# Introduction to Neural Computation

Prof. Michale Fee
MIT BCS 9.40 — 2018

## Lecture 12 - Spectral analysis II

## Game plan for Lectures 11, 12, and 13 —
## *Develop a powerful set of methods for understanding the temporal structure of signals*

- Fourier series, Complex Fourier series, Fourier transform, Discrete Fourier transform (DFT), Power Spectrum
- Convolution Theorem
- Noise and Filtering
- Shannon-Nyquist Sampling Theorem
  - https://markusmeister.com/2018/03/20/death-of-the-sampling-theorem/
- Spectral Estimation
- Spectrograms
- Windowing, Tapers, and Time-Bandwidth Product
- Advanced Filtering Methods

# Discrete Fourier transform

- Some code

WSpec.m | recordaudio.m | continuous_cos.m

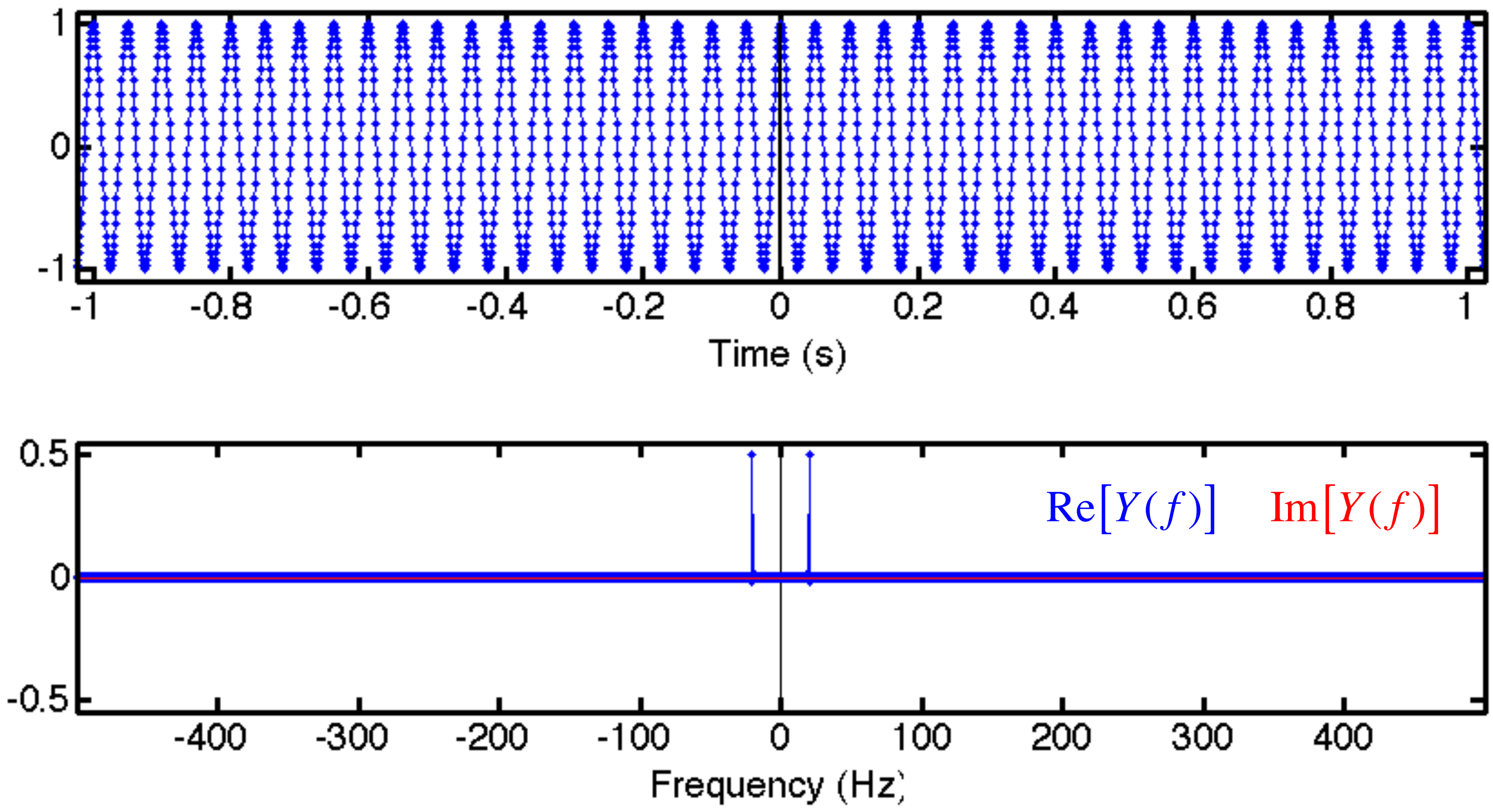
```
%
N=2048;                    % number of samples in time
                           %
dt=.001;                   % sampling interval
Fs=1./dt;               % sampling frequency
time=dt*[-N/2:N/2-1];      % timebase
%
freq=20.;                  % frequency of sine wave in Hz
y=cos(2*pi*freq*time);
%
%
yshft=circshift(y,[0,N/2]);     % First shift zero point from center to
                                    % first point in the array
ffty=fft(yshft, N)/N;           % Now compute the FFT

Y=circshift(ffty,[0,N/2]);          % Now shift the spectrum to put zero frequency
                                    % at the middle of the array
%
%Compute the vector of frequencies
df=Fs/N;
Fvec=df*[-N/2:N/2-1];
%
```

# Discrete Fourier transform

- Some examples – sine and cosine

$$y(t) = \cos(2\pi f_0 t) \qquad f_0 = 20Hz$$

Continuous_cos.m

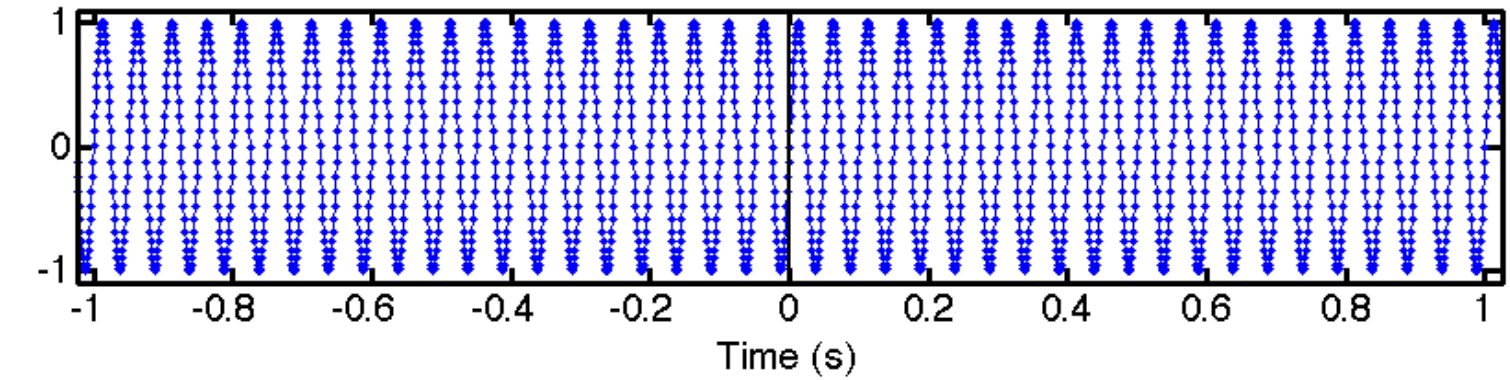


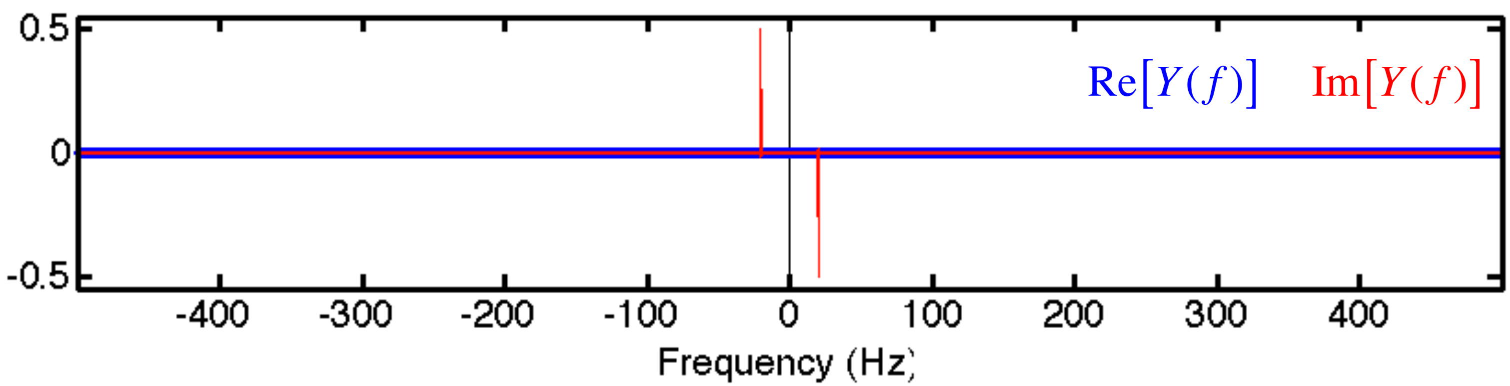
# Discrete Fourier transform

- Some examples – sine and cosine

$$y(t) = \sin(2\pi f_0 t) \qquad f_0 = 20Hz$$

Continuous_sin.m

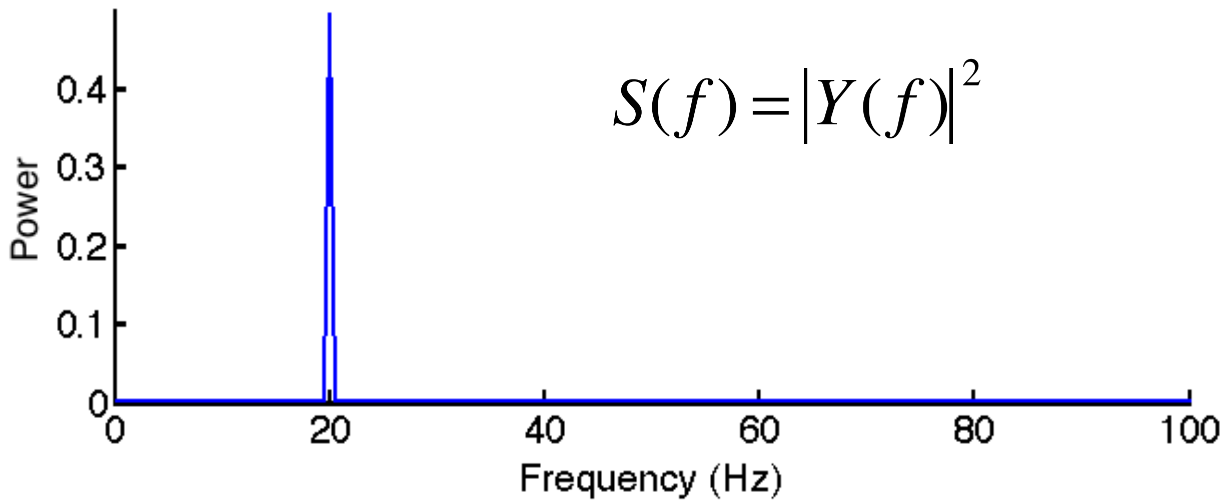


# Discrete Fourier transform

- Power spectrum of sine and cosine

Continuous_sin.m

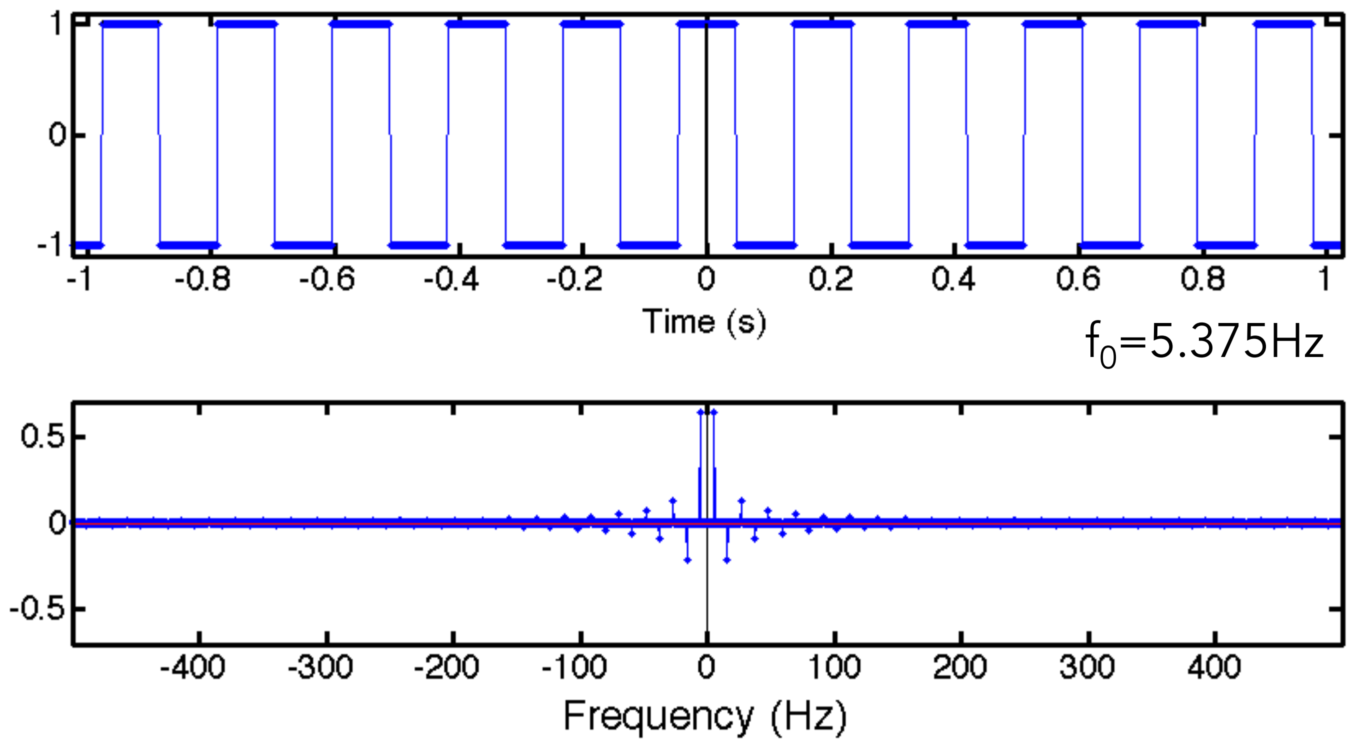



$$S(f) = |Y(f)|^2$$

# Discrete Fourier transform

- Some examples – square waves

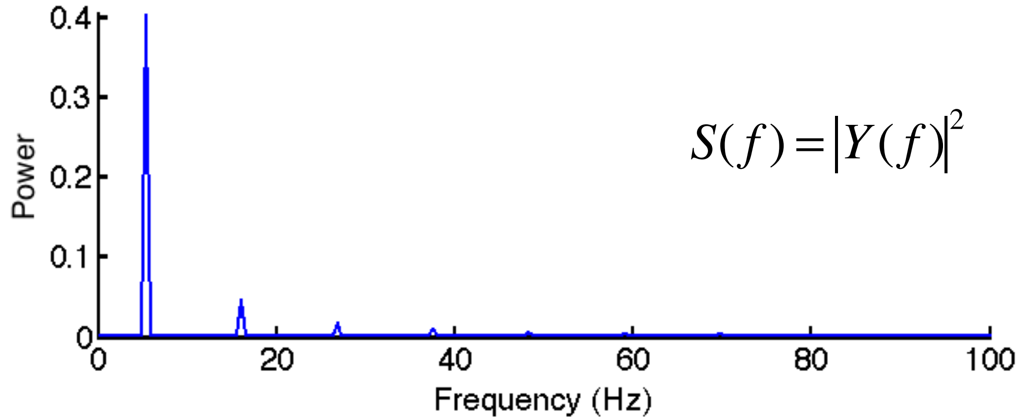
Continuous_square.m



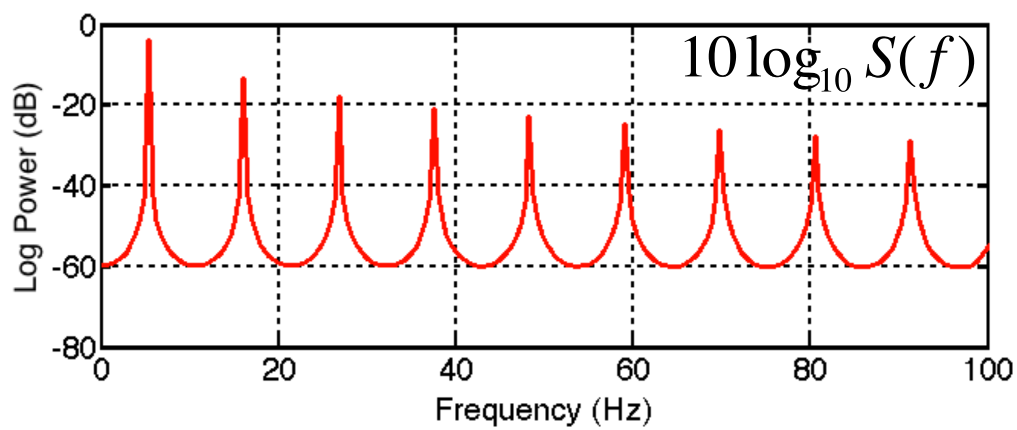
# Discrete Fourier transform

- Power spectrum– square wave

Continuous_square.m



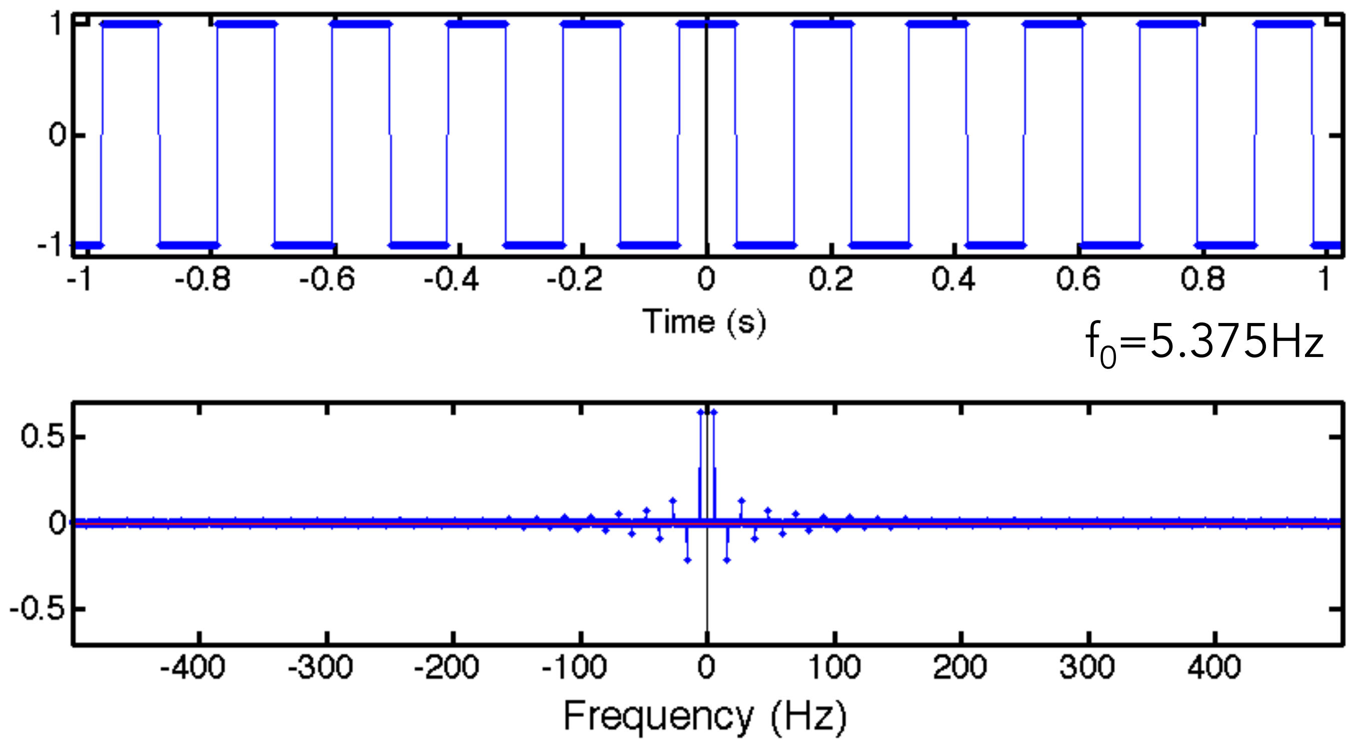

Spectrum plotted in units of decibels (dB)



# Discrete Fourier transform

- Some examples – square waves

Continuous_square.m

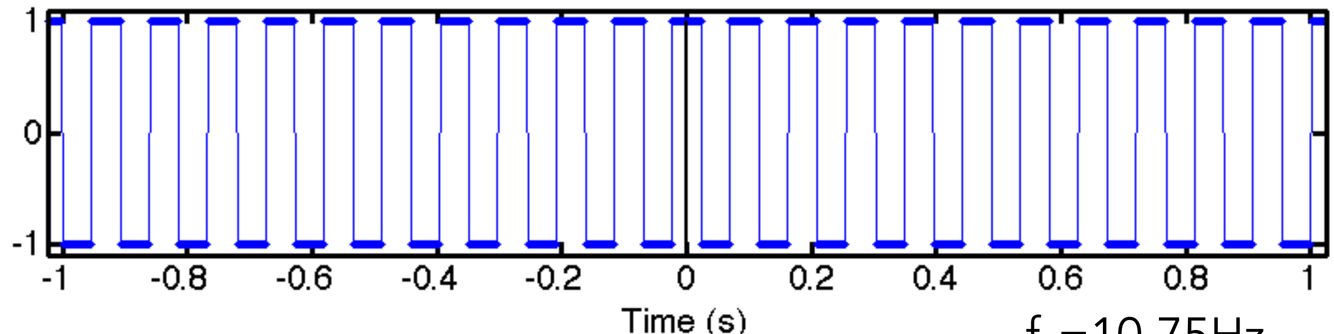



$f_0$=5.375Hz

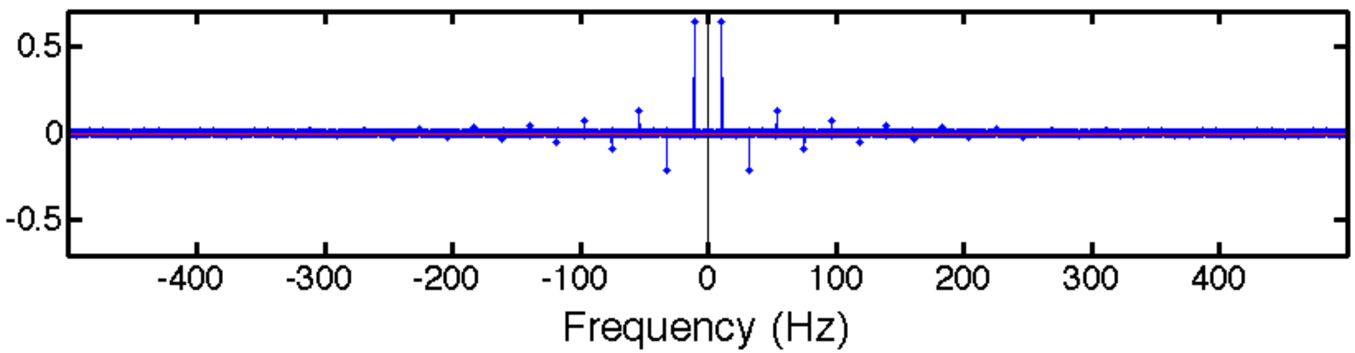
# Discrete Fourier transform

- Some examples – square waves

Continuous_square.m




$f_0$=10.75Hz



# Learning Objectives for Lecture 12

- Fourier Transform Pairs
- Convolution Theorem
- Gaussian Noise (Fourier Transform and Power Spectrum)
- Spectral Estimation
  - Filtering in the frequency domain
  - Wiener-Kinchine Theorem
- Shannon-Nyquist Theorem (and zero padding)
- Line noise removal

# Learning Objectives for Lecture 12

- Fourier Transform Pairs
- Convolution Theorem
- Gaussian Noise (Fourier Transform and Power Spectrum)
- Spectral Estimation
  - Filtering in the frequency domain
  - Wiener-Kinchine Theorem
- Shannon-Nyquist Theorem (and zero padding)
- Line noise removal

# Fourier transform pair

Square pulse

$$y(t) = \begin{cases} 1 \; if \; |t| < \Delta T / 2 \\ 0 \quad otherwise \end{cases}$$

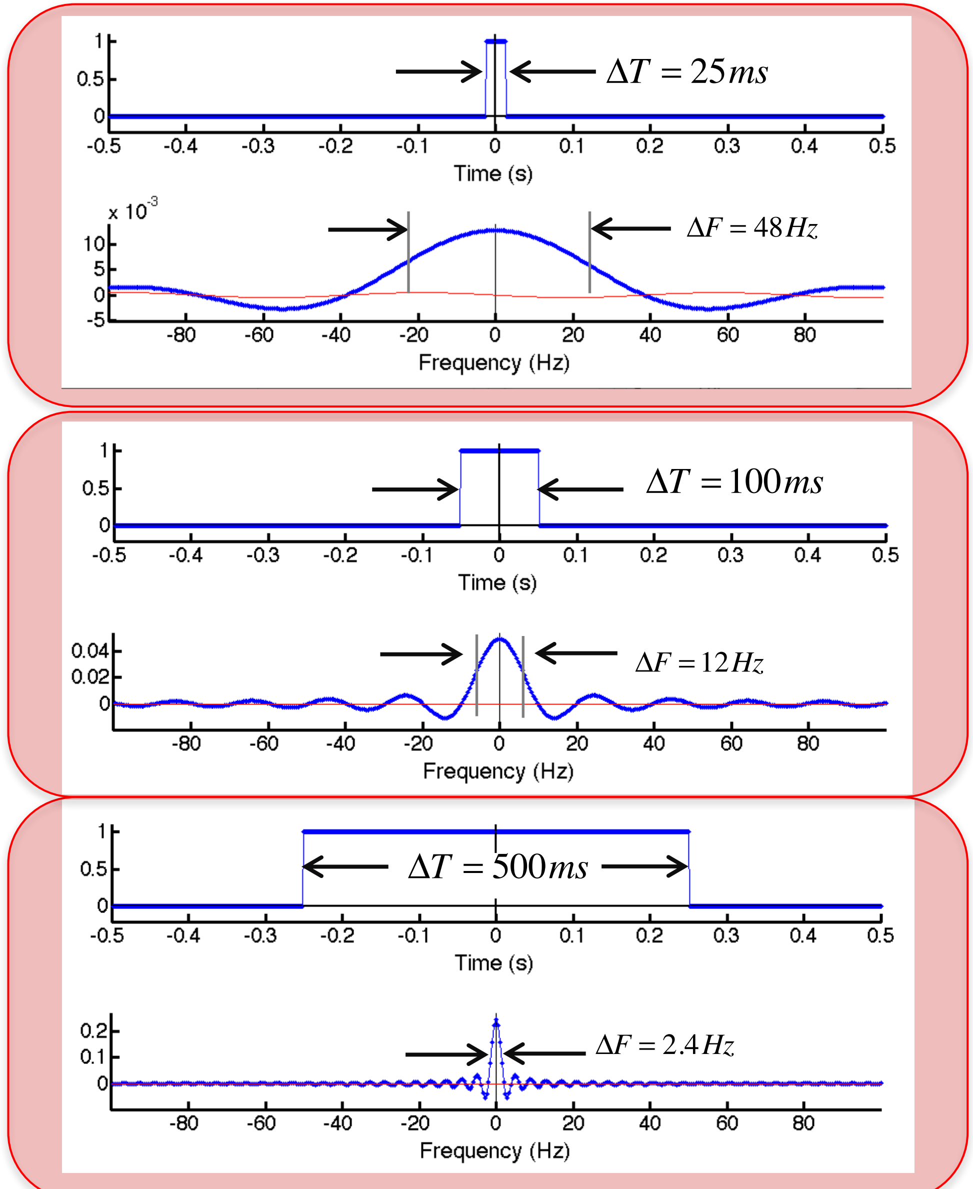
Sinc function

$$Y(f) = \Delta T \frac{\sin(\pi \Delta T f)}{\pi \Delta T f}$$

$$\Delta T \ , \ \Delta F \ \approx \ FWHM$$

$$\Delta F \approx \frac{1.2}{\Delta T}$$

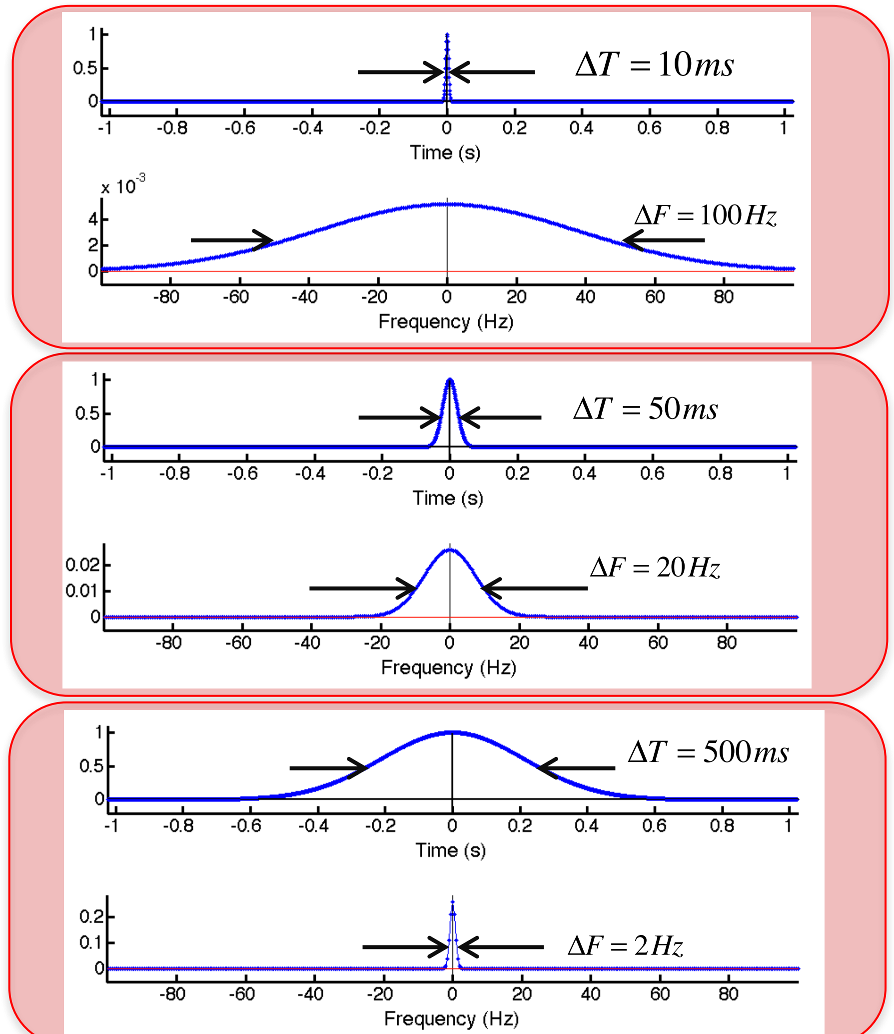
Square_window.m



# Fourier transform pair

The Fourier transform of a Gaussian
Is a Guassian!

Gaussian_window.m




$$\Delta F = \frac{1}{\Delta T}$$

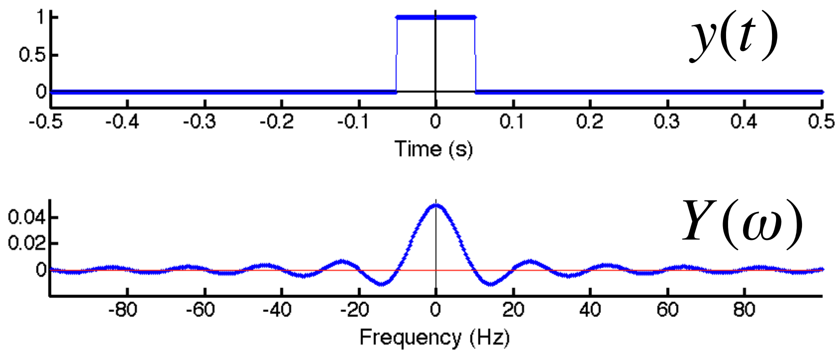
Time-bandwidth product

$$\Delta T \, \Delta F = 1$$

# Learning Objectives for Lecture 12

- Fourier Transform Pairs
- Convolution Theorem
- Gaussian Noise (Fourier Transform and Power Spectrum)
- Spectral Estimation
  - Filtering in the frequency domain
  - Wiener-Kinchine Theorem
- Shannon-Nyquist Theorem (and zero padding)
- Line noise removal

# Relation between Fourier transform and convolution




## Fourier Transform Pairs

$$y(t) \Leftrightarrow Y(\omega) \qquad g(\tau) \Leftrightarrow G(\omega) \qquad x(t) \Leftrightarrow X(\omega)$$

Convolution in the time domain

$$y(t) = \int_{-\infty}^{\infty} d\tau\, g(\tau) x(t-\tau)$$

Multiplication in the frequency domain

$$Y(\omega) = G(\omega) X(\omega)$$

# Fourier transform of a convolution

$$y(t) = \int_{-\infty}^{\infty} d\tau\, g(\tau) x(t-\tau)$$

$$Y(\omega) = G(\omega)\, X(\omega)$$

$$Y(\omega) = \int_{-\infty}^{\infty} dt\ y(t)\ e^{-i\omega t}$$

$$Y(\omega) = \int_{-\infty}^{\infty} dt \int_{-\infty}^{\infty} d\tau\, g(\tau) x(t-\tau)\ e^{-i\omega t}$$

$$= \int_{-\infty}^{\infty} d\tau \int_{-\infty}^{\infty} dt\, g(\tau) x(t-\tau) e^{-i\omega t}$$

$$= \int_{-\infty}^{\infty} d\tau g(\tau) \int_{-\infty}^{\infty} dt\, x(t-\tau) e^{-i\omega t}$$

$$= \int_{-\infty}^{\infty} d\tau g(\tau) \int_{-\infty}^{\infty} dt\, x(t-\tau) e^{-i\omega(t-\tau)} e^{-i\omega\tau}$$

$$= \int_{-\infty}^{\infty} d\tau g(\tau) e^{-i\omega\tau} \boxed{\int_{-\infty}^{\infty} dt\, x(t-\tau) e^{-i\omega(t-\tau)}}$$

$$= \int_{-\infty}^{\infty} d\tau g(\tau) e^{-i\omega\tau}\ X(\omega)$$

$$= G(\omega) X(\omega)$$

# Relation between Fourier transform and convolution

Fourier Transform Pairs

$$y(t) \Leftrightarrow Y(\omega) \qquad g(\tau) \Leftrightarrow G(\omega) \qquad x(t) \Leftrightarrow X(\omega)$$

Convolution in the time domain

$$y(t) = \int_{-\infty}^{\infty} d\tau \, g(\tau) x(t-\tau)$$

Multiplication in the frequency domain

$$Y(\omega) = G(\omega) X(\omega)$$

Convolution in the frequency domain

$$Y(\omega) = \int_{-\infty}^{\infty} d\omega' G(\omega') X(\omega - \omega')$$

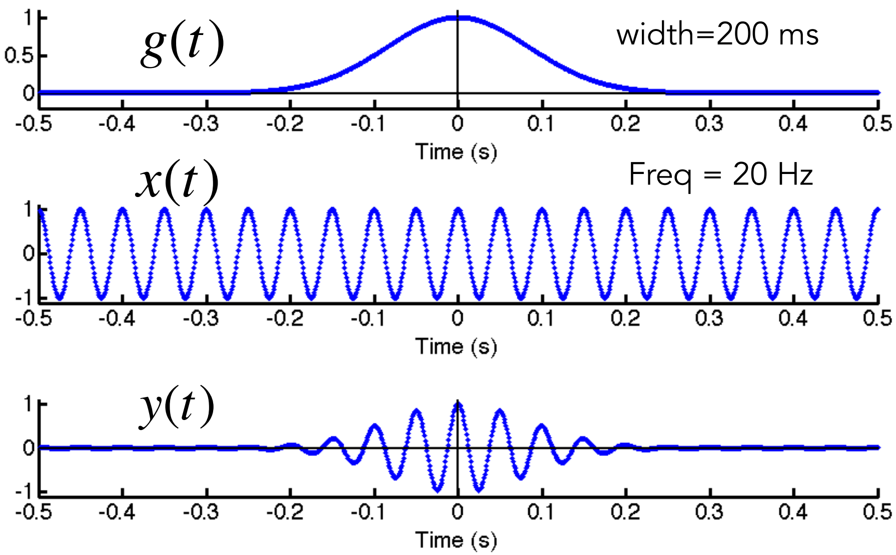
Multiplication in the time domain

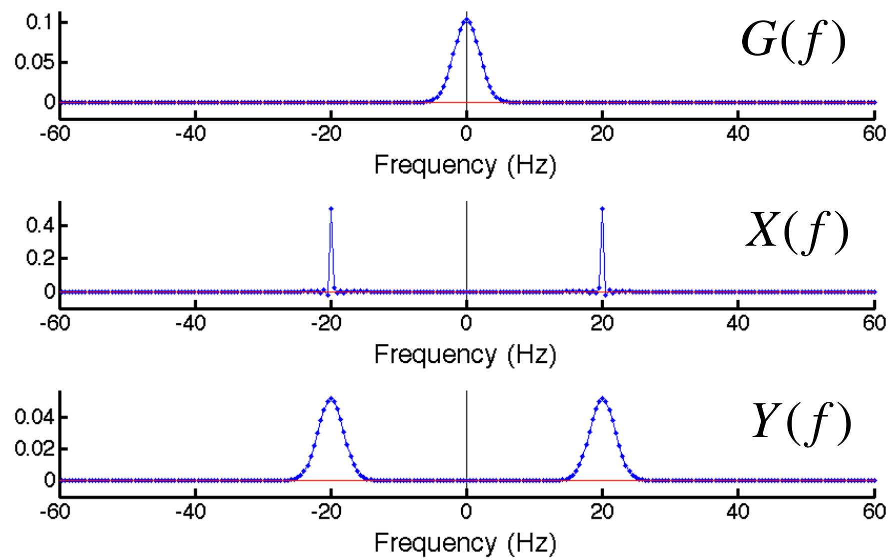
$$y(t) = g(t)\, x(t)$$

# Using the Convolution Theorem

## Gaussian-windowed cosine

Cos_Gauss_pulse.m




$$y(t) = g(t)\,x(t)$$

Product in the time-domain

$$Y(f) = G(f) * X(f)$$

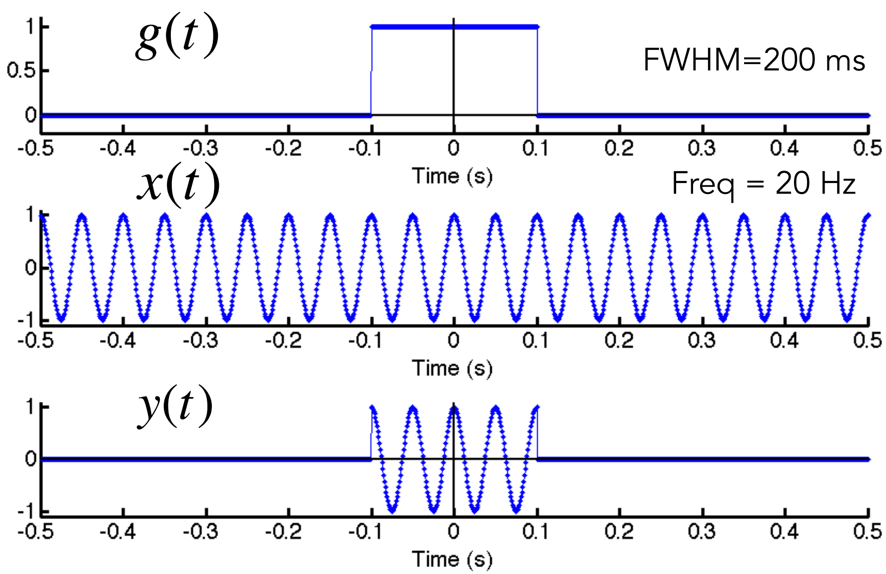
Convolution in the frequency-domain!

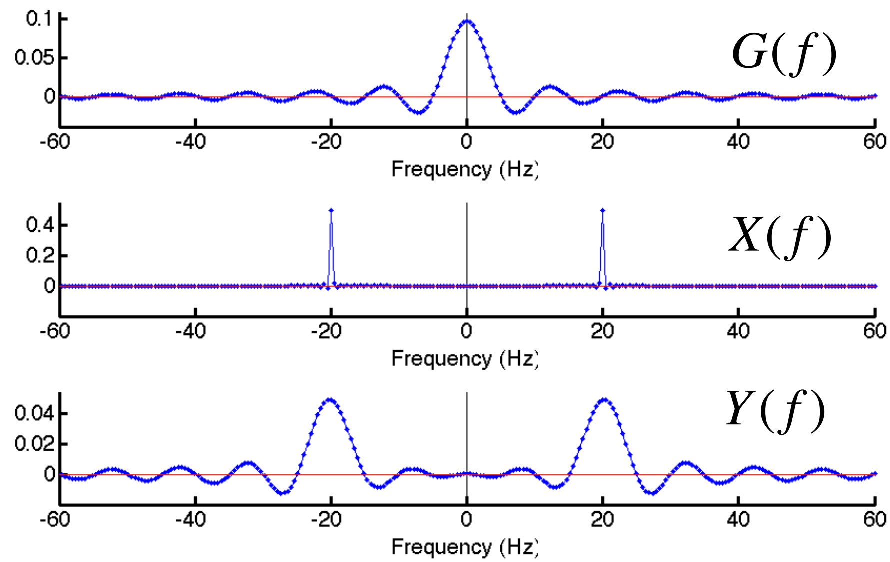
# Using the Convolution Theorem

## Square-windowed cosine

$g(t) = square \qquad x(t) = \cos(2\pi f_0 t)$

cos_Gauss_pulse.m




$$y(t) = g(t)x(t)$$

Product in the time-domain

$$Y(f) = G(f) * X(f)$$

Convolution in the frequency-domain!

# Learning Objectives for Lecture 12

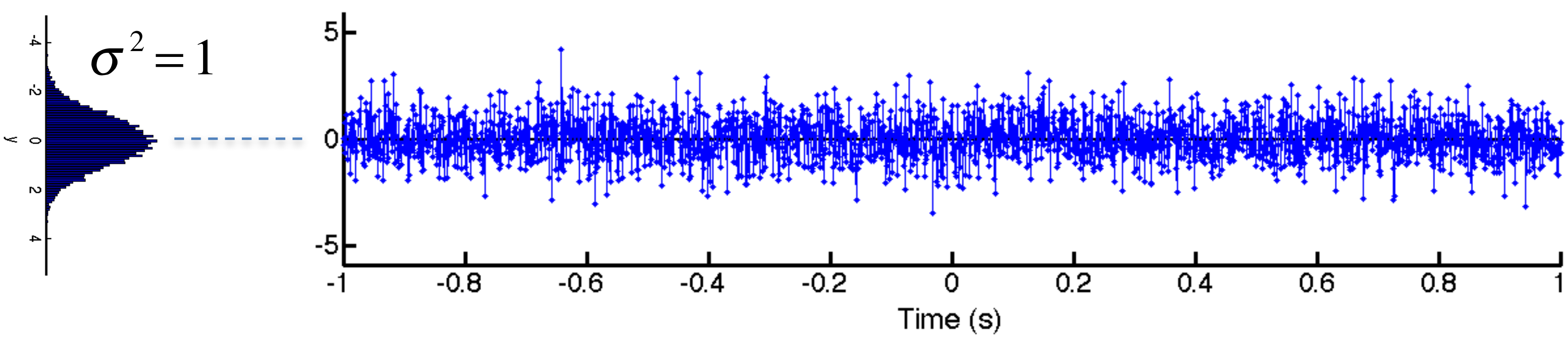
- Fourier Transform Pairs
- Convolution Theorem
- Gaussian Noise (Fourier Transform and Power Spectrum)
- Spectral Estimation
  - Filtering in the frequency domain
  - Wiener-Kinchine Theorem
- Shannon-Nyquist Theorem (and zero padding)
- Line noise removal

# Gaussian noise

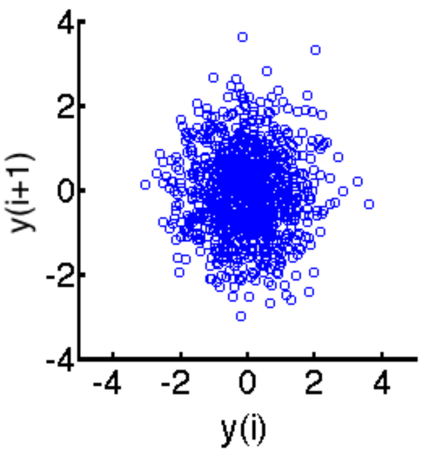
- Each sample drawn independently from a Gaussian distribution

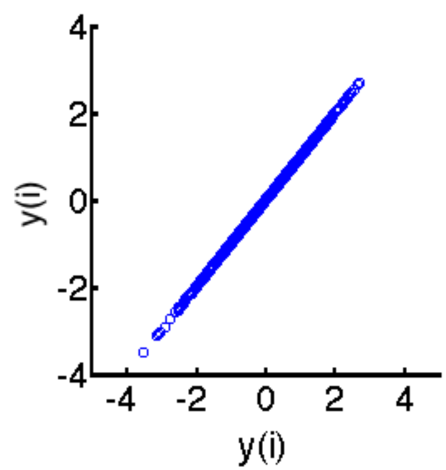
y=randn(1,N);

white_noise.m

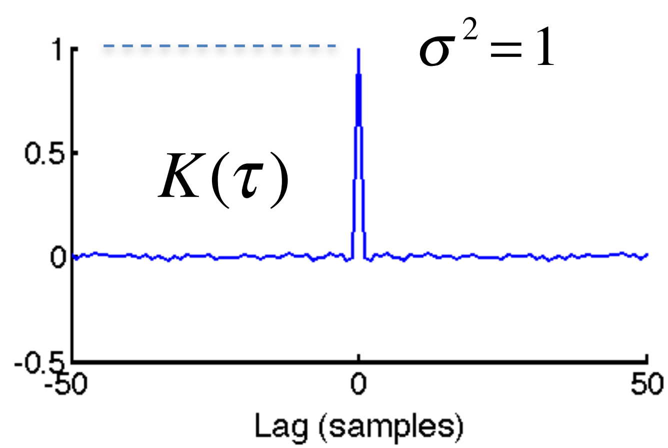



play_tone_in_noise.m




Autocorrelation

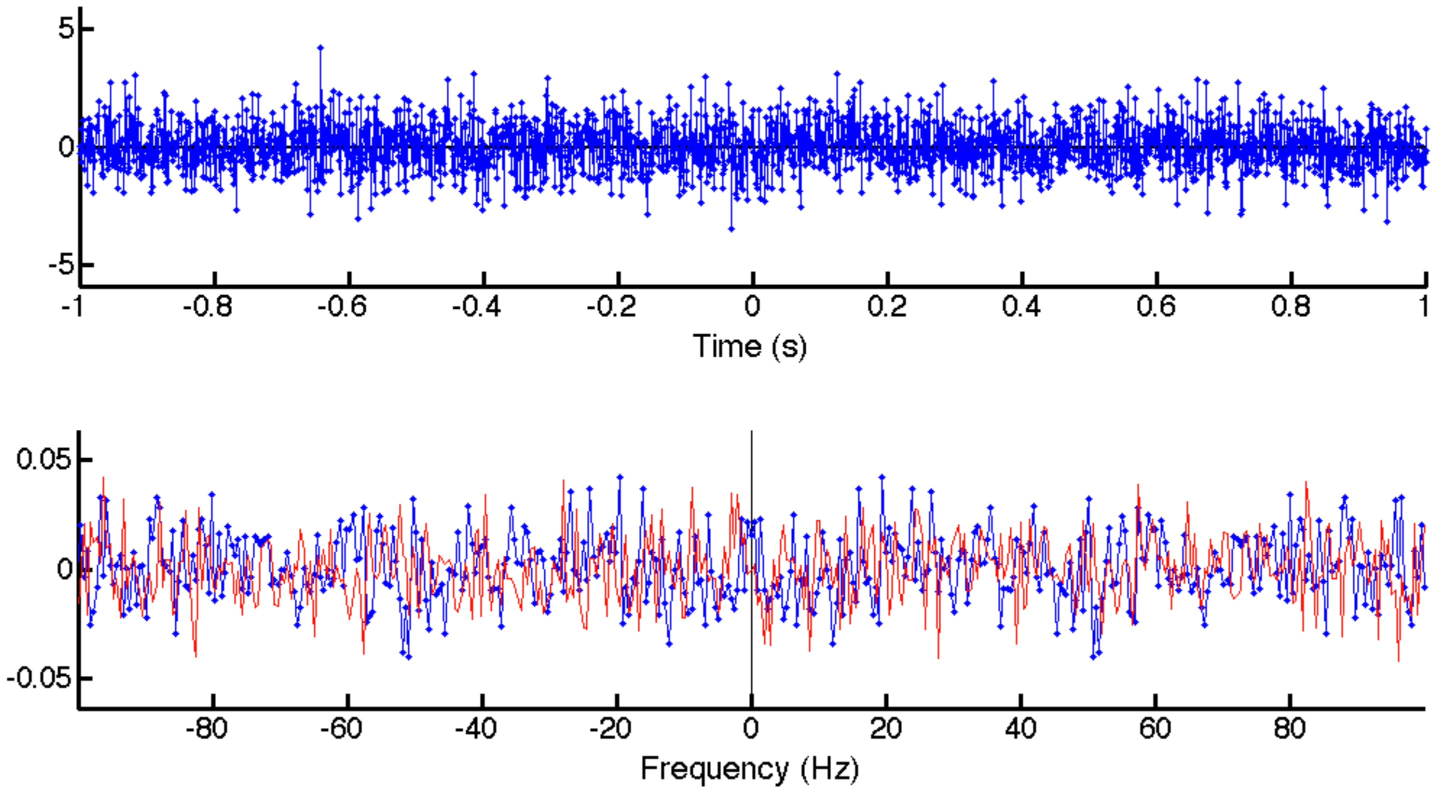



randn_dist.m

# Fourier transform of Gaussian noise

- Some examples – Gaussian white noise

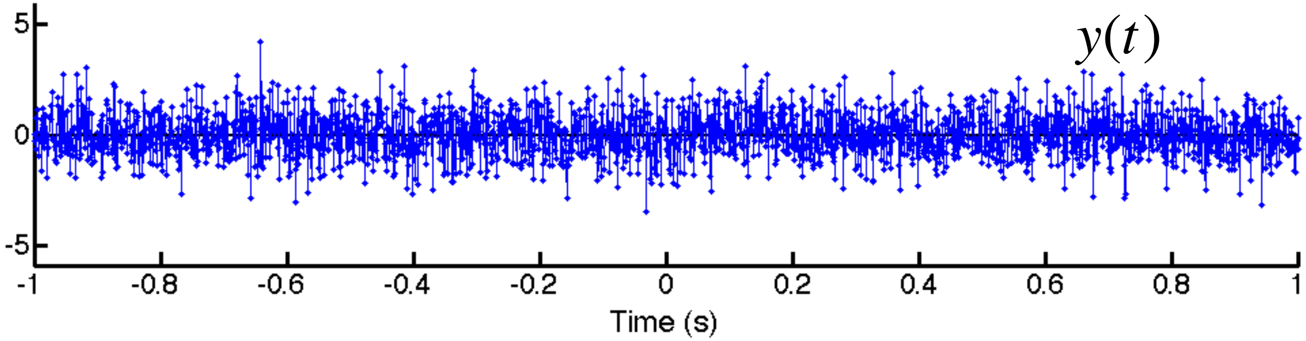
white_noise.m

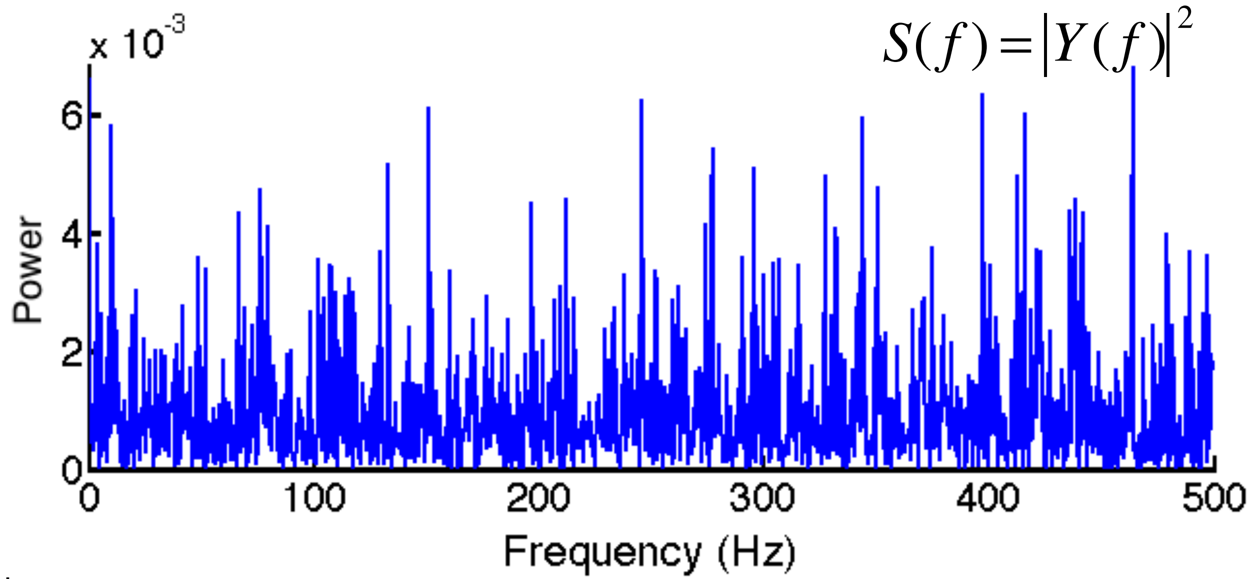



The Fourier transform of Gaussian noise, is just Gaussian noise!

# Power spectrum of noise

- Is very…




white_noise.m

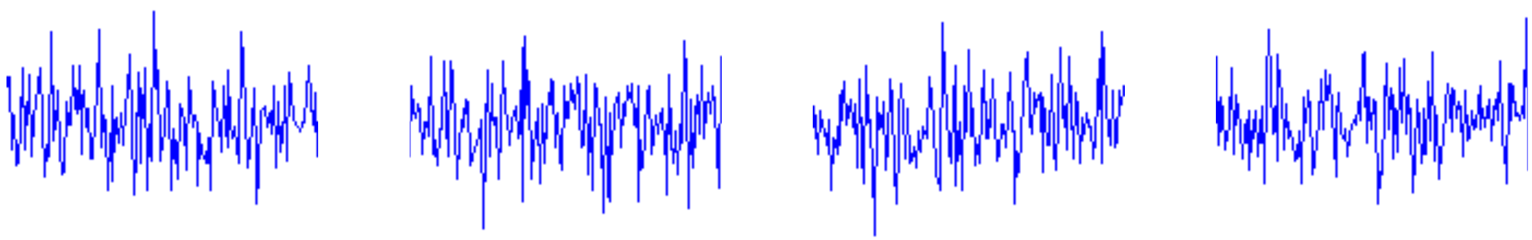
# Learning Objectives for Lecture 12

- Fourier Transform Pairs
- Convolution Theorem
- Gaussian Noise (Fourier Transform and Power Spectrum)
- Spectral Estimation
  - Filtering in the frequency domain
  - Wiener-Kinchine Theorem
- Shannon-Nyquist Theorem (and zero padding)
- Line noise removal

# Spectral estimation

- Say we want to find the spectrum $S(f)$ of a signal y(t).
- Often we only have short measurements of y(t) (e.g. trials)

Trial 1 Trial 2 Trial 3 Trial 4



…

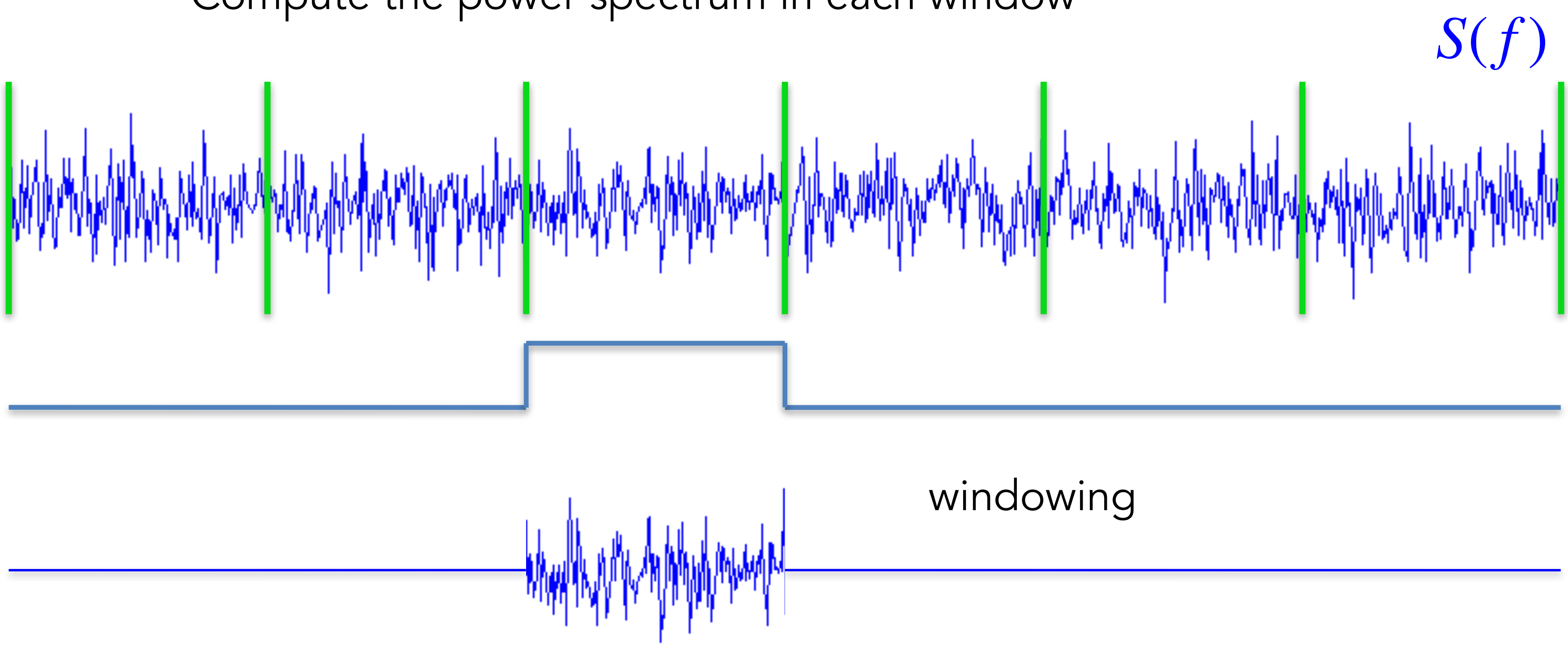
$\hat{S}_1(f)$ $\hat{S}_2(f)$ $\hat{S}_3(f)$ $\hat{S}_4(f)$

We can just average!

$$\hat{S}(f) = \frac{1}{N}\sum_{i=1}^{N}\hat{S}_i(f)$$

# Spectral estimation

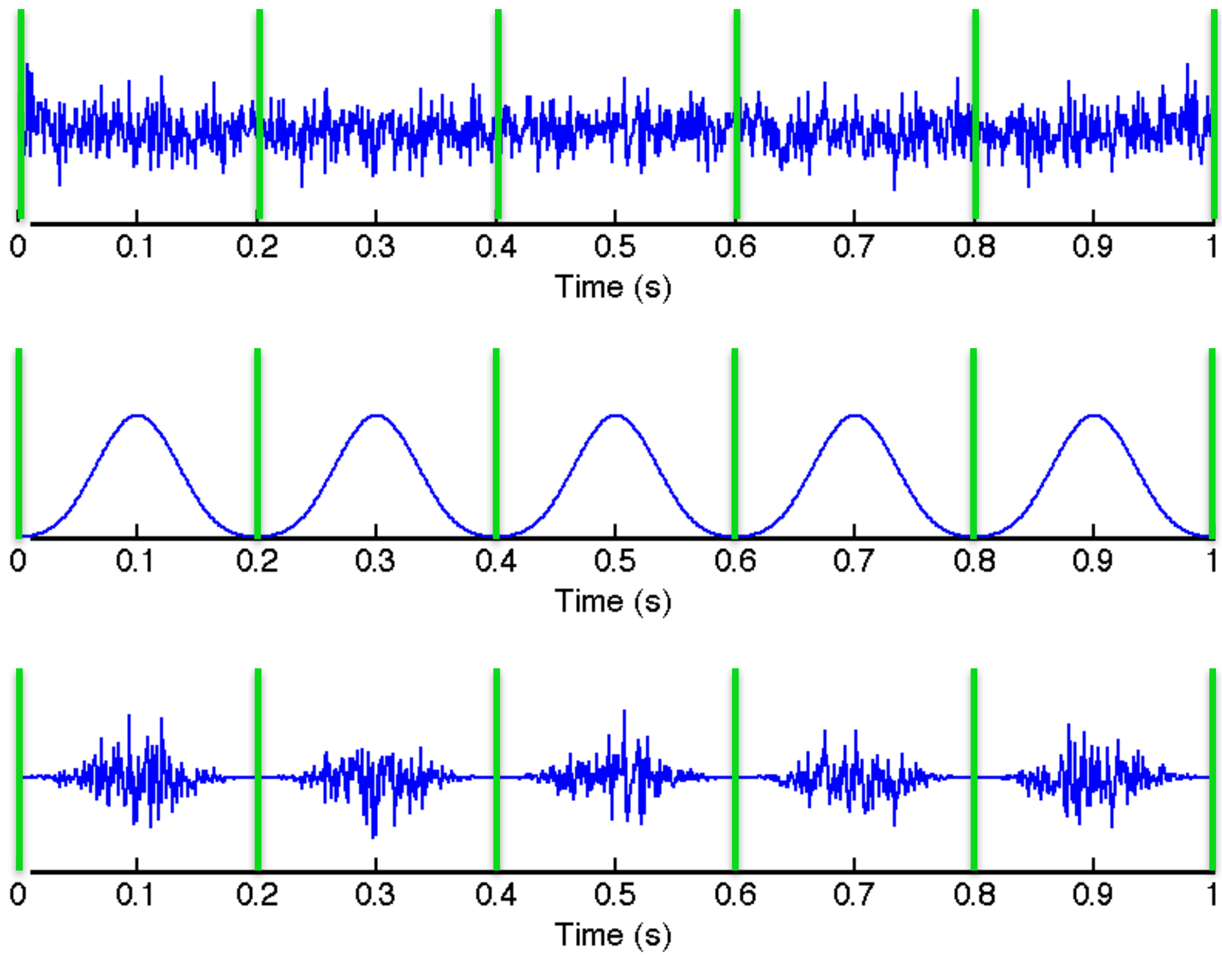
- The same principle applies to longer signals.
  - Break the signal into shorter pieces
  - Compute the power spectrum in each window




- We could just take the FFT of each piece.
  - But we know that a 'square windowing' means that the spectrum becomes convolved with the spectrum of the square window!

# Spectral estimation

- We will multiply each window by a smooth function called a 'taper'.

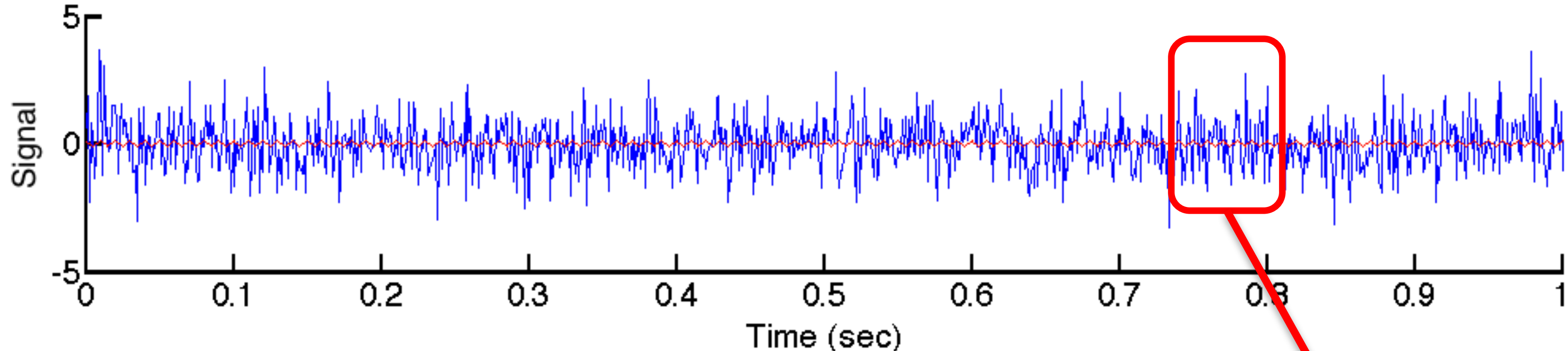


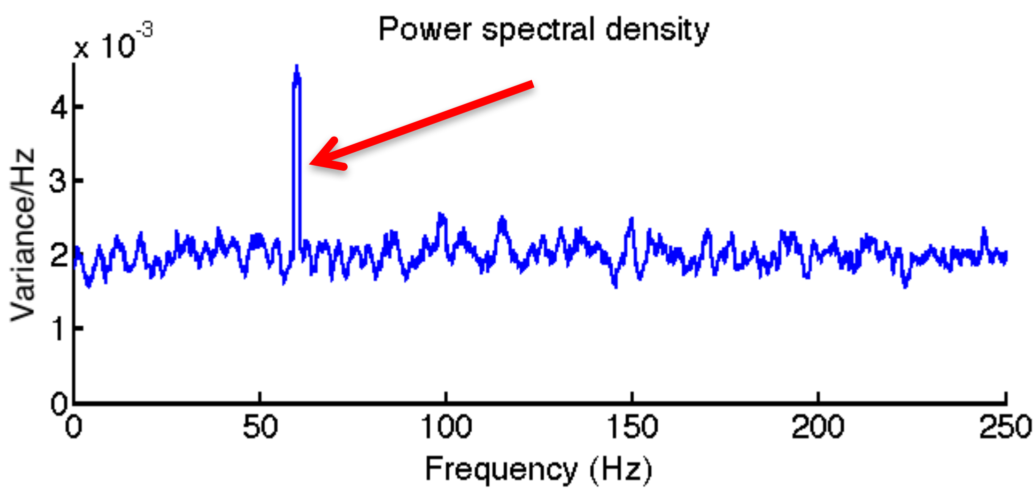

$$\hat{S}(f) = \frac{1}{N}\sum_{i=1}^{N}\hat{S}_i(f)$$

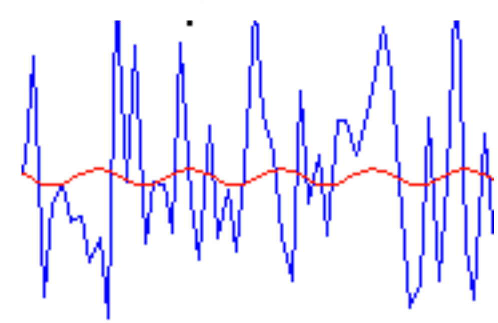
$\hat{S}_1(f)$ $\hat{S}_2(f)$ $\hat{S}_3(f)$ $\hat{S}_4(f)$ $\hat{S}_5(f)$

# Spectral estimation

- A common problem is to find a small signal in noise
  - This can be a challenge



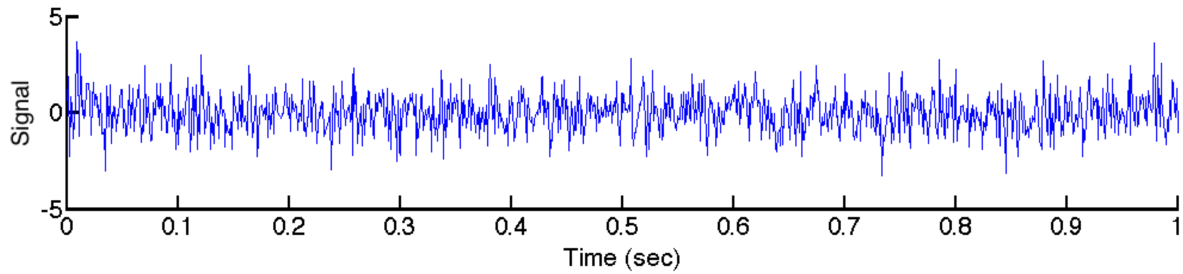



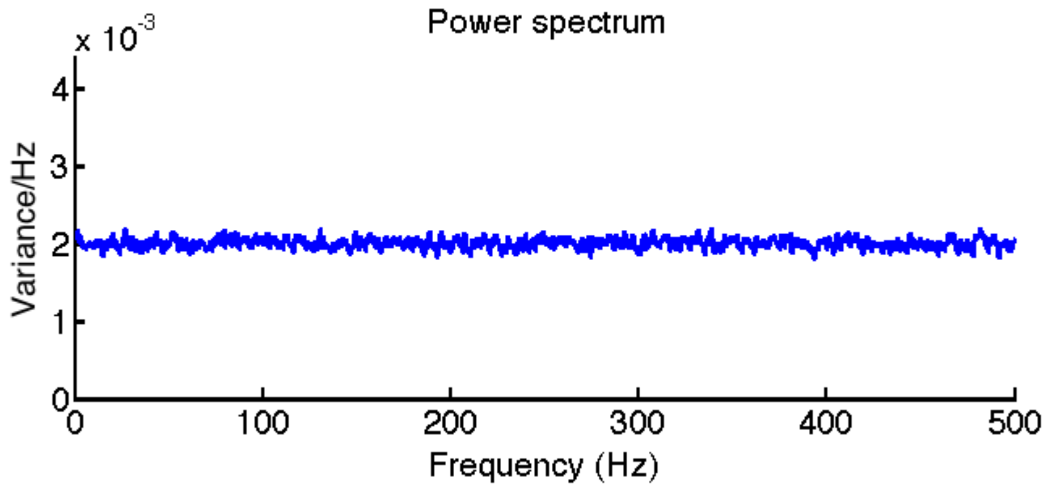
$$y(t) = 0.1 * \sin(2\pi f_0 t)$$

Powerspec_osc_in_noise.m

# Power spectrum of noise

- But the average spectrum of white noise is flat!




Powerspec_noise.m

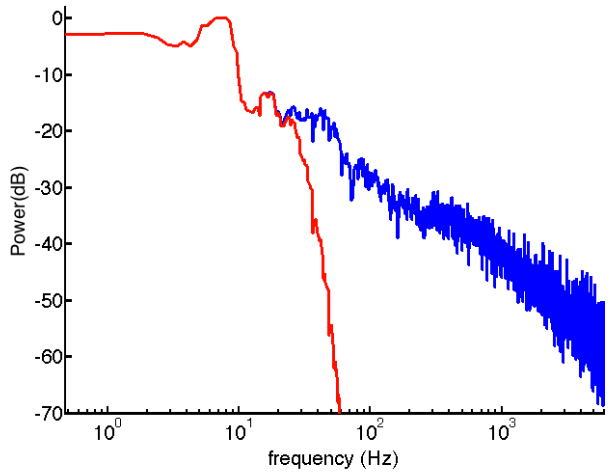
The power spectrum is a power spectral density. It tells us the amount of variance per unit frequency. Variance is power density times bandwidth.

$$\sigma^2 = 0.002 \frac{V^2}{Hz} x \ 500 Hz = 1$$

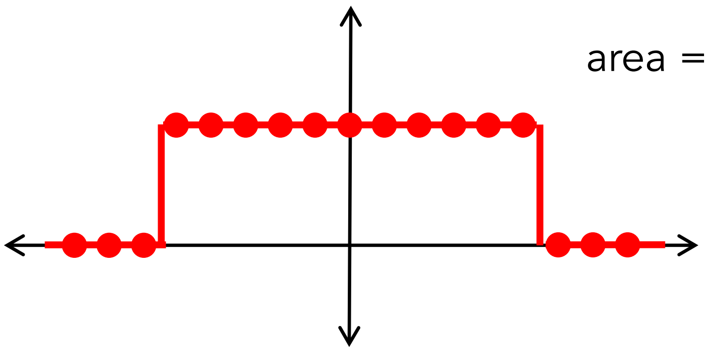
# Learning Objectives for Lecture 12

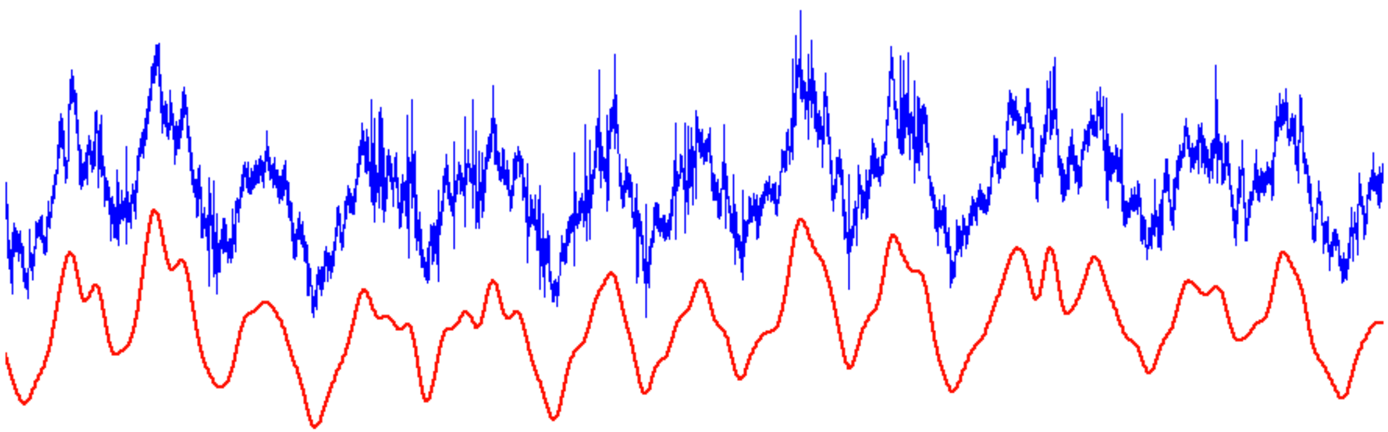
- Fourier Transform Pairs
- Convolution Theorem
- Gaussian Noise (Fourier Transform and Power Spectrum)
- Spectral Estimation
  - Filtering in the frequency domain
  - Wiener-Kinchine Theorem
- Shannon-Nyquist Theorem (and zero padding)
- Line noise removal

# Low-pass filtering




Low-pass filtering can be done by convolving the signal with a kernel like this.



# Convolution as a filter

- If the convolution is equivalent to a multiplication in the frequency domain…

$$Y(\omega) = G(\omega)X(\omega)$$

what does this do to the power spectrum?

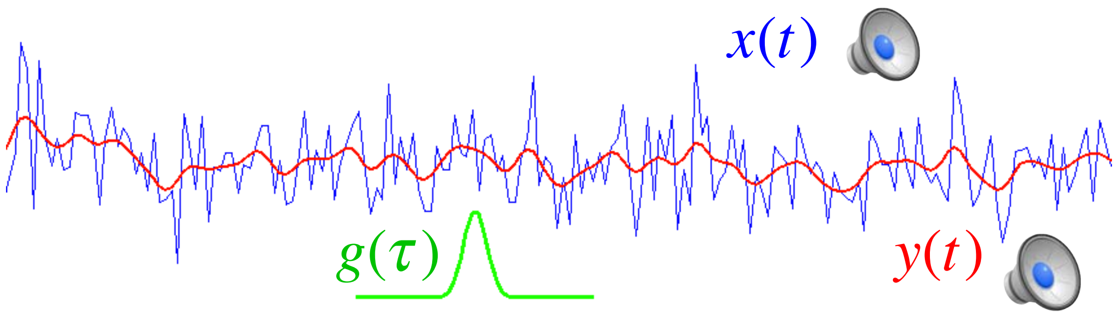
$$S(\omega) = |Y(\omega)|^2 = |G(\omega)|^2 |X(\omega)|^2$$

- The power spectrum of the filtered signal is just the power spectrum of the original signal times the spectrum of the filter!

# Power spectrum of filtered noise

- Power spectrum of white noise convolved with a Gaussian window

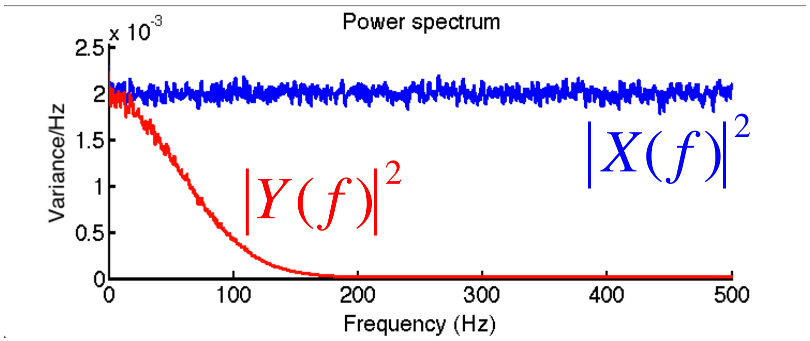
play_tone_in_noise.m

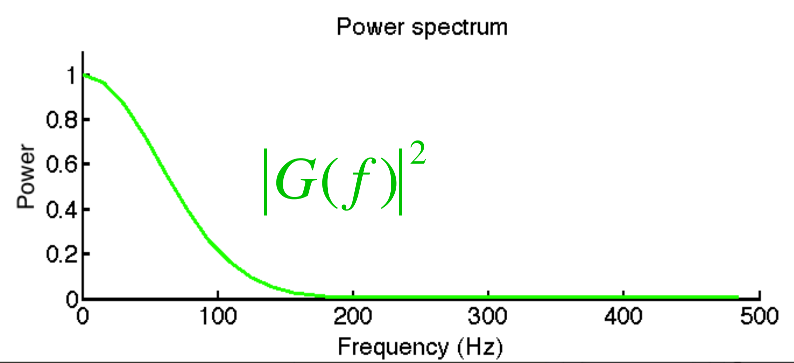


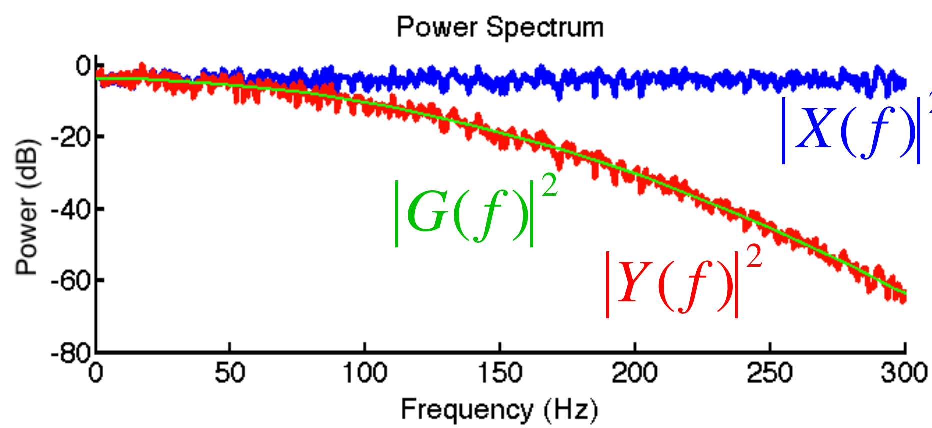

$$y(t) = \int_{-\infty}^{\infty} d\tau\, g(\tau) x(t-\tau)$$

$$Y(f) = G(f)X(f)$$

$$|Y(f)|^2 = |G(f)|^2 |X(f)|^2$$



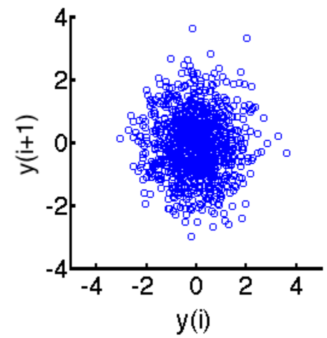



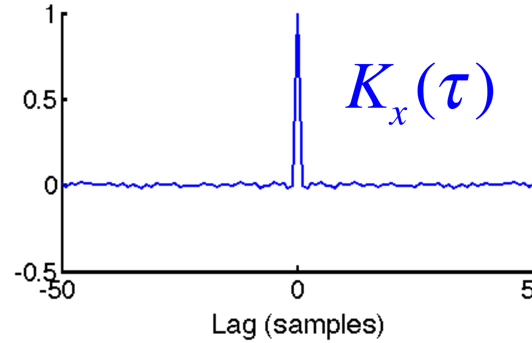




Powerspec_FilteredNoise.m

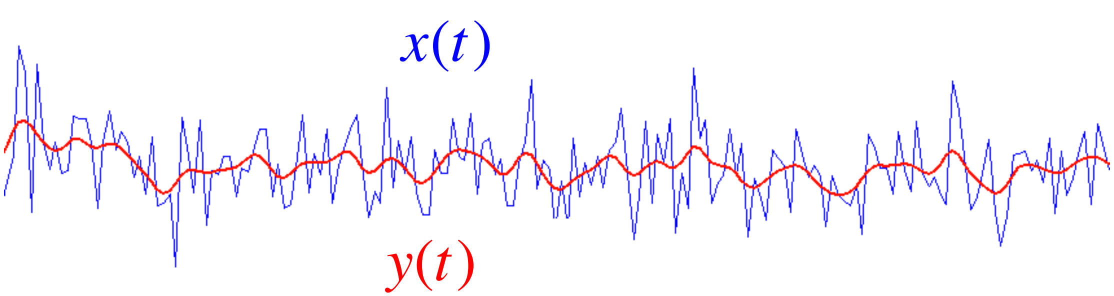
# Autocorrelation of filtered noise

Autocorrelation

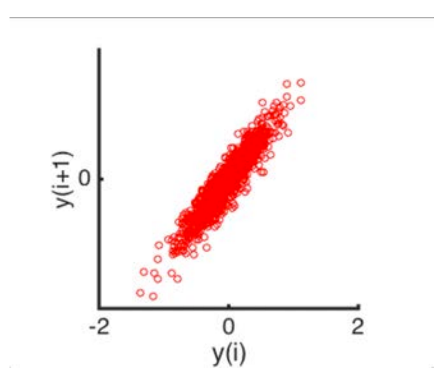


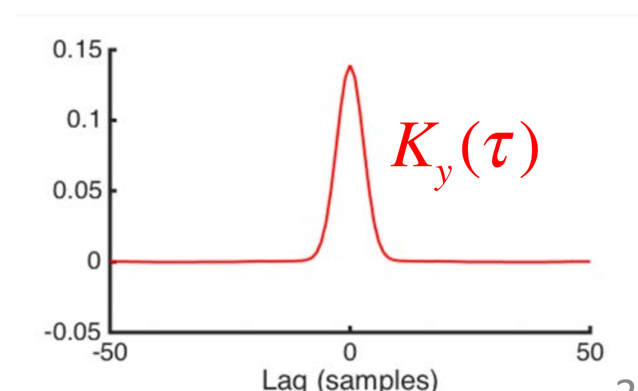




$$K_x(\tau) = \int_{-\infty}^{\infty} dt\, x(t)x(t+\tau)$$




Autocorrelation



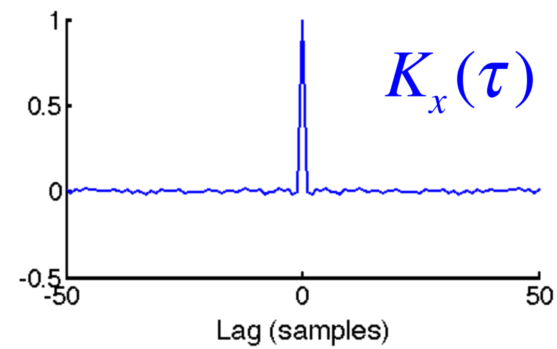



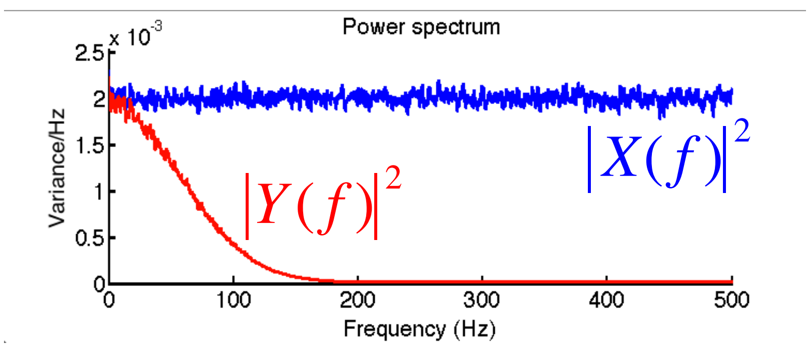
# Wiener-Kinchine theorem

- The power spectrum and autocorrelation functions are related by Fourier Transform

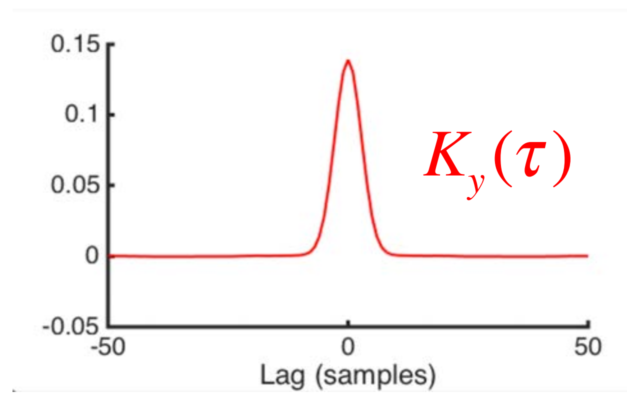
Autocorrelation

$$S_x(f) = |X(f)|^2 \quad \xleftrightarrow{\text{FT}} \quad K_x(\tau)$$




Autocorrelation



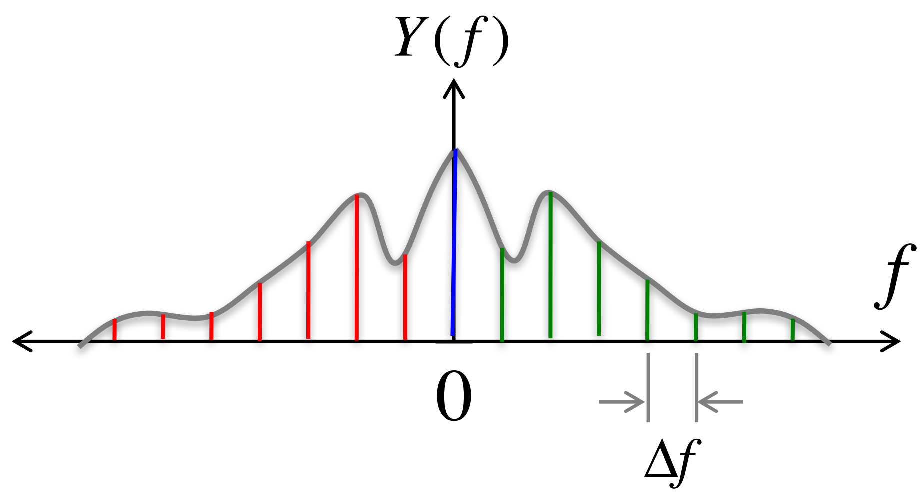

$$S_y(f) = |Y(f)|^2 \quad \xleftrightarrow{\text{FT}} \quad K_y(\tau)$$

# Learning Objectives for Lecture 12

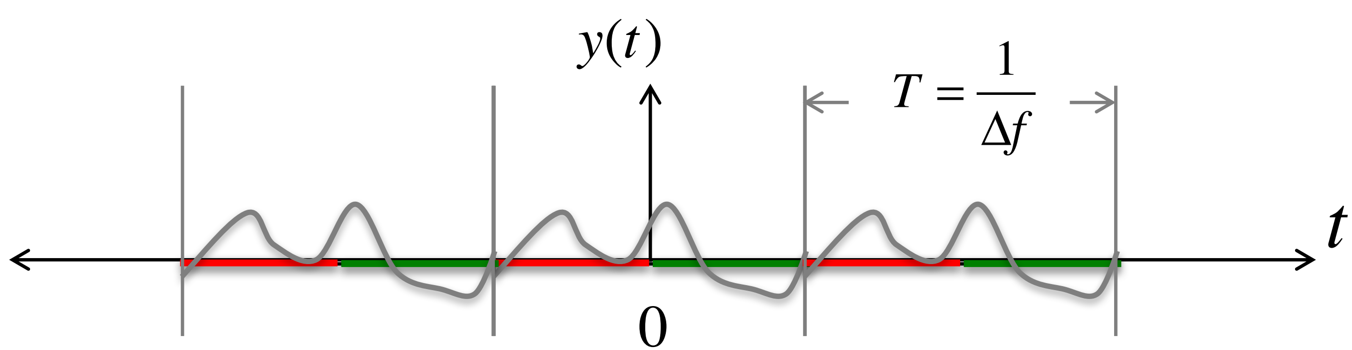
- Fourier Transform Pairs
- Convolution Theorem
- Gaussian Noise (Fourier Transform and Power Spectrum)
- Spectral Estimation
  - Filtering in the frequency domain
  - Wiener-Kinchine Theorem
- Shannon-Nyquist Theorem (and zero padding)
- Line noise removal

# Nyquist-shannon theorem

- Remember that frequency is discretized…

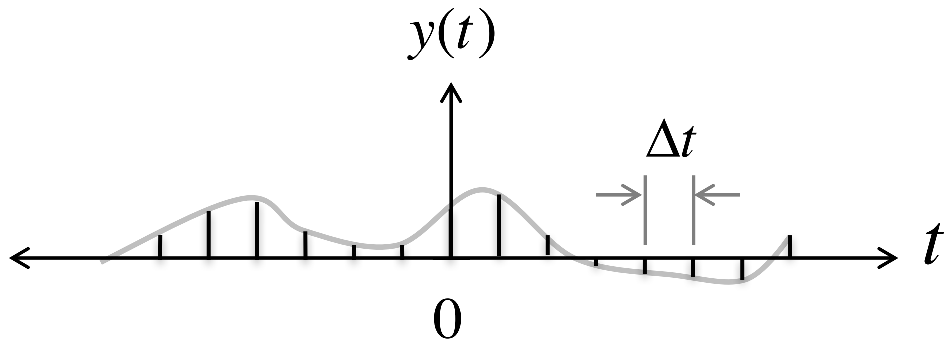



- Which means that our function is periodic in time!

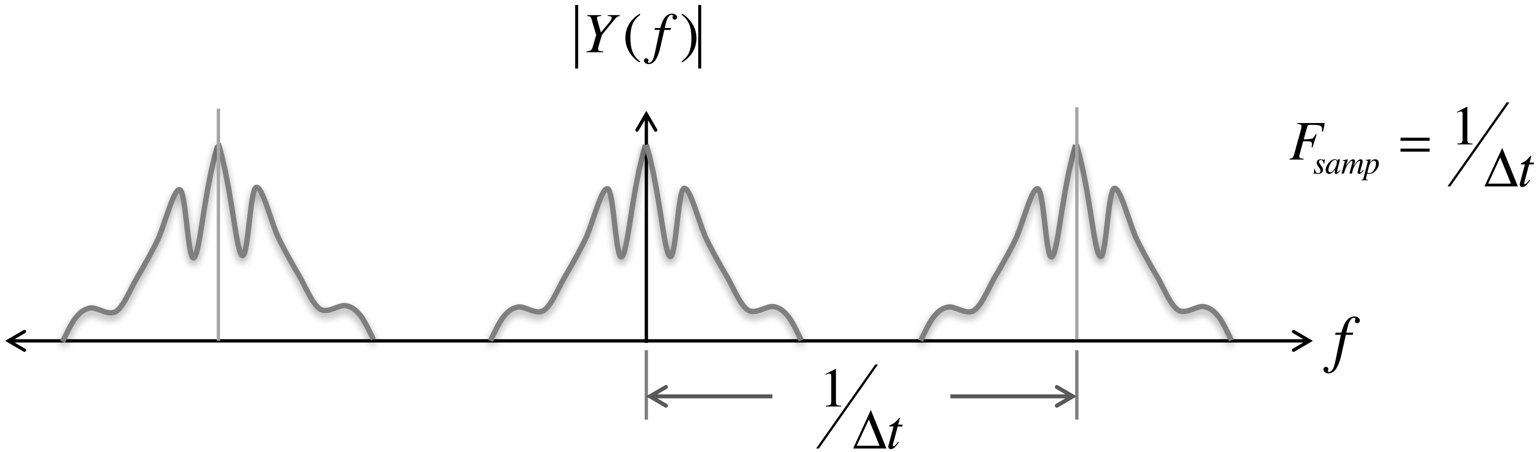


# Nyquist-shannon theorem

- But time is also discretized…




- Which means that our FFT is periodic in frequency!

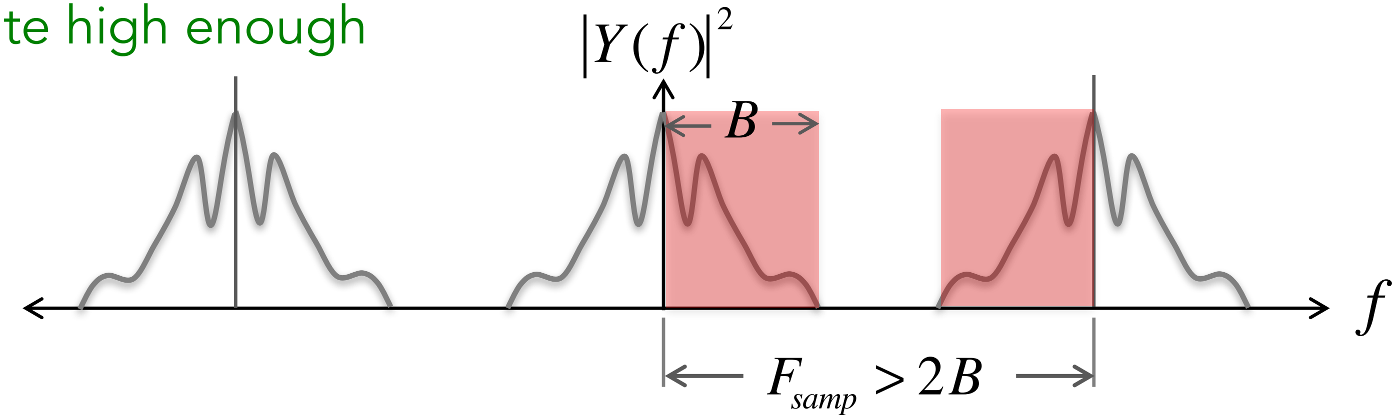



… and the separation between the spectra in the frequency-domain is given by the sampling rate.

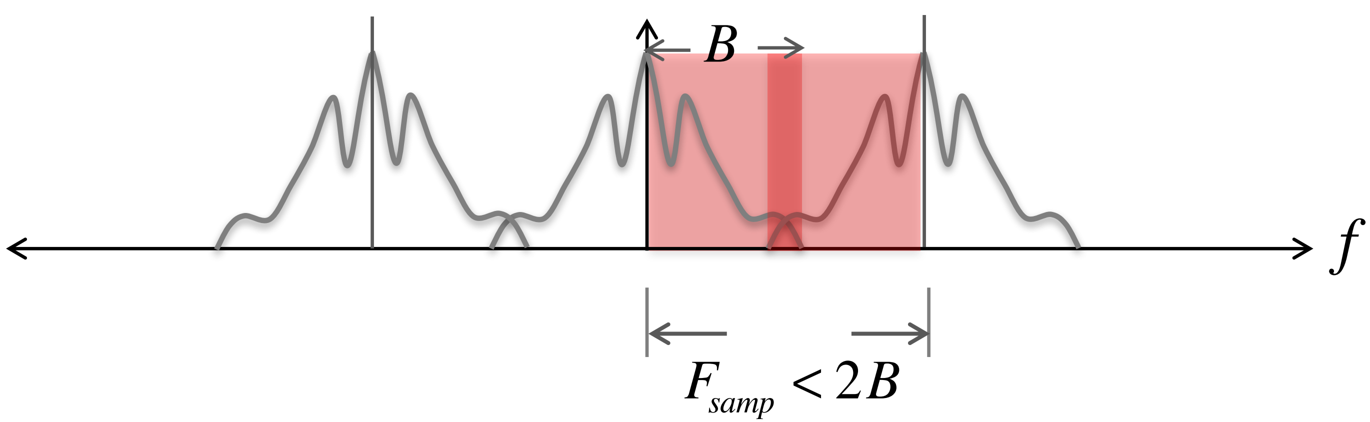
# Nyquist-shannon theorem

- Sampling rate must be greater than twice the bandwidth of the signal $F_{samp} > 2B$.

Sampling rate high enough

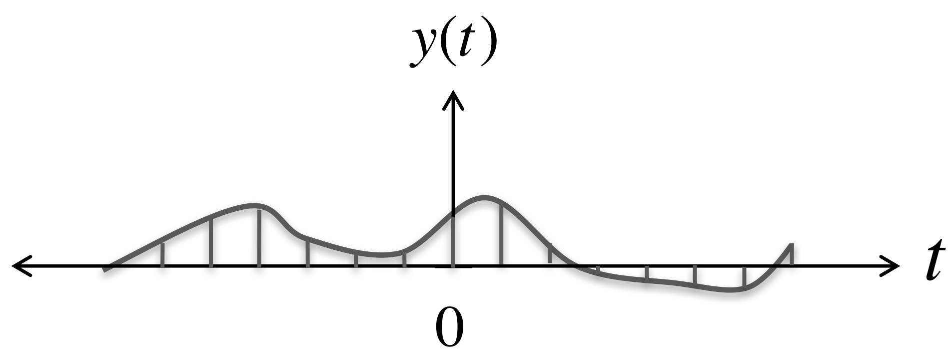



Sampling rate too low



# Nyquist-shannon theorem

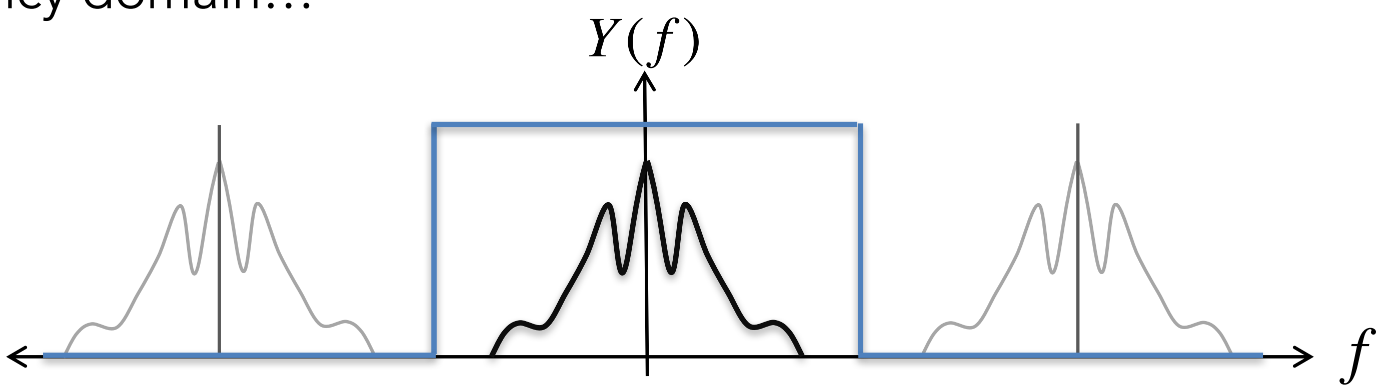
- If the sampling rate is greater than twice the bandwidth of the signal $F_{samp} > 2B$
- Then you can perfectly reconstruct the original signal.
- Not just at the sampled points, but continuously, at every point!



# Nyquist-shannon theorem

- Remember the Convolution Theorem…

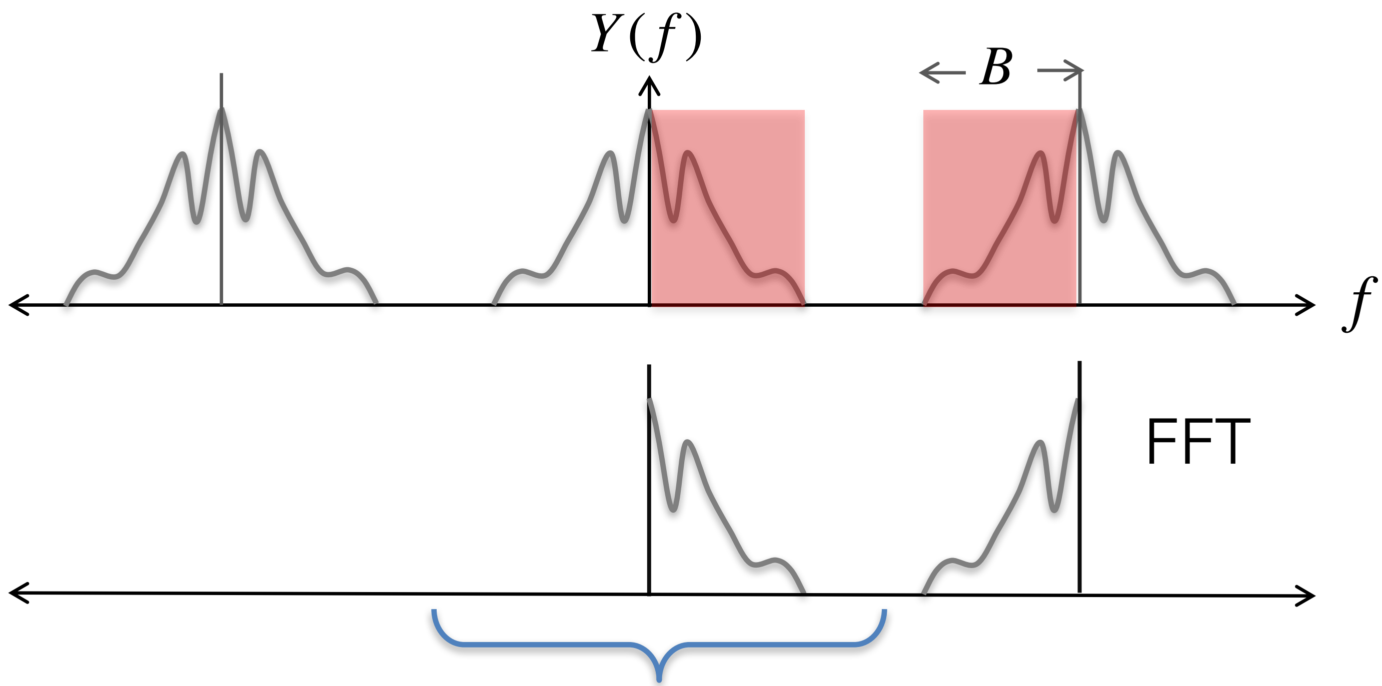
- Multiply the periodic Fourier transform by a square window in the frequency domain…




- This convolves the time-domain signal with a Kernel that is the Fourier transform of the square window… In other words, we can smooth it, so that it is no longer sampled. The smoothed signal has the same spectrum as the sampled signal!

# Zero-padding

- If $F_{samp} > 2B$ you can interpolate the values of the function y(t) with arbitrarily high temporal resolution by zero padding.




'Pad' with zeros!

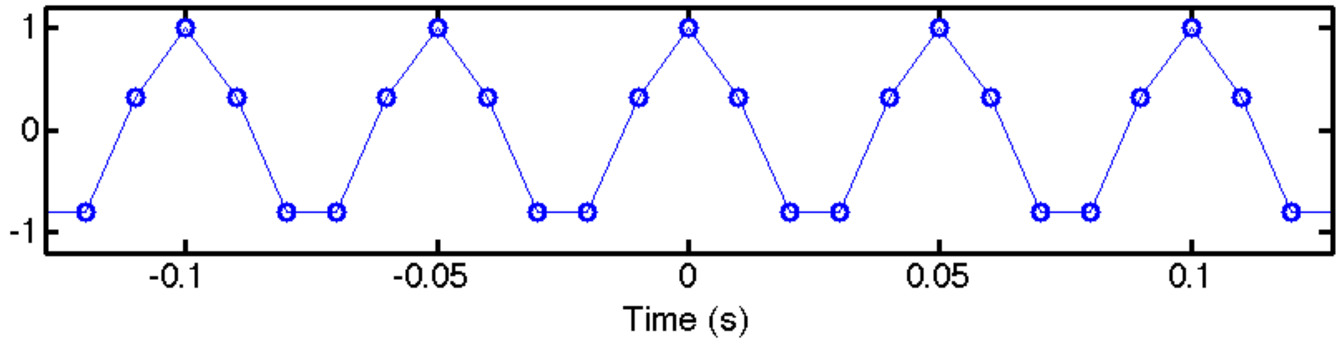
$$F_{samp} = N\,\Delta f$$

And then inverse FFT !

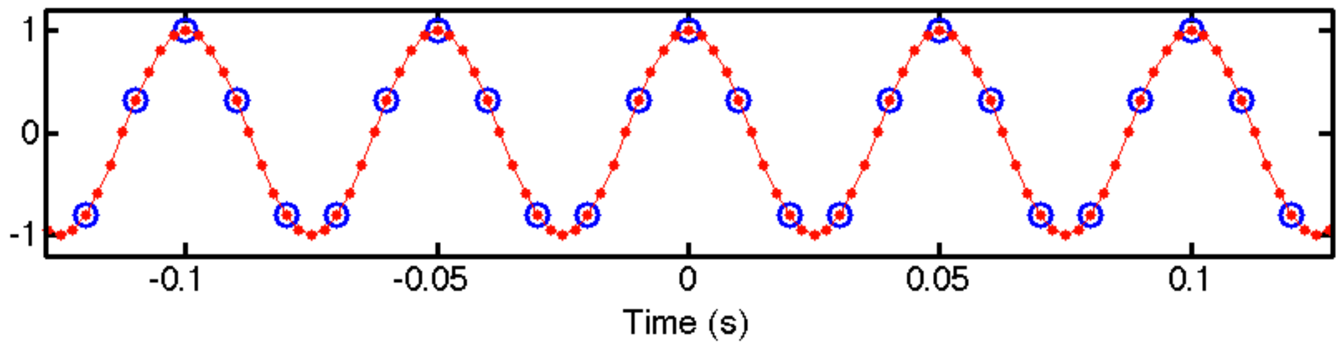
$$F_{samp} = 1/\Delta t$$

# Zero-padding

20Hz cos wave




$$F_{samp} = 100\,Hz$$
$$\Delta t = 10\,ms$$




$$F_{samp} = 400\,Hz$$
$$\Delta t = 2.5\,ms$$

```
Y=fft(y, N)/N;                          % Compute the FFT
zero_pad_factor=4.;
Nresamp=N*zero_pad_factor;              % new number of frequency bins
Yresamp=zeros(1,Nresamp);               % fill a new array with zeros
Yresamp(1:N/2)=Y(1:N/2);                % insert the positive frequency part
yRes=2*real(ifft(Yresamp)*Nresamp);     % compute the inverse FFT
```

# Zero-padding

## Or Fun with FFT and IFFT

- Zero padding in the time domain gives finer spacing in the frequency domain.
- Just add zeros to the end of a tapered window before you FFT. The spectrum that you get is the same, but it has samples that are more finely spaced. (No higher frequency resolution though.)
- MATLAB® makes this easy…

```
zero_pad_factor=4.;
y=Data;                        % Data is a vector in time with N samples
Nfft=N*zero_pad_factor;        % number of points you want in the spectrum
Y=fft(y, Nfft);                % Compute the FFT
                               % The array Y will now have 4 times the samples
```

# Learning Objectives for Lecture 12

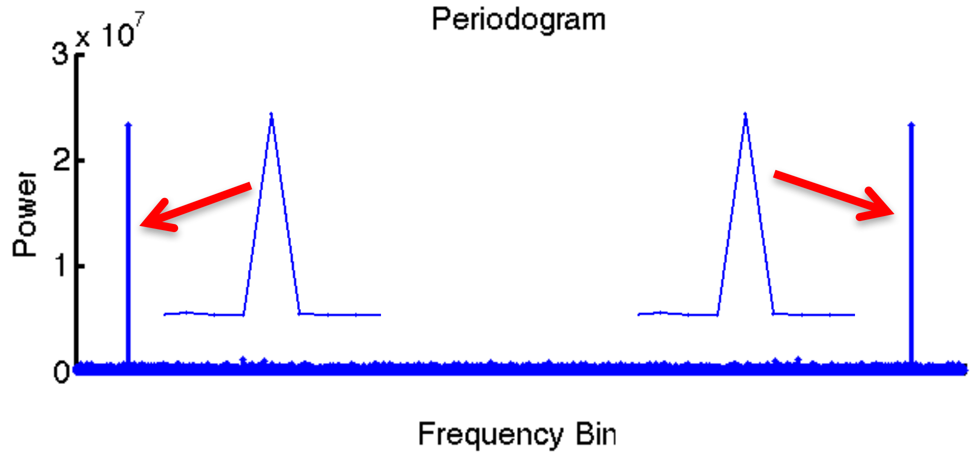
- Fourier Transform Pairs
- Convolution Theorem
- Gaussian Noise (Fourier Transform and Power Spectrum)
- Spectral Estimation
  - Filtering in the frequency domain
  - Wiener-Kinchine Theorem
- Shannon-Nyquist Theorem (and zero padding)
- Line noise removal

# Line noise removal

- Another common problem is to remove a small periodic noise in your signal.

Periodogram

$$S(f) = |Y(f)|^2$$



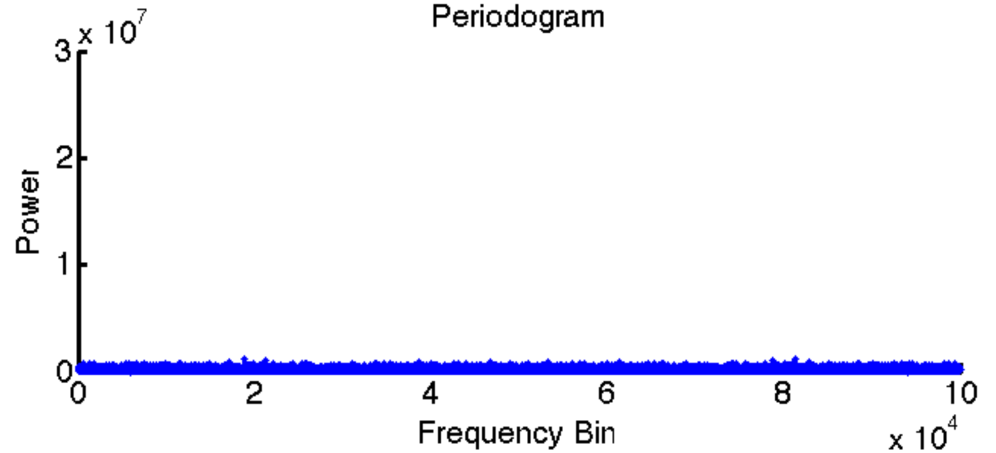

- While the periodogram is a terrible spectral estimator for non-periodic broadband signals, it is a great estimator for perfectly stationary single-frequencies… like contamination from 60Hz.

- So, if you have a single offending frequency component…

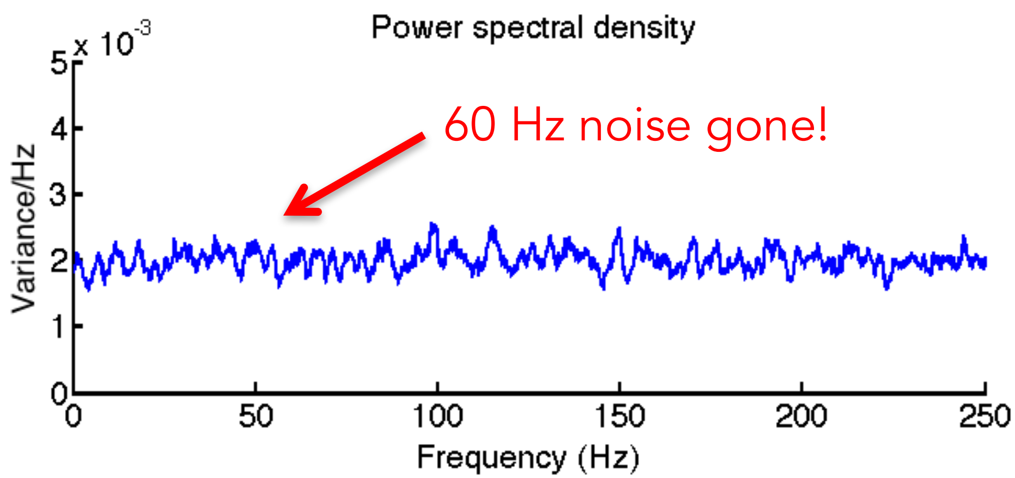
Off with its head!

# Line noise removal

- Just find those lines in Y(f) and set them to zero!




- Then inverse FFT Y(f) to get the cleaned up signal…



# Learning Objectives for Lecture 12

- Fourier Transform Pairs
- Convolution Theorem
- Gaussian Noise (Fourier Transform and Power Spectrum)
- Spectral Estimation
  - Filtering in the frequency domain
  - Wiener-Kinchine Theorem
- Line noise removal
- Shannon-Nyquist Theorem (and zero padding)