EE123
Digital Signal Processing

Lecture 13C
Lab 4 Part II - AFSK
Compressive Sampling

M. Lustig, EECS UC Berkeley

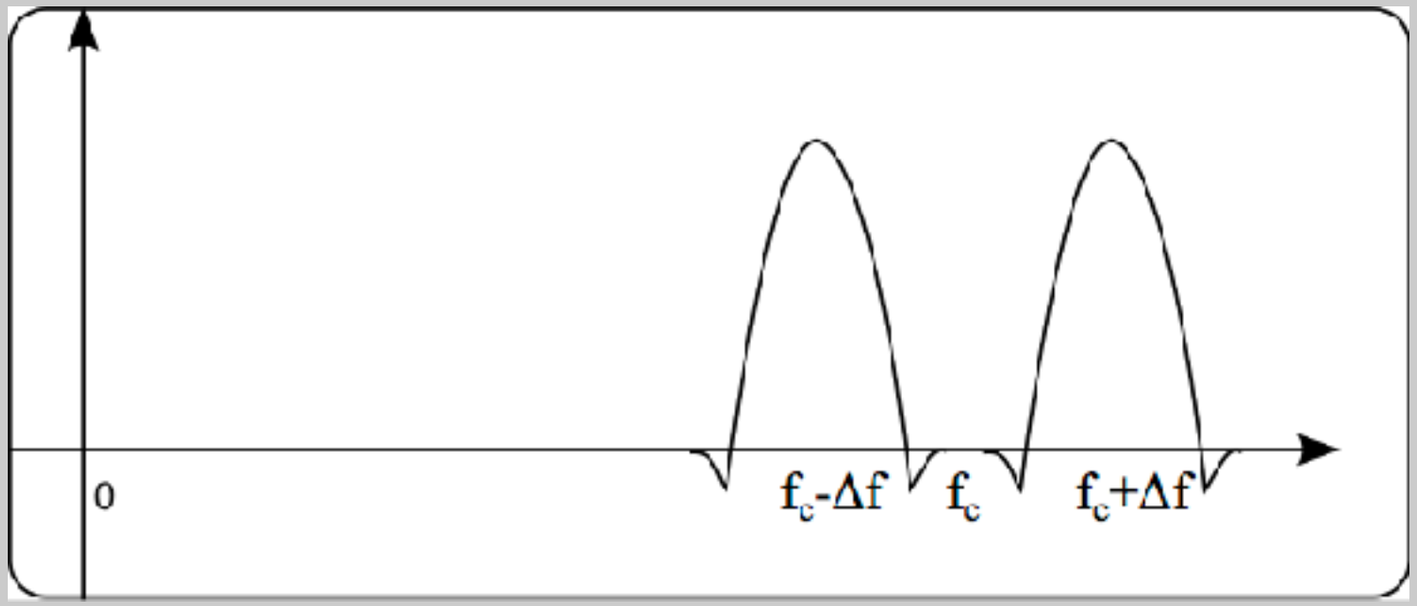
AFSK1200 / Bell 202 modem

- Audio FSK
  - Encodes digital data at 1200b/s
  - Use audio frequencies 1200/2200Hz
  - Within the bandwidth of the audio input BP filter of your radios
  - Still(!) popular for ham packet networks

$$s(t) = cos\left(2\pi f_c t + 2\pi\Delta f \int_{-\infty}^{t} m(\tau)d\tau\right)$$

  - fc = 1700, Δf = 500, m(t) = ± 1
  - Phase is not the same for each bit -- must use non coherent detection.

AFSK 1200



- For spectrum to be narrow, need continuous phase
- NRZ signal, m(t) = 1, or -1 for a duration of a bit, then

$$s(t) = cos\left(2\pi f_c t + 2\pi\Delta f \int_{\infty}^{t} m(\tau)d\tau\right)$$

$$2\pi f_c + 2\pi\Delta f m(t)$$

Write a modulator

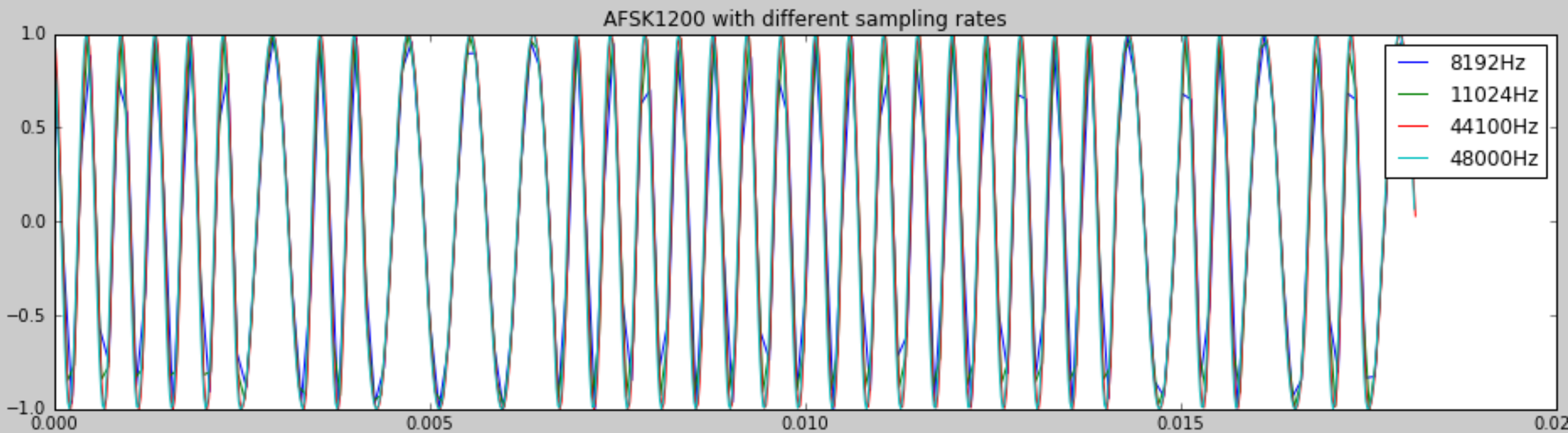
- sig = afsk1200(bits,fs)

- If fs does not divide with 1200, then each bit has a fractional sample
  - Can cause drift

- Solution: Generate signal at higher rate that divides with 1200, and downsample

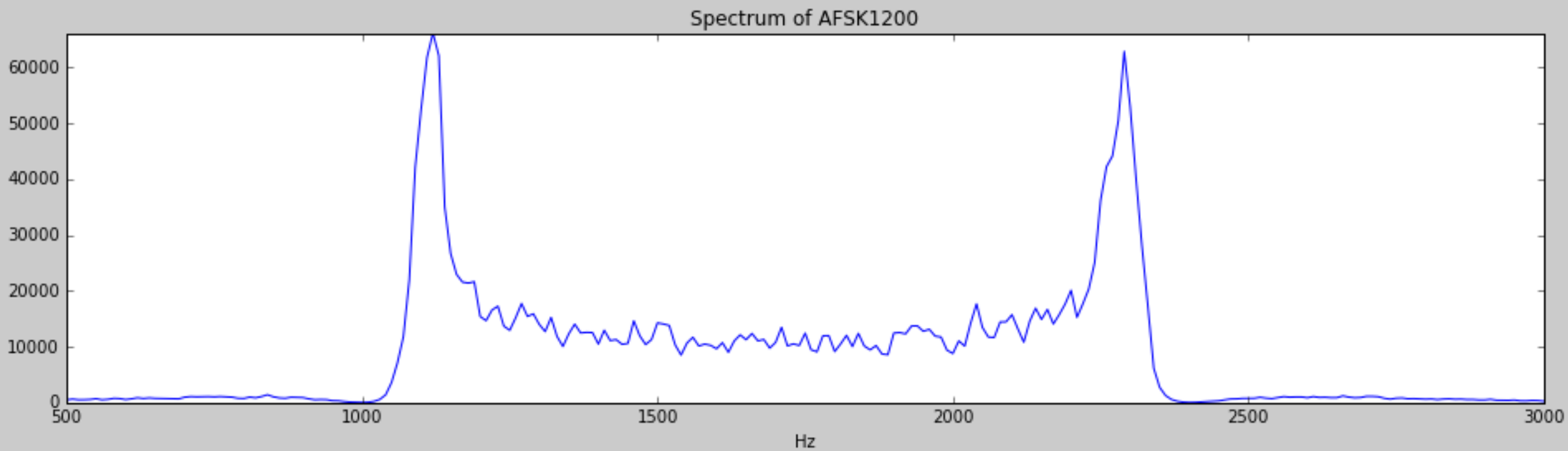
Modulator

- We will give you a sequence to compare to our implementation



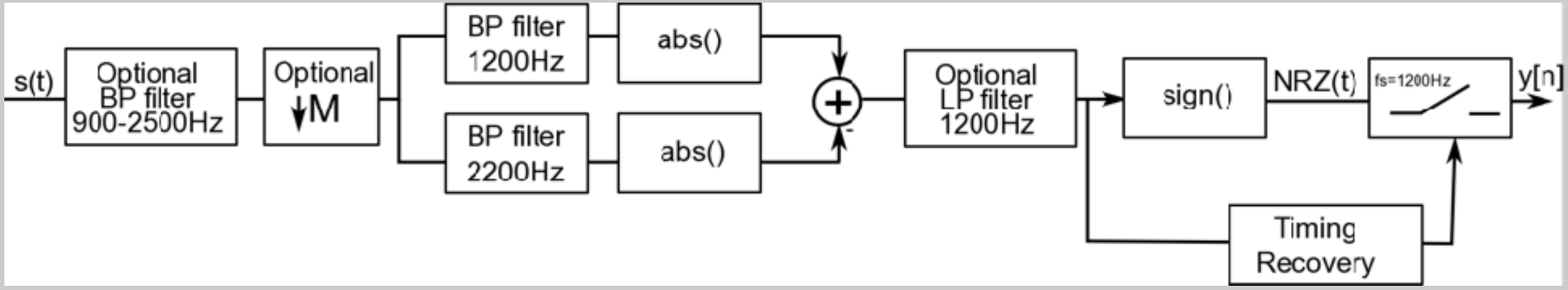
Spectrum of AFSK1200

- Generate random 4096 bits, and compute average power spectrum

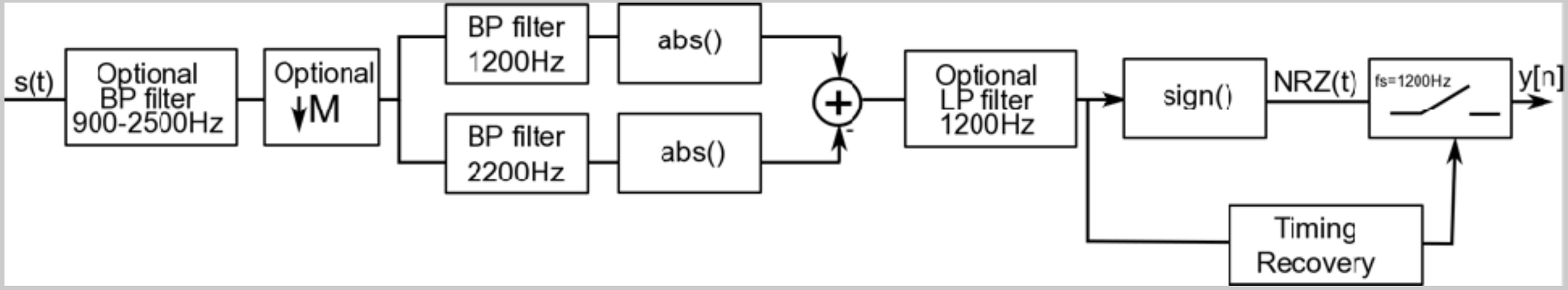


AFSK Demodulation

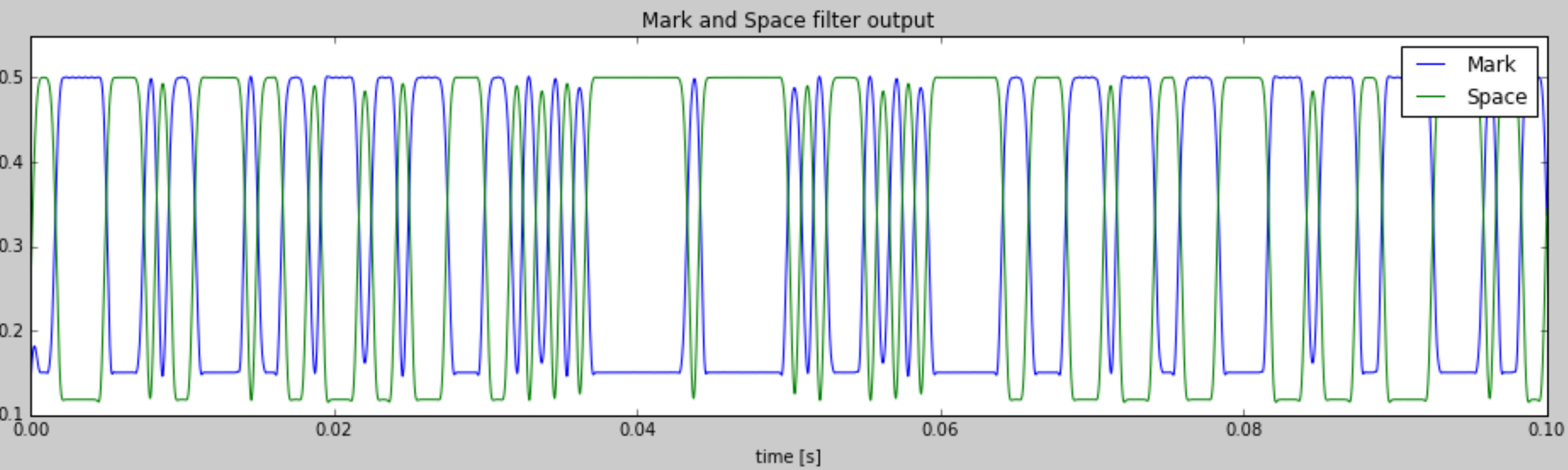
- While afsk is also digital FM -- we will use a non-coherent demodulator
- Based on article by Sivan Toledo (4x6IZ) and DireWolf aprs package.



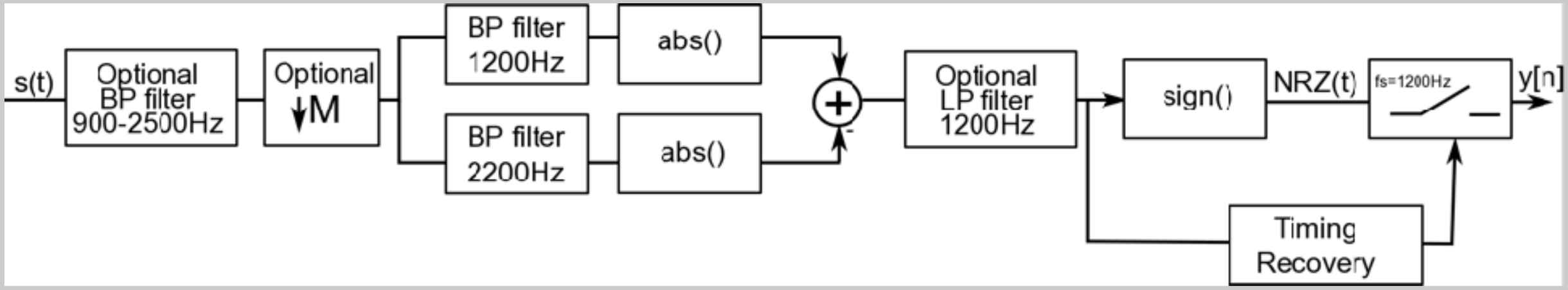
Matched filter detection

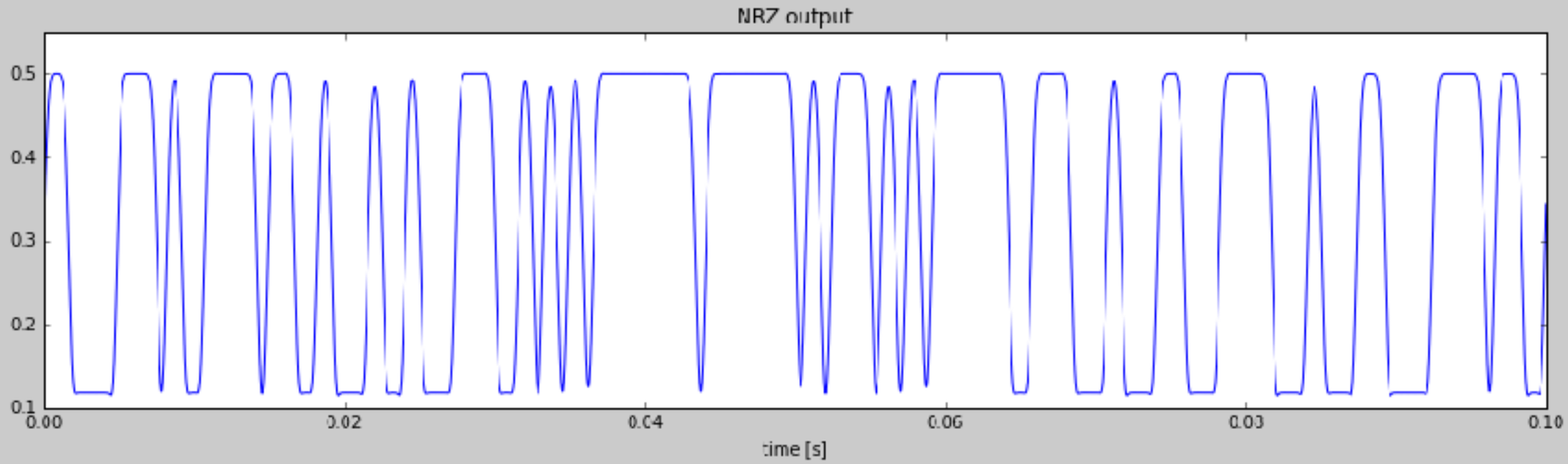


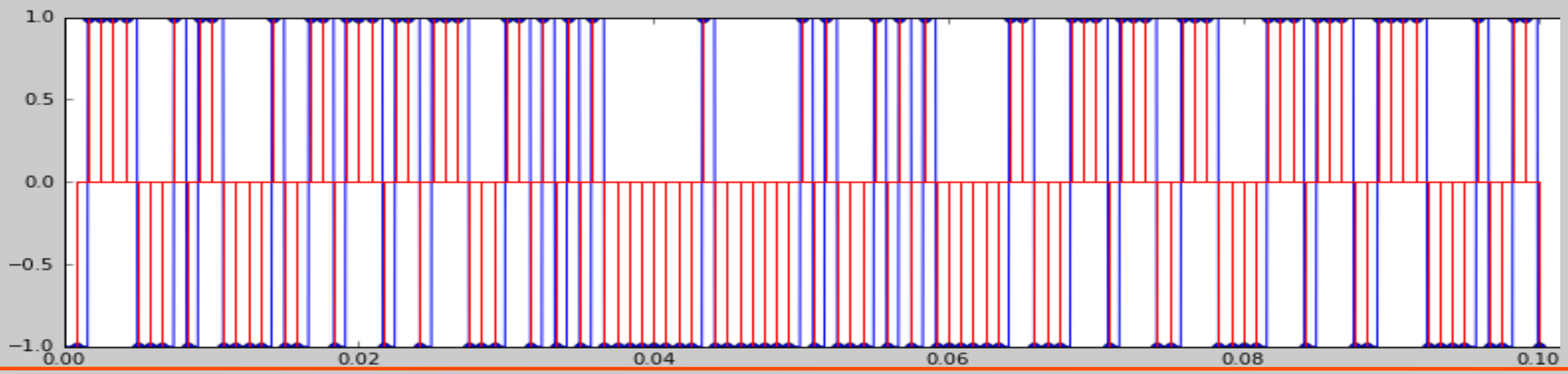
- Design narrow low pass filters around 1200Hz and 2200Hz, compute envelope



Recover NRZ

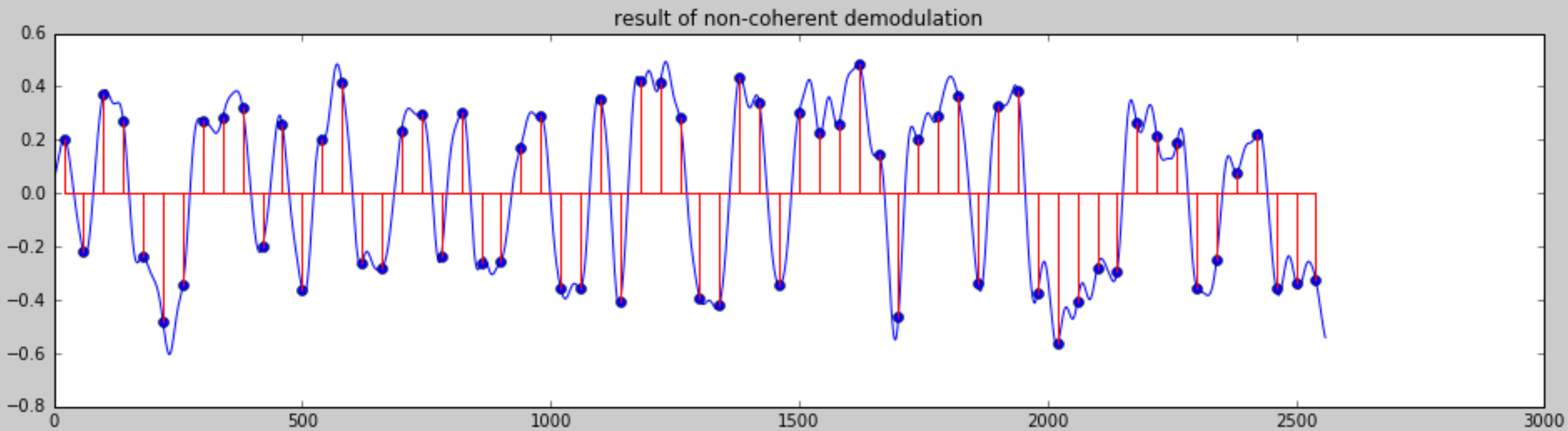

s(t)
Optional BP filter 900-2500Hz
Optional ↓M
BP filter 1200Hz
BP filter 2200Hz
abs()
abs()
+
-
Optional LP filter 1200Hz
sign()
NRZ(t)
fs=1200Hz
y[n]
Timing Recovery


NRZ output
0.5
0.4
0.3
0.2
0.1
0.00
0.02
0.04
0.06
0.08
0.10
time [s]


1.0
0.5
0.0
−0.5
−1.0
0.00
0.02
0.04
0.06
0.08
0.10

Compute Bit Error Rate

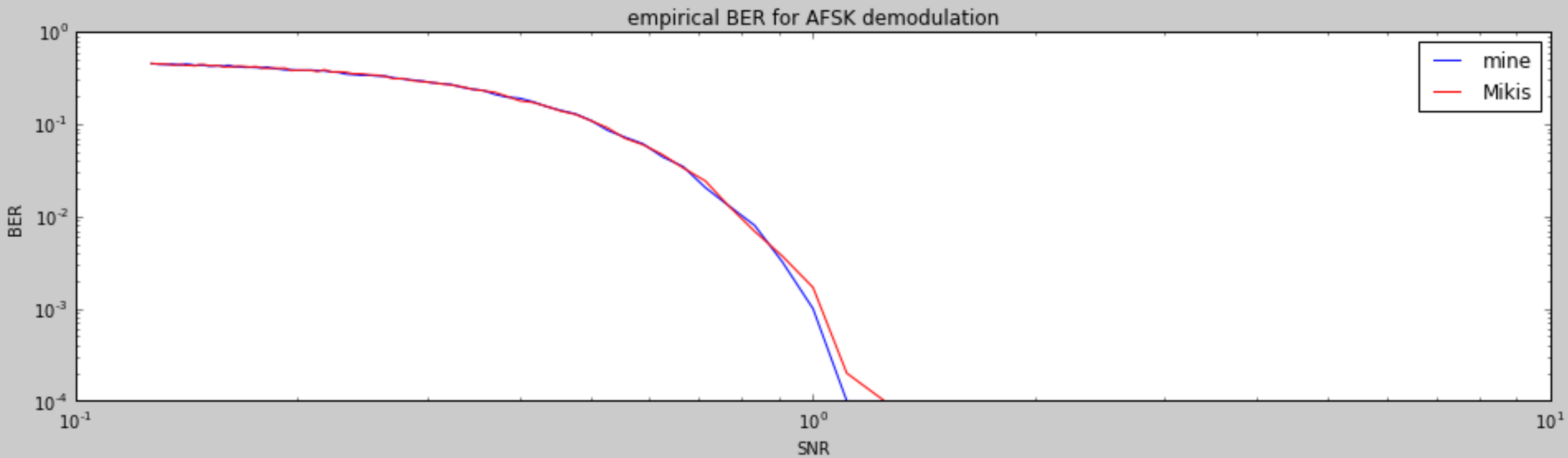
- BER = #altered bits/ Total bits
- Simulate noisy data with gaussian noise
- Run for 10000 long bitstream




```
('BER of non-coherent:', 0.0012)
```

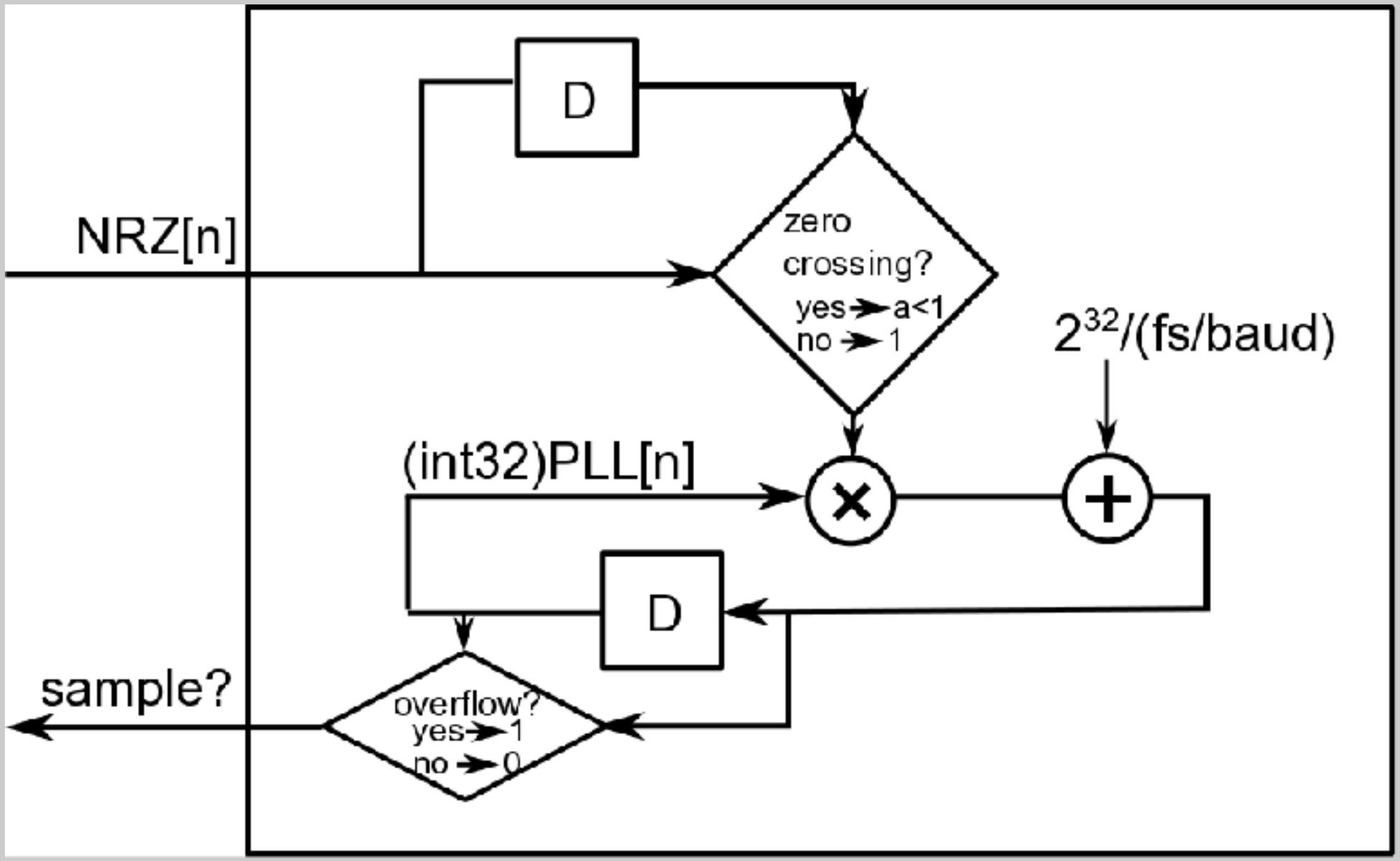
Compute empirical BER curves

- Simulate, and compare to Miki's
- Should be similar performance!

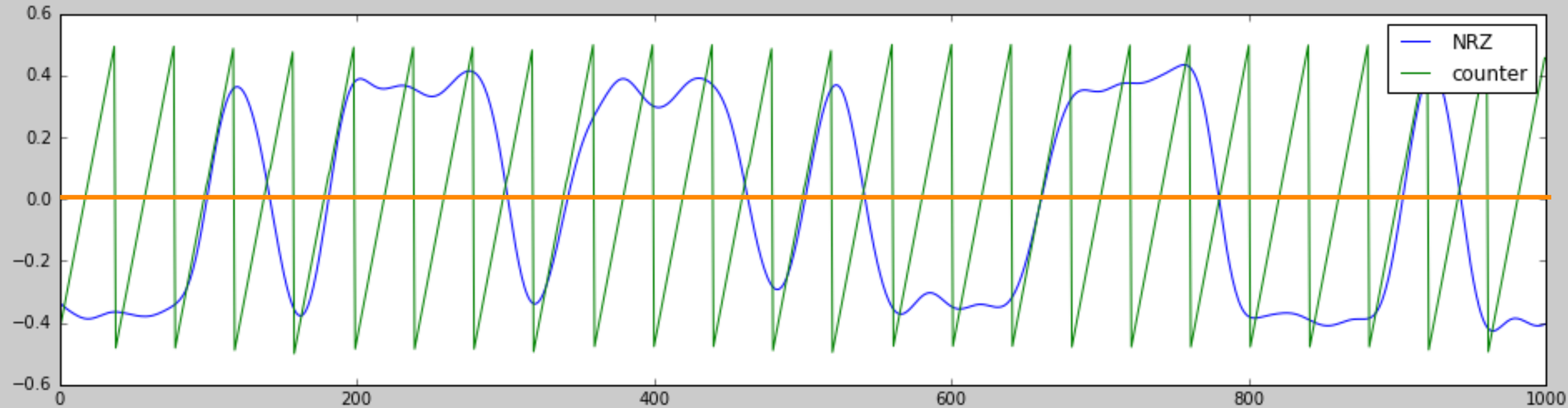


Timing Recovery

- Where to sample???????
- Implement a simple phase-locked loop (PLL)



PLL



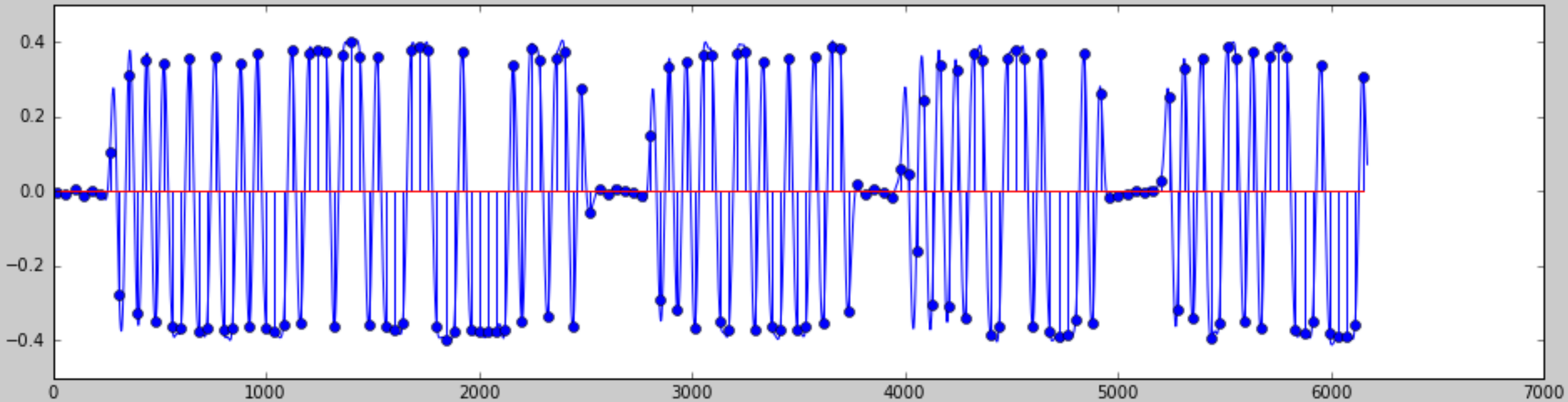
$$c[n] = c[n-1](1 - (\text{sign}(m[n])! = \text{sign}(m[n-1])) * 0.25) + \Delta c$$

- minimizes |C[n] - m[n]|
  - for zero-crossings of m[n]
- C[n] “nudges” is zero-crossing not aligned
- First order non-linear difference eqn.

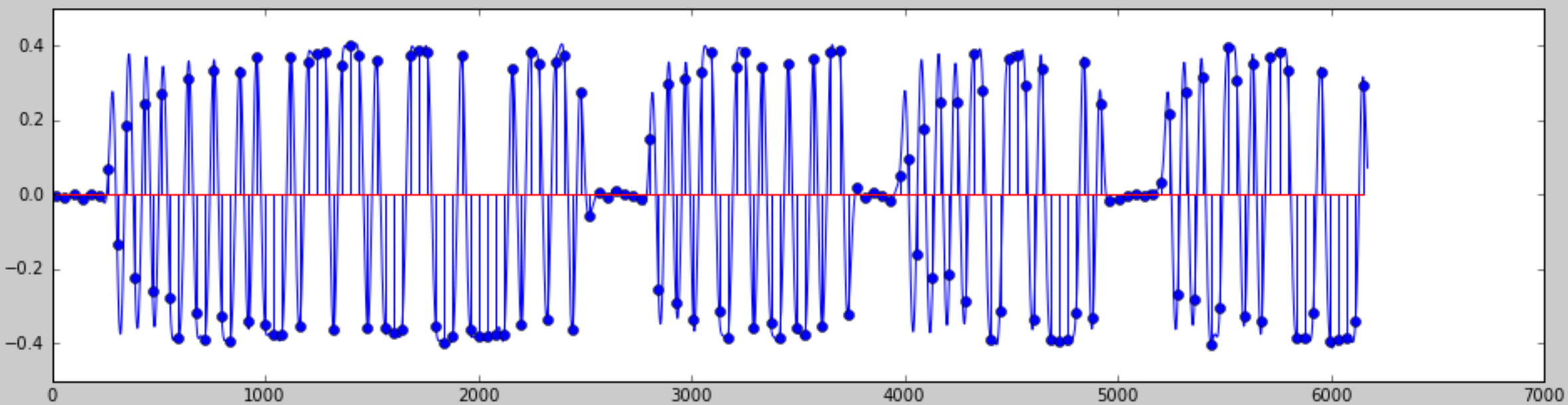
$$c[n] = 0.75c[n-1] + \Delta c \qquad H(z) = \frac{\Delta c}{1 - 0.75z^{-1}}$$

PLL Lock

$a = 0.75$

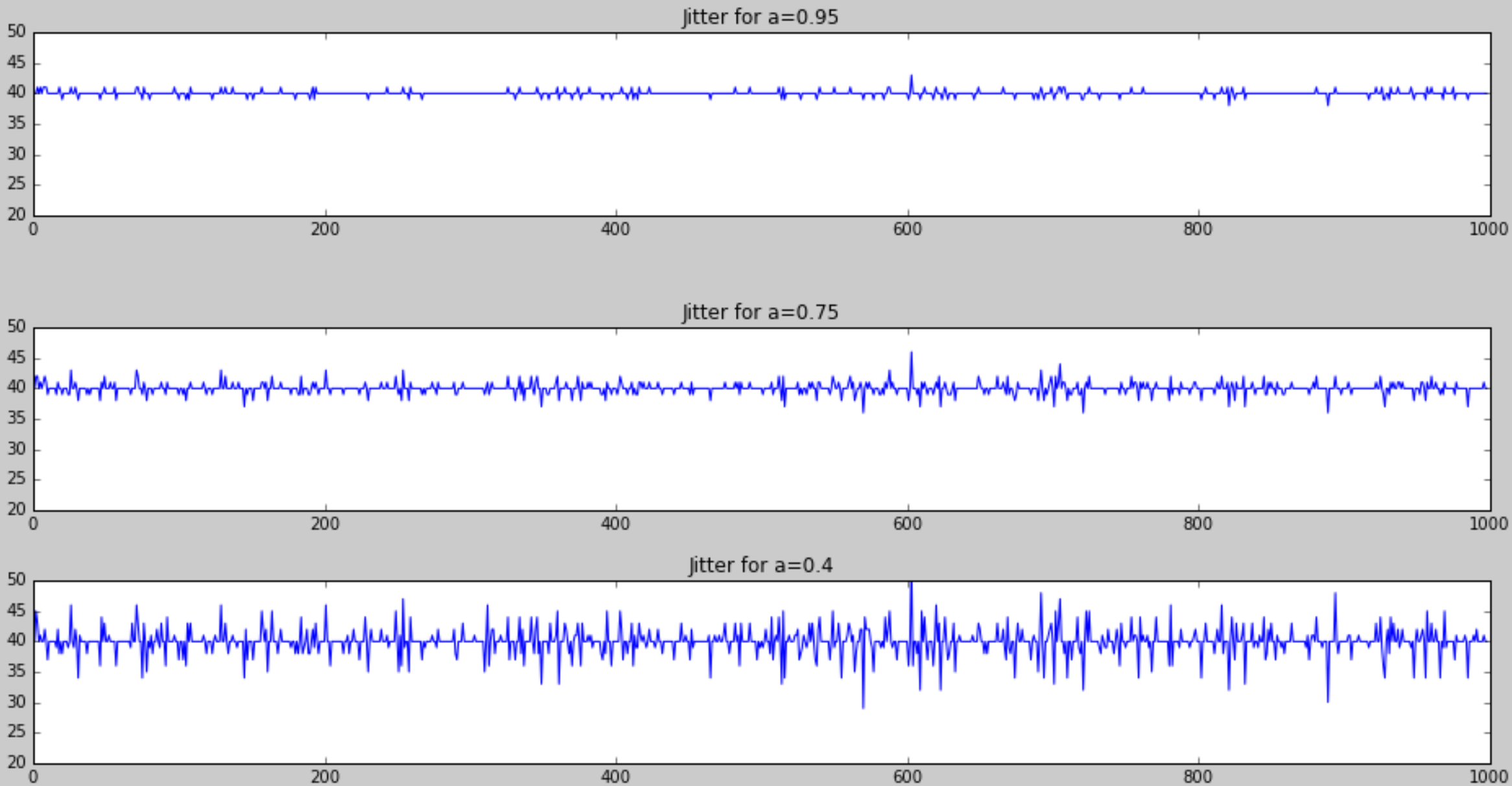


$a = 0.9$



PLL Jitter

- Estimate timing on noisy data

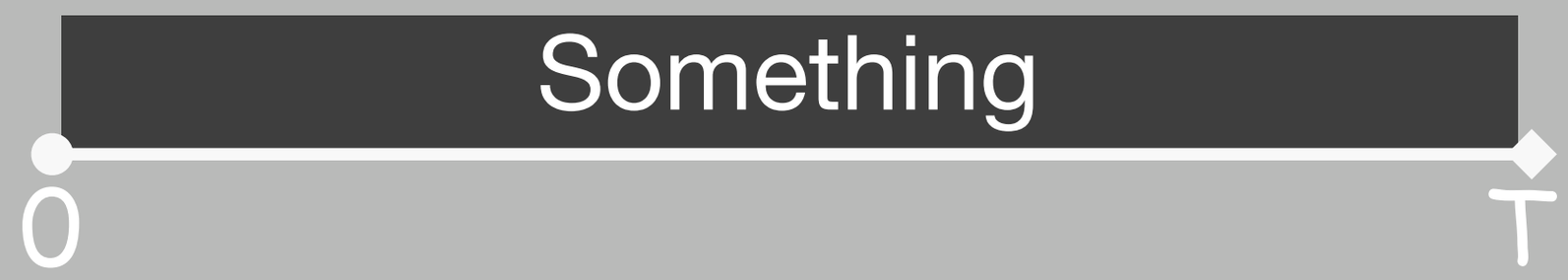


Compressive Sampling

Q: What is the rate you need to sample at?

A: At least Nyquist!

Compressive Sampling



Q: What is the rate you need to sample at?

A: Maybe less than Nyquist....

Image Compression

Images are compressible

Standard approach: First collect, then compress

1001101001101
0001001110101
0100110100010
0010101101010
1010101100101
1101110111010
1010110110110
10100111111

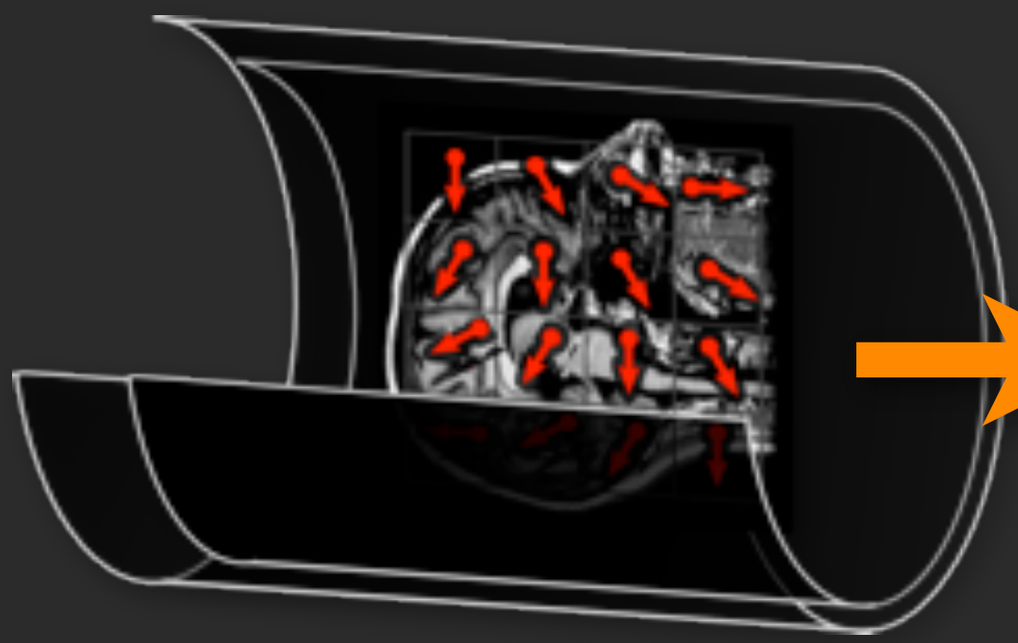
Compressio

101000110100
1101011

Image Compression

Medical images are compressible

Standard approach: First collect, then compress



1001101001101
0001001110101
0100110100010
0010101101010
1010101100101
1101110111010
1010110110110
10100111111

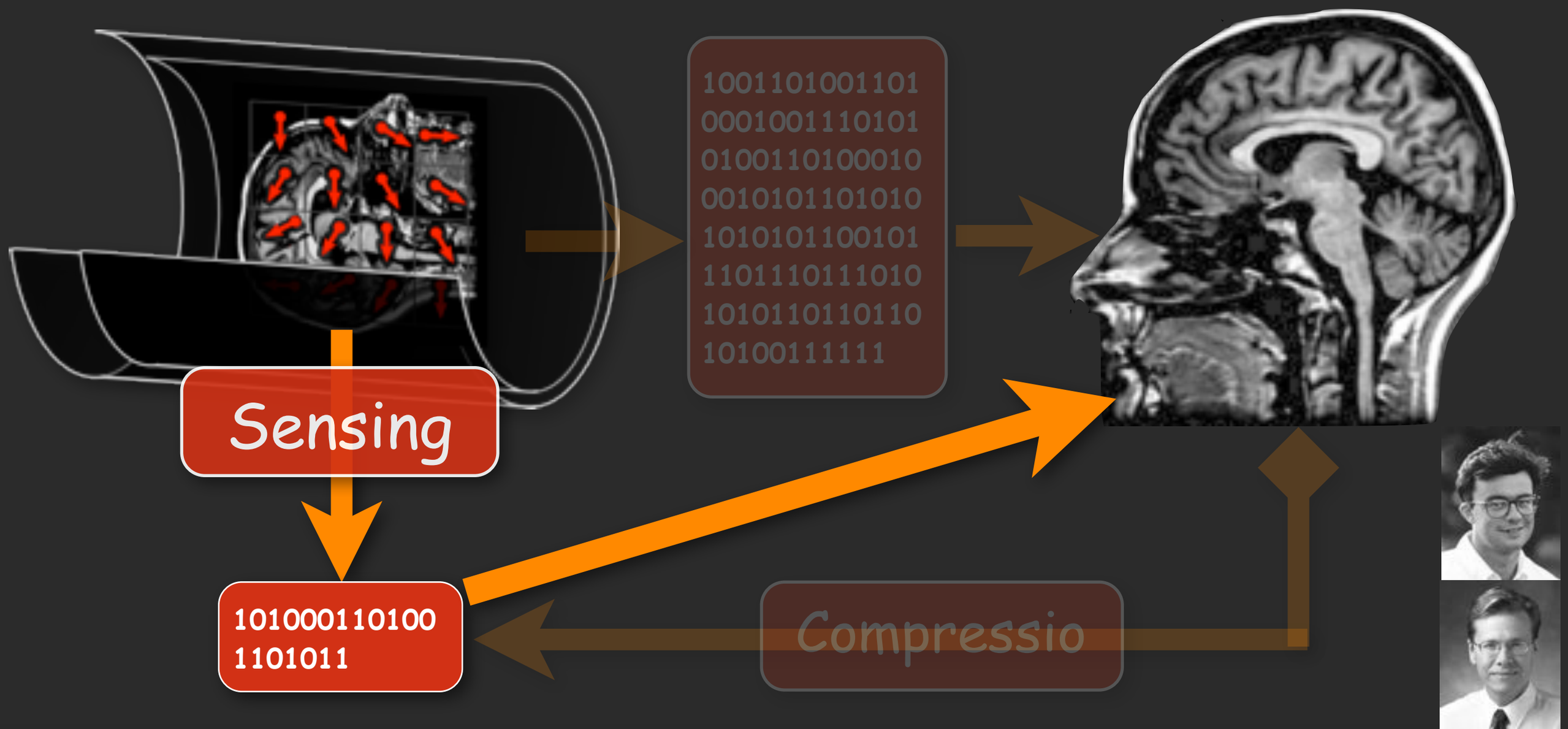
Compressio

101000110100
1101011

Compressed Sensing

Medical images are compressible

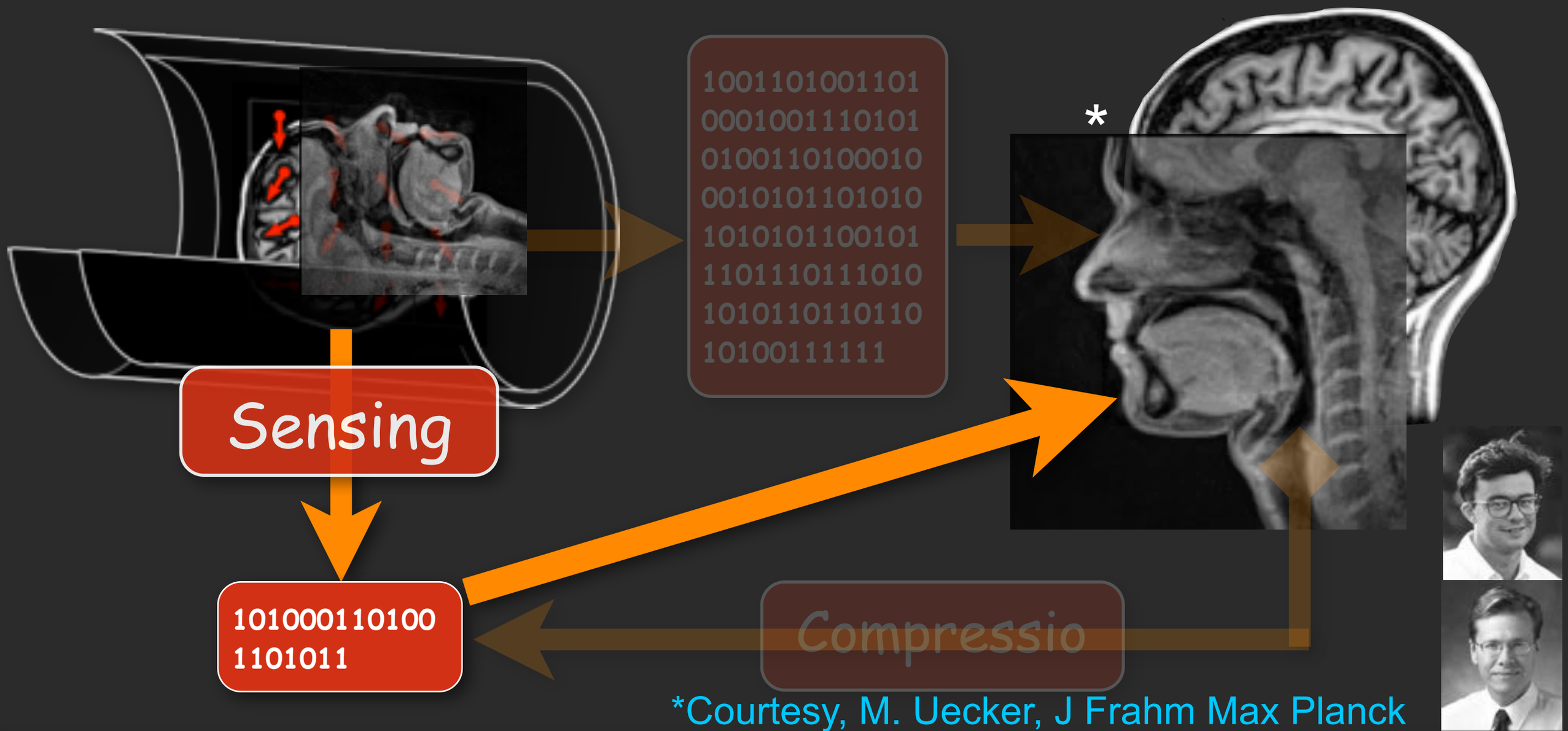
Standard approach: First collect, then compress



Compressed Sensing

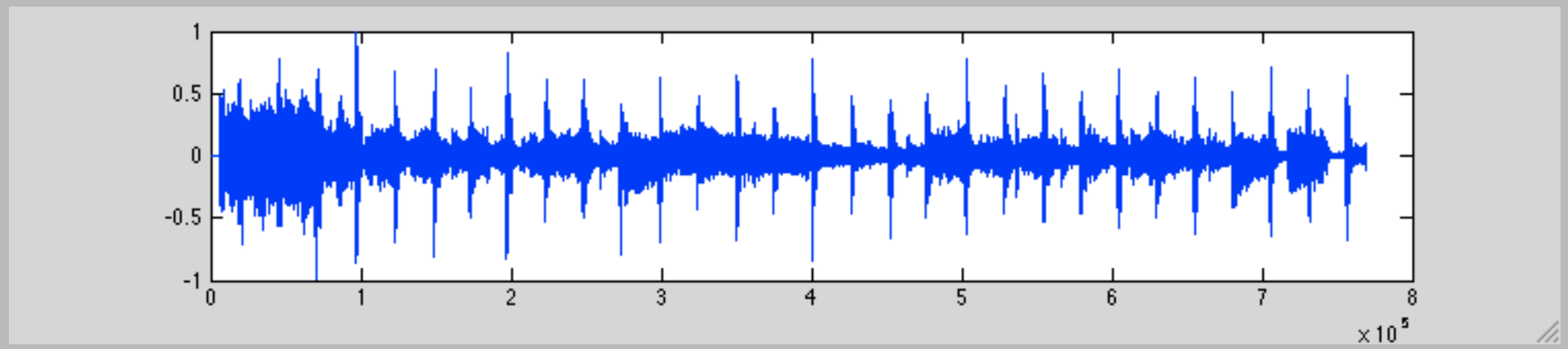
Medical images are compressible

Standard approach: First collect, then compress



*Courtesy, M. Uecker, J Frahm Max Planck

Example I: Audio



Raw audio: 44.1Khz, 16bit, stereo = 1378 Kbit/sec

MP3: 44.1Khz, 16bit, stereo = 128 Kbit/sec

10.76 fold!

Example II: Images

Raw image (RGB): 24 bit/pixel

JPEG : 1280x960, normal = 1.09 bit/pixel

22 fold!

Example III: Videos

Raw Video: (480x360)p x 24b/p x 24fps + 44.1Khz x 16b x 2 = 98,578 Kb/s

MPEG4 : 1300 Kb/s

75 fold!

Compression

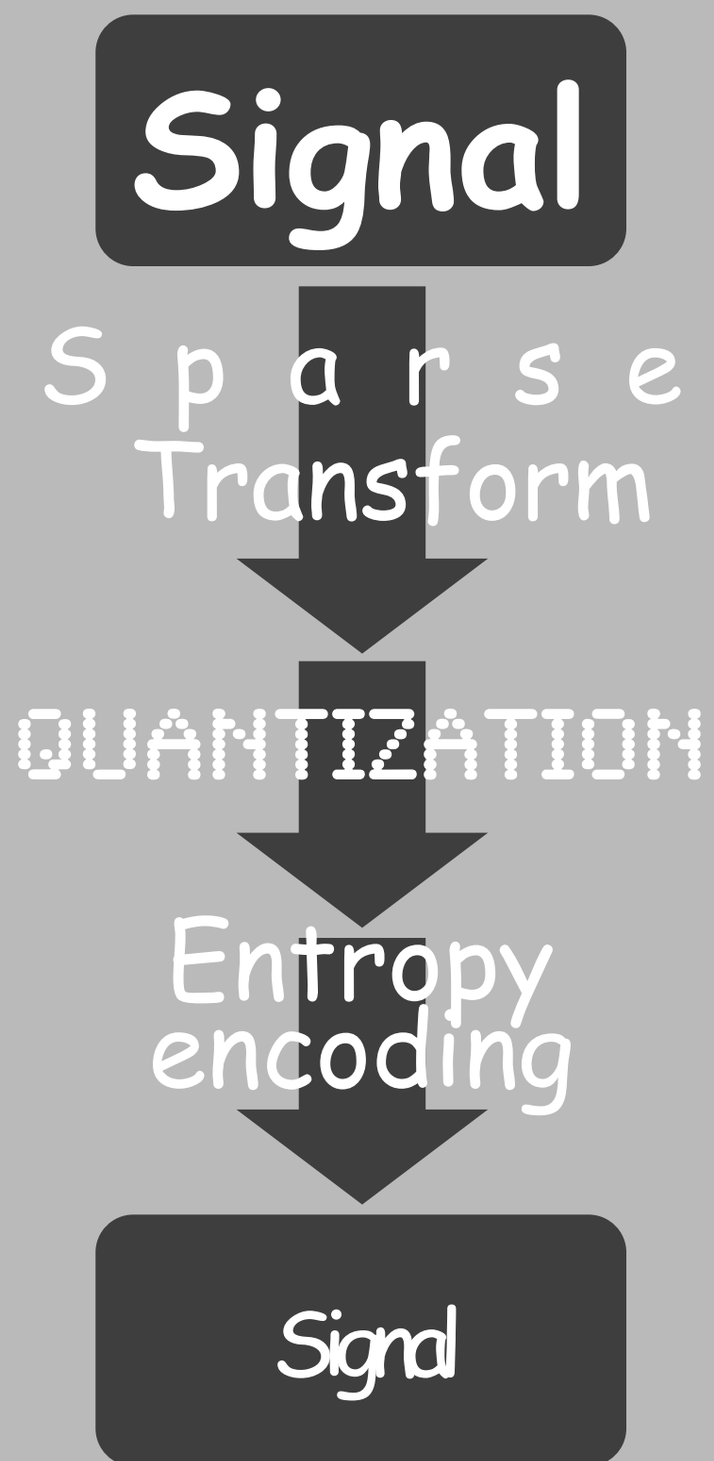
Almost all compression algorithm use transform coding

mp3: DCT

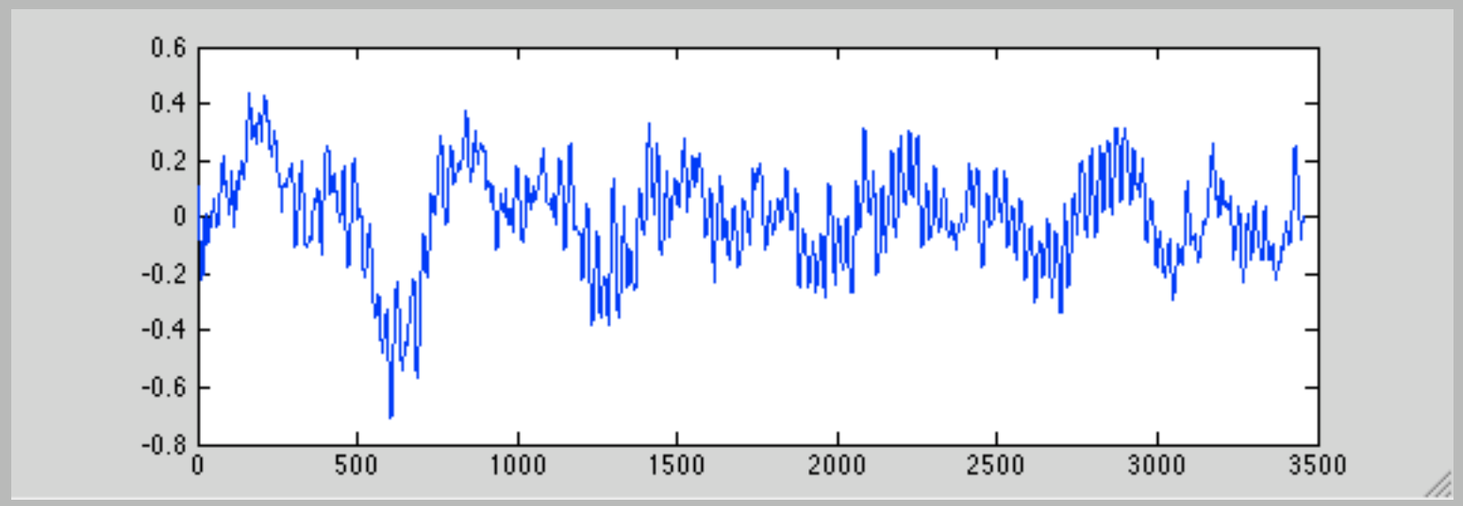
JPEG: DCT

JPEG2000: Wavelet

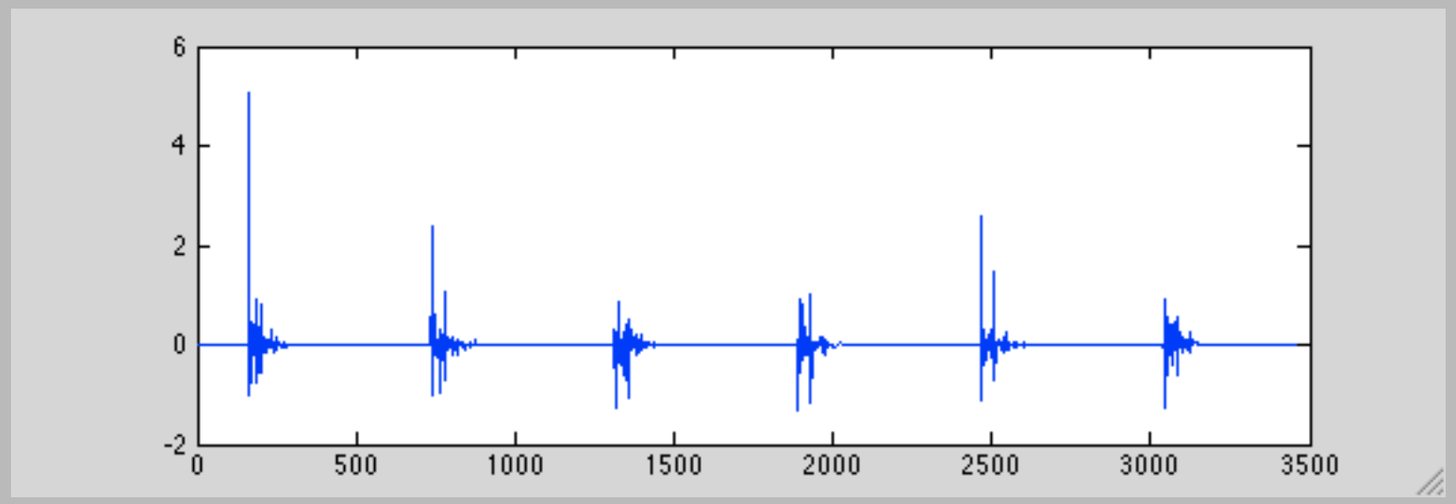
MPEG: DCT & time-difference



Sparse Transform

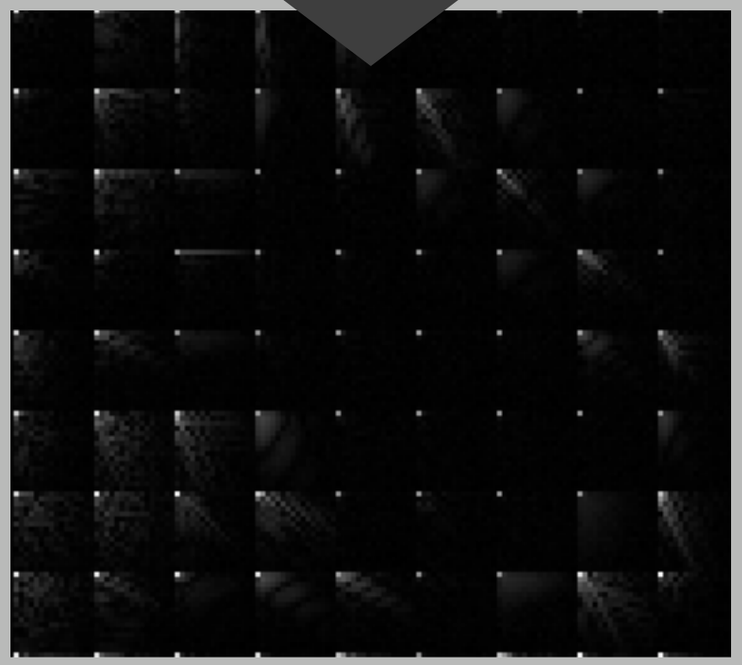


DCT

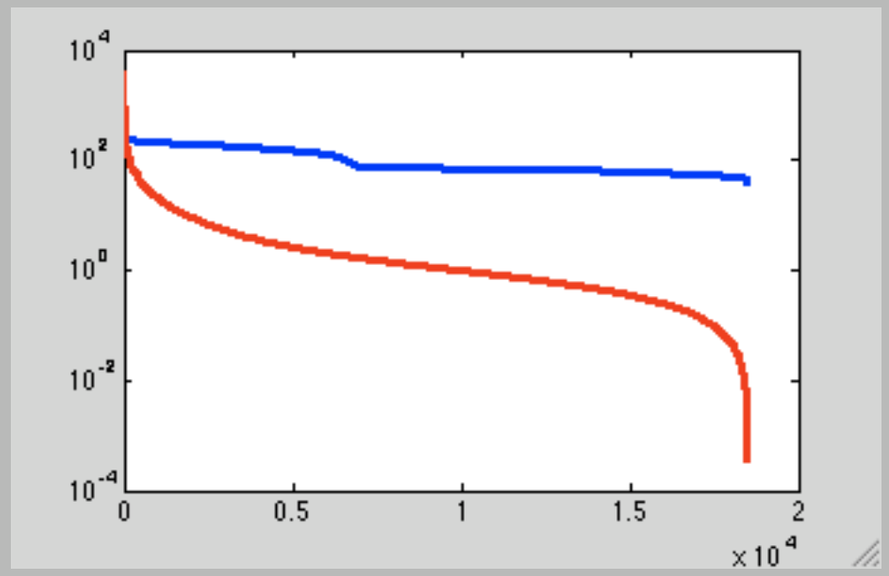


Sparse Transform

DCT



sorted coefficients



Signal

Sparse Transform

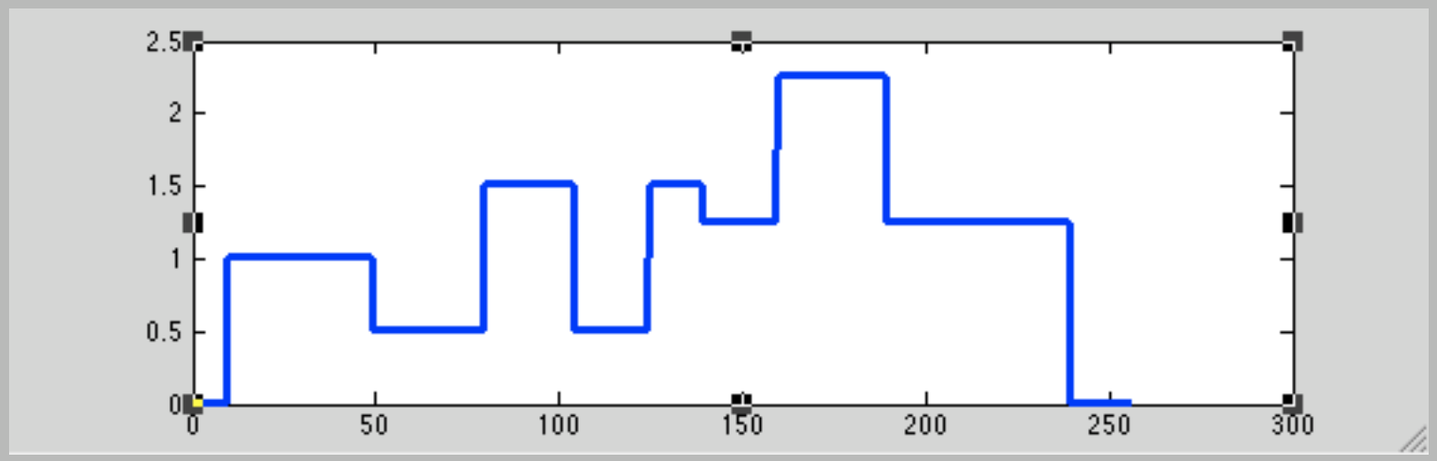
QUANTIZATION

Entropy encoding

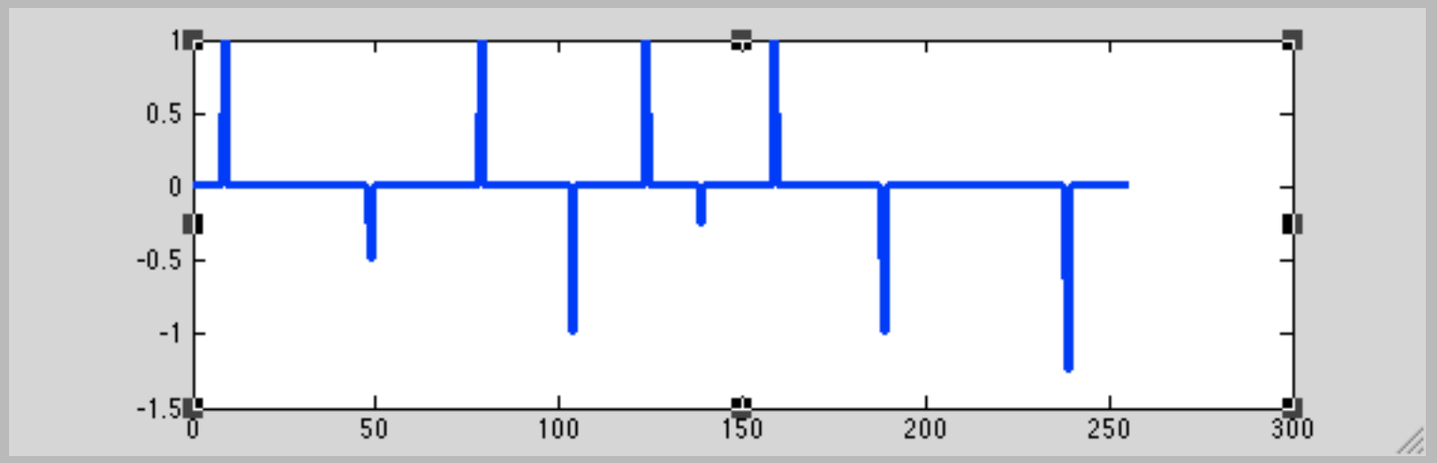
Signal

Sparse Transform

What sparsifying transform would you use here?



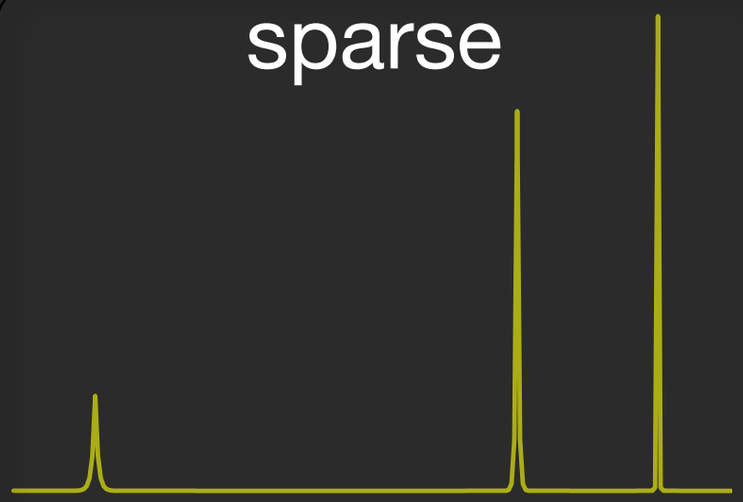
Difference



S p a r s i t y
&
Compressibility

Sparsity and Noise

sparse



not sparse

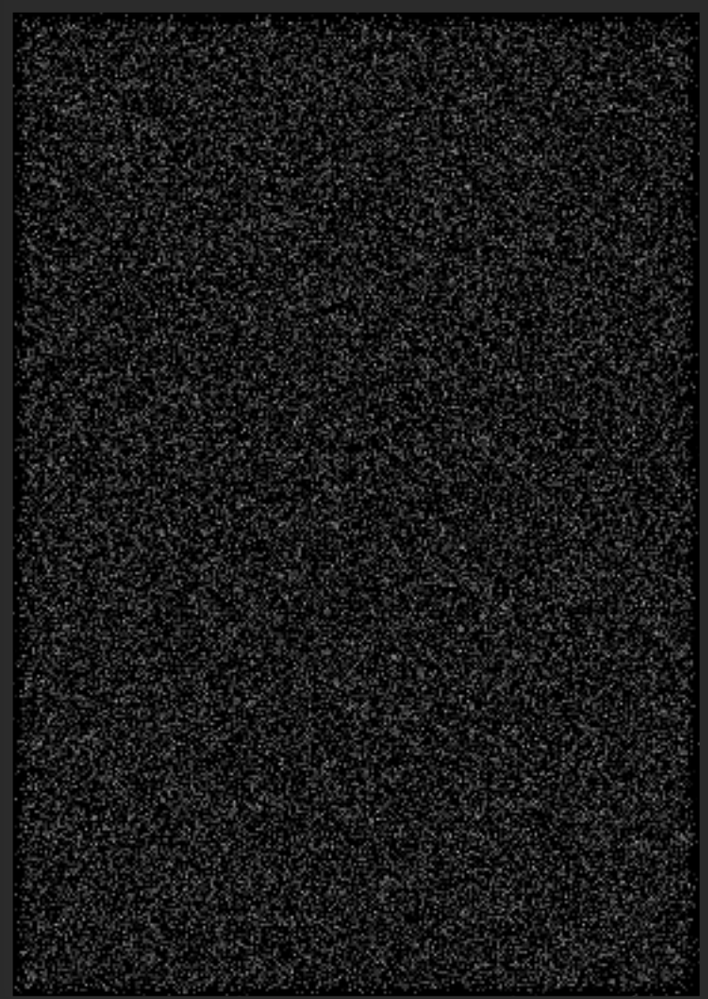


*image courtesy of J. Trzasko

Sparsity and Noise

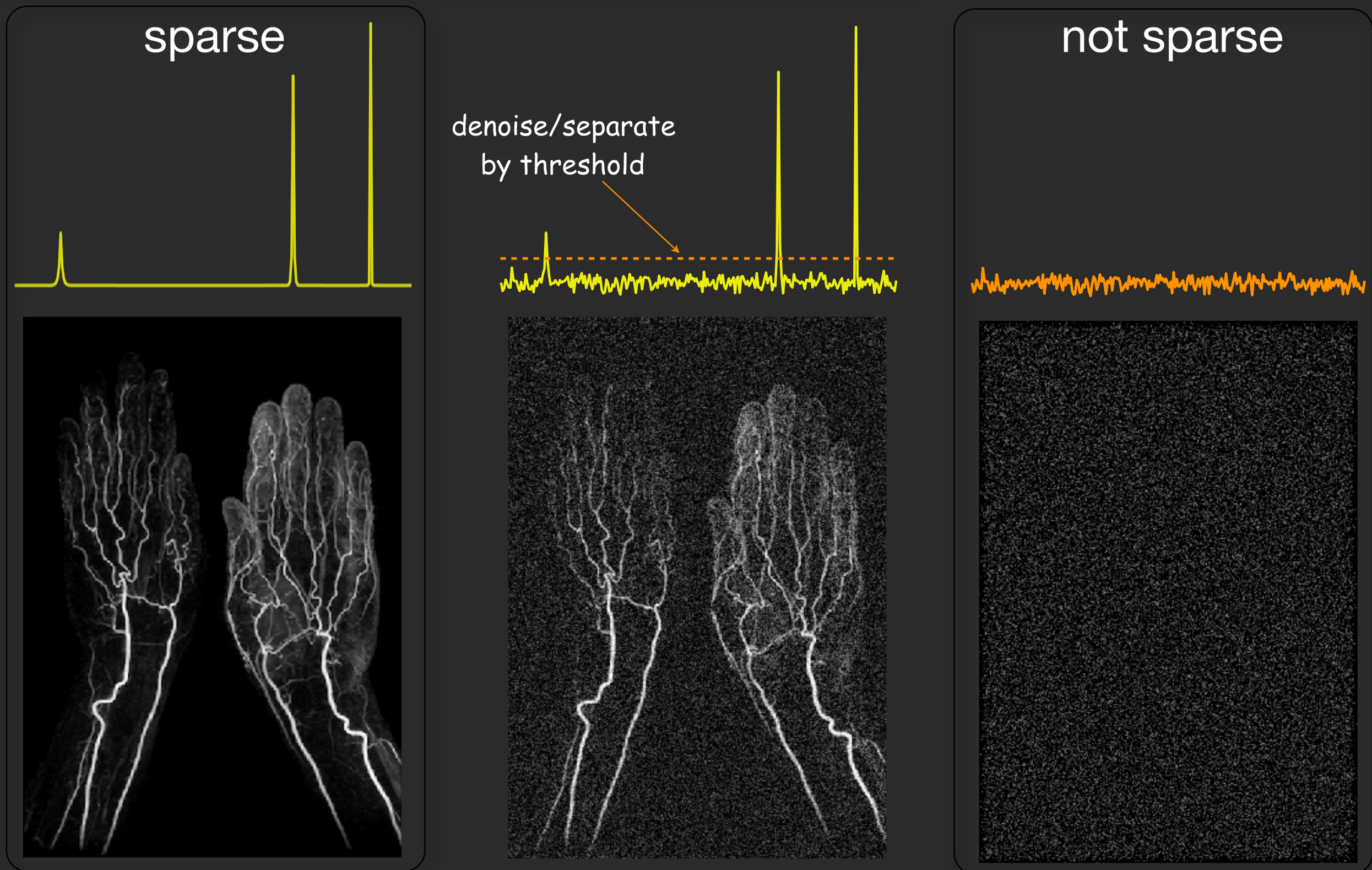
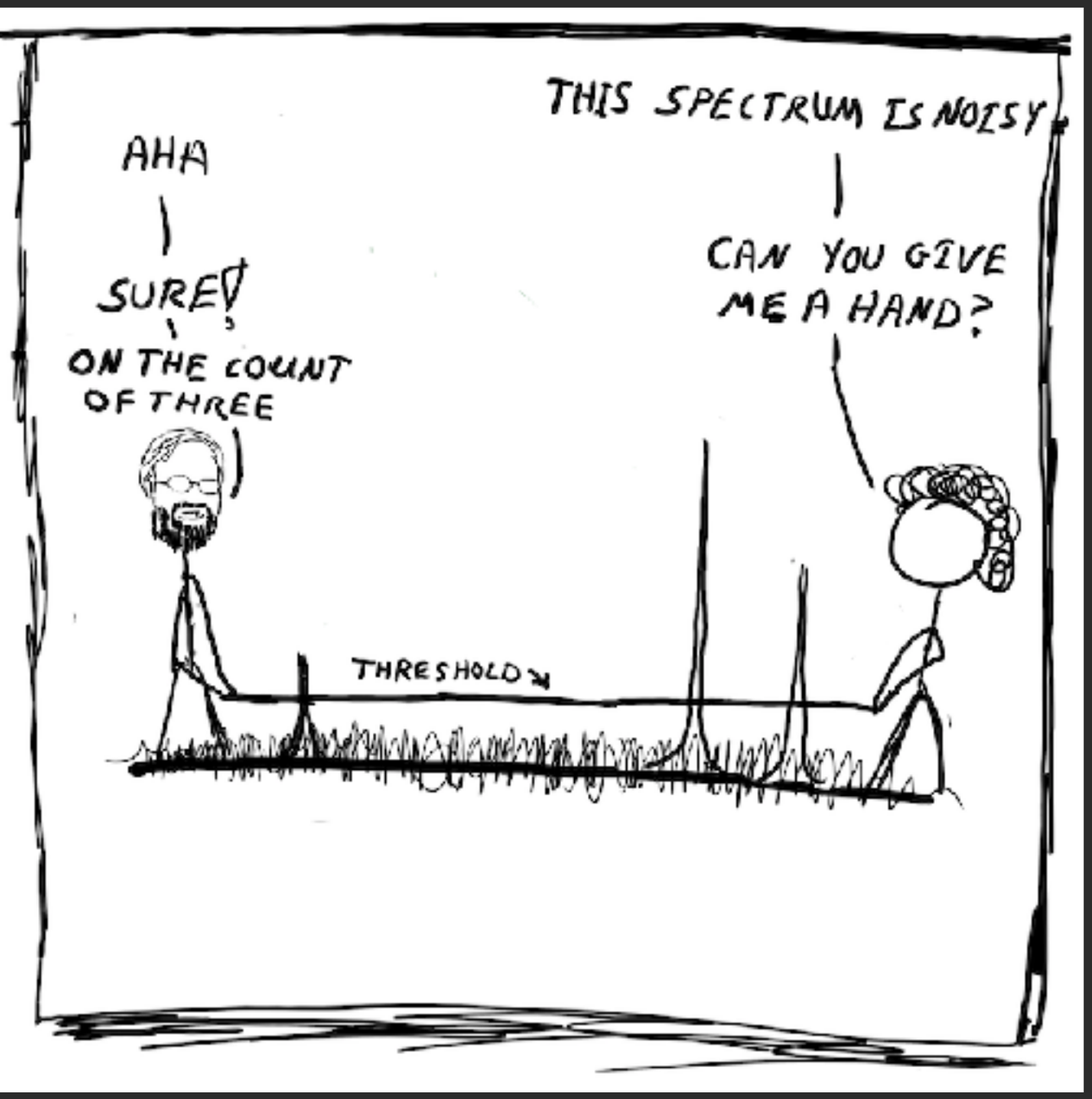
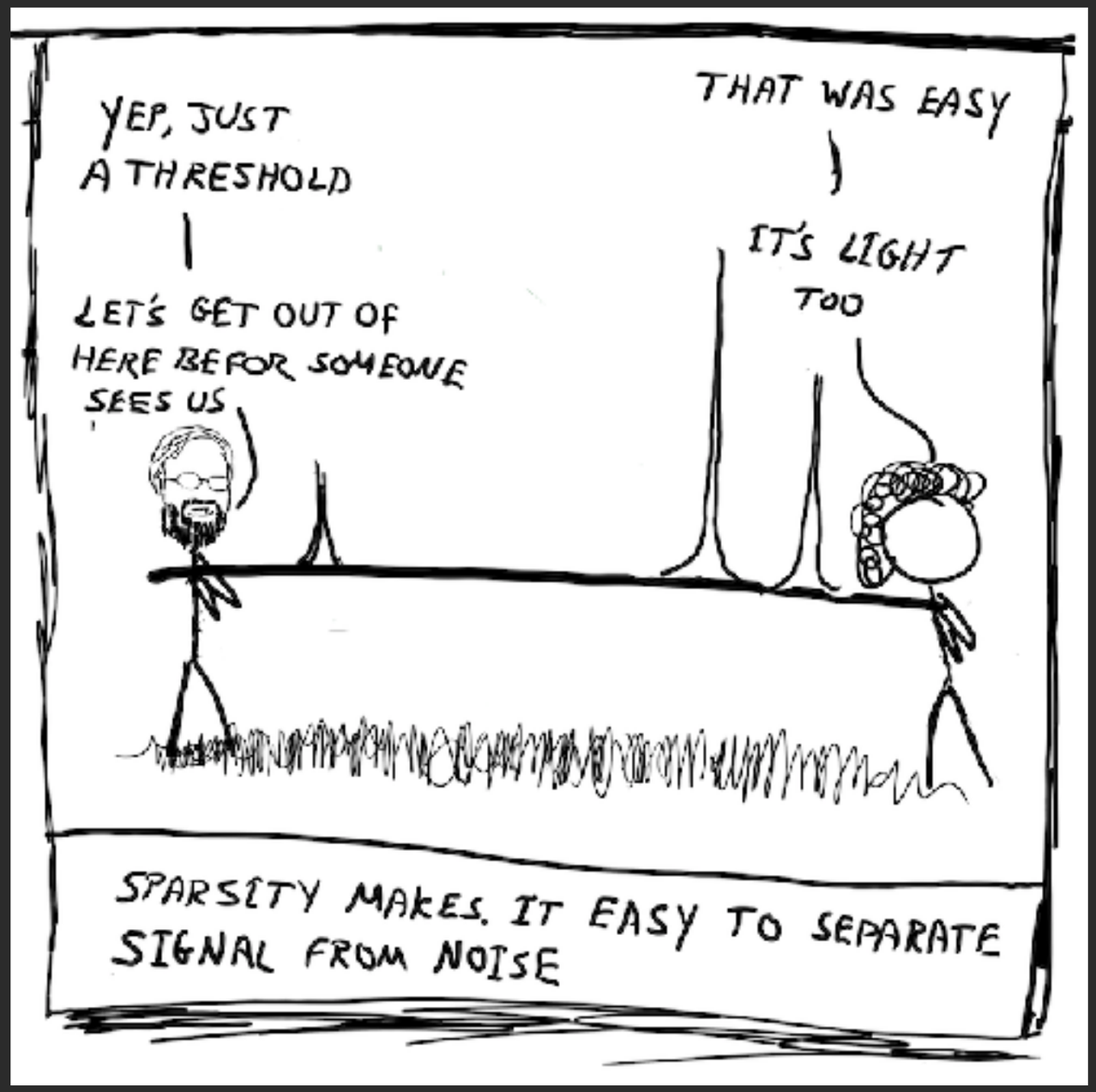


*image courtesy of J. Trzasko

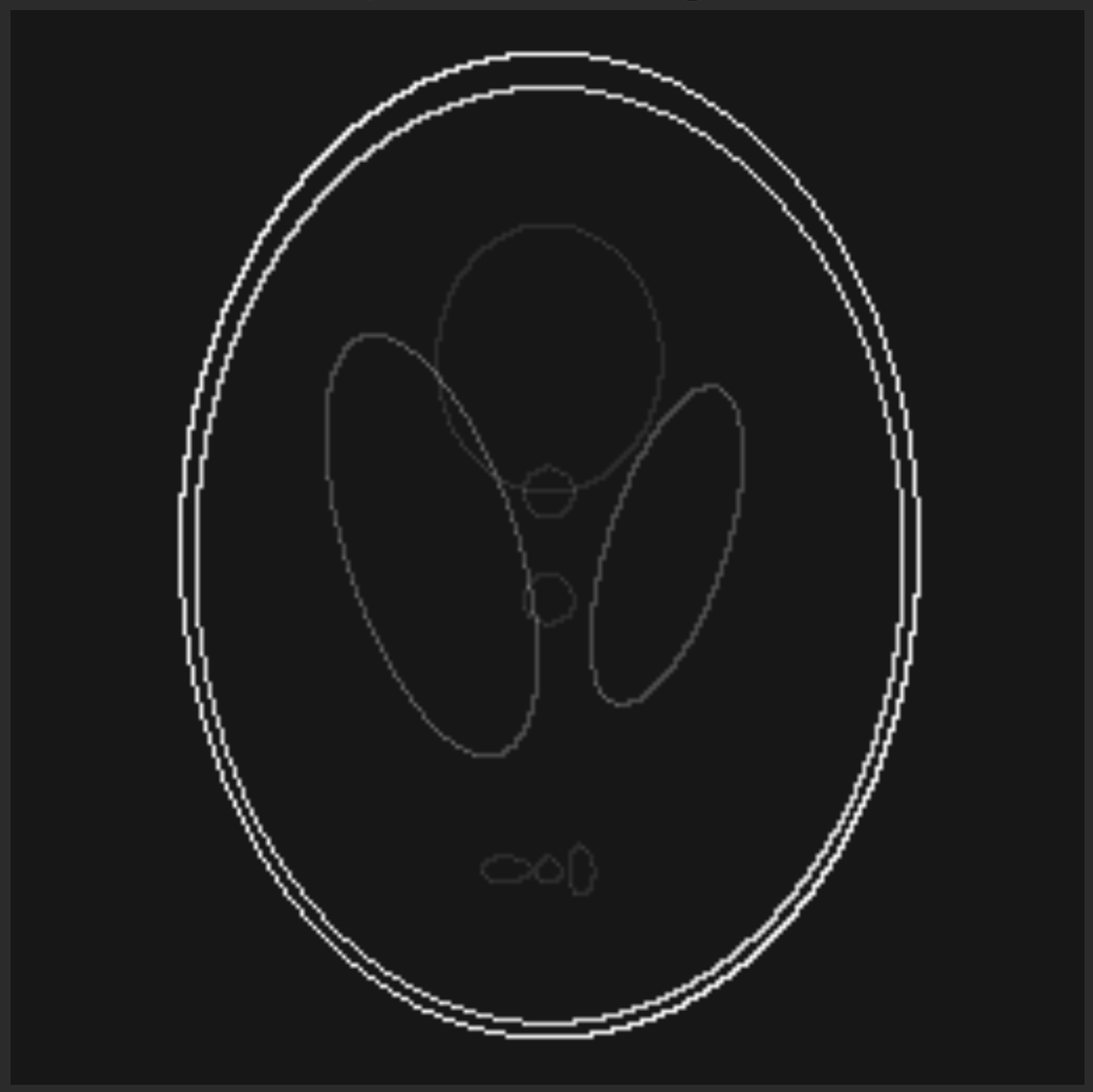

AHA
SURE!
ON THE COUNT OF THREE
THRESHOLD
THIS SPECTRUM IS NOISY
CAN YOU GIVE ME A HAND?


YEP, JUST A THRESHOLD
LET'S GET OUT OF HERE BEFOR SOMEONE SEES US
THAT WAS EASY
IT'S LIGHT TOO
SPARSITY MAKES IT EASY TO SEPARATE SIGNAL FROM NOISE

Transform Sparsity

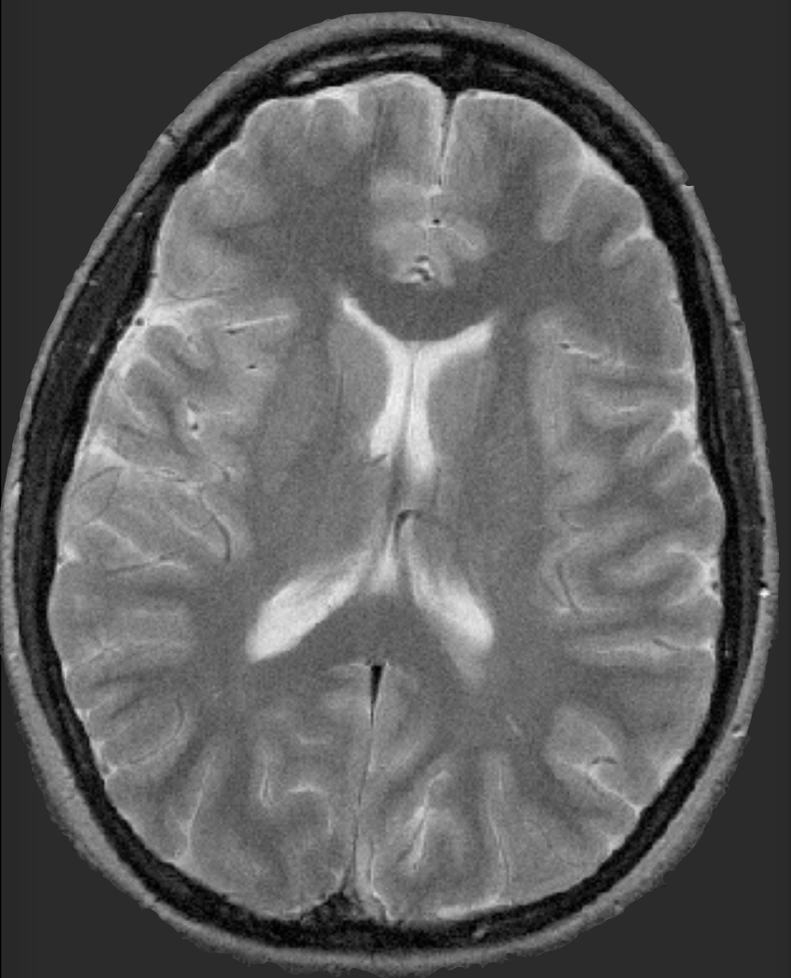
not sparse

Sparse Edges



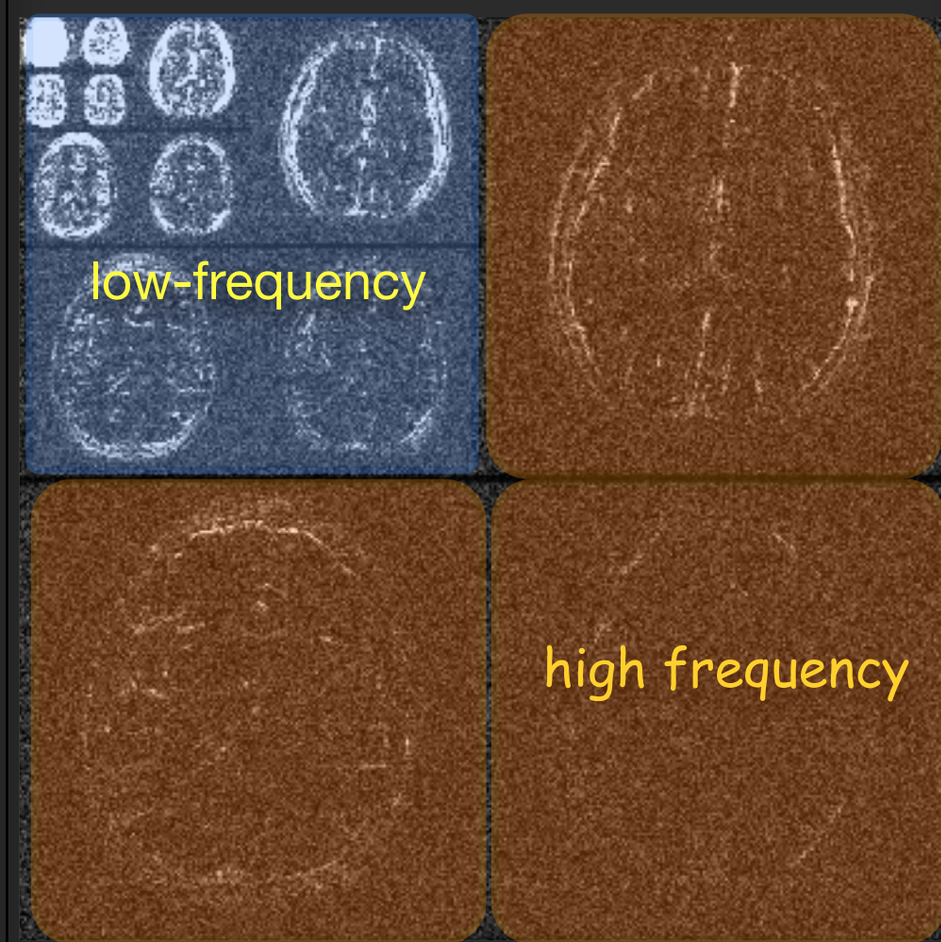
Transform Sparsity and Denoising

not sparse



sparse

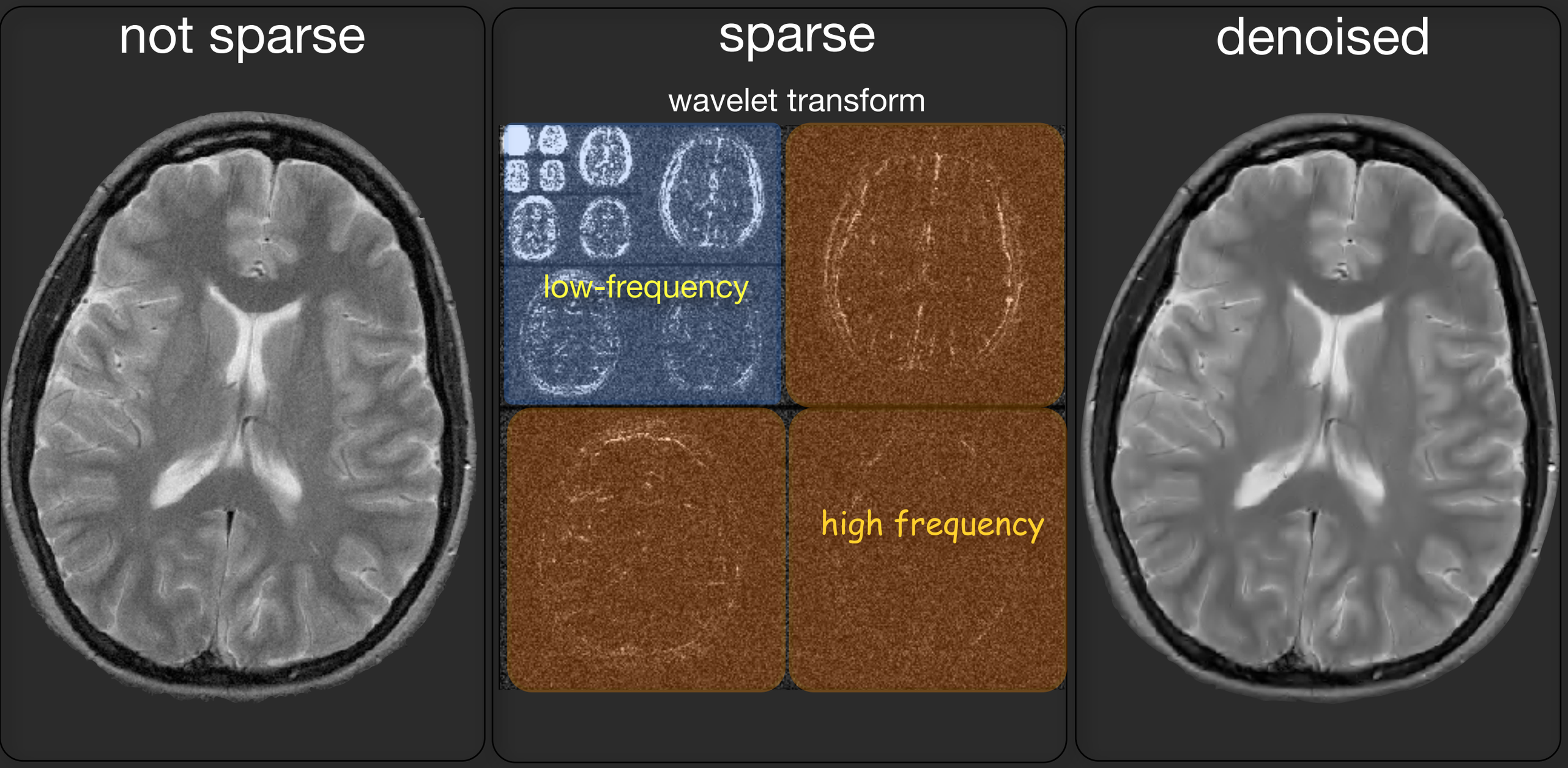
wavelet transform



denoised

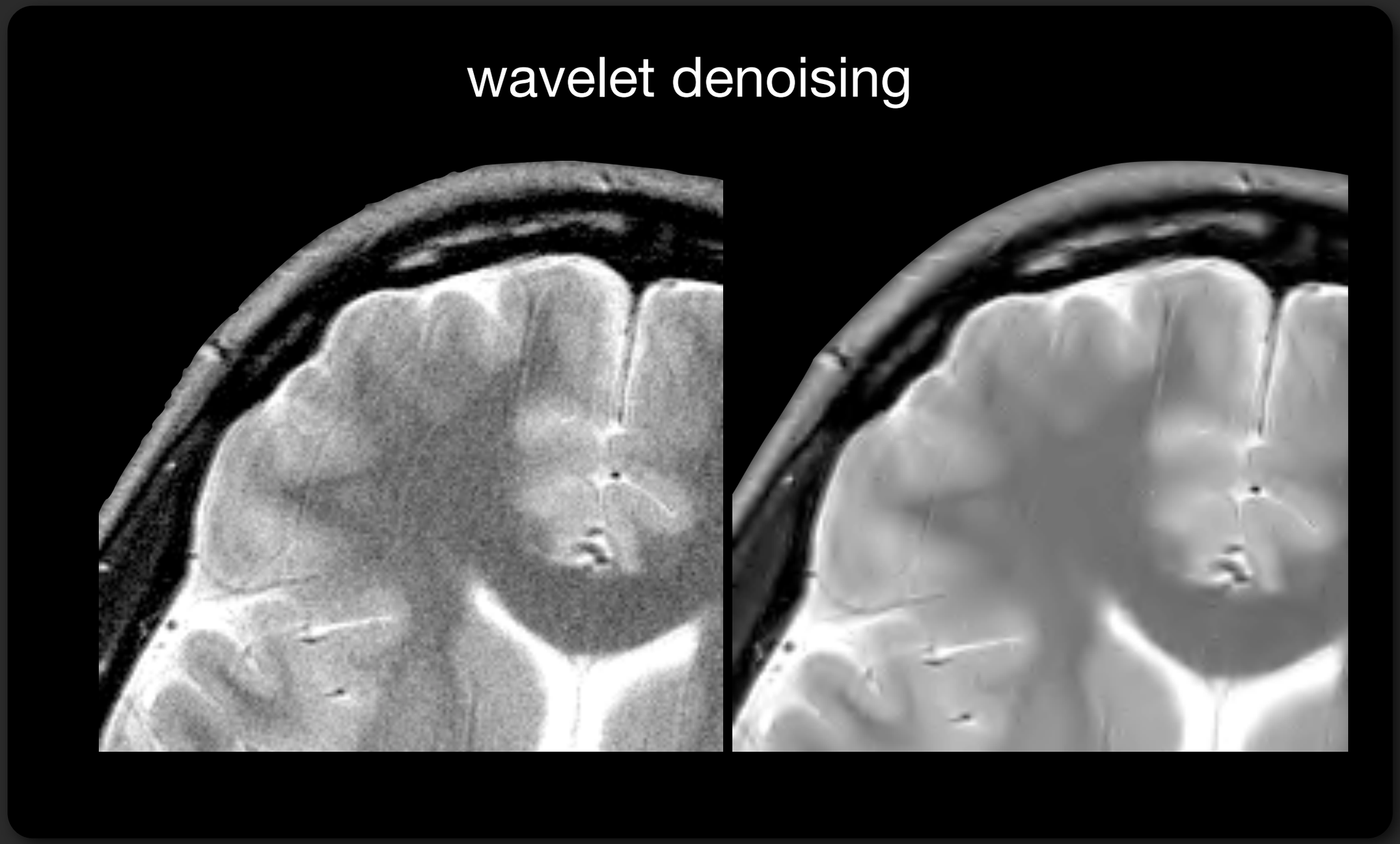
DL Donoho, I Johnstone Biometrika 1994;81(3):425-55

Transform Sparsity and Denoising



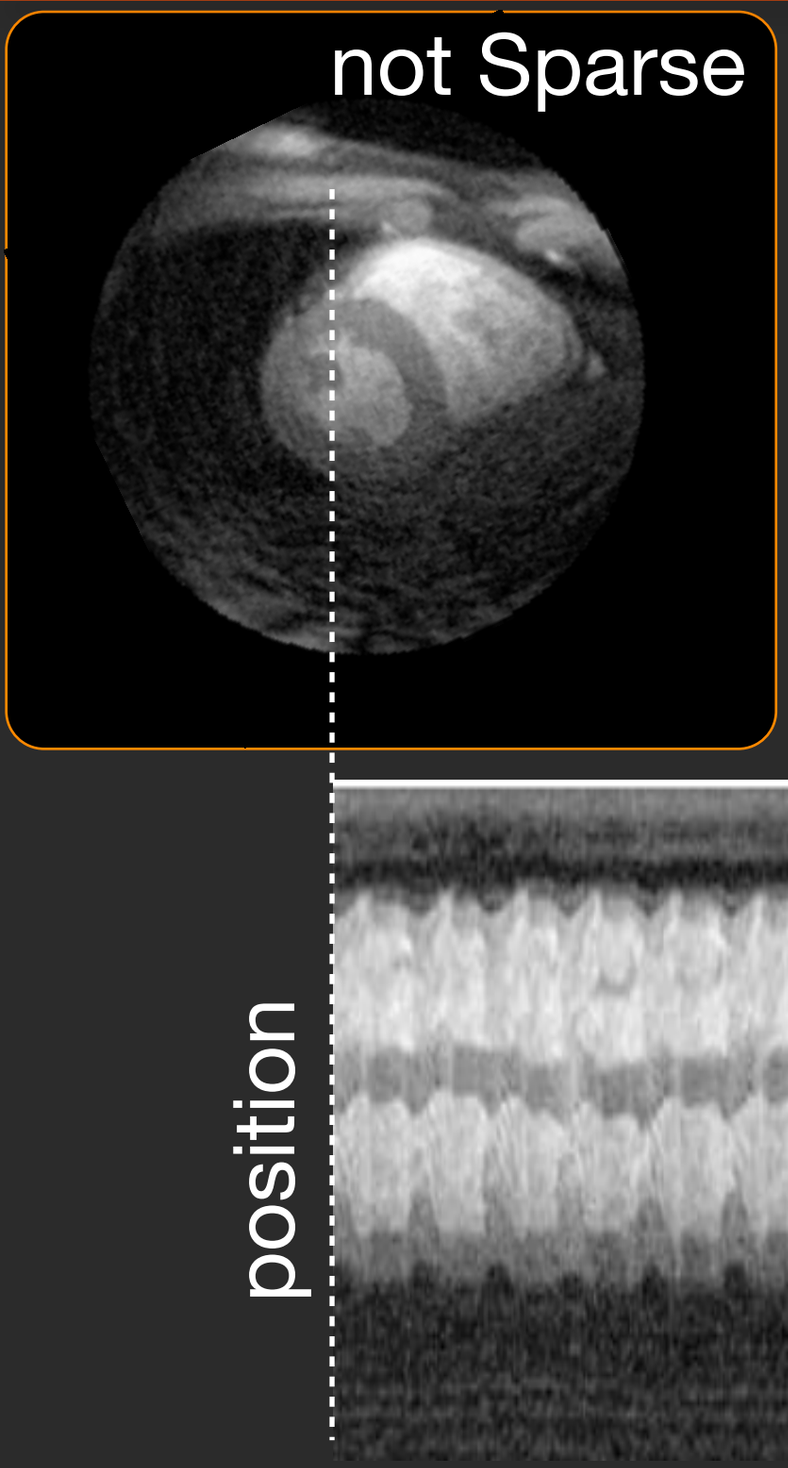
DL Donoho, I Johnstone Biometrika 1994;81(3):425-55

Transform Sparsity and Denoising

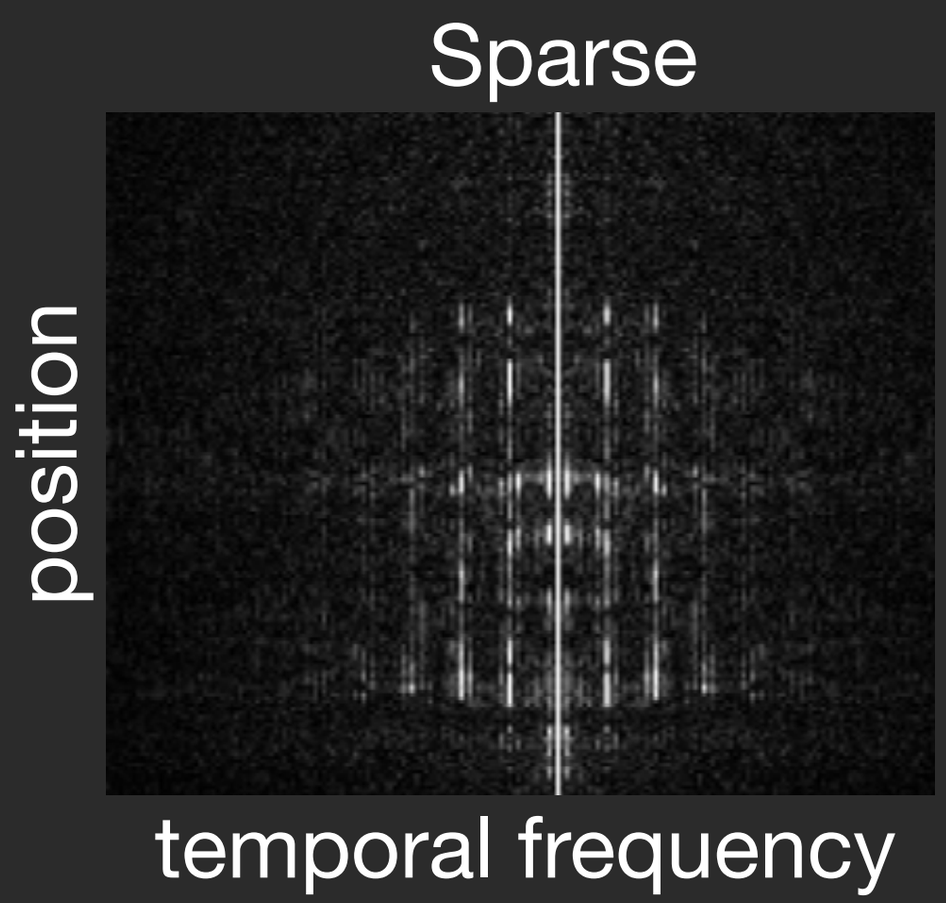


DL Donoho, I Johnstone Biometrika 1994;81(3):425-55

More Sparse Transforms



*Video courtesy of Juan Santos, Heart Vista



Sparsity and Compression

- Only need to store non-zeros