From Fourier Series to Analysis of Non-stationary Signals – VIII

prof. Miroslav Vlcek

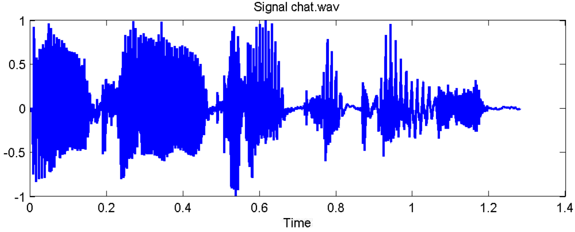
November 29, 2016

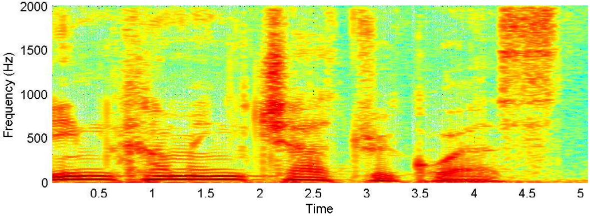
Contents

Contents

Contents

Comments to project on speech analysis





click to play

Comments to project on speech analysis

- Speech is non-stationary signal where properties change quite rapidly over time.
- For most phonemes the properties of the speech remain invariant for a short period of time ($\approx$ 5 to 100 ms).
- These segments are assumed to be stationary and we can use DFT for any $\approx$ 5 to 100 ms segment.

Comments to project on speech analysis

- Most of speech processing is done by taking short overlapping windows and processing them.
- Windowing: a long signal is multiplied with a window function of finite length, giving finite length weighted version of the original signal.

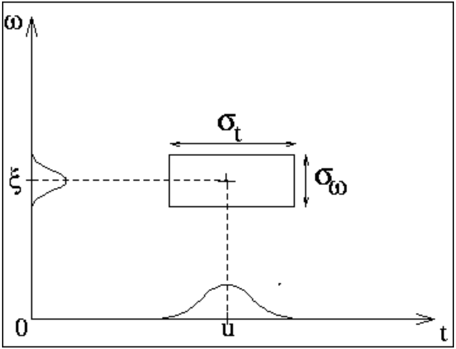
Comments to project on speech analysis

- In speech processing the shape of the window function is not that crucial.
- Usually some soft window like Hanning, or Hamming is used. Their sideband lobes as substantially smaller than in a rectangular window.
- In speech recognition the windows are usually overlapping 10 ms each other.

Principle of uncertainty

- If $f(t)$ is non-zero with a compact support, then its Fourier transform cannot be zero on a whole interval.
- If its Fourier transform $F(\jmath\omega)$ is compactly supported, then it cannot be zero on a time interval.
- Hence, even if the Heisenberg constraints are verified, it is impossible to have an function in space $\mathbb{L}^2$ which is compactly supported both in the time and frequency domains.

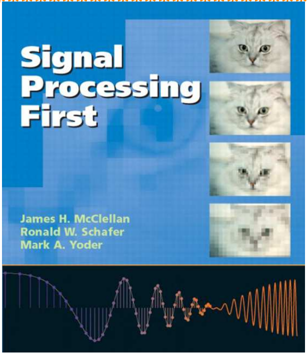
Principle of uncertainty



- In particular, there is no instantaneous frequency analysis for finite energy signals.

MATLAB project with music scale

```
% J. H. McClellan, R. W. Schafer, and M. A. Yoder
% Signal Processing First, ISBN 0-13-065562-7.
% Prentice Hall (c) 2003
% spectrogram of a music scale
% M. Vlcek, Prague, 2010
% ******************************************
```



MATLAB project with music scale

```
% make a scale for C major
c1=40; cis=41; d=42; dis=43; e=44; f=45;
fis=46; g=47; gis=48; a=49; b=50; h=51 c2=52;
keys = [ c1 d e f g a h c2 ];
% key #49 is A
% ********************************************
```

tone	C	D	E	F	G	A	H	C
frequency in Hz	262	294	330	349	392	440	494	523

Table: C major scale

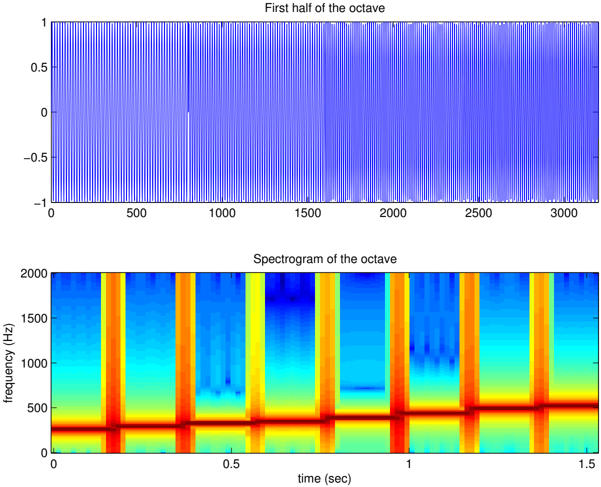
MATLAB project with music scale

```
Fs = 4000;
t0 = 0.2;
tt = 0:(1/Fs):t0;
y2 = [ ];
for k = 1:length(keys)
 keynum = keys(k);
 %add 12 to move up 1 octave
 freq(k) = 440 * (2 .^((keynum-49)/12));
 % based on A=440 Hz
 y2 = [ y2, cos( 2*pi*freq(k)*tt - pi/2 ) ];
end
```

MATLAB project with music scale

```
figure(1)
sound(y2,Fs)
Fmax = Fs/4;
Nfft = 256; Nover = 200;
[B,F,T] = spectrogram(y2,Nfft,Fmax,[],Nover);
imagesc(T,F,db(B,40));
title(['Spectrogram of the octave'])
axis xy, colormap('default')
ylabel('frequency (Hz)'), xlabel('time (sec)')
```

MATLAB project music scale



MATLAB project with music scale

1. Carefully read the help for `spectrogram` and learn how to use the command!
2. Save your sound using `audiowrite(y2, Fs, 'CDscale')`.

   click to play
3. It sounds rather artificially, can you find the way of improving the record? Use a window function – type `help window` in MATLAB!

Christmas HW

- Compose your own carol as Jana Kuklová did !
- Make your composition smart enough and submit as HW6 on December 16, 2016!!

Carol

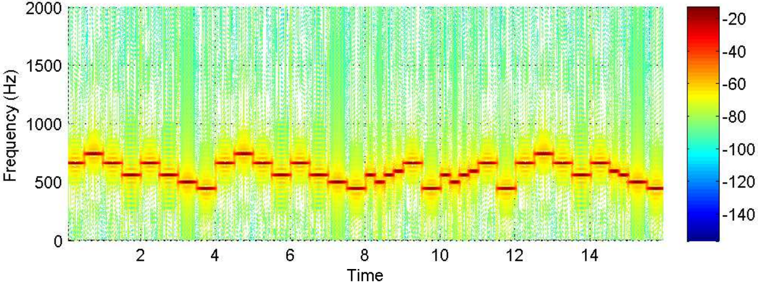


Figure: Spectrogram of a carol

Better sound . . .

How to improve the performance of your music composition

1. Include some harmonics to an original tone, e.g. to the tone A, 440 Hz add 880 Hz, 1320 Hz . . .
2. If you make a spectrogram of a real piano tone A  you can see relative duration of harmonics