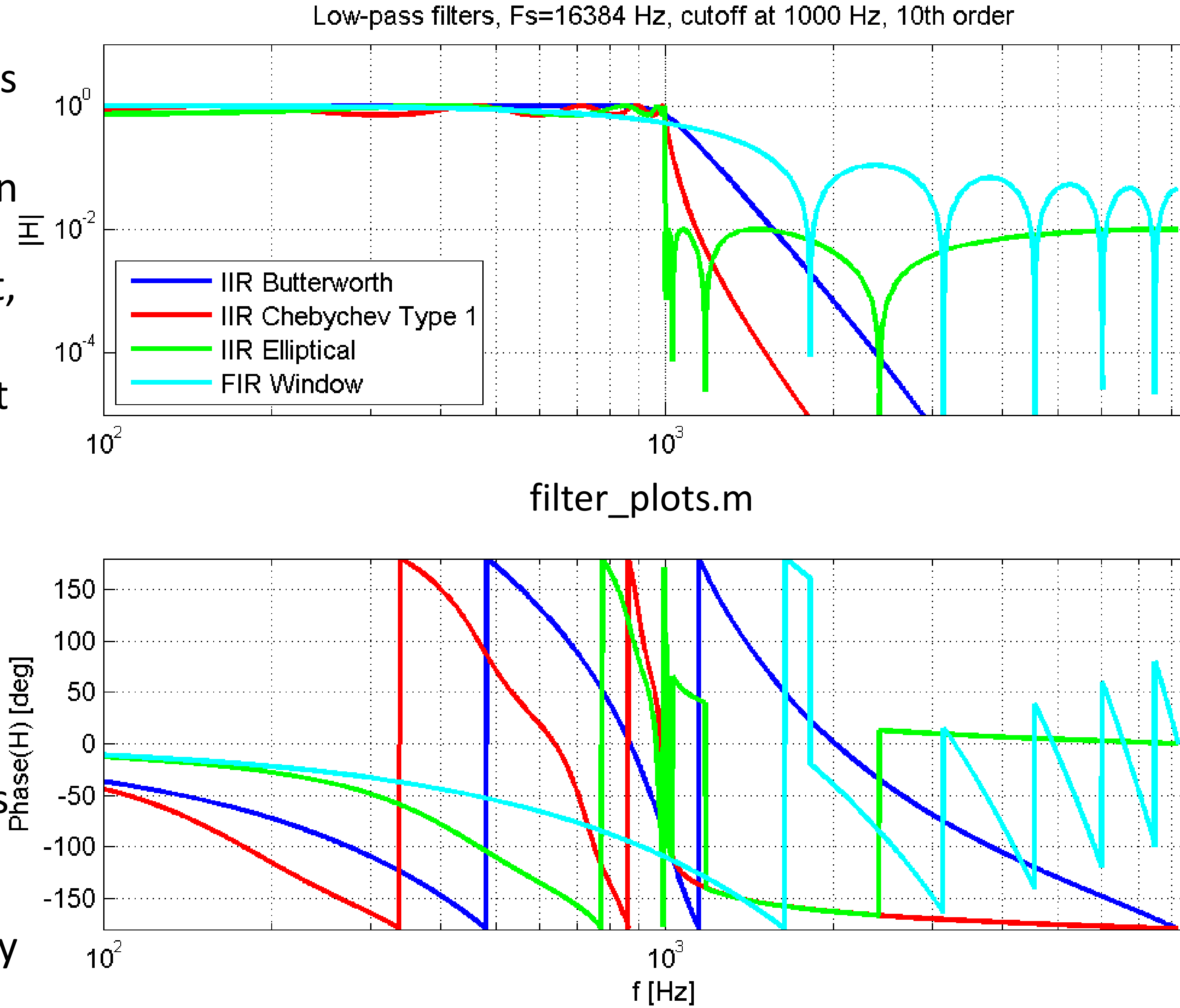
Difficult comparison: specifications for each filter can be very different



filter_plots.m

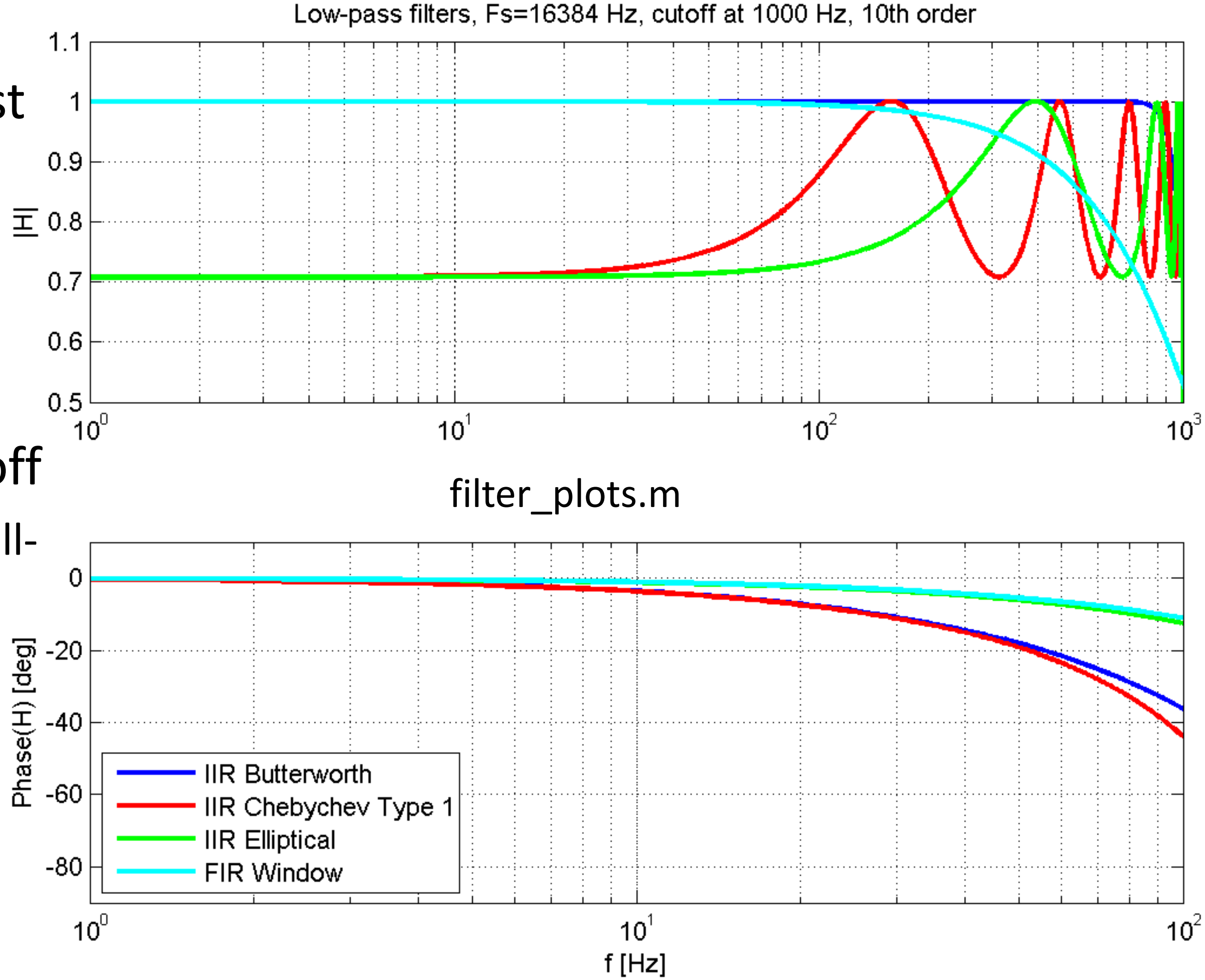
Comments

- Chebyshev filter has a steeper roll-off with respect to the Butterworth filter
- The elliptical filter has the fastest roll-off
- Elliptical's attenuation factor at high frequency is constant, unlike the others.
- Elliptical has the least phase delay with respect to the others
- Notice: the performance of a FIR window filter of 10th order is also shown. For it to achieve the same performance as the others, the filter order must be increased significantly



Comments

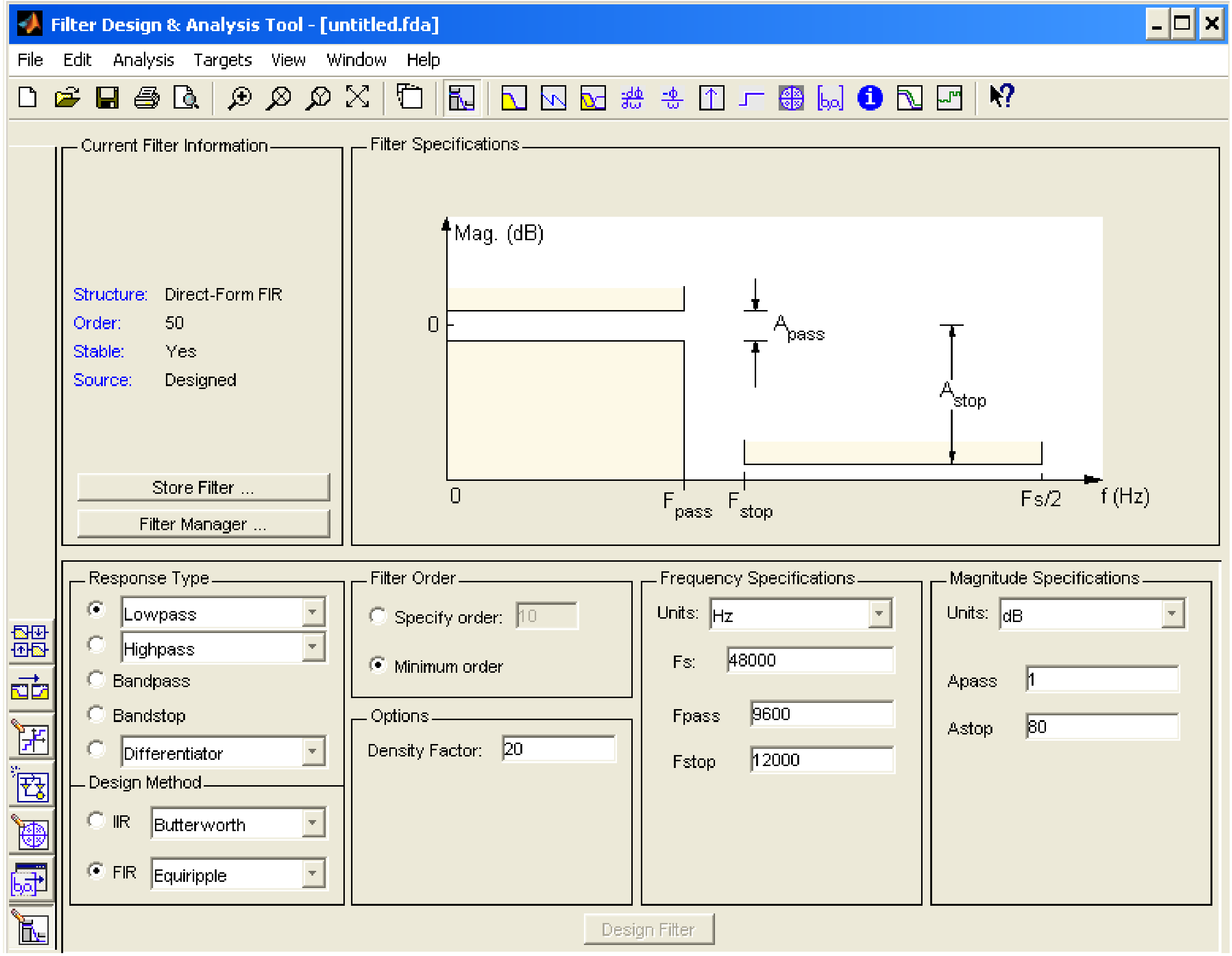
- The Butterworth filter has a flattest response when compared to the others.
- There is a trade off
  - The faster the roll-offs, the greater the ripples



filter_plots.m

MATLAB's fdatool

- Filter Design and Analysis Tool
- Allows you to design (visually) a digital filter
- Can export the filter into different formats
  - Filter coefficients
  - MATLAB's transfer function object
  - …



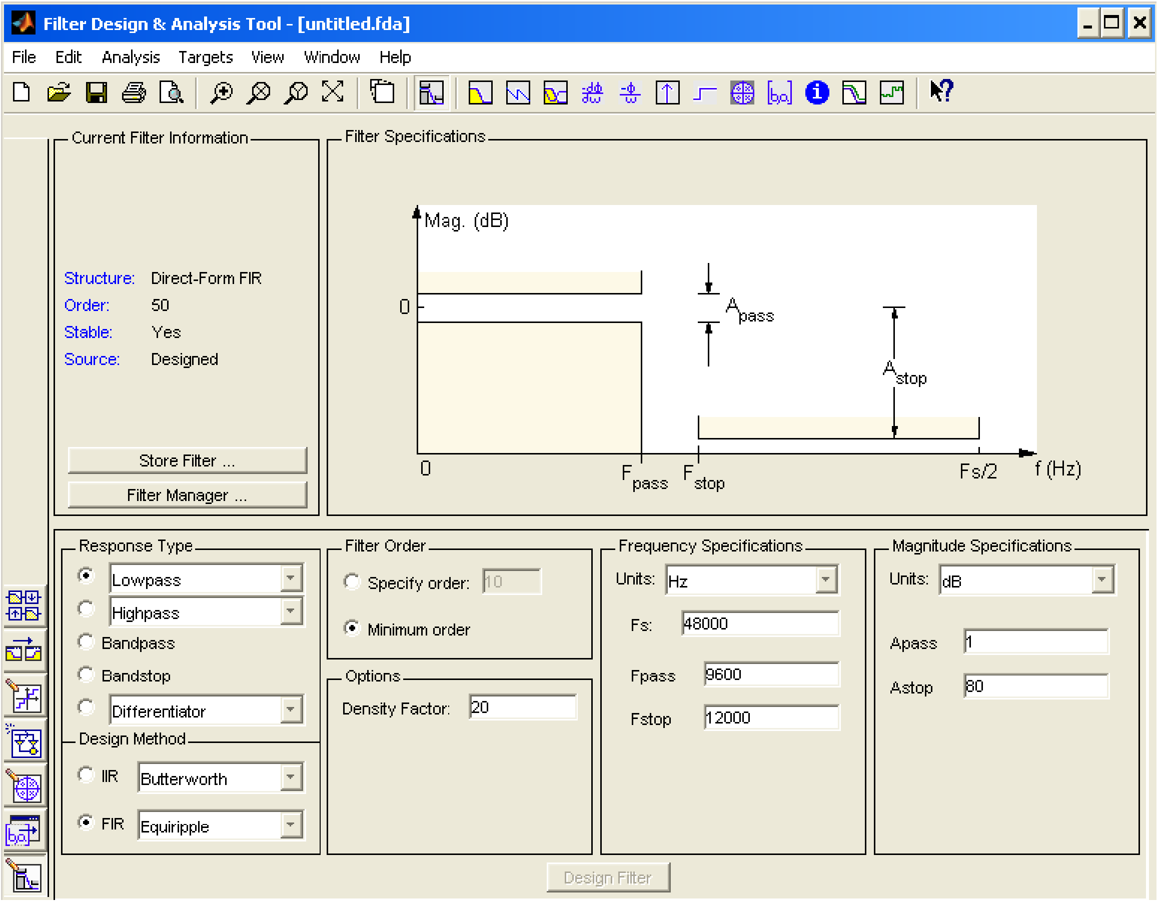

```
>> fdatool
```

MATLAB's fdatool

Exporting

- Coefficients `a, b`
- Transfer function object `Hd`
- Second-order-sections `sos`

The system function H(z) can be factored into second-order-sections. The system is then represented as a product of these sections.

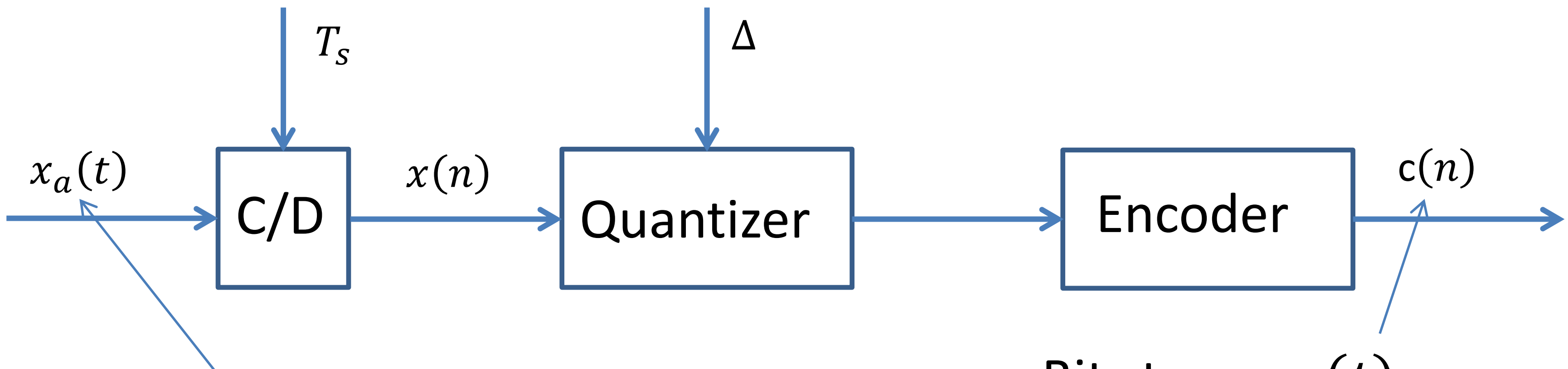


Assuming input signal `x`, the output y:

```
y = filter(Hd,x)
y = filter(b,a,x)
y = sosfilt(sos,x)
```

Sampling: Analog-to-Digital conversion

- Transforms analog signal to digital sequence
- Main components of an A/D converter



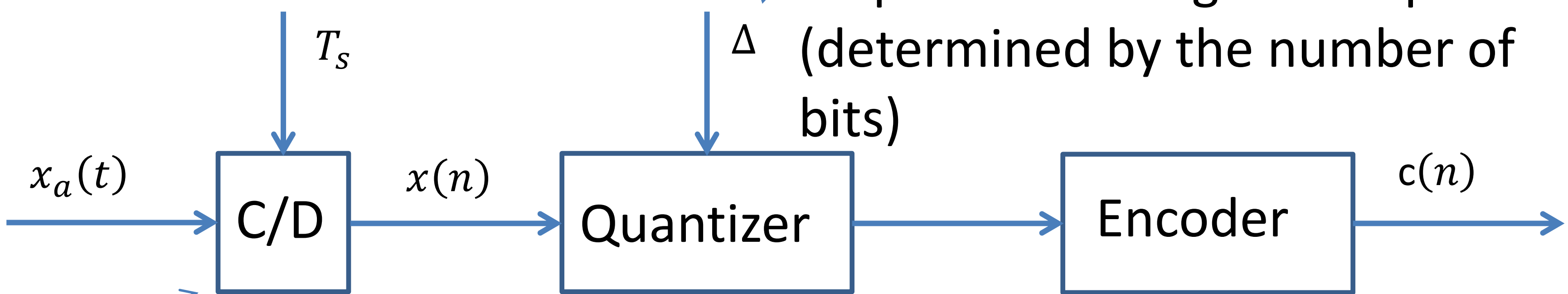
Analog signal $x_a(t)$: real valued function of a continuous variable t

Bit stream $c(t)$: corresponds to discrete time sequence $x(n)$

Sampling: Analog-to-Digital conversion

- Transforms analog signal to digital sequence
- Main components of an A/D converter

<u>Quantization interval Δ</u>: amplitude is integer multiple of Δ (determined by the number of bits)



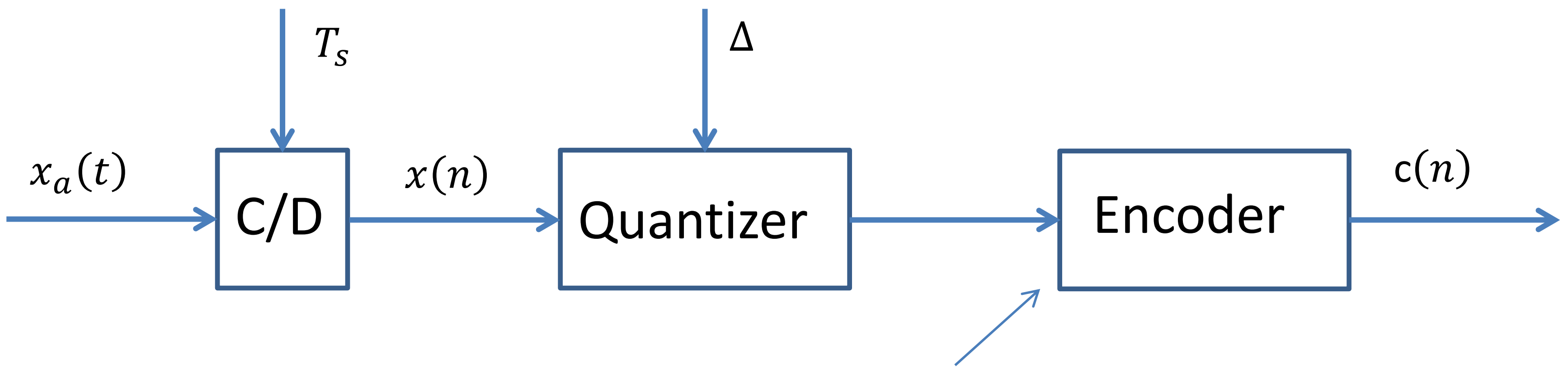
<u>Sampler C/D</u>: continuous-to-discrete converter or an ideal A/D converter:

$$x(n) = x_a(nT_s)$$

<u>Quantizer</u>: maps continuous range of possible amplitudes into a discrete set of amplitudes

Sampling: Analog-to-Digital conversion

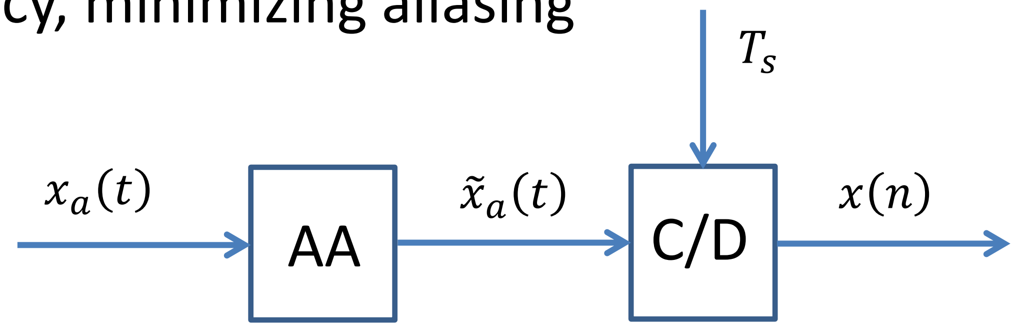
- Transforms analog signal to digital sequence
- Main components of an A/D converter



Encoder: produces a sequence of binary codewords

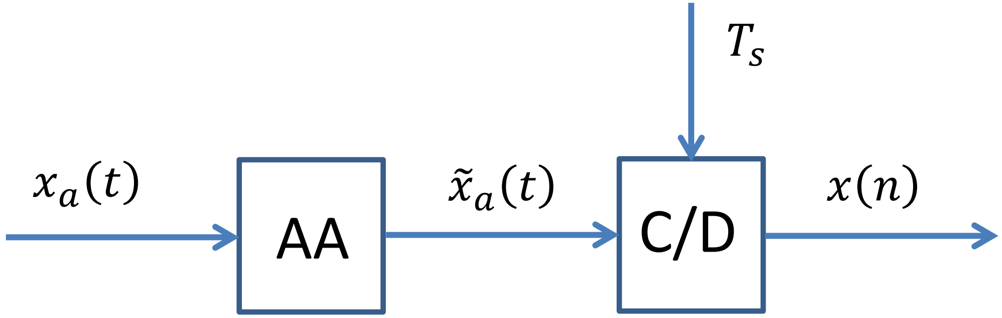
Sampling: Analog-to-Digital conversion

- Anti-aliasing filter
  - Signals in physical systems will never be exactly bandlimited, aliasing can occur
  - (Analog) lowpass at the Nyquist frequency.
    - This minimizes signal energy above the Nyquist frequency, minimizing aliasing



Sampling: Analog-to-Digital conversion

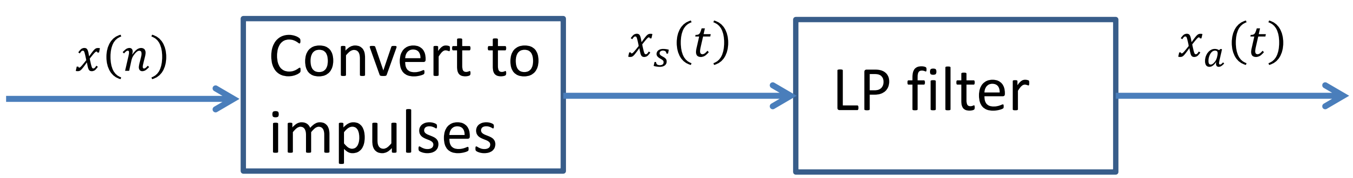
- Anti-aliasing filter
  - Signals in physical systems will never be exactly bandlimited, aliasing can occur
  - Analog lowpass filter that minimizes signal energy above the Nyquist frequency



And back: Digital-to-Analog conversion

Two steps involved

- Conversion to rectangular pulses
- Pulses cause multiple harmonics above the Nyquist frequency
- This excess noise is reduced with an (analog) low pass filter (or reconstruction filter)



THANK YOU!!