Analiza i Przetwarzanie Dźwięku - Sprawozdanie z projektu 2.

Szymon Tomulewicz

17 VI 2020

Spis treści

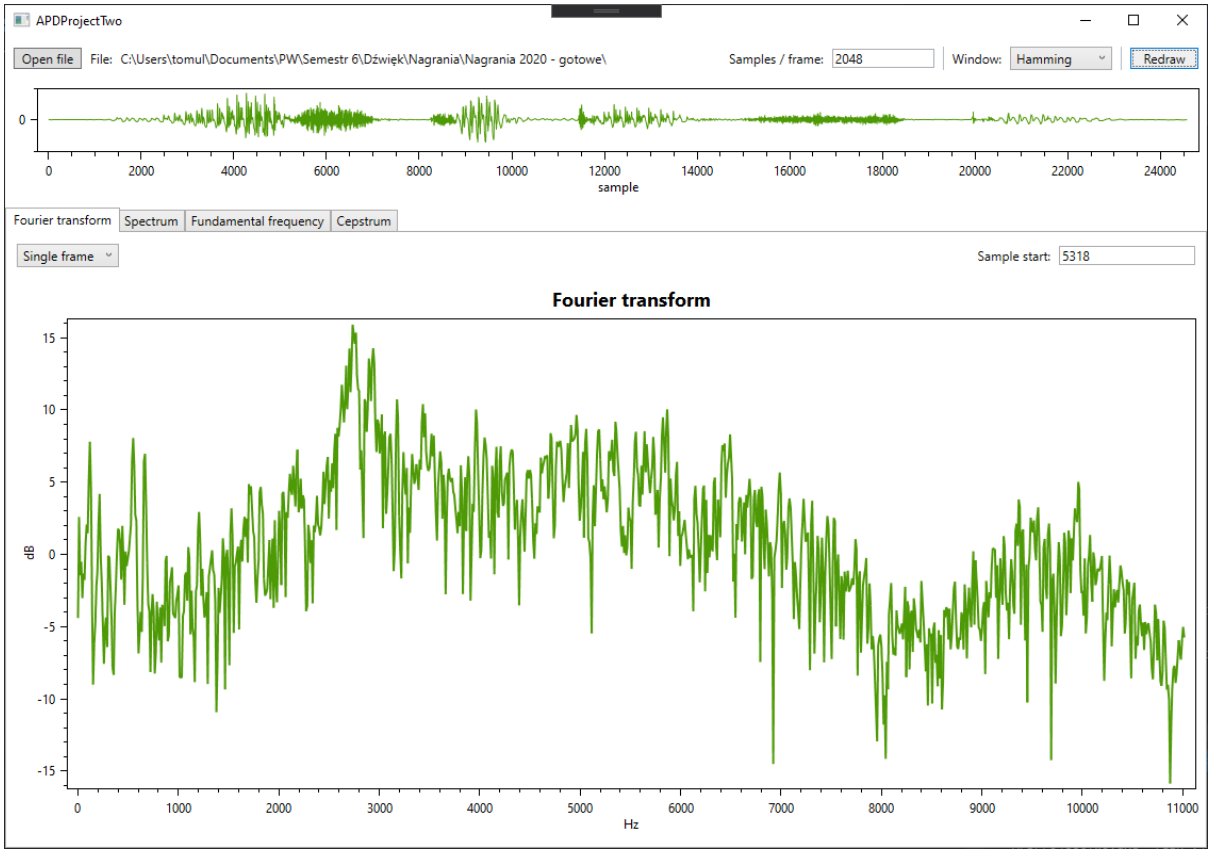
1 Wprowadzenie

Celem projektu było stworzenie aplikacji okienkowej, umożliwiającej analizę częstotliwościową sygnałów dźwiękowych. Aplikacja umożliwia rysowanie wykresu sygnału w dziedzinie czasu, rysowanie widma częstotliwościowego, rysowanie spektrogramu oraz rysowanie wykresu częstotliwości krtaniowej.

1.1 Aplikacja

Aplikacja została wykonana w języku C#, przy użyciu biblioteki *NAudio* do przetwarzania i analizy plików dźwiękowych, bibliotek *WPF* i *OxyPlot* do stworzenia interfejsu oraz wyświetlania parametrów, a także biblioteki *Math.NET Numerics* do celów obliczeniowych.

W górnej części interfejsu znajduje się menu umożliwiające wybór pliku do analizy, długości ramki (w próbkach) oraz funkcji okienkowej. Wyświetlany jest także przebieg czasowy wczytanego sygnału. Dolna część interfejsu składa się z zakładek prezentujących różne wykresy - transformatę Fouriera, spektrogram, wykres częstotliwości krtaniowej od czasu, a także cepstrum dla wybranej ramki. Na poniższym rysunku (Rys. 1) znajduje się przykładowy stan apliakcji po załadowaniu pliku *wav*.



Rysunek 1: Aplikacja po załadowaniu pliku i wybraniu zakładki *Fourier transform*

Wybór rozmiaru ramki i funkcji okienkowej wpływa na wszystkie wykresy, natomiast wybór współczynnika nakładania się ramek wpływa na spektrogram i wykres częstotliwości krtaniowej. Ich wpływ opisany jest w kolejnych sekcjach.

2 Obliczane wartości i metody

Wszystkie obliczenia w aplikacji, które są wykonywane na próbkach, używają typów zmiennoprzecinkowych. Parametry opisane w kolejnych sekcjach są obliczane, a następnie zapisywane i wyświetlane w formie wykresów. N występujące we wzorach oznacza zwykle długość ramki w próbkach.

Na początku wczytywane są wszystkie próbki z pliku, ale analizowane są wyłącznie dla jednego kanału:

```
// Add samples from single channel to FFT aggregator
int limit = fftLength * channels + offset;
for (int n = offset; n < limit; n += channels)
{
    aggregator.Add(samples[n]);
}
```

2.1 Widmo częstotliwości

Do wyznaczenia widma częstotliwości aplikacja używa FFT (z biblioteki *Math.NET Numerics*). Transformata może być wykonana dla całego sygnału. Jeśli ilość próbek w całym sygnale nie mieści się w najbliższej potędze dwójki, jest "obcinana":

$$m = \lfloor log_2 N \rfloor,$$

$$N := 2^m.$$

Transformata może być również wykonana dla pojedynczej ramki z uwzględnieniem próbki od której ma się rozpoczynać.

Podczas wykonywania FFT na sygnał nakładana jest jedna z dostępnych funkcji okienkowych

- Hamminga:
$$w(n) = 0.54 - 0.46cos(2\pi\frac{n}{N}), 0 \leqslant n \leqslant N.$$
- Hanna:
$$w(n) = 0.5 * (1 - cos(2\pi\frac{n}{N}), 0 \leqslant n \leqslant N.$$
- Prostokątna:
$$w(n) = n, 0 \leqslant n \leqslant N.$$

Efektem jest wykres widma częstotliwości, którego przykład zaprezentowany jest na Rys. 1.

2.2 Spektrogram

Spektrogram zrealizowany jest poprzez narysowanie na mapie cieplnej wyników FFT liczonych dla kolejnych ramek. Czas jest na osi poziomej, częstotliwości na osi pionowej, natomiast wartość dla danej częstotliwości oznaczany jest kolorem. Kolory cieplejsze oznaczają wysokie wartości, a kolory chłodne - niskie wartości. Przykładowy spektrogram przedstawiony jest na Rys. 6.

2.3 Cepstrum

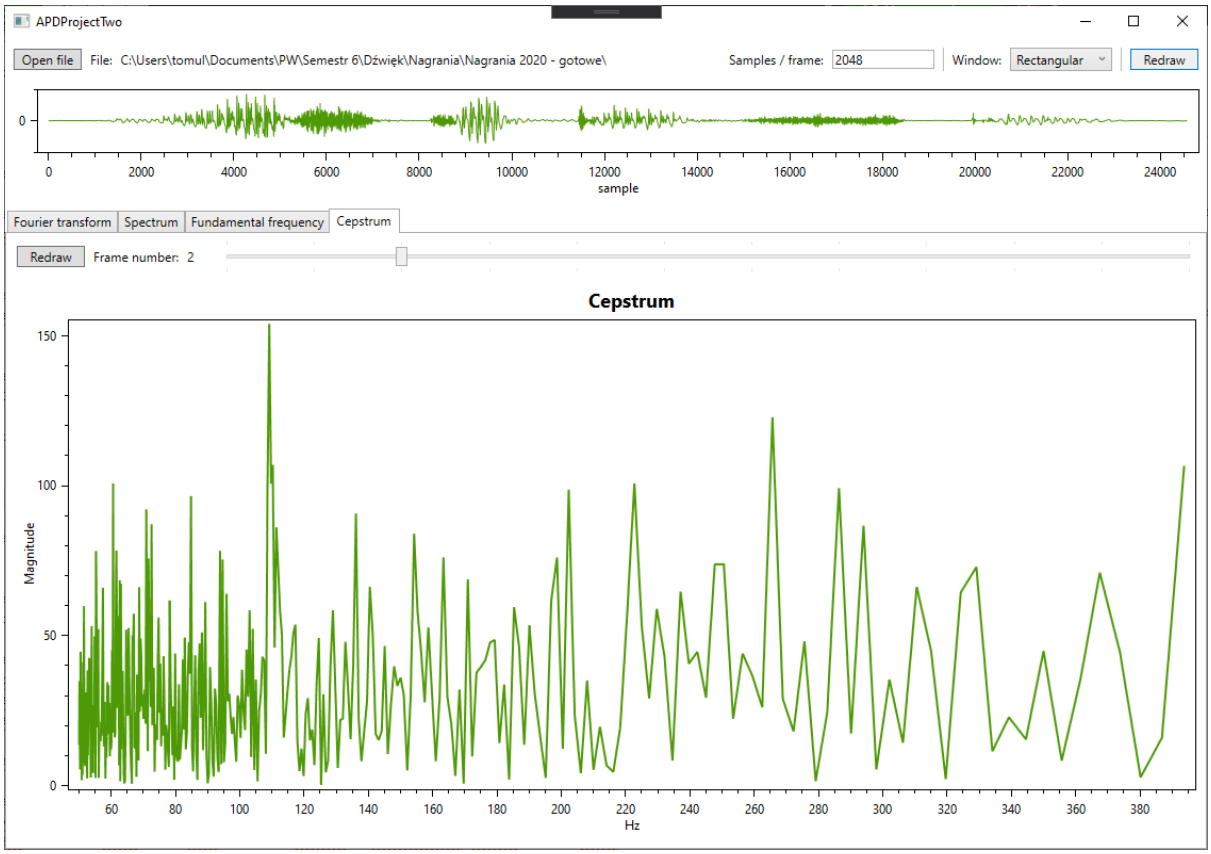
Aplikacja oblicza cepstrum dla każdej z ramek w następujący sposób:

$$C(\tau) = F^{-1}(log|F(s(t))|),$$

gdzie F jest transformatą Fouriera, a $s(t)$ wartością próbki w czasie t. Następnie w zakładce *Cepstrum* wyświetla dla wybranej ramki wykres w przedziale $[50Hz, 400Hz]$. Konwersja z τ na częstotliwość f odbywa się według wzoru

$$f = f_s/\tau,$$

gdzie f_s jest częstotliwością próbkowania. Wartość na osi rzędnych jest moduł $C(\tau)$. Przykładowy wykres cepstrum znajduje się na Rys. 2.



Rysunek 2: Lektor 201 "Błaszczykowski" - cepstrum w okolicy pierwszej samogłoski

2.4 Częstotliwość tonu podstawowego

Cepstrum przedstawiane jest w zakresie $[50Hz, 400Hz]$, ponieważ jest to przedział istotny przy znajdowaniu częstotliwości tonu podstawowego. Posiadając już obliczone cepstrum dla każdej ramki, aplikacja liczy następnie zmieniające się w czasie (jednostką czasu jest w tym przypadku ramka) przybliżenie częstotliwości f_0 według następującego wzoru:

$$f_0 = \frac{1}{\tau_{max}},$$

gdzie τ_{max} wyznaczne z równania

$$C(\tau_{max}) = \max_{\tau} C(\tau), \frac{f_s}{\tau} \in [50, 400].$$

Przykładowy wykres częstotliwości krtaniowej znajduje się na Rys. 3.

Rysunek 3: Lektor 201 "Błaszczykowski" - częstotliwość krtaniowa

3 Wyniki działania

3.1 Formanty

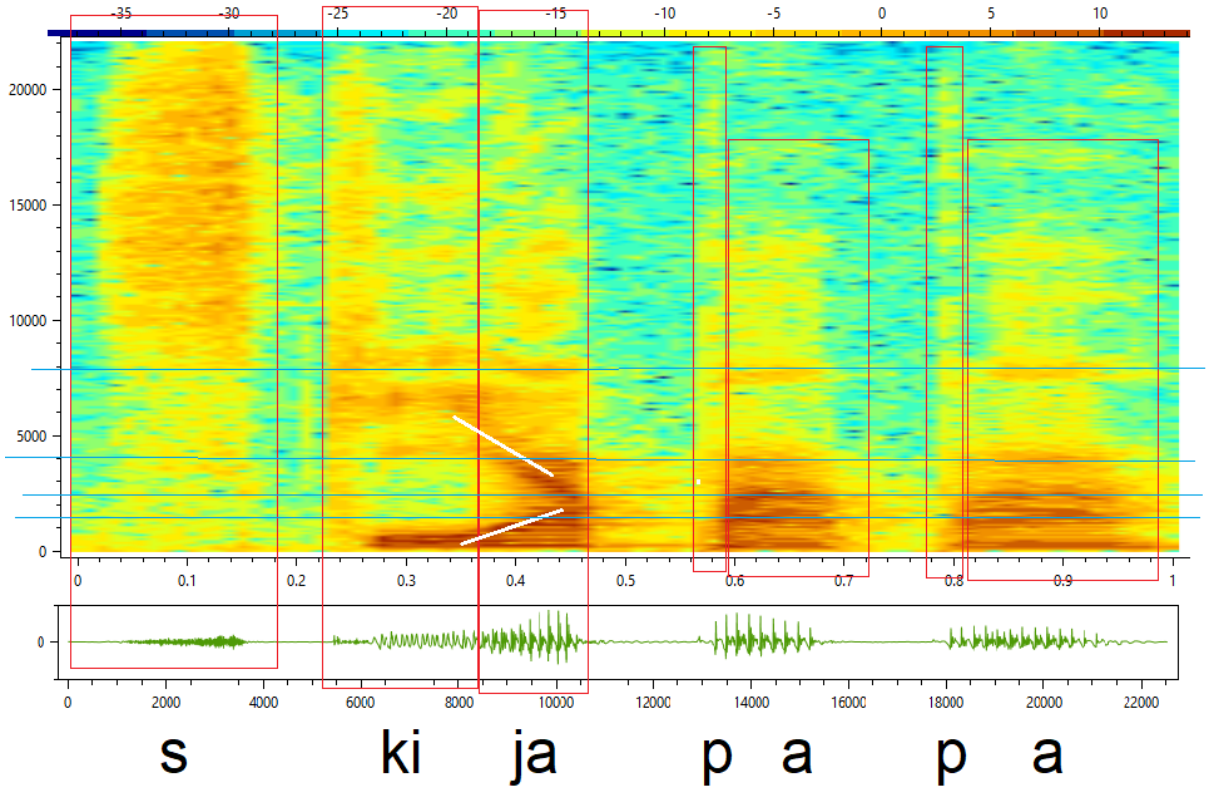
Formant – pasmo częstotliwości w dźwięku (np. głosu ludzkiego lub instrumentu muzycznego), w granicach którego wszystkie tony składowe ulegają szczególnemu wzmocnieniu. Zbiór wszystkich formantów danego dźwięku określa jego barwę [1].

Z moich obserwacji wynika, że dla pojedynczych lektorów formanty zawsze zostają w podobnej odległości od siebie w realizacjach tego samego fonemu. Na przykład głoska "a" lektora 201 zawsze ma formant F_1 1500 Hz i formant F_2 w okolicach 2300 Hz. Natomiast różnią się między lektorami. Lektor 204 z niższym głosem niż lektor 201 wykazuje niższe częstotliwości w formantach samogłosek (dla fonemu "a" F_1 to około 1300 Hz, a F_2 to około 1900Hz).

3.2 Różnice między spółgłoskami i samogłoskami

Jako przykład różnic miedzy spółgłoskami i samogłoskami wybrałem słowo “skijapapa”. Na spektrogramie (Rys. 4) doskonale przedstawiają się kolejne głoski.

- “s” charakteryzuje się dużą ilością szumu w wysokich częstotliwościach i brakiem wyraźnego tonu podstawowego,
- “ki” rozpoczyna się szumem we wszystkich częstotliwościach (głoska “k”), a nastepnie przechodzi w głoskę “i” z charakterystyczną przerwą między tonem podstawowym a wyższymi formantami,
- “ja” widoczne jest jako płynne przejście (zaznaczone na Rys. 4 białymi kreskami) pomiędzy formantami widocznymi w “i” a formantami widocznymi w “a”,
- “p” charakteryzuje się tutaj szumem, widocznym głównie w niższych częstotliwościach z racji następującego po nim “a”,
- “a” ma charakterystyczny układ formantów, który został zaznaczony na Rys. 4 niebieską poziomą linią.



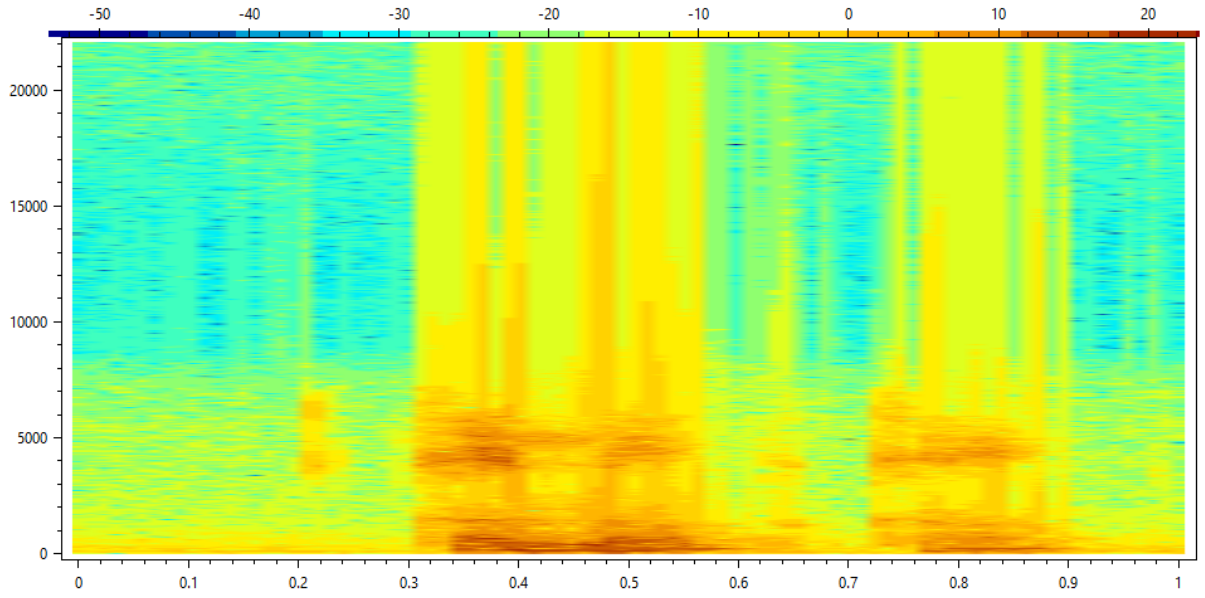
Rysunek 4: Lektor 201 “skijapapa” - analiza spektrogramu

Jak widać spółgłoski bezdźwięczne charakteryzują się dużą ilością szumu różnie rozłożoną na spektrum częstotliwości, a samogłoski - wyklarowanymi formantami, po których można je między sobą rozróżnić. Nie zawsze łatwo jest rozróżnić pojedyncze formanty, dopóki nie wiemy gdzie mniej więcej można się ich spodziewać. Jednak sam fakt, że formanty występują w danej głosce jest zwykle oczywistą wskazówką, że patrzymy

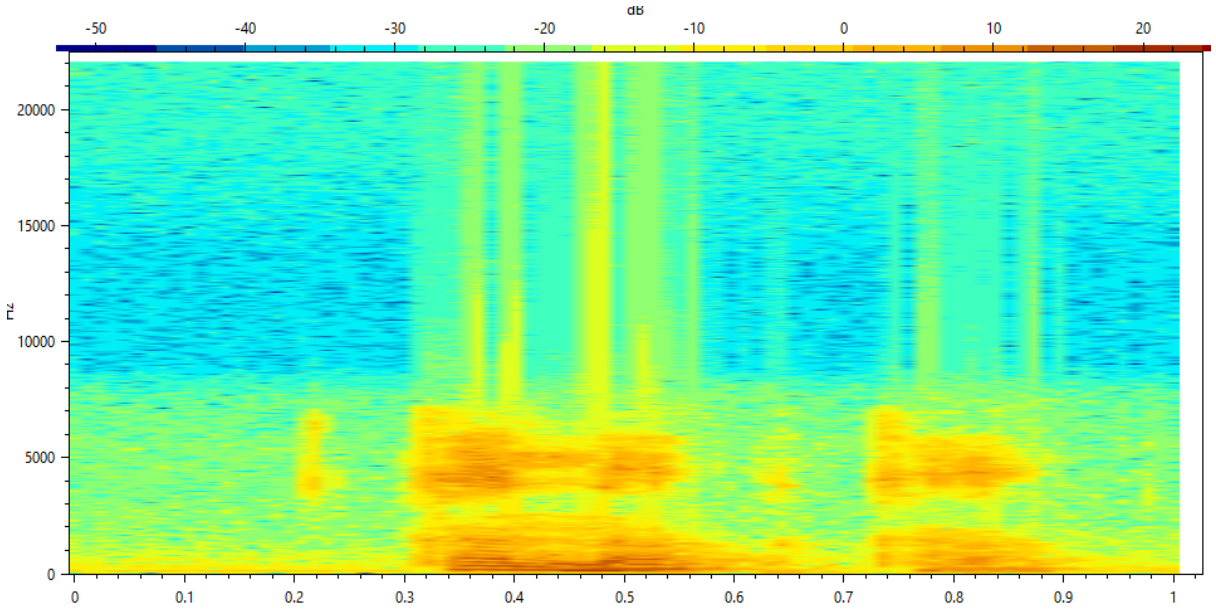
na samogłoskę. Znając "wygląd" poszczególnych głosek jesteśmy w stanie z dość dużą dokładnością odczytać ze spektrogramu wypowiadane słowo. [2]

3.3 Wpływ funkcji okienkowych

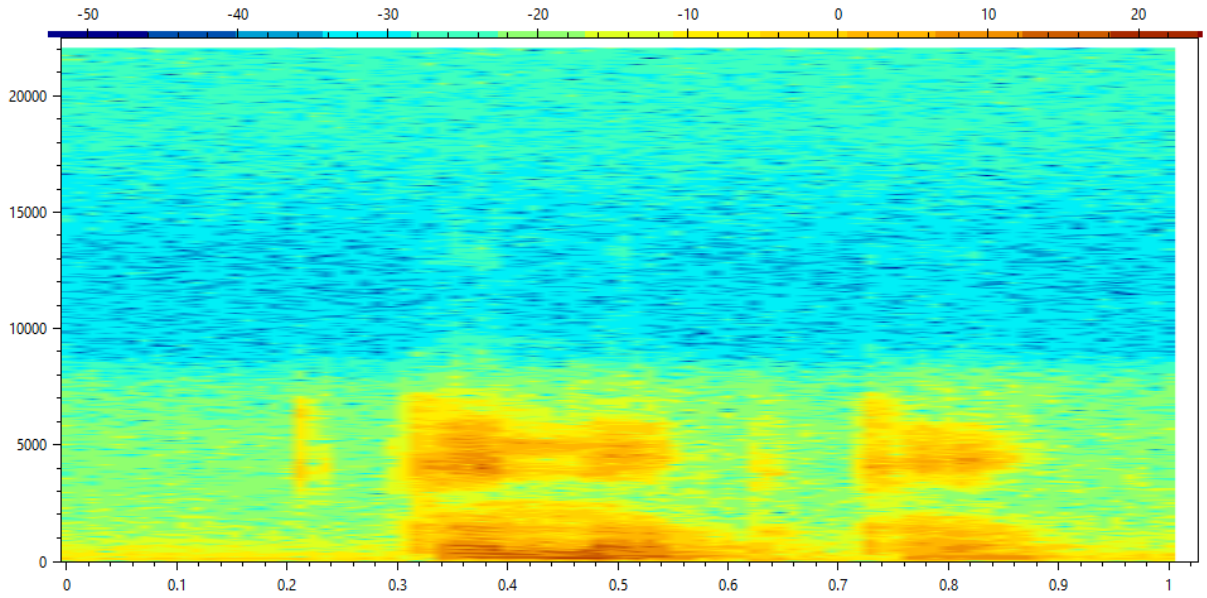
Tak jak możnaby się spodziewać, okno kwadratowe wprowadza dużo zakłóceń w wysokich częstotliwościach, natomiast okna Hamminga i Van Hanna potrafią zależnie od sygnału niwelować te zakłócenia mniej lub bardziej. W przykładzie na Rys. 5, 6, 7 okno Van Hanna pozostawia najmniej zakłóceń na spektrum.



Rysunek 5: Lektor 204 "kanapka" - okno prostokątne



Rysunek 6: Lektor 204 "kanapka" - okno Hamminga

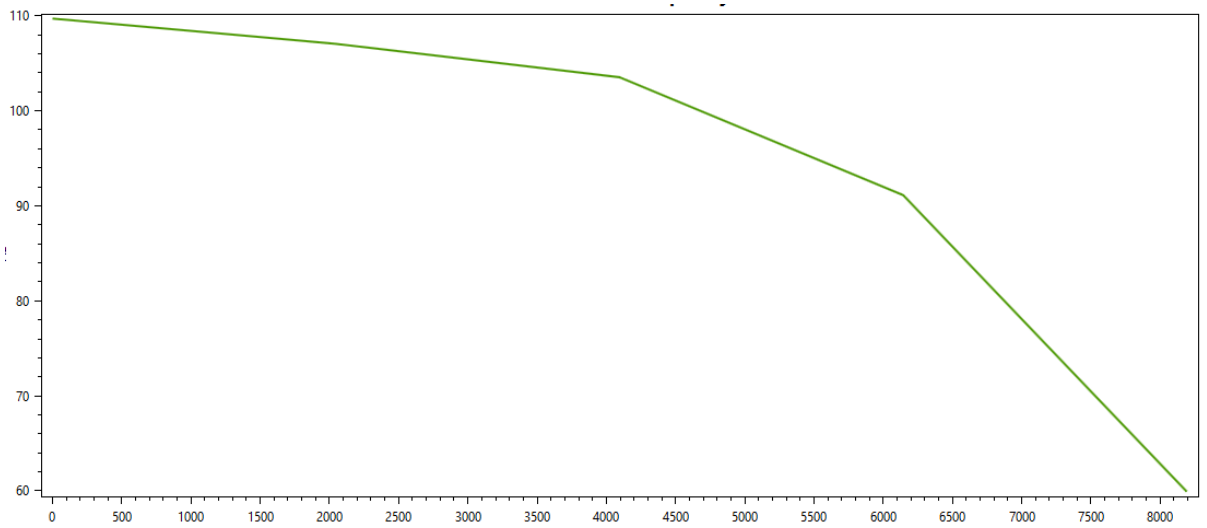


Rysunek 7: Lektor 204 “kanapka” - okno Van Hanna

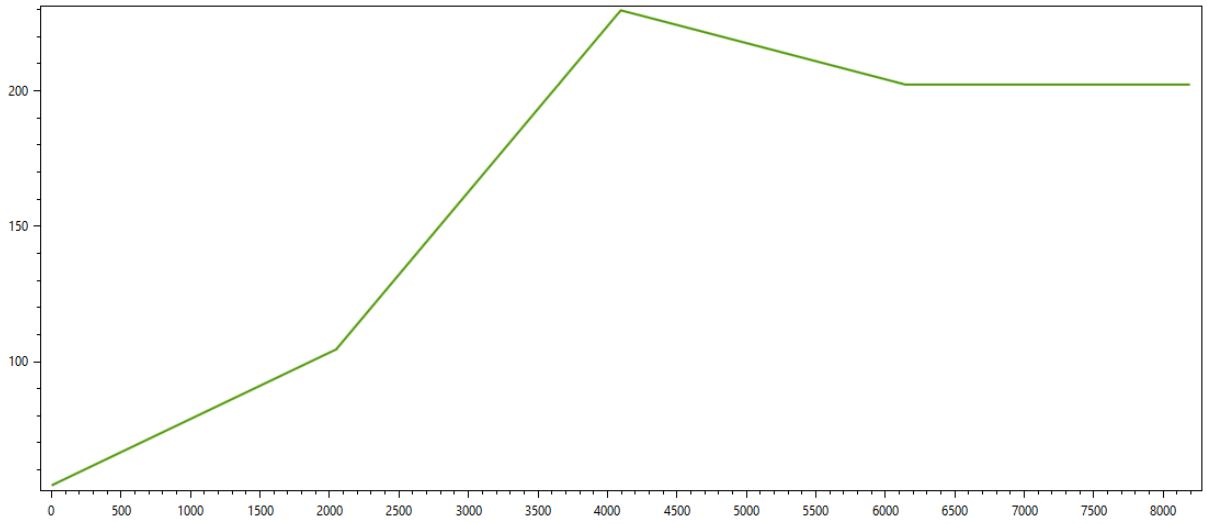
3.4 Ton podstawowy

Jako przykład działania wybrałem logotom “aba” z racji różnorodności intonacji jaką zauważyłem wśród lektorów. Lektor 201 ma na nagraniu intonację opadającą, co widać na malejącym wykresie f_0 (Rys. 8). Lektorzy 203 i 205 mają intonację rosnącą, co widać na rosnących wykresach (Rys. 9 oraz 10). Natomiast lektor 208 ma stałą intonację, co widoczne jest jako prawie płaski wykres na Rys. 11 (słowo “aba” pojawia się około 5100. próbki).

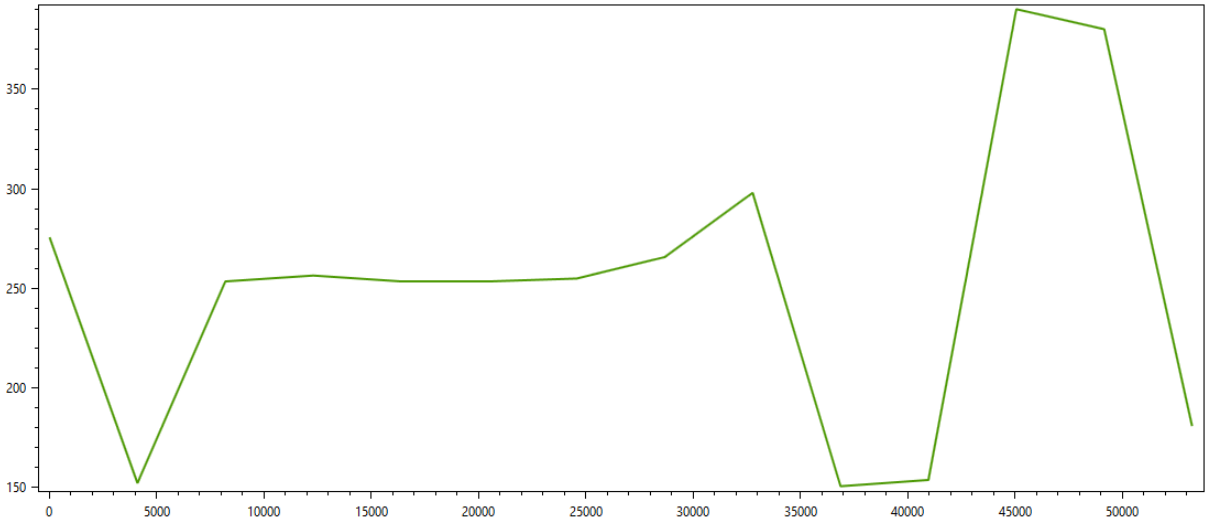
Można także zauważyć różnicę w zakresie tonów podstawowych. Lektorzy 201 i 208 (Rys. 8 i 11) to mężczyźni o niższych głosach (odpowiednio $100Hz$ i $130Hz$) niż lektorzy 203 i 205 (Rys. 9 i 10) - kobiety o wyższych głosach (odpowiednio $150 - 200Hz$ i $250 - 290Hz$).



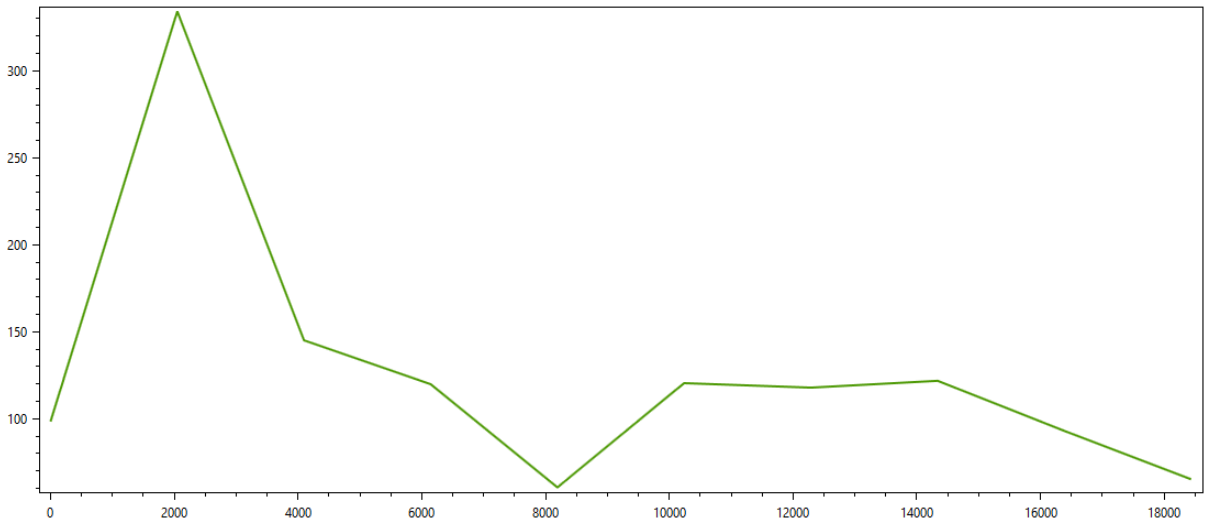
Rysunek 8: Lektor 201 “aba” - wykres częstotliwości krtaniowej



Rysunek 9: Lektor 203 “aba” - wykres częstotliwości krtaniowej



Rysunek 10: Lektor 205 “aba” - wykres częstotliwości krtaniowej



Rysunek 11: Lektor 208 “aba” - wykres częstotliwości krtaniowej

4 Wnioski

Na podstawie wyników i implementacji projektu nasuwają się następujące wnioski:

4.1 Czy metody zawsze działają dobrze?

Cepstrum wydaje się być dobrą metodą znajdowania częstotliwości krtaniowej w sygnałach dźwiękowych głosu, jednak w wypadku na przykład fal kwadratowych czy trójkątnych nie radzi sobie za dobrze (a przynajmniej moja implementacja).

4.2 Modyfikacja parametrów

Czasami, aby uzyskać miarodajne wyniki należy dokładnie manipulować parametrami. W szczególności długość ramki i funkcja okna muszą być odpowiednio dobrane. Współczynnik nakładania ramek również bywa pomocny w odsiewaniu "zaszumionych" wyników.

4.3 Wnioski natury technicznej

Nieporządne implementacje metod numerycznych takich jak transformata Fouriera potrafią wskazywać wyniki dalekie od prawdy w niektórych specyficznych sytuacjach. Na szczęscie implementecje w bibliotece *Math.NET Numerics* nie powodują takich problemów.

Literatura

[1] Golachowski S. Drobner M. *Akustyka muzyczna*. Polskie Wydawnictwo Muzyczne, Kraków, 1953.

[2] Russell K. Identifying sounds in spectrograms. `https://home.cc.umanitoba.ca/~krussll/phonetics/acoustic/spectrogram-sounds.html`.