Laboratorium – filtracja przebiegów czasowych (audio), zastosowanie okien kroczących, FFT

Streszczenie

Celem laboratoriów jest zapoznanie się z możliwościami analizy i ekstrakcji informacji z przebiegów czasowych, na przykładzie sygnału audio. Wszystkie ćwiczenia powinny być wykonywane na własnym nagraniu (patrz punkt 1.1), a w razie problemów z wydzieleniem odpowiednich fragmentów posiłkować się będzie można dodatkowymi nagraniami zawierającymi wybrane głoski nagrane indywidualnie.

1 Sygnał audio

1. Korzystając z mikrofonu w komputerze lub dyktafonu w smartfonie nagraj sekwencję audio "Jestem studentem informatyki".

2. Wczytaj nagrany plik dźwiękowy do zmiennej *s*. Odczytaj/oblicz podstawowe parametry sygnału (czas trwania, częstotliwość próbkowania, rozdzielczość bitowa, liczba kanałów). Przydatne funkcje:

   - import sounddevice as sd
   - import soundfile as sf
   - s, fs = sf.read('audio.wav', dtype='float32') - *s* to tablica zawierająca wartości próbek sygnału, każda z jej kolumn zawiera osobny kanał wartości (lewy i prawy), chyba że sygnał jest monofoniczny; *fs* - to częstotliwość próbkowania, określa liczbę próbek na sekundę;
   - sd.play(s, fs) - rozpoczęcie odtwarzania;
   - status = sd.wait() - odczekanie z wykonaniem kolejnych instrukcji do momentu, aż dźwięk skończy się odtwarzać.

3. Wyświetlić sygnał tak, aby na osi poziomej znajdowała się jednostka czasu [ms] (konieczność przeliczenia zakresu). Jeżeli po wczytaniu sygnał nie jest znormalizowany, należy przeprowadzić normalizację wartości do zakresu [-1;1]. W przypadku sygnału stereo wykorzystaj dowolny z kanałów: lewy lub prawy (do końca instrukcji pracujemy wyłącznie na jednym kanale, jeżeli nasz sygnał jest stereofoniczny).

4. Sprawdź, czy dynamika sygnału jest odpowiednia? Czy zakres amplitudy jest odpowiednio wykorzystany? Czy nie występuje przesterowanie? Jaka jest amplituda szumu na początku i na końcu nagrania? Czy szum ma charakter losowy?

5. Jeżeli występują problemy warto powtórzyć nagranie, zadbać o ciszę w pomieszczeniu, ewentualnie zmienić mikrofon. Poprawne nagranie dźwięku ułatwia też odpowiedni program, z darmowych przykładowo Audacity.

2 Zastosowanie okien kroczących

1. Podzielić sygnał na ramki (okna) długości 10 ms i obliczyć dla każdej ramki dwie statystyki – funkcję energii E oraz funkcję przejść przez zero Z:

$$E_j = \sum_{i=1}^{n} (s_i)^2$$

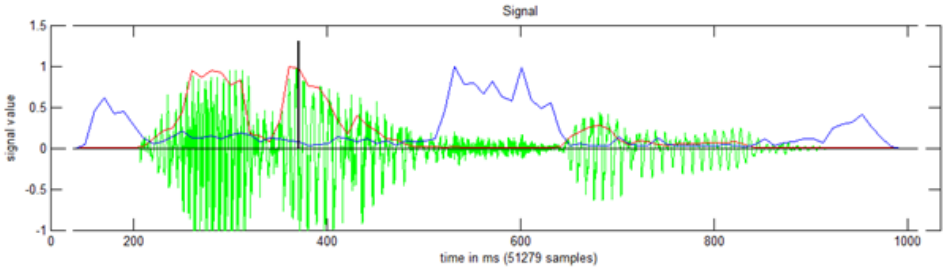
$$Z_j = \sum_{i=1}^{n-1} z_i$$

gdzie: $\begin{cases} z_i = 0 \; jeżeli \; s_i s_{i+1} \geqslant 0 \\ z_i = 1 \; jeżeli \; s_i s_{i+1} < 0 \end{cases}$

s – wektor sygnału, j – numer ramki, n – długość ramki w próbkach.
Wskazówka 1: W zależności od wartości parametru częstotliwości próbkowania, dziesięciu milisekundom odpowiada różna liczba próbek.

Wskazówka 2: Dla każdej ramki, składającej się z n próbek, wyznaczamy dwie liczby (energię i liczbę przejść przez 0).

2. Uzyskane funkcje (wektory) energii E oraz przejść przez zero Z należy znormalizować. Dokonać wizualizacji tych funkcji. Przykład poniżej: wykresy funkcji E (czerwony) i Z (niebieski) zostały naniesione na badany sygnał.



3. Na co wskazują maksima oraz minima obliczonych funkcji? Czy można ich użyć do podziału sygnału na segmenty dźwięczne i bezdźwięczne?

Jakie typy głosek można rozdzielić automatycznie na podstawie funkcji Z i E? Czy istnieją sekwencje głosek nierozdzielnych?

4. Zbadać jaki wpływ na segmentowanie sygnału ma długość okna. Sprawdzić i pokazać wyniki dla okien równych: 5ms, 20ms i 50ms. Jakie elementy nagrania zostają rozdzielone przy bardzo długim, a jakie przy bardzo krótkim oknie?

5. Podzielić sygnał na nakładające się ramki np. w stopniu 50% i przeprowadzić powyższe analizy. Jak wpływa nakładanie ramek na precyzję granic segmentów?

3 Analiza częstotliwościowa

1. Zlokalizować (manualnie) fragment nagrania stanowiący samogłoskę ustną i skopiować jej fragment długości 2048 próbek do nowej zmiennej.

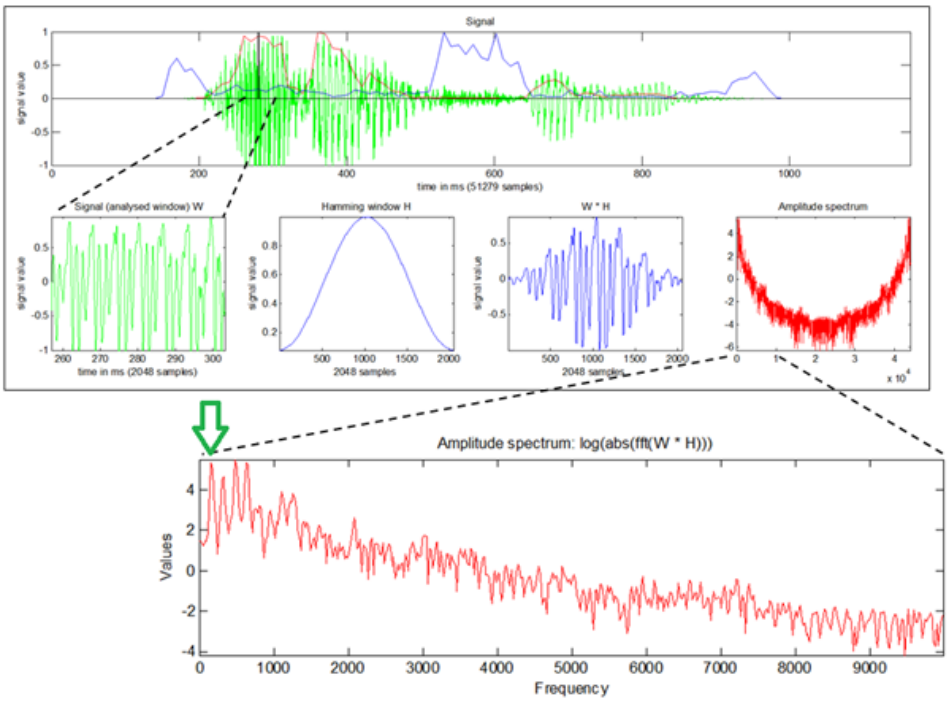
2. Dokonać maskowania sygnału oknem Hamminga (wykorzystać wbudowaną funkcję numpy.hamming).

3. Obliczyć logarytmiczne widmo amplitudowe dla badanego (zamaskowanego) okna. Przydatne funkcje: numpy.log oraz numpy.abs. Transformatę Fouriera wyznaczamy poleceniem:

   - yf = scipy.fftpack.fft(okno_zamaskowane);

   Pamiętamy o zaimportowaniu biblioteki: import scipy.fftpack.

4. Wyświetlić widmo w taki sposób, aby na osi poziomej znajdowała się jednostka częstotliwości Hz w zakresie 0 - 10000 Hz. Schematyczne zobrazowanie wykonanych do tej pory zadań ukazane zostało na poniższym rysunku.



5. Z wykresu odczytać F_0, czyli częstotliwość podstawową (częstotliwość drgania strun głosowych) na podstawie pierwszego dominującego maksimum w przebiegu widma. Na rysunku przedstawionym powyżej, częstotliwość podstawowa zaznaczona została zieloną strzałką.

   Wskazówka: Seria maksimów występujących w równych odległościach to struktura harmoniczna, należy zatem odczytać częstotliwość pierwszej harmonicznej.

6. Sprawdzić, które z badanych głosek posiadają strukturę harmoniczną i można odczytać z nich F_0 (częstotliwość drgania strun głosowych)? Czym różni się wizualnie widmo głosek dźwięcznych od bezdźwięcznych? Czy widma samogłosek różnią się między sobą? W jaki sposób? Czy częstotliwość F_0 jest stała dla tego samego nagrania, ale różnych głosek?

4 Rozpoznawanie samogłosek ustnych /a, e, i, o u, y/

1. Zlokalizować fragment nagrania stanowiący samogłoskę ustną i skopiować jej fragment długości 2048 próbek do zmiennej **okno**.

2. Wyznaczyć dla tego fragmentu p=20 współczynników liniowego filtra LPC. Wykorzystać należy gotową funkcję obliczającą Liniowe Kodowanie Predykcyjne (LPC) z pakietu librosa:

- a = librosa.lpc(okno, p);

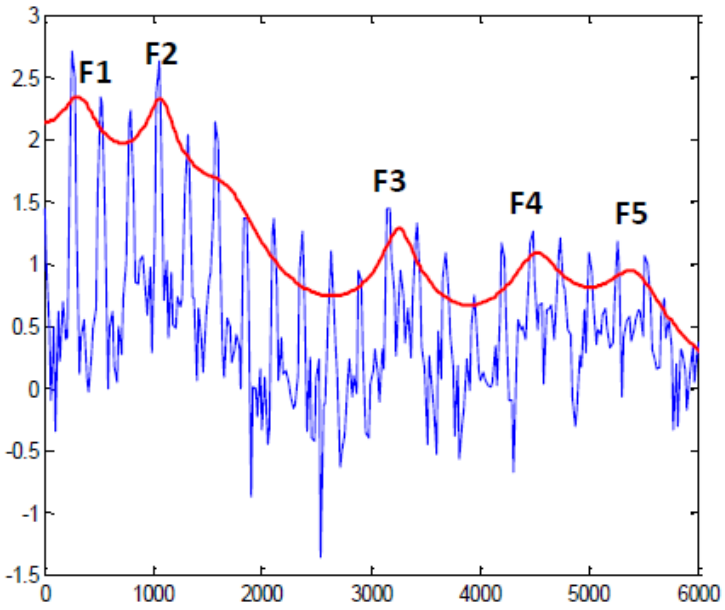
Szczegóły: https://librosa.org/doc/main/generated/librosa.lpc.html

Pamiętamy o zaimportowaniu biblioteki: import librosa; i ewentualnie jej wcześniejszej instalacji.

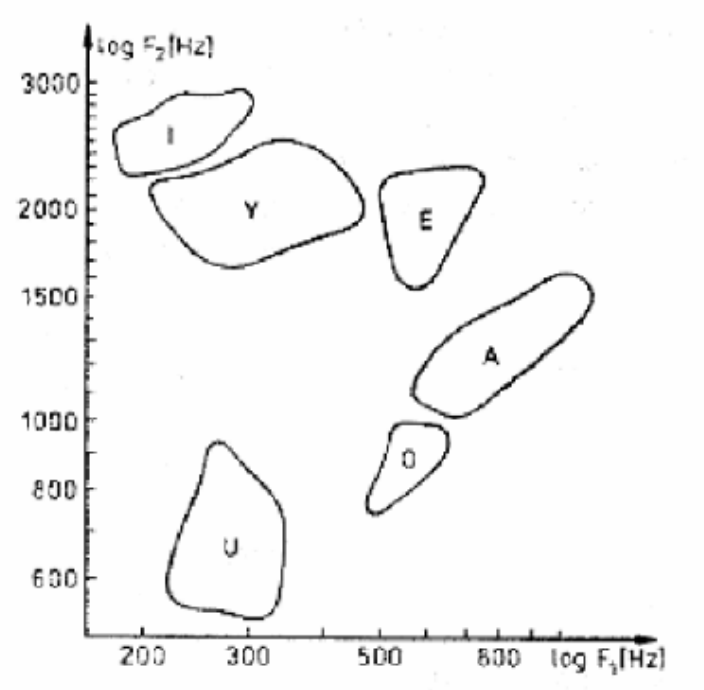
3. Dowiedzieć się i krótko opisać, co to jest i do czego służy Liniowe Kodowanie Predykcyjne.

4. Uzupełnić otrzymany wektor a długości p, zerami do długości okna sygnału (2048).

5. Wyznaczyć wygładzone widmo amplitudowe na bazie wektora a: widmoLPC = log(abs(fft(a))). Odbić otrzymane wygładzone widmo w poziomie (pomnożyć przez -1). Nałożyć wykres na właściwe widmo amplitudowe fragmentu sygnału tak, aby oba znalazły się na podobnej wysokości na osi y (prawdopodobnie będzie konieczność przeskalowania lub/i przesunięcia). Należy zadbać o prawidłową oś częstotliwości [Hz]! Przykład na poniższym rysunku.



Widmo apmplitudowe pokazane jest na niebiesko. Wygładzone widmo amplitudowe jest w kolorze czerwonym. Lokalne maksima na wygładzonym widmie to FORMANTY, oznaczane kolejno F1, F2... Mogą one służyć do automatycznego rozpoznawania samogłosek. Wykorzystać do tego można wykres pierwszych dwóch formantów dla polskich samogłosek, zgodnie z poniższym rysunkiem.



6. Wyznacz dla kilku samogłosek częstotliwości F1 i F2. Odnieś wyznaczone wartości do wzorcowego wykresu. Jakie samogłoski udało się właściwie rozpoznać? Jak wpływa ilość wybranych współczynników filtra p na proces rozpoznawania? Przetestować kilka możliwości. W przypadku, gdy istnieje problem ekstrakcji niezakłóconych samogłosek z nagrania należy dokonać kolejnego nagrania audio, tym razem zawierającego izolowane samogłoski: /a,e,i,o,u,y/.