FACULTY OF ELECTRICAL ENGINEERING CTU IN PRAGUE

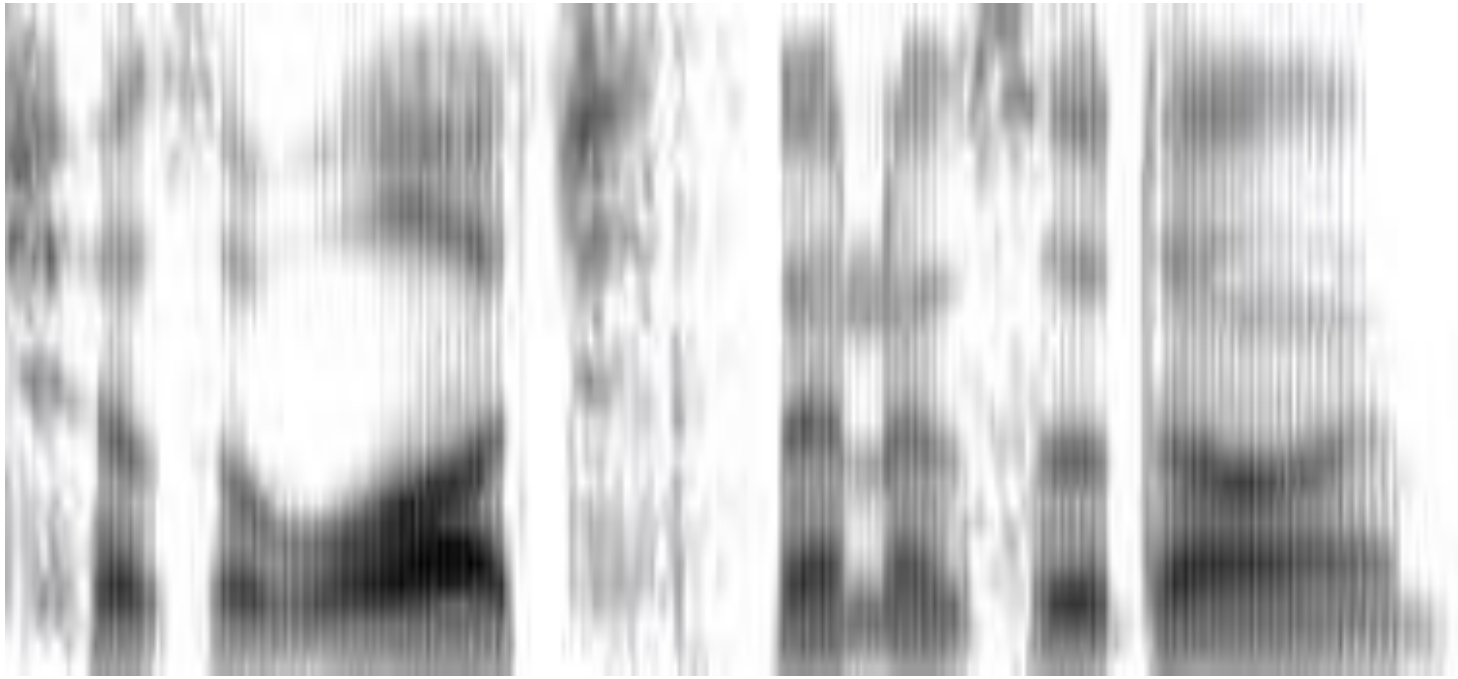
7. PŘEDNÁŠKA – LPC Syntéza

Lineární predikční kódování v řeči a hudbě

- **Model „zdroj a filtrace“**
- **Lineární Predikční kódování (LPC):** Analýza a syntéza
- **LPC Modelování hudebních nástrojů**
- **LPC Vokodér:** buzení a transformace
- **LPC vzájemná syntéza:** audio efekty
- **Příklady analýzy a syntézy LPC**

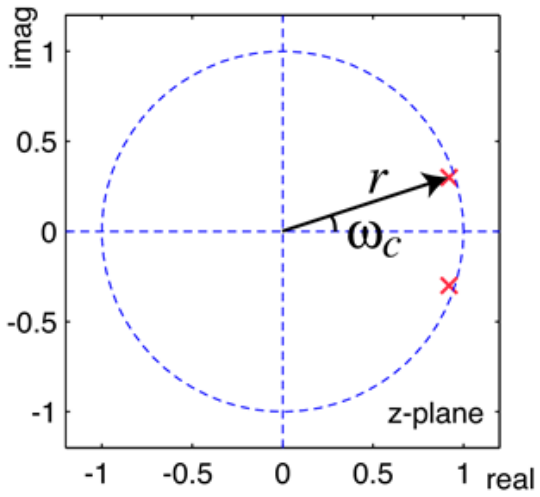
Akustický model

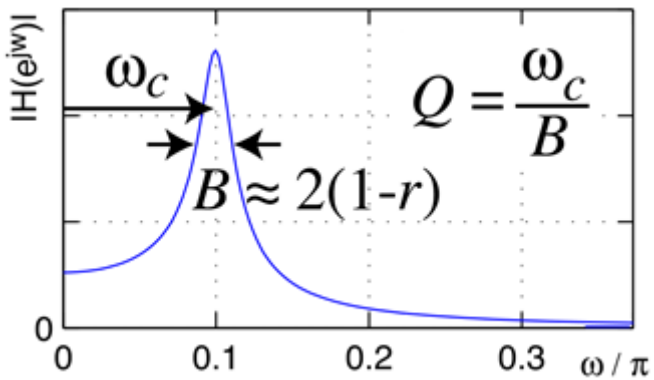
- **Rezonance v řeči**
  - Hlasové ústrojí (hrtan + jazyk + rty) působí jako variabilní rezonátor
  - **rezonance = „formanty“**

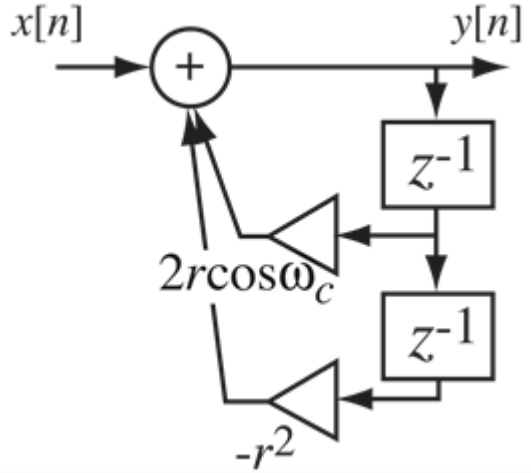


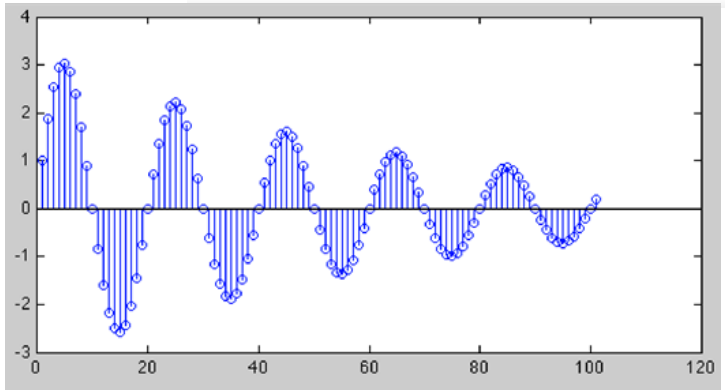
Akustický model

- **Rezonance s IIR filtry 2. řádu**
  - Rezonance v hlasovém ústrojí
  - Jednoduchá rezonance = IIR 2. řádu (pásmová propust)
    - Póly
    - Snadno implementovatelné filtry



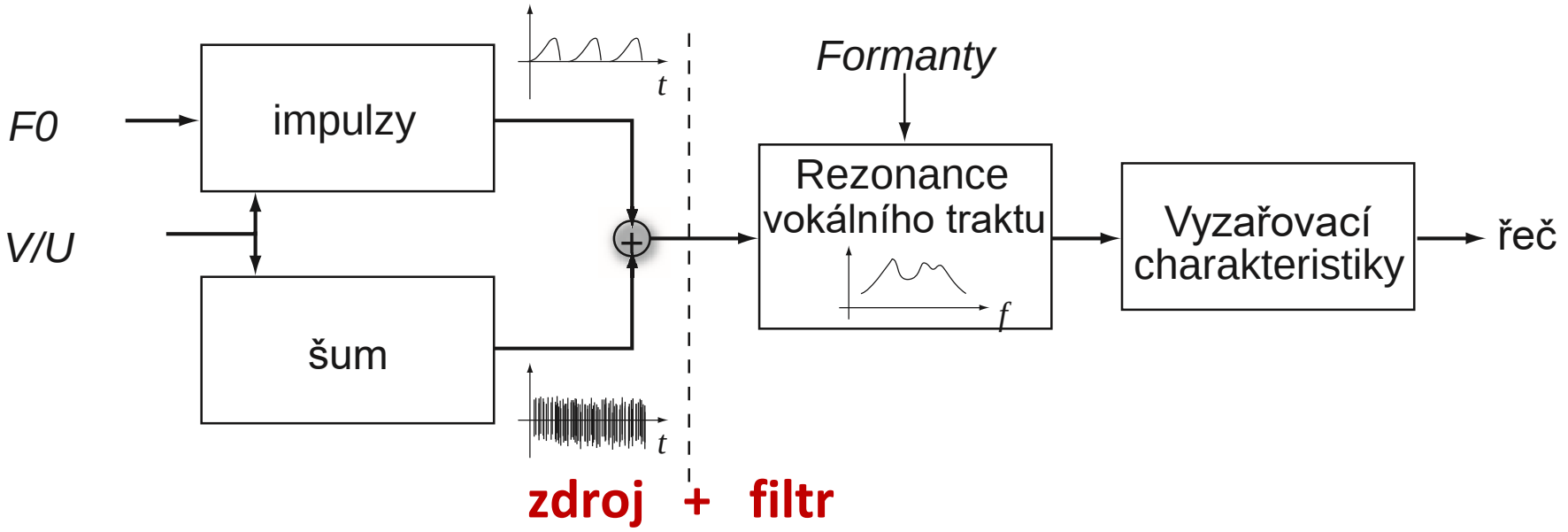






Akustický model

- **Model „zdroj + filtr“**
  - Fant v 60. letech a Rabiner v 70. letech 20. století
  - Model odděluje hlasivky (**zdroj**) a hlasové ústrojí (**filtr**)
  - LPC (Linear Predictive Coding) modeluje řeč jako výstup autoregresního filtru s buzením *e*[*n*].



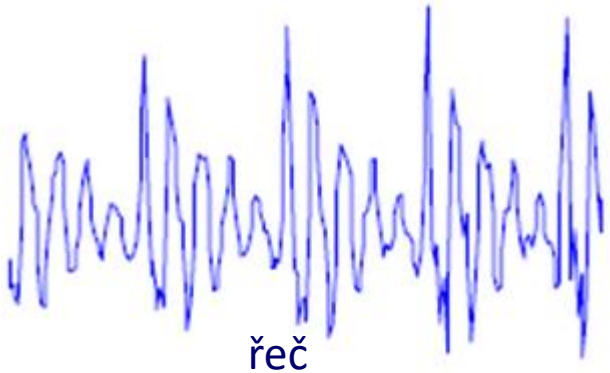
AR modelování řeči

LPC = lineární predikční kódování

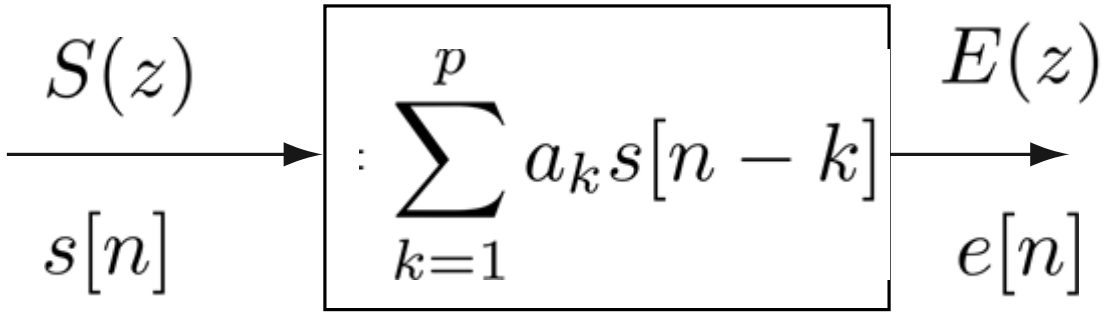
- LPC rovnice: predikce následujícího vzorku jako lineární kombinace předchozích hodnot

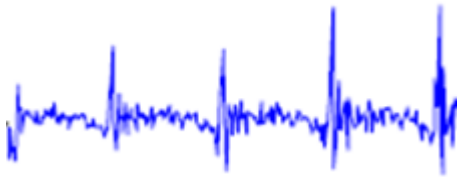
$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + e[n]$$

- {$a[k]$} jsou lineární predikční koeficienty p-tého řádu
- e[n] je reziduální signál (chyba predikce)



řeč

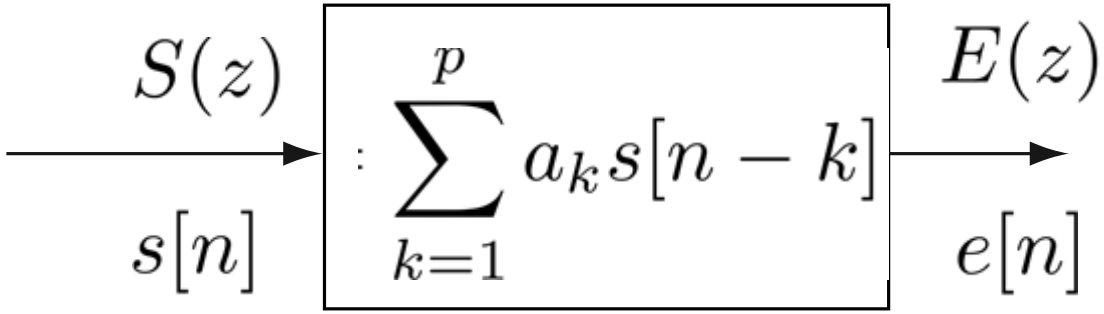




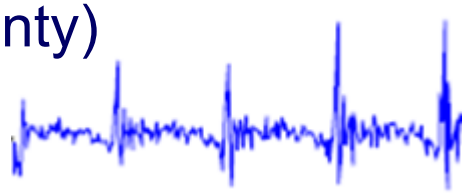
chyba predikce

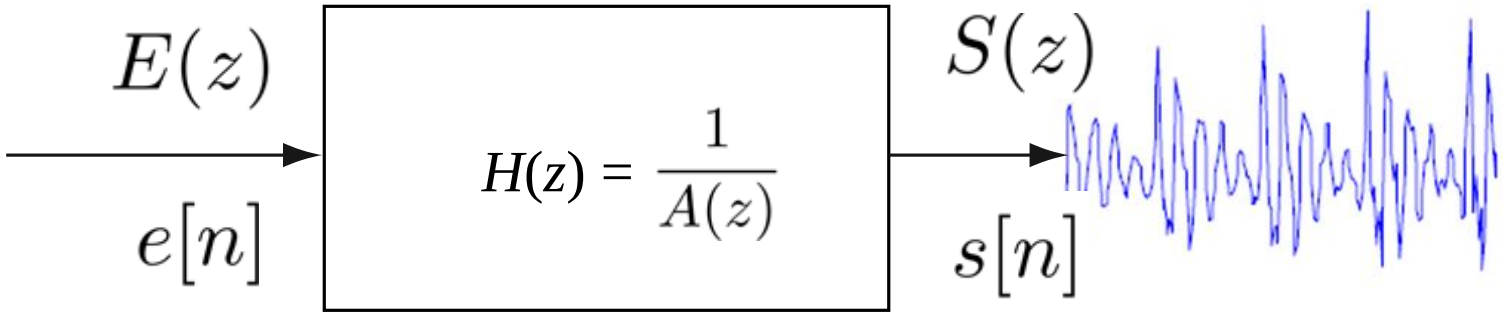
AR modelování řeči

- **Analyzující filtr**
  spektrální obálku (formanty) „popíše“ koeficienty



- **Syntetizující filtr**
  z koeficientů „obnoví“ obálku (formanty)



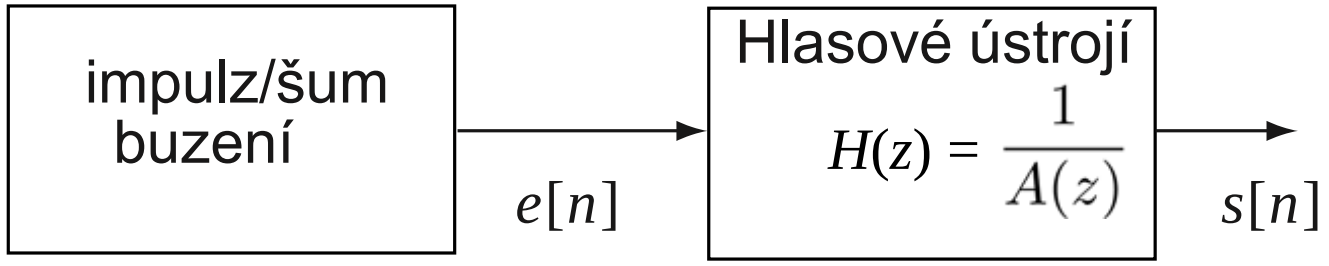


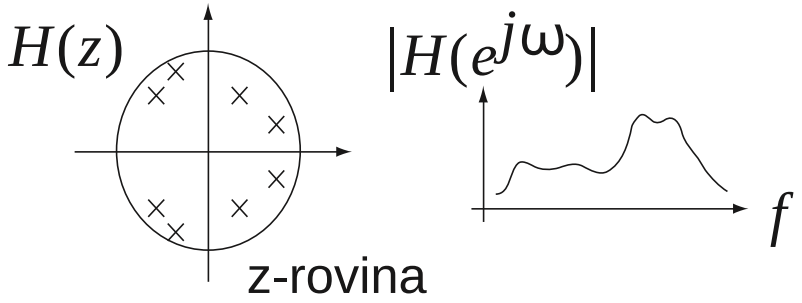
- **Přenosová funkce** představuje "autoregresivní" (AR) model (all-pole)

$$\frac{S(z)}{E(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{1}{A(z)}$$

AR modelování řeči

- **Spektrální obálka jako model hlasového ústrojí**
  - Filtr (1/*A*(*z*)): Syntetizující filtr *H*(*z*) je inverzní k analyzujícímu *A*(*z*).
  - Časově proměnné filtry modelují spektrální obálku vokálního ústrojí.
  - Buzení může být posloupnost pulsů (znělé) nebo šum (neznělé).





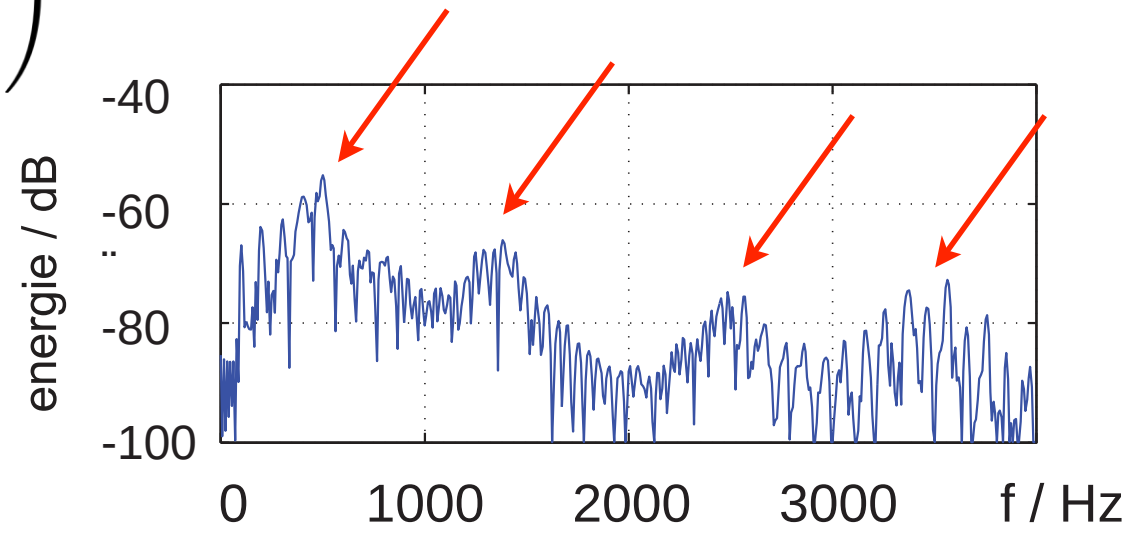
AR modelování řeči

- **Odhad AR koeficientů pomocí LPC**
  - Odhad AR koeficientů:
    - Minimalizace energie chyby predikce
    - Použití metody nejmenších čtverců vede k Yule-Walkerovým rovnicím založených na autokorelaci řeči

$$\sum_n e^2[n] = \sum_n \left(s[n] - \sum_{k=1}^{p} a_k s[n-k] \right)^2$$

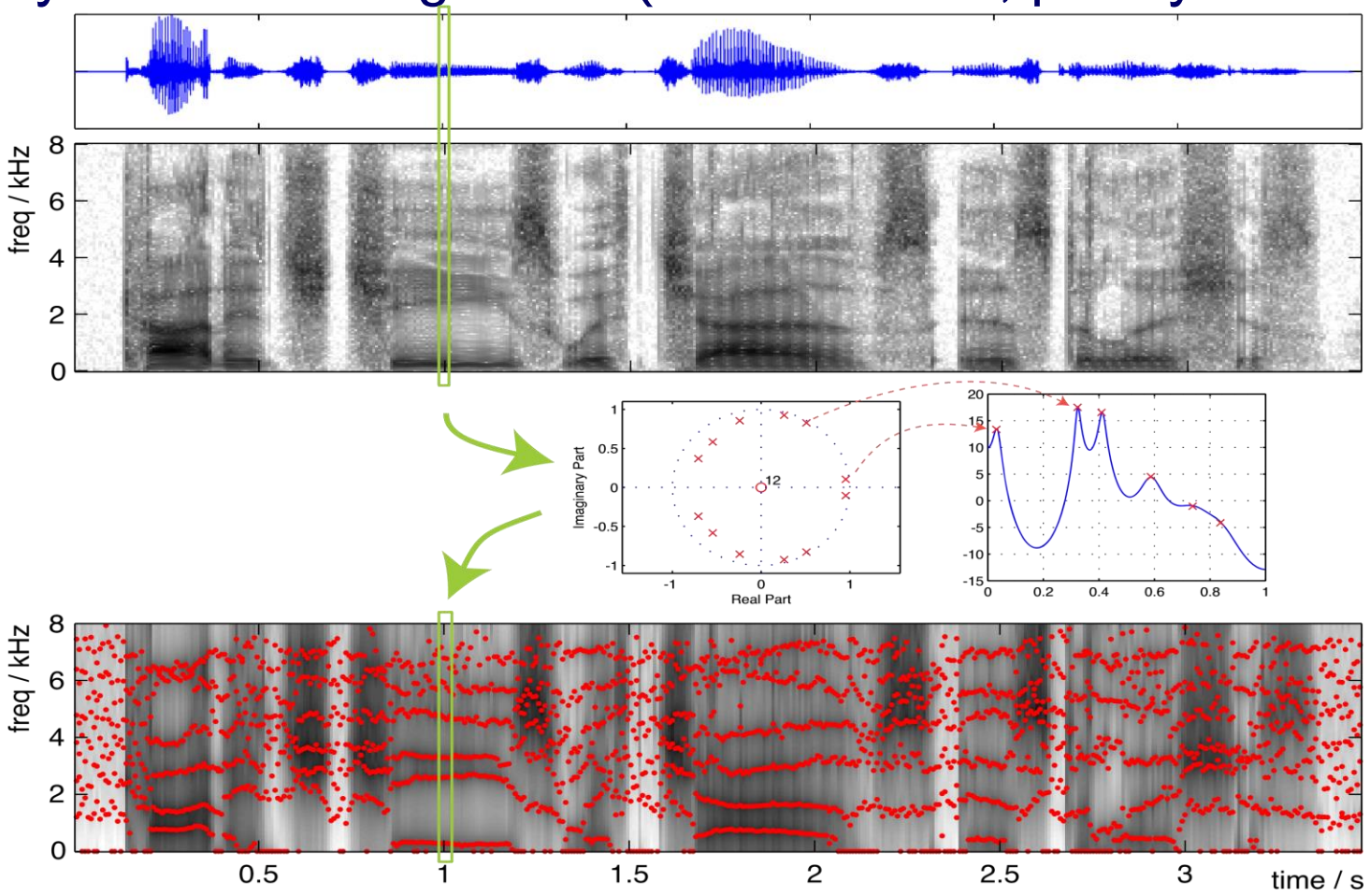
    - LPC se zaměřuje na spektrální obálku, nikoli na přesný průběh signálu

a = lpc(signal, p)



AR modelování řeči

- **LPC analýza**
  - LPC výpočty v každém segmentu (okno ~30 ms, překrytí ~10 ms)



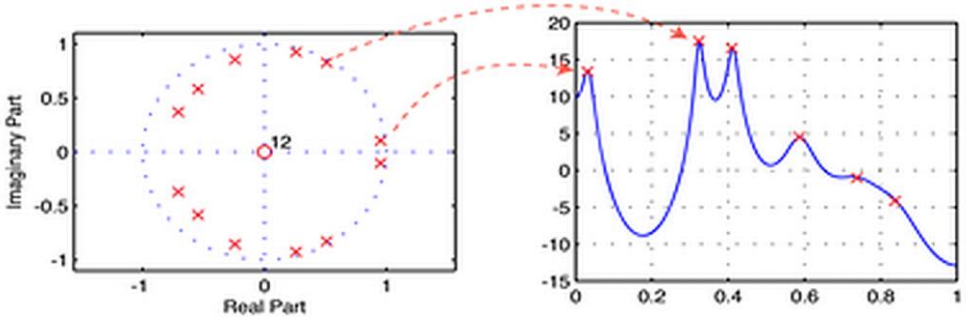
AR modelování řeči

- **Odhady formantových frekvencí pomocí LPC**
  - Výpočet LPC v každém segmentu
  - Použití kořenů polynomů k odvození formantových frekvencí

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_p z^{-p}$$

$$r_i = |r| e^{j\omega};$$

- `a = lpc(vowel_signal, order);`
- `r = roots(a);`



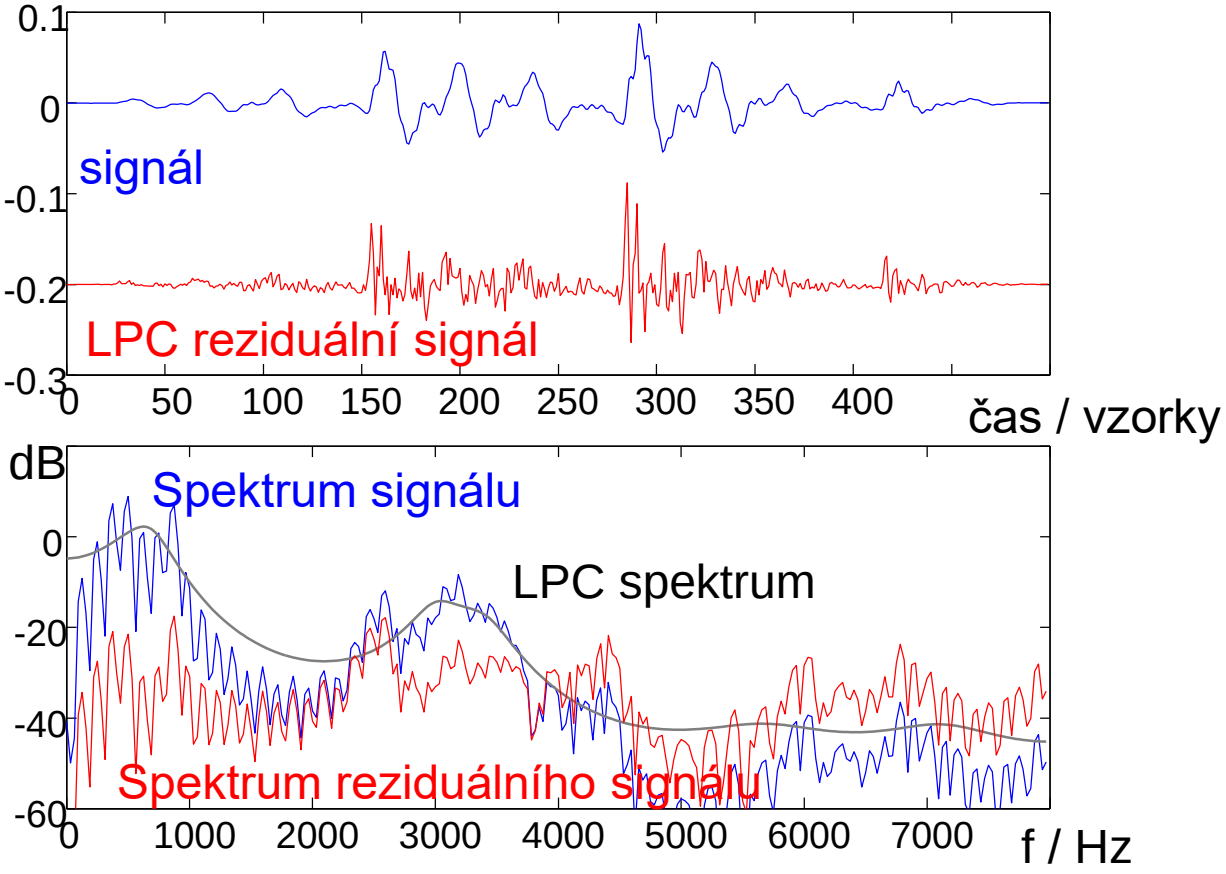
  - Přepočet kořenů na frekvenci $f_i = \dfrac{\omega_i \cdot f_s}{2\pi}$

  - Šířka pásma $B_i = -\dfrac{\ln(|r|) \cdot f_s}{\pi}$

- `formant (i) = angle(r(i))*fs/(2*pi);`
- `bw(i)       = -log(abs(r(i)))*fs/pi;`

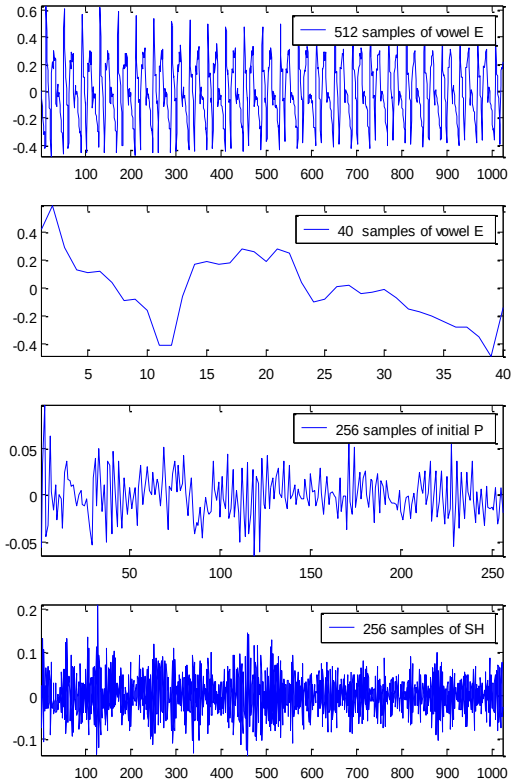
AR modelování řeči

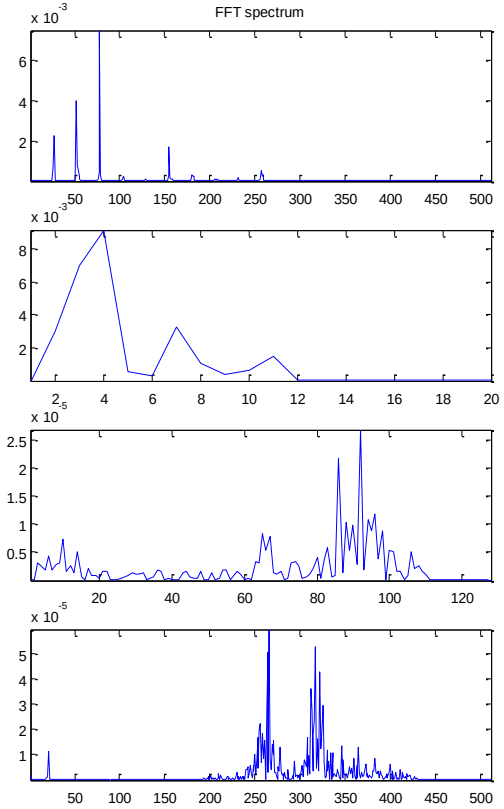
- **Porovnání LPC a FFT spekter**
  - FFT spektrum zachycuje celý frekvenční obsah, zatímco LPC se zaměřuje na spektrální obálku.



LPC analýza a syntéza

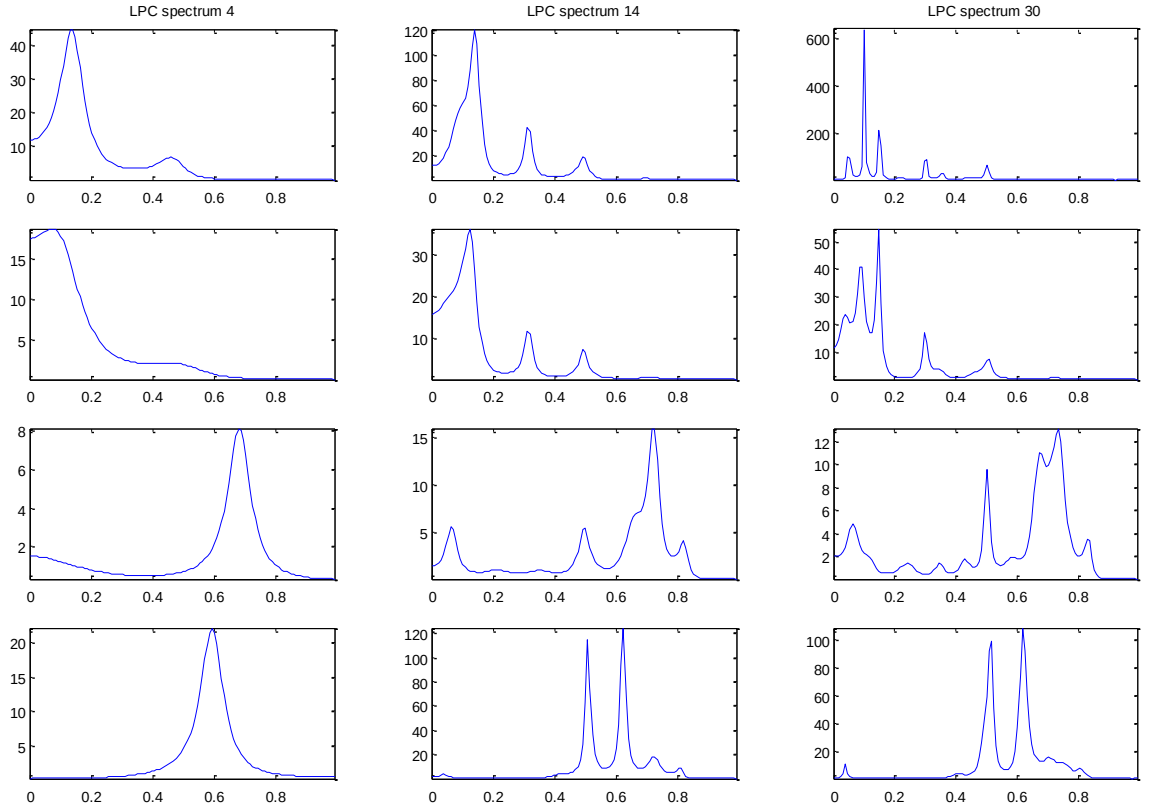
- **Vliv řádu modelu na spektrum**





FFT Spectra of Speech Sounds

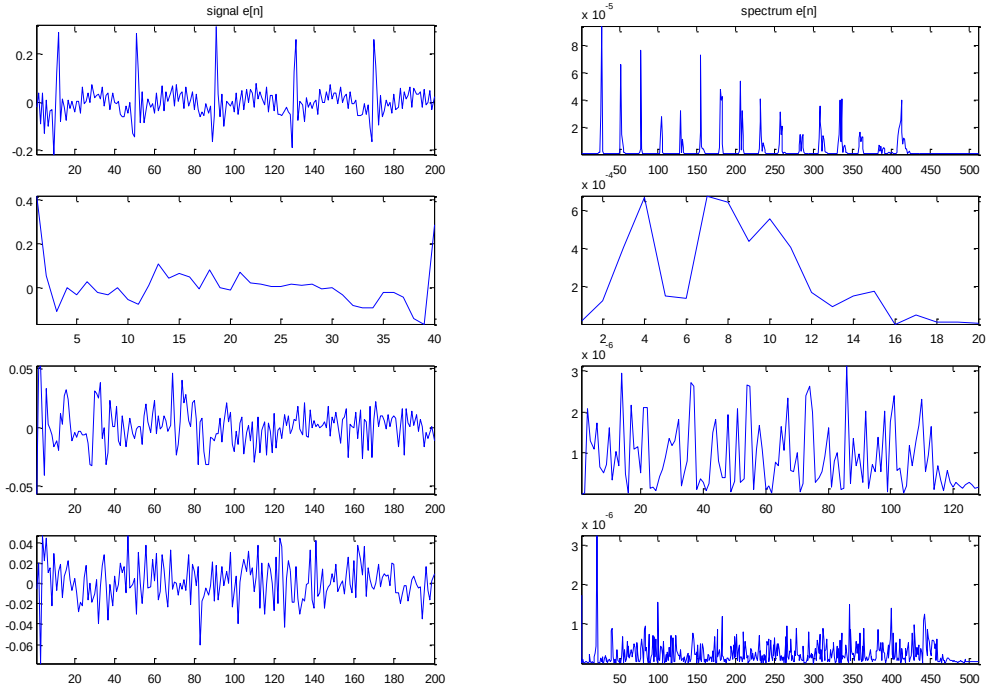
Different model orders (4, 14, 30)



Increasing model order increases spectral detail but can lead to overfitting.

LPC analýza a syntéza

- **Analýza chyby predikce v LPC**
  - Chyba predikce *e*[*n*] představuje **rozdíl mezi skutečným** signálem **a** signálem **predikovaným** modelem LPC.
  - Zachycuje detaily signálu, které model nezohledňuje, což z něj činí klíčovou součást pro různé aplikace.
  - Použití *e*[*n*]
    - Buzení pro syntézu
    - Kódování pro přenos
    - Sledování základní frekvence
    - Výběr řádu modelu



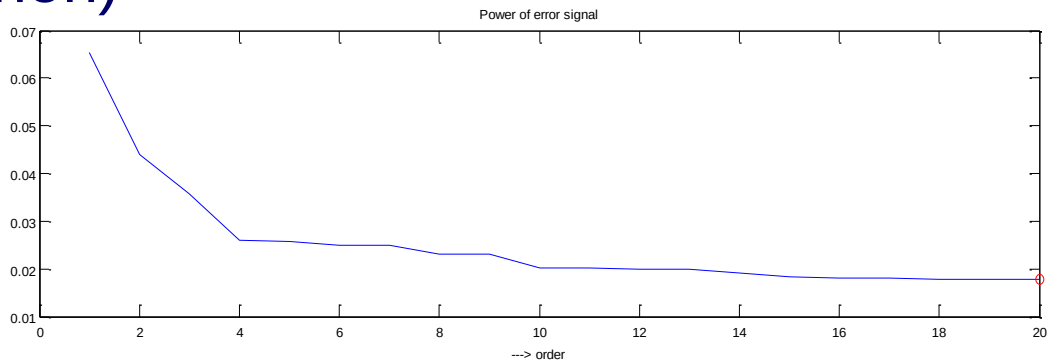
LPC analýza a syntéza

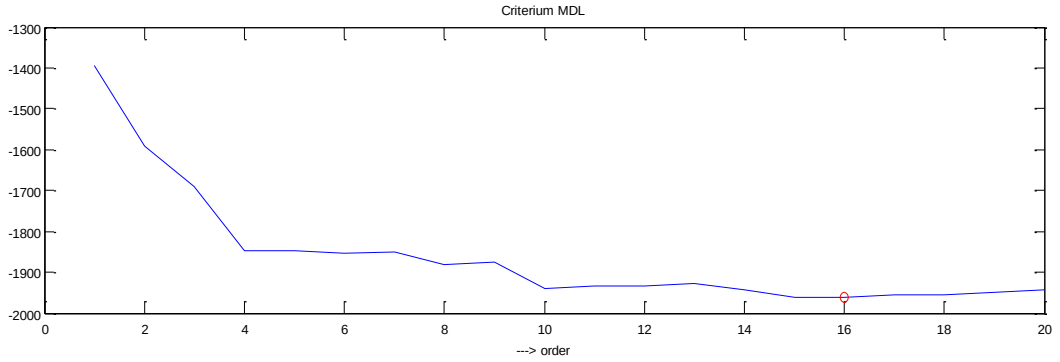
- **Výběr řádu modelu pomocí penaltových kritérií**

alfa	Criterion	
2	AIC	(Akaike Information Criterion)
ln(*N*)	MDL	(Minimum Description Length Error)
ln(ln(*N*))	HQ	(Hannan Quinn Criterion)
2ln(ln(*N*))	PHI	(Pukkila Criterion)

$$C(p) = N \ln s^2 + \alpha \cdot p$$

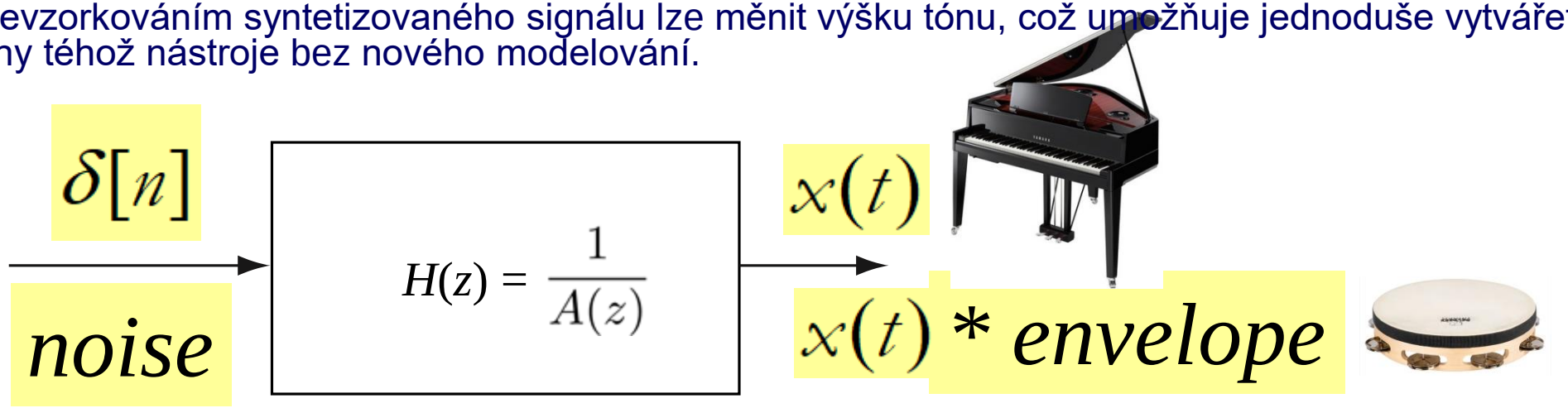
  - AIC(p) = N ln(MSE(p)) + 2 * p
    - Tends to overestimate order
  - MDL(p) = N ln(MSE(p)) + p * ln(N)
    - Statistically consistent





Modelování hudebních nástrojů pomocí LPC

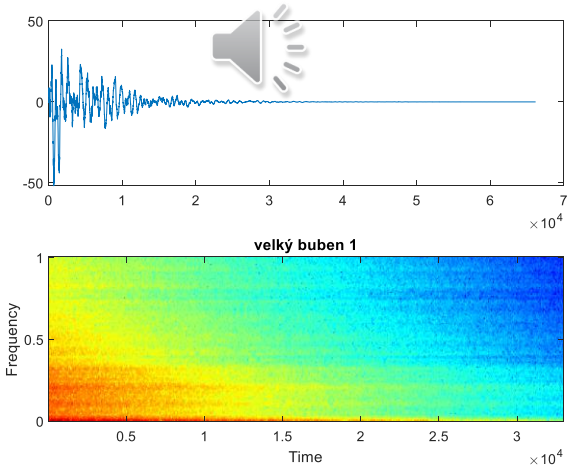
- **Simulace hudebních nástrojů s využitím LPC a buzení**
  1. LPC a hudební signály
     - Metoda vhodná pro analýzu a syntézu zvuků hudebních nástrojů.
     - Zvláště dobře se uplatňuje u perkuzních nástrojů (bicí, tamburína apod.), jejichž zvuk má krátký a přechodový charakter.
  2. Modelování pomocí různých buzení
     - **Impulsní buzení** → napodobuje ostrý atak nástrojů (např. klavír, buben).
     - **Buzení bílým šumem** → simuluje rozptýlenější zvuky (např. tamburína, virbl, činely).
  3. Změna výšky tónu (pitch shifting)
     - Převzorkováním syntetizovaného signálu lze měnit výšku tónu, což umožňuje jednoduše vytvářet různé tóny téhož nástroje bez nového modelování.

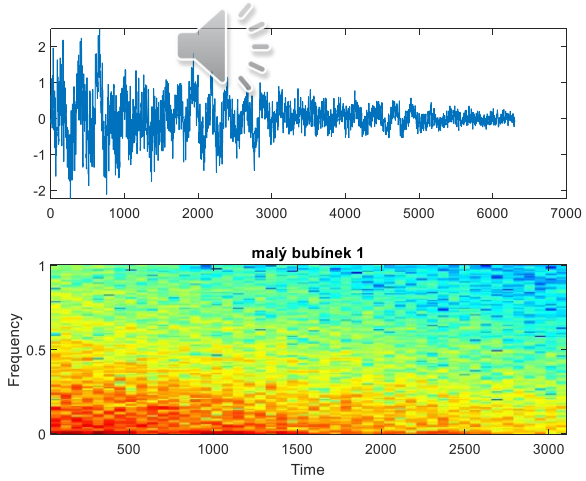


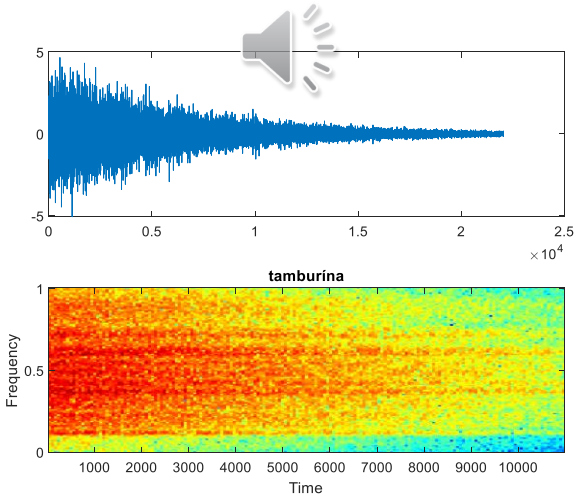
LPC model zde plní roli akustického filtru, který tvaruje spektrum budicího signálu

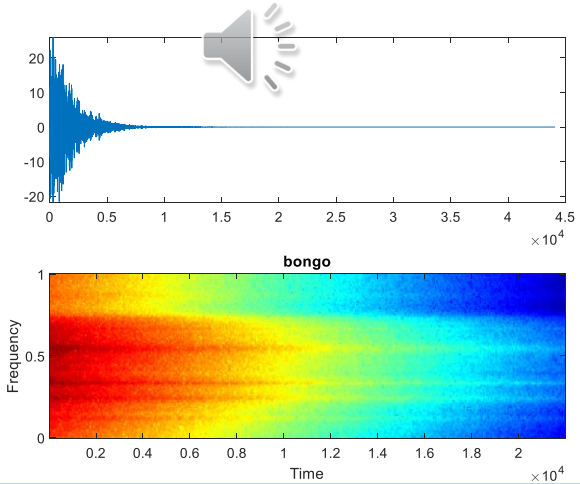
LPC modelování hudebních nástrojů

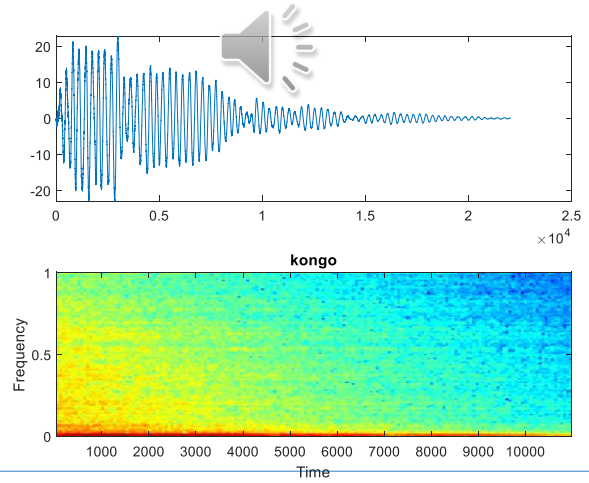
- **Simulace nástrojů s využitím LPC a buzení**

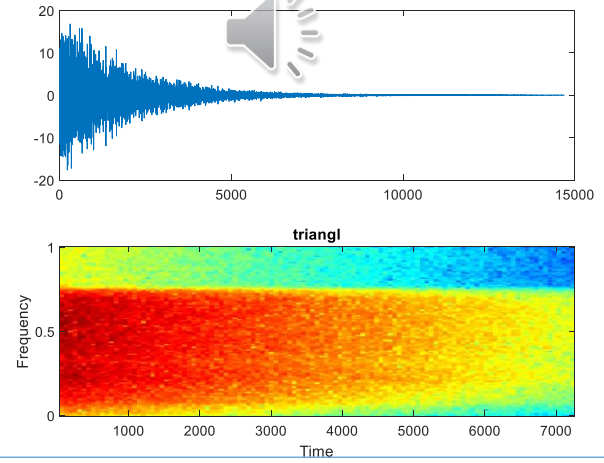






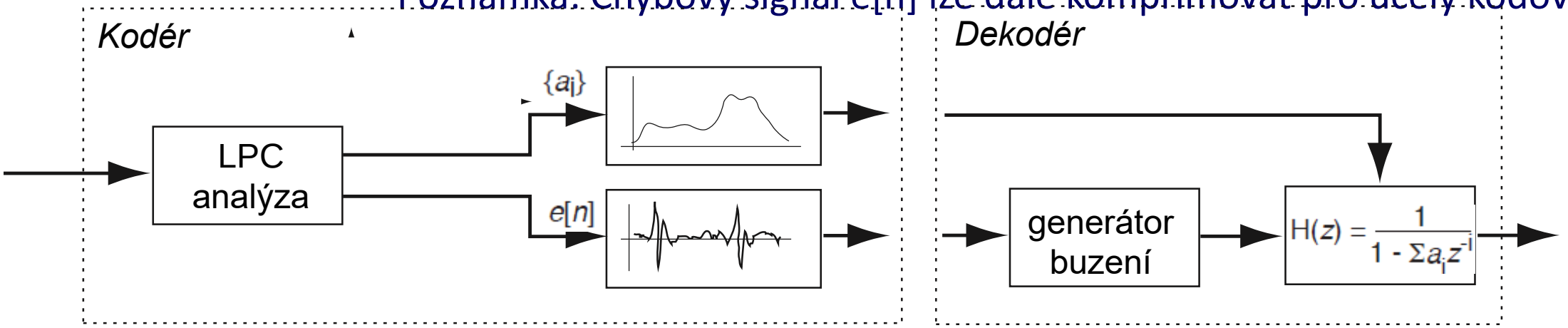






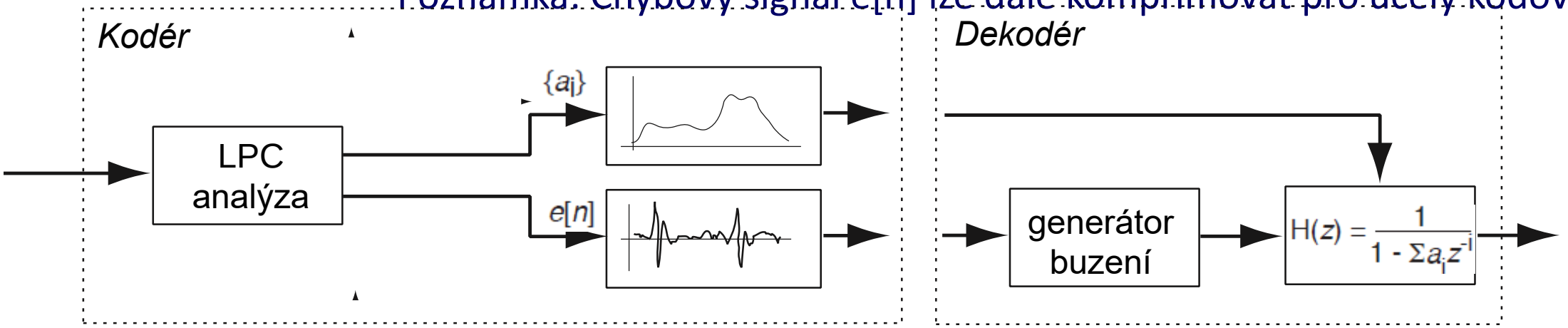
LPC vokodér: typy buzení a transformace

- **Krátkodobá LPC analýza a syntéza řeči**
  - Analýza řeči probíhá **po krátkých úsecích** (~20 ms). V každém segmentu se vypočítají **predikční koeficienty filtru** A(z) a **chybový (excitační) signál** e[n]
    - Znovusložením A(z) a e[n] lze přesně **obnovit původní řečový signál** s[n].
    - Tento princip tvoří základ LPC vokodéru.
      - Poznámka: Chybový signál e[n] lze dále komprimovat pro účely kódování řeči.

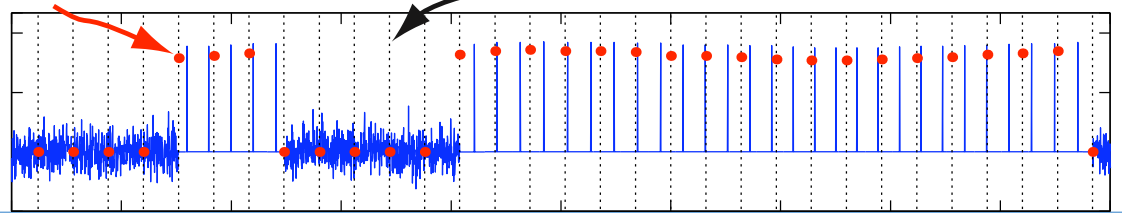


LPC vokodér: typy buzení a transformace

- **Krátkodobá LPC analýza a syntéza řeči**
  - Analýza řeči probíhá **po krátkých úsecích** (~20 ms). V každém segmentu se vypočítají **predikční koeficienty filtru** A(z) a **chybový (excitační) signál** e[n]
    - Znovusložením A(z) a e[n] lze přesně **obnovit původní řečový signál** s[n].
    - Tento princip tvoří základ LPC vokodéru.
      - Poznámka: Chybový signál e[n] lze dále komprimovat pro účely kódování řeči.

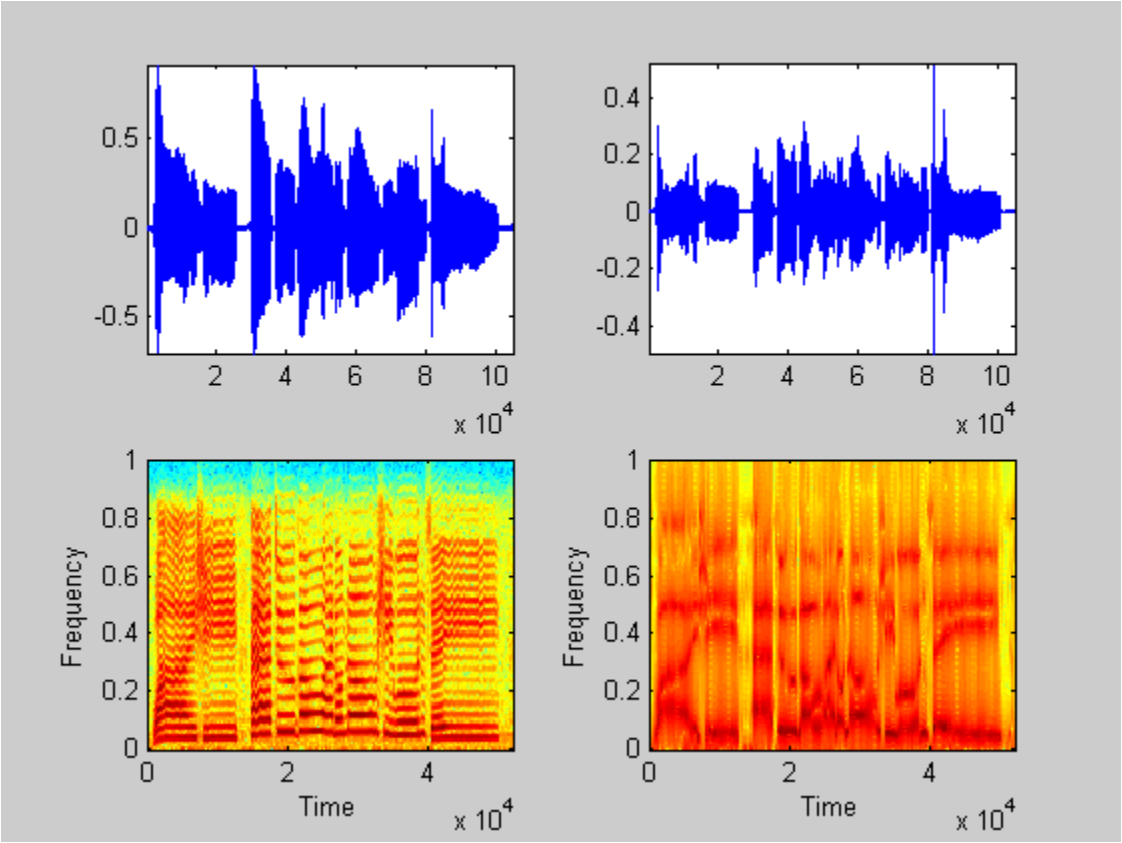


Hodnoty „ periody základního tónu“ jsou hodnoceny po 16ms segmentech



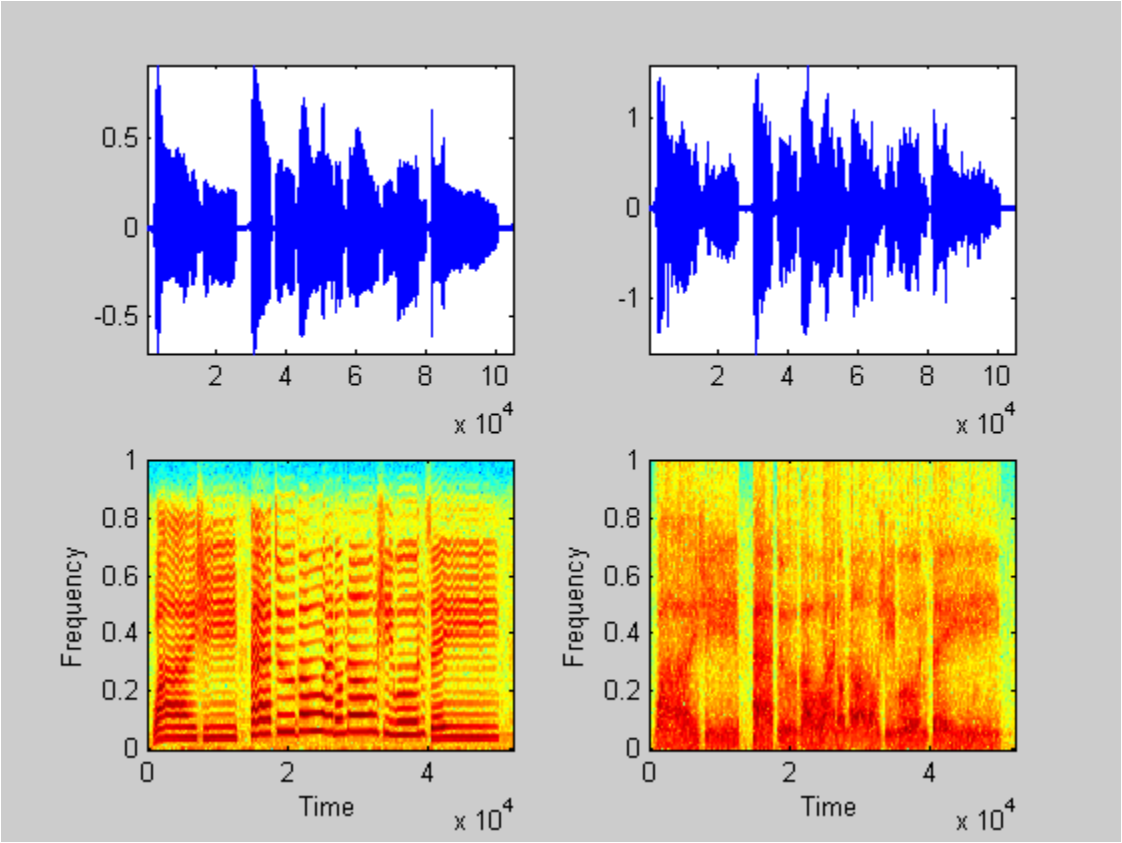
LPC vokodér: typy buzení a transformace

- **Syntéza znělé řeči** (impulsní buzení)
  - Impulsní buzení se používá pro znělé hlásky.
  - Tento přístup napodobuje periodické chvění hlasivek a je zásadní pro věrné zachování výšky tónu (pitch).



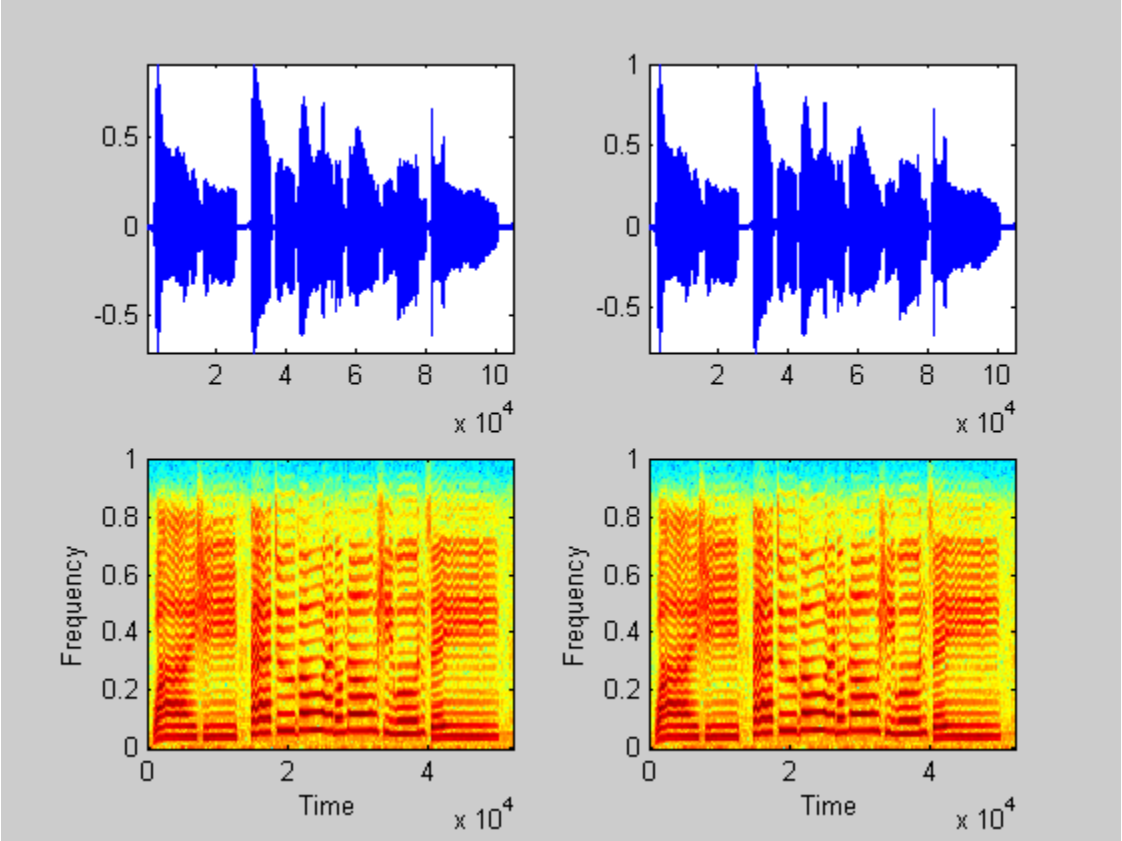
LPC vokodér: typy buzení a transformace

- **Syntéza neznělé řeči** (šumové buzení)
  - Náhodné (bílý šum) buzení se používá pro neznělé zvuky, např. frikativy (š, f, ch), kde je potřeba šumový charakter.
  - Kombinace šumu a filtrů LPC modelu umožňuje realistickou rekonstrukci těchto zvuků.



LPC vokodér: typy buzení a transformace

- **Syntéza řeči pomocí signálu chyby predikce**
  - V aplikacích vokodéru lze jako buzení použít chybový signál e[n].
  - Tento způsob se používá, když je potřeba věrně zachovat jemné detaily originálního zvuku, např. v případech přirozené řeči nebo zpěvu.



LPC vokodér: typy buzení a transformace

- **Úprava délky a výšky tónu v LPC vokodéru**

  - **coef1**: Řídí délku okna a dobu trvání signálu
    - coef1 < 1: kratší okna → rychlejší přehrávání (kratší signál)
    - coef1 > 1: delší okna → pomalejší přehrávání (delší signál)

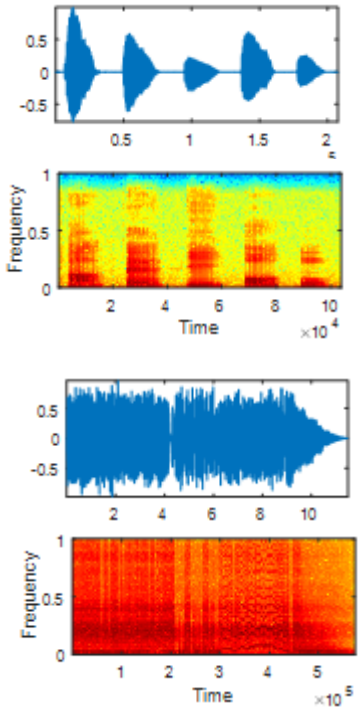
```
window_length2 = coef1 * window_length
```

  - **coef2**: Řídí výšku tónu bez změny délky signálu
    (tj. kompenzací přehrávací rychlosti)
    - coef2 < 1: nižší tón
    - coef2 > 1: vyšší tón

```
window_length2 = coef1 * coef2 * window_length
Playback Rate  = coef2 * fs
```

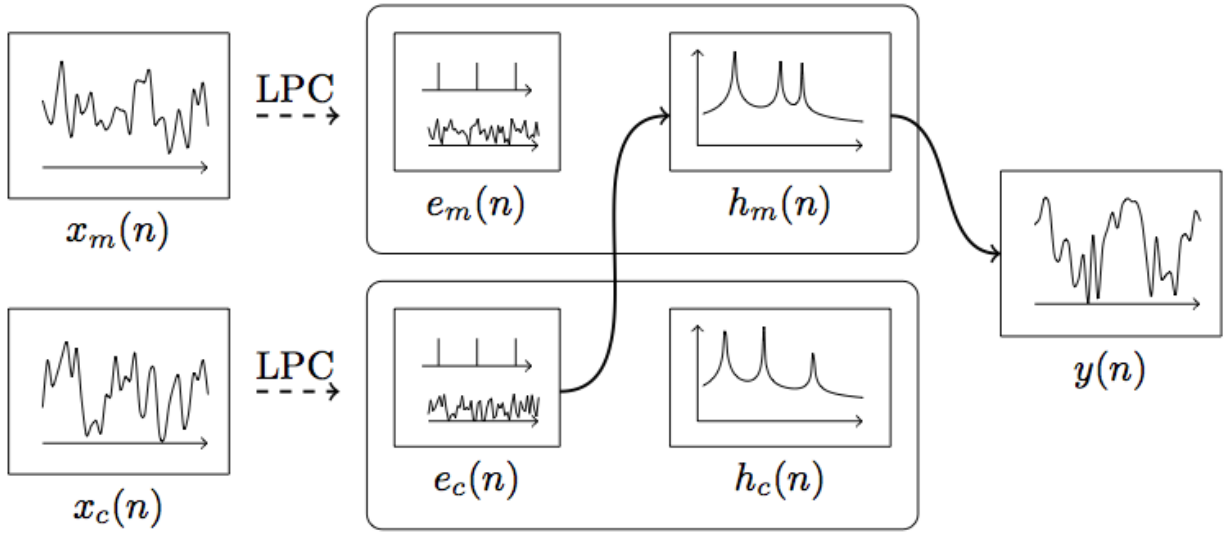
LPC vzájemná syntéza: audio efekty

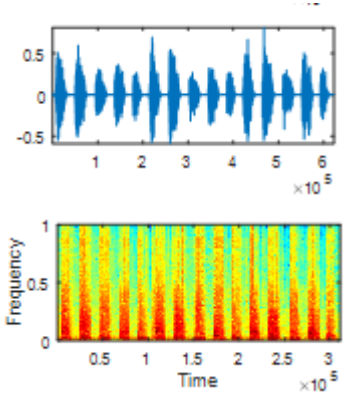
- **Využití LPC filtrů pro kreativní zvukový design**
  - Populární audioefekt „**cross-synthesis**“ (vzájemná syntéza) využívá **spektrální obálku** jednoho signálu k tvarování jiného.
    → Lze tak vytvořit efekty typu „*zpívající nástroj*“ nebo „*mluvící dveře*“.
  - Např. u efektu „*zpívající kytara*“ se **filtrační koeficienty odvozené z řeči** aplikují na signál kytary.



řeč (filtr)

hudba (zdroj)



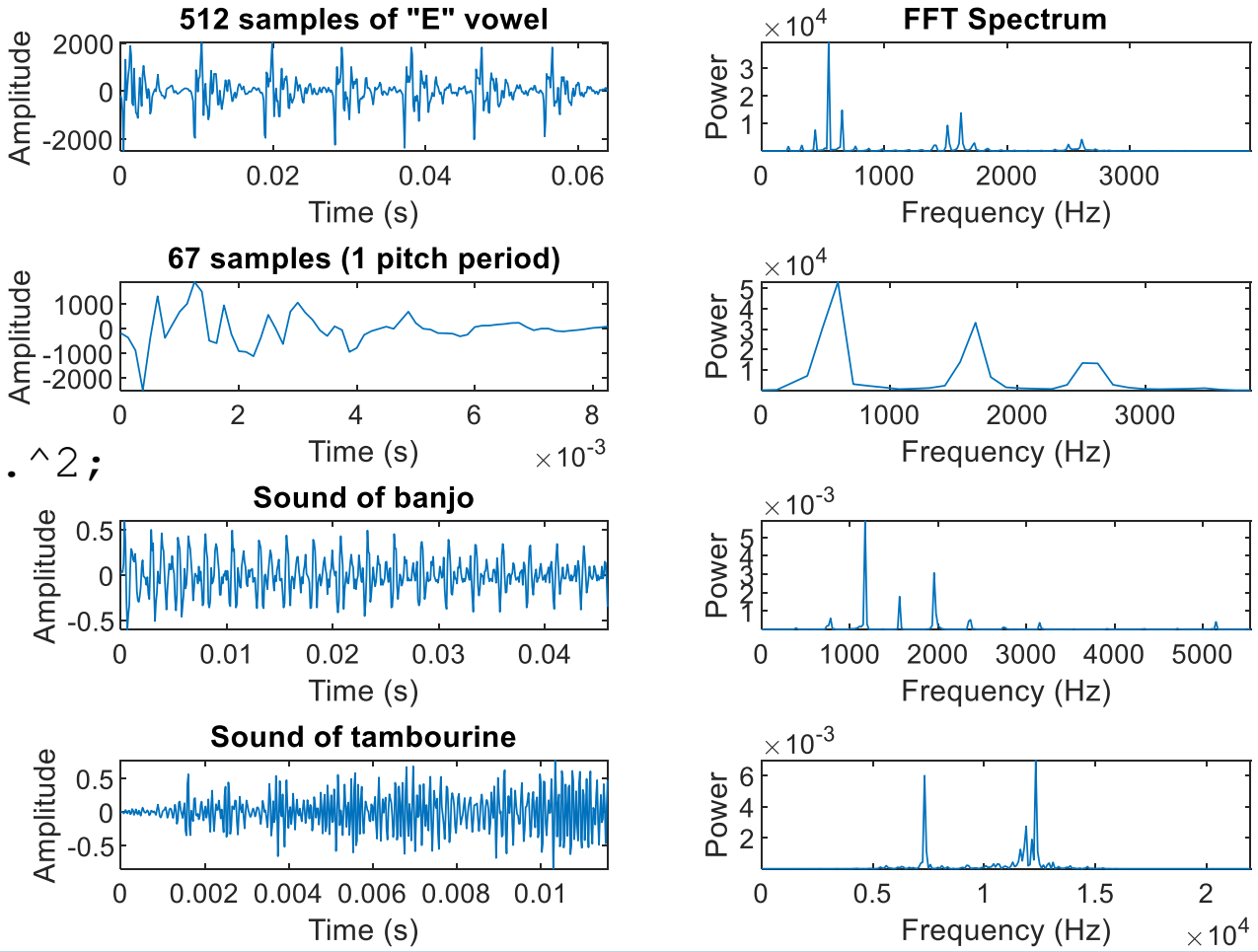


Výsledný zvuk má spektrální obálku řeči, ale časový průběh hudebního signálu

Příklady analýzy a syntézy LPC

- **Příklad 7.1: Audio signál a jeho FFT spektrum**
  - MATLAB:

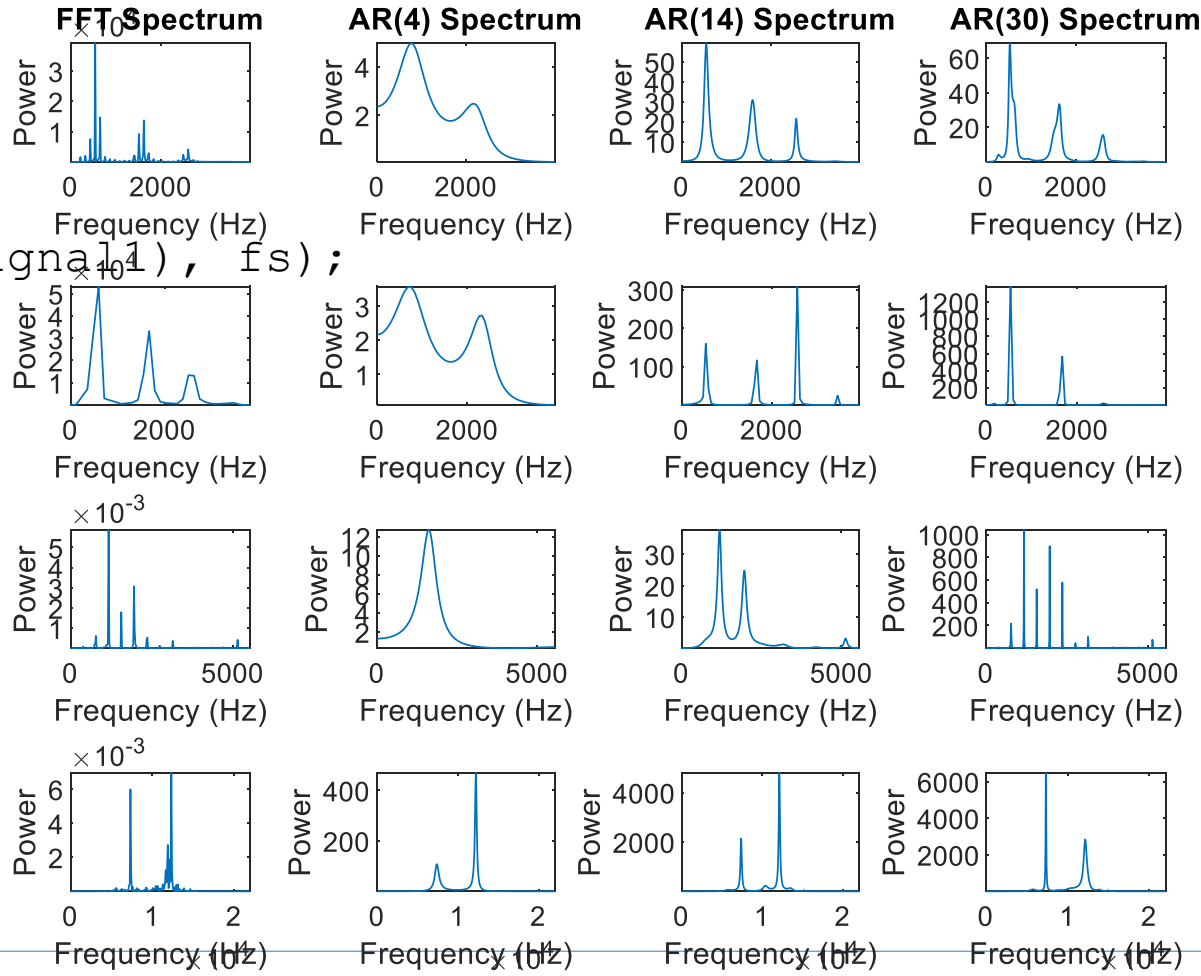
```
• load('pet1.txt')
• audioread('banjo.wav')

• fft_spectrum = fft(signal1);
• freq1 = (0:N1/2-1) * (fs1 / N1);
• psd_fft1 = (abs(fft(signal1) / N1)).^2;
• plot(freq_axis, abs(fft_spectrum));
```



Příklady analýzy a syntézy LPC

- **Příklad 7.2: AR spektra audio signálů**
  - MATLAB:

```
ar_order = 14;
a = lpc(signal1, ar_order);
[psd_ar, freq_ar] = freqz(1, a, length(signal1), fs);
plot(freq_ar, abs(psd_ar).^2);
```



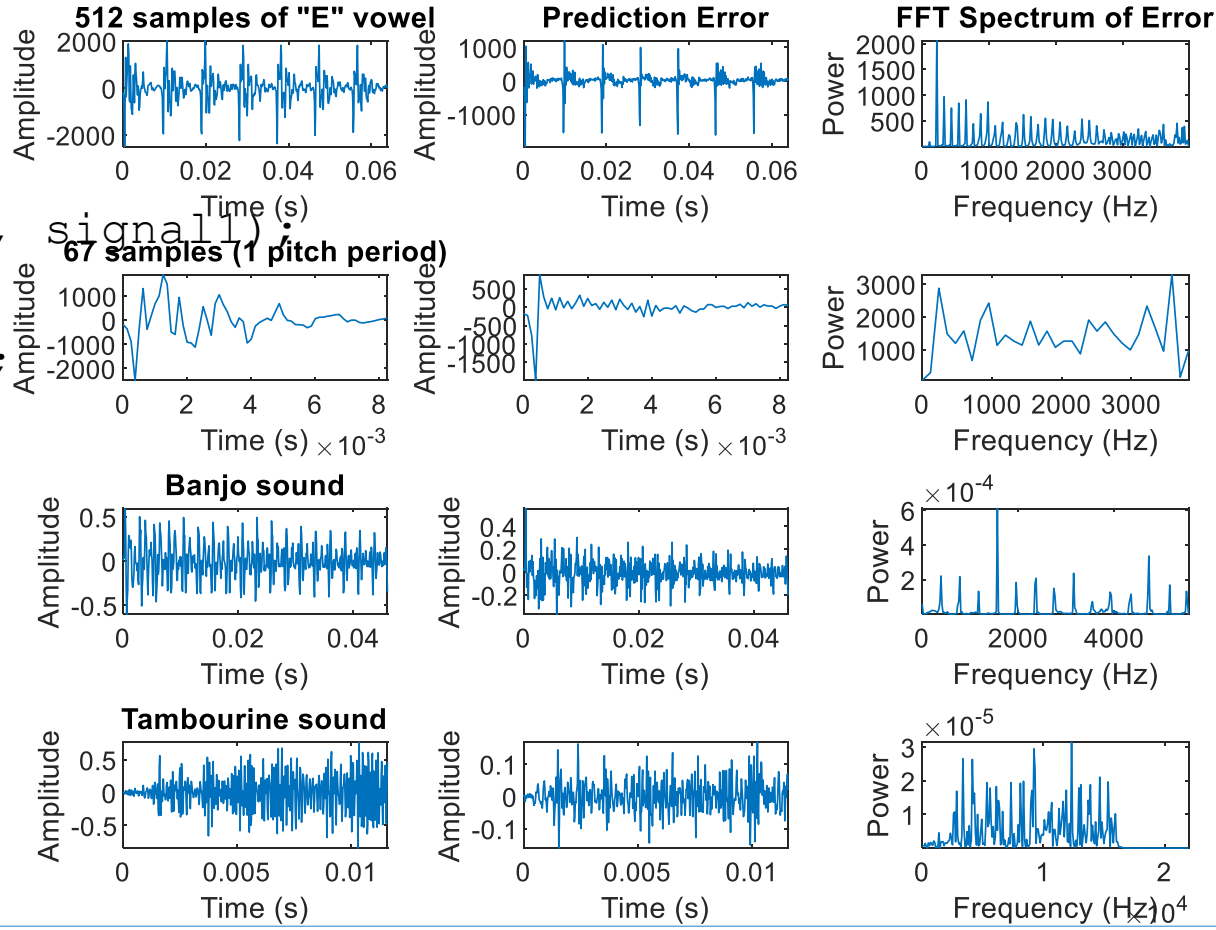
Příklady analýzy a syntézy LPC

- **Příklad 7.3: Signál chyby predikce**
  - MATLAB:

```
ar_order = 16;
a = lpc(signal1, ar_order);
pred_signal = filter([0 -a(2:end)], 1, signal1);
error_signal = signal1 - pred_signal;
psd_error = abs(fft(error_signal)).^2;
plot(freq_axis, psd_error);
```

% Alternative method to compute prediction error

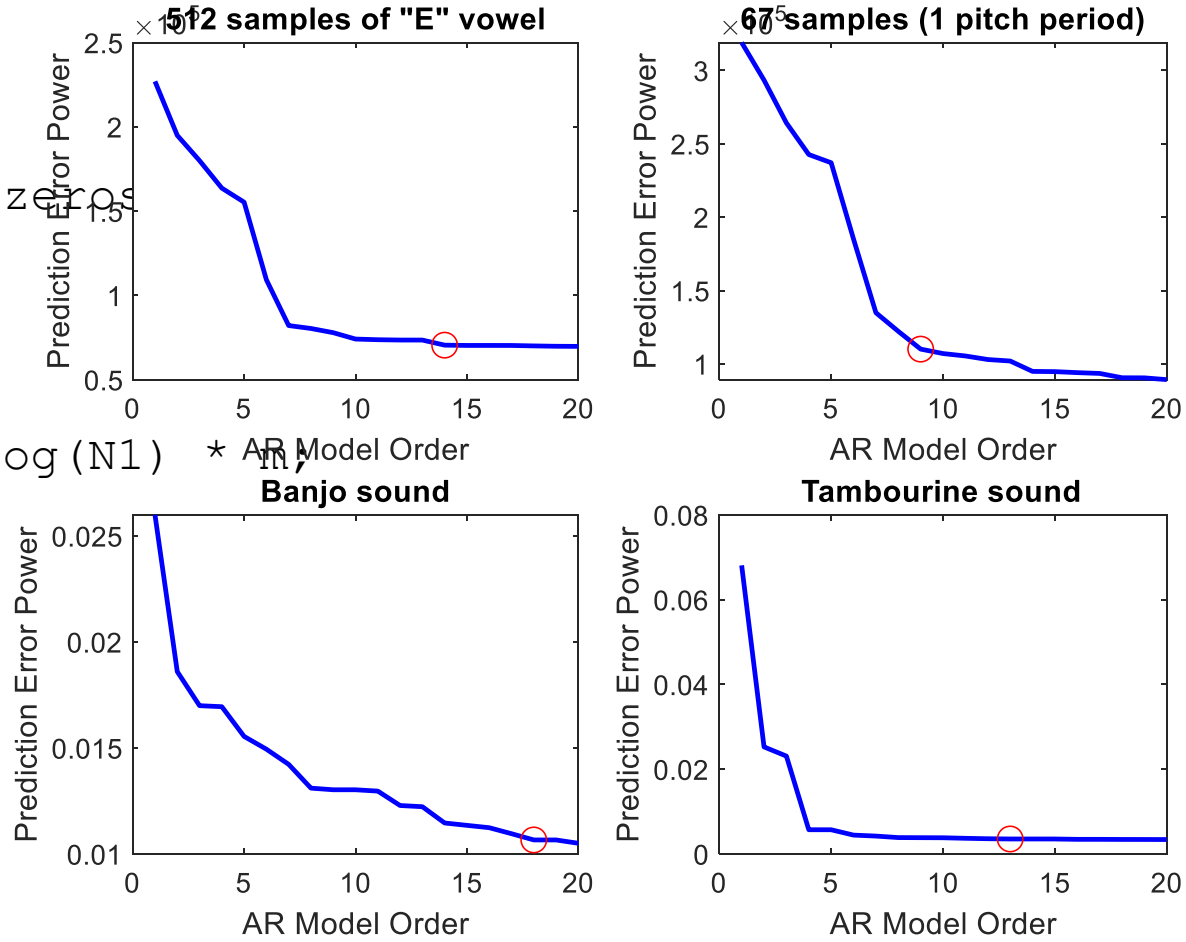
```
error_signal = filter(a, 1, signal1)
```



Příklady analýzy a syntézy LPC

- **Příklad 7.4: Odhad řádu AR modelu**
  - MATLAB:

```
max_order = 20;
    P1 = zeros(1, max_order); MDL1 = zeros
    for m = 1:max_order
        [a, p] = lpc(signal1, m);
        P1(m) = p;
        MDL1(m) = N1 * log(P1(m)) + log(N1) * m;
    end
    plot(1:max_order, MDL1);
```

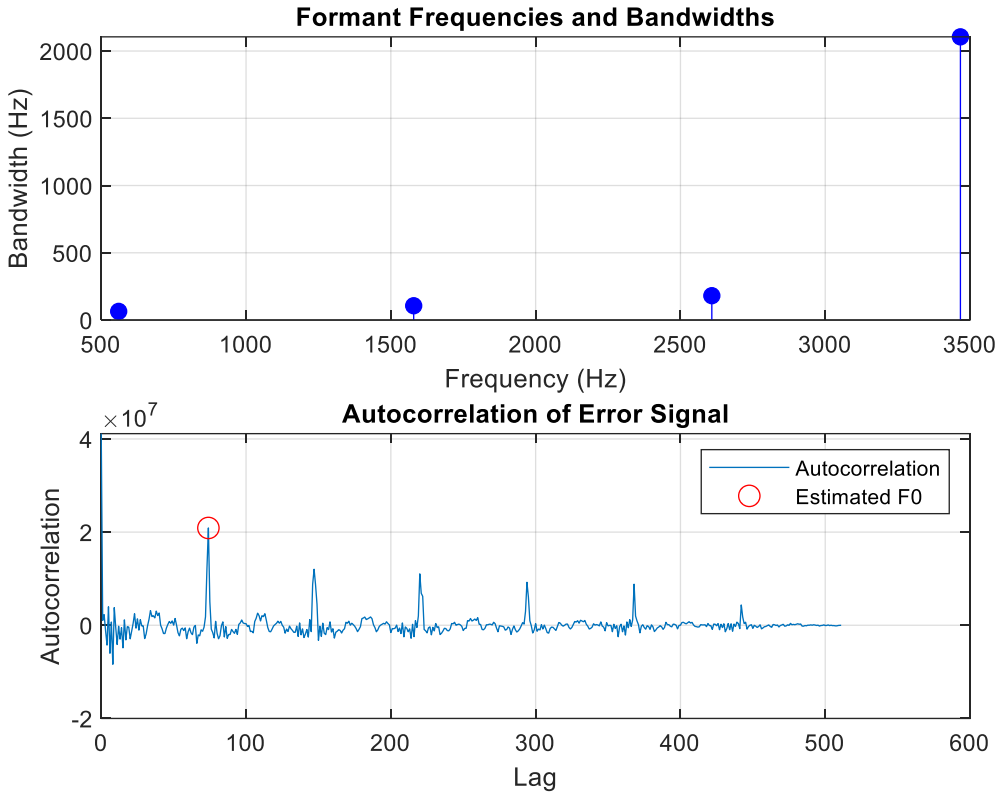


Příklady analýzy a syntézy LPC

- **Příklad 7.5: AR model – odhad výšky tónu a formantů**

  - MATLAB:

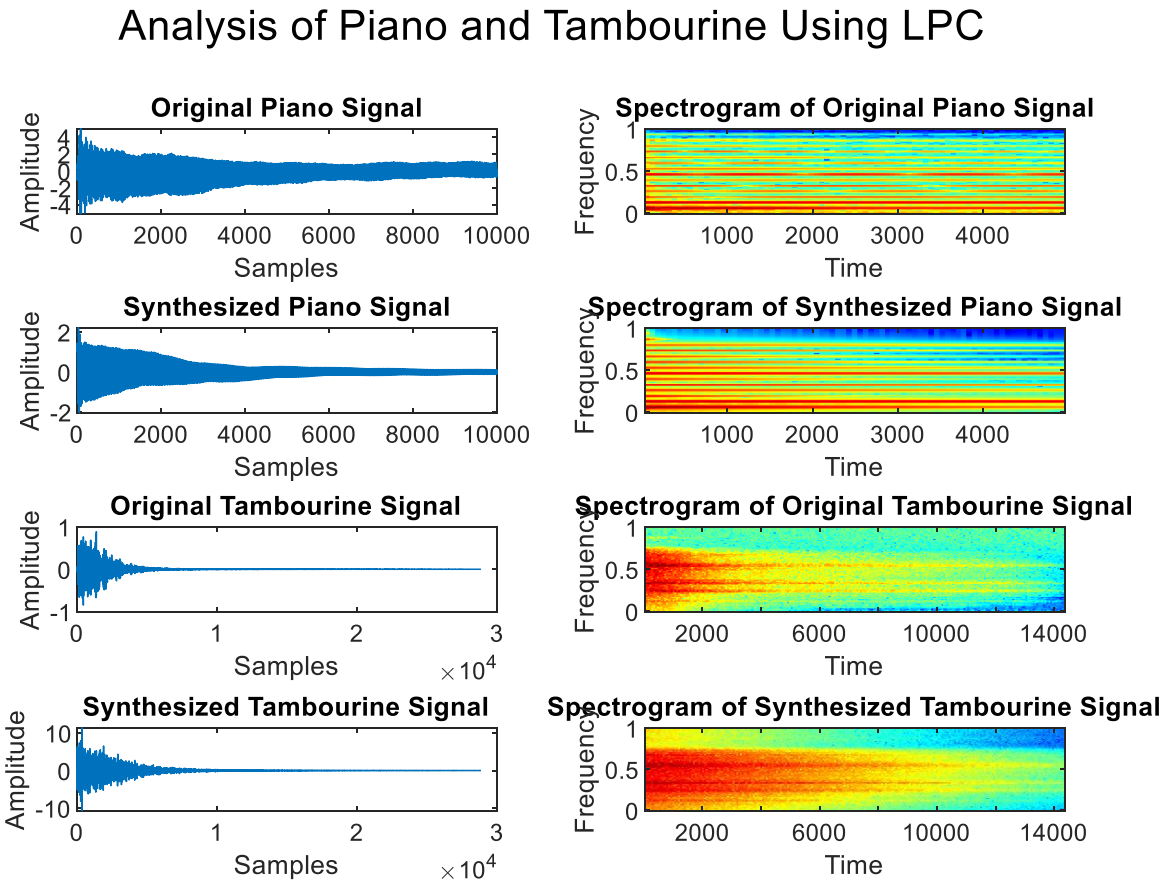
```
order = 8;
[a, ~] = lpc(signal1, order);
r = roots(a);
formant_freqs = angle(r) * Fs / (2 * pi);
bandwidths = -log(abs(r)) * Fs / pi;
[auto_corr, lags] = xcorr(errorSignal);
plot(lags, auto_corr);
```



Příklady analýzy a syntézy LPC

- **Příklad 7.6: AR modelování hudebních nástrojů pomocí impulsní odezvy**
  - MATLAB:

- `a = lpc(x, 50)`
- `y=filter(1, a, [1 zeros(1, 10000)])`

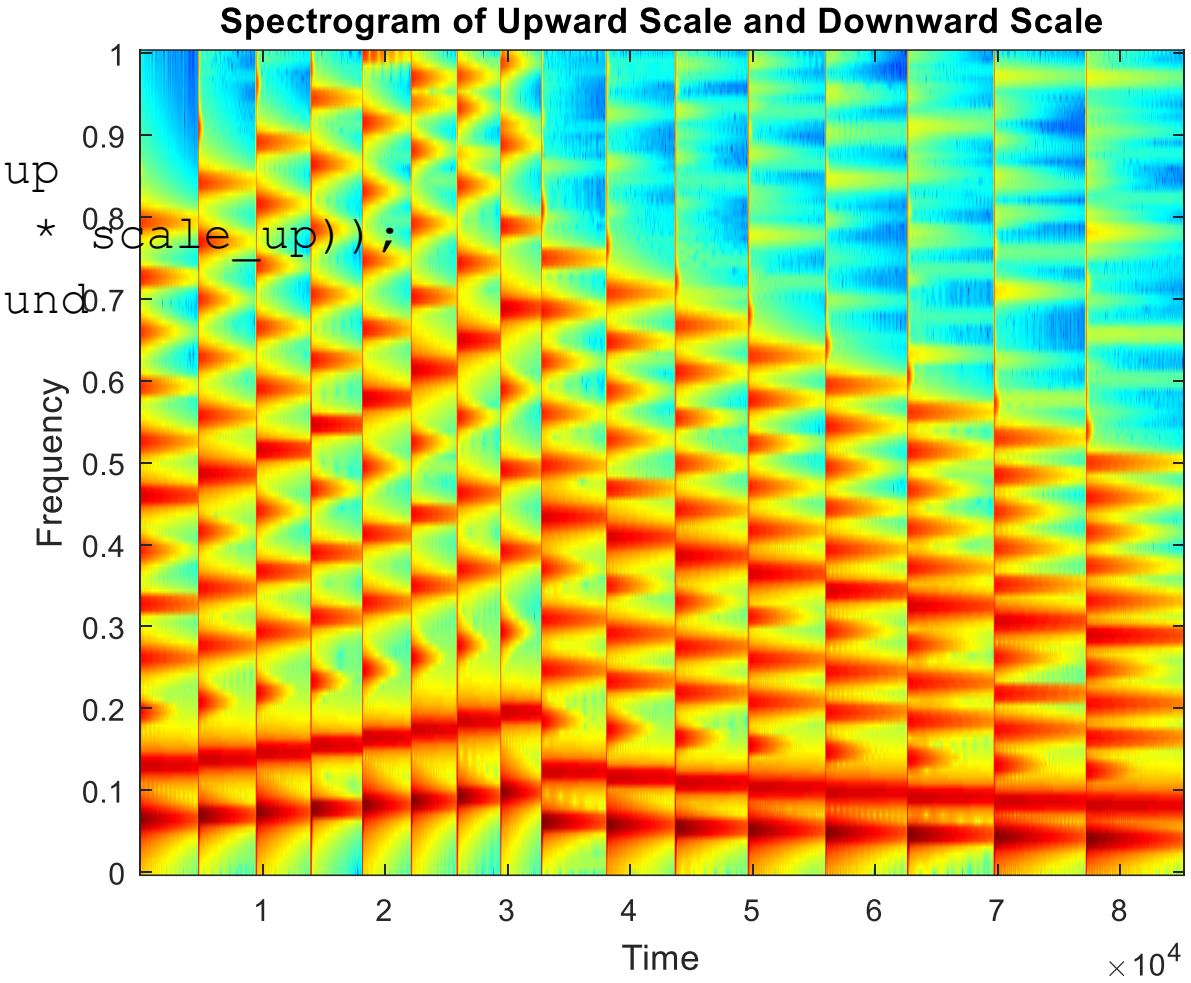


- `filter(1, a, [rand(1, length(x)) - 0.5]) .* exp(-[1:length(x)] ./ length(x) ./ 0.1)`

Příklady analýzy a syntézy LPC

- **Příklad 7.7: Posun výšky tónu pomocí převzorkování**
  - MATLAB:

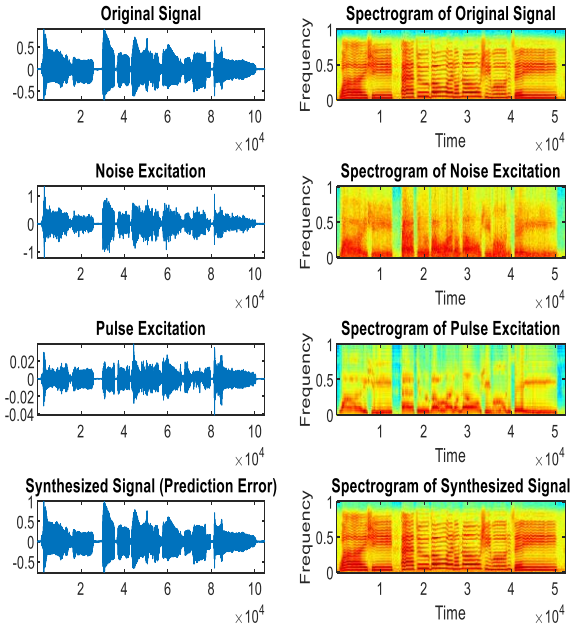
```
scale_up = 2^(1/12); % One semitone up
y_up = resample(y_orig, fs, round(fs * scale_up));
soundsc(y_up, fs); % Play shifted sound
```



Příklady analýzy a syntézy LPC

- **Příklad 7.8: LPC vokodér – různá buzení**
  - MATLAB:

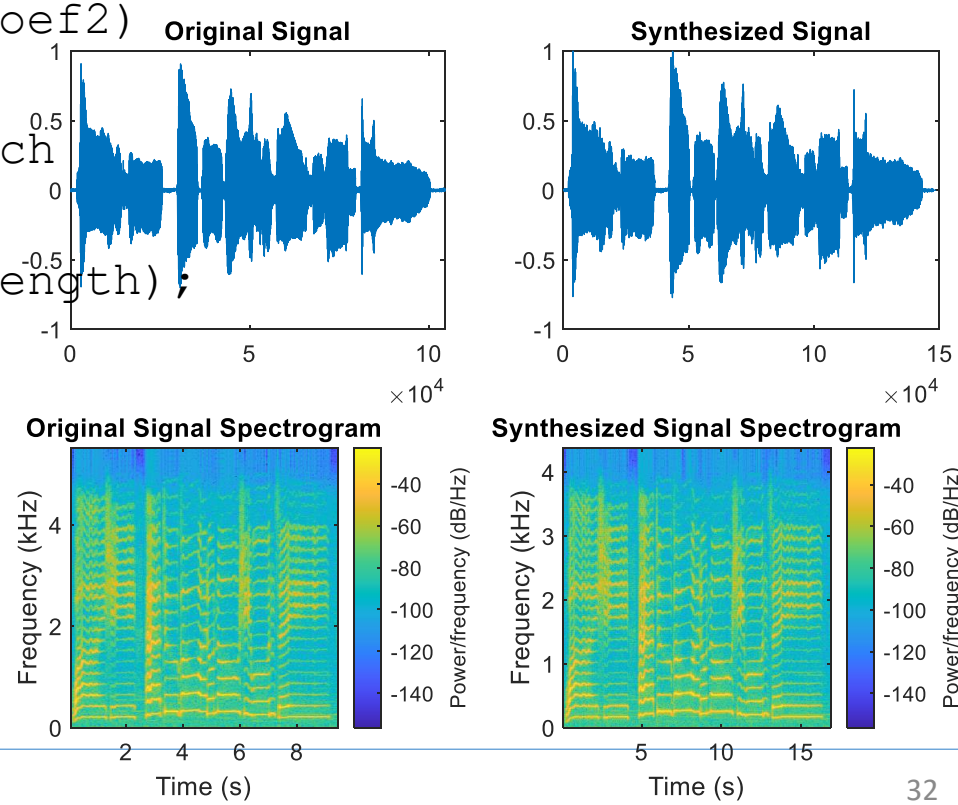
```
• lpc(frame, 8)
• segment_noise = filter(1, A(:, k), P(k) * randn(window_length, 1));
• segment_pulse = filter(1, A(:, k), P(k) * [1; zeros(window_length - 1, 1)]);

• Window:
• interp1([0, window_length / 3, 2 * window_length / 3,
  window_length], [0, 1, 1, 0], 0:window_length - 1)
```



Příklady analýzy a syntézy LPC

- **Příklad 7.9: LPC vokodér – časové a frekvenční transformace**
  - MATLAB:

```
% Adjustments for time (coef1) and frequency (coef2)
% coef1 < 1 for shorter; > 1 for longer time
% coef2 < 1 for lower pitch; > 1 for higher pitch

window_length2 = round(coef1 * coef2 * window_length);
window_shift2 = round(2 * window_length2 / 3);
soundsc(signal_y, coef2 * fs);
```



Příklady analýzy a syntézy LPC

- **Příklad 7.10: „Cross-Synthesis“ efekt**
  - MATLAB:

```
cross_synth = filter(1, A1(:, k), EE1(k) / EE2(k) * err2(:, k));
```