An Animated Introduction to the Discrete Wavelet Transform

Revised Lecture Notes

New Delhi
December 2001

Arne Jensen

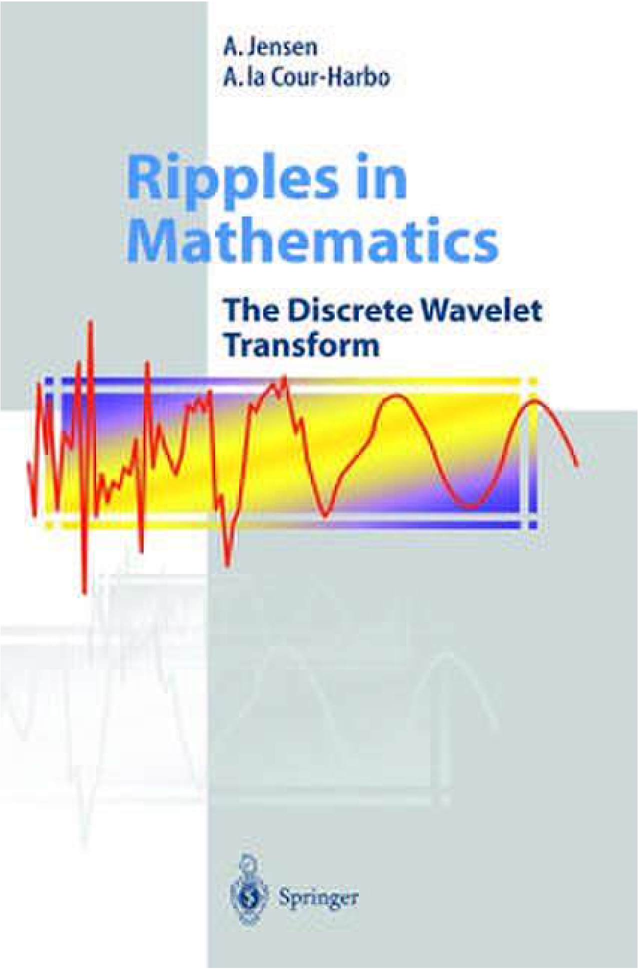
Aalborg University

Reference

This is a tutorial introduction to the discrete wavelet transform. It is based on the book

> A. Jensen and A. la Cour-Harbo:
> *Ripples in Mathematics*
> *The Discrete Wavelet Transform*
> Springer-Verlag 2001.

Book cover


A. Jensen
A. la Cour-Harbo
Ripples in Mathematics
The Discrete Wavelet Transform
Springer

A first example 1

A signal with 8 samples:

$$56, \quad 40, \quad 8, \quad 24, \quad 48, \quad 48, \quad 40, \quad 16$$

We compute a transform as shown here:

56	40	8	24	48	48	40	16
48	16	48	28	8	-8	0	12
32	38	16	10	8	-8	0	12
35	-3	16	10	8	-8	0	12

To interpretation

A first example 2

First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the *means*. Differences are kept in each step.

A first example 2

First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the *means*. Differences are kept in each step.

56	40	8	24	48	48	40	16

A first example 2

First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the *means*. Differences are kept in each step.

$$\frac{56+40}{2}$$

56	40	8	24	48	48	40	16
48				8			

$$56-48$$

A first example 2

First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the *means*. Differences are kept in each step.

56	40	8	24	48	48	40	16
48	16			8	-8		

A first example 2

First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the *means*. Differences are kept in each step.

56	40	8	24	48	48	40	16
48	16	48		8	-8	0	

A first example 2

First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the *means*. Differences are kept in each step.

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12

A first example 2

First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the *means*. Differences are kept in each step.

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
				8	−8	0	12

A first example 2

First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the *means*. Differences are kept in each step.

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
32		16		8	−8	0	12

A first example 2

First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the *means*. Differences are kept in each step.

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
32	38	16	10	8	−8	0	12

A first example 2

First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the *means*. Differences are kept in each step.

56	40	8	24	48	48	40	16
48	16	48	28	8	-8	0	12
32	38	16	10	8	-8	0	12
		16	10	8	-8	0	12

A first example 2

First row is the original signal. The second row in the table is generated by taking the mean of the samples pairwise, put them in the first four places, and then the difference between the the first member of the pair and the computed mean. Computations are repeated on the *means*. Differences are kept in each step.

56	40	8	24	48	48	40	16
48	16	48	28	8	-8	0	12
32	38	16	10	8	-8	0	12
35	-3	16	10	8	-8	0	12

A first example 3

The transform is invertible. We start from the bottom row. We add and subtract the difference to the mean, and repeat the process up to the first row.

A first example 3

The transform is invertible. We start from the bottom row. We add and subtract the difference to the mean, and repeat the process up to the first row.

35	-3	16	10	8	-8	0	12

A first example 3

The transform is invertible. We start from the bottom row. We add and subtract the difference to the mean, and repeat the process up to the first row.

32	38						
35	-3	16	10	8	-8	0	12

A first example 3

The transform is invertible. We start from the bottom row. We add and subtract the difference to the mean, and repeat the process up to the first row.

32	38	16	10	8	-8	0	12
35	-3	16	10	8	-8	0	12

A first example 3

The transform is invertible. We start from the bottom row. We add and subtract the difference to the mean, and repeat the process up to the first row.

48	16	48	28				
32	38	16	10	8	-8	0	12
35	-3	16	10	8	-8	0	12

A first example 3

The transform is invertible. We start from the bottom row. We add and subtract the difference to the mean, and repeat the process up to the first row.

48	16	48	28	8	−8	0	12
32	38	16	10	8	−8	0	12
35	−3	16	10	8	−8	0	12

A first example 3

The transform is invertible. We start from the bottom row. We add and subtract the difference to the mean, and repeat the process up to the first row.

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
32	38	16	10	8	−8	0	12
35	−3	16	10	8	−8	0	12

A first example 4

We replace samples in the transformed signal below 4 by zero (thresholding) and then repeat the reconstruction procedure:

A first example 4

We replace samples in the transformed signal below 4 by zero (thresholding) and then repeat the reconstruction procedure:

35	0	16	10	8	-8	0	12

A first example 4

We replace samples in the transformed signal below 4 by zero (thresholding) and then repeat the reconstruction procedure:

35	35						
35	0	16	10	8	-8	0	12

A first example 4

We replace samples in the transformed signal below 4 by zero (thresholding) and then repeat the reconstruction procedure:

35	35	16	10	8	-8	0	12
35	0	16	10	8	-8	0	12

A first example 4

We replace samples in the transformed signal below 4 by zero (thresholding) and then repeat the reconstruction procedure:

51	19	45	25				
35	35	16	10	8	-8	0	12
35	0	16	10	8	-8	0	12

A first example 4

We replace samples in the transformed signal below 4 by zero (thresholding) and then repeat the reconstruction procedure:

51	19	45	25	8	-8	0	12
35	35	16	10	8	-8	0	12
35	0	16	10	8	-8	0	12

A first example 4

We replace samples in the transformed signal below 4 by zero (thresholding) and then repeat the reconstruction procedure:

59	43	11	27	45	45	37	13
51	19	45	25	8	-8	0	12
35	35	16	10	8	-8	0	12
35	0	16	10	8	-8	0	12

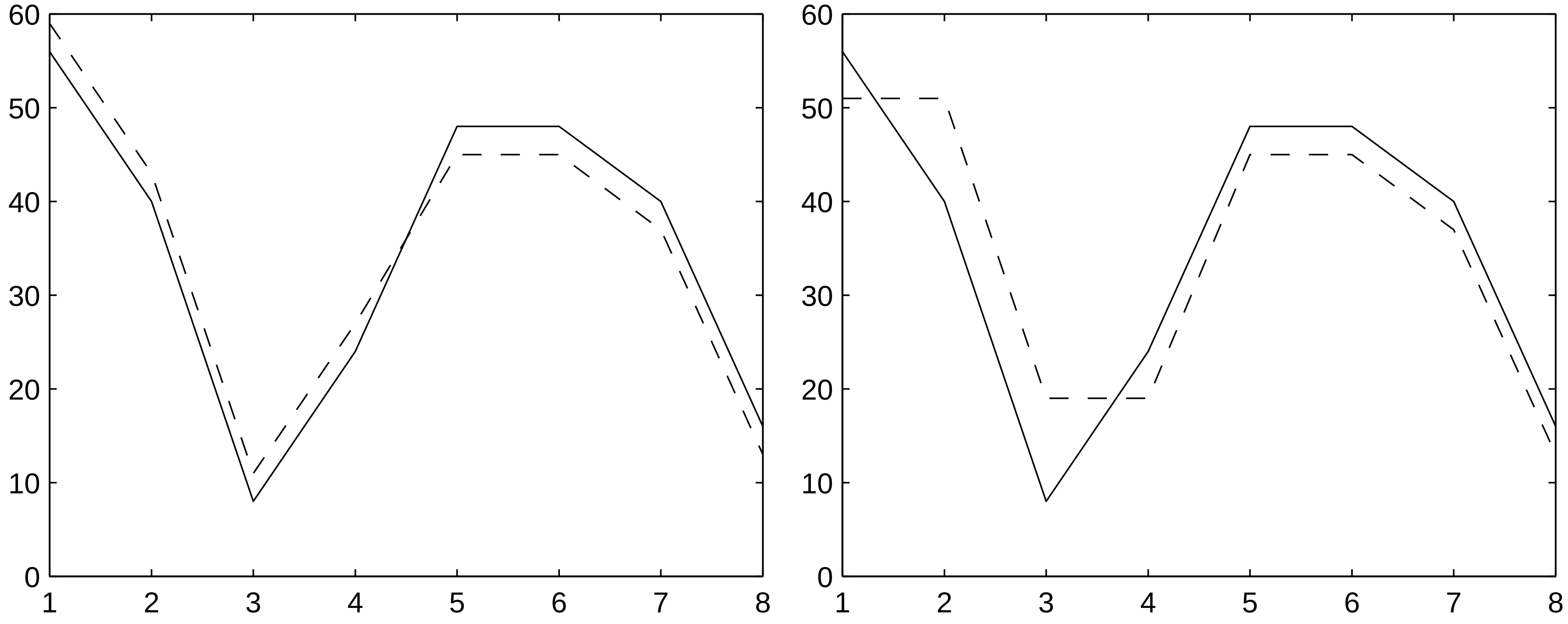
A first example 5

We now replace samples in the transformed signal below 9 by zero (thresholding) and then repeat the reconstruction procedure. The final result is:

51	51	19	19	45	45	37	13
51	19	45	25	0	0	0	12
35	35	16	10	0	0	0	12
35	0	16	10	0	0	0	12

A first example 6

Here is now a graphical representation of the results. Full line original signal, and dashed line for thresholding, left hand side 4, right hand side 9.



Lifting 1

We now look at the transform in the first example. The direct transform $(a, b) \to (d, s)$ is given by

$$s = \frac{a+b}{2},$$
$$d = a - s.$$

and the inverse $(d, s) \to (a, b)$ by

$$a = s + d;,$$
$$b = s - d.$$

Lifting 2

They can be realized as in-place transforms in two steps. The direct transform as

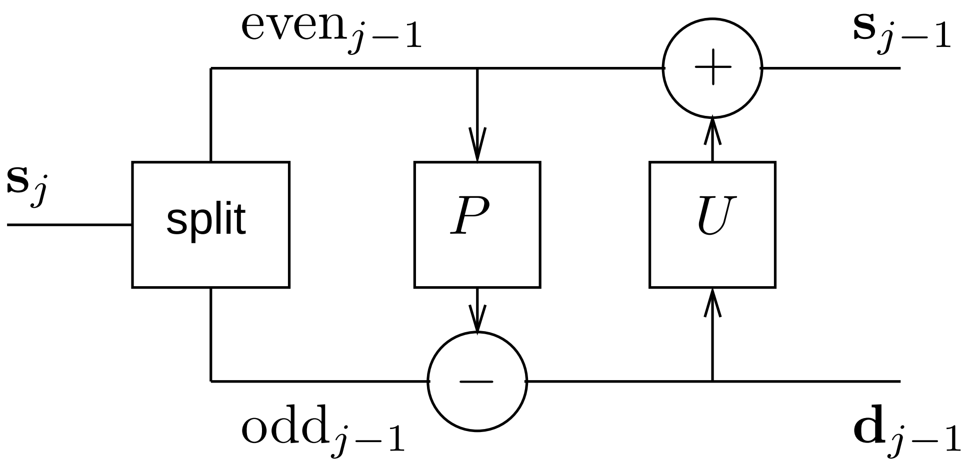
$$\begin{array}{ll} \text{First step:} & a, b \;\rightarrow\; a, \tfrac{1}{2}(a+b) \\ \text{Second step:} & a, s \;\rightarrow\; a - s, s. \end{array}$$

and the inverse transform as

$$\begin{array}{ll} \text{First step:} & d, s \;\rightarrow\; d + s, s \\ \text{Second step:} & a, s \;\rightarrow\; a, 2s - a. \end{array}$$

Lifting 3

Notation: Finite sequence of numbers (samples of a signal) of length 2^j is denoted by $\mathbf{s}_j = \{s_j[1], s_j[2], \ldots, s_j[2^j]\}$.
Basic idea in lifting is given in this figure:



P: Predict
U: Update

Lifting 4

An alternative to the first example is difference and mean computation, in that order:

$$a, b \longrightarrow \delta, \mu$$

where

$$\delta = b - a$$
$$\mu = \frac{a + b}{2} = a + \frac{\delta}{2}$$

Lifting 5

Predict: In the difference-mean case:

$$d_{j-1}[n] = s_j[2n+1] - s_j[2n].$$

In general:

$$\mathbf{d}_{j-1} = \text{odd}_{j-1} - P(\text{even}_{j-1}).$$

Update: In the difference-mean case:

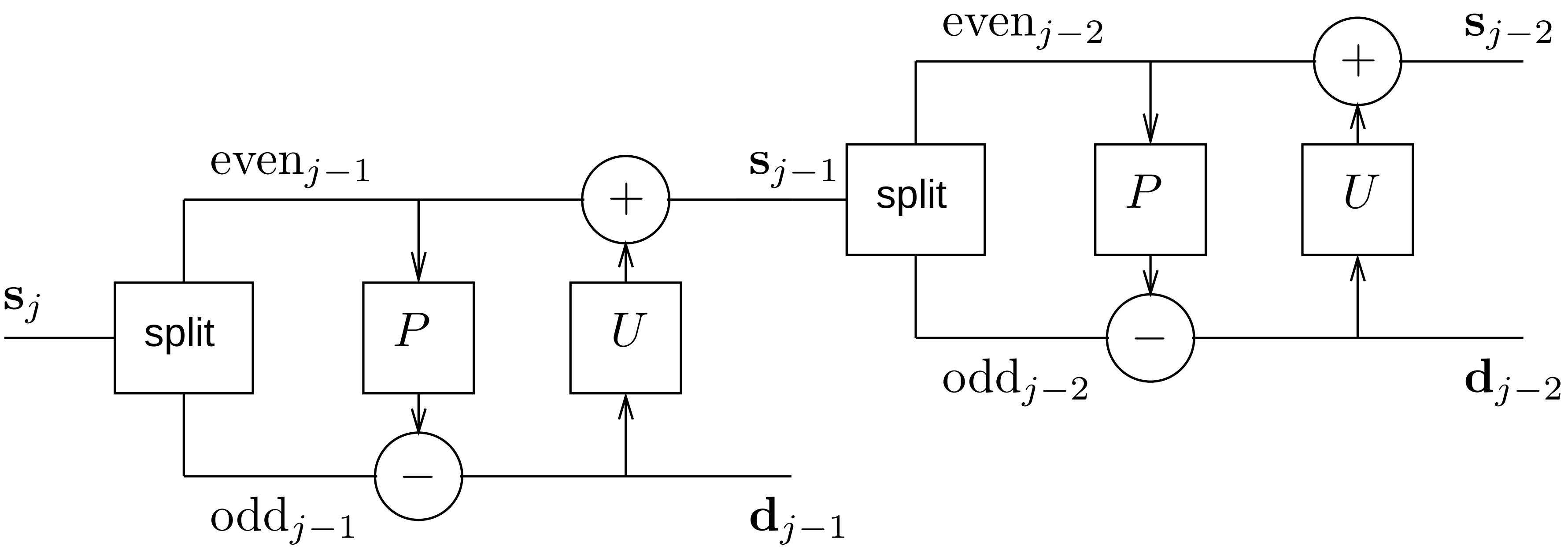
$$s_{j-1}[n] = s_j[2n] + d_{j-1}[n]/2.$$

In general:

$$\mathbf{s}_{j-1} = \text{even}_{j-1} + U(\mathbf{d}_{j-1}).$$

Lifting 6

The transform $\mathbf{s}_j \to \mathbf{s}_{j-1}, \mathbf{d}_{j-1}$ is called one step lifting. In the the first example we repeatedly applied the transform to the $\mathbf{s}$-components, ending with $\mathbf{s}_0$ of length 1. Two step discrete wavelet transform:



Lifting 7

The difference and mean computations in the in place form:

$s_3[0]$	$s_3[1]$	$s_3[2]$	$s_3[3]$	$s_3[4]$	$s_3[5]$	$s_3[6]$	$s_3[7]$	
$s_3[0]$	$d_2[0]$	$s_3[2]$	$d_2[1]$	$s_3[4]$	$d_2[2]$	$s_3[6]$	$d_2[3]$	P
$s_2[0]$	$d_2[0]$	$s_2[1]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$s_2[3]$	$d_2[3]$	U
$s_2[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	P
$s_1[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$s_1[1]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	U
$s_1[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$d_0[0]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	P
$s_0[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$d_0[0]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	U

Lifting 8

The in place transform step by step:

Lifting 8

The in place transform step by step:

$s_3[0]$	$s_3[1]$	$s_3[2]$	$s_3[3]$	$s_3[4]$	$s_3[5]$	$s_3[6]$	$s_3[7]$

Lifting 8

The in place transform step by step:

$s_3[0]$	$s_3[1]$	$s_3[2]$	$s_3[3]$	$s_3[4]$	$s_3[5]$	$s_3[6]$	$s_3[7]$	
$s_3[0]$	$d_2[0]$	$s_3[2]$	$d_2[1]$	$s_3[4]$	$d_2[2]$	$s_3[6]$	$d_2[3]$	P

Lifting 8

The in place transform step by step:

$s_3[0]$	$s_3[1]$	$s_3[2]$	$s_3[3]$	$s_3[4]$	$s_3[5]$	$s_3[6]$	$s_3[7]$	
$s_3[0]$	$d_2[0]$	$s_3[2]$	$d_2[1]$	$s_3[4]$	$d_2[2]$	$s_3[6]$	$d_2[3]$	P
$s_2[0]$	$d_2[0]$	$s_2[1]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$s_2[3]$	$d_2[3]$	U

Lifting 8

The in place transform step by step:

$s_3[0]$	$s_3[1]$	$s_3[2]$	$s_3[3]$	$s_3[4]$	$s_3[5]$	$s_3[6]$	$s_3[7]$	
$s_3[0]$	$d_2[0]$	$s_3[2]$	$d_2[1]$	$s_3[4]$	$d_2[2]$	$s_3[6]$	$d_2[3]$	P
$s_2[0]$	$d_2[0]$	$s_2[1]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$s_2[3]$	$d_2[3]$	U
$s_2[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	P

Lifting 8

The in place transform step by step:

$s_3[0]$	$s_3[1]$	$s_3[2]$	$s_3[3]$	$s_3[4]$	$s_3[5]$	$s_3[6]$	$s_3[7]$	
$s_3[0]$	$d_2[0]$	$s_3[2]$	$d_2[1]$	$s_3[4]$	$d_2[2]$	$s_3[6]$	$d_2[3]$	P
$s_2[0]$	$d_2[0]$	$s_2[1]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$s_2[3]$	$d_2[3]$	U
$s_2[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	P
$s_1[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$s_1[1]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	U

Lifting 8

The in place transform step by step:

$s_3[0]$	$s_3[1]$	$s_3[2]$	$s_3[3]$	$s_3[4]$	$s_3[5]$	$s_3[6]$	$s_3[7]$	
$s_3[0]$	$d_2[0]$	$s_3[2]$	$d_2[1]$	$s_3[4]$	$d_2[2]$	$s_3[6]$	$d_2[3]$	P
$s_2[0]$	$d_2[0]$	$s_2[1]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$s_2[3]$	$d_2[3]$	U
$s_2[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	P
$s_1[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$s_1[1]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	U
$s_1[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$d_0[0]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	P

Lifting 8

The in place transform step by step:

$s_3[0]$	$s_3[1]$	$s_3[2]$	$s_3[3]$	$s_3[4]$	$s_3[5]$	$s_3[6]$	$s_3[7]$	
$s_3[0]$	$d_2[0]$	$s_3[2]$	$d_2[1]$	$s_3[4]$	$d_2[2]$	$s_3[6]$	$d_2[3]$	P
$s_2[0]$	$d_2[0]$	$s_2[1]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$s_2[3]$	$d_2[3]$	U
$s_2[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	P
$s_1[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$s_1[1]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	U
$s_1[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$d_0[0]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	P
$s_0[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$d_0[0]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	U

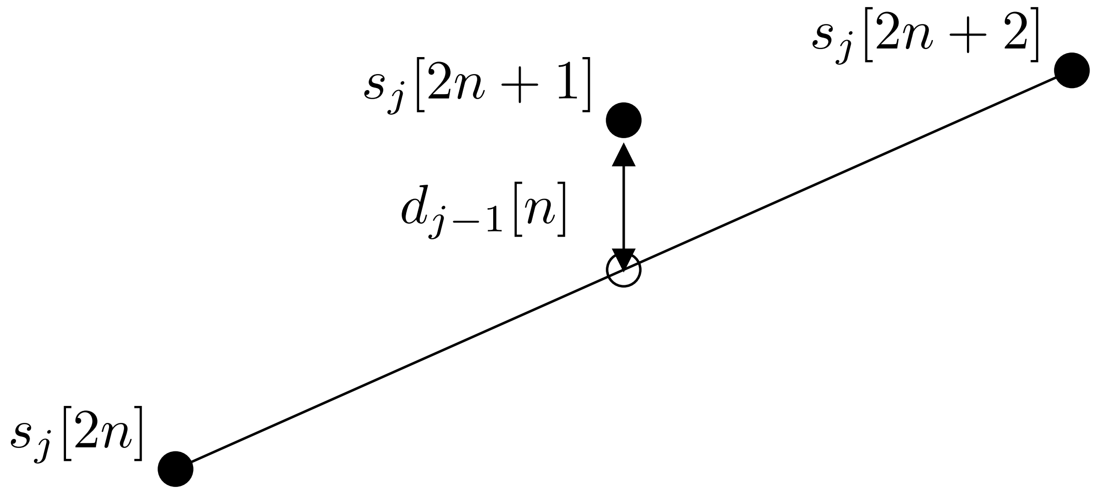
Lifting 8

In place transform with pattern of computed values:

$s_3[0]$	$s_3[1]$	$s_3[2]$	$s_3[3]$	$s_3[4]$	$s_3[5]$	$s_3[6]$	$s_3[7]$	
$s_3[0]$	$d_2[0]$	$s_3[2]$	$d_2[1]$	$s_3[4]$	$d_2[2]$	$s_3[6]$	$d_2[3]$	P
$s_2[0]$	$d_2[0]$	$s_2[1]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$s_2[3]$	$d_2[3]$	U
$s_2[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$s_2[2]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	P
$s_1[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$s_1[1]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	U
$s_1[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$d_0[0]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	P
$s_0[0]$	$d_2[0]$	$d_1[0]$	$d_2[1]$	$d_0[0]$	$d_2[2]$	$d_1[1]$	$d_2[3]$	U

Lifting 9

A second example of lifting: Base prediction on assumption that signal is linear, ie $s_j[n] = \alpha n + \beta$. Prediction of $s_j[2n+1]$ is then $\frac{1}{2}(s_j[2n] + s_j[2n+2])$, and we need to save only $d_{j-1}[n] = s_j[2n+1] - \frac{1}{2}(s_j[2n] + s_j[2n+2])$.



Lifting 10

The update step: Keep mean of $s_j[n]$ sequence equal to mean of $s_{j-1}[n]$ sequence. Final result is

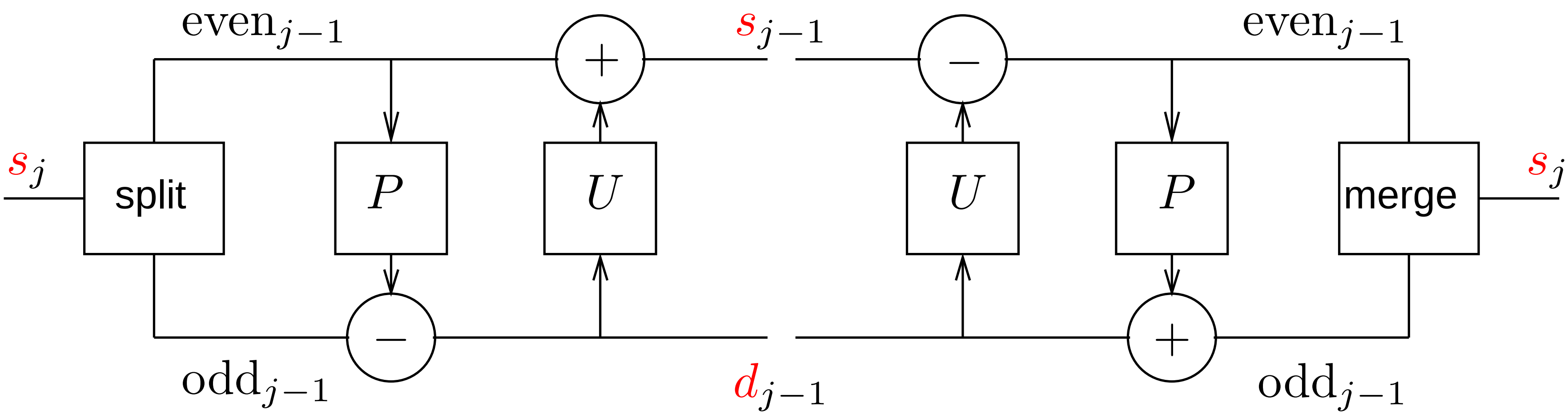
$$d_{j-1}[n] = s_j[2n+1] - \tfrac{1}{2}(s_j[2n] + s_j[2n+2]),$$
$$s_{j-1}[n] = s_j[2n] + \tfrac{1}{4}(d_{j-1}[n-1] + d_{j-1}[n]).$$

Inverse transform:

$$s_j[2n] = s_{j-1}[n] - \tfrac{1}{4}(d_{j-1}[n-1] + d_{j-1}[n]),$$
$$s_j[2n+1] = d_{j-1}[n] + \tfrac{1}{2}(s_j[2n] + s_j[2n+2]).$$

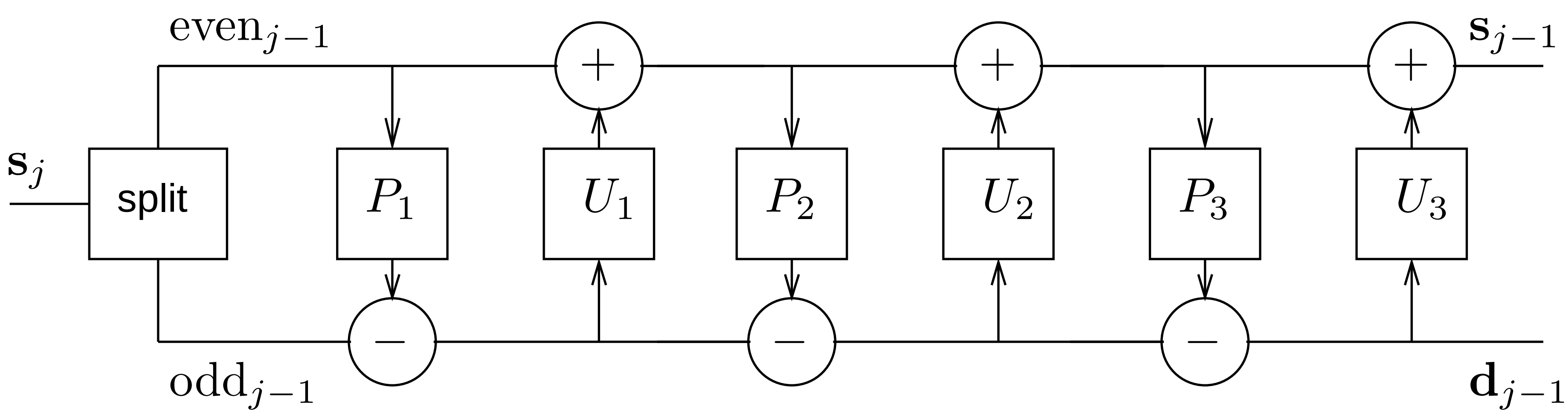
Lifting 11

Summary of one step lifting and inverse lifting:



Generalized lifting 1

One can generalize the lifting step by allowing several pairs of predictions and updates.



Generalized lifting 2

An example, Daubechies 4

$$s_{j-1}^{(1)}[n] = s_j[2n] + \sqrt{3}s_j[2n+1]$$

$$d_{j-1}^{(1)}[n] = s_j[2n+1] - \tfrac{1}{4}\sqrt{3}s_{j-1}^{(1)}[n] - \tfrac{1}{4}(\sqrt{3}-2)s_{j-1}^{(1)}[n-1]$$

$$s_{j-1}^{(2)}[n] = s_{j-1}^{(1)}[n] - d_{j-1}^{(1)}[n+1]$$

$$s_{j-1}[n] = \frac{\sqrt{3}-1}{\sqrt{2}}s_{j-1}^{(2)}[n]$$

$$d_{j-1}[n] = \frac{\sqrt{3}+1}{\sqrt{2}}d_{j-1}^{(1)}[n]$$

Generalized lifting 3

Last two steps are normalization steps, in order to preserve the energy in the transform, ie

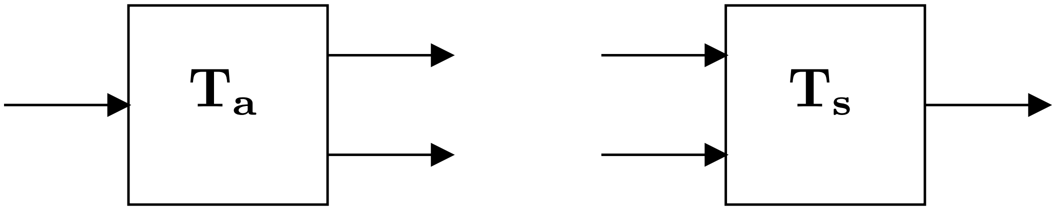
$$\sum_n |s_j[n]|^2 = \sum_n |s_{j-1}[n]|^2 + \sum_n |d_{j-1}[n]|^2$$

now holds. Note that

$$\frac{\sqrt{3}-1}{\sqrt{2}} \cdot \frac{\sqrt{3}+1}{\sqrt{2}} = 1 \; .$$

DWT 1

Finally we can introduce the Discrete Wavelet Transform (DWT). Block diagrams are used for our lifting and inverse lifting based one step transforms:

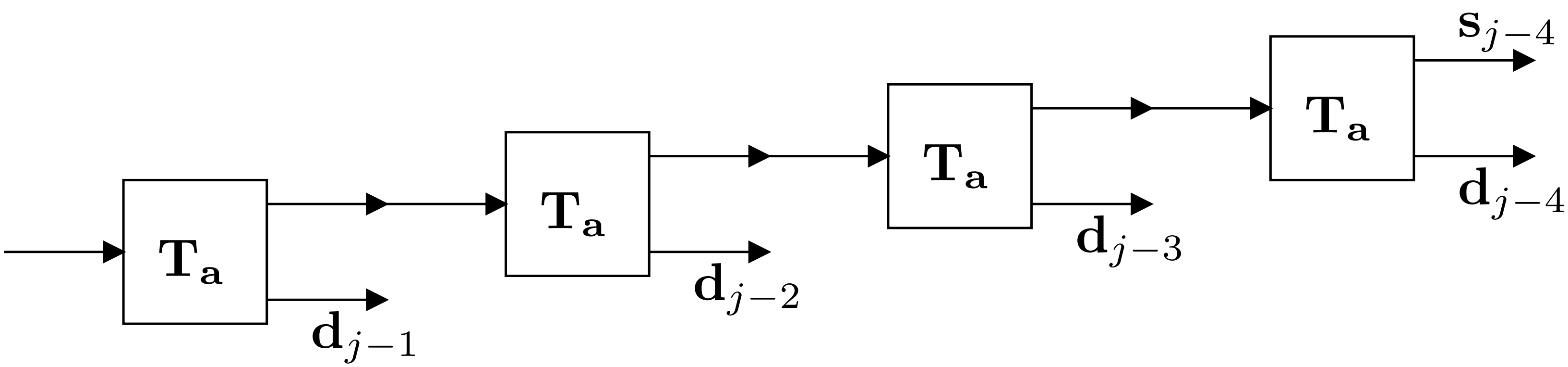


DWT 2

A DWT over four scales

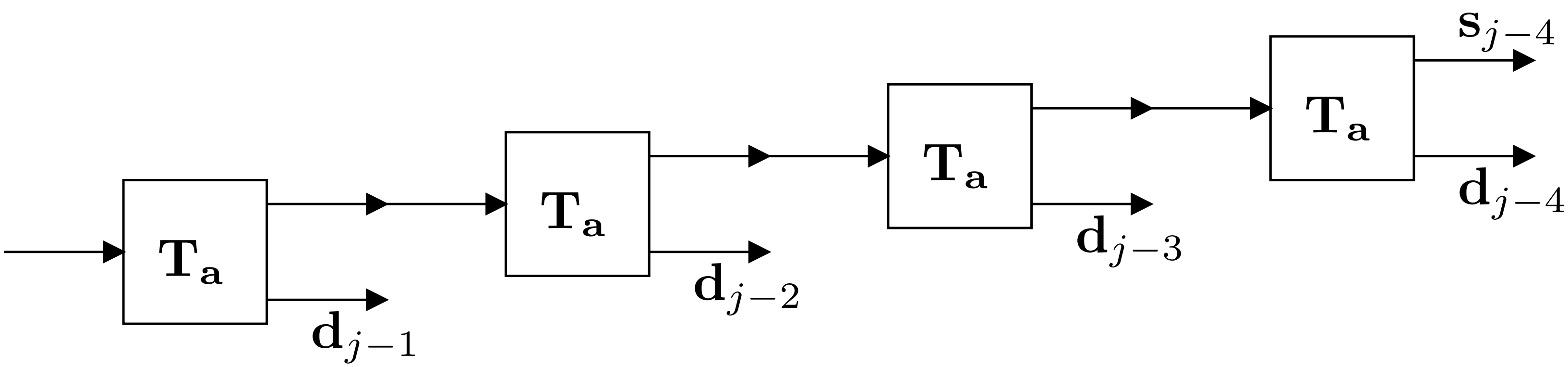
DWT 2

A DWT over four scales



DWT 2

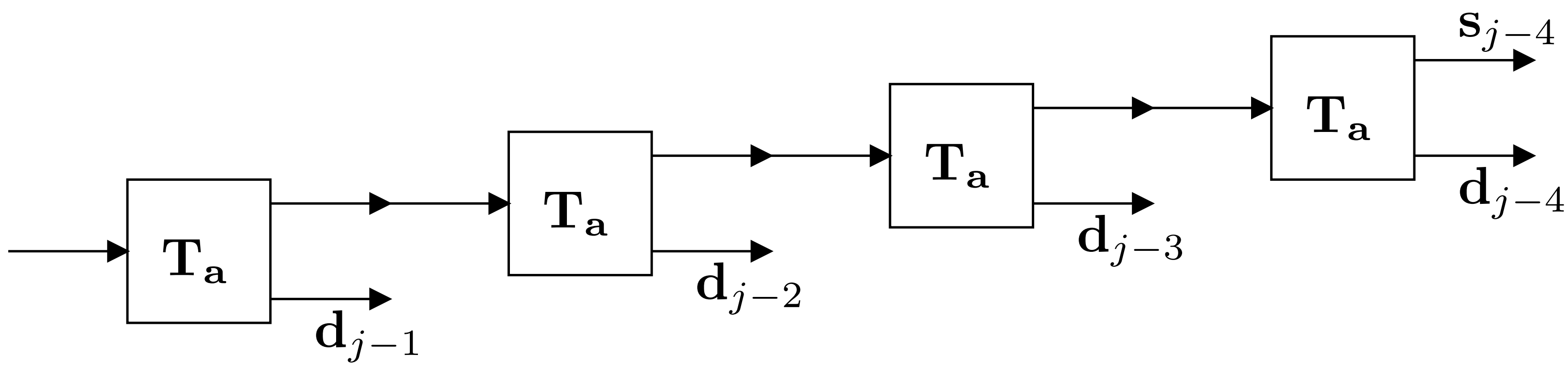
A DWT over four scales



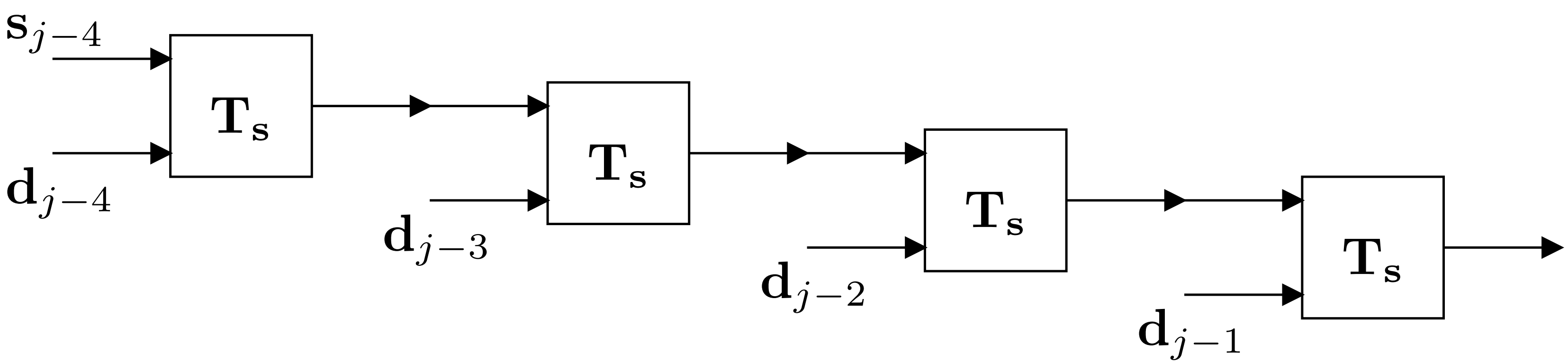
The inverse DWT over four scales

DWT 2

A DWT over four scales



The inverse DWT over four scales



DWT 3

A family of transforms (Cohen, Daubechies, Faveau)

$$d^{(1)}_{j-1}[n] = s_j[2n+1] - \tfrac{1}{2}(s_j[2n] + s_j[2n+2])$$

CDF(2,2) $$s^{(1)}_{j-1}[n] = s_j[2n] + \tfrac{1}{4}(d_{j-1}[n-1] + d_{j-1}[n])$$

CDF(2,4) $$s^{(1)}_{j-1}[n] = s_j[2n] - \tfrac{1}{64}(3d_{j-1}[n-2] - 19d_{j-1}[n-1] - 19d_{j-1}[n] + 3d_{j-1}[n+1])$$

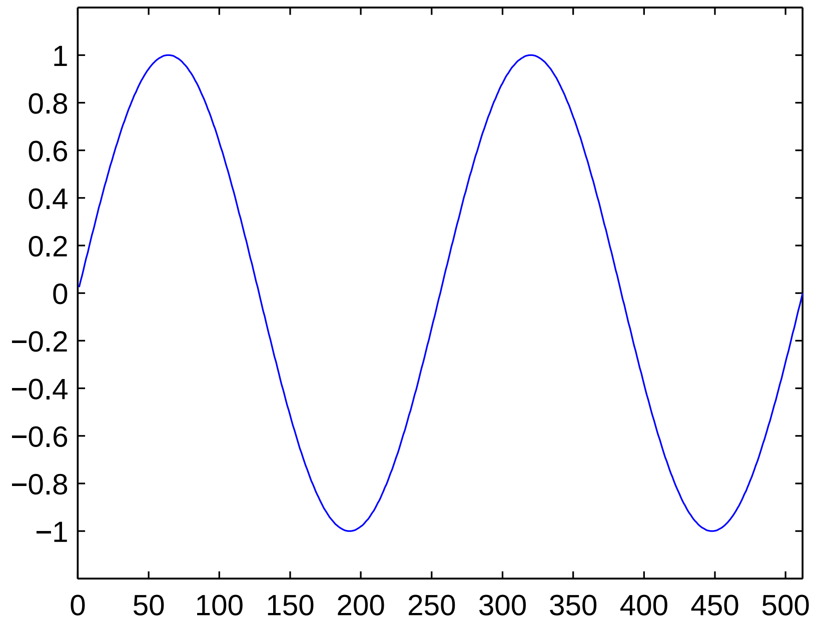
CDF(2,6) $$s^{(1)}_{j-1}[n] = s_j[2n] - \tfrac{1}{512}(-5d_{j-1}[n-3] + 39d_{j-1}[n-2] - 162d_{j-1}[n-1] - 162d_{j-1}[n] + 39d_{j-1}[n+1] - 5d_{j-1}[n+2])$$

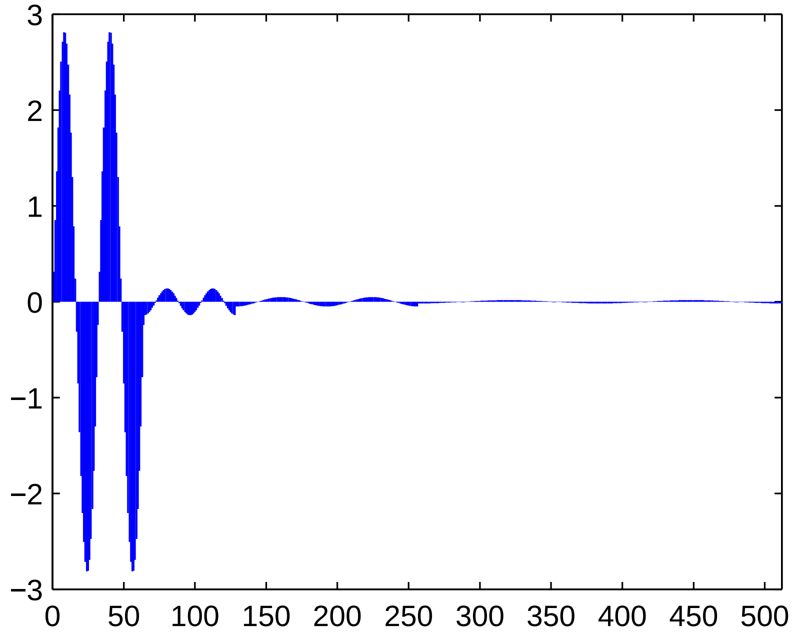
$$d_{j-1}[n] = \tfrac{1}{\sqrt{2}} d^{(1)}_{j-1}[n]$$

$$s_{j-1}[n] = \sqrt{2} s^{(1)}_{j-1}[n]$$

Examples 1

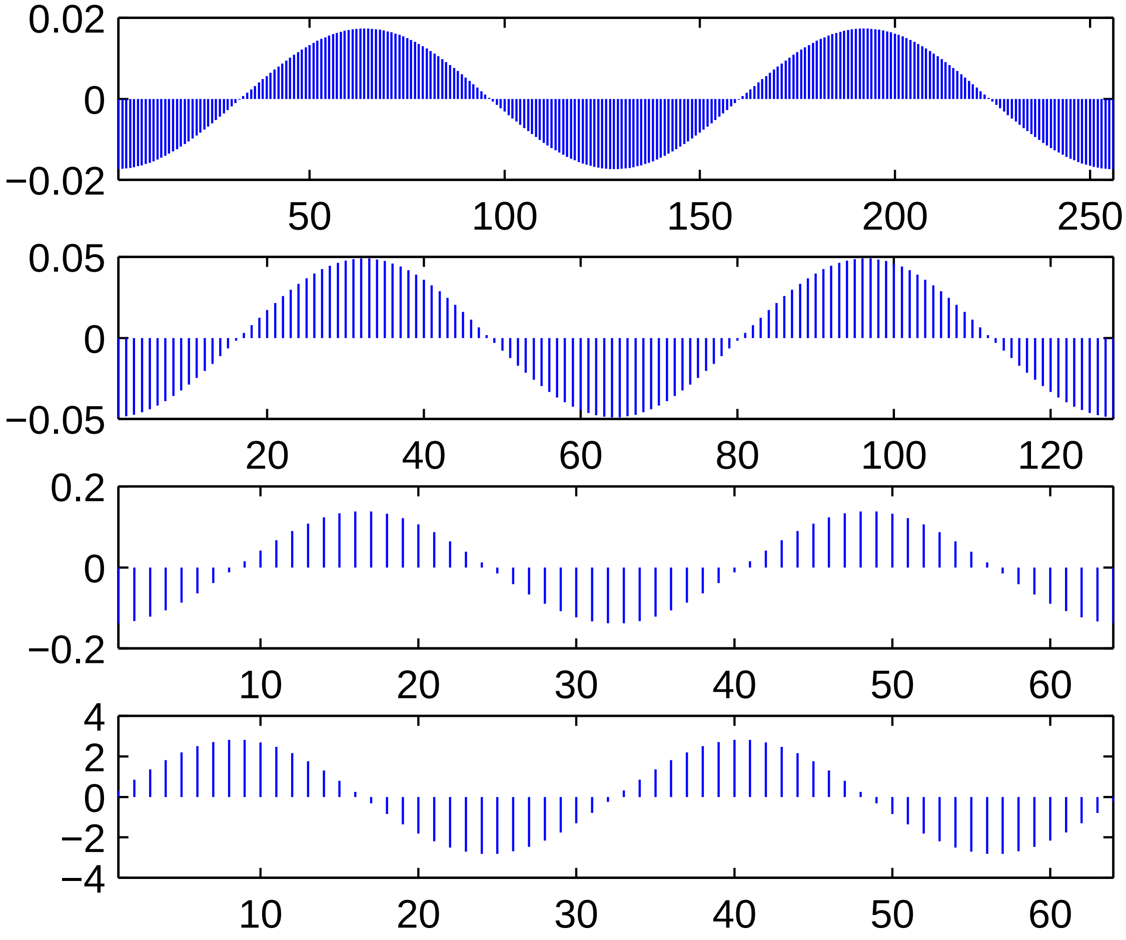
Now some examples on synthetic signals: The first problem is how to visualize the action of the wavelet transform. We start with a simple signal and perform a three-scale Haar transform.





Examples 2

The coefficients separately. Note vertical range in plots.



Examples 3

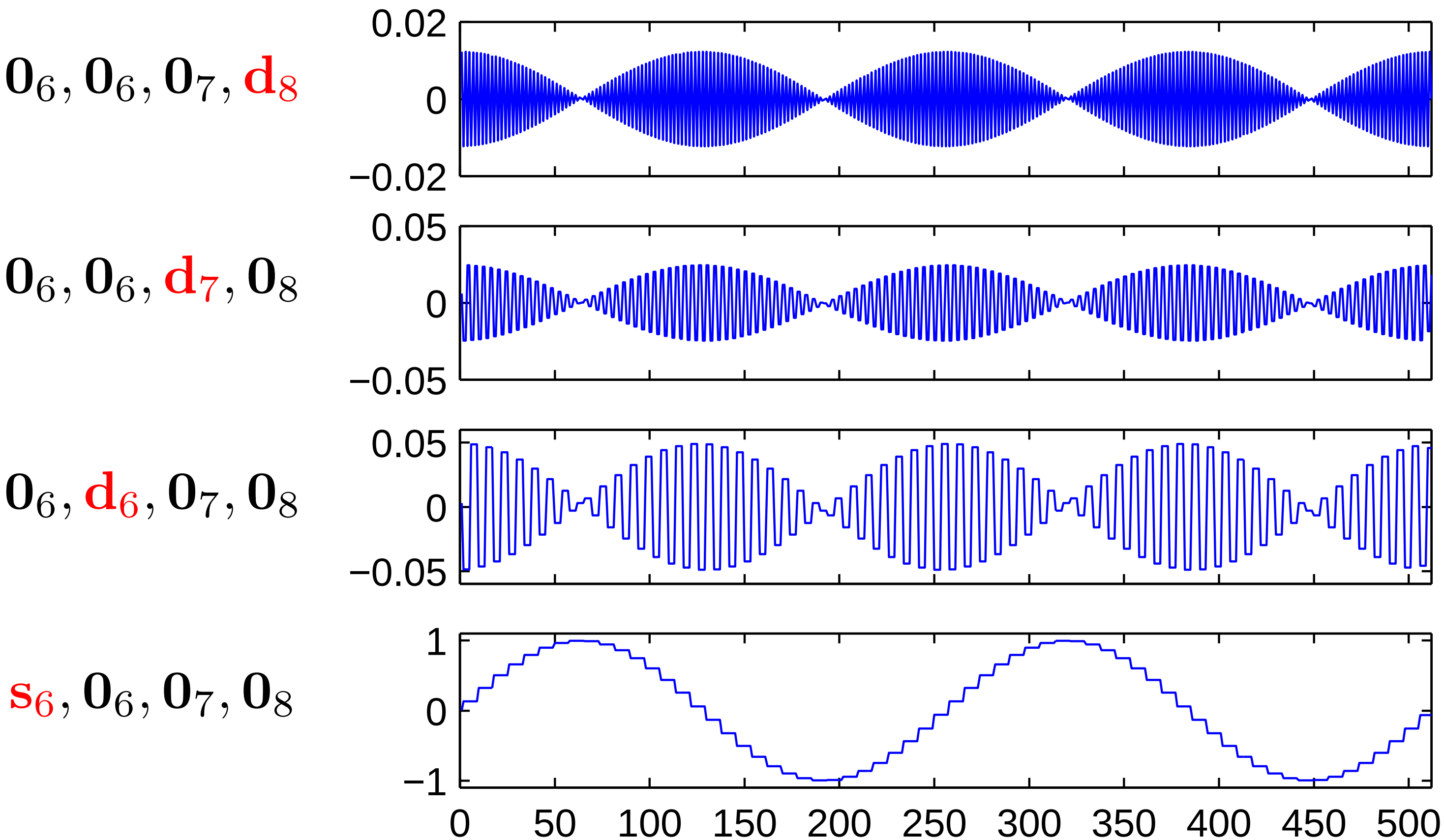
Multiresolution representation of the DWT of a signal:

Transform a signal $W_{\mathrm{a}}^{(3)}: \mathbf{s}_9 \to \mathbf{s}_6, \mathbf{d}_6, \mathbf{d}_7, \mathbf{d}_8$. Replace all entries but one in the transform by zeroes, and do the inverse transform. Schematically

$$
\begin{array}{rcccl}
W_{\mathrm{a}}^{(3)}: \mathbf{s}_9 & \longrightarrow & \underbrace{\mathbf{s}_6, \mathbf{d}_6, \mathbf{d}_7, \mathbf{d}_8} & & \\
 & & \downarrow & & \\
 & W_{\mathrm{s}}^{(3)}: & \overbrace{\mathbf{0}_6, \mathbf{d}_6, \mathbf{0}_7, \mathbf{0}_8} & \longrightarrow & \mathbf{s}_9'
\end{array}
$$

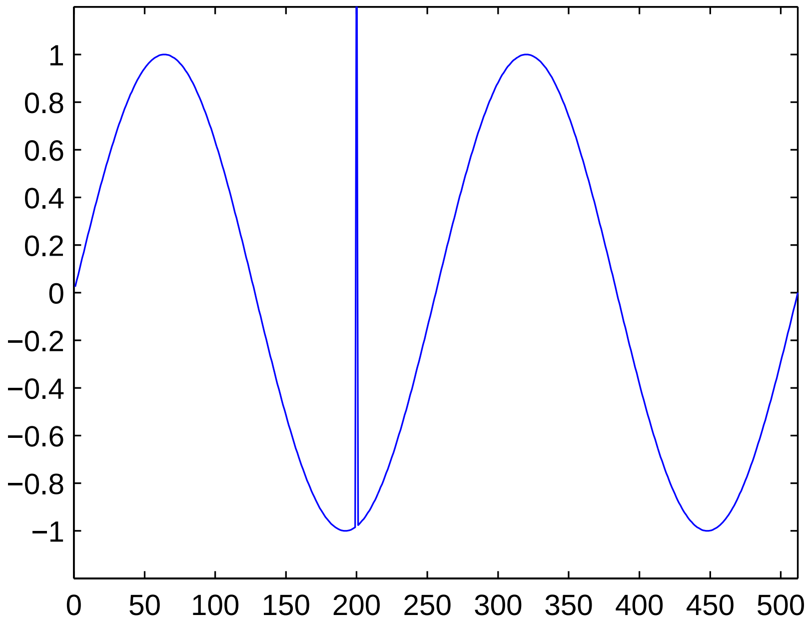
Examples 4

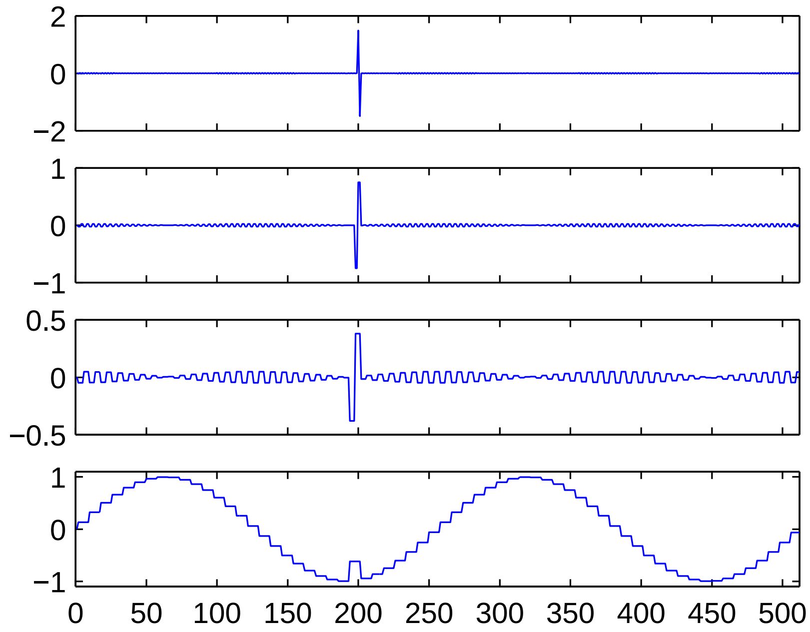
Multiresolution representation of sine signal, three scales, Haar transform.



Examples 5

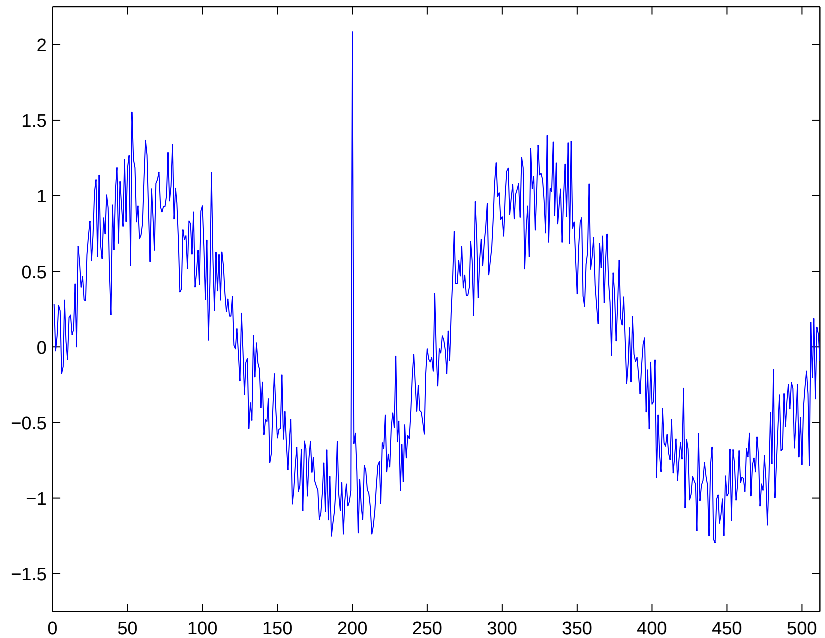
Singularity detection. Singularities can be localized in time using DWT. A sine plus a spike located at position 200:

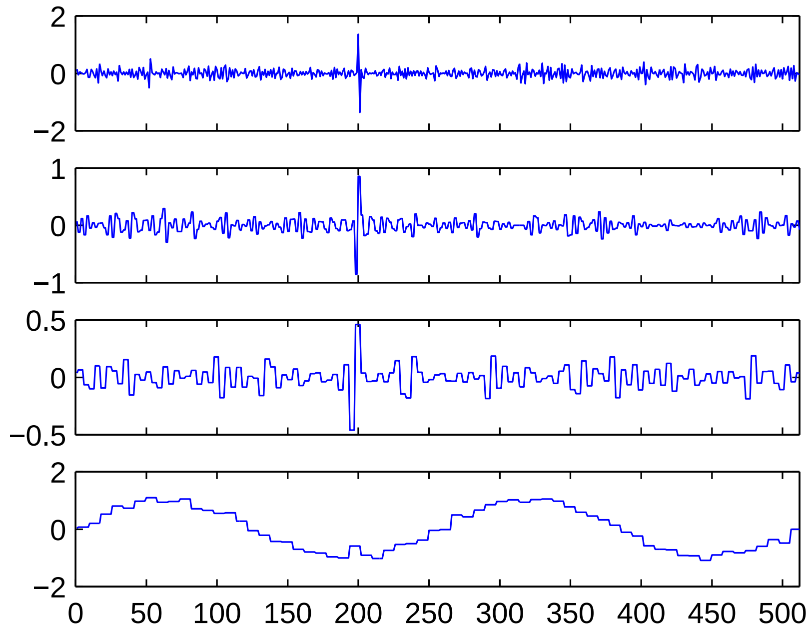




Examples 6

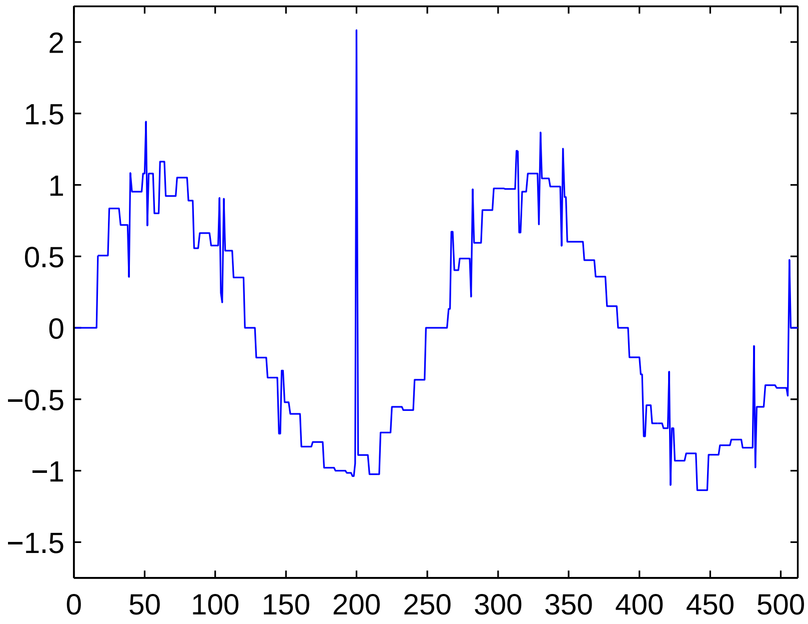
We do some denoising examples. First based on the Haar transform. Here is the sine plus spike, and its multiresolution representation:

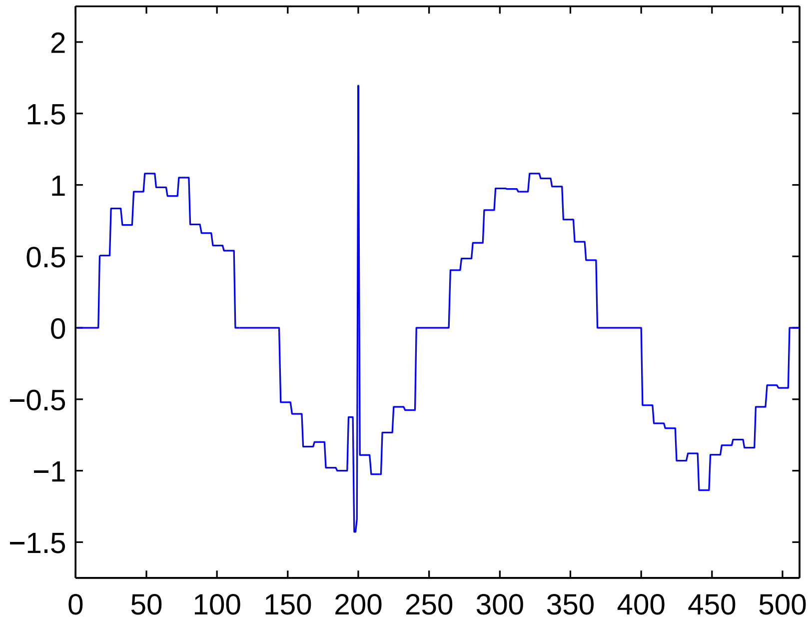




Examples 7

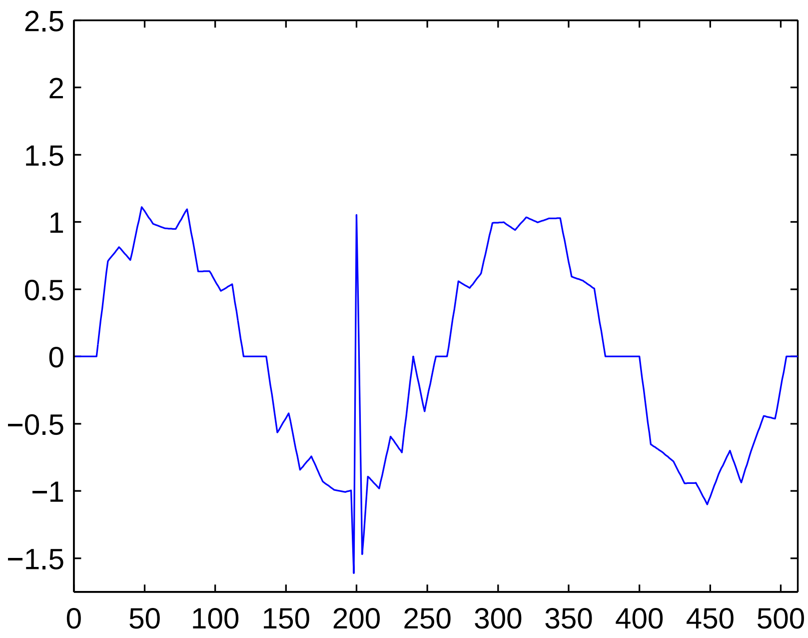
The idea in denoising is to keep the largest coefficients. On the left hand side we kept the 15% largest coefficients, and on the right hand side the 10% largest coefficients.

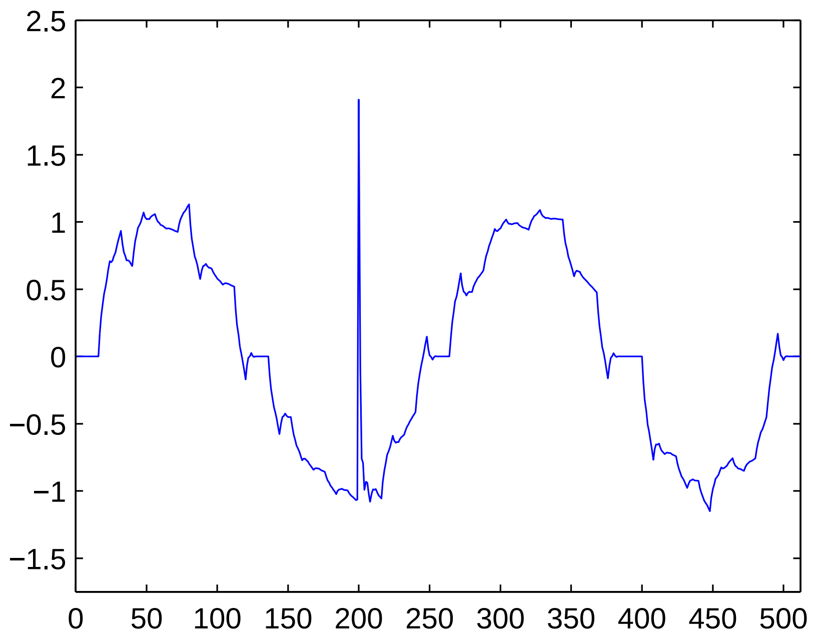




Examples 8

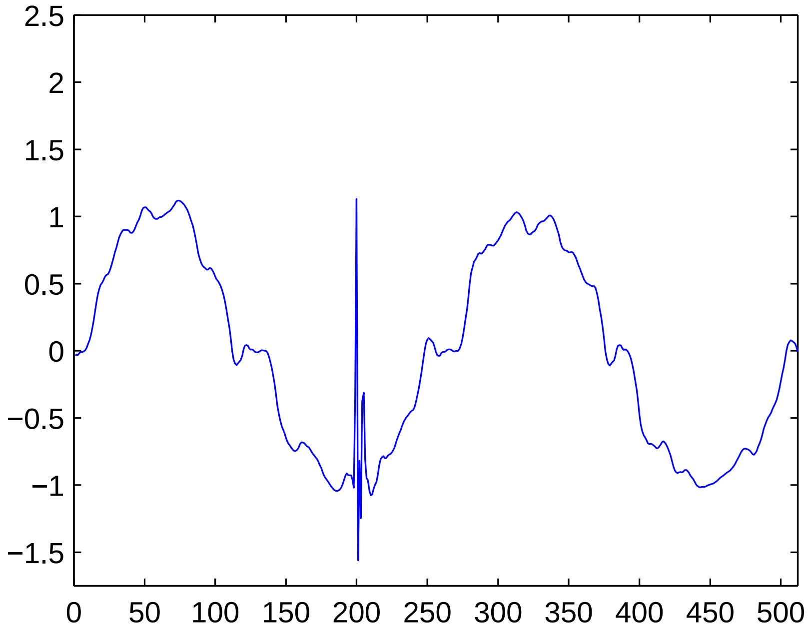
To get better performance one must use better wavelets. Same example, with CDF(2,2) (linear prediction) on the left, Daubechies 4 on the right. Largest 10% coefficients retained.

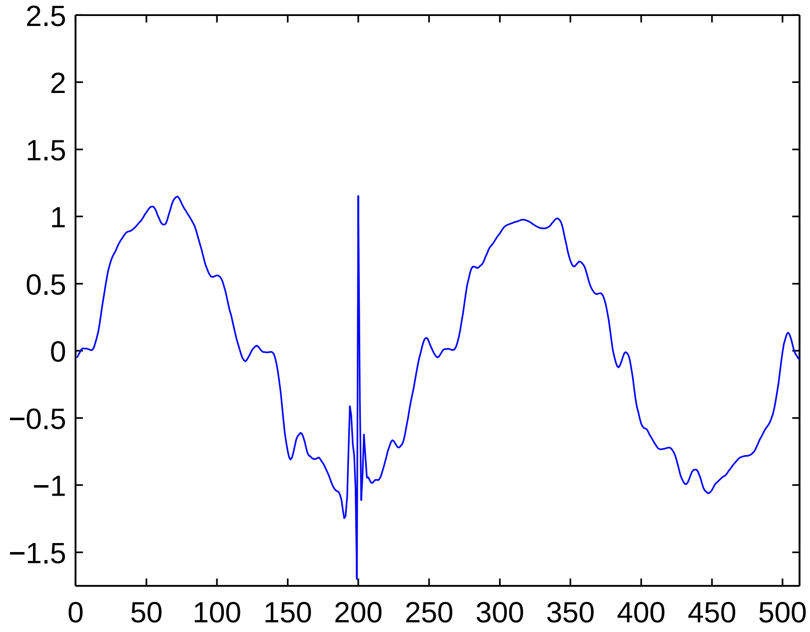




Examples 9

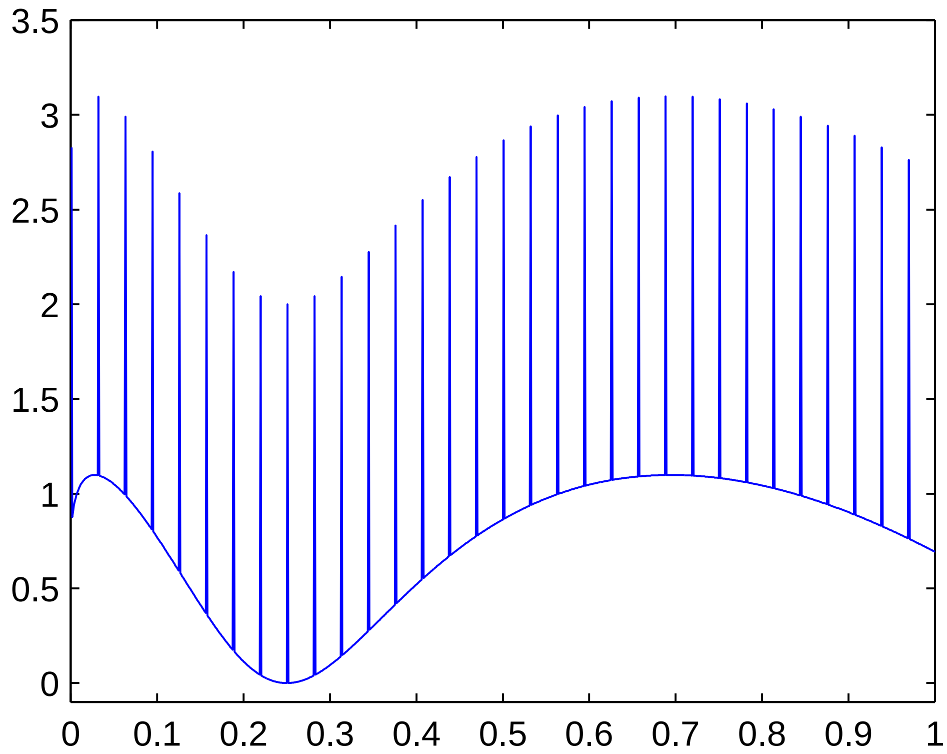
Same example with Daubechies transforms of length 8 and 12.





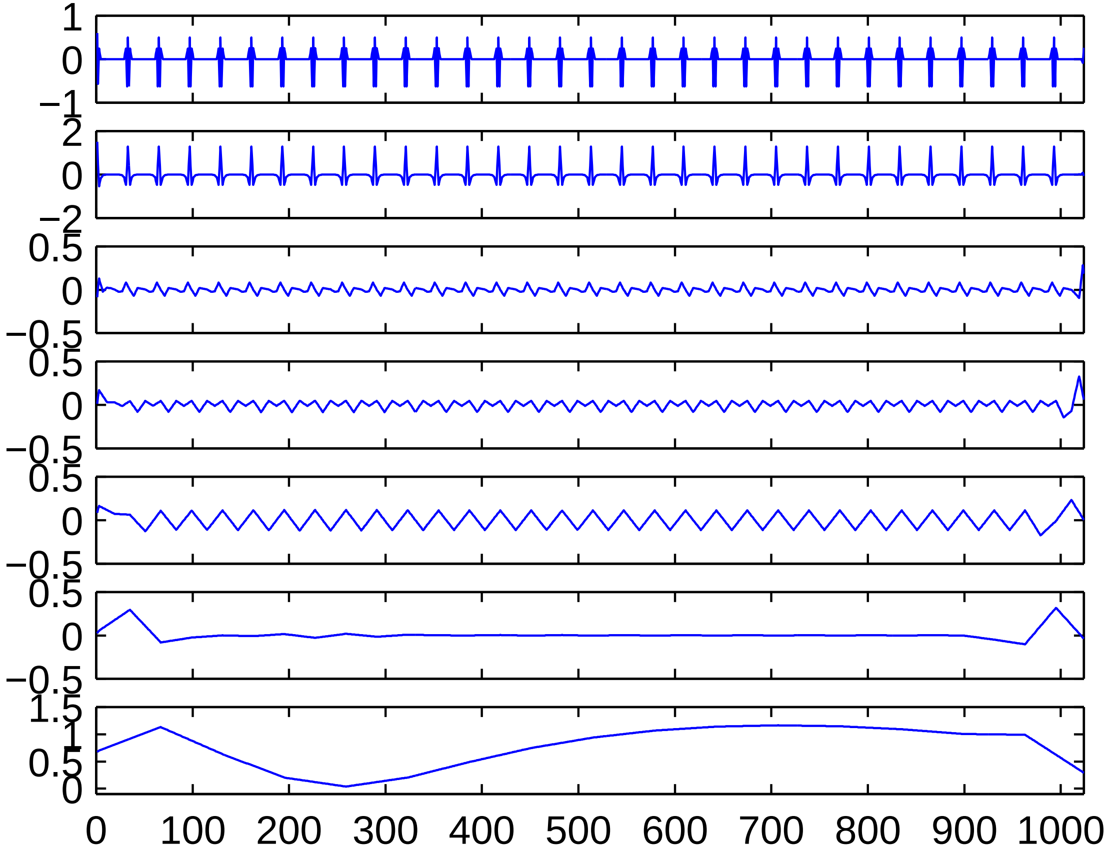
Examples 10

In the last example we show how to separate slow and fast variations in a signal. The function $\log(2 + \sin(3\pi\sqrt{t}))$, $0 \le r \le 1$, sampled 1024 times, and spikes added:



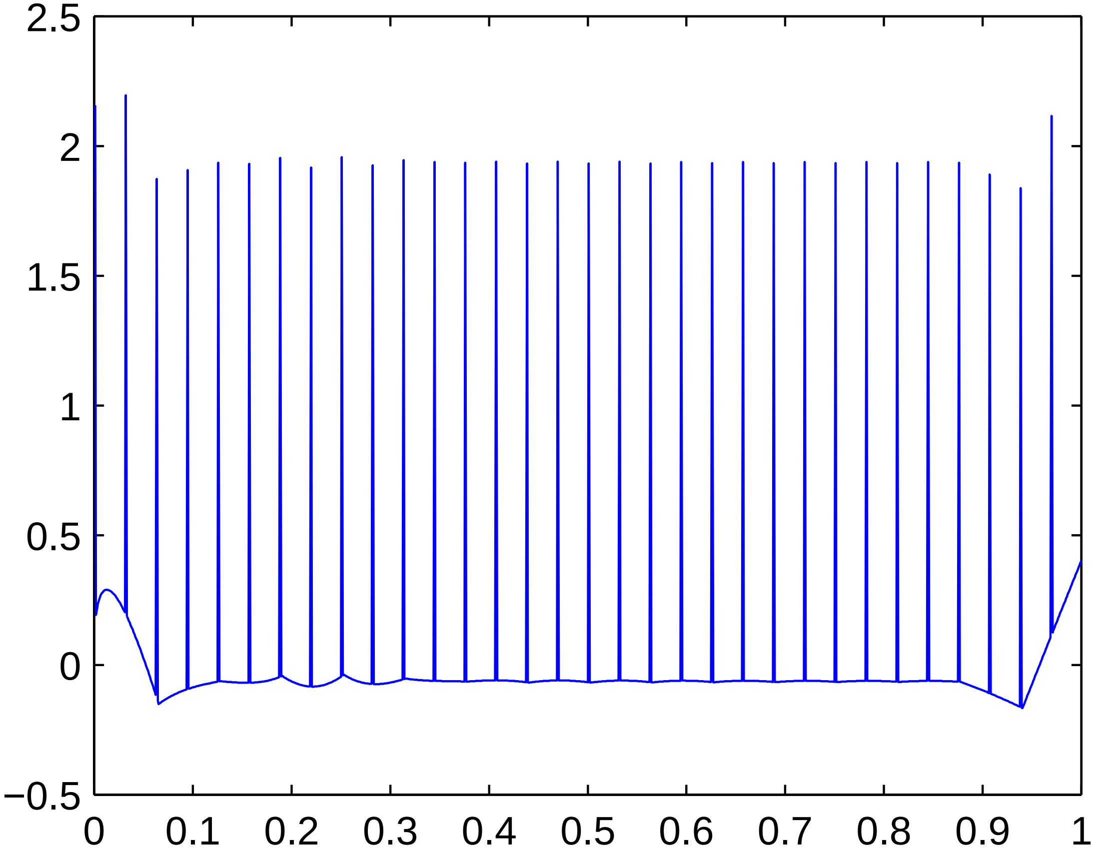
Examples 11

Multiresolution analysis, 6 scales, CDF(2,2):



Examples 12

Slow variation removed: Reconstruction based on $\mathbf{d}$-components.



Interpretation 1

We recall the first example. We now apply the inversion procedure to the signals $[1, 0, 0, 0, 0, 0, 0, 0]$, $[0, 1, 0, 0, 0, 0, 0, 0]$, and $[0, 0, 1, 0, 0, 0, 0, 0]$.

1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0

Interpretation 2

1	1	1	1	-1	-1	-1	-1
1	1	-1	-1	0	0	0	0
1	-1	0	0	0	0	0	0
0	1	0	0	0	0	0	0

1	1	-1	-1	0	0	0	0
1	-1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0

Interpretation 3

Linear algebra interpretation as a matrix:

$$\mathbf{W}_{\mathrm{s}}^{(3)} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & -1 & 0 & 1 & 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & 1 \\ 1 & -1 & 0 & -1 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

Interpretation 4

We do the same for the direct transform. Here is one example computation:

1	0	0	0	0	0	0	0
$\frac{1}{2}$	0	0	0	$\frac{1}{2}$	0	0	0
$\frac{1}{4}$	0	$\frac{1}{4}$	0	$\frac{1}{2}$	0	0	0
$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$	0	$\frac{1}{2}$	0	0	0

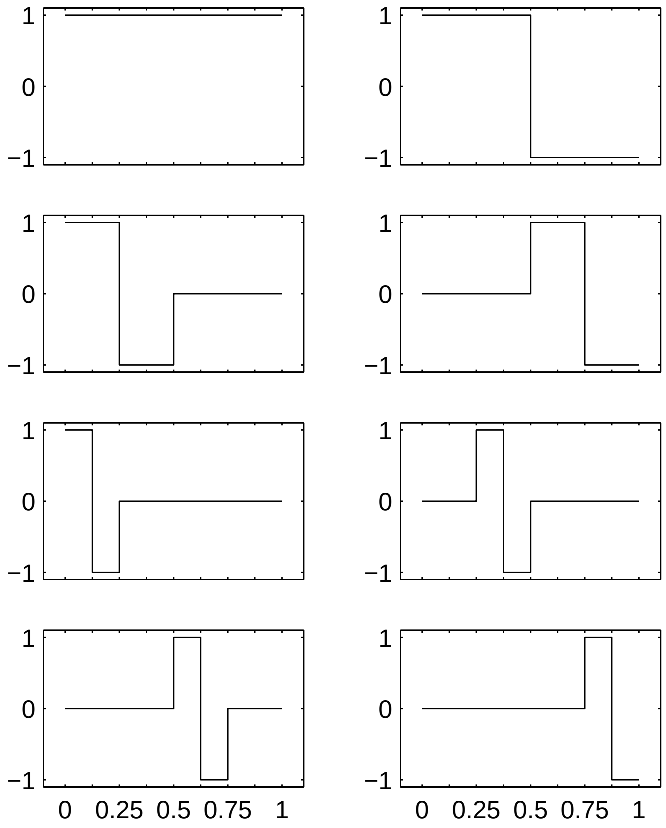
Interpretation 5

The result in matrix form for direct transform:

$$\mathbf{W}_{\mathrm{a}}^{(3)} = \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} & -\frac{1}{8} \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}.$$

Interpretation 6

Here is a graphical representation of the contents of $\mathbf{W}_a^{(3)}$:



Interpretation 7

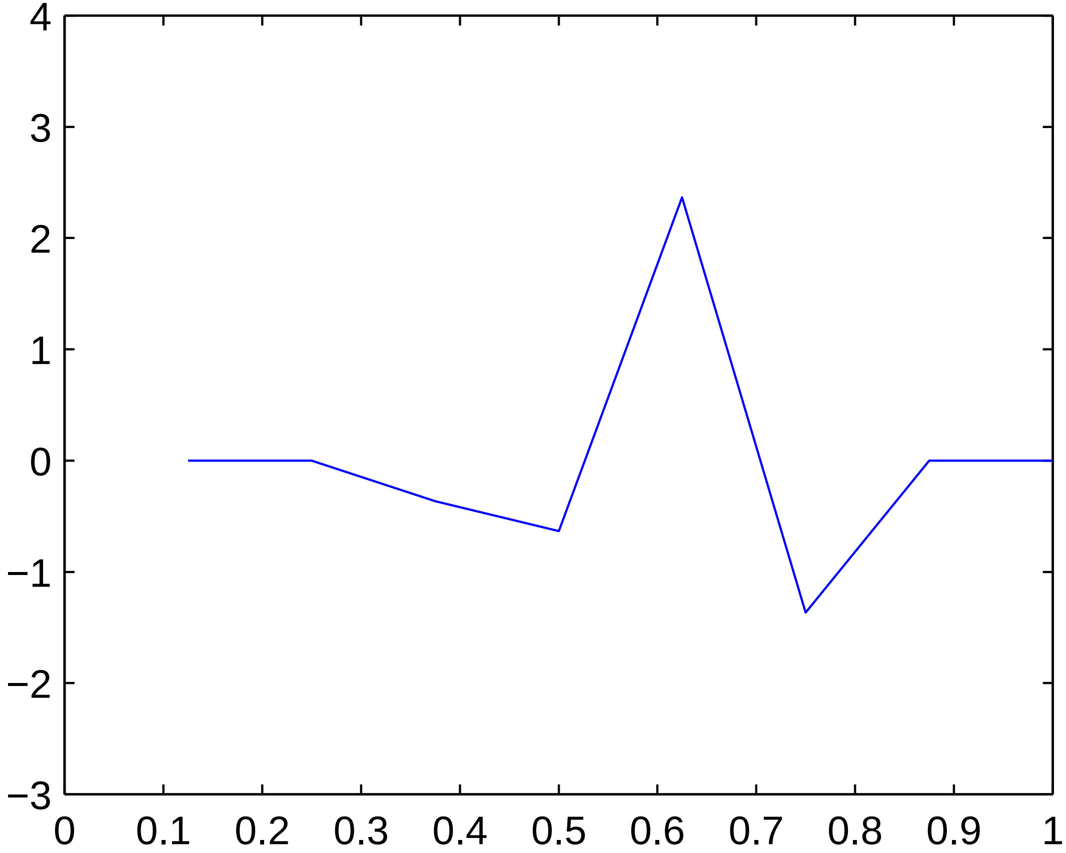
It is one of the nontrivial results in wavelet theory that there always are either 2 or 4 waveforms behind each DWT. These waveforms get scaled and translated. By reconstructing from signals with zeroes except a single 1, one can find these waveforms. Here is an example using the inverse of the Daubechies 4 transform. We take the inverse transform of a signal with a one at place 6, and take lengths 8, 32, 128, 512, and 2048. The result is shown on the next slide.

Interpretation 8

Iterations, signal lengths 8, 32, 128, 512, 2048.

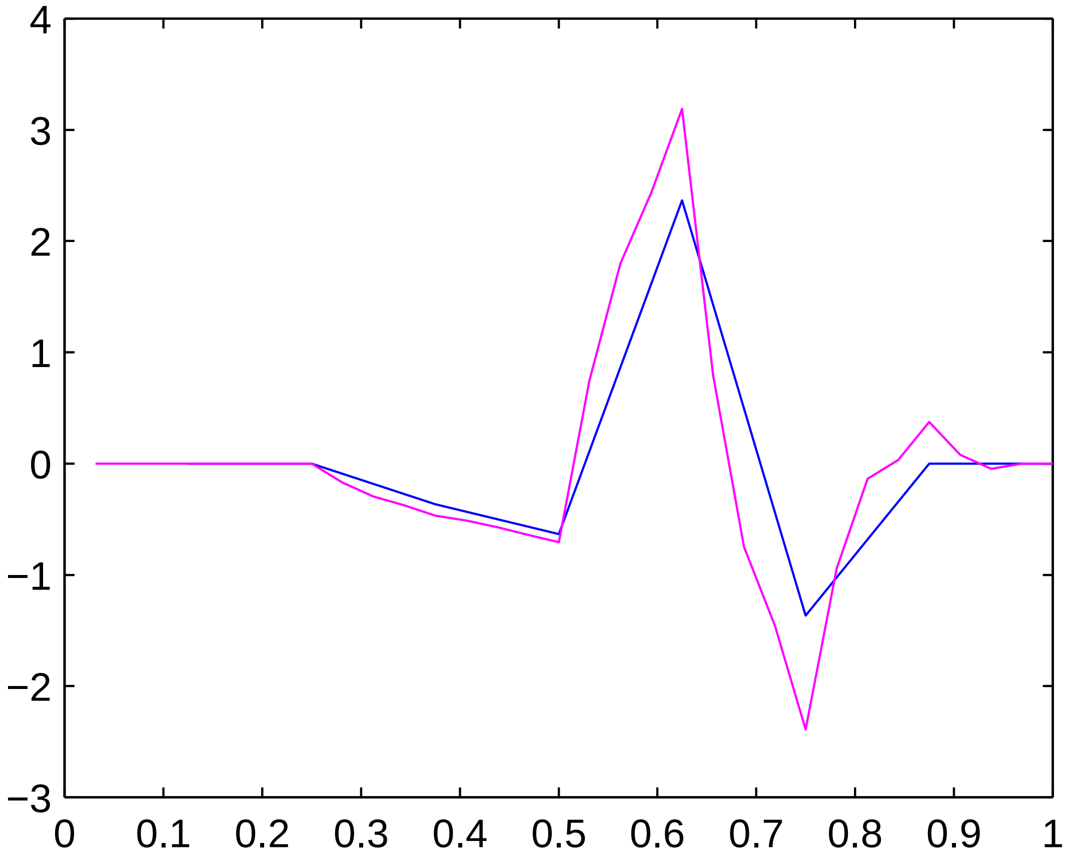
Interpretation 8

Iterations, signal lengths 8, 32, 128, 512, 2048.



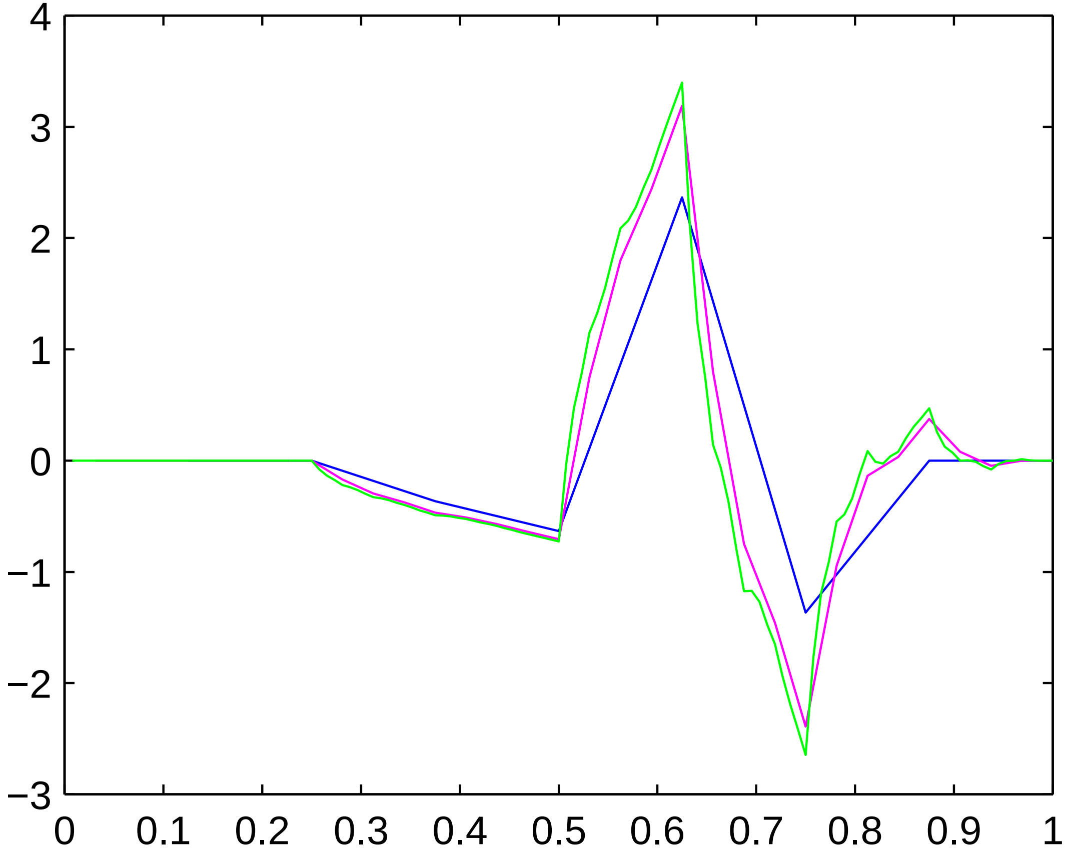
Interpretation 8

Iterations, signal lengths 8, 32, 128, 512, 2048.



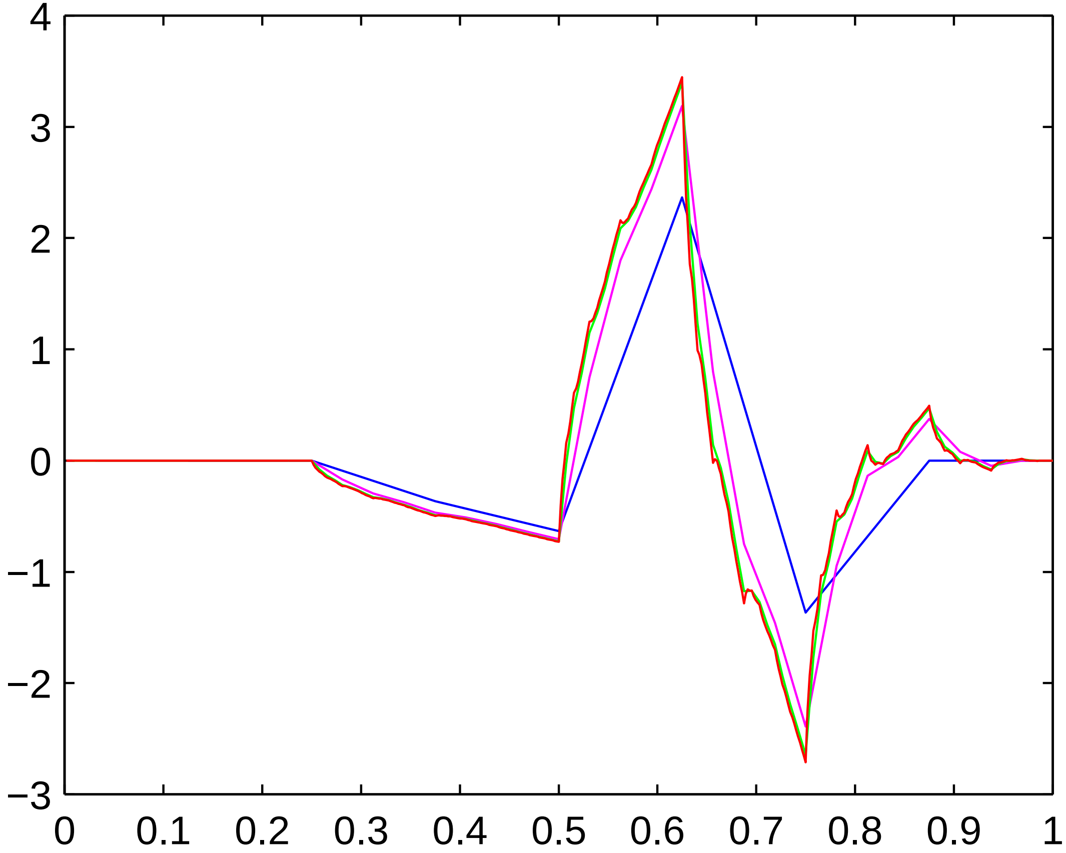
Interpretation 8

Iterations, signal lengths 8, 32, 128, 512, 2048.



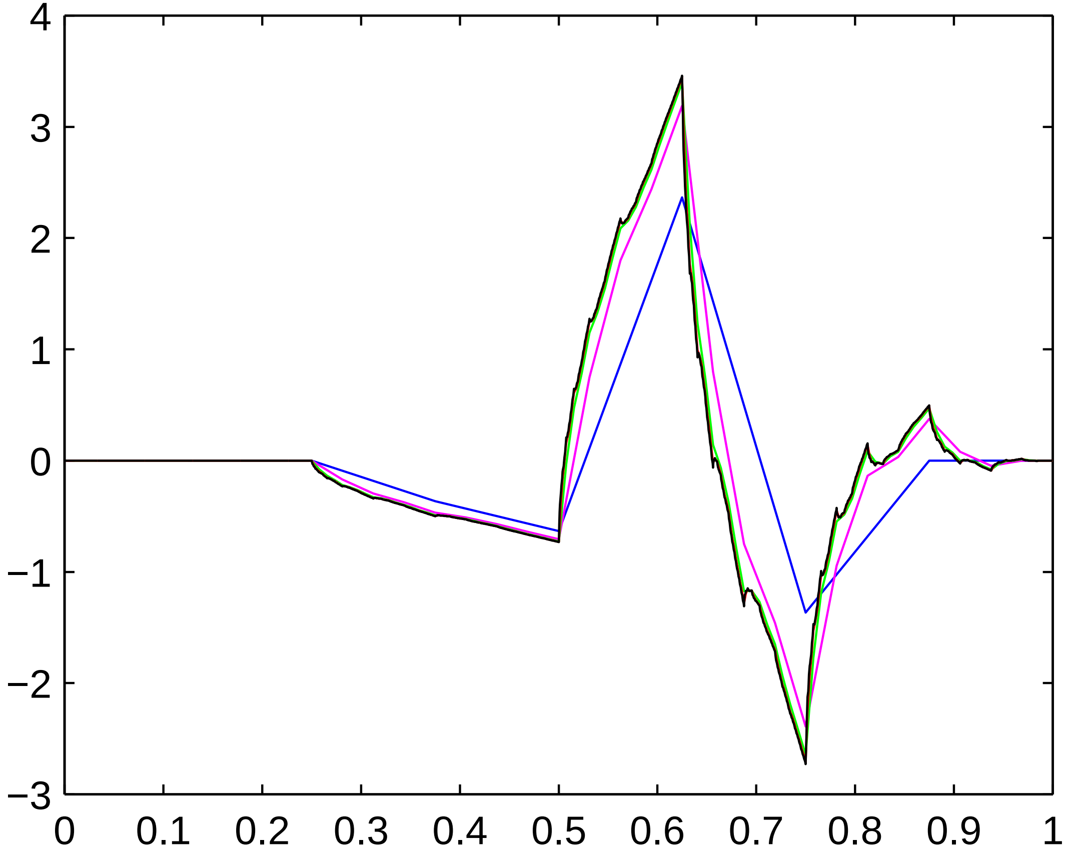
Interpretation 8

Iterations, signal lengths 8, 32, 128, 512, 2048.



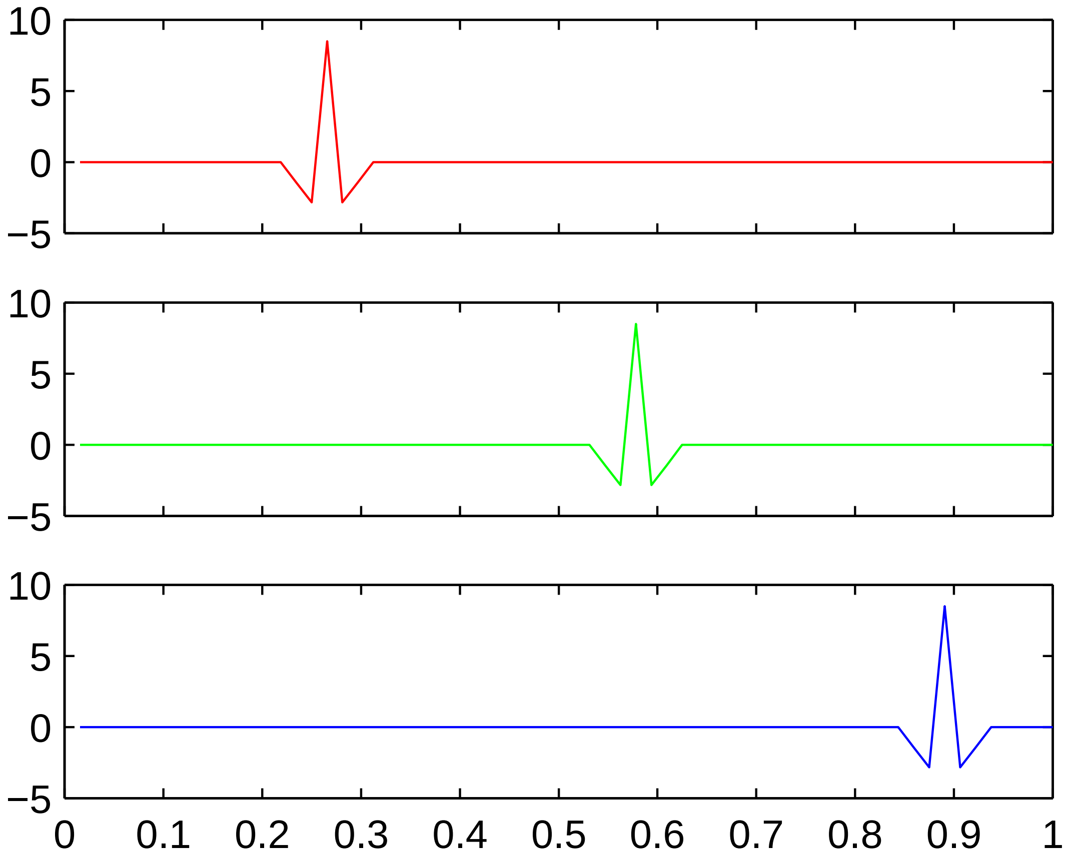
Interpretation 8

Iterations, signal lengths 8, 32, 128, 512, 2048.



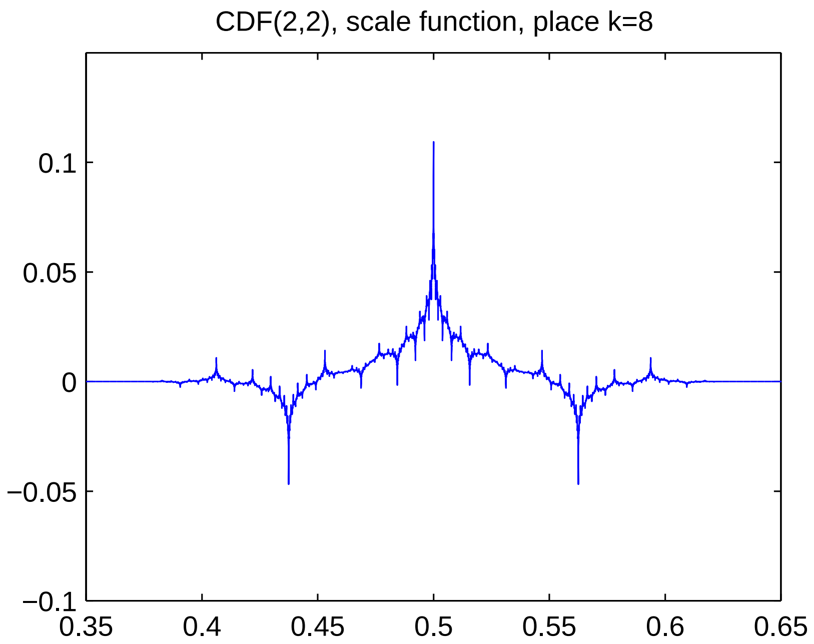
Interpretation 9

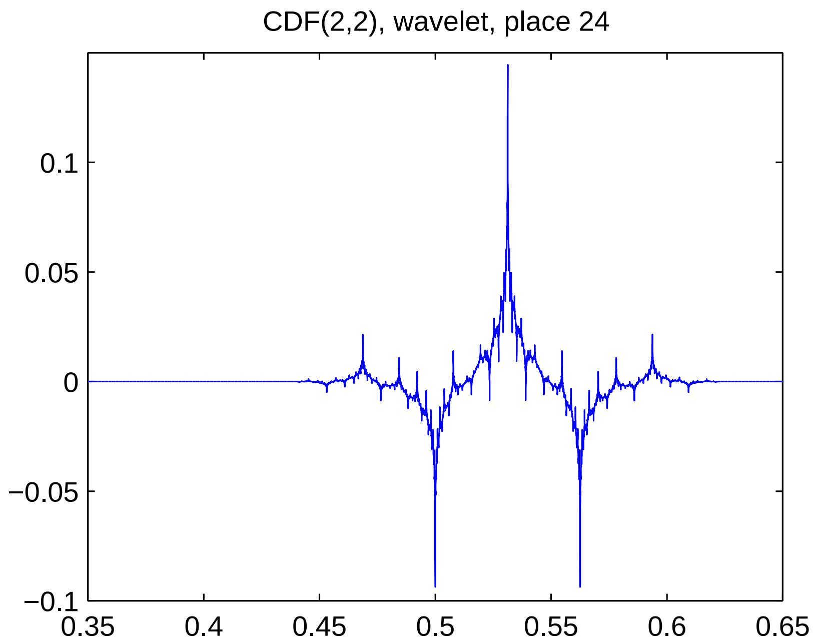
Another example: inverse CDF(2,2), signal length 64, 1 at positions 40, 50, and 60.



Interpretation 10

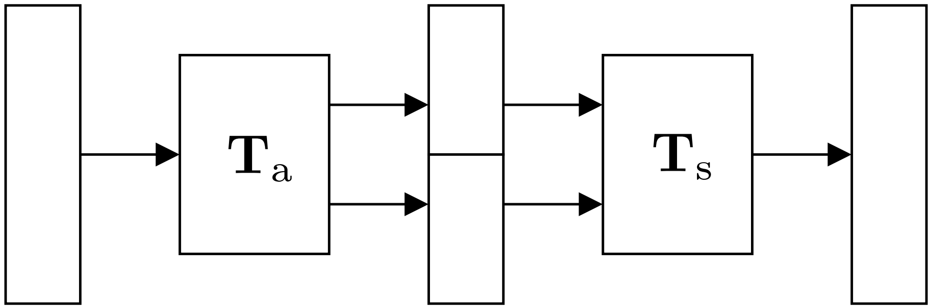
Example using direct CDF(2,2):





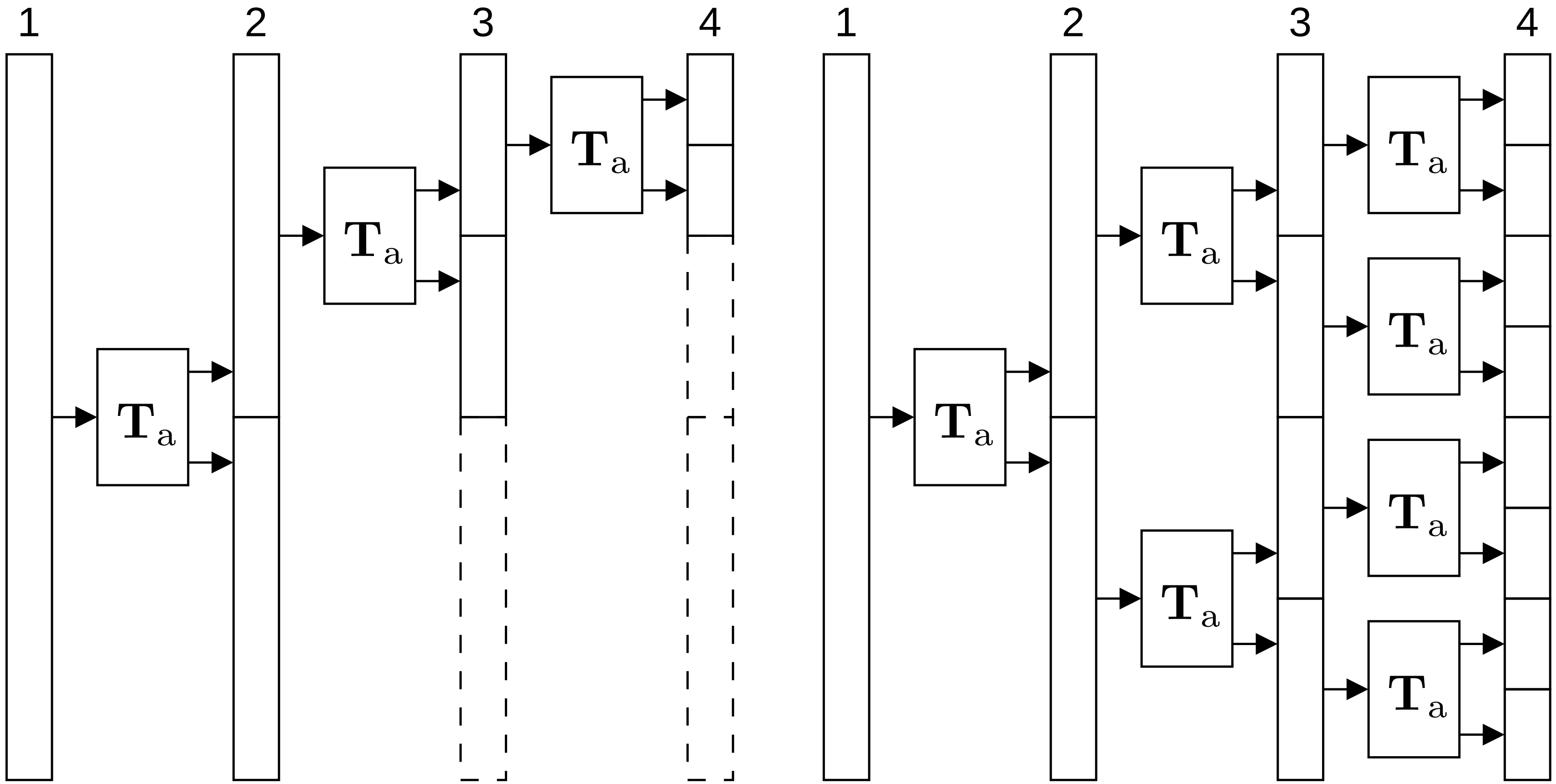
A generalization 1

We now present a generalization of the DWT to the Wavelet Packet Transform. Block diagram representation of one step DWT:



Note that we now put the average $\mathbf{s}$ components on the top, and the difference $\mathbf{d}$ components on the bottom, in this one step representation.

A generalization 2



(a) (b)

A generalization 3

Our first example, full decomposition:

A generalization 3

Our first example, full decomposition: Recall example

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
32	38	16	10	8	−8	0	12
35	−3	16	10	8	−8	0	12

A generalization 3

Our first example, full decomposition:

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
32	38	16	10	0	6	8	−6
35	−3	13	3	3	−3	1	7

A generalization 3

Our first example, full decomposition:

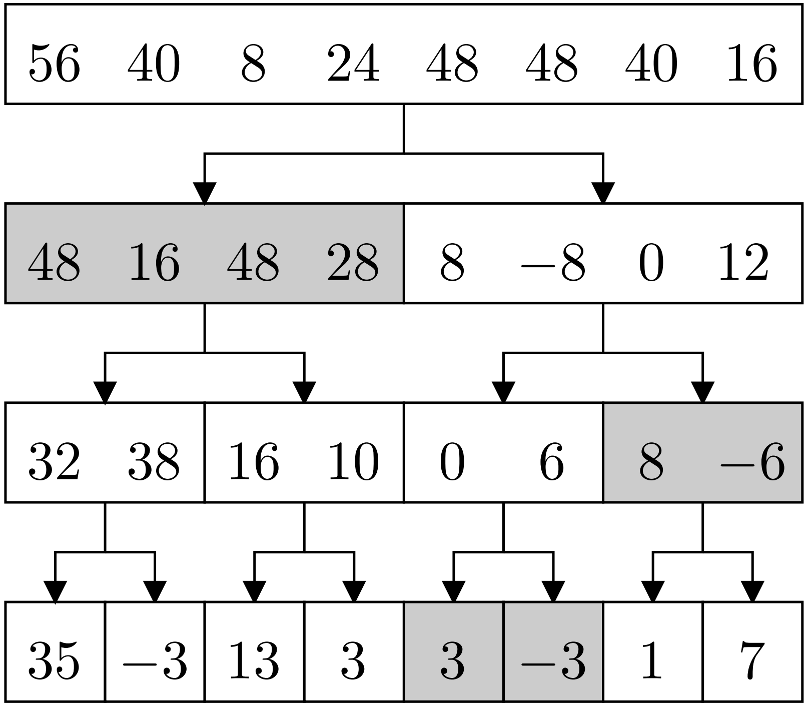
$$\frac{8+(-8)}{2}$$

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
32	38	16	10	0	6	8	−6
35	−3	13	3	3	−3	1	7

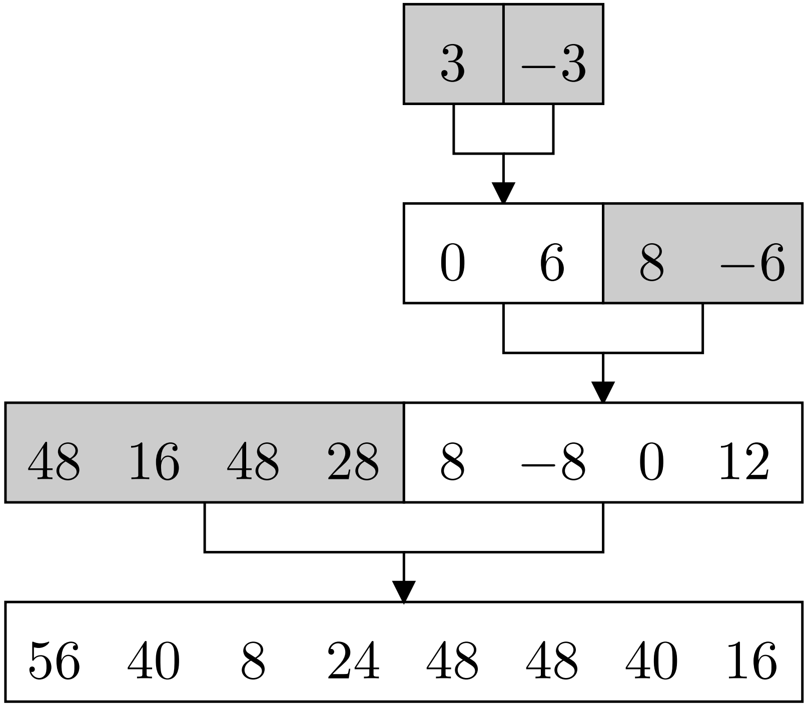
$$8-0$$

A generalization 4

Decomposition

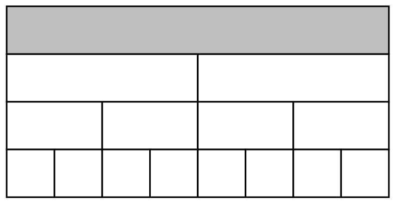


Reconstruction

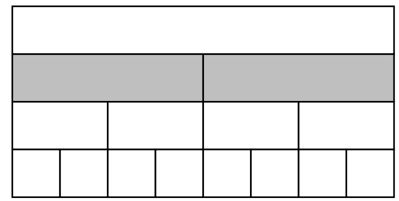


A generalization 5

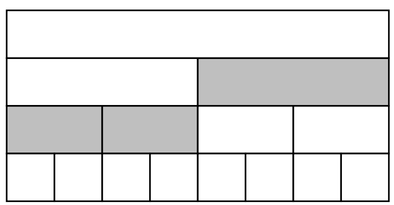
Possible representations of the signal:



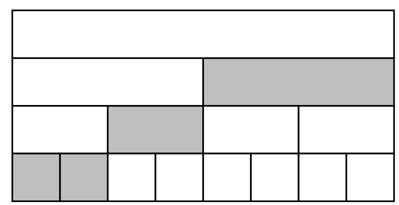
56 40 8 24 48 48 40 16



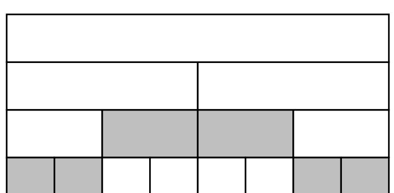
48 16 48 28 | 8 −8 0 12



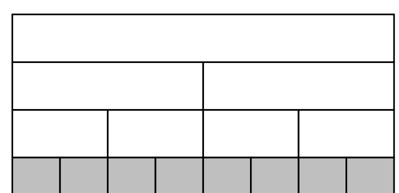
32 38 | 16 10 | 8 −8 0 12



35 | −3 | 16 10 | 8 −8 0 12



35 | −3 | 16 10 | 0 6 | 1 | 7



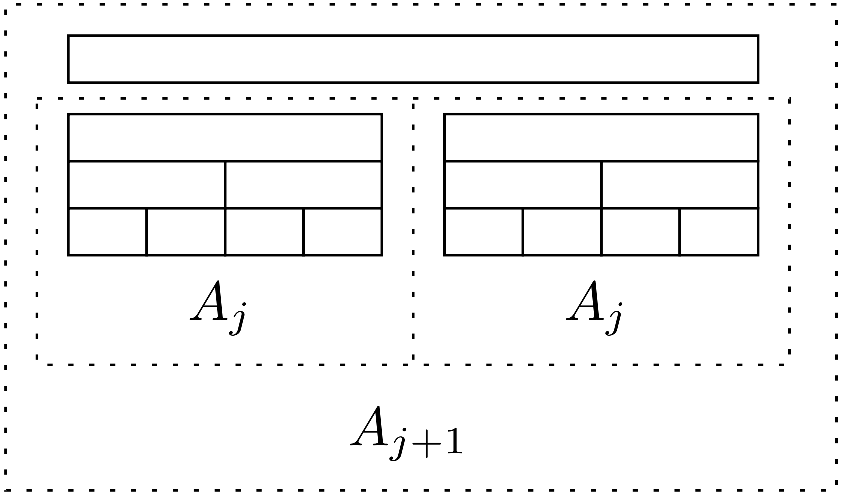
35 | −3 | 13 | 3 | 3 | −3 | 1 | 7

WPT complexity 1

The number of possible representations of a signal grows very fast with the number of decomposition steps. We have:

Number of levels	Minimum signal length	Number of bases
1	1	1
2	2	2
3	4	5
4	8	26
5	16	677
6	32	458330
7	64	210066388901
8	128	44127887745906175987802

WPT complexity 2



The number of possible decompositions of a signal using j levels is denoted by A_j. We have $A_{j+1} = 1 + A_j^2$. We have the estimate $2^{2^{j-1}} < A_j < 2^{2^j}$. Example $j = 10$: $2^{2^9} \approx 10^{154}$ and $2^{2^{10}} \approx 10^{308}$.

Best basis algorithm 1

Solution to complexity problem is the best basis algorithm. This is a very flexible algorithm, based on a cost function. A cost function is denoted by $\mathcal{K}$. It maps a finite length signal $\mathbf{a}$ to a number $\mathcal{K}(\mathbf{a})$. $[\mathbf{a}\,\mathbf{b}]$ denotes the concatenation of two signals $\mathbf{a}$ and $\mathbf{b}$. We require two properties:

- $\mathcal{K}(\mathbf{0}) = 0$

Best basis algorithm 1

Solution to complexity problem is the best basis algorithm. This is a very flexible algorithm, based on a cost function. A cost function is denoted by $\mathcal{K}$. It maps a finite length signal $\mathbf{a}$ to a number $\mathcal{K}(\mathbf{a})$. $[\mathbf{a}\,\mathbf{b}]$ denotes the concatenation of two signals $\mathbf{a}$ and $\mathbf{b}$. We require two properties:

- $\mathcal{K}(\mathbf{0}) = 0$
- $\mathcal{K}([\mathbf{a}\,\mathbf{b}]) = \mathcal{K}(\mathbf{a}) + \mathcal{K}(\mathbf{b})$

Best basis algorithm 1

Solution to complexity problem is the best basis algorithm. This is a very flexible algorithm, based on a cost function. A cost function is denoted by $\mathcal{K}$. It maps a finite length signal $\mathbf{a}$ to a number $\mathcal{K}(\mathbf{a})$. $[\mathbf{a}\,\mathbf{b}]$ denotes the concatenation of two signals $\mathbf{a}$ and $\mathbf{b}$. We require two properties:

- $\mathcal{K}(\mathbf{0}) = 0$
- $\mathcal{K}([\mathbf{a}\,\mathbf{b}]) = \mathcal{K}(\mathbf{a}) + \mathcal{K}(\mathbf{b})$

An example: $\mathcal{K}(\mathbf{a}) =$ number of nonzero entries in $\mathbf{a}$.

$$\begin{aligned}5 &= \mathcal{K}([1,\ 0,\ -1,\ 22,\ 0,\ 0,\ 2,\ -7]\\ &= \mathcal{K}([1,\ 0,\ -1,\ 22]) + \mathcal{K}([0,\ 0,\ 2,\ -7]) = 3 + 2\end{aligned}$$

Best basis algorithm 2

Cost functions

Threshold $\mathcal{K}_{\mathrm{thres}}(\mathbf{a})$ equals number of elements in $\mathbf{a}$ with absolute value greater than the threshold ε. Example:

$$\varepsilon = 2.0: \quad \mathcal{K}_{\mathrm{thres}}([1,\ 2,\ 3\,0,\ -1,\ -4]) = 2$$

$$\varepsilon = 1.0: \quad \mathcal{K}_{\mathrm{thres}}([1,\ 2,\ 3\,0,\ -1,\ -4]) = 3$$

$$\varepsilon = 0.5: \quad \mathcal{K}_{\mathrm{thres}}([1,\ 2,\ 3\,0,\ -1,\ -4]) = 5$$

Problem: Look out for rescaling hidden in transforms.

Best basis algorithm 3

Cost functions

ℓ^p-norm

Notation: $\mathbf{a} = \{a[n]\}$, $0 < p < \infty$ (useful values are $0 < p < 2$)

$$\mathcal{K}_{\ell^p}(\mathbf{a}) = \sum_n |a[n]|^p.$$

Note that for $p = 2$ this is the energy in the signal.

Shannon entropy

$$\mathcal{K}_{\text{Shannon}}(\mathbf{a}) = \sum_n |a[n]|^2 \log(|a[n]|^2)$$

Best basis algorithm 4

The best basis algorithm through the first example. Do a full decomposition. Result is:

56	40	8	24	48	48	40	16
48	16	48	28	8	−8	0	12
32	38	16	10	0	6	8	−6
35	−3	13	3	3	−3	1	7

Cost function: Number of entries with absolute value > 1. Compute cost of each vector in full decomposition:

Best basis algorithm 5

<table>
<tr><td colspan="8">8</td></tr>
<tr><td colspan="4">4</td><td colspan="4">3</td></tr>
<tr><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2">2</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Cost values are computed, and components are marked with cost values.

Best basis algorithm 5

8							
4				3			
2		2		1		2	
1	1	1	1	1	1	0	1

Last row is marked. Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary.

Best basis algorithm 5

$$2 = 1 + 1$$

<table>
<tr><td colspan="8">8</td></tr>
<tr><td colspan="4">4</td><td colspan="4">3</td></tr>
<tr><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2">2</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary.

Best basis algorithm 5

<table>
<tr><td colspan="8">8</td></tr>
<tr><td colspan="4">4</td><td colspan="4">3</td></tr>
<tr><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2">2</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary.

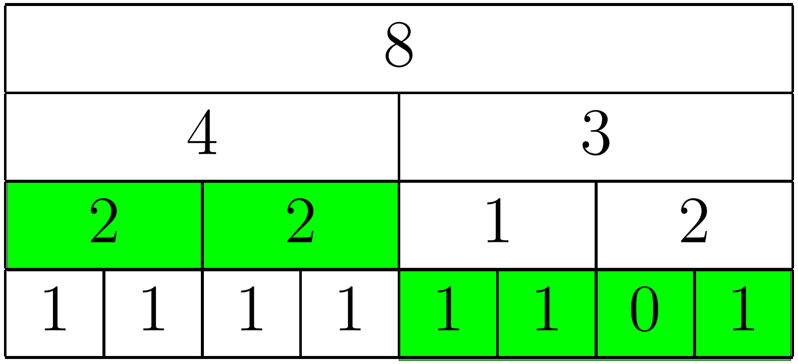
Best basis algorithm 5

$$2 = 1 + 1$$

<table>
<tr><td colspan="8">8</td></tr>
<tr><td colspan="4">4</td><td colspan="4">3</td></tr>
<tr><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2">2</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary.

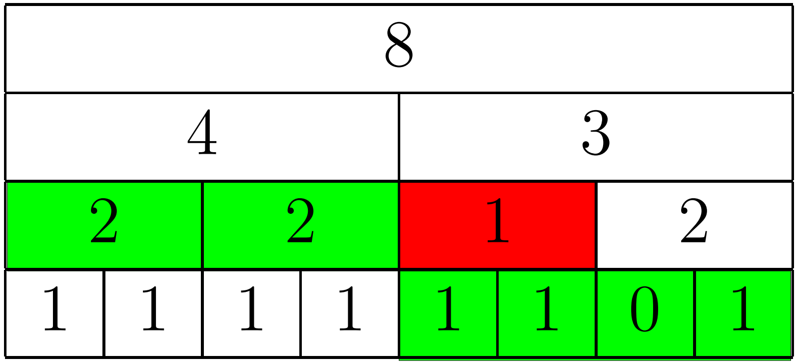
Best basis algorithm 5



Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary.

Best basis algorithm 5

$$1 < 1 + 1$$



Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary.

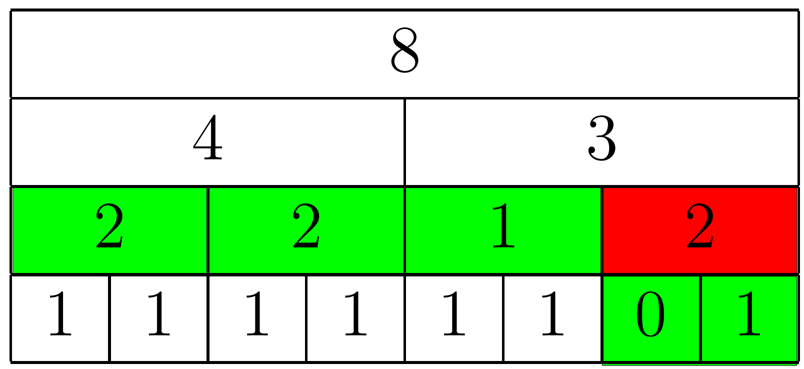
Best basis algorithm 5

<table>
<tr><td colspan="8" align="center">8</td></tr>
<tr><td colspan="4" align="center">4</td><td colspan="4" align="center">3</td></tr>
<tr><td colspan="2" align="center">2</td><td colspan="2" align="center">2</td><td colspan="2" align="center">1</td><td colspan="2" align="center">2</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary.

Best basis algorithm 5

$$2 > 0 + 1$$



Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary. If lower component is cheaper, keep, and replace cost value above with total cost of components kept.

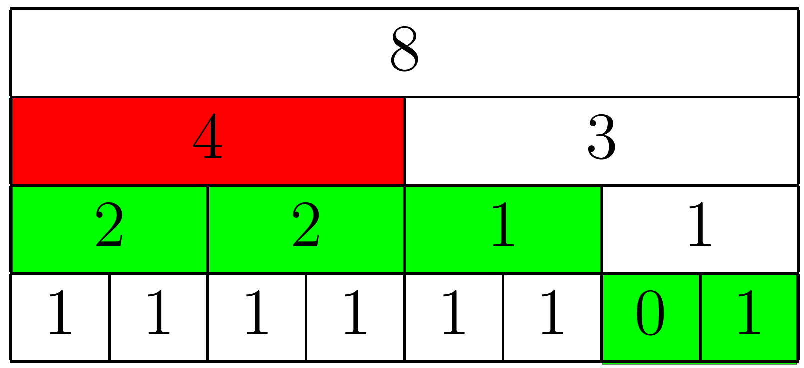
Best basis algorithm 5

<table>
<tr><td colspan="8">8</td></tr>
<tr><td colspan="4">4</td><td colspan="4">3</td></tr>
<tr><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2">1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary. If lower component is cheaper, keep, and replace cost value above with total cost of components kept.

Best basis algorithm 5

$$4 = 2 + 2$$



Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary. If lower component is cheaper, keep, and replace cost value above with total cost of components kept.

Best basis algorithm 5

<table>
<tr><td colspan="8">8</td></tr>
<tr><td colspan="4">4</td><td colspan="4">3</td></tr>
<tr><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2">1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary. If lower component is cheaper, keep, and replace cost value above with total cost of components kept.

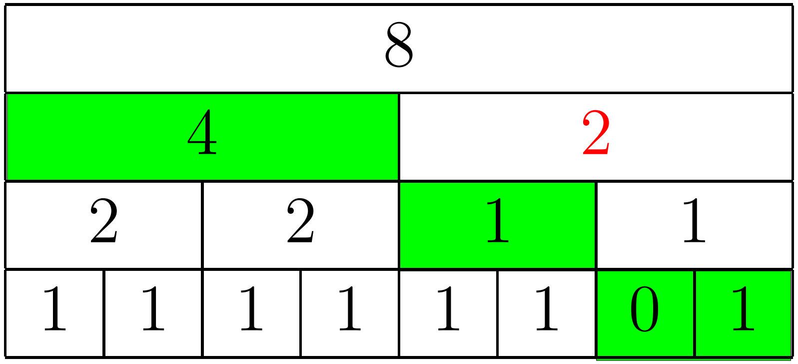
Best basis algorithm 5

$3 > 1 + 1$

<table>
<tr><td colspan="8">8</td></tr>
<tr><td colspan="4">4</td><td colspan="4">3</td></tr>
<tr><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2">1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary. If lower component is cheaper, keep, and replace cost value above with total cost of components kept.

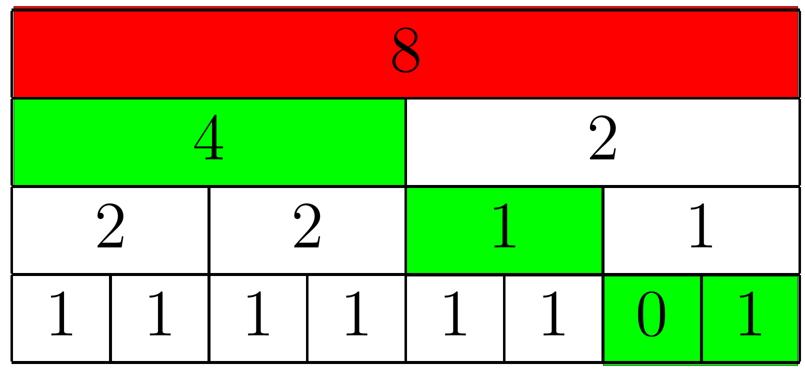
Best basis algorithm 5



Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary. If lower component is cheaper, keep, and replace cost value above with total cost of components kept.

Best basis algorithm 5

$$8 > 4 + 2$$



Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary. If lower component is cheaper, keep, and replace cost value above with total cost of components kept.

Best basis algorithm 5

<table>
<tr><td colspan="8">6</td></tr>
<tr><td colspan="4">4</td><td colspan="4">2</td></tr>
<tr><td colspan="2">2</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2">1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
</table>

Compare cost of a pair of elements with the one just above. In case of lower or equal cost, move up. Adjust marking, if necessary. If lower component is cheaper, keep, and replace cost value above with total cost of components kept.

Best basis algorithm 6

Some things to note:

- The best basis is not unique.

Best basis algorithm 6

Some things to note:

- The best basis is not unique.
- A best basis with all components at the same level is called a best level basis.

Best basis algorithm 6

Some things to note:

- The best basis is not unique.
- A best basis with all components at the same level is called a best level basis.
- With J levels the search algorithm is of order $O(J \log J)$. The full decomposition and the costs have to be computed only once.

Best basis algorithm 6

Some things to note:

- The best basis is not unique.
- A best basis with all components at the same level is called a best level basis.
- With J levels the search algorithm is of order $O(J \log J)$. The full decomposition and the costs have to be computed only once.
- The size of the tree to be searched is independent of the length of the signal.

Time and frequency 1

Discrete signal with finite energy

$$\mathbf{x} = \{x[n]\}_{n\in\mathbf{Z}}, \qquad \sum_{n\in\mathbf{Z}} |x[n]|^2 < \infty$$

Frequency contents ($j = \sqrt{-1}$):

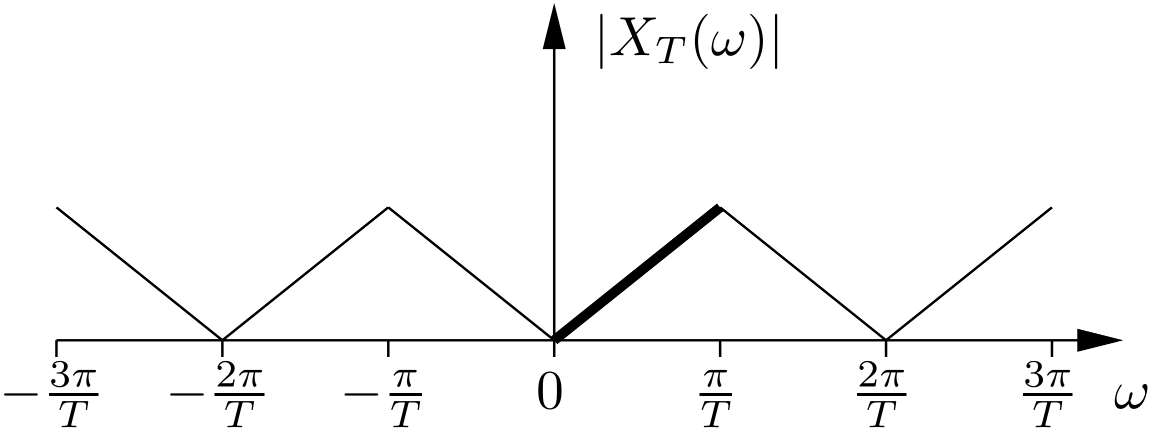
$$X(\omega) = \sum_n x[n]e^{-jn\omega},$$

or with period T, ie n corresponds to sampling time nT,

$$X_T(\omega) = \sum_n x[n]e^{-jnT\omega}.$$

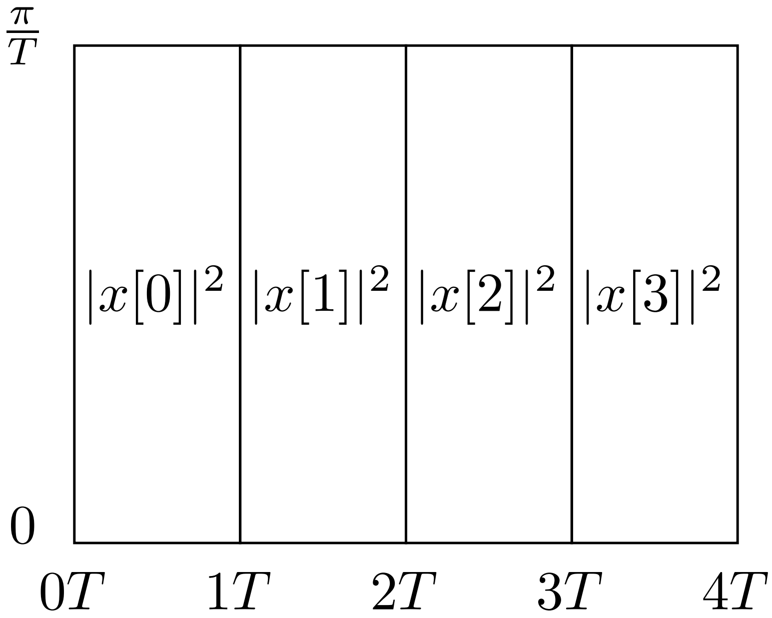
Time and frequency 2

For a real signal $\overline{X_T}(\omega) = X_T(-\omega)$. Frequency contents in any interval $[k\pi/T, (k+1)\pi/T]$.



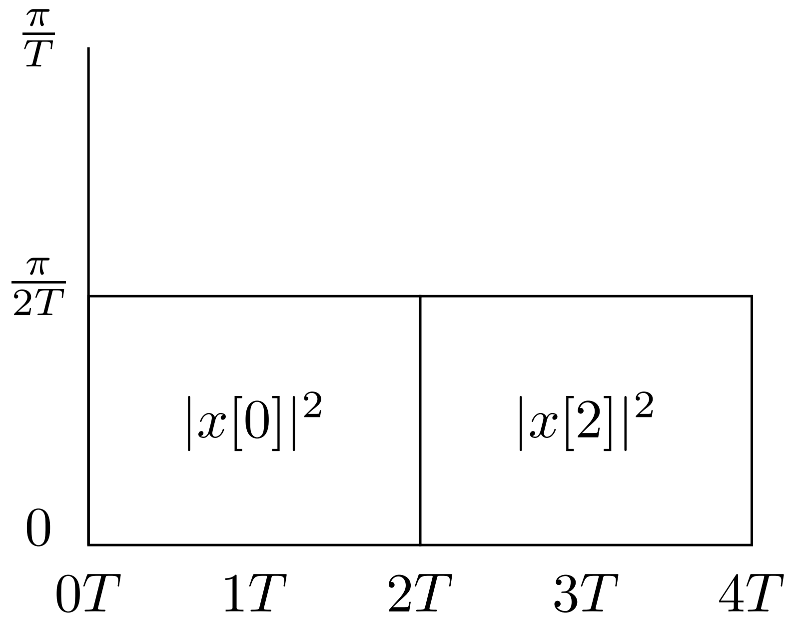
Time and frequency 3

Discrete signal $x[0], x[1], x[2], x[3]$, frequency interval $[0, \pi/T]$.

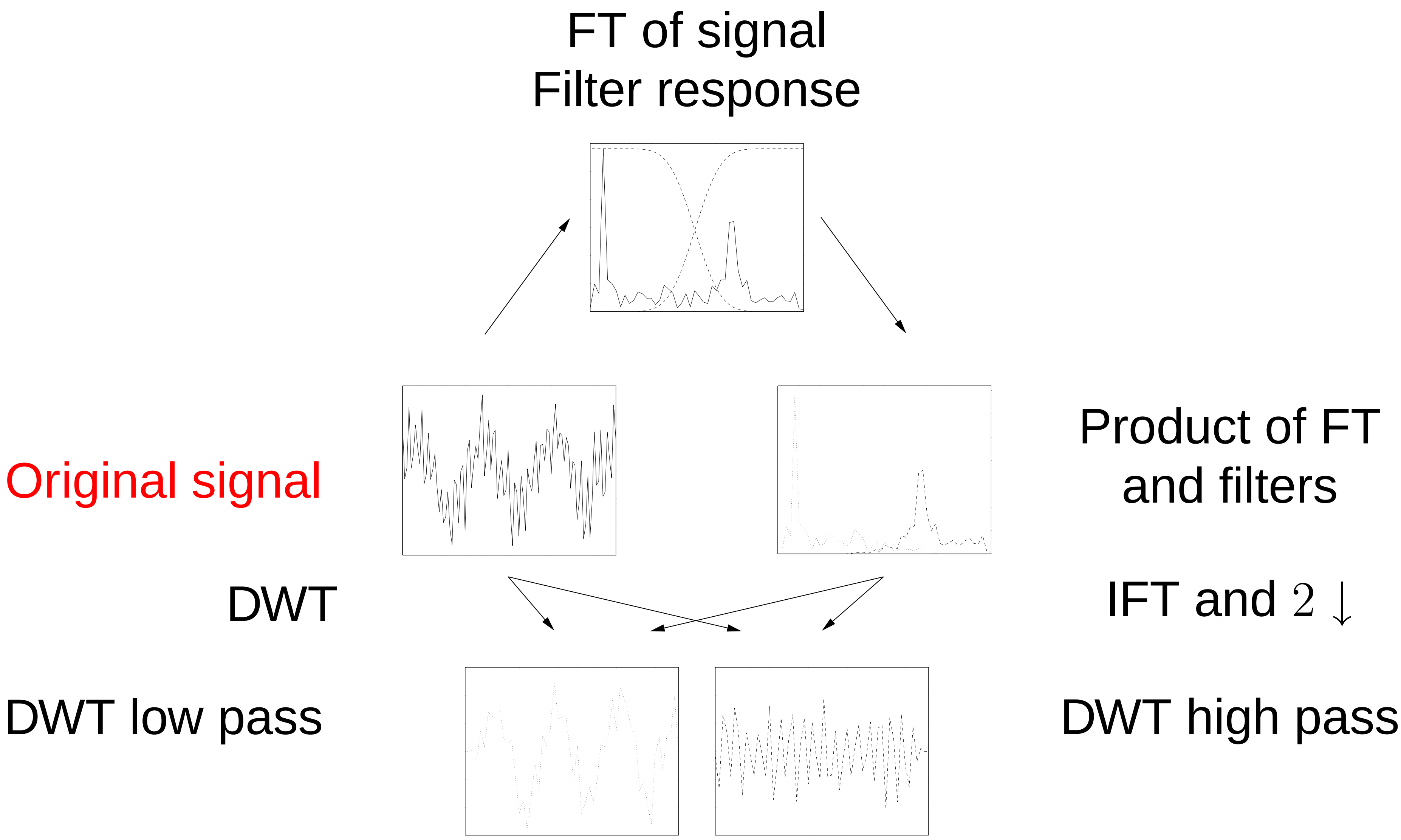


Time and frequency 4

Same signal downsampled by 2, frequency interval $[0, \pi/2T]$.

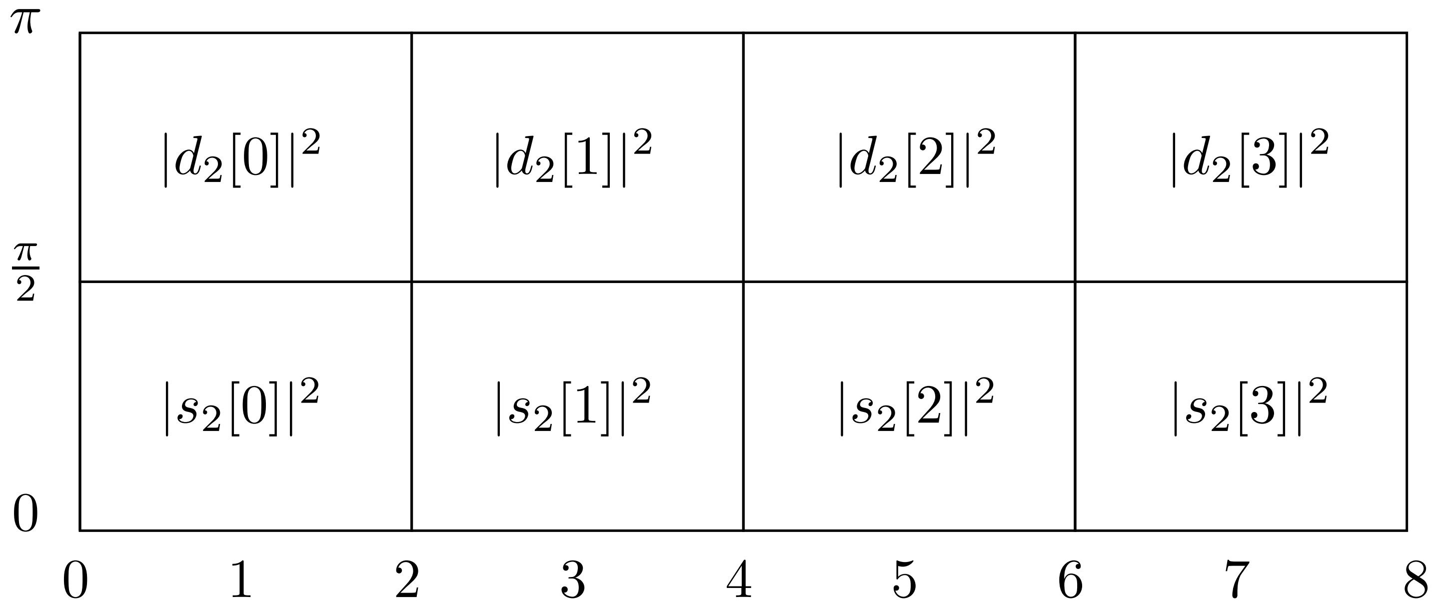


Time and frequency 5



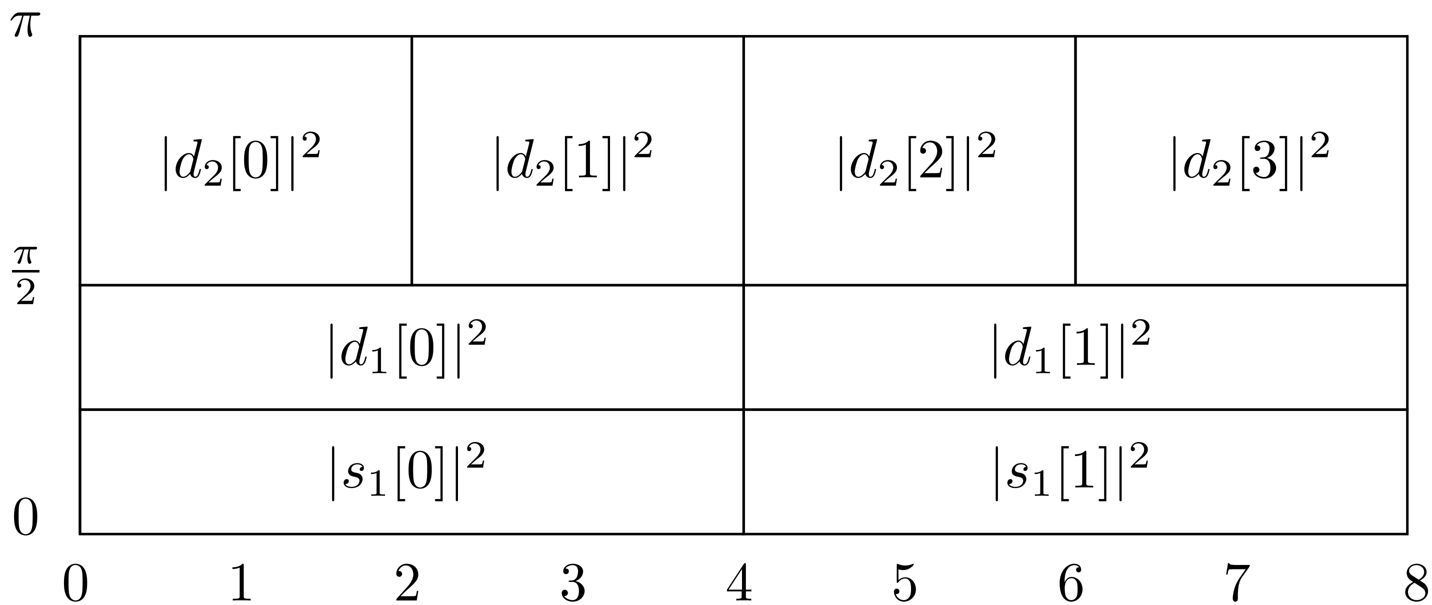
Time and frequency 6

One step DWT, eight samples. Energy distribution.



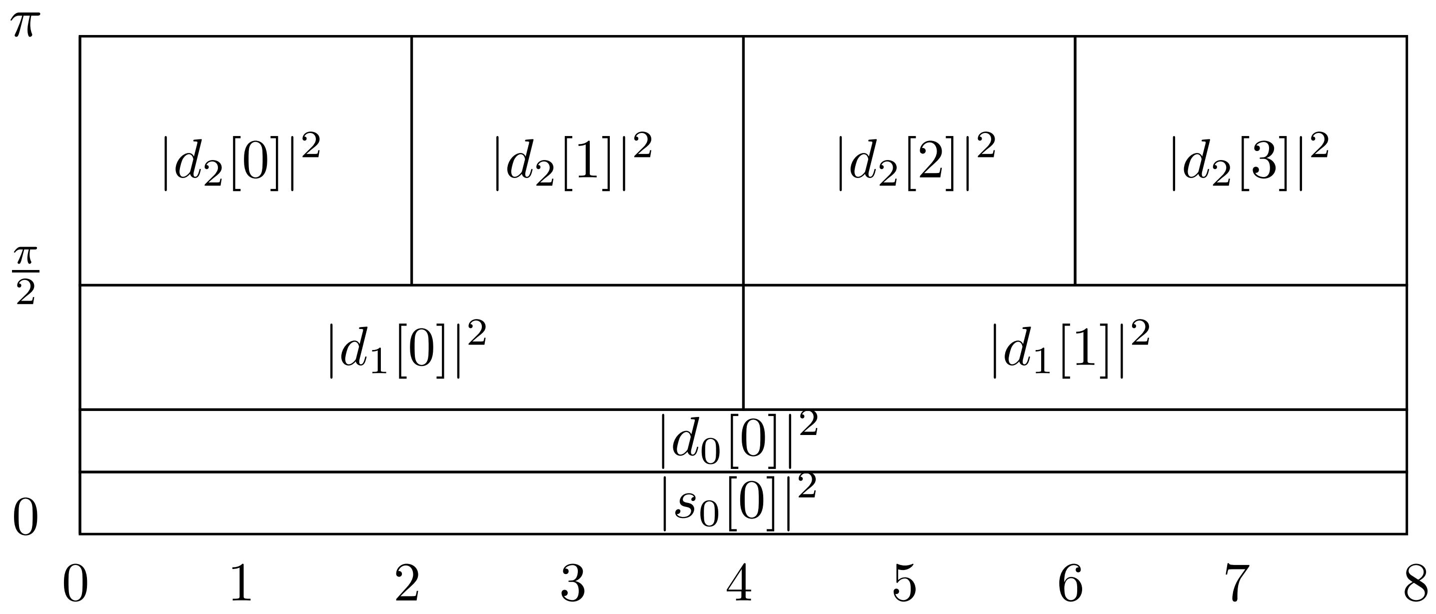
Time and frequency 7

Two step DWT, eight samples. Energy distribution.



Time and frequency 8

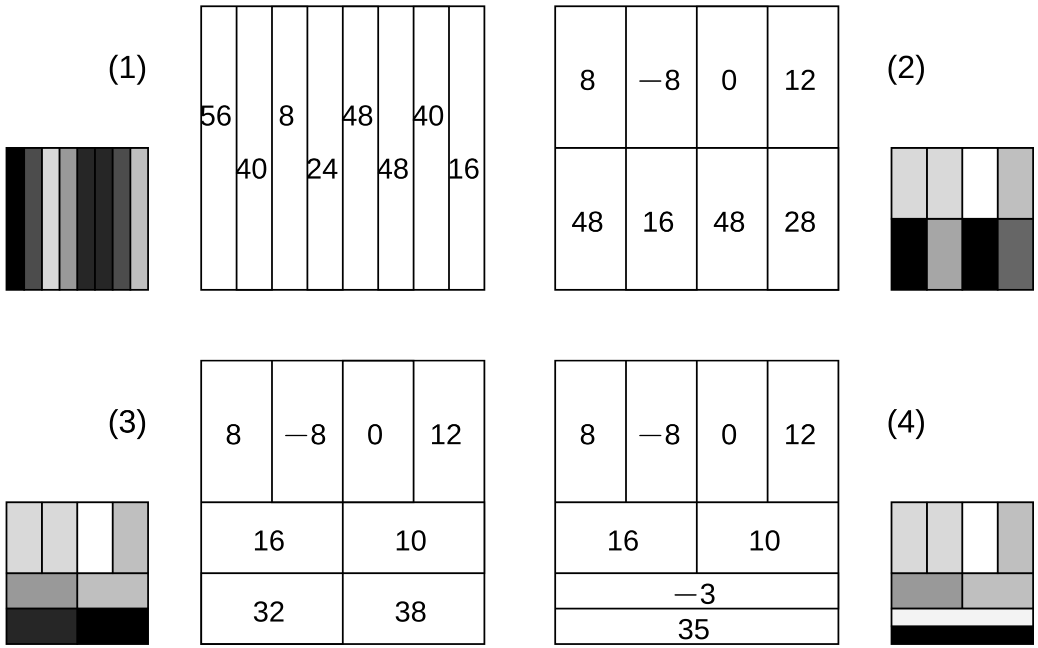
Three step DWT, eight samples. Energy distribution.



Time and frequency 9

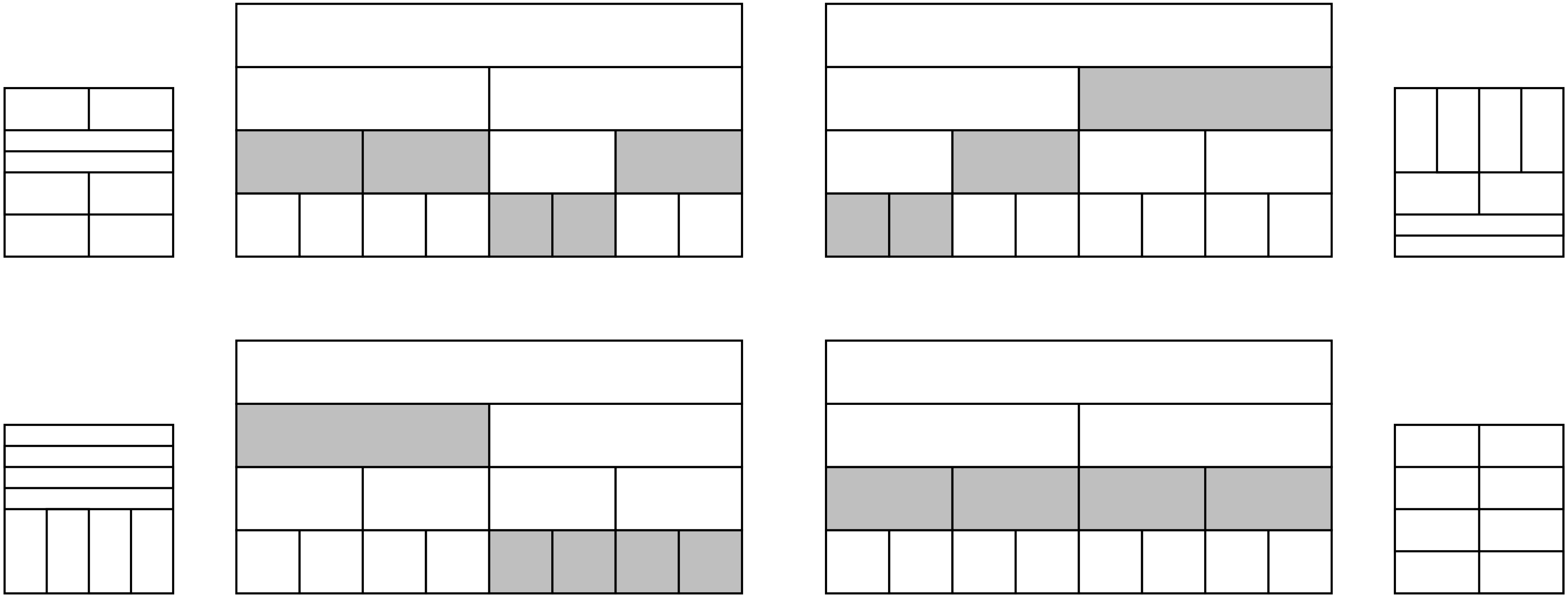
The first example, again:

(1)	56	40	8	24	48	48	40	16
(2)	48	16	48	28	8	−8	0	12
(3)	32	38	16	10	8	−8	0	12
(4)	35	−3	16	10	8	−8	0	12



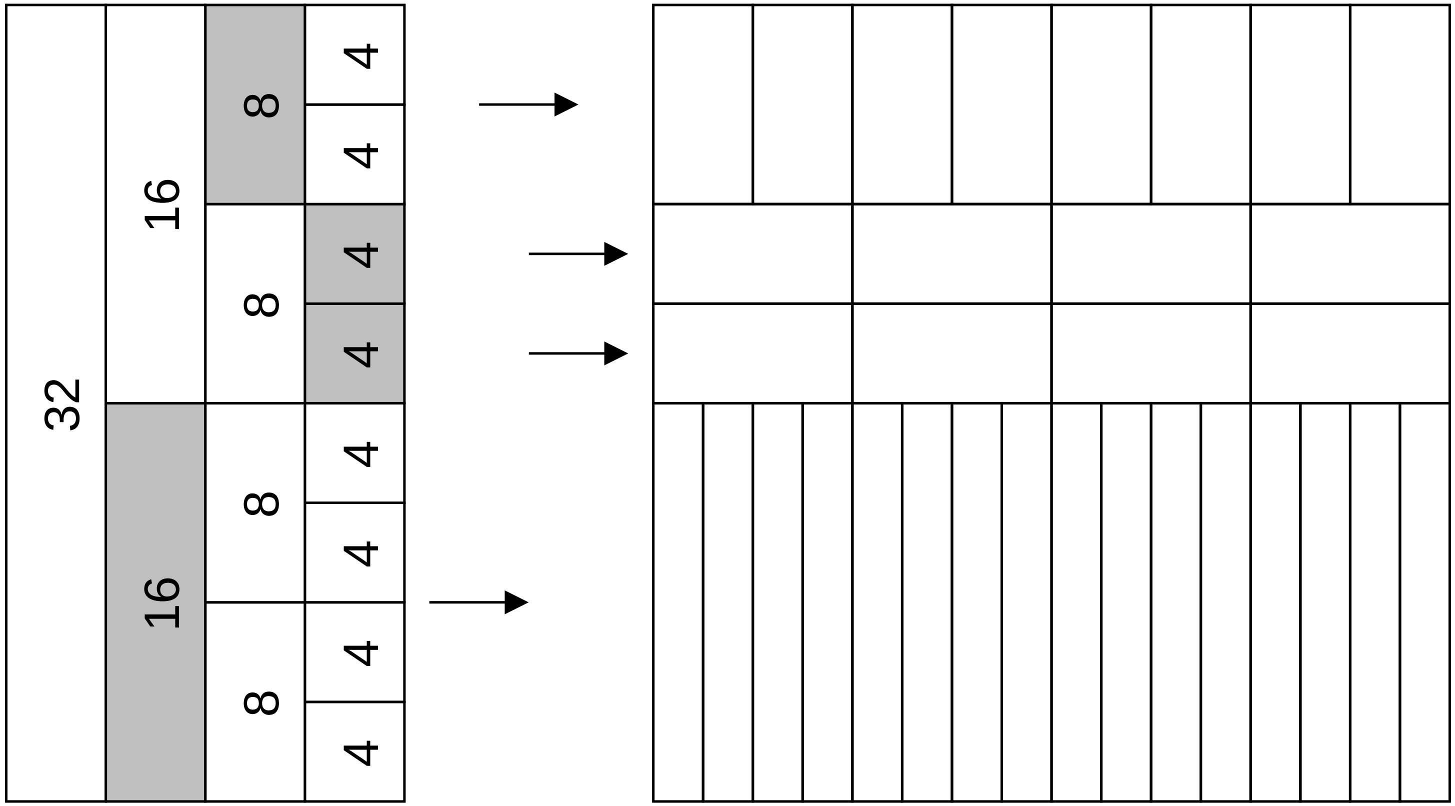
Time and frequency 10

More examples:



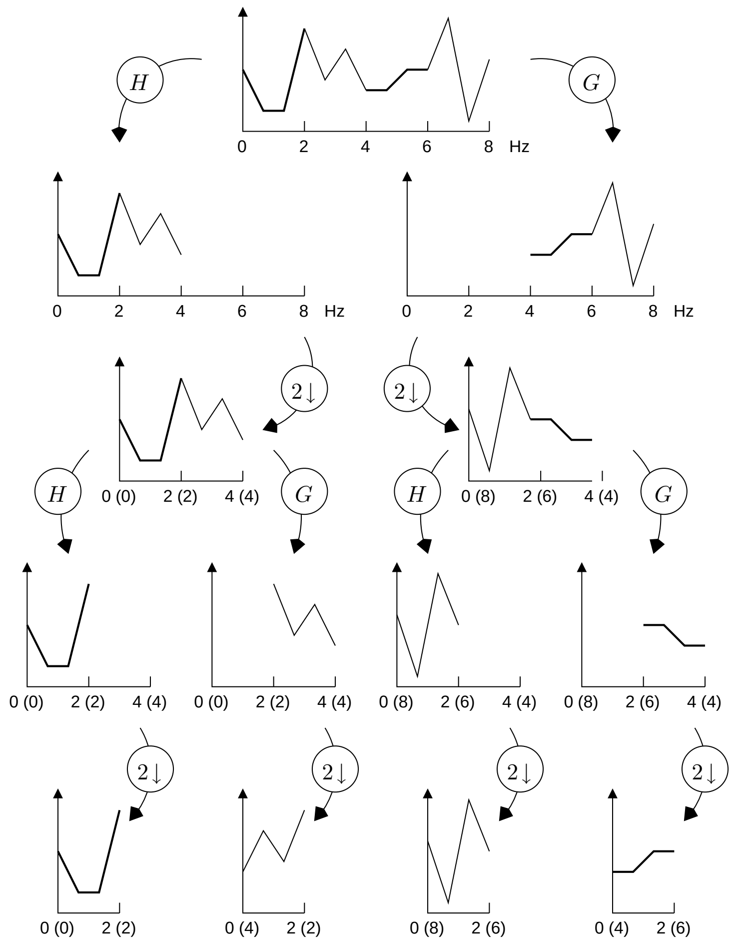
Time and frequency 11

Explanation for previous example:

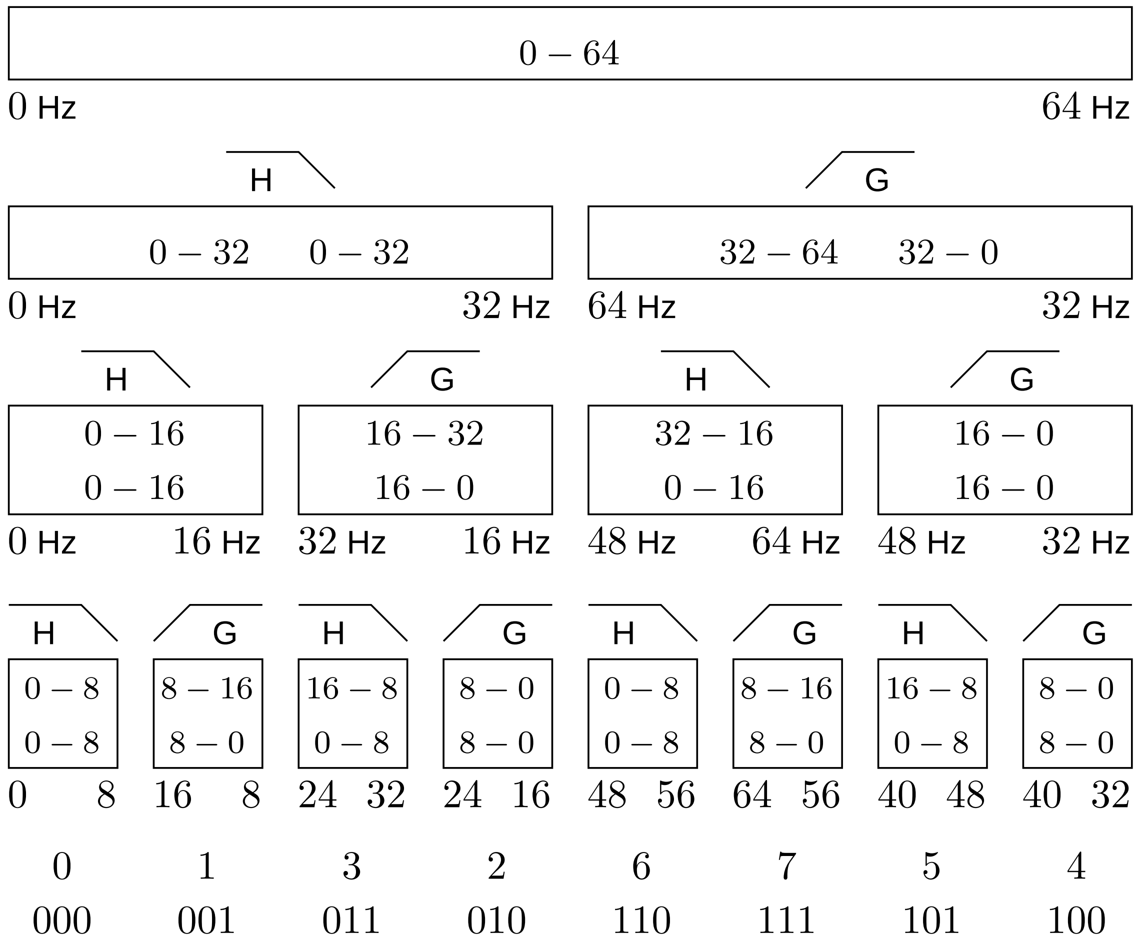


Time and frequency 12

Frequency contents in WP decomposition, ideal filters:

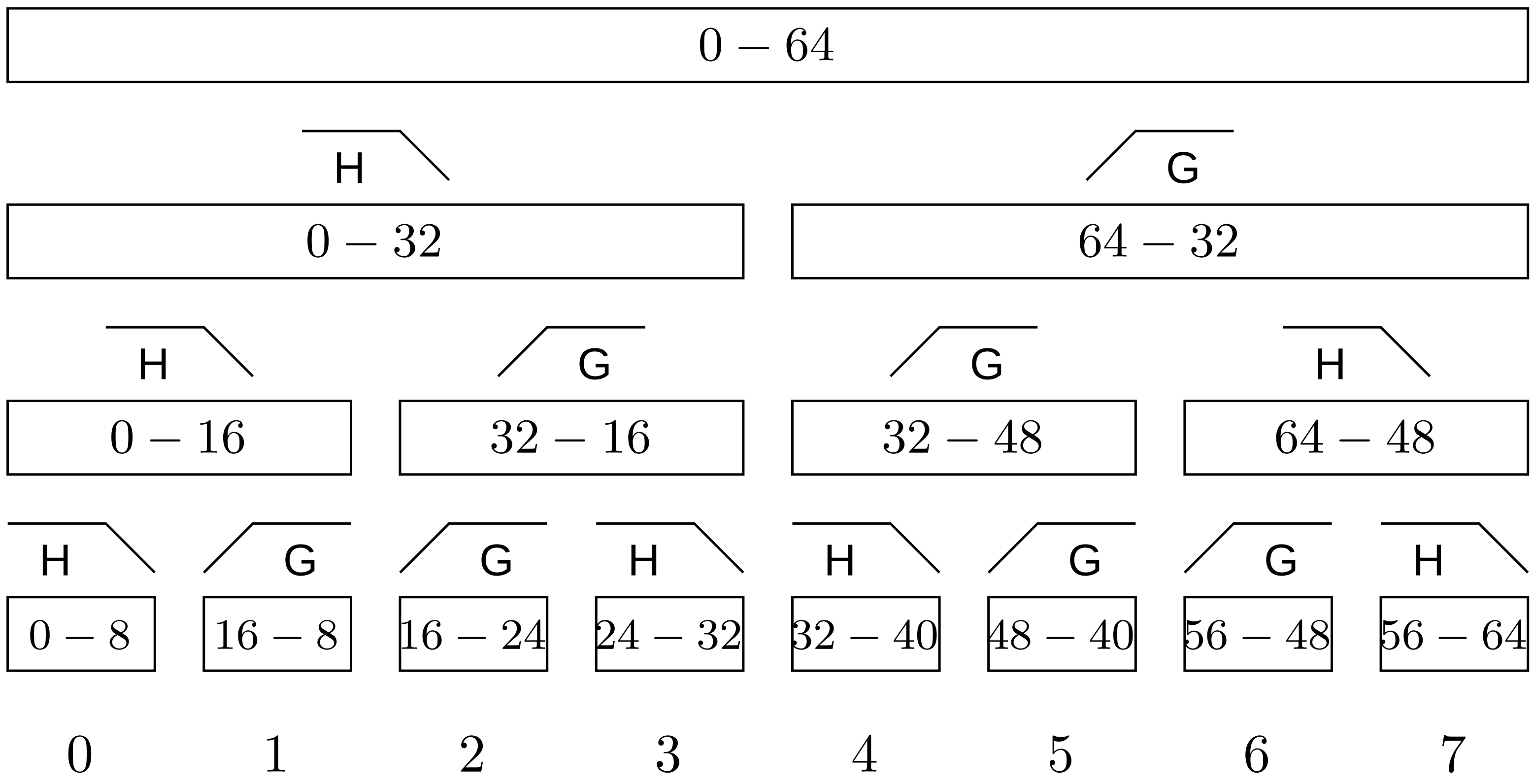


Time and frequency 13



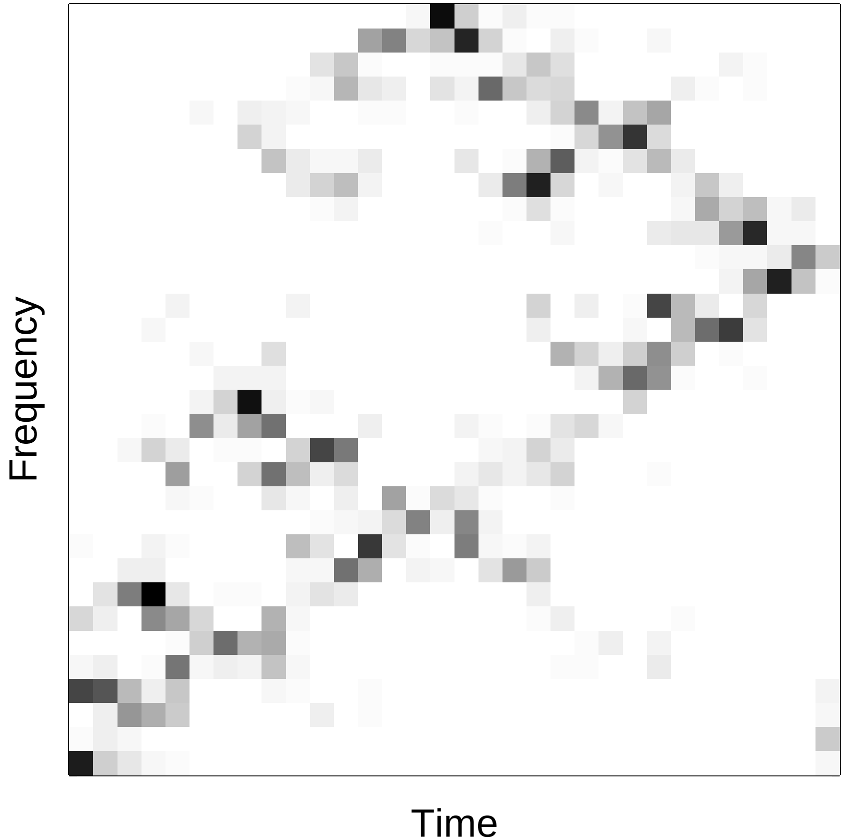
Time and frequency 14

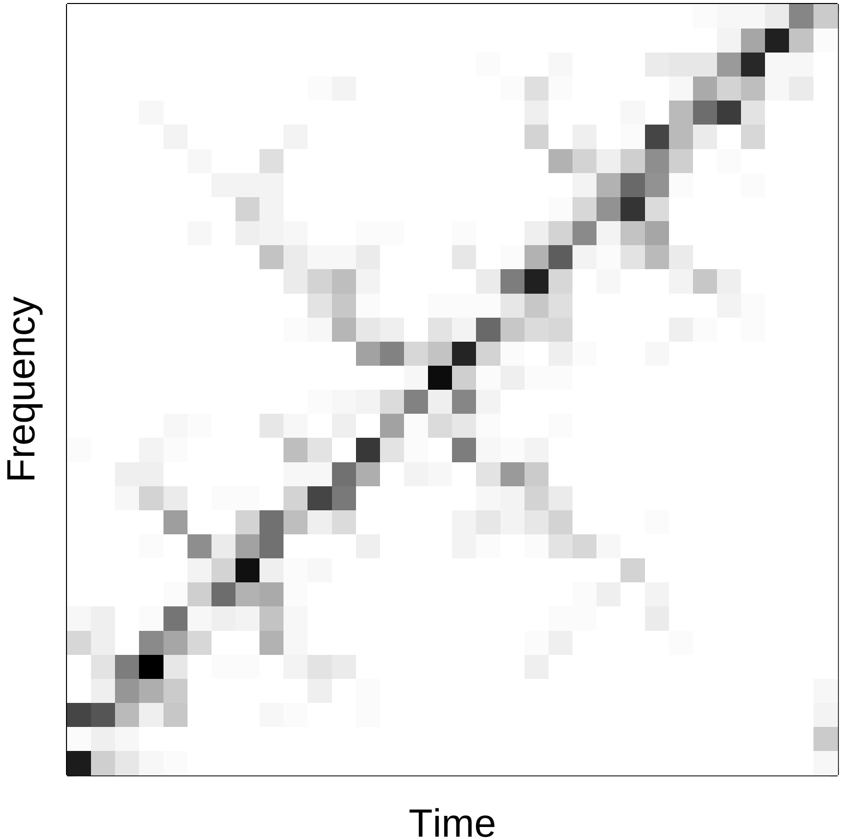
Solution: Swap order in every other application of the DWT:



Time and frequency 15

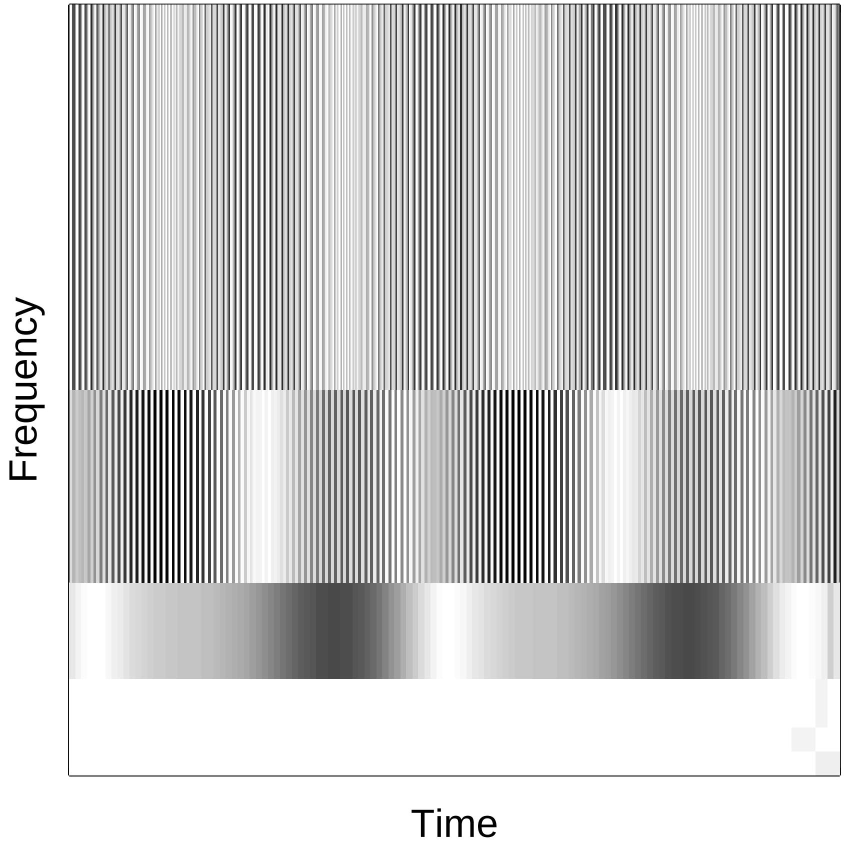
Significance of ordering, linear chirp. Left filter bank order, right natural frequency order.

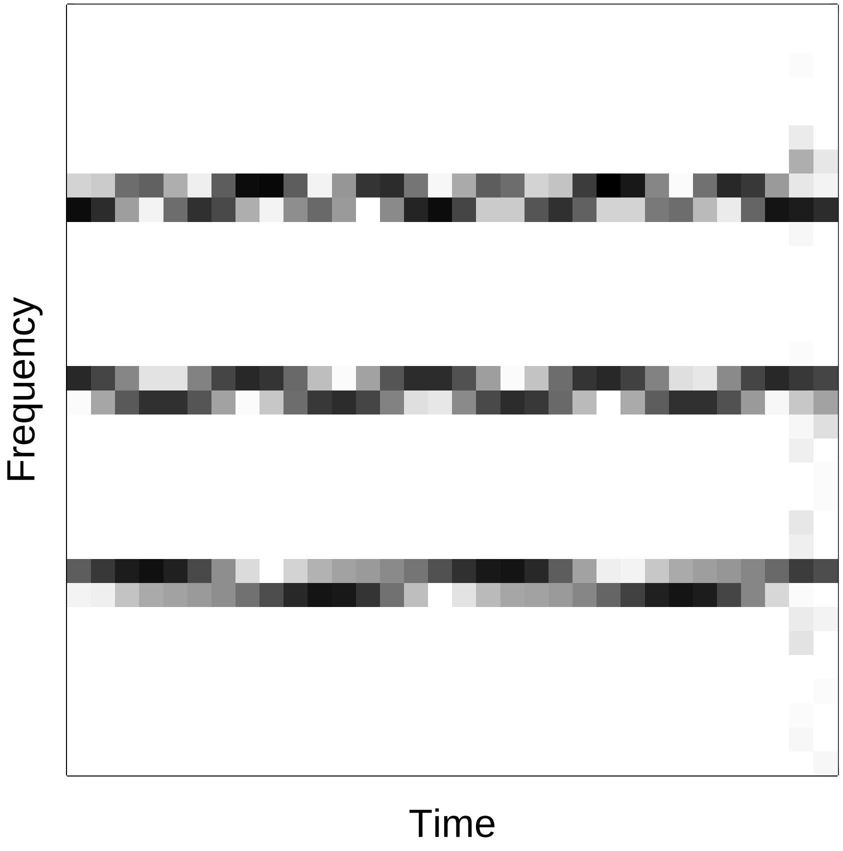




Time and frequency 16

Three frequencies, DWT and best level, $J = 6$.





Time and frequency 17

A complicated signal, length 1024: Sum of

$$x[n] = \begin{cases} 25 & \text{if } n = 300 \ , \\ 1 & \text{if } 500 \leq n \leq 700 \ , \\ 15 & \text{if } n = 900 \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

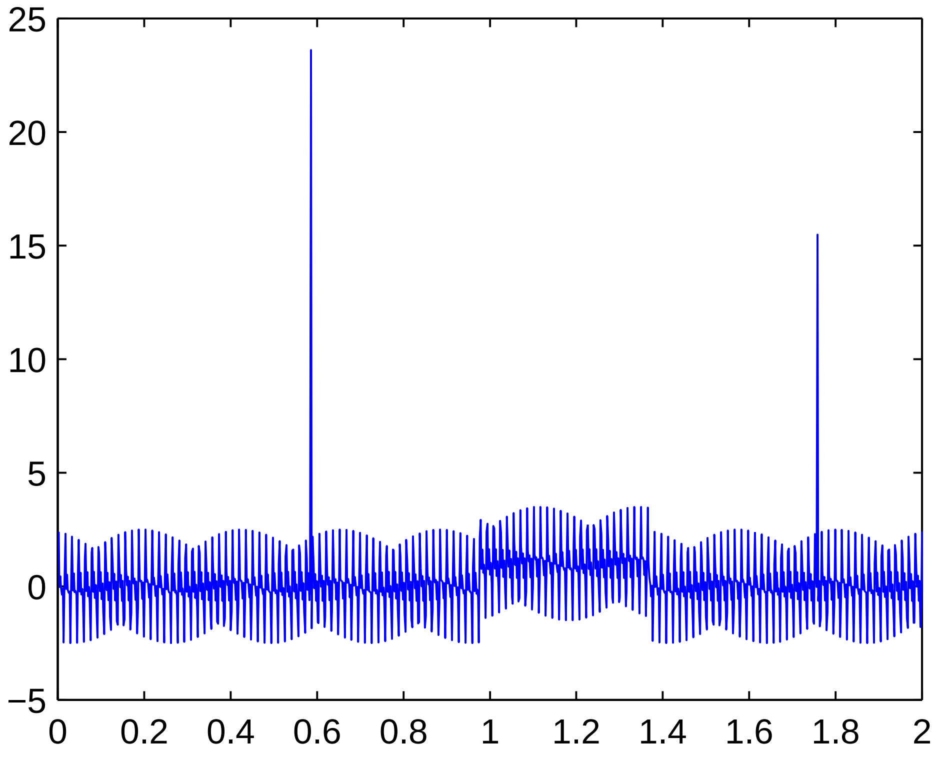
and

$$\sin(\omega_0 t) + \sin(2\omega_0 t) + \sin(3\omega_0 t) \ ,$$

with $\omega_0 = 405.5419$.

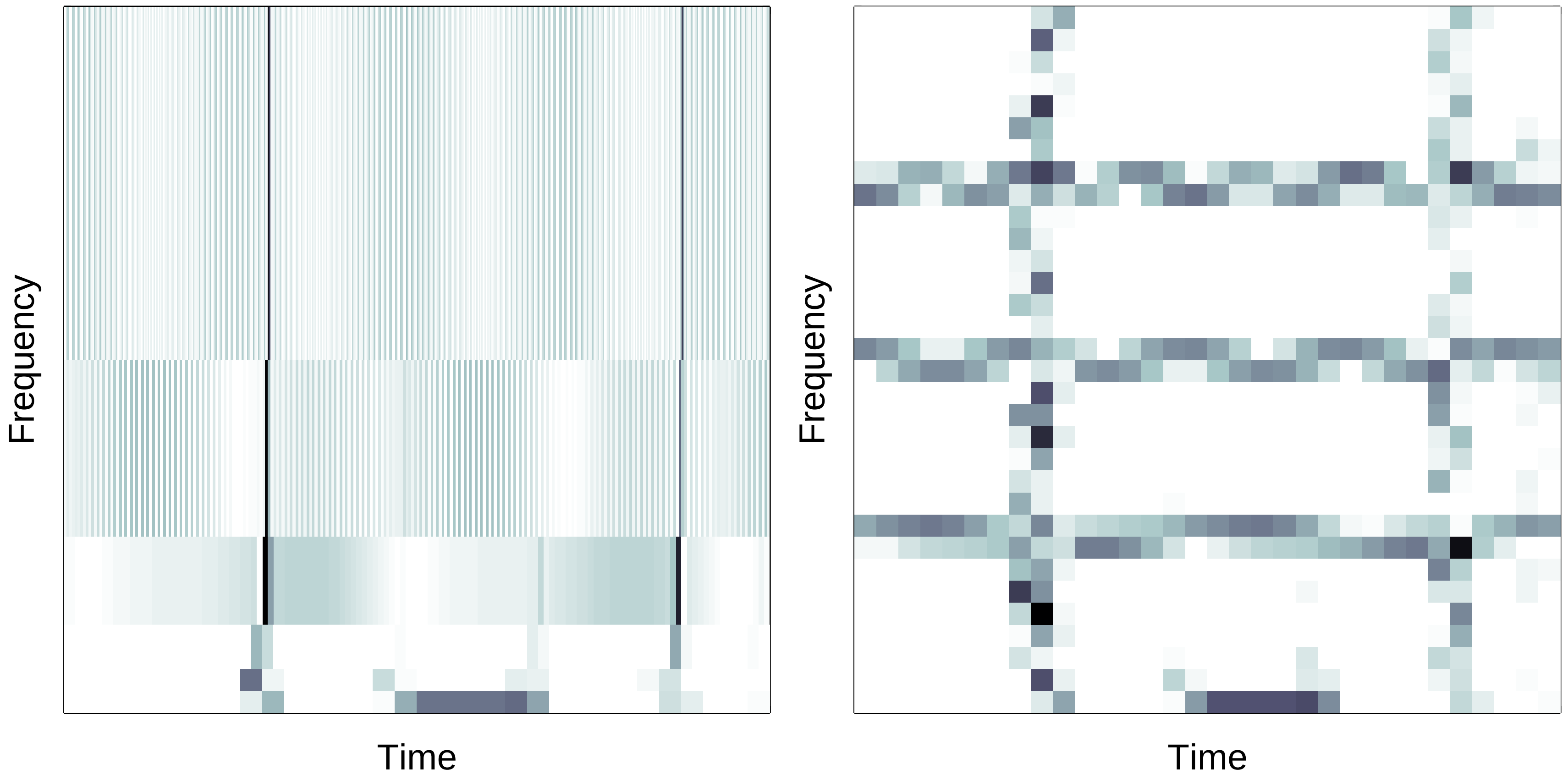
Time and frequency 18

The signal



Time and frequency 19

Time-frequency plane, Daubechies 4, DWT and best level, $J = 6$.



The Fourier transform 1

Review of the Fourier transform. There are at least four variants:

Acronym	Time	Frequency
CTCFFT	Continuous	Continuous
DTCFFT	Discrete	Continuous
CTDFFT	Continuous	Discrete
DTDFFT	Discrete	Discrete

The Fourier transform 2

CTCFFT $x(t) \longleftrightarrow \hat{x}(\omega)$

$$\hat{x}(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \qquad x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{x}(t) e^{j\omega t} d\omega$$

$$\int_{-\infty}^{\infty} |x(t)|^2 dt == \frac{1}{2\pi} \int_{-\infty}^{\infty} |x(\omega)|^2 d\omega$$

$x(t)$ real-valued:

$$\overline{\hat{x}(\omega)} = \hat{x}(-\omega)$$

The Fourier transform 3

DTCFFT $x[n] \longleftrightarrow X(\omega)$

$$X(\omega) = \sum_{n \in \mathbf{Z}} x[n] e^{-jn\omega} \qquad x[n] = \frac{1}{2\pi} \int_0^{2\pi} X(\omega) e^{in\omega} d\omega$$

$$\sum_{n \in \mathbf{Z}} |x[n]|^2 = \frac{1}{2\pi} \int_0^{2\pi} |X(\omega)|^2 d\omega$$

$x[n]$ real-valued:

$$\overline{X(\omega)} = X(-\omega)$$

CTDFFT Interchange role of time and frequency above.

The Fourier transform 4

DTDFFT $\mathbf{x} \longleftrightarrow \hat{\mathbf{x}}$
Orthogonal basis for $\mathbf{C}^N$ $\{\mathbf{e}_k\}_{k=0,\ldots,N-1}$ given by

$$e_k[n] = e^{j2\pi nk/N}, \quad k, n = 0, \ldots, N-1$$

$$\hat{x}[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N} \qquad x[n] = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}[k] e^{j2\pi nk/N}$$

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |\hat{x}[k]|^2$$

The Fourier transform 5

$\mathbf{x} \in \mathbf{C}^N$ realvalued. Then

$$\overline{\hat{x}}[k] = \sum_{n=0}^{N-1} x[n] e^{j2\pi nk/N} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi n(N-k)/N} = \hat{x}[N-k]$$

Comparing DTDF with DTCF we see that $\hat{\mathbf{x}}$ is obtained by sampling $X(\omega)$ at the frequencies $0, 2\pi/N, \ldots, 2\pi(N-1)/N$, ie

$$\hat{x}[k] = X(2\pi k/N)$$

Sampling 1

A continuous signal $x(t)$ is sampled at times nT, $n \in \mathbf{Z}$.
Fourier series with this time unit:

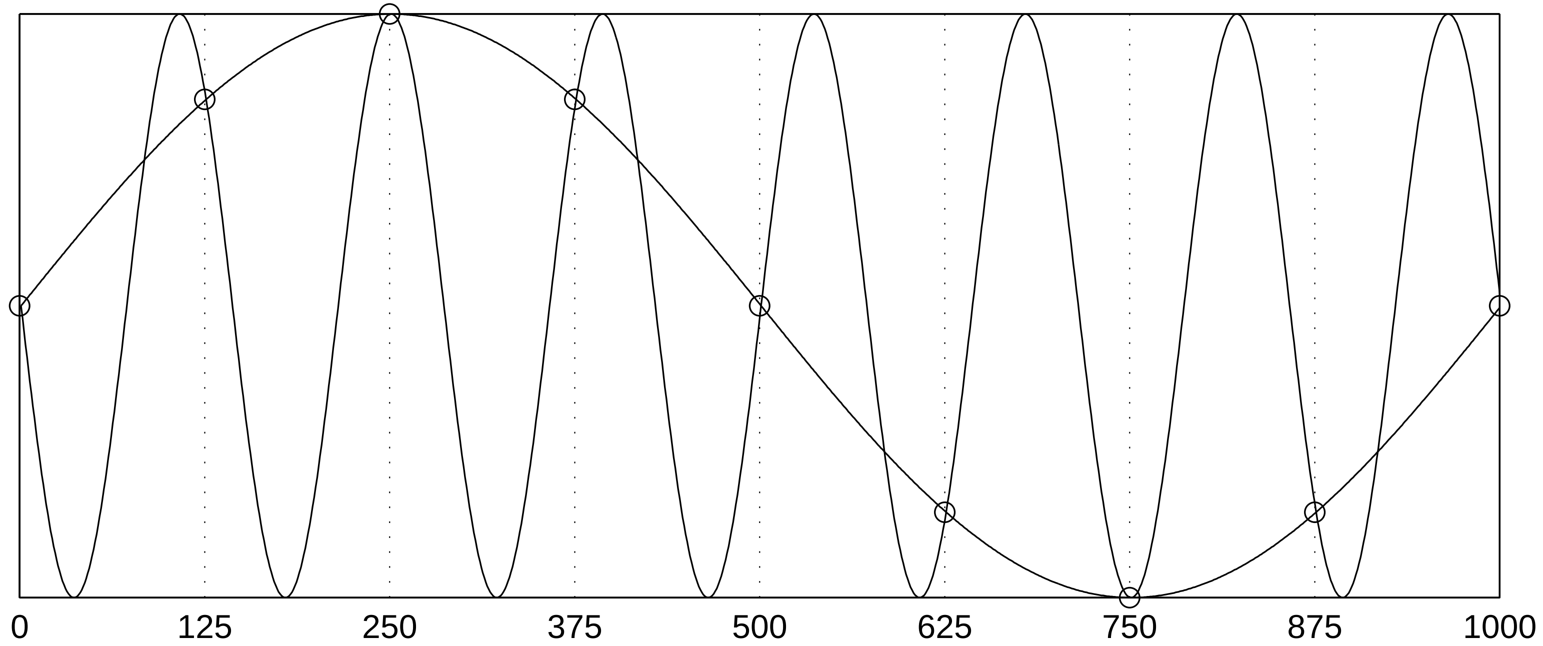
$$X_T(\omega) = \sum_n x[n]e^{-jnT\omega}$$

Relation to the CTCFFT:

$$X_T(\omega) = \frac{1}{T}\sum_{k\in\mathbf{Z}} \hat{x}\Big(\omega - \frac{2k\pi}{T}\Big)$$

Sampling 2

Illustration of aliasing effect (undersampling):



Short Time Fourier Transform 1

The Short Time Fourier Transform (STFT) is based on DTCFFT and a window function:

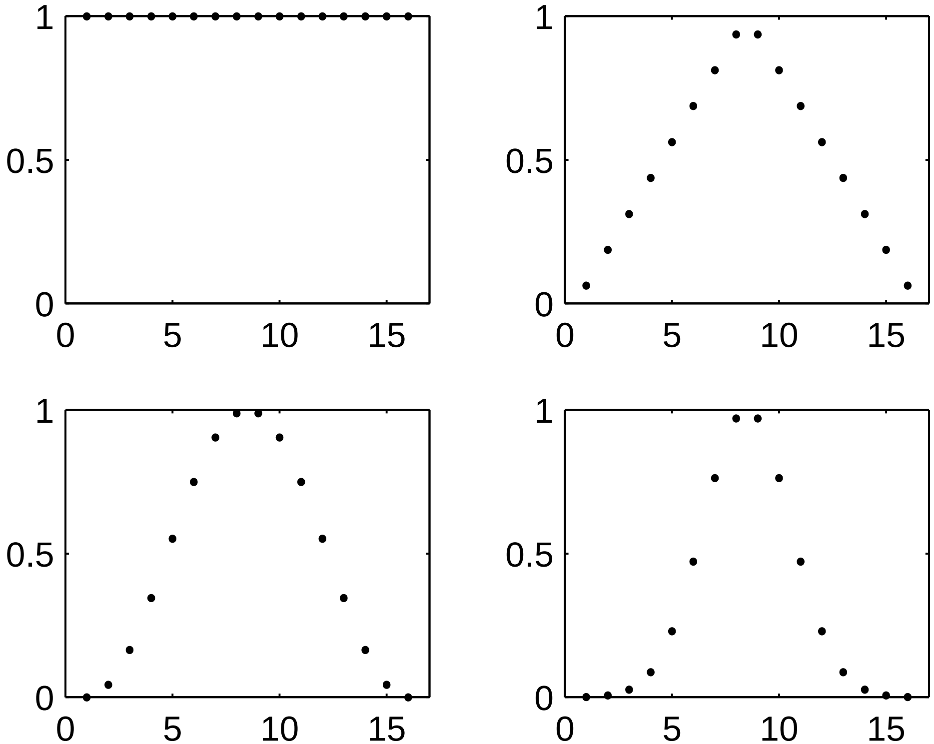
$$X_{\mathrm{STFT}}(k,\omega) = \sum_{n\in\mathbf{Z}} w[n-k]x[n]e^{-jnT\omega}$$

Let $\mathbf{x}$ be a signal of length N. Usual choice of k is for N even is $k = mN/2$, $m \in \mathbf{Z}$, and for N odd $k = m(N-1)/2$, $m \in \mathbf{Z}$.
The window function $\mathbf{w}$ gives a localization in time.
Example is Hanning window:

$$w[n] = \sin^2(\pi(n-1)/N), \quad n = 1,\ldots,N$$

Short Time Fourier Transform 2

Examples with $N = 16$: Rectangular, triangular, Hanning and Gaussian windows.



Short Time Fourier Transform 3

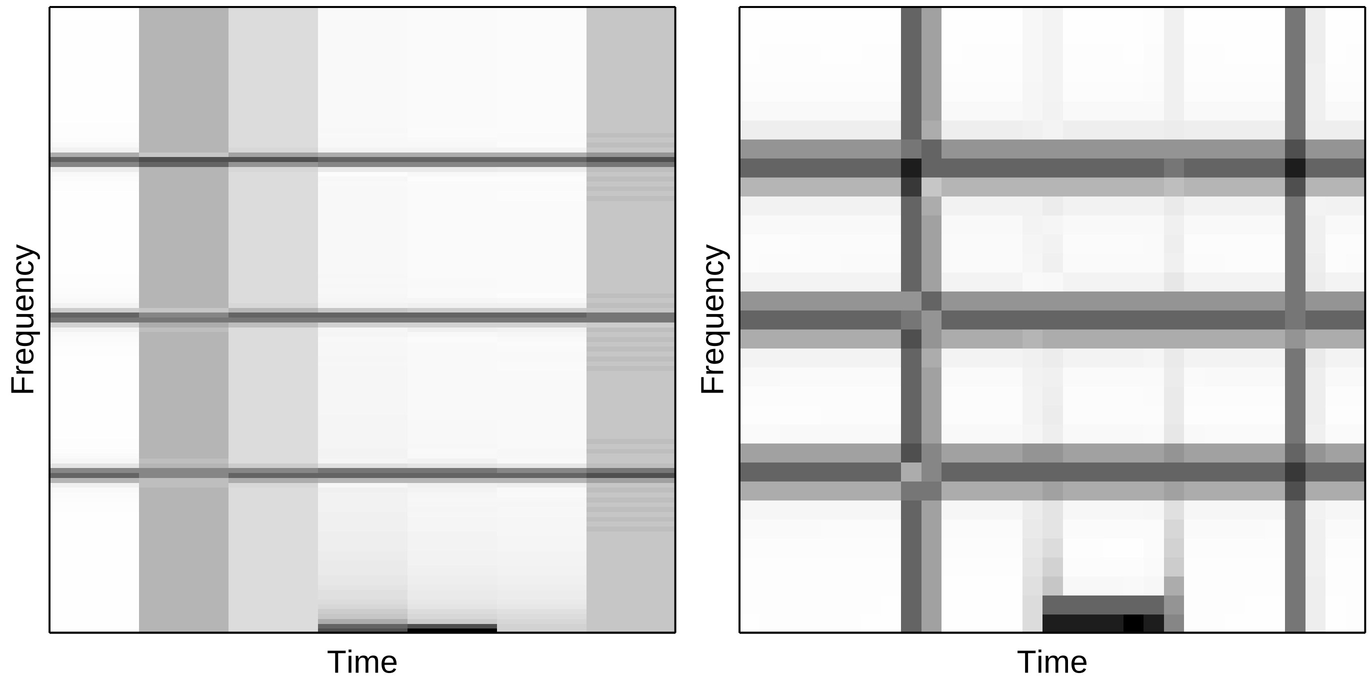
The spectrogram is obtained by plotting

$$\frac{1}{2\pi}|X_{\mathrm{STFT}}(k, 2\pi n/N)|^2$$

for values of k determined by the length of the window, and for $n = 0, \ldots, N-1$. Visualized in the time-frequency plane by using cells of a size determined by the length of the window in the frequency direction and by the length of the signal and the overlap in the time direction.

An example

We compute the spectrogram of the signal used above. On the left hand side we use a Hanning window of length 256, on the right hand side the length is 64.

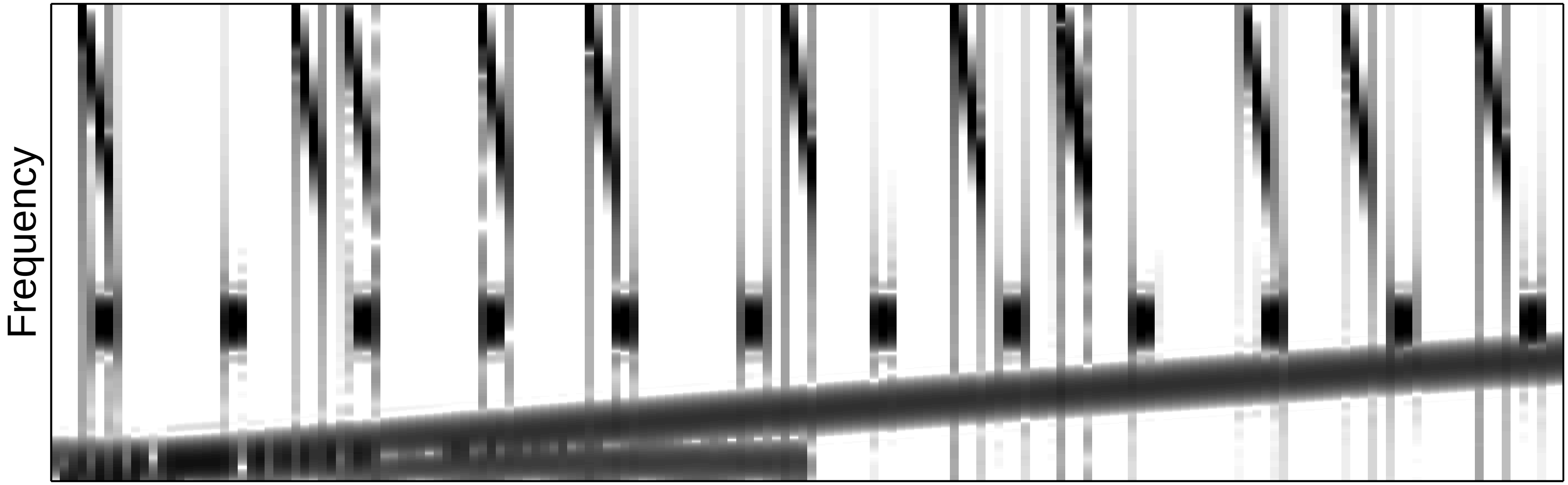


Comparative example 1

In the final example we compare the methods on a complicated signal. We perform two wavelet and two Fourier based analyses of the signal. The first two are STFT based, with a long and a short window. forcing us to identify either slow or fast oscillations, but not both. The wavelet based analysis shows first the result of a level basis analysis. The final one uses the best basis algorithm with the Shannon entropy. This clearly gives a superior result.

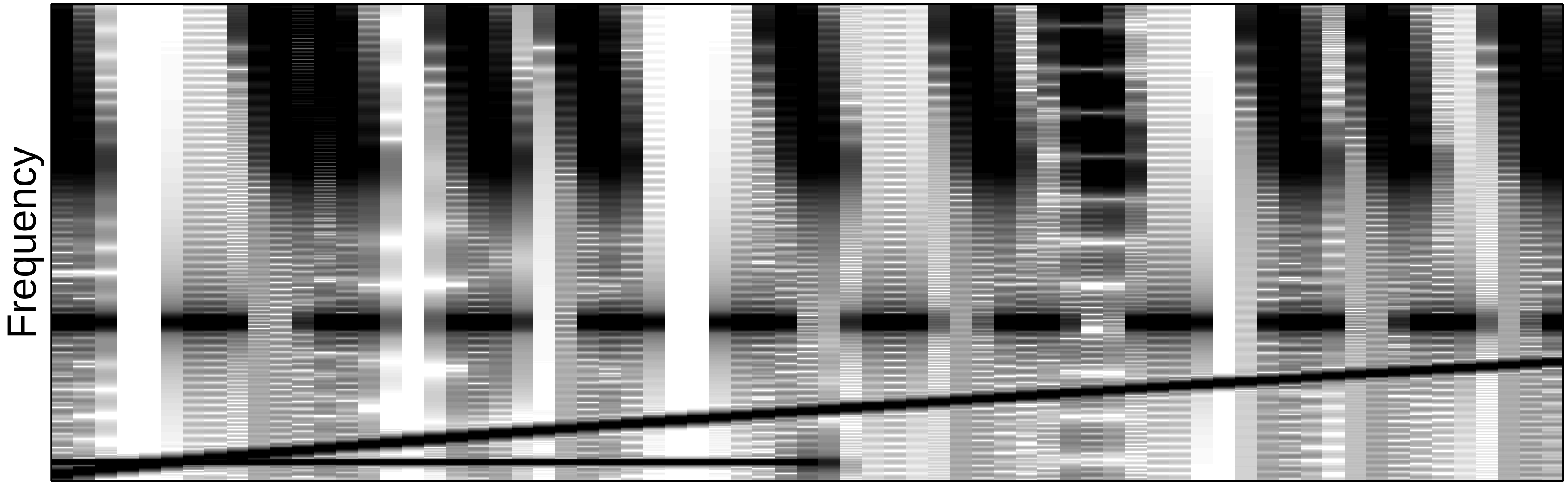
Comparative example 2

Spectrogram, 1024 point FFT, windows 64, overlap 16.



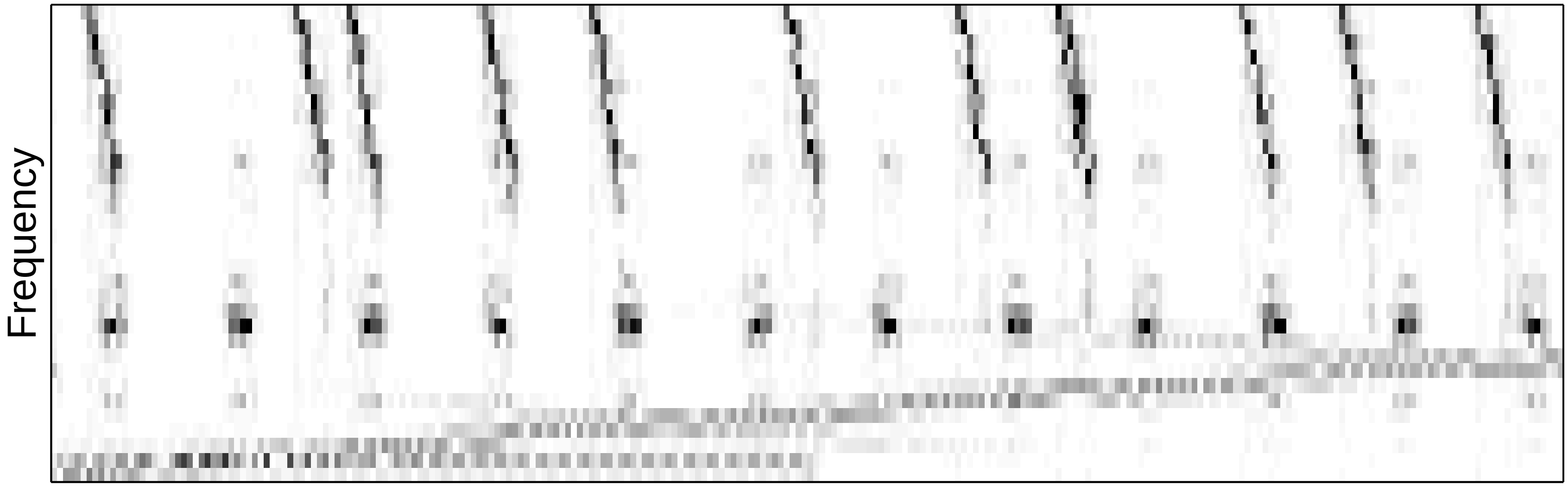
Comparative example 3

Spectrogram, 1024 point FFT, windows 512, overlap 400.



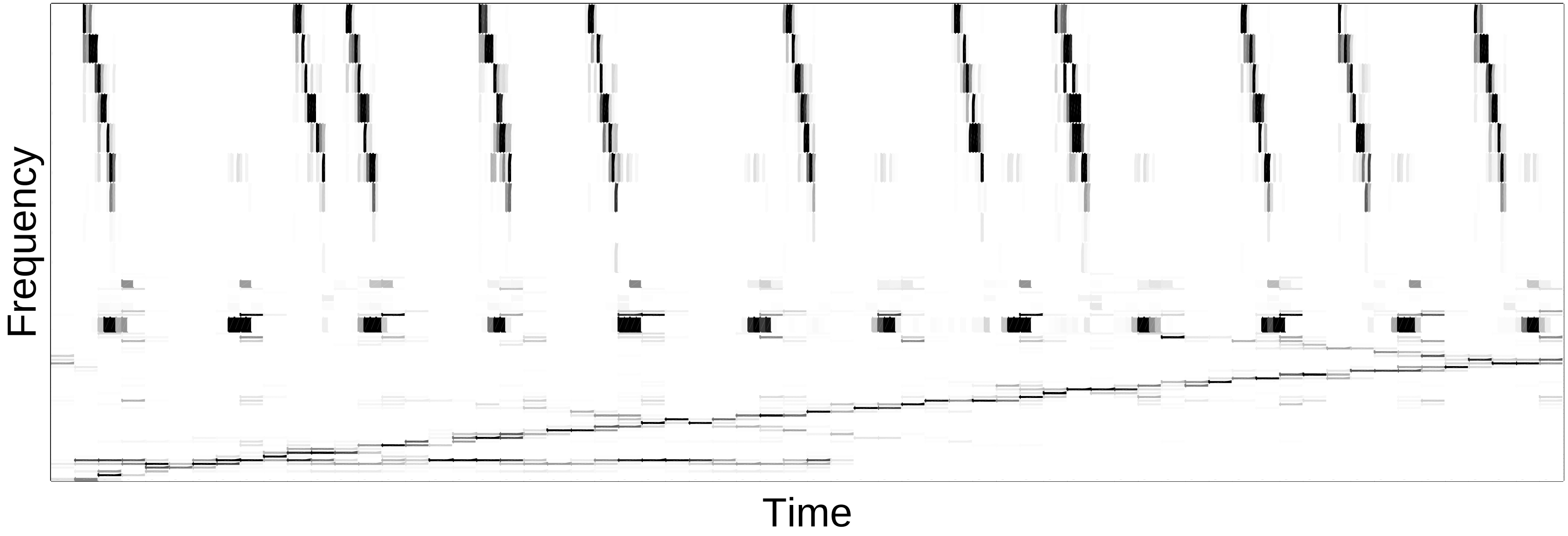
Comparative example 4

Wavelet packet level basis, symlet 12.



Comparative example 5

Wavelet packet, best basis, Shannon entropy, symlet 12.



Final remarks

I have introduced you to the discrete wavelet transform and its generalization, the wavelet packet transform. I have also reviewed some results from Fourier analysis, and shown you a comparative study on two signals.
If you want to learn more, start by reading the book mentioned in the introduction, and then start experimenting with the transforms, both on synthetic signals, and on real world signals.

Thank your for your attention!

Technical afterword

One question often asked, after someone has seen this presentation, is how it is produced.
It is produced using LaTeX with the document class `prosper`. It can be found at

`http://prosper.sourceforge.net`

It is of course in the public domain.