# 4. PŘEDNÁŠKA – Spektrální manipulace

- **Spektrální analýza**
    - Fourierovy řady
    - Diskrétní Fourierovy řady
    - Fourierova transformace
    - Diskrétní Fourierova transformace
- **Fázový vokodér**
    - Analýza
    - Transformace
    - Syntéza
- **Spektrální manipulace**
    - Časové a frekvenční modifikace
    - Audio efekty (robotizace, šepot)
    - Potlačení šumu

FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

# Spektrální analýza

- **komplexní (exponenciální) tvar Fourierovy řady**

$$\boxed{x(t) = \sum_{k=-\infty}^{\infty} \mathbf{C}_k \mathrm{e}^{\mathrm{j}k\omega_0 t}}$$

$\mathbf{C}_k$ … komplexní koeficient

$$\mathbf{C}_k = \frac{1}{T}\int_0^T x(t)\mathrm{e}^{-jk\omega_0 \mathrm{t}}\,\mathrm{d}t$$

$$\mathbf{C}_k = \frac{1}{2}(a_k - \mathrm{j}b_k)$$

$$A_{mk} = 2|\mathbf{C}_k|$$

# Spektrální analýza

- **Diskrétní Fourierova transformace**
  - K výpočtu spektra **periodických číslicových** signálů lze použít vztah:

$$X[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]\mathrm{e}^{-\mathrm{j}2\pi nk/N}$$

$$X_k = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)\mathrm{e}^{-\mathrm{j}2\pi k f_0 t}dt$$

complex coefficient

  - Pro signál popsaný N vzorky je podstatných **N/2 hodnot** spektra. Dalších N/2 hodnot jsou čísla komplexně sdružená a není třeba je počítat.
  - Výpočtem podle výše uvedeného vztahu dostaneme **diskrétní spektrum** s hodnotami komplexních koeficientů na frekvencích k·Fs/N.
  - Spektrum můžeme počítat i pro k > N, dostaneme však stejné hodnoty jako pro základní interval -N/2 < k < N/2. Spektrum číslicových signálů je **periodické** s periodou Fs.

# Spektrální analýza

- **Od periodických signálů k neperiodickým**

$$X_k = \frac{1}{T}\int_{-\frac{T}{2}}^{\frac{T}{2}} x(t)\mathrm{e}^{-\mathrm{j}2\pi k f_0 t}dt$$

  - **Spojité signály** (popsané analytickou funkcí)
  - Na neperiodický signál se nahlíží jako na signál, jehož T→∞
  - Místo FŘ se používá FT
  - Protože T→∞, Δf→0 … spektrum spojité

$$X(f) = \int_{-\infty}^{\infty} x(t)\mathrm{e}^{-\mathrm{j}2\pi f t}dt$$

# Spektrální analýza

- **Od FT k DFT**

$$X(f)=\int_{-\infty}^{\infty} x(t)\mathrm{e}^{-\mathrm{j}2\pi ft}dt$$

  - **Číslicové signály** (nejsou spojité a nejsou nekonečně dlouhé)
  - Numerický ekvivalent FT bude

$$X[k]=\sum_{n=0}^{N-1} x[n]\mathrm{e}^{-\mathrm{j}2\pi nk/N}$$

  - DFT je popsána stejným vztahem jako DFŘ
  - Spektrum číslicového signálu je **diskrétní a periodické**

# Spektrální analýza

- **Diskrétní Fourierova transformace (DFT)**
  - Definiční vztah:

$$X[k]=\sum_{n=0}^{N-1} x[n]\mathrm{e}^{-\mathrm{j}2\pi nk/N}$$

  - **Vstupem** je N hodnot čísl. signálu (podle předpokladu jde o 1 periodu).
  - **Výstupem** je N hodnot komplexních koeficientů spektra na normovaných frekvencích k/N, tj. na reálných frekvencích k·Fs/N.
  - Spektrum je periodické s periodou Fs, tj pro k > N dostaneme tytéž hodnoty.
  - Hodnoty koeficientů pro N/2 < k < N jsou komplexně sdružené s prvními N/2 hodnotami, netřeba je počítat. Při určování modulu jednostranného spektra je nutné násobit dvěma.
  - Pokud vybraných N vzorků signálu netvoří jednu periodu (v praxi je to téměř vždy), jsou výsledné hodnoty zatíženy chybami (objeví se neexistující složky).

# Spektrální analýza

- **Zpětná (inverzní) DFT**

DFT

$$X[k]=\sum_{n=0}^{N-1} x[n]\mathrm{e}^{-\mathrm{j}2\pi nk/N}$$

IDFT

$$x[n]=\frac{1}{N}\sum_{n=0}^{N-1} X[k]\mathrm{e}^{\mathrm{j}2\pi nk/N}$$

  - Vztah pro IDFT se liší od DFT pouze ve znaménku exponenciální funkce. Normovací koeficient 1/N se někdy uvádí u DFT.
  - Do IDFT vstupuje vždy N hodnot dvoustranného spektra, tj. nejenom N/2 hodnot jednostranného spektra.
  - Pokud na signál aplikujeme nejprve DFT a následně IDFT, dostaneme tentýž signál. Vyplývá to z toho, že popis signálu v časové i ve frekvenční oblasti je ekvivalentní co do úplnosti informace. (Ve spektrální oblasti však musíme vždy uvažovat jak modul, tak i fázi.)

# Spektrální analýza

- **Fast Fourier Transform (FFT)**
  - FFT – rychlý algoritmus výpočtu DFT.
  - Poskytuje stejné hodnoty jako DFT, ale mnohem rychleji
  - Vysoké rychlosti je dosaženo optimalizovaným způsobem symetričnost exponenciálních členů, podobnost mezi lichými a sudými koeficienty…
  - Nejrychleji funguje v případech, že N je mocninou 2 Např. pro N=1024 je FFT 200x rychlejší než DFT
  - V MATLABu
    - fft(x) …spočítá DFT pro signál x
    - ifft(x)…spočítá IDFT pro spektrum x

# Spektrální analýza

- **Předpoklad periodicity**

Reálné signály jsou nekonečné

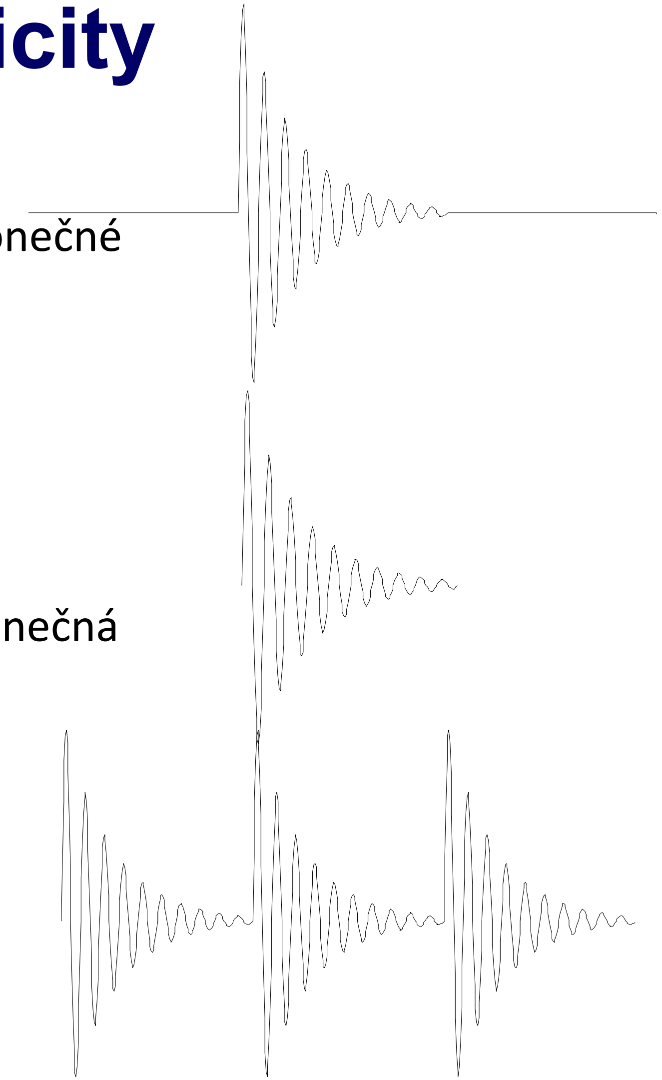
Signál

Vzorkovaná data jsou konečná

FFT vidí pouze toto

FFT je periodická

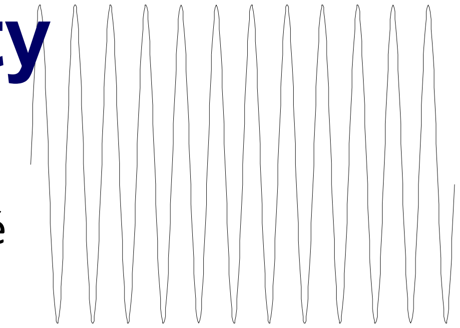
FFT funguje, jako by viděla toto



# Spektrální analýza

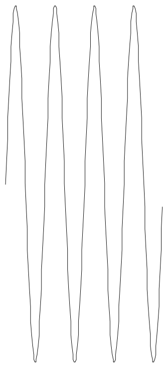
- **Předpoklad periodicity**

Reálné signály jsou nekonečné



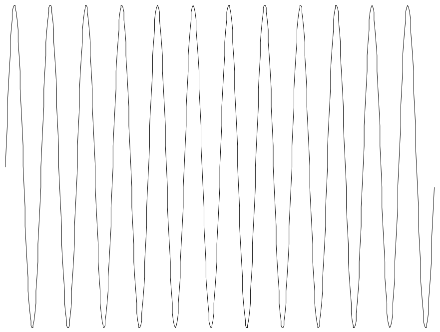
Toto je signál

Perioda odpovídá času vzorkování



FFT vidí pouze toto

Předpoklad FFT je správný



FFT funguje, jako by viděla toto

# Spektrální analýza

- **Předpoklad periodicity**

Reálné signály jsou nekonečné

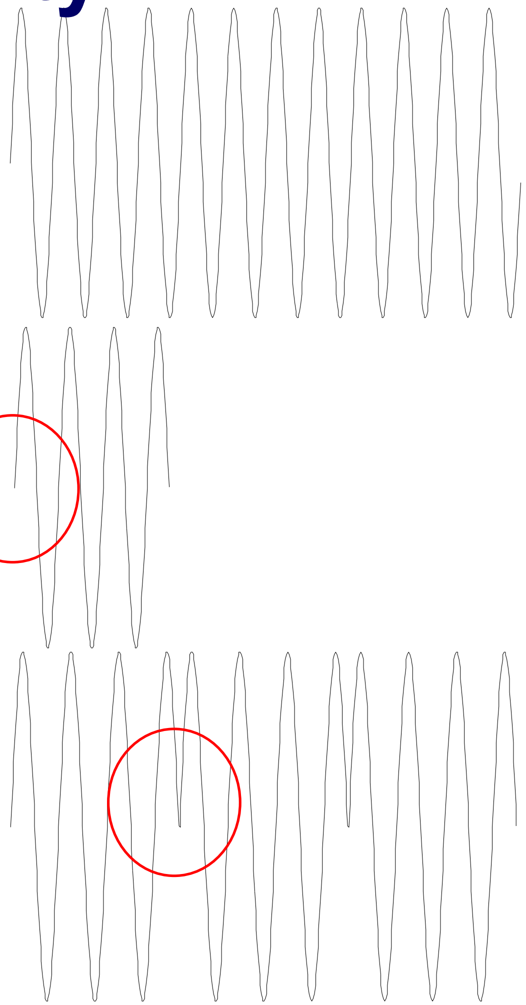
Toto je signál

Vzorkovaná data trochu chybí

FFT vidí pouze toto

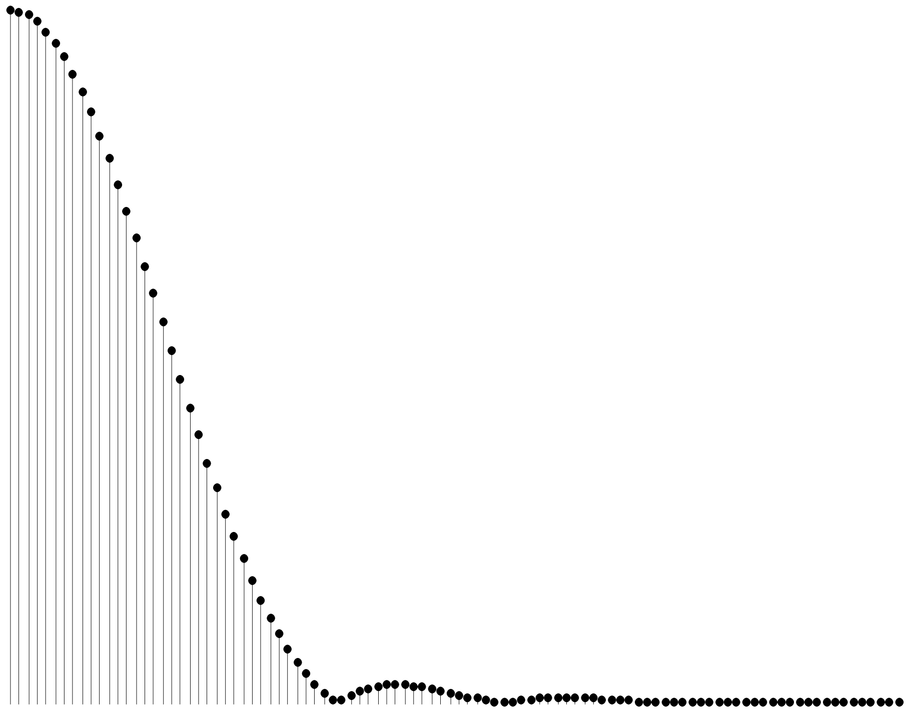
FFT předpokládá nespojitosti

které FFT vidí



# Spektrální analýza

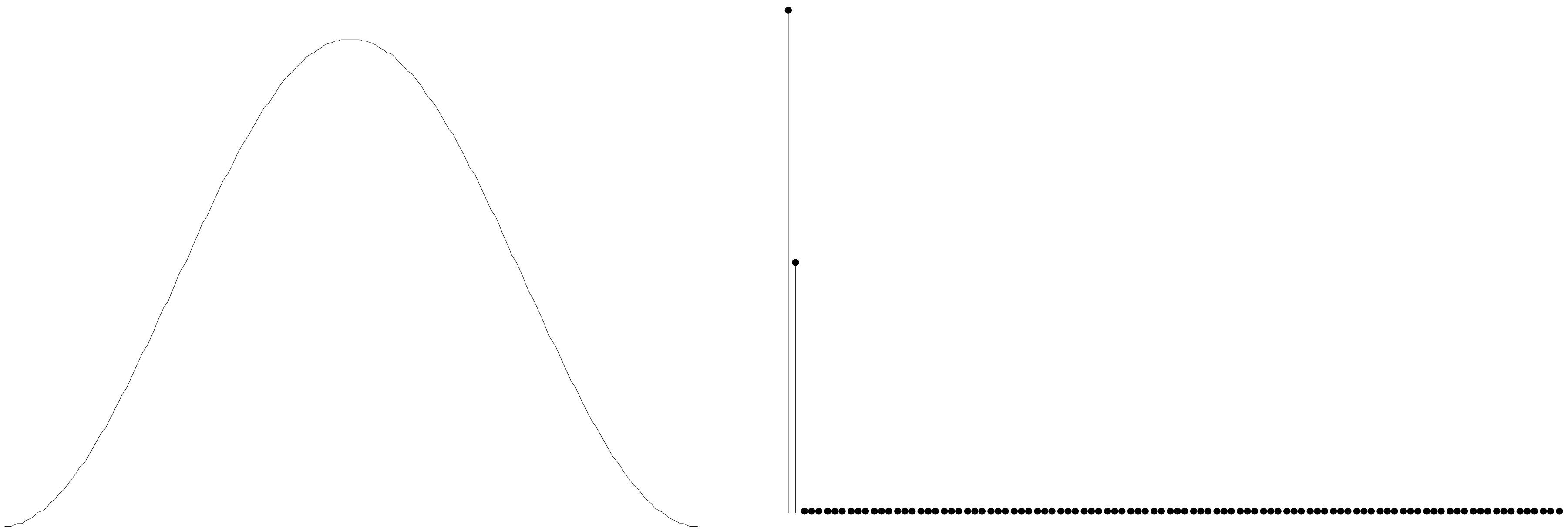
- **Krátké signály**



Signál krátkého trvání má široké spektrum

# Spektrální analýza

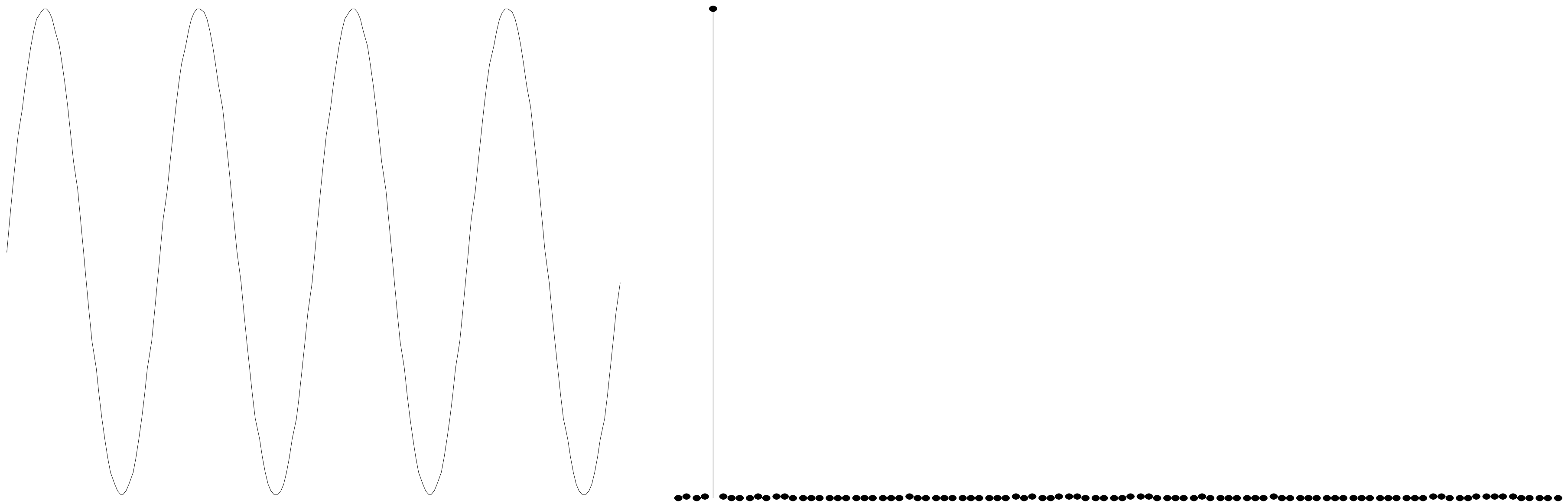
- **Krátké signály**



Signál s dlouhým trváním má úzké spektrum

# Spektrální analýza

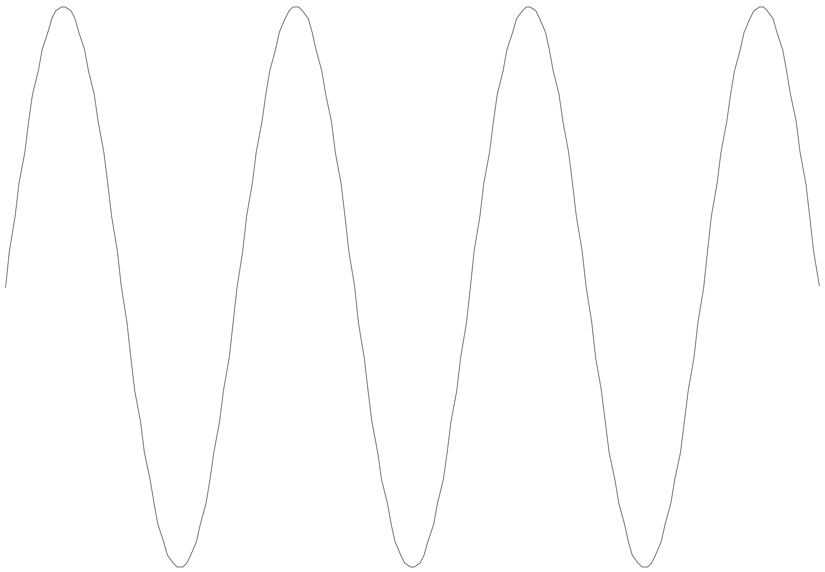
- **Předpoklad periodicity**

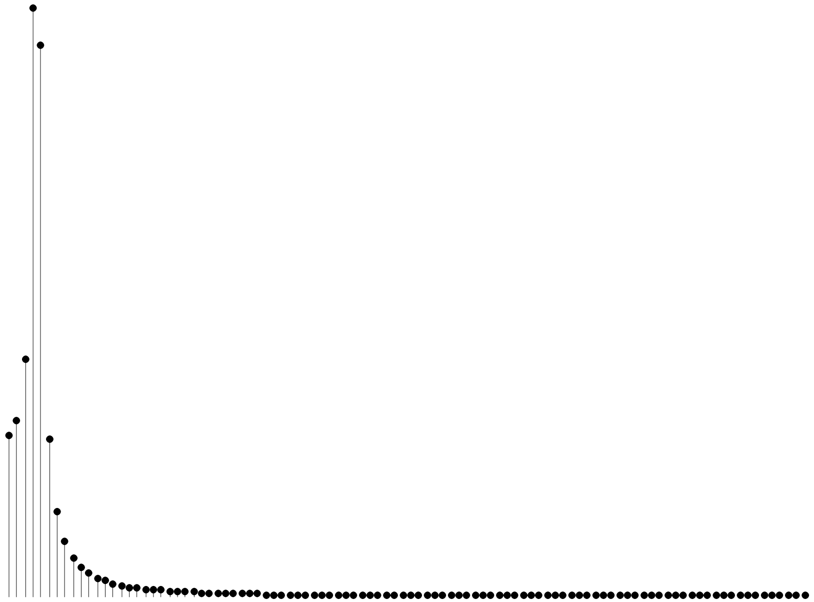


Pokud perioda odpovídá času, spektrum je správné.

# Spektrální analýza

- **Předpoklad periodicity**

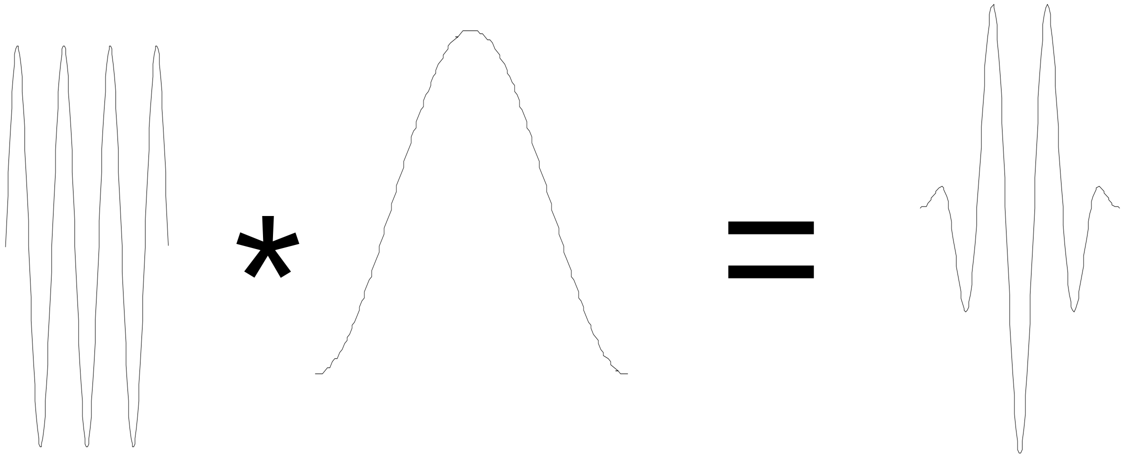


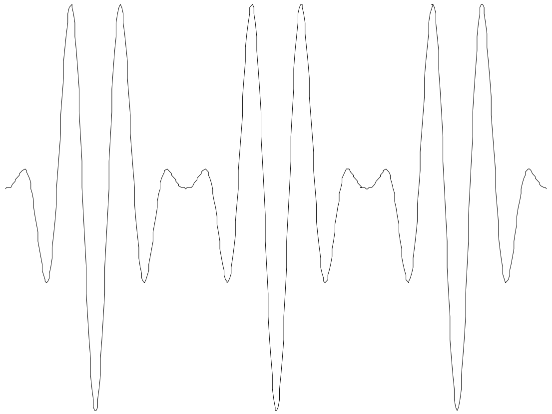


Pokud perioda neodpovídá času, vznikají falešné spektrální čáry.

# Spektrální analýza

- **Váhování**

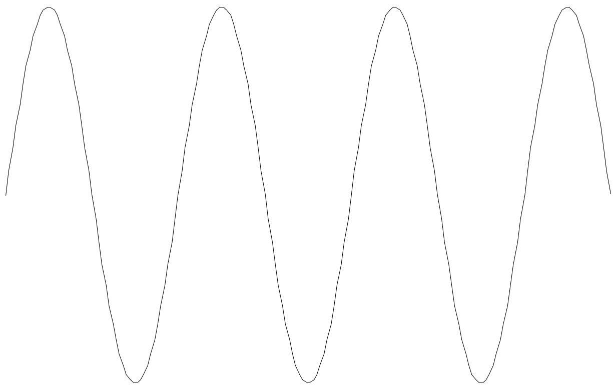


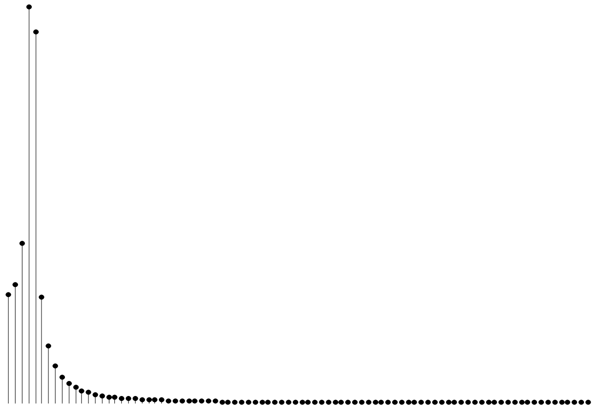


Váhování potlačuje nespojitosti a vzorkovaná data se blíží periodičnosti.

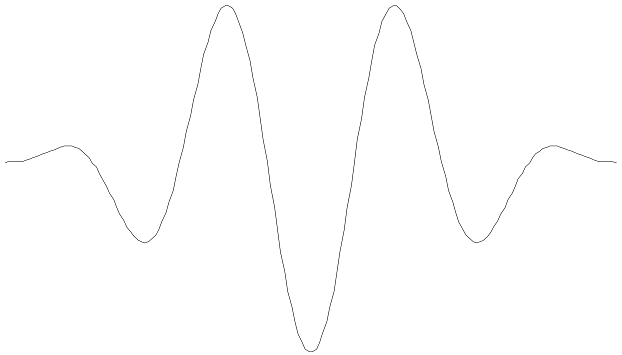
# Spektrální analýza

- **Váhování**





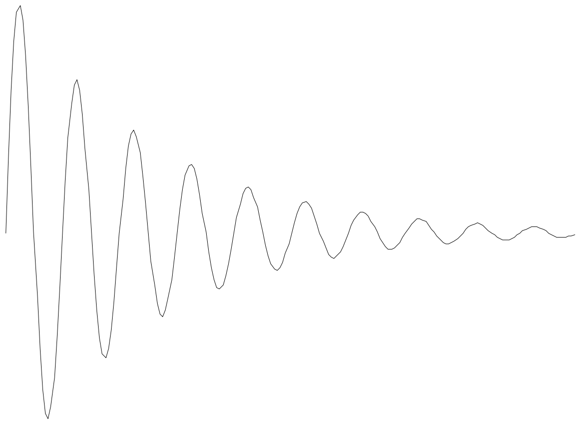
Pokud perioda neodpovídá času, vznikají falešné spektrální čáry.

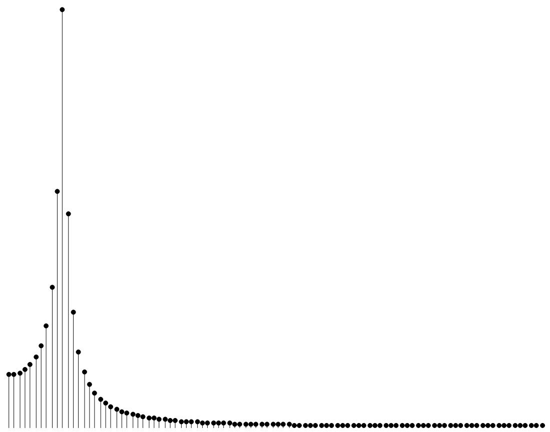


Ale váhování dat může zúžit spektrum.

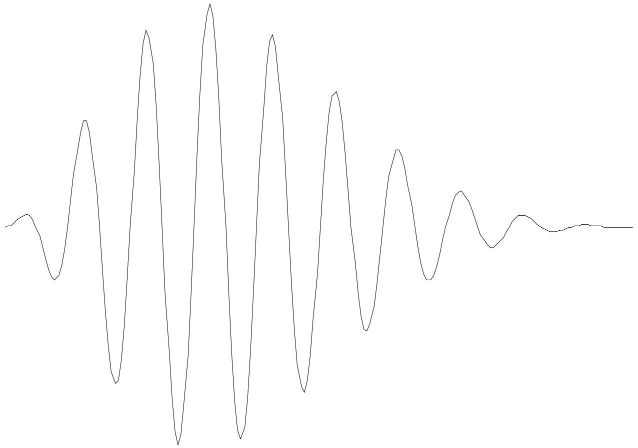
# Spektrální analýza

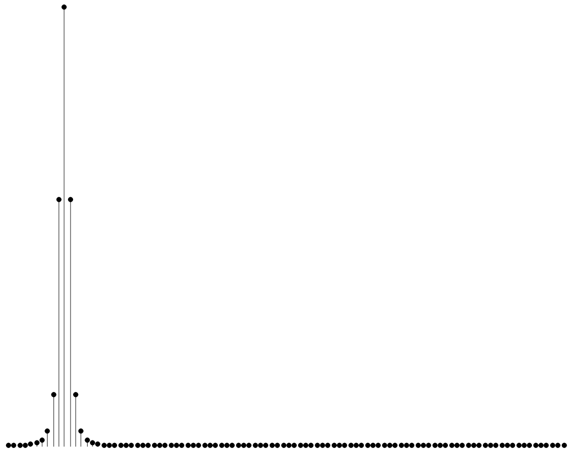
- **Váhování**





Tato přechodná odezva má skutečně široké spektrum





Ale váhování dat to vytváří dojmu, že to vypadá spíše jako jeden tón.

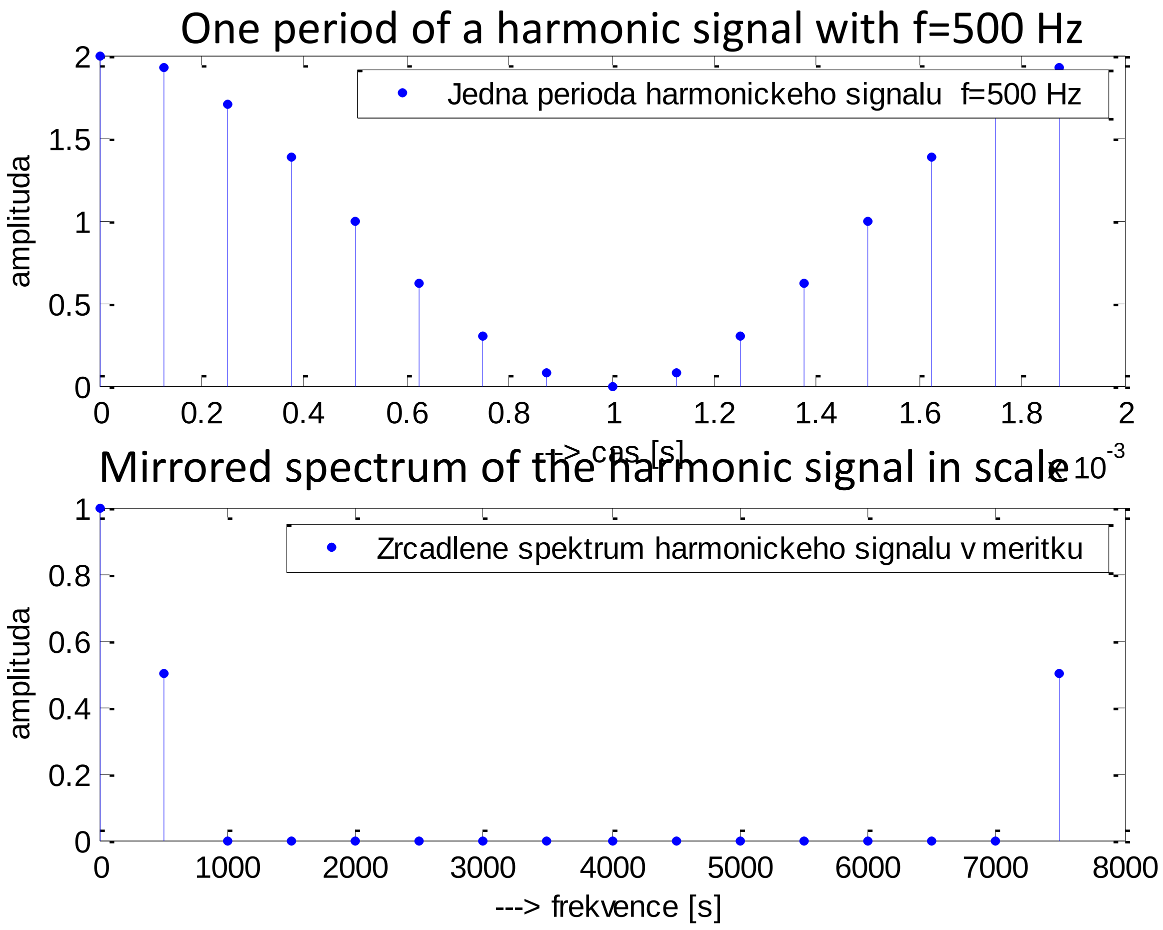
# Spektrální analýza

- **Praktická aplikace DFT a FFT**

    - Analyzovaný signál rozdělíme do kratších úseků.
    - Délku segmentu volíme tak, aby se počet vzorků N rovnal mocnině 2.
    - Pokud nelze zvolit N jako mocninu 2, doplníme signál nulami.
    - Vzorky vynásobíme vhodným oknem.
    - Pomocí FFT vypočítáme komplexní koeficienty spektra.
    - Do časové oblast se lze vrátit pomocí IFFT.

# Spektrální analýza

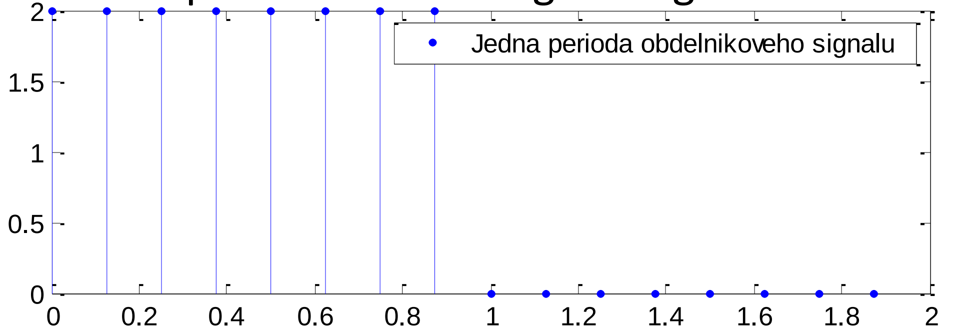
- **Zobrazení signálů ve frekvenční oblasti**



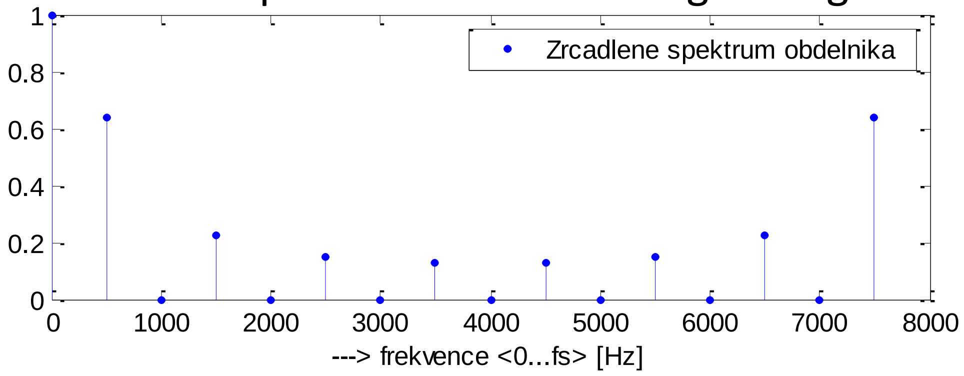
# Spektrální analýza

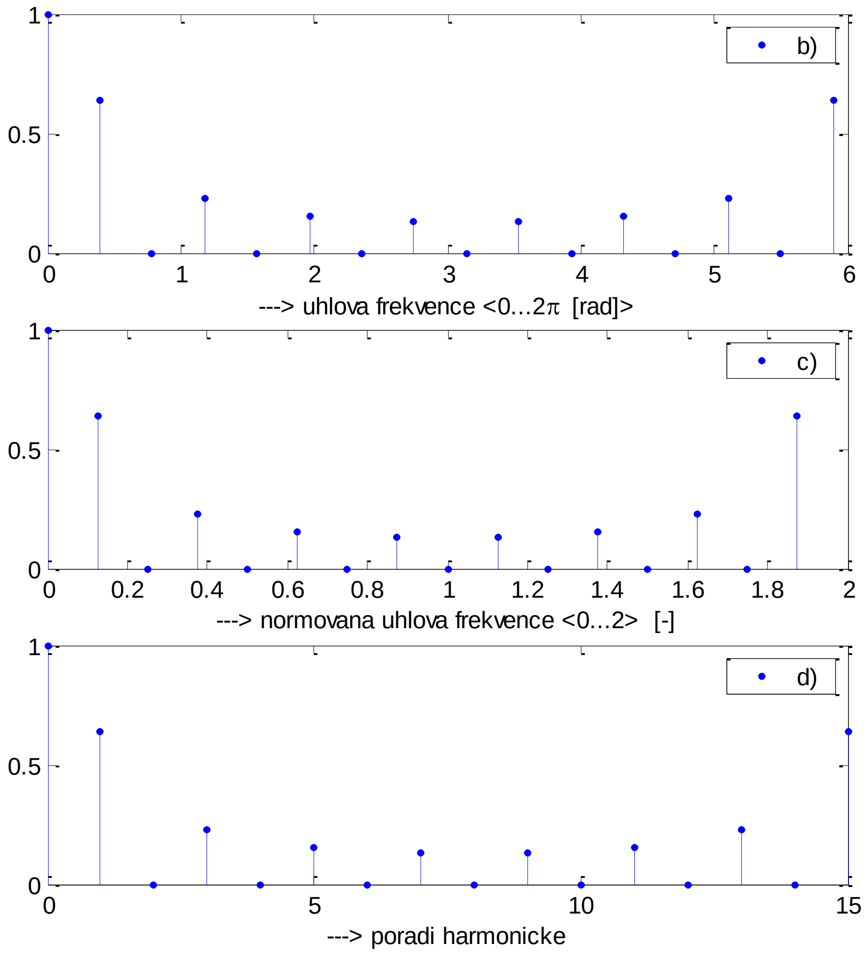
- **Frekvenční osa v periodogramu**

One period of a rectangular signal




Mirrored spectrum of the rectangular signal

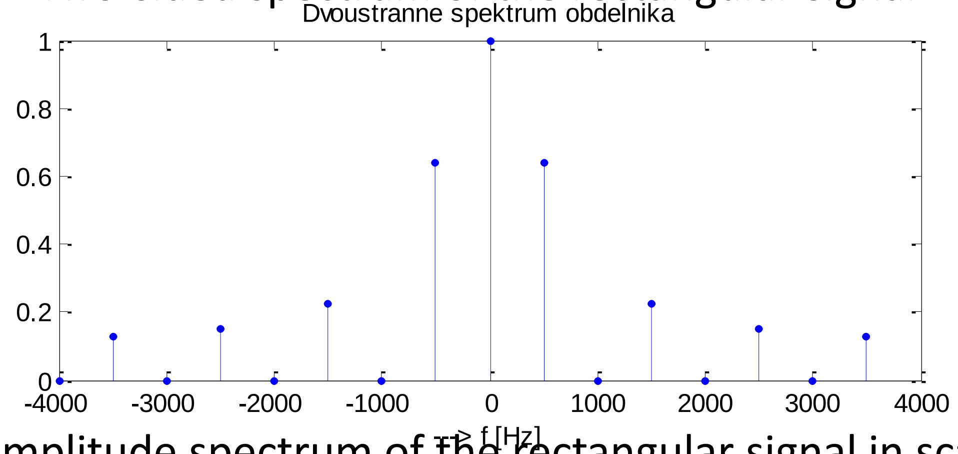


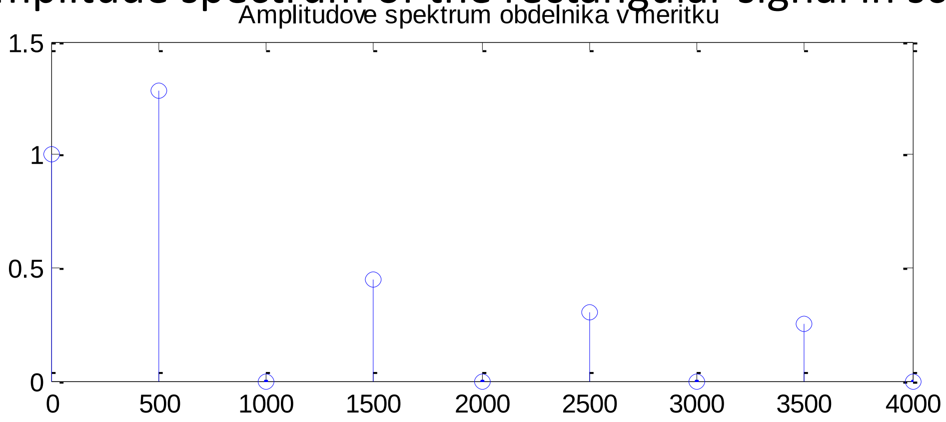
# Spektrální analýza

- **Jednostranné a dvoustranné spektrum**

Two-sided spectrum of the rectangular signal

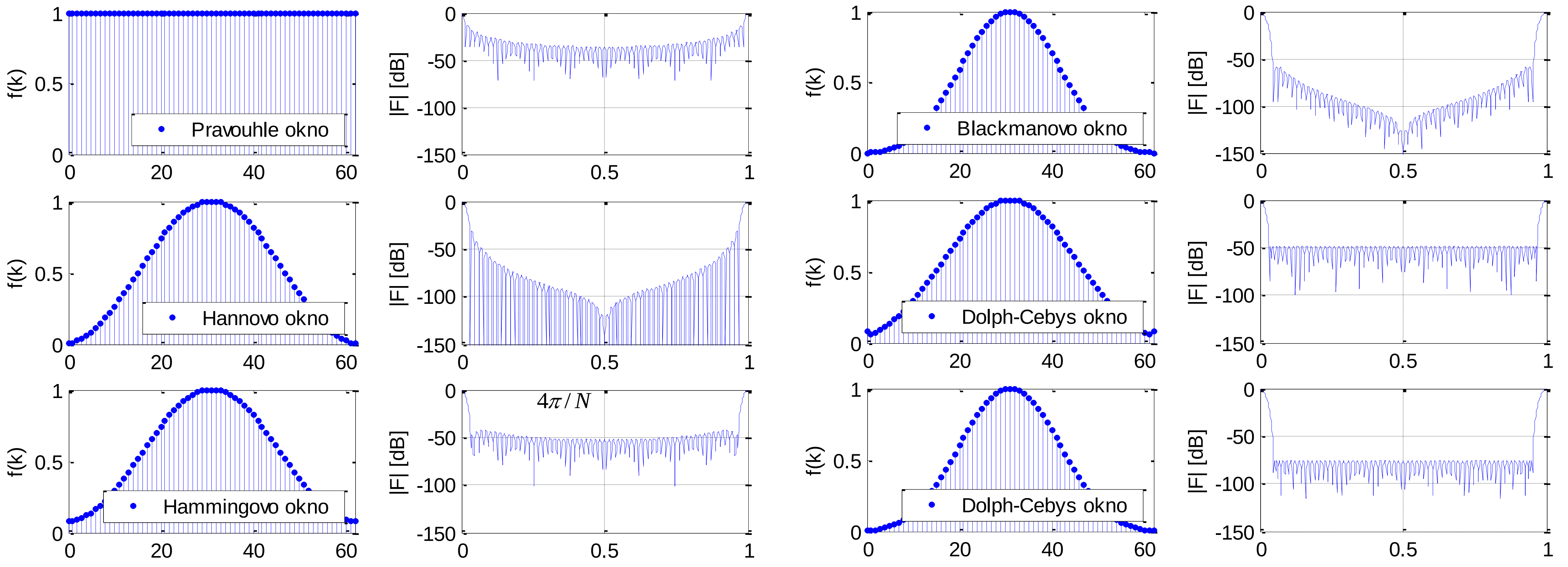



Amplitude spectrum of the rectangular signal in scale



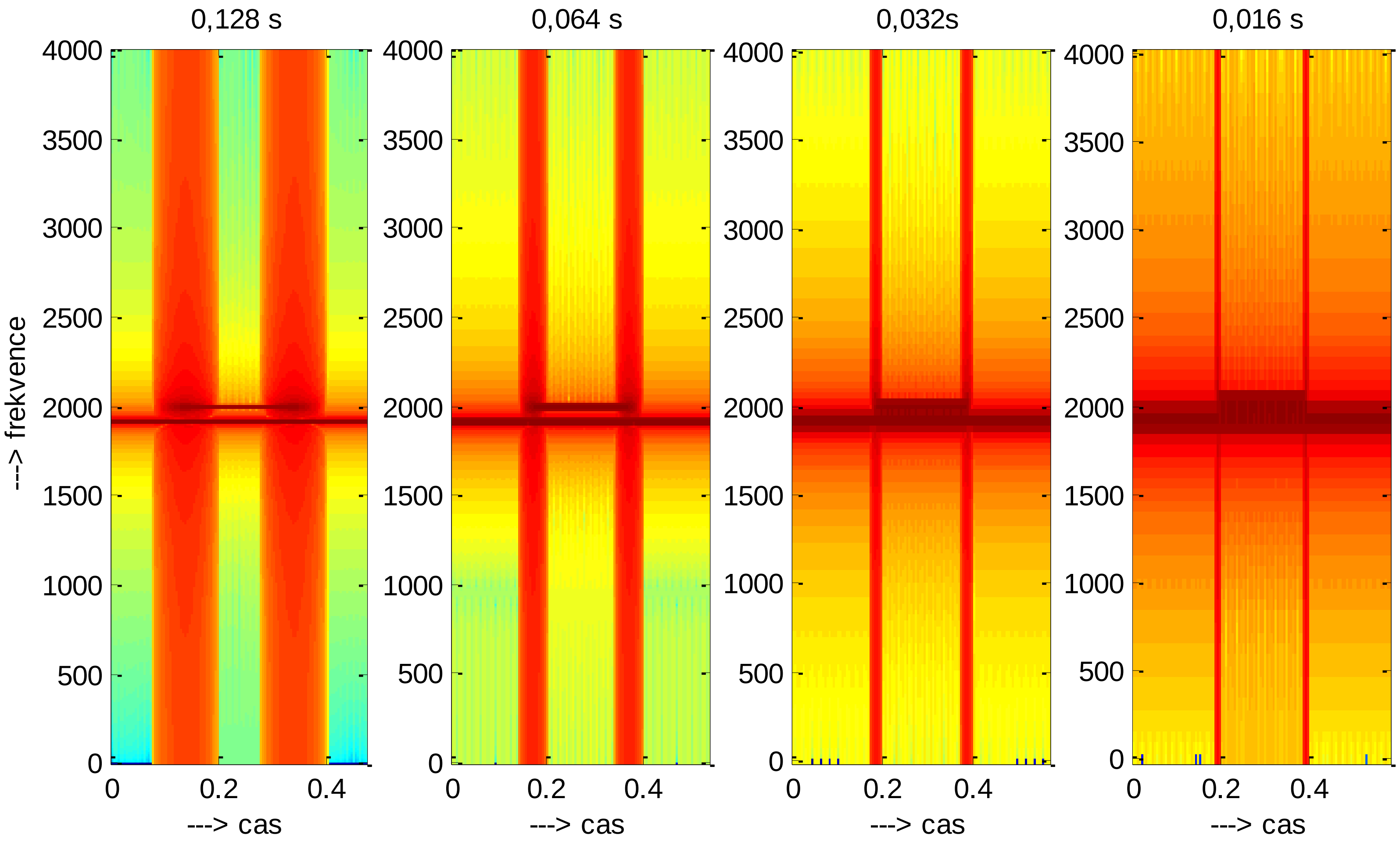
# Spektrální analýza

- **Porovnání oken**



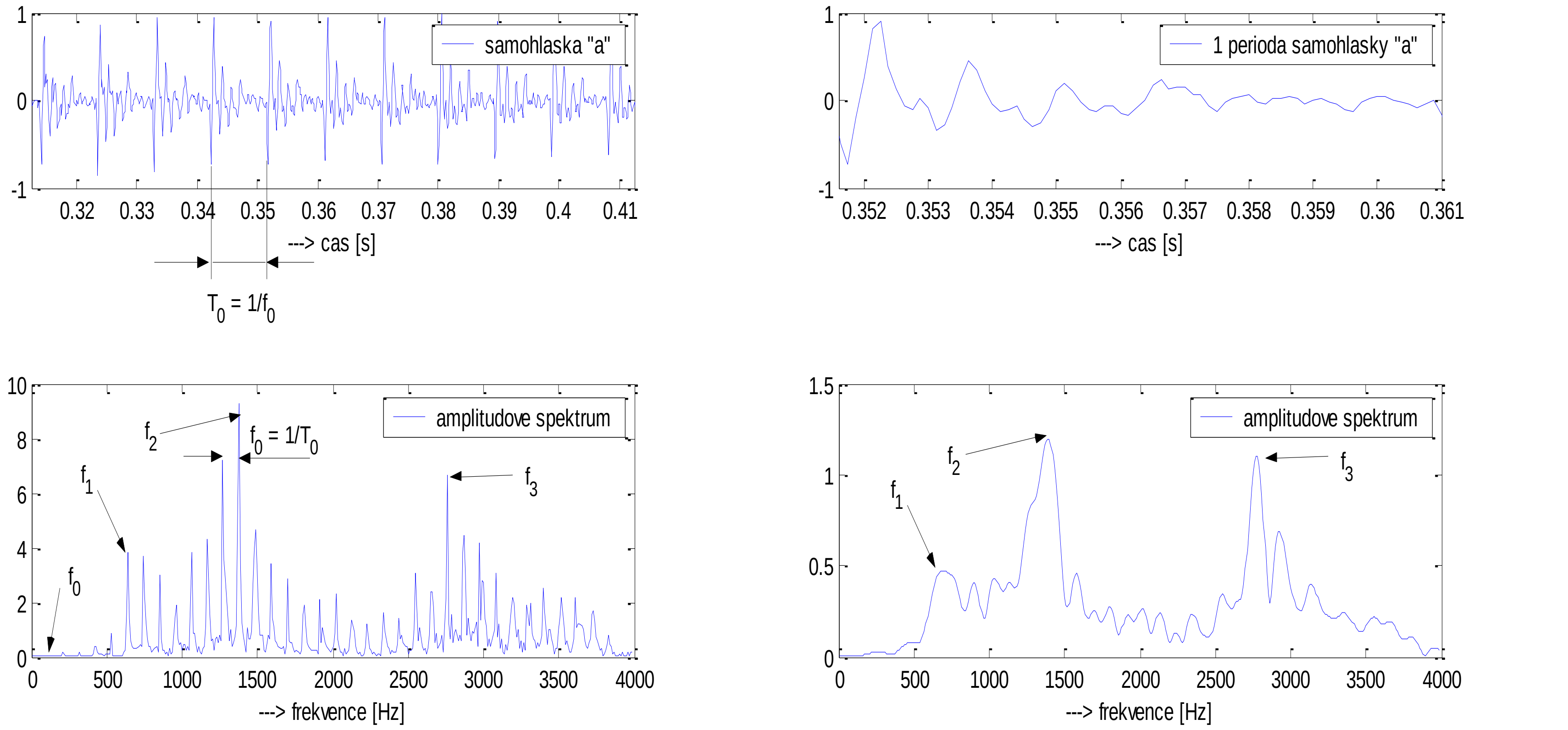
# Spektrální analýza

- **Spektrogramy – rozlišení**



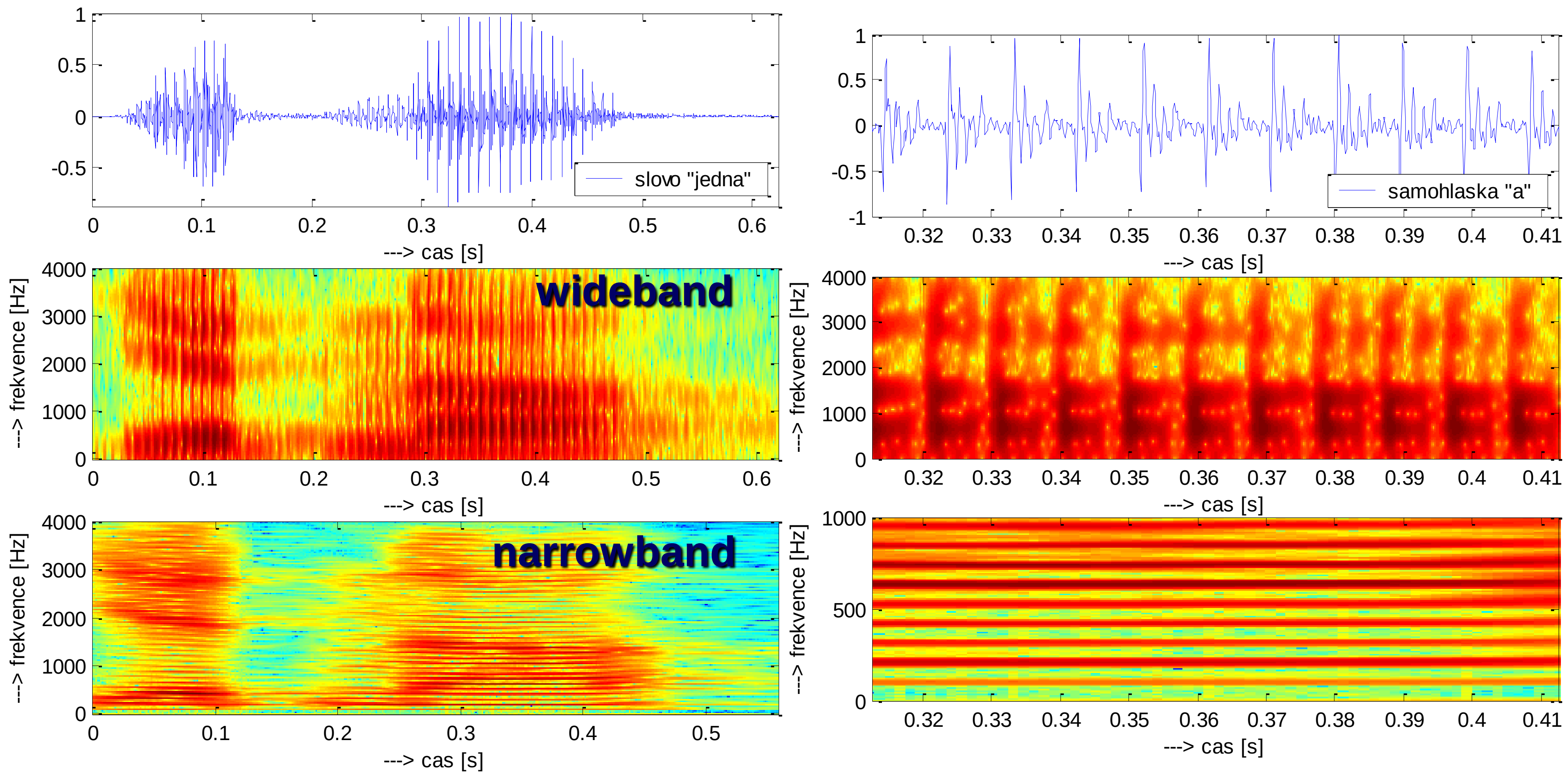
# Spektrální analýza

- **Spektrum samohlásek**



# Spektrální analýza

- **Spektrogramy**



# Spektrální analýza

- **Spektrogramy** spectrogram(X,WINDOW,NOVERLAP,NFFT,Fs)

*a) širokopásmový spektrogram – dobré rozlišení v čase*
*(základní frekvence je určena vertikálními pruhy)*

```
load osum.asc;
sig = osum; fs = 8000;
subplot(2,1,1); spectrogram(sig, 32, 30, 1024, fs, 'yaxis');
title('Wideband Spectrogram')
xlabel(''), ylabel('Frequency [Hz]')
```

*b) úzkopásmový spektrogram – dobré frekvenční rozlišení*
*(základní frekvence je určena horizontálními pruhy)*

```
subplot(2,1,2); spectrogram(sig, hamming(512), 500, 1024, fs, yaxis');
title('Narrowband Spectrogram')
xlabel('---> Time [s]'), ylabel('Frequency [Hz]')
```

# Fázový vokodér

- **Oblíbený a výkonný audio nástroj**
  - **Zvuková studia**
    - Úprava délky záznamu při zachováních frekvenčních vlastností
    - Frekveční posuny (půltóny, oktávy)
  - **Řečové technologie**
    - TTS synzéza
    - Rozpoznávače
- **Fázový vokodér** jsou techniky, které:
  - segmentují signál,
  - transformují jej do frekvenční oblasti,
  - provedou modifikace ve spektru,
  - signál rekonstruují v časové oblasti.

https://musicandcomputersbook.com/chapter5/05_04.php

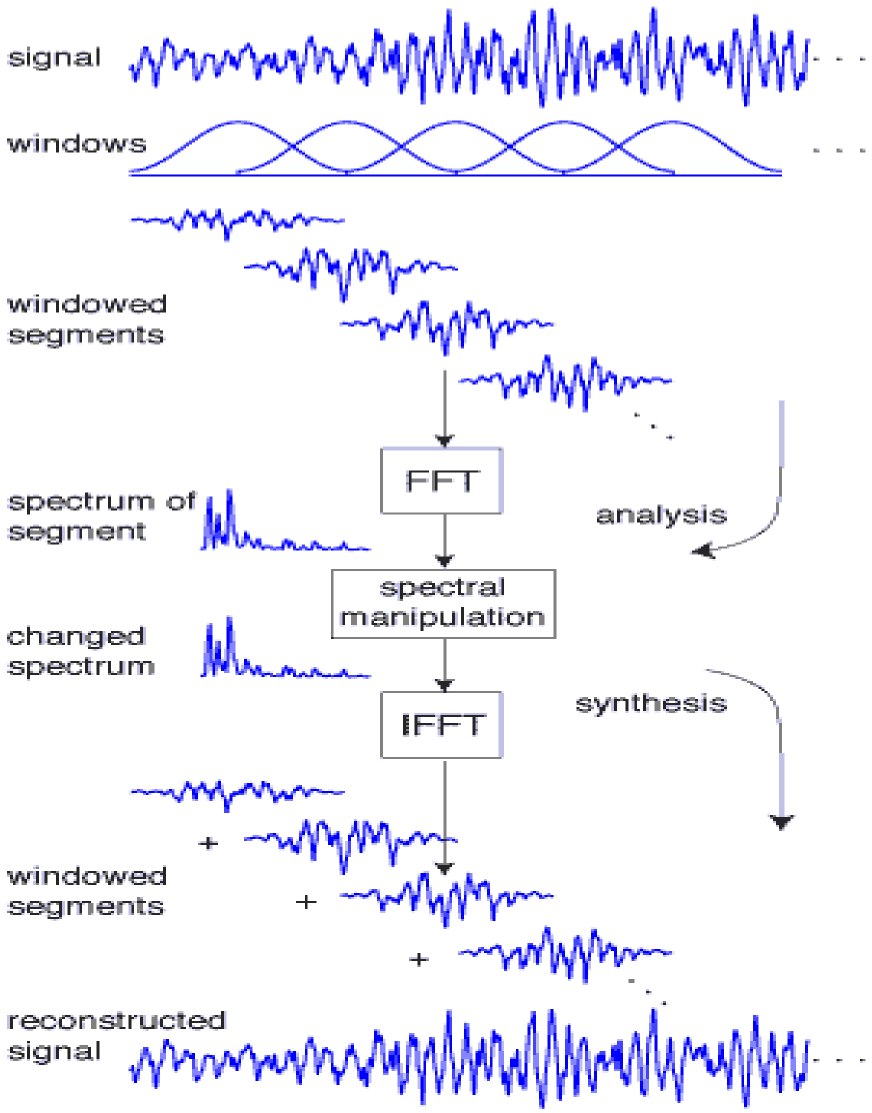
# Fázový vokodér

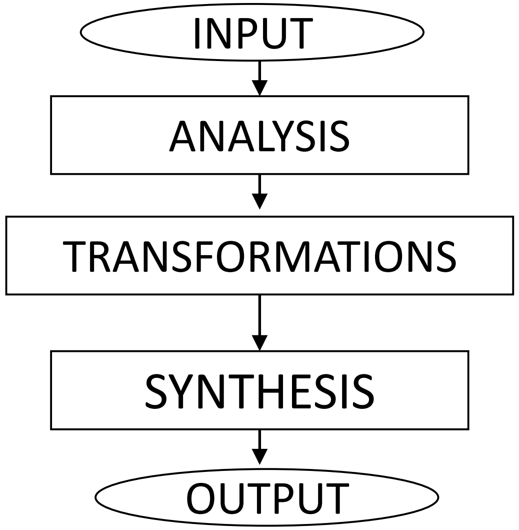
Krátkodobá Fourierova transformace

$$X_m = \frac{1}{n}\sum_{k=0}^{n-1} x_k e^{-j\frac{2\pi mk}{n}}$$

Re-syntéza

$$x_k = \frac{1}{n}\sum_{m=0}^{n-1} X_m e^{j\frac{2\pi mk}{n}}$$



# Fázový vokodér – analýza

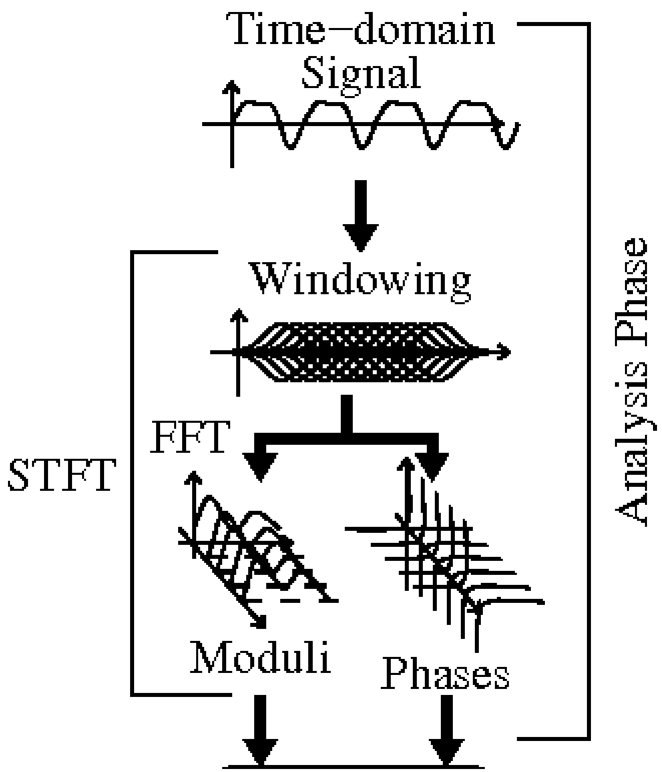
- **Délka segmentu (M):** počet analyzovaných vzorků signálu.
  - delší segment = lepší frekvenční a nižší časové rozlišení.
  - kratší segment = lepší časové a menší frekvenční rozlišení.
  - ovlivňuje dobu výpočtu (doporučená délka je 30 .. 40 ms)

- **Typ okna:** jakékoliv nepravoúhlé (např., Hamming, Hanning, Blackman).

- **Počet vzorků FFT:** optimalizováno na 2^n (např., 1024).
  - Obvykle je velikost FFT nejbližší mocninou dvou, která je dvojnásobkem délky segmentu.
  - Např., pro délku segmentu 512, bude počet vzorků FFT roven 1024.

- **Překrytí (overlap** nebo **hop size):** zlomek M, např., M/8.

# Fázový vokodér – analýza



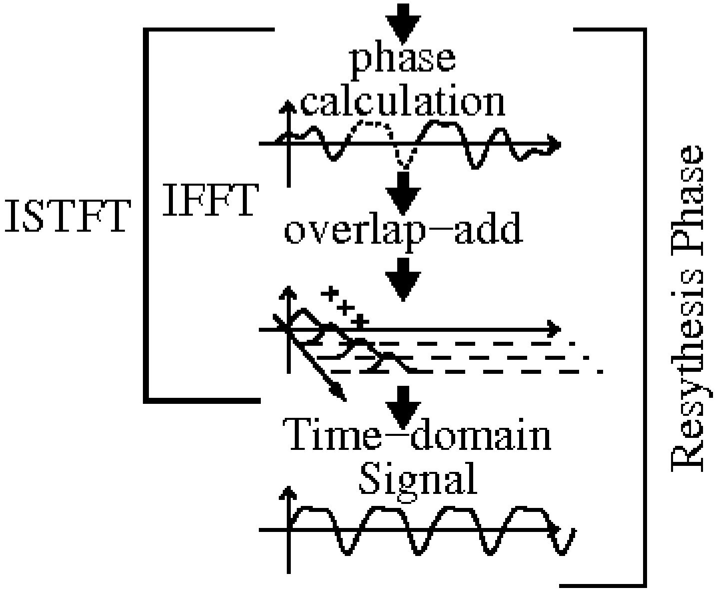

```
% Spectral analysis
% (Short-Time Fourier Transform)

X = []; k=1;
for start = 0:window_shift:N-window_length,
    frame = sig(start+1:start+window_length)...
     .*hamming(window_length);
    X(:,k)=fft(frame);
    k=k+1;
end
```

# Fázový vokodér – syntéza

- **Inverzní Fourierova transformace:**
  - Modifikované komplexní spektrum se transformuje zpět do časové oblasti.

- **Metoda Overlap-Add (překrývání a přidávání):**
  - Jednotlivé nově transformované segmenty signálu se v časové oblasti sčítají s překrytím, čímž se rekonstruuje spojitý audio signál.
  - Překrývání segmentů zajišťuje plynulé přechody a potlačuje artefakty.

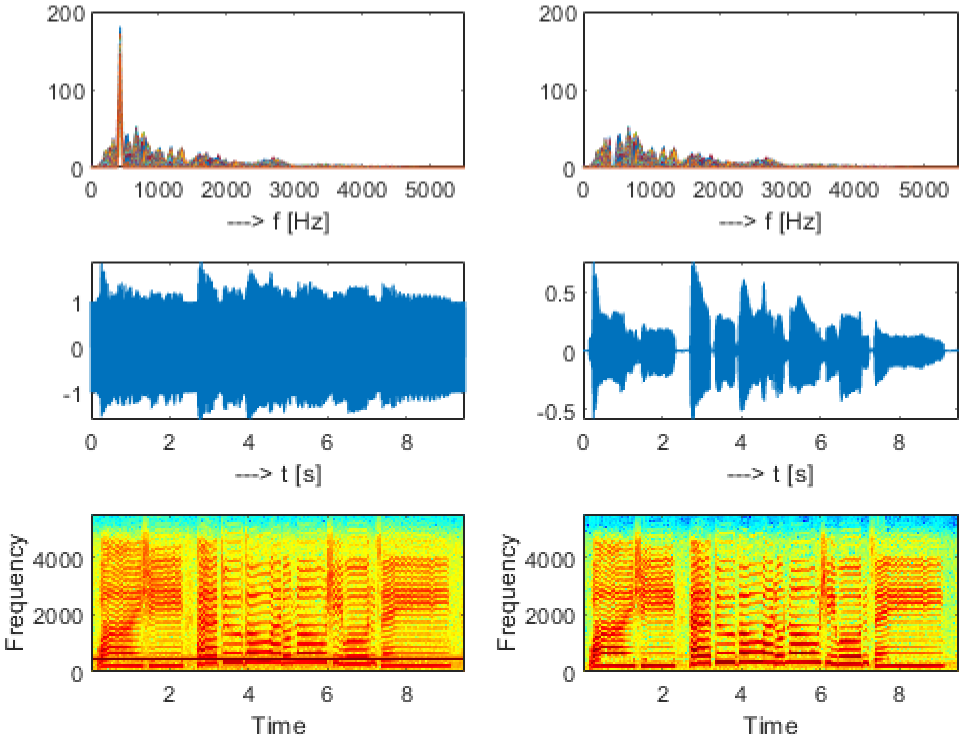
# Fázový vokodér – syntéza



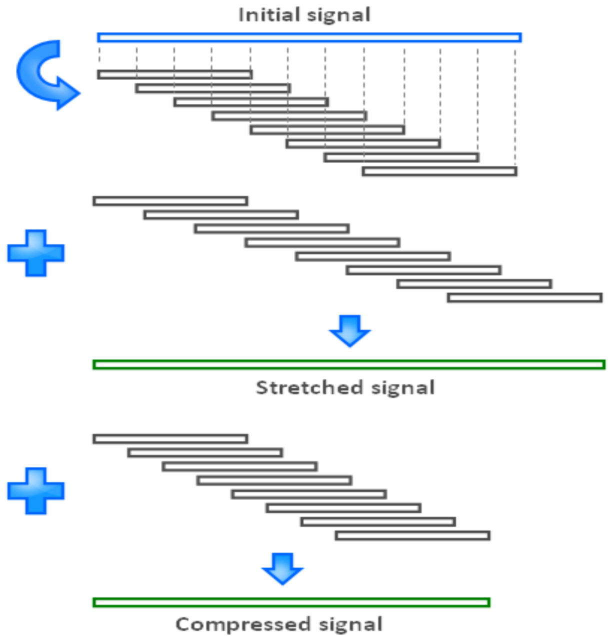

```
% Synthesis (Inverse FFT and overlap-add method)
k=1; N_new=size(Y,2)*window_shift+window_length;
signal_y=zeros(N_new,1);
for start = 0:window_shift:(size(Y,2)-1)*window_shift,
    segment=real(ifft(Y(:,k),window_length))…
        .*hamming(window_length);
    signal_y((start+1):(start+window_length))...
      =signal_y((start+1):(start+window_length))+segment;
    k=k+1;
end
```

# Fázový vokodér – manipulace ve spektru

- **Transformace: filtrace ve spektru**
  - Potlačení 430 Hz



# Fázový vokodér – manipulace ve spektru

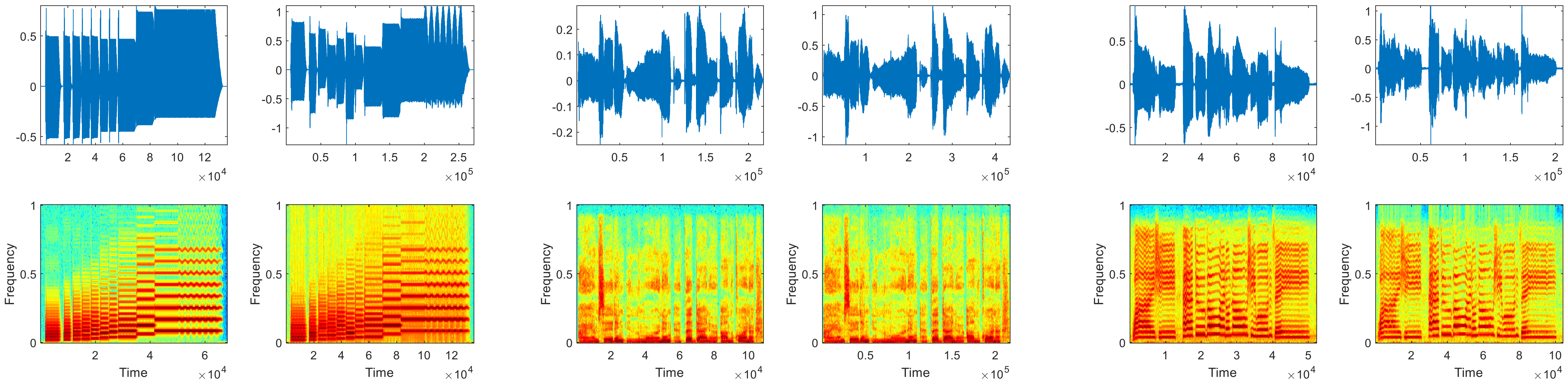



- **Transformace: změna délky signálu v čase**
  - R = N2 / N1 (činitel prodloužení)
  - R > 1 → prodloužení (délka signálu se zvětší)
  - R < 1 → zkrácení (délka signálu se zmenší)

  - **Interpolace amplitudy** Xmag
  - **Modifikace fáze**
    - Nová časová mřížka `new_grid = floor(0:1/R:size(X, 2)-2) + 1;`
    - Výpočet rozdílů fází `D = diff(angle(X'))';`
    - Mapování rozdílů fází do nové mřížky `D_new = D(:, new_grid);`
    - Rekonstrukce fáze `phaseX = cumsum(D_new');`
  - **Rekonstrukce spektra**
    - Kombinace amplitudy a fáze `Y = (Xmag .* exp(j * phaseX))';`

# Fázový vokodér – manipulace ve spektru

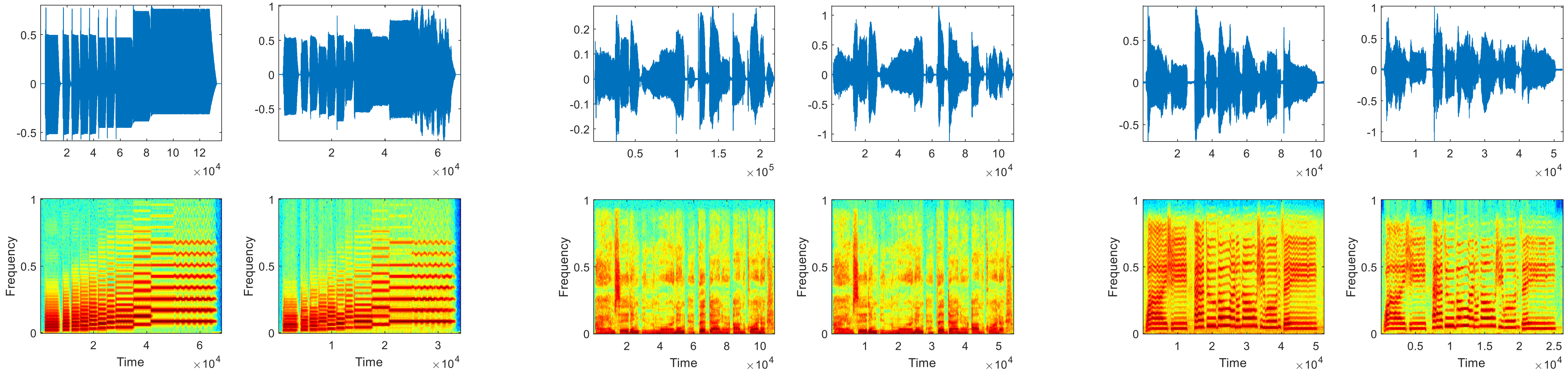
- **Prodloužení v čase: R = 2**



FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

# Fázový vokodér – manipulace ve spektru

- **Zkrácení v čase (komprese): R = 0.5**



# Fázový vokodér – manipulace ve spektru

- **Transformace: změna výšky tónu (frekvenční transpozice)**
  - **Postup:**
    1. **Prodloužení signálu v čase**
       - Činitel prodloužení: R = N2 / N1
         - R > 1 pro prodloužení
         - R < 1 pro zkrácení (komprese)
    2. **Převzorkování prodlouženého signálu**
       - Převzorkováním s poměrem N1 / N2 se vrátíme k původní délce signálu, avšak s posunutou výškou tónu
  - **Příklad 1:**
    - Posun o oktávu nahoru: R = 2 / 1, pak převzorkování 1 / 2 (interpolace / decimace).
    - Posun o oktávu dolů: R = 1 / 2, pak převzorkování 2 / 1.
  - **Příklad 2:**
    - Posun o půltóny ST nahoru: ST = 12; % Počet půltónů (12 = posun o oktávu)

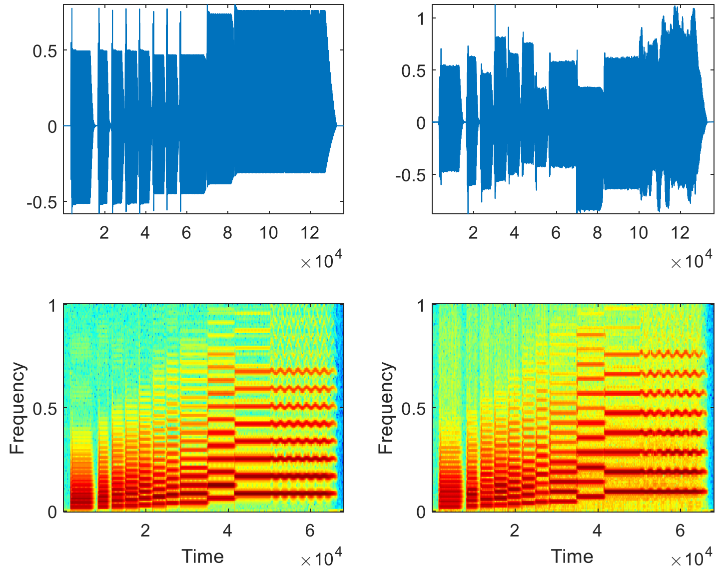
```
N1 = round(window_shift);  N2 = round(2^(ST/12) * N1);
R = N2/N1; resample(y, N1, N2)
```

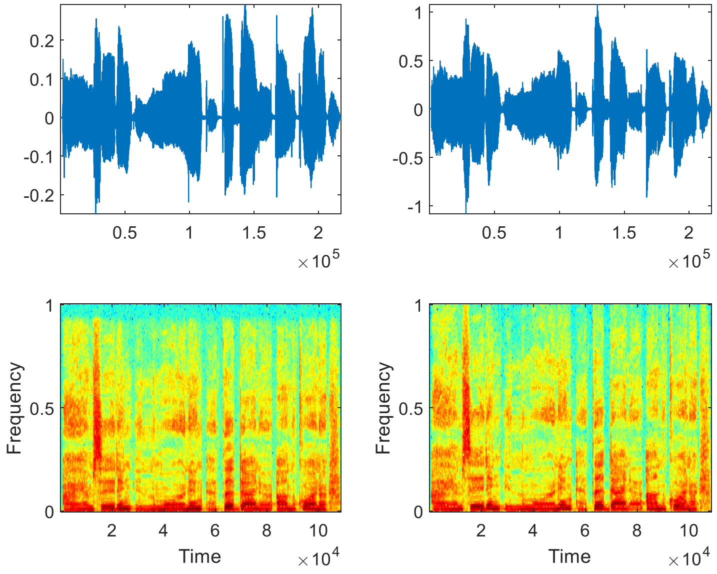
    - Posun o půltóny dolů: ST = -12; % Počet půltónů

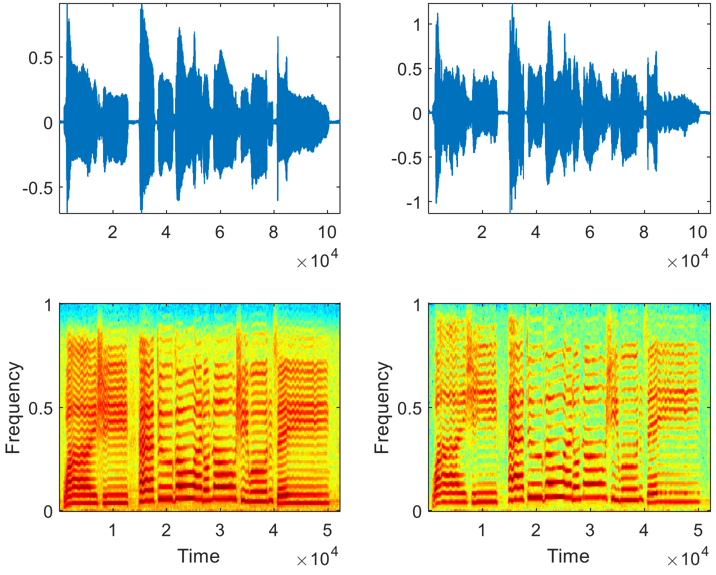
$$2^{(s/12)} = \frac{hop_s}{hop_a}$$

# Fázový vokodér – manipulace ve spektru

- **Zvýšení frekvence - f↑**

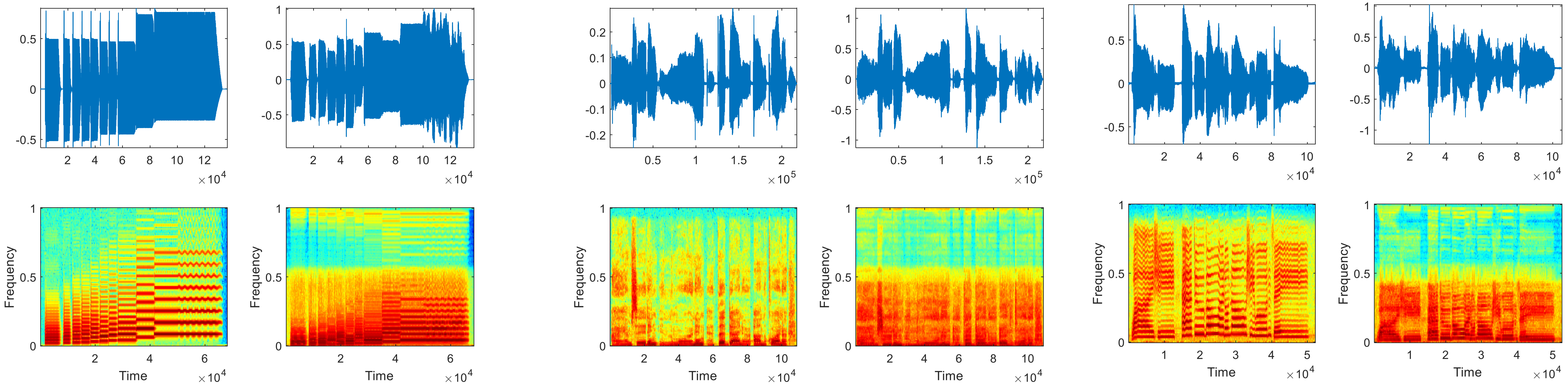


# Fázový vokodér – manipulace ve spektru

- **Snížení frekvence - f↓**

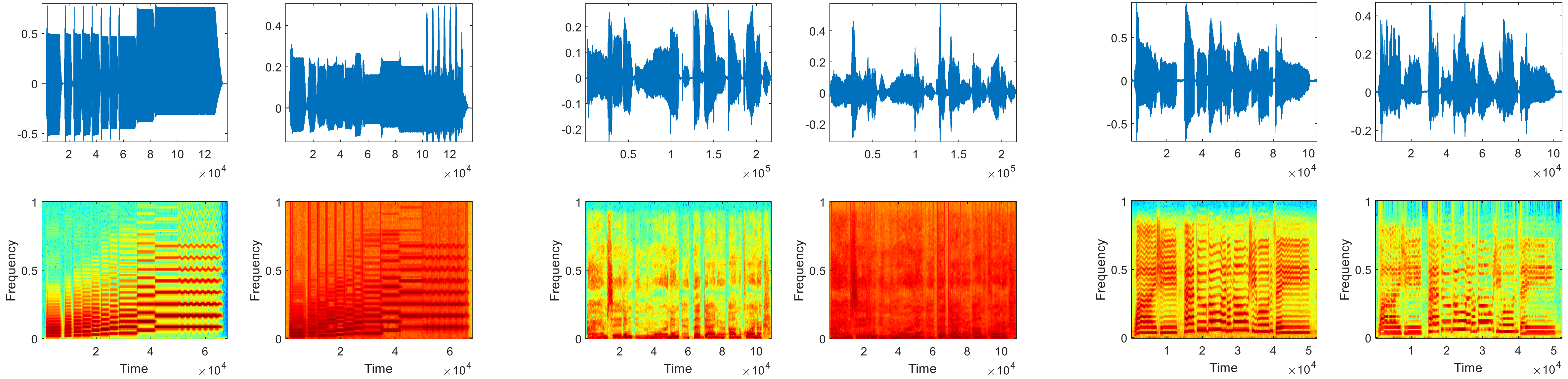


# Fázový vokodér – spektrální manipulace

- **Transformační fáze: efekty – robotický hlas**
  - o Zploštění fáze

```
phaseX = 0.*cumsum(D_new');
```

    - Zploštění fáze zahrnuje nastavení fáze na nulu, čímž se vytvoří monotónní nebo robotický efekt.

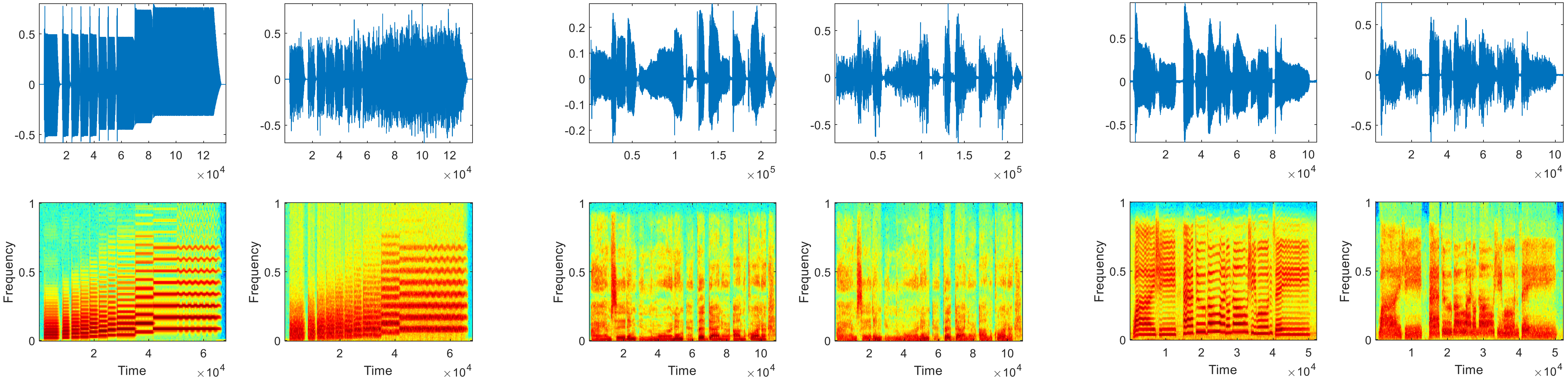


# Fázový vokodér – spektrální manipulace

- **Transformační fáze: efekty – šepot**
  - Náhodná změna fáze

    ```
    phaseX  = 2*pi*rand(size(D_new,2),size(D_new,1));
    ```

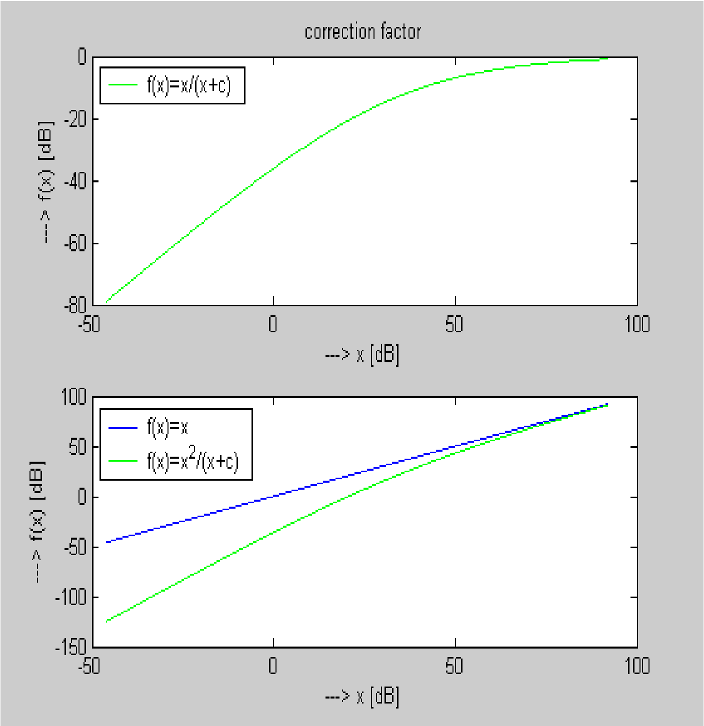
    - Náhodná změna fáze aplikuje náhodný fázový posun napříč spektrem, čímž vzniká šeptavý efekt.



# Fázový vokodér – spektrální manipulace

- **Transformační fáze:**
  **potlačení šumu pomocí nelineárního spektrálního odečítání**
  - Hlasité části signálu jsou zachovány, zatímco šum (obvykle tišší) je redukován zeslabením jeho magnitudy.

```
noise=0.0169*(randn(length(signal),1));
snr=10*log10(sum(signal.^2)/sum(noise.^2))
```




```
coef=0.02;
r=abs(X')./window_length;
Y = (X'.*r./(r+coef))';
```