EE434
Biomedical Signal Processing
Lecture # 4

I would like to thank Professor Robi Polikar for using his lecture notes.

Nonstationary Signal Processing

STFT & Wavelets

How Does FT Work?

- Recall that **FT** uses complex exponentials (sinusoids) as building blocks.

$$e^{j\omega t} = \cos(\omega t) + j\sin(\omega t)$$

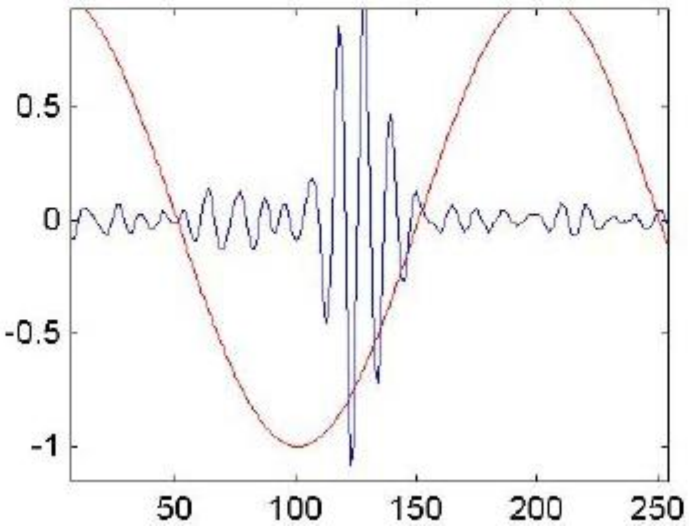
- For each frequency of complex exponential, the sinusoid at that frequency is compared to the signal.

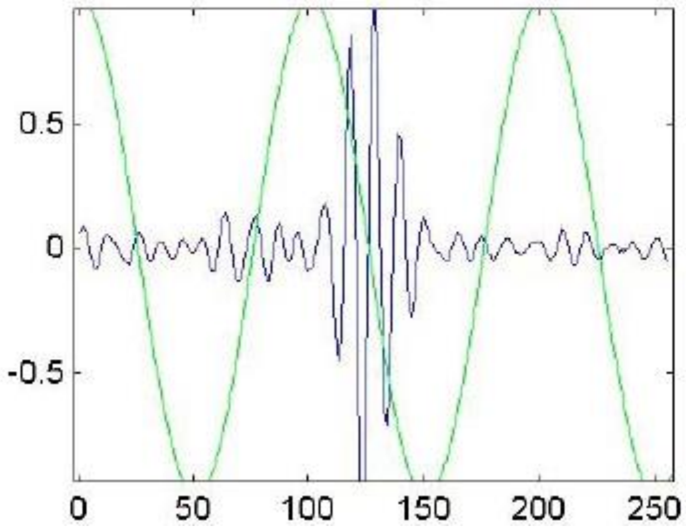
$$F(\omega) = \int f(t)e^{-j\omega t}dt \quad \Leftrightarrow \quad f(t) = \frac{1}{2\pi}\int F(\omega)e^{j\omega t}d\omega$$

- If the signal consists of that frequency, the correlation is high → large FT coefficients.

- If the signal does not have any spectral component at a frequency, the correlation at that frequency is low / zero, → small / zero FT coefficient.

FT at Work

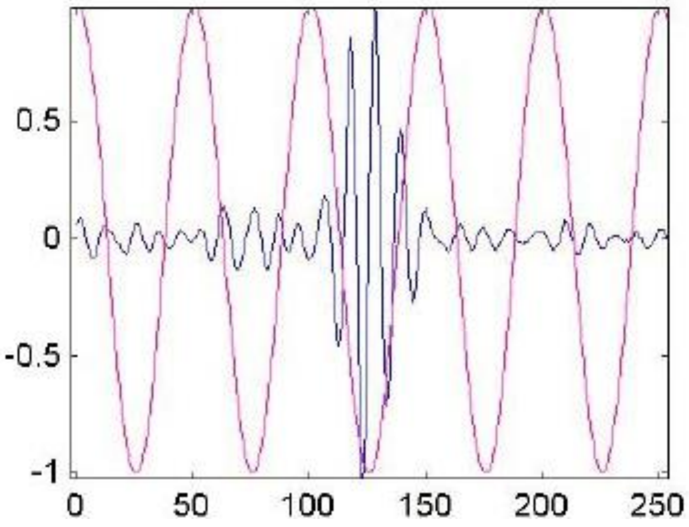


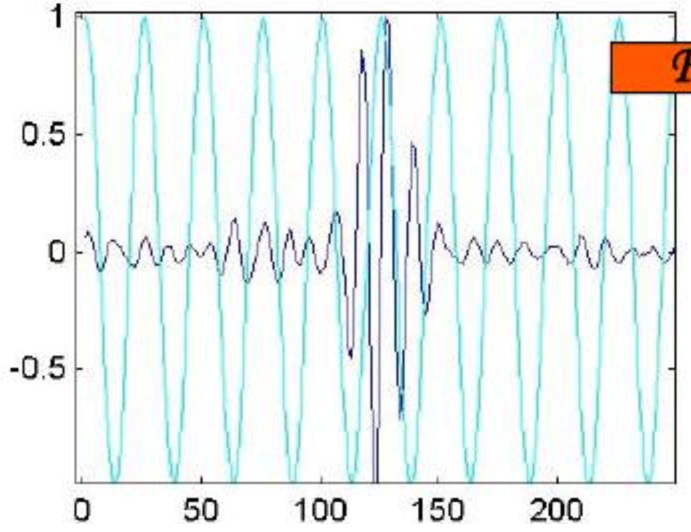


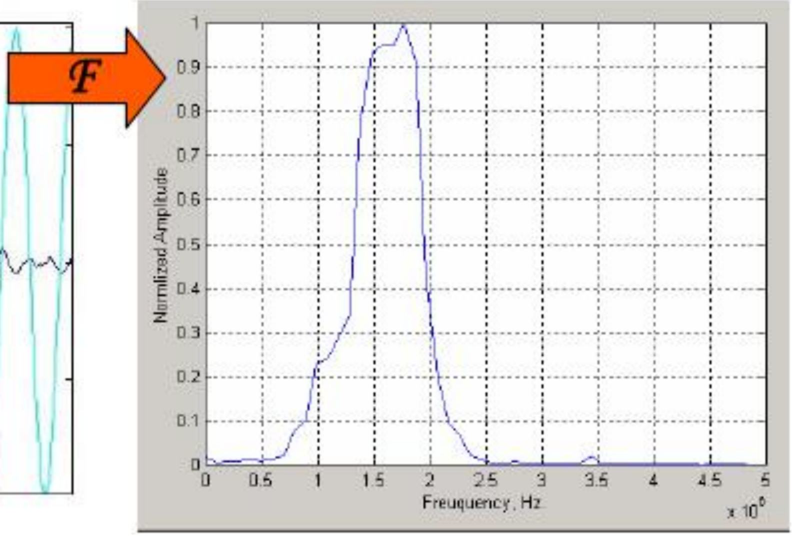
Complex exponentials (sinusoids) as basis functions:

$$X(j\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

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







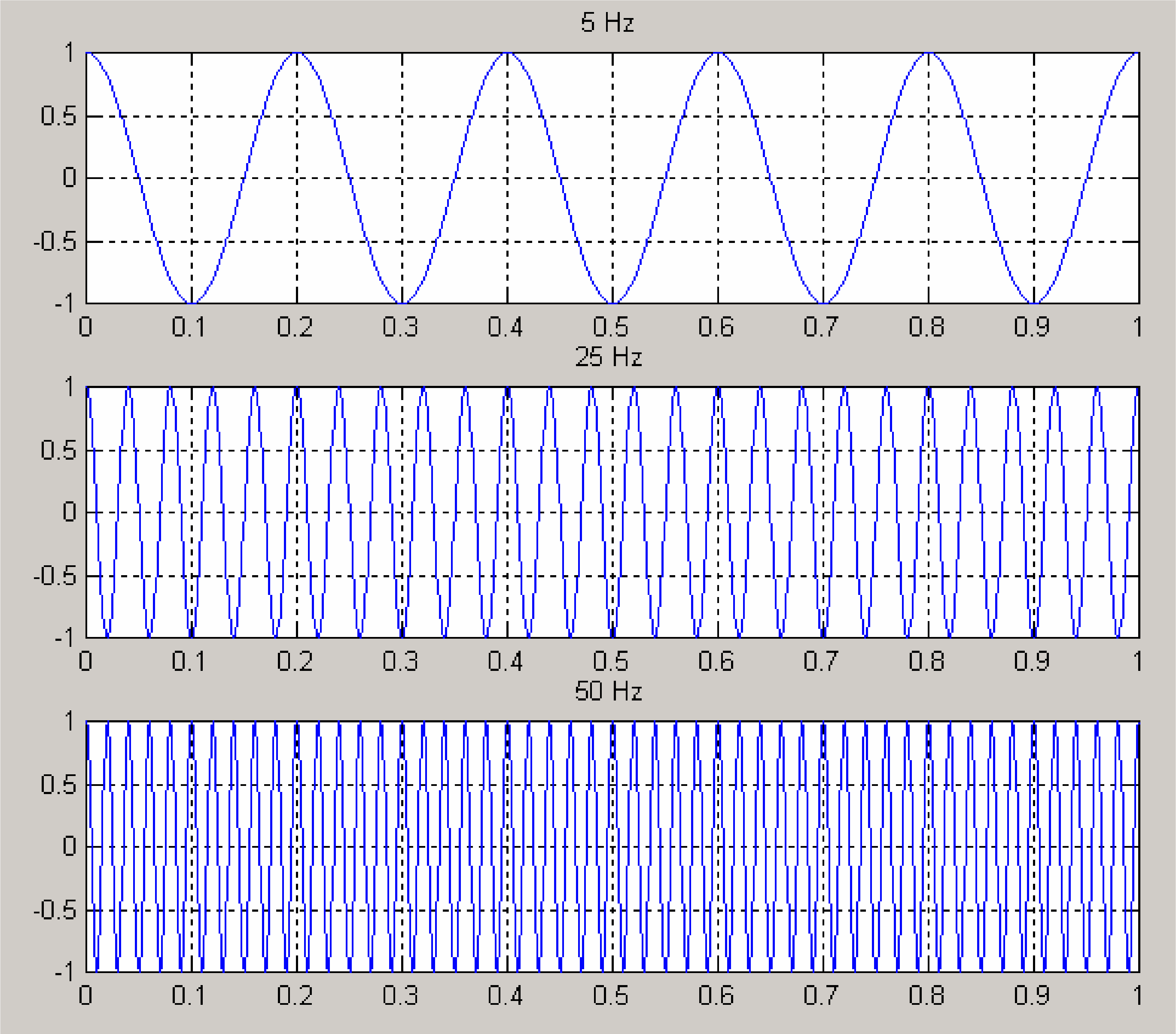
An ultrasonic A-scan using 1.5 MHz transducer, sampled at 10 MHz

FT at Work

$$x_1(t) = \cos\,(2\pi\cdot 5\cdot t)$$

$$x_2(t) = \cos\,(2\pi\cdot 25\cdot t)$$

$$x_3(t) = \cos\,(2\pi\cdot 50\cdot t)$$

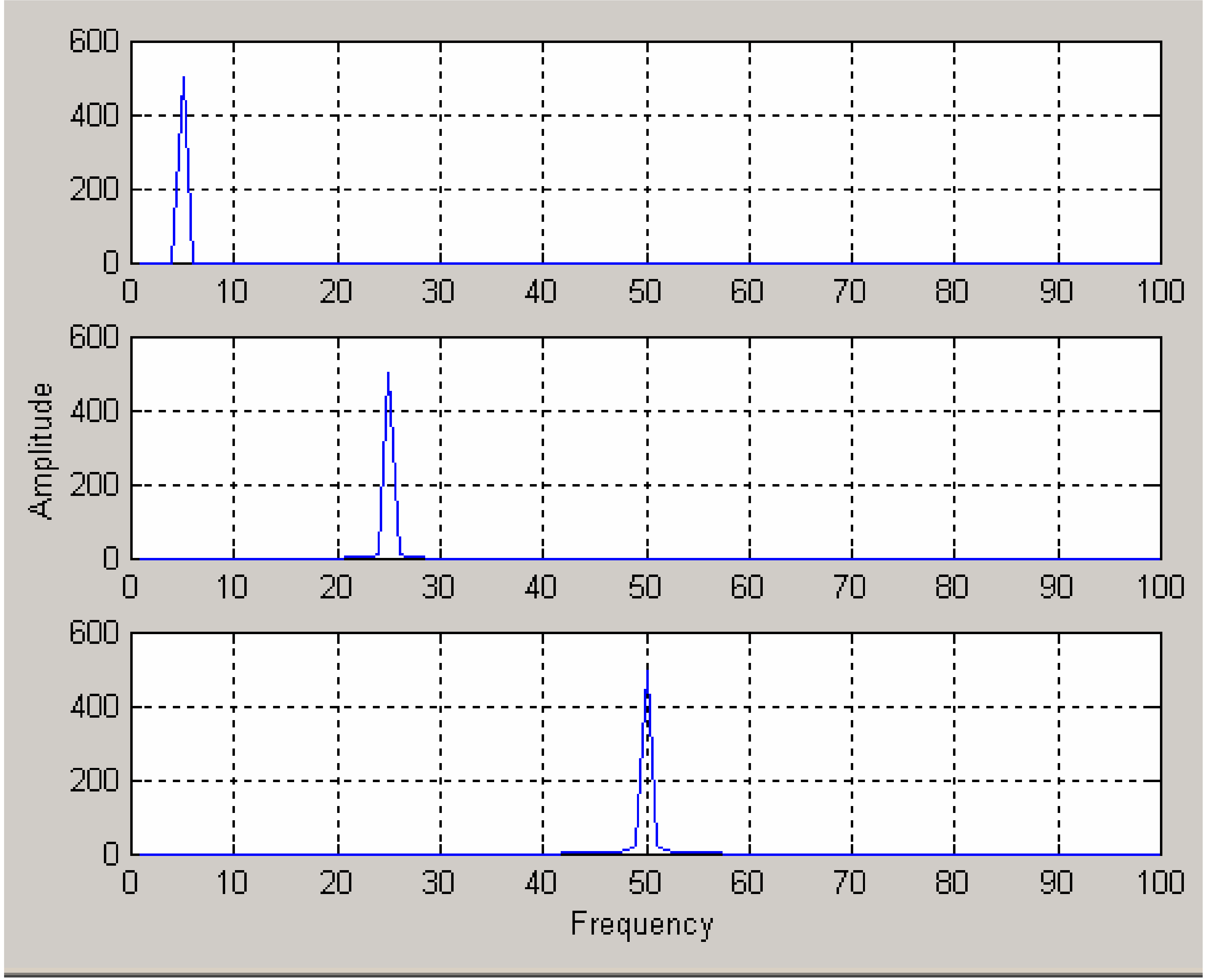


FT at Work

$$x_1(t) \overset{\Im}{\leftrightarrow} X_1(\omega)$$

$$x_2(t) \overset{\Im}{\leftrightarrow} X_2(\omega)$$

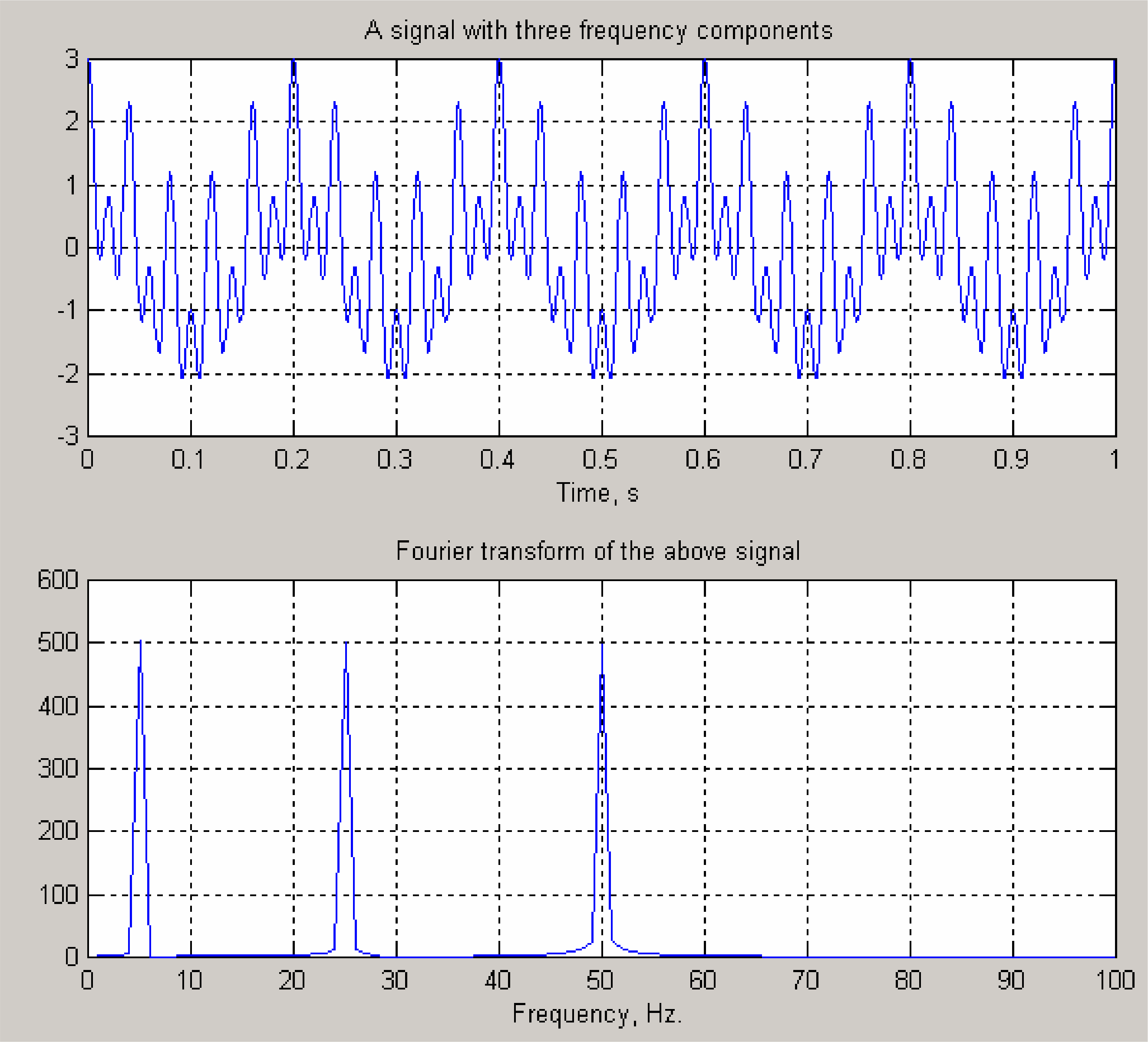
$$x_3(t) \overset{\Im}{\leftrightarrow} X_3(\omega)$$



FT at Work

$$x_4(t) = \cos(2\pi\cdot 5\cdot t) + \cos(2\pi\cdot 25\cdot t) + \cos(2\pi\cdot 50\cdot t)$$

$$x_4(t) \overset{\Im}{\leftrightarrow} X_4(\omega)$$



Stationary and Non-stationary Signals

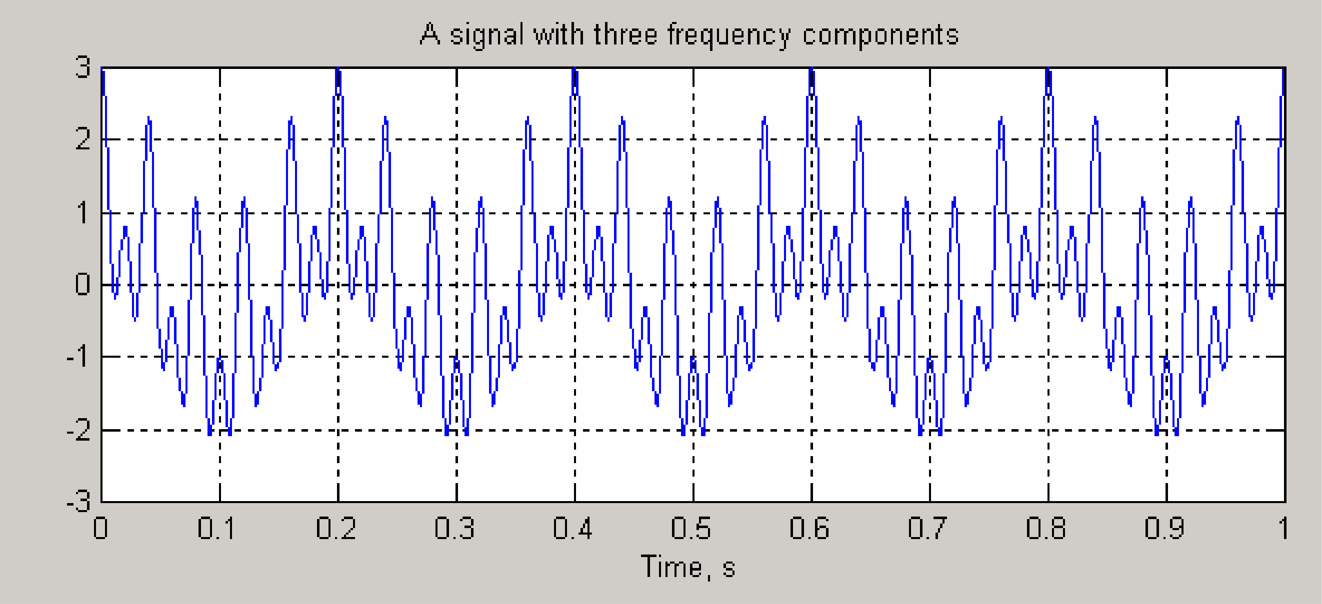
- **FT** identifies all spectral components present in the signal, however it does not provide any information regarding the temporal (time) localization of these components. Why?
- **Stationary signals** consist of spectral components that do not change in time*
  - all spectral components exist at all times
  - no need to know any time information
  - **FT** works well for stationary signals
- However, **non-stationary signals** consists of time varying spectral components
  - How do we find out which spectral component appears when?
  - **FT** only provides **what spectral components exist** , **not where in time they are located**.
  - Need some other ways to determine **time localization of spectral components**

* Formally, a **stationary signal** is one whose statistical properties (distribution) does not change in time – it is a very strict requirement that is rarely met by real world signals

Stationary and Non-stationary Signals

- **Stationary signals′** spectral characteristics do not change with time

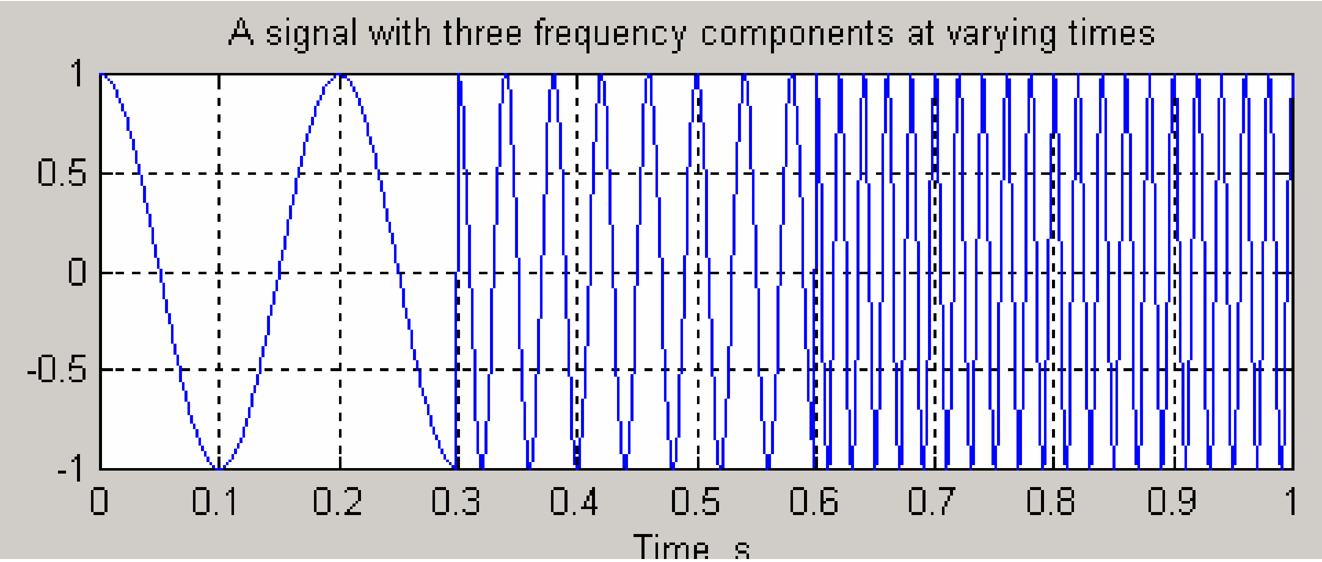
$$x_4(t) = \cos(2\pi\cdot 5\cdot t) + \cos(2\pi\cdot 25\cdot t) + \cos(2\pi\cdot 50\cdot t)$$



- **Non-stationary signals** have time varying spectra

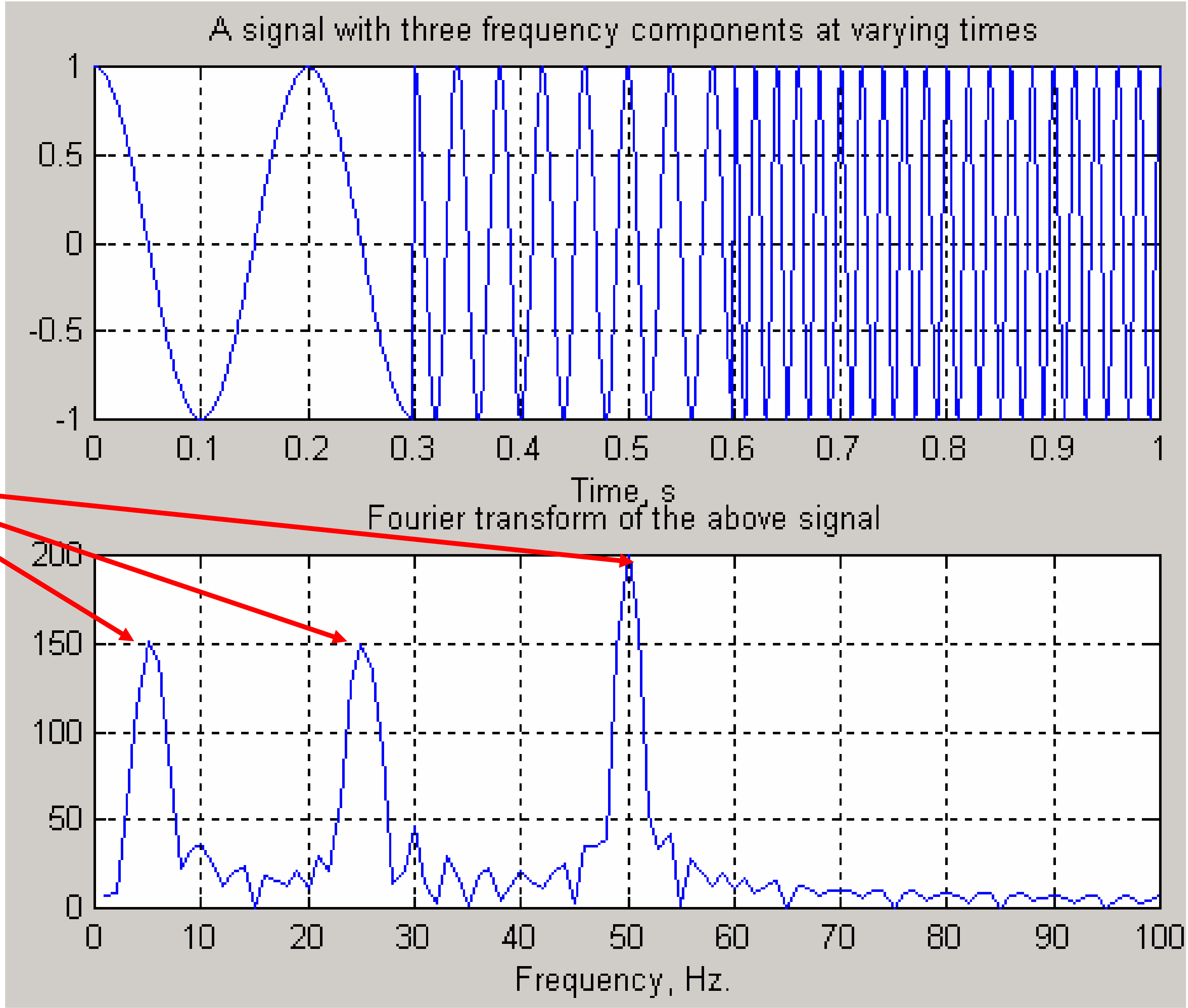
$$x_5(t) = [x_1(t) \oplus x_2(t) \oplus x_3(t)]$$

$\oplus$: Concatenation



Non-stationary Signals

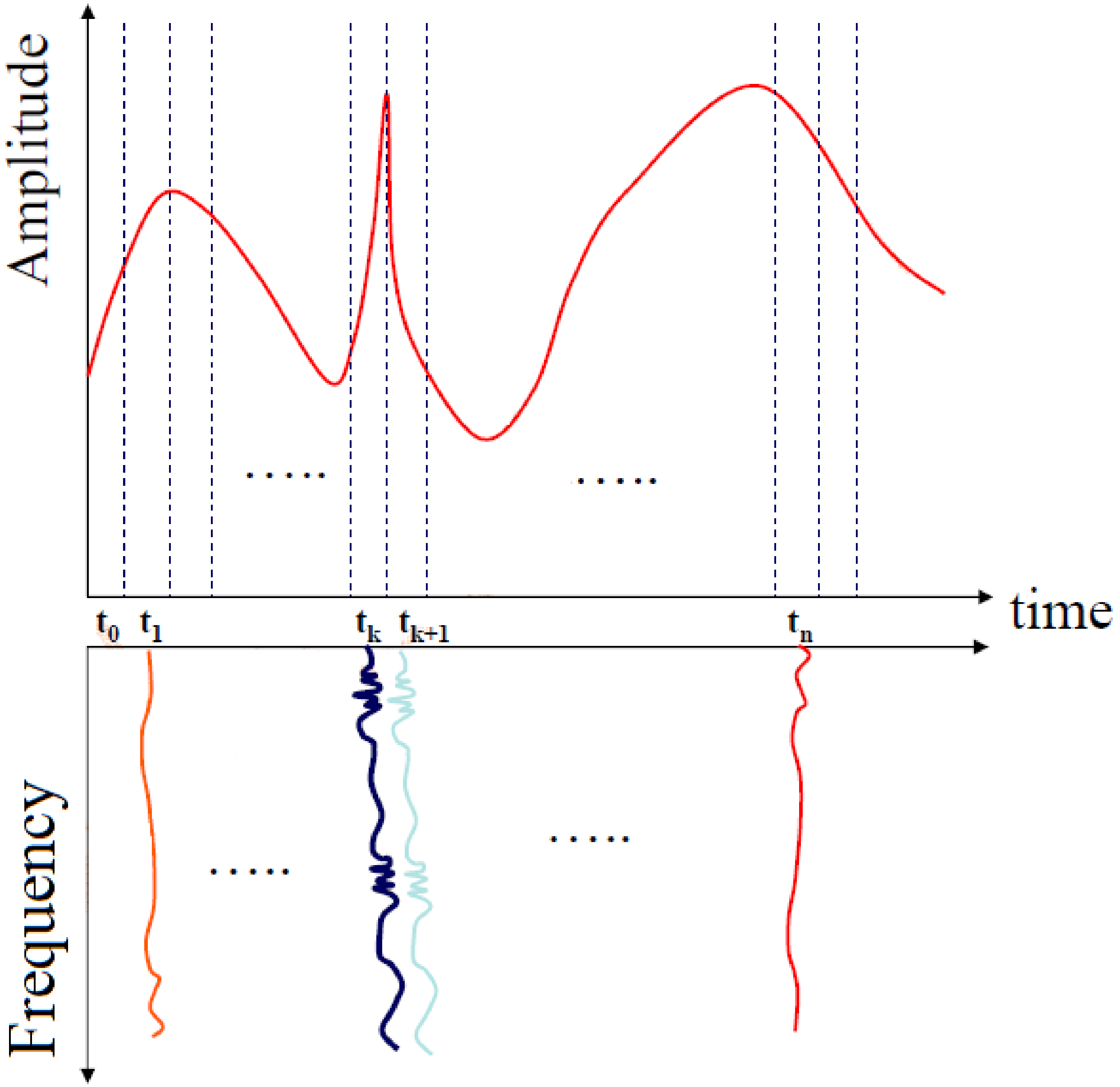
Perfect knowledge of what frequencies exist, but no information about where these frequencies are located in time



FT Shortcomings

- Complex exponentials stretch out to infinity in time
  - They analyze the signal globally, not locally
  - Hence, **FT** can only tell what frequencies exist in the entire signal, but cannot tell, at what time instances these frequencies occur
  - In order to obtain time localization of the spectral components, the signal need to be analyzed locally
  - How?

Short Time Fourier Transform (STFT)



The Short Time Fourier Transform

- Take **FT** of segmented consecutive pieces of a signal.
- Each **FT** then provides the spectral content of that time segment only
  - Spectral content for different time intervals
  - → Time-frequency representation

Time parameter

Frequency parameter

Signal to be analyzed

FT Kernel (Basis function)

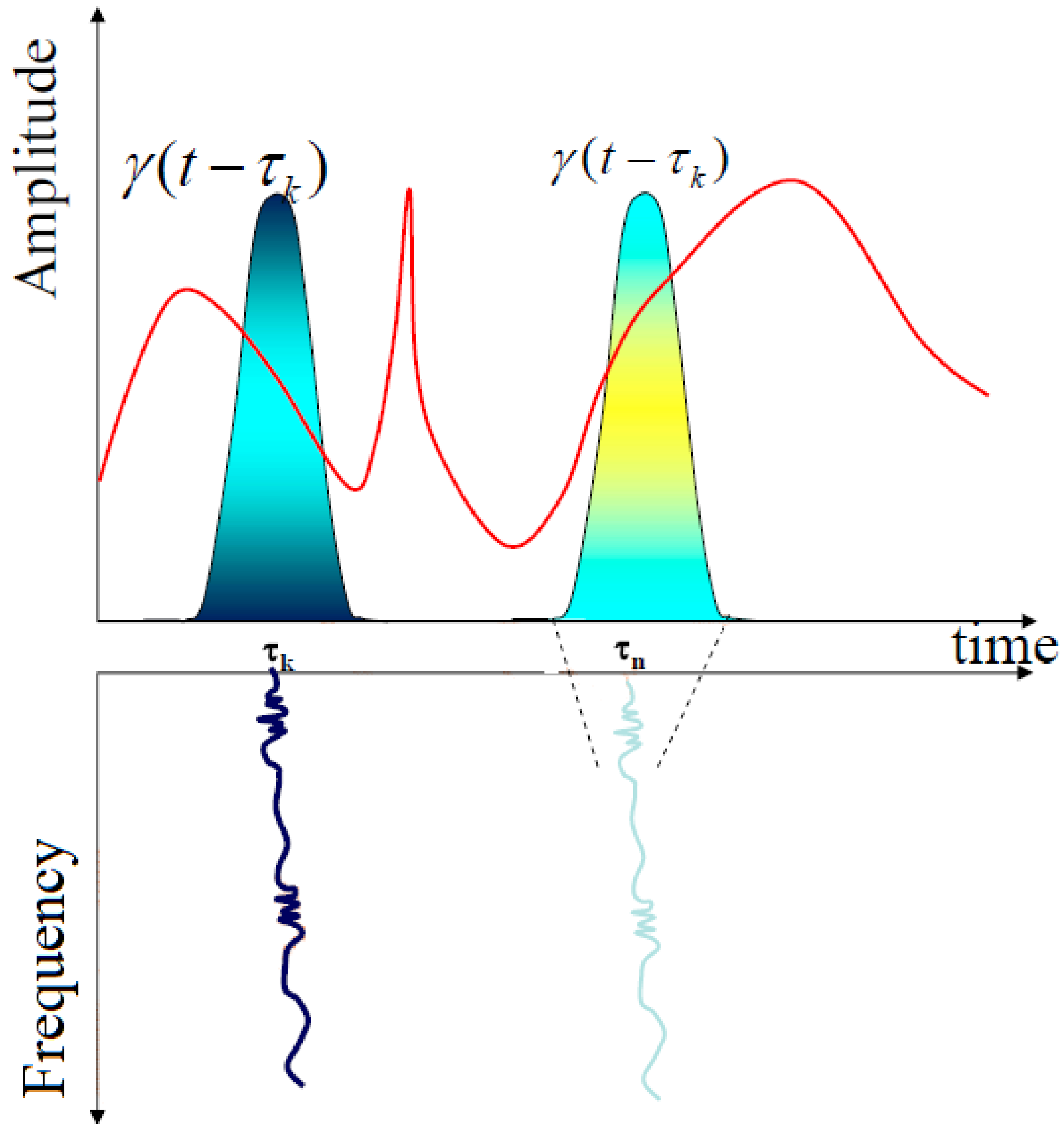
$$STFT_x^{\omega}(\tau,\omega)=\int_t [x(t)\cdot\gamma(t-\tau)]\cdot e^{-j\omega t}dt$$

STFT of signal $x(t)$: Computed for each window centered at $t=\tau$ (localized spectrum)

Windowing function (analysis window)

Windowing function centered at $t=\tau$

Resolution Issues



All signal attributes located within the local window interval around "t" will appear at "t" in the STFT

Time-Frequency Resolution

- Closely related to the choice of analysis window
  - Narrow window → good time resolution
  - Wide window (narrow band) → good frequency resolution
- Two extreme cases:
  - $\gamma(T) = \delta(t)$ → excellent time resolution, no frequency resolution
  - $\gamma(T) = 1$ → excellent frequency resolution, no time info!!!
  - How to choose the **window length**?

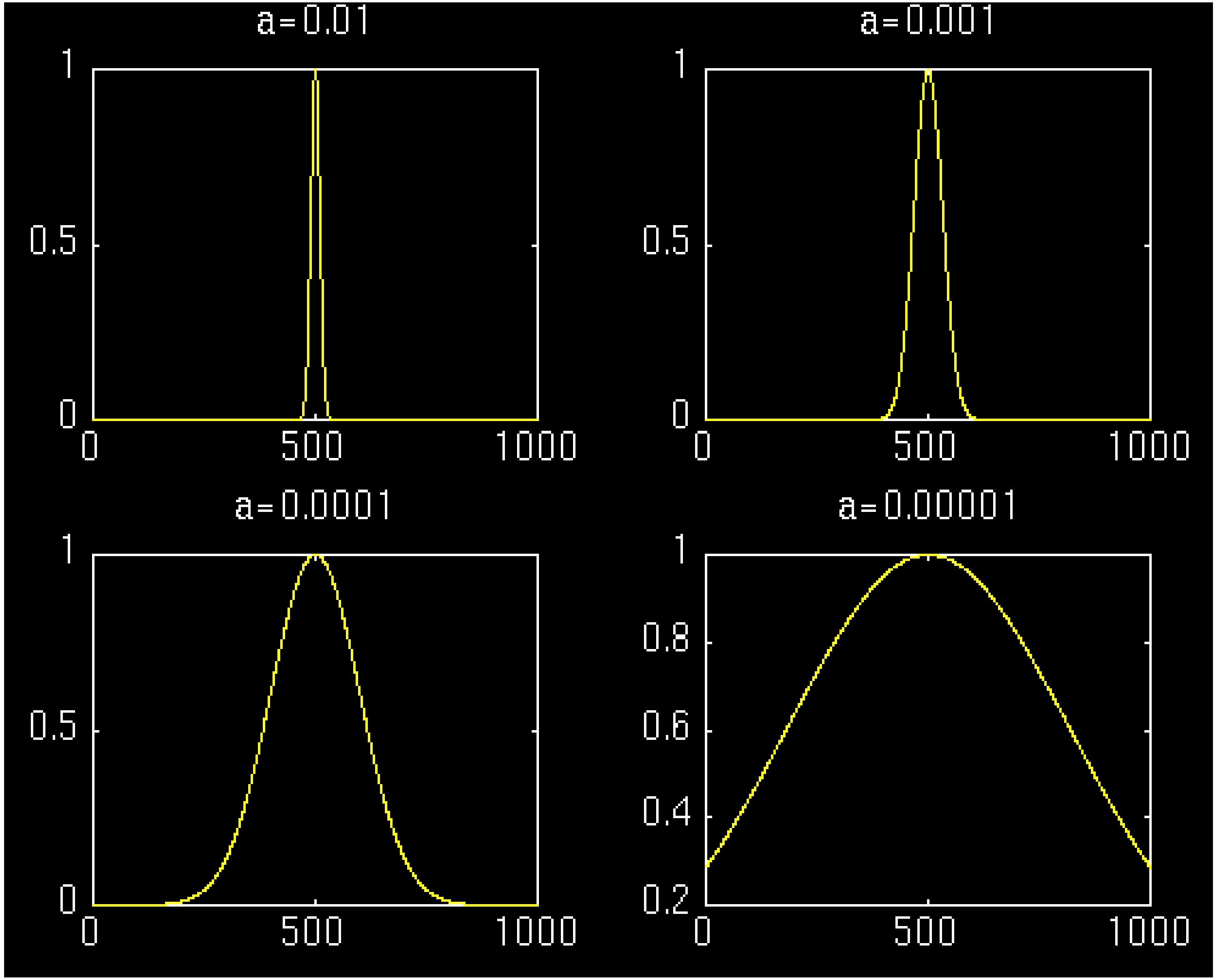
$$STFT_x^{\omega}(\tau,\omega) = \int_t [x(t)\cdot\gamma(t-\tau)]\cdot e^{-j\omega t}dt$$

$$\gamma(t) = \delta(t) \rightarrow STFT(\tau,\omega) = x(\tau)e^{-j\omega\tau}$$

$$\gamma(t) = 1 \rightarrow STFT(\tau,\omega) = X(\omega)$$

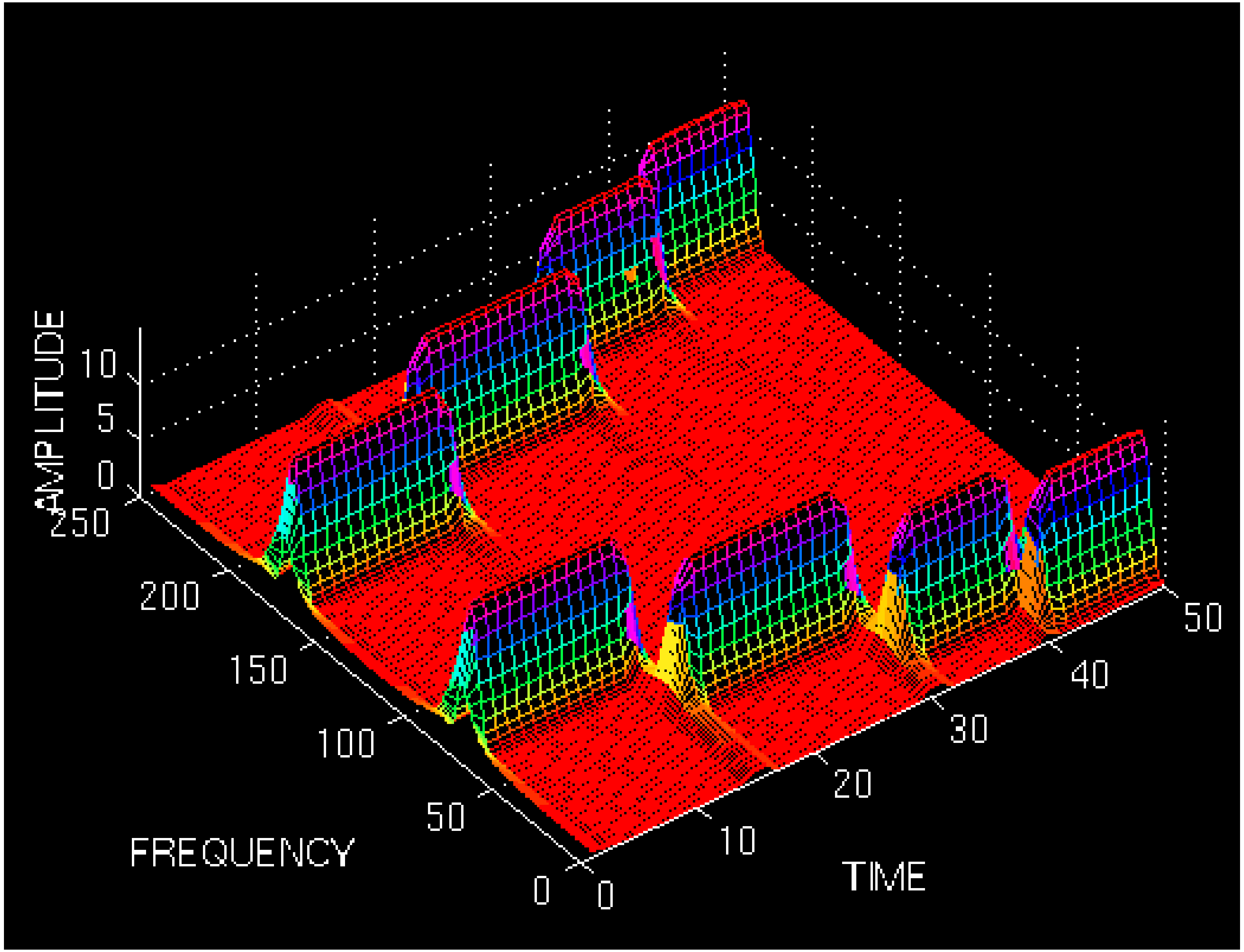
- Window length defines the time and frequency resolutions
- Heisenberg's inequality
  - Cannot have arbitrarily good time and frequency resolutions. One must trade one for the other. Their product is bounded from below.

STFT Example



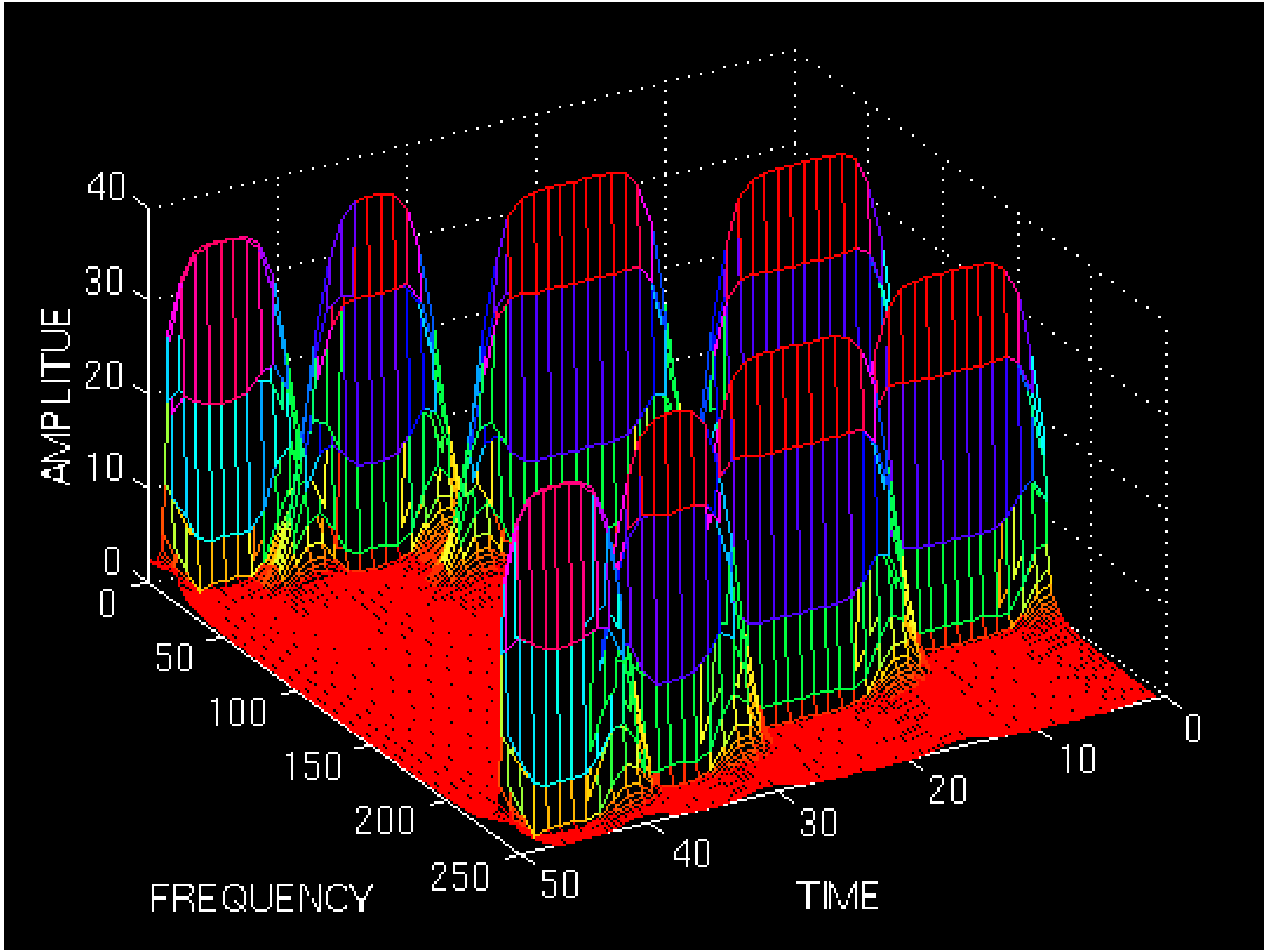
$$\gamma(t) = e^{-at^2/2}$$

STFT Example



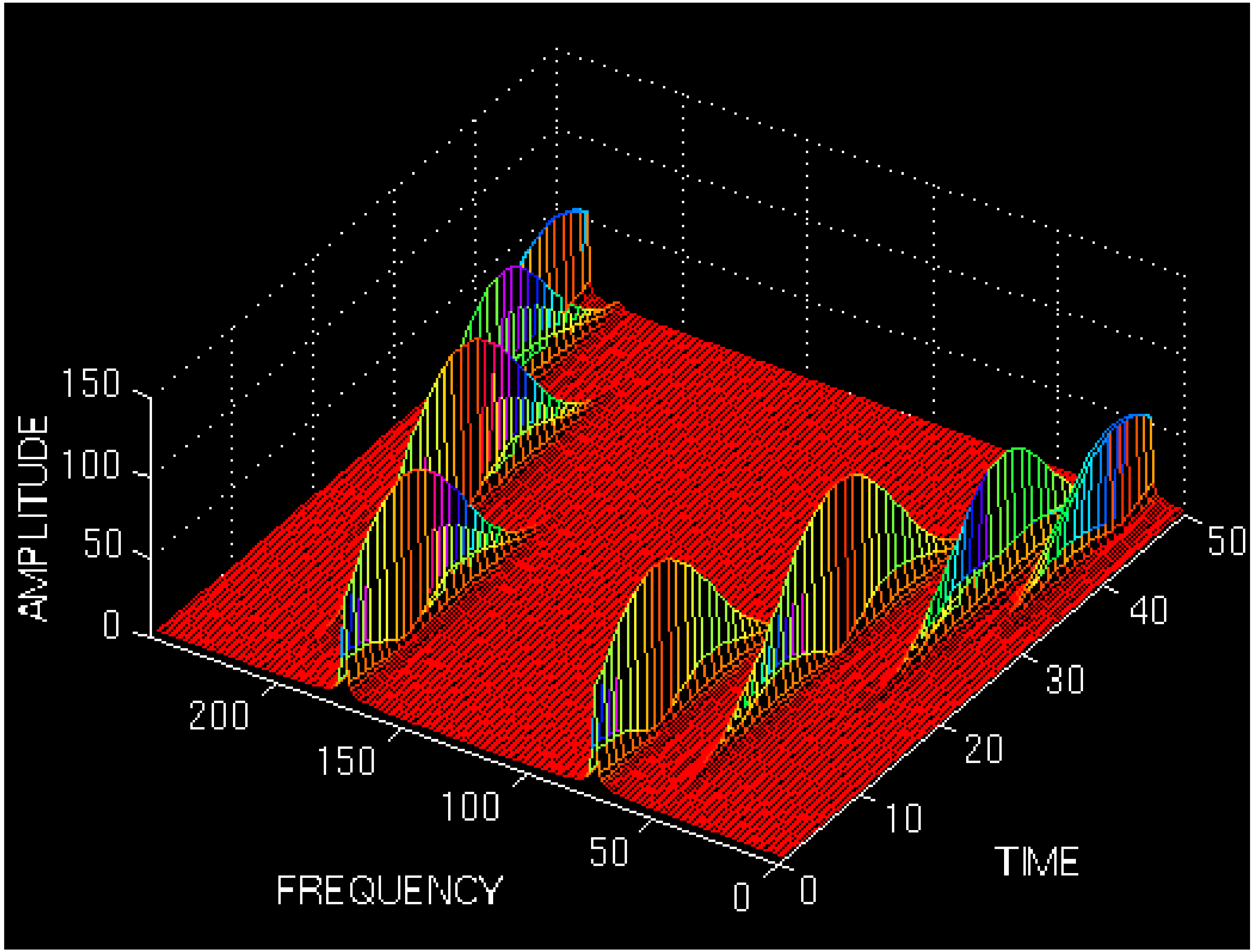
$a = 0.01$

STFT Example



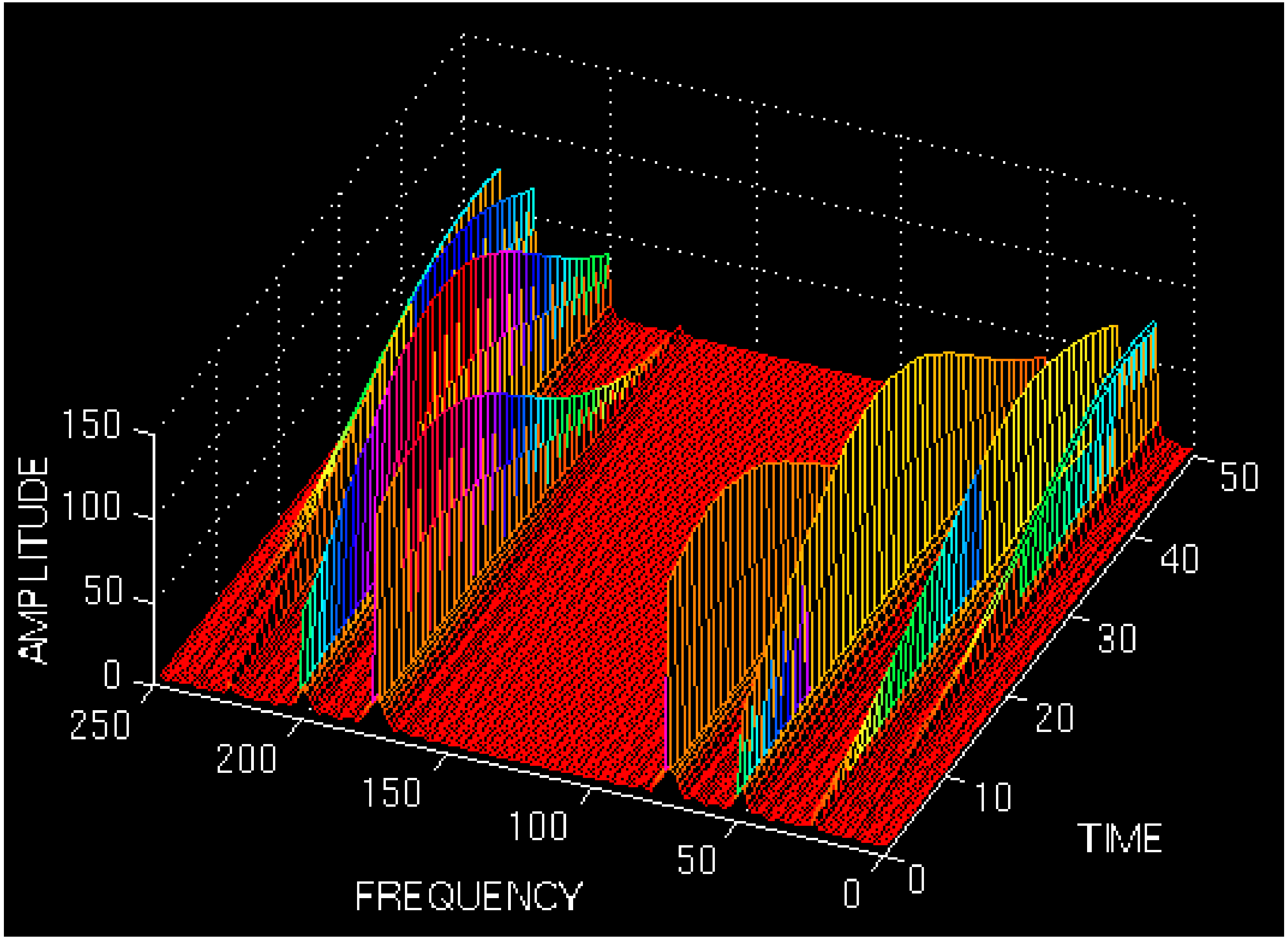
a = 0.001

STFT Example



a = 0.0001

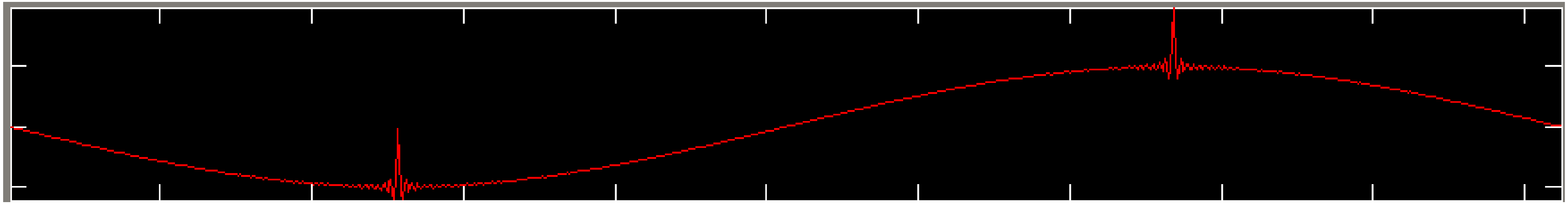
STFT Example



$a = 0.00001$

Time-Frequency Resolution

- Time – frequency resolution problem with STFT
  - Analysis window dictates both time and frequency resolutions, once and for all
  - Narrow window → Good time resolution
  - Narrow band (wide window) → Good frequency resolution
- When do we need good time resolution, when do we need good frequency resolution?
  - Consider the following signal…



Heisenberg Principle

$$\Delta t \cdot \Delta f \geq \frac{1}{4\pi}$$

Time resolution: How well two spikes in time can be separated from each other in the transform domain

Frequency resolution: How well two spectral components can be separated from each other in the transform domain

Both time and frequency resolutions cannot be arbitrarily high!!!

→→ We cannot precisely know at what time instance a frequency component is located. We can only know what *interval of frequencies* are present in which *time intervals*

The Wavelet Transform

- Overcomes the preset resolution problem of the **STFT** by using a variable length window
- Analysis windows of different lengths are used for different frequencies:
    - Analysis of high frequencies → Use narrower windows for better time resolution
    - Analysis of low frequencies → Use wider windows for better frequency resolution
- This works well, if the signal to be analyzed mainly consists of slowly varying characteristics with occasional short high frequency bursts.
- Heisenberg principle still holds!!!
- The function used to window the signal is called the **wavelet**

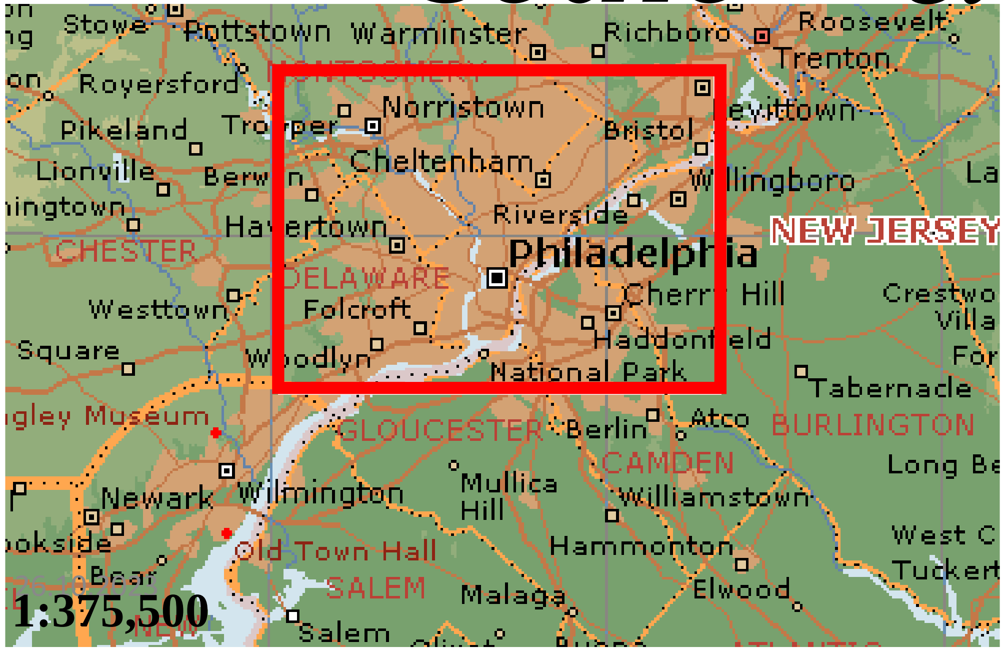
Scale & Translation

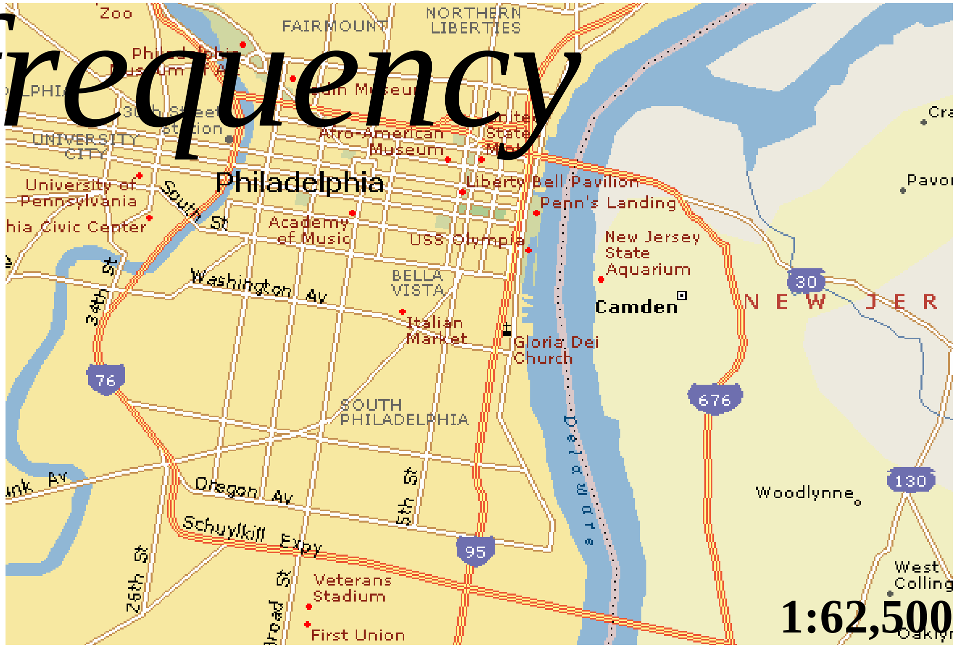
- **Translation** → time shift

- **Scaling** → Similar meaning of scale in maps
  - Large scale: Overall view, long term behavior
  - Small scale: Detail view, local behavior
- $f(t) \rightarrow f(a \cdot t),\ a > 0$
  - If $0 < a < 1$ → dilation, expansion → lower frequency
  - If $a > 1$ → contraction → higher frequency
- $f(t) \rightarrow f(t/a),\ a > 0$
  - If $0 < a < 1$ → contraction → low scale (high frequency)
  - If $a > 1$ → dilation, expansion → large scale (lower frequency)

Scale & Translation

$$scale \ \alpha \ \frac{1}{frequency}$$





- The kernel functions used in **Wavelet transform** are all obtained from one prototype function, by scaling and translating the prototype function.
- This prototype is called the **mother wavelet**

Translation parameter

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$$

Scale parameter

$\frac{1}{\sqrt{a}}$: Normalization factor to ensure that all wavelets have the same energy

$$\int_{-\infty}^{\infty} \left|\psi_{(a,b)}(t)\right|^2 dt = \int_{-\infty}^{\infty} \left|\psi_{(1,0)}(t)\right|^2 dt = \int_{-\infty}^{\infty} \left|\psi(t)\right|^2 dt$$

$$\psi_{1,0}(t) = \psi(t)$$

Continuous Wavelet Transform

Mother wavelet

Translation

$$CWT_x^{(\psi)}(a,b) = W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \cdot \psi^* \left(\frac{t-b}{a} \right) dt$$

CWT of $x(t)$ at scale *a* and translation *b*
Note: low scale → high frequency

Normalization factor

Scaling: Changes the support of the wavelet based on the scale (frequency)

Alternate Notation

Translation parameter, measure of time

Scale parameter, measure of frequency

A normalization constant

Signal to be analyzed

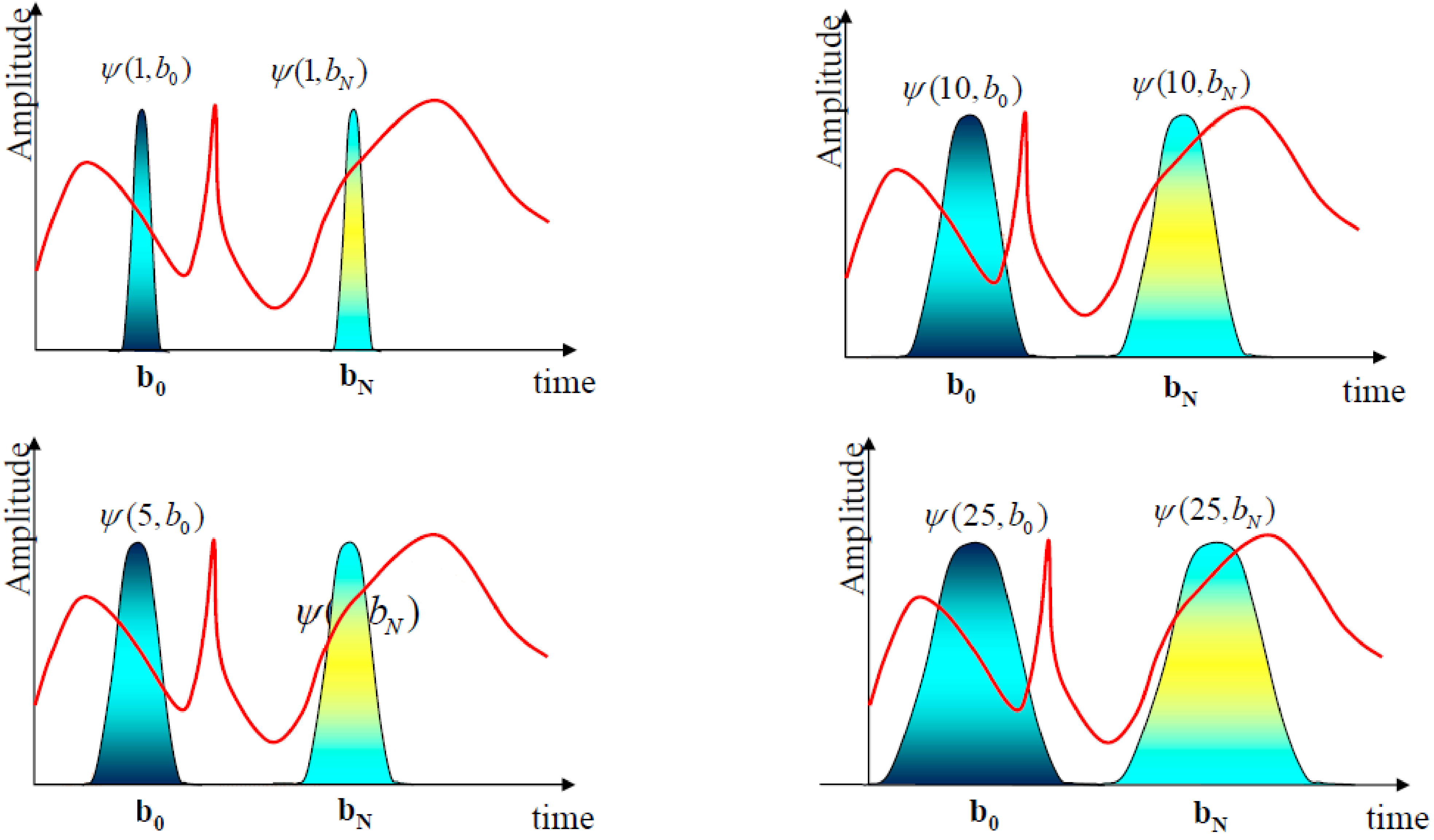
$$CWT_x^{\psi}(\tau, s) = \Psi_x^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int_t x(t) \cdot \psi^* \left(\frac{t - \tau}{s} \right) dt$$

Continuous wavelet transform
of the signal $x(t)$ using the analysis wavelet $\psi(.)$

The mother wavelet. All kernels are obtained by translating (shifting) and/or
scaling the mother wavelet

scale = 1/frequency

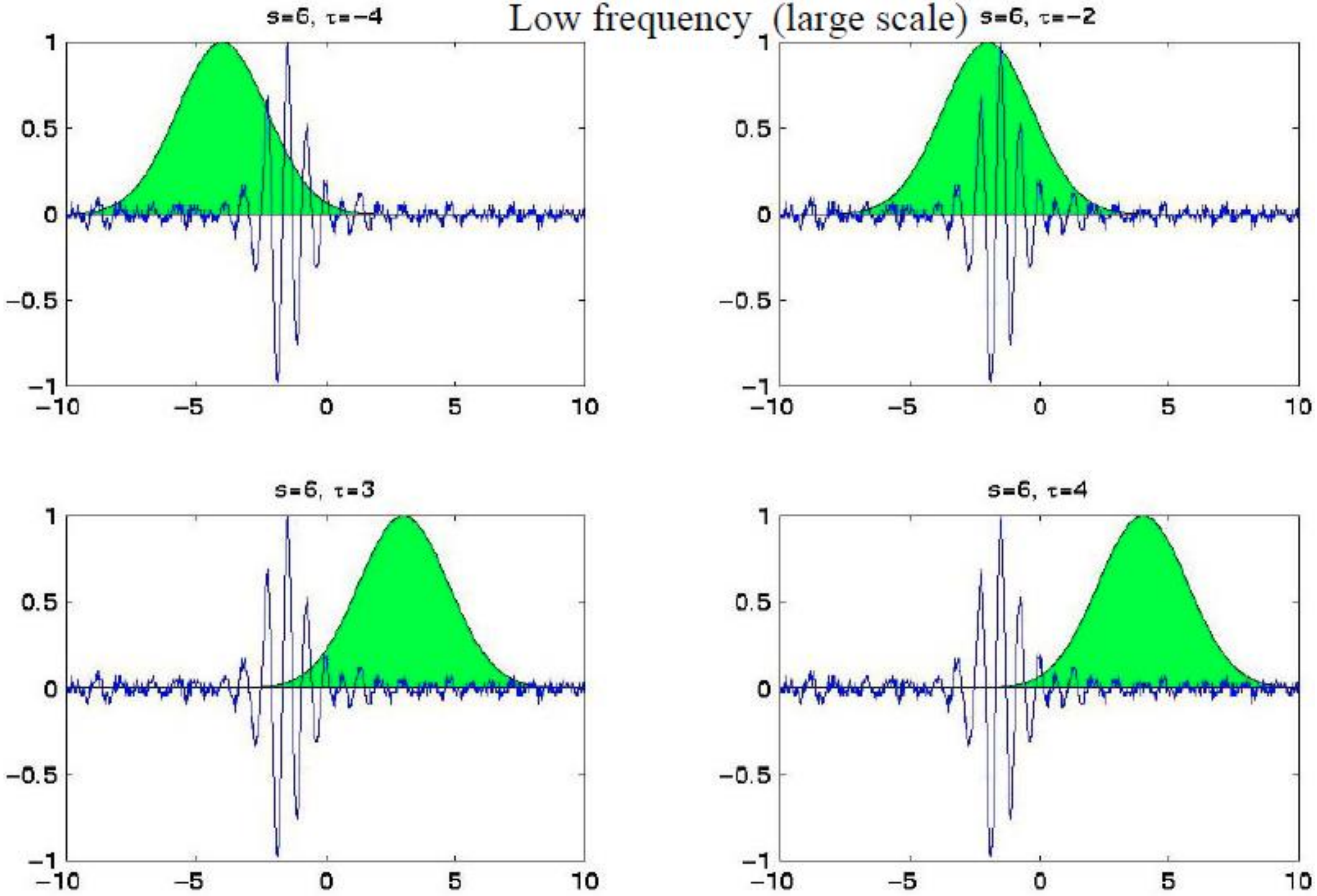
Computation of CWT



$$CWT_x^{(\psi)}(a,b) = W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \cdot \psi^*\left(\frac{t-b}{a}\right) dt$$

CWT at Work

Low frequency (large scale)



$$CWT_x^{\psi}(\tau,s)=\Psi_x^{\psi}(\tau,s)=\frac{1}{\sqrt{|s|}}\int_t x(t)\cdot\psi^*\left(\frac{t-\tau}{s}\right)dt$$

Why Wavelet?

- We require that the wavelet functions, at a minimum, satisfy the following:

$$\int_{-\infty}^{\infty} \psi(t)dt = 0$$

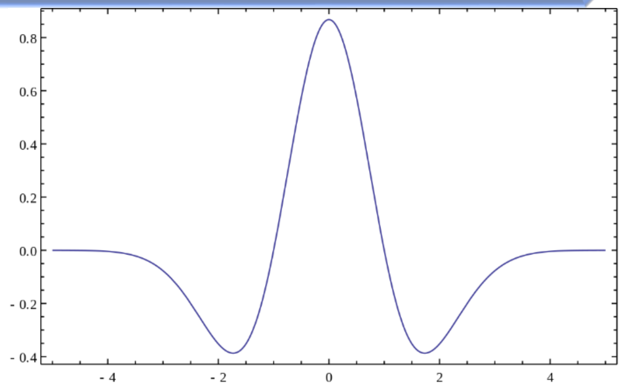
Wave…

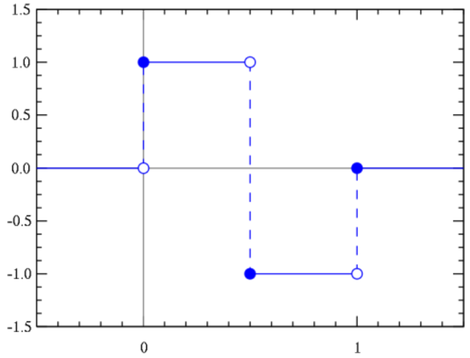
$$\int_{-\infty}^{\infty} |\psi(t)|^2 \, dt < \infty$$

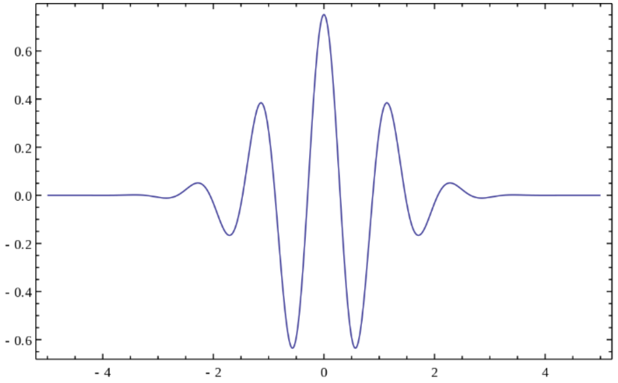
…let

What Do Wavelets Look Like??

- **Mexican Hat** (**Ricker** or **Marr**) Wavelet
- **Haar** Wavelet
- **Morlet** (**Gabor**) Wavelet







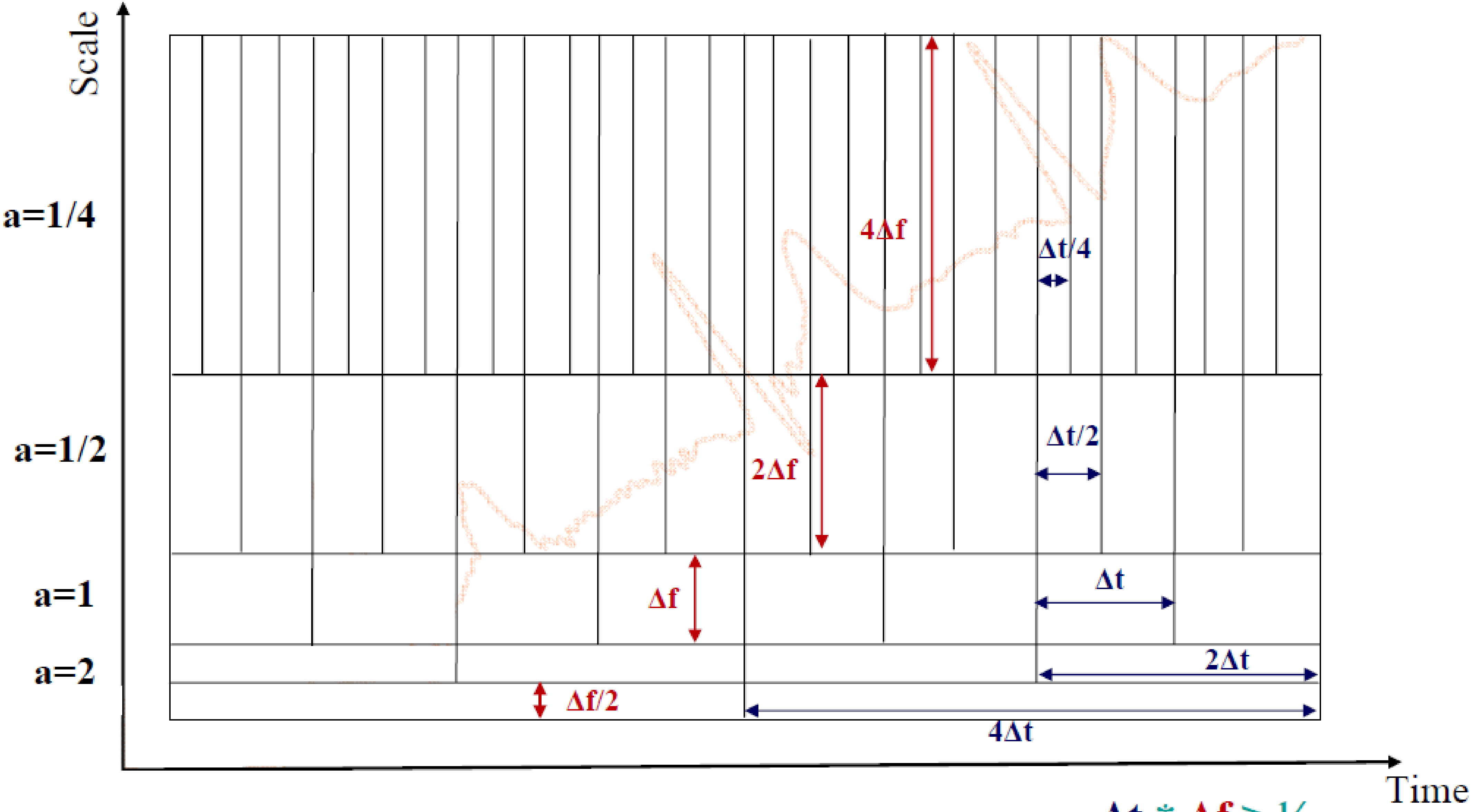
Mexican and **Morlet** have an effectively finite support, whereas **Haar** has a strictly finite support.

$$\psi_{\text{Mex}}(t) = \left(\frac{2}{\sqrt{3}}\pi^{-1/4}\right)\left(1-t^2\right)e^{-t^2/2}$$

$$\psi_{\text{haar}}(t) = \begin{cases} 1, & 0 \le t < 1/2 \\ -1, & 1/2 \le t < 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\psi_{\text{morlet}}(t) = e^{j\omega_0 t}e^{-t^2/2}, \quad \omega_0 = 5.336$$

Time Frequency Grid in CWT



$$\Delta t * \Delta f \geq \frac{1}{2}$$

Scale vs. Frequency

- Can we obtain the **frequency** corresponding to a **scale**?
  - The answer depends on the wavelet used
  - We really should be talking about **pseudo-frequency**, since "**frequency**" does not appear in the analysis.
  - Nevertheless,

$$F_a = \frac{F_\psi}{a \cdot T_s}$$

Center frequency of the wavelet ψ in Hz.

Pseudo-frequency corresponding to scale a

Sampling period of the signal being analyzed

The scale at which we want the frequency

The CWT Transform As a Correlation

$$CWT_x^{(\psi)}(a,b) = W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \cdot \psi^*\left(\frac{t-b}{a}\right) dt$$

- Now, in the L^2 space (of square integrable functions) an **inner product** is defined as

$$< f(t), g(t) >= \int f(t) g^*(t) dt$$

then

$$W(a,b) =< x(t), \psi_{a,b}(t) >$$

- Also, **cross correlation** is defined as:

$$R_{xy}(\tau) = \int x(t) \cdot y^*(t-\tau) dt$$
$$=< x(t), y(t-\tau) >$$

then

$$W(a,b) =< x(t), \psi_{a,0}(t-b) >$$
$$= R_{x,\psi_{a,0}}(b)$$

- Meaning of wavelets: $W(a,b)$ is the **cross correlation** of the signal $x(t)$ with the mother wavelet at scale a, at the lag of b. If $x(t)$ is similar to the mother wavelet at this scale and lag, then $W(a,b)$ will be large.

Inverse CWT

$$x(t) = \frac{1}{C} \int_{a=0}^{\infty} \int_{b=-\infty}^{\infty} \frac{1}{a^2} W(a,b) \cdot \psi_{a,b}(t)\, db\, da$$

$$C = \int_{-\infty}^{\infty} \frac{|\Psi(\omega)|}{|\omega|} d\omega$$

$$0 < C < \infty$$

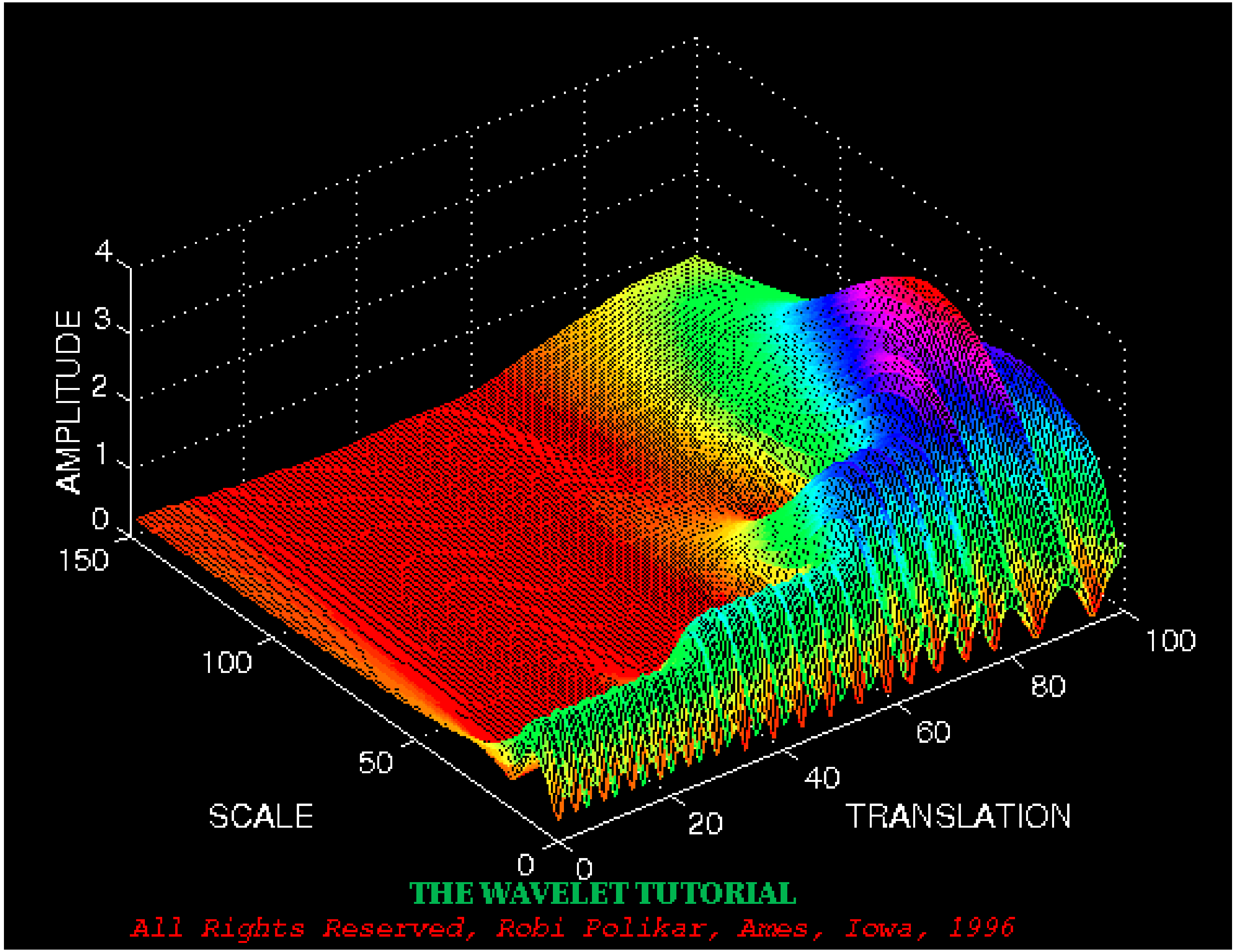
which implies that $\int_{-\infty}^{\infty} \psi(t)\, dt = 0$

However, inverse transform is rarely performed in continuous time, because the CWT is infinitely redundant – More about this later!

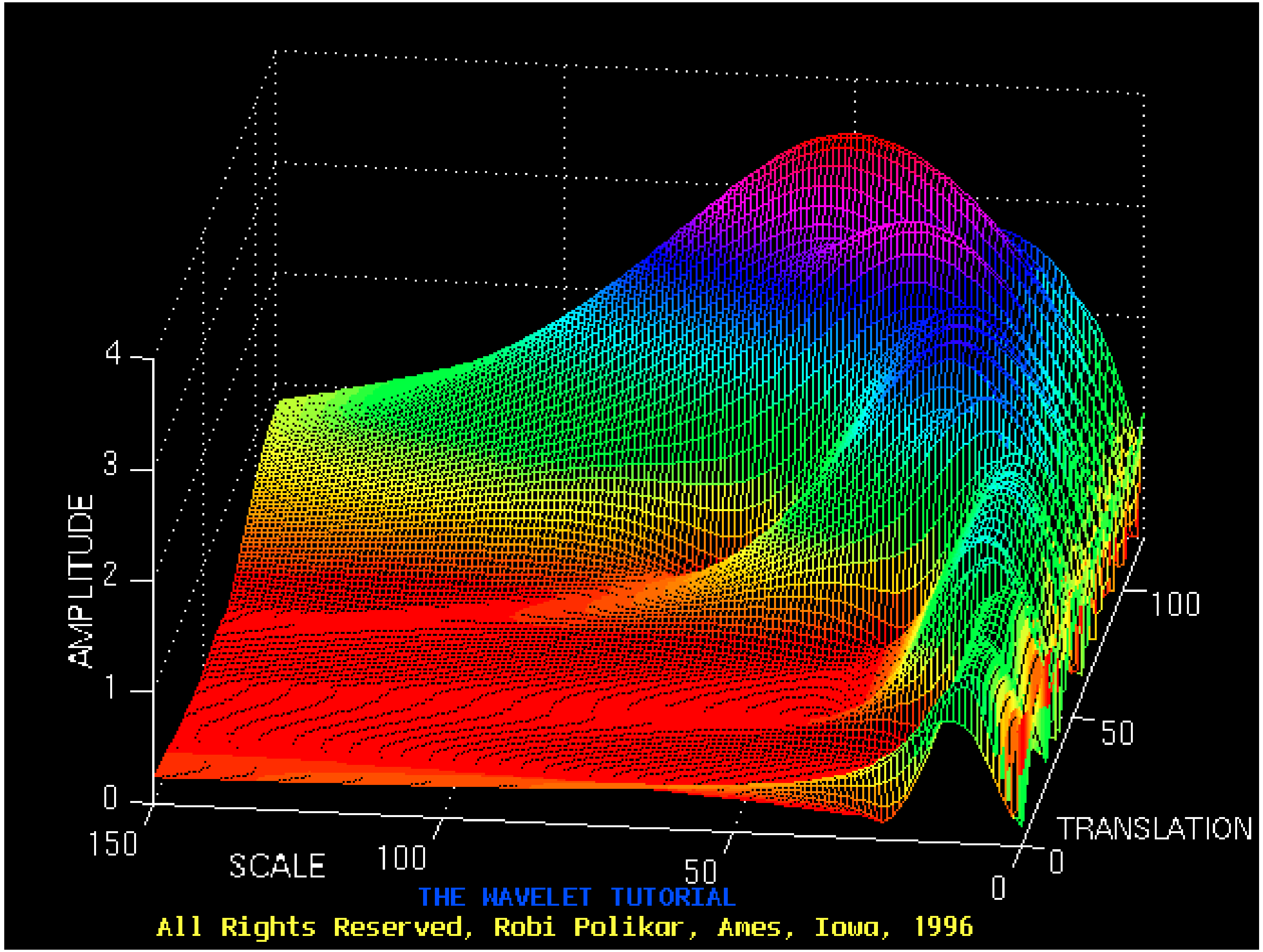
Example

30 Hz
20 Hz
10 Hz
5 Hz
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
200
400
600
800
1000
Time, ms

Example

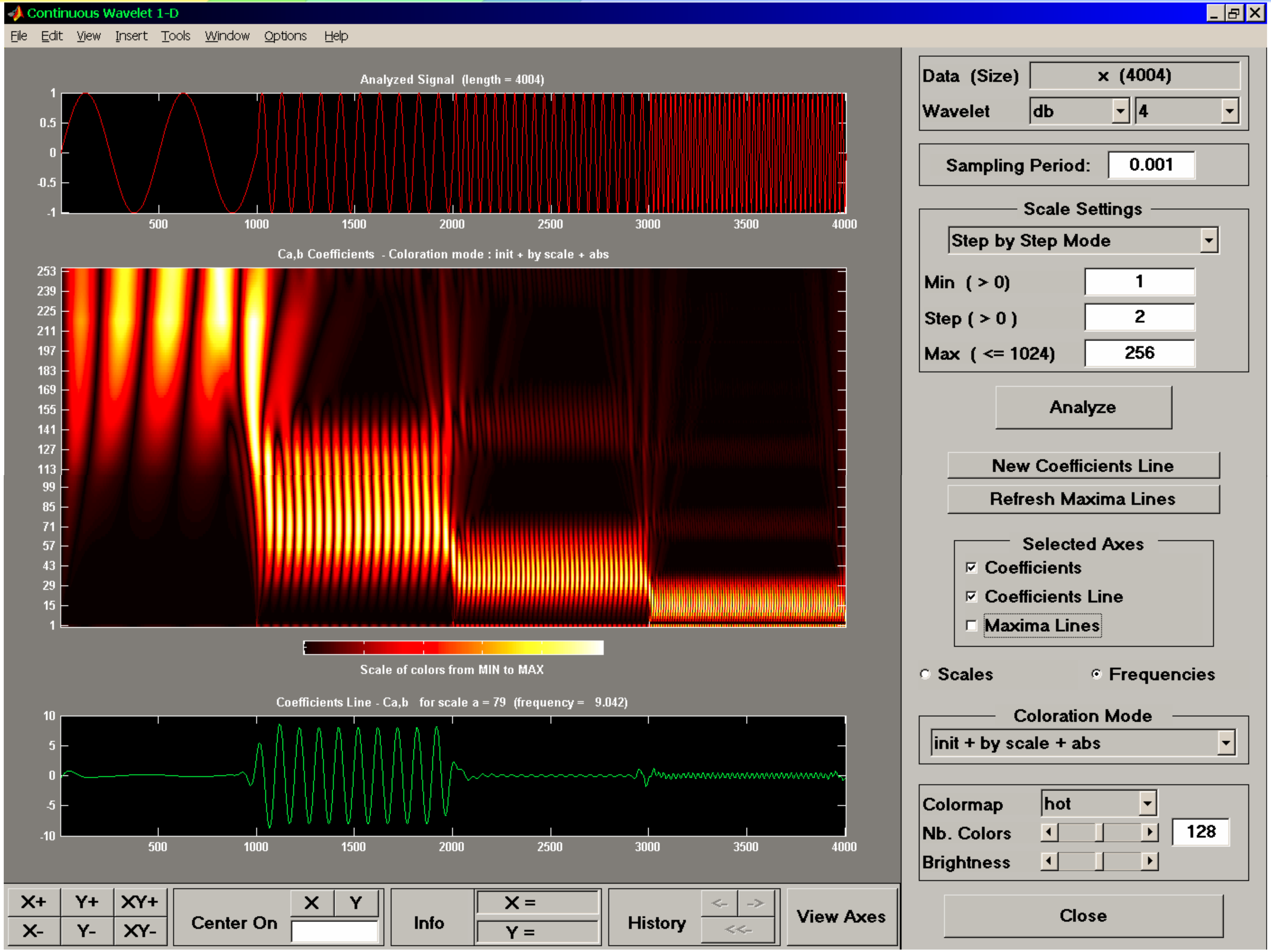

AMPLITUDE
4
3
2
1
0
150
100
50
0
SCALE
0
20
40
60
80
100
TRANSLATION
THE WAVELET TUTORIAL
All Rights Reserved, Robi Polikar, Ames, Iowa, 1996

Example



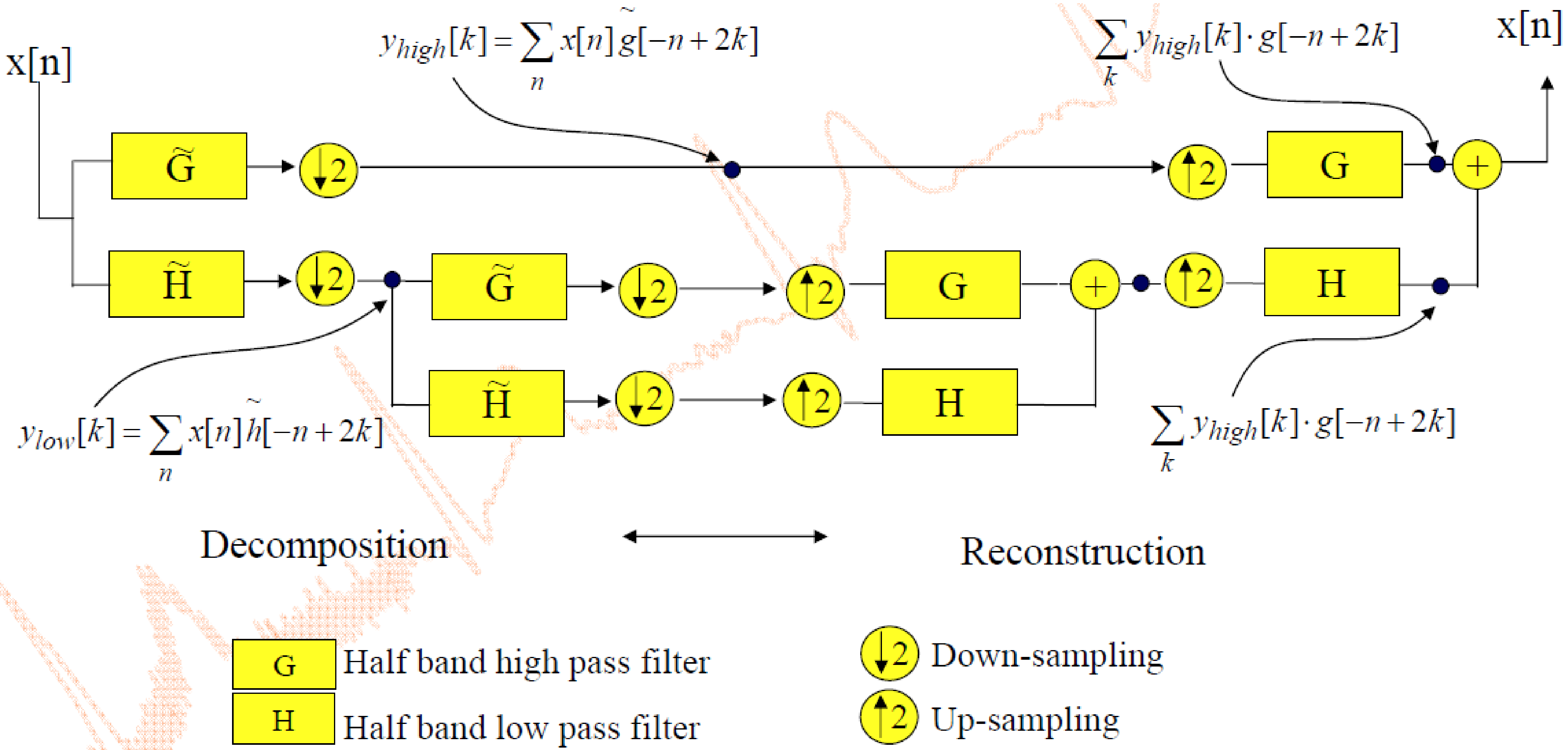
4
3
2
1
0
AMPLITUDE
150
SCALE
100
50
0
100
50
TRANSLATION
0
THE WAVELET TUTORIAL
All Rights Reserved, Robi Polikar, Ames, Iowa, 1996

Matlab Demos on CWT



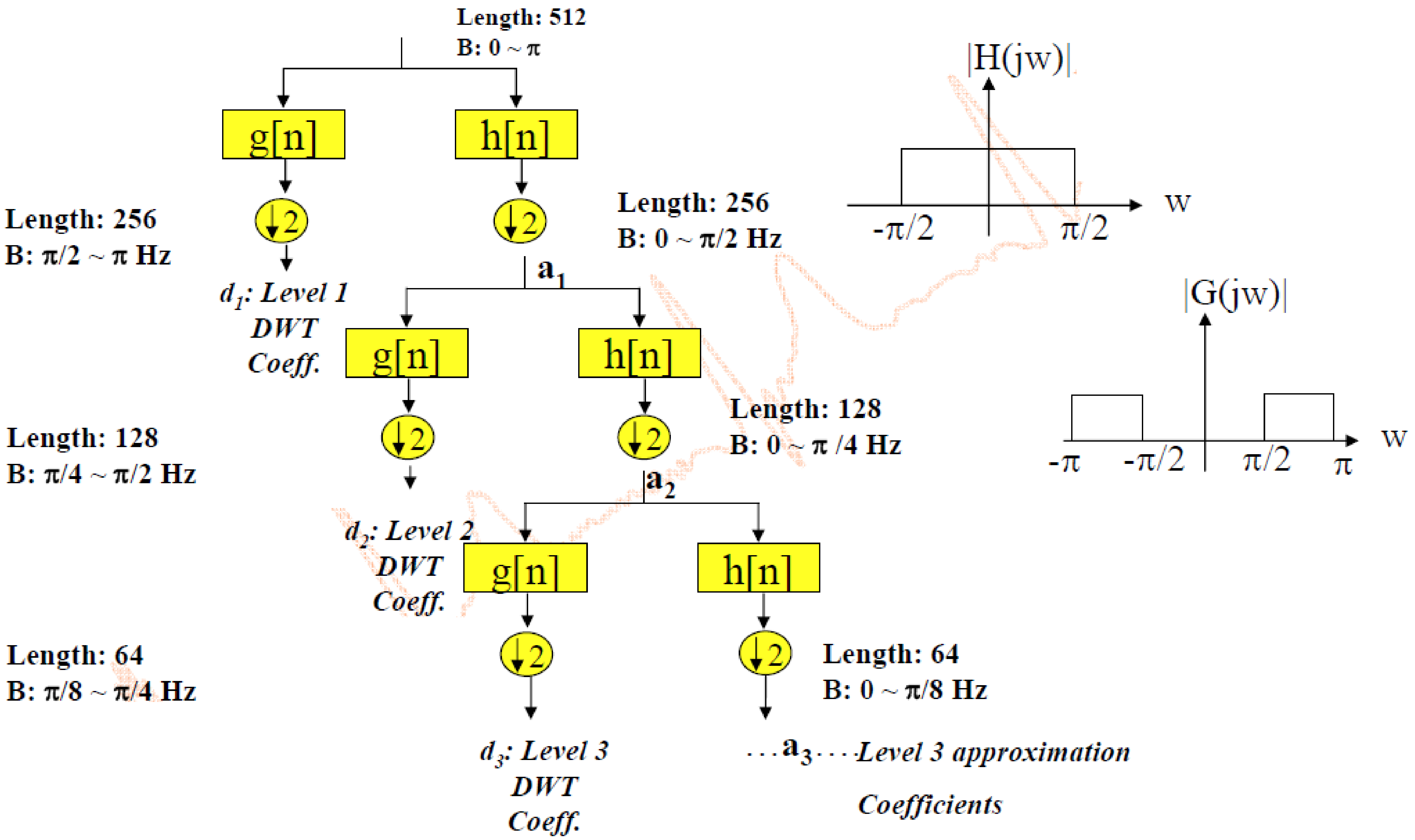
DWT Wavelet Transformation Implementation

- There is a close relationship between the **DWT** and **filter banks**
  - This relationship allows us to compute the **DWT** in a very efficient manner
- In this implementation, the signal is decomposed into its **approximations** and **details** using a series of lowpass and highpass filters.

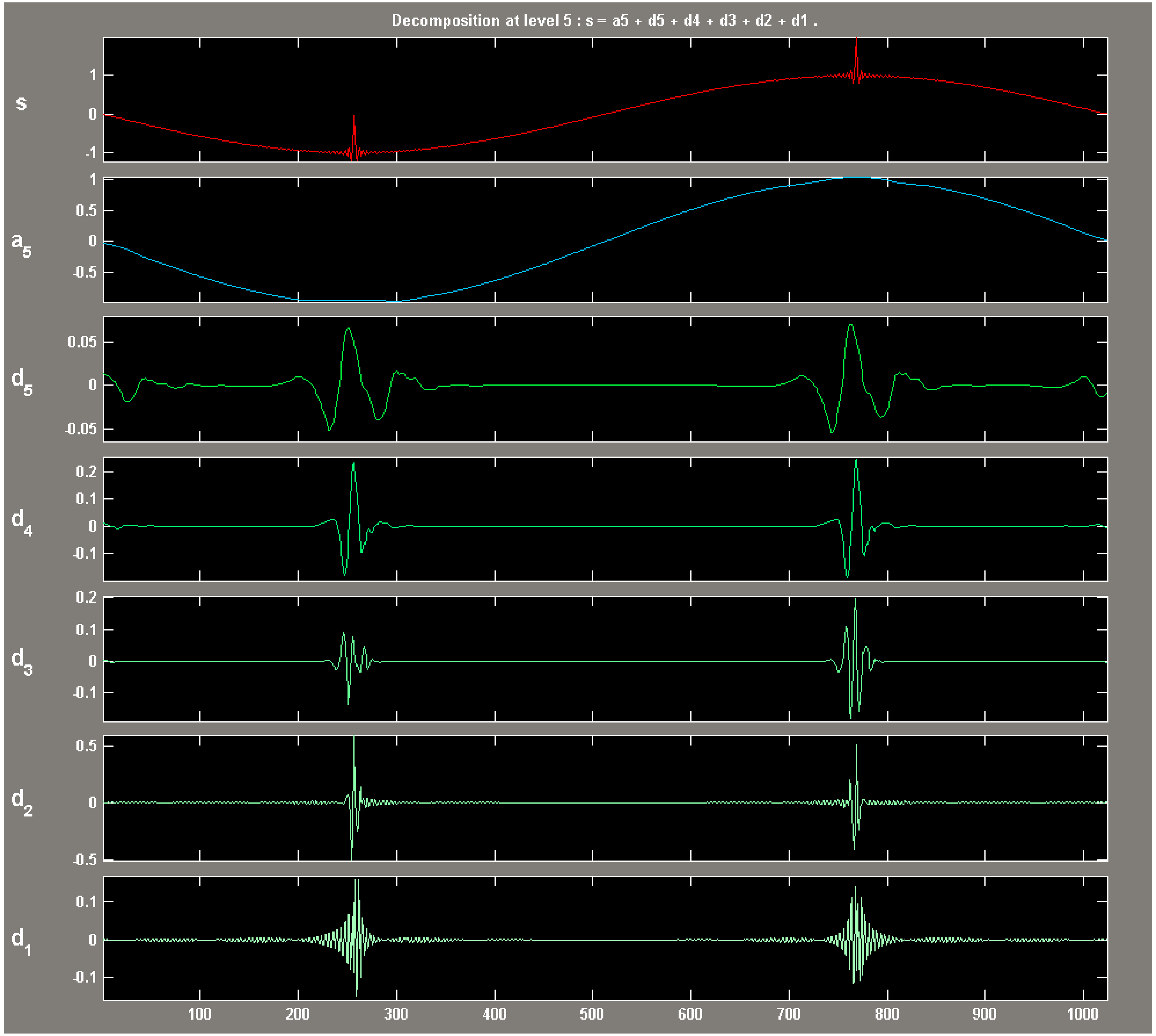


2-level DWT decomposition. The decomposition can be continues as long as there are enough samples for down-sampling.

DWT – Demystified



Details vs. Approximations



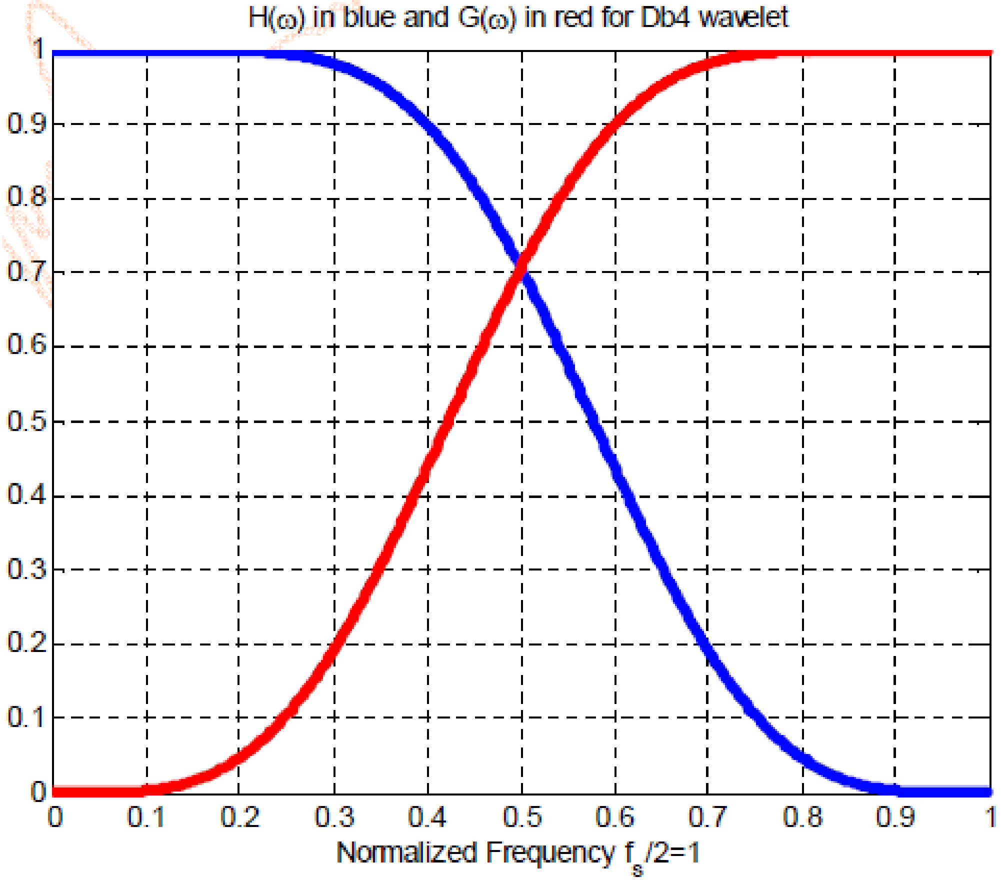
Quadrature Mirror Filters

- It can be shown that

$$\left|H(j\omega)\right| + \left|H(j\omega + \pi/2)\right| = 1$$

$$\left|H(j\omega)\right|^2 + \left|G(j\omega)\right|^2 = 1 \Leftrightarrow \left|\tilde{H}(j\omega)\right|^2 + \left|\tilde{G}(j\omega)\right|^2 = 1$$

that is, $h[\ ]$ and $g[\ ]$ filters are related to each other: In fact, $h[L\text{-}1\text{-}n] = (-1)^n\, g[n]$ that is, $h[\ ]$ and $g[\ ]$ are mirrors of each other, with every other coefficient negated. Such filters are called **quadrature mirror filters**. For example, Daubechies wavelets with 4 vanishing moments.....



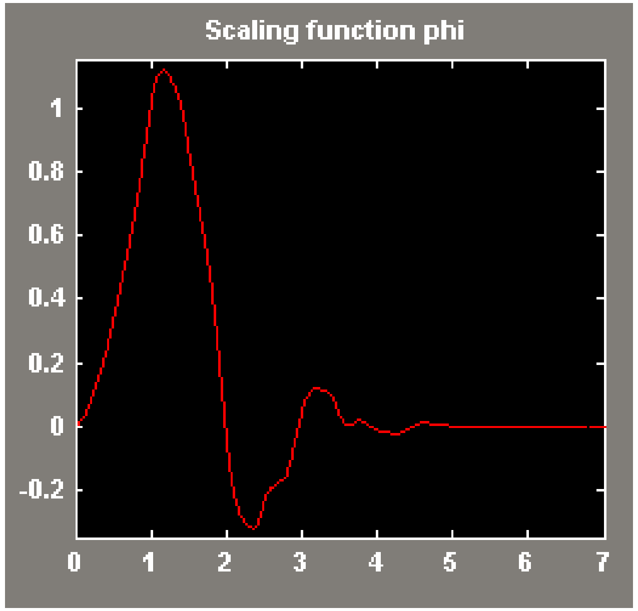
DB-4 Wavelets

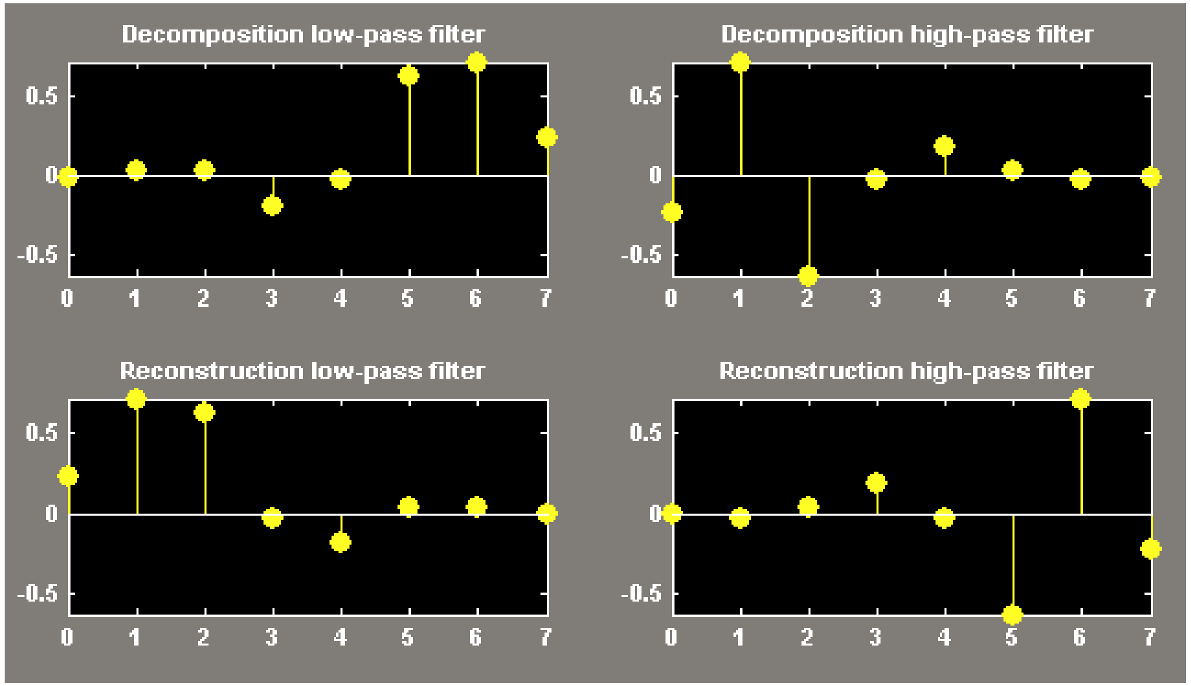
LO_D $\tilde{h}$ =	-0.0106	0.0329	0.0308	-0.1870	-0.0280	0.6309	0.7148	0.2304
HI_D $\tilde{g}$ =	-0.2304	0.7148	-0.6309	-0.0280	0.1870	0.0308	-0.0329	-0.0106
LO_R h =	0.2304	0.7148	0.6309	-0.0280	-0.1870	0.0308	0.0329	-0.0106
HI_R g =	-0.0106	-0.0329	0.0308	0.1870	-0.0280	-0.6309	0.7148	-0.2304

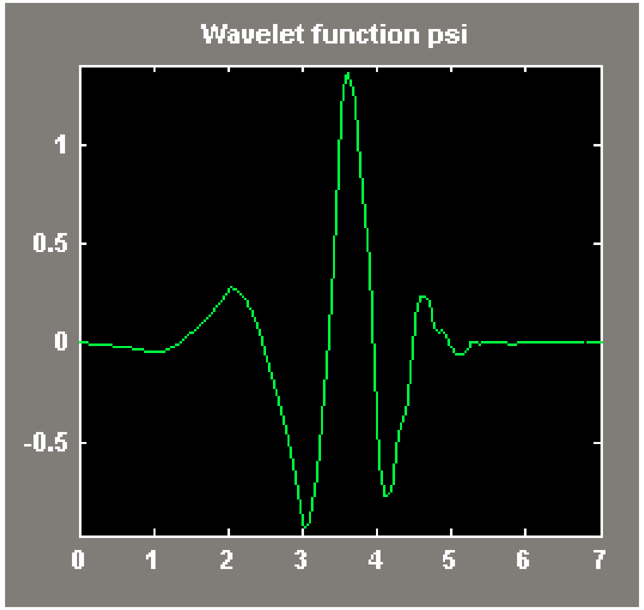
$$h[L-1-n] = (-1)^n g[n]$$

$$h[n] = \tilde{h}[-n], \quad g[n] = \tilde{g}[-n]$$

L = 8, n = 0,1,…,L-1







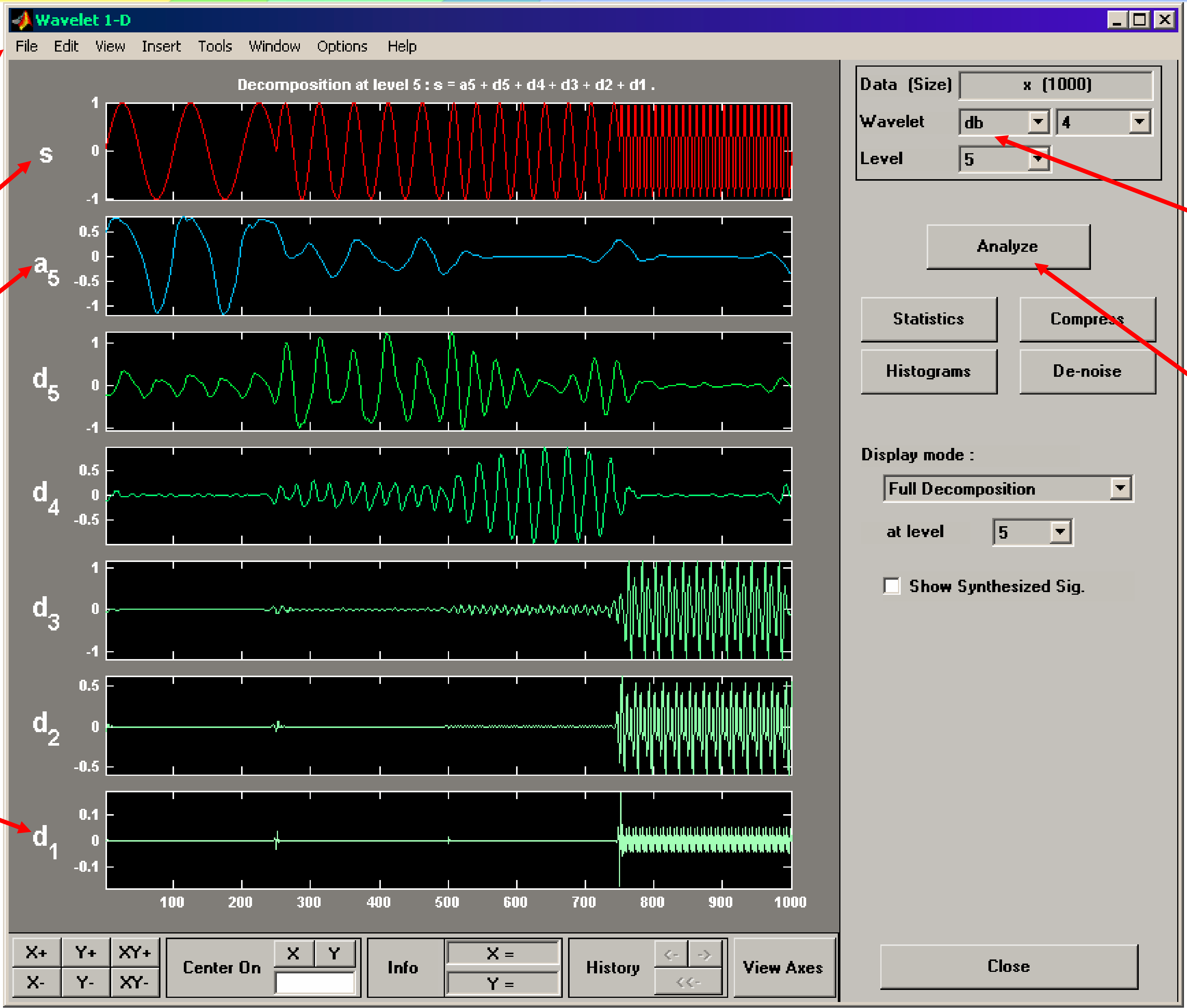
Implementation of DWT on Matlab

Load signal

$s = a_5 + d_5 + \ldots + d_1$

Approx. coeff. at level 5

Level 1 coeff. Highest freq.

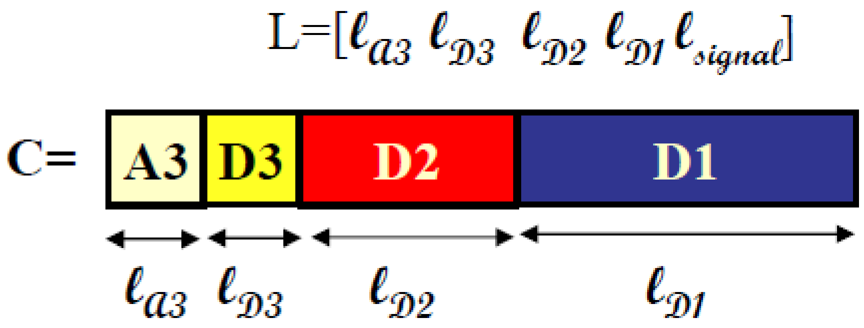


Choose wavelet and number of levels

Hit Analyze button

Matlab Implementation Command Line

- Single level decomposition and reconstruction
  - [cA1 cD1]=**dwt**(x, 'db4') : single level decomposition
  - x=**idwt**(cA1, cD1, 'db4') : single level inverse transform
- Direct reconstruction of approximation and detail signals
  - D1=**upcoef**('d', cD1, 'db4', 1, N) : one-level reconstruction of detail signal from detail coefficient
  - A1=**upcoef**('a', cA1, 'db4', 1, N) : one level reconstruction of approximation signal from approximation coefficients
- Multi-level decomposition & reconstruction
  - [C L]=**wavedec**(x, K, 'db4') : k-level decomposition
  - A0=**waverec**(C, L, 'db4') : reconstruct the signal
- Extracting coefficients
  - cA3=**appcoef**(C, L, 'db4', 3);
  - cD5=**detcoef**(C, L, 5);
- Single branch reconstruction
  - A3=**wrcoef**('a', C, L, 'db4', 3); : Reconstruct level 3 approx. signal
  - D2=**wrcoef**('d', C, L, 'db4', 2); : Reconstruct level 2 detail signal

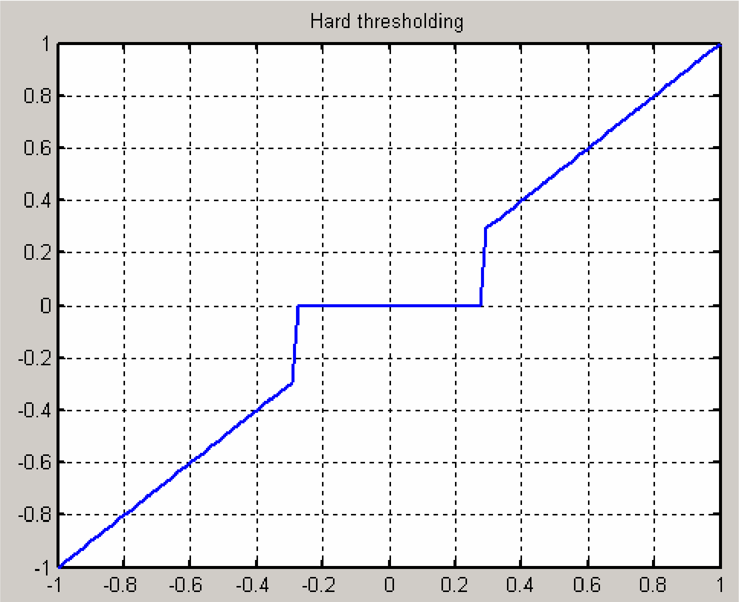


Wavelet Shrinkage Denosing:

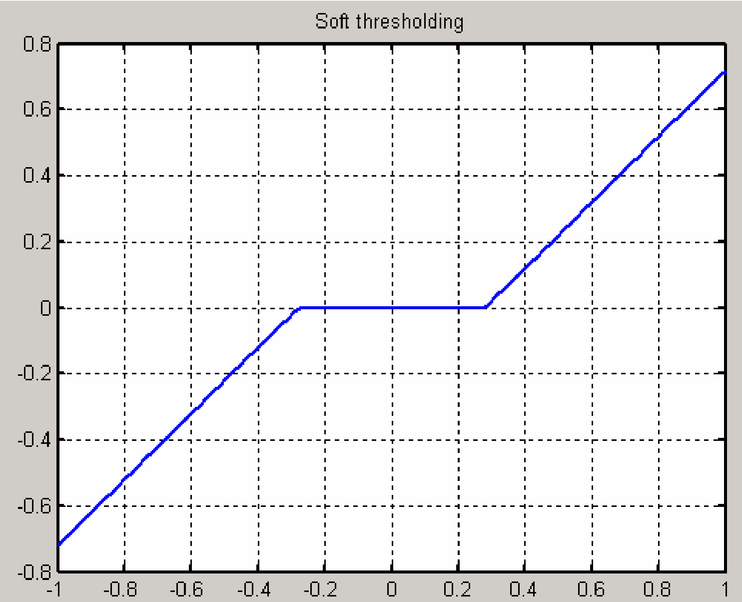
- Based on reducing the values of certain coefficients (at each level) that are believed to correspond to noise.
- Better then regular filtering, because no significant signal information is lost, even when signal and noise spectra overlap!
- Two types of thresholding are used:

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

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



$$\delta = 0.28$$



Three step procedure:

1. Decompose signal using DWT; choose wavelet and number of decomposition levels
2. Shrink coefficients by thresholding (hard /soft). Do we pick a single threshold or pick different thresholds at different levels?
3. Reconstruct the signal from thresholded DWT coefficients

How Do We Choose The Threshold?

- There are many models, such as Stein's unbiased risk estimate, universal threshold estimate, combination of the above two, minimax criterion, etc. Among them, universal threshold estimate is the one used most often.
- According to this model, $x[n] = s[n]+\sigma\, e[n]$, where $s[n]$ is the clean signal, $e[n]$ is the noise, σ^2 is the noise power, and $x[n]$ is the noisy signal. The noise is considered as Additive White Gaussian Noise (AWGN).
- This model estimates the universal threshold (subband dependent, of course) as

$$\delta_k = \sigma_k \sqrt{2\log(N_k)}$$

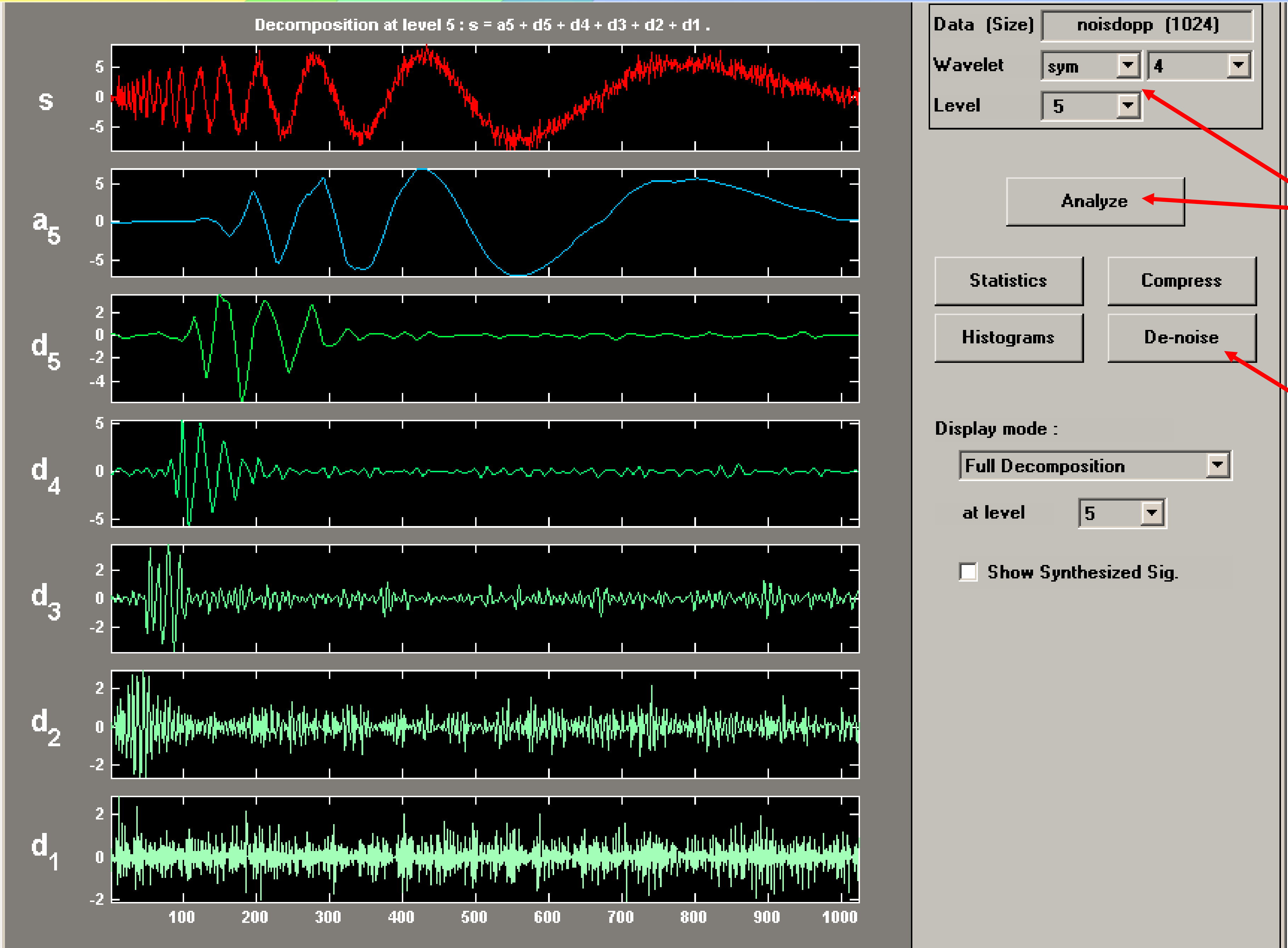
Threshold for subband k

Noise std. deviation for subband k

Length of the DWT coefficients at level k

[XD, CXD, LXD]=wden(X, TPTR, SORH, SCAL, N, 'wname');

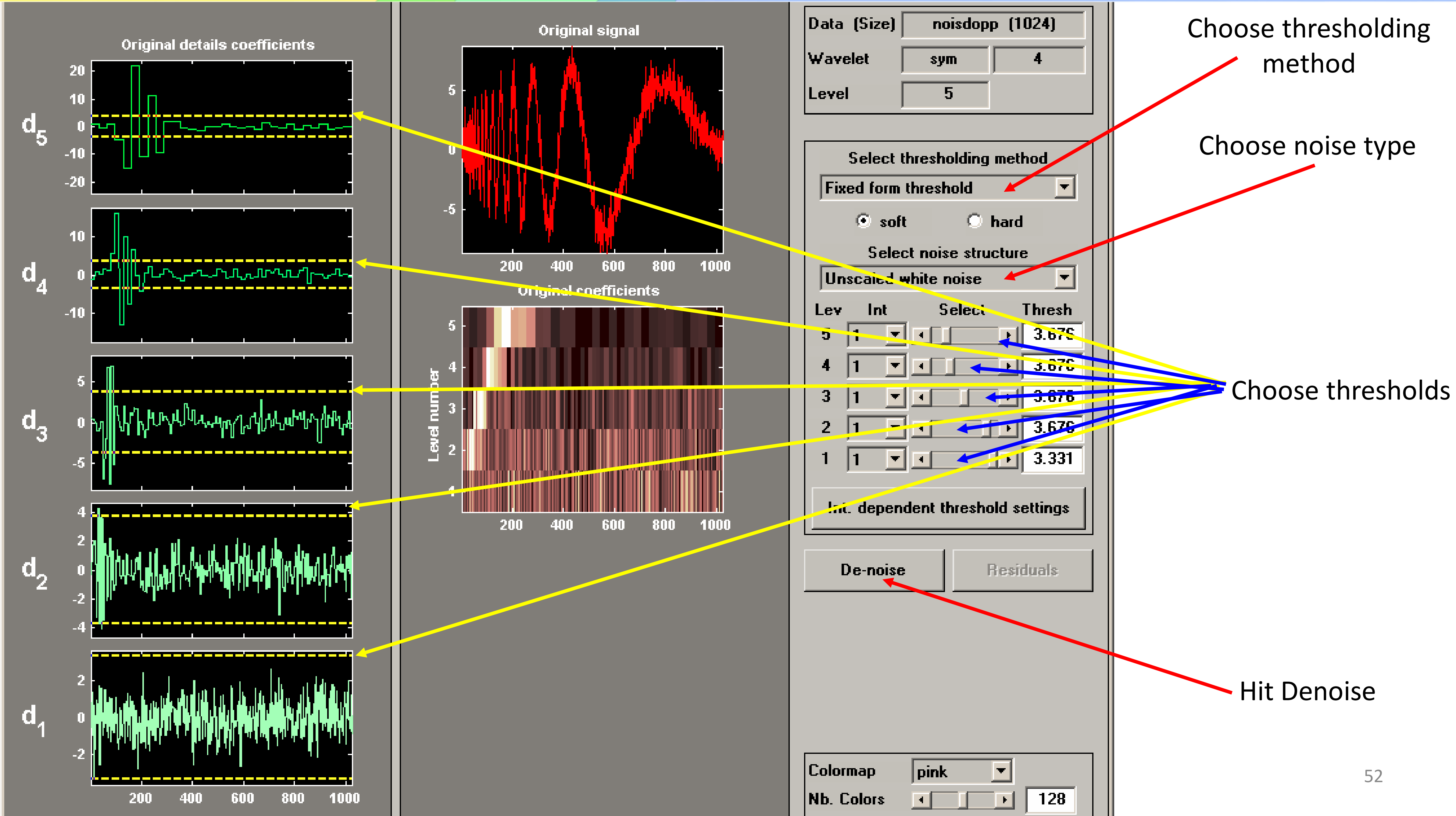
Denoising Implementation in Matlab



First, analyze the signal with appropriate wavelets

Hit Denoise

Denoising Implementation in Matlab



Denoising Using Matlab