

# LPC

Jan Černocký, FIT VUT Brno

Při modelování tvorby řeči metodou LPC vycházíme z toho, že buzení prochází lineárním filtrem  $H(z) = \frac{G}{A(z)}$ , kde  $A(z)$  je polynom  $P$ -tého řádu:

$$A = 1 + a_1 z^{-1} + \dots + a_P z^{-P}, \quad (1)$$

a  $G$  je **gain** tohoto filtru.

Koeficienty filtru a jeho gain lze vypočítat z autokorelačních koeficientů. V Matlabu to za Vás udělá funkce `lpc`. **POZOR !!!** Definice gainu vypočítaného touto funkcí není shodná s přednáškami (Matlab označuje “gainem” druhou odmocninu *nenormované* energie chybového signálu). Do souladu s přednáškami lze Matlabovský gain uvést tímto vztahem:

$$G = \frac{G_{Matlab}}{\sqrt{l_{ram}}}, \quad (2)$$

kde  $l_{ram}$  je délka rámce.

Z koeficientů polynomu  $A(z)$  a gainu lze odhadnout *spektrální hustotu výkonu* pro daný úsek řečového signálu:

$$\hat{G}_{LPC}(f) = \left| \frac{G}{A(z)} \right|_{z=e^{j2\pi f}}, \quad (3)$$

kde  $f$  je normalizovaná frekvence  $f = F/F_s$ . V Matlabu můžete pro výpočet tohoto výrazu s výhodou využít funkce `freqz` (nezapomeňte na absolutní hodnotu a druhou mocninu).

## Matlab

Pro zkoumání metody LPC použijte známý signál `test.116`. Před analýzou LPC signál rozdělte na rámce o délce 30 ms (240 vzorků) s překrytím 20 ms (160 vzorků). Frame-shift bude tedy 10 ms a dostaneme 100 rámců za sekundu.

- Vyberte jeden znělý, např. č. 12. Uložte jej do vektoru `x`.
- Pomocí funkce `lpc` proveďte LPC analýzu:  
$$[a, g] = \text{lpc}(x, 10);$$
- Odhadněte spektrální hustotu výkonu z parametrů LPC a zobrazte její logaritmus  
pomůcka: `Glpc=10*log(abs(freqz(g, a)).^2);`
- Srovnajte s odhadem spektrální hustoty výkonu pomocí FFT.  
pomůcka: `Gfft=10*log(abs(fft(x)).^2 / 200);`

## C

Implementujte výpočet koeficientů LPC pro každý rámec pomocí algoritmu Levinsona-Durbina. Jeho vstupem je sada autokorelačních koeficientů, funkci pro jejich výpočet byste už měli mít k dispozici. Vlastní L-D je definován rovnicemi (22)–(26) v přednášce o LPC.

Výsledky uložte opět do binárního souboru, neukládejte nultý koeficient LPC, který je podle teorie vždy jedna.

## Instrukce a pomůcky

1. Pro výpočet LPC koeficientů použijete SW tvořený na minulém cvičení, jen dopíšete jednu funkci.
2. Počítejte  $P = 10$  koeficientů LPC.

- Ukládat budete pouze LPC-koefficienty  $a_1 \dots a_P$ , což by Vás mohlo vést k tomu, abyste pole pro výpočet LPC koeficientů udělali pouze o délce  $P$  a v L-D algoritmu zmenšovali všechny indexy. **Nedělejte to ;-)** Nadefinujte si raději pole o velikosti  $P + 1$ , pak budete moci algoritmus L-D přímo z rovnic přepisovat do kódu v C. Až budete hotoví, vypusťte na výstup pouze koeficienty  $a_1 \dots a_P$ , např takto:  
`memcpy (params, A + 1, LPC_COEFFS * sizeof(float));`
- Z popisu algoritmu v přednášce to není moc zřejmé ... vnější cyklus pro  $i$  musí probíhat od 1 do  $P$ .
- Nezapomeňte na alokaci paměti pro všechny typy koeficientů :-)

## Kontrola

Při vlastním programování funkcí v C se dá dopustit spousty chyb, proto je třeba mnoho kontrol:

- Přidejte do mainu Vašeho programu výpis parametrů na obrazovku. Autokorelační koeficienty by měly být v rozsahu cca od -240 do 240 (uvědomte si, že násobíte čísla menší než jedna a pak jich maximálně 200) sčítáte. Koeficienty LPC by měly být v rozsahu od cca -10 do +10.
- Uložený soubor načtete do Matlabu, srovnajte koeficienty s těmi z Matlabu, vypočítanými pomocí funkce `lpc`. Mohou se mírně lišit, ale až na 3.-4. místě za desetinnou čárkou.

```
% cteni vysledku - lpc
ff = fopen('/tmp/lpc','r');
A = fread (ff, [10 inf], 'float');
fclose (ff);
% vybereme si pro konrolu napr. 12. frame:
a = A(:,12);
% LPC matlab - 12. ramec
am = lpc(mm(:,12),10);
% ... srovnani ...
a
am
```

## 1 Když to vše pěkně půjde ... LPC-spektrogram

Vytvořte v Matlabu funkci pro zobrazení LPC-spektrogramu. Můžete s výhodou využít funkce `freqz(1,a)` (nezapomeňte zobrazovat absolutní hodnotu nebo absolutní hodnotu na druhou v *logaritmu*) a `imagesc`, kterou jste viděli na minulém cviku při hraní s FFT.