

Katedra Biosensorów i Przetwarzania Sygnałów Biomedycznych

Programowanie Python

Projekt

Narzędzie CLI do konwersji częstotliwości próbkowania

Suchanek Kamil

PiAIB semestr 2

Spis treści

1. Cel i założenia 3

2. Wstęp teoretyczny 3

2.1. Interpolacja 3

2.2. Ekspansja sygnału 4

3. Realizacja projektu 4

4. Specyfikacja wewnętrzna 4

4.1. Wskazówki dla użytkownika 4

4.2. Załadowanie potrzebnych modułów 5

4.3. Funkcja resampleit(path, target\_frequency) 6

5. Specyfikacja zewnętrzna 7

6. Podsumowanie 8

# Cel i założenia

Celem projektu jest stworzenie narzędzia linii poleceń, służącego do konwersji częstotliwości próbkowania jednowymiarowych sygnałów bioelektrycznych.

Narzędzie umożliwi konwersję częstotliwości próbkowania z dowolnym czynnikiem poprzez wprowadzenie odpowiednich flag-argumentów oraz wskazania ścieżki do pliku z sygnałem.

# Wstęp teoretyczny

Konwersja częstotliwości próbkowania sygnału realizowana jest poprzez ekspansję i podpróbkowanie sygnału.

## Interpolacja

Wartości sygnału cyfrowego znane są w dyskretnych punktach czasu. W przypadku równomiernie próbkowanych sygnałów, odstęp między próbkami jest stały i równy okresowi próbkowania , gdzie oznacza częstotliwość próbkowania, czyli liczbę próbek przypadających na jedną sekundę sygnału. Zdarza się, że zachodzi potrzeba uzyskania innej częstotliwości próbkowania niż pierwotna , albo potrzebna jest wartość w chwili nie zdeterminowanej dyskretną próbką sygnału [1].

Interpolacja sygnału cyfrowego polega na obliczeniu wartości sygnału w dowolnym punkcie między próbkami. Należy zaznaczyć, że interpolacja nie tworzy nowych danych sygnału, powstały sygnał nie będzie identyczny z odpowiednikiem rejestrowanym z większą częstotliwością próbkowania. Wartości w dodatkowych punktach są estymowane na podstawie dostępnych [1].

## Ekspansja sygnału

Ekspansja polega na “poszerzeniu” sygnału o dodatkowe próbki, zwiększając częstotliwość próbkowania. Operacja ta przeprowadzona w઀dziedzinie czasu powoduje zwężenie się widma częstotliwościowego, objawiające się poprzez wystąpienie niskich częstotliwości.

# Realizacja projektu

Zastosowano moduł docopt do sformułowania narzędzia CLI ze skryptu Pythona. W celu przetwarzania sygnałów zastosowano wrapper do narzędzi WFDB służących do tworzenia, odczytywania, przetwarzania i zaopatrywania w adnotacje sygnałów w serwisie PhysioNet.

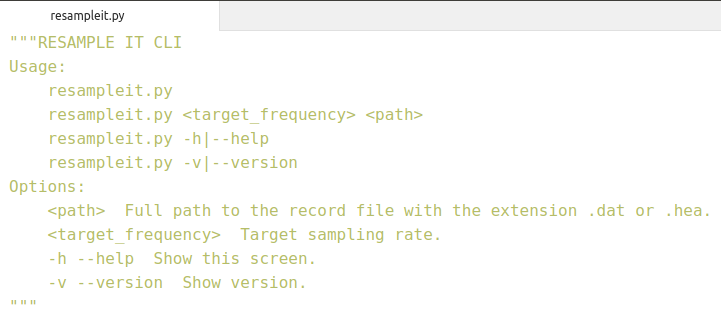
# Specyfikacja wewnętrzna

Skrypt narzędzia CLI składa się z czterech części:

* Wskazówki dla użytkownika,
* Załadowanie potrzebnych modułów,
* Funkcja spełniająca zadanie narzędzia,
* Fragment warunkujący uruchomienie narzędzia z argumentami i bez nich.

## Wskazówki dla użytkownika

Na samej górze skryptu znajduję się notatka wskazująca na prawidłowe użycie narzędzia.

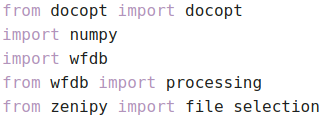


Rys. 1 - Wskazówka do wykorzystania narzędzia CLI

## Załadowanie potrzebnych modułów

Narzędzie wykorzystuje następujące moduły:

* docopt - narzędzie do formułowania narzędzi CLI,
* numpy - pakiet narzędzi numerycznych i nie tylko,
* wfdb - narzędzia wspomniane w punkcie 3 - Realizacja projektu,
* zenipy - narzędzie do interaktywnego wyboru plików i folderów, wygodne i zachowujące natywny wygląd bieżącego interfejsu.



Rys. 2 - Zależności narzędzia

## Funkcja resampleit(path, target\_frequency)

Funkcja wykonuje kolejno następujące czynności:

* usunięcie z pełnej ścieżki do pliku rozszerzenia .hea / .dat,
* wczytanie rekordu WFDB jako obiektu Python,
* dostosowanie formy tablicy do celów obliczeniowych,
* przepróbkowanie sygnału,
* stworzenie rekordu WFDB z przepróbkowanym sygnałem,
* informacja zwrotna.



Rys. 3 - Funkcja resampleit.

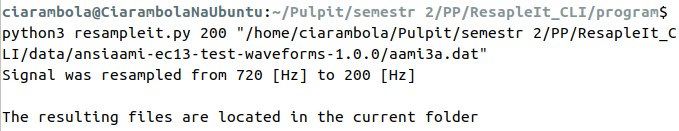
# Specyfikacja zewnętrzna

Istnieją dwie metody użycia narzędzia:

* Wraz z argumentami:
  + przykład zastosowania:
    - wywołanie interpretera “python3”,
    - wskazanie pliku narzędzia “resampleit\_py”,
    - argument pierwszy - częstotliwość docelowa “200”,
    - argument drugi - pełna ścieżka do rekordu wfdb, jak poniżej:

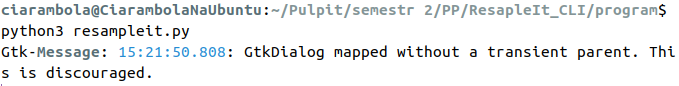
**>> python3 resampleit.py 200 "/home/ciarambola/Pulpit/semestr 2/PP/ResapleIt\_CLI/data/ansiaami-ec13-test-waveforms-1.0.0/aami3a.dat"**

* + widok terminala:



Rys. 4 - Widok terminala przy zastosowaniu narzędzia z argumentami.

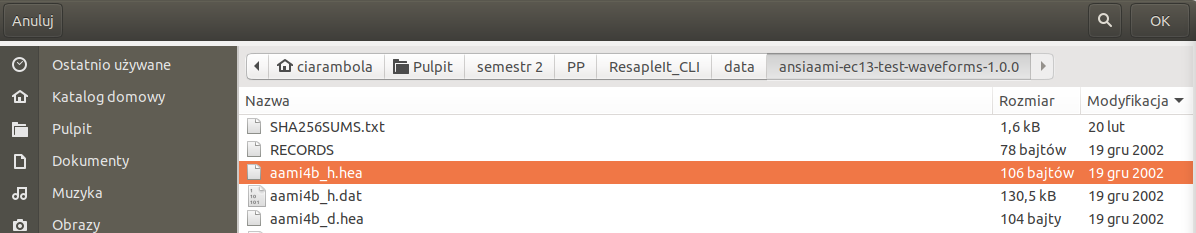
* Bez argumentów:
  + przykład zastosowania:
    - wywołanie interpretera python3 oraz użycie narzędzia resampleit.py bez argumentów:



Rys. 5 - Widok terminala przy zastosowaniu narzędzia bez argumentów.

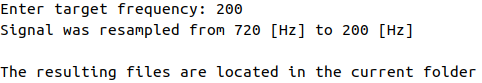
Ewentualne powiadomienia w terminalu należy zignorować o ile nie informują o błędzie, przerywając działanie narzędzia.

Od razu po wpisaniu polecenia do konsoli powinno pojawić się okno wyboru pliku.



Rys. 6 - Okno wyboru pliku.

* + - następnie należy wprowadzić w oknie terminala docelową częstotliwość próbkowania i zatwierdzić wybór.



Rys. 7 - Widok terminala po wprowadzeniu i zatwierdzeniu docelowej częstotliwości próbkowania.

Wynik działania narzędzia znajduje się w bieżącej pozycji terminala, w przykładach jest to folder “program”.

# Podsumowanie

Utworzono zgodne z założeniami narzędzie do konwersji częstotliwości próbkowania. Można go używać zarówno w formie interaktywnej oraz w innym skrypcie podając potrzebne argumenty.

Bibliografia

1. Opracowanie w notatniku Jupyter - <https://sound.eti.pg.gda.pl/~greg/dsp/05-Interpolacja.html> [dostęp: 19.10.2019]
2. Wykłady, Zaawansowane metody analizy sygnałów biologicznych, Prof. dr hab. inż. Ewaryst Tkacz
3. Instrukcja do Laboratorium Cyfrowego przetwarzania sygnałów, Ćwiczenie 5 - Wieloczęstotliwościowe przetwarzanie sygnałów - intepolacja i decymacja - [https://ioisp.el.pcz.pl/images/instrukcje/air/Cyfrowe%20Przetwarzanie%20Sygnalow/Laboratoria/Cw.5%20Wieloczestotliwosciowe%20przetwarzanie%20sygnalow%20–%20interpolacja%20i%20decymacja.pdf](https://ioisp.el.pcz.pl/images/instrukcje/air/Cyfrowe%20Przetwarzanie%20Sygnalow/Laboratoria/Cw.5%20Wieloczestotliwosciowe%20przetwarzanie%20sygnalow%20%E2%80%93%20interpolacja%20i%20decymacja.pdf) [dostęp: 19.10.2019]
4. Przetwarzanie sygnałów, Ćwiczenie 3 - Właściwości przekształcenia Fouriera - http://www.w12.pwr.wroc.pl/ps/instrukcje/PS4.pdf [dostęp: 19.10.2019]