ESE 5310: Digital Signal Processing

Lecture 24: April 23, 2024

Wavelet Transform and Compressive Sampling

Penn ESE 5310 Spring 2024 – Khanna
Adapted from M. Lustig, EECS Berkeley

Today

- Wavelet Transform
- Compressive Sampling/Sensing

Discrete Time-Dependent Fourier Transform

$$X[n,\lambda) = \sum_{m=-\infty}^{\infty} x[n+m]w[m]e^{-j\lambda m}$$

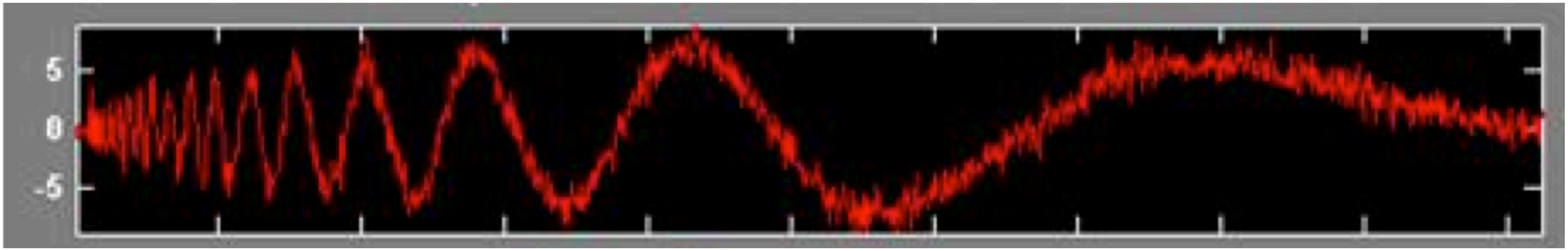
$$X[rR,k] = X[rR, 2\pi k / N) = \sum_{m=0}^{L-1} x[rR+m]w[m]e^{-j(2\pi/N)km}$$

$$X_r[k] = \sum_{m=0}^{L-1} x[rR+m]w[m]e^{-j(2\pi/N)km}$$

Wavelet Transform

Motivation

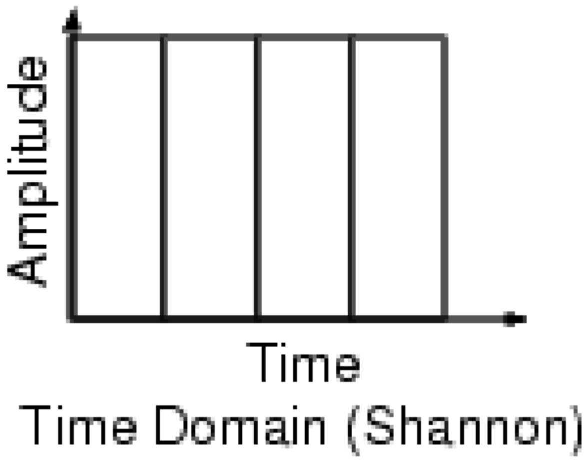
- Some signals obviously have spectral characteristics that vary with time

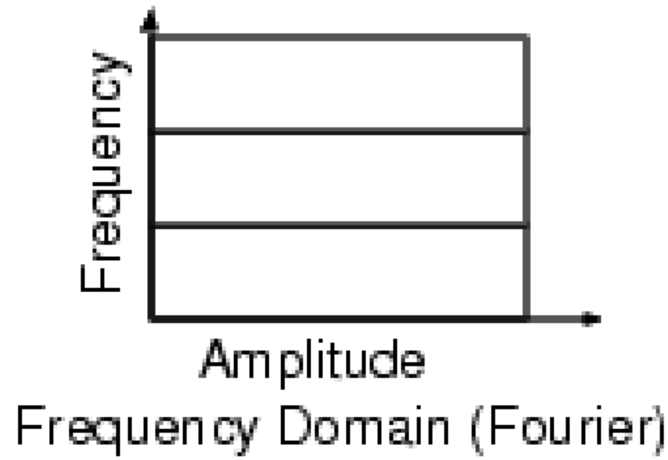


Criticism of Fourier Spectrum

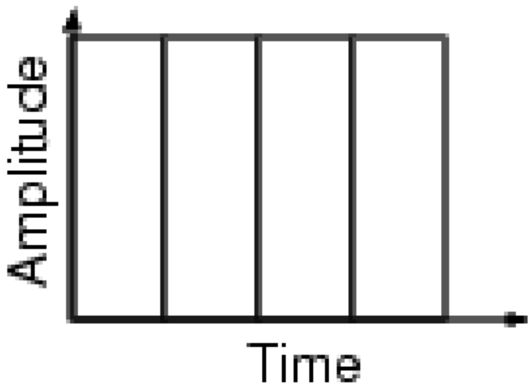
- It's giving you the spectrum of the 'whole time-series'
- Which is OK if the time-series is stationary. But what if it's not?
- We need a technique that can "march along" a time series and that is capable of:
  - Analyzing spectral content in different places
  - Detecting sharp changes in spectral character

Transform Comparison

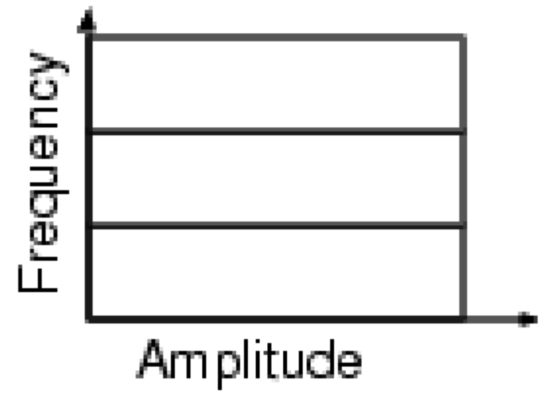




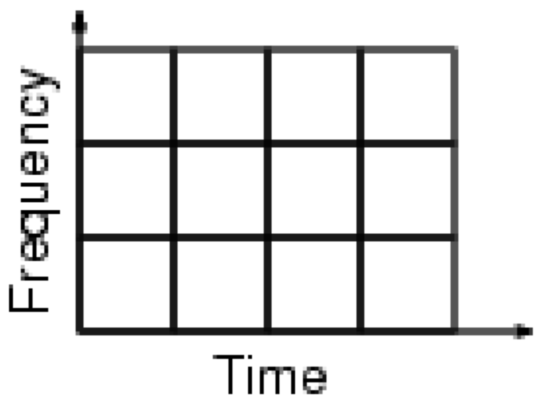
Transform Comparison



Time Domain (Shannon)



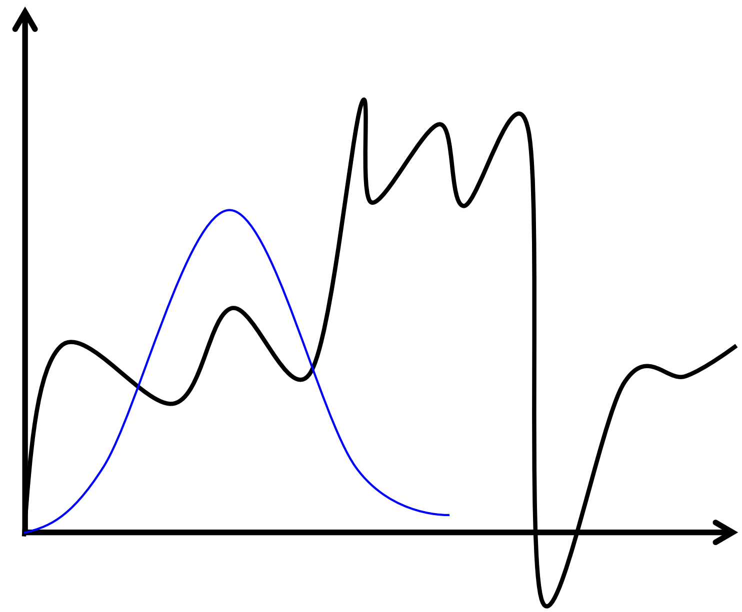
Frequency Domain (Fourier)



STFT (Gabor)

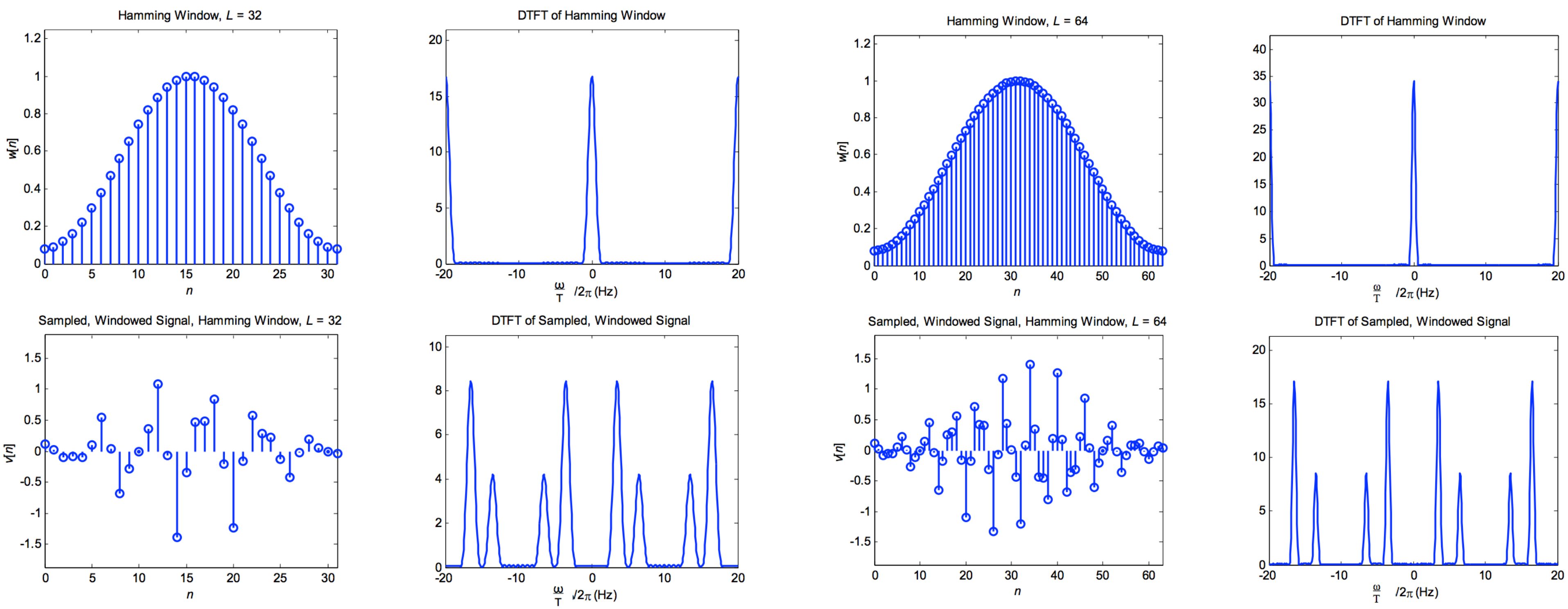
Discrete Time-Dependent FT

- Fixed window size, shift in time (Gabor)

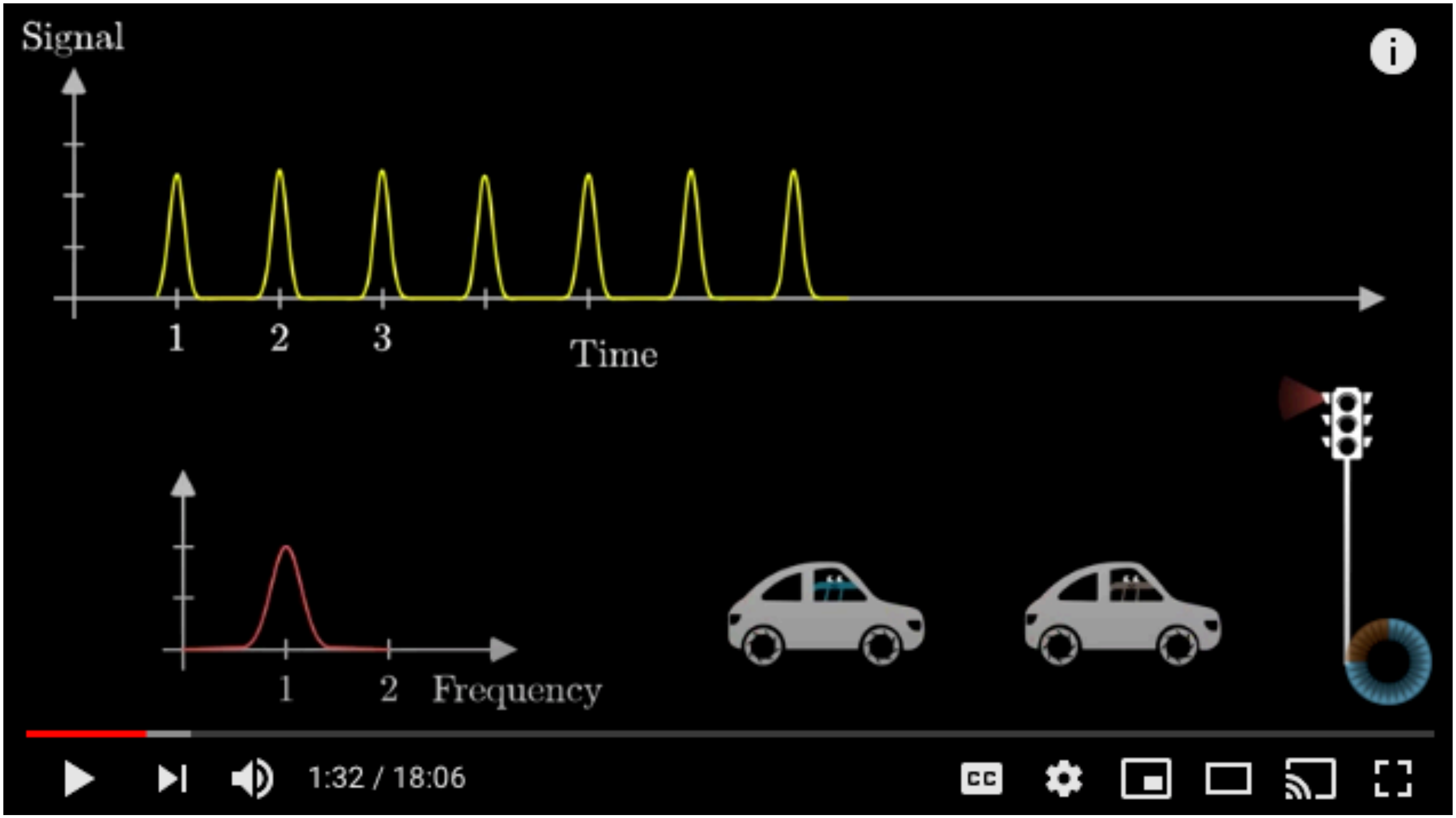


Windowed Sampled CT Signal Example

- As before, the sampling rate is $\Omega s/2\pi = 1/T = 20Hz$
- Hamming Window, L = 32 vs. L = 64



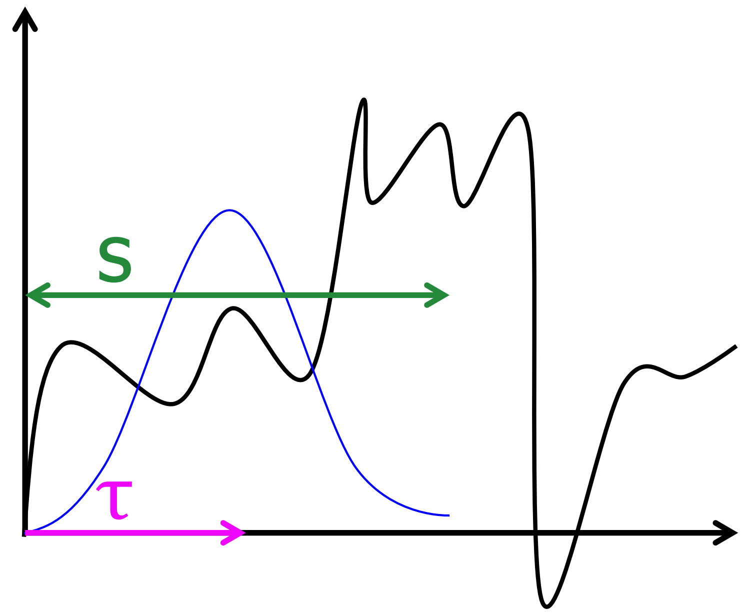
Uncertainty Principle



https://youtu.be/MBnnXbOM5S4?t=49

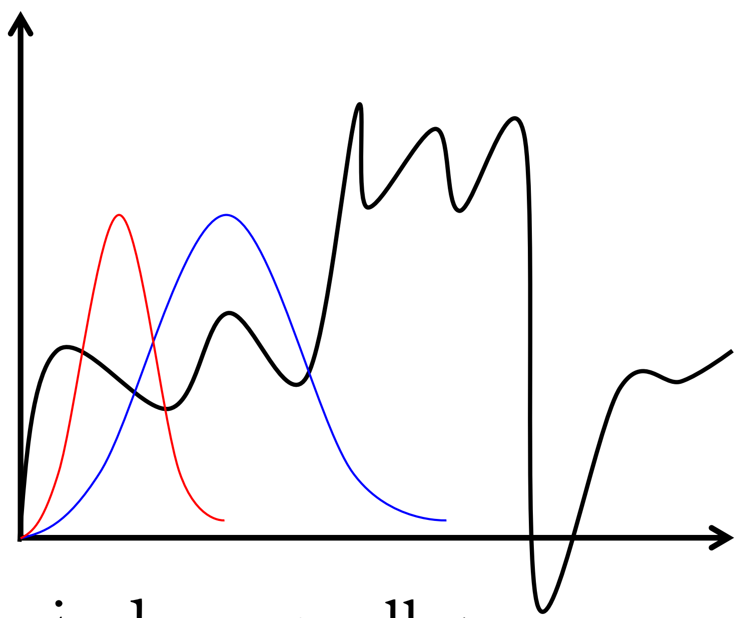
Discrete Time-Dependent FT

- Fixed window size, shift in time (Gabor)



Discrete Time-Dependent FT

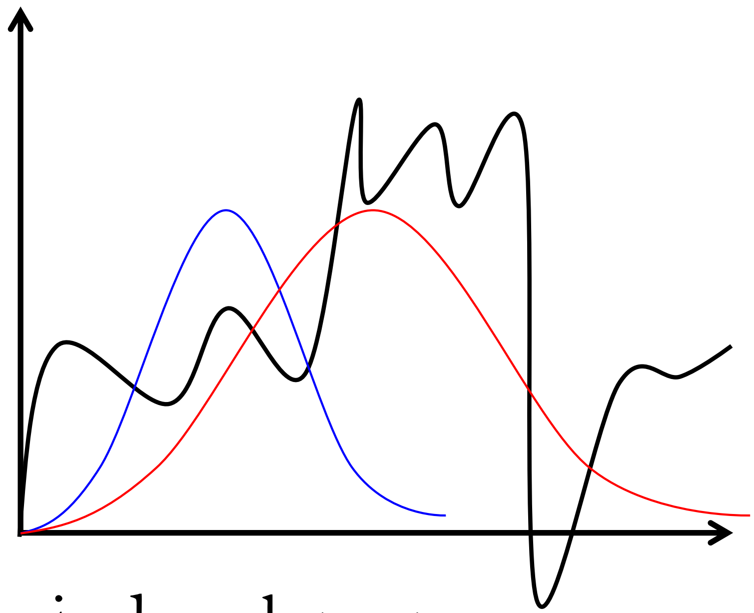
- Fixed window size, shift in time (Gabor)



- Make the window smaller
  - Better localization in time
  - Less spectral resolution

Discrete Time-Dependent FT

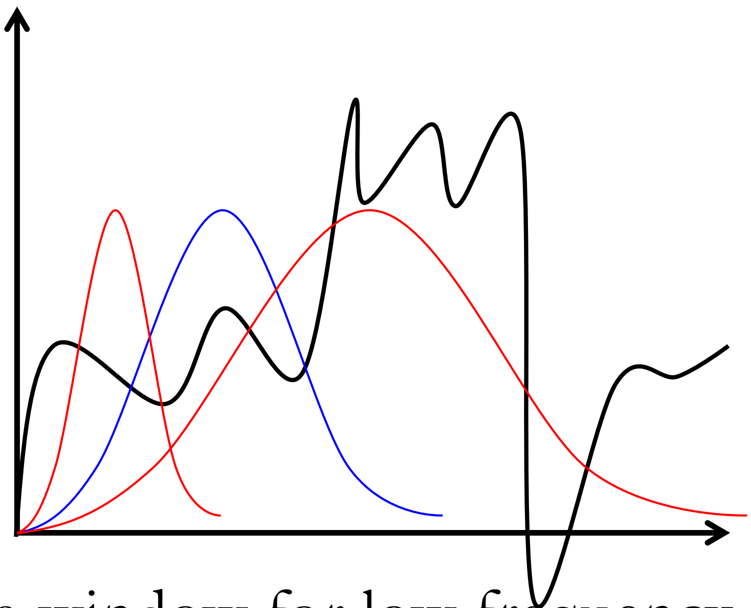
- Fixed window size, shift in time (Gabor)



- Make the window larger
  - Worse localization in time
  - More spectral resolution

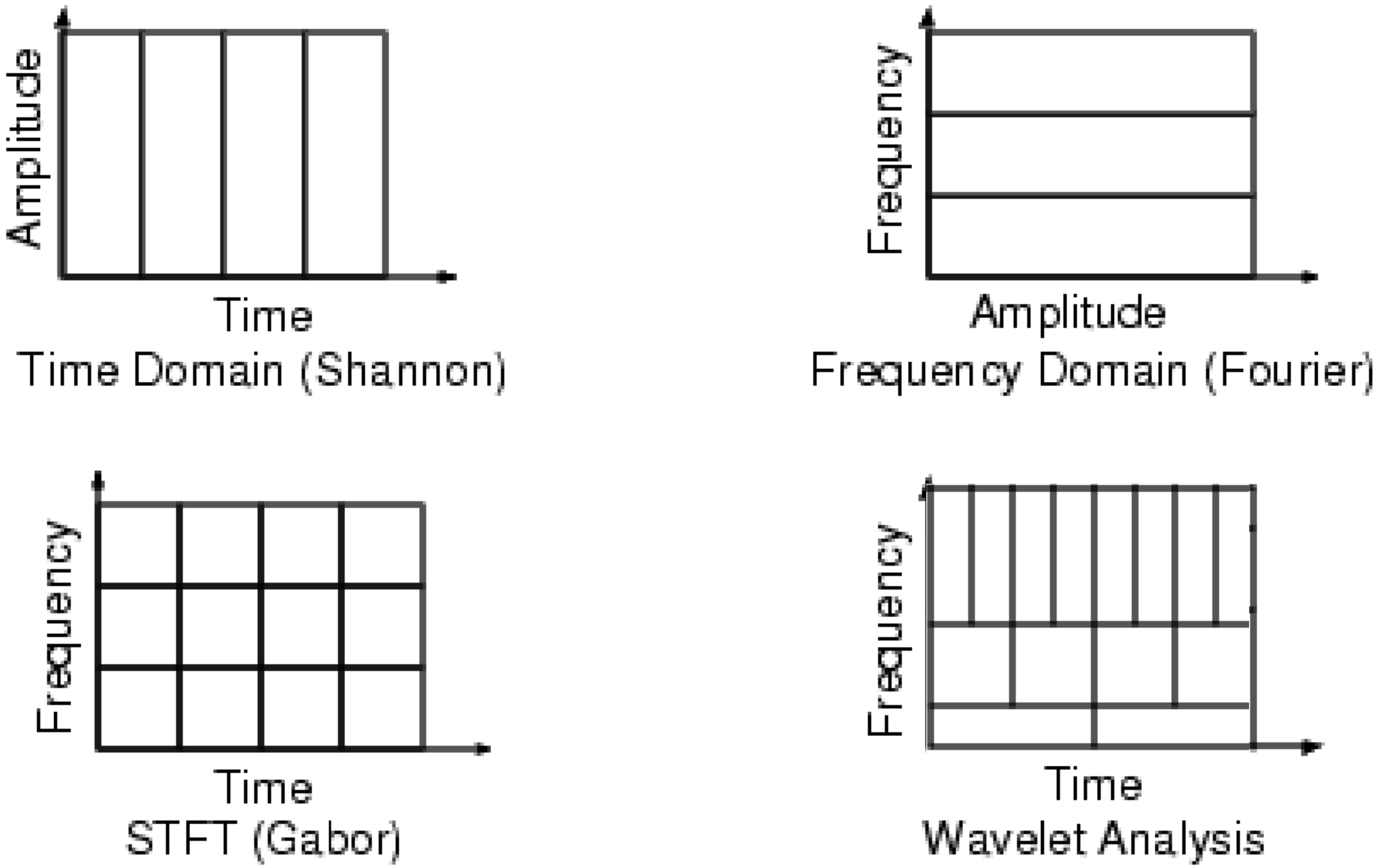
Discrete Time-Dependent FT

- Fixed window size, shift in time (Gabor)



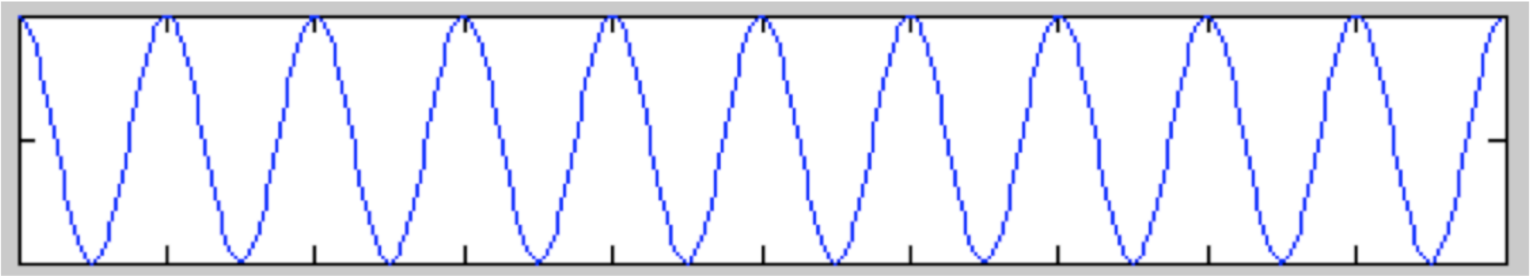
  - Use a big window for low frequency content that is not localized in time
  - Use a small window for high frequency content that is localized in time

Transform Comparison

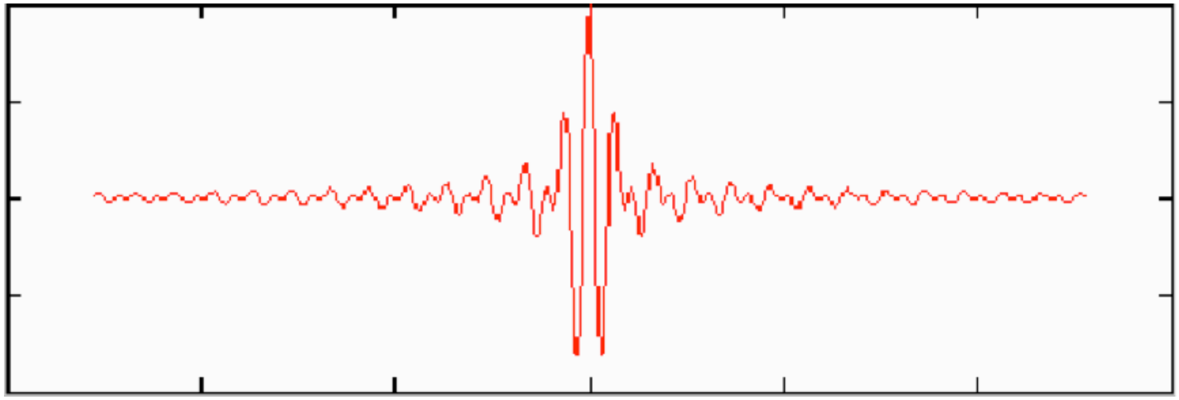


Fourier vs. Wavelet

- Fourier Analysis is based on an indefinitely long cosine wave of a specific frequency



- Wavelet Analysis is based on a short duration wavelet of a specific center frequency

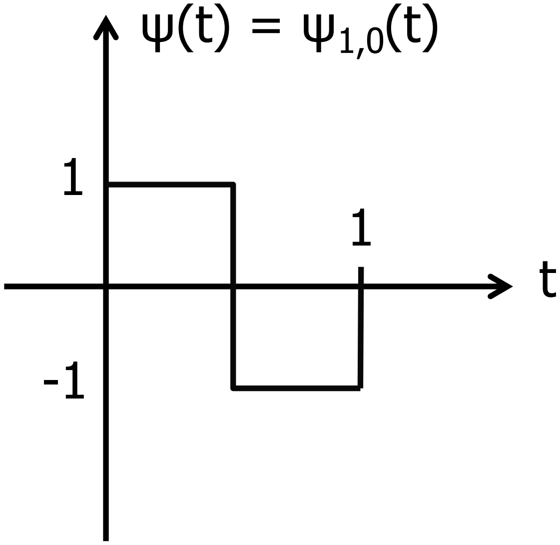


Wavelet Transform

- All wavelets derived from *mother* wavelet

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

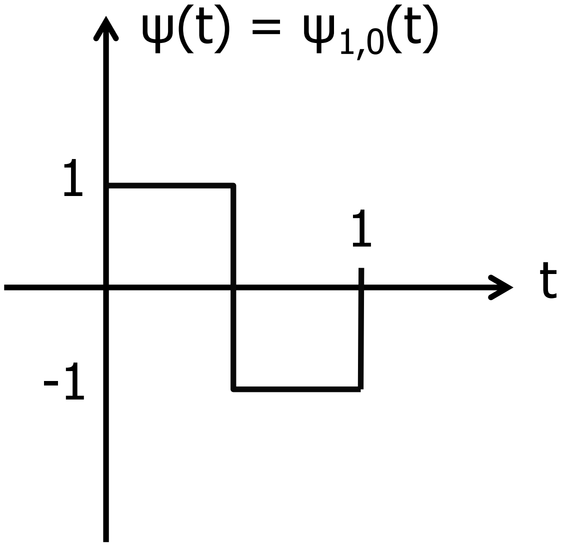
Example: Haar Wavelet



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

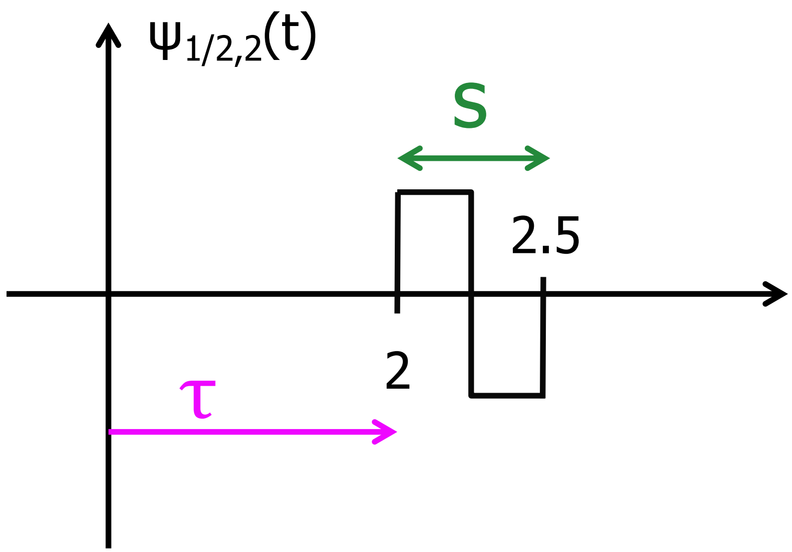
$s=1/2$, $\tau=2$

Example: Haar Wavelet

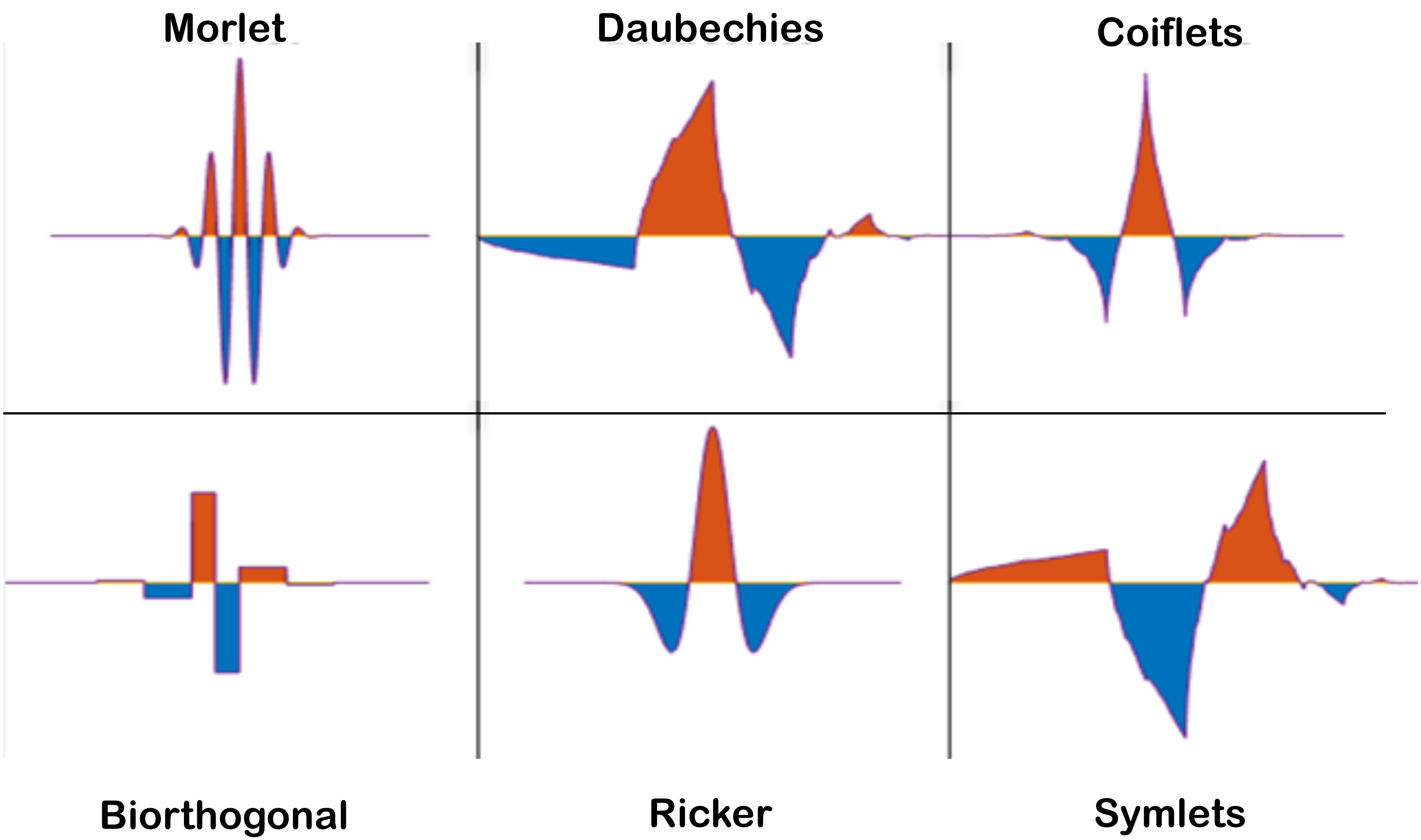


$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right)$$

s=1/2, τ=2



Examples of Wavelets



Ingrid Daubechies

- Defined worldwide standard for image compression
- https://www.fi.edu/en/laureates/ingrid-daubechies#:~:text=Her%20contributions%20have%20revolutionized%20and,the%20JPEG2000%20image%20processing%20standard.

Wavelet – Scaled and Shifted

normalization

shift in time

$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\,\psi\!\left(\frac{t-\tau}{s}\right)$$

change in scale

wavelet with scale, s and translation, τ

Mother wavelet

Continuous Wavelet Transform

time-series

$$\gamma(s,\tau) = \int f(t)\Psi_{s,\tau}(t)dt$$

coefficient of wavelet
with
scale, s and time, τ

wavelet with
scale, s, and shift, τ

Inverse Wavelet Transform

- Build up a time-series as sum of wavelets of different scales, s, and positions, t

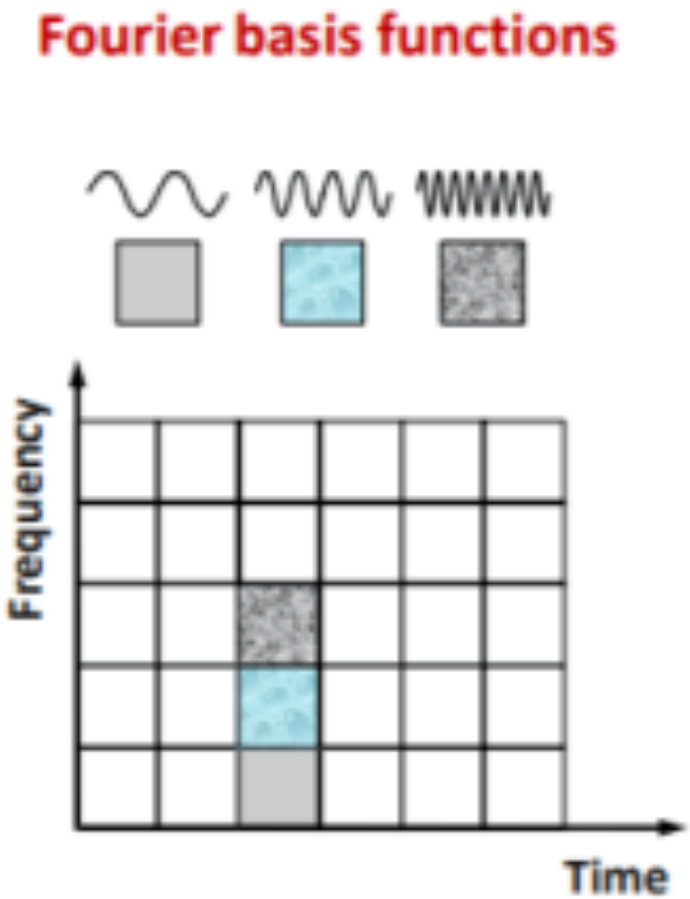
$$f(t) = \int\int \gamma(s,\tau)\psi_{s,\tau}(t)\,d\tau\,ds$$

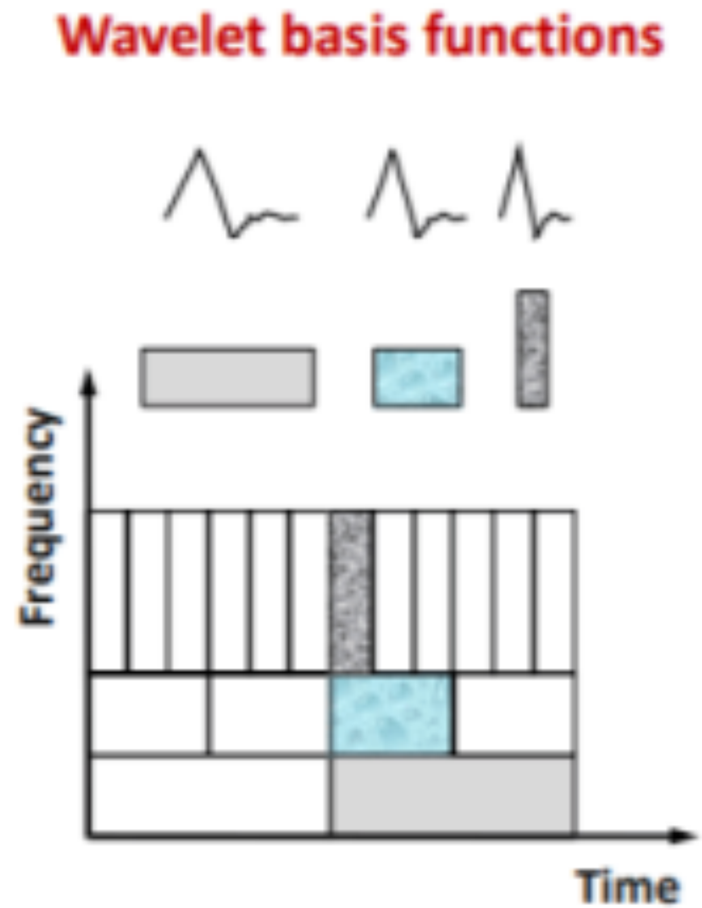
time-series

coefficients of wavelets

wavelet with scale, s and time, τ

Wavelet Basis Functions



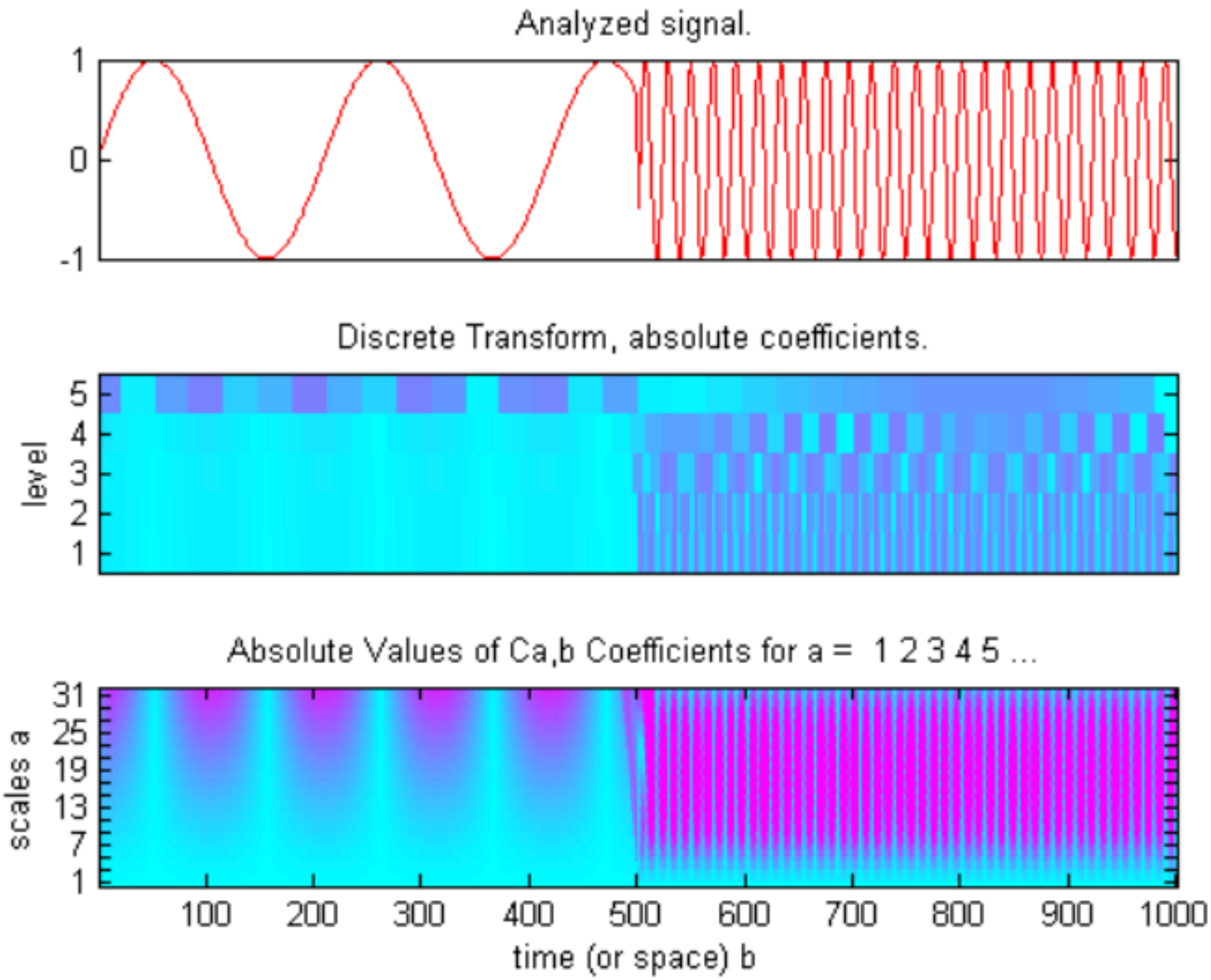


Discrete wavelets:

- Scale wavelets only by integer powers of 2
  - $s_j = 2^j$
- And shifting by integer multiples of s_j for each successive scale
  - $\tau_{j,k} = k2^j$
- Then $\gamma(s_j, \tau_{j,k}) = \gamma_{jk}$
  - where $j = 1, 2, \ldots\infty$, and $k = -\infty \ldots -2, -1, 0, 1, 2, \ldots \infty$

$$\gamma_{j,k} = \frac{1}{\sqrt{2^j}} \int f(t)\Psi\left(\frac{t-k2^j}{2^j}\right)dt$$

DWT vs CWT



Analyzed signal.
1
0
-1
Discrete Transform, absolute coefficients.
level
5
4
3
2
1
Absolute Values of Ca,b Coefficients for a = 1 2 3 4 5 ...
scales a
31
25
19
13
7
1
100 200 300 400 500 600 700 800 900 1000
time (or space) b

Wavelet Transform

- For a fixed scale, *s*, determining the wavelet coefficients can be thought of as a filtering operation

$$\gamma(s,\tau) = \int f(t)\Psi_{s,\tau}(t)dt$$

$$\gamma_s(\tau) = \int f(t)\Psi_s(t-\tau)dt = f(\tau) * \Psi_s(-\tau)$$

If wavelet is even,
$\Psi(-\tau) = \Psi(\tau)$

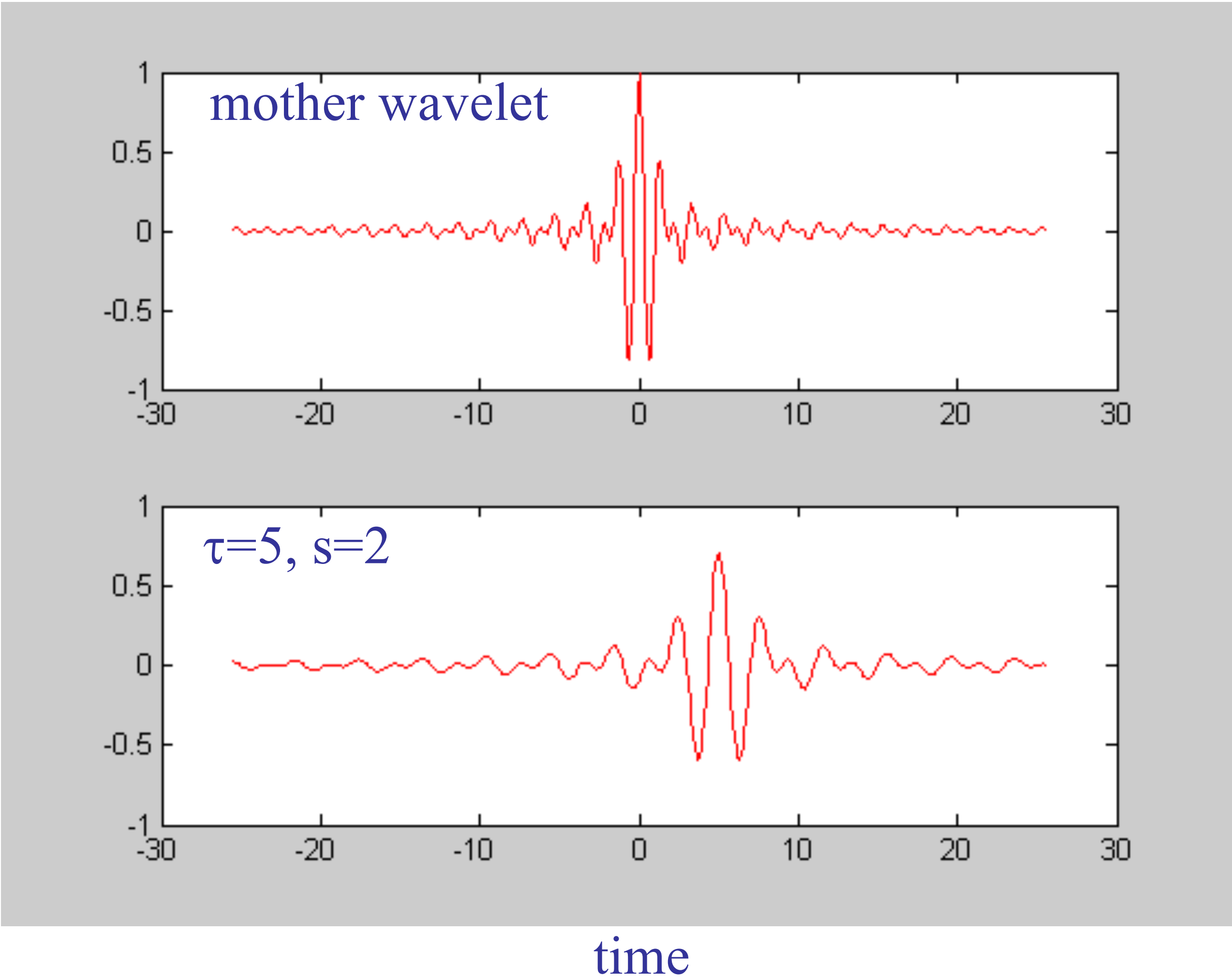
- where

$$\Psi_s(t) = \frac{1}{\sqrt{s}}\Psi(\frac{t}{s})$$

Shannon Wavelet

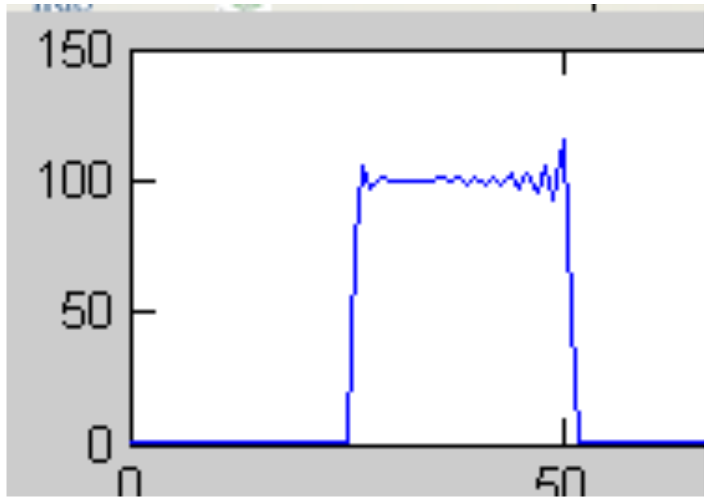
$$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right)$$

- $\Psi(t) = 2\,\mathrm{sinc}(2t) - \mathrm{sinc}(t)$



Fourier spectrum of Shannon Wavelet

$\Psi_s(j\Omega)$



frequency, Ω

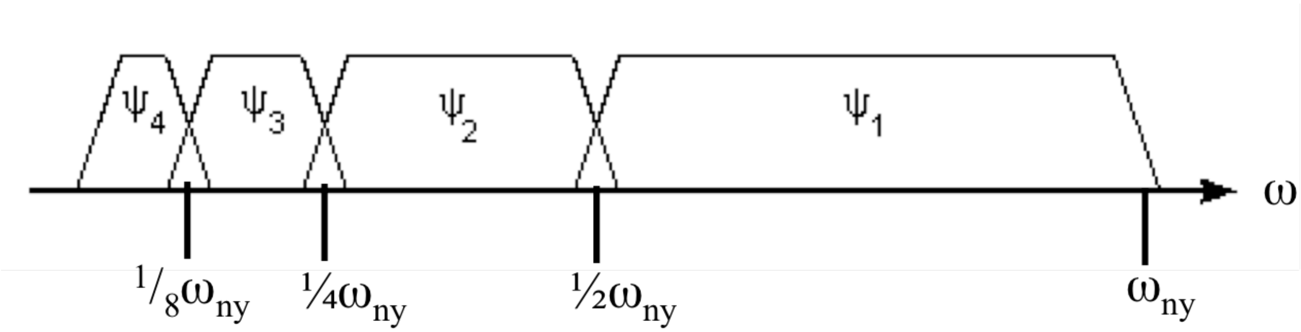
- Wavelet coefficients are a result of bandpass filtering

Discrete Wavelet Transform

- The coefficients of Ψ is just the band-pass filtered time-series, where Ψ is the wavelet, now viewed as the impulse response of a bandpass filter.
  - Discrete wavelet → $s = 2^j$

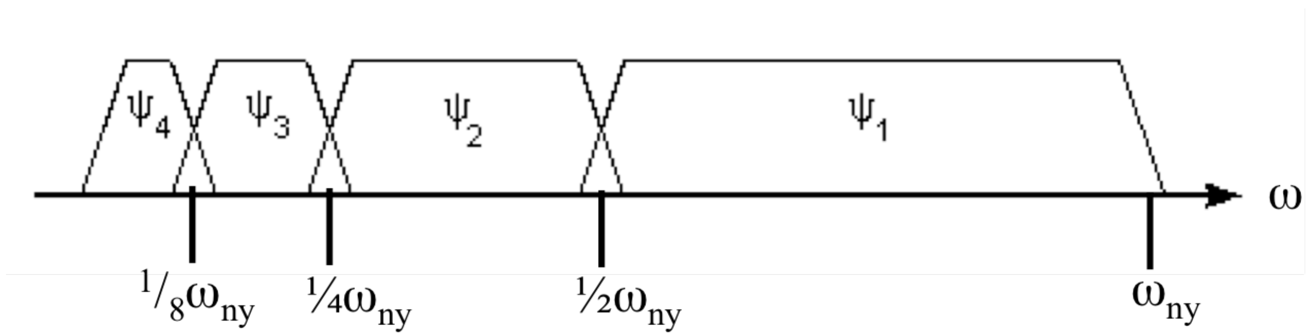
Discrete Wavelet Transform

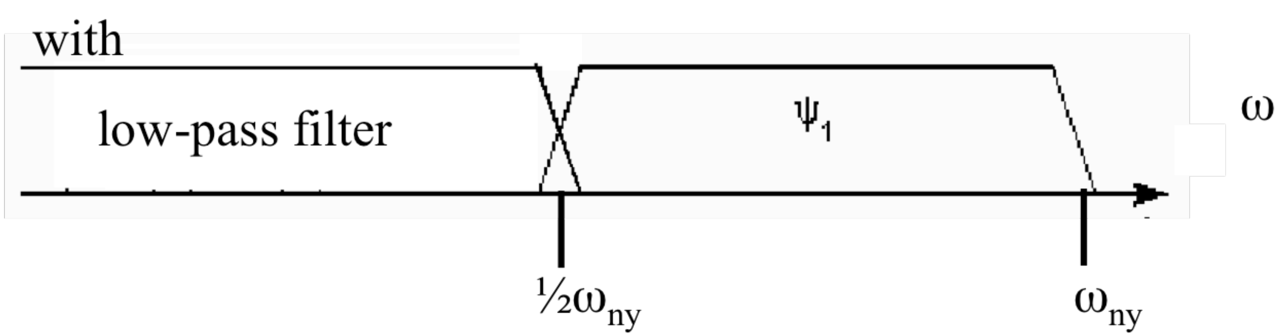
- The coefficients of Ψ is just the band-pass filtered time-series, where Ψ is the wavelet, now viewed as the impulse response of a bandpass filter.
  - Discrete wavelet → $s = 2^j$



Discrete Wavelet Transform

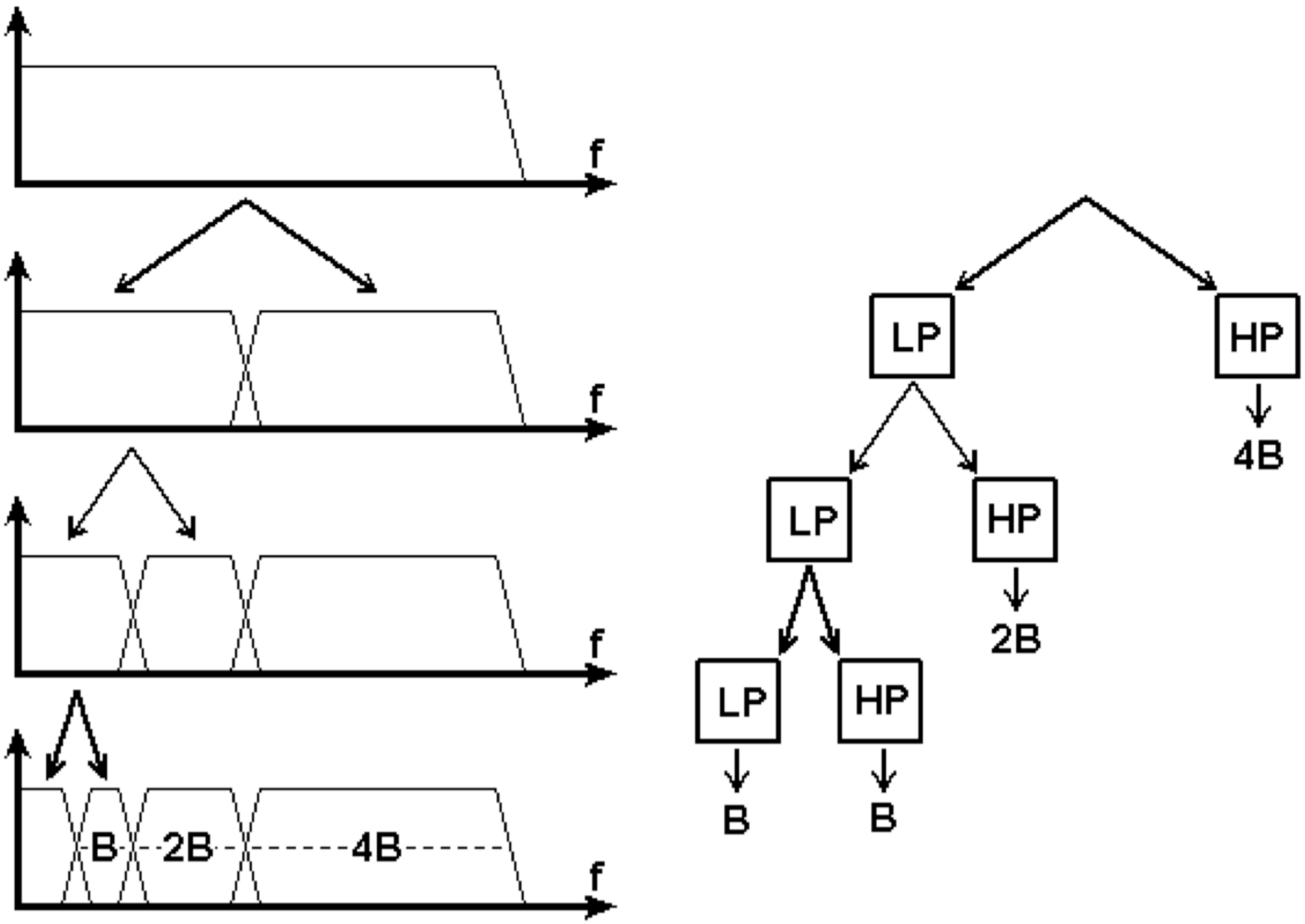
- The coefficients of Ψ is just the band-pass filtered time-series, where Ψ is the wavelet, now viewed as the impulse response of a bandpass filter.
  - Discrete wavelet → $s = 2^j$



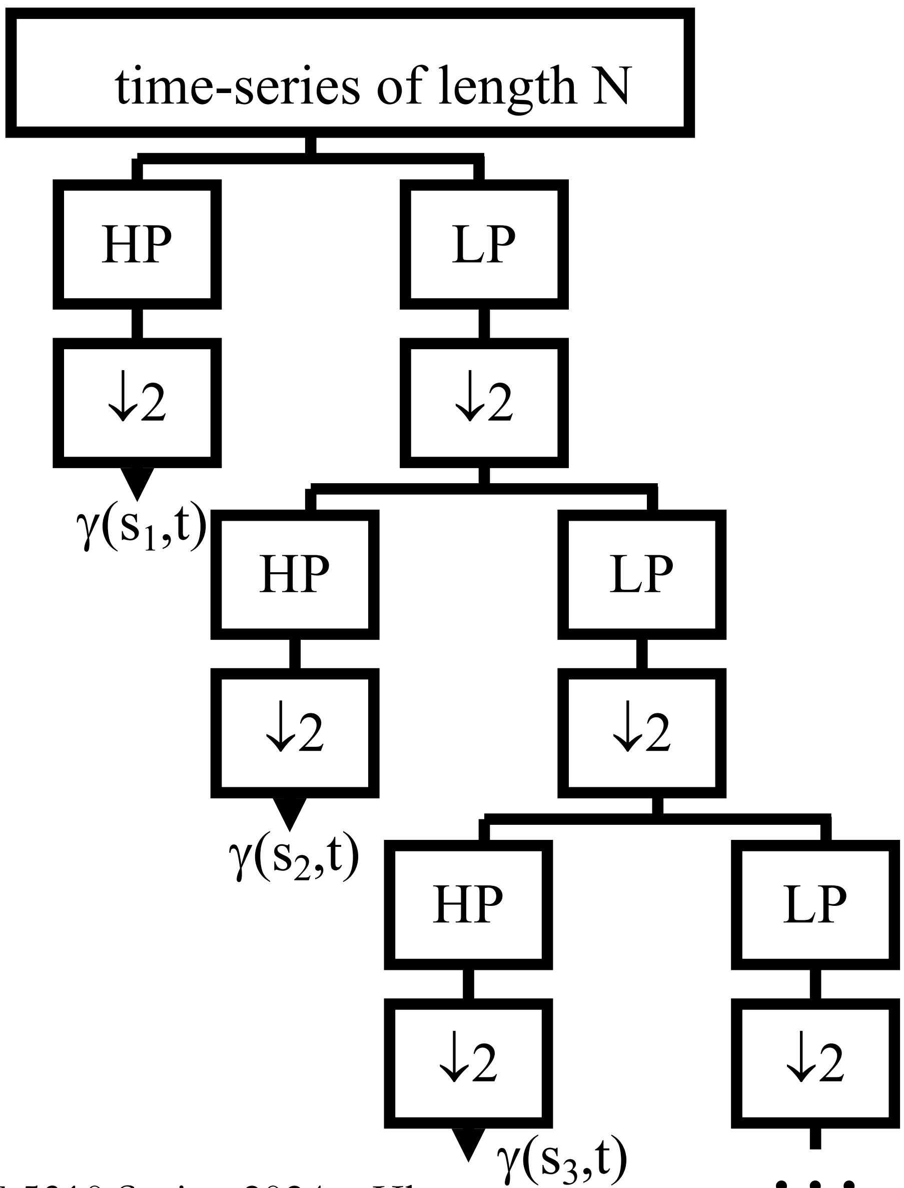


Digital Wavelet as Multirate Filter Bank

- Repeat recursively!



Digital Wavelet as Multirate Filter Bank



$\gamma(s_1,t)$: N/2 coefficients

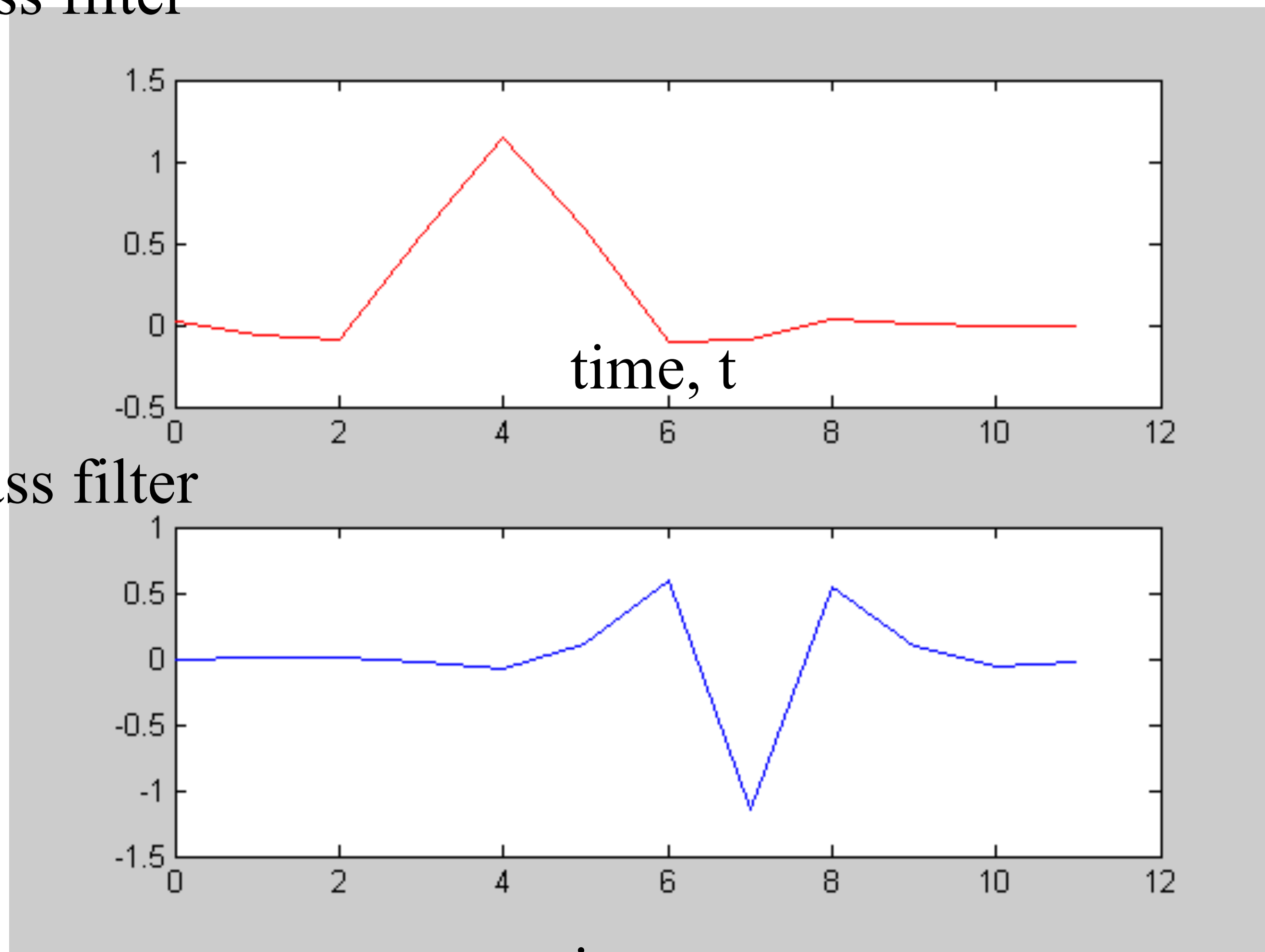
$\gamma(s_2,t)$: N/4 coefficients

$\gamma(s_2,t)$: N/8 coefficients

Total: N coefficients

Impulse Responses

Coiflet low pass filter



time, t

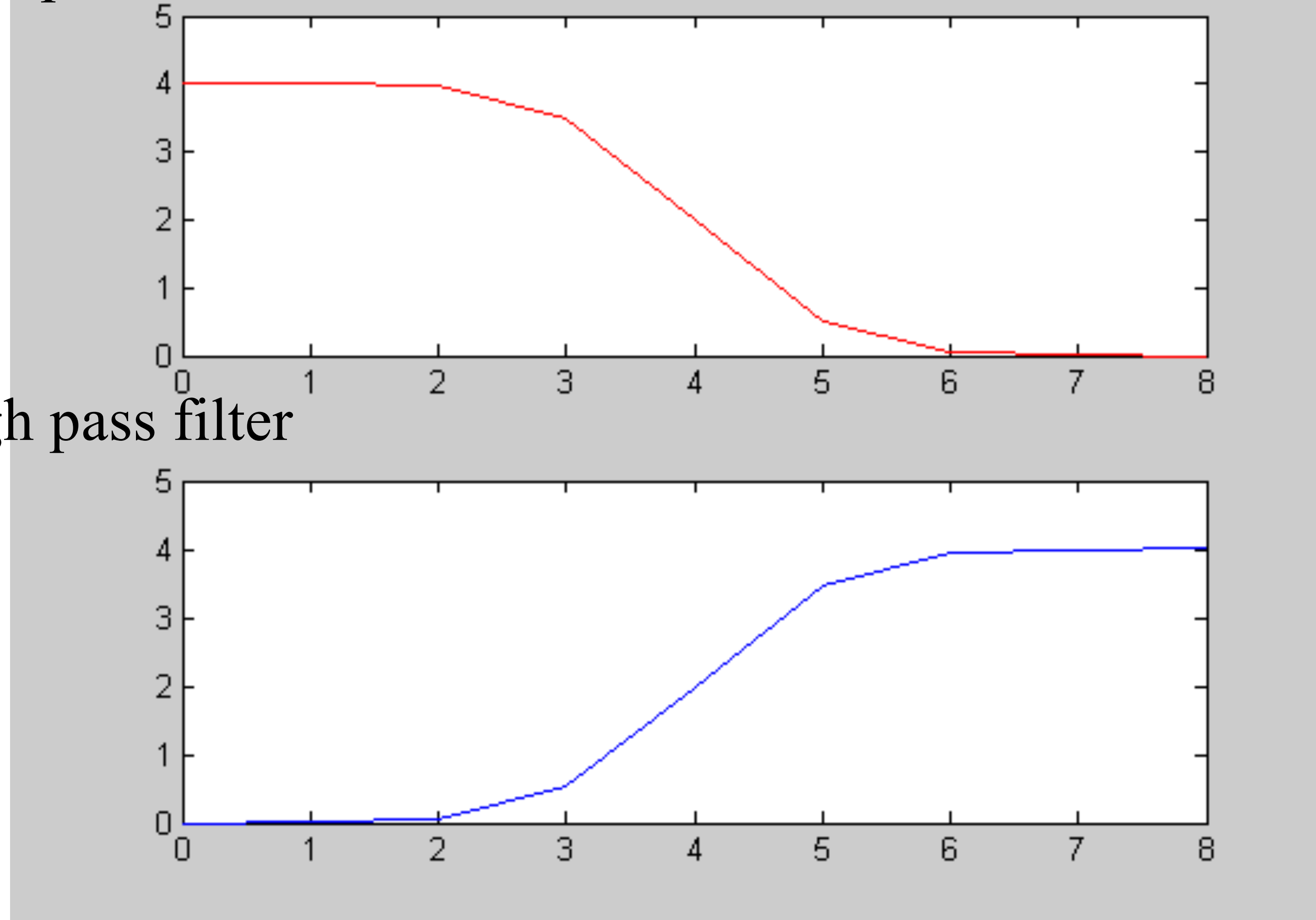
Coiflet high-pass filter

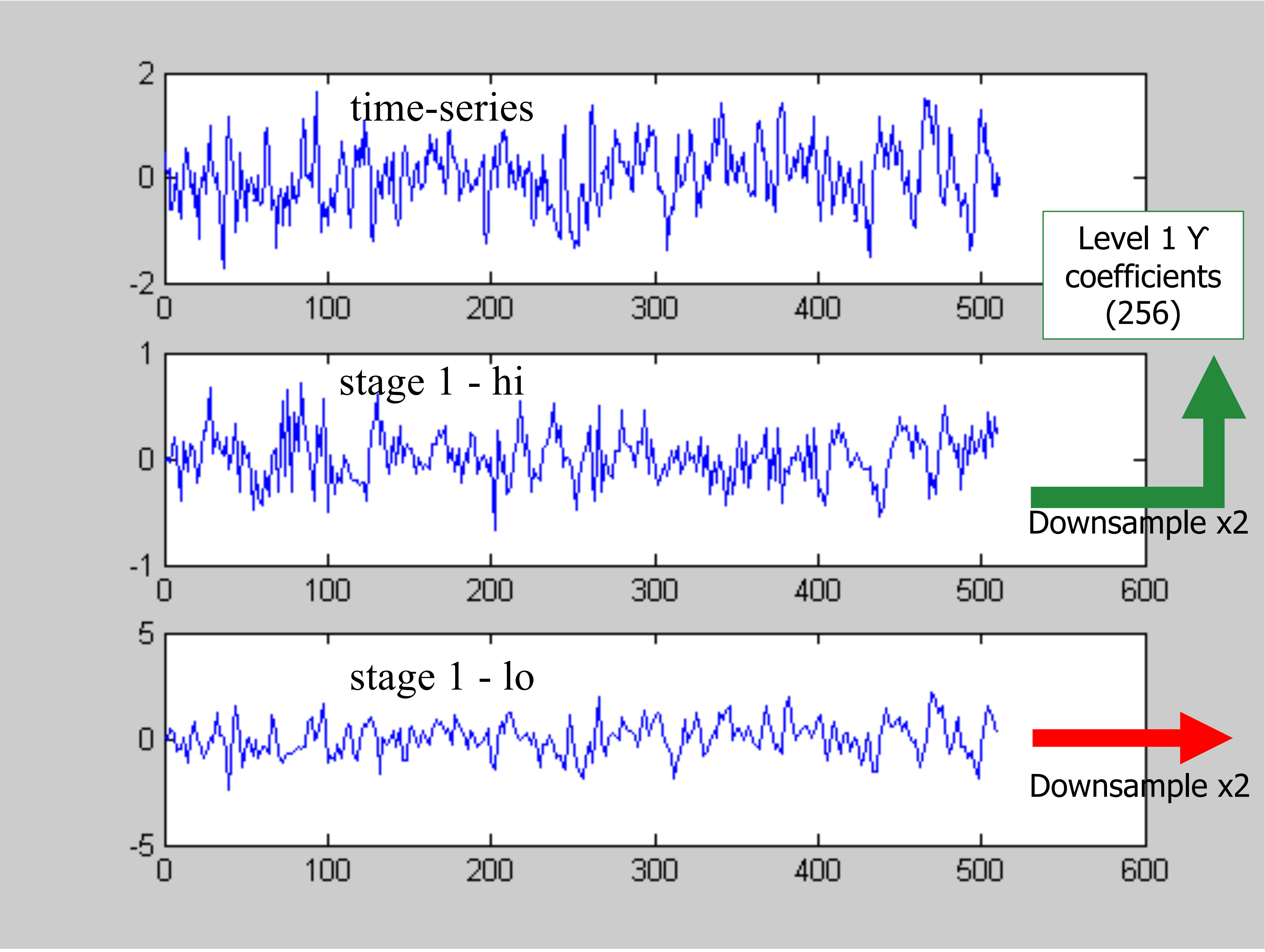
time, t

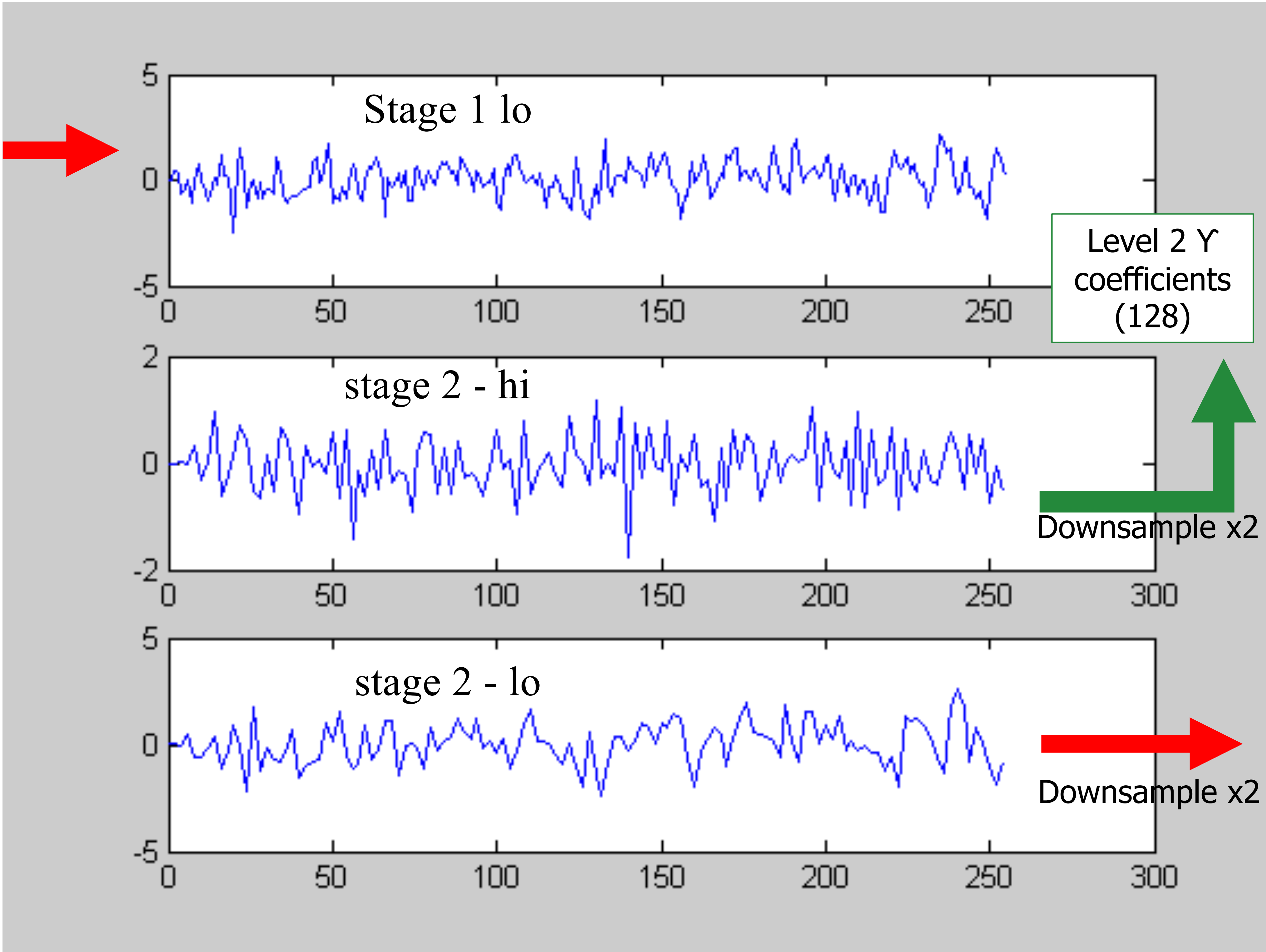
Filter Responses

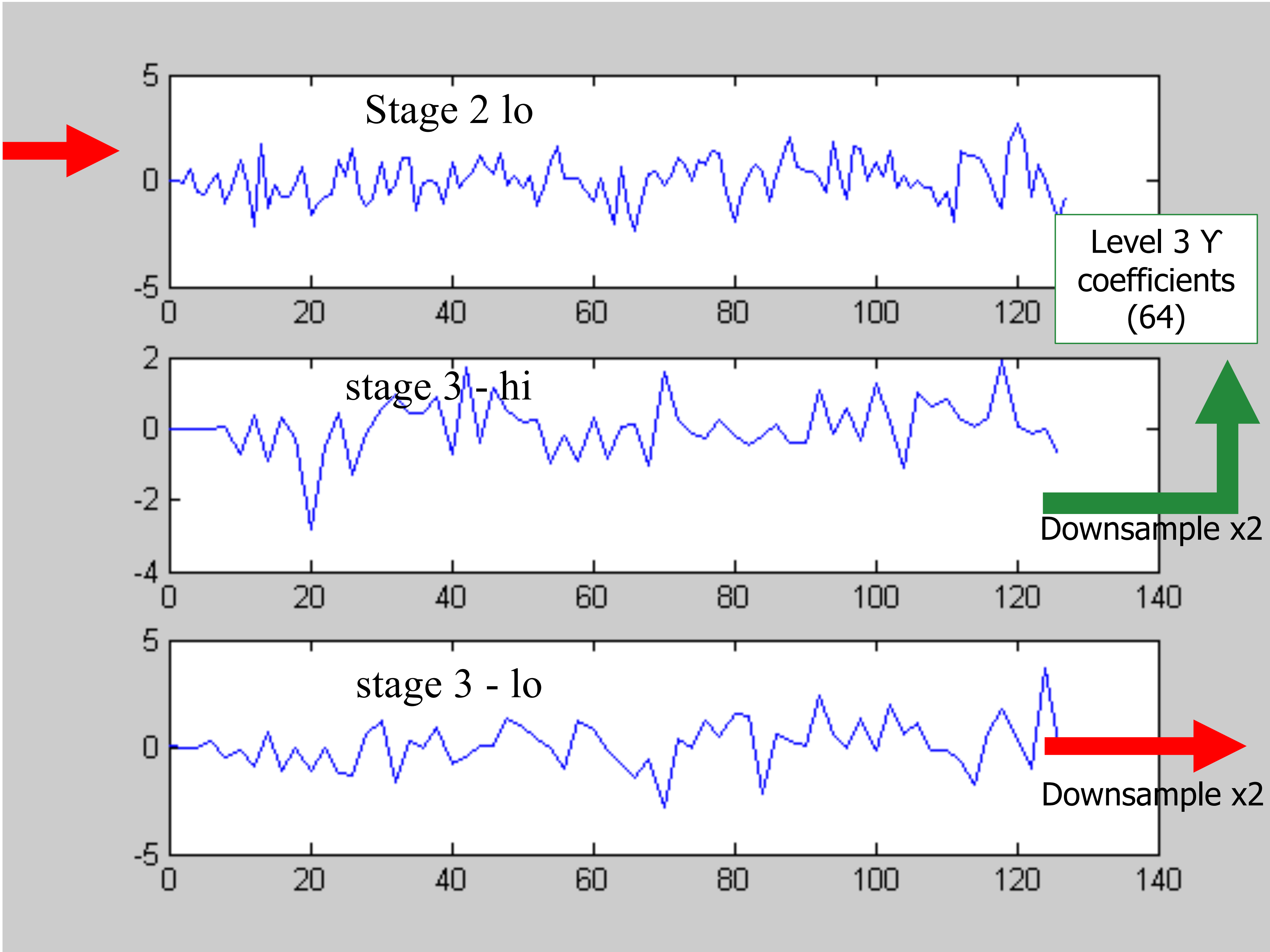
Spectrum of low pass filter

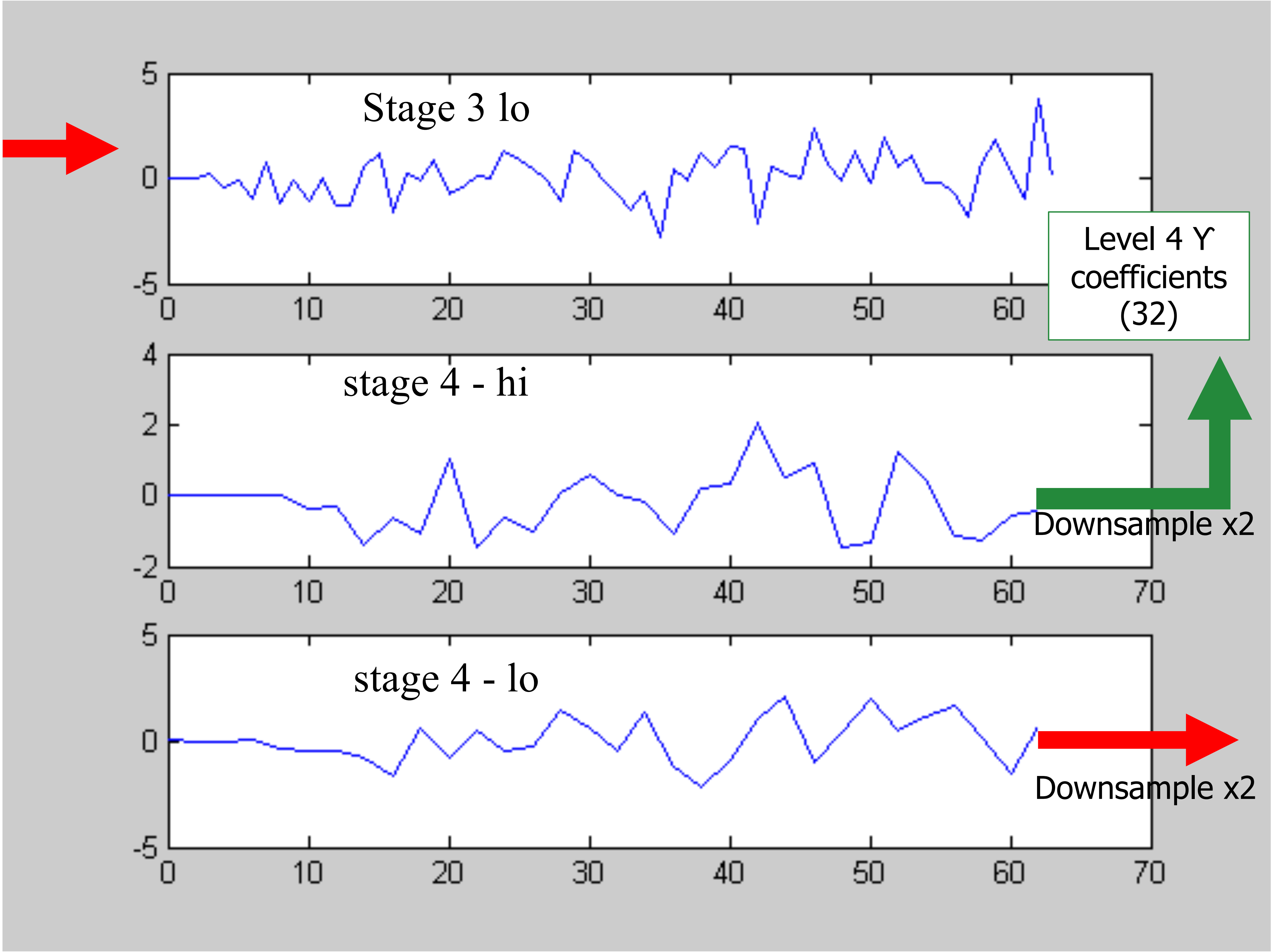
Spectrum of high pass filter

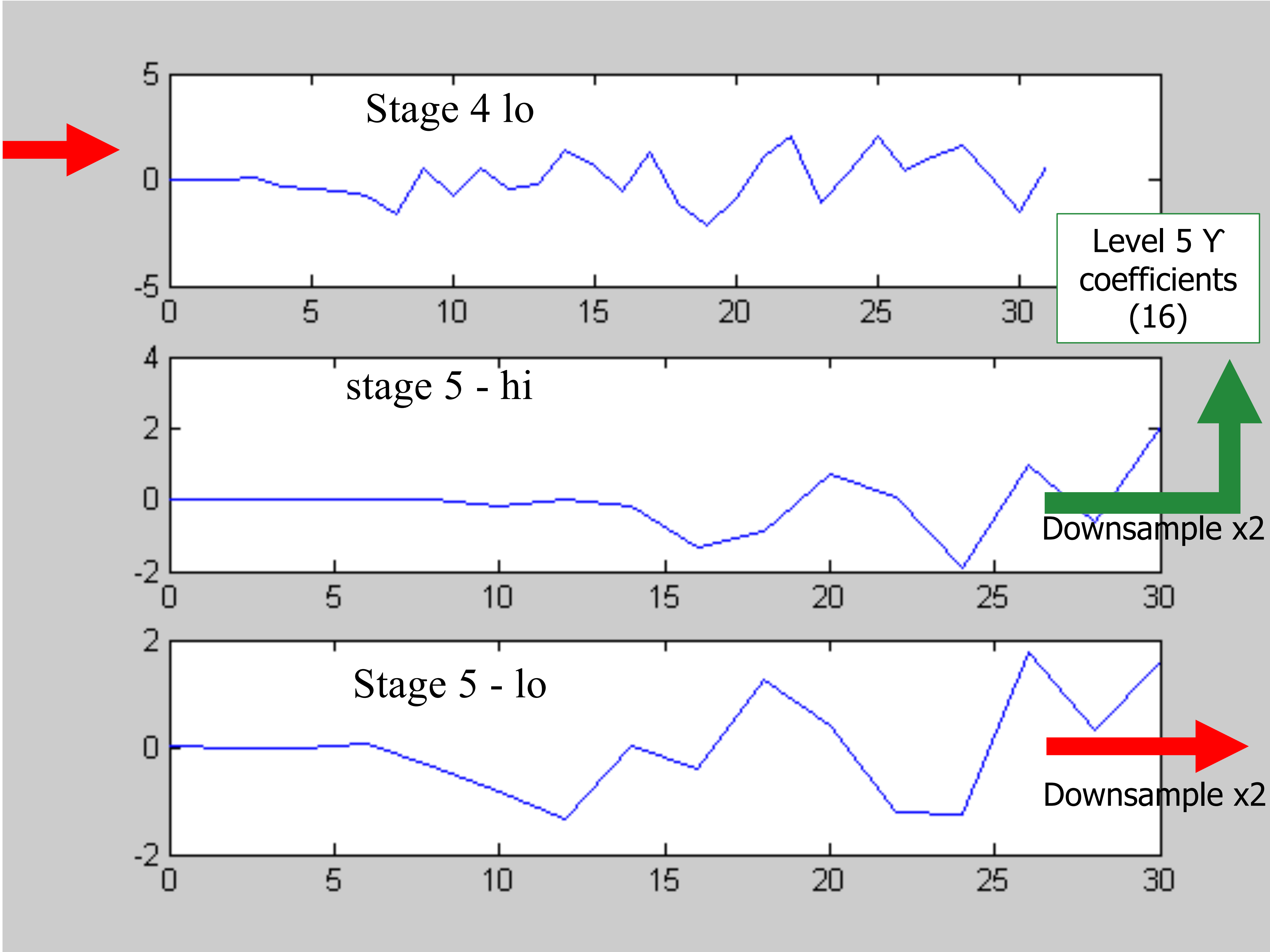



time-series
stage 1 - hi
stage 1 - lo
Level 1 Y coefficients (256)
Downsample x2
Downsample x2
2
0
-2
1
0
-1
5
0
-5
0
100
200
300
400
500
600

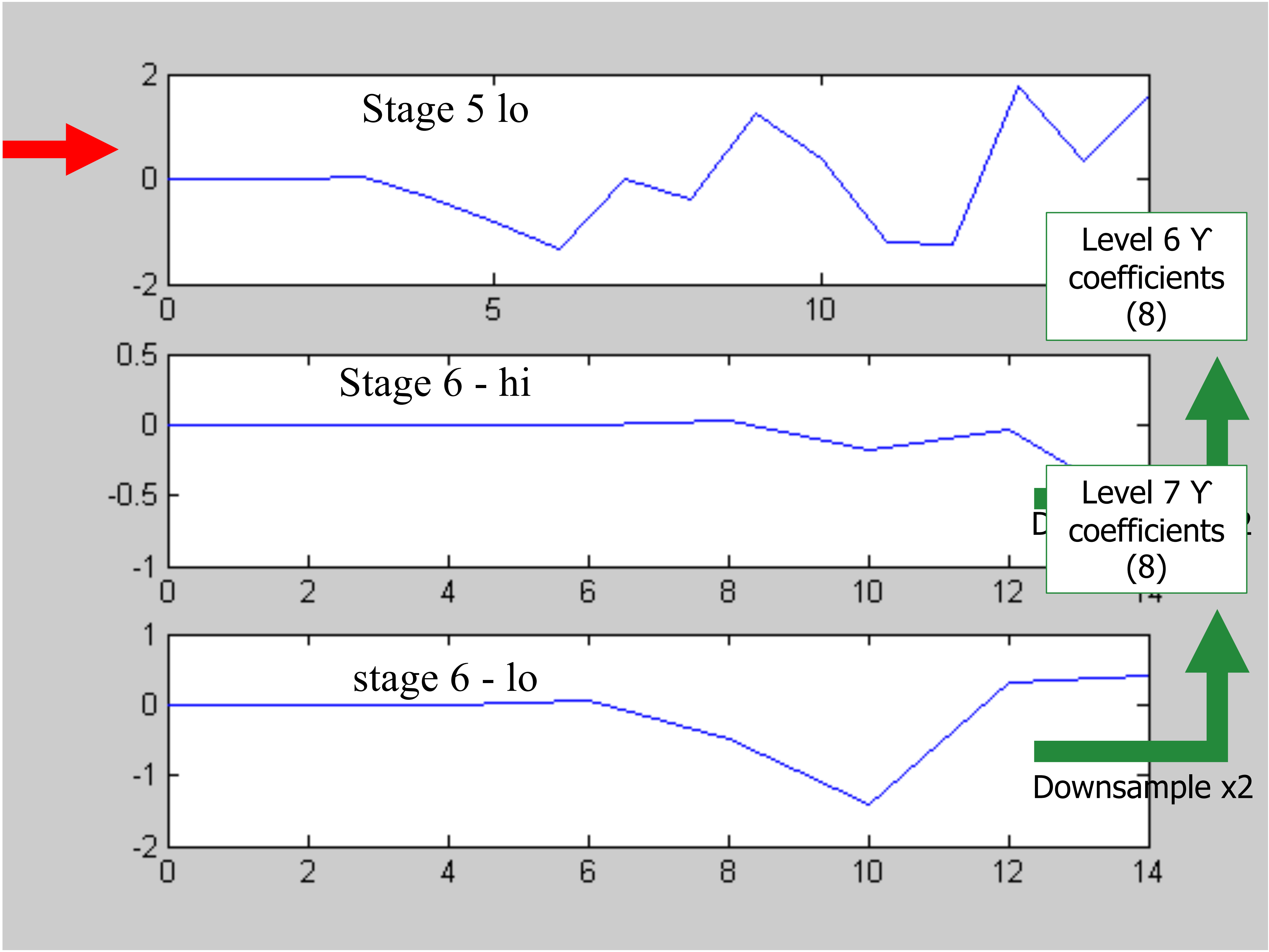

Stage 1 lo
stage 2 - hi
stage 2 - lo
Level 2 Y coefficients (128)
Downsample x2
Downsample x2
5
0
-5
0
50
100
150
200
250
2
-2
300


Stage 2 lo
Level 3 Y`
coefficients
(64)
stage 3 - hi
Downsample x2
stage 3 - lo
Downsample x2

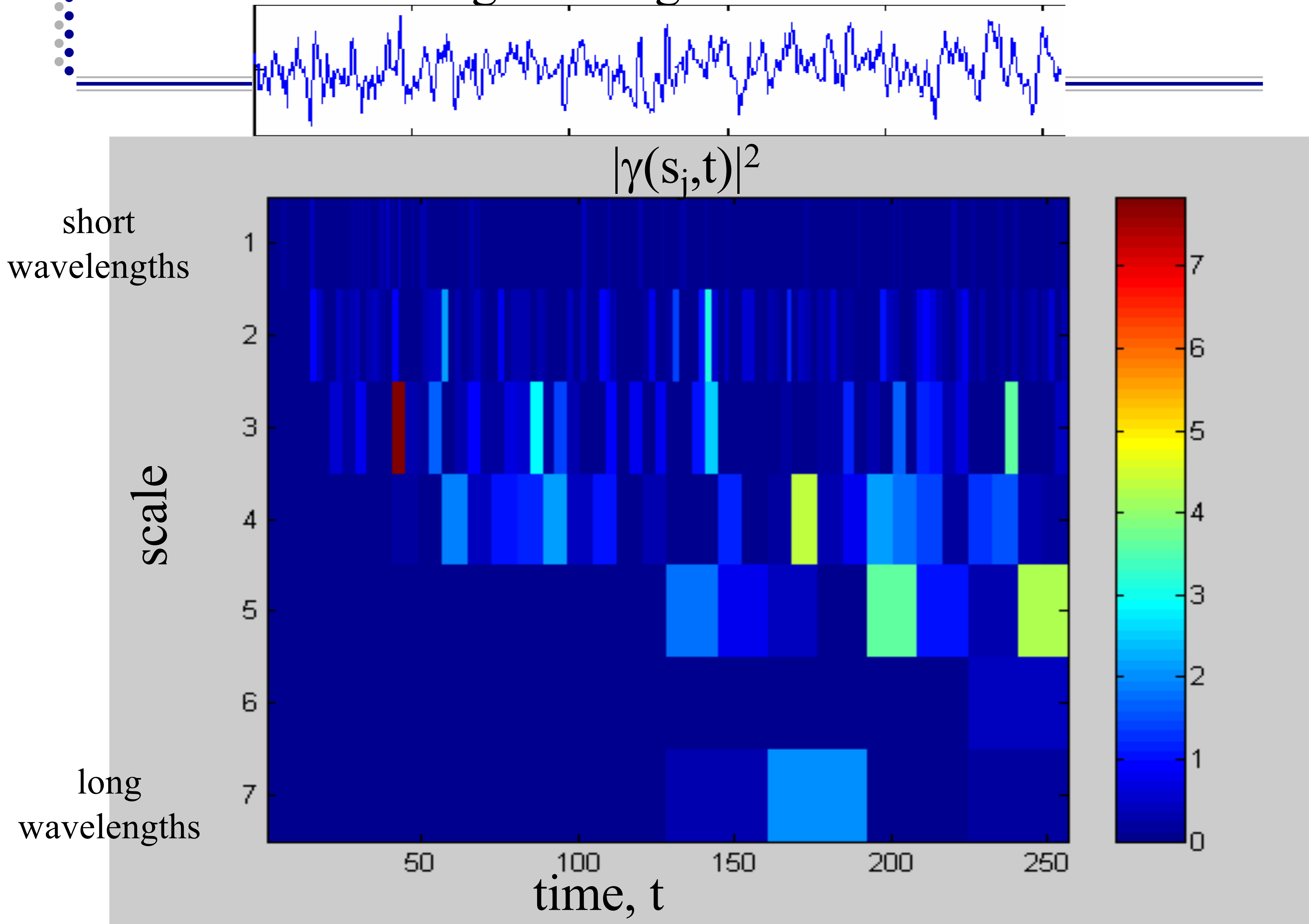

Stage 3 lo
Level 4 Y
coefficients
(32)
stage 4 - hi
Downsample x2
stage 4 - lo
Downsample x2



Stage 4 lo
stage 5 - hi
Stage 5 - lo
Level 5 Y coefficients (16)
Downsample x2
Downsample x2


Stage 5 lo
Stage 6 - hi
stage 6 - lo
Level 6 Y coefficients (8)
Level 7 Y coefficients (8)
Downsample x2

Putting it all together …



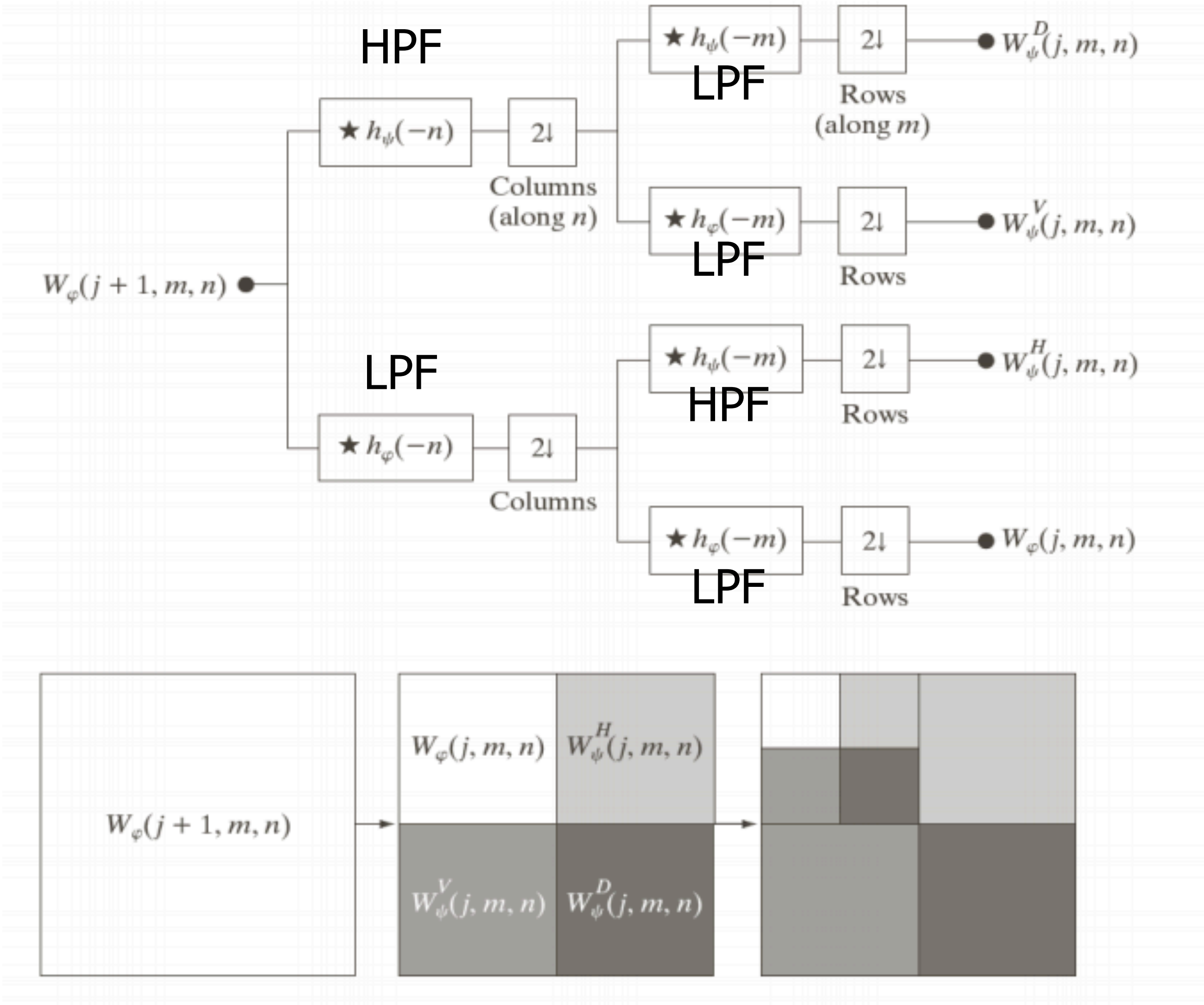
Expanding to Two Dimensions

- In two dimensions, a 2D scaling function $\phi(x, y)$ and three 2D wavelet functions $\psi^H(x, y), \psi^V(x, y), \psi^D(x, y)$ are required
- We can create these from the 1D scaling and wavelet functions:
  - $\phi(x, y) = \phi(x)\phi(y)$
  - $\psi^V(x, y) = \psi(x)\phi(y)$
  - $\psi^H(x, y) = \phi(x)\psi(y)$
  - $\psi^D(x, y) = \psi(x)\psi(y)$

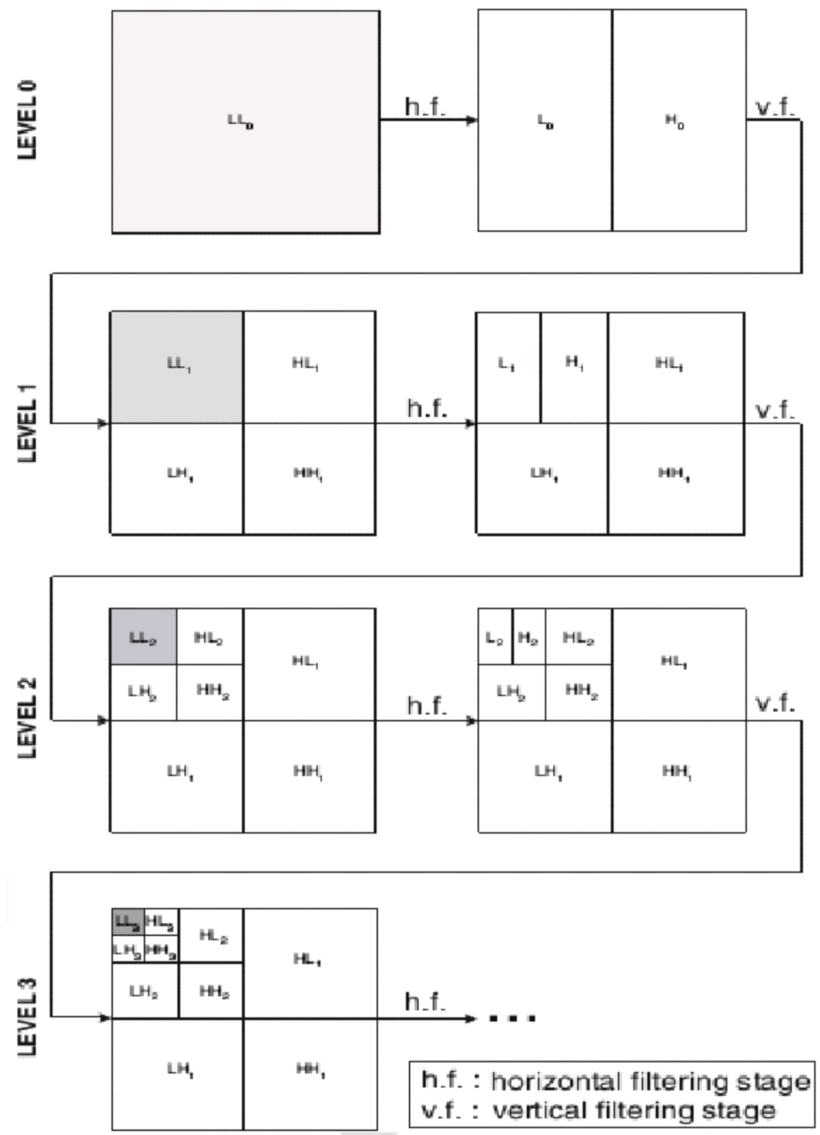
$\psi(x)$ - *HPF*

$\phi(y)$ - *LPF*

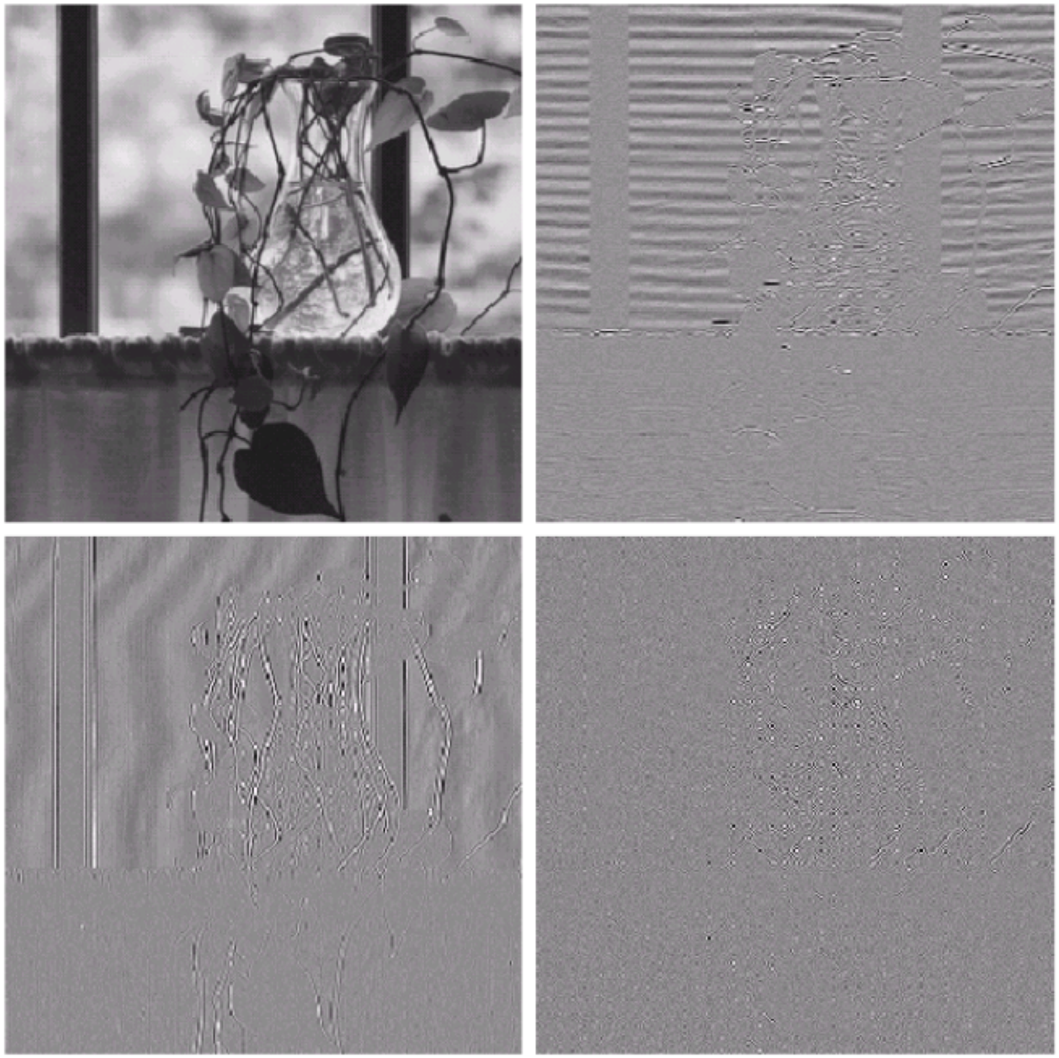
Expanding to Two Dimensions



Expanding to Two Dimensions



Expanding to Two Dimensions



Colorado School of Mines   Image and Multidimensional Signal Processing

FIGURE 7.7 A four-band split of the vase in Fig. 7.1 using the subband coding system of Fig. 7.5.

a	d^V
d^H	d^D

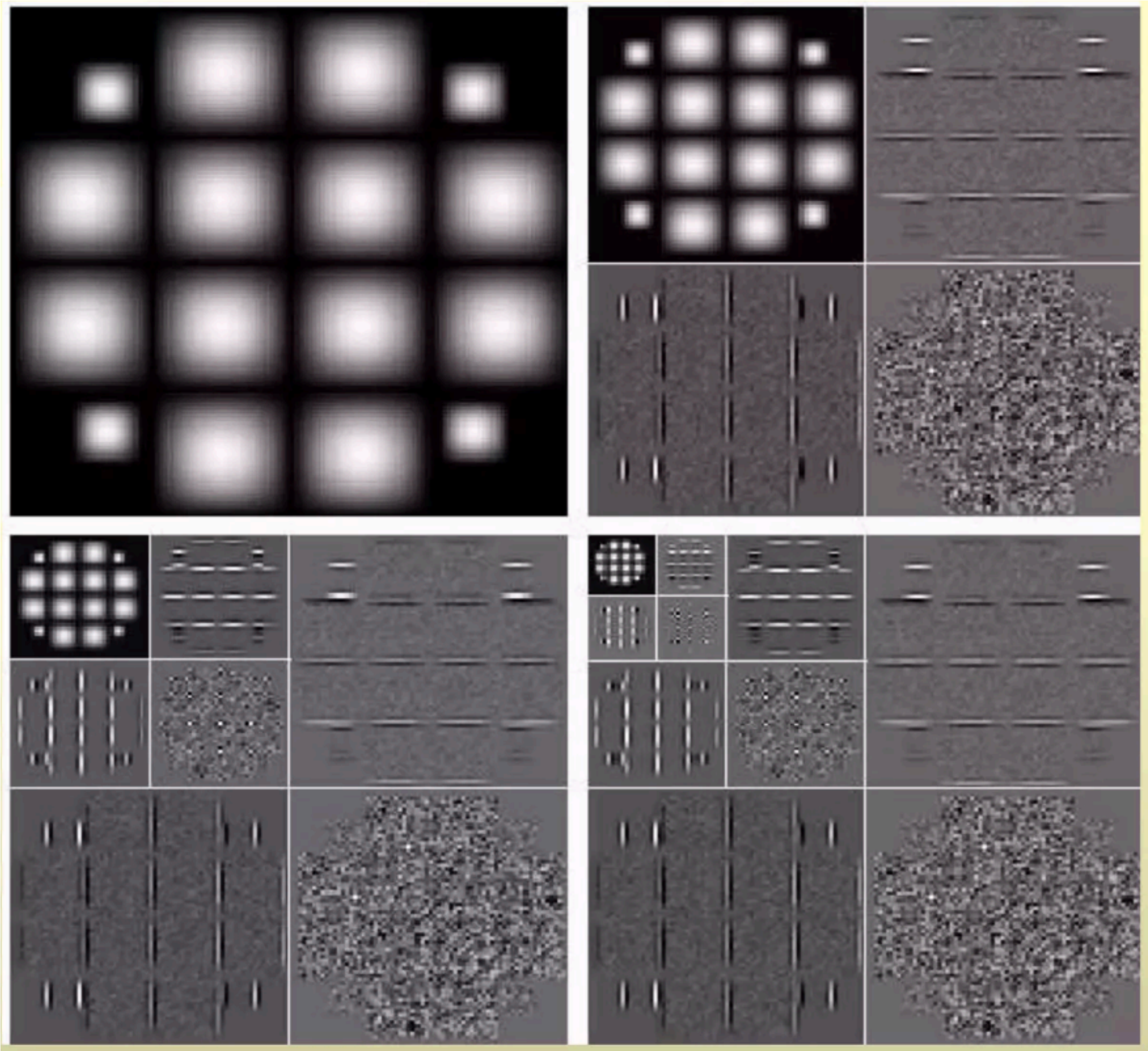
a(m,n): approximation

d^V(m,n): detail in vertical

d^H(m,n): detail in horizontal

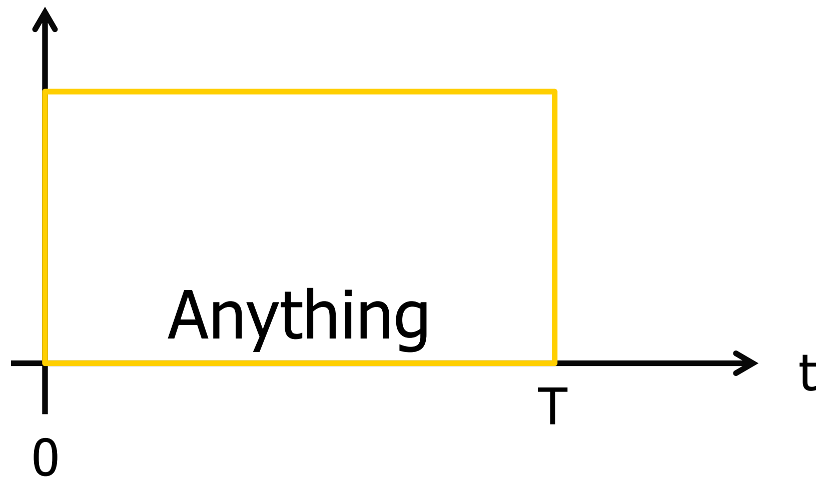
d^D(m,n): detail in diagonal

Expanding to Two Dimensions



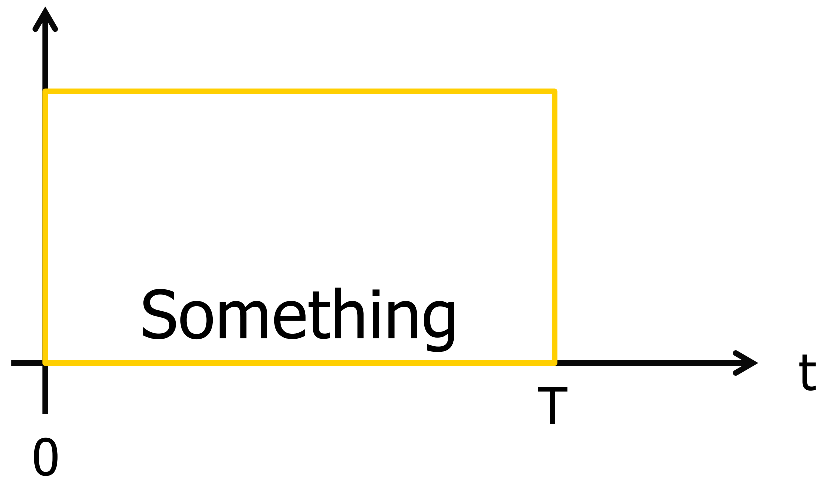
Compressive Sampling

Compressive Sampling



- What is the rate you need to sample at?
  - At least Nyquist

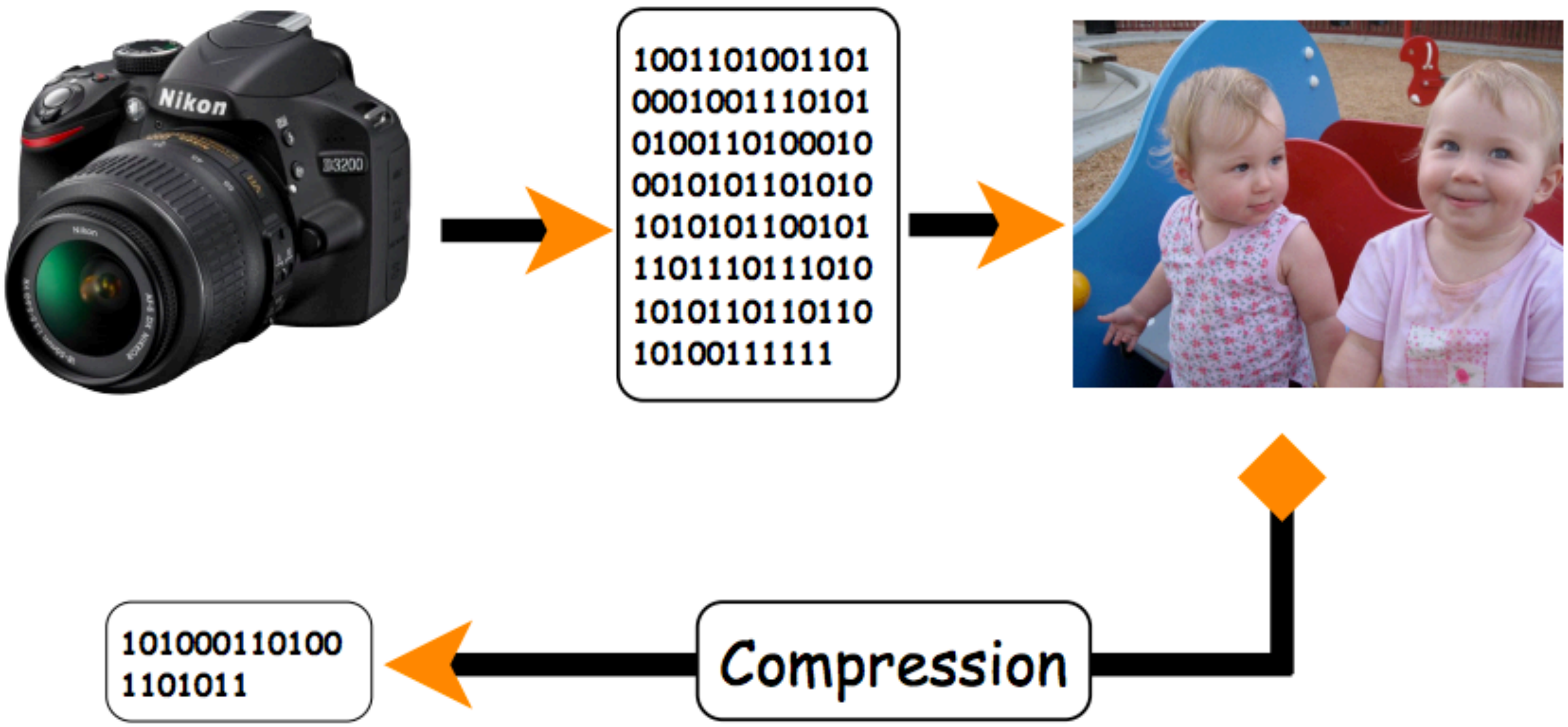
Compressive Sampling



- What is the rate you need to sample at?
  - Maybe less than Nyquist…

First: Compression

- Standard approach
  - First collect, then compress
    - Throw away unnecessary data



First: Compression

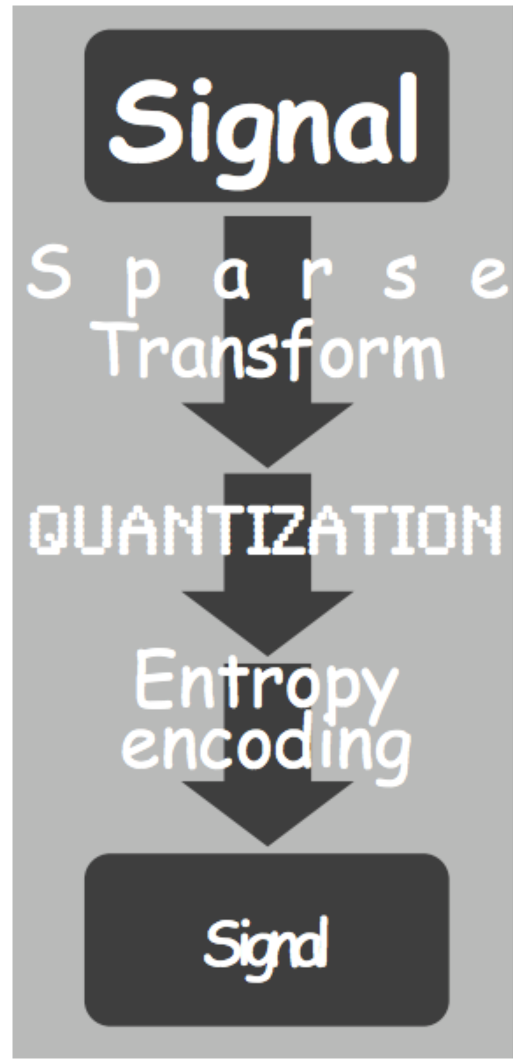
- Examples
  - Audio – 10x
    - Raw audio: 44.1kHz, 16bit, stereo = 1378 Kbit/sec
    - MP3: 44.1kHz, 16 bit, stereo = 128 Kbit/sec
  - Images – 22x
    - Raw image (RGB): 24bit/pixel
    - JPEG: 1280x960, normal = 1.09bit/pixel
  - Videos – 75x
    - Raw Video: (480x360)p/frame x 24b/p x 24frames/s + 44.1kHz x 16b x 2 = 98,578 Kbit/s
    - MPEG4: 1300 Kbit/s

First: Compression

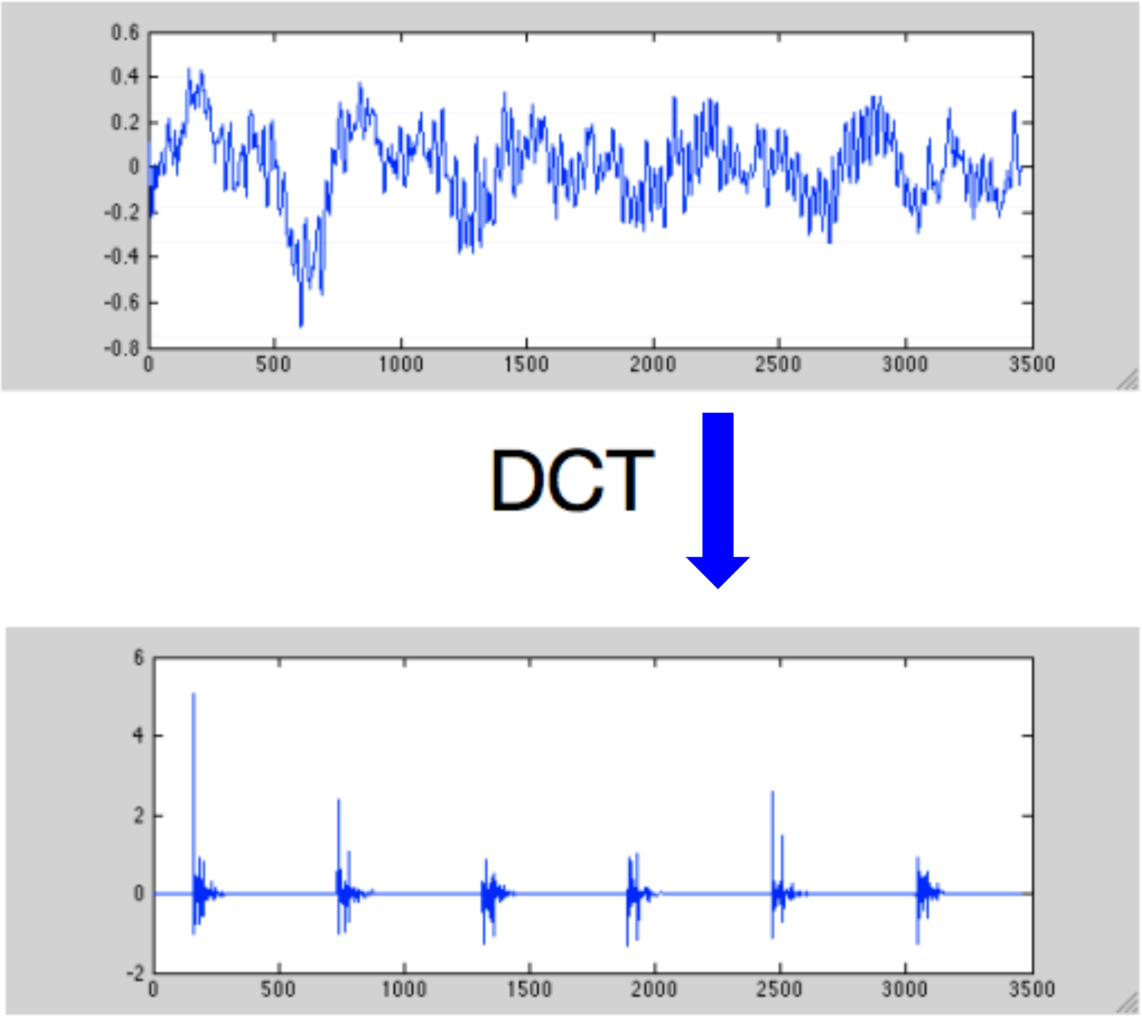
- Almost all compression algorithm use transform coding
  - mp3: DCT
  - JPEG: DCT
  - JPEG2000: Wavelet
  - MPEG: DCT & time-difference

First: Compression

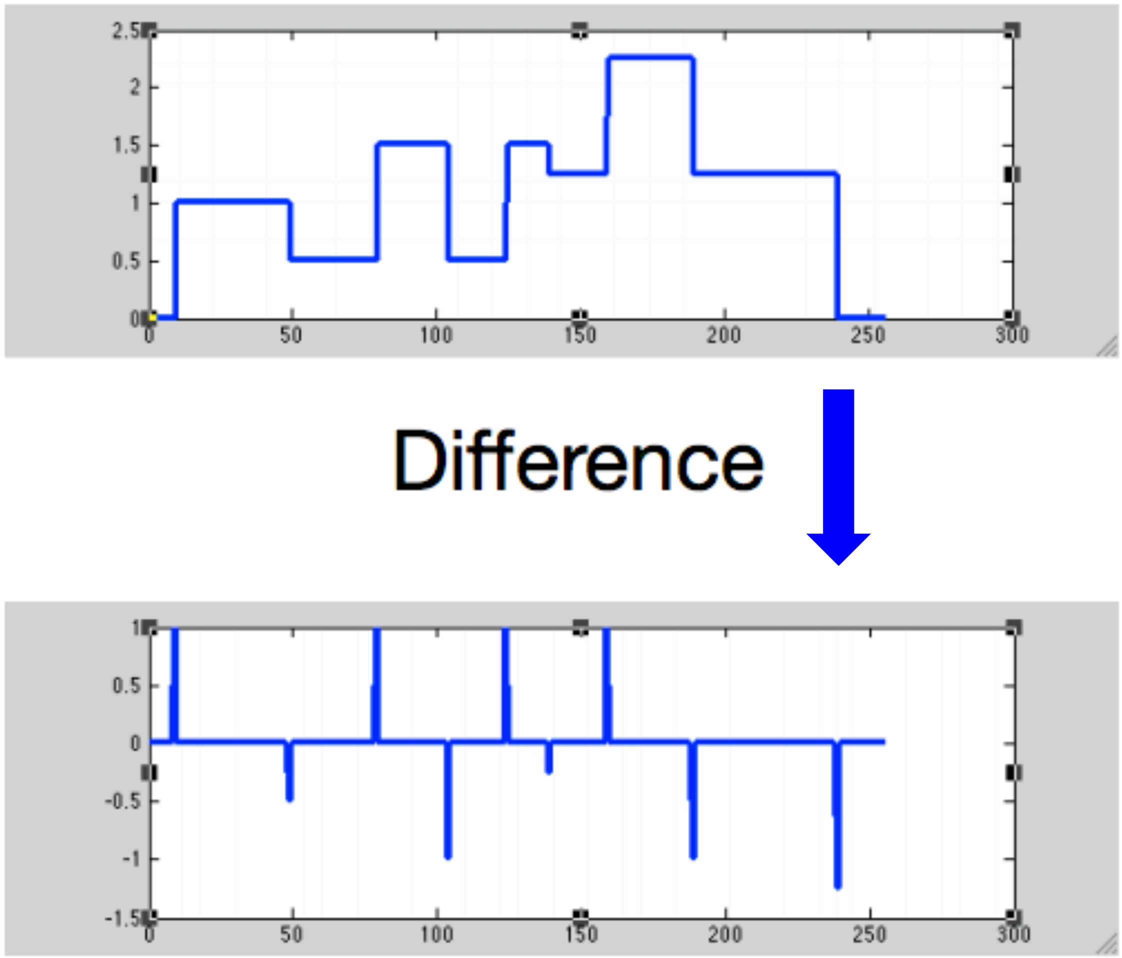
- Almost all compression algorithm use transform coding
  - mp3: DCT
  - JPEG: DCT
  - JPEG2000: Wavelet
  - MPEG: DCT & time-difference



Sparse Transform



Sparse Transform



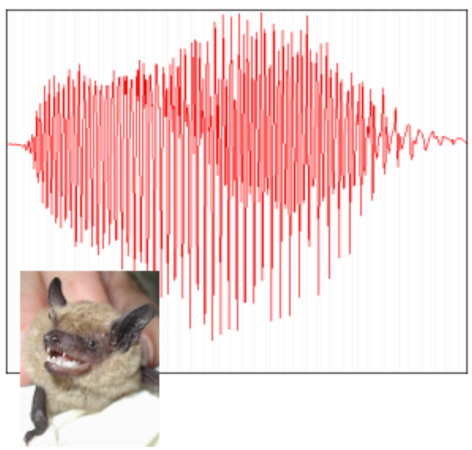
Sparsity

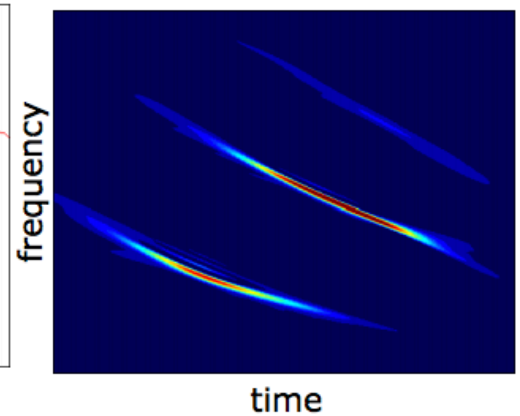
N
pixels

$K \ll N$
large
wavelet
coefficients

(blue = 0)

N
wideband
signal
samples





$K \ll N$
large
Gabor (TF)
coefficients

Signal Processing Trends

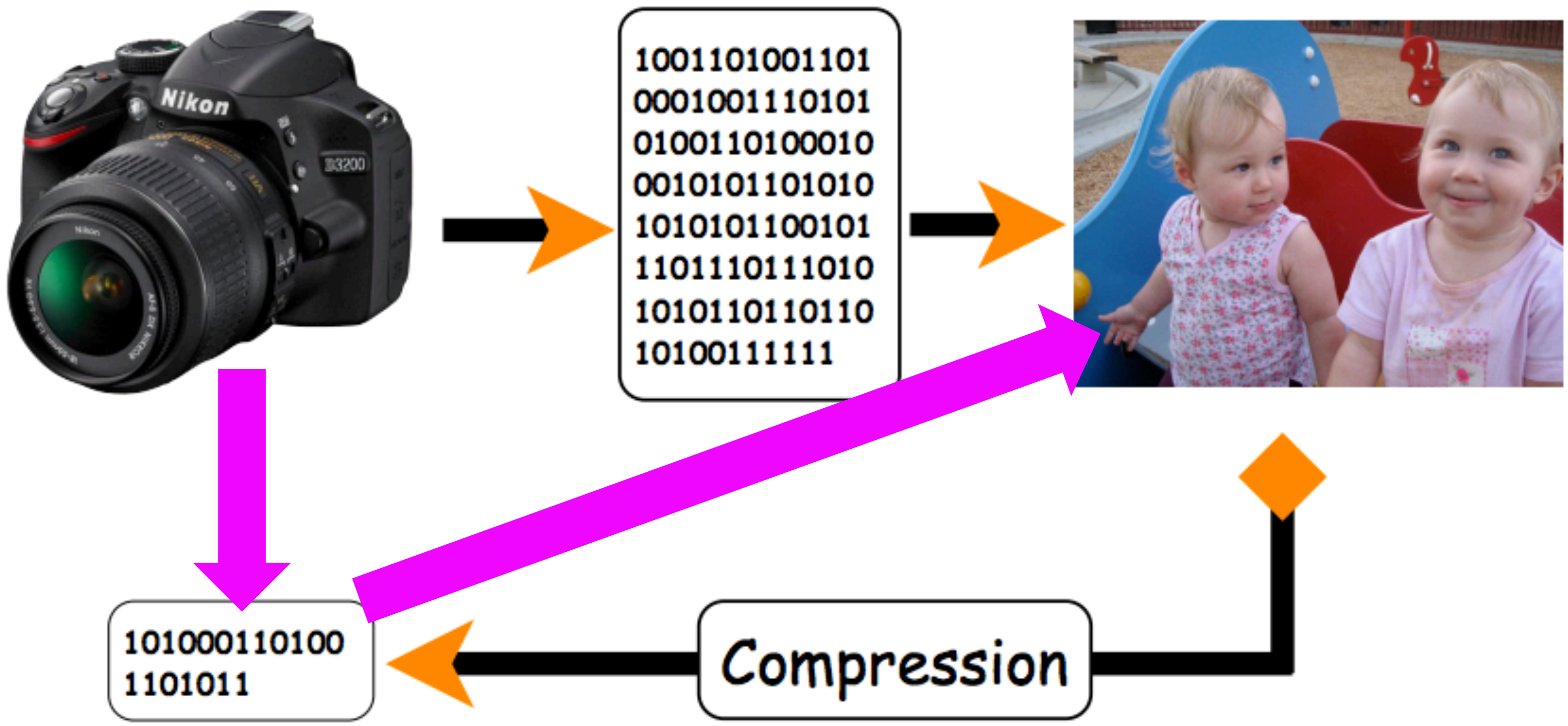
- Traditional DSP → sample first, ask questions later

Signal Processing Trends

- Traditional DSP → sample first, ask questions later
- Explosion in sensor technology/ubiquity has caused two trends:
  - Physical capabilities of hardware are being stressed, increasing speed/resolution becoming expensive
    - gigahertz+ analog-to-digital conversion
    - accelerated MRI
    - industrial imaging
  - Deluge of data
    - camera arrays and networks, multi-view target databases, streaming video...
- Compressive Sensing → sample smarter, not faster

Compressive Sensing/Sampling

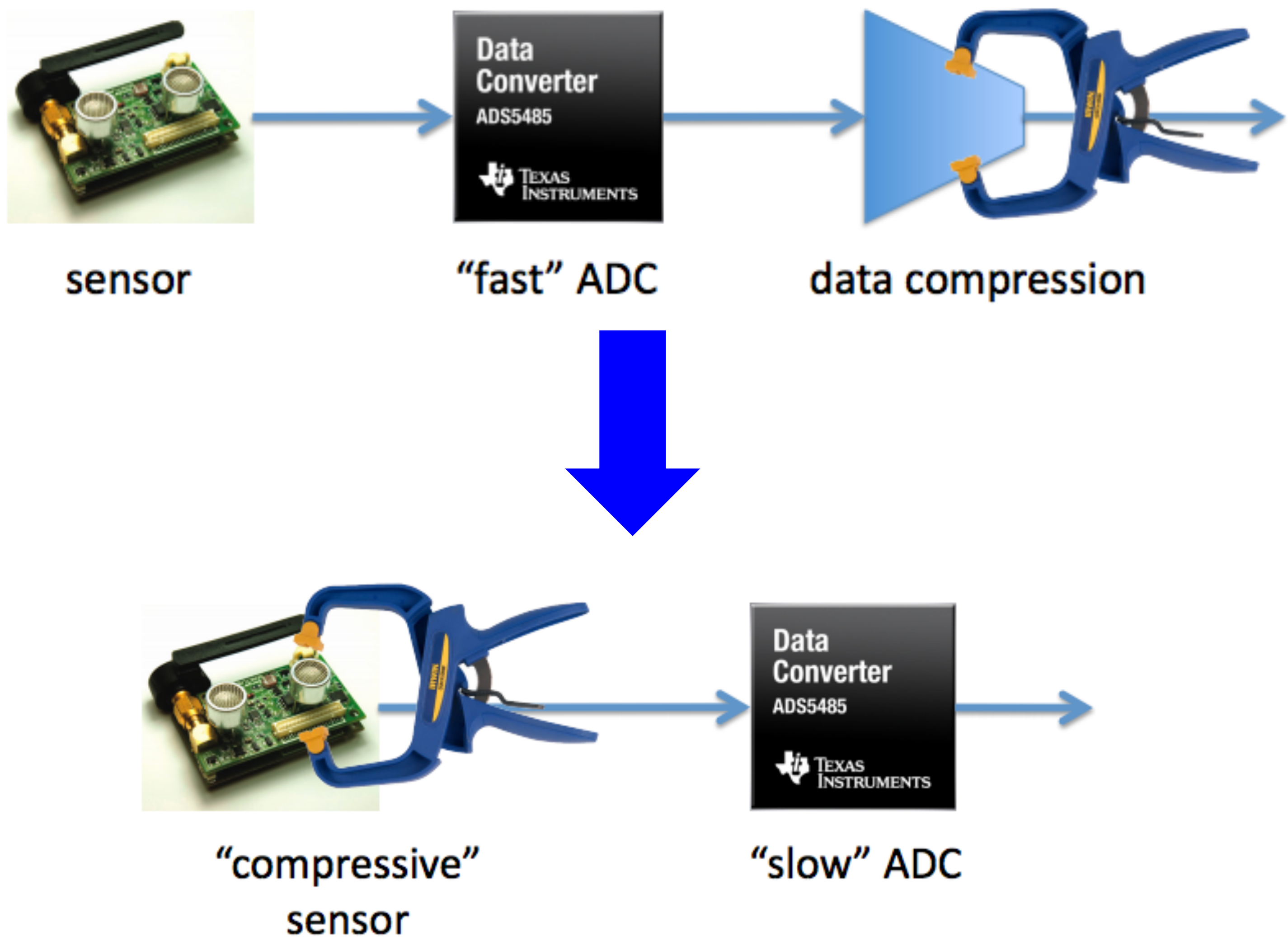
- Standard approach
  - First collect, then compress
    - Throw away unnecessary data



Compressive Sensing

- Shannon/Nyquist theorem is pessimistic
  - 2 × bandwidth is the worst-case sampling rate — holds uniformly for any bandlimited data
  - sparsity/compressibility is irrelevant
  - Shannon sampling based on a linear model, compression based on a nonlinear model
- Compressive sensing
  - new sampling theory that leverages compressibility
  - key roles played by new uncertainty principles and randomness

Sensing to Data



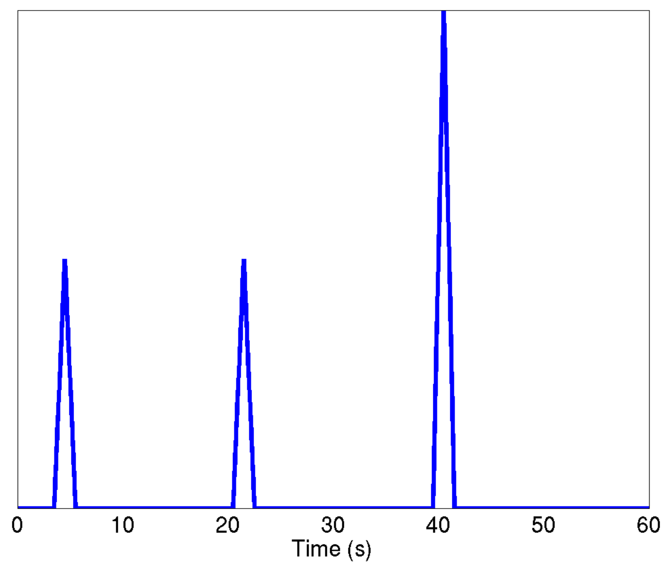
Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in most cases *exactly* recover

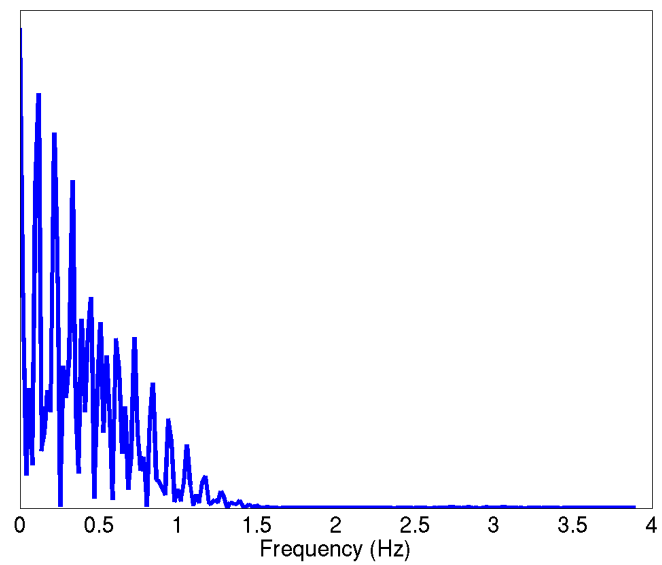
Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in most cases *exactly* recover

Sparse signal in time



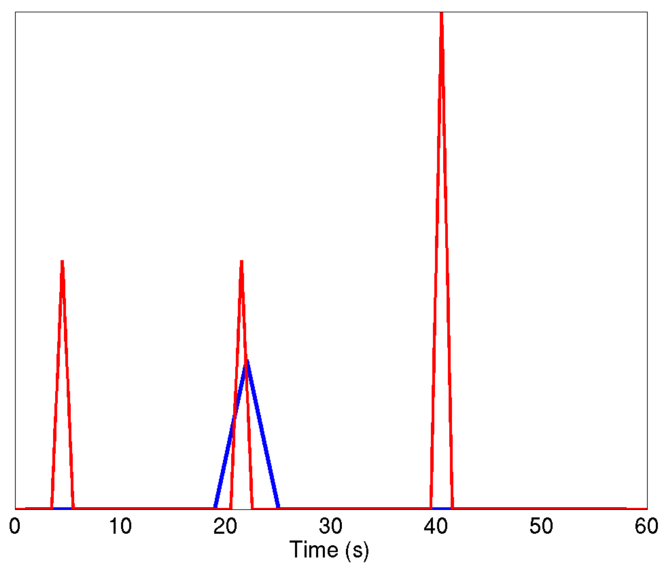
Frequency spectrum



Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in most cases *exactly* recover

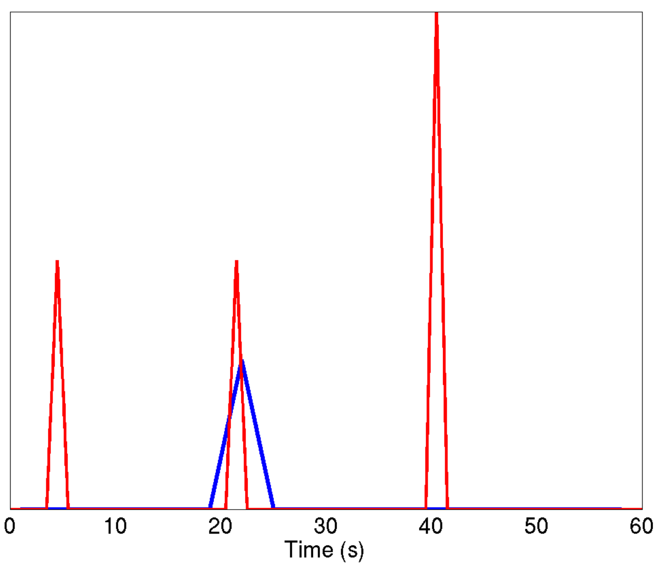
Undersampled in time

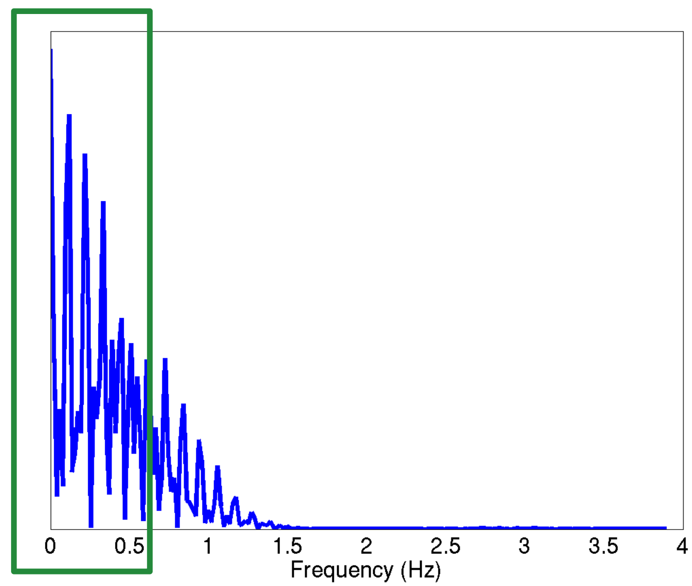


Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in most cases *exactly* recover

Undersampled in time

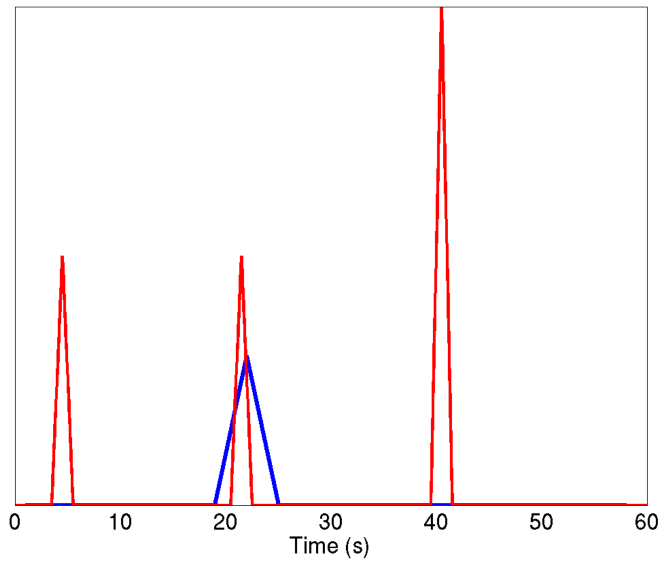




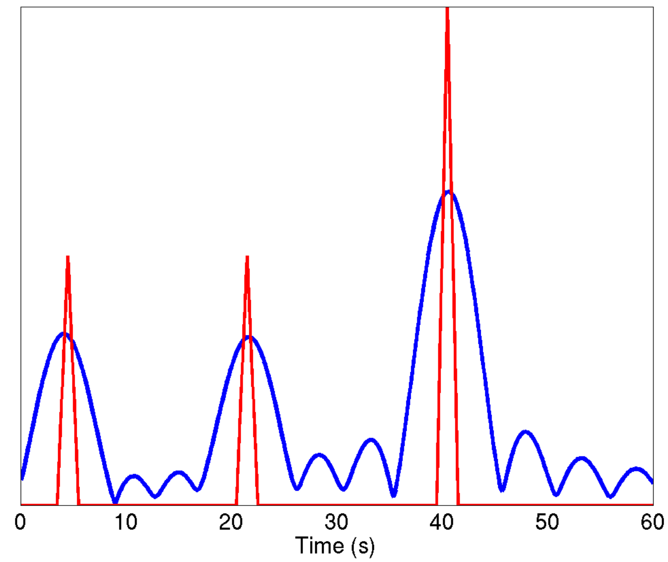
Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in most cases *exactly* recover

Undersampled in time



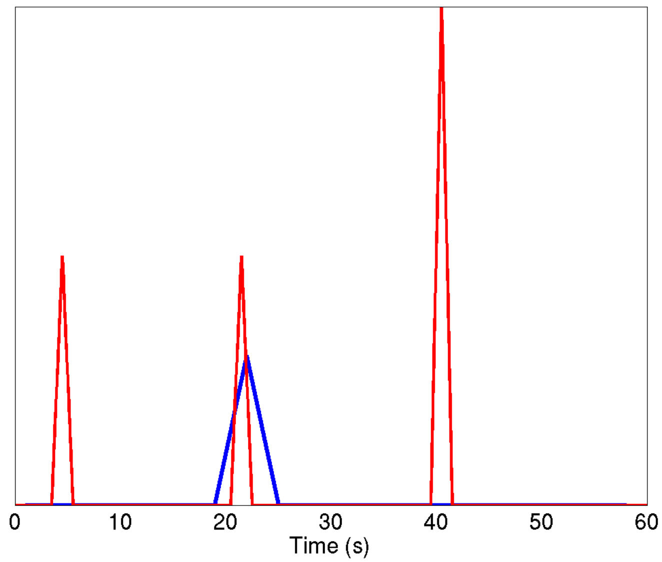
Undersampled in frequency
(reconstructed in time with IFFT)



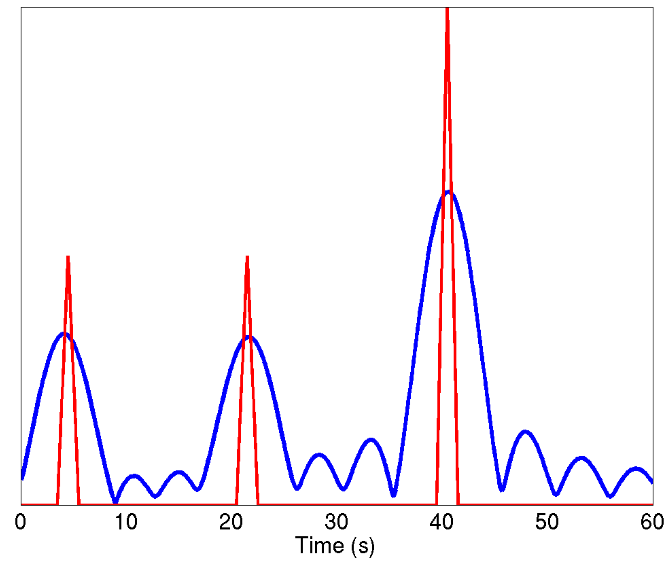
Compressive Sampling

- Sample at lower than the Nyquist rate and still accurately recover the signal, and in most cases *exactly* recover

Undersampled in time

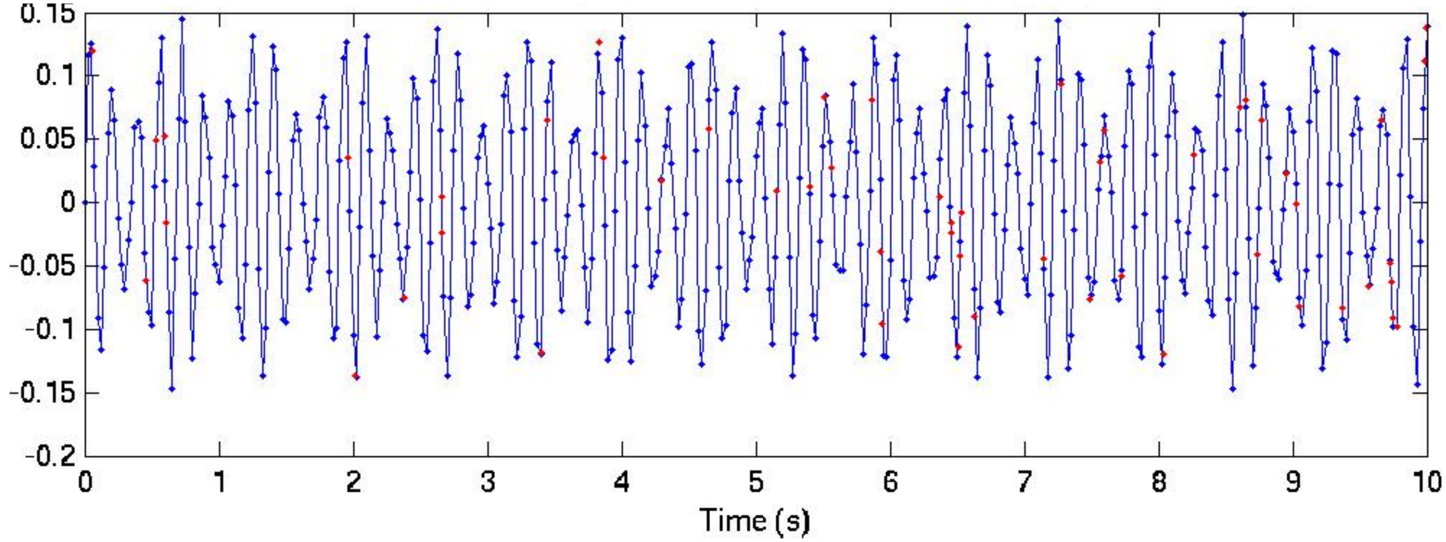


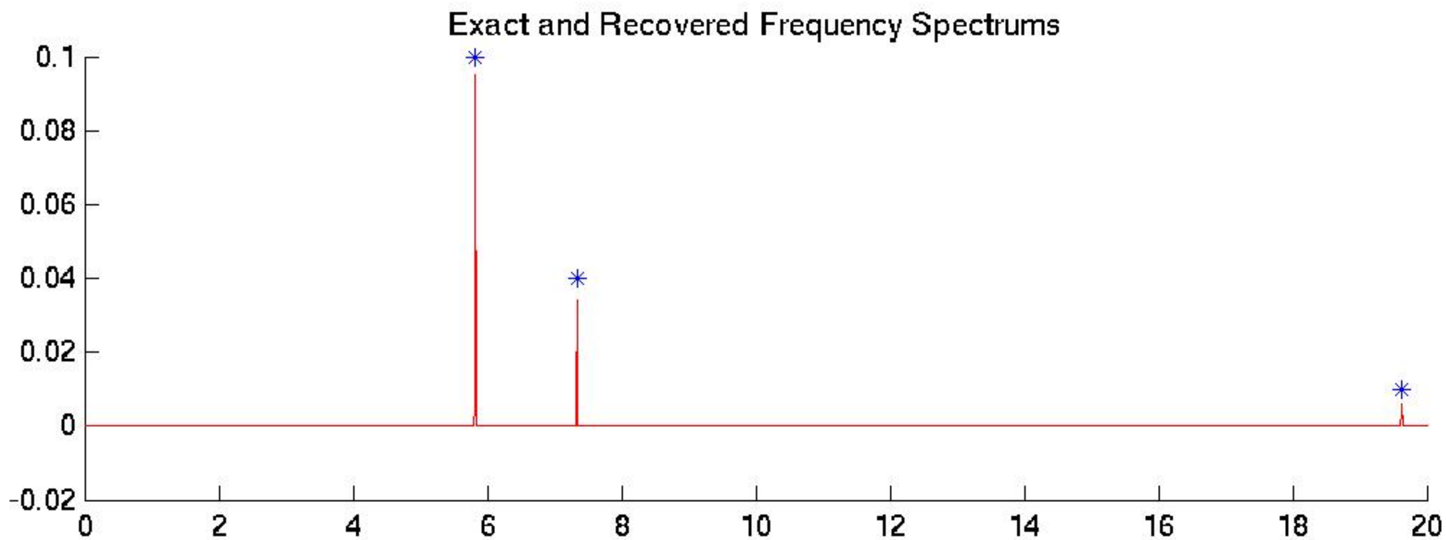
Undersampled in frequency
(reconstructed in time with IFFT)



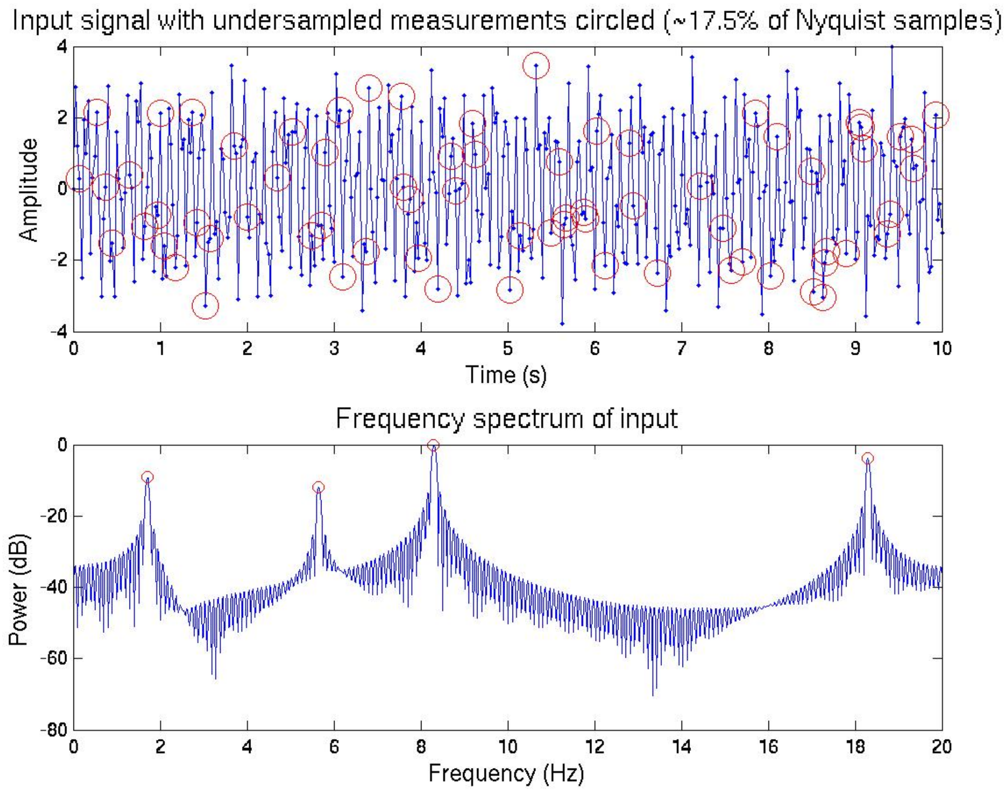
Requires sparsity and incoherent sampling

Compressive Sampling: Simple Example





Compressive Sampling

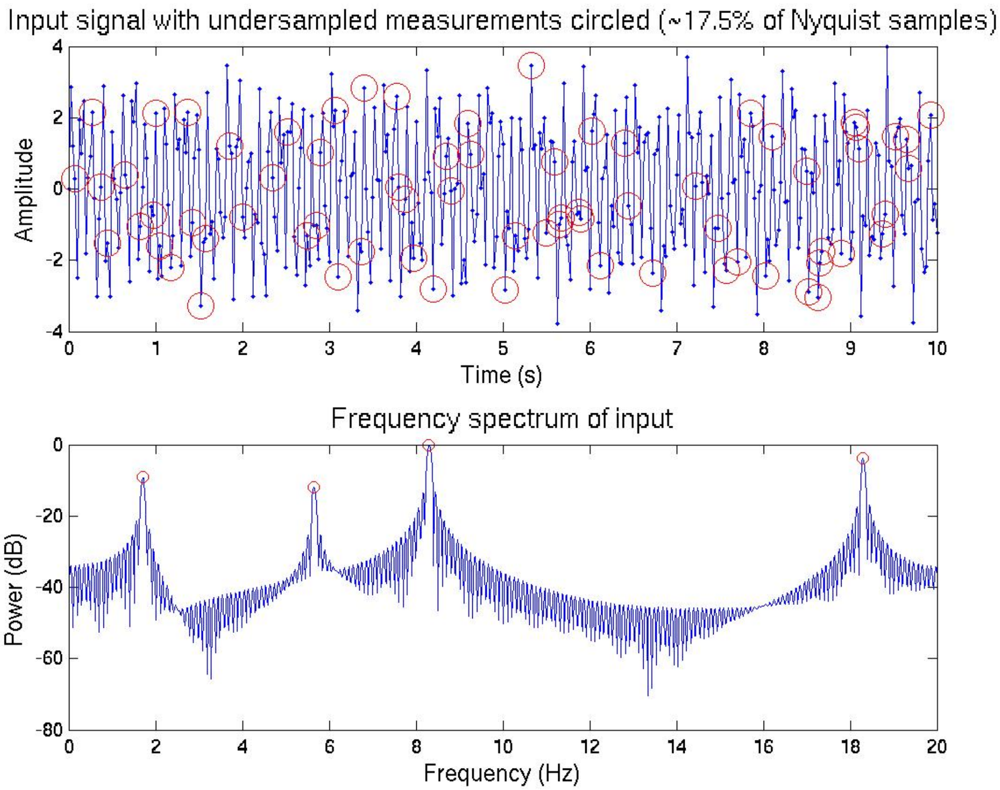


- Sense signal M times
- Recover with linear program

$$\min \sum_{\omega} |\hat{g}(\omega)| \quad \text{subject to} \quad g(t_m) = f(t_m), \quad m = 1, \ldots, M$$

Compressive Sampling

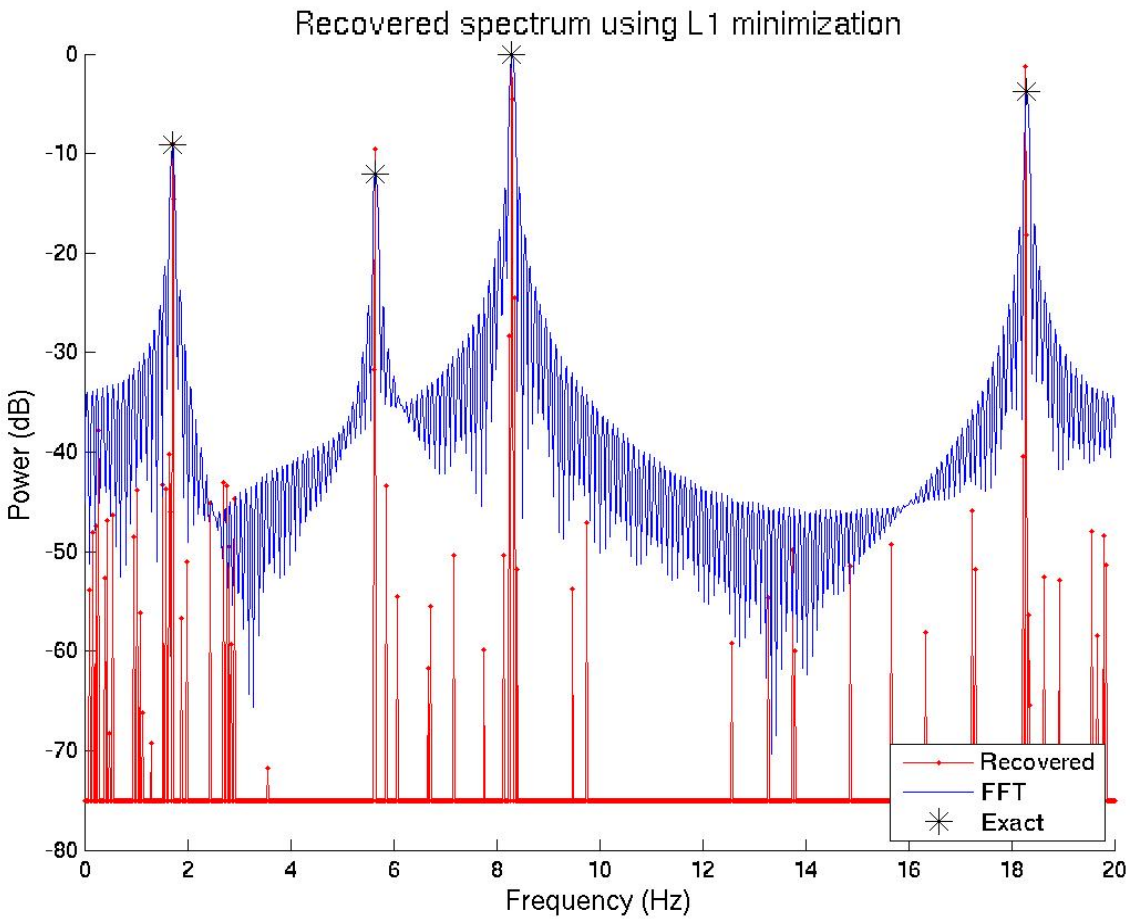
$$\hat{f}(\omega) = \sum_{i=1}^{K} \alpha_i \delta(\omega_i - \omega) \overset{\mathcal{F}}{\Leftrightarrow} f(t) = \sum_{i=1}^{K} \alpha_i e^{i\omega_i t}$$



- Sense signal M times
- Recover with linear program

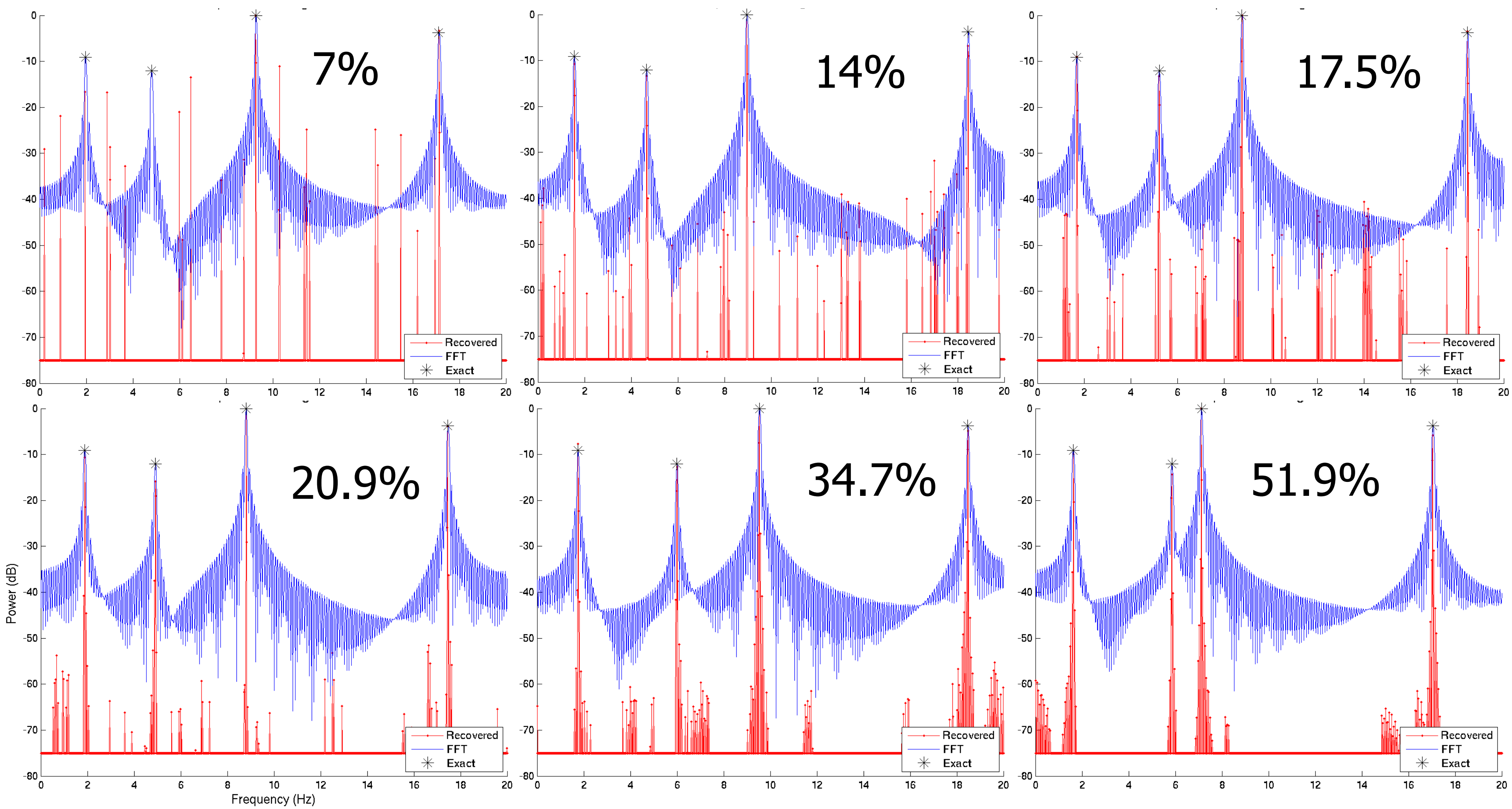
$$\min \sum_{\omega} |\hat{g}(\omega)| \quad \text{subject to} \quad g(t_m) = f(t_m), \quad m = 1, \ldots, M$$

Example: Sum of Sinusoids

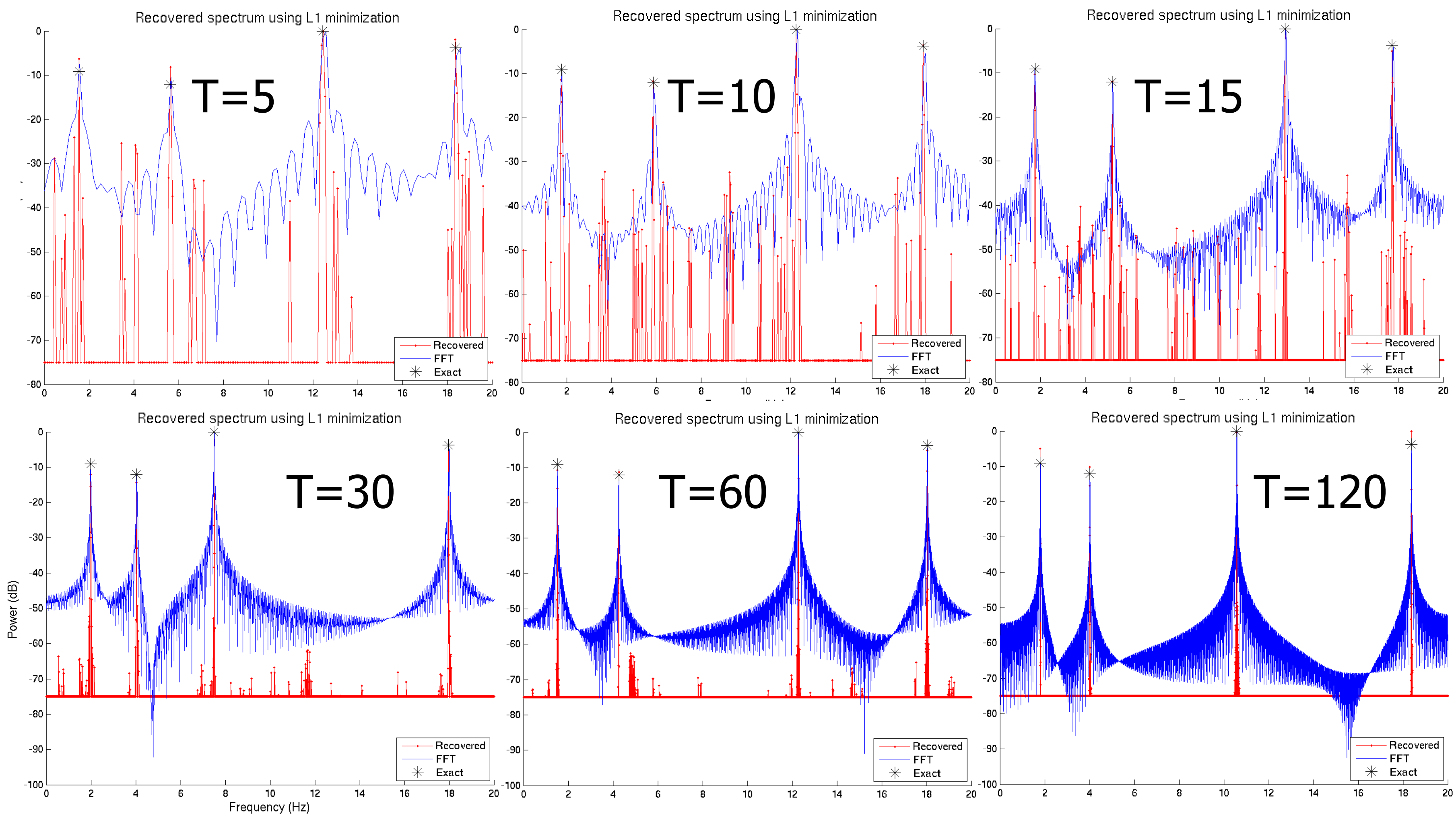


- Two relevant "knobs"
  - percentage of Nyquist samples as altered by adjusting number of samples, M
  - input signal duration, T
    - Data block size

Example: Increasing M

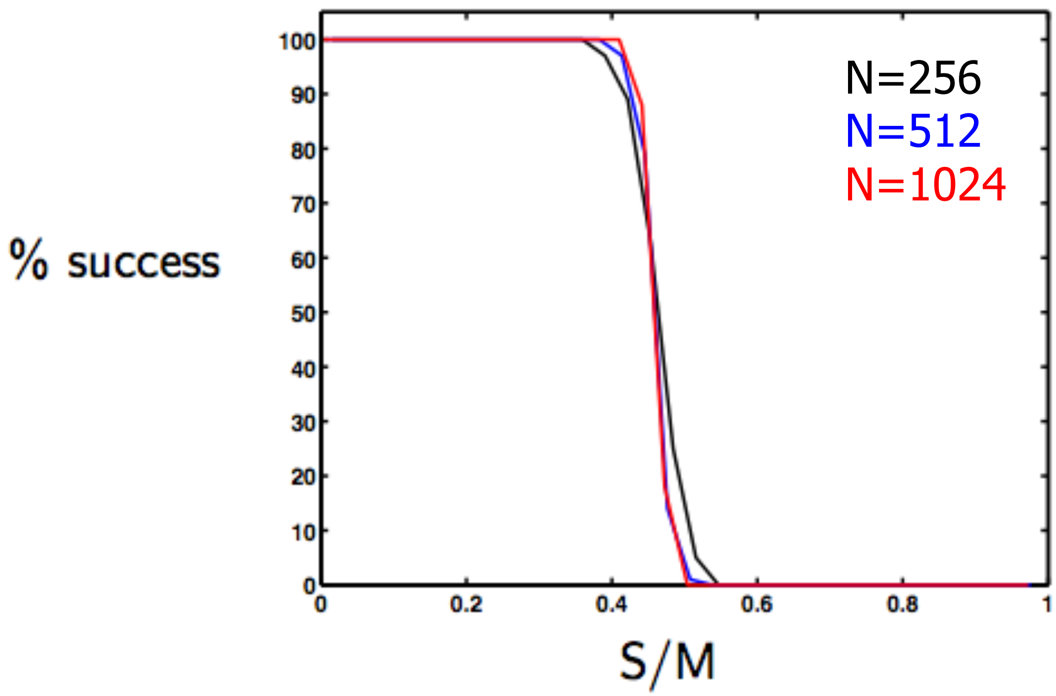


Example: Increasing T



Numerical Recovery Curves

- Sense S-sparse signal of length N randomly M times



  - In practice, perfect recovery occurs when $M \approx 2S$ for $N \approx 1000$

A Non-Linear Sampling Theorem

- Exact Recovery Theorem (Candès, R, Tao, 2004):
  - Select M sample locations $\{t_m\}$ "at random" with

$$M \geq \text{Const} \cdot S \log N$$

- Take time-domain samples (measurements)

$$y_m = x_0(t_m)$$

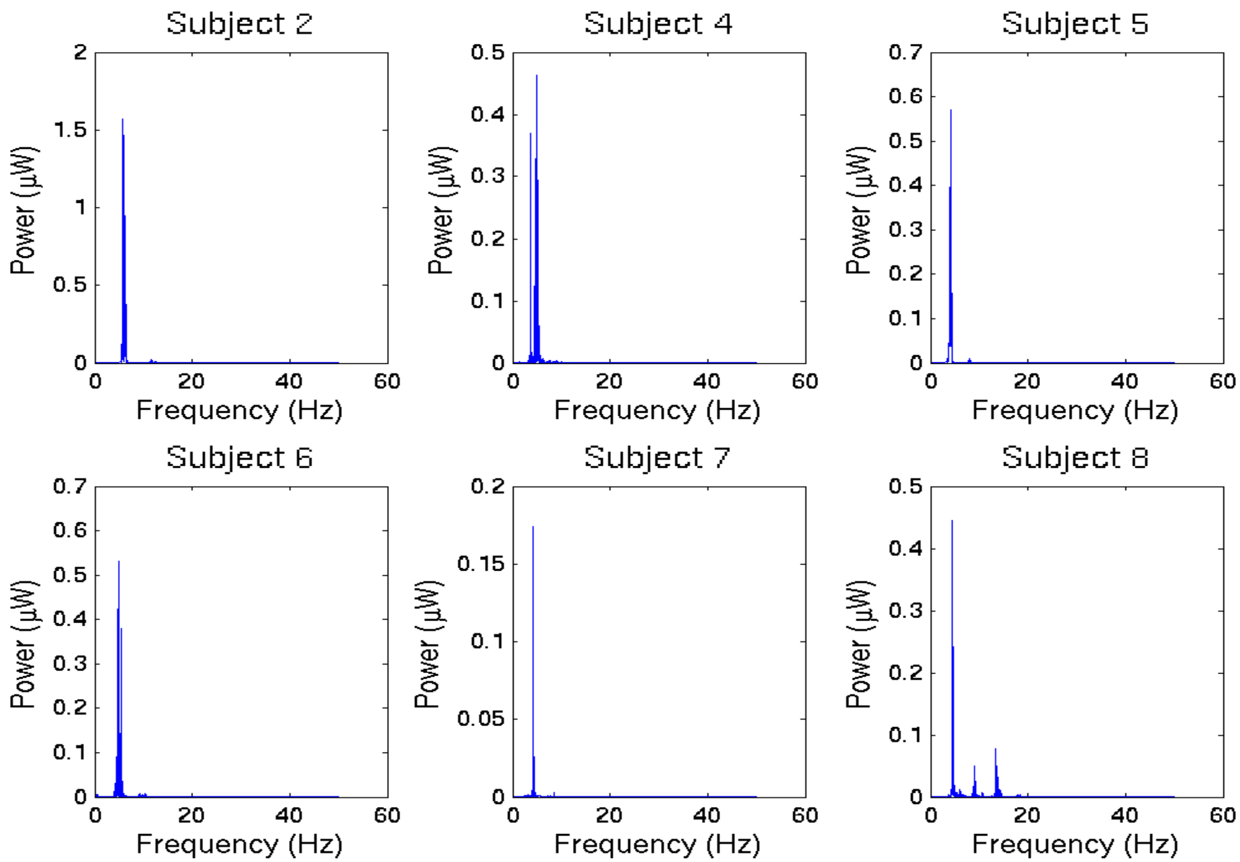
- Solve

$$\min_x \|\hat{x}\|_{\ell_1} \quad \textbf{subject to} \quad x(t_m) = y_m, \quad m = 1, \ldots, M$$

- Solution is exactly recovered signal with extremely high probability

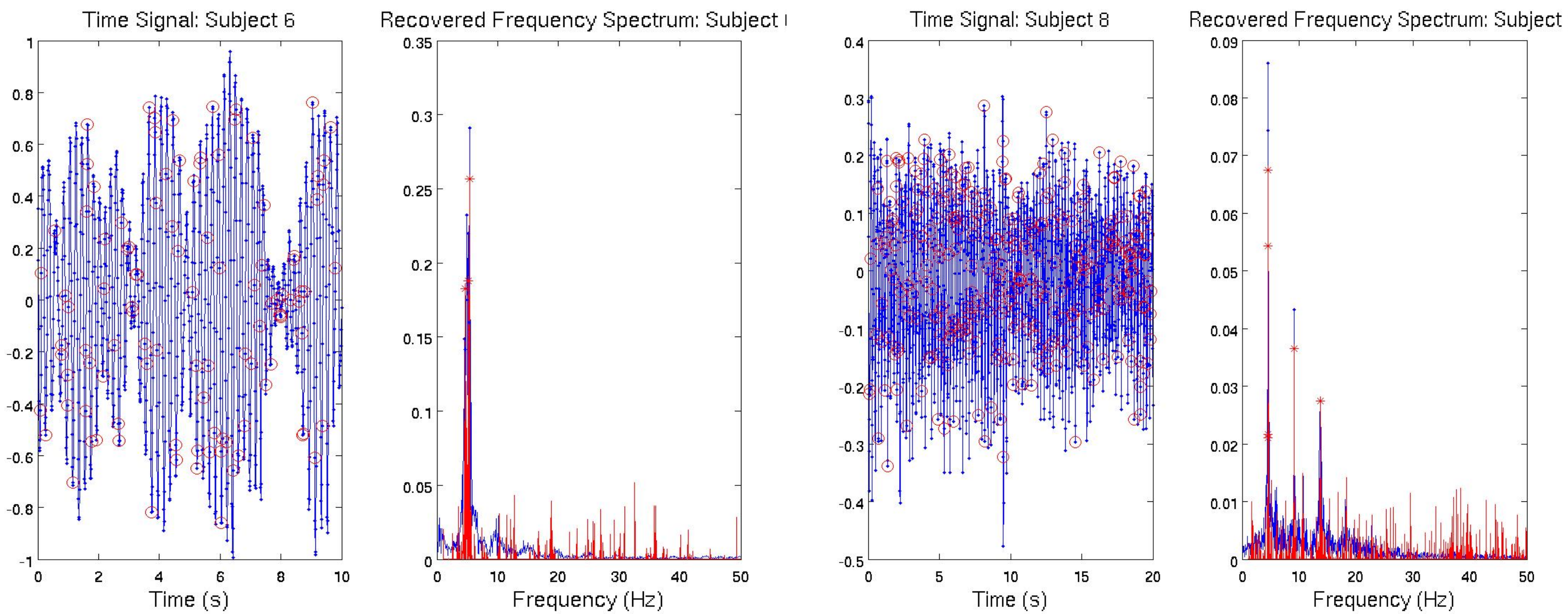
$M > C \cdot \mu^2(\Phi, \Psi) \cdot S \cdot \log N$

Biometric Example: Parkinson's Tremors

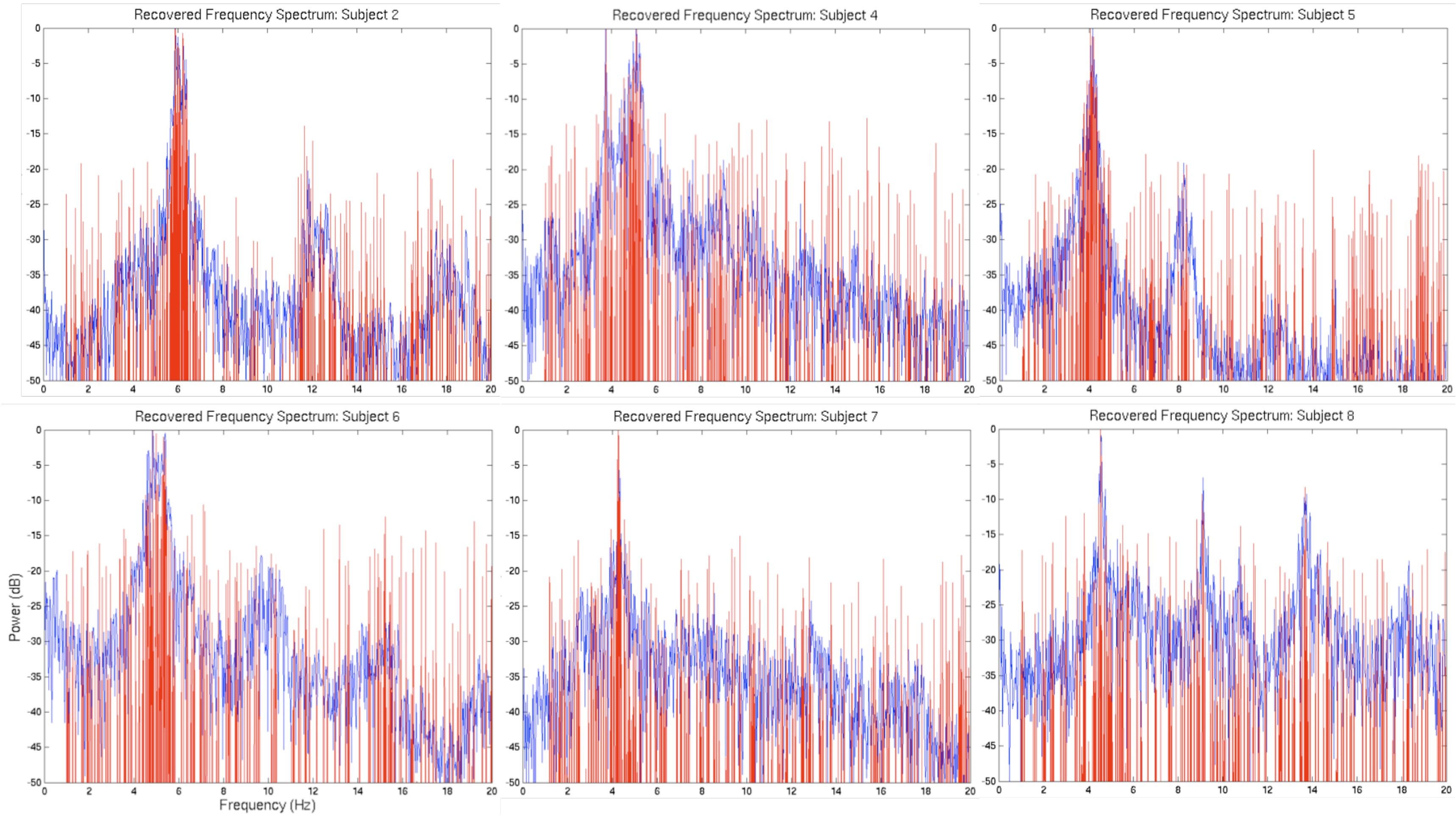


- 6 Subjects of real tremor data
  - collected using low intensity velocity-transducing laser recording aimed at reflective tape attached to the subjects' finger recording the finger velocity
  - All show Parkinson's tremor in the 4-6 Hz range.
  - Subject 8 shows activity at two higher frequencies
  - Subject 4 appears to have two tremors very close to each other in frequency

Compressive Sampling: Real Data

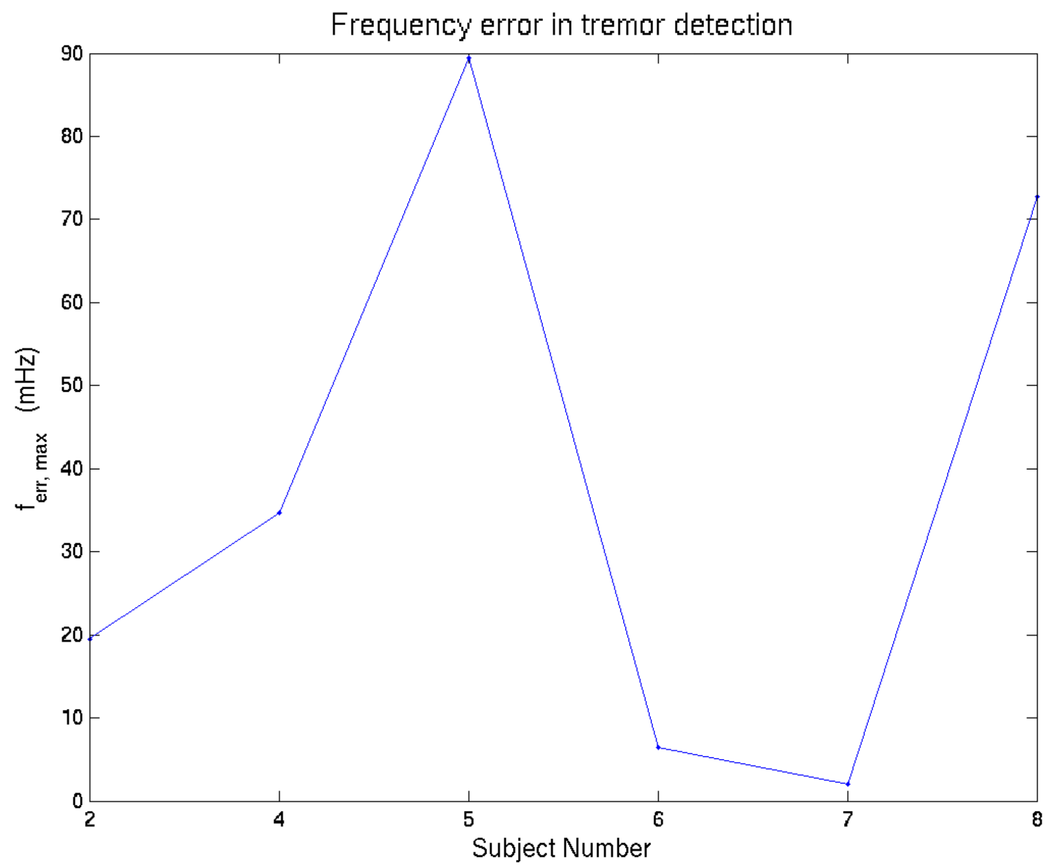


Biometric Example: Parkinson's Tremors



- **C=10.5, T=30**
  - 20% Nyquist required samples

Biometric Example: Parkinson's Tremors



- **Tremors detected within 100 mHz**
- **randomly sample 20% of the Nyquist required samples**

Requires post processing to randomly sample!

Implementing Compressive Sampling

- Devised a way to randomly sample 20% of the Nyquist required samples and still detect the tremor frequencies within 100mHz
  - Requires post processing to randomly sample!

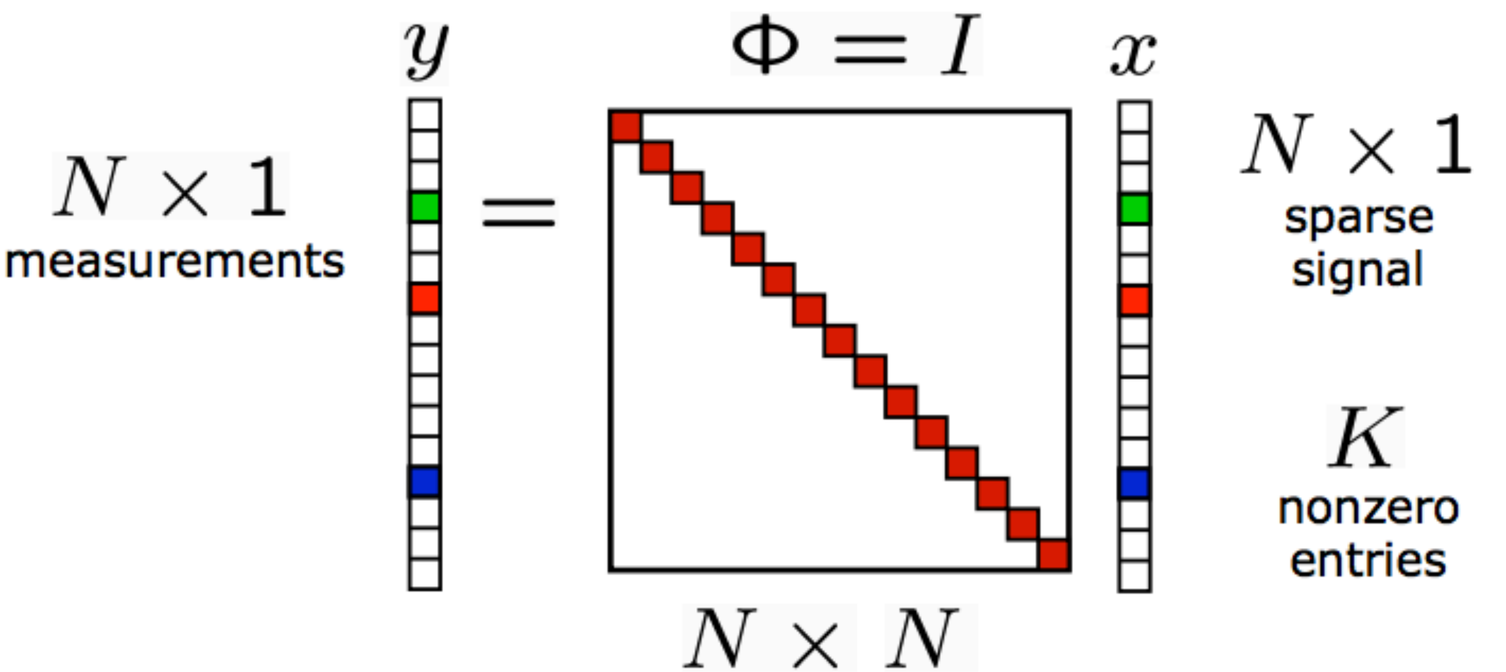
- Implement hardware on chip to "choose" samples in real time
  - Only write to memory the "chosen" samples
    - Design random-like sequence generator
  - Only convert the "chosen" samples
    - Design low energy ADC

CS Theory

Why does it work?

Sampling

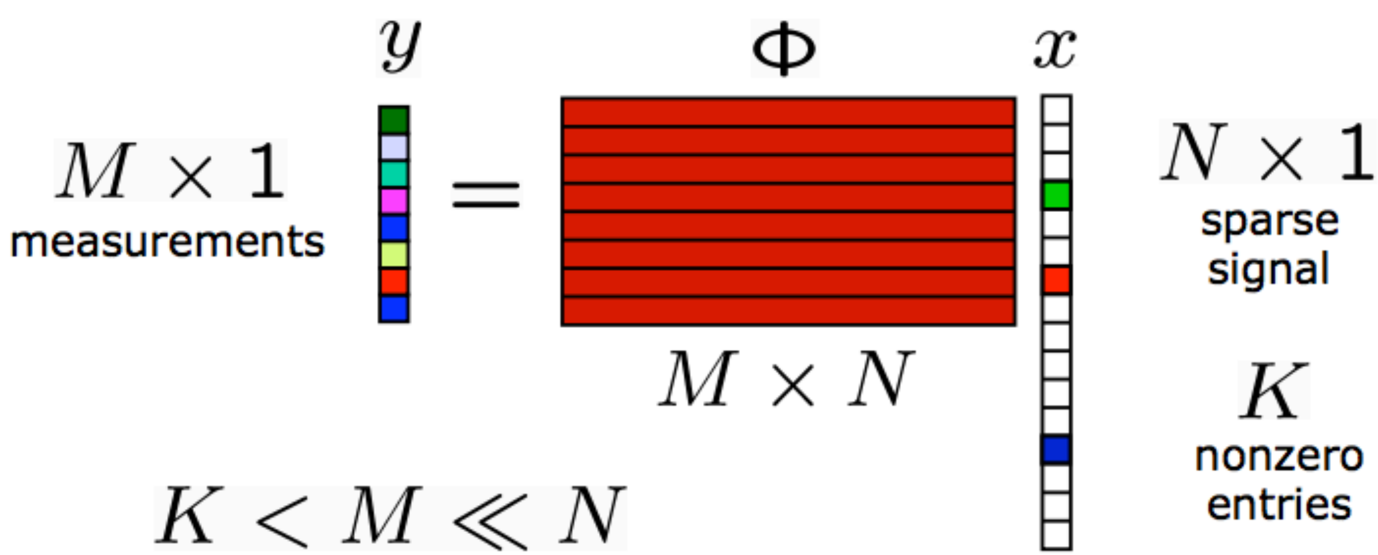
- Signal x is K-*sparse* in basis/dictionary Ψ
  - WLOG assume sparse in space domain $\Psi = I$
- **Sampling**



Compressive Sampling

- When data is sparse/compressible, can directly acquire a ***condensed representation*** with no/little information loss through linear ***dimensionality reduction***

$$y = \Phi x$$

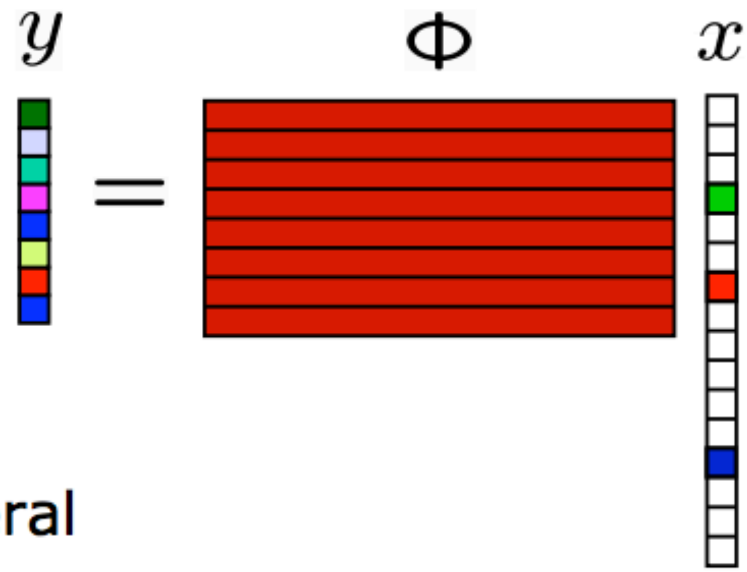


How Can It Work?

- Projection Φ
  not full rank...

  $M < N$

  ... and so
  loses information in general



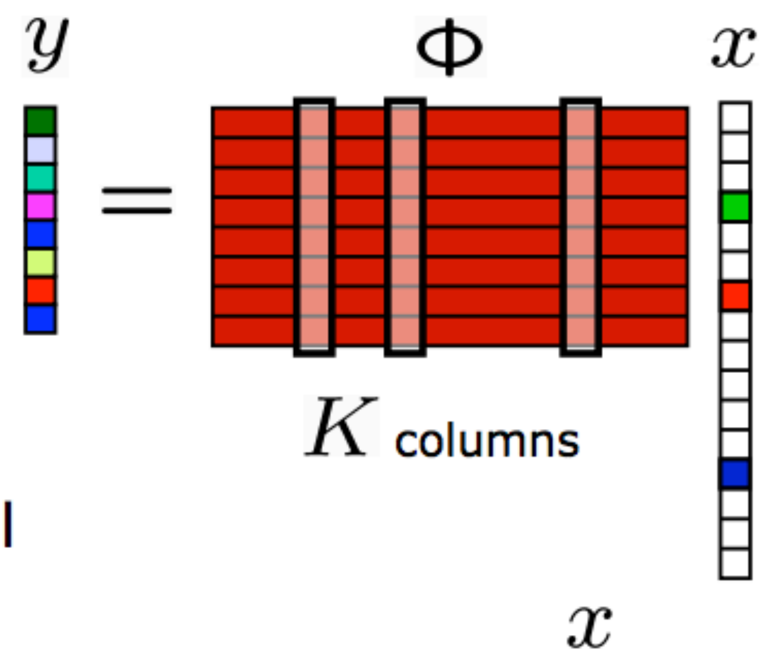
- Ex: Infinitely many x's map to the same y (null space)

How Can It Work?

- Projection Φ
  not full rank...

  $M < N$

  ... and so
  loses information in general



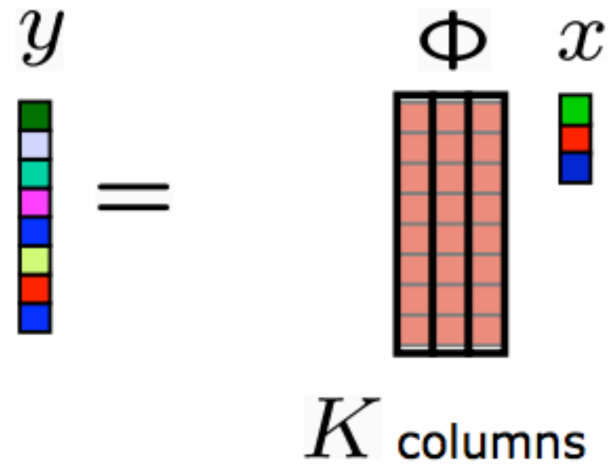
- But we are only interested in ***sparse*** vectors

How Can It Work?

- Projection Φ
  not full rank…

  $M < N$

  … and so
  loses information in general



- But we are only interested in ***sparse*** vectors

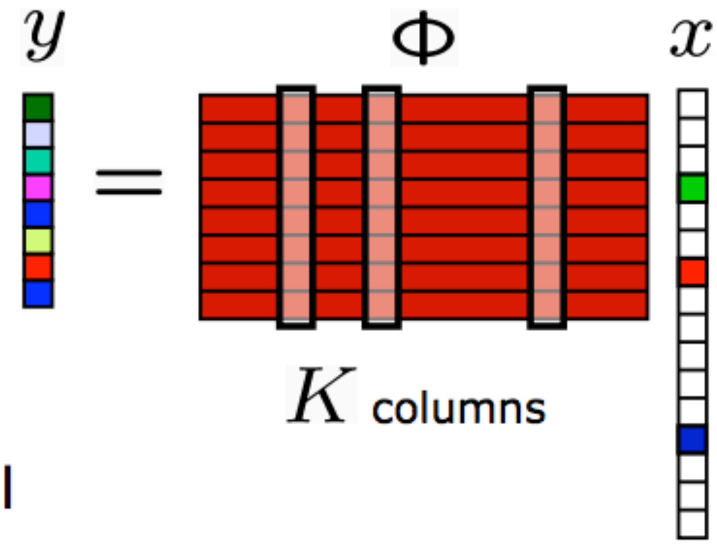
- Φ is effectively *MxK*

How Can It Work?

- Projection Φ not full rank…

  $M < N$

  … and so loses information in general



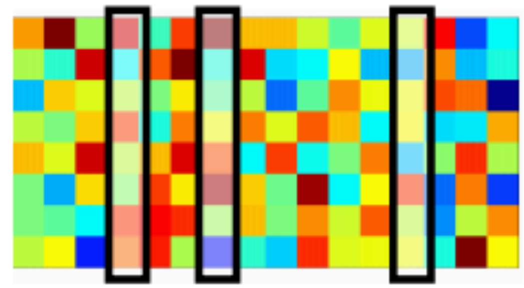
- But we are only interested in ***sparse*** vectors

- **Design** Φ so that each of its *MxK* submatrices are full rank (ideally close to orthobasis)
  - **Restricted Isometry Property (RIP)**

Restricted Isometric Property (RIP)

- Draw Φ at **random**
  - iid Gaussian
  - iid Bernoulli ± 1

  ...

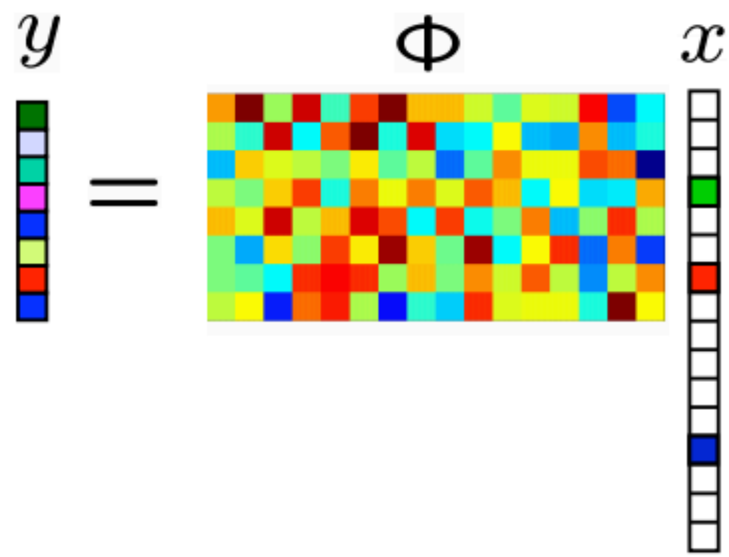


- Then Φ has the RIP with high probability provided

$$M = O(K \log(N/K)) \ll N$$

CS Signal Recovery

- **Goal**: Recover signal x from measurements y

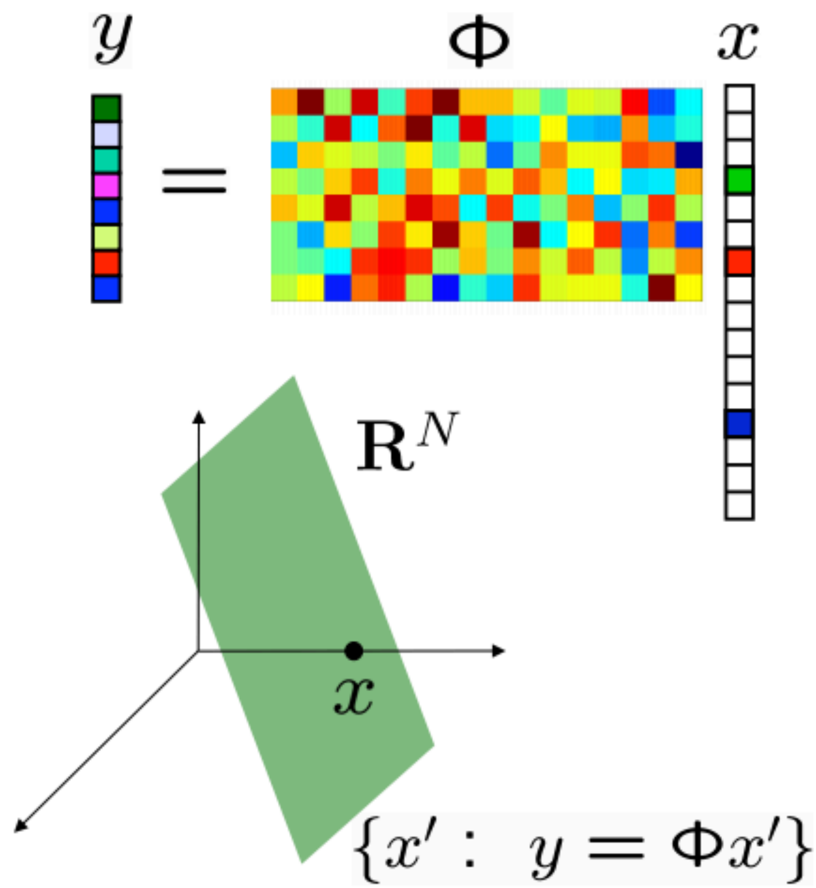


- **Problem**: Random projection Φ not full rank (ill-posed inverse problem)

- **Solution**: Exploit the sparse/compressible ***geometry*** of acquired signal x

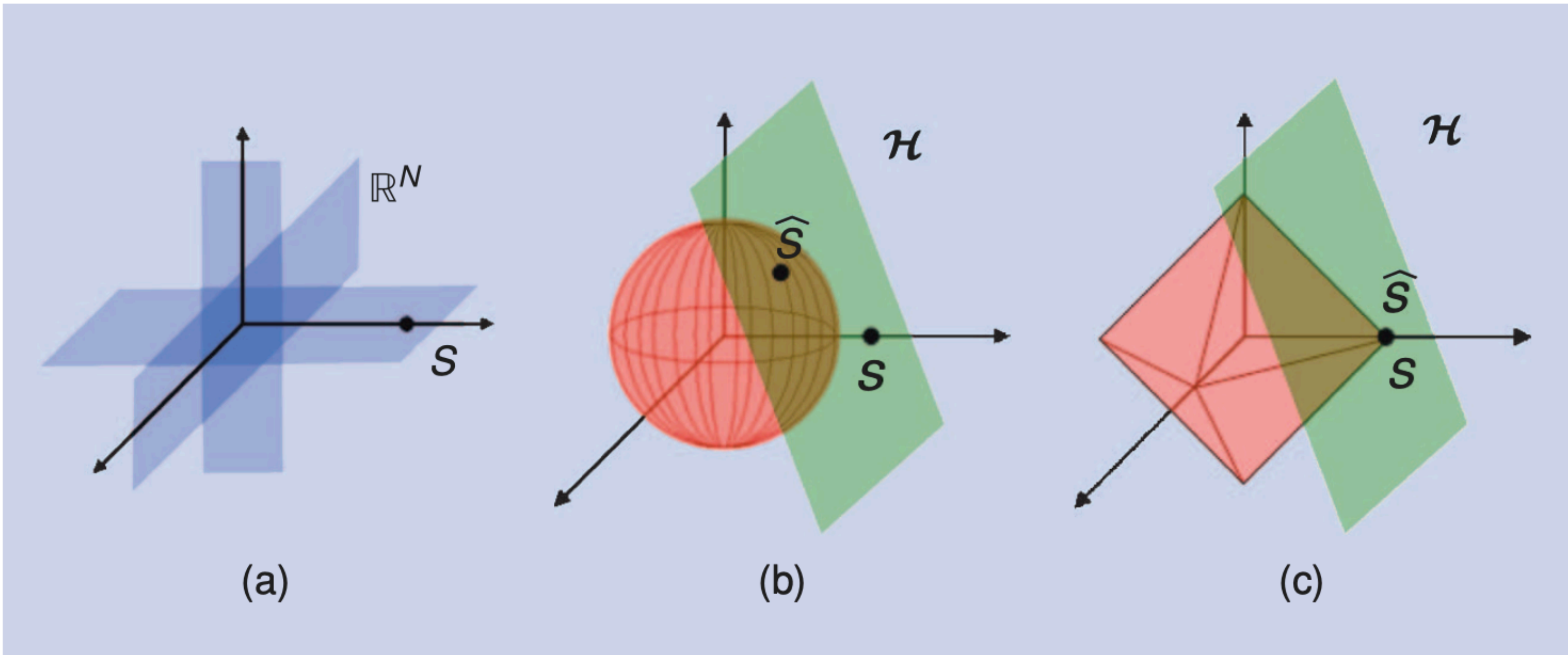
CS Signal Recovery

- Random projection Φ not full rank
- Recovery problem: given $y = \Phi x$ find x
- **Null space**
- Search in null space for the "best" x according to some criterion
  - ex: least squares



*(N-M)-dim hyperplane at **random angle***

CS Recovery

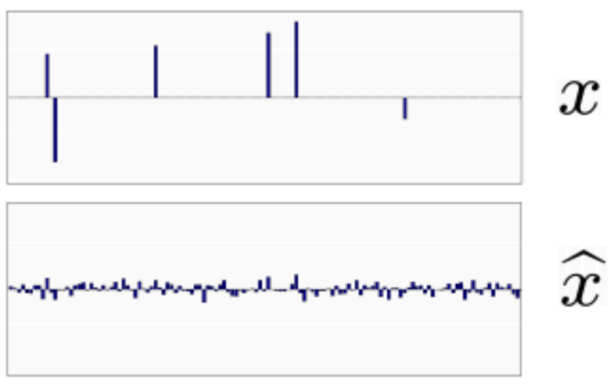


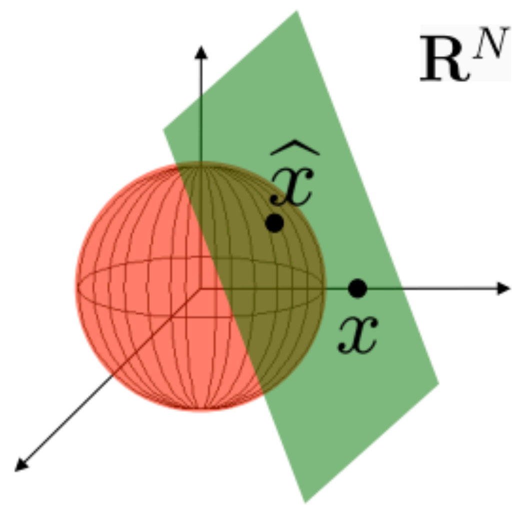
[FIG2] (a) The subspaces containing two sparse vectors in $\mathbb{R}^3$ lie close to the coordinate axes. (b) Visualization of the ℓ_2 minimization (5) that finds the non-sparse point-of-contact $\hat{s}$ between the ℓ_2 ball (hypersphere, in red) and the translated measurement matrix null space (in green). (c) Visualization of the ℓ_1 minimization solution that finds the sparse point-of-contact $\hat{s}$ with high probability thanks to the pointiness of the ℓ_1 ball.

Baraniuk, Richard. "Compressive Sensing [Lecture Notes]." *IEEE Signal Processing Magazine* 24 (2007): 118-121.

L_2 Signal Recovery

- **Recovery:** (ill-posed inverse problem) — given $y = \Phi x$, find x (sparse)
- **Optimization:** $\widehat{x} = \arg\min_{y=\Phi x} \|x\|_2$
- **Closed-form solution:** $\widehat{x} = (\Phi^T \Phi)^{-1} \Phi^T y$
- **Wrong answer!**





L_0 Signal Recovery

- Recovery:
  (ill-posed inverse problem)

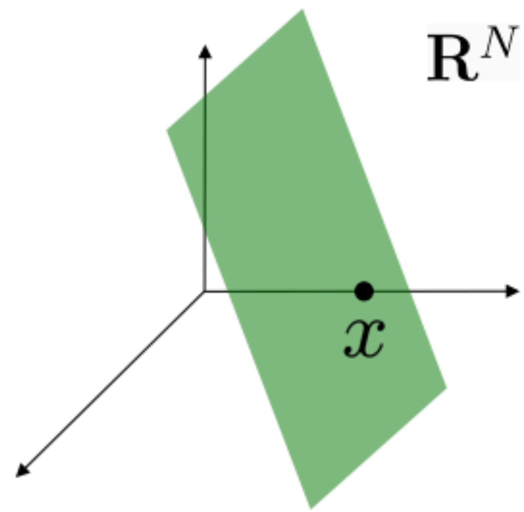
given $y = \Phi x$
find x (sparse)

- Optimization:

$$\hat{x} = \arg\min_{y=\Phi x} \|x\|_0$$

- **Correct!**

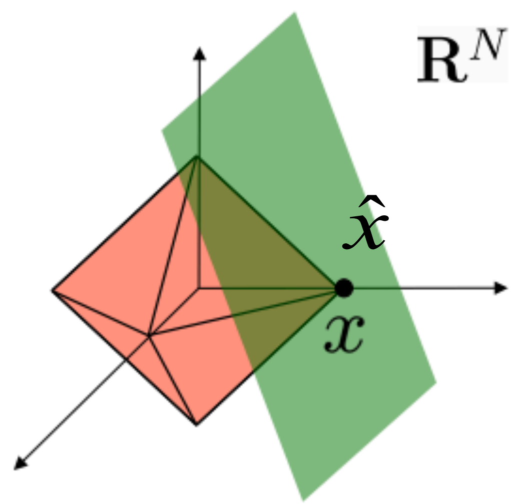
"find sparsest vector in translated nullspace"



- But **NP-Complete** alg

L_1 Signal Recovery

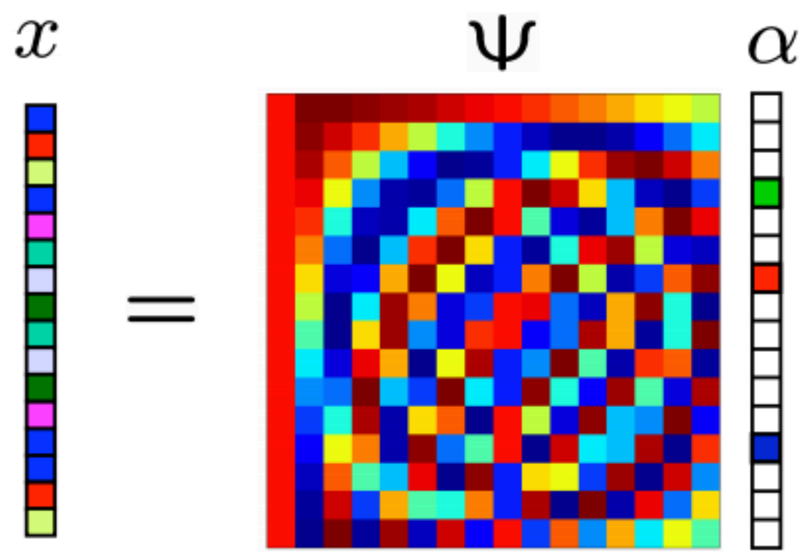
- Recovery: given $y = \Phi x$
  (ill-posed inverse problem) find x (sparse)
- Optimization: $\hat{x} = \arg\min_{y=\Phi x} \|x\|_1$
- **Convexify** the ℓ_0 optimization
- **Correct!**
- **Polynomial time** alg
  (linear programming)
- Much recent alg progress
  - greedy, Bayesian approaches, …



Universality

- Random measurements can be used for signals sparse in *any* basis

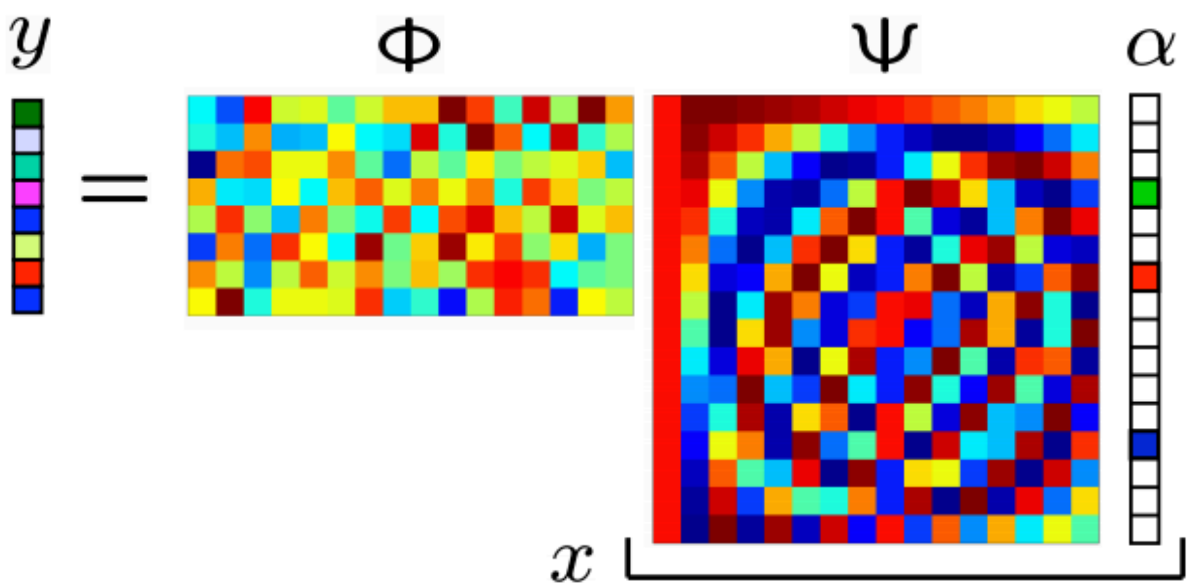
$$x = \Psi\alpha$$



Universality

- Random measurements can be used for signals sparse in *any* basis

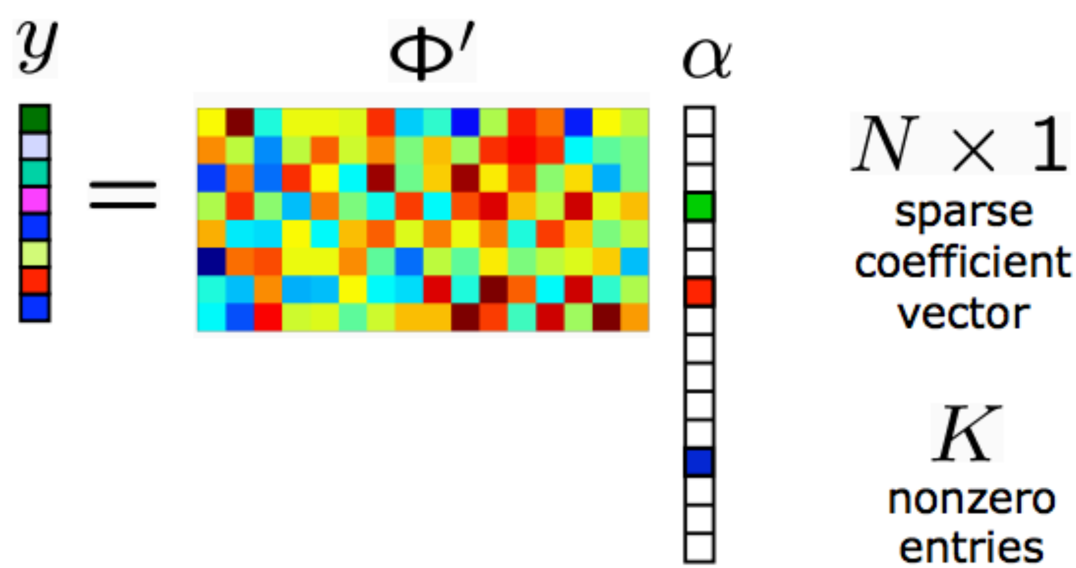
$$y = \Phi x = \Phi \Psi \alpha$$



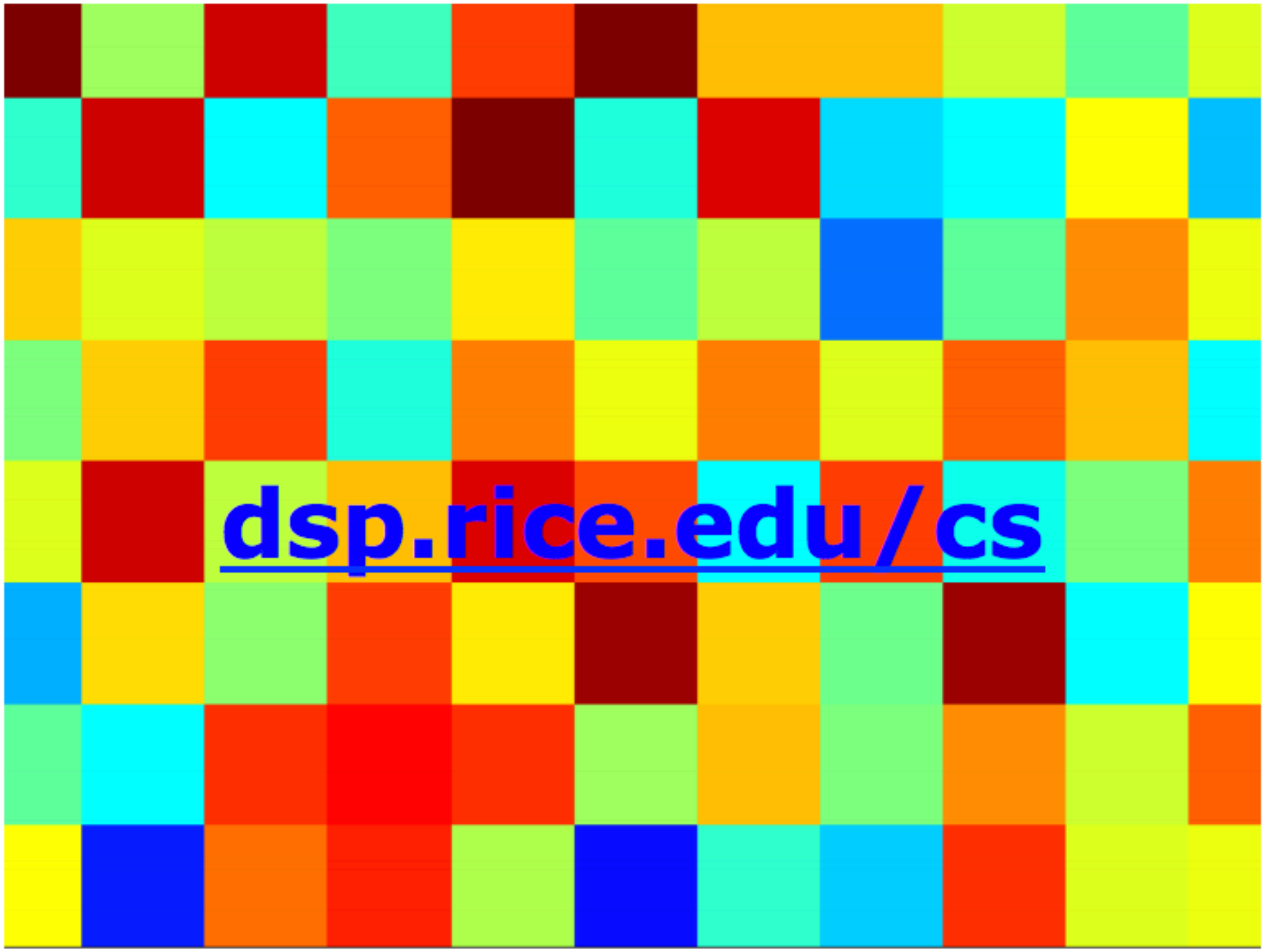
Universality

- Random measurements can be used for signals sparse in *any* basis

$$y = \Phi x = \Phi \Psi \alpha = \Phi' \alpha$$



Reference Slide



Big Ideas

- Wavelet transform
  - Capture temporal data with fewer coefficients than STFT
  - Use scaling and translation to get different resolution at different levels
- Compressive Sampling
  - Integrated sensing/sampling, compression and processing
  - Based on sparsity and incoherency

Admin

- Project 2 - Due 5/1
- Final Exam – 5/10 3-5pm in DRLB A8
  - Cumulative – covers lec 1-22
    - Except data converters, noise shaping (lec 11)
    - Closed book
    - 2 8.5x11 two-sided cheat sheets allowed
    - Calculators allowed, no smart phones
      - Can't share. Bring your own.
  - Old exams posted
    - Disclaimers: old exams had different coverage for different years
  - Review Session – poll in Ed
  - Last day of office hours – May 9th