Filip Kuś | NTwK gr. 9:45 | 24.04.2024

Fonoskopia

Sprawozdanie z laboratorium 7

Analiza cech sygnału mowy pod kątem fonoskopii. Formanty

W ramach zadania należało dla nagrań samogłosek wyznaczyć ich obwiednię widmową oraz wskazać F1 i F2, a wyniki zweryfikować za pomocą lpc().

Rozpocząłem od wczytania plików audio:

[y1, fs] = audioread("17\_a\_norm.wav");

[y2] = audioread("17\_e\_norm.wav");

[y3] = audioread("17\_i\_norm.wav");

[y4] = audioread("17\_u\_norm.wav");

Nim przeszedłem do zasadniczej części zadania, dokonałem wstępnego przetworzenia sygnałów, by polepszyć ich jakość i pracować z najistotniejszymi informacjami. Najpierw odjąłem od sygnałów ich składowe stałe:

y1 = y1 - mean(y1);

y2 = y2 - mean(y2);

y3 = y3 - mean(y3);

y4 = y4 - mean(y4);

Następnie zastosowałem dla sygnałów okno Hamminga:

y1 = y1 .\* hamming(length(y1));

y2 = y2 .\* hamming(length(y2));

y3 = y3 .\* hamming(length(y3));

y4 = y4 .\* hamming(length(y4));

W końcu dokonałem preemfazy:

h = [1 -0.95];

y1 = filter(h, 1, y1(:,1));

y2 = filter(h, 1, y2(:,1));

y3 = filter(h, 1, y3(:,1));

y4 = filter(h, 1, y4(:,1));

By móc narysować obwiednię sygnału, należało przejść do dziedziny częstotliwości. W tym celu wykonałem N-punktową szybką transformację Fouriera:

N = 8192;

df = fs/N;

f = (0:N-1)\*df;

Y1 = fft(y1, N);

Y2 = fft(y2, N);

Y3 = fft(y3, N);

Y4 = fft(y4, N);

Aby narysować obwiednie widm, należało skorzystać z funkcji envelope(). Kluczowe znaczenie ma w nim parametr np – im jest mniejszy, tym dopasowanie bardziej dokładne. Celem jednak było w pewnym sensie uogólnienie przebiegu, by móc łatwiej wskazać lokalne ekstrema stanowiące formanty. Stąd np nie mogło być zbyt małe, ale również zbyt wysokie, by przebieg nie został spłaszczony. Metodą prób i błędów arbitralnie ustaliłem wartości na podane poniżej:

np1 = 38;

np2 = 35;

np3 = 15;

np4 = 15;

[up1] = envelope(abs(Y1), np1, 'peak');

[up2] = envelope(abs(Y2), np2, 'peak');

[up3] = envelope(abs(Y3), np3, 'peak');

[up4] = envelope(abs(Y4), np4, 'peak');

Widmo sygnałów wraz z ich obwiednią prezentują się następująco:

Obraz zawierający tekst, linia, diagram, Wykres

Opis wygenerowany automatycznie

Na wykresach ręcznie zaznaczyłem znaczące w kontekście zadania (czyli te, których argumenty mieszczą się w podanych w instrukcji do laboratorium zakresów częstotliwości formantowych) ekstrema lokalne:

Obraz zawierający tekst, diagram, linia, zrzut ekranu

Opis wygenerowany automatycznie

Częstotliwości im odpowiadające porównałem z tymi modelowymi. Zielonym kolorem zaznaczyłem wyniki, które mieszczą się w podanych w tabeli po lewej zakresach, czerwonym zaś te, które są poza zakresem. Zasadniczo są one zgodne – nieduża rozbieżność wystąpiła tylko w jednym przypadku.

|  |  |
| --- | --- |
| **F1 [Hz]** | **F2 [Hz]** |
| **Otrzymane wyniki** | |
| **630** | **1168** |
| **630** | **1529** |
| **231** | **2299** |
| **328** | **754** |

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Do weryfikacji wyników w ramach podpunktu b) utworzyłem pomocniczą funkcję:

function [formants] = formants\_lpc(signal, fs)

A = lpc(signal, 6); % oczekuję 2 formantów, zatem zgodnie z dokumentacją 2\*2+2=6;

rts = roots(A);

rts = rts(imag(rts)>=0);

angz = atan2(imag(rts),real(rts));

[frqs,indices] = sort(angz.\*(fs/(2\*pi)));

bw = -1/2\*(fs/(2\*pi))\*log(abs(rts(indices)));

formants = zeros(length(frqs));

nn = 1;

for kk = 1:length(frqs)

if (frqs(kk) > 90 && bw(kk) < 400)

formants(nn) = frqs(kk);

nn = nn+1;

end

end

Wywołałem ją w głównym pliku:

formants1 = formants\_lpc(y1, fs);

formants2 = formants\_lpc(y2, fs);

formants3 = formants\_lpc(y3, fs);

formants4 = formants\_lpc(y4, fs);

Niestety, mimo wzorowania się na dokumentacji MATLABa (artykuł *estymacja formantów na podstawie współczynników LPC*), nie udało mi się potwierdzić wyznaczonych w punkcie a) wyników. Po ustawieniu parametru model order na 6 (wg dokumentacji *model order = oczekiwana liczba formantów \* 2 + 2*), macierze formantów odpowiednich głosek były puste lub zawierały jeden wynik, w dodatku niekorespondujący z oczekiwanym.

W kolejnym podejściu próbowałem jeszcze apriorycznie ustalić model order, by jego wartość pokrywała się z tymi oczekiwanymi. Wyniki jednak znów nie były zadowalające. Ponadto nie widzę uzasadnienia, by w tym konkretnym wypadku dobierać parametr metodą prób i błędów, ponieważ lpc() ma służyć za potwierdzenie przeprowadzonej uprzednio analizy. Dobieranie zaś wartości pod tezę zdaje się nie mieć sensu i być wbrew celowi postawionego zadania.