Spectral analysis II

Honza Černocký, ÚPGM

Please open Python notebook **03_spectral_2**

Agenda

- **DFT refresher**
- DFT of typical signals
- Properties of DFT
- Some more math on DFT
- Spectrogram and what can we do with it
- Summary and “take-home” messages

Discrete Fourier transform (DFT) analysis

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n}$$

- Input: signal *x[n]* of *N* samples
- Output: *N* **complex** coefficients *X[k]* providing the information on
  - Frequency – index *k*, corresponds to normalized frequency *k/N* and regular frequency $k/N\ F_s$. Normalization and de-normalization of frequency involves a simple division or multiplication by sampling frequency F_s.
  - How much ? Magnitude (absolute value, modul) |*X[k]*|.
  - How shifted ? Phase (angle, argument, phase shift) arg *X[k]*.
- Easy to implement by definition but using FFT is much faster (for $N = 2^b$)

DFT analysis II.

- For real input signals $x[n]$, coefficients $X[k]$ are symmetrical (complex conjugate $X[k] = X^*[N-k]$:
  - $|X[k]| = |X[N-k]|$ and $\arg X[k] = -\arg X[N-k]$
- … so that it is enough to store and visualize the spectrum only from $k = 0$ to $k = N/2$. This corresponds to normalized frequencies $0 \ldots \frac{1}{2}$ and regular frequencies $0 \ldots F_s/2$.
- Attention, the number of coefficients to retain is **NOT** $N/2$ but $N/2+1$!!!

#dft_anal

DFT synthesis - IDFT

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{+j2\pi \frac{k}{N} n}$$

- Input: *N* **complex** coefficients *X[k]*
- Output: *N* samples of signal $x_s[n]$.
  - In case we did nothing with the spectrum, $x_s[n]$ is exactly the same as *x[n]*.
- Synthesis by DFT involves summing pairs of complex exponentials at *k* and *N-k* running “against each other”, creating one shifted cosine – see derivation in the last lecture.
  - When computing, DFT/FFT can produce very small imaginary components, better to kill them using `np.real()`.

#dft_synt

Agenda

- DFT refresher
- **DFT of typical signals**
- Properties of DFT
- Some more math on DFT
- Spectrogram and what can we do with it
- Summary and “take-home” messages

DFT of a D.C. (stejnosměrný) / constant signal

- *x[n] = a*

$$X[k] = \sum_{n=0}^{N-1} a e^{-j2\pi \frac{k}{N} n}$$

- For *k=0*, the sum is *X[0] = Na*.
- For everything else, the sum of *k* “revolutions” of the complex exponential is zero.
- Intuition check: D.C. has just zero frequency, everything else should be zero - **OK**

Very slow signal <=> narrow spectrum

#dft_of_constant

DFT of a pulse (impuls) at time zero

- *x[0] = a,* otherwise x[n] = 0.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n} = a e^{-j2\pi \frac{k}{N} 0} = a e^{0} = a$$

- Only the term for *n=0* will produce something - here, the complex exponential is exp*(0) = 1*, so *X[k] = a* for all *k*'s
- Intuition check: the pulse is very fast (like a click), should contains lots of frequencies - **OK**

Very fast signal <=> wide spectrum

#dft_of_pulse

DFT of a shifted pulse (posunutý impuls)

- $x[g] = a$, otherwise x[n] = 0. g is the position of the pulse.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n} = a e^{-j2\pi \frac{k}{N} g}$$

- Only the term for $n=g$ will produce something - here, the complex exponential is exp$(-2\pi\, g\, k\, /\, N)$, so
  - $X[k] = a$ exp$(-2\pi\, g\, k\, /\, N)$.
  - Magnitude is $|X[k]| = a$
  - Phase is arg (a exp$(-2\pi\, g\, k\, /\, N)) = -2\pi\, g\, k\, /\, N$ … linear with slope $-2\pi\, g\, /\, N$
- For visualization, you might try using `np.unwrap` to see values of phase outside of $-\pi$ … $+\pi$.
- Intuition check: the signal only shifted, the magnitudes should not change, shift should be reflected in phases - **OK**

#dft_of_shifted_pulse

DFT of a rectangular pulse (pravoúhlý / obdélníkový impuls)

- $x[n] = a$ for $n = 0...g-1$, zero otherwise (so it will have g non-zero samples).
- No derivation (we'll see this later) but remember that wide signal should have narrow spectrum and narrow signal should have wide spectrum …

#dft_of_rectangle

- The function we see is $sin(x) / x$, so called **cardinal sine** (kardinální sinus). We'll be seeing this a lot …
- For narrow rectangles, the cardinal sine lobes (laloky) seem to be distorted as we move towards $F_s/2$ – this is caused by **aliasing** (more about it in the lecture on sampling)

DFT of a “symmetrical” rectangular pulse

- Trying to make the rectangle symmetrical so that the phase is not shifted.
- Will work only for odd (liché) *g* by setting the same amount of samples for negative *n* as for positive *n*. For example, for *g = 5*, set *x[-2 … +2] = a*
- However, we can not work with negative indices, so we need to consider the end of input buffer as negative indices …
  x[0 … 2] = a, x[N-2] = a, x[N-1] = a

#dft_of_symmetrical_rectangle

- The phase is either 0 or $\pm\pi$, this means that this spectrum is real – just positive or negative numbers !

Symmetrical signal <=> real spectrum

DFT of cosine with the period of *N* samples

$$x[n] = A\cos(2\pi \frac{1}{N} n + \phi)$$

- We could run the DFT analysis, but let's work the "lazy" way:
  - Split the cosine using our well known formula $\cos\alpha = \dfrac{e^{j\alpha} + e^{-j\alpha}}{2}$

$$x[n] = \frac{A}{2} e^{j\phi} e^{j2\pi \frac{1}{N} n} + \frac{A}{2} e^{-j\phi} e^{-j2\pi \frac{1}{N} n}$$

  - Write the synthesis formula next to it and try to see if we can find respective coefficients $x_s[n] = \dfrac{1}{N} \displaystyle\sum_{k=0}^{N-1} X[k] e^{+j2\pi \frac{k}{N} n}$
  - We easily find $X[1] = N\frac{A}{2}e^{j\phi}$
  - The next one we could find is X[-1] which is not there (can use only indices from 0 … N-1) but we already know about the symmetry: $X[-1] = X[N-1] \quad X[N-1] = N\frac{A}{2}e^{-j\phi}$
- Coefficients *X[1]* and *X[N-1]* for a cosine are almost the same, except for the phase $|X[1]| = |X[N-1]| = N\frac{A}{2} \qquad \arg X[1] = \phi, \qquad \arg X[N-1] = -\phi \qquad X[1] = X^\star[N-1]$

#dft_of_cos1 - OMG, what's going on with the phase ???

DFT of cosine with more periods in *N* samples

$$x[n] = A\cos(2\pi \frac{g}{N} n + \phi)$$

- We could find the values of coefficients exactly in the same way, just looking at index *g* instead of 1
- Exactly the same, only the index will change: $X[g] = N\frac{A}{2}e^{j\phi}$

  $X[-g] = X[N-g]$ $\quad$ $X[N-g] = N\frac{A}{2}e^{-j\phi}$

  - Magnitudes are the same, arguments opposite:

  $|X[g]| = |X[N-g]| = N\frac{A}{2}$ $\quad$ $\arg X[g] = \phi,$ $\quad$ $\arg X[N-g] = -\phi$ $\quad$ $X[g] = X^\star[N-g]$

#dft_of_cos_multiperiod

DFT of a cosine that does not fit exactly *N* samples …

- In real life, we can not say "I'm going to analyze only signals with period od 128 samples". We obtain an input and we have to go …
- So it can happen (and it regularly happens) that our cos is "badly cut".
- Remember: any sharp edge in a signal generates high frequencies.

#dft_of_general_cos

For a period that does not fit into N exactly k times ("wrong cut" of cos), the spectrum is "leaking" to neighboring frequencies.

Why ? Because we're windowing our signal …

- $x[n] = w[n]\, s[n]$, where $w[n]$ is the window function.
- In case of multiplication of signals in time, we'll have a **convolution** (konvoluce) of spectra in frequency.
- More about convolution in the filtering lecture, for cosine signal, it simply means, that the spectrum of $w[n]$ shifts to its frequency
- So what is the spectrum of our window ?
  - Analyzing N samples is not enough, we'd get a constant (and we know it's spectrum is a single coefficient).
  - Increasing N to more samples by zero padding: 256 -> N_{fft} = 4096 (16x)
  - Zoom only on low frequencies to see details … and showing only magnitude
  - … also showing where are the points of length *N* DFT !

#dft_rectangle_zeropad

Better frequency resolution for a “good” cosine

$$x[n] = A\cos(2\pi \frac{g}{N} n + \phi)$$

#dft_of_cos_multiperiod_detail

- Here, we see the cardinal sine, but the frequencies of the original DFT sample it exactly at zero values !

Better frequency resolution for a “bad” cosine

#dft_of_cos_generalcos_detail

- We see the cardinal sine, and we are not lucky, so the theoretical 1 line “leaks” into neighboring frequencies.
- We can try to fix it by another window … Hamming

$$w[n] = 0.54 - 0.46 \cos \frac{2\pi n}{N-1} \quad \text{for} \quad 0 \leq n \leq N-1, \qquad 0 \text{ elsewhere}$$

#dft_of_generalcos_Hamming

- Less “messy” spectrum than the rectangular window (the edges resulting from “bad cutting” are attenuated) but also worse frequency resolution.
- Remember: smoother signal <=> wider spectrum.

DFT of something periodic

- Periodic rectangular pulses with period N_{pulse}
- Fitting N samples with an integer number of periods

#dft_pulsetrain

- Compared to the spectrum of one rectangular pulse, the spectrum is **sampled**.
- The positions of samples correspond to $1/ N_{pulse}$ (normalized frequency) and F_s / N_{pulse} (regular frequency).

Periodic signal <=> sampled “discrete” spectrum

DFT of a periodic signal with a “wild” period

- Periodic rectangular pulses with period N_{pulse}
- Not fitting N samples with an integer number of periods (which is the usual situation …)

#dft_pulsetrain_general

- It does something but there is a brutal “leaking” into neighboring frequencies impacting also the shape of the spectrum.
- However, this is what we’re doing most of the time – see the speech signal #dft_anal at the beginning of this lecture

In case we know the period, or are able to measure it precisely, DTFT might be a safer option.

Agenda

- DFT refresher
- DFT of typical signals
- **Properties of DFT**
- Some more math on DFT
- Spectrogram and what can we do with it
- Summary and “take-home” messages

Scaling

- If the signal is multiplied with a constant, *y[n] = a x[n]*, the DFT coefficients will be *Y[k] = a X[k]*
- Proof:

$$X[k] = \sum_{n=0}^{N-1} a x[n] e^{-j2\pi \frac{k}{N} n} = a \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n} = a X[k]$$

#dft_scaling

Additivity

- If
  - DFT for signal $x_1[n]$ is $X_1[k]$
  - and DFT for signal $x_2[n]$ is $X_2[k]$
- Then, if we sum the signals: $y[n] = x_1[n] + x_2[n]$
- The resulting DFT will be also the sum of the original ones:
  $Y[k] = X_1[k] + X_2[k]$
- Proof:

$$X[k] = \sum_{n=0}^{N-1} (x_1[n] + x_2[n]) e^{-j2\pi \frac{k}{N} n} = \sum_{n=0}^{N-1} x_1[n] e^{-j2\pi \frac{k}{N} n} + \sum_{n=0}^{N-1} x_2[n] e^{-j2\pi \frac{k}{N} n} = X_1[k] + X_2[k]$$

#dft_additivity

… why don't the values of phases match at some points ?

Linearity

- Scaling and additivity together
- If DFT for signal $x_1[n]$ is $X_1[k]$ and DFT for signal $x_2[n]$ is $X_2[k]$
- Then, if we do a **weighted sum** the signals:
  $y[n] = a_1 x_1[n] + a_2 x_2[n]$
- the resulting spectrum will be exactly the same weighted combination of the original ones: $Y[k] = a_1 X_1[k] + a_2 X_2[k]$

$$X[k] = \sum_{n=0}^{N-1} (a_1 x_1[n] + a_2 x_2[n]) e^{-j2\pi \frac{k}{N} n} = a_1 \sum_{n=0}^{N-1} x_1[n] e^{-j2\pi \frac{k}{N} n} + a_2 \sum_{n=0}^{N-1} x_2[n] e^{-j2\pi \frac{k}{N} n} = a_1 X_1[k] + a_1 X_2[k]$$

#dft_linearity

Linearity is absolutely crucial in many applications, and we intentionally use non-linearities in others – neural networks.

Shifts of short signals

- Consider a relatively short signal *x [n]* that has DFT *X[k].*
- What happens if this signal is shifted to the right (delayed) by *m* samples*: y[n] = x[n-m]* (let's consider that the signal does not leave the interval *0 … N-1*)
- The complex exponentials "seeing" the signal will be *m* samples "older", so that we can re-write the DFT as

$$Y[k] = \sum_{n=0}^{N-1} y[n] e^{-j2\pi \frac{k}{N} n} = \sum_{n=0}^{N-1} x[n-m] e^{-j2\pi \frac{k}{N} n} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} (n+m)} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} (n+m)} =$$

$$\sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n} e^{-j2\pi \frac{k}{N} m} = e^{-j2\pi \frac{k}{N} m} \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n} = e^{-j2\pi \frac{k}{N} m} X[k]$$

- The resulting coefficient is the original one multiplied with a "correction factor": $Y[k] = e^{-j2\pi\frac{k}{N}m}X[k]$
  - The magnitude of the correction is 1, so $|Y[k]| = |e^{-j2\pi\frac{k}{N}m}|\;|X[k]| = |X[k]|$
  - The phase of the correction is shifted by a linear function.

$$\arg Y[k] = \arg e^{-j2\pi\frac{k}{N}m} + \arg X[k] = \arg X[k] - 2\pi\frac{k}{N}m$$

The magnitude spectrum will be the same, the phase one will be inclined "downhill" for delay and "uphill" for advancing the signal.

#dft_shifted_lin

Shift of a signal that covers the whole period

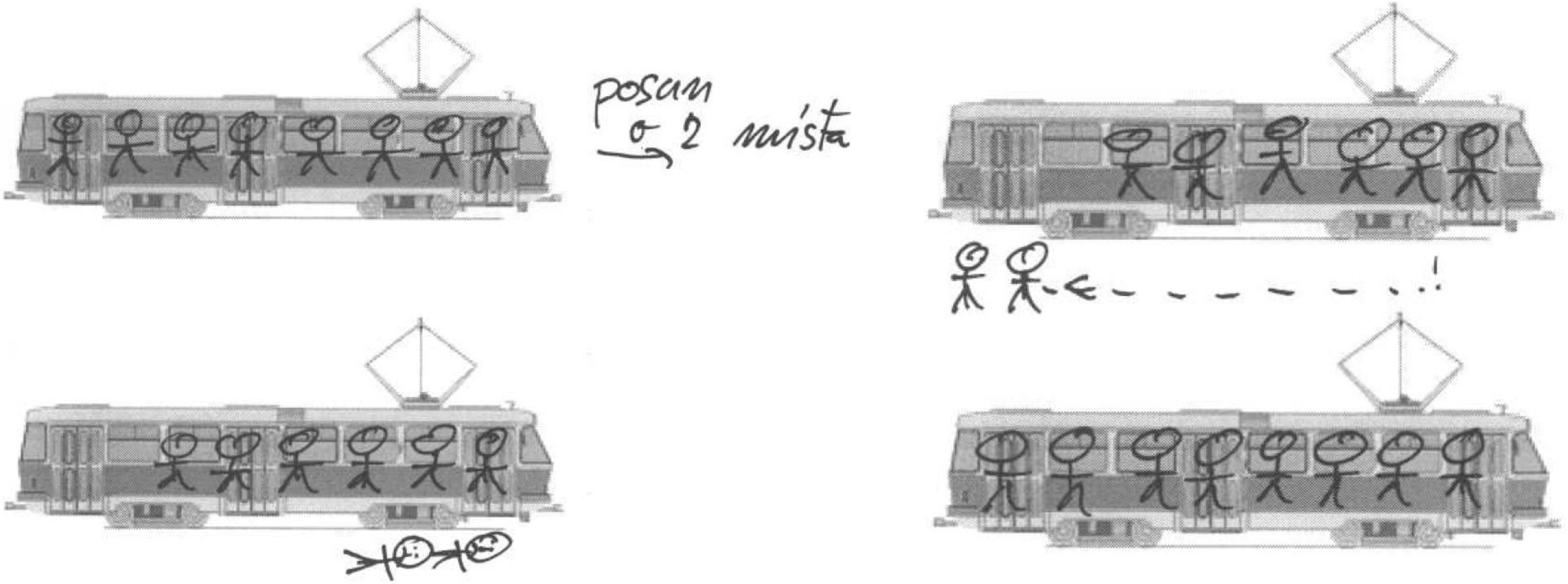
- Consider a longer signal, for example a cosine nicely fitting in N samples
- When shifted, “gaps” appear, and creates edges generating unwanted frequencies …

#dft_shifted_lin_long

Ooops, that’s a mess ☹

Correct way to shift signals for DFT …

- Remember: we need to always stay in the buffer *0 … N-1*
- In case *m* samples of the signal are pushed out at the end, they need to return to the beginning.



Circular shift mathematically

- **Modulo** *N* function always returns values from *0 … N-1*
- Implementing circular shift by modulo indexing:
  $y[n] = x[\,\mathrm{mod}_N\,(n-m)]$
- For such a circular shift, we can safely use the correction computed above - the resulting coefficient is the original one multiplied with a "correction factor": $Y[k] = e^{-j2\pi\frac{k}{N}m} X[k]$
  - The magnitude of the correction is 1, so $|Y[k]| = |e^{-j2\pi\frac{k}{N}m}|\;|X[k]| = |X[k]|$
  - The phase of the correction is shifted by a linear function.

$$\arg Y[k] = \arg e^{-j2\pi\frac{k}{N}m} + \arg X[k] = \arg X[k] - 2\pi\frac{k}{N}m$$

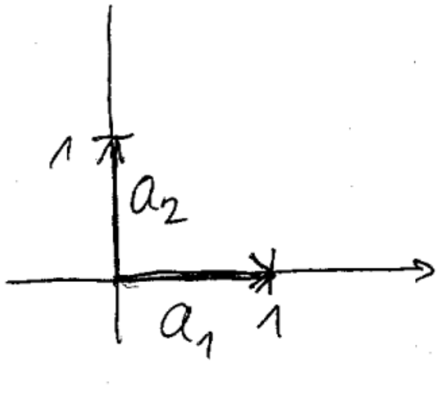
The magnitude spectrum will be the same, the phase one will be inclined "downhill" for delay and "uphill" for advancing the signal.
#dft_shift_circular

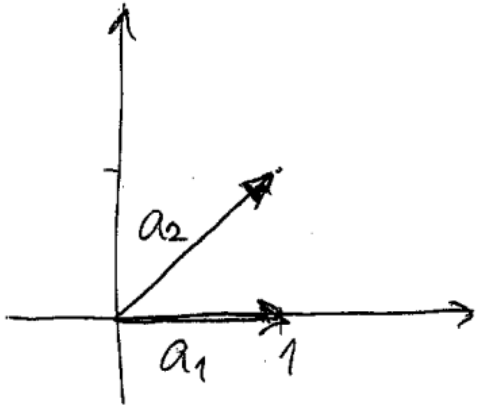
Agenda

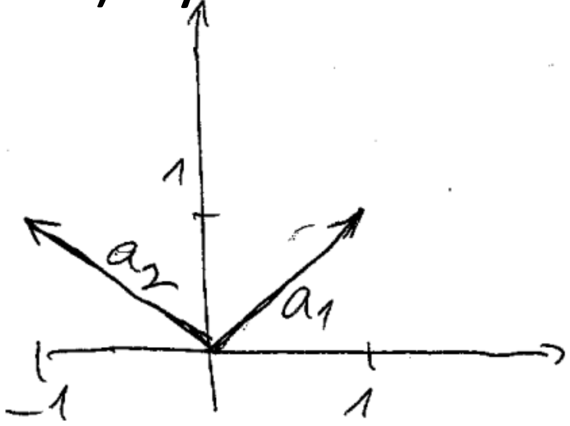
- DFT refresher
- DFT of typical signals
- Properties of DFT
- **Some more math on DFT**
- Spectrogram and what can we do with it
- Summary and "take-home" messages

Orthogonality

- For 2D, we can check the orthogonality (kolmost) by hand …







- Mathematically, checking by dot-product
  - $[1 \;\; 0]\,[0 \;\; 1]^T = 1 \times 0 + 0 \times 1 = 0$ **orthogonal**
  - $[1 \;\; 0]\,[1 \;\; 1]^T = 1 \times 0 + 1 \times 1 = 1$ **not orthogonal**
  - $[1 \;\; 1]\,[-1 \;\; 1]^T = 1 \times (-1) + 1 \times 1 = 0$ **orthogonal**

Are DFT bases orthogonal ?

- We'll check it by a standard dot-product over *N* samples.

$$c = \sum_{n=0}^{N-1} a_k[n] a_l[n] = 0, \quad \text{for} \quad k \neq l$$

- But we have complex bases, so need to do complex conjugation (as for projection)

$$c = \sum_{n=0}^{N-1} a_k[n] a_l^{\star}[n] = 0, \quad \text{for} \quad k \neq l$$

- For DFT bases, we have

$$c = \sum_{n=0}^{N-1} e^{-j2\pi \frac{k}{N} n} e^{-(-j2\pi \frac{k}{N} l)} = \sum_{n=0}^{N-1} e^{j2\pi \frac{l-k}{N} n} = 0, \quad \text{for} \quad k \neq l$$

DFT bases ARE orthogonal

Are DFT bases normal ?

- Norm of each basis must be 1 …

$$||a[n]||_2 = \sum_{n=0}^{N-1} |a[n]|^2$$

- For DFT:

$$||a[n]||_2 = \sum_{n=0}^{N-1} |e^{+j2\pi \frac{k}{N} n}|^2 = \sum_{n=0}^{N-1} 1 = N$$

- The norm is not 1, but N, so we need to correct in the IDFT when synthesizing the signal:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{+j2\pi \frac{k}{N} n}$$

- Remember the multiplication by *N* when computing coefficients for a cosine !

$$|X[g]| = |X[N-g]| = N\frac{A}{2}$$

After IDFT fix, DFT bases are normal, so they are also orthonormal.

DFT extending beyond *k=0 ... N-1* – periodicity

- We have defined DFT to strictly compute the spectrum for *k=0 ... N-1,* this corresponds to
  - Normalized frequencies from *0* to almost *1* (precisely *(N-1) / N*)
  - Regular frequencies from *0* to almost F_s (precisely $F_s(N-1) / N$)
  - And we usually visualize it only till *N/2* (i.e. till ½ in normalized frequencies and $F_s/2$ in regular ones).
- What happens if we go beyond interval *k=0 ... N-1* ?
- Augmenting *k* by a multiple of the number of samples *N*:

$$X[k+gN] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k+gN}{N} n} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi (\frac{k}{N} + \frac{gN}{N}) n} \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n} e^{-j2\pi \frac{gN}{N} n} =$$

$$\sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n} e^{-j2\pi g n} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{k}{N} n} 1 = X[k]$$

- DFT is periodic with *N* samples. In frequencies, this corresponds to periodicity
  - With period of 1 (normalized frequencies)
  - With period of F_s (regular frequencies)
- This is the "mathematical" proof of periodicity, we will see another one in the lecture on sampling !
- We can do a similar proof for DTFT

#dft_periodicity

$$\tilde{X}(e^{j(\omega+g2\pi)}) = \sum_{n=0}^{N-1} x[n]e^{-j(\omega+g2\pi)n} = \sum_{n=0}^{N-1} x[n]e^{-j(\omega n+g2\pi n)} =$$

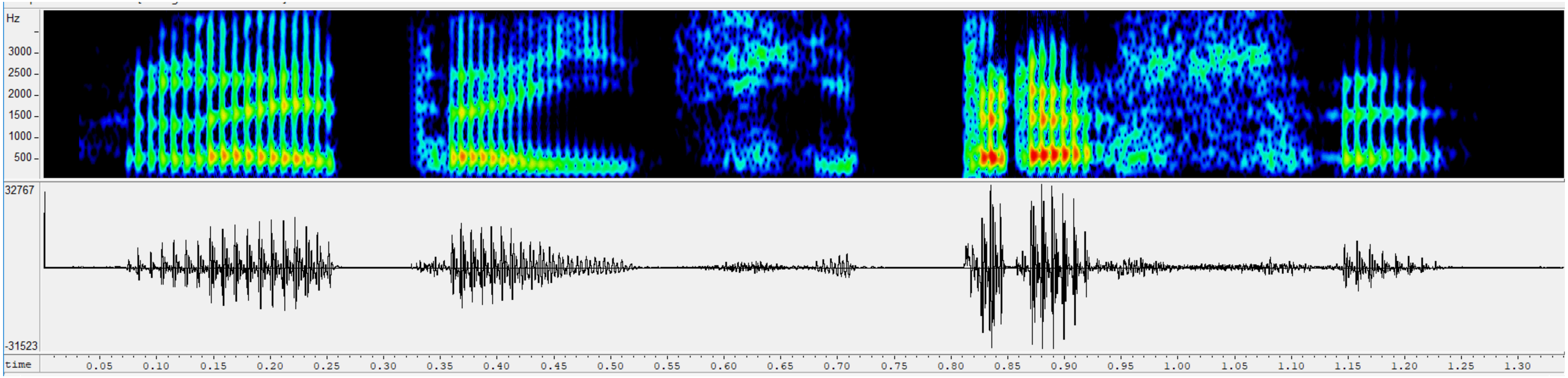
$$\sum_{n=0}^{N-1} x[n]e^{-j\omega n}e^{-jgn2\pi} = \sum_{n=0}^{N-1} x[n]e^{-j\omega n}1 = \tilde{X}(e^{j\omega})$$

Agenda

- DFT refresher
- DFT of typical signals
- Properties of DFT
- Some more math on DFT
- **Spectrogram and what can we do with it**
- Summary and “take-home” messages

Spectrogram

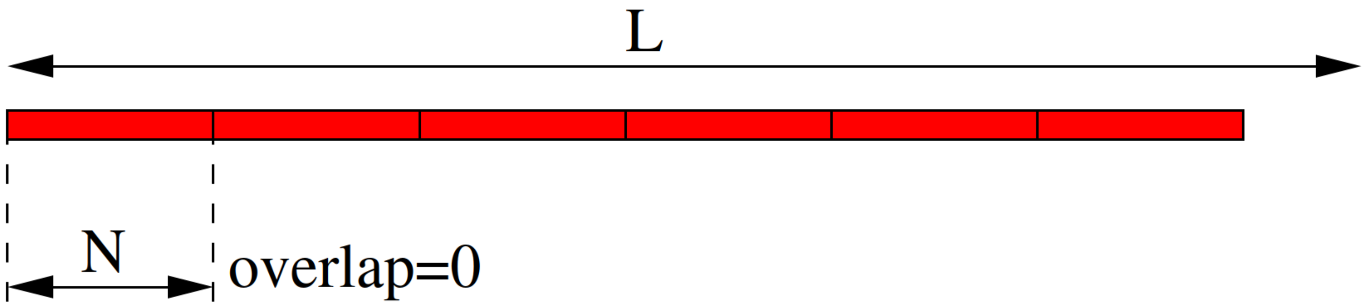
- We want to see the evolution of spectrum over time.
- For our speech signal, want something like this:



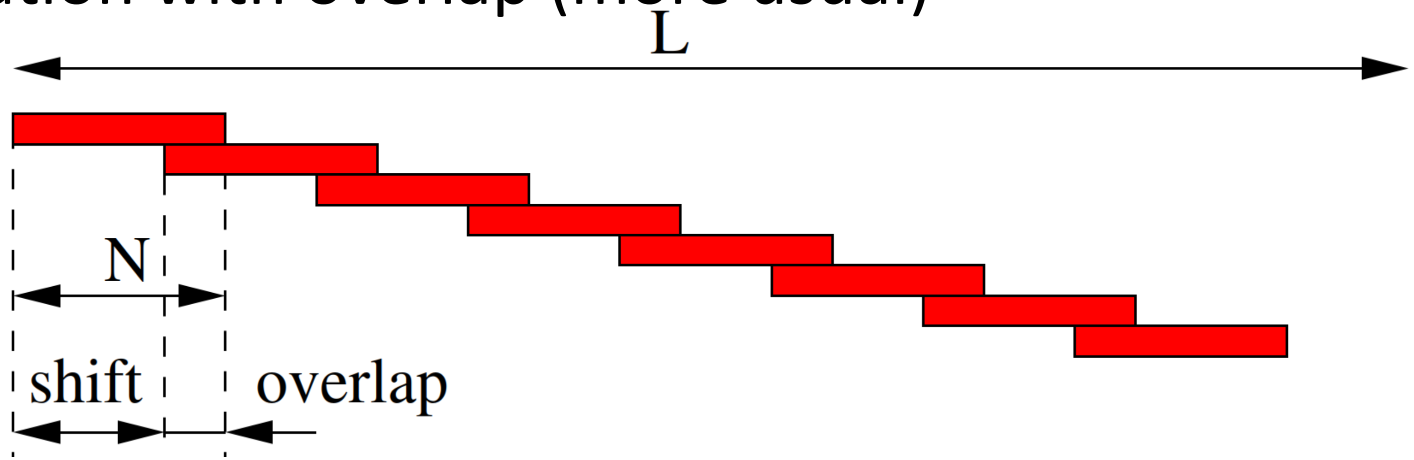
- Calling #spectrogram_blackbox without knowing what it really does.
- We better learn **what the spectrogram does exactly**

Pre-processing of the signal before computation

- Division of a long signal into short segments – frames (rámce)
- Segmentation without overlap



- Segmentation with overlap (more usual)



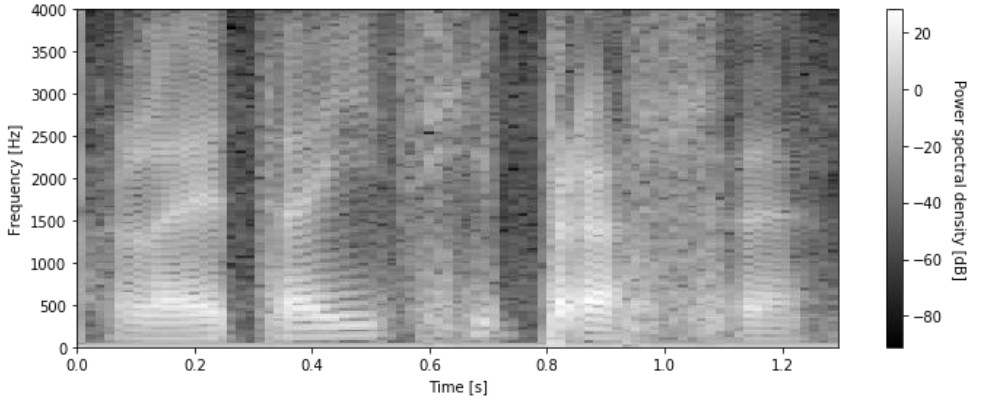
Windowing the frames and running DFT for all …

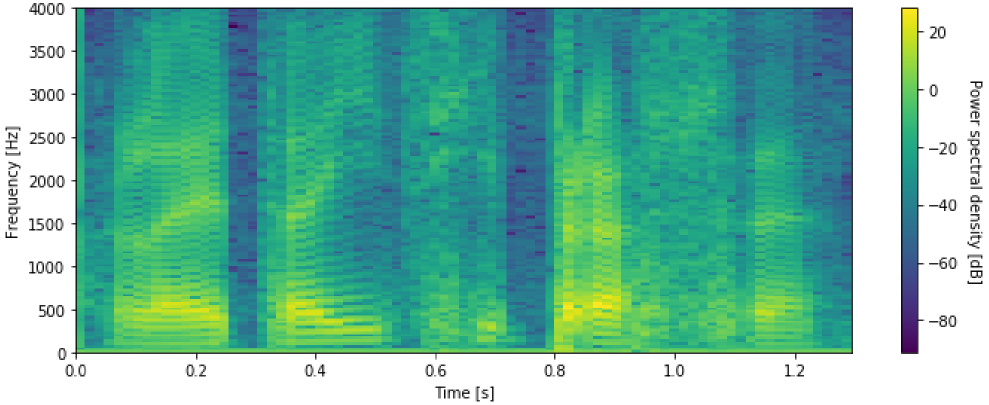
- Window each frame by a selected window
  - Rectangular – good frequency selectivity, more mess at higher frequencies.
  - Hamming – worse frequency selectivity, less mess at higher frequencies.
  - Hanning (or Hann) window – a sequence overlapping by N/2 sums up to 1 – see later. Beware of confusion ‘mm’ vs ‘nn’ …
- If needed, zero-pad to reach the desired number of point N_{fft}.
- As usual, select only $0 \ldots N/2$, resp $0 \ldots N_{fft}/2$ points.
- Visualize

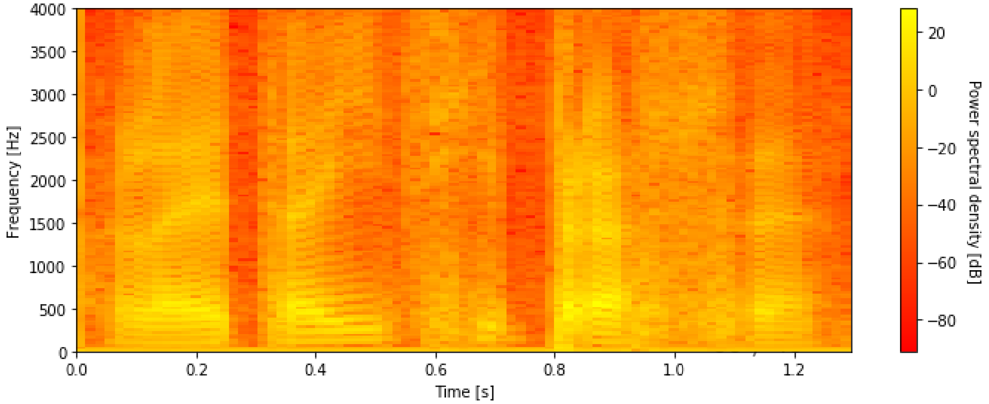
#spectrogram

More on spectrogram visualization

- We usually take the log of so called power spectral density – square of spectrum $10 \log_{10} |X[k]|^2$ and claim it is in decibel [dB]
- Selection of colormap …
- Playing with brightness, contrast, etc …







Short-term vs long-term spectrogram

- Wanted: more details in frequency => set longer frame length => however, less precision in time (averaging details over longer time…) – **long-term spectrogram**
- Wanted: more details in time => set shorter frame length (eventually with more FFT points to have a nice figure) => however, less precision in frequency (the frame “sees” less signal) – **short-term spectrogram**
- Wanted: more frames per second => set more overlap (less frame shift) => however, will not be able to detect short events (still averaged by the frame length…

#spectrogram_long_short_term

Setting good spectrogram parameters is a tricky business and there is no universal solution – ask colleagues and then tune the parameters.

Synthesizing signal from spectrogram

- Need to be careful with the analysis – define frames with ½ frame overlap and use Hanning (Hann) window – their sequence sums exactly to 1.
- When synthesizing, initialize a buffer with the length of the resulting signal by zeros, then run a cycle:
  - For each frame, take spectrum and complete the upper (from $N/2+1 \ldots N-1$) by the symmetry $X[k] = X^*[N-k]$
  - Perform IDFT that generates the current frame.
  - Add it to the buffer
  - Move to the next frame, shift the position where signal will be added to the buffer by frame overlap.

#dft_synt

Playing with spectrogram I – filtering

- Without changes to the spectrogram, the Hanning window can be applied either in the analysis (before DFT) or in the synthesis (after IDFT)
- However, if spectrogram is modified, it is good to perform the windowing on both ends. However, the whole thing needs to be still Hanning window, so:
  - Apply sqrt of Hanning before DFT
  - Apply sqrt of Hanning after IDFT
  - sqrt Hanning * sqrt Hanning = Hanning.

#spectrogram_filtering – deleting part of spectrogram + up to you to try anything else

Playing with spectrogram II – duration modification

- Taking every 2nd frame from the spectrum – shortening 2x – “Ostrava accent”
- Repeating every frame – lengthening 2x – “Prague accent”

#spectrogram_duration_modif

The quality of audio is not perfect – the phases would need better processing (continuity of phase across frames).

Agenda

- DFT refresher
- DFT of typical signals
- Properties of DFT
- Some more math on DFT
- Spectrogram and what can we do with it
- **Summary and “take-home” messages**

SUMMARY

- DFT (computed by FFT) is the basic tool for frequency analysis.
- Basic truths on signals and spectra:
  - Narrow signal <=> wide spectrum
  - Wide (smooth) signal <=> narrow spectrum
  - Periodic signal <=> discrete spectrum
  - Discrete signal <=> periodic spectrum
- Shifts of signals
  - Delay => magnitude spectrum stays the same, phase goes downhill.
  - Advance => magnitude spectrum stays the same, phase goes uphill.
  - For DFT, we need to do circular shifts to stay in *0 … N-1*

SUMMARY II.

- Spectrogram is evolution of spectrum in time
  - Contains magnitude and phase, only magnitude is shown
  - Lots of tuning can be done by non-linearity (log), colormaps, and setting brightness and contrast of the image.
  - Better frequency or better time resolution, but not both – long vs. short-term spectrogram
- Synthesis of signal from spectrogram
  - Analysis should be done carefully – shift is ½ of the frame-length, Hanning window.
  - Spectrum needs to be completed by the 2nd half before IDFT.
  - In case modifications are done, it's safer to use sequence of sqrt(Hanning) and sqrt(Hanning)
- Filtering is usually done by other means, but spectrogram modifications are of great use in machine learning (learned T-F masks, etc).