Denoising using wavelets

Dorit Moshe

In today's show

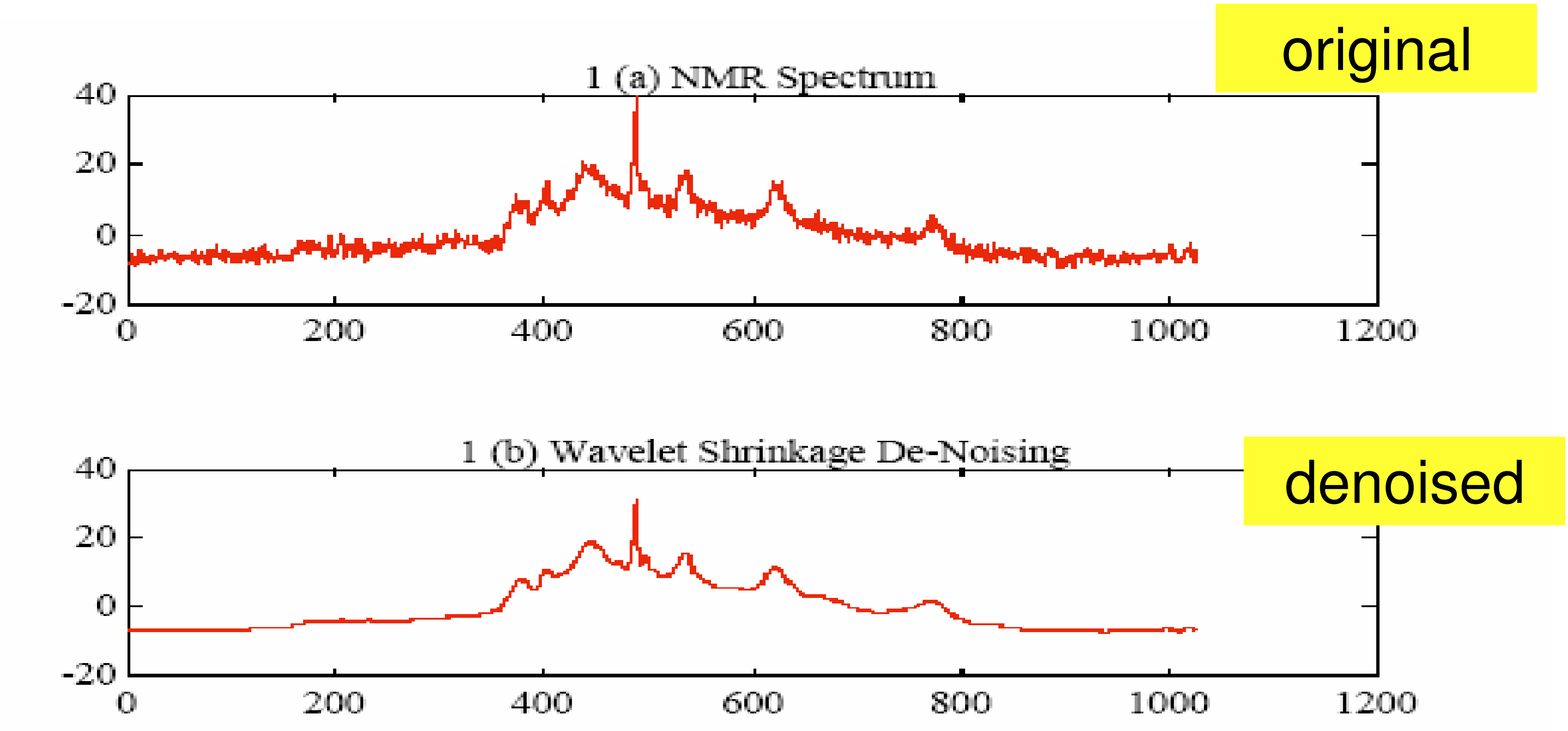
- Denoising – definition
- Denoising using wavelets vs. other methods
- Denoising process
- Soft/Hard thresholding
- Known thresholds
- Examples and comparison of denoising methods using WL
- Advanced applications
- 2 different simulations
- Summary

In today's show

- **Denoising – definition**
- Denoising using wavelets vs. other methods
- Denoising process
- Soft/Hard thresholding
- Known thresholds
- Examples and comparison of denoising methods using WL
- Advanced applications
- 2 different simulations
- Summary

Denoising

➲ ***Denosing*** is the process with which we reconstruct a signal from a noisy one.



In today's show

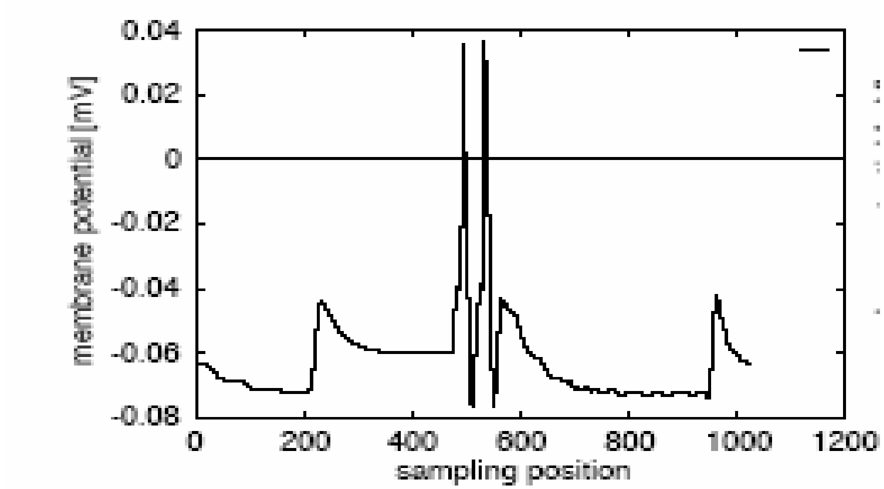
- Denoising – definition
- **Denoising using wavelets vs. other methods**
- Denoising process
- Soft/Hard thresholding
- Known thresholds
- Examples and comparison of denoising methods using WL
- Advanced applications
- 2 different simulations
- Summary

Old denoising methods

What was wrong with existing methods?

➲Kernel estimators / Spline estimators

Do not resolve local structures well enough. This is necessary when dealing with signals that contain structures of different scales and amplitudes such as neurophysiological signals.



➲ **Fourier based signal processing**

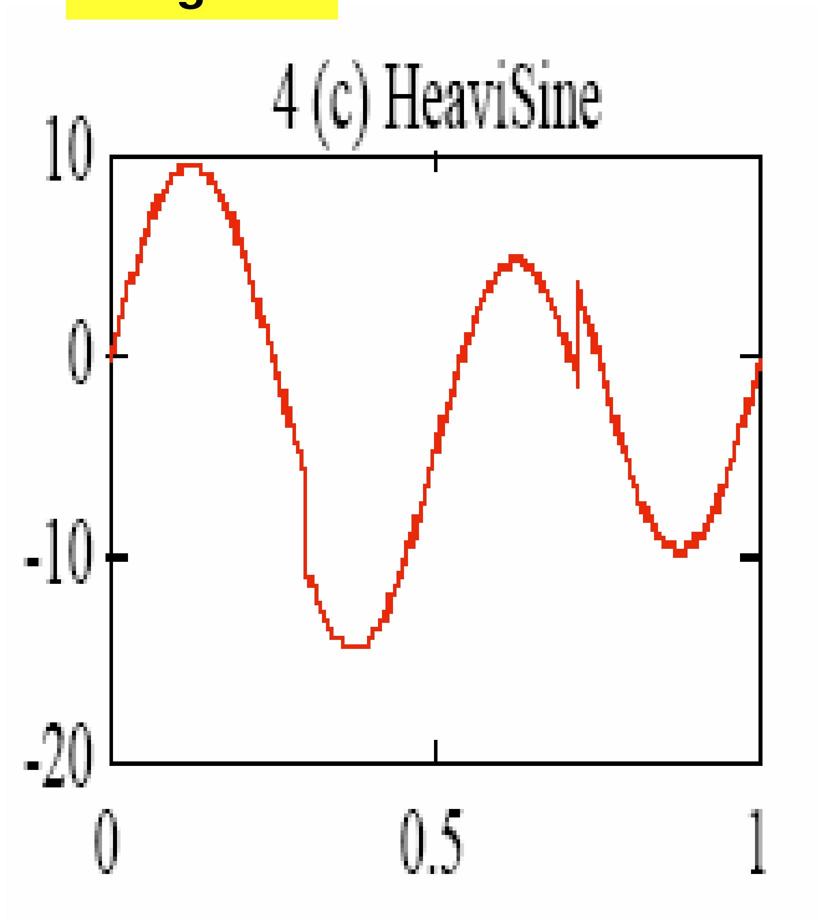
- we arrange our signals such that the signals and any noise overlap as little as possible in the frequency domain and linear time-invariant filtering will approximately separate them.

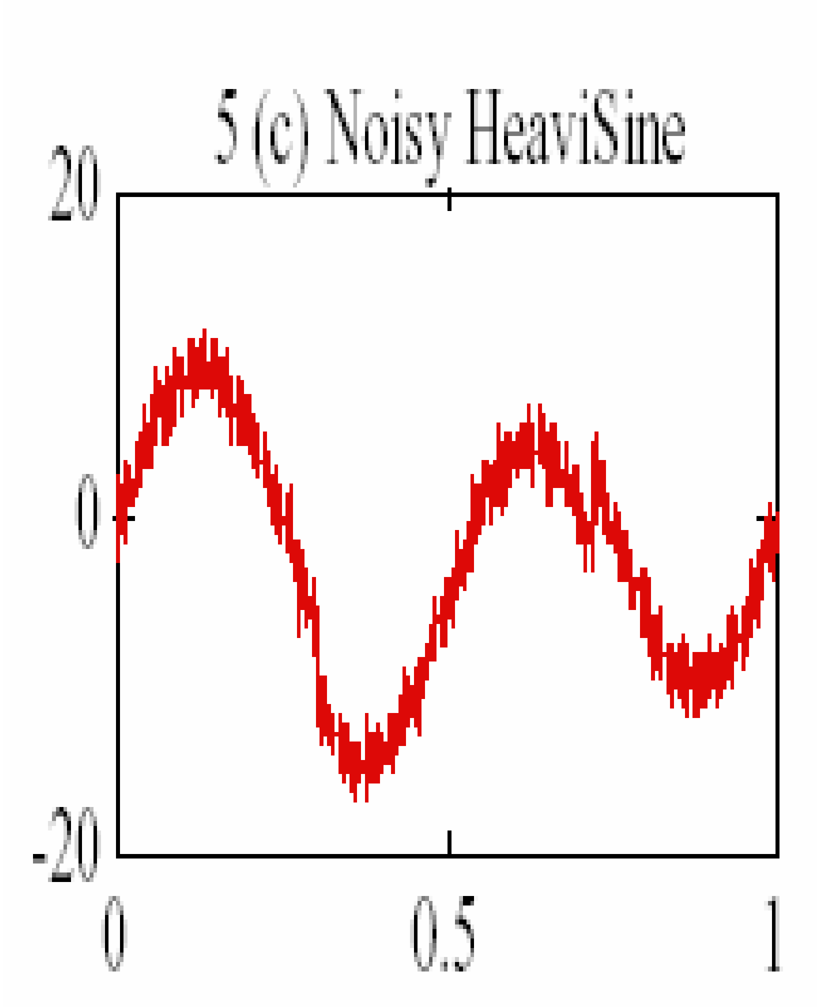
- *This linear filtering approach cannot separate noise from signal where their Fourier spectra overlap.*

Motivation

- Non-linear method
- The spectra can overlap.
- The idea is to have the amplitude, rather than the location of the spectra be as different as possible for that of the noise.
- This allows shrinking of the amplitude of the transform to separate signals or remove noise.

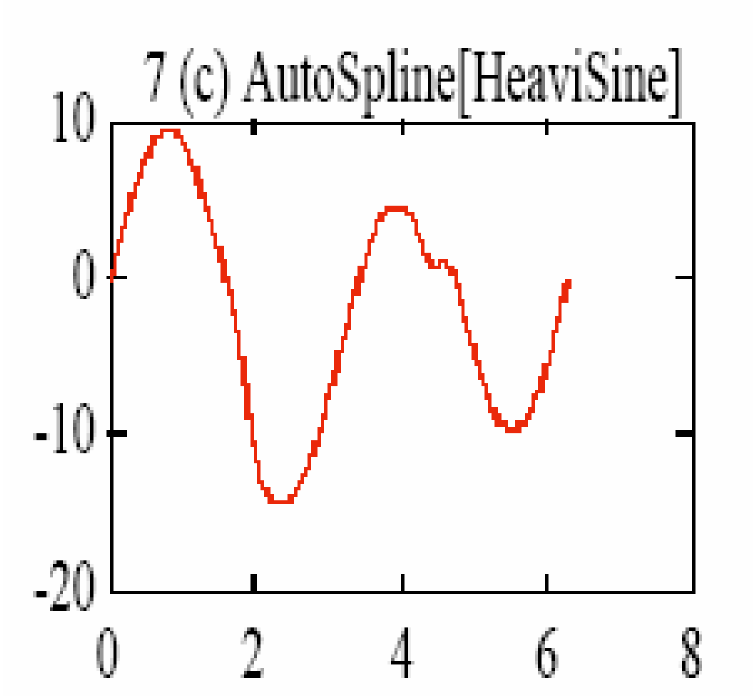
original



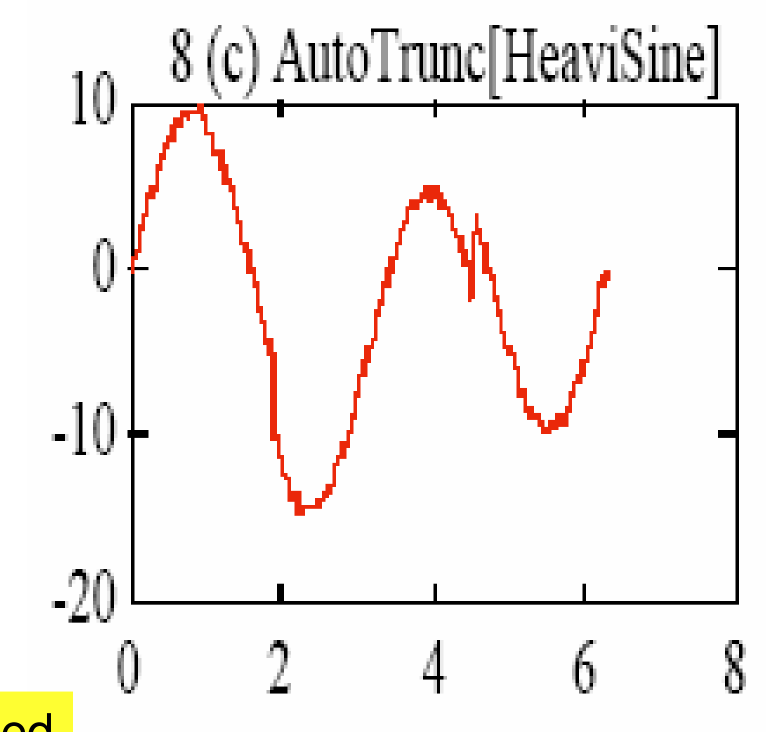


noisy

➲ Spline method - suppresses noise, by broadening, erasing certain features



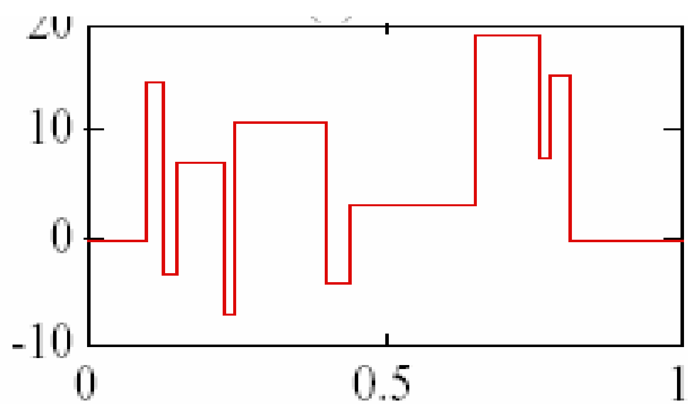
➲ Fourier filtering – leaves features sharp but doesn't really suppress the noise

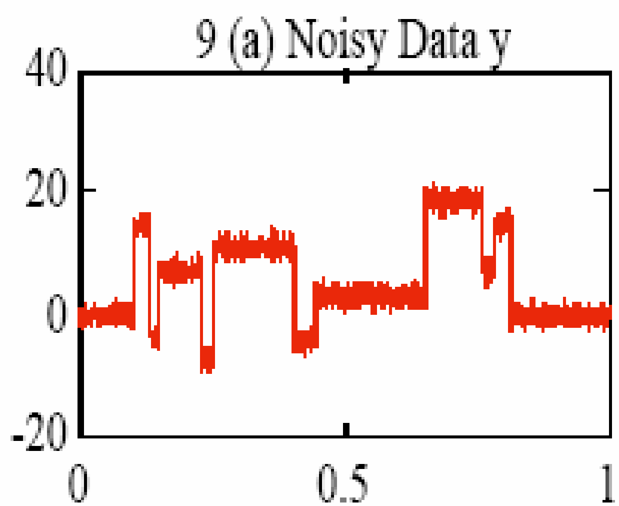


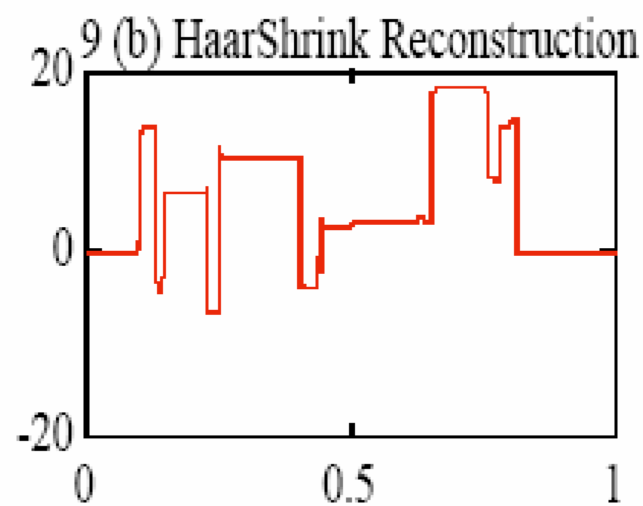
denoised

- Here we use Haar-basis shrinkage method

original







Why wavelets?

- The Wavelet transform performs a correlation analysis, therefore the output is expected to be maximal when the input signal most resembles the mother wavelet.
- If a signal has its energy concentrated in a small number of WL dimensions, its coefficients will be relatively large compared to any other signal or noise that its energy spread over a large number of coefficients

Localizing properties + concentration

- This means that shrinking the WL transform will remove the low amplitude noise or undesired signal in the WL domain, and an inverse wavelet transform will then retrieve the desired signal with little loss of details

- Usually the same properties that make a system good for denoising or separation by non linear methods makes it good for **compression,** which is also a nonlinear process

In today's show

- Denoising – definition
- Denoising using wavelets vs. other methods
- **Denoising process**
- Soft/Hard thresholding
- Known thresholds
- Examples and comparison of denoising methods using WL
- Advanced applications
- 2 different simulations
- Summary

Noise (especially white one)

- Wavelet denoising works for additive noise since wavelet transform is linear

$$W(a,b)[f+\eta;\psi] = W(a,b)[f;\psi] + W(a,b)[\eta;\psi]$$

- **White** noise means the noise values are not correlated in time
- Whiteness means noise has equal power at all frequencies.
- Considered the most difficult to remove, due to the fact that it affects every single frequency component over the whole length of the signal.

Denoising process

- Finite Length Signal with Additive Noise:

$$y[n] = x[n] + \sigma\, r[n],\ 0 \le n \le N-1$$

Observations of x + noise

Signal

Standard Deviation

Noise : $r[n] \sim N(0,1)$

N-1 = 2^{j+1} -1 dyadic sampling

Goal : recover *x*

In the Transformation Domain: $Y = X + \sigma R$

where: $Wy = Y$ (W transform matrix).

$\hat{X}$ estimate of X from Y, $\hat{x}$ estimate of x from y

Define diagonal linear projection: $\hat{X} = \Delta Y$

$$\Delta = diag\left(\delta_0, \delta_1, \ldots, \delta_{N-1}\right), \delta_i \in [0,1]$$

Estimate: $\boxed{\hat{x} = W^{-1}\hat{X} = W^{-1}\Delta Y = W^{-1}\Delta W y}$

We define the risk measure :

$$R(\hat{X}, X) = E[\| x - \hat{x} \|_2^2] =$$

$$E[\| W^{-1}(\hat{X} - X) \|_2^2] = E[\| \hat{X} - X \|_2^2] =$$

$$E[\| \Delta Y - X \|_2^2] = \begin{Bmatrix} \| Y_i - X_i \|_2^2 = \| N_i \|_2^2, & X_i > \varepsilon \\ \| 0 - X_i \|_2^2 = \| X_i \|_2^2, & X_i < \varepsilon \end{Bmatrix}$$

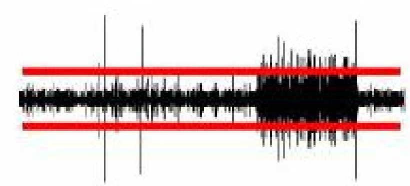
$$\delta_i = 1_{x_i > \varepsilon}$$

$$R_{id}(\hat{X}, X) = \sum_{n=1}^{N} \min(X^2, \varepsilon^2)$$

Is the lower limit of l_2 error

3 step general method

1. Decompose signal using DWT;
   Choose wavelet and number of decomposition levels.
   Compute $\boldsymbol{Y=Wy}$
2. Perform thresholding in the Wavelet domain.
   Shrink coefficients by thresholding (hard /soft)



3. Reconstruct the signal from thresholded DWT coefficients
   Compute $$\hat{x} = W^{-1}\hat{X}$$

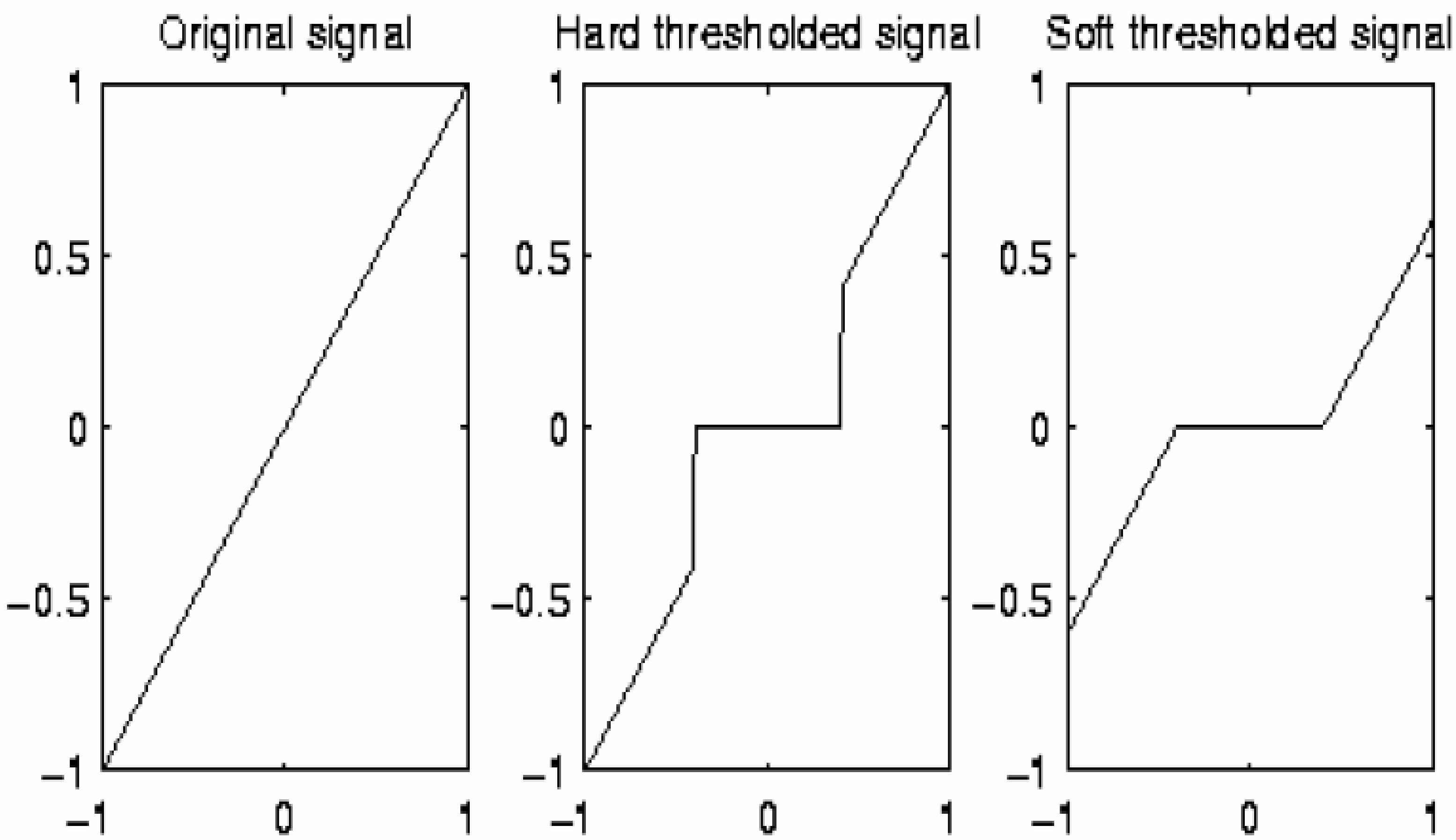
Questions

- Which thresholding method?
- Which threshold?
- Do we pick a single threshold or pick different thresholds at different levels?

In today's show

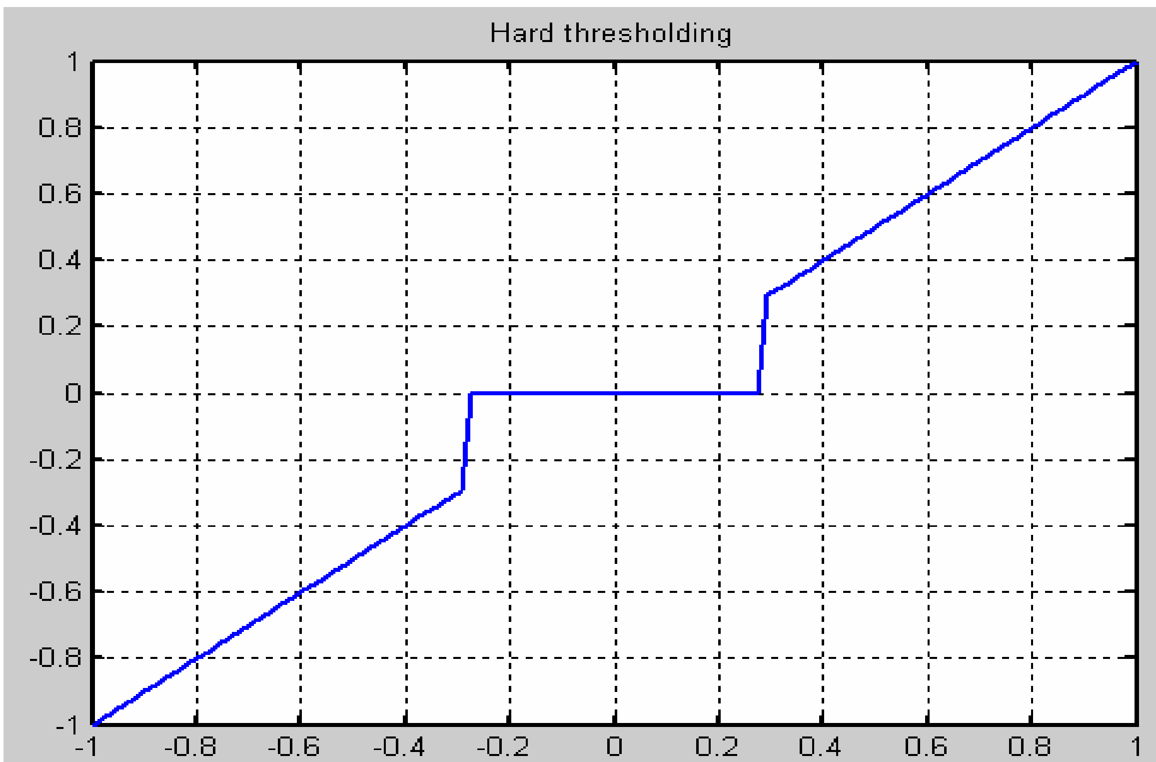
- Denoising – definition
- Denoising using wavelets vs. other methods
- Denoising process
- **Soft/Hard thresholding**
- Known thresholds
- Examples and comparison of denoising methods using WL
- Advanced applications
- 2 different simulations
- Summary

Thresholding Methods



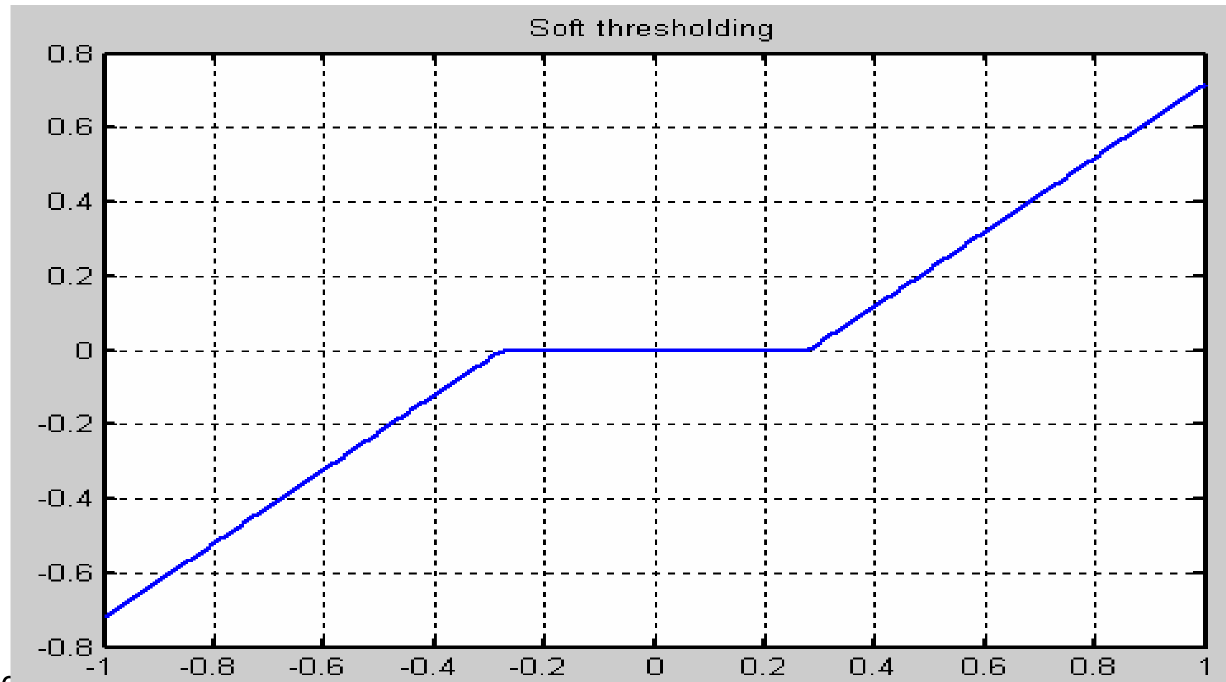
Hard Thresholding

$$y_{hard}(t) = \begin{cases} x(t), & |x(t)| > \delta \\ 0, & |x(t)| < \delta \end{cases}$$



Soft Thresholding

$$y_{soft}(t) = \begin{cases} \operatorname{sgn}(x(t)) \cdot (|x(t) - \delta|), & |x(t)| > \delta \\ 0, & |x(t)| < \delta \end{cases}$$



Soft Or Hard threshold?

- It is known that soft thresholding provides smoother results in comparison with the hard thresholding.
- More visually pleasant images, because it is continuous.
- Hard threshold, however, provides better edge preservation in comparison with the soft one.
- Sometimes it might be good to apply the soft threshold to few detail levels, and the hard to the rest.

Noisy Signal

Hard Thresholding

Edges are kept, but the noise wasn’t fully suppressed

Edges aren’t kept. However, the noise was almost fully suppressed

Soft Thresholding

In today's show

- ➲ Denoising – definition
- ➲ Denoising using wavelets vs. other methods
- ➲ Denoising process
- ➲ Soft/Hard thresholding
- ➲ **Known thresholds**
- ➲ Examples and comparison of denoising methods using WL
- ➲ Advanced applications
- ➲ 2 different simulations
- ➲ Summary

Known soft thresholds

VisuShrink (Universal Threshold)

- Donoho and Johnstone developed this method
- Provides easy, fast and automatic thresholding.
- Shrinkage of the wavelet coefficients is calculated using the formula

$$\lambda = \sigma\sqrt{2\log(n)}$$

No need to calculate λ foreach level (sub-band)!!

σ is the standard deviation of the noise of the noise level
n is the sample size.

- The rational is to remove all wavelet coefficients that are smaller than the expected maximum of an assumed i.i.d normal noise sequence of sample size *n*.
- It can be shown that if the noise is a white noise z_i i.i.d N(0,1)

- **Probablity $\{\max_i |z_i| > (2\log n)^{1/2}\} \to 0, \quad n \to \infty$**

SureShrink

A threshold level is assigned *to each resolution level* of the wavelet transform. The threshold is selected by the principle of minimizing the Stein Unbiased Estimate of Risk (SURE).

min

$$SURE(t;x) = d - 2\cdot\#\{i : |x_i| \le t\} + \sum_{i=1}^{d} (|x_i| \wedge t)^2$$

where d is the number of elements in the noisy data vector and x_i are the wavelet coefficients. This procedure is smoothness-adaptive, meaning that it is suitable for denoising a wide range of functions from those that have many jumps to those that are essentially smooth.

- If the unknown function contains jumps, the reconstruction (essentially) does also;
  if the unknown function has a smooth piece, the reconstruction is (essentially) as smooth as the mother wavelet will allow.
- The procedure is in a sense optimally smoothness-adaptive: it is near-minimax simultaneously over a whole interval of the Besov scale; the size of this interval depends on the choice of mother wavelet.

Estimating the Noise Level

- In the threshold selection methods it may be necessary to estimate the standard deviation σ of the noise from the wavelet coefficients. A common estimator is shown below:

$$\sigma = \frac{MAD}{0.6745}$$

where *MAD* is the median of the absolute values of the wavelet coefficients.

In today's show

- Denoising – definition
- Denoising using wavelets vs. other methods
- Denoising process
- Soft/Hard thresholding
- Known thresholds
- **Examples and comparison of denoising methods using WL**
- Advanced applications
- 2 different simulations
- Summary

Example

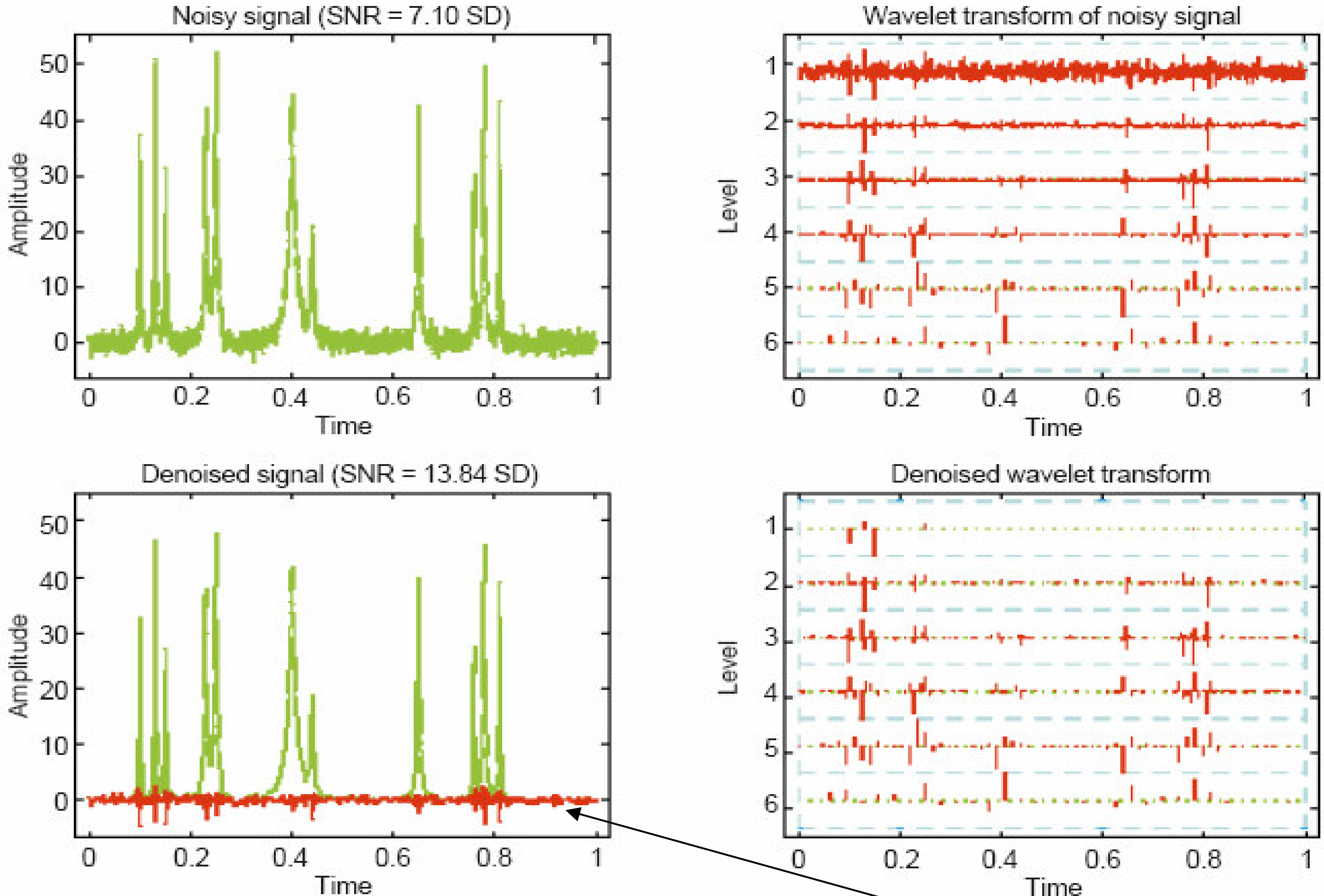
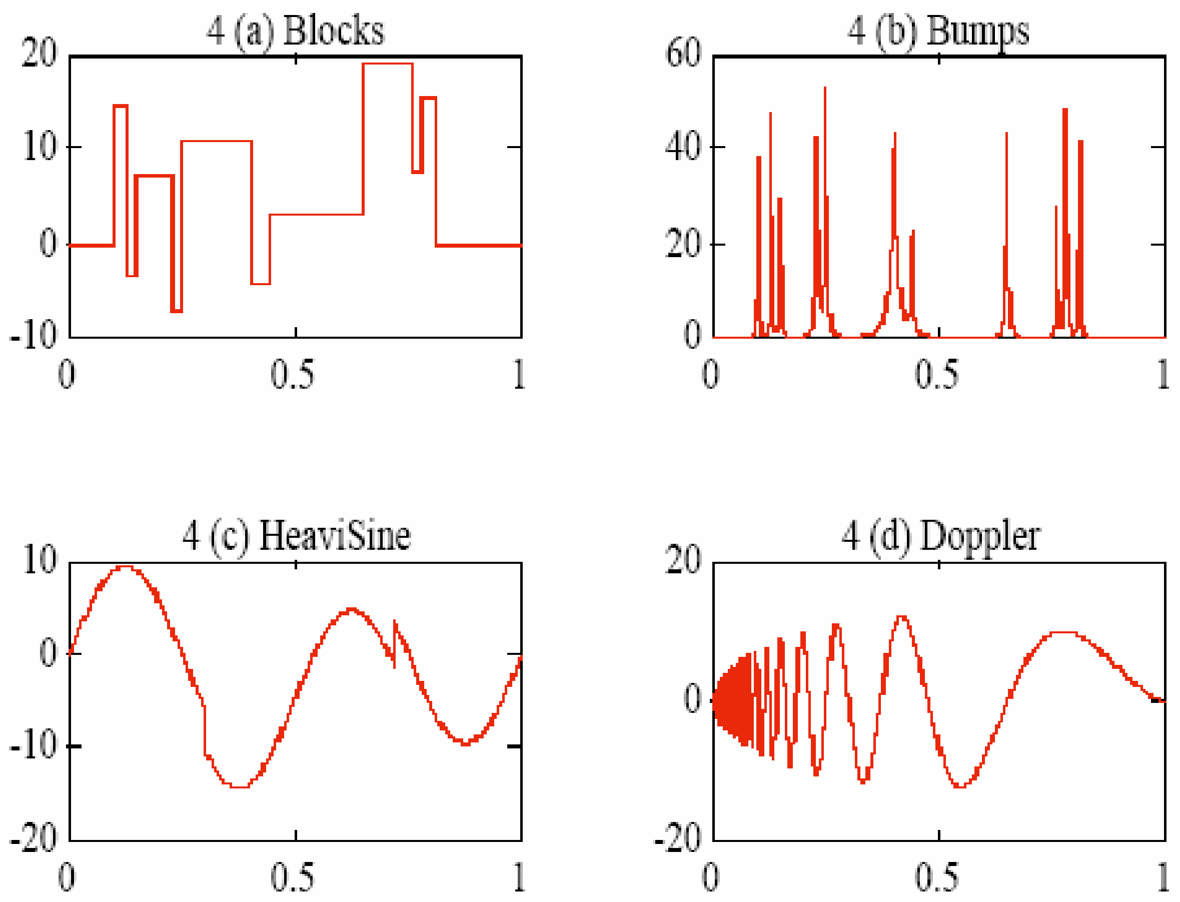


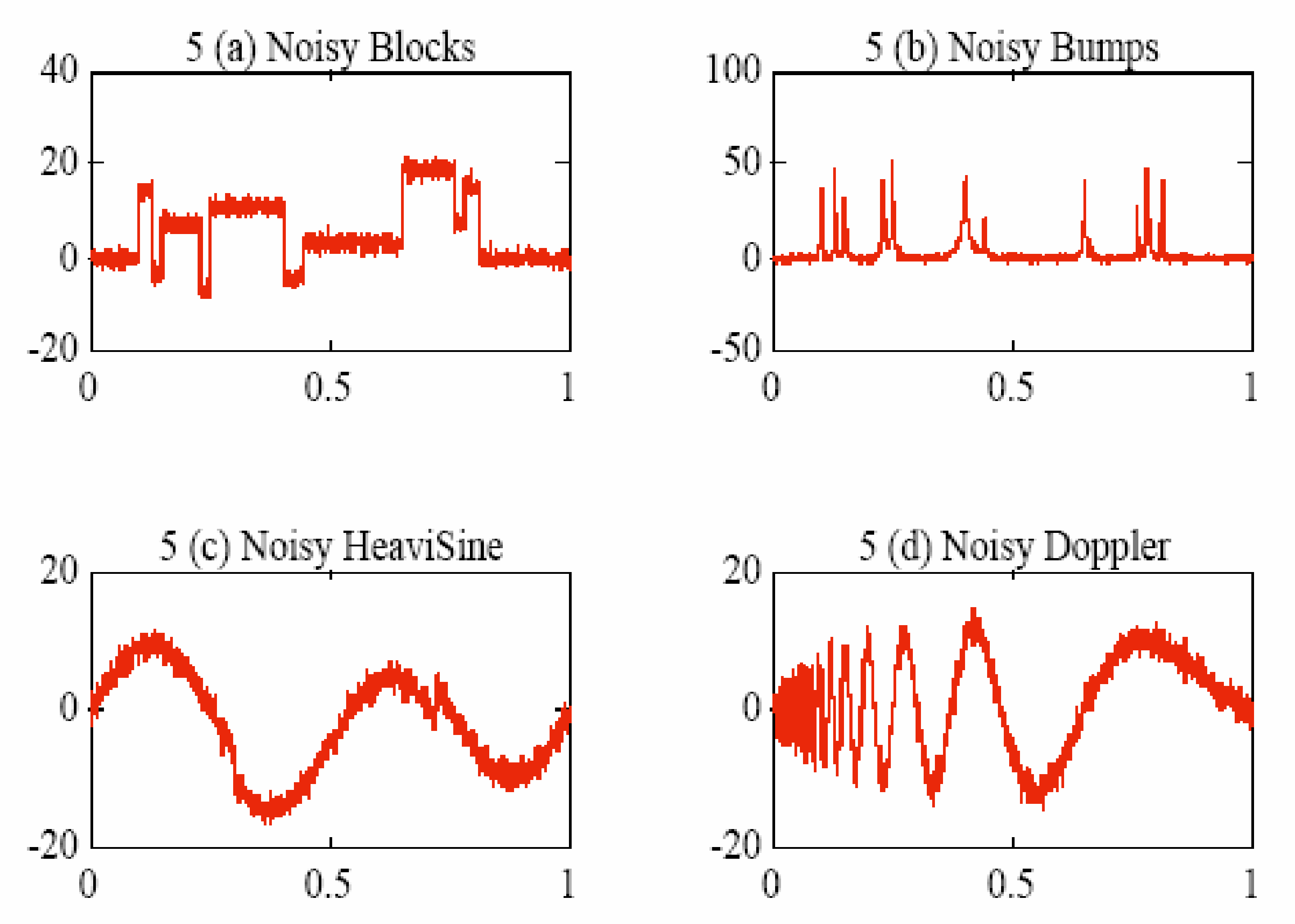
Figure 1. Wavelet shrinkage de-noising with 'SUR,' DROLA (16; 8), n = 2,048, L = 6.

Difference!!

More examples



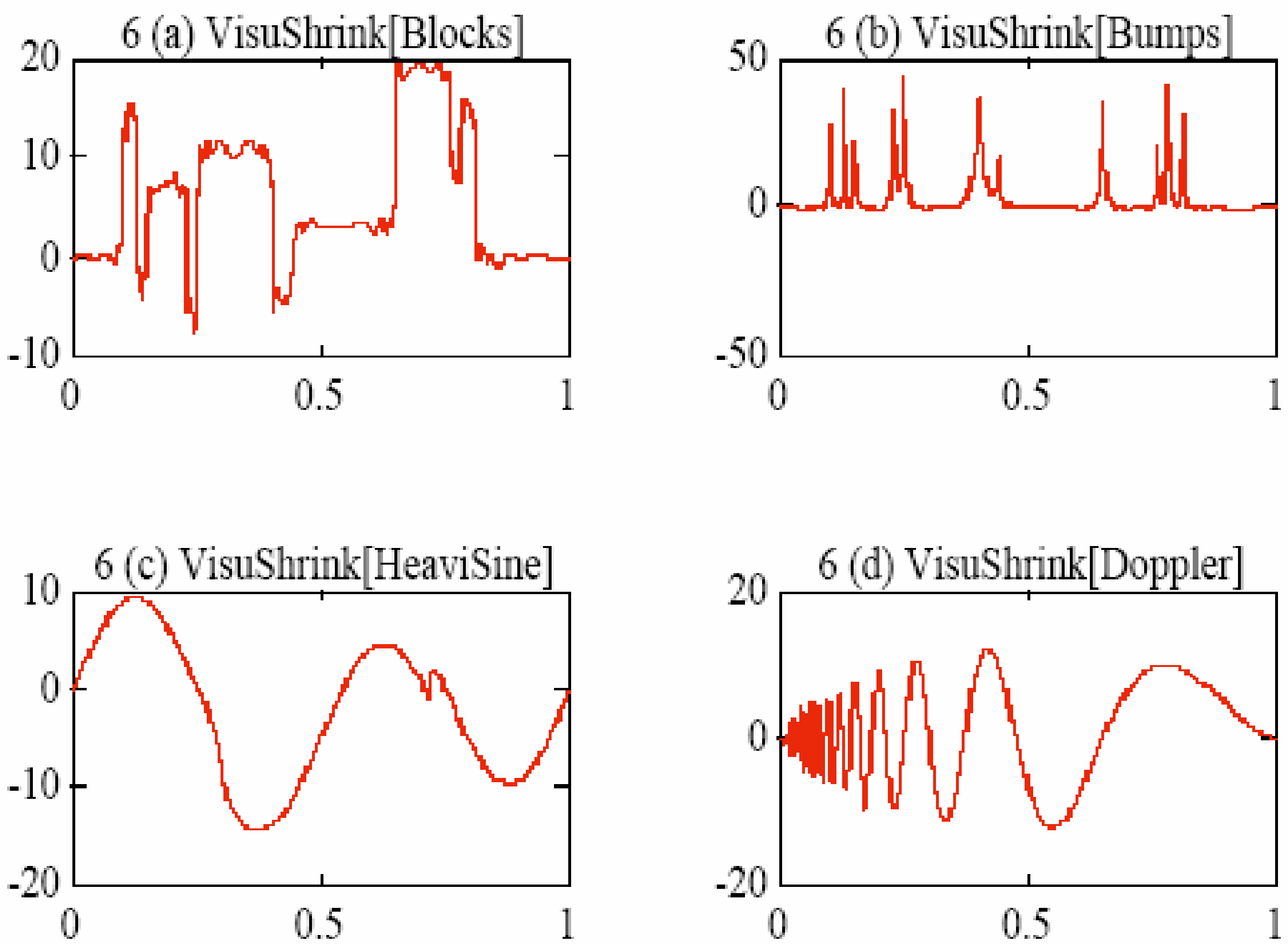
Original signals



Noisy signals

$$n = 2^{J+1} \text{ data } y_i = f(t_i) + \sigma z_i, \ i = 1, \ldots, n$$

N = 2048 = 2^{11}



Denoised signals

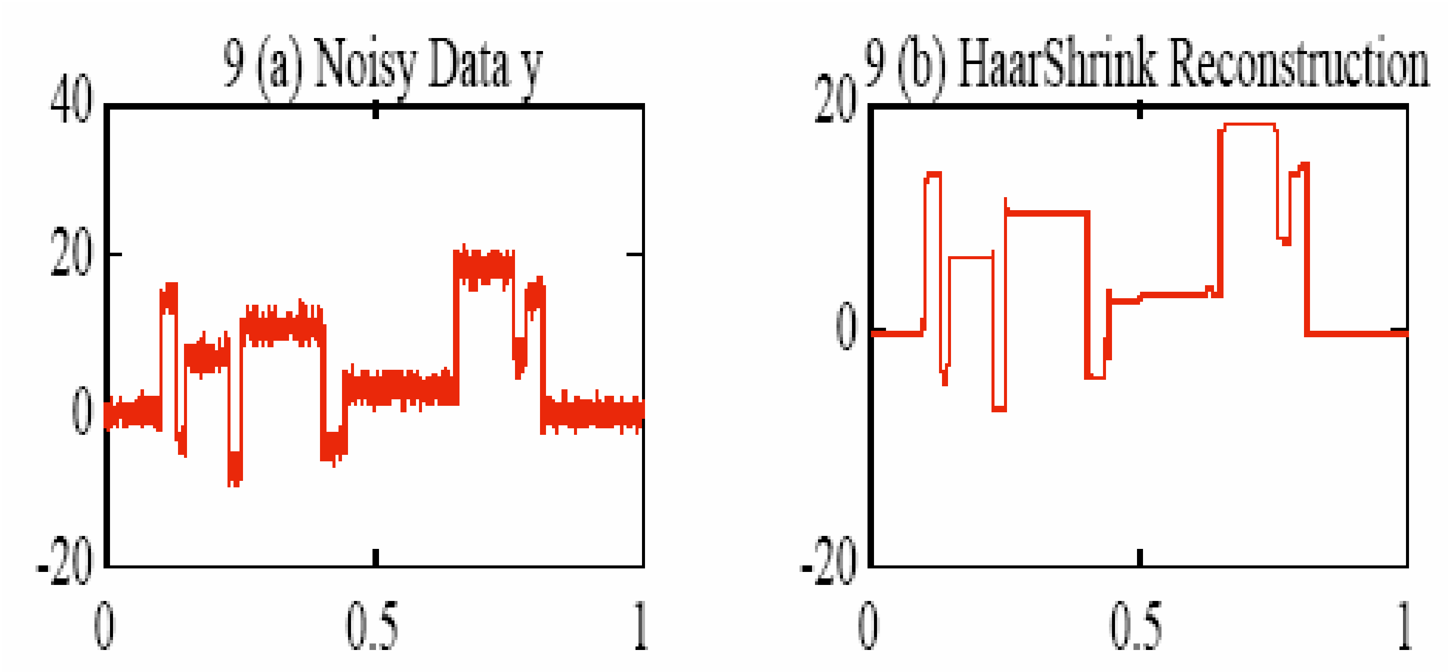
Soft threshold

$$t = \sqrt{2\log(n)}\sigma/\sqrt{n}$$

The reconstructions have two properties:

1. The noise has been almost entirely suppressed
2. Features sharp in the original remain sharp in reconstruction

- Here we use Haar-basis shrinkage method



- The Haar transform of the noiseless object *Blocks* compresses the l^2 energy of the signal into a very small number of consequently) very large coefficients.
- On the other hand, Gaussian white noise in any one orthogonal basis is again a white noise in any other.
- → In the Haar basis, the few nonzero signal coefficients really stick up above the noise
- →the thresholding kills the noise while not killing the signal

Formal:

- Data: $d_i = \theta_i + \varepsilon z_i$, $i=1,\ldots,n$
- z_i standard white noise
- Goal : recovering θ_i
- Ideal diagonal projector : keep all coefficients where θ_i is larger in amplitude than ε and 'kill' the rest.
- The ideal is unattainable since it requires knowledge on θ which we don't know

The ideal mean square error is

$$R(\hat{\theta}^{IDEAL}, \theta) = \sum_i \min(\theta_i^2, \epsilon^2).$$

Define the "compression number" c_n as follows.

With $|\theta|_{(k)}$ = k-th largest amplitude in vector θi set

$$c_n \equiv \sum_{k>n} |\theta|^2_{(k)}$$

This is a measure of how well the vector θi can approximated by a vector with n nonzero entries.

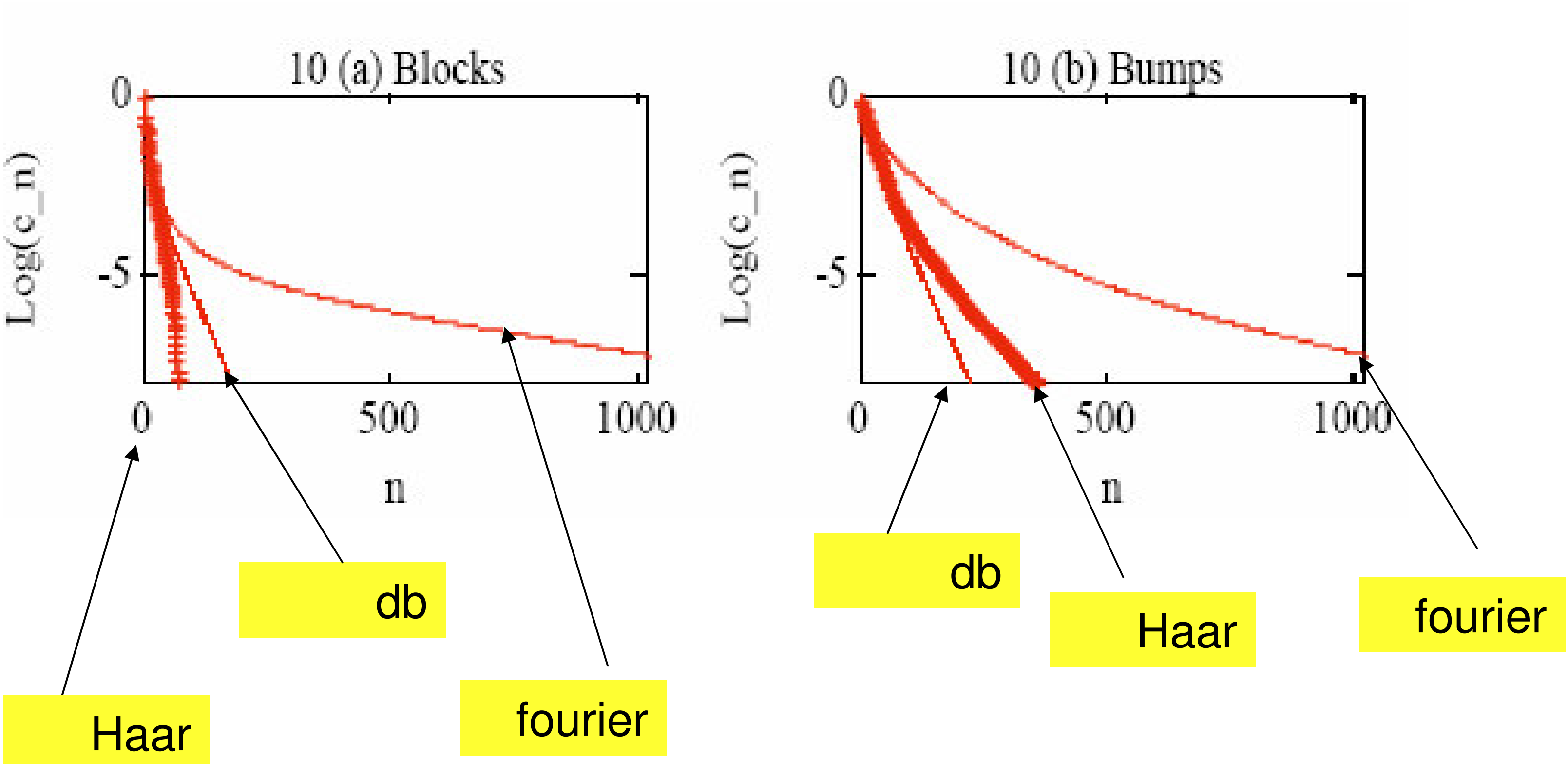
Setting $$N(\epsilon) = \#\{i : |\theta_i| \geq \epsilon\}$$

$$\sum_i \min(\theta_i^2, \epsilon^2) = \epsilon^2 \cdot \#\{i : |\theta_i| \geq \epsilon\}$$

$$+ \quad \sum_i \theta_i^2 1_{\{i:|\theta_i| \leq \epsilon\}} = \epsilon^2 \cdot N(\epsilon) + c_{N(\epsilon)},$$

so this ideal risk is explicitly a measure of the extent to which the energy is compressed into a few big coefficients.

➲ We will see the extend to which the different orthogonal basses compress the objects



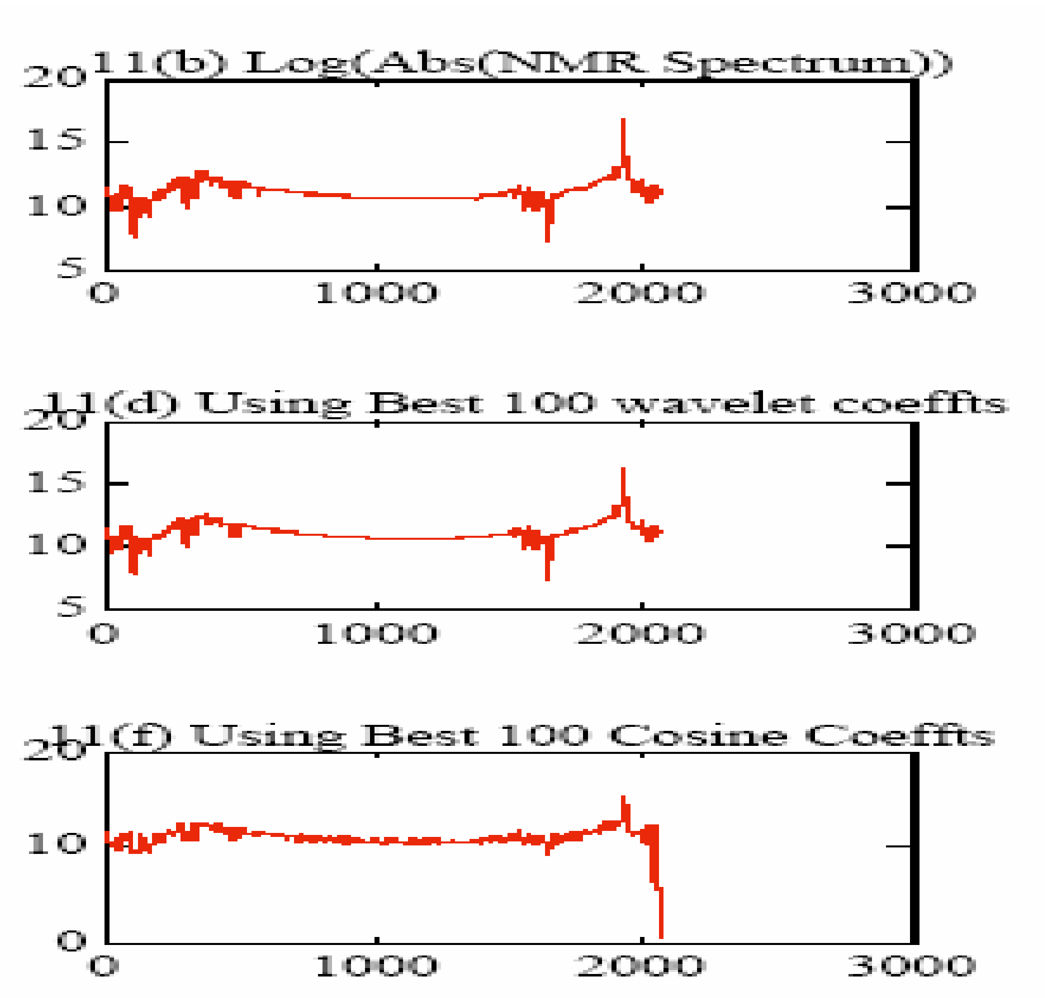
Another aspect - Vanishing Moments

- The mth moment of a wavelet is defined as $\int t^m \psi(t) dt$
- If the first *M* moments of a wavelet are zero, then all polynomial type signals of the form $x(t) = \sum_{0<m<M} c_m t^m$

  have (near) zero wavelet / detail coefficients.
- Why is this important? Because if we use a wavelet with enough number of vanishing moments, *M*, to analyze a polynomial with a degree less than *M*, then all detail coefficients will be zero ➔ excellent compression ratio.
- All signals can be written as a polynomial when expanded into its Taylor series.
- This is what makes wavelets so successful in compression!!!

Why it works?(II)
Unconditional basis

- A very special feature of wavelet bases is that they serve as unconditional bases, not just of L^2, but of a wide range of smoothness spaces, including Sobolev and Hölder classes.
- As a consequence, “shrinking" the coefficients of an object towards zero, as with soft thresholding, acts as a “smoothing operation" in any of a wide range of smoothness measures.
- Fourier basis isn’t such basis



Original singal

Denoising using the 100 biggest **WL** coefficients

Denoising using the 100 biggest **Fourier** coefficients

Worst MSE+ visual artifacts!!

In today's show

- Denoising – definition
- Denoising using wavelets vs. other methods
- Denoising process
- Soft/Hard thresholding
- Known thresholds
- Examples and comparison of denoising methods using WL
- **Advanced applications**
- 2 different simulations
- Summary

Advanced applications

Discrete inverse problems

Assume : $\boldsymbol{y_i = (Kf)(t_i) + \varepsilon z_i}$

- *Kf* is a transformation of *f* (Fourier transformation, laplace transformation or convolution)
- Goal : reconstruct the singal t_i
- Such problems become problems of recovering wavelets coefficients in the presence of ***non-white noise***

Example :

we want to reconstruct the discrete signal $(x_i)_{i=0..n-1}$, given the noisy data :

$$d_i = \left(\sum_{t=0}^{i} x_t\right) + \sigma z_i, \ldots i = 1, \ldots, n.$$

White gaussian noise

We may attempt to invert this relation, forming the differences :

$y_i = d_i - d_{i-1}, \quad y_0 = d_0$

This is equivalent to observing

$y_i = x_i + \sigma(z_i - z_{i-1})$ ***(non white noise)***

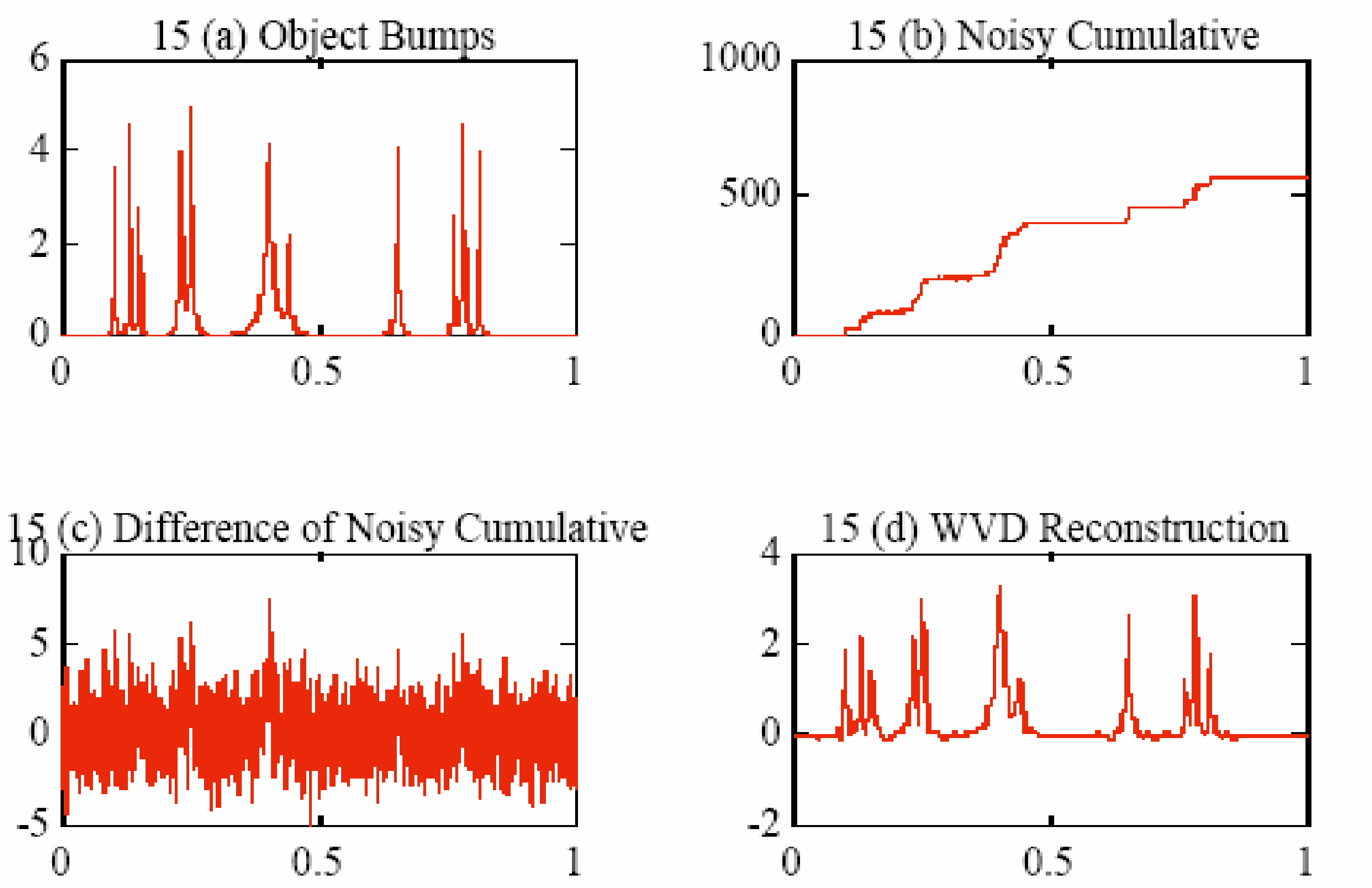
- Solution : reconstructing x_i in three-step process, with level-dependent threshold.

$$t_{j,n} = \sqrt{2\log(n)} \cdot (2\sigma)/\sqrt{n} \cdot 2^{(J-j)/2}, \qquad j = j_0, \ldots, J;$$

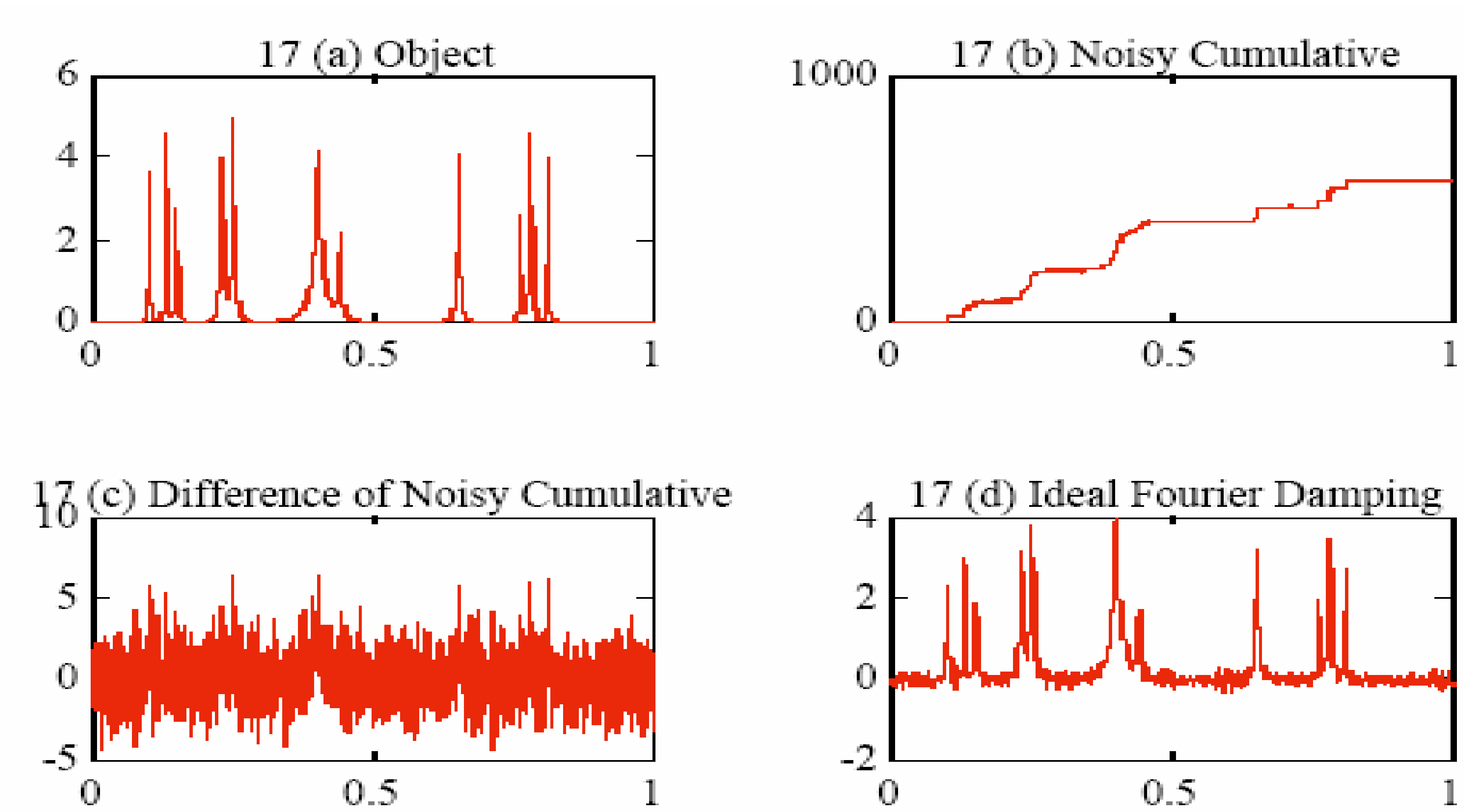
The threshold is much larger at high resolution levels than at low ones (j_0 is the coarse level. J is the finest)

Motivation : the variance of the noise in level j grows roughly like 2^j

The noise is heavily damped, while the main structure of the object persists



WL denoising method supresses the noise!!



Fourier is unable to supress the noise!!

In today's show

- Denoising – definition
- Denoising using wavelets vs. other methods
- Denoising process
- Soft/Hard thresholding
- Known thresholds
- Examples and comparison of denoising methods using WL
- Advanced applications
- **2 different simulations**
- Summary

Monte Carlo simulation

- The Monte Carlo method (or simulation) is a statistical method for finding out the answer to a problem that is too difficult to solve analytically, or for verifying the analytical solution.
- It Randomly generates values for uncertain variables over and over to simulate a model
- It is called Monte Carlo because of the gambling casinos in that city, and because the Monte Carlo method is related to rolling dice.

- We will describe a variety of wavelet and wavelet packet based denoising methods and compare them with each other by applying them to a simulated, noised signal
- f is a known signal. The noise is a free parameter
- The results help us choose the best wavelet, best denoising method and a suitable denoising threshold in praticical applications.

- A noised singal f_i $i=0,\ldots,2^{jmax-1}$
- Wavelet

$$\tilde{f} = \tilde{c}_0^0 \cdot \phi_0^0 + \sum_{j=0}^{jmax-1} \sum_{i=0}^{2^j-1} \tilde{d}_i^j \cdot \psi_i^j$$

- Wavelet pkt

$$\tilde{f} = \sum_{B_{jfi}} \tilde{wc}_{jfi} \, w_{jfi}$$

Denoising methods

- Linear – Independent on the size of the signal coefficients. Therefore the coefficient size isn't taken into account, but the scale of the coefficient. It is based on the assumption that signal noise can be found mainly in fine scale coefficients and not in coarse ones. Therefore we will cut off all coefficients with a scale finer that a certain scale threshold S_0.

WL

$$\hat{d}_i^j = \begin{cases} 0 & , j \geq S_0 \\ \tilde{d}_i^j & , j < S_0 \end{cases}$$

In packet wavelets, fine scaled signal structures can be represented not only by fine scale coefficients but also by coarse scale coefficients with high frequency. Therefore, it is necessary to eliminate not only fine scale coefficients through linear denoising, but also coefficients of a scale and frequency combination which refer to a certain fine scale structure.

$$\hat{w}c_{jfi} = \begin{cases} 0 & , j \geq S_0 \\ 0 & , (j < S_0) \wedge (f > 2^{S_0 - j}) \\ \tilde{w}c_{jfi} & , else \end{cases}$$

PL

Non linear – cutting of the coefficients (hard or soft), threshold = λ

$$\hat{w}c_{jfl} = \begin{cases} 0 & , \left|\tilde{w}c_{jfl}\right| < \lambda \\ \tilde{w}c_{jfl} & , else \end{cases} \qquad hard(PNLH)$$

$$\hat{w}c_{jfl} = \begin{cases} 0 & , \left|\tilde{w}c_{jfl}\right| < \lambda \\ sgn(\tilde{w}c_{jfl}) \cdot \left(\left|\tilde{w}c_{jfl}\right| - \lambda\right) & , else \end{cases} \qquad soft(PNLS)$$

Measuring denoising errors

Lp norms (p=1,2) :

$$l_p - norm: \quad \|f - \hat{f}\|_p = (\sum_i |f(t_i) - \hat{f}(t_i)|^p)^{\frac{1}{p}} \ , \ p = 1,2$$

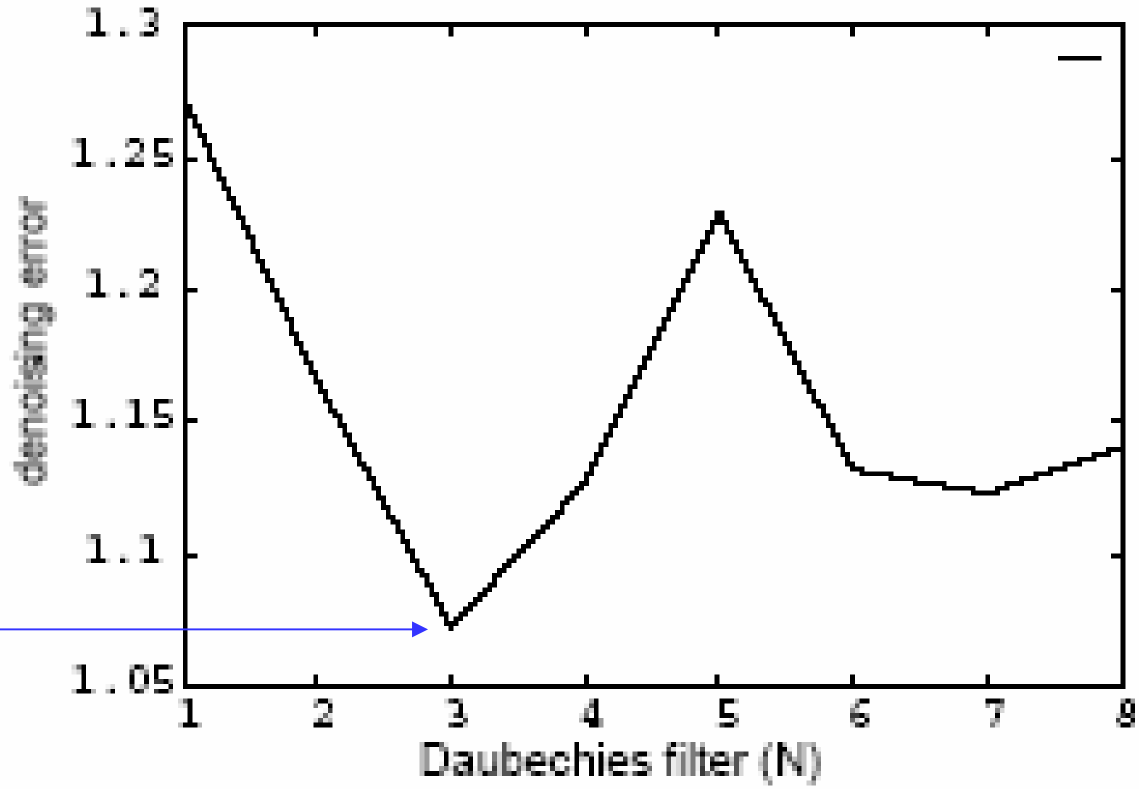
Entropy -

$$entropy - norm: \quad -\sum_i \frac{|f(t_i) - \hat{f}(t_i)|^2}{\|f - \hat{f}\|_2^2} \cdot \log \frac{|f(t_i) - \hat{f}(t_i)|^2}{\|f - \hat{f}\|_2^2}$$

Choosing the best threshold and basis

➲ Using Monte Carlo simulation DB with 3 vanishing moments has been chosen for PNLS method.

Min Error

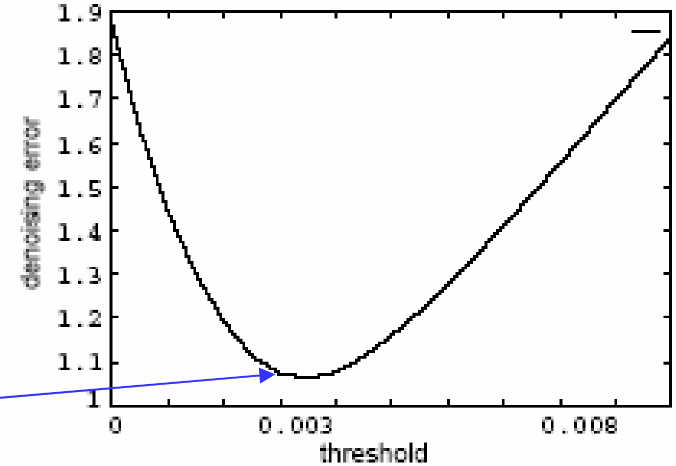


Threshold – universal soft threshol $\lambda_u = \sqrt{2log(2^{jmax})}\hat{\sigma}$

For normally distibuted noise, $\lambda_u = 0.008$

However, it seems that λ_u lies above the optimal threshold.

Using monte carlo to evaluate the 'best' threshold for PNLS, 0.003 is the best



Min error

- For each method a best basis and an optimal threshold is collected using Monte Carlo simulations.
- Now we are ready to compare!
- The comparison reveals that WNLH has the best denoising performance.
- We would expect wavelet packets method to have the best performance. It seems that for this specific signal, even with Donoho best cost function, this method isn’t the optimal.

Best!!

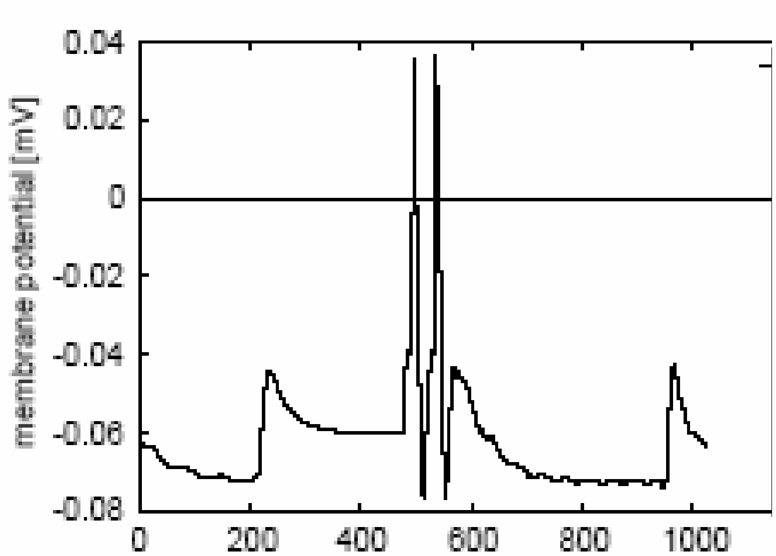
DJ WP close to the Best!!

Type of Noise	N(0,0.00225)			U[-0.004,0.004]		
Method Daub. N=3	*threshold* $\hat{E}\| \|_1$ *std.dev.*	*threshold* $\hat{E}\| \|_2$ *std.dev.*	*threshold* $\hat{E}(Entr.)$ *std.dev.*	*threshold* $\hat{E}\| \|_1$ *std.dev.*	*threshold* $\hat{E}\| \|_2$ *std.dev.*	*threshold* $\hat{E}(Entr.)$ *std.dev.*
WL	8	8	8	8	8	8
	1.02776	0.05027	0.39545	1.07019	0.05115	0.40694
	0.04472	0.00135	0.01378	0.04008	0.00104	0.01171
WNLH	0.007	0.007	0.007	0.006	0.006	0.006
	0.73024	0.03369	0.30076	0.71280	0.03229	0.29159
	0.04978	0.00194	0.01597	0.05099	0.00214	0.01798
WNLS	0.003	0.003	0.003	0.003	0.003	0.003
	0.87034	0.03839	0.34352	0.89008	0.03857	0.34697
	0.04943	0.00187	0.01547	0.04419	0.00095	0.01138
PL	8	8	8	8	8	8
	1.14461	0.05406	0.45877	1.19165	0.05626	0.48823
	0.08135	0.00287	0.04322	0.07182	0.00271	0.03880
PNLH	0.00419	0.00390	0.00396	0.00416	0.00387	0.00409
	0.88898	0.04138	0.35604	1.04809	0.04740	0.40225
	0.06849	0.00329	0.02257	0.56825	0.01513	0.13982
PNLS	0.00311	0.00300	0.00300	0.00362	0.00300	0.00312
	0.98594	0.04260	0.37870	0.99727	0.04364	0.38481
	0.07954	0.00371	0.02737	0.05407	0.00248	0.01855
DJL	8	8	8	7	8	8
	1.61946	0.06612	0.57474	2.02705	0.07428	0.67049
	0.23498	0.00637	0.06918	0.37177	0.01042	0.09041
DJNLH	0.00154	0.00142	0.00150	0.00160	0.00148	0.00156
	0.87264	0.03957	0.34804	0.89884	0.04166	0.36053
	0.07421	0.00343	0.02504	0.07024	0.00296	0.02181
DJNLS	0.00300	0.00290	0.00296	0.00346	0.00296	0.00300
	0.98796	0.04269	0.37982	1.13178	0.04730	0.42033
	0.06553	0.00258	0.02064	0.46297	0.01403	0.12289

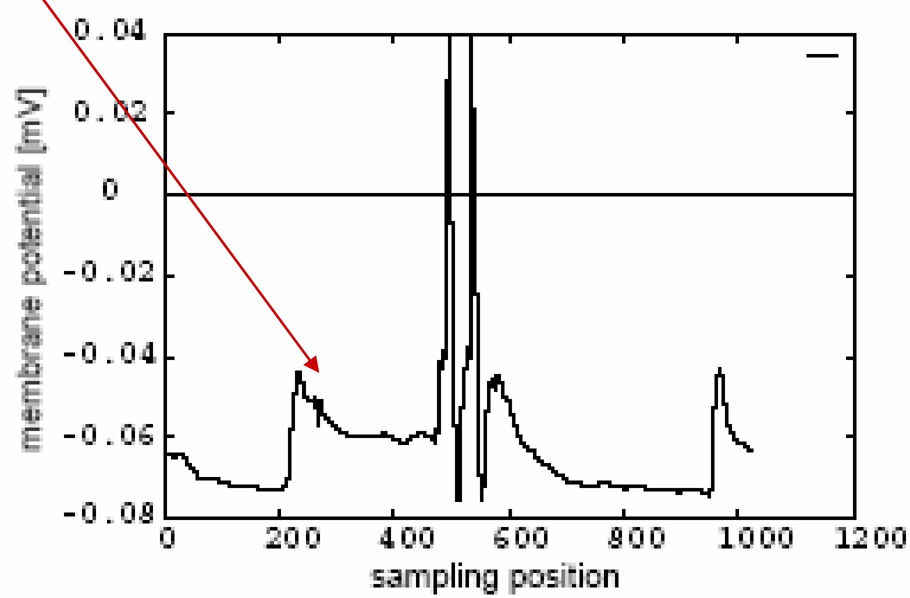
Improvements

- Even with the minimal denoising error, there are small artifacts.

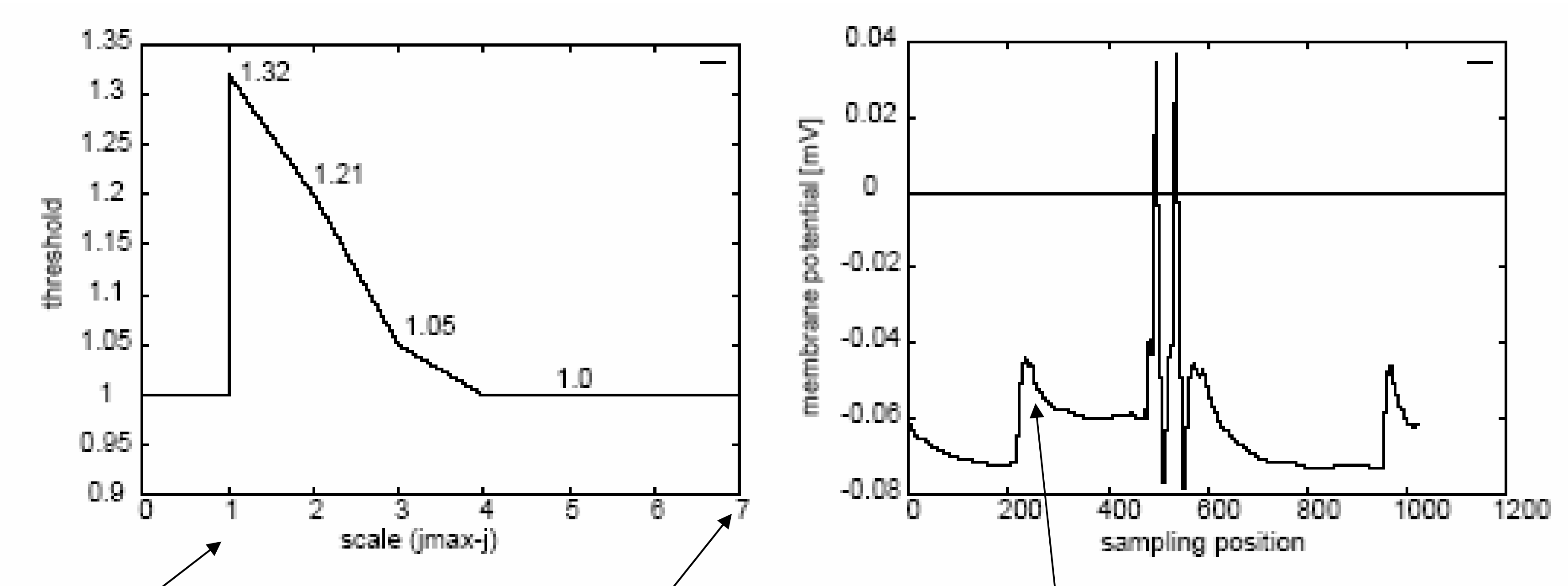
Original



Denoised



- **Solution** : the artifacts live only on fine scales, we can adapt λ to the scale *j* $\quad \lambda_j = \lambda * \mu_j$



Finest scale

Most coarse scale

Artifacts have disappeared!

Thresholds experiment

- In this experiment, 6 known signals were taken at n=1024 samples.
- Additive white noise (SNR = 10dB)
- The aim – to compare all thresholds performance in comparison to the ideal thresholds.
- RIS, VIS – global threshold which depends on n.
- SUR – set for each level
- WFS, FFS – James thresholds (WL, Fourier)
- IFD, IWD – ideal threshold (if we knew noise level)

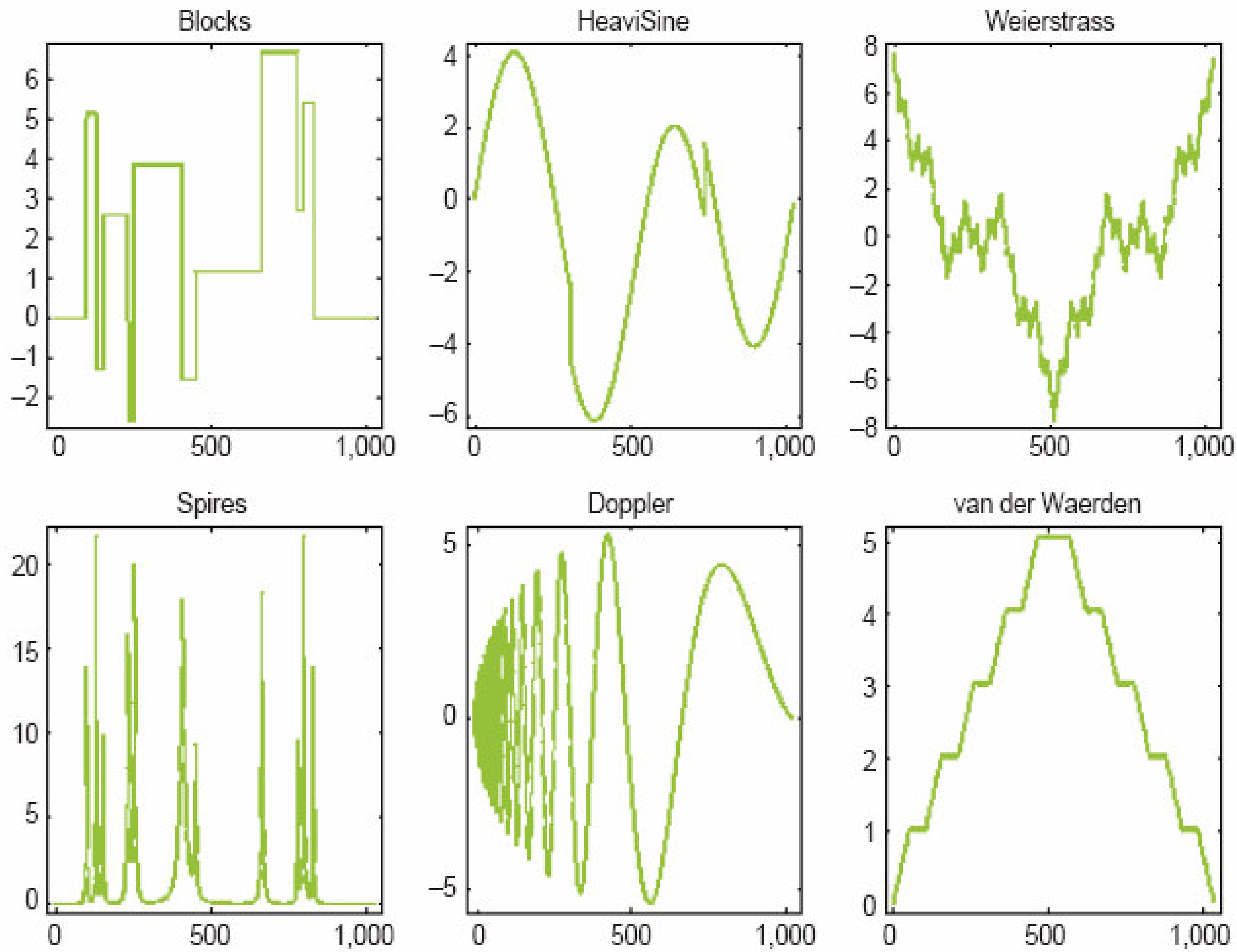
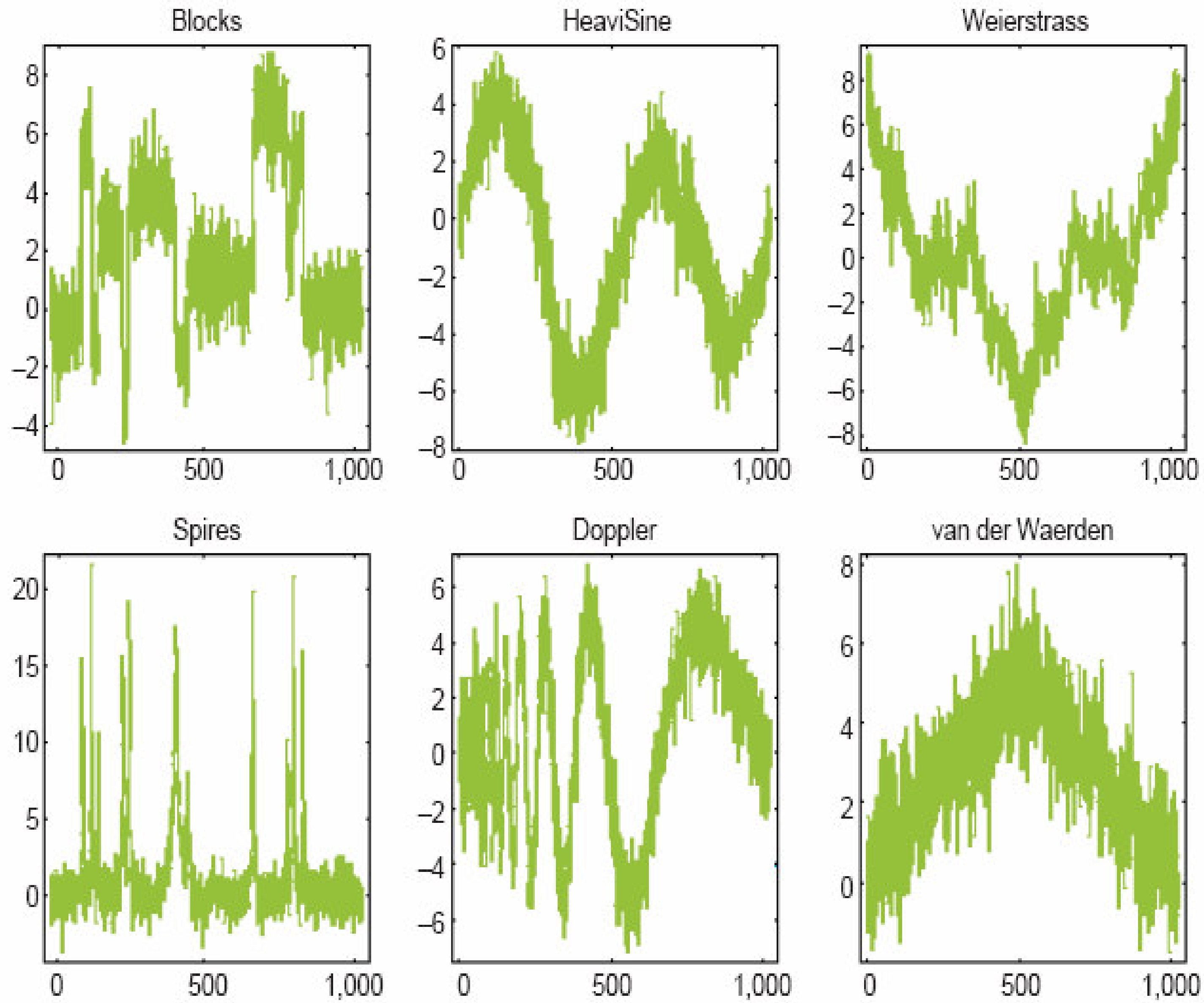


Figure 2. Standardized test signals with n = 1,024.

original

Figure 3. Noisy test signals with $n = 1{,}024$, SNR = 10.



Noisy signals

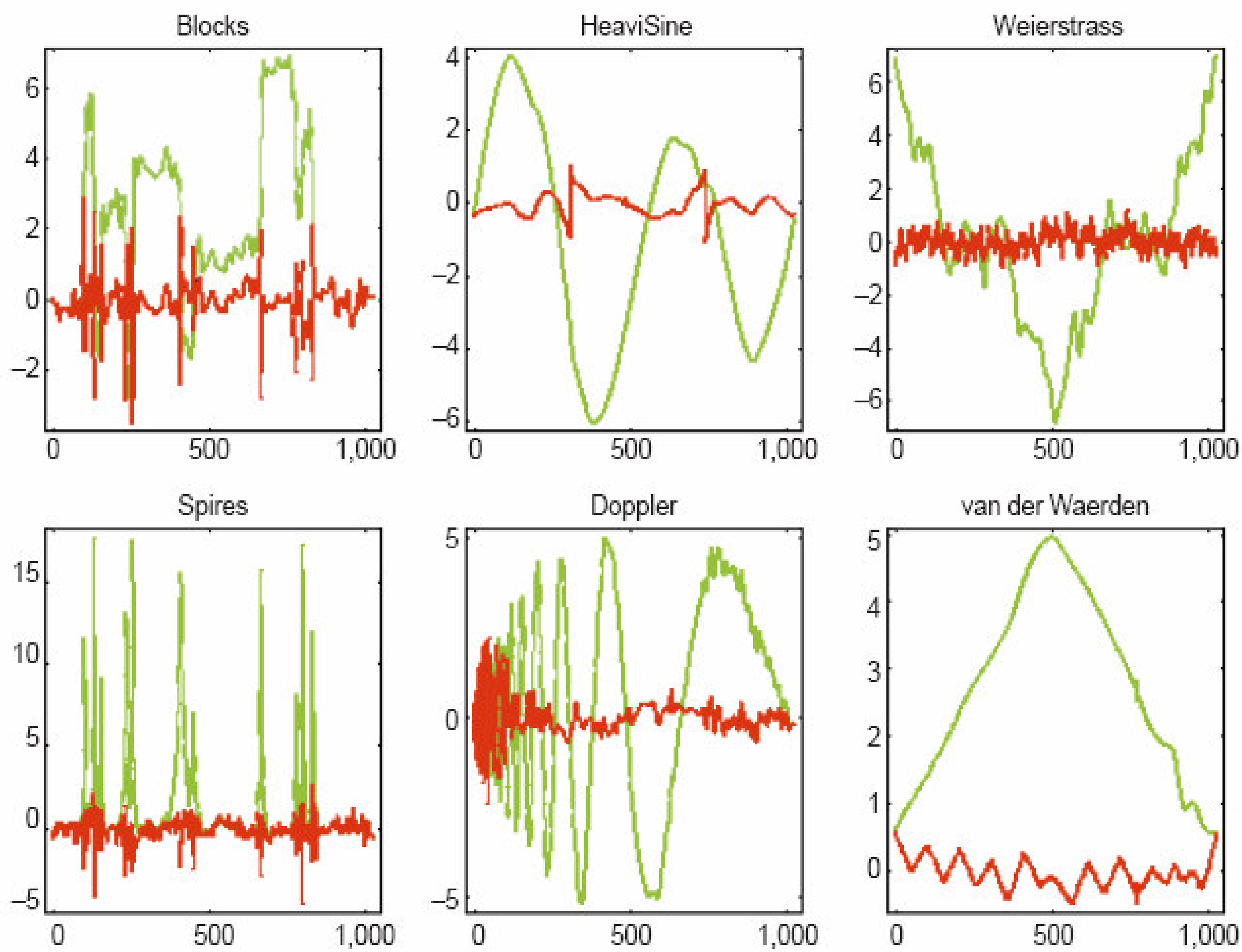


Figure 4. Wavelet shrinkage denoising with 'SUR,' DROLA (10; 5), $n = 2{,}048$, $L = 5$.

Denoised signals

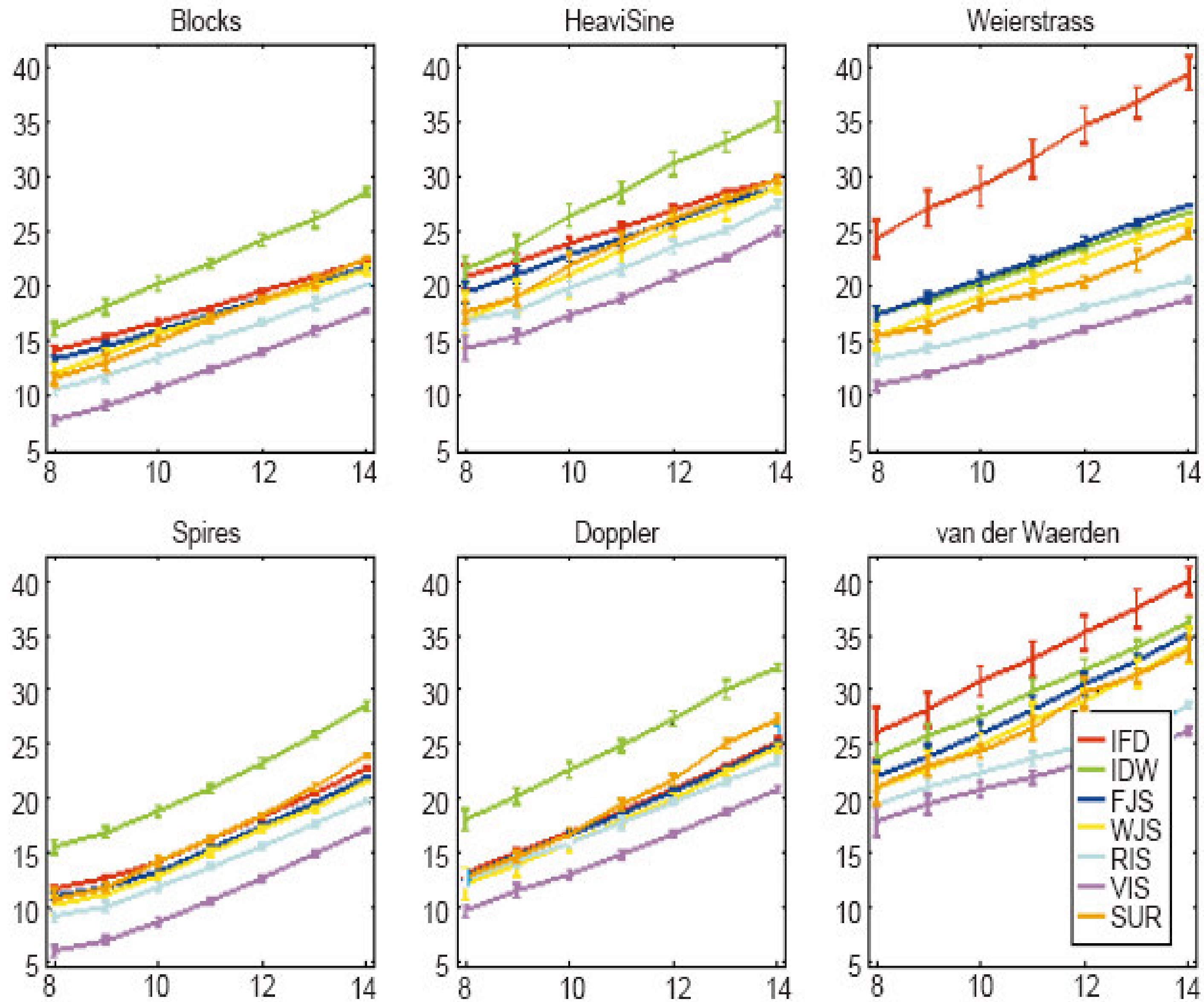


Figure 5. Monte Carlo experiment comparing various denoising methods. Each curve plots SNR versus *L*.

Results

- Surprising, isn't it?
- VIS is the worst for all the signals.
- Fourier is better?
- What about the theoretical claims of optimality and generality?
- We use SNR to measure error rates
- Maybe should it be judged visually by the human eye and mind?

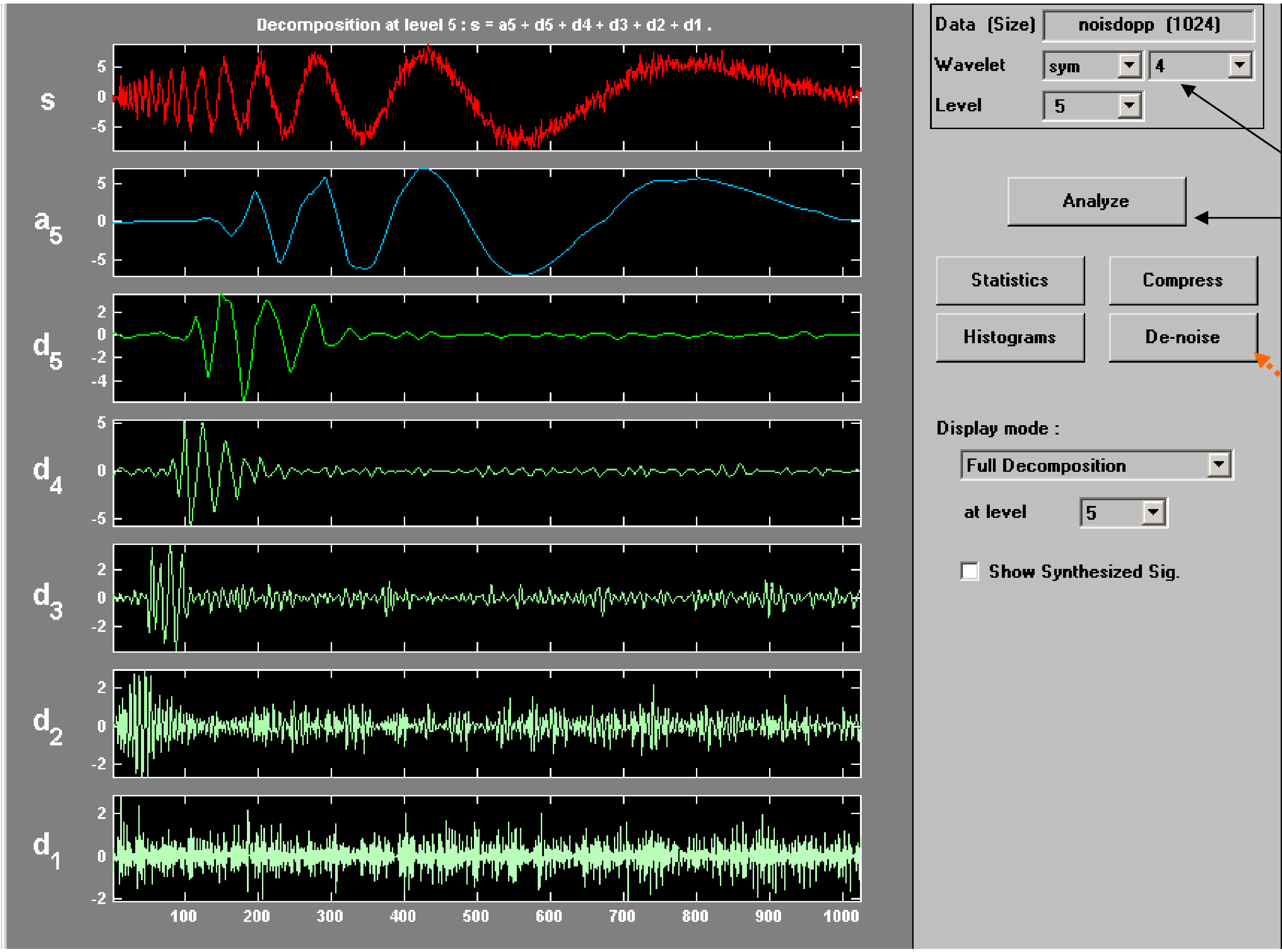
➲ [DJ] In this case, VIS performs best.

Original

Noisy (SNR = 10 dB)

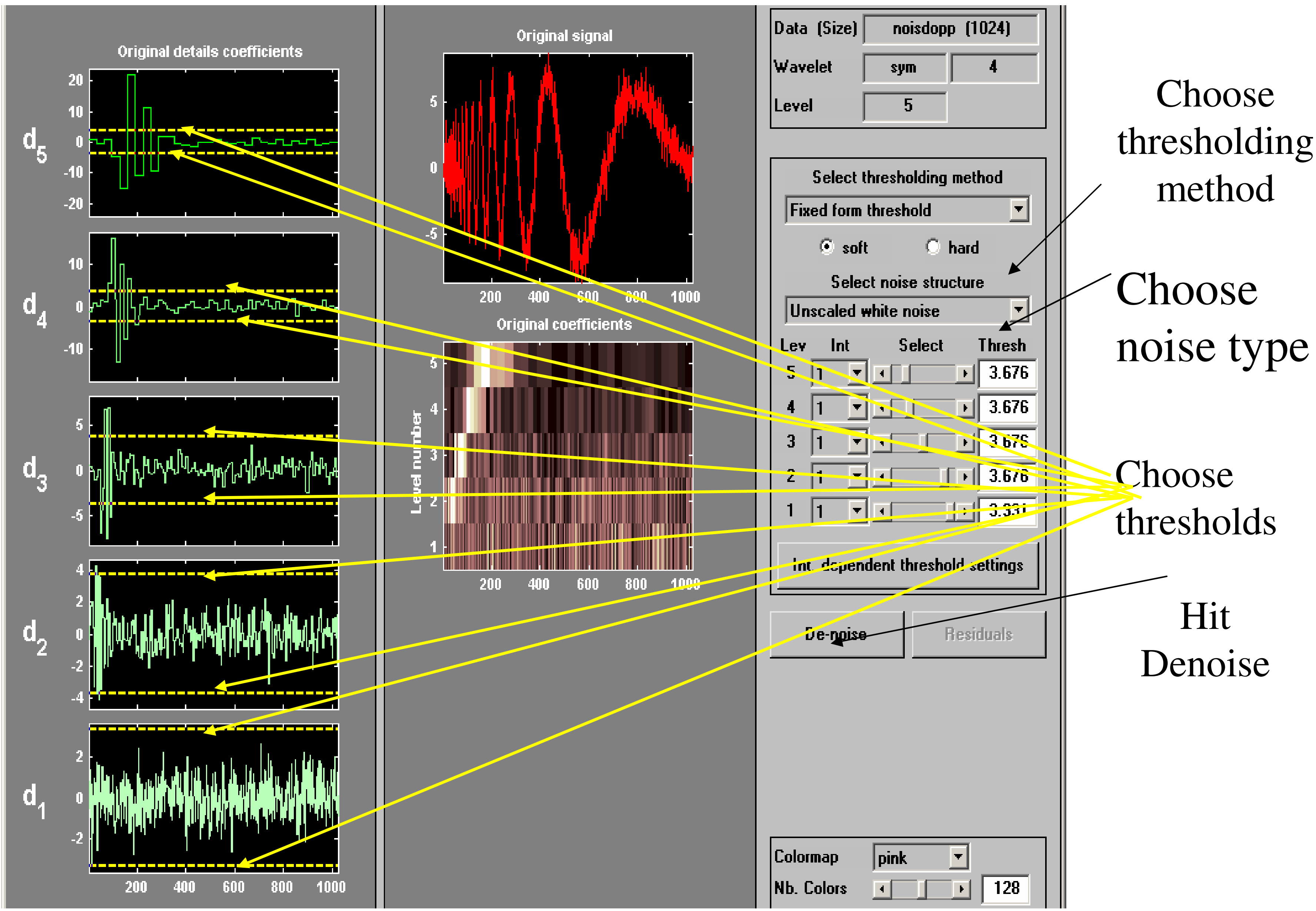
Denoised (SNR = 18 dB)

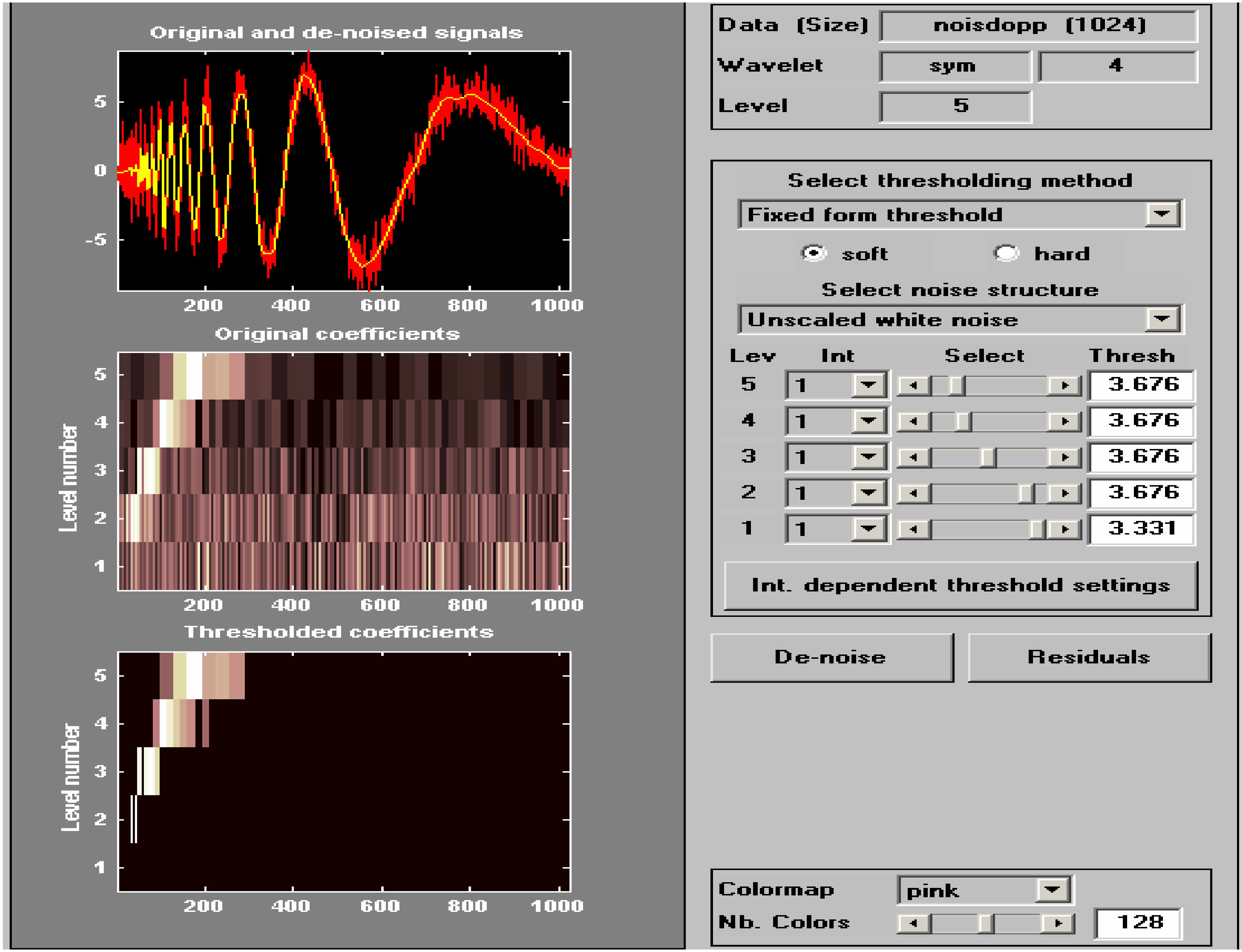
Denoising Implementation in Matlab



First, analyze the signal with appropriate wavelets

Hit Denoise


Original details coefficients
d5
d4
d3
d2
d1
Original signal
Original coefficients
Level number
Data (Size)
noisdopp (1024)
Wavelet
sym
4
Level
5
Select thresholding method
Fixed form threshold
soft
hard
Select noise structure
Unscaled white noise
Lev
Int
Select
Thresh
3.676
Int. dependent threshold settings
De-noise
Residuals
Colormap
pink
Nb. Colors
128
Choose thresholding method
Choose noise type
Choose thresholds
Hit Denoise



Original and de-noised signals
Original coefficients
Thresholded coefficients
Level number
Data (Size)
noisdopp (1024)
Wavelet
sym
4
Level
5
Select thresholding method
Fixed form threshold
soft
hard
Select noise structure
Unscaled white noise
Lev
Int
Select
Thresh
3.676
3.331
Int. dependent threshold settings
De-noise
Residuals
Colormap
pink
Nb. Colors
128

In today's show

- Denoising – definition
- Denoising using wavelets vs. other methods
- Denoising process
- Soft/Hard thresholding
- Known thresholds
- Examples and comparison of denoising methods using WL
- Advanced applications
- 2 different simulations
- **Summary**

Summary

- We learn how to use wavelets for denoising
- We saw different denoising methods and their results
- We saw other uses of wavelets denoising to solve discrete problems
- We saw experiments and results

Thank you!

Bibliography

- Nonlinear Wavelet Methods for Recovering Signals, Images, and Densities from indirect and noisy data [D94]
- Filtering (Denoising) in the Wavelet Transform Domain *Yousef M. Hawwar, Ali M. Reza, Robert D. Turney*
- Comparison and Assessment of Various Wavelet and Wavelet Packet based Denoising Algorithms for Noisy Data F. Hess, M. Kraft, M. Richter, H. Bockhorn
- De-Noising via Soft-Thresholding, Tech. Rept., Statistics, Stanford, 1992.
- Adapting to unknown smoothness by wavelet shrinkage, Tech. Rept., Statistics, Stanford, 1992. D. L. Donoho and I. M. Johnstone
- Denoising by wavelet transform [Junhui Qian]
- Filtering denoising in the WL transform domain[Hawwr,Reza,Turney]
- The What,how,and why of wavelet shrinkage denoising[Carl Taswell, 2000]