Wavelet Analysis / Line Removal

Presented by S.Klimenko

University of Florida

- Outline
  - ➢Wavelet Analysis Tool
    - ➢Lifting wavelets
    - ➢Daubechie's wavelets
  - ➢Line Removal
    - ➢LineFilter class in the DMT
  - ➢Summary & Plans

WAT

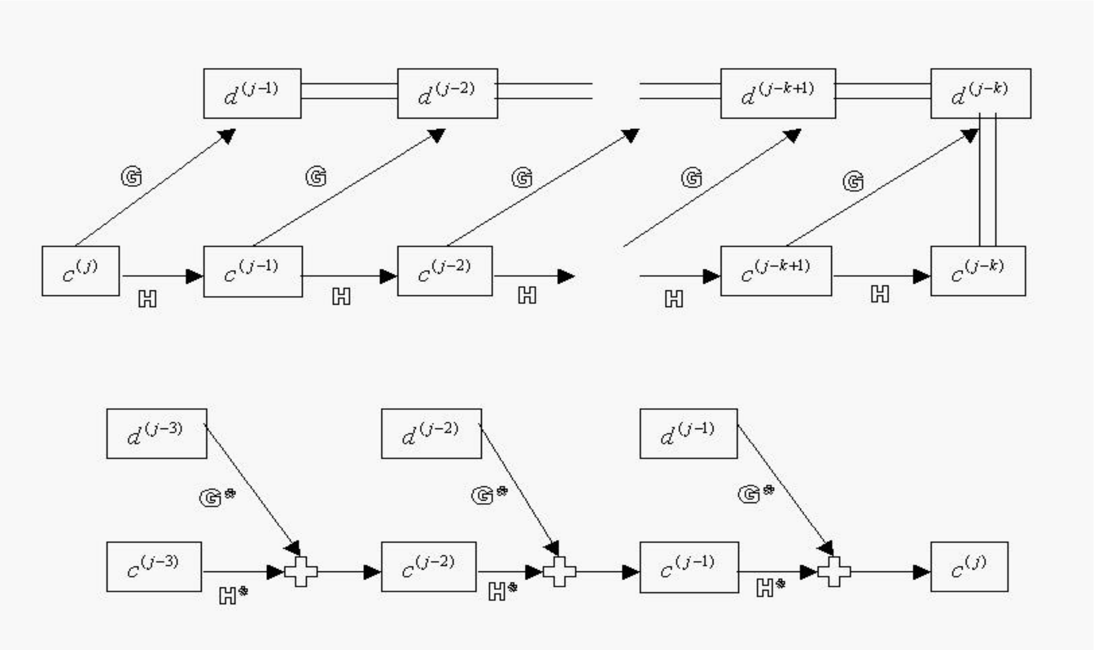
- Objects:
  - WaveletL - family of lifting wavelets
  - WaveletD - family of Daubechies wavelets
  - WaveletI - integer version of lifting wavelets (maps integers to integers)
  - few constructors are provided (wavelet tree, wavelet binary tree, default)
- methods
  - *t2w(ts,level)* - decomposition, *ts* - input time series, *level* - decomposition depth
  - *w2t(ts)* - reconstruction, *ts* - output time series.
  - *getLayer(a,L)* - extract wavelet coefficients from layer *L* into array *a*.
  - *putLayer(a,L)* - fill wavelet layer *L* with data from array *a*.
  - ……….
- ROOT macro-files
  - *WSpectrum(w,"title",pallet)* - time-frequency plot for wavelet *w*.
  - *Plot(w,L)* - plot wavelet coefficients for layer *L*

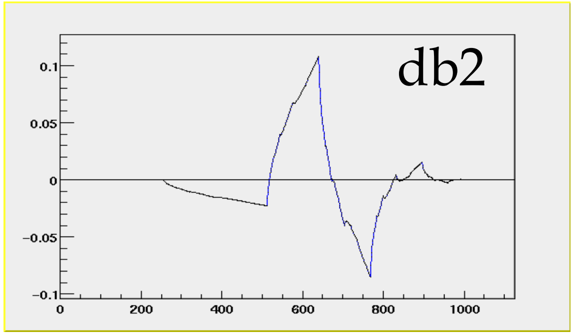
Daubechies Wavelets

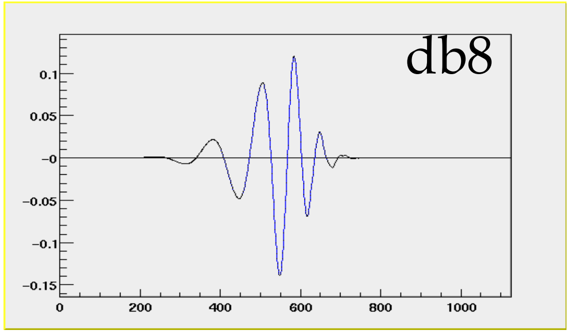
- first compactly supported orthogonal wavelets with pre-assigned degree of smoothness.

  I.Daubechies. Ten Lectures on Wavelets, #61 in CBMS-NSF series in App.Math. SIAM, Philadelphia, 1992.

- Fast Wavelet Transform using cascade algorithm





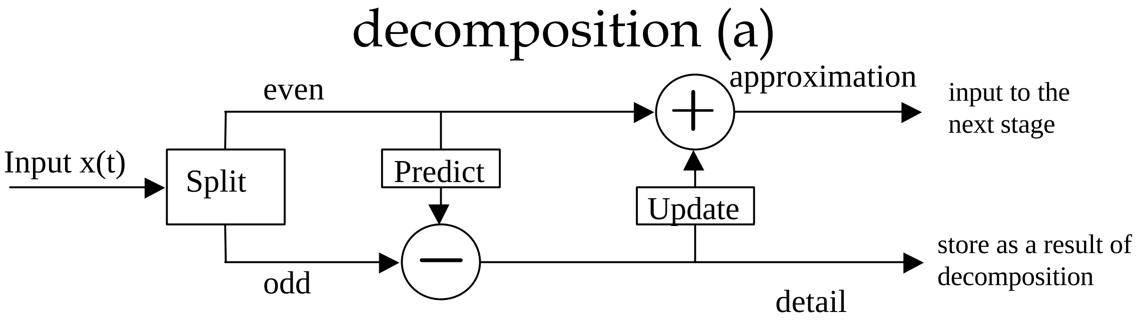


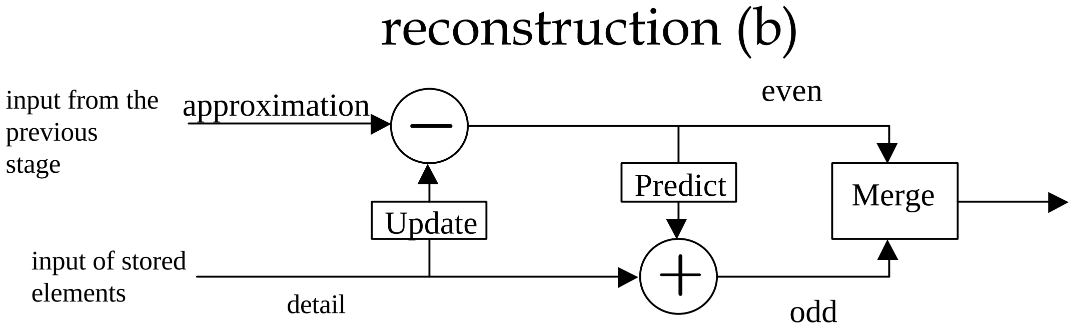
$(Hx)_k = Sh_{n-k}x_n$, $(Gx)_k = Sg_{n-k}x_n$; h,g - wavelet filters

S.Klimenko

Lifting Wavelet Transform

R.C.Calderbank, I.Daubechies, W,Sweldwns, B.L.Yeo. ACHA, V5, N3, pp. 332-369, 199
Wavelet Transforms that Maps Integers to Integers.



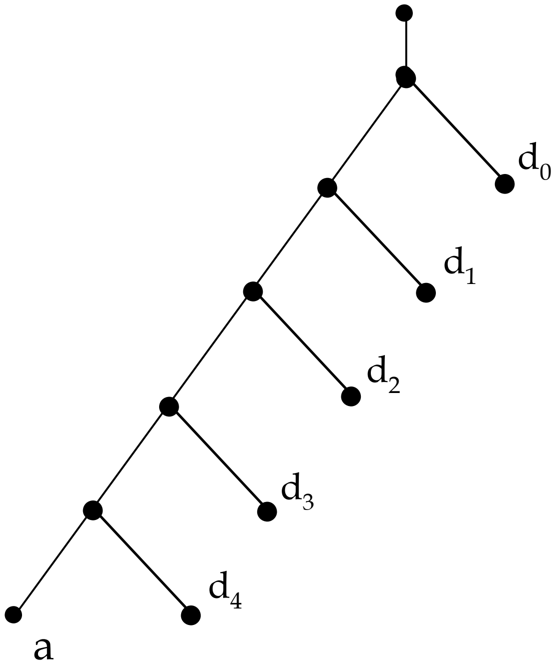


- twice faster than Fast Wavelet Transform.
- allows transforms that map integers to integers:

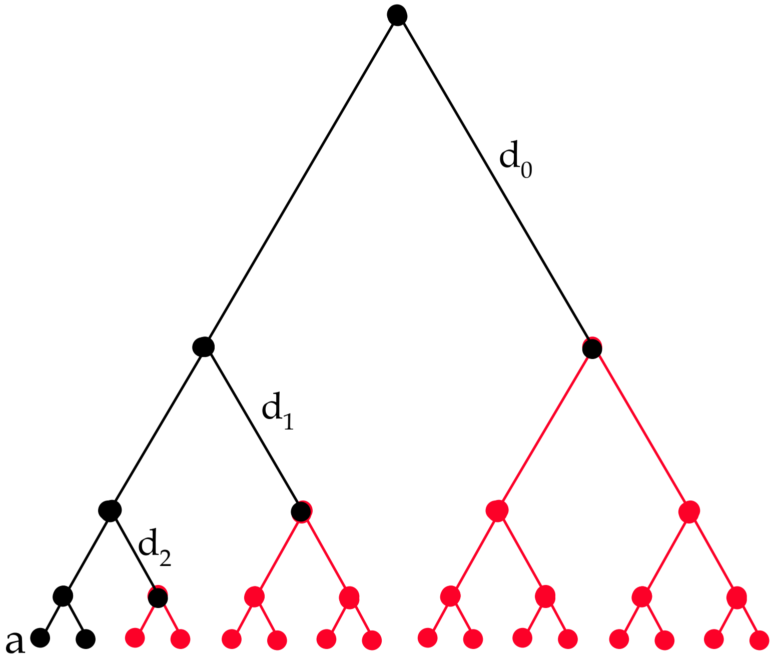
P_I = int(P), U_I = int(U) (for lossless compression)

S.Klimenko

Wavelet Transform Tree



a. wavelet transform tree



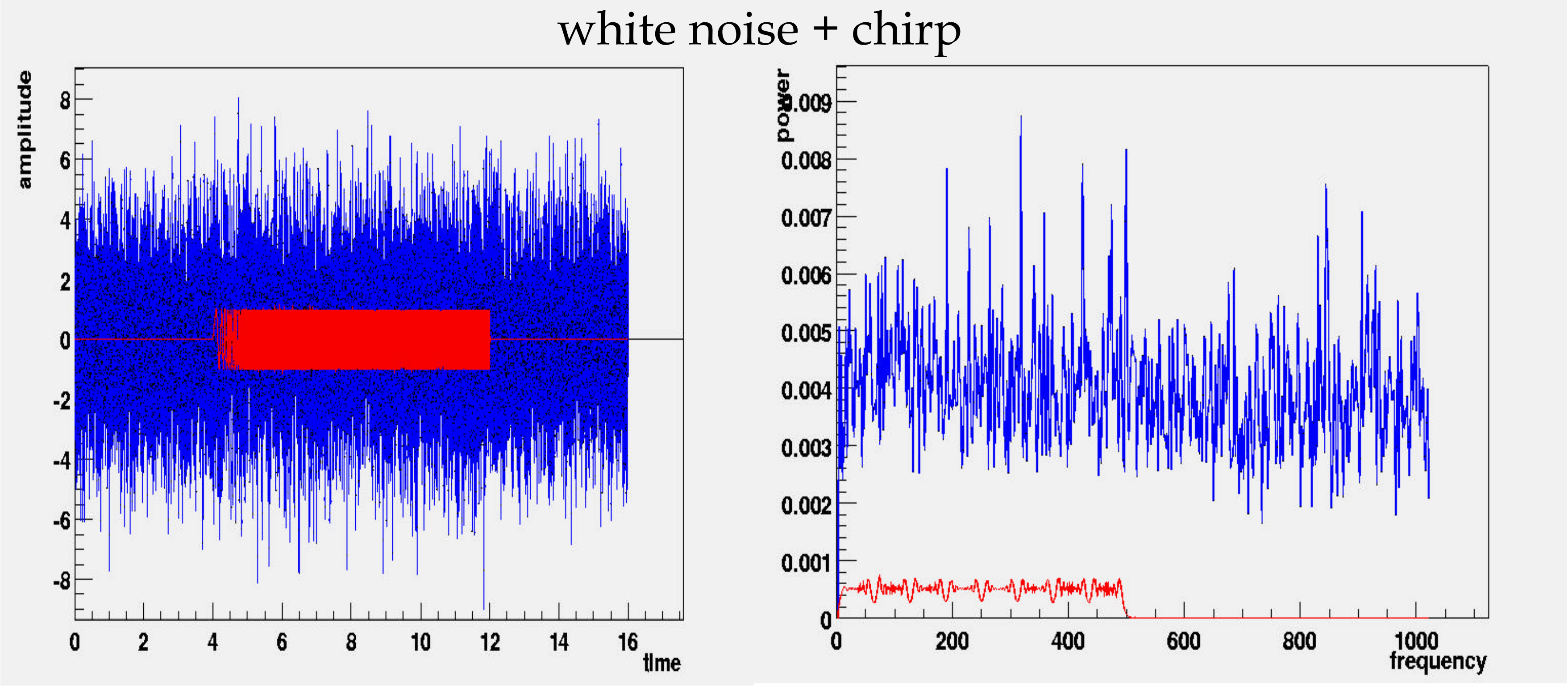
b. wavelet transform binary tree

- detail coefficients d_i represent data in different frequency bands
  - a. $df = f/2, f/4, f/8,$ ….. - dyadic basis
  - b. $df = f/n$, n - number of nodes in last layer - linear basis

S.Klimenko

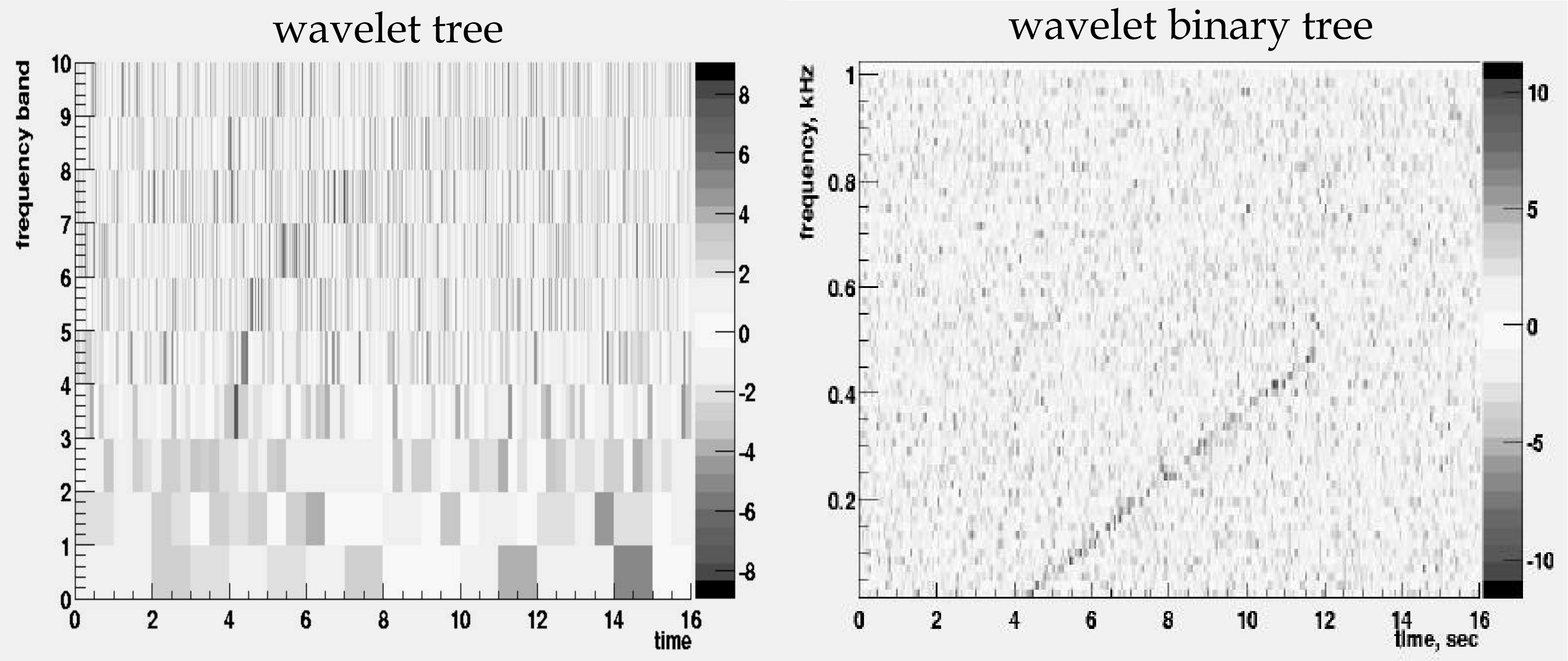
example

white noise + chirp



Wavelet time-frequency plots

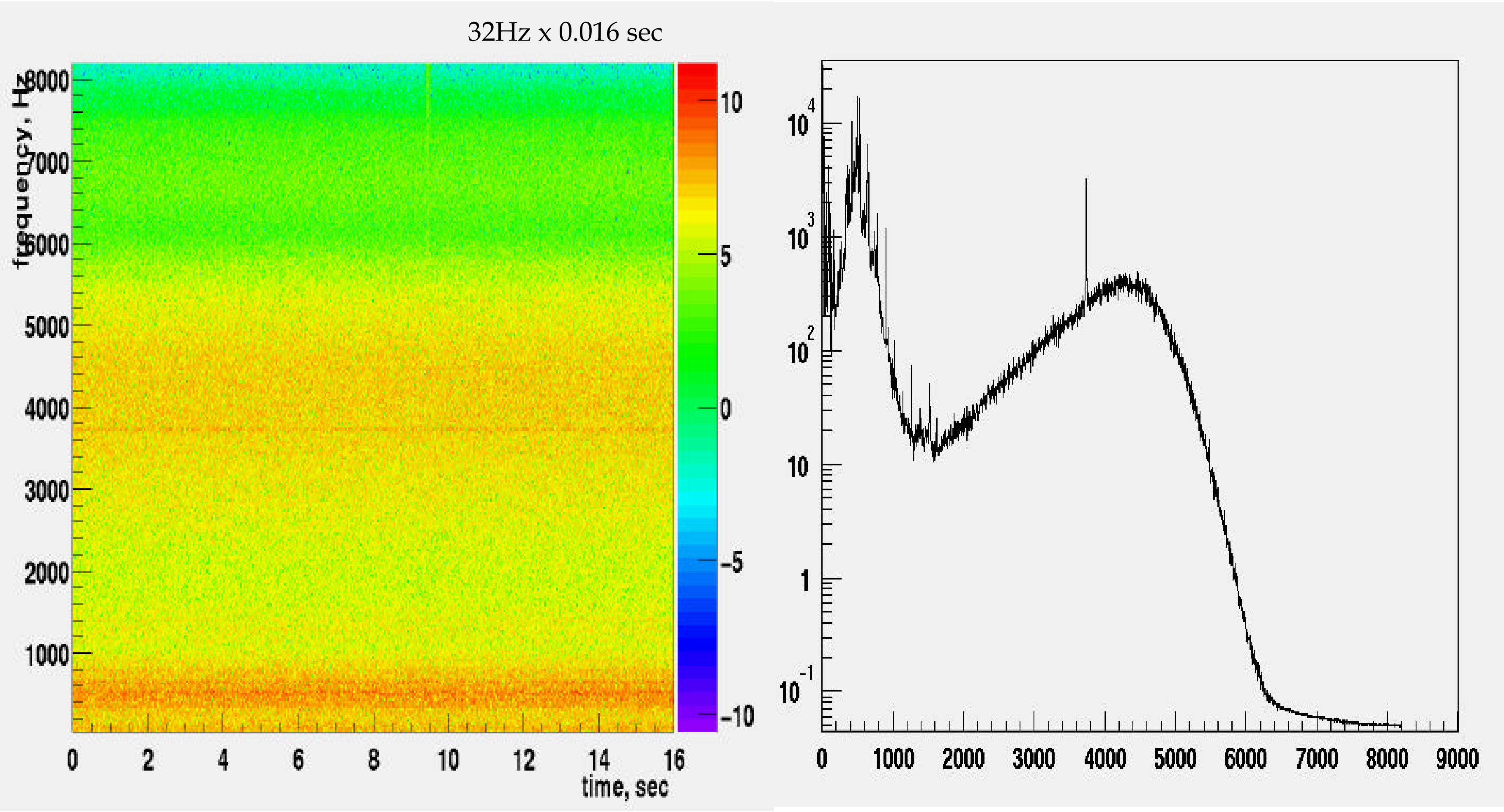
- data: white Gaussian noise + linear chirp
- wavelet: db6 (class WaveletD)
- wavelet tree: 10 frequency bands - *df* = 1/2, 1/4, 1/8, 1/16, ….
- wavelet binary tree: 64 frequency bands - *df* = 1/64



S.Klimenko

Wavelet time-frequency plot

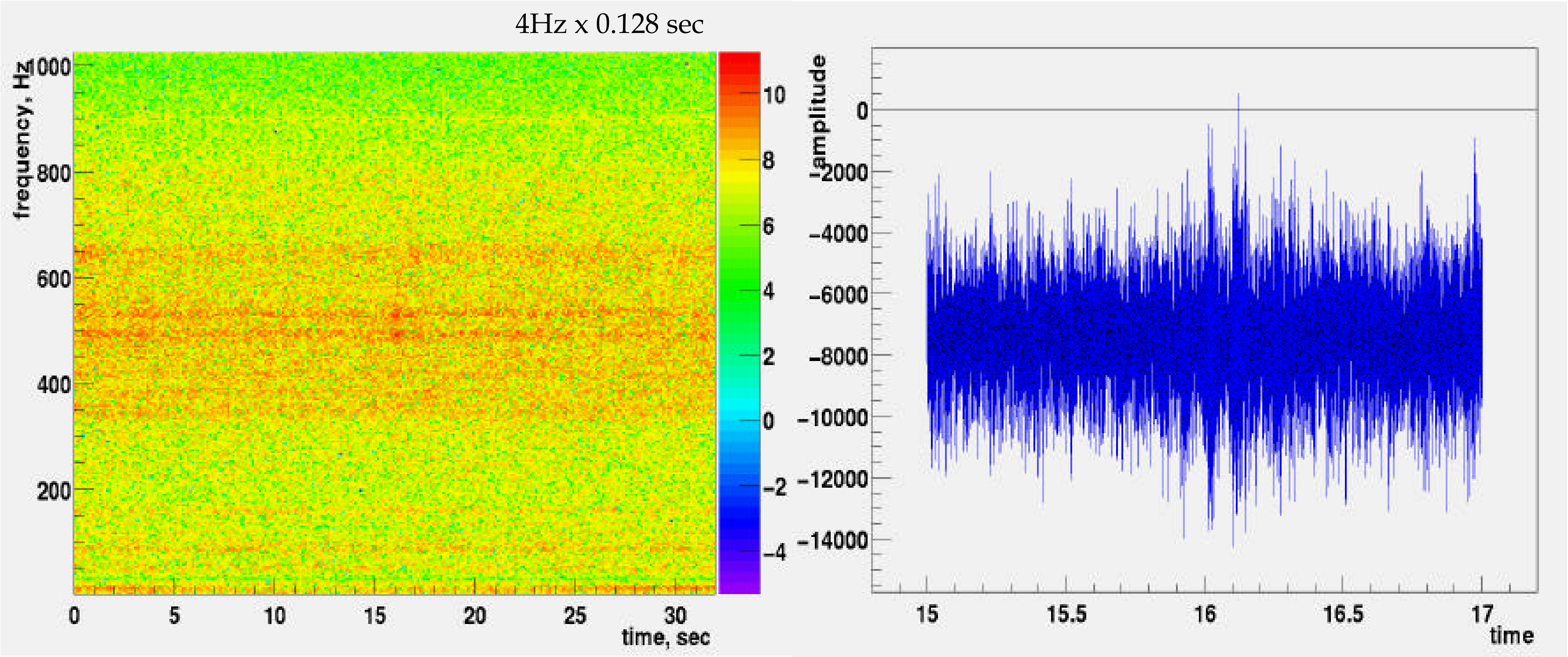
- LSC-AS_I_TEMP
  - *w.t2w(x,8) - 256 wavelet layers*



S.Klimenko

Combined WT-WBT time-frequency plot

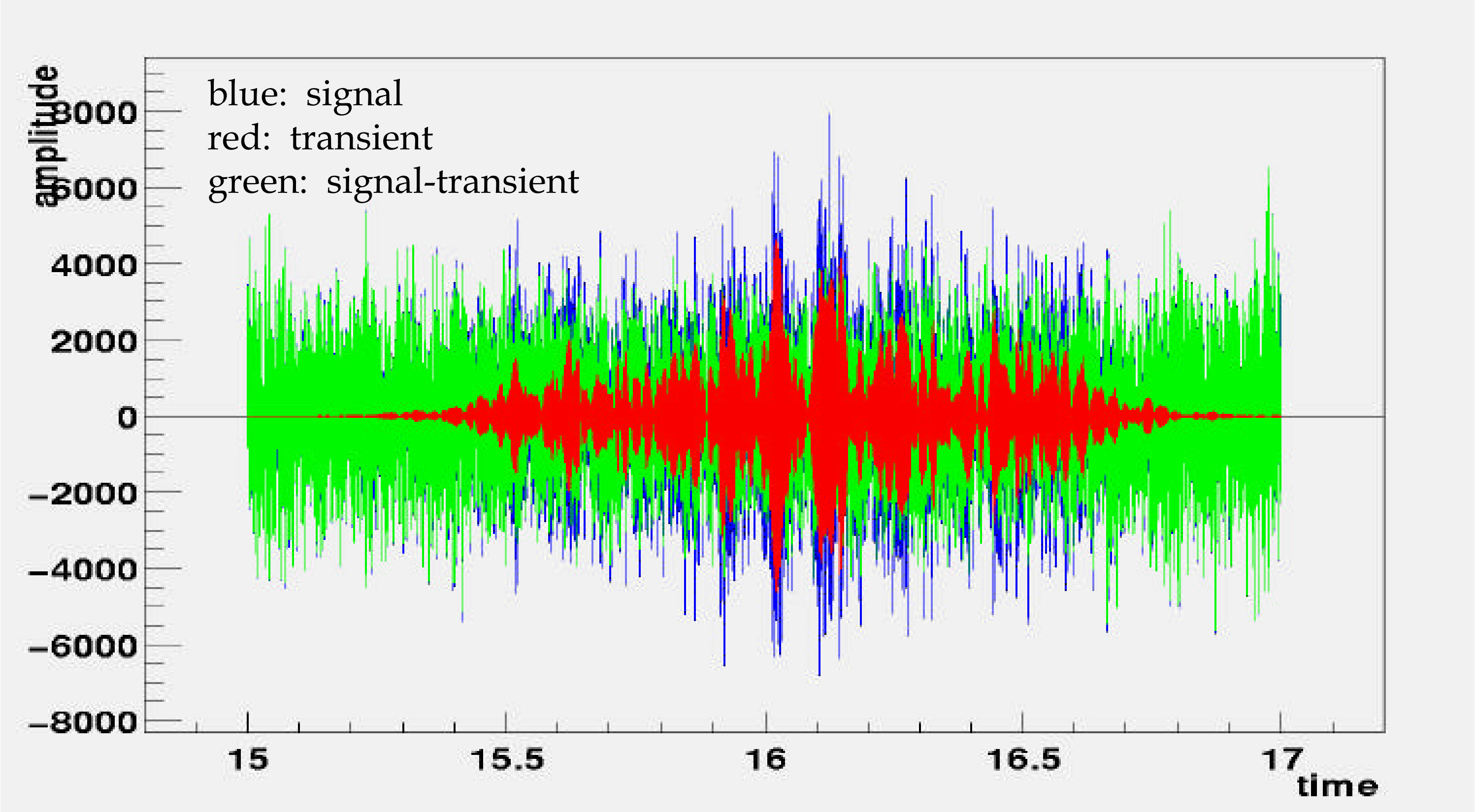
- LSC-AS_I_TEMP
  - ➢ *wt.t2w(x,3);* - wavelet tree transform *(WT)*
  - ➢ *w.getLayer(y,0);* - get approximation coefficients
  - ➢ *wbt.t2w(y,8);* - wavelet binary tree transform *(WBT)*



S.Klimenko

Transients

- *detection*
- *identification (transient frequency, energy, shape, …..)*
- *removal*



- *Useful tool for Transient Analysis*

S.Klimenko

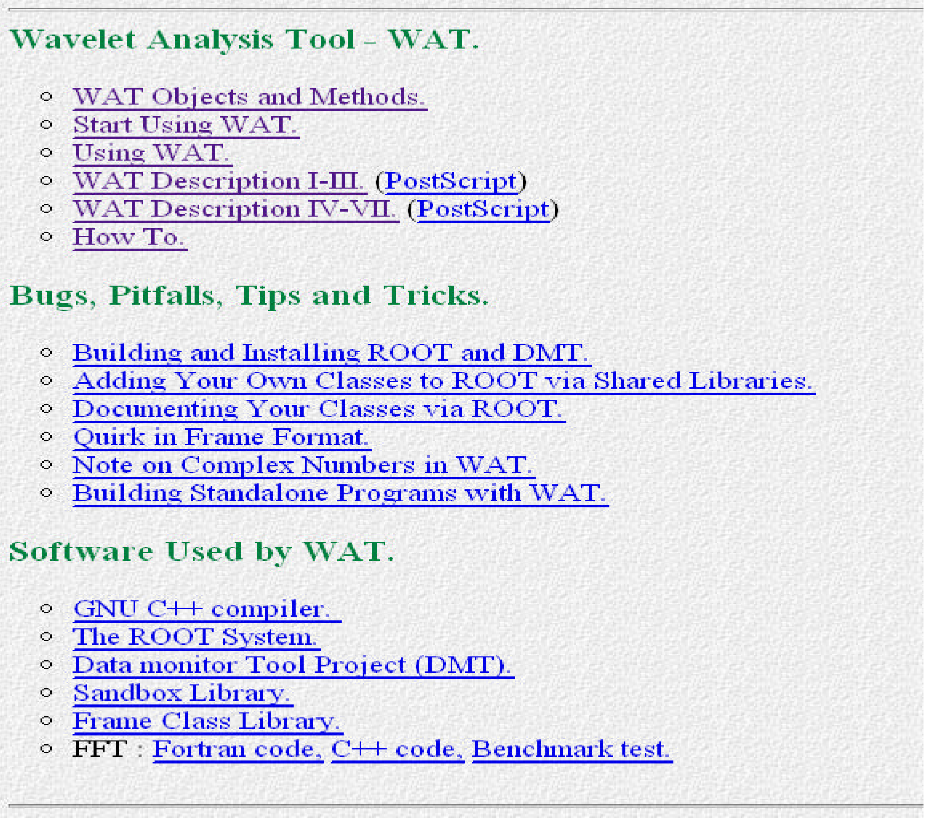
Line Filter

- class to find and remove lines (harmonics with the same fundamental frequency) has been added to DMT.
- construction: *LineFilter(f, fid)*
  - creates *LineFilter* object to remove harmonic lines with fundamental frequency *f* using filter *fid*
- data
  - linked list $(f, E, a_{n1} : a_{n2})$, where E - total energy and $a_{n1} : a_{n2}$ are amplitudes of harmonics *n1:n2*
- methods
  - *setFilter(w, n1, n2)* - set filter, w - min time interval to remove lines, *n1:n2* - range of harmonics.
  - t_{out} = *apply*(t_{in}) - find and remove specified harmonics. t_{in} - input time series, t_{out} - cleaned time series.
  - *findLines*(t_{in}) - find parameters $(f, E, a_{n1} : a_{n2})$ for input t_{in} .
  - currently a quasi-Monochromatic Line Removal algorithm from UF is implemented (*fid* = *0,1*). Other methods can be added.
- *LineFilter* class can be used to develop Line Monitor (a background process to track selected lines)

S.Klimenko

Documentation

- WAT: http://www.phys.ufl.edu/LIGO/wavelet/index.html



Wavelet Analysis Tool - WAT.

- WAT Objects and Methods.
- Start Using WAT.
- Using WAT.
- WAT Description I-III. (PostScript)
- WAT Description IV-VII. (PostScript)
- How To.

Bugs, Pitfalls, Tips and Tricks.

- Building and Installing ROOT and DMT.
- Adding Your Own Classes to ROOT via Shared Libraries.
- Documenting Your Classes via ROOT.
- Quirk in Frame Format.
- Note on Complex Numbers in WAT.
- Building Standalone Programs with WAT.

Software Used by WAT.

- GNU C++ compiler.
- The ROOT System.
- Data monitor Tool Project (DMT).
- Sandbox Library.
- Frame Class Library.
- FFT : Fortran code, C++ code, Benchmark test.

- Line Removal: http://www.phys.ufl.edu/~klimenko

Summary & Plans

- Summary
  - Wavelet transforms (lifting & Daubechie's) are available for use in the DMT.
  - Set of ROOT macro files is available to process and plot data
  - *LineFilter* class to track and remove lines is included into the DMT. Currently the *qMLR* method is implemented. Other methods (*multi-taper*, *CLR*, …) can be added.
  - The *LineFilter* class can be used to develop the Line Monitor.
- Plans
  - Add family of Gaussian wavelets (example: Mexican hat).
  - Add family of Symmlets (symmetric Daubechies wavelets).
  - Transient analysis with wavelets: Development of statistical algorithms for transients detection, identification and removal. Try wavelet algorithms on the engineering run data.