Topics for Today's Lecture

- Spectrogram: Compute time-varying spectra
- Short-Time-Fourier-Transform (STFT): Spectrogram=$|STFT|^2$; almost always use
- Data windows: Applied to spectrogram

Time-varying spectral content

- "fft" is useful for analyzing periodic signals: $x(t)=c_0+c_1\cos(2\pi t/T+\theta_1)+c_2\cos(4\pi t/T+\theta_2)+\ldots$ >>2/N*abs(fft(X,N)) computes $[2c_0\ c_1\ c_2\ldots]$.

Time-varying spectral content

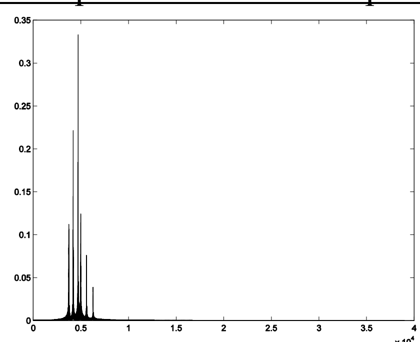
- "fft" is useful for analyzing periodic signals: $x(t)=c_0+c_1\cos(2\pi t/T+\theta_1)+c_2\cos(4\pi t/T+\theta_2)+\ldots$ >>2/N*abs(fft(X,N)) computes $[2c_0\ c_1\ c_2\ldots]$.
- **BUT**: Many (e.g., music) signals look like:

$c_{11}\cos(2\pi t/T_1+\theta_{11})+c_{21}\cos(4\pi t/T_1+\theta_{21})+\ldots t_0<t<t_1$

$c_{12}\cos(2\pi t/T_2+\theta_{12})+c_{22}\cos(4\pi t/T_2+\theta_{22})+\ldots t_1<t<t_2$

$c_{13}\cos(2\pi t/T_3+\theta_{13})+c_{23}\cos(4\pi t/T_3+\theta_{23})+\ldots t_2<t<t_3$

Example: "The Victors" spectrum



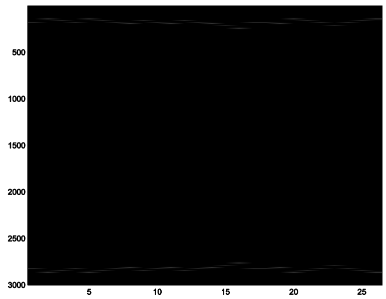
Time-varying spectral content

- **IDEA**: Segment (chop up) signal in time.
- **THEN**: Apply "fft" to each time segment.
- **HOW**: imagesc(abs(fft(reshape(X',N,L),N)))
- **WHERE**: L=#segments and N=length(X)/L;
- **WHAT**: Computes "fft" of each of L segments.
- **SHOW**: Display freq. vertical, time horizontal.

Example: "The Victors" spectrogram

- X=Tonal "The Victors" sampled 8192 Hertz.
- 26 notes of length 3000/8192 seconds each.
- Length(X)=78000=26(3000). L=26. N=3000.
- >>imagesc(abs(fft(reshape(X',3000,26)))) , colormap(gray) is shown on the next slide.
- This is called the "**spectrogram**" of X.

Example: “The Victors: spectrogram



Example: “The Victors: spectrogram

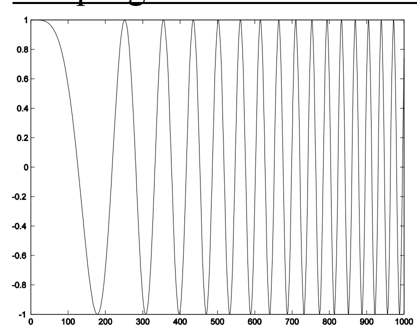
- Frequency actually displayed from top down, but due to “fft” also displayed from down up.
- Time displayed increasing from left to right.
- Vertical slices are spectra at different times.
- Horizontal slices are presence/absence of a specific frequency as time varies.
- Brightness indicates strength at that time-freq.

Example: chirp signal

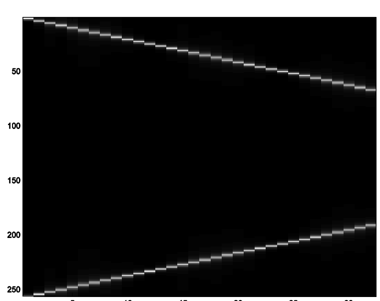
- **Chirp**: x(t)=cos(2πFt^2): birds, dolphins.
- **Frequency** increases linearly with time.
- Instantaneous frequency=2Ft (not Ft) Hertz.

- >>X=cos([0:8191].^2/10000);plot(X(1:1000))
- >>imagesc(abs(fft(reshape(X',256,32))))
- >>colormap(gray) shown on next 2 slides.

Chirp signal: time waveform



Chirp signal: spectrogram

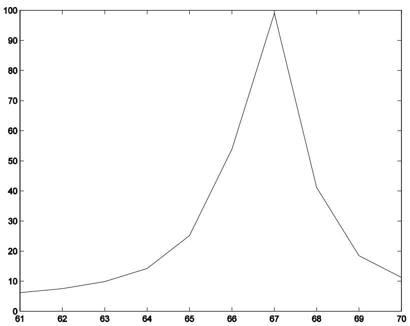


Chirp: Instantaneous frequency

>>X=cos([0:8191].^2/10000) means this:

- x(t)=cos(t^2) sampled at t=n/100; duration=81.9
- Instantaneous frequency (2t)/(2π) Hertz.
- Increases from 0 to 2(81.9)/(2π)=26.08 Hertz.
- Interpret spectrogram: F=100; N=256; T=2.56
- Freq. in final window: (67-1)100/256=25.8 Hz. This is average of freqs in final time window.

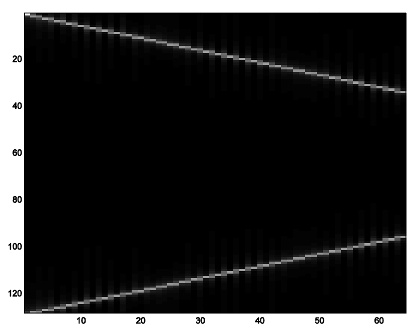
Vertical slice of final (32nd) time window



Time-Frequency Resolution Tradeoff

- Previous spectrogram used 32 slices @256.
- Next slide spectrogram uses 64 slices @128.
- More resolution in time; less in frequency.
- Note vertical “smearing” in frequency bands.
- Tradeoff: time vs. frequency resolution by altering #windows=L and N=length(X)/L.
- Leads to next topic of lecture: data windows.

Chirp signal: spectrogram

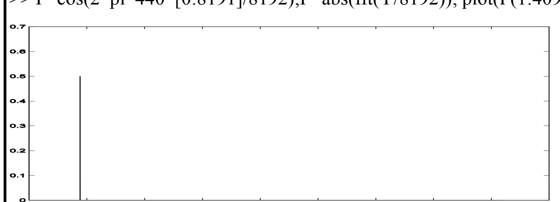


Windows: Interpretation of data

- **Data windows**: used to aid interpretation of spectra computed from real-world data.
- Deliberately distort the spectrum to aid in intepreting it—determine peak locations.
- The correct answer is not always best one: What are you attempting to determine?
- What is the problem? See following spectra:

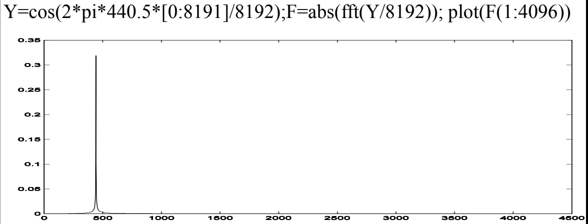
Spectrum: cos(2π440t): length=1 sampled at 8192 samples/second

>>Y=cos(2*pi*440*[0:8191]/8192);F=abs(fft(Y/8192)); plot(F(1:4096))



Spectrum: cos(2π440.5t): length=1 sampled at 8192 samples/second

Y=cos(2*pi*440.5*[0:8191]/8192);F=abs(fft(Y/8192)); plot(F(1:4096))



Why broadening at the base?

- Do not have an integral number of periods.
- So periodic extension of sinusoid is NOT the same as the sinusoid—discontinuities!
- A sinusoid is only a sinusoid over $-\infty<t<\infty$!
- A truncated sinusoid is not same as sinusoid, and it does not have a single-line spectrum.

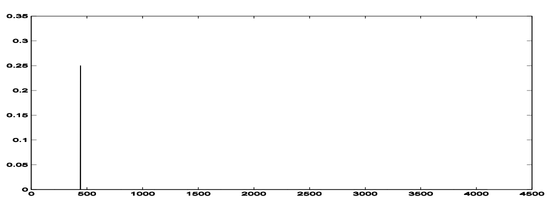
What to do: Data windows

- **Window** data segment to zero at each end.
- Then no discontinuities in periodic extension.
- Effect of windowing: no base, peak broader.
- Easier to interpret peaks without their bases.

- Use N=length(X);W=sin(pi*[0:N-1]/(N-1));
- Many others: (hanning, hamming, kaiser…)

Spectrum: cos(2π440.5t): length=1 sampled at 8192 samples/second

```
>>Y=cos(2*pi*440.5*[0:8191]/8192);Y=Y.*sin(pi*[0:8191]/8192);
>>F=abs(fft(Y/8192)); plot(F(1:4096))  sin window eliminates the base!
```



Use data window on spectrogram

- Previous high-time low-frequency resolution: X=cos([0:8191].^2/10000);colormap(gray); imagesc(abs(fft(reshape(X',128,64))))
- Now use data window on each vertical slice:
- W=ones(64,1)*(sin(pi*[0:127]/127)); imagesc(abs(fft(reshape(X',128,64).*W')))
- Shown on next slide. Note less vertical smearing

Use data window on spectrogram