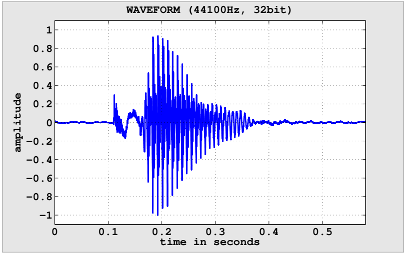
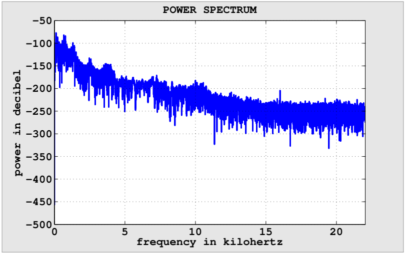
Acoustic signals/Short-time analysis

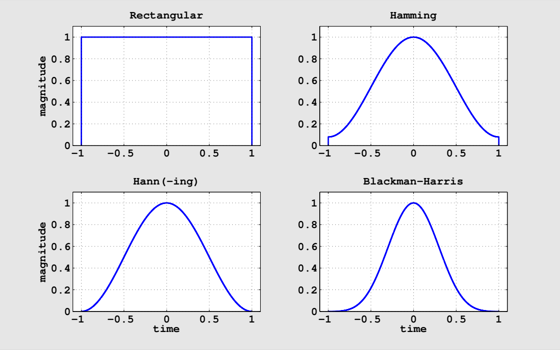
- **spectral and temporal analysis** is essential for speech acoustics
- problem:
  - **power spectrum** has no temporal information anymore (`matlab/tam.wav`)





- solution:
  - choose **short overlapping segments** (windows) at different time points
  - length of the segments (**window size**) is crucial
  - **overlap** and **window function** control spectral leakage
  - aligning Fourier transforms of these (altered) segments leads to **spectrograms**

- **example**: `matlab/windows.m`



- **optimal overlapping** for minimal spectral leakage

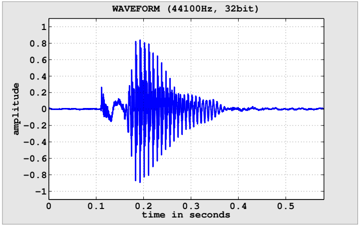
Rectangular:	any value
Hamming:	50%
Hann(-ing):	50%
Blackman-Harris:	66.1%

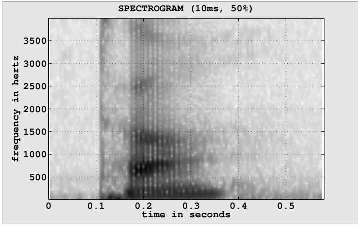
- other commonly used window functions: **Welch, Kaiser, Gaussian**, …

Acoustic signals/Spectrograms

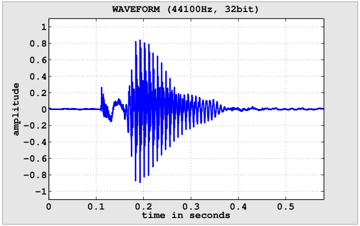
Spectrograms

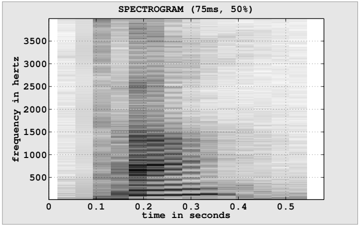
- **example:** `matlab/specgram.m` (`matlab/tam.wav`)





- **example:** `matlab/specgram.m` (`matlab/tam.wav`)





- **exercise:**
  - impact of **window size** → **broad-band** vs. **narrow-band** spectrogram

- **broad-band spectrograms** have good temporal but poor spectral resolution
- **narrow-band spectrograms** have poor temporal but good spectral resolution

spectrogram:	**broad-band**	**narrow-band**
window size:	< 20 ms	> 20 ms
structures:	**formants**	**harmonics**

- set up windowing

```
>> wsize = 10; % window size in milliseconds
>> woverlap = 66; % window overlap in percent
>> wfunc = @blackmanharris; % window function
```

- compute the spectrogram

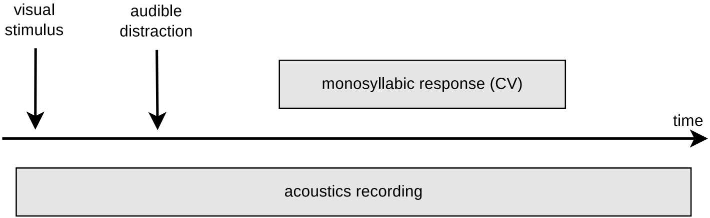
```
>> [Xk, fk, ti] = spectrogram( xi, ... % signal
      wfunc( ceil( wsize/1000 * fS ) ), ... % window function values
      ceil( woverlap/100 * wsize/1000 * fS ), ... % window overlap samples
      4096, fS ); % fourier transform
```

- plot the spectrogram

```
>> colormap( flipud( colormap( 'gray' ) ) );  % set color coding
>> imagesc( ti, fk, Pk .^ 0.1 );  % plot spectral powers
```

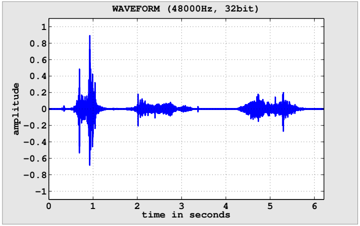
Acoustic signals/Activity detection

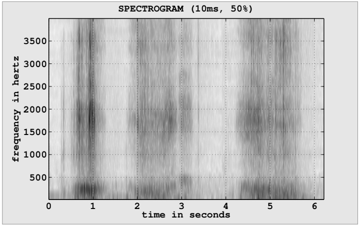
- **experimental data** often contain a lot of noise and little of information
- for **automatic processing** restriction to important parts is essential
- consider the following experiment:



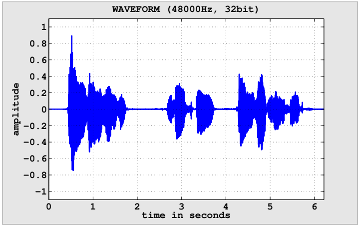
- with features of interest:
  - responded syllable (out of a specific set → classifcation task)
  - voice onset time (→ **landmarks detection**)
  - formants onsets (frequency and time → **formants tracking/detection**)
- all of these require (human) **activity detection** as an initial processing pass

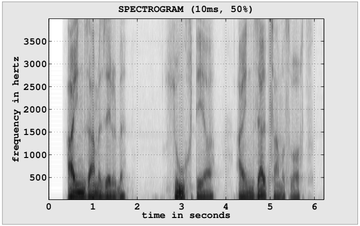
- in literature usually called **voice activity detection** (VAD)
- exploiting **spectral differences** in human speech and ambient sound/noise
- **example:** `matlab/specgram.m` (`matlab/chair.wav`)



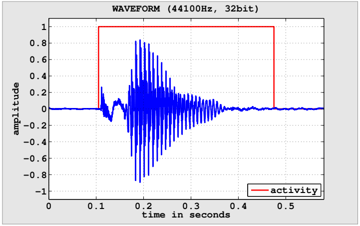


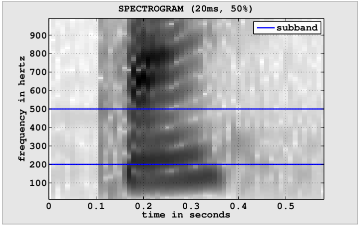
- **example:** `matlab/specgram.m` (`matlab/haiku.wav`)



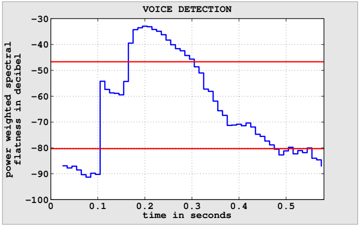


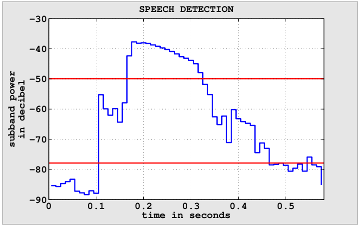
- **example:** `matlab/activity.m` (`matlab/tam.wav`)
- applying an **equal loudness filter** and limiting to **frequency band** 0 . . . 1000 Hz





- **adaptive thresholds** for power-weighted **spectral flatness** and **subband power**





- combining **activity states** based on thresholds gives overall **voice activity**

- D. Robinson. *Equal loudness filter.* 2001.
- D. Burileanu, L. Pascalin, C. Burileanu, M. Puchiu. *An adaptive and fast speech detection algorithm.* Springer, 2000.
- M. Prcin, L. Müller. *Heuristic and statistical methods for speech/non-speech detector design.* Springer, 2002.
- Y. Ma, A. Nishihara. *Efficient voice activity detection algorithm using long-term spectral flatness measure.* Springer, 2013.