Introduction to Field Analysis Techniques:

The Power Spectrum and Coherence

Tutorial

Mark Kramer
SFN 2013

Outline

- Tutorial: Hands on MATLAB examples
- An introduction . . .
- Power spectrum
  - Frequency resolution
  - Nyquist frequency
  - Tapering
- Coherence
- Please ask questions

Assumptions

<u>MATLAB</u>

- Running on your computer.
- Basic knowledge
  - Loading variables, navigating directories, executing commands, indexing, etc.

Get the data

- Download example data and code:

  http://math.bu.edu/people/mak/sfn-2013/

The Science of Large Data Sets: Spikes, Fields, and Voxels

Society for Neuroscience Short Course #2

Data
Download the data set for power spectrum analysis: d1.mat
Download the data set for coherence analysis: d2.mat
Load this data set in MATLAB using the *load* command.

MATLAB code
Download an M-file that includes MATLAB code to analyze these data: sfn_tutorial.m

Tutorial slides : As a PDF.

Software Links
Chronux
EEGLab

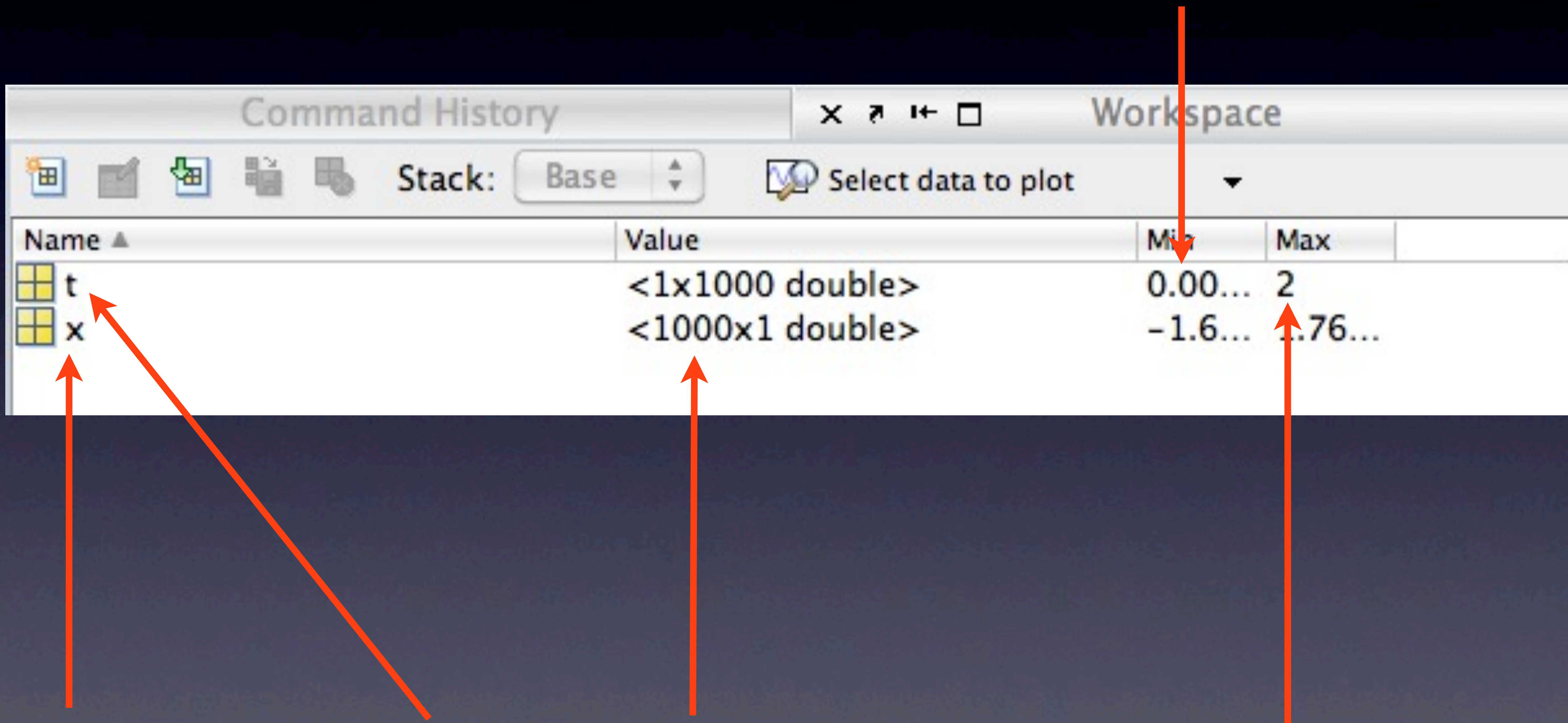
Book Links
Matlab for Neuroscientists
Signal Processing for Neuroscientists
Observed Brain Dynamics

Contact: Email Mark Kramer

Load data

```
>> load d1.mat
```

Start time



Data from sensor

Time

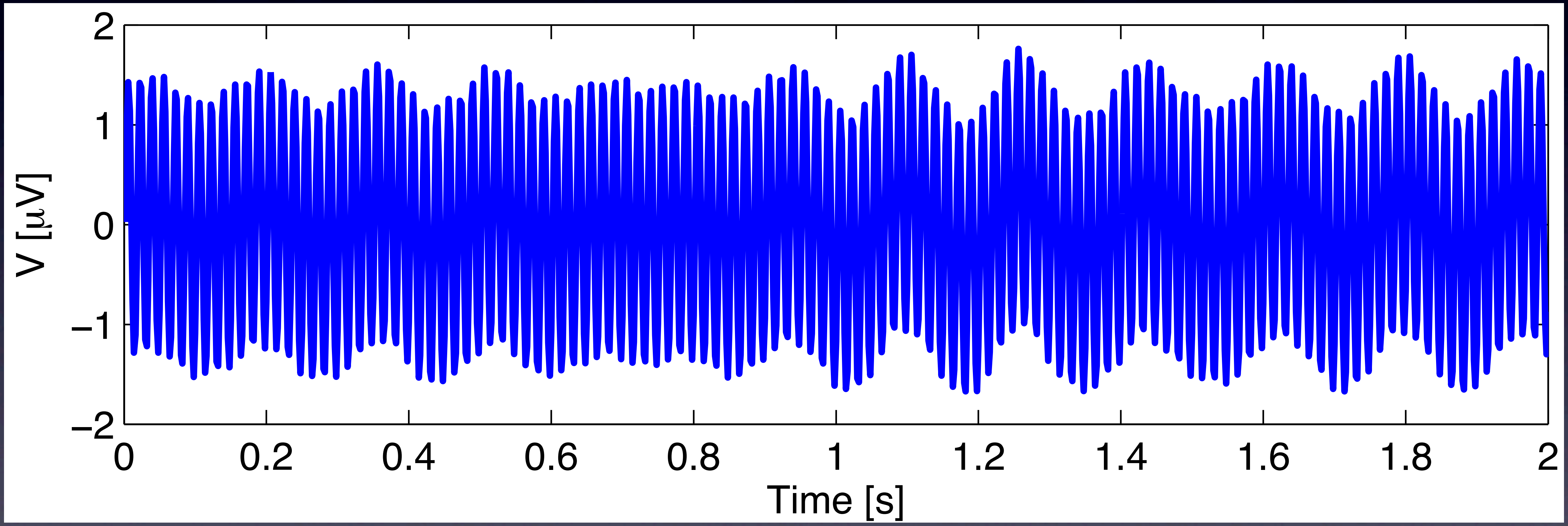
1000 samples

Stop time

Load data & visualize

```
>> load d1.mat
>> plot(t,x)
```

```
>> T=2;
```



Visual inspection:

- Rhythmic
- It's complicated
- How can characterize?

Load data & visualize

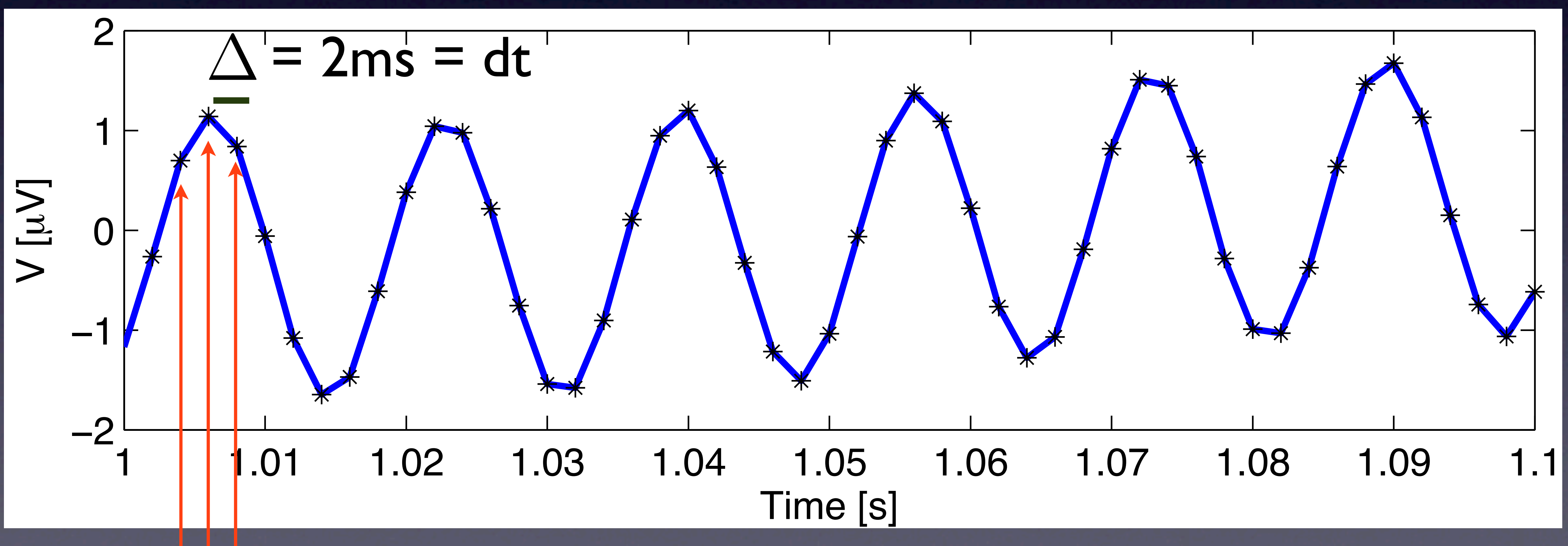
Zoom in ...

```
>> hold on
>> plot(t,x, 'k*')
>> xlim([1 1.1])
```

Hold the graphics window

Plot as black *

Adjust x-axis

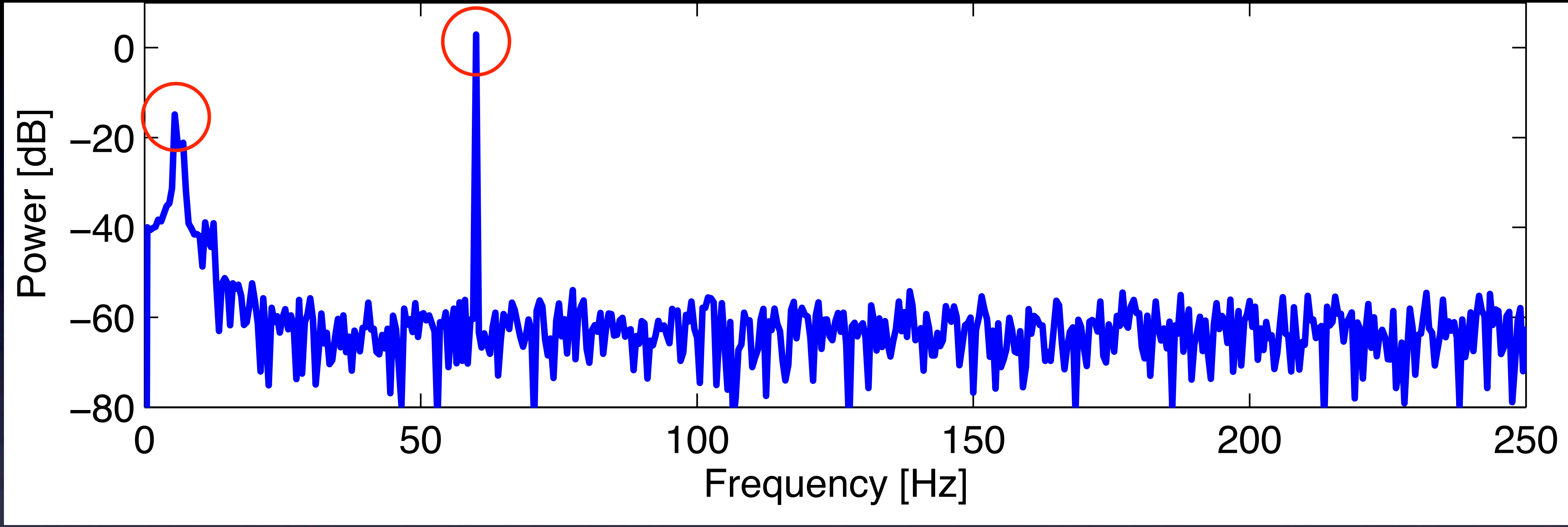


Discrete sampling

```
>> dt=0.002;
```

Power spectrum

Our goal:



- Axes: Power [dB] vs Frequency [Hz]
- A simpler representation in frequency domain. Two peaks at ~5-8 Hz, 60 Hz
- How do we compute it?

MATLAB code

Complex conjugate

Equation:

(Power spectrum)

$$\boxed{S_{xx,f}} = \frac{\boxed{2\Delta^2}}{\boxed{T}}\boxed{X_f}\boxed{X_f^*}$$

Sampling interval (0.002 s)

Duration of recording (2 s)

Fourier transform of x at frequency f

MATLAB:

```
Sxx=2*dt^2/T* fft(x).*conj(fft(x))
```

MATLAB code

```
Sxx=2*dt^2/T* fft(x).*conj(fft(x));
Sxx=10*log10(Sxx);
plot(Sxx)
```

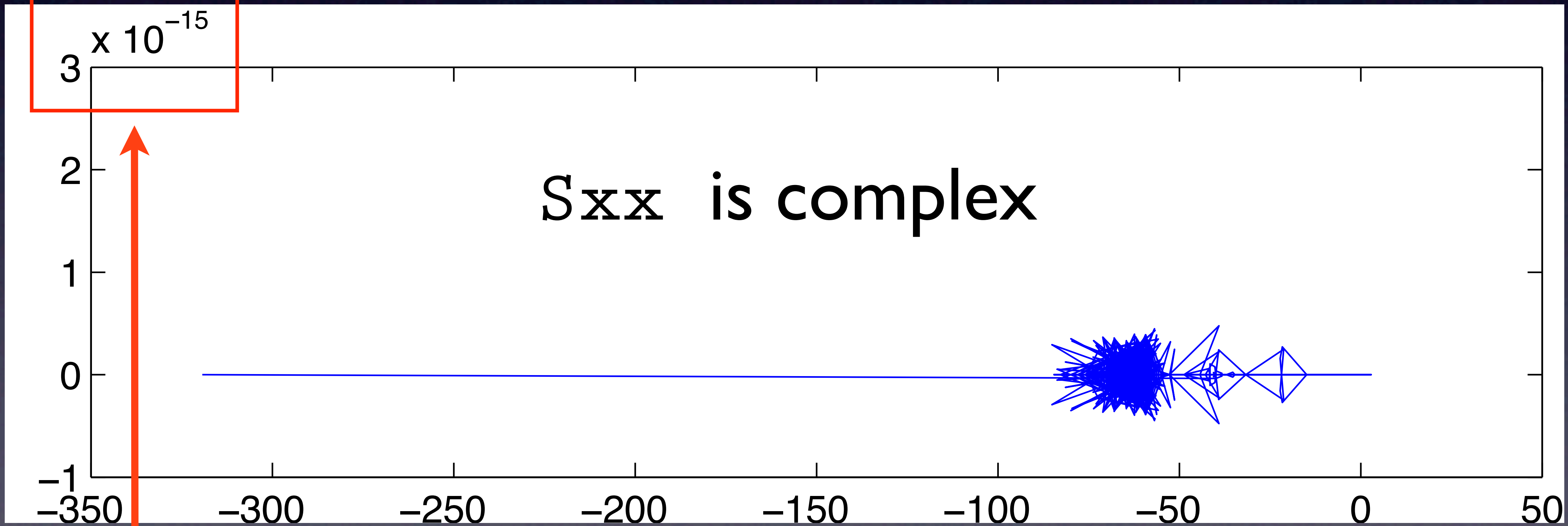
Compute the power.

Use decibel scale.

Plot it.

Hmm . . .

Imag part



Real part

Imaginary part is really tiny.

It's actually 0.

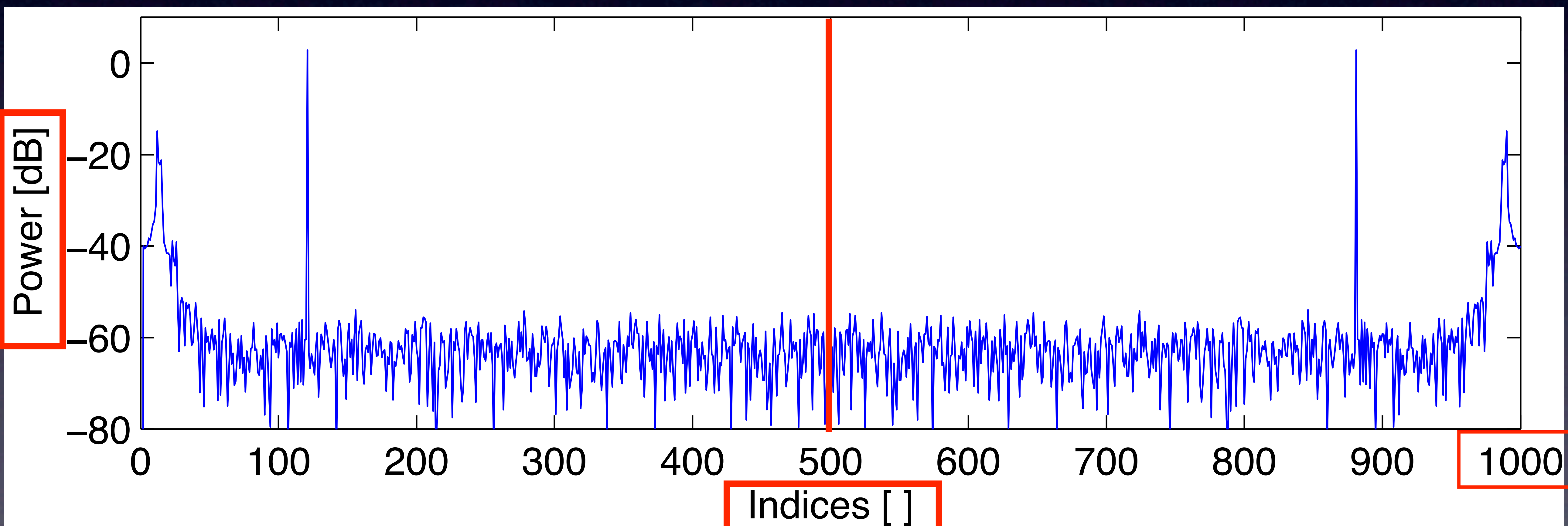
Lab Example 4

MATLAB code

```
Sxx=2*dt^2/T*fft(x).*conj(fft(x));
Sxx=10*log10(real(Sxx));
plot(Sxx)
```

Keep only the real part.

Clue?



Matches length of x

Incomplete: Label the x-axis.

Power spectrum x-axis

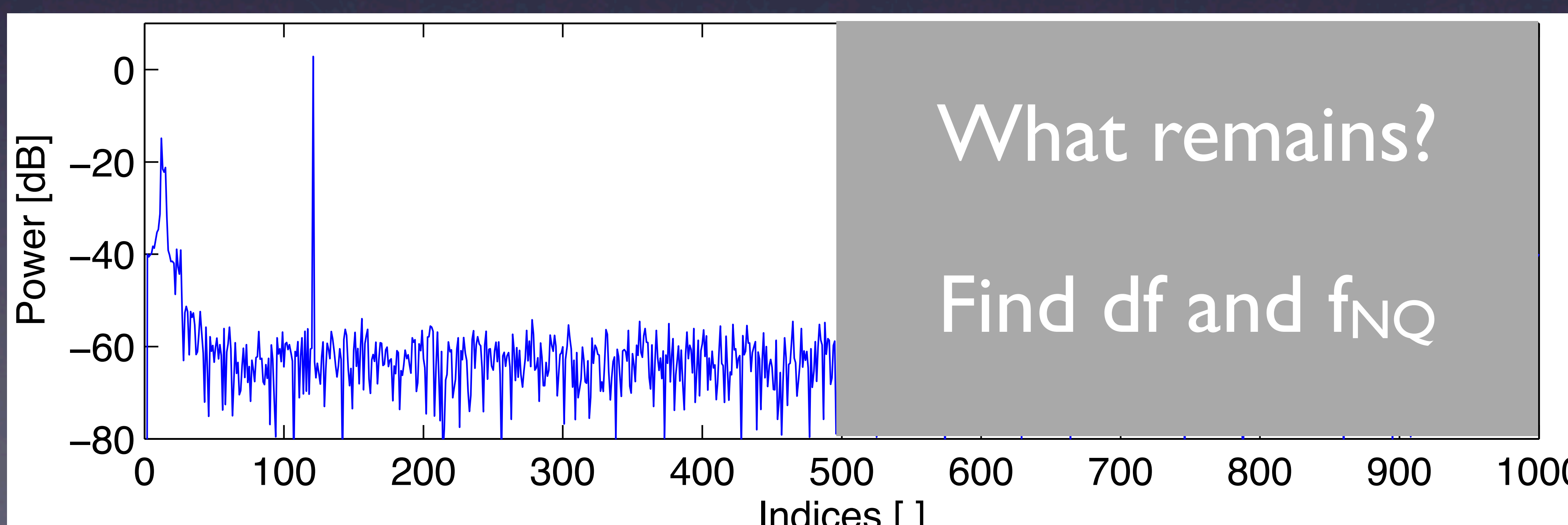
- Indices & frequencies related in a particular way . . .
  Examine vector `Sxx`:

1000

Freq	0	df	2df	...	f_{NQ} - df	f_{NQ}	
Index	1	2	3		500	501	

Frequency resolution f > 0 **Nyquist frequency** f < 0

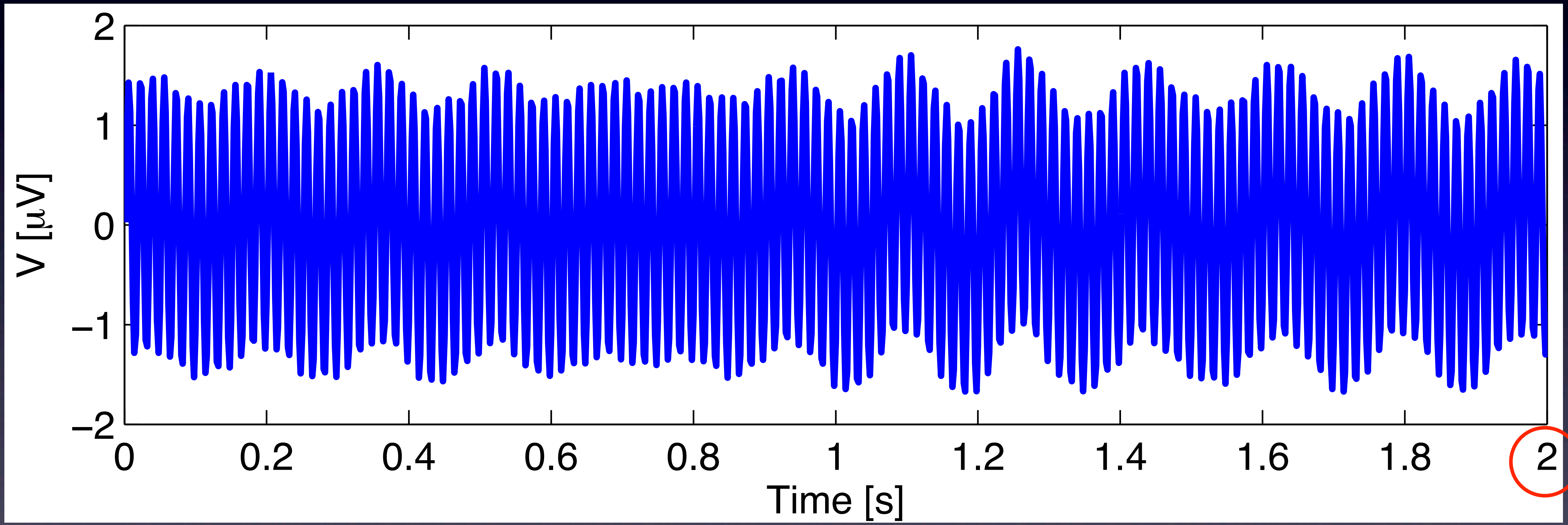
- Because data is real, f < 0 is redundant.



Power spectrum x-axis

- What is df? $$df = \frac{1}{T}$$ where T = Total time of recording.

Ex:



MATLAB:

```
>> df = 1/T;
```

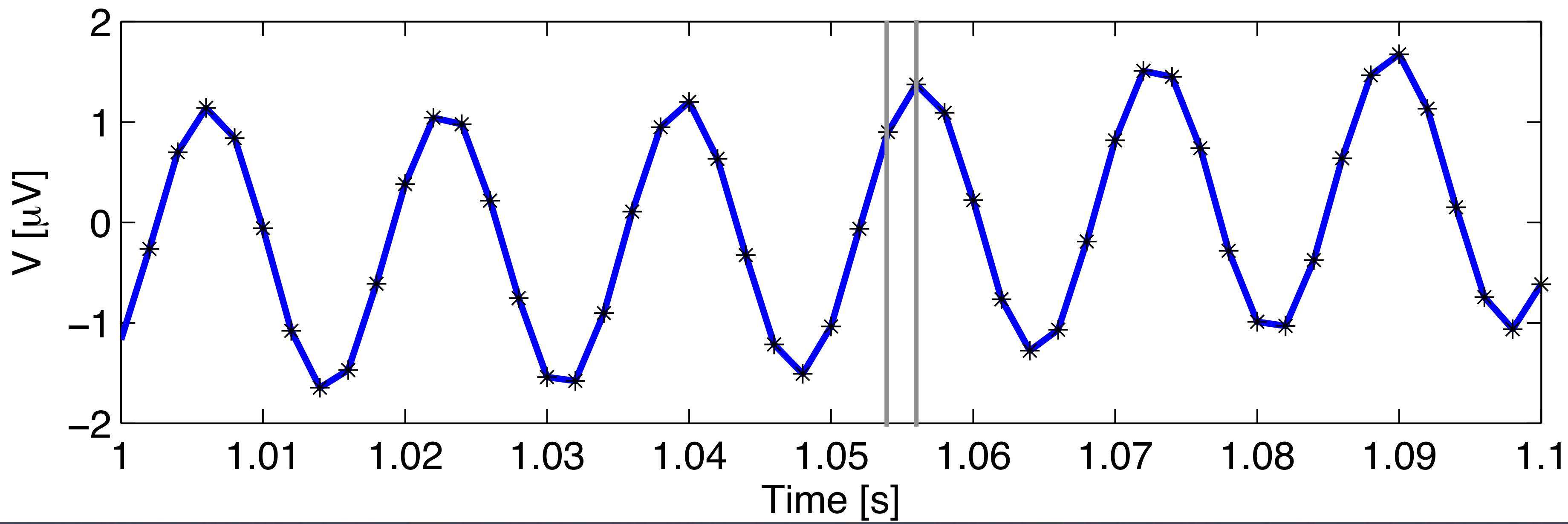
Q: How do we improve frequency resolution?

A: Increase T or record for longer time.

Power spectrum x-axis

- What is f_{NQ}? $$f_{NQ} = \frac{f_0}{2}$$ The Nyquist frequency where f_0 = sampling frequency.

Ex: Sampling interval: dt = 2 ms



Sampling frequency:

$f_0 = 1/dt$

$f_0 = 500$ Hz

$f_{NQ} = 250$ Hz

MATLAB:

```
>> fNQ=1/dt/2;
```

$\underbrace{\texttt{1/dt}}_{f_0}$

Q: How do we increase the Nyquist frequency?

A: Increase the sampling rate f_0.

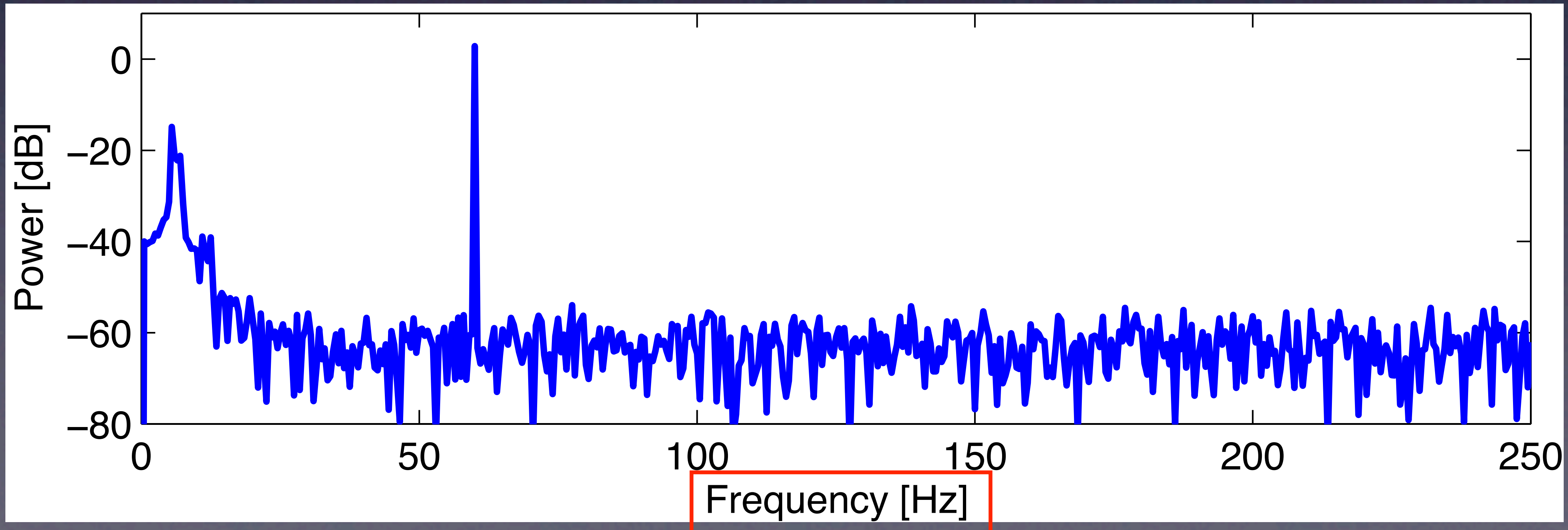
MATLAB code

```
>> Sxx = 2*dt^2/T * fft(x).*conj(fft(x));
>> Sxx = 10*log10(Sxx);
>> Sxx = Sxx(1:length(x)/2+1);
>> df = 1/T;  fNQ = 1/dt/2;
>> faxis = (0:df:fNQ);
>> plot(faxis, Sxx);  ylim([-80 10])
```

First half of pow

Define df & f_{NQ}

Frequency axis



Summary

```
>> Sxx=2*dt^2/T*fft(x).*conj(fft(x));
```

Frequency resolution $df = \frac{1}{T}$

```
>> df = 1/T;
```

Nyquist frequency $f_{\mathrm{NQ}} = \frac{f_0}{2}$

```
>> fNQ=1/dt/2;
```

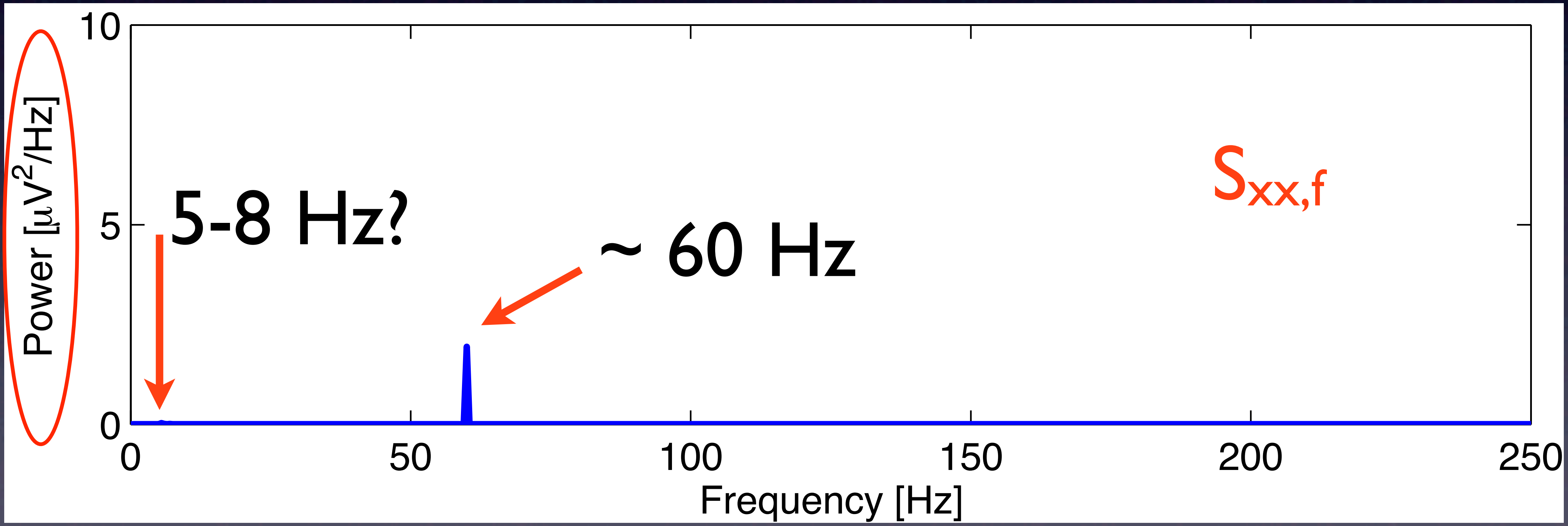
- For finer frequency resolution: record more data.
- To observe higher frequencies: increase sampling rate.
- Built-in routines: `>> periodogram(...)`
  Requires Signal Processing Toolbox
- Many subtleties ...

Power spectrum

A note on scale . . .

~~`>> Sxx = 10*log10(Sxx);`~~ Decibel scale

Consider not using the decibel scale . . .

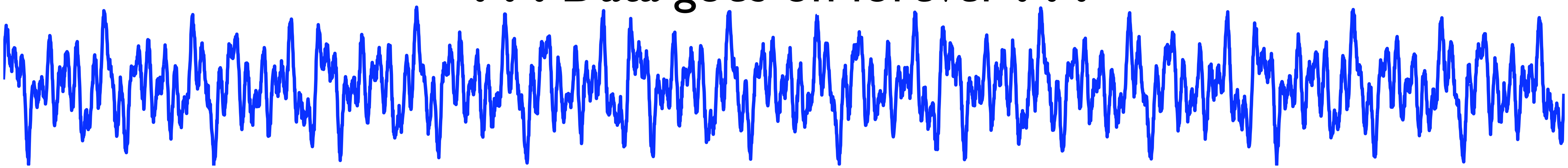


- Use decibel scale to reveal low power features.

Tapers

- Doing nothing, we make an implicit taper choice . . .

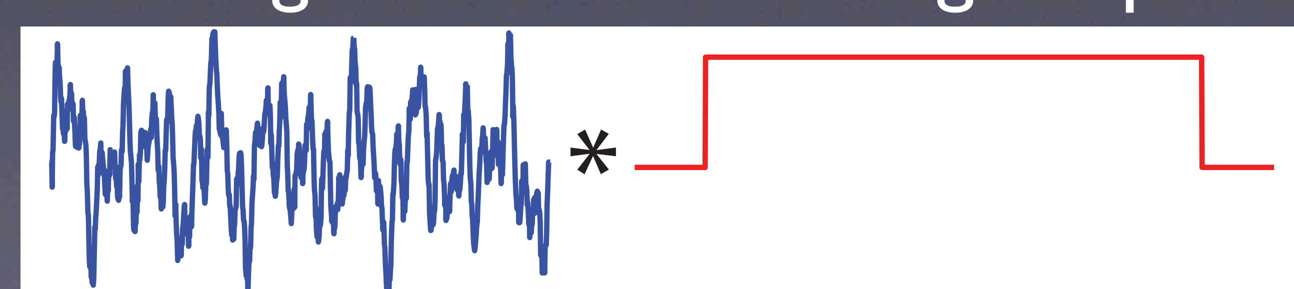
. . . Data goes on forever . . .



What we're observing:

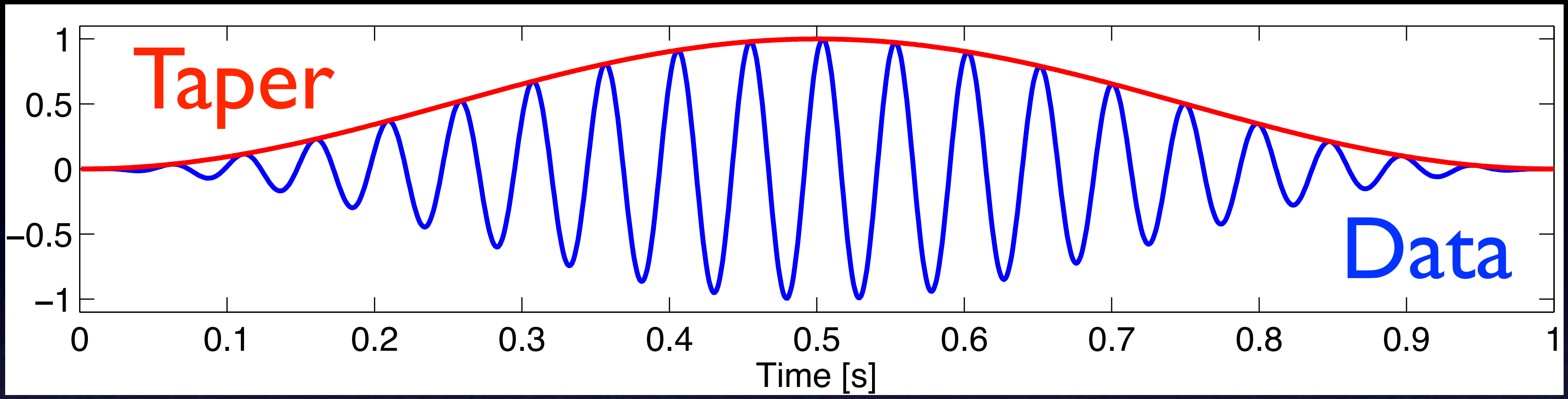
Signal

Rectangle taper

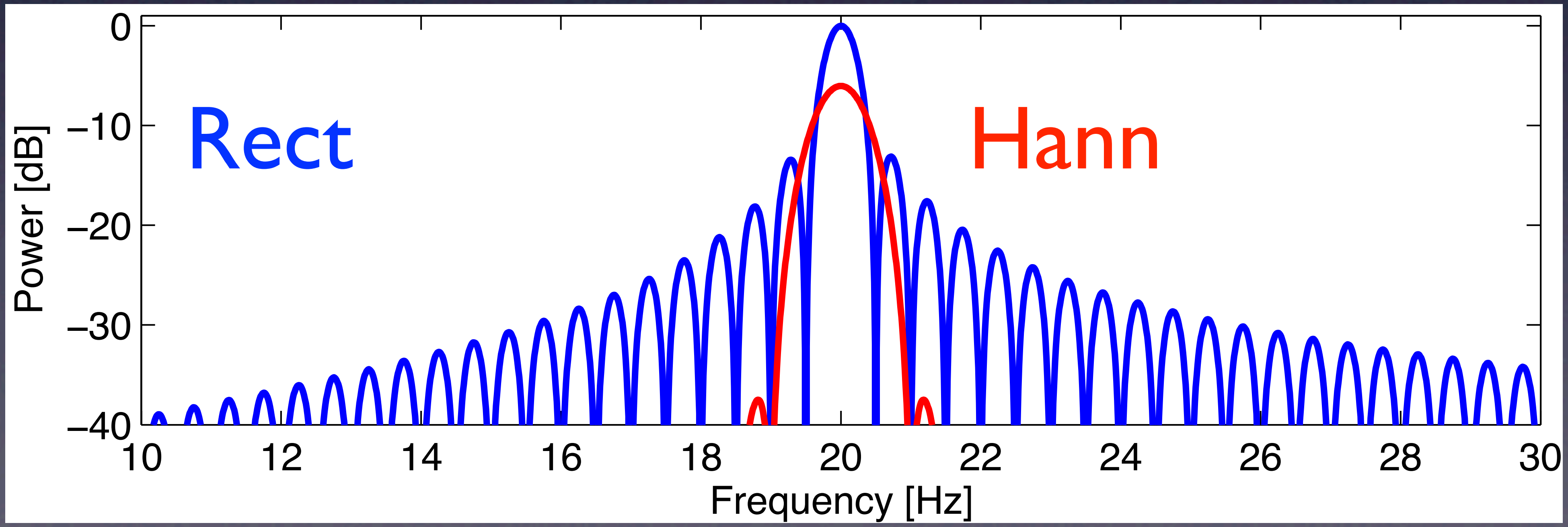


Hann taper

- Idea: smooth the sharp edges of rectangle taper.



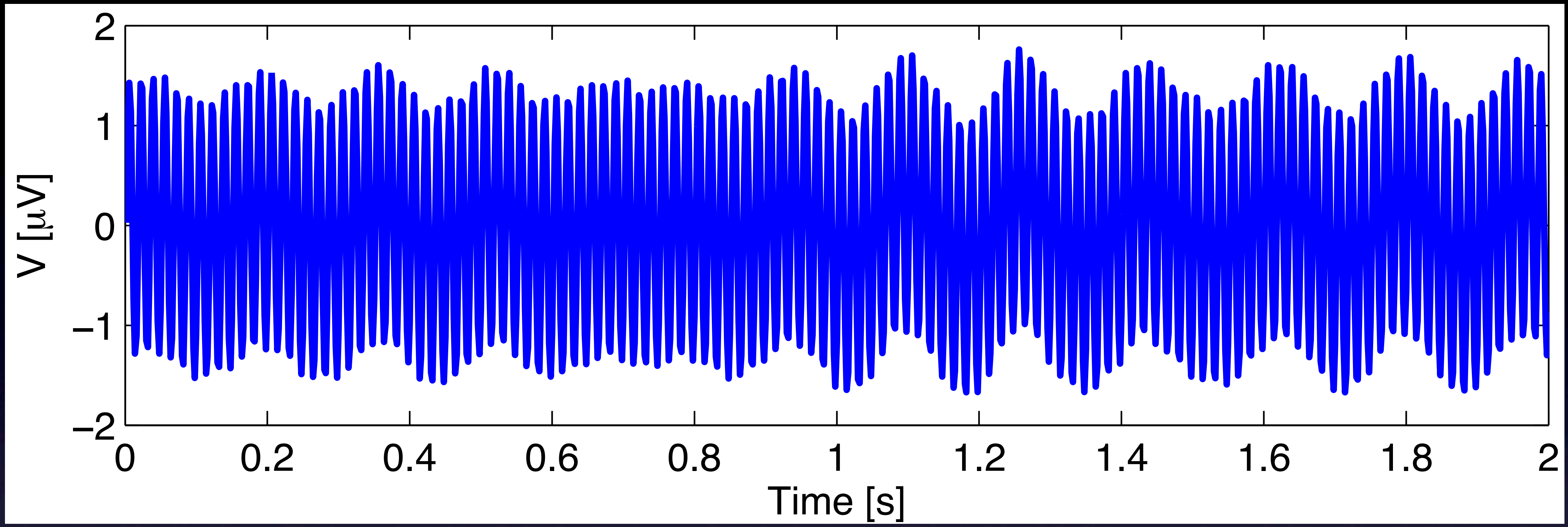
- Compute power spectrum of tapered data.



Taper reduces the "sidelobes".

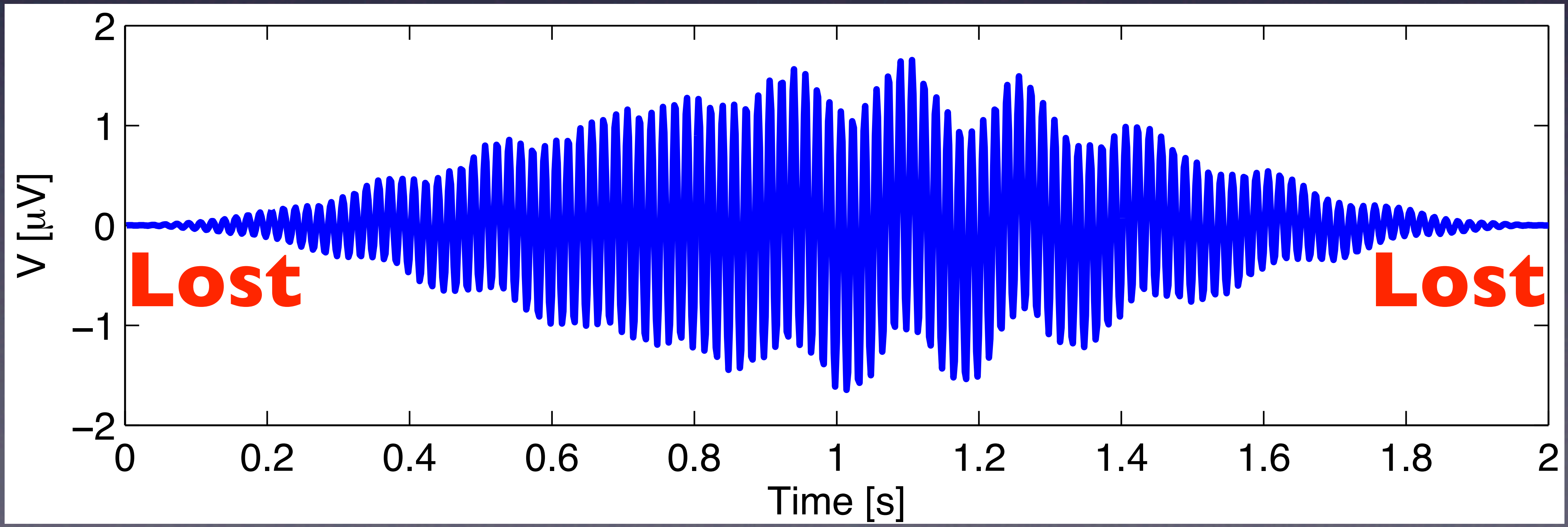
Ex: Hann taper

x =



MATLAB: `>> xh = hann(length(x)).*x;`

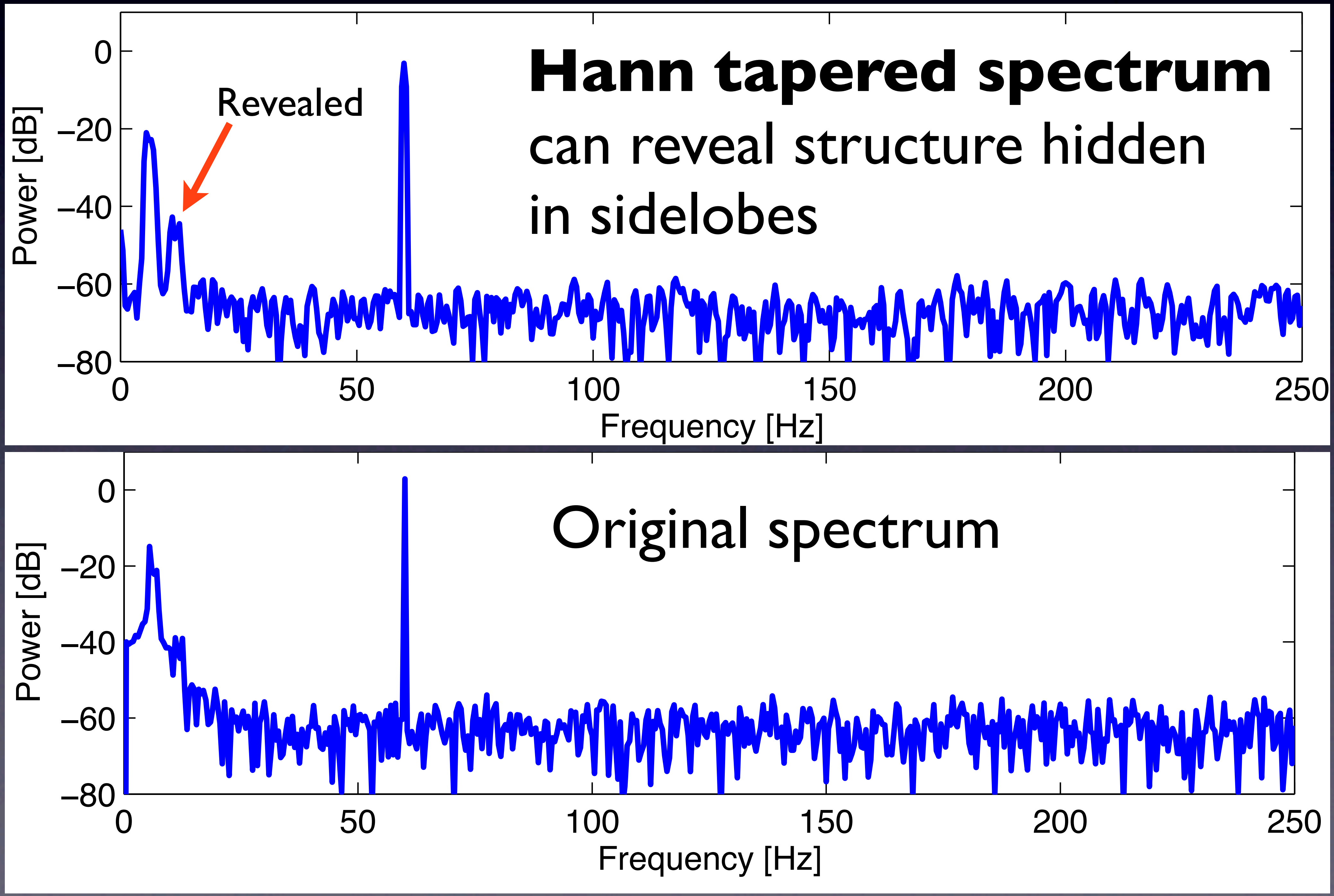
xh =



Ex: Hann taper

Compute the power spectrum of Hann tapered data

```
>> Sxx = 2*dt^2/T * fft(xh).*conj(fft(xh));
```



Multi-sensor data

Download data: http://math.bu.edu/people/mak/sfn-2013/

The Science of Large Data Sets: Spikes, Fields, and Voxels

Society for Neuroscience Short Course #2

Data
Download the data set for power spectrum analysis: d1.mat
Download the data set for coherence analysis: d2.mat
Load this data set in MATLAB using the *load* command.

MATLAB code
Download an M-file that includes MATLAB code to analyze these data: sfn_tutorial.m

Tutorial slides : As a PDF.

Software Links
Chronux
EEGLab

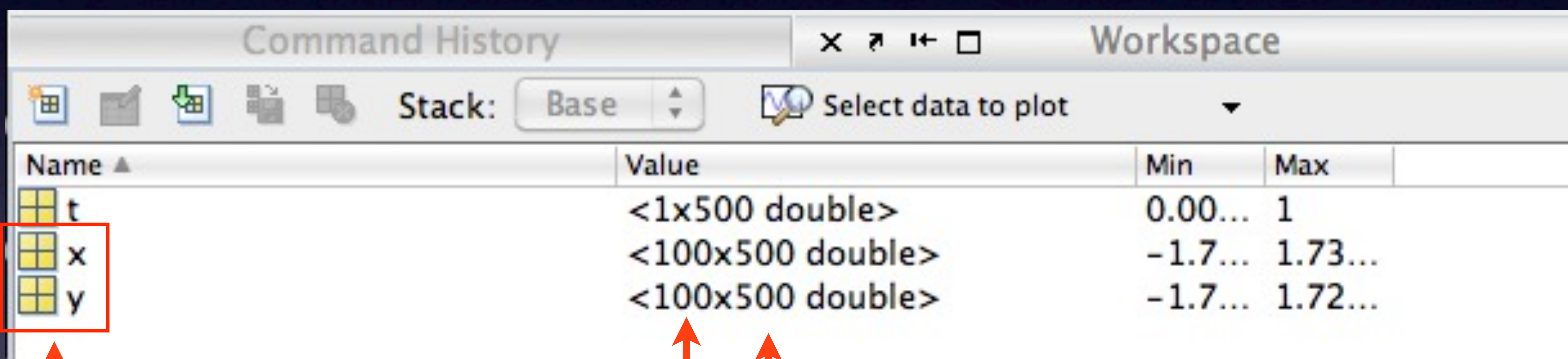
Book Links
Matlab for Neuroscientists
Signal Processing for Neuroscientists
Observed Brain Dynamics

Contact: Email Mark Kramer

Multi-sensor data

Download data: http://math.bu.edu/people/mak/sfn-2013/

```
>> clear
>> load d2.mat
```



Data from 2 sensors

Trials

100 total trials

Time

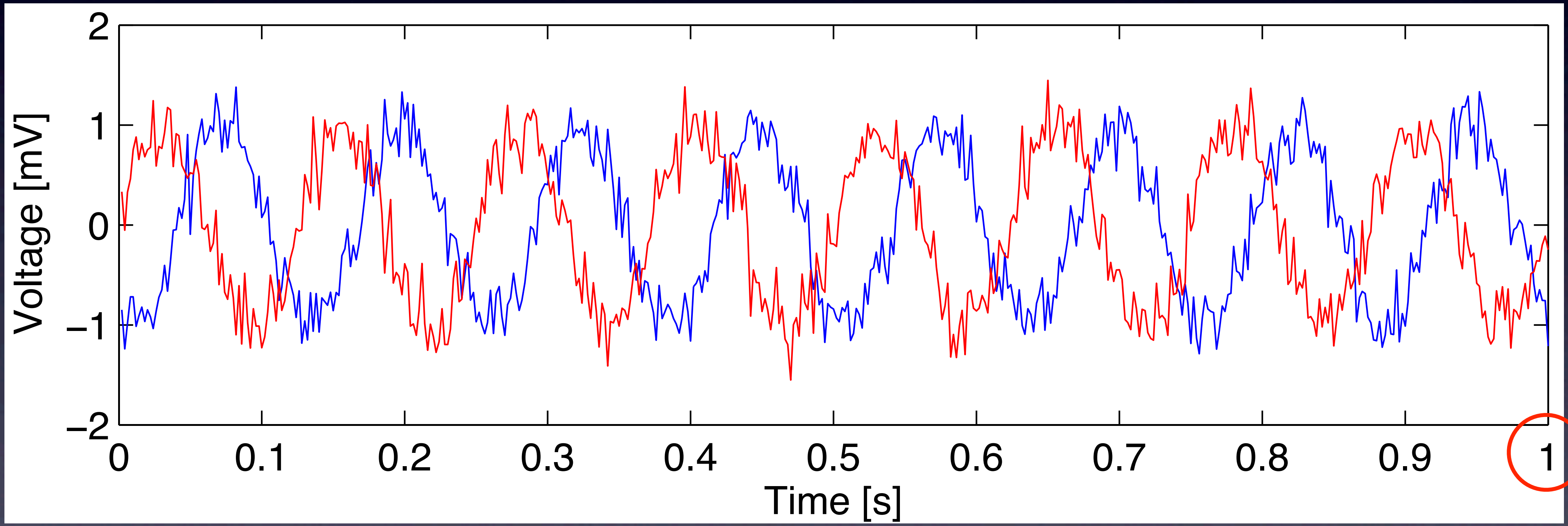
500 time samples per trial

Visualize

```
>> plot(t,x(1,:))
>> hold on
>> plot(t,y(1,:), 'r')
>> hold off
```

Sensor *x*, first trial.

Sensor *y*, first trial.



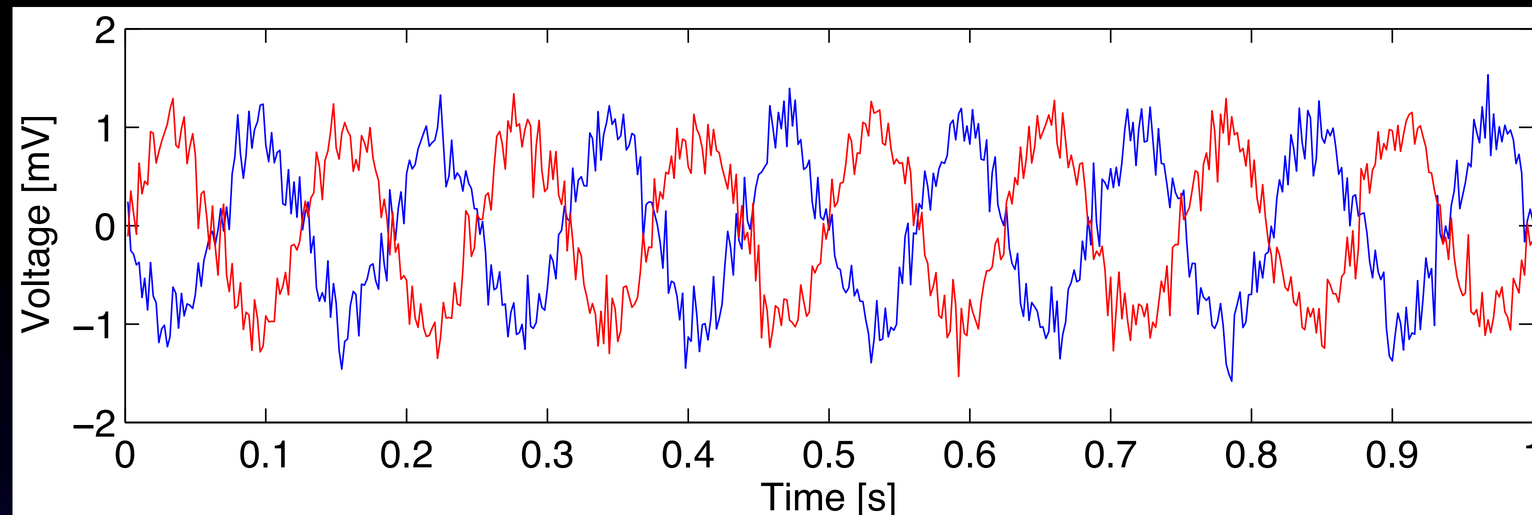
Visual inspection:

- Rhythmic
- Q: Are the signals at the two sensors “related”?

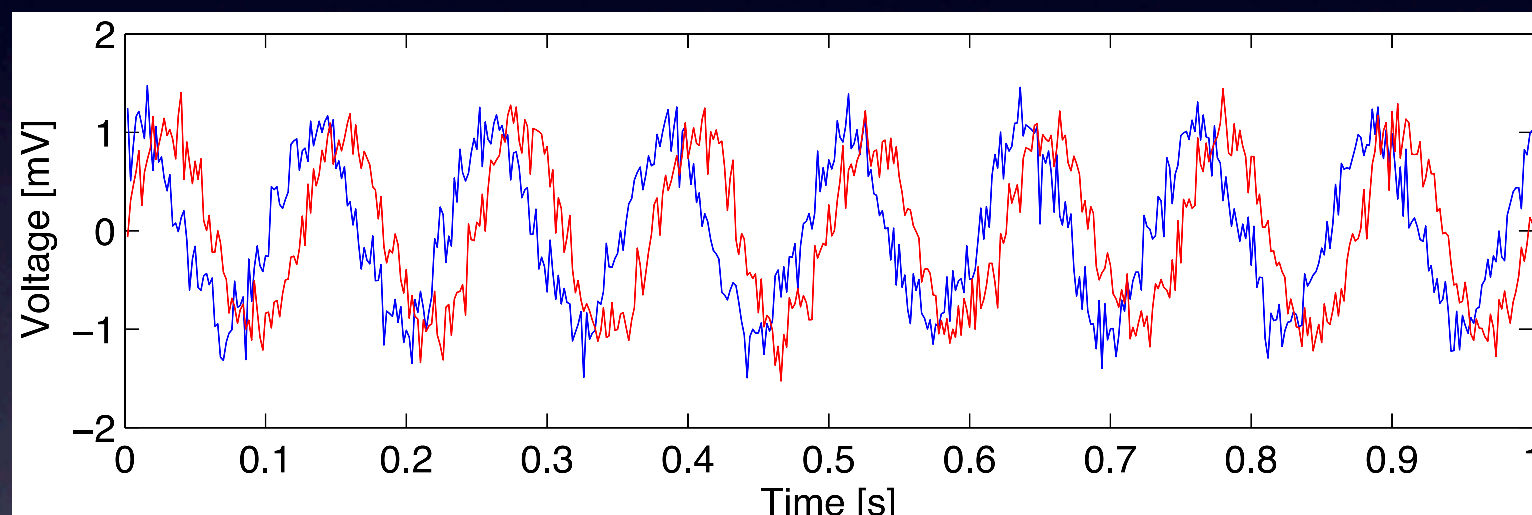
```
>> T=1;
>> dt=0.002;
```

Visualize

Trial 2

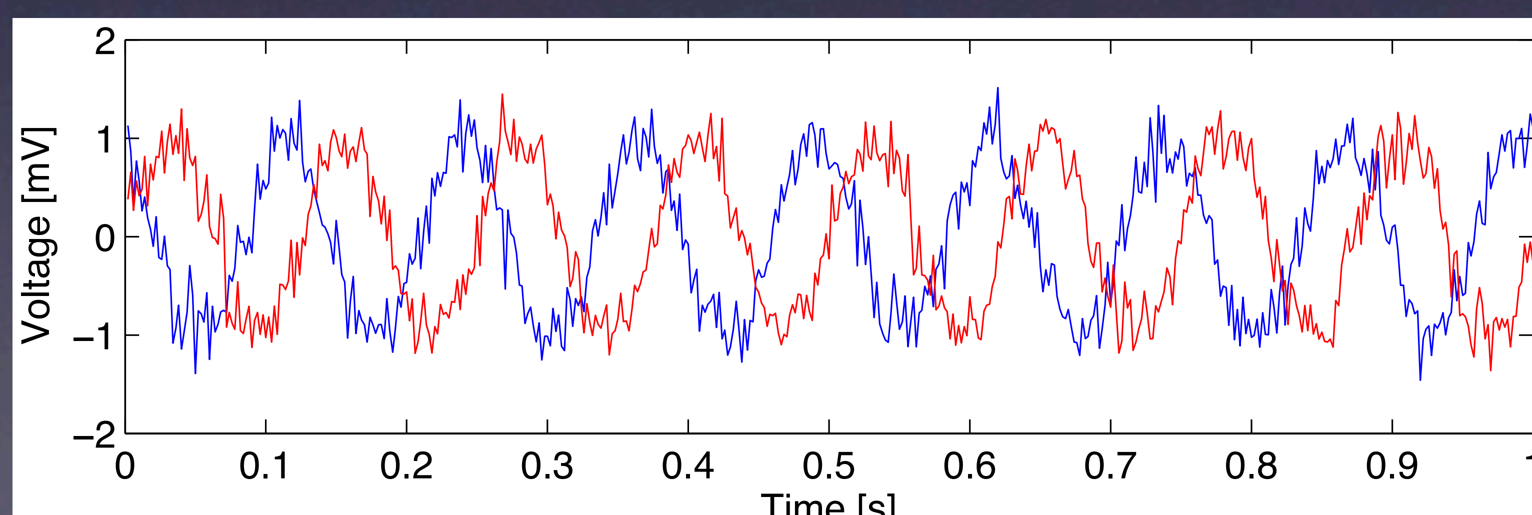


Trial 3



. . .

Trial 100



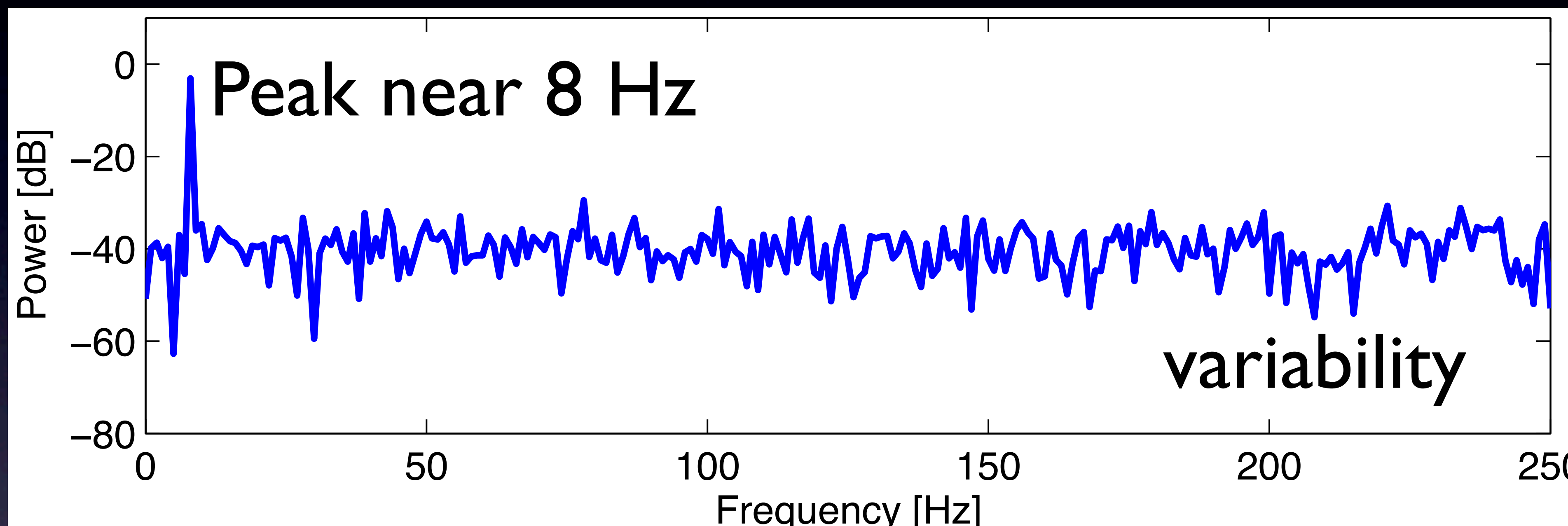
- They're rhythmic . . .

Power spectrum

For a <u>single</u> trial . . . same as before

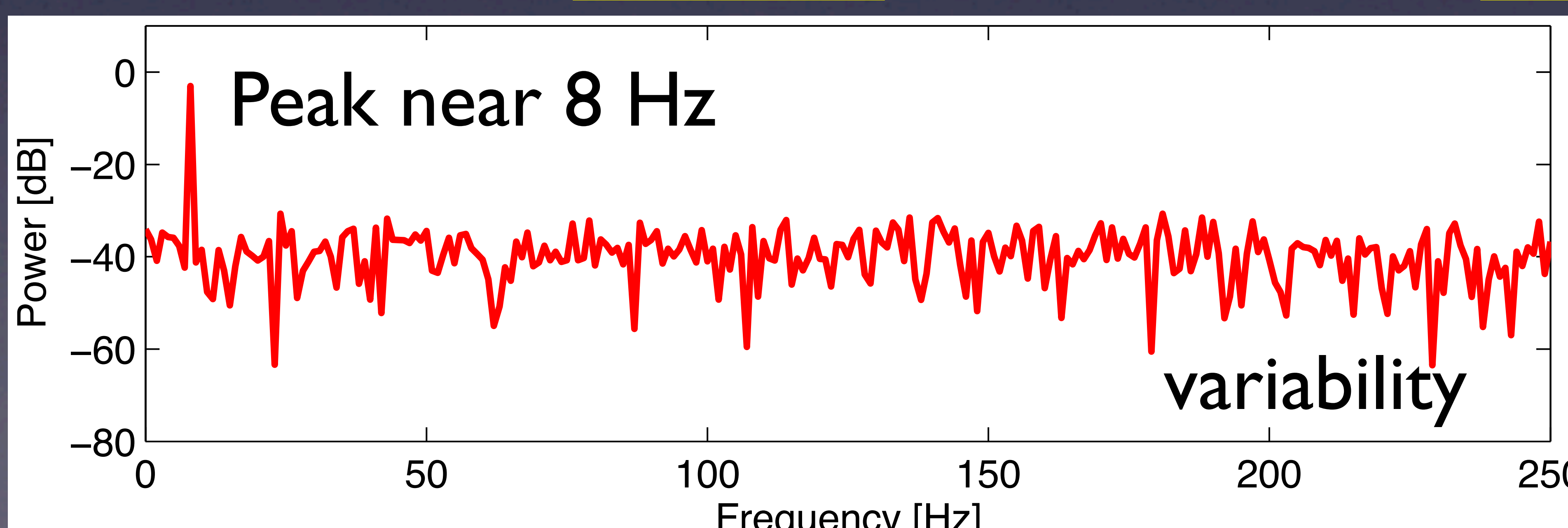
```
Sxx=2*dt^2/T*fft(x(1,:)).*conj(fft(x(1,:)))
```

Power spectrum of x for Trial 1




```
Syy=2*dt^2/T*fft(y(1,:)).*conj(fft(y(1,:)))
```

Power spectrum of y for Trial 1



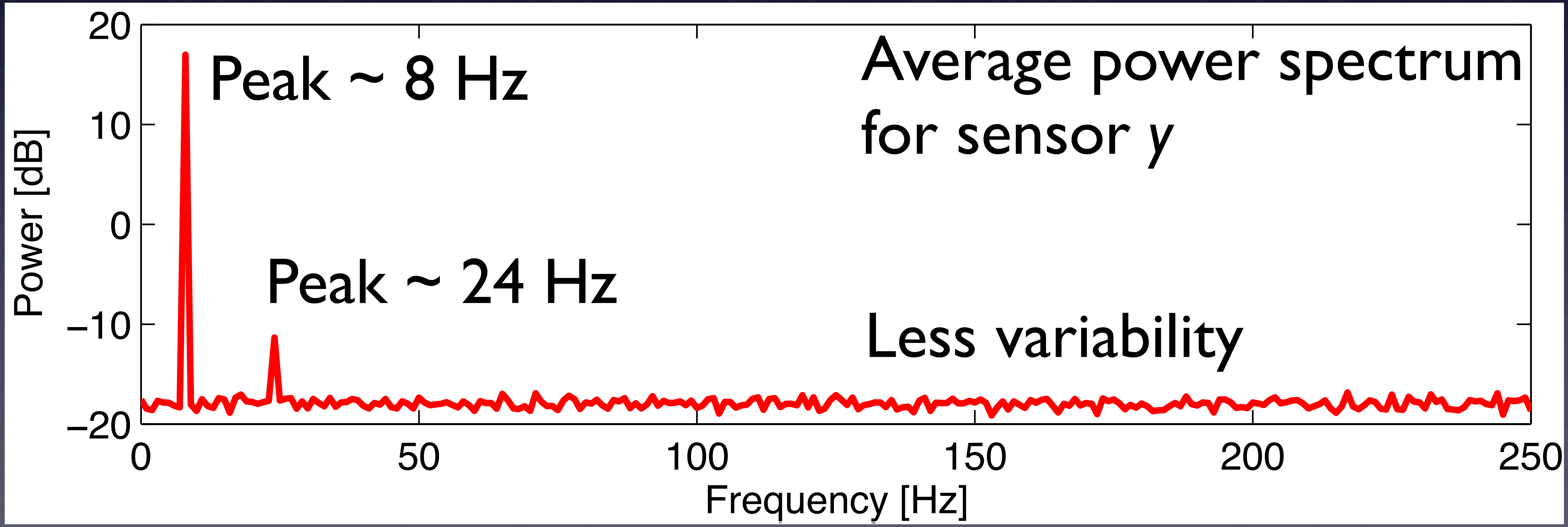
Power spectrum

Averaged across trials . . .

```
for k=1:K
  x0 = x(k,:);
  Sxx = Sxx+2*dt^2/T* fft(x0).*conj(fft(x0));
end
```

Get the data for trial k ... and compute the power spectrum

Accumulate in the sum.



- Q: Are the signals at the two sensors “related”?

Coherence

Equation:

$\boxed{\text{Cross-spectrum}}$ between *x*,*y* at frequency f

$\boxed{\text{Trial average}}$

Absolute value

$$\kappa_{xy,f} = \frac{|\boxed{<}\boxed{S_{xy,f}}\boxed{>}|}{\sqrt{\boxed{<}S_{xx,f}\boxed{>}<S_{yy,f}>}}$$

Power spectrum of *x* at frequency f $\boxed{\text{averaged across trials.}}$

Power spectrum of *y* at frequency f averaged across trials.

We know:

Coherence

Equation (cross spectrum):

Complex conjugate

Trial average

$$< S_{xy,f} > = \frac{2\Delta^2}{T} \frac{1}{K} \sum_{k=1}^{K} X_{f,k} Y^*_{f,k}$$

Number of trials

Sum over trials

Fourier transform of *x* & *y* at frequency f, trial k

MATLAB:

```
for k=1:K
  Sxy(k,:) =...
    2*dt^2/T * fft(x(k,:)).*conj(fft(y(k,:)));
end
```

Coherence

```
K   = size(x,1);
N   = size(x,2);
```

Helpful variables: # trials, # time points

```
Sxx = zeros(K,N);
Syy = zeros(K,N);
Sxy = zeros(K,N);
```

Create variables to save the spectra.

For each trial ...

```
for k=1:K
    Sxx(k,:) = 2*dt^2/T * fft(x(k,:)) .* conj(fft(x(k,:)));
    Syy(k,:) = 2*dt^2/T * fft(y(k,:)) .* conj(fft(y(k,:)));
    Sxy(k,:) = 2*dt^2/T * fft(x(k,:)) .* conj(fft(y(k,:)));
end
```

Power *x*

Power *y*

Cross spectra

```
Sxx = Sxx(:,1:N/2+1);
Syy = Syy(:,1:N/2+1);
Sxy = Sxy(:,1:N/2+1);
```

Keep the positive frequencies.

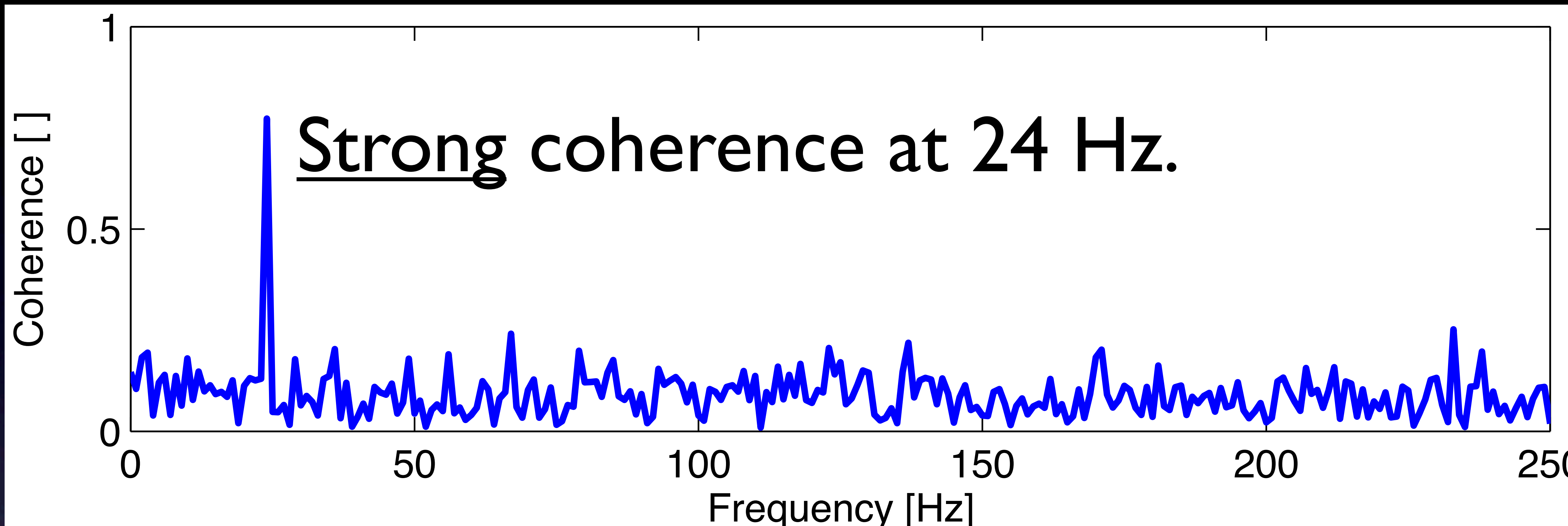
```
Sxx = mean(Sxx,1);
Syy = mean(Syy,1);
Sxy = mean(Sxy,1);
```

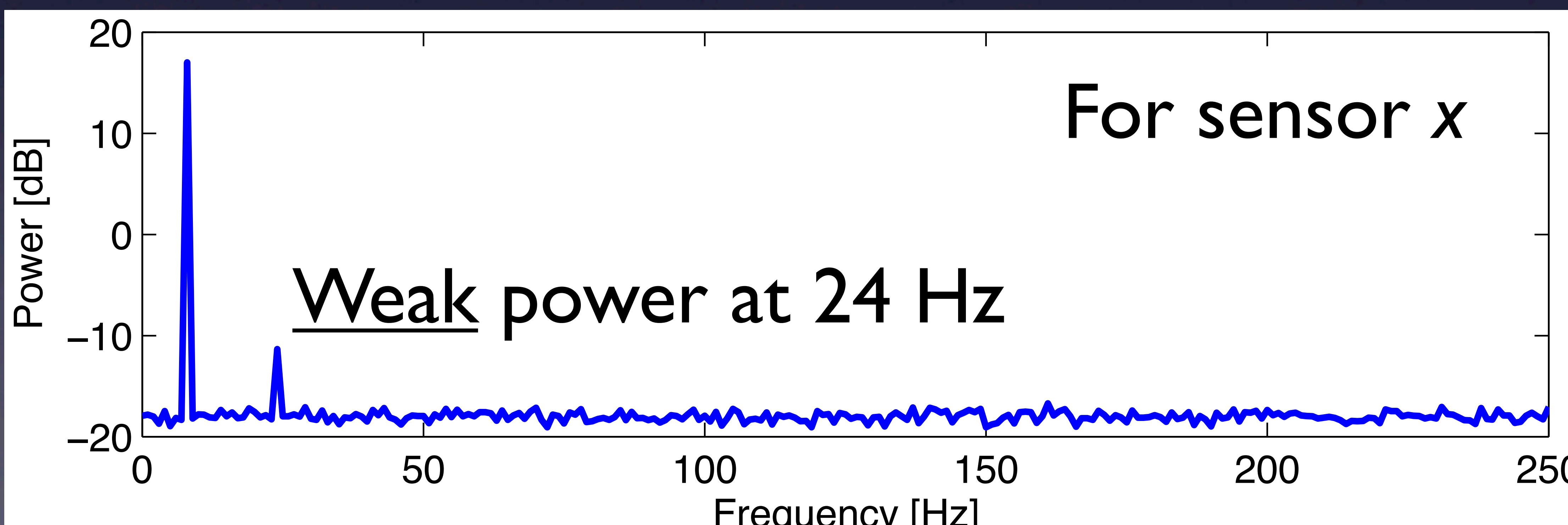
Average across trials.

```
cohr = abs(Sxy) ./ (sqrt(Sxx) .* sqrt(Syy));
```

Compute coherence.

Coherence





- High power does not imply high coherence

Coherence

Q: What is the coherence between two signals for a single trial?

Claim: 1 for all frequencies.

MATLAB:

```
x0 = x(1,:);
y0 = y(1,:);
```

Select data from the first trial.

```
Sxx = 2*dt^2/T * fft(x0) .* conj(fft(x0));
Syy = 2*dt^2/T * fft(y0) .* conj(fft(y0));
Sxy = 2*dt^2/T * fft(x0) .* conj(fft(y0));
```

Power *x*
Power *y*
Cross spectra

```
Sxx = Sxx(1:N/2+1);
Syy = Syy(1:N/2+1);
Sxy = Sxy(1:N/2+1);
```

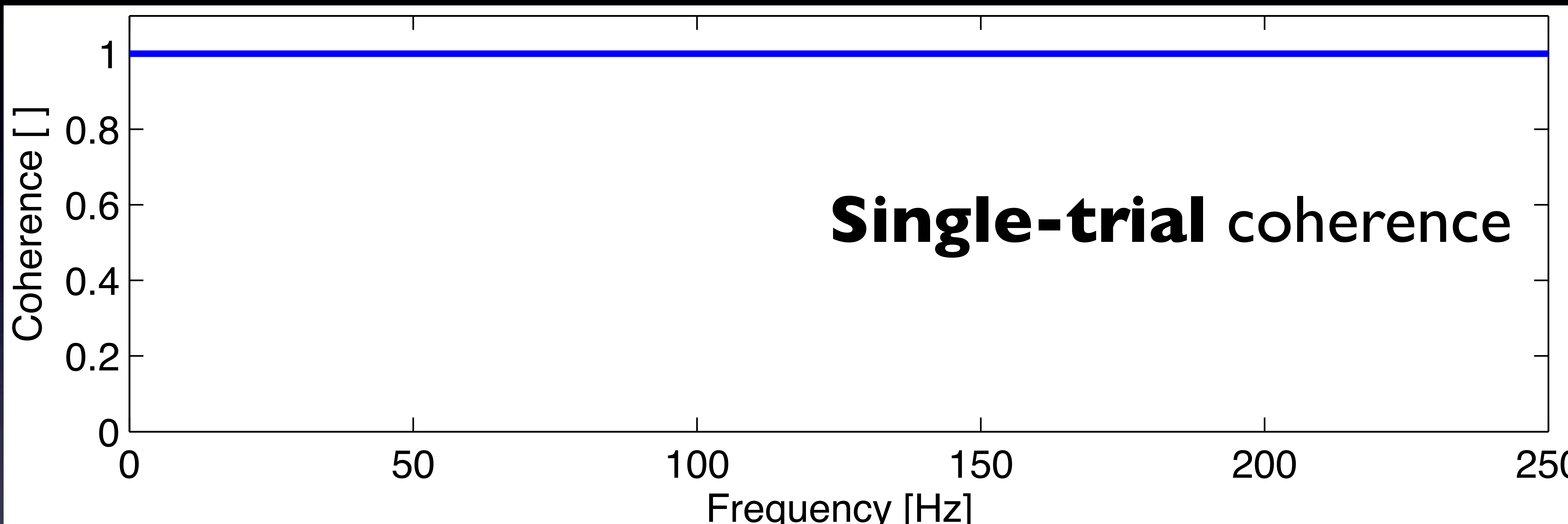
Keep the positive frequencies.

```
cohr = abs(Sxy) ./ (sqrt(Sxx) .* sqrt(Syy));
```

Compute coherence.

Coherence

Compute the result:



Observation: Perfect coherence for all frequencies.

Maybe data unique . . .

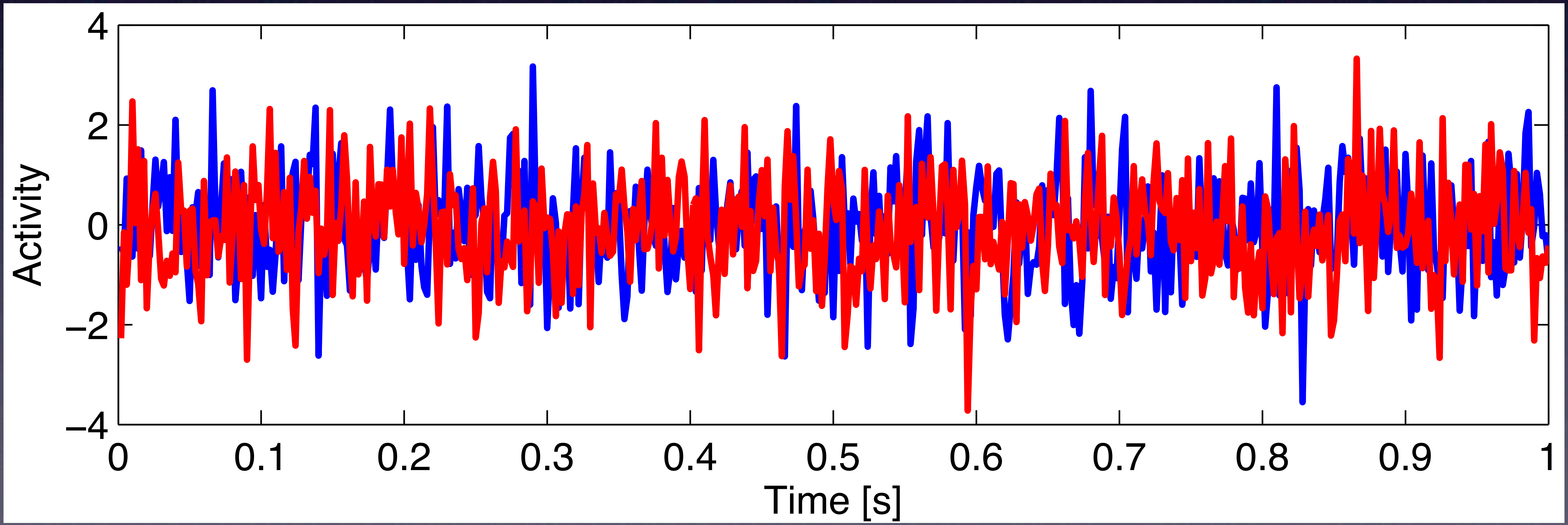
Coherence

Q: What is the coherence between two signals for a single trial?

Consider "artificial" random data.

`x0 = randn(1,N);` Sensor *x*, one trial of random data.

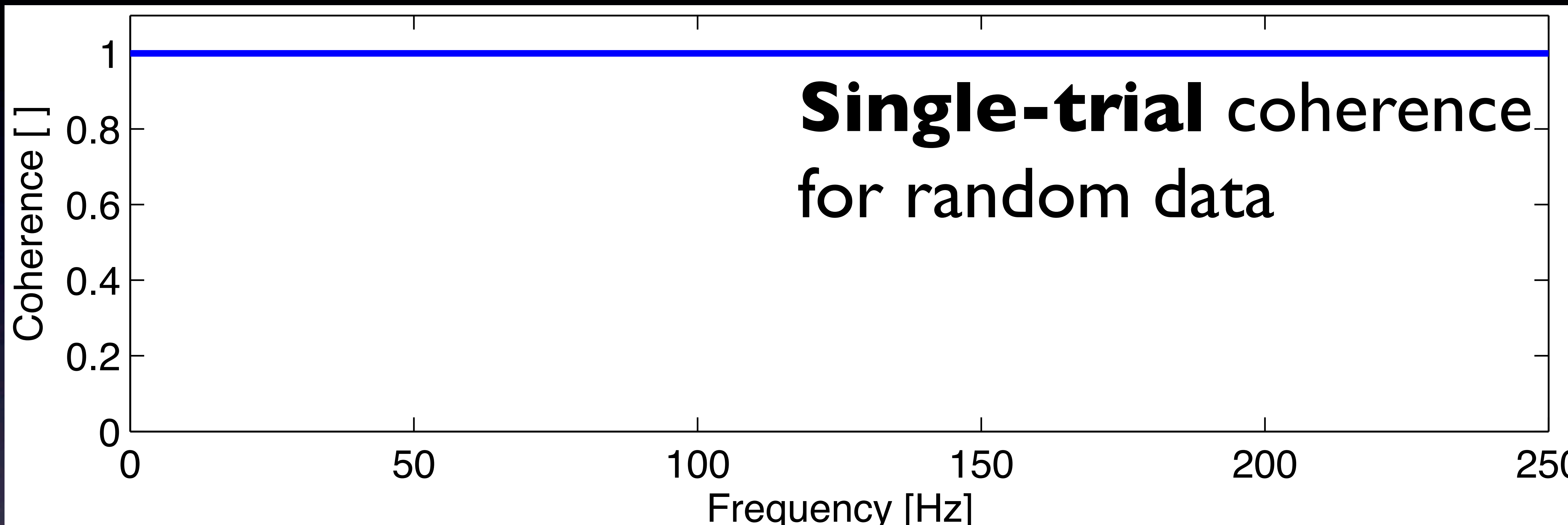
`y0 = randn(1,N);` Sensor *y*, another trial of random data.



Are these two signals coherent?

Coherence

Compute the result:



Observation: Perfect coherence for all frequencies.

Two sensors are coherent "across trials" for a single trial.

Alternative: multi-taper method.

Conclusions

In MATLAB:

- Power spectrum
- Coherence

Only scratched the surface . . .

MATLAB for Neuroscientists, Wallisch et al
Observed Brain Dynamics, Mitra & Bokil
Chronux.org, EEGLab

Spectral Analysis and Time Series, Priestley
Spectral Analysis for Physical Applications, Percival & Walden

Stay tuned . . .

Kramer, Eden 2014-15