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

Miroslav Vlček, Jan Přikryl

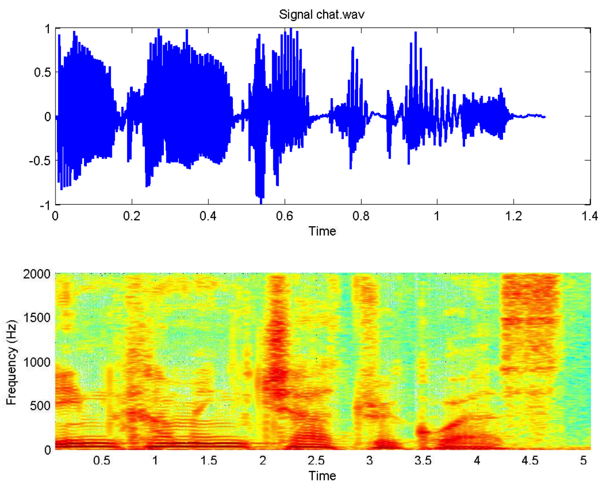
November 25, 2019

Department of Applied Mathematics, CTU FTS

Contents

Non-stationary and Stationary Signals

Comments to project on speech analysis



click to play

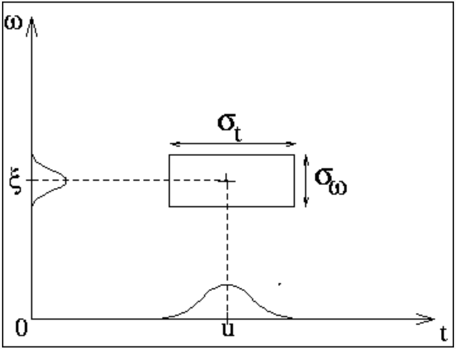
- Speech is non-stationary signal where properties change quite rapidly over time.
- For most phonemes the properties of the speech remainS invariant for a short period of time ($\approx$ 5–100 ms).
- These segments are assumed to be stationary and we can use DFT for any $\approx$ 5–100 ms segment.

- 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

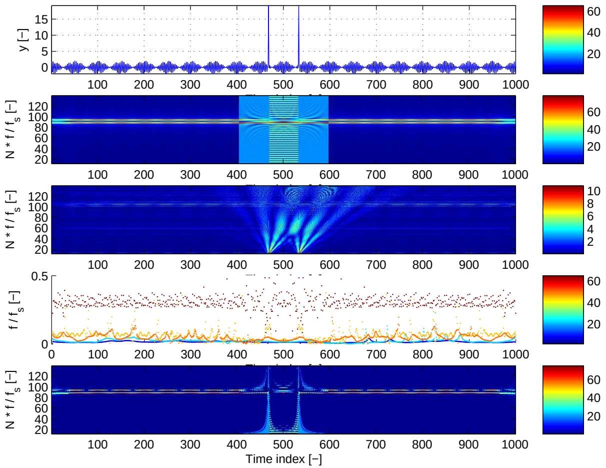
- In speech processing the shape of the window function is not that crucial.
- Usually some "soft" window like Hanning, or Hamming are used. Their sideband lobes are substantially smaller than those of a rectangular window.
- In speech recognition the windows are usually overlapping 10 ms each other.

- 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 $F(\mathrm{j}\omega)$ 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.



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

Discrete Zolotarev Transform

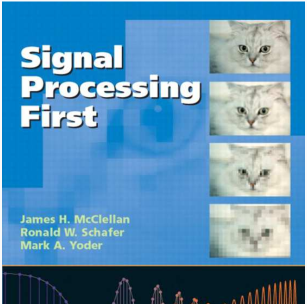


Harmonic signal with two pulses, STFT, WT, HHT and DZT spectrogram

MATLAB project

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
c4=40; cis4=41; d4=42; dis4=43; e4=44; f4=45;
fis4=46; g4=47; gis4=48; a4=49; ais4=50; b4=51;
c5=52;
keys = [ c1 d e f g a h c2 ];
% Remember: key #49 is a4 (i.e. 440 Hz)
```

How to generate tone frequencies?

ČVUT
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

tone	C_4	D_4	E_4	F_4	G_4	A_4	B_4	C_5
f [Hz]	261.63	293.66	329.63	349.23	392.00	440.00	493.88	523.25

Note:

- Every octave has 12 tones
- Every octave doubles the frequency
- Tone frequencies form a geometric series

Example (Frequency of C5 based on A4)

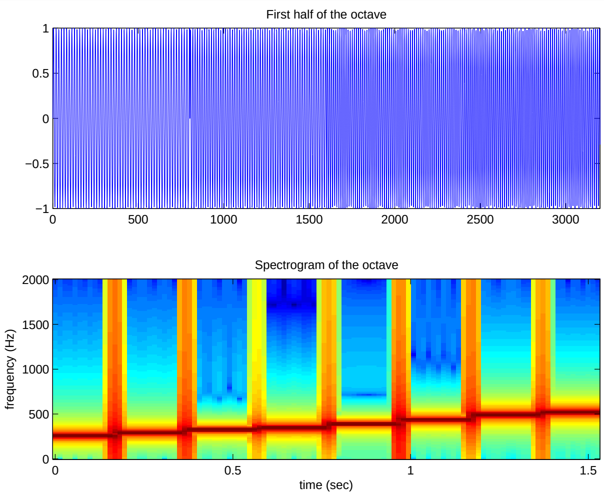
The tone C4 is nine half-tones below A4, and the tone C5 is three half-tones above A4, therefore

$$f(C_4) = f(A_4) \cdot 2^{(-9/12)} = 440 \cdot 2^{-0.75} = 261.63$$
$$f(C_5) = f(A_4) \cdot 2^{(3/12)} = 440 \cdot 2^{0.25} = 523.25$$

```
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 = 440 * (2 .^((keynum - 49)/12));
  % based on A=440 Hz
  y2 = [ y2, cos( 2*pi*freq*tt - pi/2 ) ];
end
% play it
sound(y2, Fs);
```

```
Fmax = Fs/4;
Nfft = 256;
Nover = 200;
% Old approach to generating a spectrogram
[B,F,T] = specgram(y2, Nfft, Fmax, [], Nover);
figure(1);
imagesc(T, F, db(B, 40)); % Amplitude in decibels!
title('Spectrogram of the octave');
axis('xy');
colormap('default');
ylabel('Frequency [Hz]');
xlabel('Time [sec]');
```

MATLAB project with music scale



1. Replace the old Matlab command `specgram` with `spectrogram`.
2. Before applying this, carefully read the help for `spectrogram`!
3. Save your sound using `audiowrite('CDscale',y2, Fs)`.

   click to play
4. It sounds rather artificially, can you find the way to improve the generated record?

ČVUT ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

1. Replace the old Matlab command `specgram` with `spectrogram`.
2. Before applying this, carefully read the help for `spectrogram`!
3. Save your sound using `audiowrite('CDscale',y2, Fs)`.

   click to play
4. It sounds rather artificially, can you find the way to improve the generated record? Hint: Use a window function to overlap the tones!

ČVUT ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

1. Replace the old Matlab command `specgram` with `spectrogram`.
2. Before applying this, carefully read the help for `spectrogram`!
3. Save your sound using `audiowrite('CDscale',y2, Fs)`.

   click to play
4. It sounds rather artificially, can you find the way to improve the generated record? Hint: Use a window function to overlap the tones! Hint: Add harmonics!

MATLAB project with music scale

ČVUT ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

1. Replace the old Matlab command `specgram` with `spectrogram`.
2. Before applying this, carefully read the help for `spectrogram`!
3. Save your sound using `audiowrite('CDscale',y2, Fs)`.

   click to play

4. It sounds rather artificially, can you find the way to improve the generated record? Hint: Use a window function to overlap the tones! Hint: Add harmonics! If you want to be even cooler, look at additive, subtractive or FM synthesis.

Homework

Homework 6

a) Select a Christmas carol that has not been composed in C-major scale.

b) Generate the tone scale using the non-windowed approach.

c) Compose the Carol using Matlab commands

d) Compose a Christmas carol with a tone scale improved by windowing.

e) Deliver the code that generates the `.wav` format as ZIP file by December 11, 2019.