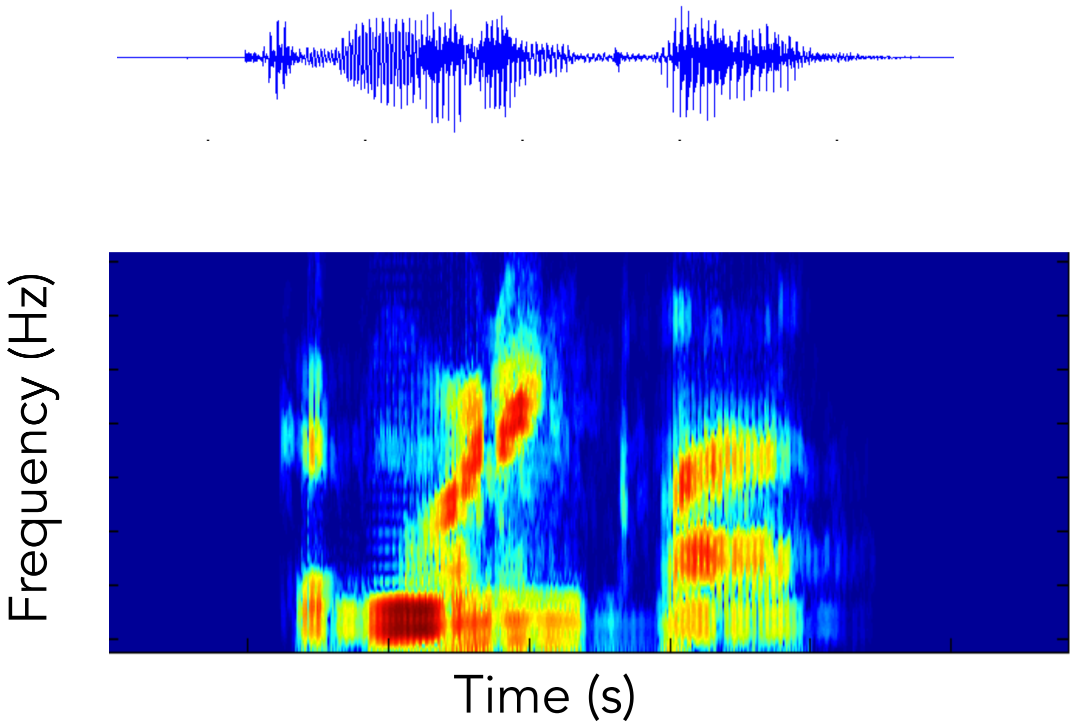
Introduction to Neural Computation

Prof. Michale Fee
MIT BCS 9.40 — 2018

Lecture 13 - Spectral analysis III

Spectral Analysis



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

Nyquist-shannon theorem

- How do we ensure that the sampling rate is greater than twice the bandwidth of the signal? $F_{samp} > 2B$

- You don’t want to sample at unnecessarily high frequencies because:
  - High-speed analog to digital converters are expensive
  - Large data files are computationally expensive to process and store

Nyquist-shannon theorem

- How do we ensure that the sampling rate is greater than twice the bandwidth of the signal? $F_{samp} > 2B$

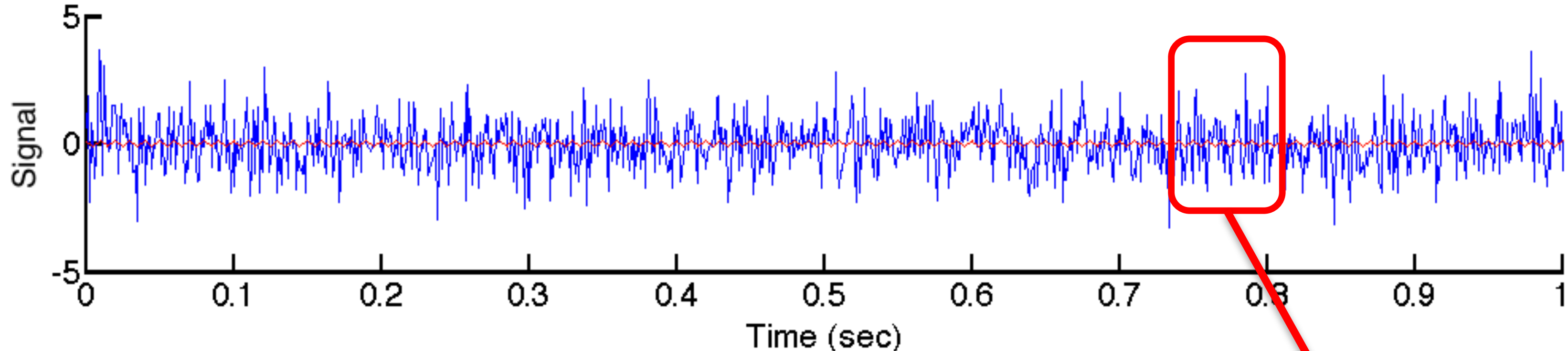
1. Use your understanding of the problem you are studying to estimate the highest frequencies you need to keep.
   - For example, the highest important frequency for recording spike waveforms is 5-10kHz

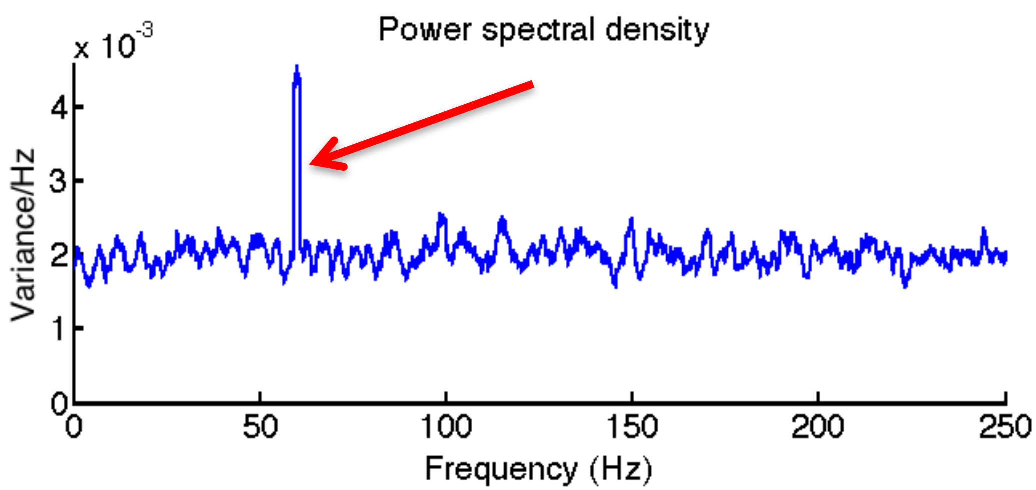
2. Use a low-pass (anti-aliasing) filter to cut out frequencies higher than the highest frequencies of interest.
   - For example, use a low pass filter that cuts off above 10-15 kHz

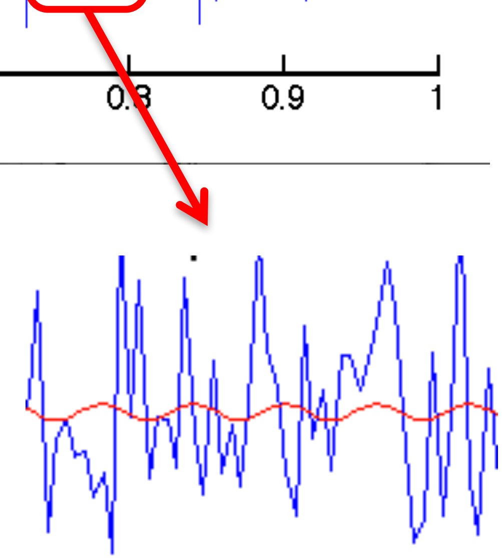
3. Sample at 2-4 times the low-pass filter cutoff.
   - For example, sample at 20-40 kHz

Spectral estimation

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







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

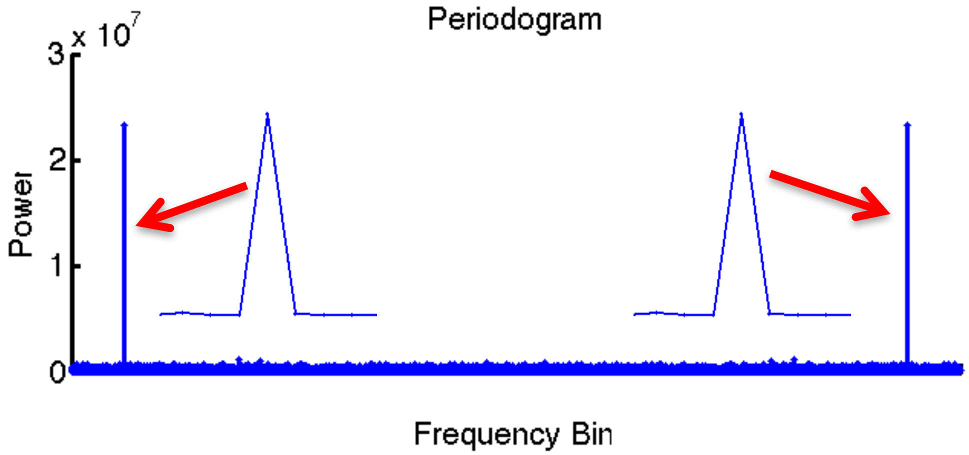
Powerspec_osc_in_noise.m

Line noise removal

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

Periodogram

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



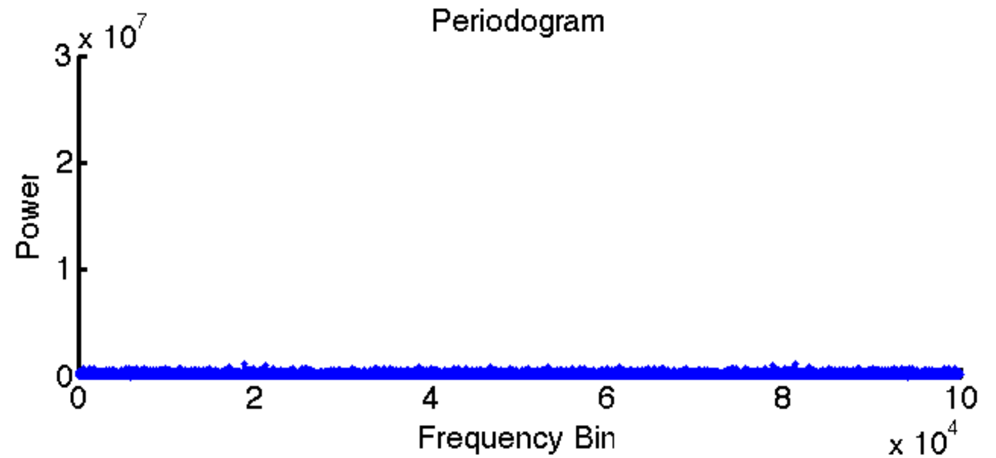
- While the periodogram is a terrible spectral estimator for non-periodic broadband signals, it is a great estimator for perfectly stationary single-frequencies… like contamination from 60Hz.

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

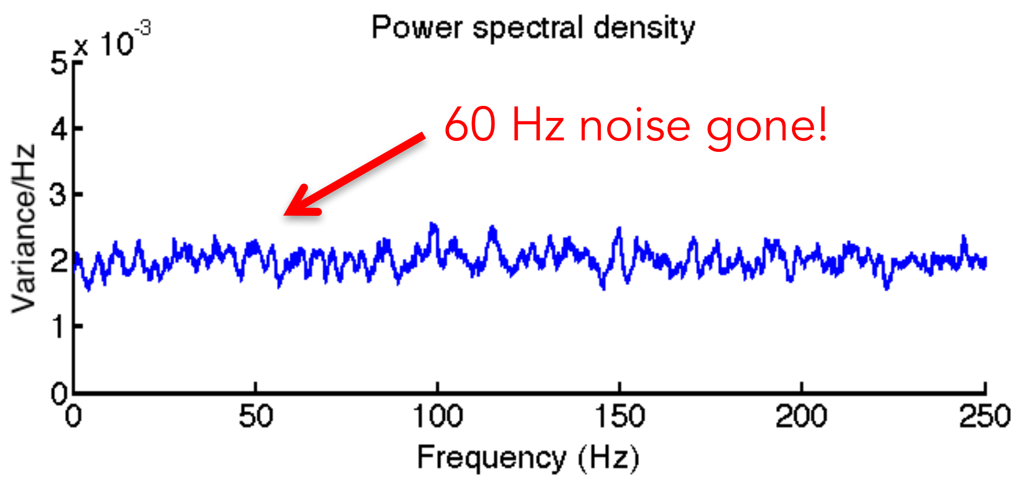
Off with its head!

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 13

- Brief review of Fourier transform pairs and convolution theorem
- Spectral estimation
  - Windows and Tapers
- Spectrograms
- Multi-taper spectral analysis
  - How to design the best tapers (DPSS)
  - Controlling the time-bandwith product
- Advanced filtering methods

Learning Objectives for Lecture 13

- Brief review of Fourier transform pairs and convolution theorem

- Spectral estimation
  - Windows and Tapers

- Spectrograms

- Multi-taper spectral analysis
  - How to design a taper (DPSS)
  - Controlling the time-bandwith product

- Advanced filtering methods

Fourier transform pair

Square pulse

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

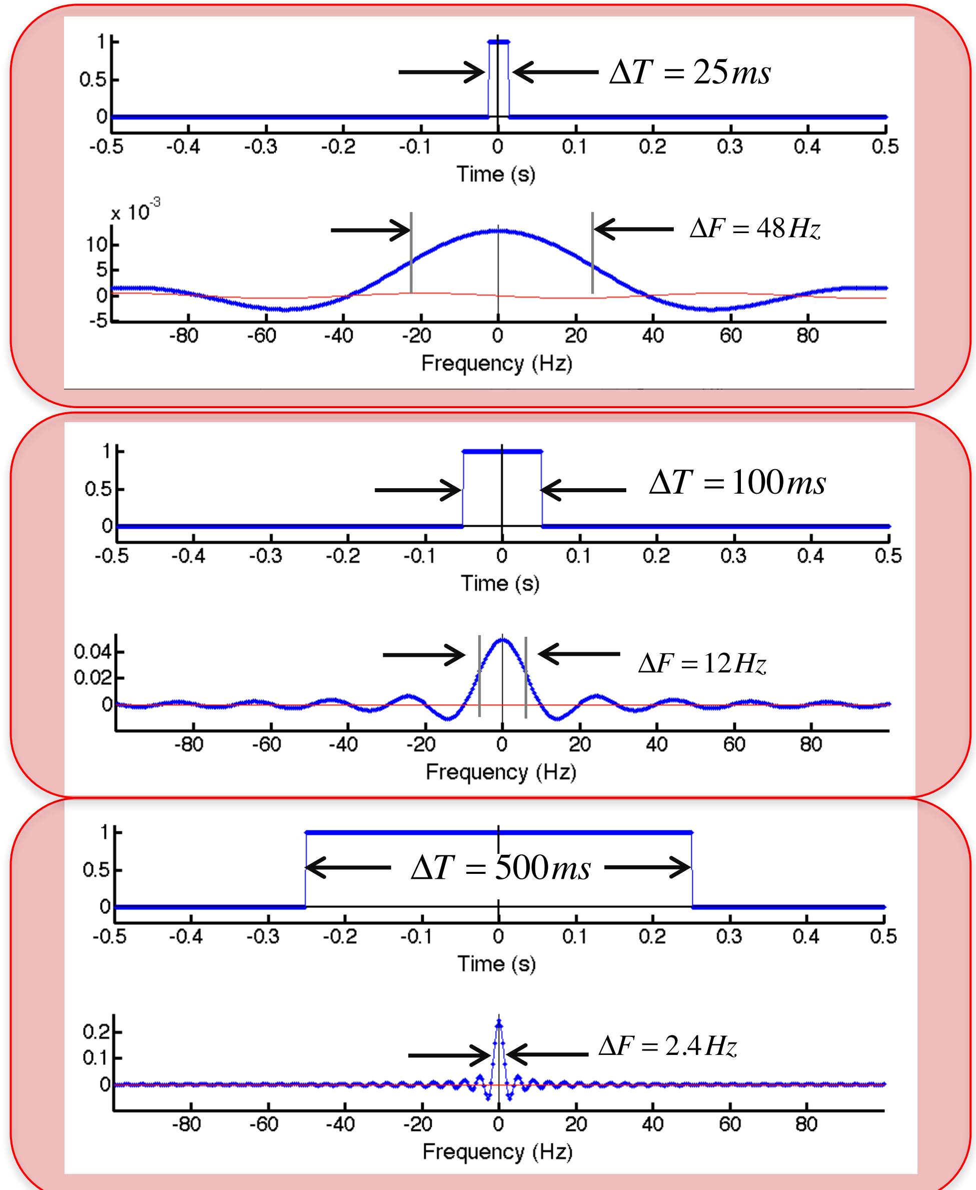
Sinc function

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

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

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

Square_window.m

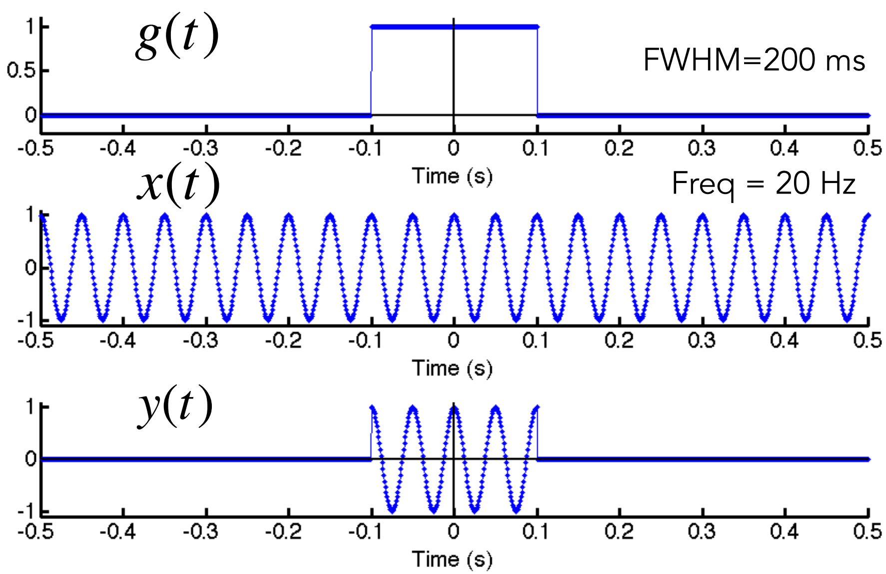


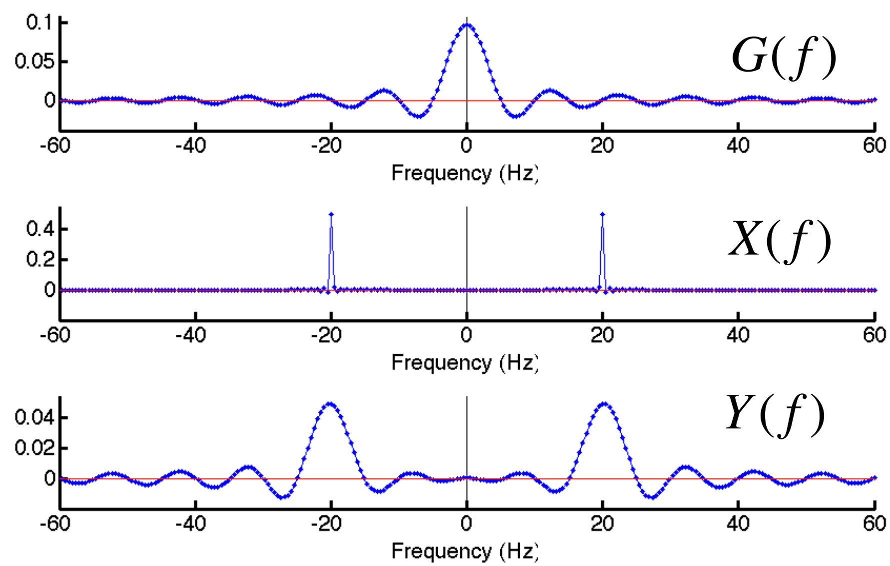
Discrete Fourier transform

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

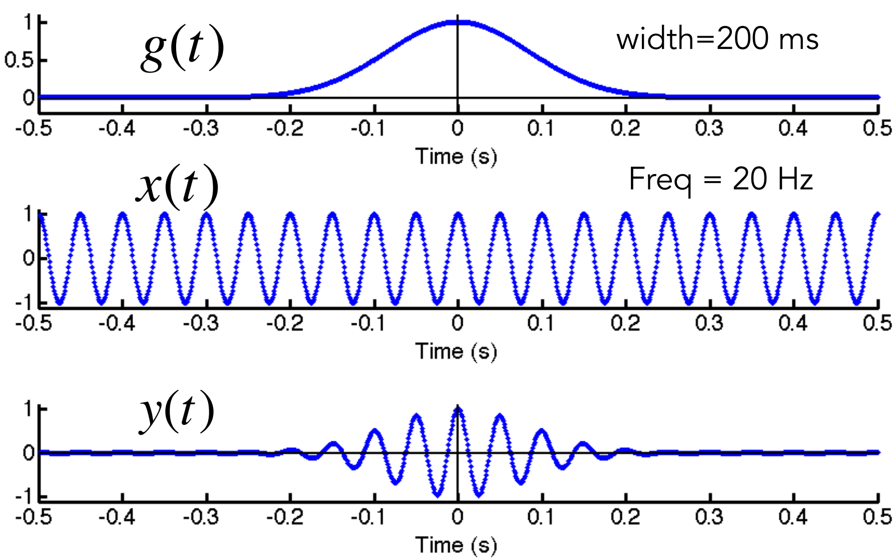
$$Y(f) = G(f) * X(f)$$

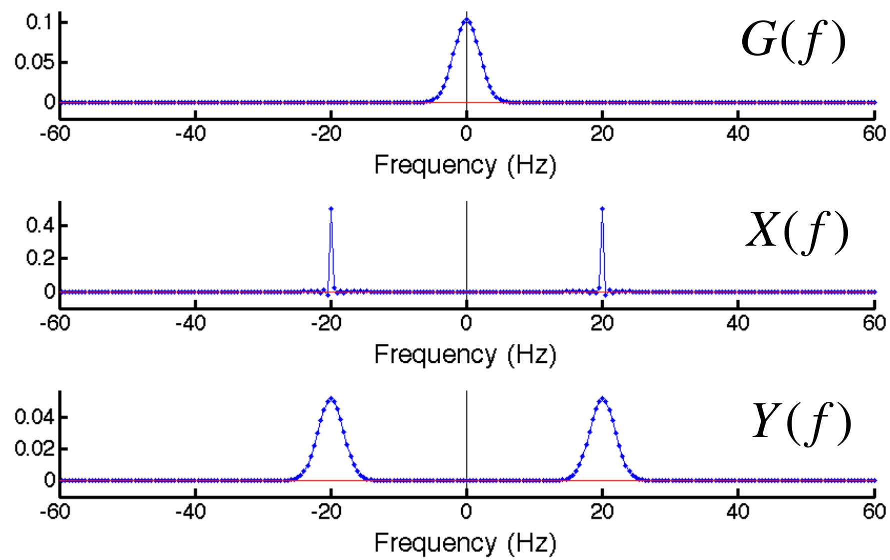
Convolution in the frequency-domain!

Using the Convolution Theorem

Gaussian-windowed cosine

Cos_Gauss_pulse.m





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

Product in the time-domain

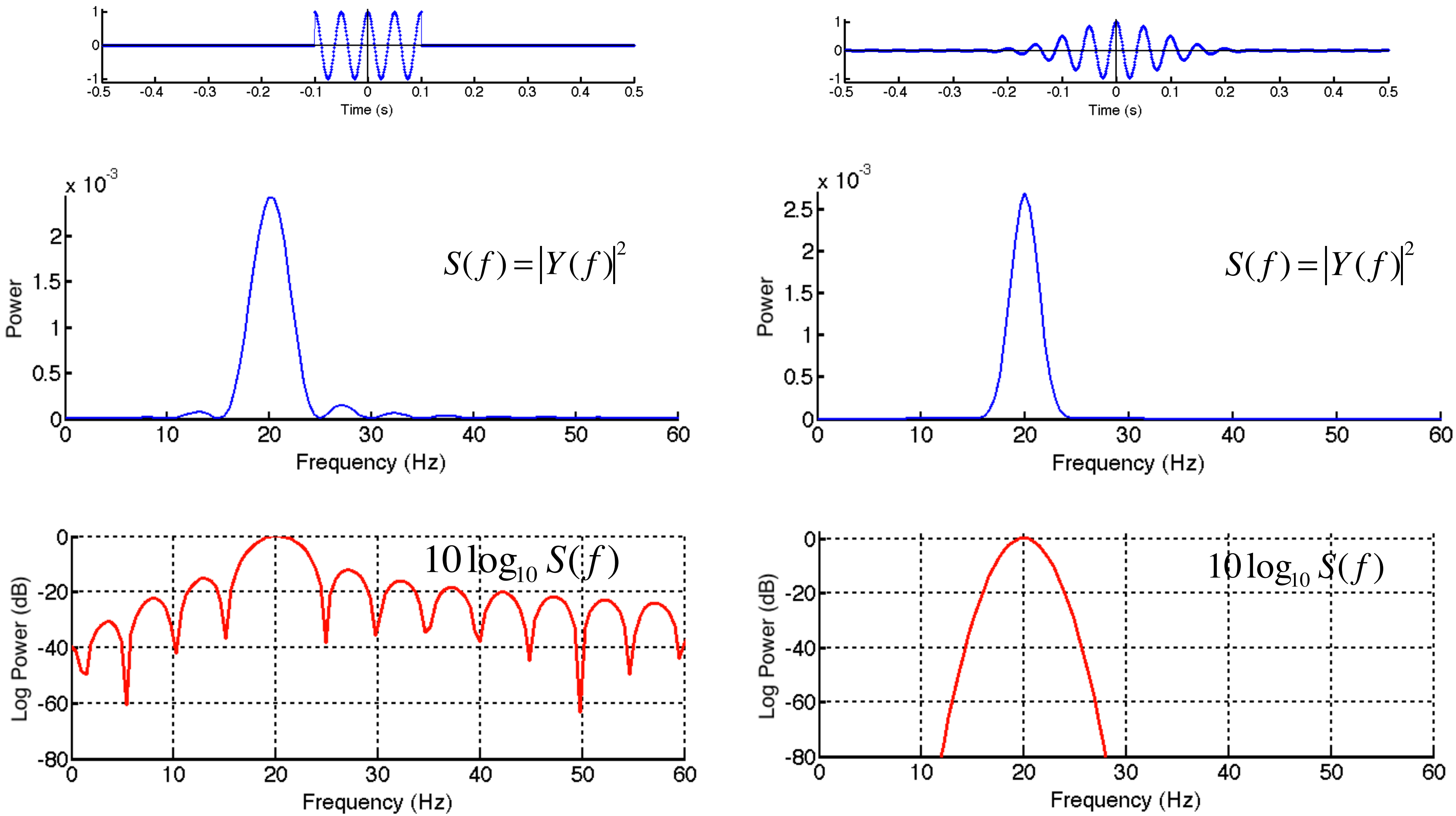
$$Y(f) = G(f) * X(f)$$

Convolution in the frequency-domain!

Discrete Fourier transform

- Square vs. Gaussian windowing

cos_Gauss_pulse.m



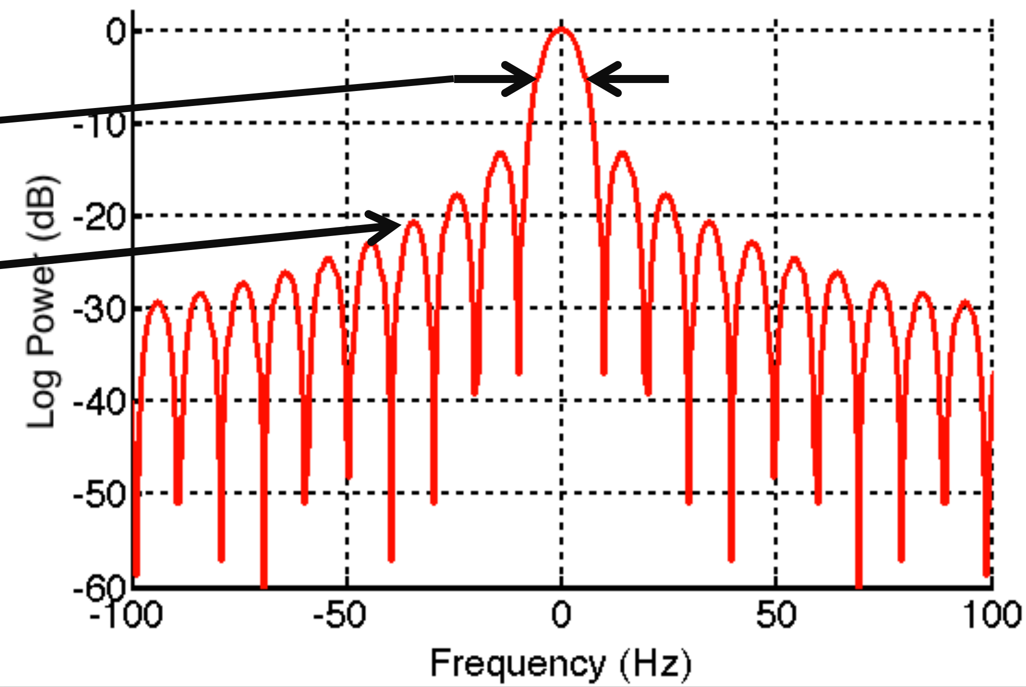
Spectral estimation

- This ‘kernel’ is called the Dirichlet Kernel
- The finite time-window introduces two errors into the spectral estimate.

Narrowband bias

Broadband bias

Large sidelobes



Learning Objectives for Lecture 13

- Brief review of Fourier transform pairs and convolution theorem

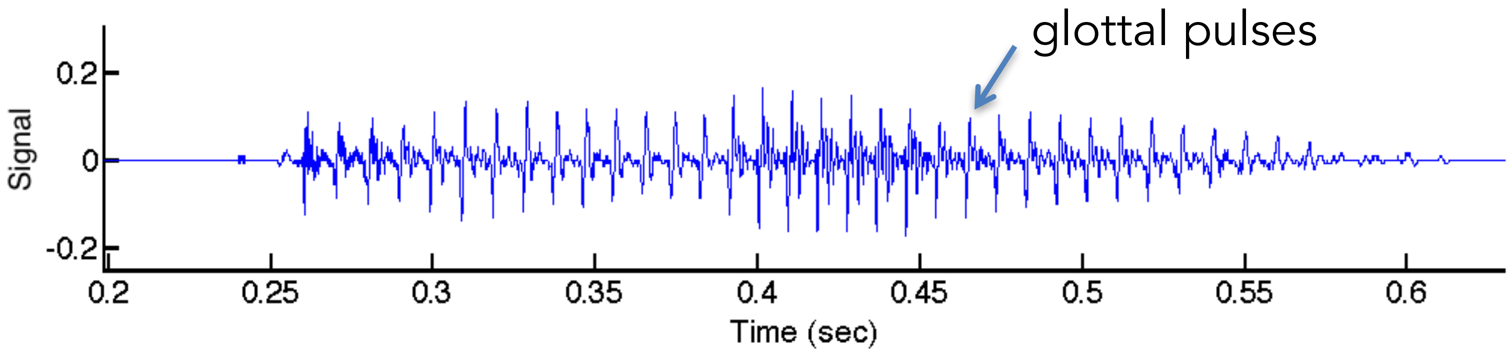
- Spectral estimation
  - Windows and Tapers

- Spectrograms

- Multi-taper spectral analysis
  - How to design a taper (DPSS)
  - Controlling the time-bandwith product

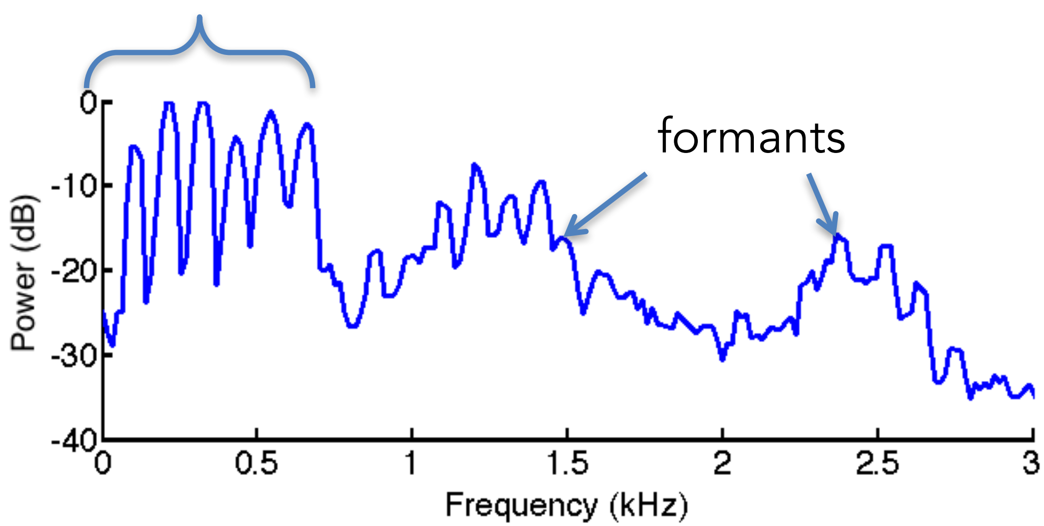
- Advanced filtering methods

Spectrum of speech signals



What will the spectrum look like?

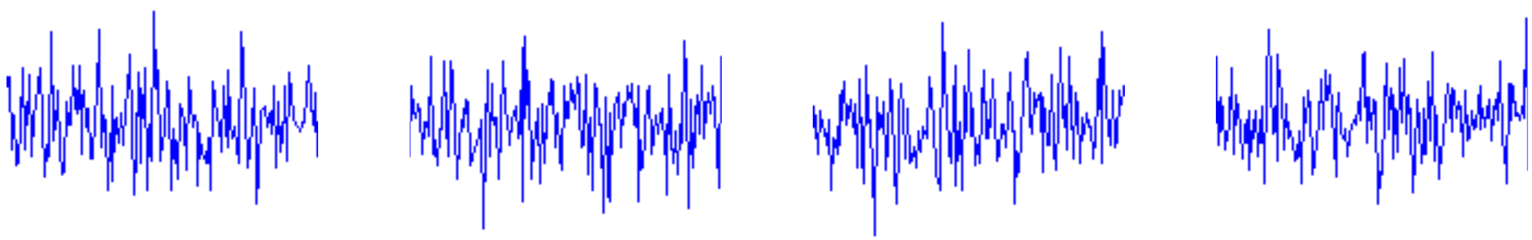
What causes these broad bumps?



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



…

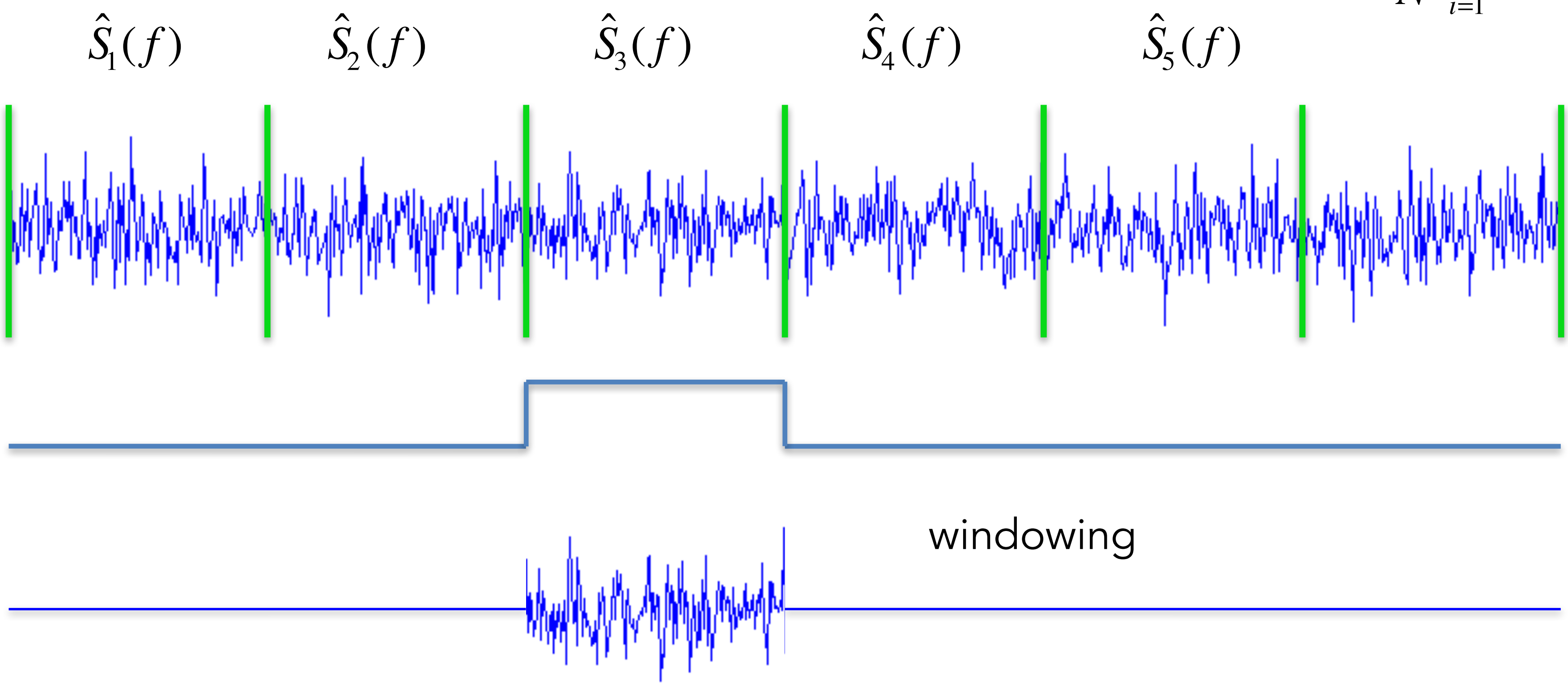
$\hat{S}_1(f)$ $\hat{S}_2(f)$ $\hat{S}_3(f)$ $\hat{S}_4(f)$

We can just average!

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

Spectral estimation

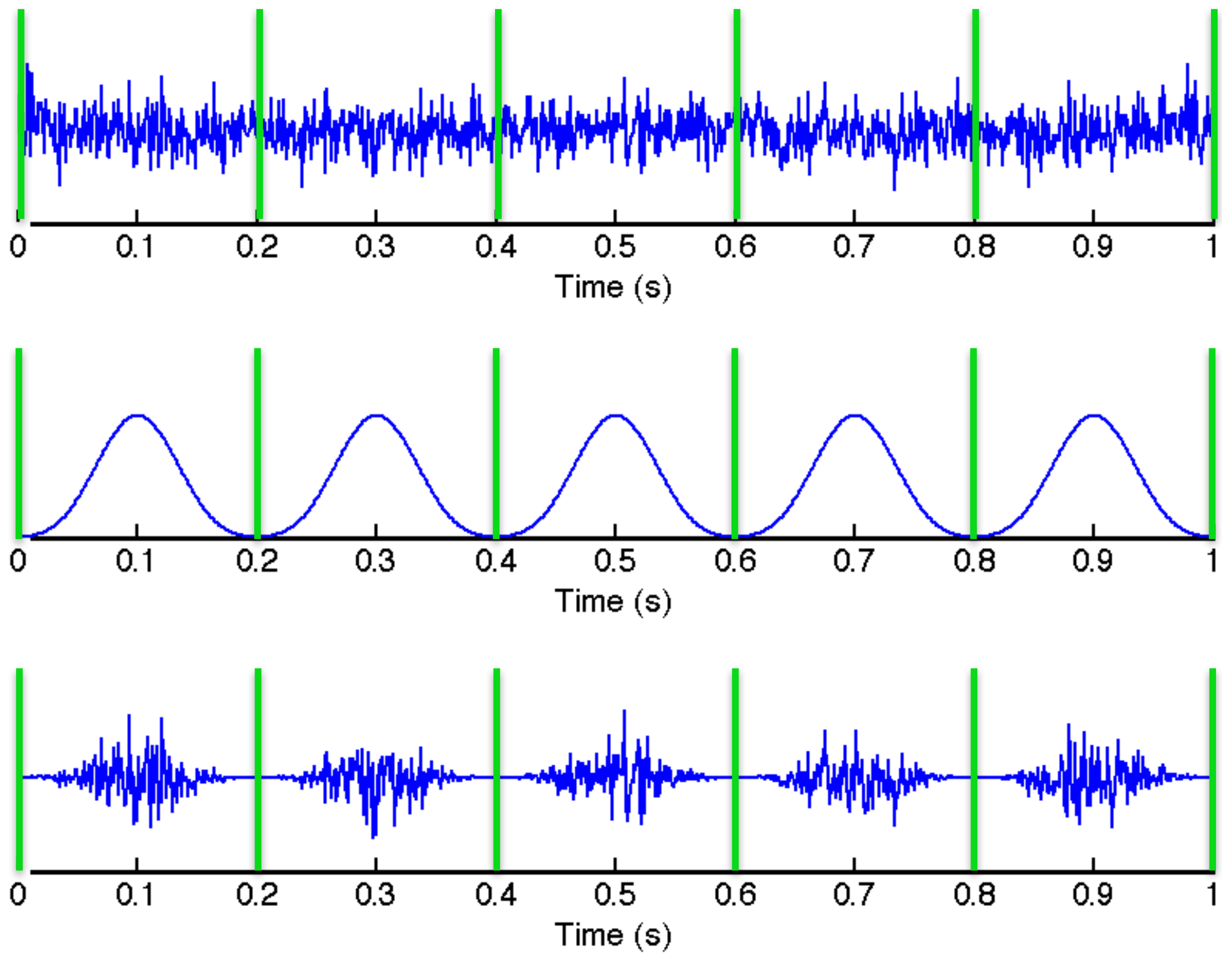
$$\hat{S}(f) = \frac{1}{N}\sum_{i=1}^{N}\hat{S}_i(f)$$



- 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'.



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

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

Learning Objectives for Lecture 13

- Brief review of Fourier transform pairs and convolution theorem

- Spectral estimation
  - Windows and Tapers

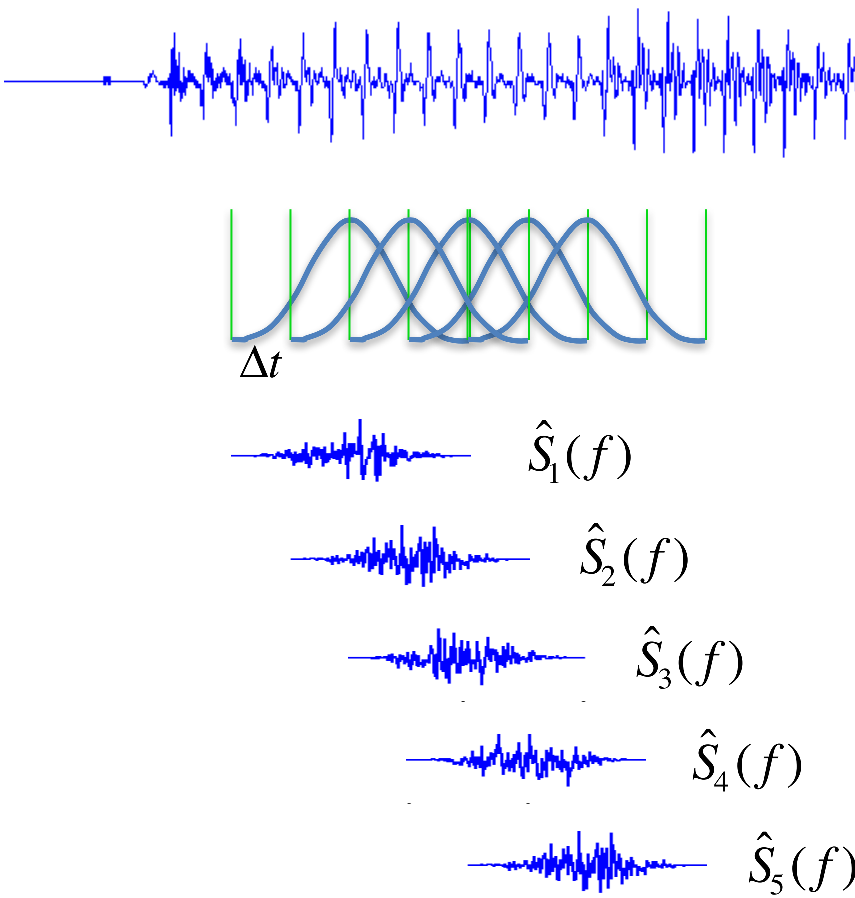
- Spectrograms

- Multi-taper spectral analysis
  - How to design a taper (DPSS)
  - Controlling the time-bandwith product

- Advanced filtering methods

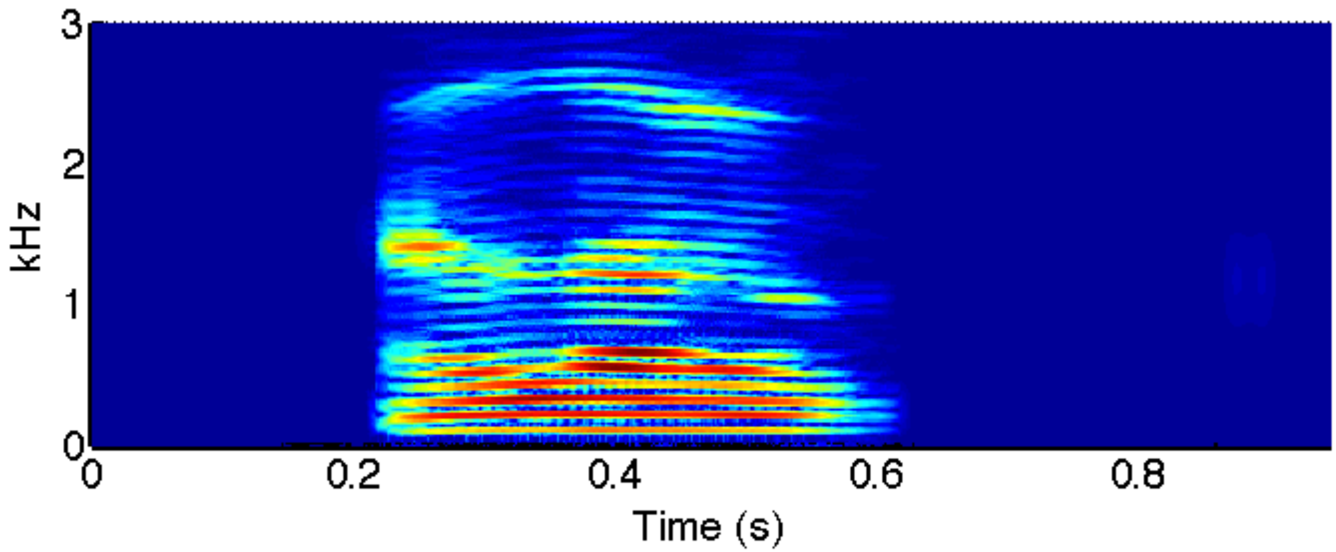
Time-varying spectrum (or Spectrogram)

- Compute the spectrum in short time windows of length T
  — slide the window in small steps of size Δt.



$$\hat{S}(t_i, f) = \hat{S}_i(f)$$
$$where \;\; t_i = i\,\Delta t$$

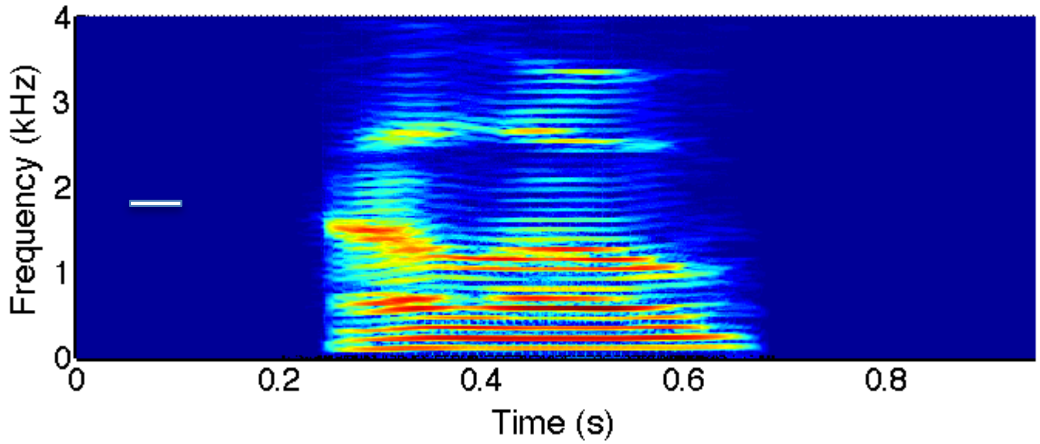
Spectrogram of speech signals

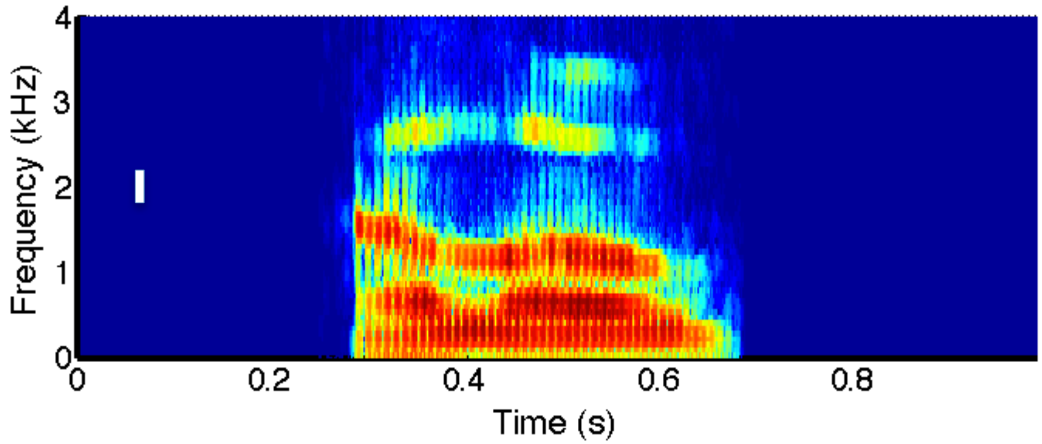


WSpecgram.m

What you see depends on the taper!

- How do I choose the length of the window?
- What kind of taper do I use?





Learning Objectives for Lecture 13

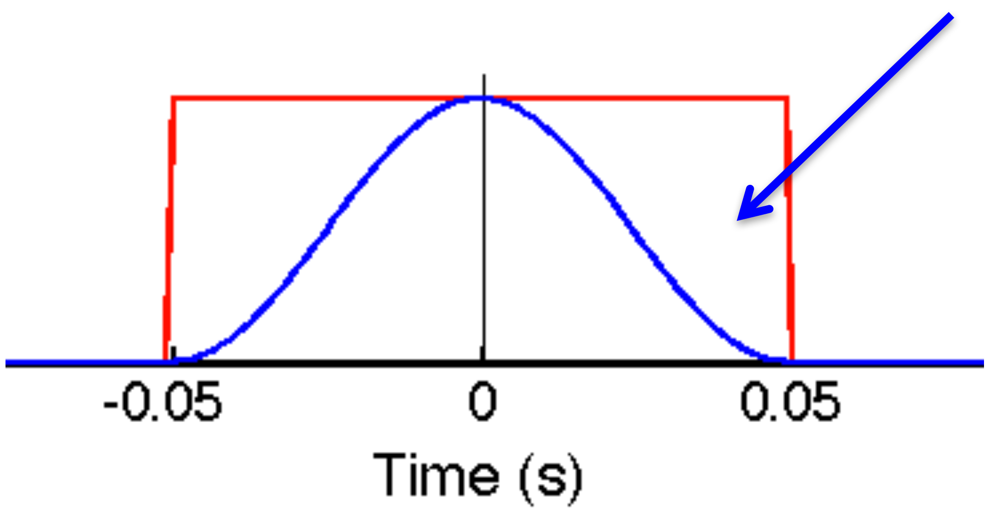
- Brief review of Fourier transform pairs and convolution theorem
- Spectral estimation
  - Windows and Tapers
- Spectrograms
- Multi-taper spectral analysis
  - How to design a taper (DPSS)
  - Controlling the time-bandwith product
- Advanced filtering methods

Tapers

- Is there a perfect taper?

  No, because a function that is strictly limited to a time window between –T/2 to T/2 has a spectrum that extends to infinity in frequency.

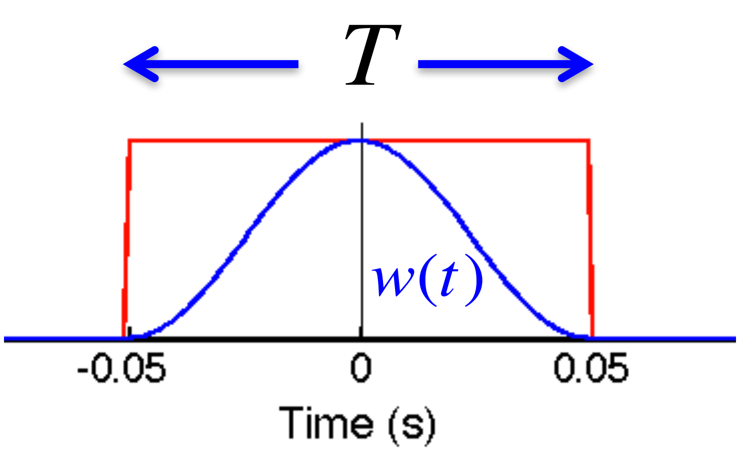
  Another problem with tapering is that, when we make a 'smooth' function that goes to zero at the edges, we lose data!



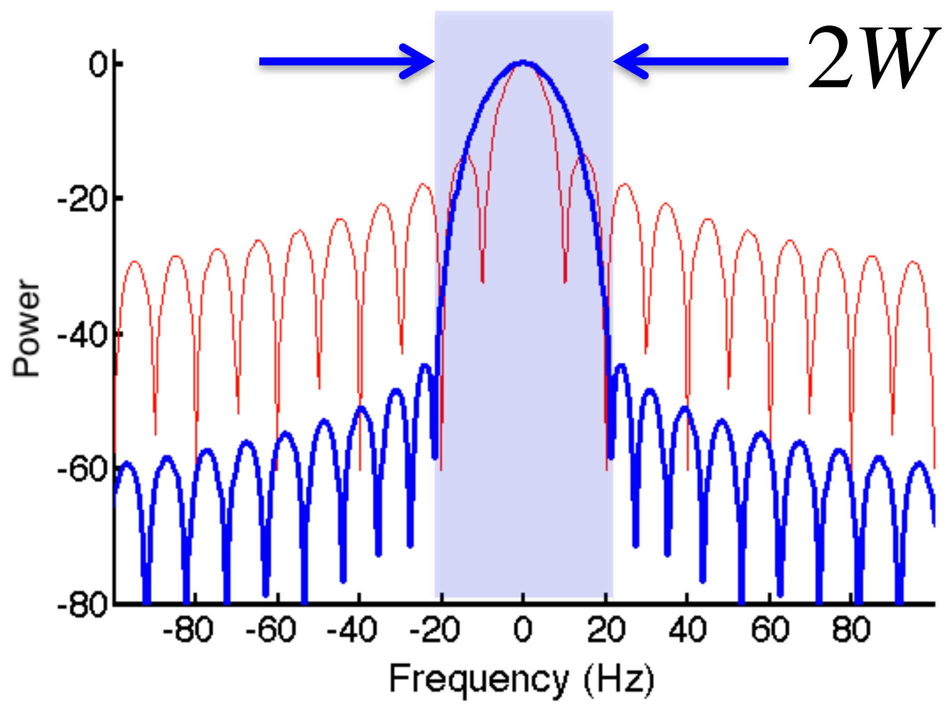
Tapers

- First we consider the spectral concentration problem

  We want to find a strictly time-localized function [-T/2,T/2] whose Fourier Transform is maximally localized within a finite window in the frequency domain [-W,W].



dpss_taper.m.m



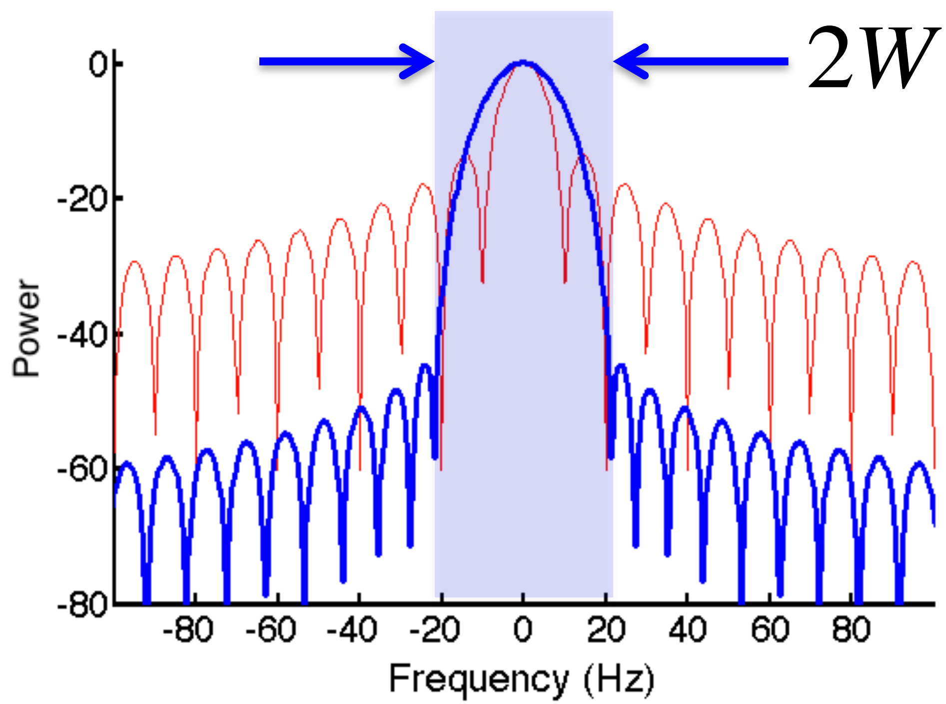
Tapers

- We want to find a function of time w(t) that maximizes the spectral concentration.

$$\lambda = \frac{\int_{-W}^{W} \left|U(f)\right|^2 df}{\int_{-\infty}^{\infty} \left|U(f)\right|^2 df}$$

U(f) is the F.T. of w(t)

$$U(f) = \int_{-\infty}^{\infty} w(t) e^{-i2\pi f t}\, dt$$

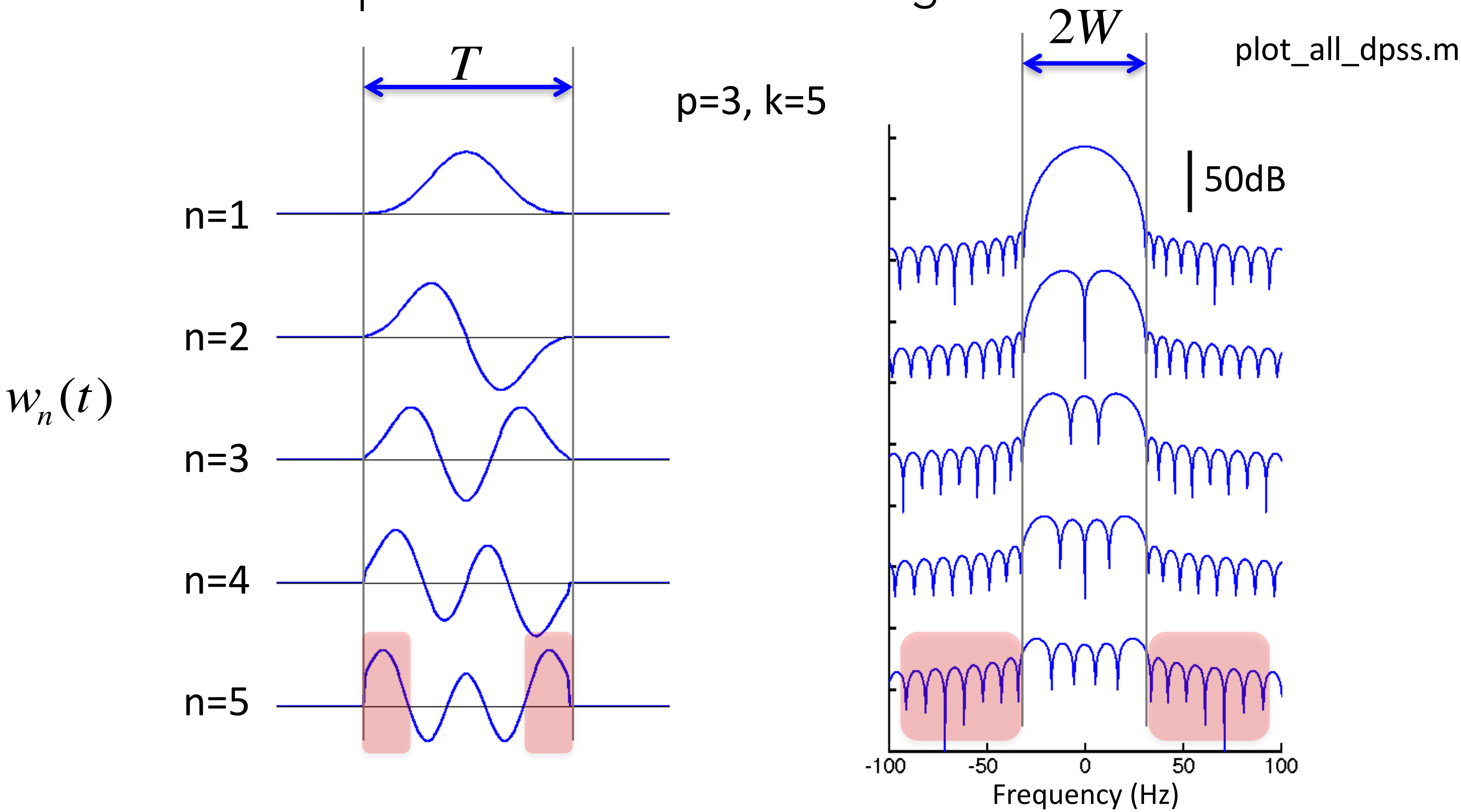


- Maximizing λ gives a set of k=2WT-1 functions called Slepian functions for which λ is very close to 1.

… also called discrete prolate spheroid sequence (dpss)

DPSS Tapers

- The set of dpss functions is also orthogonal.



- Because they are orthogonal, each will give an independent estimate of the spectrum!

Multi-taper spectral estimation

- Select a time window width T (temporal resolution).
- Select a time-bandwidth product p=WT (i.e. set the frequency resolution).
- Compute the set of set of dpss tapers using T and p=WT
- Estimate the spectrum using each of the k= 2*p-1 tapers

$$\hat{S}_n(f) = \left| \sum_{t=1}^{N} w_n(t) y(t) e^{-i2\pi f t} \right|^2$$

- Average the estimates to get the spectrum!

$$S(f) = \frac{1}{k} \sum_{n=1}^{k} \hat{S}_n(f)$$

- You get multiple spectral estimates from the same piece of data. Which means you can get error bars !

Learning Objectives for Lecture 13

- Brief review of Fourier transform pairs and convolution theorem

- Spectral estimation
  – Windows and Tapers

- Spectrograms

- Multi-taper spectral analysis
  – How to design a taper (DPSS)
  – Controlling the time-bandwith product

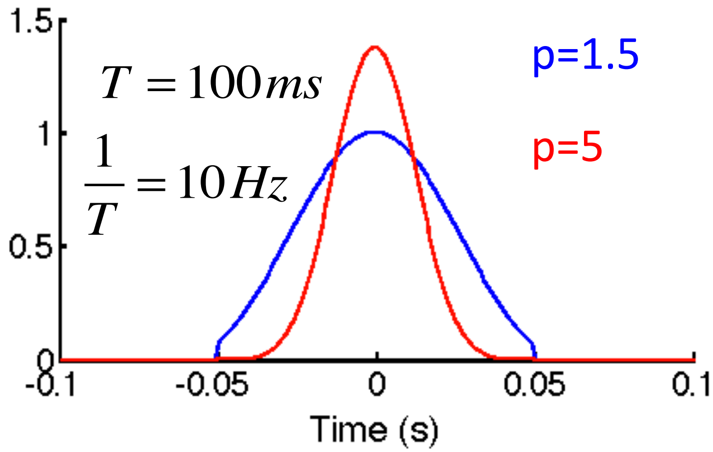
- Advanced filtering methods

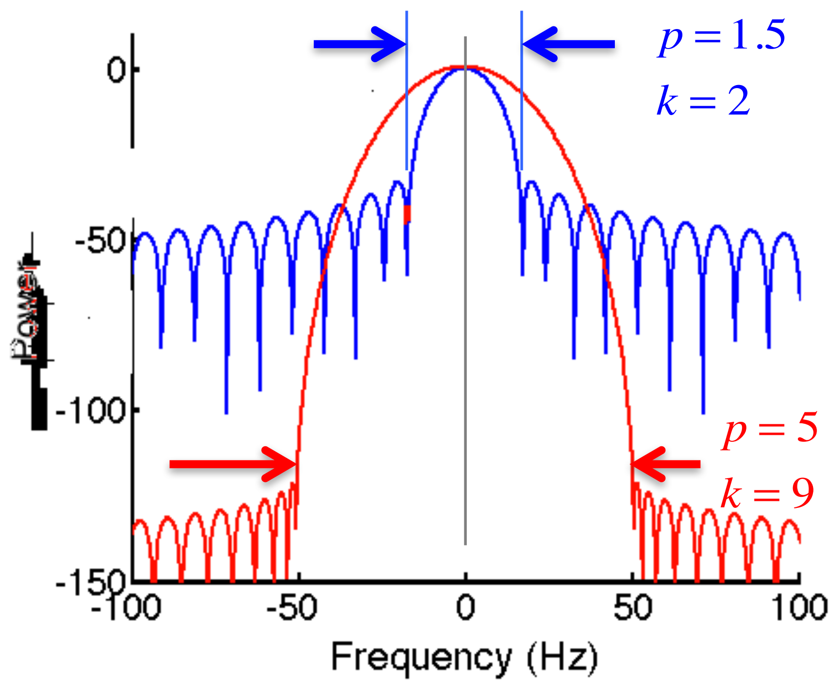
Time-bandwidth product

- With a larger p, you get more suppression of the side-lobes, and you increase the bandwidth.

- But you also get more tapers, you get more spectral estimates from the same piece of data, and more averaging.

$$k = 2WT - 1$$
$$= 2p - 1$$





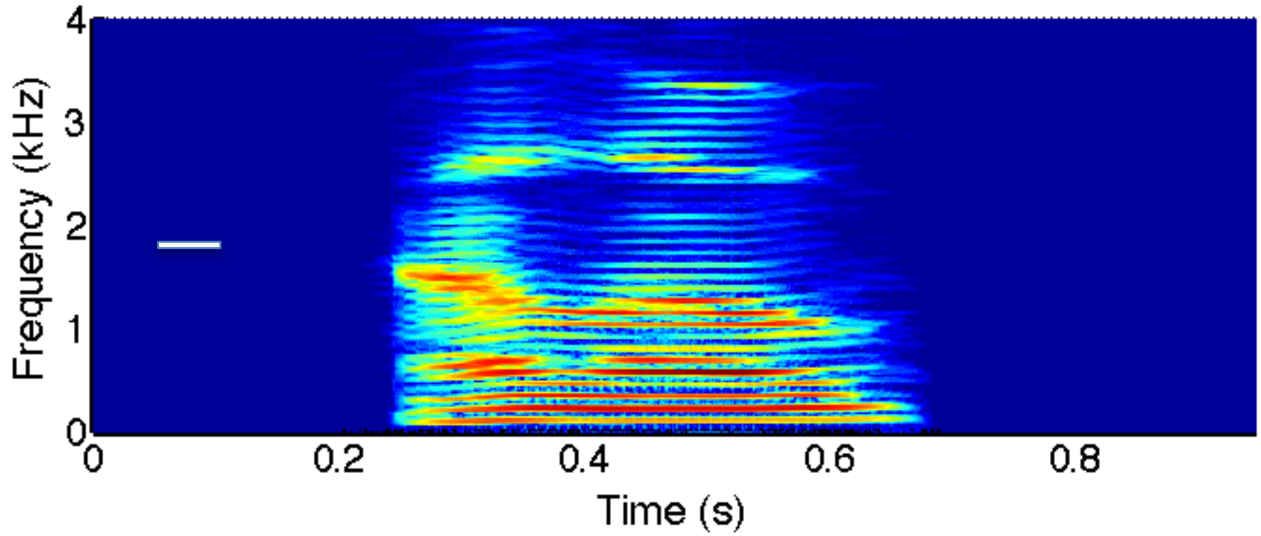
Dpss_comp_WT.m

Time-bandwidth product

$$T = 50\,ms$$

$$2W = 60\,Hz$$

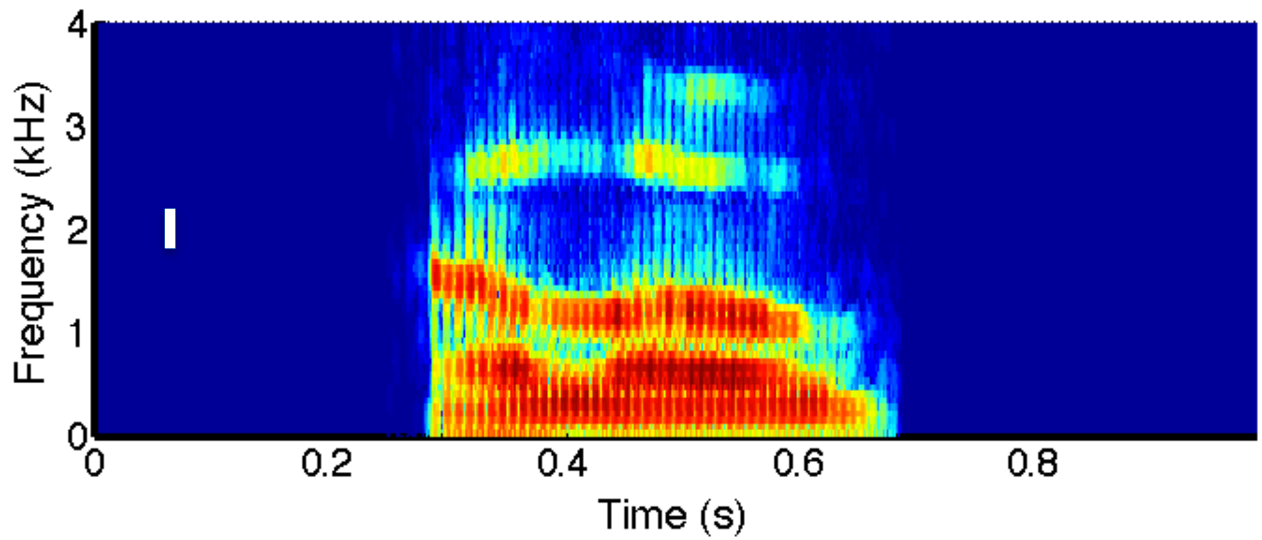
$$p = 1.5 \quad k = 2$$



$$T = 8\,ms$$

$$2W = 375\,Hz$$

$$p = 1.5 \quad k = 2$$



demo_specgram.m

Time-bandwidth product

- There is a fundamental limit to the resolution in time and frequency.

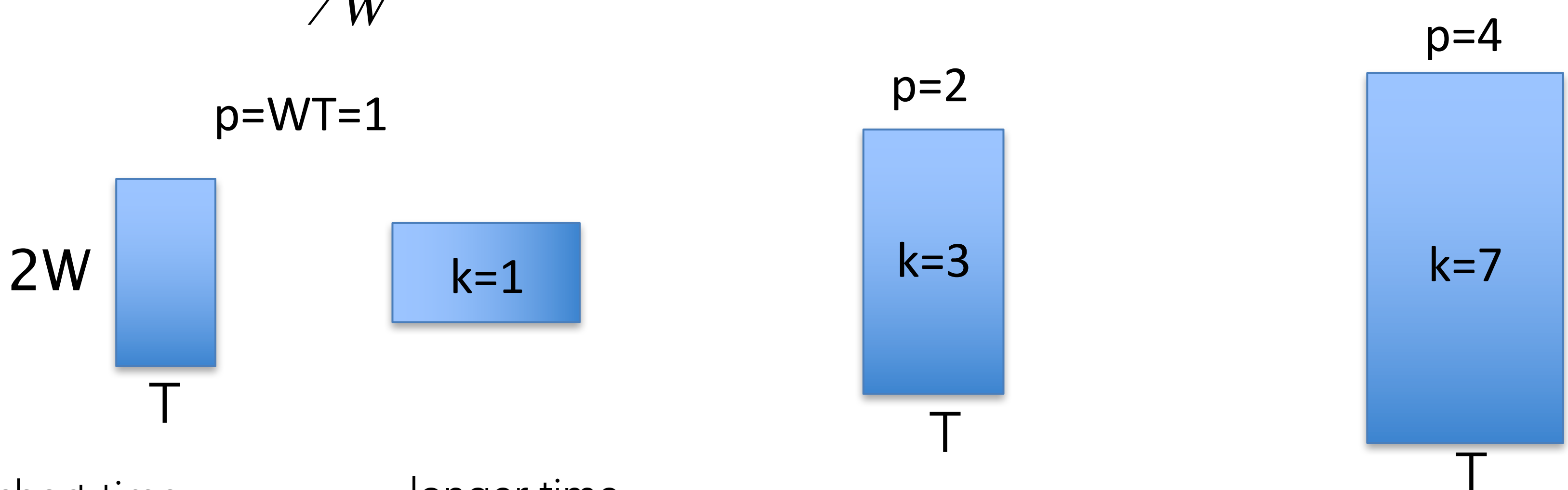
$$WT > 1$$

- If you want a temporal resolution of T, the bandwidth has to be greater than $W > 1/T$

$W = 1/T$ for a square taper

$W > 1/T$ for 'narrower' tapers

- If you want a bandwidth of W, the time window has to be greater than $T > \frac{1}{W}$



short time
large bandwidth

longer time
smaller bandwidth

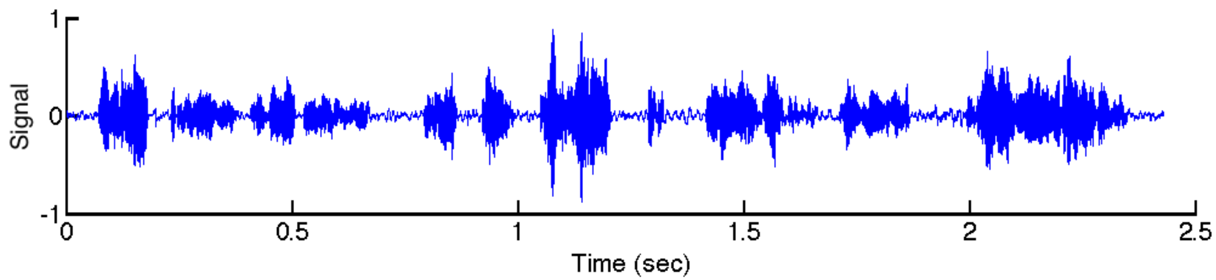
p=1.5, 2, 2.5, 3, 4

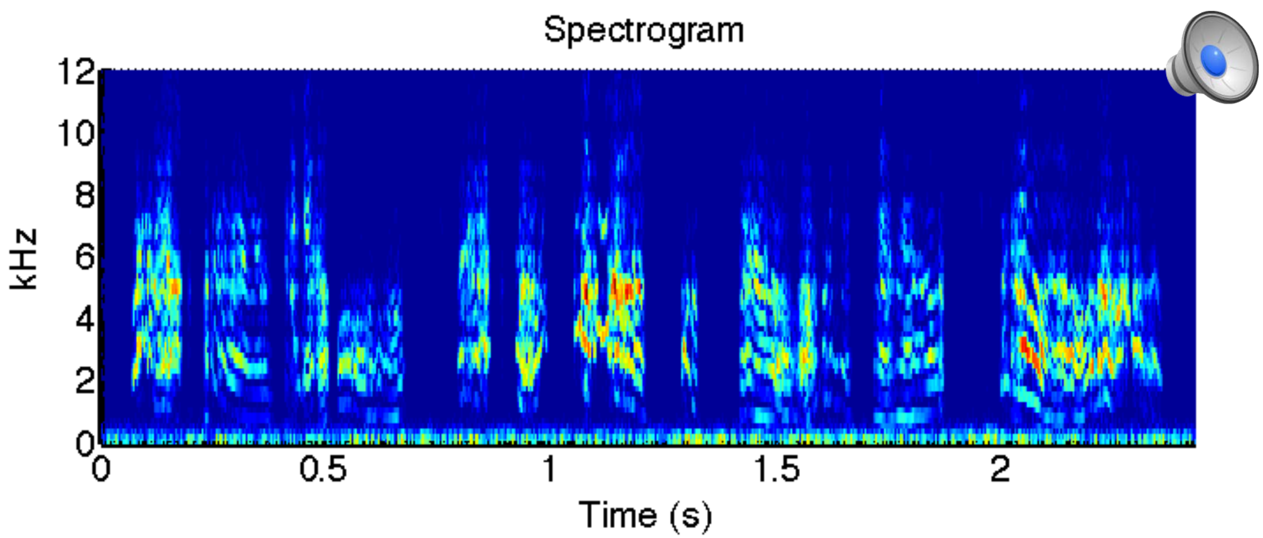
Learning Objectives for Lecture 13

- Brief review of Fourier transform pairs and convolution theorem
- Spectral estimation
  - Windows and Tapers
- Spectrograms
- Multi-taper spectral analysis
  - How to design a taper (DPSS)
  - Controlling the time-bandwith product
- Advanced filtering methods

Filtering

- Sometimes offending noise is not a single line. But if it is well enough separated from your signal, then you can use filtering.

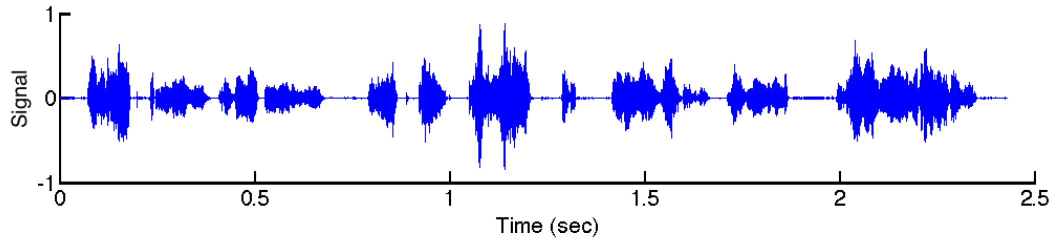


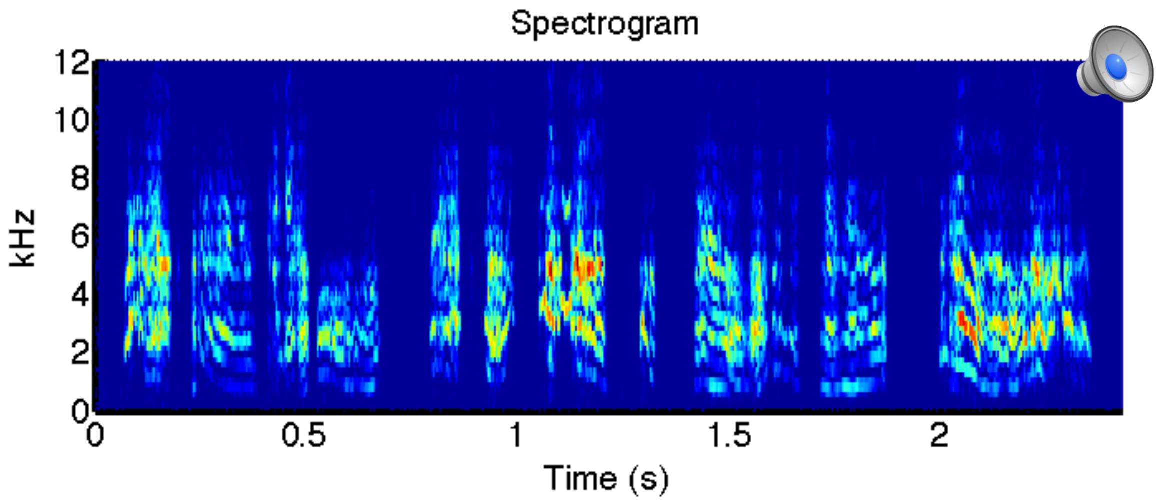


- We talked about using convolution for high-pass or low-pass filtering, but there are very powerful tools built into MATLAB® for this.

High-pass filtering

filter_demo.m

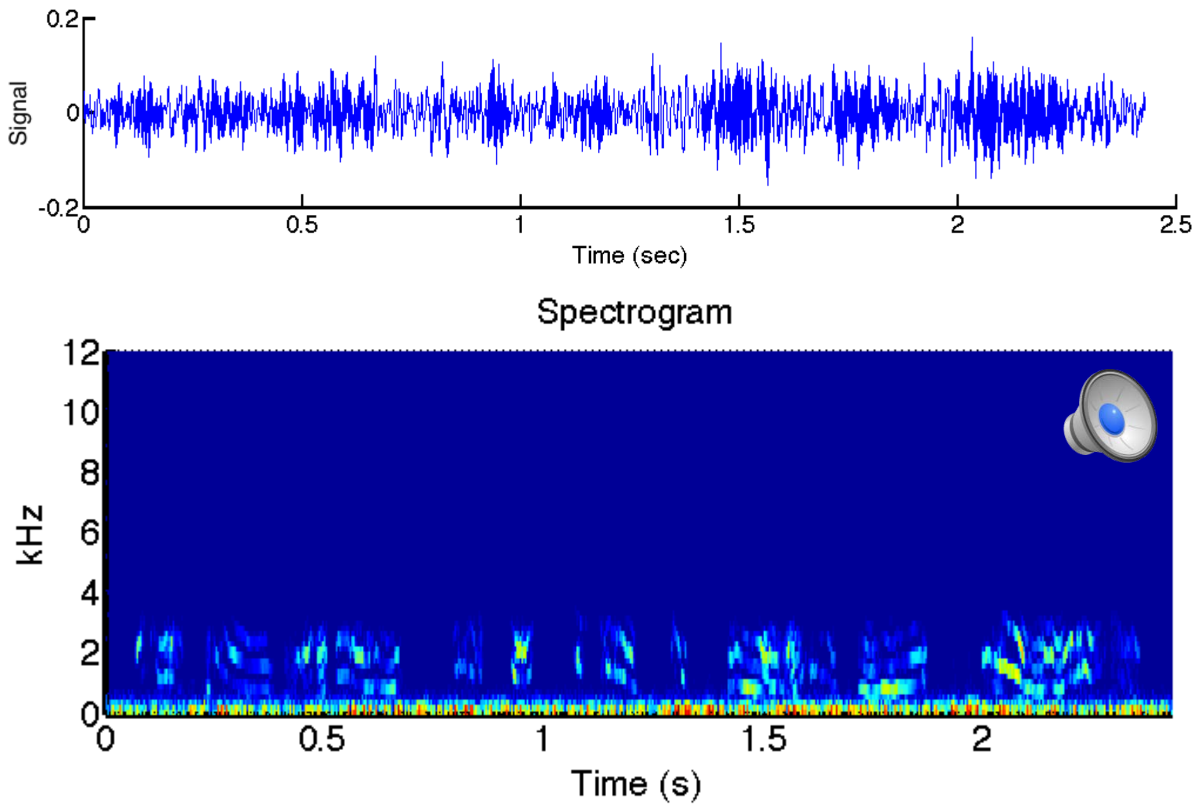





```
Fnyq=Fs/2.;        % Nyquist frequency (samples/sec)
cutoff = 500;      % Set cutoff frequency (Hz)
Wn = (cutoff/Fnyq);
[b,a]= butter(4, Wn, 'high');   % Butterworth high-pass
Data = filtfilt(b,a,DataIn);    %Run the filter!
```

Low-pass filtering

- Sometimes offending noise is not a single line. But if it is well enough separated from your signal, then you can use filtering.

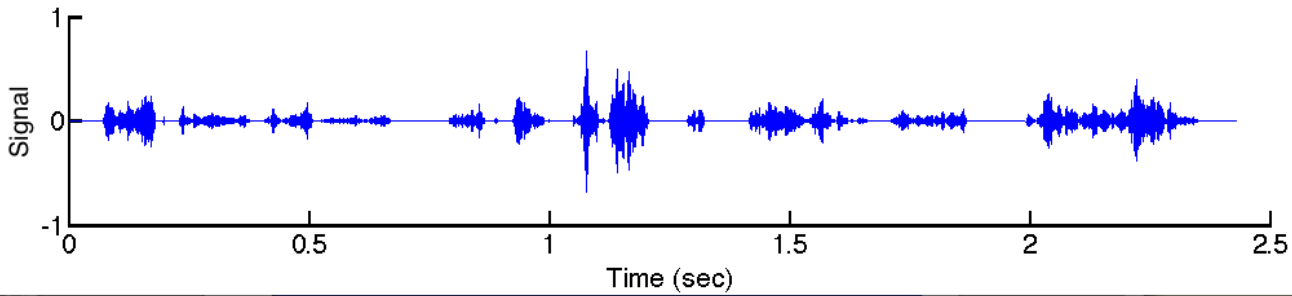


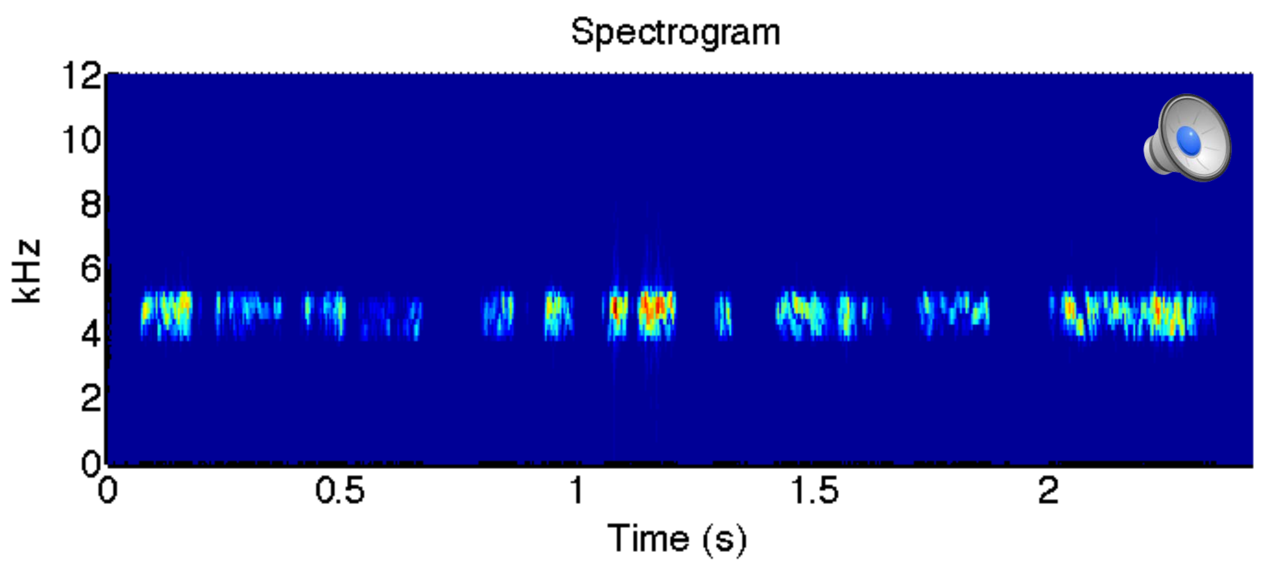
filter_demo.m

```
Fnyq=Fs/2.;        % Nyquist frequency
cutoff = 2000;      % Set cutoff frequency
Wn = (cutoff/Fnyq);
[b,a]= butter(4, Wn, 'low');   % Butterworth low-pass
Data = filtfilt(b,a,DataIn);    % Run the filter!
```

Band-pass filtering

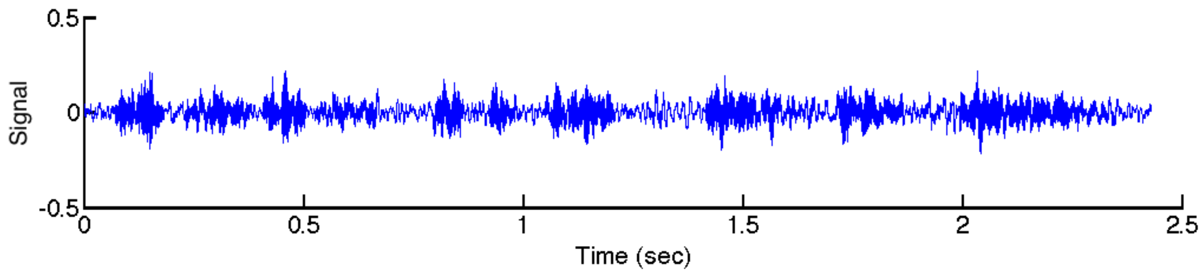
filter_demo.m

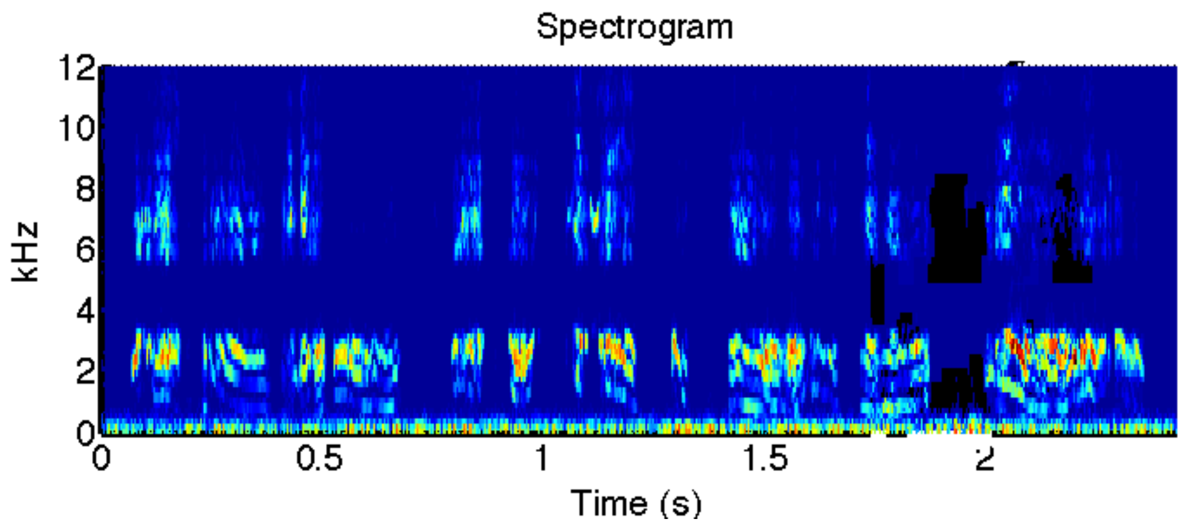





```
Fnyq=Fs/2.;          % Nyquist frequency
cutoff = [4000 5000];      % Set cutoff frequency
Wn = (cutoff/Fnyq);
[b,a]= butter(4, Wn);   % Butterworth band-pass
Data = filtfilt(b,a,DataIn);    % Run the filter!
```

Band-stop filtering

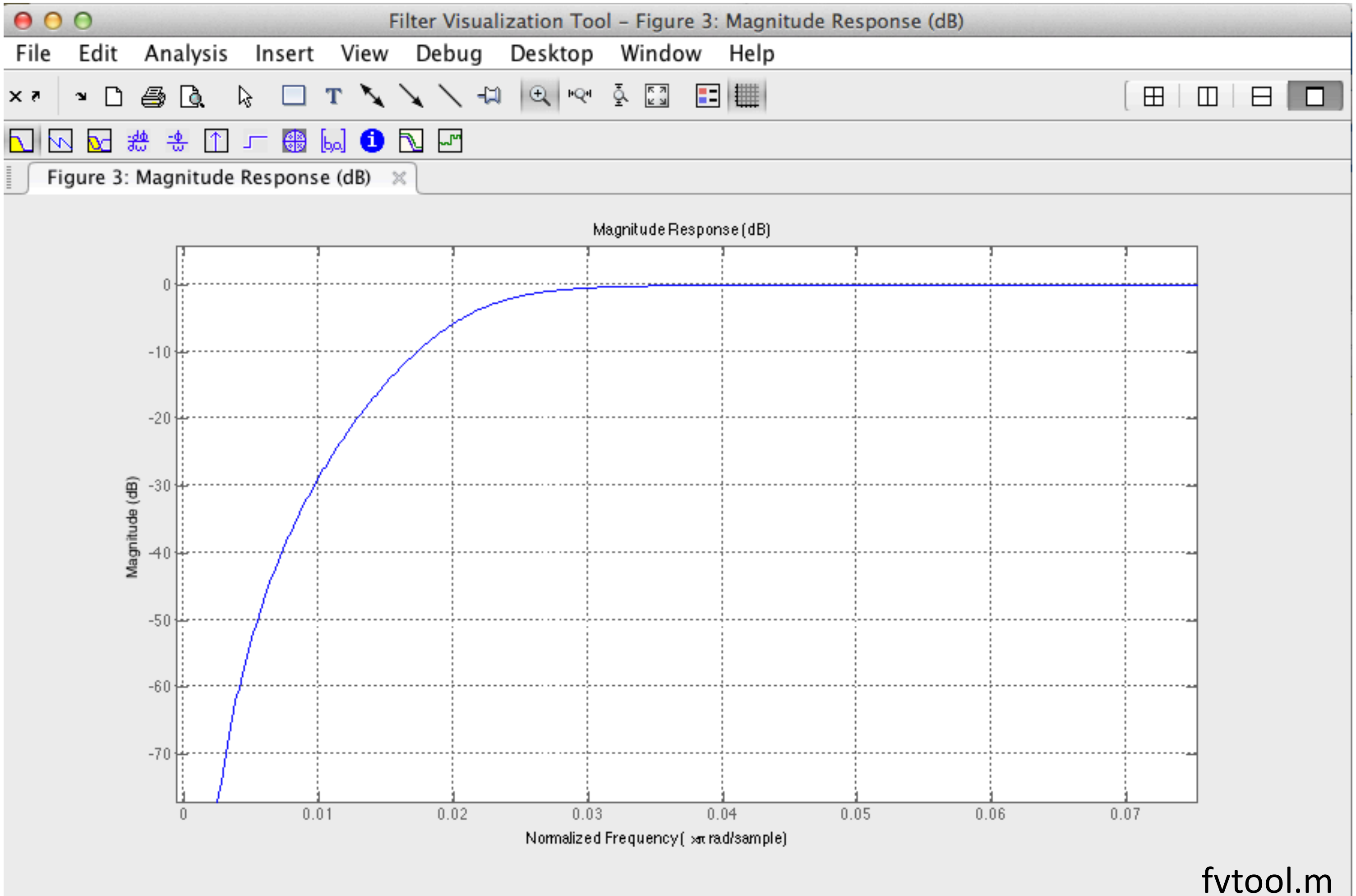




filter_demo.m

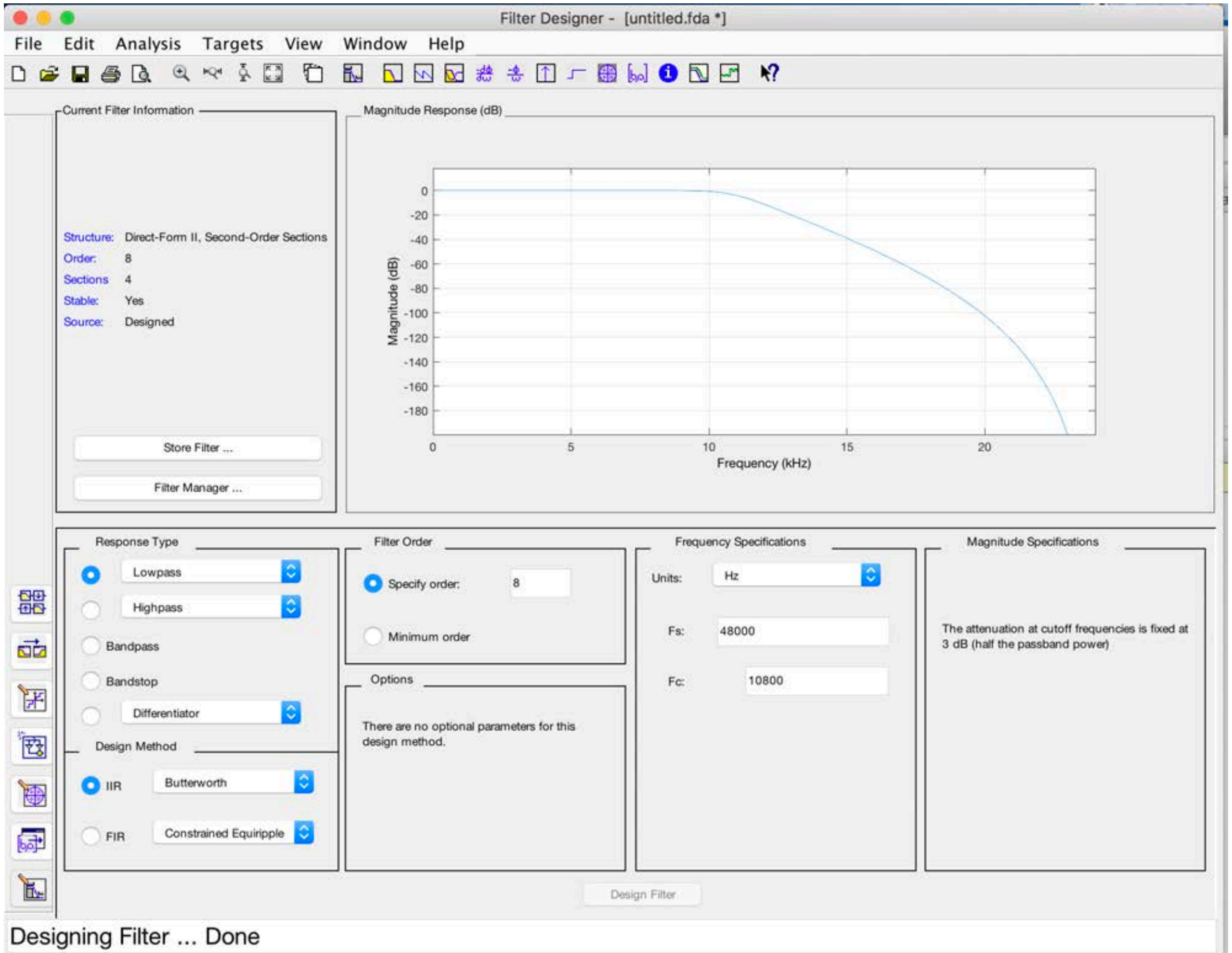
```
Fnyq=Fs/2.;          % Nyquist frequency
cutoff = [3000 6000];       % Set cutoff frequency
Wn = (cutoff/Fnyq);
[b,a]= butter(4, Wn, 'stop');   % Butterworth band-stop
Data = filtfilt(b,a,DataIn);    % Run the filter!
```

MATLAB® Filter Visualization Tool



fvtool.m

MATLAB® Filter Designer



filterDesigner.m

Learning Objectives for Lecture 13

- Brief review of Fourier transform pairs and convolution theorem

- Spectral estimation
  - Windows and Tapers

- Spectrograms

- Multi-taper spectral analysis
  - How to design the best tapers (DPSS)
  - Controlling the time-bandwith product

- Advanced filtering methods