

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.L16` nebo si nahrajte signál, kde bude hláska ‘e’. Před analýzou LPC signál rozdělte na rámce o délce 25 ms (200 vzorků) s překrytím 15 ms (120 vzorků). Frame-shift bude tedy 10 ms a dostaneme 100 rámců za sekundu.

- Vyberte jeden znělý. Pokud použijete `test.l16`, doporučuji 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++

Tato část je přípravou na kódování řeči. Použijte vzorkovací frekvenci $F_s = 8000$ Hz a stejné parametry rámců jako v Matlabu – bude tedy třeba korigovat hodnoty v souboru `config.h`. Výsledné koeficienty budou uloženy do binárního souboru s následující strukturou. Pro každý rámeček:

- 10 čísel `float` (4 byty) pro uložení parametrů LPC: $a_1 \dots a_{10}$.
- 1 číslo `float` (4 byty) pro energii rámce.
- 1 číslo `float` (4 byty) pro periodu základního tónu.

V tomto cvičení budeme plnit pouze koeficienty LPC, připravte si ale strukturu už i pro ostatní parametry, zapisujte do ní pro každý rámeček nulu.

Úkoly

Definujte třídu `Lpc` (děděním z `Frames`), která bude pro každý rámeček počítat LPC koeficienty. Budete muset napsat následující funkce:

1. pro výpočet 11-ti autokorelačních koeficientů $R(0) \dots R(10)$.
2. pro algoritmus Levinsona-Durbina pro převod autokorelačních koeficientů na 10 LPC koeficientů $a_1 \dots a_{10}$.
3. Nezapomeňte na alokaci paměti pro všechny typy koeficientů :-)

Kontrola

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

1. Přidejte do mainu Vašeho programu výpis parametrů na obrazovku. Autokorelační koeficienty by měly být v rozsahu cca od -200 do 200 (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.
2. Uložený soubor (do mikrofonu říkejte 'eeee') načtěte do Matlabu, z koeficientů LPC odhadněte spektrální hustotu výkonu a podívejte se na ni. Musí vypadat "inteligentně", tzn. měli byste vidět podobné formanty jako v "jen-matlabové" části.

```
ff = fopen ('file_with_params.bin', 'r');
aux = fread (ff, [12 inf], 'float'); % nacte vse do aux
% toto cteni mozna skonci s chybou, pokud jste
% program ukoncili v okamziku, kdy nebyly dopsany parametry
% ramce, ale to by nemelo vadit.

A = aux(1:10,:); % vyber koeficientu filtru
ene = aux(11,:); % vyber energie - zatim nuly...
lag = aux(12,:); % vyber zakl. tonu - zatim nuly...
fclose (ff);

% vyber jednoho ramce - cislo nnn si musite zvolit
a = A(:,nnn);
% pozor, pro funkci freqz je potreba doplnit koeficient a0=1 !
a = [1; a];
Glp=10*log(abs(freqz(1, a)).^2); % gain nas ted nezajima, dosazujeme 1
... zobrazeni ...
```