Time series analysis
Matlab tutorial

Joachim Gross

Outline

- Terminology
- Sampling theorem
- Plotting
- Baseline correction
- Detrending
- Smoothing
- Filtering
- Decimation

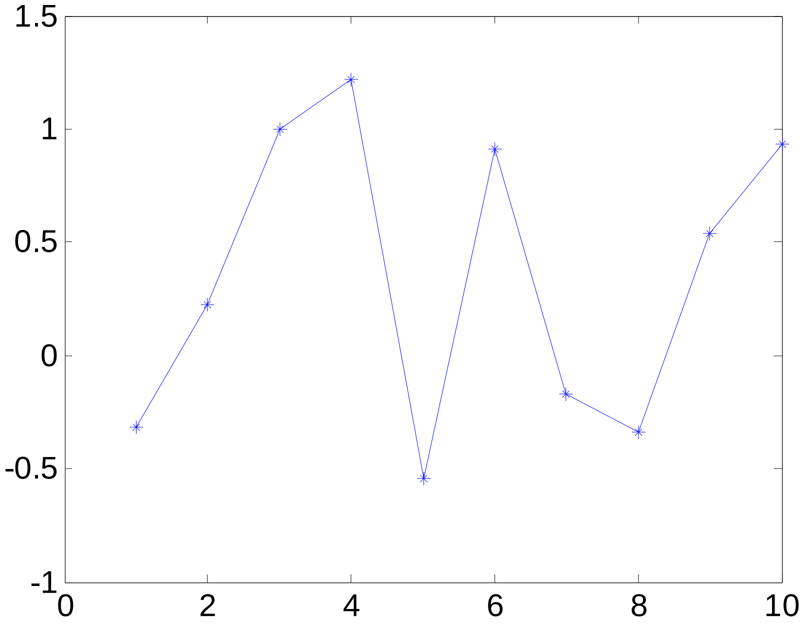
Remarks

- Focus on practical aspects, exercises, getting experience (not on equations, theory)
- Focus on “How to do …”
- Learn some basic skills for TS analysis

- Note: Usually there is not a single perfectly correct way of doing a TS operation!
=> learn the limitations!

What is a time series?

A sequence of measurements over time



Terminology

- Continuous TS: continuous observations
- Discrete TS: observations at specific times usually equally spaced

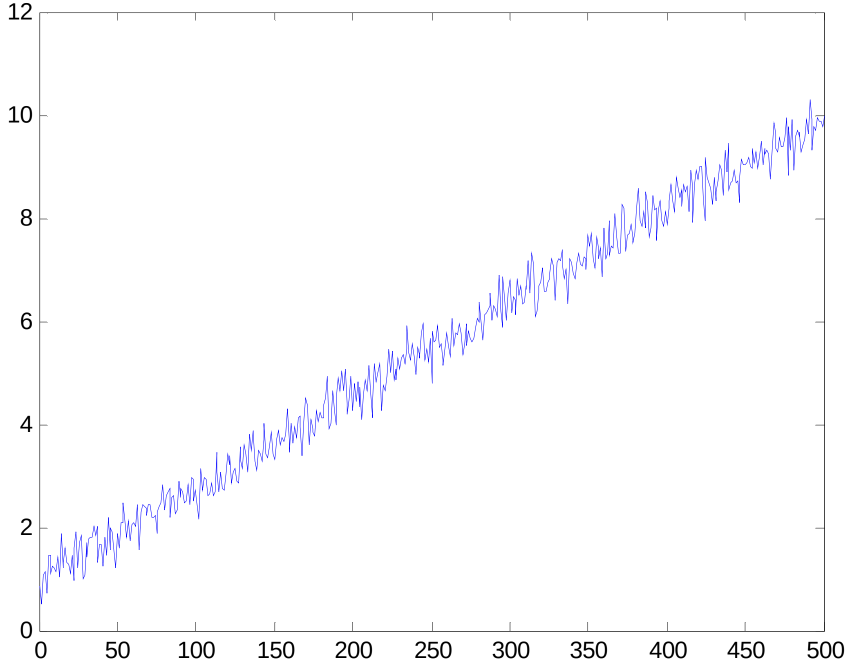
- Deterministic TS: future values can be exactly predicted from past values
- Stochastic TS: exact prediction not possible

Objectives of TS analysis

- Description
- Explanation
- Prediction
- Control

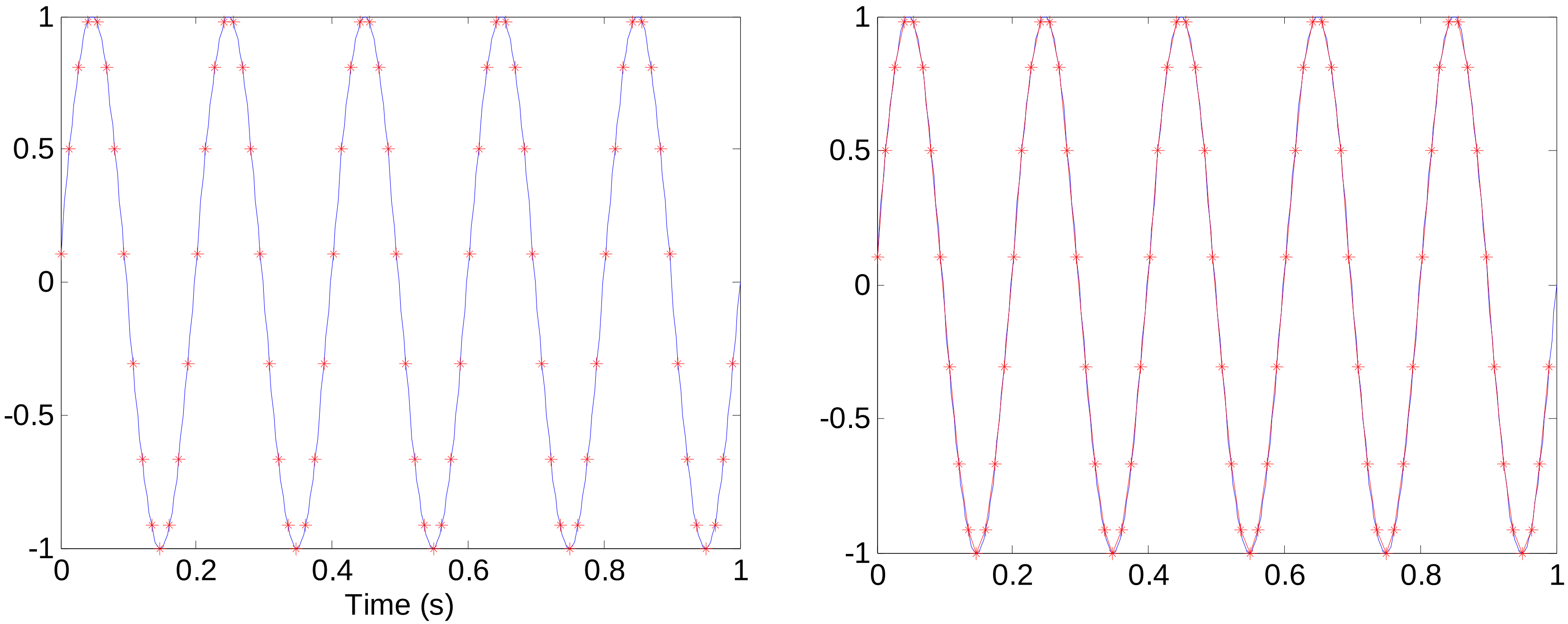
Simple descriptive analysis

Summary statistics (mean, std) is not always meaningful for TS



Sampling

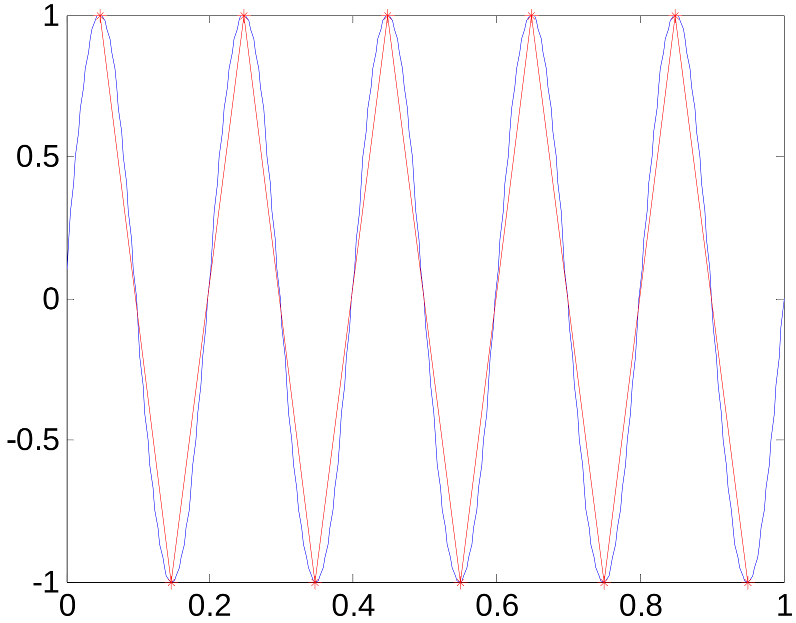
- Converting a continuous signal into a discrete time series
- **Reconstruction is possible if sampling frequency is greater than twice the signal bandwidth**



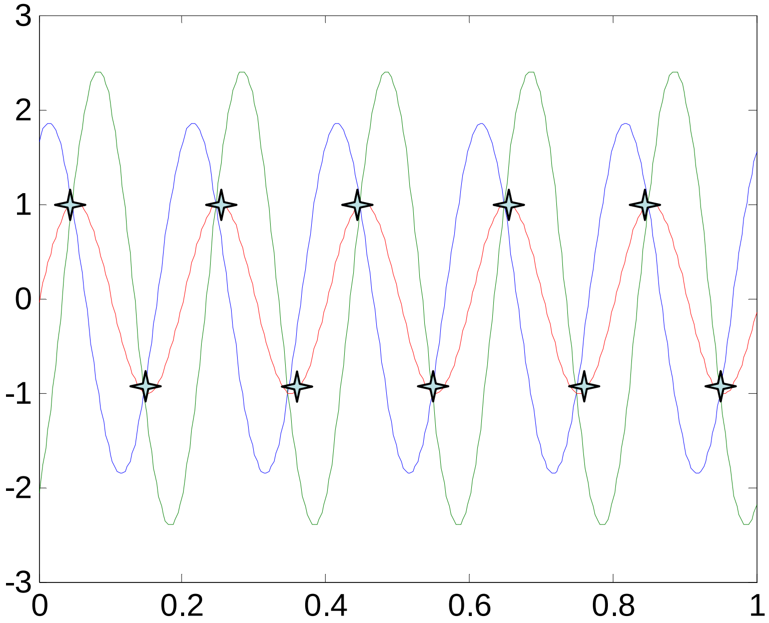
75 Hz sampling

Sampling

- Nyquist frequency: half of sampling frequency



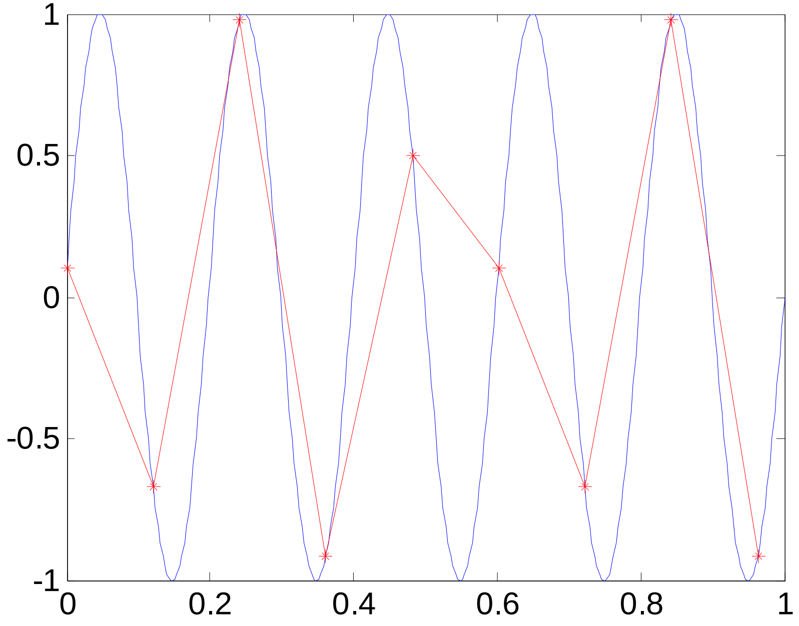
10 Hz sampling



10 Hz reconstruction

Sampling

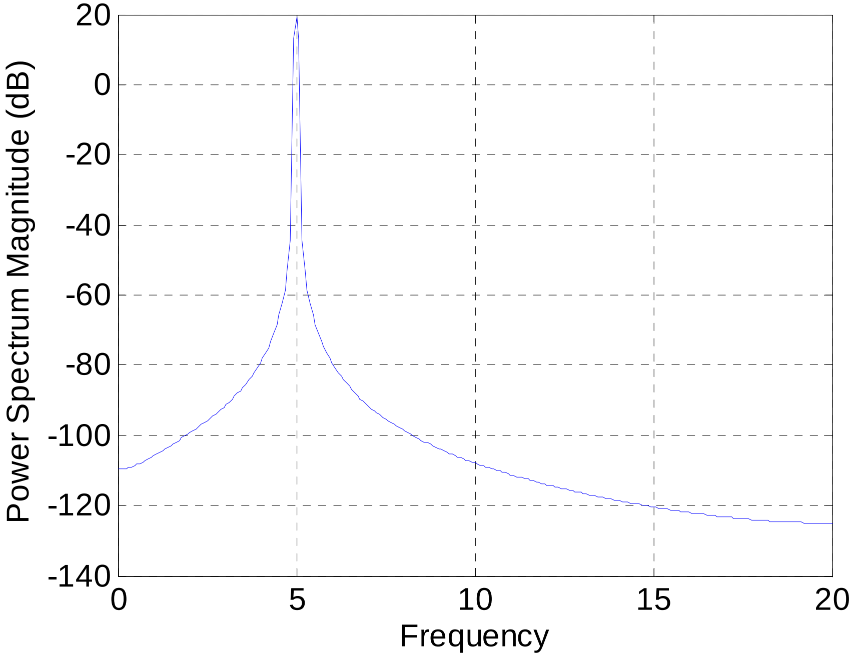
- Aliasing: Frequencies above Nyquist frequency are reconstructed below Nyquist frequency



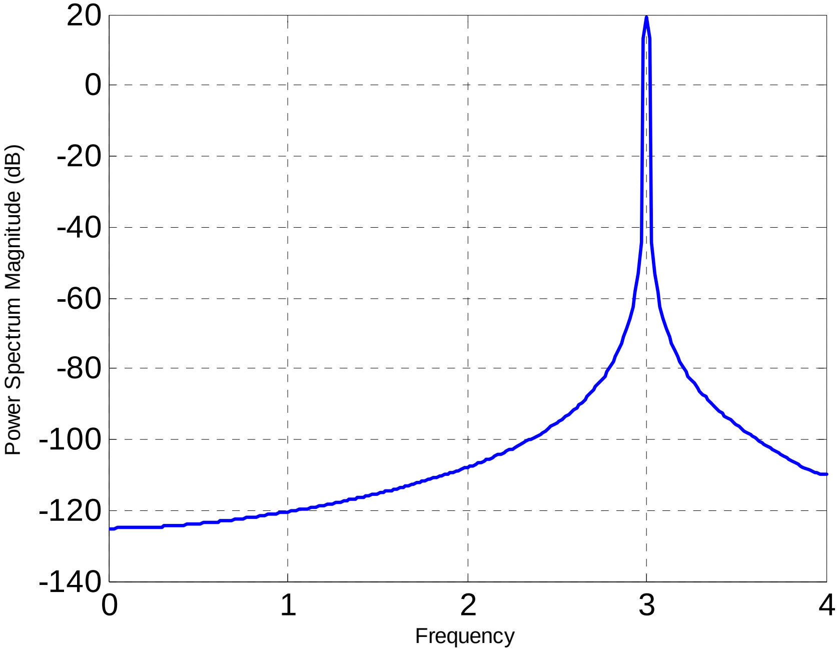
8 Hz sampling

Sampling

- Aliasing: Frequencies above Nyquist frequency are reconstructed below Nyquist frequency



40 Hz sampling



8 Hz sampling

Simple operations on TS

- Plotting
- Removing a baseline
- Removing a trend
- Smoothing
- Filtering
- Decimation

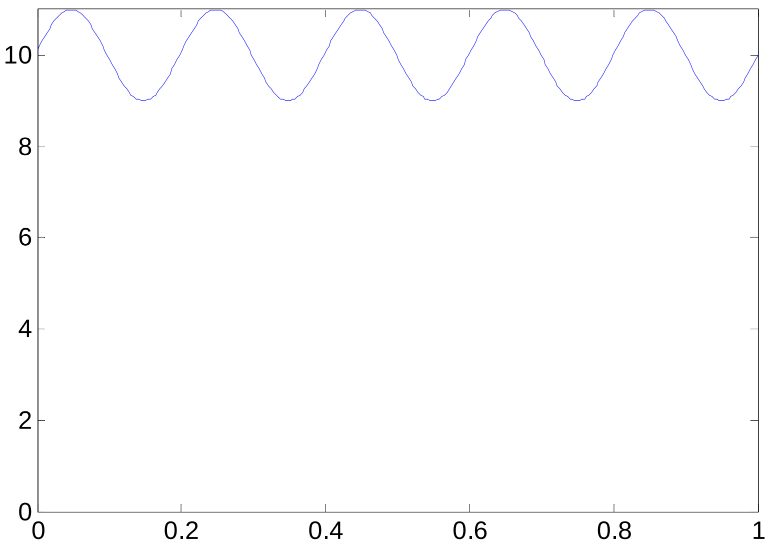
Plotting in Matlab

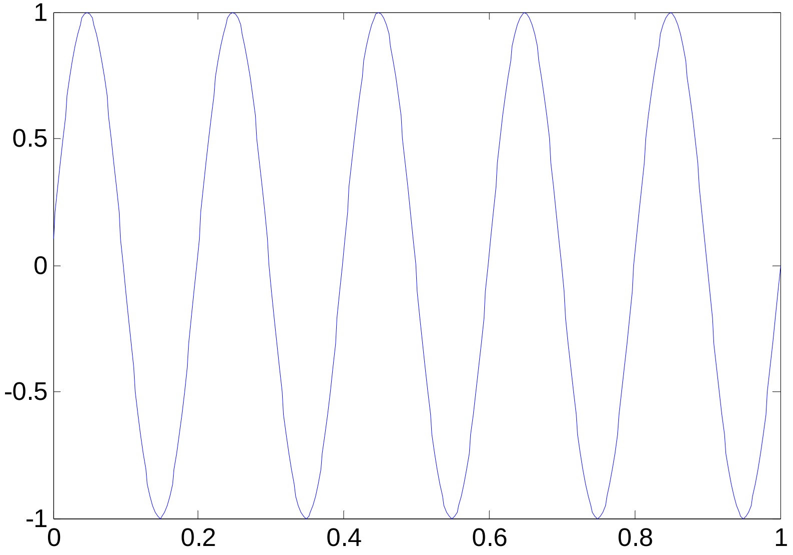
- For visual inspection of TS
- For publications/talks

- plot
- sptool

Data preprocessing I

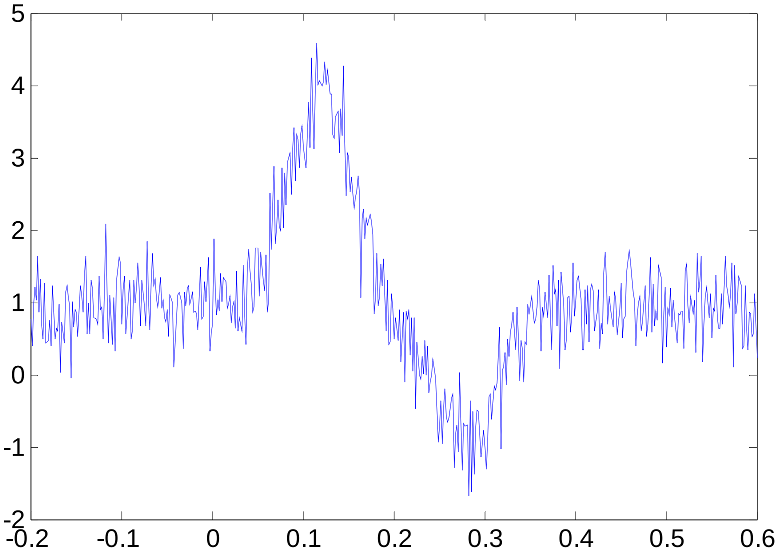
- Removing offset
- ts=ts-mean(ts);

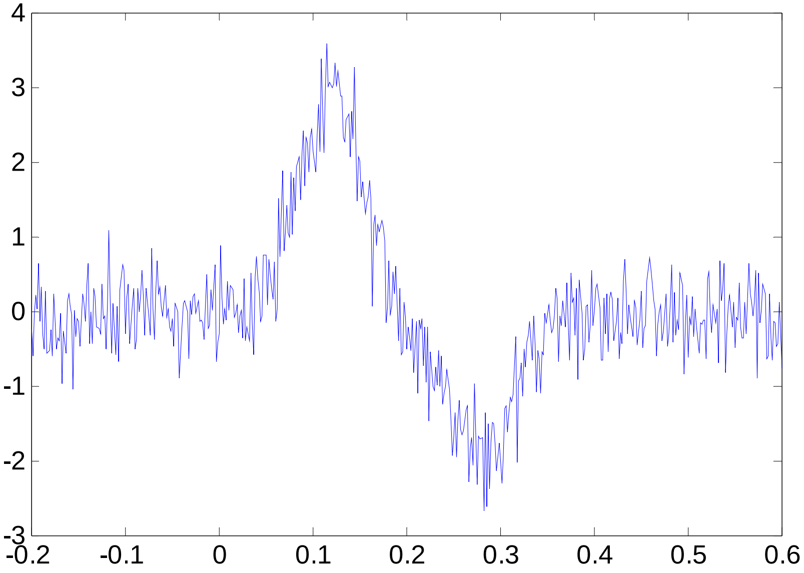




Data preprocessing I

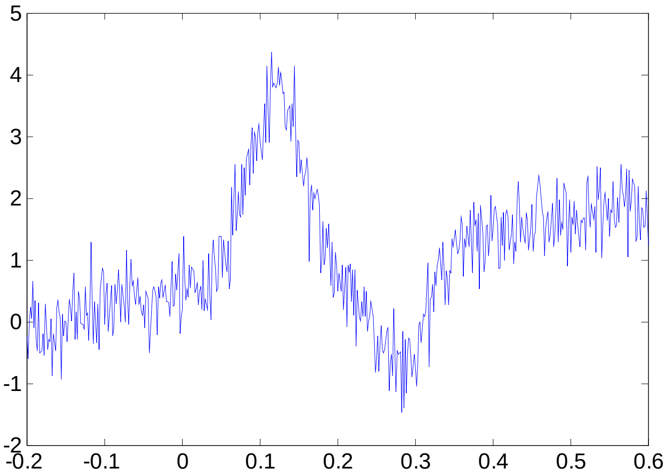
- Removing a baseline
- basel=find(t<=0);
- ts=ts-mean(ts(basel));

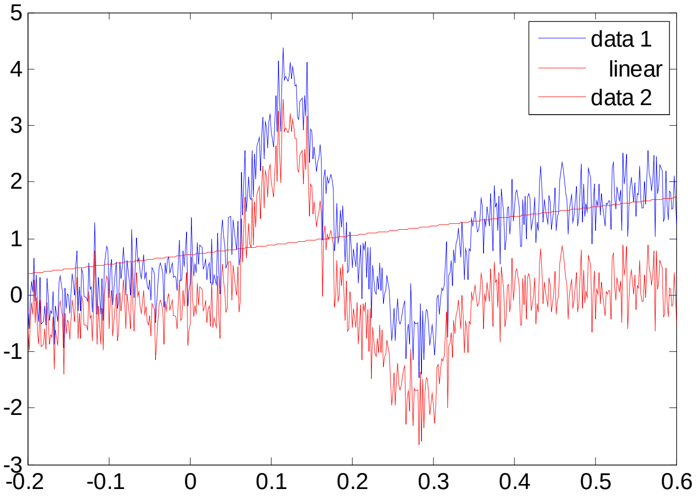




Data preprocessing II

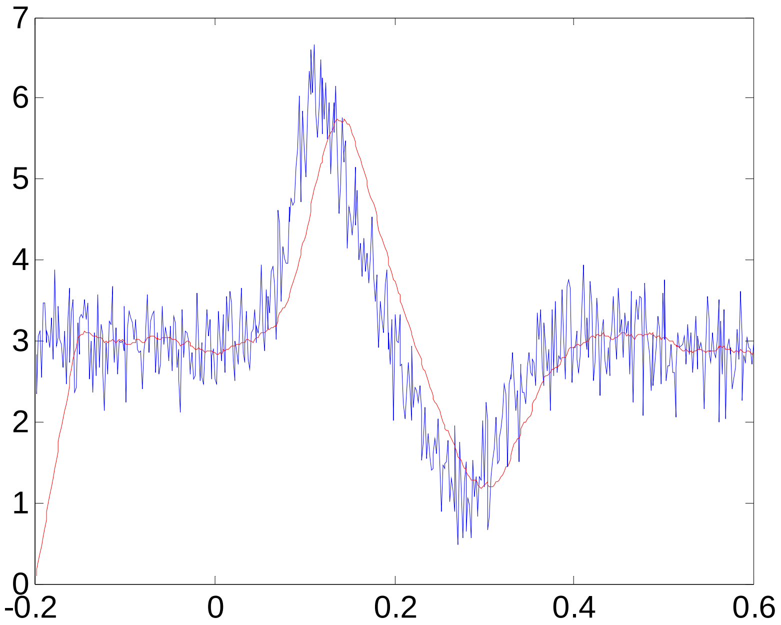
- Removing a trend
- ts=detrend(ts);
- subtracts best fitting line
- detrend can be used to subtract mean: detrend(ts,’constant’)





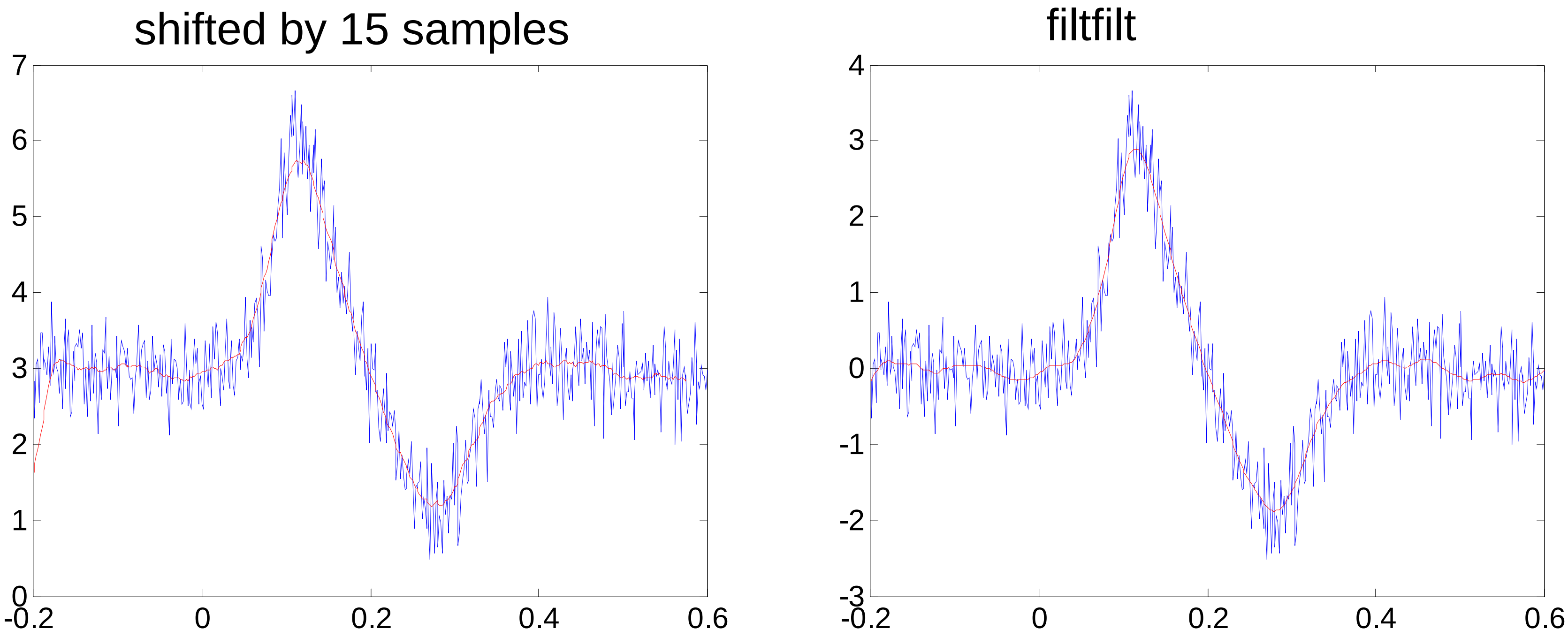
Data preprocessing III

- Smoothing
- ts=filter(ones(1,30)/30,1,ts); %mean filter, moving average
- uses zeros at beginning!
- => baseline correction or do not use first 30 samples



Data preprocessing III

- introduces a shift! => either correct for it or
- ts=filtfilt(ones(1,15)/15,1,ts); %mean filter, forward and reverse
- no shift!
- filter can take any smoothing kernel (gaussian, etc)

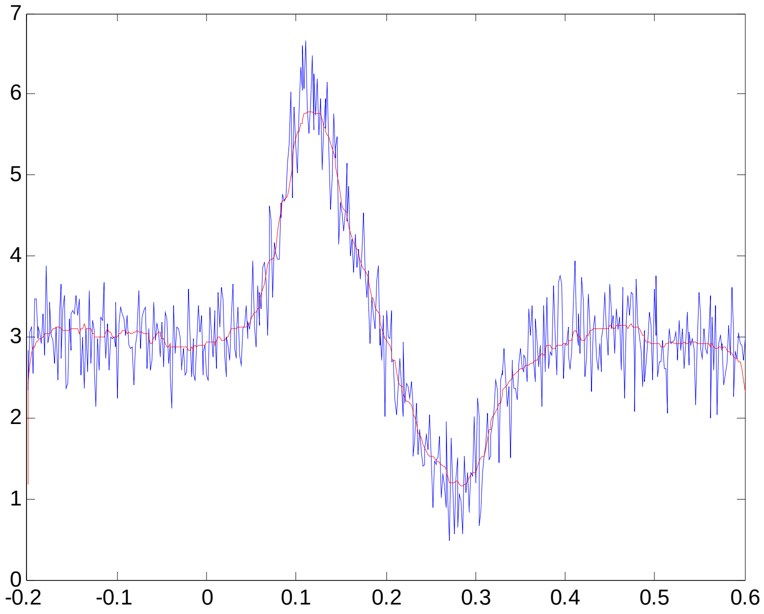


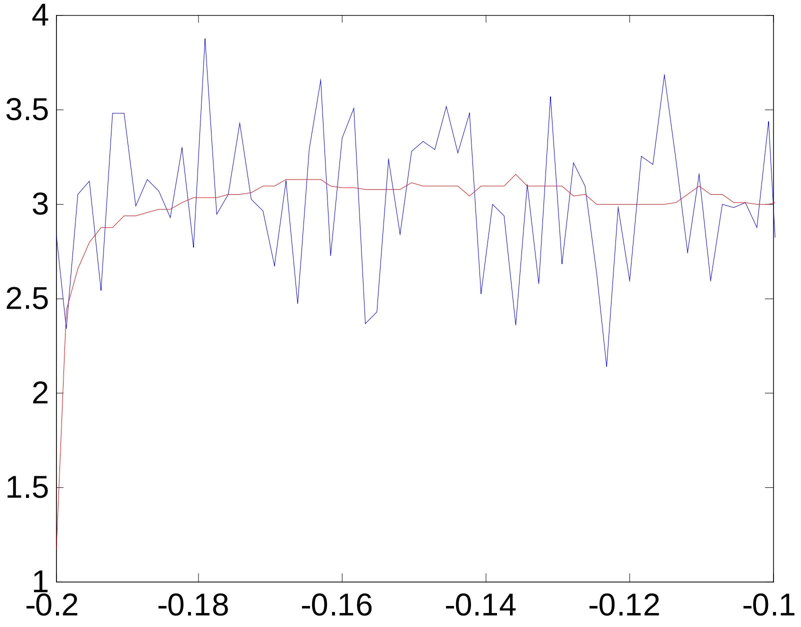
shifted by 15 samples

filtfilt

Data preprocessing III

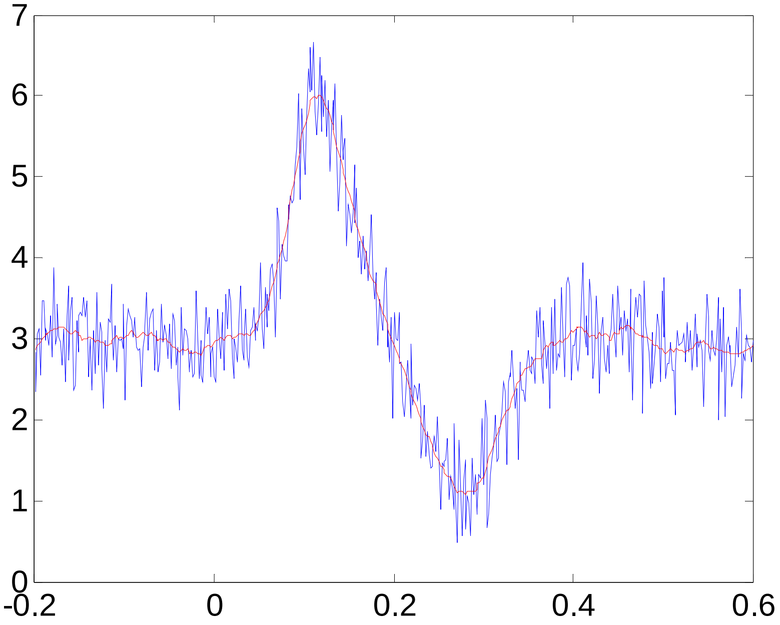
- Smoothing
- ts=medfilt1(ts,30); %median filter, takes into account the shift
- uses 0 at beginning and end !





Data preprocessing III

- Smoothing
- ts=sgolayfilt(ts,3,41); %Savitzky-Golay filter
- fits 3rd order polynomial to frames of size 41
- good at preserving high frequencies in the data



Data preprocessing III

- Smoothing
- compare unsmoothed and smoothed data
- check for shift
- check beginning (and end) of the smoothed time series

Exercise 1

Data preprocessing IV

- Filtering
- **FIR-Filter** (finite impulse response)
- stable
- high filter order
- usually have linear phase

(phase change is proportional to frequency)

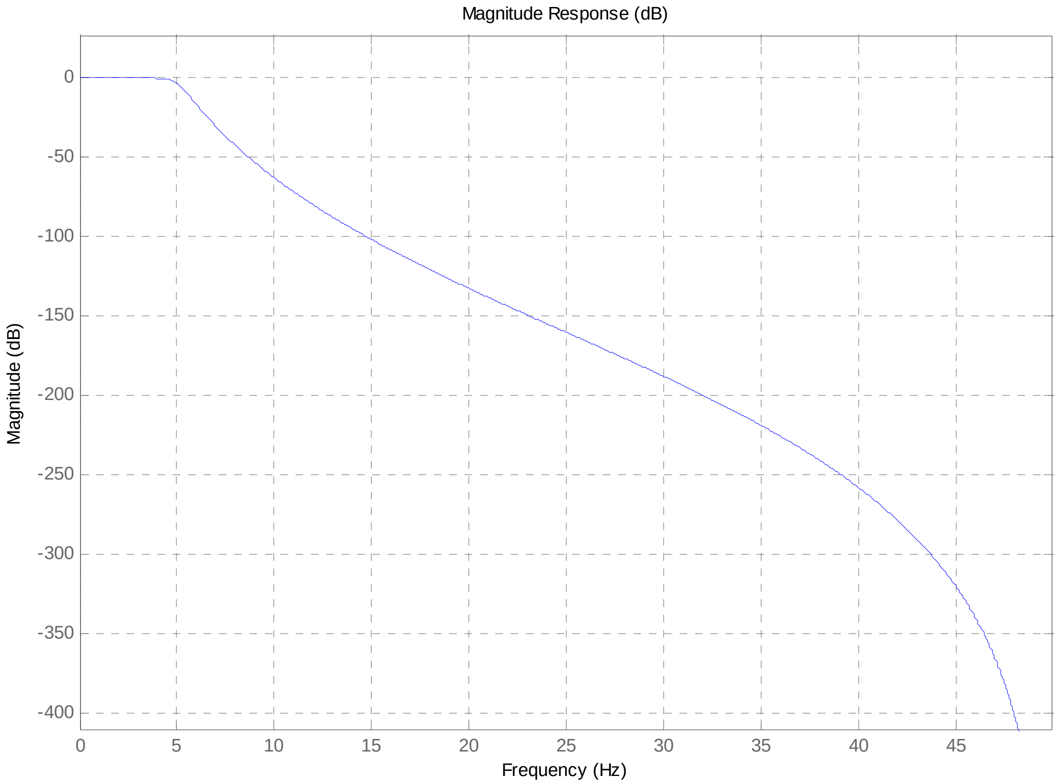
- **IIR-Filter** (infinite impulse response)
- potentially unstable
- low filter order
- non-linear phase distortion
- computationally efficient

Data preprocessing IV

- IIR-Filter:
  - **Butterworth**
  - Elliptic
  - Chebychev Typ 1
  - Chebychev Typ 2
  - Bessel
- FIR-Filter:
  - fir1

Data preprocessing IV

- **lowpass**
- highpass
- bandpass
- bandstop

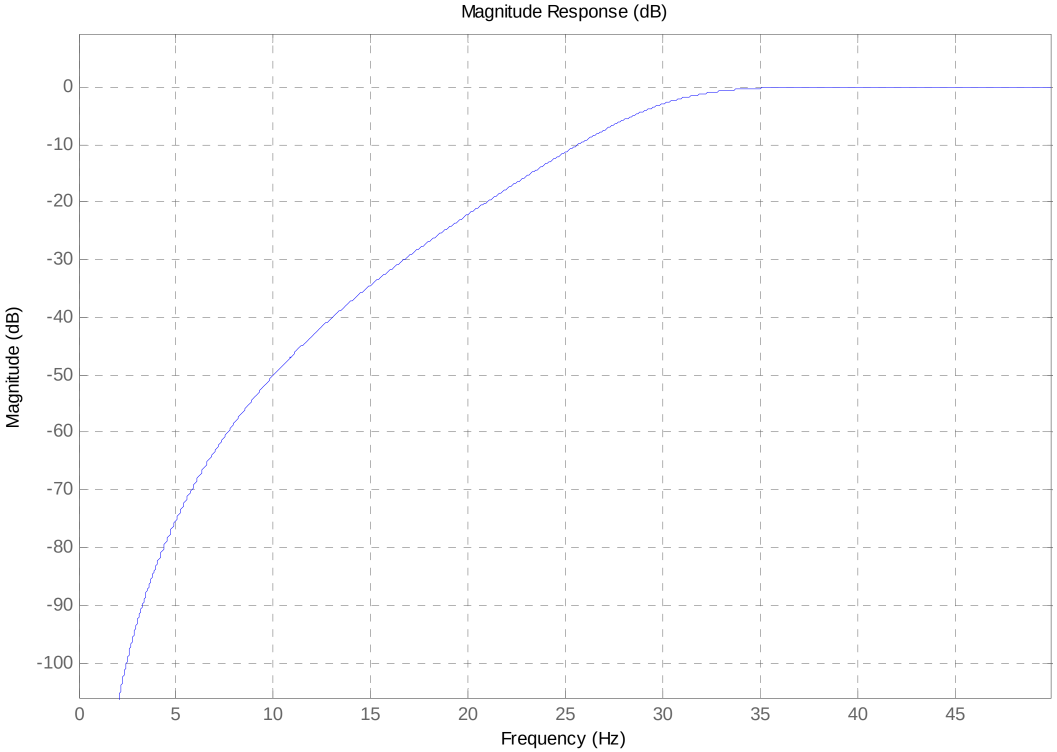


dB is logarithmic unit
0dB = factor of 1
3dB = factor of 2
10dB= factor of 10

5 Hz lowpass

Data preprocessing IV

- lowpass
- **highpass**
- bandpass
- bandstop

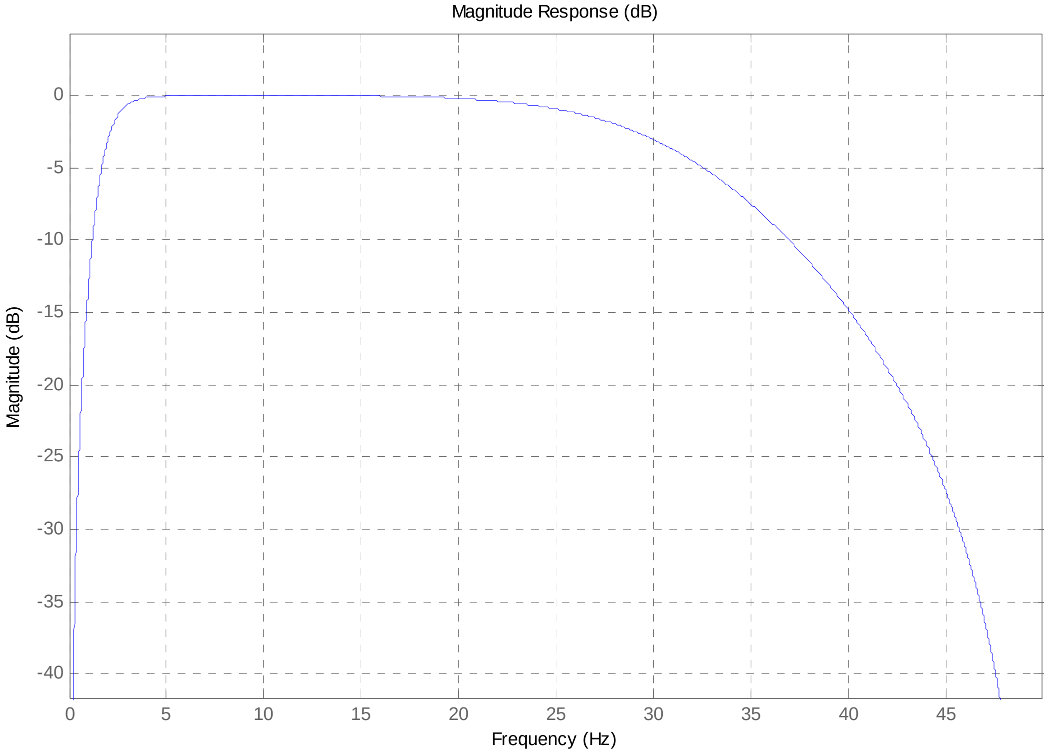


dB is logarithmic unit
0dB = factor of 1
3dB = factor of 2
10dB= factor of 10

30 Hz highpass

Data preprocessing IV

- lowpass
- highpass
- **bandpass**
- bandstop

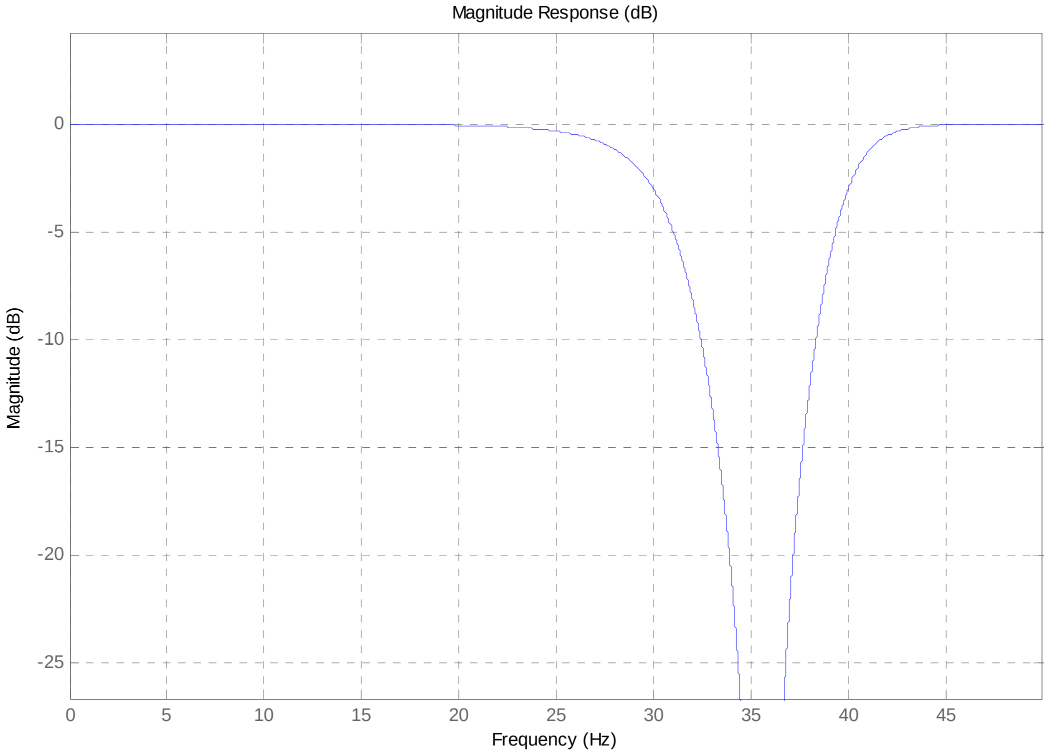


dB is logarithmic unit
0dB = factor of 1
3dB = factor of 2
10dB= factor of 10

2-30 Hz bandpass

Data preprocessing IV

- lowpass
- highpass
- bandpass
- **bandstop**



dB is logarithmic unit
0dB = factor of 1
3dB = factor of 2
10dB= factor of 10

30-40 Hz bandstop

Simple design: FIR

- [b]=fir1(4,2*4/sf); %4 Hz lowpass
- [b]=fir1(4,2*4/sf,’high’); %4 Hz highpass
- [b]=fir1(4,2*[4 10]/sf); %4-10 Hz bandpass
- [b]=fir1(4,2*[4 10]/sf,’stop’); %4-10 Hz bandstop

- tsf=filter(b,1,ts);
- tsf=filtfilt(b,1,ts); %forward and reverse

Simple design: IIR

- [b,a]=butter(4,2*4/sf); %4 Hz lowpass
- [b,a]=butter(4,2*4/sf,’high’); %4 Hz highpass
- [b,a]=butter(4,2*[4 10]/sf); %4-10 Hz bandpass
- [b,a]=butter(4,2*[4 10]/sf,’stop’); %4-10 Hz bandstop

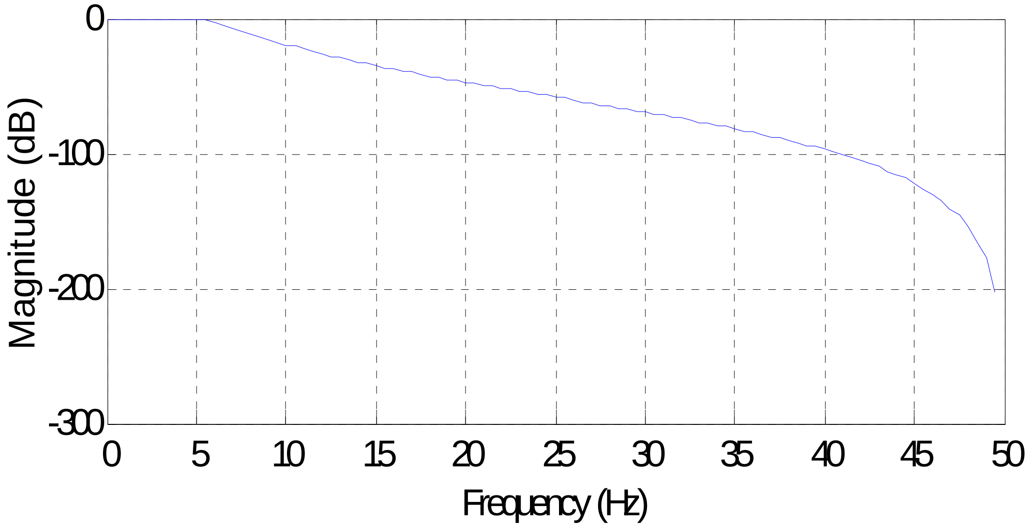
- tsf=filter(b,a,ts);
- tsf=filtfilt(b,a,ts); %forward and reverse

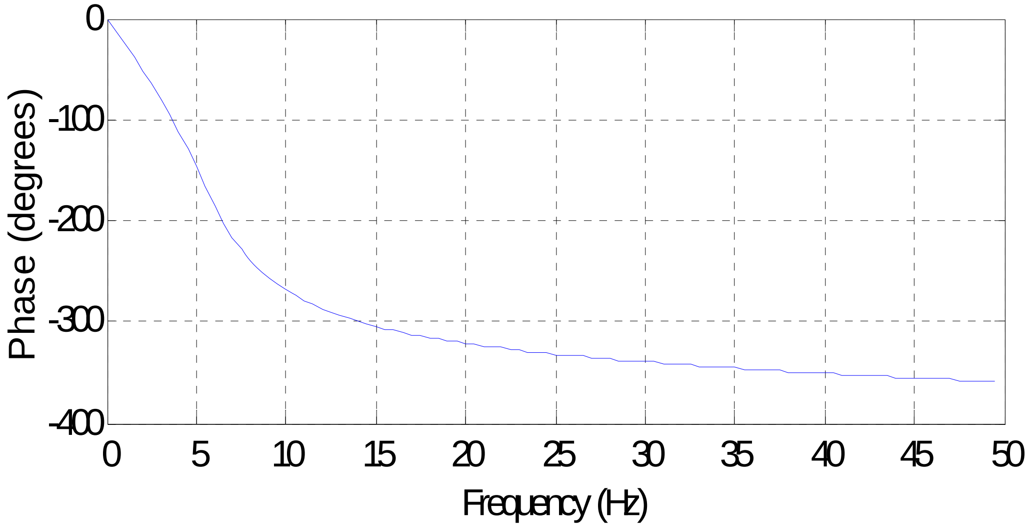
Simple Inspection

freqz(b,a,100,100);

sf

number of frequencies





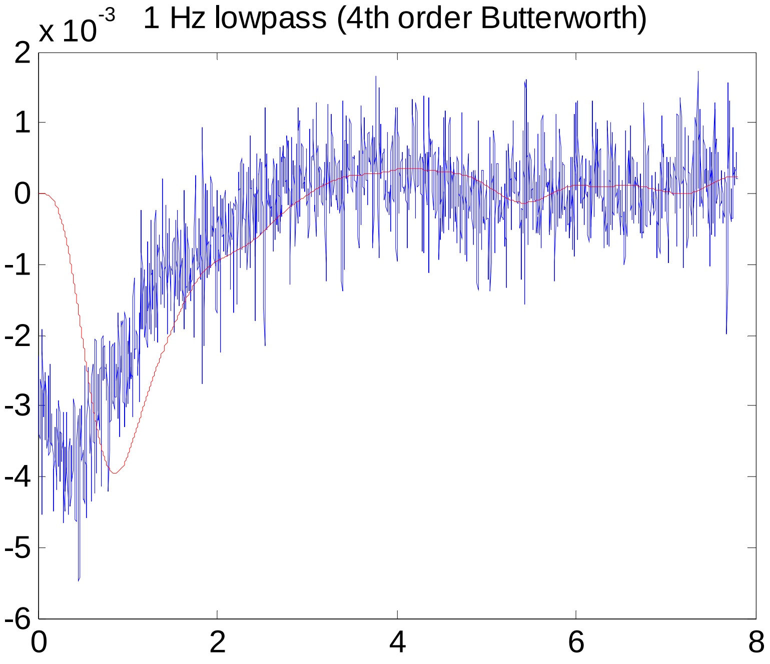
Complex design

- fdatool
  - magnitude response
  - phase response
  - impulse response
  - compare filters
  - effect of changing filter order

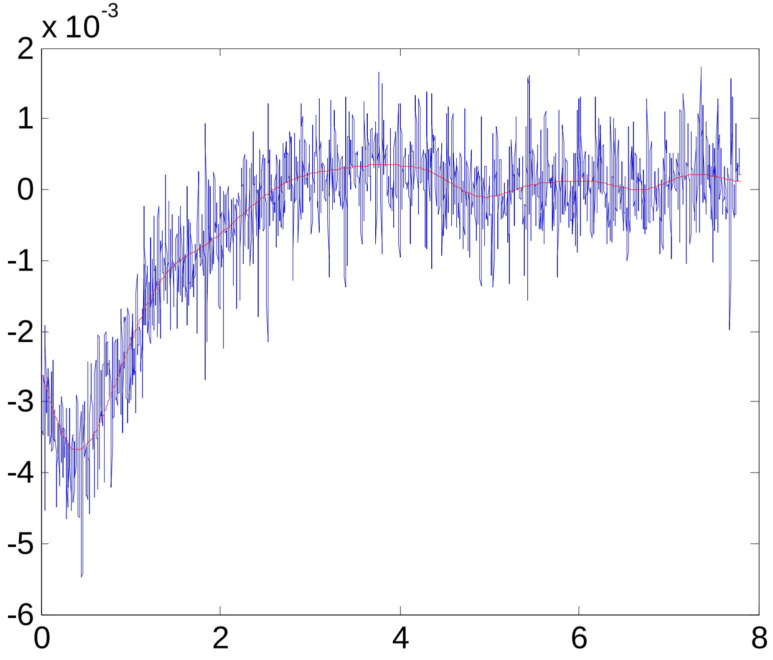
Filter artifacts

- onset transients

filter

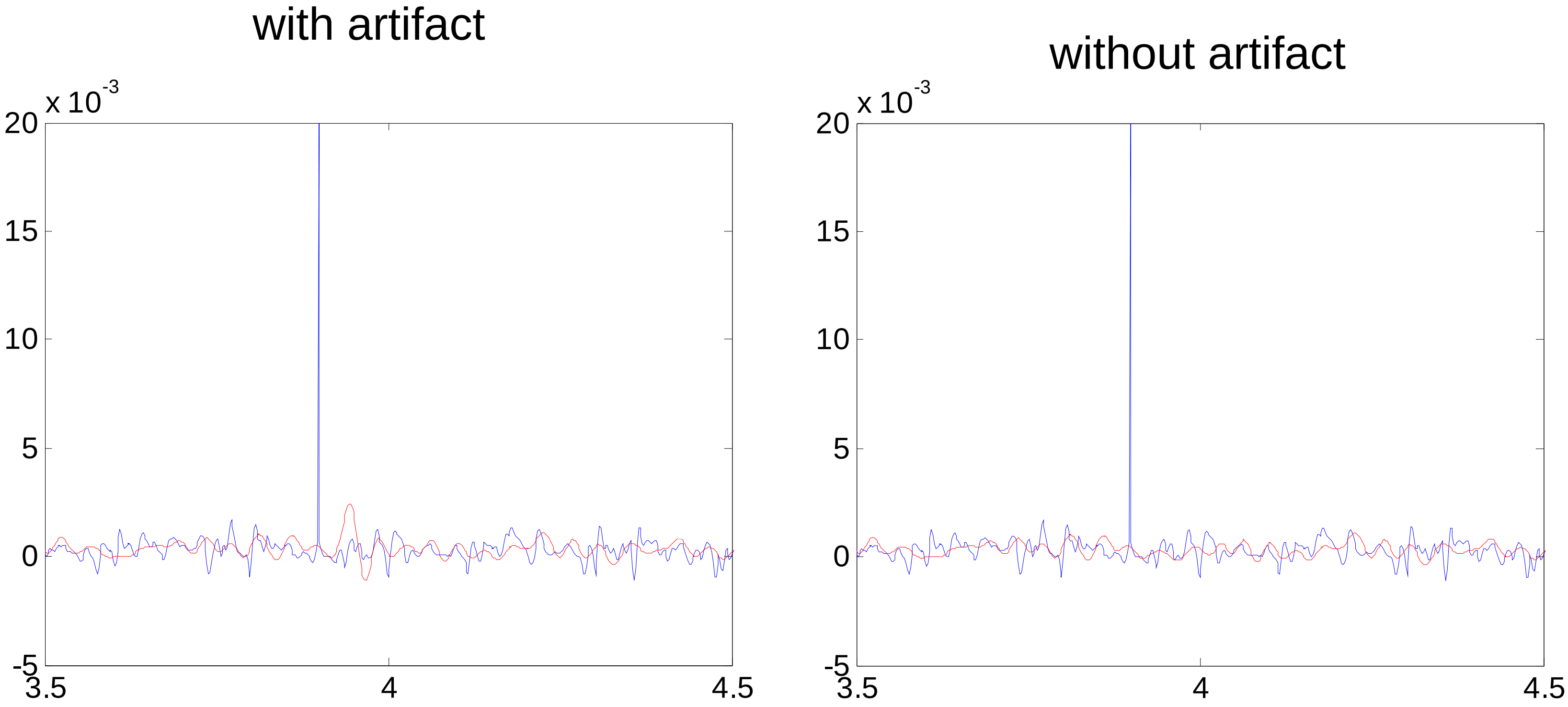


filtfilt



Filter artifacts

- ringing



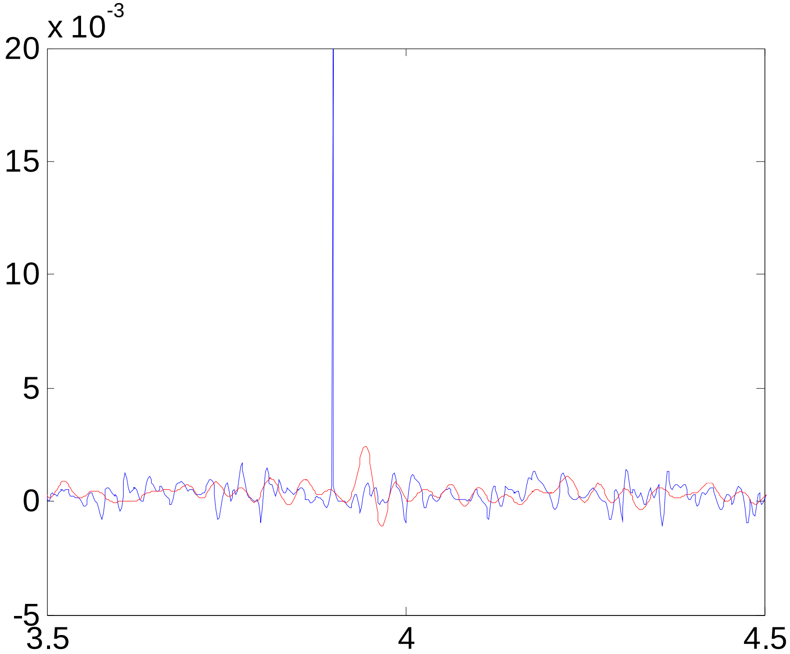
with artifact

without artifact

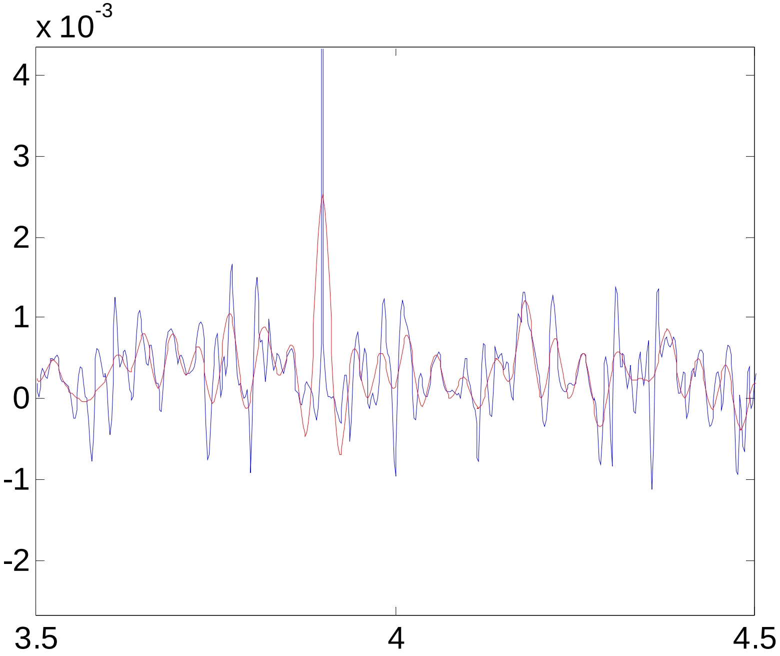
Filter artifacts

- ringing

filter, 20 Hz lowpass (12th order Butterworth)



filtfilt



Filter artifacts

- beginning and end of filtered ts is distorted
- filtering artifacts is dangerous
- filtering may change the latency of effects!
- filtering may change the phase

Suggestions

- be careful with low frequencies
- use low order butterworth forward and reverse (to avoid phase distortions)
- carefully check beginning and end of filtered ts
- make sure you don’t have artifacts in the data
- use surrogate data (filtered noise)

Data preprocessing V

- Decimation
- ts=decimate(ts,4);
- decimate uses a lowpass filter to avoid aliasing artifacts

Exercises 2-4