Acoustic Emission

Wavelet analysis
background

Wavelets

- Fourier transform: local information in time domain is spread to all frequencies, and vice versa.
- wavelet transformations : also a fast, linear operation, but wavelets are more localized.
- w(t)<->W(k); invertable and orthogonal, transpose of matrix is inverse of matrix.
- renders matrices sparse: information reduction.
- Daubechies wavelets, from highly localized to highly smooth.
- Simplest class has 4 coefficients, c0…c3.
- Transformation matrix given as:

Wavelet basics

- Wavelets: complete orthogonal set, like Fourier base
  - Localized in time and in the conjugate dimension
  - Example: Daubechy wavelets (See Numerical Recipes, CH. 13.10):

$$W[j]=A[i,j]x[i]$$

$$A=\begin{pmatrix} c_0 & c_1 & c_2 & c_3 & & & \\ c_3 & -c_2 & c_1 & -c_0 & & & \\ & & c_0 & c_1 & c_2 & c_3 & \ldots \\ & & c_3 & -c_2 & c_1 & -c_0 & \ldots \\ \ldots & & & & & & \\ c_2 & c_3 & & & & c_0 & c_1 \\ c_1 & -c_0 & & & & c_3 & -c_2 \end{pmatrix}$$

A is orthogonal: transpose is inverse.
This yields:

$$c_0^2+c_1^2+c_2^2+c_3^2=1$$
$$c_0c_2-c_1c_3=0$$

2 remaining conditions are used to make the first 2 moments disappear:

$$c_3-c_2+c_1-c_0=0$$
$$-c_2+2c_1-3c_0=0$$

DAUB4 : 4 coefficients. Continuous wavelets but cusps.
Higher order Daubechy sets: higher-order moments are cancelled.

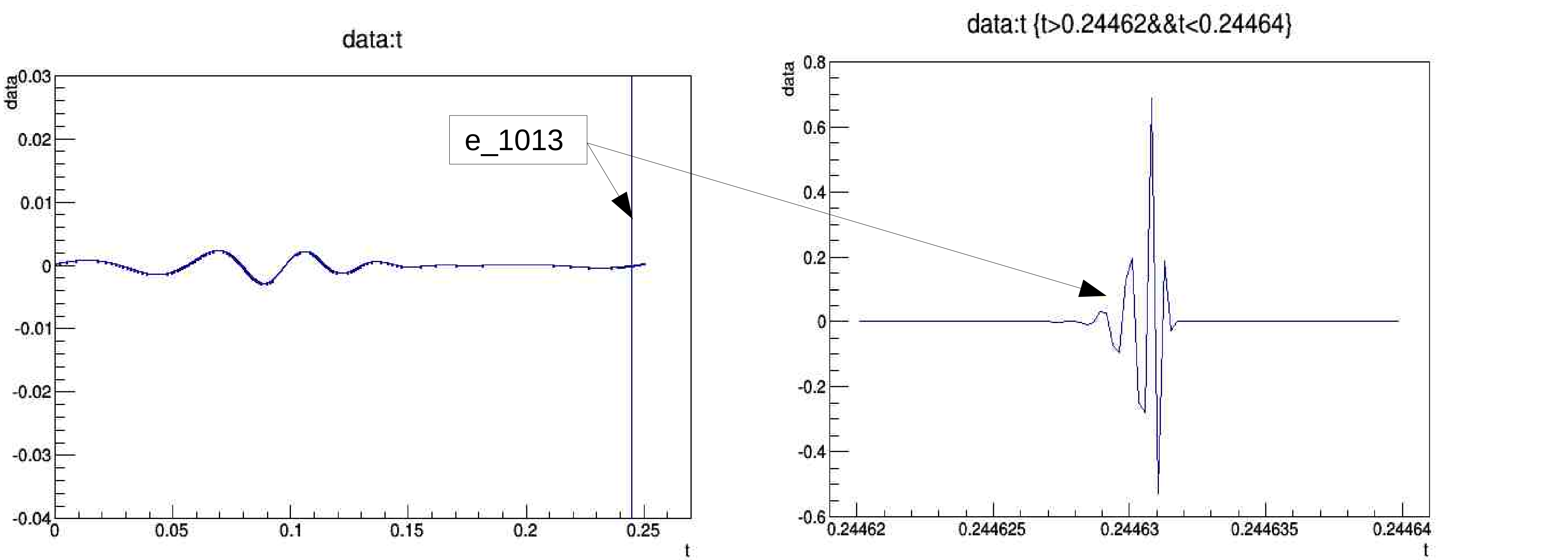
Wavelet basics

- As can be seen from the matrix, the first line takes a weighted sum of a few consecutive samples. One can think of this as a kind of averaging, to smoothen the function. The second row enhances the differences between consecutive data points, and can be viewed as a differencing operation. The symmetric or smooth responses are called mother functions and the difference responses wavelets.
- After one transform, the result is a vector with interleaved smooth (s) and difference (d) results.
- The difference results are kept and moved to the end, and the transform is repeated on the smooth results (but now the length of the transform is halved).
- This is repeated until you end up with the 2 “super-smooth” mother functions S and N-2 wavelets D:

$$
\left[\begin{array}{c} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \end{array}\right]
\xrightarrow{13.10.1}
\left[\begin{array}{c} s_0 \\ d_0 \\ s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \\ s_5 \\ d_5 \\ s_6 \\ d_6 \\ s_7 \\ d_7 \end{array}\right]
\xrightarrow{\text{permute}}
\left[\begin{array}{c} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ \hline d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array}\right]
\xrightarrow{13.10.1}
\left[\begin{array}{c} S_0 \\ D_0 \\ S_1 \\ D_1 \\ S_2 \\ D_2 \\ S_3 \\ D_3 \\ \hline d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array}\right]
\xrightarrow{\text{permute}}
\left[\begin{array}{c} S_0 \\ S_1 \\ S_2 \\ S_3 \\ \hline D_0 \\ D_1 \\ D_2 \\ D_3 \\ \hline d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array}\right]
\xrightarrow{\text{etc.}}
\left[\begin{array}{c} \mathcal{S}_0 \\ \mathcal{S}_1 \\ \hline \mathcal{D}_0 \\ \mathcal{D}_1 \\ \hline D_0 \\ D_1 \\ D_2 \\ D_3 \\ \hline d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array}\right]
\qquad (13.10.7)
$$

Wavelet basics

- The wavelet transform is a linear operation, and the base is orthogonal and complete. The inverse transform exists. Also it is computationally fast (NlogN), just as the Fourier transform. Many relations exist just as in the case of the FFT (like conservation of power).
- Example of DAUB20 wavelets: the different unit vectors are translations in time (for the same order of d) and in scale (for every next transform, the amplitude is reduced by a factor of 2 and the time doubled). Shown are the inverse transform of the 9^{th} and 1013^{th} wavelet for the Daubechy-20 set
- This transform was for 0.25s of data. 1013 is close to the last wavelet (1024,2048,8192,....) so in time domain it is close to the end (around 0.245 s). Events around a certain time contribute to wavelets N, N*2, N*4, N/2, N/4 etc. (e.g. for time around 0.8T you expect wavelets around 112,225,450,900,1800,3600,7200,14400,.....) to contribute.

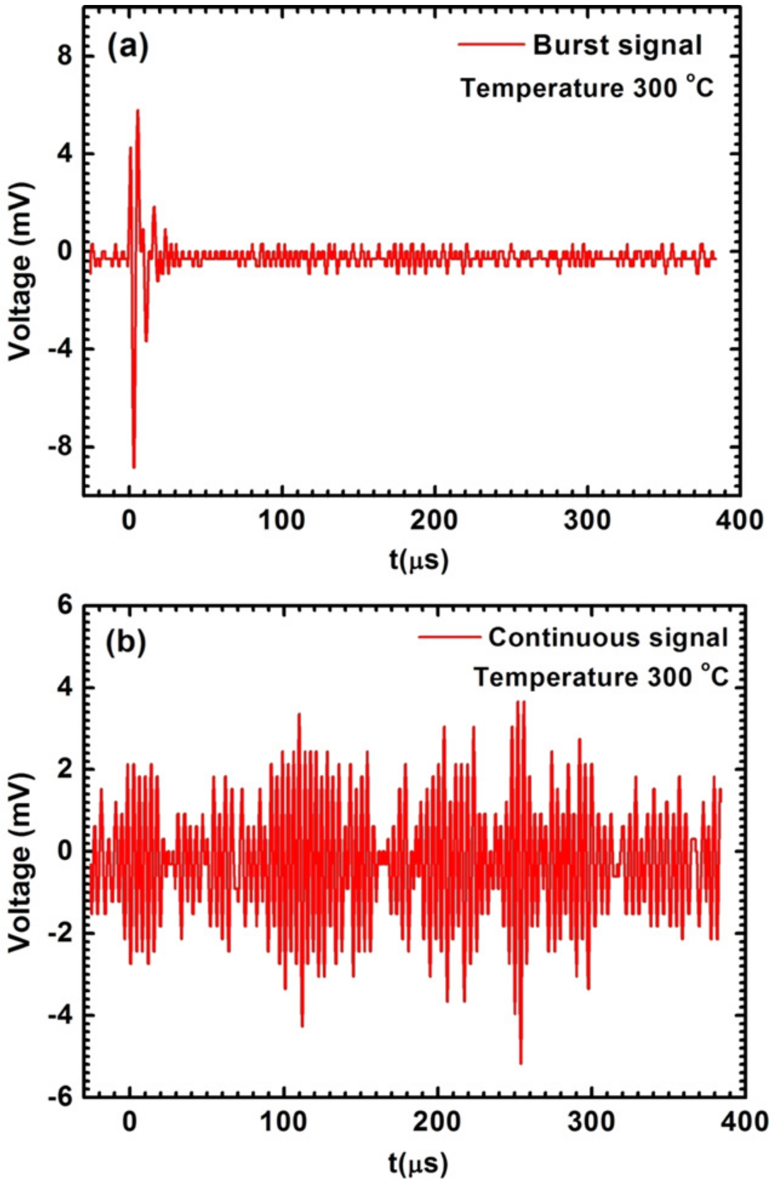


Acoustic Emission

- Many (>1000) papers and even magazines. Still it was difficult for me to see what to expect for normal steel under static load. Typically, one has cyclic loading and the emission rate is then maybe 1 event/cycle/kg material or so.
- For static load one typically expects no events, so maybe a few events per day can be obtained.
- Also, the waveforms can be quite different. I expect that a real acoustic emission exhibits power far in excess of the 30 dB background that the sensor shows.
- To have some guidance of expected signal shapes, I found some explicit discussions about that in Xu (where corrosion crack expansion for steel under static load has been measured. A solution with chemical agents is induced to enforce crack growth):
- Acoustic emission response of 304 stainless steel during constant load test in high temperature aqueous environment, Jian Xu, En-Hou Han ⇑ , Xinqiang Wu ⇑ Xu, Corrosion Science 63 (2012) 91

Wave forms (Xu 2012)

Signals: short duration and fast risetime or many sub-peaks, long duration.



94 J. Xu et al./Corrosion Science 63 (2012) 91–99

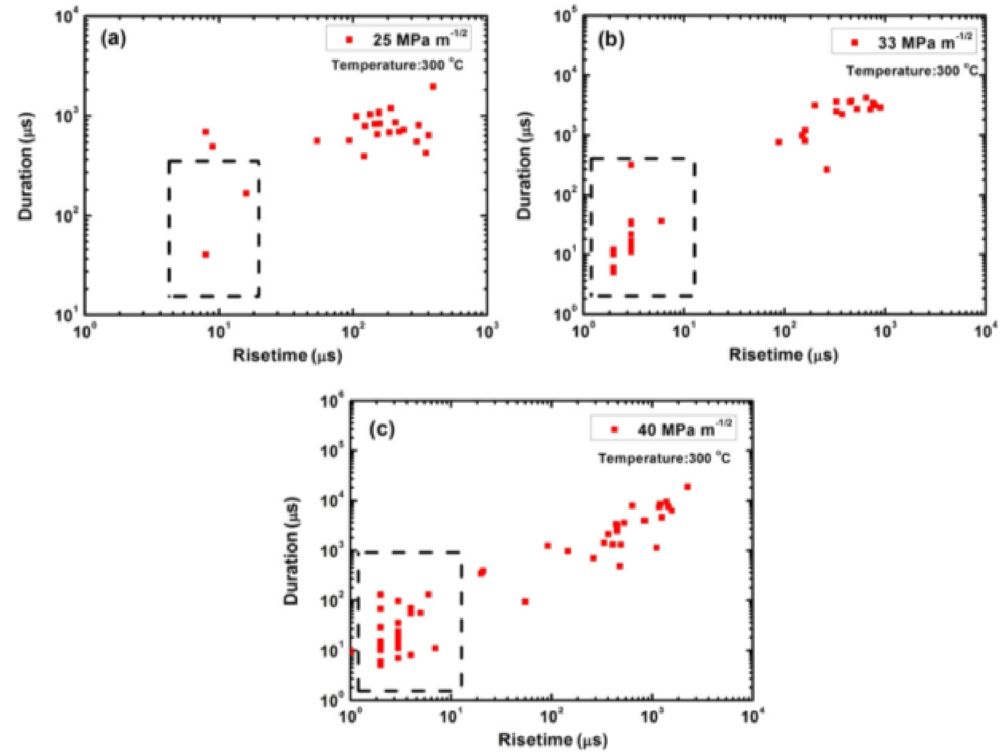
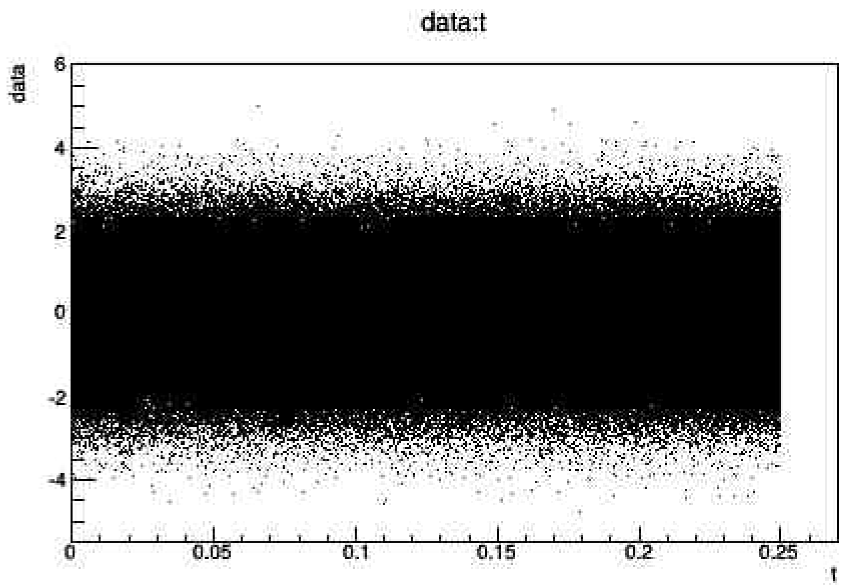
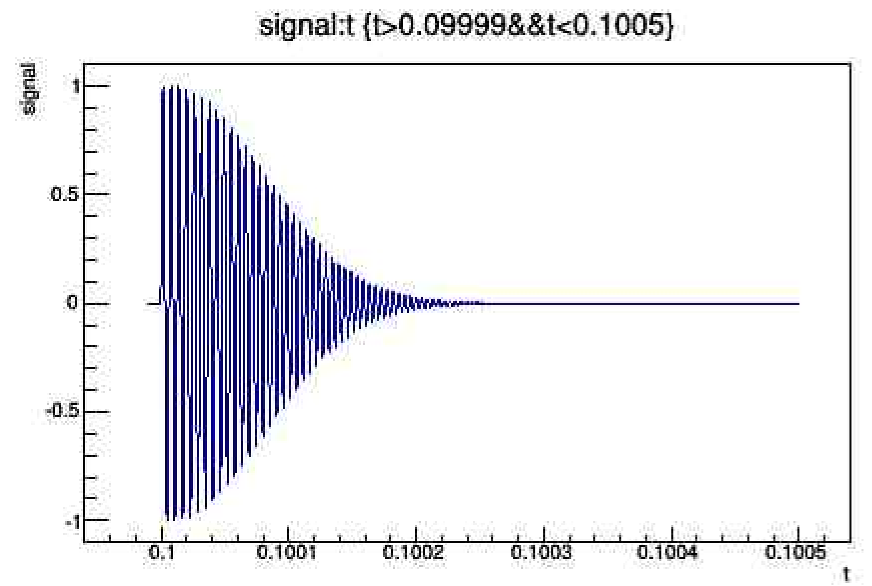


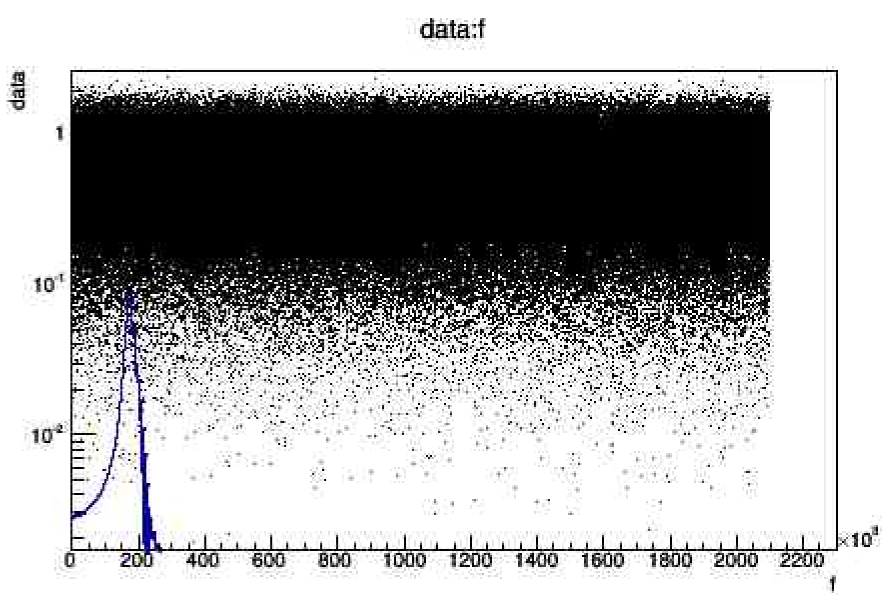
Fig. 4. The cross-plot of risetime and duration of the AE signals detected under different stress intensity factors: (a) 25 MPa m$^{1/2}$, (b) 33 MPa m$^{1/2}$ and (c) 40 Mpa m$^{1/2}$.

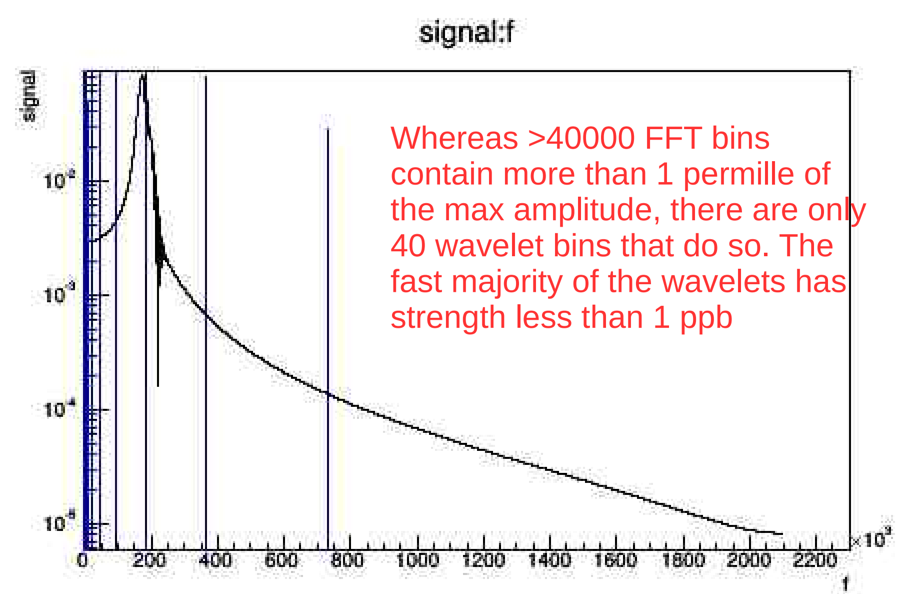
Detection using wavelets

As test, I created 4 M samples of white noise and induced 2 signals. I analyzed this with FFT and wavelet transform. Shown are the data and first signal in time and FFT frequency domain. Lower right plot shows the wavelet bins too.

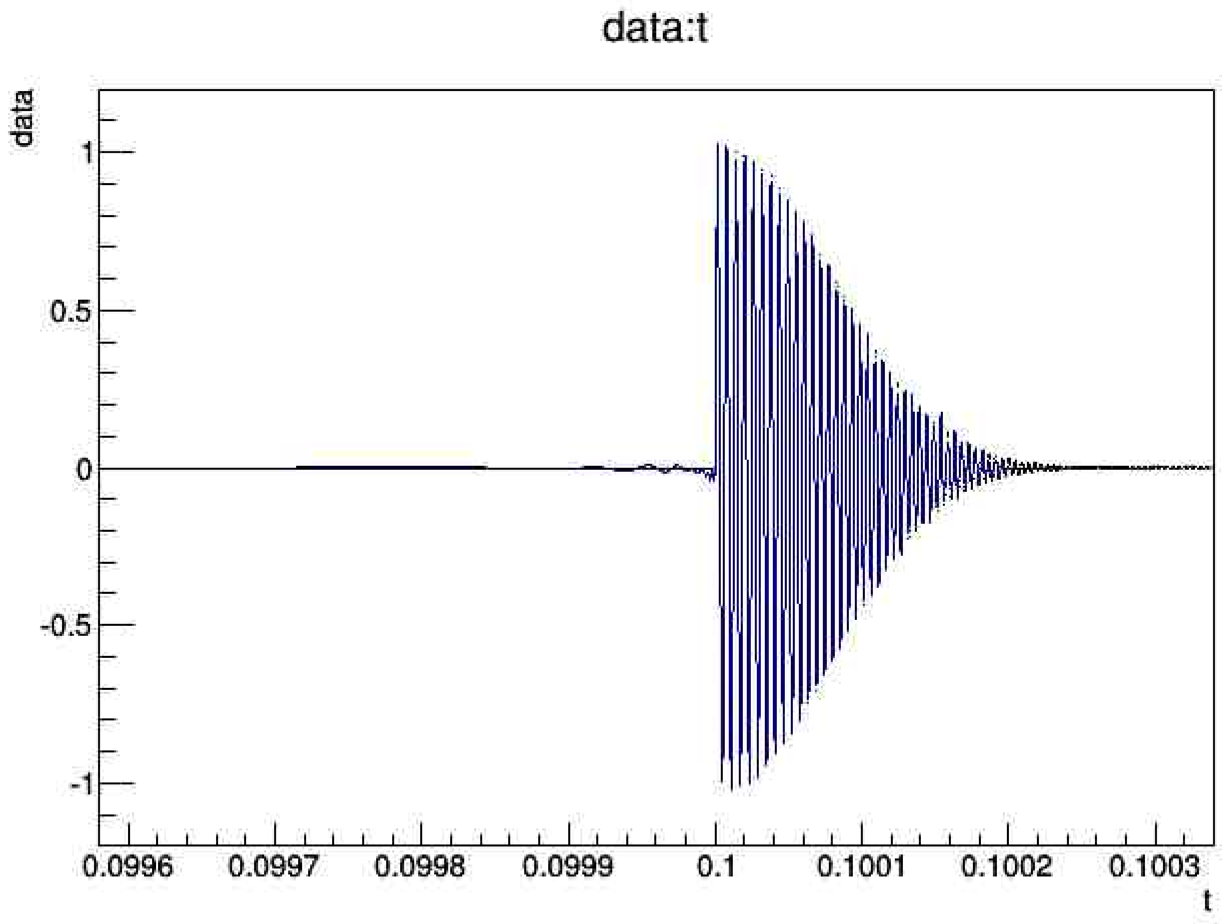






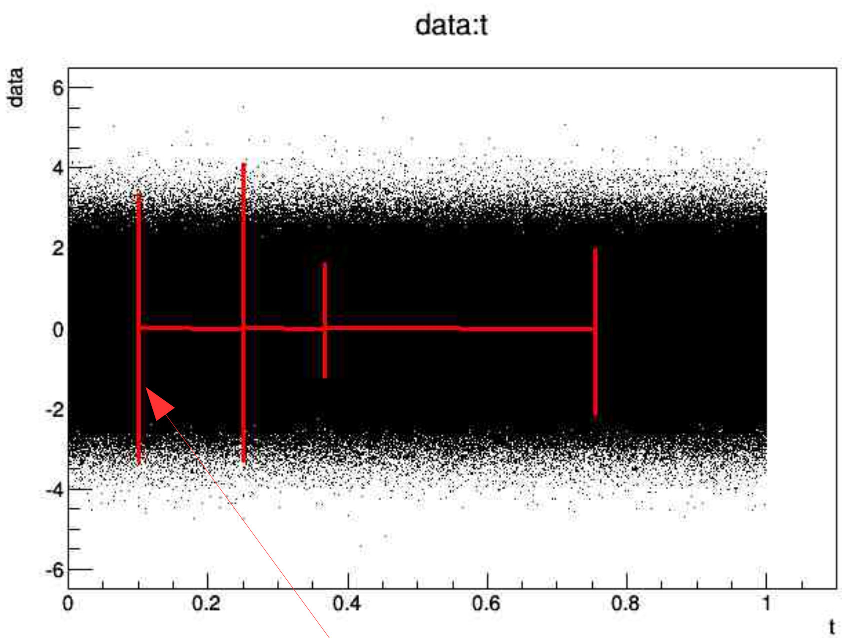


- Denoising operation performed on the signal. The maximum wavelet plus some 40 neighboring wavelets are used. (For the FFT, one would need to include >10000 bins for a similar denoising). Black: injected signal. Blue: wavelet description.

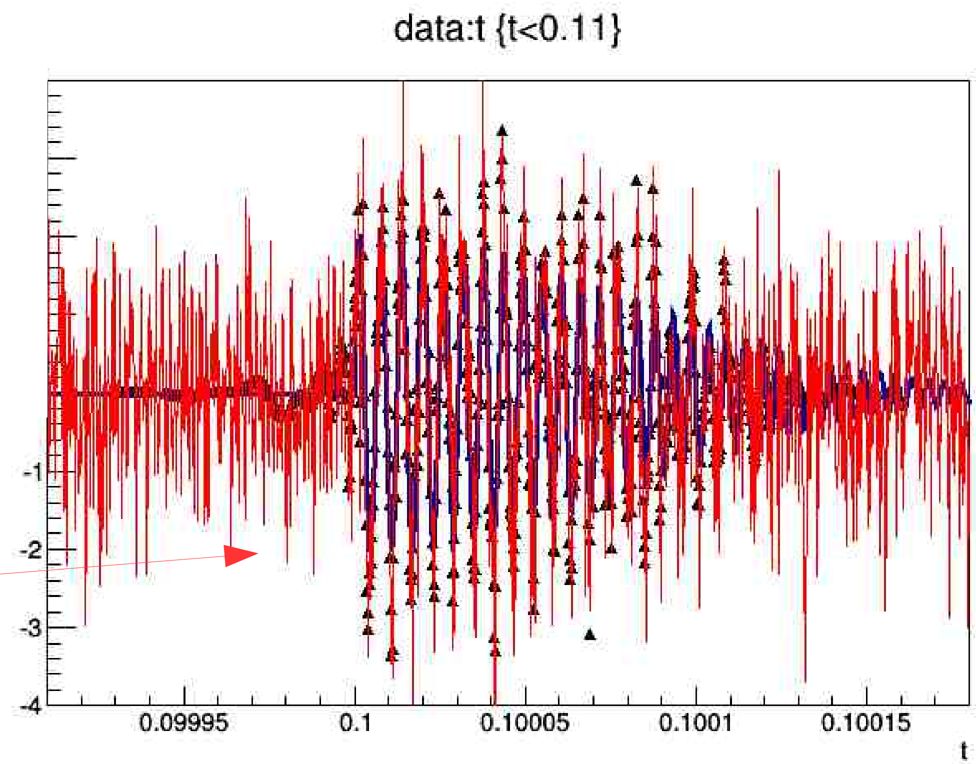


Detection with wavelets

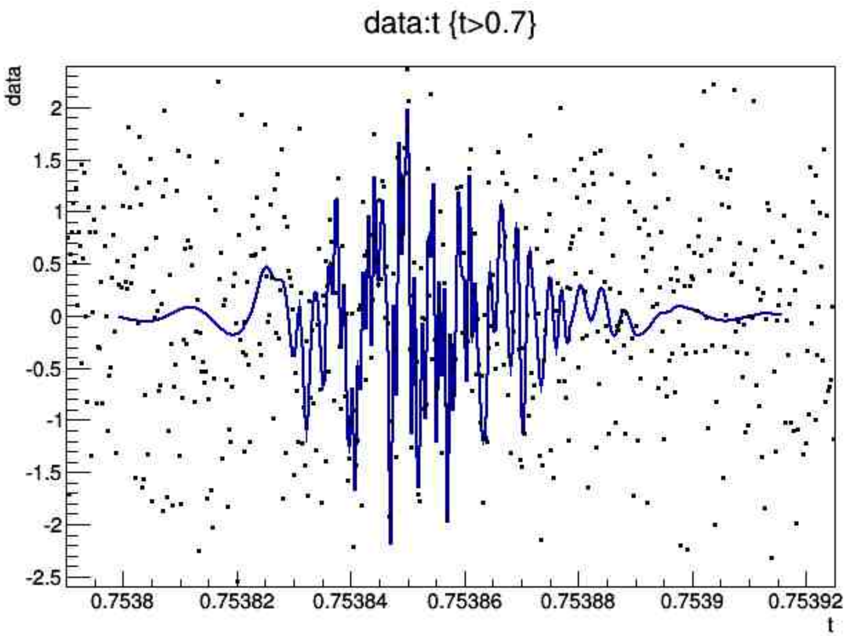
- Denoising: only use the wavelets with a signal > 5 RMS. Set all other wavelets to 0 (except for some neighbouring wavelets, the ones at n-4...n+12, some around n/2, n/4, n*2, n*4).
- Perform inverse transform. Shown are the full data (black) and the cleaned data (red). 4 hits are observed (2 noise hits with sigma>5, and the 2 injected events).

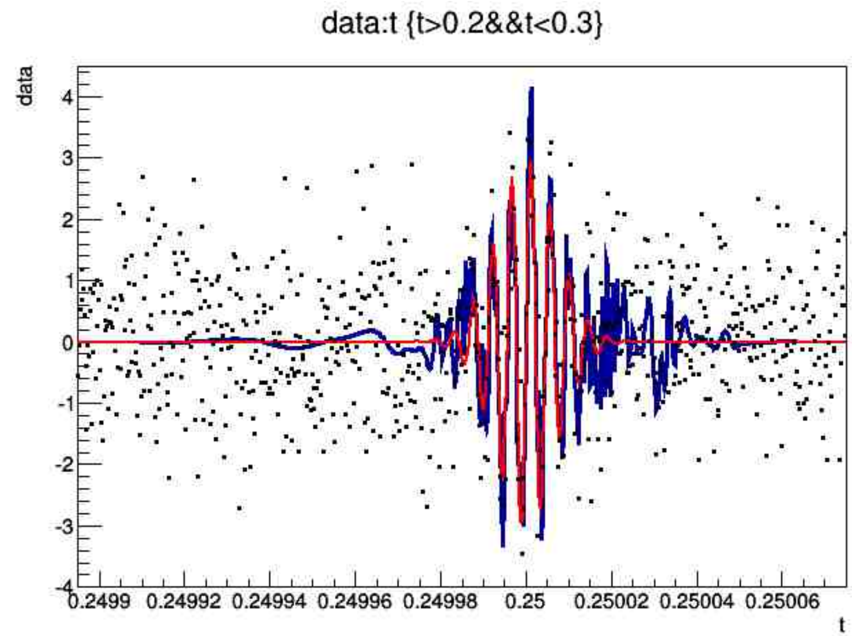


Zoom-in at t=0.1 s. Red: data plus noise. Blue: injected signal. Markers: denoised wavelet transform.



Detection with wavelets





Left: one of the noise events that just came above threshold (black markers are the data and blue the denoised wavelet data)
Right: same for the injected event at t=0.25s. Here, the injected event is shown in red.

Wavelet analysis

- Wavelets seem very well suited to denoise data
  - Localized events in time and in frequency domain
  - Do the wavelet transform; filter out noise e.g. by using a weight on each amplitude, and transform back.
  - Here applied for acoustic emissions: short bursts of ~200 kHz acoustic noise emitted in steel when micro-cracks occur. It is also used in e.g. filtering seismic data, etc.

Optimal filters, Fourier

- Noise is colored: the noise at different frequencies has different amplitudes.
- Divide the Fourier-transform of the data by the expected amplitude of the noise in frequency domain – you are left with a Gaussian-distributed number of samples of the data in frequency domain, with a mean of 0 and an RMS of 1

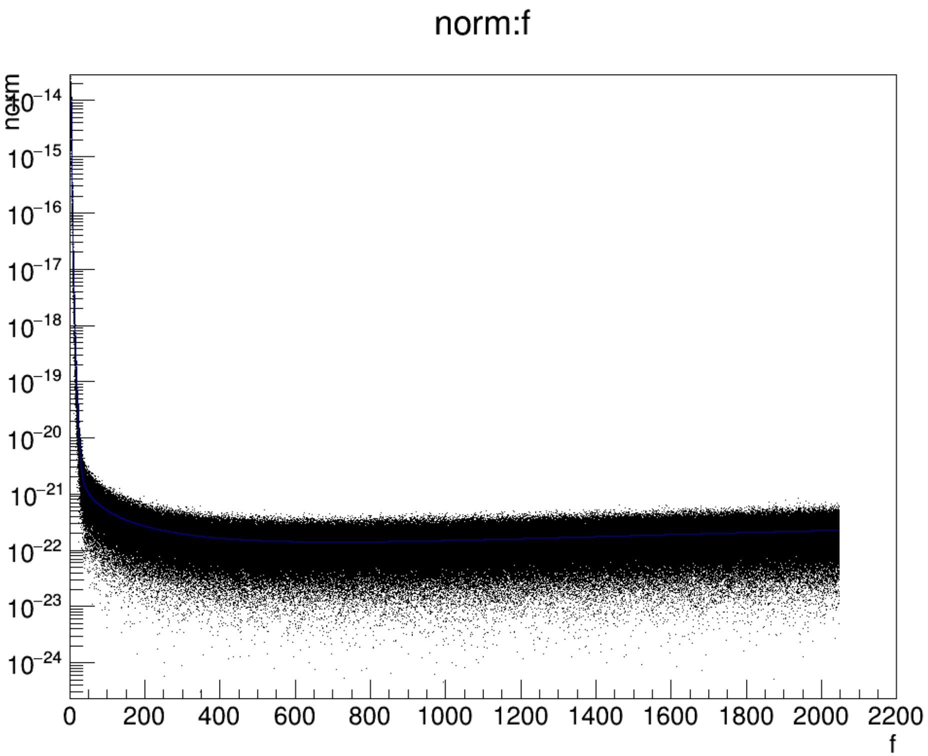
  Each frequency bin can be considered to contain random noise.

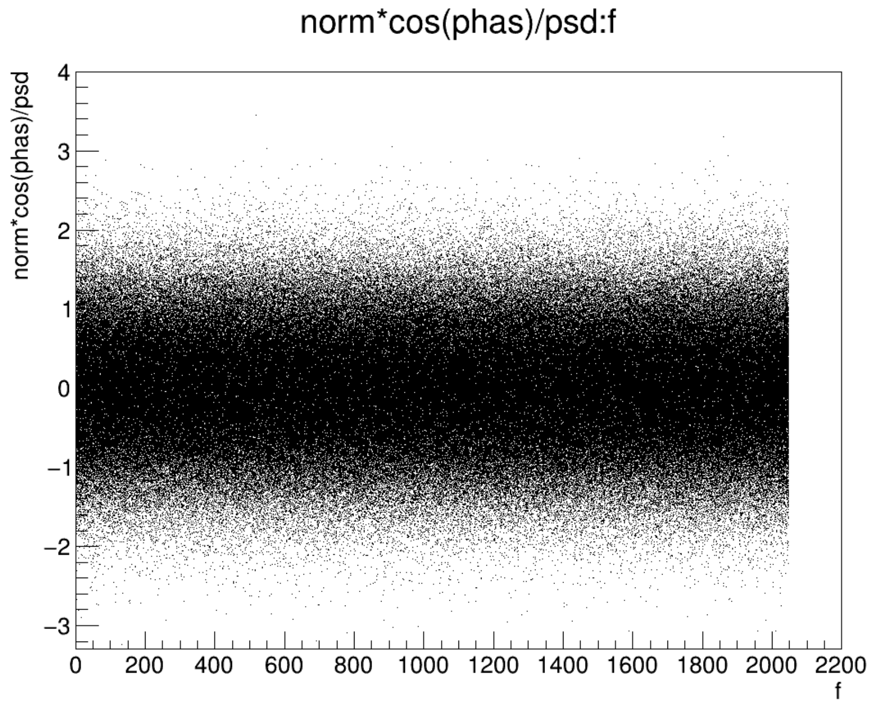
  - If data is present in each bin, it is smeared by the noise
  - If you create a filter in which the weights sum up to 1 $\sum_{i=0}^{N-1} f_i^2 = 1$

    then the product of the filter with the noise looks like a random trial: on average you will find 0 , with a RMS of 1. This is because random noise trials can be added quadratically.
  - If there is a signal present in the data, and it resembles the shape of the filter, then it adds coherently: for every bin the phase of the signal and the filter are the same. Thus instead of adding N random numbers with random phases, one adds linearly N numbers.
  - The signal-to-noise ratio then grows linearly
  - The largest overlap is when the s^2/(s^2+n^2) is maximized. If the noise is larger than the signal, one obtains that by dividing the data by the square root of the power spectral density (the average amplitude in Fourier-domain) and the signal shape as well (since the signal in the data has been divided by that amount, the filter has the same shape as the signal when you apply the same normalization
  -

Exercise 6

- Amplitude of the Noise, and real part of Fourier-transform of the noise divided by sqrt(PSD)





Exercise 6

- Signals in Fourier-domain, normalized to 1e-36
- Since the data is divided by sqrt(PSD), the signal inside the data is also divided by that number, and the optimal filter should be the true signal shape divided by sqrt(PSD) as well (right plot),

signals normalized to 1e-32,divided by PSD