Engin. 100: Music Signal Processing
Lab #4: Spectrogram

Filtering signals to remove interference

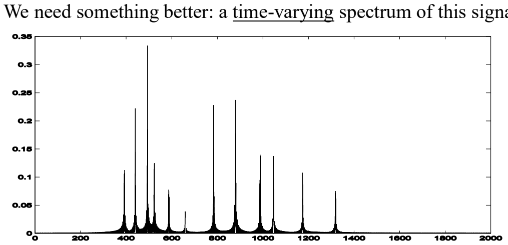
Spectrogram: Time-varying spectrum

Removing Interference from Signals

- Noise: Unknown, unpredictable, but usually dominated by high frequency components.
- Interference: Known signal; want to eliminate. Easy if we have a record of the interference, or if we can generate it. But usually we have neither.
- Lab #3: Two trumpets playing simultaneously. Goal: Filter out one. How? Set its harmonics=0.
- Lab #4: Tonal versions of "The Victors" and awful interference. Goal: Eliminate interference.

Removing Interference from Signals

Spectrum of the signal. How can we tell signal from interference? We need something better: a time-varying spectrum of this signal.



>>FZ=abs(fft(Z)); FZ=FZ(1:20000)*2/78000; F=[0:19999]*8192/78000; plot(F,FZ) where Z is the signal. This doesn't help very much, does it?

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:

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

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

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

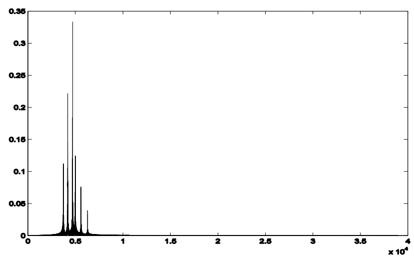
Time-varying spectral analysis

- **IDEA**: Segment (chop up) signal in time.
- **THEN**: Apply "fft" to each signal 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.
- **WHY**: Gives a spectrum that varies with time.

Example: "The Victors" spectrum

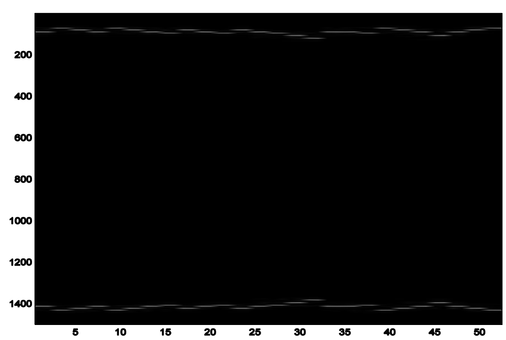
Just a *histogram* of note frequencies (#times each frequency appears)



Example: "The Victors" spectrogram

- X=Tonal "The Victors"; sampled at 8192 Hertz.
- Has 26 notes of length 3000/8192 seconds each.
- Length(X)=78000=26(3000). So L=26, N=3000.
- $\boxed{\text{>>imagesc(abs(fft(reshape(X',3000,26))))}}$, colormap(gray). This is shown on the next slide.
- This is called the "**spectrogram**" of X.
- You can *see* that the signal is a single sinusoid whose frequency *jumps* every 3000/8192 seconds.
- Much more information than just the spectrum!

Example: "The Victors: spectrogram



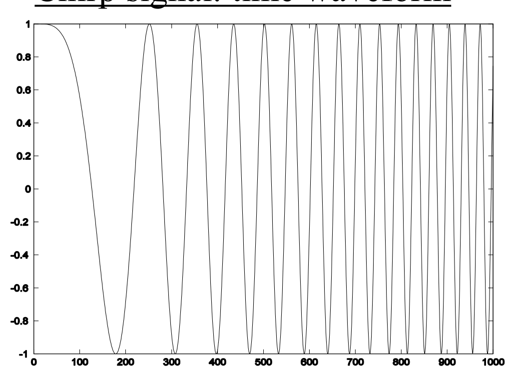
Example: "The Victors: spectrogram

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

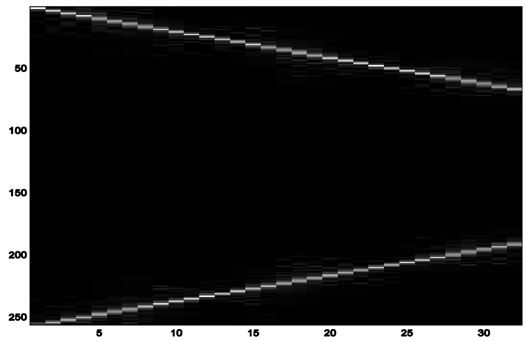
Example: chirp signal

- **Chirp**: $x(t)=\cos(2\pi Ft^2)$: 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

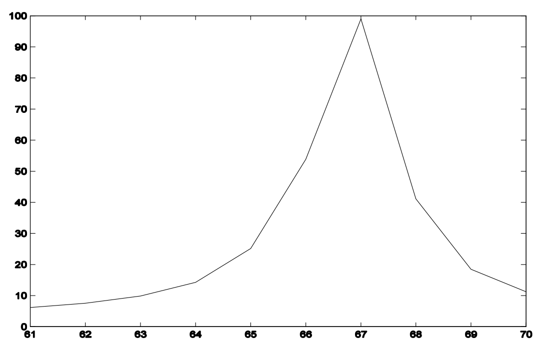


Chirp: Instantaneous frequency

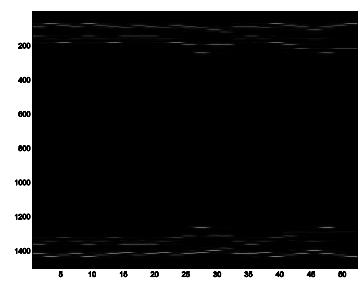
>>X=cos([0:8191].^2/10000) means this:

- x(t)=cos(t^2) sampled at t=n/100; duration=81.9
- The "*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



Removing Interference, Continued: Spectrogram of signal+interference. Now we can see what's happening.



Now you can see what you need to do to eliminate the interference. The rest is up to you in Lab.

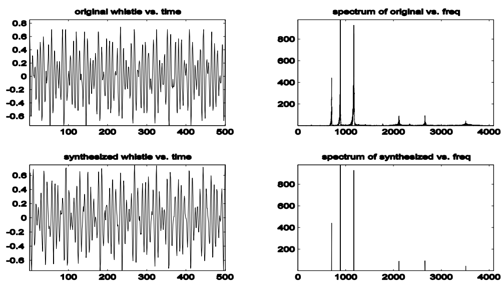
Conclusion

- Can filter out noise from a noisy signal y using: fy=fft(y); fy(K:N+2-K)=0; z=real(ifft(fy)); where the signal has no components above index K.
- Noise tends to be high-frequency, so lowpass filter
- To remove interference, time-varying spectrum, computed as spectrogram, can give a much better picture (visualization!) of what is happening.
- Spectrogram can also aid in interpreting signal.

MORE EXAMPLES OF INTERPRE-TING RESULTS OF MATLAB's "fft"

Not presented in lecture unless time.

Matlab train whistle spectrum: Approximate using 6 sinusoids



Matlab's train whistle signal

- Signal duration=12880 samples=1.57 sec. Take *periodic extension* (repeats; T=1.57).
- Sampling rate=8192 Hertz. So N=ST=12880.
- >>load train.mat;length(y)[=12880];plot(y); F=fft(y);plot(abs(F(1:12880/2))) [1^{st} half only]
- F(k) is component at frequency f=(k-1)S/N. f=(index-1)(8192/12880)=(index-1)/1.57 Hertz.
- [2/N*abs(F) angle(F)] gives numerical values.

Matlab train whistle spectrum

- Only about 6 significant sinusoids present.
- Closer examination of spectrum: Peaks at:

Index	1109	1394	1840	3326	4180	5521
Hertz	705	886	1170	2115	2658	3511

$440\cos(2\pi 705t-1.70)+979\cos(2\pi 886t+2.93)+928\cos(2\pi 1170t+1.35)+88\cos(2\pi 2115t+1.41)+93\cos(2\pi 2658t+0.84)+43\cos(2\pi 3511t+3.03)$

Interpreting results of Matlab's "fft"

- Data acquisition system samples@1024 Hertz. Results loaded into Matlab, in a vector X. **GOAL**: Determine spectrum of loaded data.
- Step #1: >>length(X) gives 3072. Means what?
- Step #1: (3072 samples)/(1024 samples/second). Duration=3 seconds. Take periodic extension.

Interpreting results of Matlab's "fft"

- Step #2: >>F=(2/3072)*abs(fft(X,3072));
- F=0 except indices 97,193,289,2785,2881,2977. F=7 at those indices. How do we interpret this?
- Step #2: Last 3 nonzero mirror images of 1^{st} 3: 2977=3074-97;2881=3074-193;2785=3074-289.
- Three sinusoidal components at frequencies determined by table on next slide.

Interpreting results of Matlab's "fft"

K=Matlab indices of nonzero values of F	97	193	289
Hertz=(K-1)1024/3072=(K-1)/T (N=FT)	32	64	96

$\text{Data}(t)=7\cos(2\pi 32t+\theta_1)+7\cos(2\pi 64t+\theta_2)+7\cos(2\pi 96t+\theta_3)$.

Phase θ_1: >>angle(F(97)). θ_2: >>angle(F(193)). θ_3: >>angle(F(289)).

Harmonic frequencies: Data(t) periodic with period=1/32 second.

Don't confuse this with periodic extension of Data(t) (3 seconds).