



Imię i nazwisko studenta: Amadeusz Lisiecki

Nr albumu: 134709

Studia pierwszego stopnia

Forma studiów: stacjonarne

Międzywydziałowy kierunek studiów: Inżynieria Biomedyczna

Realizowany na:

Wydziale Elektroniki, Telekomunikacji i Informatyki;

Wydziale Chemicznym;

Wydziale Fizyki Technicznej i Matematyki Stosowanej

Specjalność/profil: Informatyka w Medycynie

PROJEKT DYPLOMOWY INŻYNIERSKI

Tytuł projektu w języku polskim: Program komputerowy do wyznaczania charakterystycznych załamków w przebiegu EKG

Tytuł projektu w języku angielskim: ECG waves detection software

Potwierdzenie przyjęcia projektu	
Opiekun projektu	Kierownik Katedry/Zakładu
<i>Podpis</i>	<i>Podpis</i>
dr inż. Mateusz Moderhak	dr hab. inż. Jacek Rumiński, prof. nadzw. PG

Data oddania projektu do dziekanatu

OŚWIADCZENIE

Imię i nazwisko: Amadeusz Lisiecki

Data i miejsce urodzenia: 24.12.1991r., Szczecin

Nr albumu: 134709

Międzywydziałowy kierunek studiów: Inżynieria Biomedyczna

Realizowany na:

Wydziale Elektroniki, Telekomunikacji i Informatyki;

Wydziale Chemicznym;

Wydziale Fizyki Technicznej i Matematyki Stosowanej

Poziom studiów: pierwszy

Forma studiów: stacjonarne

Ja, niżej podpisany(a), wyrażam zgodę/~~nie wyrażam zgody~~* na korzystanie z mojego projektu dyplomowego zatytułowanego: Oprogramowanie generowania profilu osoby na podstawie analizy zachowania w czasie interakcji z gramami komputerowymi.¹

Gdańsk, dnia.....

.....
podpis studenta

Świadomy(a) odpowiedzialności karnej z tytułu naruszenia przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. z 2006 r., nr 90, poz. 631) i konsekwencji dyscyplinarnych określonych w ustawie Prawo o szkolnictwie wyższym (Dz. U. z 2012 r., poz. 572 z późn. zm.),² a także odpowiedzialności cywilno-prawnej oświadczam, że przedkładany projekt dyplomowy został opracowany przeze mnie samodzielnie.

Niniejszy projekt dyplomowy nie był wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem tytułu zawodowego.

Wszystkie informacje umieszczone w ww. projekcie dyplomowym, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami zgodnie z art. 34 ustawy o prawie autorskim i prawach pokrewnych.

Potwierdzam zgodność niniejszej wersji projektu dyplomowego z załączoną wersją elektroniczną.

Gdańsk, dnia.....

.....
podpis studenta

Upoważniam Politechnikę Gdańską do umieszczenia ww. projektu dyplomowego w wersji elektronicznej w otwartym, cyfrowym repozytorium instytucjonalnym Politechniki Gdańskiej oraz poddawania jego procesom weryfikacji i ochrony przed przywłaszczeniem jej autorstwa.

Gdańsk, dnia.....

.....
podpis studenta

¹ Zarządzenie Rektora Politechniki Gdańskiej nr 34/2009 z 9 listopada 2009 r., załącznik nr 8 do instrukcji archiwalnej PG.

² Ustawa z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym:

Art. 214 ustęp 4. W razie podejrzenia popełnienia przez studenta czynu polegającego na przypisaniu sobie autorstwa istotnego fragmentu lub innych elementów cudzego utworu rektor niezwłocznie poleca przeprowadzenie postępowania wyjaśniającego.

Art. 214 ustęp 6. Jeżeli w wyniku postępowania wyjaśniającego zebrany materiał potwierdza popełnienie czynu, o którym mowa w ust.4, rektor wstrzymuje postępowanie o nadanie tytułu zawodowego do czasu wydania orzeczenia przez komisję dyscyplinarną oraz składa zawiadomienie o popełnieniu przestępstwa.

STRESZCZENIE

Praca zajmuje się zagadnieniem wykrywania charakterystycznych załamków zarejestrowanych w Elektrokardiogramie. Celem było stworzenie aplikacji desktopowej, która pozwoli na wczytanie sygnału EKG i jego automatyczną analizę, detekcję charakterystycznych załamków PQRST i prezentację wyników w dogodnej formie. W procesie analizy EKG stosowano metody z dziedziny cyfrowego przetwarzania sygnałów oraz konieczne było wykazanie się umiejętnościami programowania w języku Python – zarówno pod względem konstruowania aplikacji użytkowych, jak i pisanie skryptów do naukowego przetwarzania danych. Niezbędne były również umiejętności matematyczne na poziomie akademickim.

ABSTRACT

The thesis is dealing with an issue of detecting characteristic wavelets recorded on Electrocardiogram. The goal was to create a desktop application giving a functionality to load an ECG signal, analyze it automatically, detect characteristic PQRST waves and present the results in convenient fashion. During the process of ECG analysis there were used methods of digital signal processing and it was essential to improve oneself with the skills of Python programming – both in terms of constructing user applications, and scientific data processing. Academic level mathematical skills were also essential.

SPIS TREŚCI

STRESZCZENIE	5
ABSTRACT	6
SPIS TREŚCI	7
WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW	9
1. WSTĘP I CEL PRACY	10
2. ANALIZA STANU WIEDZY	11
2.1. Elektrokardiografia	11
2.1.1. <i>Potencjał czynnościowy komórek mięśnia sercowego</i>	11
2.1.2. <i>Elektrokardiograf</i>	12
2.1.3. <i>Elektrokardiogram i charakterystyczne załamki</i>	13
2.2. Charakterystyka szumu i artefaktów w sygnale EKG	14
2.3. Algorytm detekcji QRS Pan-Tompkins	15
2.3.1. <i>Filtracja dolnoprzepustowa</i>	15
2.3.2. <i>Filtracja górnoprzepustowa</i>	16
2.3.3. <i>Różniczkowanie</i>	16
2.3.4. <i>Potęgowanie</i>	17
2.3.5. <i>Całkowanie w ruchomym oknie</i>	17
2.3.6. <i>Wyznaczanie załamka R adaptacyjnym progiem</i>	17
2.4. Algorytm detekcji odstępu QT Laguna-Thakor	18
2.4.1. <i>Różniczkowanie</i>	19
2.4.2. <i>Filtracja dolnoprzepustowa</i>	19
2.4.3. <i>Wyznaczenie zer i ekstremów zróżniczkowanego sygnału</i>	19
2.4.4. <i>Detekcja załamków Q, R i S</i>	19
2.4.5. <i>Obliczanie odstępu RR</i>	20
2.4.6. <i>Detekcja początku QRS</i>	20
2.4.7. <i>Detekcja załamka T</i>	21
2.4.8. <i>Obliczenie odstępu QT</i>	22
3. ANALIZA PROJEKTOWA	23
3.1. Cel i zakres projektu	23
3.2. Wymagania	23
3.2.1. <i>Wymagania funkcjonalne</i>	23
3.2.2. <i>Wymagania pozafunkcjonalne</i>	24
3.3. Przypadki użycia	24
4. PROJEKT	25
4.1. Diagram klas aplikacji	25
4.2. Algorytm	26
4.2.1. <i>Preprocessing</i>	26
4.2.2. <i>Algorytm Pan-Tompkins</i>	26
4.2.3. <i>Algorytm Laguna-Thakor</i>	26
4.3. Opis klas i pakietów	29
4.4. Baza danych PhysioNet	29
5. IMPLEMENTACJA	30
5.1. Implementacja algorytmów	30
5.1.1. <i>Preprocessing</i>	30
5.1.2. <i>Algorytm Pan-Tompkins</i>	31

5.1.3. Algorytm Laguna-Thakor.....	32
5.2. Implementacja GUI	34
5.3. Podsumowanie	36
6. TESTY	37
6.1. Testy funkcjonalne	37
6.1.1. Scenariusz testowy nr 1	37
6.1.2. Scenariusz testowy nr 2	37
6.1.3. Scenariusz testowy nr 3	38
6.1.4. Scenariusz testowy nr 4	38
6.2. Analiza otrzymanych wyników	38
6.3. Wpływ zakłóceń na skuteczność detekcji.....	39
6.4. Podsumowanie	41
7. PODSUMOWANIE	42
WYKAZ LITERATURY	43
WYKAZ RYSUNKÓW	45
WYKAZ LISTINGÓW	46
WYKAZ TABEL	47

WYKAZ WAŻNIEJSZYCH OZNACZEŃ I SKRÓTÓW

P	—	Argument największej bezwzględnej wartości w załamku P
Q	—	Argument największej bezwzględnej wartości w załamku Q
R	—	Argument największej bezwzględnej wartości w załamku R
S	—	Argument największej bezwzględnej wartości w załamku S
T	—	Argument największej bezwzględnej wartości w załamku T
QRS	—	Zespół QRS
d(n)	—	Zrózniczkowany sygnał EKG
f(n)	—	Przefiltrowany sygnał EKG
GUI	—	Graficzny interfejs użytkownika
PTB	—	Physikalisch-Technische Bundesanstalt, niemiecki narodowy instytut miar

1. WSTĘP I CEL PRACY

Choroby układu krążenia są najczęstszą przyczyną zgonu w Polsce, wg GUS były one powodem 46% wszystkich zgonów w 2015 roku. Jedną z najpopularniejszych metod diagnostycznych wykorzystywanych w celu wykrywania chorób serca jest Elektrokardiografia, w skrócie EKG.

Metoda ta polega na mierzeniu potencjałów elektrycznych, na powierzchni klatki piersiowej pacjenta, będących wynikiem czynności elektrycznej serca podczas jego pracy. Otrzymany w ten sposób przebieg EKG nie tylko pozwala na obliczenie tempa bicia serca, ale także pokazuje charakterystykę depolaryzacji i repolaryzacji przedsionków oraz komór serca. Widoczne jest to na przebiegu EKG w postaci załamków oznaczonych literami P, Q, R, S i T - odpowiadają one różnym etapom pracy mięśnia sercowego. Obecność i umiejscowienie tych załamków w przebiegu EKG jest ściśle związana z kondycją serca, a ich nieprawidłowa lokalizacja może być objawem choroby.

Automatyzacja procesu detekcji załamków PQRST, a także obliczenie charakterystycznych odległości i wartości w przebiegu EKG może znacznie usprawnić i przyspieszyć diagnostykę chorób mięśnia sercowego. Dalszy rozwój oprogramowania diagnostycznego tego typu może pozwolić na stworzenie systemów sugerujących lekarzowi diagnozę, a co za tym idzie wykrywanie nieprawidłowości w pracy serca na wczesnym etapie rozwoju choroby.

Celem pracy jest stworzenie oprogramowania, z graficznym interfejsem użytkownika, pozwalającego na wykrywanie załamków PQRST w oczyszczonym sygnale EKG. Program umożliwi na wprowadzenie pliku zawierającego sygnału EKG, wyświetlenie przebiegu na wykresie i przeprowadzenie detekcji – zaznaczenie pozycji załamków na wykresie.

2. ANALIZA STANU WIEDZY

W rozdziale dotyczącym analizy stanu wiedzy uwaga zostanie skupiona na metodzie Elektrokardiografii, zostanie opisane w jakim celu stosuje się tę metodę, z jakich narzędzi się korzysta i co jest efektem całego badania. Następnie zostaną pokrótce opisane trudności towarzyszące pomiarowi i analizie sygnału z EKG. Ostatnim elementem rozdziału będzie przedstawienie dwóch algorytmów wykorzystywanych do analizy sygnału pozyskiwanego z elektrokardiografu i wyznaczania charakterystycznych załamków.

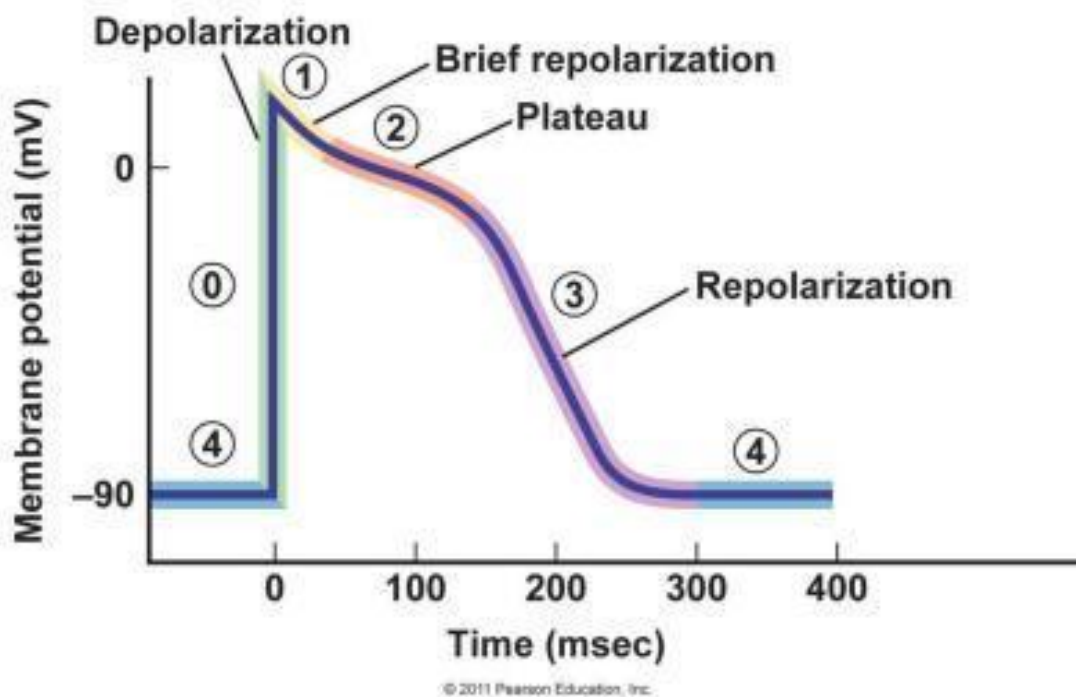
2.1. Elektrokardiografia

2.1.1. Potencjał czynnościowy komórek mięśnia sercowego

Celem elektrokardiografii jest zarejestrowanie czynności elektrycznej mięśnia sercowego, a dokładnie zmian potencjału elektrycznego wywołanych depolaryzacją i repolaryzacją błon komórek mięśni przedsionków oraz komór.

W potencjale czynnościowym mięśnia sercowego wyróżniamy następujące fazy:

- Faza 0 – szybka depolaryzacja, nagły wzrost potencjału błony komórkowej,
- Faza 1 – wstępna szybka repolaryzacja, nieznaczne obniżenie potencjału,
- Faza 2 – powolna repolaryzacja, faza plateau,
- Faza 3 – szybka repolaryzacja, ustalenie się potencjału spoczynkowego,
- Faza 4 – polaryzacja, utrzymanie potencjału spoczynkowego.

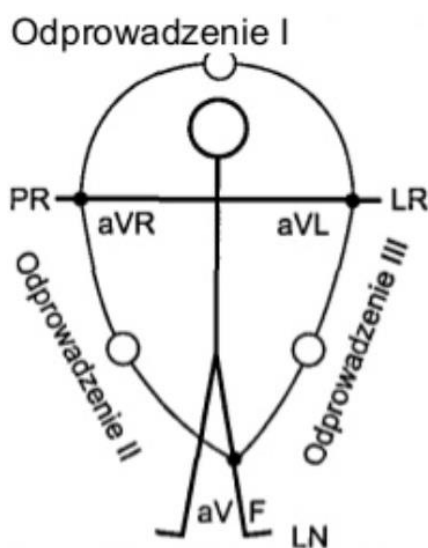


Rys. 2.1. Wykres zmiany potencjału błony komórkowej mięśnia sercowego [1]

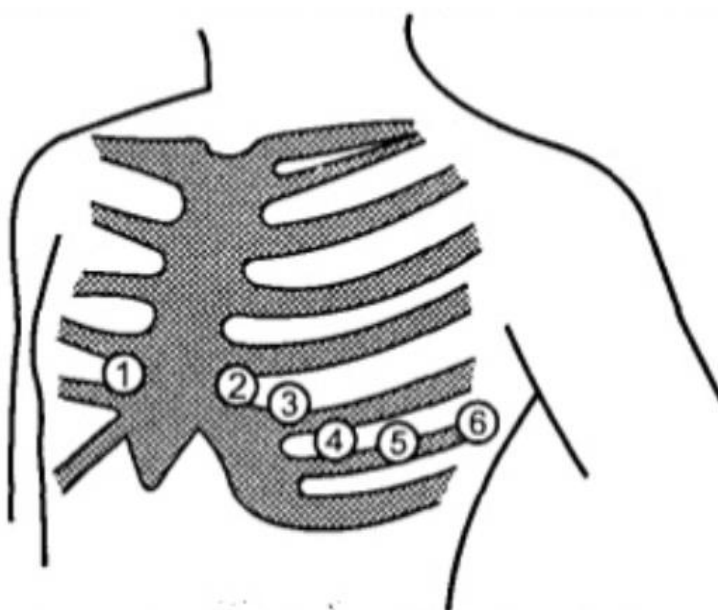
2.1.2. Elektrokardiograf

Pomiar potencjału czynnościowego dokonujemy za pomocą urządzenia zwanego elektrokardiografem, z użyciem elektrod umiejscowionych na ciele pacjenta. Standardowo badanie wykonuje się przy użyciu 12 odprowadzeń [2]:

- 3 dwubiegunowe kończynowe (I, II, III) – „Einthovena”,
- 3 jednobiegunowe kończynowe (aVR, aVL, aVF) – „Goldbergera”,
- 6 jednobiegunowych przedsercowych (V₁, V₂, V₃, V₄, V₅, V₆) – „Wilsona”.



Rys. 2.2. Odprowadzenia kończynowe Einthovena i Goldbergera [3]



Rys. 2.3. Umiejscowienie odprowadzeń Wilsona na klatce piersiowej [3]

W przypadku 3 odprowadzeń dwubiegunowych powstają one z połączenia lewego (LR) i prawego (PR) przedramienia (odprowadzenie I), prawego przedramienia (PR) i lewego golenia (LN) tworząc odprowadzenie II oraz odprowadzenie III pomiędzy lewym goleniem (LN) i lewym przedramieniem (LR). Elektroda LN stanowi punkt odniesienia – masę układu pomiarowego.

Wzmocnione jednobiegunowe odprowadzenia Goldbergera umiejscowione są na prawej ręce (PR) – odprowadzenie aVR, lewej ręce (LR) – odprowadzenie aVL, lewym goleniu (LN) – odprowadzenie aVF. Sygnał każdej z elektrod mierzony jest w stosunku do sumy potencjałów dwóch pozostałych. Litera „a” oznacza, że sygnał jest wzmocniony (ang. „augmented” - wzmocniony) natomiast „V” oznacza odprowadzenie jednobiegunowe.

Odprowadzenia Wilsona umiejscawiamy w przestrzeni międzyżebrowej co jest widoczne na Rys. 2.3. i dokonujemy pomiaru w stosunku do potencjału odniesienia, bliskiemu zeru [3].

Efektom pomiaru elektrokardiografem jest elektrokardiogram.

2.1.3. *Elektrokardiogram i charakterystyczne załamki*

Sygnał przedstawiony w elektrokardiogramie cechuje okresowość – powtarzające się skurcze mięśnia sercowego generują sygnał o powtarzającej się charakterystyce. Można w nim wyróżnić pewne charakterystyczne elementy. Punktem odniesienia jest linia izoelektryczna czyli pozioma linia przedstawiająca wartość potencjału w momencie braku aktywności elektrycznej serca. Wychylenia od tej linii nazywamy „załamkami” i są one wynikiem zmiany potencjału czynnościowego w różnych regionach serca. Wyróżniamy następujące załamki [4] [5]:

- Załamek P – depolaryzacja przedsionków,
- Załamek Q – depolaryzacja przegrody międzykomorowej,
- Załamek R – depolaryzacja komór,
- Załamek S – depolaryzacja włókien Purkiniego (podwsierdziowych),
- Załamek T – repolaryzacja komór,
- Załamek U.

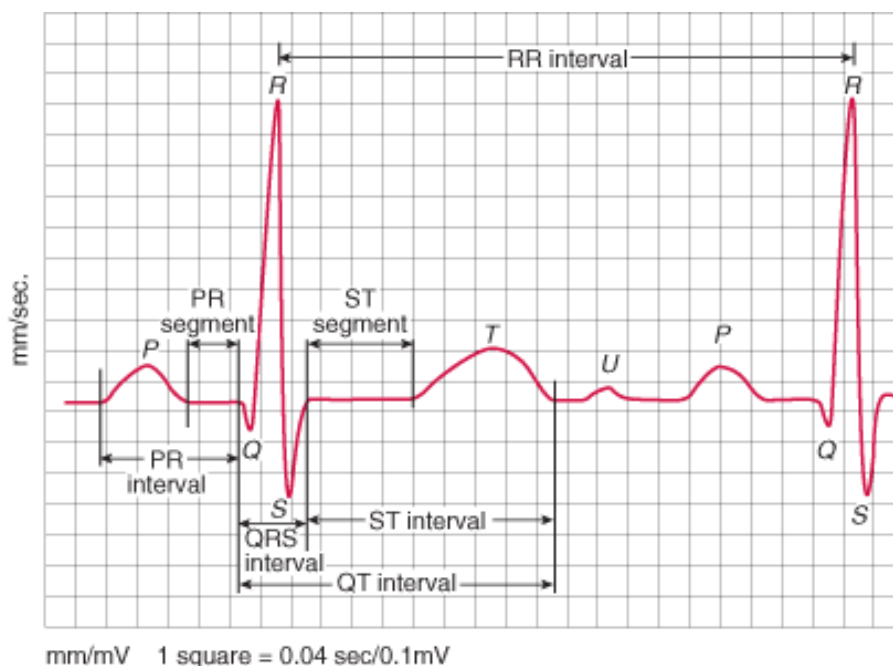
Pochodzenie załamka U jest nieznane i jest on widoczny jedynie w 25% przypadków. Załamek U wzrasta wraz ze spadkiem tętna. Jedną z kilku teorii pochodzenia załamka U to opóźniona repolaryzacja włókien Purkiniego [6].

Załamki Q, R i S tworzą razem zespół QRS, który jest największą pod względem amplitudy i czasu trwania strukturą w sygnale EKG. QRS przysłania trwającą w tym samym czasie repolaryzację przedsionków.

W sygnale EKG wyróżnia się także odcinki (ang. „segment”) i odstępy (ang. „interval”) [5]:

- Odstęp PQ (ang. „PR interval”) – od początku załamka P do początku zespołu QRS, przedstawia czas fali depolaryzacji idącej przez przedsionki, węzły przedsionkowo-komorowe, pęczek Hisa i włókna Purkiniego,
- Odcinek PQ (ang. „PR segment”) – od końca załamka P do początku zespołu QRS to okres przewodnictwa pobudzenia przedsionkowo-komorowego,
- Odstęp QRS (ang. „QRS interval”) – czas trwania zespołu QRS to od 0,06s do 0,11s,

- Odstęp QT (ang. „QT interval”) – od początku załamka Q do końca załamka T, wyraża czas depolaryzacji i repolaryzacji (potencjału czynnościowego) komór serca,
- Odstęp ST (ang. „ST interval”) – od końca załamka S do końca załamka T to fazy 2 i 3 potencjału czynnościowego komór,
- Odcinek ST (ang. „ST segment”) – od końca załamka S do początku załamka T, przedstawia czas pomiędzy depolaryzacją, a repolaryzacją komór,
- Odstęp RR (ang. „RR interval”) – od jednego do kolejnego szczytu załamka R, to pełen cykl pracy serca i służy do wyznaczenia rytmu serca – od 0,6s do 1,2s.



Rys. 2.4. Charakterystyczne załamki EKG [5]

Wszelkie odstępstwa od normy co do morfologii (kształtu) i czasu trwania charakterystycznych załamków, odstępów i odcinków w sygnale EKG mogą być objawami chorób kardiologicznych będących zagrożeniem dla zdrowia i życia pacjenta.

2.2. Charakterystyka szumu i artefaktów w sygnale EKG

Pomiar sygnału EKG wiąże się z dużym prawdopodobieństwem powstania artefaktów oraz silnego szumu, których źródłami mogą być [7]:

- Ruchy mięśni – zakłócenia 5-50 Hz,
- Oddychanie (respiracja) – zakłócenia 0,12-0.5 Hz, dryf linii izoelektrycznej,
- Sieć elektryczna – 50 lub 60 Hz,
- Oderwanie elektrody – prowadzi do powstania artefaktów.

Z drugiej strony interesujące z punktu widzenia analizy EKG komponenty to [7]:

- Tętno – 0,67-5 Hz,
- Załamek P – 0.67-5 Hz,

- Zespół QRS – 10-50 Hz,
- Załamek T – 1-7 Hz,
- Składowe o wysokich częstotliwościach – 100-500Hz.

Amplituda EKG mieści się w przedziale 0.1-2 mV co oznacza, że filtracja jest bardzo istotnym elementem pracy nad sygnałem. Konieczne jest użycie filtra pasmowo-przepustowego. W [8] Pan i Tompkins zaproponowali pasmo przepustowe 5-15 Hz, a w praktyce zastosowali filtr dolnoprzepustowy o częstotliwości odcięcia 11 Hz i górnoprzepustowy o odcięciu 5 Hz. W [9] Laguna i Thakor zastosowali filtr dolnoprzepustowy o częstotliwości odcięcia 20 Hz.

2.3. Algorytm detekcji QRS Pan-Tompkins

Istnieje wiele różnych algorytmów pozwalających na wyznaczanie załameków, odstępów i odcinków w przebiegu EKG. Przedstawione zostaną 2 popularne [8] [9] algorytmy pozwalające na wyznaczenie kompleksów QRS w sygnale EKG.

Algorytm Pan-Tompkins [8] pozwala na detekcję zespoły QRS i odstępu RR. Dzieli się na fazę preprocessingu – filtracji oraz przygotowania sygnału, oraz fazę wykrywania kompleksu QRS adaptacyjnym progiem. Częstotliwość próbkowania f_s wynosiła 200 Hz.

2.3.1. Filtracja dolnoprzepustowa

Transmitancję filtra przedstawia wzór (2.1):

$$H(z) = \frac{(1-z^{-6})^2}{(1-z^{-1})^2} \quad (2.1)$$

gdzie:

- $H(z)$ – transmitancja filtra,
 z – operator opóźnienia.

Co w dziedzinie czasu dyskretnego przedstawia się jako:

$$y(nT) = 2y(nT - T) - y(nT - 2T) + x(nT) - 2x(nT - 6T) + x(nT - 12T) \quad (2.2)$$

gdzie:

- $y(nT)$ – sygnał wyjściowy,
 $x(nT)$ – sygnał wejściowy,
 T – okres próbkowania,
 n – numer próbki.

Częstotliwość odcięcia filtra (2.2) to $f_c = 11$ Hz, a opóźnienie to 6 próbek (30 ms). Filtr ma nieskończoną odpowiedź impulsową.

2.3.2. Filtracja górnoprzepustowa

Transmitancję filtra przedstawia wzór (2.3):

$$H(z) = \frac{-1+32z^{-16}+z^{-32}}{1+z^{-1}} \quad (2.3)$$

gdzie:

- $H(z)$ – transmitancja filtra,
- z – operator opóźnienia.

W dziedzinie czasu dyskretnego:

$$y(nT) = 32x(nT - 16T) - [y(nT - T) + x(nT) - x(nT - 32T)] \quad (2.4)$$

gdzie:

- $y(nT)$ – sygnał wyjściowy,
- $x(nT)$ – sygnał wejściowy,
- T – okres próbkowania,
- n – numer próbki.

Filtr (2.4) ma nieskończoną odpowiedź impulsową, częstotliwość odcięcia $f_c = 5$ Hz i wprowadza opóźnienie 16 próbek (80 ms).

2.3.3. Różniczkowanie

Różniczkowanie sygnału zdobyciu informacji na temat nachylenia załamek. Wzór (2.5):

$$H(z) = (1/8T)(z^{-2} - 2z^{-1} + 2z^1 + z^2) \quad (2.5)$$

gdzie:

- $H(z)$ – transmitancja filtra,
- T – okres próbkowania,
- z – operator opóźnienia.

W dziedzinie czasu dyskretnego:

$$y(nT) = (1/8T)[-x(nT - 2T) + 2x(nT - T) + 2x(nT + T) + x(nT + 2T)] \quad (2.6)$$

gdzie:

- $y(nT)$ – sygnał wyjściowy,
- $x(nT)$ – sygnał wejściowy,
- T – okres próbkowania,
- n – numer próbki.

Filtr różniczkujący (2.6) powinien przybliżać idealne różniczkowanie w paśmie do 30 Hz. Wprowadza opóźnienie 2 próbek.

2.3.4. Potęgowanie

Potęgowanie (2.7) spowoduje, że wszystkie wartości sygnału stają się dodatnie oraz dodatkowo wzmocnia składowe o wysokich częstotliwościach.

$$y(nT) = [x(nT)]^2 \quad (2.7)$$

gdzie:

- $y(nT)$ – sygnał wyjściowy,
- $x(nT)$ – sygnał wejściowy,
- T – okres próbkowania,
- n – numer próbki.

2.3.5. Całkowanie w ruchomym oknie

Celem całkowania (2.8) w ruchomym oknie jest połączenie falek zespołu QRS spotęgowanego sygnału w jeden załamek. Długość okna powinna być optymalnie wyważona, ponieważ za duże okno spowoduje połączenie zespołu QRS z załamkami P i T, a za małe okno nie spowoduje połączenia zespołu QRS. Wykorzystano okno o długości 30 próbek (150 ms).

$$y(nT) = (1/N)[x(nT - (N - 1)T) + x(nT - (N - 2)T) + \dots + x(nT)] \quad (2.8)$$

gdzie:

- $y(nT)$ – sygnał wyjściowy,
- $x(nT)$ – sygnał wejściowy,
- T – okres próbkowania,
- N – szerokość okna,
- n – numer próbki.

2.3.6. Wyznaczanie załamka R adaptacyjnym progiem

Ten etap wykonywany jest zarówno na sygnale po filtracji jak i po całkowaniu. Punkt możemy uznać za R jeżeli został jako taki oznaczony zarówno podczas wyznaczania z adaptacyjnym progiem na sygnale po filtracji jak i na sygnale po całkowaniu.

Pierwszym elementem jest wyznaczenie ekstremów sygnału, ponieważ to one są pierwszymi kandydatami na załamek R. Następnie należy wyznaczyć wartość inicjującą, np. punkt o największej wartości w sygnale. Uznajemy, że dany punkt jest R jeżeli przekracza, któryś z dwóch progów: H_1 lub H_2 . Progi wyznaczamy na podstawie dwóch adaptacyjnych parametrów wyznaczanych wg wzorów (2.9) i (2.10).

$$SPK = 0.125 \cdot PEAK + 0.875 \cdot SPK \quad (2.9)$$

gdzie:

- SPK* – ruchoma estymata pików R,
PEAK – aktualnie rozważany pik.

$$NPK = 0.125 \cdot PEAK + 0.875 \cdot NPK \quad (2.10)$$

gdzie:

- SPK* – ruchoma estymata pików szumu,
PEAK – aktualnie rozważany pik.

Jeżeli dane ekstremum (pik) przekracza wartość adaptacyjnego progu H_1 to uznajemy go za R i aktualizujemy wartość SPK wzorem (2.8). Wzór na próg H_1 to:

$$H_1 = NPK + 0.25 \cdot (SPK - NPK) \quad (2.11)$$

gdzie:

- H_1 – pierwszy próg adaptacyjny,
SPK – ruchoma estymata pików R,
NPK – ruchoma estymata pików szumu.

Jeżeli nie znaleziono, R i wymagane jest powtórne przeszukiwanie tego samego obszaru stosujemy drugi próg adaptacyjny H_2 , który stanowi 50% wartości progu H_1 . W przypadku znalezienia R z wykorzystaniem progu H_2 stosujemy następujący wzór do wyznaczenia nowej wartości SPK:

$$SPK = 0.125 \cdot PEAK + 0.75 \cdot SPK \quad (2.12)$$

gdzie:

- SPK* – ruchoma estymata pików R,
PEAK – aktualnie rozważany pik.

Jeżeli wartość badanego ekstremum nie przekroczy progu uznajemy je za szum i aktualizujemy wartość NPK wg wzoru (2.10).

2.4. Algorytm detekcji odstępów QT Laguna-Thakor

Algorytm [9] możemy podzielić na kilka postępujących po sobie etapów:

- Preprocessing – przygotowanie sygnał do analizy:
 - Różniczkowanie – uzyskanie informacji na temat nachyleń załamek,
 - Filtracja dolnoprzepustowa – usunięcie szumów,
- Wyznaczenie zer i lokalnych ekstremów (pików) sygnału zróżniczkowanego,
- Detekcja załamek Q, R i S,

- Obliczenie odstępu RR,
- Detekcja początku QRS,
- Detekcja załamka T,
- Obliczenie odstępu QT.

Wartość sygnału zróżniczkowanego w punkcie n oznaczamy jako $d(n)$.

Algorytm został przetestowany na sygnałach EKG z odprowadzenia II.

2.4.1. Różniczkowanie

Filtr różniczkujący o transmitancji (2.13) wprowadza opóźnienie 3 próbek. Zastosowanie go pozwoli uzyskać informacje na temat nachylenia zbocza załamków.

$$H(z) = 1 - z^{-6} \quad (2.13)$$

gdzie:

$H(z)$ – transmitancja filtru,
 z – operator opóźnienia.

2.4.2. Filtracja dolnoprzepustowa

Filtr dolnoprzepustowy o transmitancji (2.14) wprowadza opóźnienie 4,5 próbki i ma częstotliwość odcięcia $f_c = 20$ Hz. Służy ona pozbyciu się zakłóceń wysokich częstotliwości – wygładzeniu sygnału zróżniczkowanego.

$$H(z) = \frac{1 - z^{-8}}{1 - z^{-1}} \quad (2.14)$$

gdzie:

$H(z)$ – transmitancja filtru,
 z – operator opóźnienia.

2.4.3. Wyznaczenie zer i ekstremów zróżniczkowanego sygnału

Ekstrema lokalne (piki) sygnału zróżniczkowanego to miejsca, w których sygnał EKG ma najbardziej strome nachylenie – minima oznaczają, że sygnał maleje, a maxima, że rośnie.

Punkty przecięcia linii izoelektrycznej przez sygnał zróżniczkowany wyznaczają extrema lokalne w sygnale EKG.

2.4.4. Detekcja załamków Q, R i S

Przez wyznacznienie załamków Q, R i S mamy na myśli znalezienie argumentów Q, R i S dla największych bezwzględnych wartości załamków Q, R i S. Stosujemy adaptacyjny próg podobnie jak w 2.3.6., dany wzorem (2.15):

$$H_{n+1} = 0.8H_n + 0.16|d(PK_n)| \quad (2.15)$$

gdzie:

H_{n+1} – próg w następnej iteracji,
 H_n – aktualny próg,

$d(PK_n)$ – aktualny pik sygnału różniczkowanego,
 n – numer iteracji.

Iterujemy kolejno po szczytach sygnału różniczkowanego. Próg startowy to 80% wartości bezwzględnej największego szczytu (dodatniego lub ujemnego) w pierwszych 2 sekundach sygnału.

Jeżeli $|d(PK_n)| > |H_n|$ - definiujemy PK_a jako następny pik po PK_n i PK_b jako pik przed PK_n .

- Jeżeli $|d(PK_b)| > |d(PK_a)|$:
 - R znajduje się w pierwszym zerze sygnału różniczkowanego przed PK_n ,
- W przeciwnym wypadku:
 - R znajduje się w pierwszym zerze sygnału różniczkowanego po PK_n .

S to pierwsze zero sygnału różniczkowanego po R.

Q to pierwsze zero sygnału różniczkowanego przed R.

Jeżeli Q zostanie wykryte dalej niż 80 ms od R to uznajemy, że nie ma załamka Q.

2.4.5. Obliczanie odstęp RR

Odległość pomiędzy wyznaczonym w danej iteracji, a poprzednim R oznaczamy jako odstęp RR [ms]. Średni odstęp RR_{avg} wyznaczamy wg wzoru (2.16):

$$RR_{avg} = \begin{cases} 0.8RR_{avg} + 0.2RR & 1.5RR_{avg} > RR > 0.5RR_{avg} \\ RR_{avg} & \text{w przeciwnym wypadku} \end{cases} \quad (2.16)$$

gdzie:

RR_{avg} – średni odstęp RR,
 RR – odstęp RR.

2.4.6. Detekcja początku QRS

Poszukujemy punktu Q_i (lub adekwatnie R_i jeżeli Q nie istnieje), który jest pierwszym ekstremum lokalnym sygnału różniczkowanego przed Q – oznacza to punkt o największym nachyleniu sygnału EKG, zaraz przed Q.

Następnie wyznaczamy próg H, który liczymy wg wzoru (2.17) jeżeli Q istnieje lub (2.18) w wypadku przeciwnym.

$$H = d(Q_i)/2 \quad (2.17)$$

gdzie:

H – próg wyznaczania QRS_{start} ,
 $d(Q_i)$ – wartość sygnału różniczkowanego w punkcie Q_i ,
 Q_i – pierwsze ekstremum lokalne sygnału różniczkowanego przed Q.

$$H = d(R_i)/5 \quad (2.18)$$

gdzie:

H – próg wyznaczania QRS_{start} ,
 $d(R_i)$ – wartość sygnału różniczkowanego w punkcie R_i ,
 R_i – pierwsze ekstremum lokalne sygnału różniczkowanego przed R.

Punkt przecięcia progu H, przed Q_i (lub R_i), przez sygnał zróżniczkowany to początek zespoły QRS – QRS_{start} .

2.4.7. Detekcja załamka T

W celu odnalezienia załamka T wyznaczamy okno wg wzoru (2.19), w którym przeprowadzone zostaną poszukiwania.

$$(bwind, ewind) = \begin{cases} (140, 500)ms & RR_{avg} > 700 ms \\ ((100, 0.7RR_{avg})ms & \text{w przeciwnym wypadku} \end{cases} \quad (2.19)$$

gdzie:

$bwind$ – początek okna (odstęp od R w ms),

$ewind$ – koniec okna (odstęp od R w ms),

RR_{avg} – średni odstęp RR.

Poszukujemy wartości minimalnej (min) i maksymalnej (max) sygnału zróżniczkowanego w wyznaczonym oknie ($bwind$, $ewind$). Numery ich próbek w sygnale to min_n i max_n .

- Jeżeli ($min_n > max_n$) – max jest przed min, a załamek T jest normalny (up-down) albo tylko rosnący (up):
 - Wyznaczamy punkt $T_i = min$,
 - Jeżeli $|max| > 4|min|$:
 - Załamek T jest tylko rosnący (up),
 - W przeciwnym wypadku:
 - Załamek T jest normalny (up-down).
- Jeżeli ($min_n < max_n$) – min jest przed max, a załamek T jest normalny (up-down), odwrócony (down-up) albo tylko malejący (down):
 - Poszukujemy wartości minimalnej (min_2) sygnału zróżniczkowanego w nowym oknie od max_n do końca starego okna – (max_n , $ewind$),
 - Jeżeli $|max| < 4|min_2|$:
 - Wyznaczamy punkt $T_i = min_2$,
 - Załamek T jest normalny (up-down).
 - W przeciwnym wypadku:
 - Jeżeli $|min| > 4|max|$:
 - Wyznaczamy punkt $T_i = min$
 - Załamek T jest tylko malejący (down).
 - W przeciwnym wypadku:
 - Wyznaczamy punkt $T_i = max$
 - Załamek T jest odwrócony (down-up).

W ten sposób wyznaczono kształt załamka T (normalny, odwrócony, tylko rosnący lub tylko malejący) oraz punkt T_i .

T znajduje się w pierwszym zerze sygnału zróżniczkowanego przed T_i .

Następnie obliczamy wg wzoru (2.20) wartość progu H_T potrzebnego do wyznaczenia końca załamka T (T_{end}).

$$H_T = d(T_i)/2 \quad (2.20)$$

gdzie:

- H – próg wyznaczania QRS_{start} ,
- $d(T_i)$ – wartość sygnału zróżniczkowanego w punkcie T_i ,
- T_i – pierwsze ekstremum lokalne sygnału zróżniczkowanego po T.

Punkt przecięcia progu H_T , po T_i , przez sygnał zróżniczkowany to T_{end} .

2.4.8. Obliczenie odstępu QT

Odstęp QT liczymy wg wzoru (2.21). Definiujemy również odstęp QTP (2.22) jako czas pomiędzy początkiem zespołu QRS, a T.

$$QT = 1000(T_{end} - QRS_{start})/f_s \quad (2.21)$$

gdzie:

- QT – odstęp QT [ms],
- f_s – częstotliwość próbkowania sygnału [Hz],
- T_{end} – koniec załamka T,
- QRS_{start} – początek zespołu QRS.

$$QTP = 1000(T - QRS_{start})/f_s \quad (2.22)$$

gdzie:

- QT – odstęp QT [ms],
- f_s – częstotliwość próbkowania sygnału [Hz],
- T_{end} – koniec załamka T,
- QRS_{start} – początek zespołu QRS.

Korzystając ze wzoru Bazetta [10] możemy obliczyć skorygowane wartości QT i QTP.

$$QT_c = QT/\sqrt{RR/1000} \quad (2.23)$$

gdzie:

- QT_c – skorygowany odstęp QT [ms],
- QT – odstęp QT [ms],
- RR – odstęp RR [ms].

$$QTP_c = QTP/\sqrt{RR/1000} \quad (2.24)$$

gdzie:

- QTP_c – skorygowany odstęp QTP [ms],
- QTP – odstęp QTP [ms],
- RR – odstęp RR [ms].

3. ANALIZA PROJEKTOWA

W rozdziale dotyczącym analizy projektowej przedstawione są wymagania jakie są stawiane aplikacji, jakie funkcjonalności powinna realizować i z jakimi przypadkami użycia możemy się spotkać. Przede wszystkim najpierw zostanie przedstawiony cel i zakres projektu.

3.1. Cel i zakres projektu

Celem projektu jest stworzenie aplikacji okienkowej na komputery PC, która pozwoli na:

- Wczytanie sygnału EKG z pliku,
- Wykrycie charakterystycznych załamków w podanym sygnale,
- Prezentację wyników w przyjaznym dla użytkownika interfejsie graficznym.

Docelowymi użytkownikami aplikacji mogą być studenci kierunków technicznych i biomedycznych wykorzystujący ją w celach edukacyjnych podczas zajęć.

Gotowa aplikacja powinna zostać przetestowana pod względem skuteczności detekcji załamków, a także wpływu szumu i artefaktów na ten proces.

3.2. Wymagania

3.2.1. Wymagania funkcjonalne

Aplikacja powinna spełniać szereg wymagań funkcjonalnych:

1. Po uruchomieniu aplikacji wyświetla się ekran powitalny. Możliwe jest wczytanie sygnału EKG z pliku za pomocą przycisku „Otwórz próbkę”, który wywołuje okno dialogowe wyboru pliku, lub wyłączenie programu.
2. Otwarcie pliku w prawidłowym formacie wywołuje automatyczny proces przetwarzania sygnału i detekcji załamków.
3. Po zakończeniu obliczeń na miejscu ekranu powitalnego pojawia się wczytany, surowy sygnał EKG, wewnątrz zakładki „Sygnały”. Dostępne są zakładki:
 - a. „Sygnały”,
 - b. „Widma”,
 - c. „Odstępy”,
 - d. „Tabela załamków”.
4. Wewnątrz zakładki „Sygnały” dostępne są kolejne zakładki, które zawierają wykresy przedstawiające kolejne etapy preprocessingu sygnału oraz wyników detekcji załamków.
5. Zakładka „Widma” zawiera dwie zakładki przedstawiające:
 - a. Wykres widma surowego sygnału EKG,
 - b. Wykres widma sygnału EKG po filtracji.
6. Zakładka „Odstępy” zawiera zakładki przedstawiające:
 - a. Wykres odstępów RR,
 - b. Wykresy odstępów QT, QTP, QT_c i QTP_c,

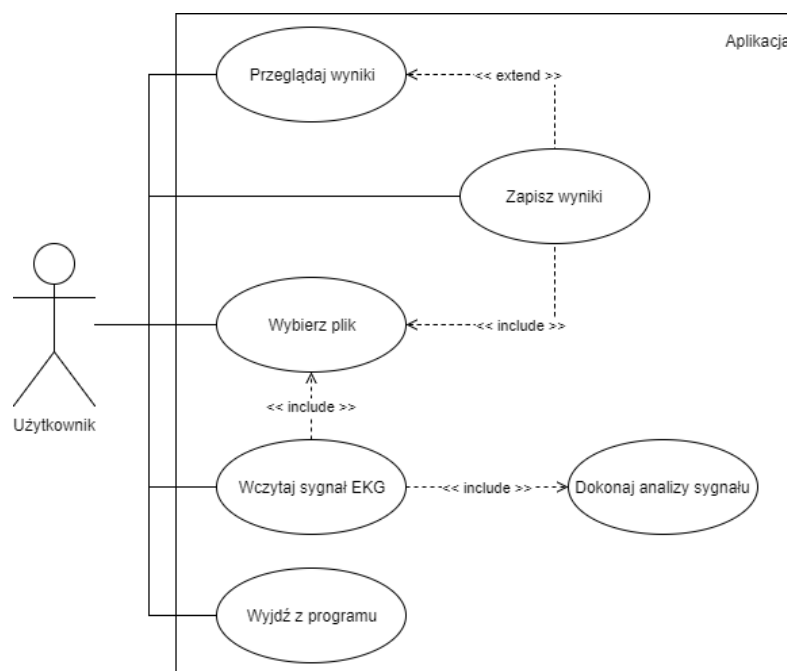
- c. Tabełaryczne zestawienie wszystkich obliczonych odstępów.
7. Zakładka „Tabela załamków” przedstawia tabelę z wynikami detekcji załamków.
8. W oknie aplikacji dostępne są dwa przyciski pozwalające na zapis danych do pliku:
 - a. „Zapisz wszystko” – pozwala na zapisanie do pliku w formacie CSV wyników preprocessingu, detekcji załamków i obliczeń odstępów,
 - b. „Zapisz dane” – dostępne wewnątrz aktywnej zakładki, pozwala na zapis do pliku w formacie CSV danych przedstawionych w tej zakładce.
9. Każdy wykres jest interaktywny, tj. możliwe jest skalowanie, przesuwanie i zapis wykresu do pliku.

3.2.2. Wymagania pozafunkcjonalne

Zidentyfikowano następujące wymagania pozafunkcjonalne:

1. Komputer PC z systemem operacyjnym Microsoft Windows 7, 8, 10 lub Linux.
2. Interpreter Python w wersji 3.6 lub nowszej.
3. Sygnał EKG w formacie MIT lub w pliku CSV.
4. Klawiatura i myszka.

3.3. Przypadki użycia

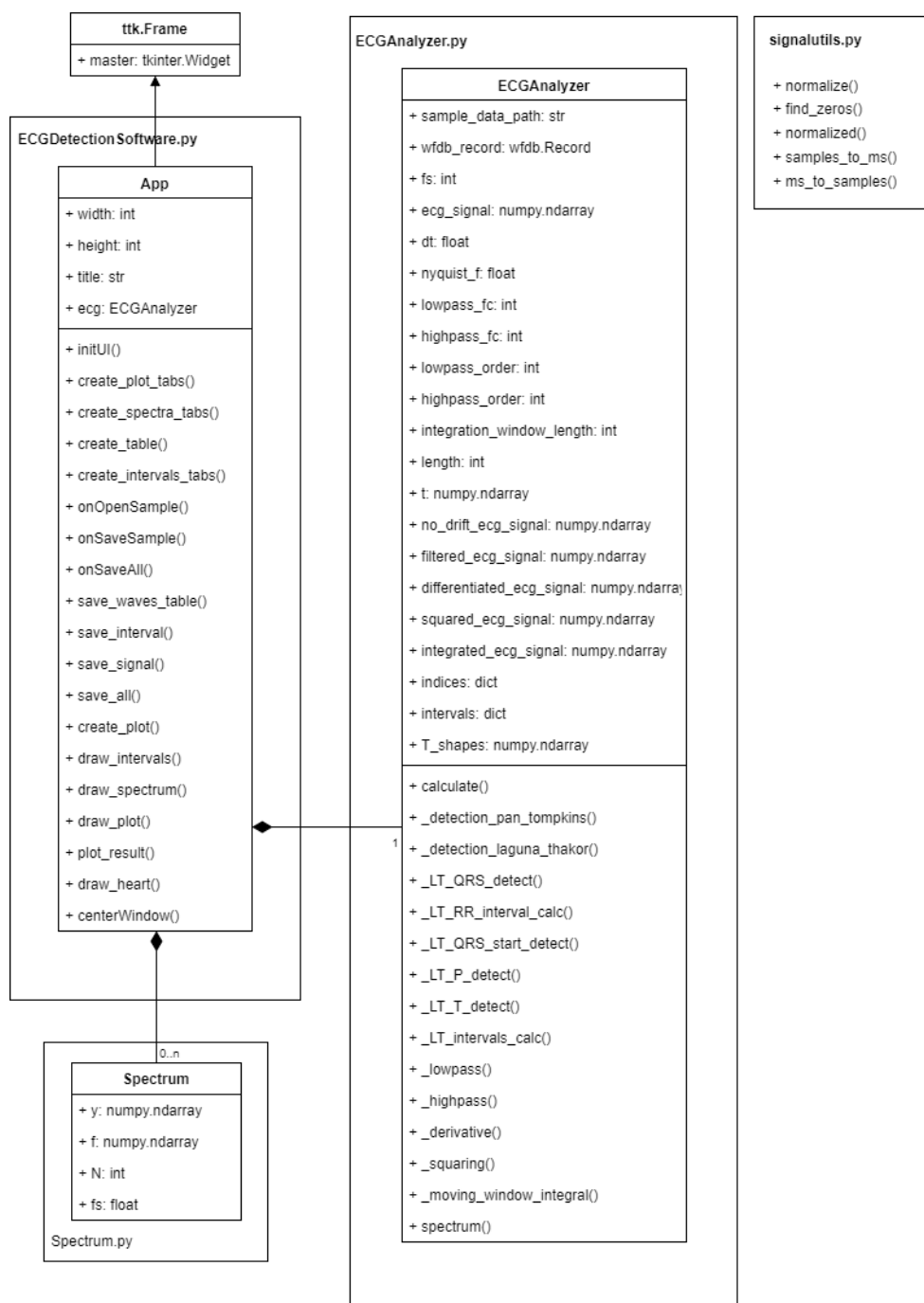


Rys. 3.1. Diagram przypadków użycia utworzony za pomocą [11]

4. PROJEKT

Rozdział projektowy to przede wszystkim strukturalny opis projektowanej aplikacji. Zostanie również przedstawiony dokładny projekt algorytmu, który został zaimplementowany w programie.

4.1. Diagram klas aplikacji



Rys. 4.1. Diagram klas utworzony za pomocą [11]

4.2. Algorytm

W aplikacji wykorzystano dwa algorytmy będące modyfikacjami algorytmów Pan-Tompkins, przedstawionego w rozdziale 2.3, i Laguna-Thakor z rozdziału 2.4.

4.2.1. Preprocessing

Pierwsze 3 etapy preprocessingu są wspólne dla implementacji obu algorytmów:

1. Filtracja górnoprzepustowa – pozwala na pozbycie się składowych o niskich częstotliwościach jak np. dryf linii izoelektrycznej wywołany respiracją. Zbyt wysoka częstotliwość odcięcia może spowodować zaburzenie załamek P, T i odcinków między załamekami. Zastosowano filtr górnoprzepustowy Butterwortha, rzędu 2, o częstotliwości odcięcia $f_c = 1\text{Hz}$.
2. Filtracja dolnoprzepustowa – eliminuje szумы o wysokich częstotliwościach takie jak zakłócenia mięśniowe i zakłócenia z sieci (50 lub 60 Hz). Niska częstotliwość odcięcia może prowadzić do wytłumienia załamek Q, R i S. Zastosowano filtr dolnoprzepustowy Butterwortha, rzędu 3, o częstotliwości odcięcia $f_c = 20\text{Hz}$.
3. Różniczkowanie – służy zdobyciu informacji na temat nachyleń załamek, np. załamek R cechuje się największym nachyleniem zbocza. Różniczkowanie przybliżono za pomocą gradientu.

Kolejne 2 etapy preprocessingu wykorzystane zostały jedynie w implementacji algorytmu Pan-Tompkins.

4. Potęgowanie – zgodnie z instrukcjami w 2.3.4 sygnał podnoszony jest do kwadratu w celu wzmocnienia sygnałów o wysokich częstotliwościach.
5. Sumowanie w ruchomym oknie – zgodnie z instrukcją w 2.3.4 zastosowano okno o szerokości 150 ms.

4.2.2. Algorytm Pan-Tompkins.

Poszukiwanie kompleksu R odbywa się zgodnie z instrukcjami przedstawionymi w 2.3.6 – stosując próg adaptacyjny zarówno na sygnale po filtracji jak i po całkowaniu w ruchomym oknie. Załamek zostaje uznany za R jeżeli zostanie tak oznaczony w obu sygnałach.

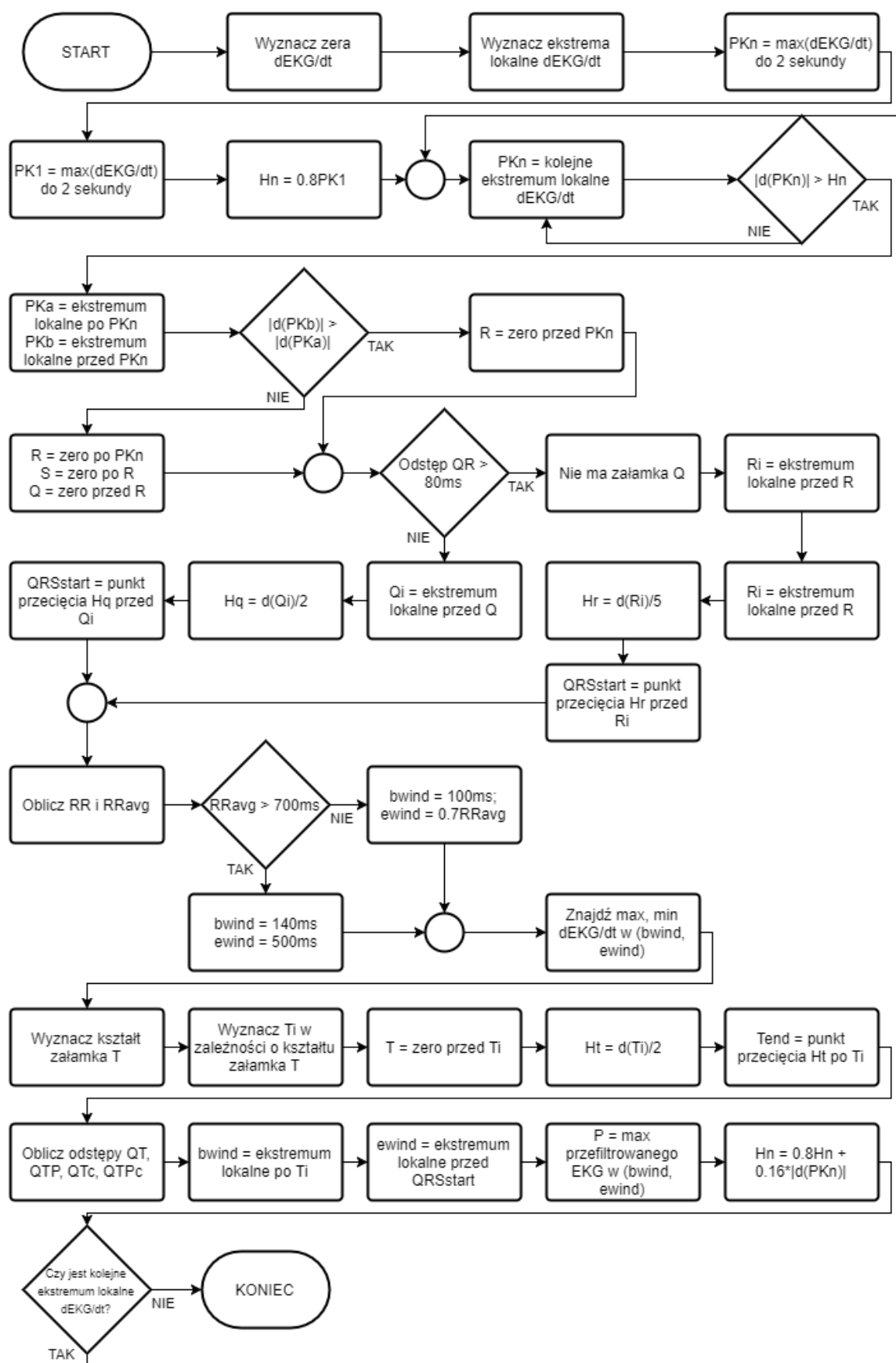
4.2.3. Algorytm Laguna-Thakor.

Algorytm przedstawiony w rozdziałach od 2.4.3 do 2.4.8 został rozbudowany o kroki pozwalające na wykrywanie załamka P. Przedstawia się następująco:

1. Wyznaczenie zer i lokalnych ekstremów sygnału zróżniczkowanego – wg. 2.4.3. Wyznaczamy zera sygnału zróżniczkowanego oraz ekstrema lokalne (szczyty) sygnału zróżniczkowanego, po których iterowany będzie algorytm.
2. Detekcja załamek Q, R i S – wg. 2.4.4.
3. Obliczanie odstępu RR – wg. 2.4.5. Odstęp RR jest liczony od momentu wykrycia drugiego załamka R.

4. Detekcja początku QRS – wg. 2.4.6.
5. Detekcja załamka T – wg. 2.4.7. Wykrywany od momentu wyliczenia pierwszego odstępu RR.
6. Obliczenie odstępu QT – wg. 2.4.8. Obliczane od momentu wykrycia pierwszego załamka T.
7. Detekcja załamka P – wykrywany od momenty wykrycia pierwszego załamka T. Ustalamy okno poszukiwań (bwind, ewind) od pierwszego ekstremum lokalnego sygnału zróżniczkowanego po T_{end} (bwind) do ostatniego ekstremum lokalnego sygnału zróżniczkowanego przed QRS_{start} (ewind). P to punkt, w którym sygnał przefiltrowany osiąga maksimum wewnątrz okna (bwind, ewind).

Schemat blokowy prezentujący zastosowany algorytm przedstawia Rys. 4.2.



Rys. 4.2. Diagram klas utworzony za pomocą [11]

4.3. Opis klas i pakietów

Program zawiera dwie bardzo istotne klasy z punktu widzenia funkcjonalności projektu i jedną klasę pomocniczą. Aplikacja została podzielona na dwa moduły reprezentowane w postaci klas:

- Klasa ECGAnalyzer – jest to klasa reprezentująca stronę obliczeniową aplikacji. Zostały w niej zaimplementowane funkcje służące do preprocessingu sygnału EKG, a także funkcje implementujące algorytmy wyznaczania załamków. Klasa ta może być z powodzeniem importowana do innego oprogramowania pełniąc rolę backendu.
- Klasa ECGDetectionSoftware – klasa reprezentująca frontend aplikacji. Pełni wyłącznie rolę interfejsu dla modułu obliczeniowego. Pozwala na prezentację wyników w postaci wykresów oraz tabel, a także wczytywanie i zapisywanie danych EKG.
- Ostatnią klasą jest klasa pomocnicza Spectrum służąca do przechowywania informacji na temat widma amplitudowego sygnału.

W programie wykorzystywanych jest 6 przydanych zewnętrznych pakietów dla Pythona 3.6:

- scipy w wersji 1.0.0 – moduł służący do zaawansowanych obliczeń matematycznych i przetwarzania sygnałów,
- matplotlib w wersji 2.1.0 – służy tworzeniu wykresów,
- numpy w wersji 1.13.3 – moduł rozbudowuje struktury danych Pythona o bardziej wydajne tablice i inne narzędzia matematyczne,
- wfdb w wersji 1.3.5 – obsługa bazy danych PhysioNet,
- peakutils w wersji 1.1.0 – do znajdowania ekstremów lokalnych w liście,
- ttkthemes w wersji 1.5.2 – rozbudowane motywy graficzne do Tkintera.

GUI został napisany z wykorzystaniem modułu Tkinter będącego częścią biblioteki standardowej Pythona.

4.4. Baza danych PhysioNet

Sygnały EKG, na których testowane było działanie algorytmów, pochodzą z internetowej bazy danych PhysioNet [12]. Udostępnia ona szeroki zbiór zarejestrowanych sygnałów fizjologicznych, np. EKG, EEG i EMG. Wybrano dane pacjentów udostępnione Physikalisch-Technische Bundesanstalt (PTB), niemiecki narodowy instytut miar. Sygnały zostały pobrane urządzeniem o parametrach [13]:

- 16 kanałów (14 kanałów EKG, 1 do respiracji, 1 do linii napięcia),
- Napięcie wejściowe: $\pm 16\text{mV}$, kompensacja napięcia niezrównoważenia do $\pm 300\text{mV}$,
- Rezystancja wejściowa: $100\Omega(\text{DC})$,
- Rozdzielczość: 16 bitów, $0.5\text{ }\mu\text{V/LSB}$,
- Pasma: 0 – 1 kHz.

Dane przechowywane w bazie PhysioNet może pobrać za pomocą modułu wfdb-python [14] do języka Python. Pobrany sygnał jest w formacie MIT, a moduł wfdb-python pozwala na jego odczytanie.

5. IMPLEMENTACJA

5.1. Implementacja algorytmów

Poniższy rozdział zawiera opis końcowego efektu implementacji algorytmów wraz z najciekawszymi fragmentami kodu.

5.1.1. Preprocessing

Do przeprowadzenia analizy potrzebny jest jedynie zapis czystego sygnału EKG i informacja o częstotliwości próbkowania sygnału. Pierwszym krokiem jest wyznaczenie wartości sygnału, częstotliwości próbkowania f_s , różnicy czasu pomiędzy próbkami dt , częstotliwości Nyquista, długości sygnału, osi czasu i parametrów preprocessingu.

```
self.fs = fs # Hz
self.ecg_signal = signal

self.dt = 1 / self.fs
self.nyquist_f = self.fs/2 # Hz
self.lowpass_fc = 20 # Hz
self.highpass_fc = 1 # Hz

self.lowpass_order = 3
self.highpass_order = 2
self.integration_window_length = int(0.15 * self.fs)

self.length = self.ecg_signal.__len__()
self.t = np.arange(self.length) / self.fs
```

Listing 5.1. Inicjalizowanie zmiennych sygnału

Następnie szeregowo są wykonywane kolejne etapy preprocessingu: filtracja górno- i dolnoprzepustowa, różniczkowanie, potęgowanie i całkowanie w oknie ruchomym. W celach analizy przygotowano także funkcję pozwalającą na stworzenie obiektu reprezentującego widmo amplitudowe danego sygnału. Funkcje `_lowpass()` i `_highpass()` wykorzystują funkcję `butter()` z pakietu `scipy.signal` w celu wyliczenia parametrów filtru Butterwortha, a następnie funkcję `filtfilt()`, która stosuje podany filtr na sygnale dwukrotnie – w przód i w tył. Efektem jest filtracja bez opóźnienia sygnału. Dodatkowo po każdym etapie preprocessingu sygnał wyjściowy jest normalizowany – odpowiedzialny jest za to dekorator `@normalized` z pliku `signalutils.py`. Plik `signalutils.py` jest zbiorem kilku przydatnych funkcji:

- `normalize()` – normalizowanie funkcji,
- `normalized()` – dekorator normalizujący sygnał zwracany przez dekorowaną funkcję,
- `find_zeros()` – zwraca tablicę zawierającą indeksy punktów przecięcia zera przez dany sygnał,

- `samples_to_ms()` – konwersja próbek na milisekundy,
- `ms_to_samples()` – konwersja milisekund na próbki.

```
@normalized
def _lowpass(self, signal):
    b, a = butter(self.lowpass_order, self.lowpass_fc / self.nyquist_f)
    return filtfilt(b, a, signal)

@normalized
def _highpass(self, signal):
    b, a = butter(self.highpass_order, self.highpass_fc / self.nyquist_f, 'highpass')
    return filtfilt(b, a, signal)

@normalized
def _derivative(self, signal):
    return np.gradient(signal)

@normalized
def _squaring(self, signal):
    return signal**2

@normalized
def _moving_window_integral(self, signal):
    window = self.integration_window_length + 1
    b = np.ones(window) / (window)
    a = [1]
    f = fft.rfftfreq(self.length, self.dt)
    w, gd = group_delay((b, a), f)
    shift = -np rint(np.mean(gd)).astype(int)
    return np.roll(lfilter(b, a, signal), shift)

def spectrum(self, signal):
    return Spectrum(signal, self.t, self.length, self.dt)
```

Listing 5.2. Funkcje preprocessingu

5.1.2. Algorytm Pan-Tompkins

Implementację algorytmu Pan-Tompkins przedstawia Listing 5.3. Funkcja powinna być wywoływana dwukrotnie: raz przyjmując argument `signal` w postaci sygnału po filtracji górno- i dolnoprzepustowej, a następnie przekazując do funkcji sygnał po całkowaniu w ruchomym oknie. Po zakończeniu obliczeń funkcja zwraca dwie tablice: indeksy szczytów załamekó uznanych za R i indeksy szczytów uznanych za szum.

Niestety implementacja tego algorytmu nie przyniosła oczekiwanych efektów. Punkty wyznaczone jednocześnie przy obu uruchomieniach funkcji (z sygnałem po filtracji i sygnałem po całkowaniu) powinny zostać uznane za R. Niestety sygnał po całkowaniu był nieznacznie przesunięty względem sygnału po filtracji (szczyty się nie pokrywały). Niepewna jest przyczyna takiego wyniku, najprawdopodobniej błąd leży po stronie preprocessingu, np. przy niwelowaniu opóźnień sygnału lub doborze długości okna całkowania. Możliwe jest też złe zrozumienie działania algorytmu. Przegląd implementacji algorytmu Pan-Tompkins [15] [16] [17] wykazał, że ostatni etap algorytmu – detekcja adaptacyjnym progiem, jest często realizowana w inny sposób niż zaproponowany w oryginalnym artykule. Niestety zabrakło czasu na dogłębsze zbadanie tematu i zarówno w pracy jak, i w aplikacji

skupiono się na działającym algorytmie Laguna-Thakor. W związku z tym w dalszej części pracy algorytm Pan-Tompkins będzie pomijany, a aplikacja nie wyświetla wyników detekcji tym algorytmem.

```
def _detection_pan_tompkins(self, signal):
    peaks = peakutils.indexes(signal, thresh=0)
    running_estimate_signal_peak = 0
    running_estimate_noise_peak = 0
    first_peak_threshold = 0

    qrs_indices = np.empty(0, dtype=int)
    noise_indices = np.empty(0, dtype=int)

    for peak_idx, peak_val in zip(peaks, signal[peaks]):
        if peak_val > first_peak_threshold:
            qrs_indices = np.append(qrs_indices, peak_idx)
            running_estimate_signal_peak = 0.125 * peak_val + 0.875 *
running_estimate_signal_peak
        else:
            noise_indices = np.append(noise_indices, peak_idx)
            running_estimate_noise_peak = 0.125 * peak_val + 0.875 *
running_estimate_noise_peak
            first_peak_threshold = running_estimate_noise_peak \
+ 0.25 * (running_estimate_signal_peak -
running_estimate_noise_peak)
    return qrs_indices, noise_indices
```

Listing 5.3. Implementacja algorytmu Pan-Tompkins

5.1.3. Algorytm Laguna-Thakor

Implementacja algorytmu Laguna-Thakor jest znacznie dłuższa i bardziej złożona niż pierwszego algorytmu. Argumentem przekazywanym do funkcji `_detection_laguna_thakor()` powinna być tablica zawierająca wartości sygnału EKG po filtracji pasmowoprzepustowej i różniczkowaniu. Wewnątrz funkcji wykorzystywane są także wartości sygnału po samej filtracji.

Pierwszym krokiem jest wyznaczenie ekstremów lokalnych (pików) i zer w sygnale, obliczenie inicjalizującej wartości adaptacyjnego progu oraz inicjalizacja tablic asocjacyjnych (słowników), do których będą zapisywane wykryte załamki i obliczone parametry.


```

def _detection_laguna_thakor(self, signal):
    peak_indices = peakutils.indexes(np.abs(signal), thresh=0)
    zero_indices = find_zeros(signal)
    peak_n = max(abs(signal[:self.fs*2]))
    threshold_n = 0.8 * peak_n
    RR_interval_avg = None

    indices = {
        'Q': np.empty(0, dtype=int),
        'R': np.empty(0, dtype=int),
        'S': np.empty(0, dtype=int),
        'QRS_start': np.empty(0, dtype=int),
        'P': np.empty(0, dtype=int),
        'T': np.empty(0, dtype=int),
        'T_end': np.empty(0, dtype=int),
    }

    intervals = {
        'RR': np.empty(0, dtype=int),
        'QT': np.empty(0, dtype=int),
        'QTP': np.empty(0, dtype=int),
        'QTc': np.empty(0, dtype=int),
        'QTPc': np.empty(0, dtype=int)
    }

    T_shapes = np.empty(0, dtype=str)

```

Listing 5.4. Etap inicjalizacji parametrów w funkcji implementującej algorytm Laguna-Thakor

Następnie rozpoczęte jest wykonywanie się pętli for, która pobiera kolejne indeksy zawarte w tablicy `peak_indices` i odpowiadającym im wartości sygnału zróżniczkowanego. Jeżeli wartość bezwzględna szczytu przekracza próg dokonujemy detekcji R i kolejnych załamek, a także aktualizujemy wartość progu. Kolejno wykonywane są funkcje odpowiedzialne za:

- Detekcję Q, R i S – funkcja `_LT_QRS_detect()`,
- Obliczenie odstępu RR – funkcja `_LT_RR_interval_calc()`,
- Detekcję początku QRS – funkcja `_LT_QRS_start_detect()`,
- Detekcję P – funkcja `_LT_P_detect()`,
- Detekcję T – funkcja `_LT_T_detect()`,
- Obliczenie odstępów QT – funkcja `_LT_intervals_calc`.

W większości funkcji wykorzystujemy funkcję `bisect` z pakietu `bisect` biblioteki standardowej Pythona, która jest implementacją algorytmu bisekcyjnego. Jest to bardzo szybka metoda poszukiwania numeru indeksu danej wartości w posortowanej tablicy o złożoności obliczeniowej $O(\log n)$. Używana jest do znajdowania najbliższego ekstremum lub punktu przecięcia zera od załamka lub innego punktu odniesienia. Przykładem jest funkcja detekcji P, przedstawiona w Listingu 5.5, gdzie wyszukujemy pierwszego ekstremum po punkcie `T_end` i ostatniego ekstremum przed punktem `QRS_start`.

```

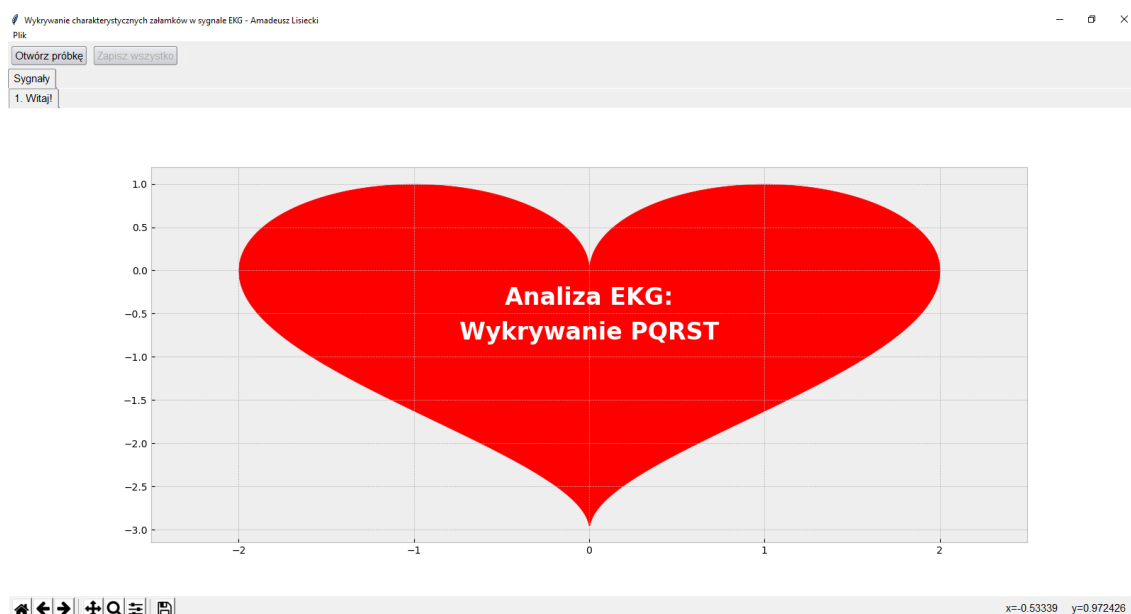
def _LT_P_detect(self, indices, QRS_start_idx, zero_indices, peak_indices,
signal):
    last_T_end = indices['T_end'][-1]
    bwind_idx = peak_indices[bisect.bisect_right(peak_indices, last_T_end)]
    ewind_idx = peak_indices[bisect.bisect_right(peak_indices, QRS_start_idx) - 1]
    if bwind_idx >= len(signal) or ewind_idx >= len(signal) or bwind_idx ==
ewind_idx:
        return None
    P_idx = bwind_idx + np.argmax(self.filtered_ecg_signal[bwind_idx:ewind_idx])
    return P_idx

```

Listing 5.5. Etap inicjalizacji parametrów w funkcji implementującej algorytm Laguna-Thakor

5.2. Implementacja GUI

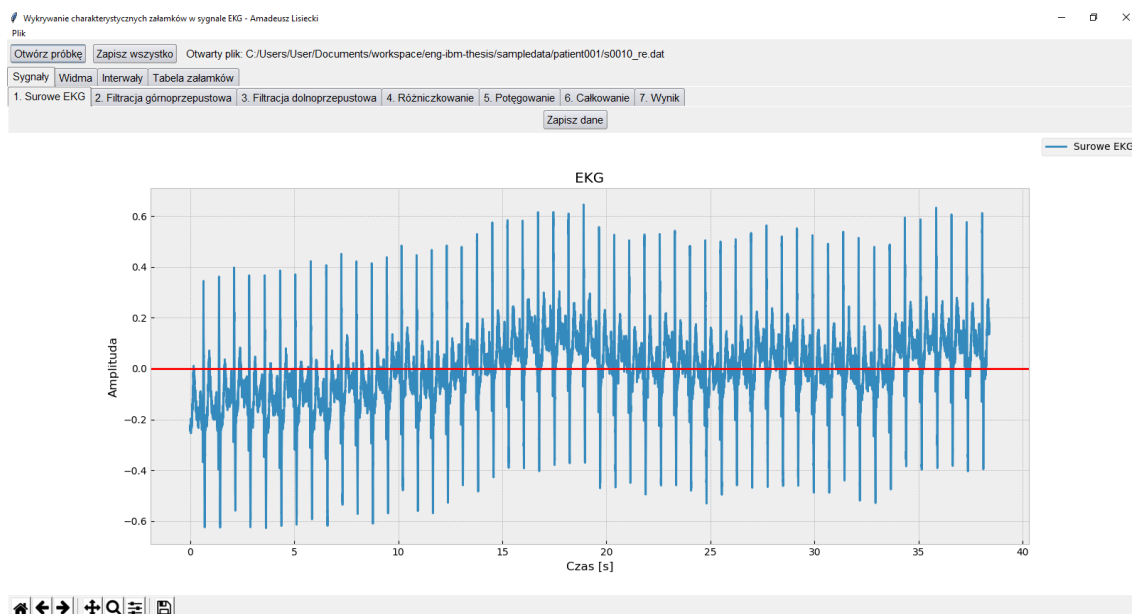
Interfejs graficzny został zaimplementowany za pomocą klasy App, w skrypcie ECGDetectionSoftware.py, który jest także skryptem uruchomieniowym całej aplikacji. Po uruchomieniu aplikacji wyświetla się ekran powitalny i istnieje możliwość wczytania sygnału.



Rys. 5.1. Ekran powitalny aplikacji

Następnie możliwe jest wczytanie sygnału z pliku o rozszerzeniu .DAT (format MIT bazy danych PhysioNet) albo .CSV przy pomocy przycisku „Otwórz próbkę”. W pierwszym przypadku wykorzystany zostaje moduł wfdb-python do sparsowania pliku i odczytania wartości sygnału i częstotliwości próbkowania. Pliki o rozszerzeniu .CSV muszą przedstawiać dane w jednej kolumnie – pierwszy wiersz zawierający częstotliwość próbkowania, a kolejne wartości sygnału. W przeciwnym wypadku aplikacja wyświetli okno dialogowe z informacją o błędzie.

W momencie wyboru pliku dokonują się automatyczne obliczenia i detekcja załamków – po kilku sekundach na ekranie pojawia się wykres przedstawiający wczytany sygnał EKG oraz szereg zakładek.



Rys. 5.2. Ekran po wczytaniu sygnału EKG

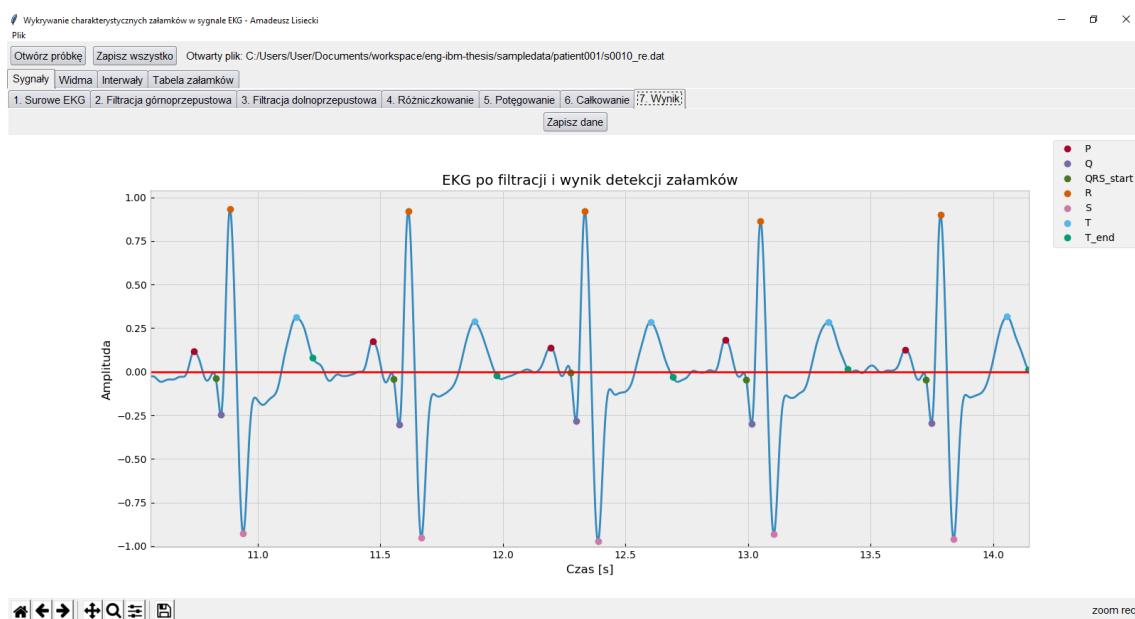
Poza przyciskiem „Otwórz próbkę” aktywny staje się przycisk „Zapisz wszystko” – pozwala on na zapisanie do pliku .CSV wszystkich wyników detekcji i analizy sygnału. Dodatkowo na ekranie pojawiają się dwa poziomy zakładki. Górny rząd to zakładki „Sygnały” – zawierająca wykresy sygnałów, „Widma” – z wykresami widm amplitudowych, „Interwały” – przedstawiająca wykresy i tablicę obliczonych odstępów i „Tablica załamek” – w której znajdują się sekwencje indeksów wykrytych załamek. Drugi rząd to szczegółowe widoki dostępne w ramach każdego z wymienionych wyżej działów. Szczegółowa struktura aplikacji z podziałem na zakładki wygląda następująco:

- „Sygnały” – wykresy efektów kolejnych etapów preprocessingu:
 - „1. Surowe EKG” – wykres wczytanego sygnału EKG,
 - „2. Filtracja górnoprzepustowa” – wykres EKG po filtracji górnoprzepustowej,
 - „3. Filtracja dolnoprzepustowa” – wykres EKG po filtracji górno- i dolnoprzepustowej,
 - „4. Różniczkowanie” – wykres EKG po filtracji górno- i dolnoprzepustowej oraz różniczkowaniu,
 - „5. Potęgowanie” – wykres EKG po filtracji, różniczkowaniu i potęgowaniu,
 - „6. Całkowanie” – wykres EKG po filtracji, różniczkowaniu, potęgowaniu i całkowaniu,
 - „7. Wyniki” – wykres EKG po filtracji i wyniki detekcji załamek.
- „Widma” – wykresy widm amplitudowych sygnału EKG przed i po filtracji:
 - „1. Surowe EKG” – wykres widma amplitudowego wczytanego EKG,
 - „2. Filtracja” – wykres widma amplitudowego po filtracji.
- „Interwały” – wykresy i tabela obliczonych odstępów RR i QT:
 - „Wykres RR” – wykres obliczonych odstępów RR,
 - „Wykres QT” – wykres obliczonych odstępów QT, QTP, QTc, QTPc,

- „Tabela” – tabela zawierająca wszystkie obliczone wartości odstępów.
- „Tabela załamków” – tabela zawierająca indeksy wszystkich wyznaczonych załamków oraz informacje na temat wykrytej morfologii załamka T.

Jeżeli w danej zakładce widoczny jest przycisk „Zapisz dane” to możliwe jest za jego pomocą zapisanie do pliku .CSV danych widocznych w danej zakładce. Dodatkowo wszystkie wykresy są interaktywne – można je dowolnie skalować i przesuwać, a także zapisywać do pliku.

Przykładowe efekty detekcji przedstawione na wykresie widoczne są, na Rys. 5.3.



Rys. 5.3. Ekran aplikacji przedstawiający wyniki detekcji załamków PQRST

5.3. Podsumowanie

Ostatecznie implementacja składa przedstawia się w postaci aplikacji posiadającej dwie warstwy: warstwę obliczeniową, w której zaprogramowano dwa algorytmy detekcji, oraz warstwę interfejsu użytkownika. Moduł obliczeniowy może być teoretycznie importowany i wykorzystywany w innym oprogramowaniu lub całkowicie niezależnie. Niestety nie udało się zaimplementować dwóch działających algorytmów – z powodów wymagających odrębnej analizy algorytm Pan-Tompkins okazał się być niefunkcjonalny. Natomiast drugi algorytm, Laguna-Thakor, po pewnych modyfikacjach wydaje się dawać dobre wyniki detekcji. Efekty testów sprawności tej implementacji będą opiane w kolejnym rozdziale.

6. TESTY

W tym rozdziale zostaną przedstawione wyniki wszelkich testów sprawdzających sprawność wytworzonej aplikacji. Przygotowano szereg scenariuszy testów funkcjonalnych. Następnie sprawdzona zostanie skuteczność detekcji na losowej próbie. Ostatnim zadaniem będzie sprawdzenie odporności algorytmu na różne rodzaje szumu i artefaktów.

6.1. Testy funkcjonalne

6.1.1. Scenariusz testowy nr 1

Nazwa: Wczytanie pliku z sygnałem EKG.	
Akcja użytkownika:	Efekt:
1. Wciśnięcie przycisku „Otwórz próbkę”.	1. Wyświetlenie się okna dialogowego wyboru pliku.
2. Wybranie formatu pliku „dat”.	2. Wyświetlenie jedynie plików w formacie „dat”.
3. Wybranie pliku.	3a. Zamknięcie okna dialogowego. 3b. Przeprowadzenie detekcji. 3c. Wyświetlenie wyników.
Wynik:	Sukces

Tabela 6.1. Scenariusz testowy nr 1

6.1.2. Scenariusz testowy nr 2

Nazwa: Przegląd wyników analizy sygnału EKG.	
Akcja użytkownika:	Efekt:
1. Wczytanie pliku	1. Wyświetlenie wyników analizy.
2. Wybranie zakładki „7. Wynik”	2. Wyświetlenie wykresu z wynikiem detekcji załamków.
3. Kliknięcie ikony lupy w lewym dolnym rogu wykresu.	3. Zmiana kursora na lupę w obrębie wykresu.
4. Wykreślenie myszką prostokąta na wykresie.	4. Powiększenie zaznaczonego prostokąta na cały obszar wykresu.
Wynik:	Sukces

Tabela 6.2. Scenariusz testowy nr 2

6.1.3. Scenariusz testowy nr 3

Nazwa: Zapis wyników do pliku	
Akcja użytkownika:	Efekt:
1. Wczytanie pliku	1. Wyświetlenie wyników analizy.
2. Kliknięcie przycisku „Zapisz wszystko”.	2. Wyświetlenie okna dialogowego wyboru pliku.
3. Wybranie pliku.	3a. Zamknięcie okna dialogowego. 3b. Zapisanie wyników do pliku o podanejścieżce, w formacie CSV.
Wynik:	Sukces

Tabela 6.3. Scenariusz testowy nr 3

6.1.4. Scenariusz testowy nr 4

Nazwa: Zapis wyników w nieprawidłowym katalogu	
Akcja użytkownika:	Efekt:
1. Wczytanie pliku	1. Wyświetlenie wyników analizy.
2. Kliknięcie przycisku „Zapisz wszystko”.	2. Wyświetlenie okna dialogowego wyboru pliku.
3. Wybranie pliku w katalogu, do którego użytkownik nie ma uprawnień.	3a. Zamknięcie okna dialogowego. 3b. Wyświetlenie okna dialogowego z komunikatem o błędzie. 3c. Powrót do aktywnej zakładki.
Wynik:	Sukces

Tabela 6.4. Scenariusz testowy nr 4

6.2. Analiza otrzymanych wyników

W celu przetestowania wyników algorytmu detekcji wylosowane zostanie 10 przebiegów z badania elektrokardiografem u 10 różnych pacjentów. Sygnał EKG będzie pochodził z odprowadzenia I. Następnie w każdym z tych sygnałów zostanie wylosowana sekunda, od której zostanie sprawdzona jakość wyników przy kolejnych 10 uderzeniach serca. Dla wyników testu istotne jest jedynie w pełni poprawne lub niepoprawne wyznaczenie wierzchołka załmków P, Q, R, S i T. Losowe próbki zostaną pobrane z bazy PTB [13], zawierającej 549 nagrań od 290 pacjentów:

- Wiek 17-87 lat,
- Średnia wieku 57,2 lata,
- 209 mężczyzn, średnia wieku 55,5 lata,
- 81 kobiet, średnia wieku 61,6 lat.

	Liczba wykryć załamek na to 10 możliwych					
Numer pacjenta	P	Q	R	S	T	Skuteczność
196	0	7	10	2	8	54%
147	8	10	10	10	9	94%
30	10	10	10	3	10	86%
293	5	10	10	10	6	82%
133	0	10	10	10	9	78%
119	9	10	10	10	10	98%
159	10	10	10	10	10	100%
260	2	10	10	10	10	84%
87	3	10	10	10	9	84%
64	9	10	10	10	5	88%
Skuteczność	56%	97%	100%	85%	86%	84,8%

Tabela 6.5. Skuteczność detekcji załameków

Zestawiając wyniki analizy 10 sygnałów otrzymaliśmy 84,8% skutecznych detekcji – tzn. wykryto 84,8% możliwych załameków w prawidłowych miejscach. Za każdym razem udało się wykryć wszystkie załamki R w prawidłowym miejscu, a w przypadku pacjenta nr 159 wykryto wszystkie poszukiwane załamki w badanym przedziale czasu. Najgorsze parametry uzyskała detekcja załamek P – jest to niewielki załamek, o niskiej częstotliwości, a metoda jego detekcji była najmniej wyszukana. Wynik ten więc nie dziwi – mocniej zaszumione sygnały wykazywały bardzo zmienny sygnał po różniczkowaniu (różniczkowanie wzmacnia wysokie częstotliwości) co stwarzało problemy zarówno przy detekcji załamek T jak i P. Bardzo dobre efekty uzyskano przy detekcji kompleksu QRS, zwłaszcza że przy załamku S algorytm nie przewiduje żadnych wyjątkowych metod, a sygnał z badania pacjenta nr 30 wykazywał brak załamek S. W tym przypadku był on zasłonięty przez odwrócony załamek T – taką morfologię udało się również zauważyć u pacjenta numer 293. Algorytm poprawnie ją sklasyfikował.

6.3. Wpływ zakłóceń na skuteczność detekcji

Znając już ogólną skuteczność algorytmu warto zbadać jaki wpływ na analizę mają konkretne rodzaje zakłóceń. W tym celu zapisano jeden z przefiltrowanych sygnałów EKG, w którym wykrywano 100% załameków. Następnie wygenerowano pewnego rodzaju szum, który miał wpłynąć negatywnie na analizę sygnału. Zastosowano biały szum - wartości sygnału były losowane z rozkładu normalnego. Biały szum, który można wygenerować korzystając również z innych rozkładów niż normalny, ma pewną wyjątkową właściwość – widmo jego mocny jest płaskie. Tzn., że ma niemal jednakową moc w każdym paśmie częstotliwości.

	Liczba wykryć załamek na to 10 możliwych					
Odchylenie stand.	P	Q	R	S	T	Skuteczność
0.1	10	10	10	10	10	100%
0.2	10	10	10	10	8	96%
0.3	9	10	10	10	6	90%
0.4	9	10	10	10	4	86%
0.5	6	9	9	9	5	76%
0.6	0	0	0	0	0	0%
0.7	0	0	0	0	0	0%
0.8	5	7	0	0	1	26%
0.9	1	5	5	5	3	38%
1.0	0	2	2	2	0	12%
Skuteczność	50%	63%	56%	56%	37%	262

Tabela 6.6. Skuteczność detekcji załameków po wprowadzeniu białego szumu

Powyższą tabelę wygenerowano za pomocą kodu przedstawionego w Listingu 6.1. i przedstawia ona procent wykrytych załameków w zależności od tego jaki szum dodano do sygnału – zastosowano rozkład normalny o średniej równej 0 i odchyleniu standardowym rosnącym od 0.1 do 1. Biorąc pod uwagę, że surowy sygnał EKG, który poddany zaszumieniu miał maksymalną bezwzględną wartość 1 to wynik 86% skuteczności przy odchyleniu standardowym na poziomie 0.4 jest zadawalający. W przypadku tego testu największe problemy algorytmowi sprawiała detekcja załamek T. Jest to załamek o niższej częstotliwości niż QRS, a metoda jego wyznaczania jest bardzo czuła na zaszumienie sygnału różniczkowanego. W pewnych wypadkach algorytm nawet nie jest w stanie rozpocząć detekcji – tak było przy sygnałach dla odchylen standardowych 0,6 i 0,7.

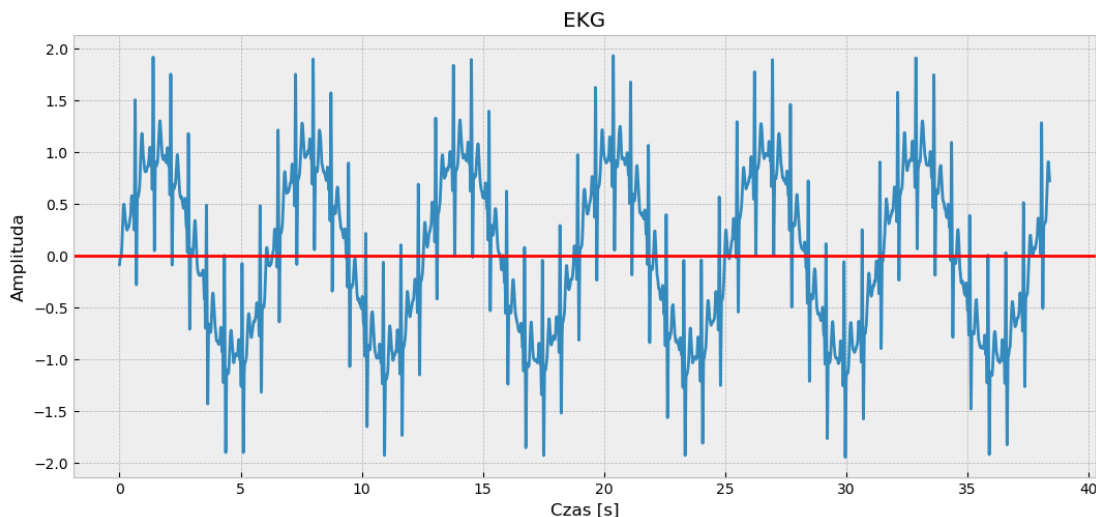
```
import numpy as np
import os

with open('testy/test2/raw.csv') as f:
    sample = np.fromstring(f.read(), dtype=np.float64, sep=os.linesep)

fs = sample[0]
y = sample[1:]
N = len(y)
t = np.arange(N)/fs
noise = np.random.normal(0, 0.01, N)
s = {}
for i in range(1, 11):
    s[i] = y + np.random.normal(0, 0.1*i, N)
    with open('testy/test2/gaussian{}.csv'.format(i), 'w') as f:
        f.write(str(fs)+'\n')
        f.writelines(list(map(lambda x: str(x)+'\n', s[i])))
```

Listing 6.1. Kod generujący biały szum

Poza białym szumem postanowiono również sprawdzić odporność algorytmu na dryf linii bazowej sygnału. W tym celu dodano do sygnału składową sinusoidalną o amplitudzie 1 i niskiej częstotliwości $f=1/2\pi$. Filtry były w stanie w pełni wytłumić tę składową, tak że efekt dryfu był niezauważalny i wykryto wszystkie załamki ze 100% skutecznością.



Rys. 6.1. Sygnał EKG z dodanym komponentem o dużej amplitudzie i niskiej częstotliwości.

6.4. Podsumowanie

W tym rozdziale udało się przeprowadzić podstawowe czynności sprawdzające sprawność aplikacji pod względem funkcjonalnym, a także jakości działania algorytmu. Testy wykazały mocne i słabe strony zastosowanych rozwiązań, m.in. niedoskonałość metod wyznaczania załamków P i T, ale też skuteczność w wykrywaniu kompleksu QRS z dużą odpornością na szum i inne zakłócenia.

7. PODSUMOWANIE

Efektem mojej pracy jest zbudowanie funkcjonalnej aplikacji pozwalającej na wyznaczenie z dużą pewnością lokalizacji charakterystycznych załamków w przebiegu EKG. Do tego została napisana pełna dokumentacja w postaci tej pracy, w której przeanalizowano teoretyczne zagadnienia stojące za analizą sygnału EKG, wyróżniono rodzące problemy kwestie, postarano się je rozwiązać lub przynajmniej zniwelować ich efekt. Zbadano również wpływ szumu na efekty pracy aplikacji.

Niestety pomimo zaimplementowania działającego algorytmu Laguna-Thakor nie udało się uzyskać podobnych wyników z wykorzystaniem metody zaproponowanej Jiau Pana i Willisa Tompkinsa. Czas poświęcony na próby modyfikacji tego algorytmu nie poszedł na marne, ponieważ przedstawiona tam wiedza na temat charakteru sygnału EKG oraz propozycje dotyczące etapu preprocessingu zostały wykorzystane do usprawnienia działania algorytmu Laguna-Thakor.

Warto zwrócić uwagę na mocne i słabe strony skonstruowanego na rzecz aplikacji algorytmu. Kompleks QRS nie stanowił dla niego problemu, a wsparcie tego rozwiązania dodatkowymi usprawnieniami, jak choćby lepsze zaprojektowanie filtru, może jeszcze poprawić jego działanie. Kompleks QRS jest w końcu najistotniejszym elementem analizowanym podczas pracy z sygnałem EKG. Co do wad zaprezentowanego rozwiązania można na pewno przedstawić metodę poszukiwania załamka P, a w pewnych warunkach również T. Na pewno konieczna jest staranna kontrola filtracji – na początku pracy z sygnałem EKG załamki P i T sprawiały mi wiele problemów, gdyż nieostrożnie zaprojektowany filtr zbyt mocno je wzmacniał lub wygładzał, a nawet zmieniał ich morfologię. Załamki R i S były mniej podatne na tego typu problemy. Warto też dodać, że w niektórych przebiegach załamek Q okazywał się niezwykle krótki, o wysokiej częstotliwości, i mógł zostać niepotrzebnie odfiltrowany. Wszystkie te aspekty są ze sobą ściśle powiązane, ponieważ najczęściej załamki wyznaczone są na bazie tych już wyznaczonych – np. chcąc poprawić skuteczność wyznaczania załamka Q możemy pogorszyć wykrywalność załamka P i vice versa.

Temat analizy EKG na pewno nie jest skończony – nie tylko można badać ten sygnał pod innym kątem, np. w poszukiwaniu innych załamków (np. U – pytanie o jego pochodzenie jest wciąż otwarte), ale także poszukiwać alternatywnych metod. Jedną z takich metod jest zastosowanie sieci neuronowych do rozpoznawania schematów i charakterystyk w powtarzającym się sygnale EKG. Warto również pracować nad analizą sygnału EKG „na żywo”, tj. w urządzeniach przenośnych (np. wearables), na bieżąco monitorując stan pacjenta lub konsumenta. Tego typu rozwiązania biomedyczne, pozwalające na natychmiastowe informowanie o stanie zdrowia, są z pewnością bardzo przyszłościowe.

WYKAZ LITERATURY

- [1] 30 11 2017. [Online]. Available: https://classconnection.s3.amazonaws.com/586/flashcards/812586/jpg/action_potential1340906012006.jpg.
- [2] R. E. Klabunde, „Cardiovascular Physiology Concepts - Electrocardiogram Leads,” 30 11 2017. [Online]. Available: <http://www.cvphysiology.com/Arrhythmias/A013>.
- [3] T. A. Traczyk W., Fizjologia człowieka z elementami fizjologii stosowanej i klinicznej, Warszawa: Wydawnictwo Lekarskie PZWL, 2001.
- [4] T. U. o. Nottingham, „Cardiology Teaching Package - The S Wave,” 30 11 2017. [Online]. Available: http://www.nottingham.ac.uk/nursing/practice/resources/cardiology/function/s_wave.php.
- [5] B. E., „ECG Basics,” LIFE IN THE FASTLANE, 30 11 2017. [Online]. Available: <https://lifeinthefastlane.com/ecg-library/basics>.
- [6] T. O. Chantal E. Conrath, „The patient U wave,” *Cardiovascular Research*, p. <https://doi.org/10.1016/j.cardiores.2005.05.027>, 1 8 2005.
- [7] M. A. A.-V. A. A.-V. F. Buendía-Fuentes, „High-Bandpass Filters in Electrocardiography: Source of Error in the Interpretation of the ST Segment,” *ISRN Cardiology*, p. doi:10.5402/2012/706217, 2012.
- [8] W. T. Jiapu Pan, „A Real-Time QRS Detection Algorithm,” 1985.
- [9] N. V. T. P. Laguna, „New algorithm for QT interval analysis in 24-hour Holter ECG: performance and applications,” *Med. & Biol. Eng. & Comput.*, pp. 67-73, January 1990.
- [10] H. C. Bazett, „An Analysis of the Time-Relations of Electrocardiograms,” *Heart*, tom 7, pp. 353-370, 1920.
- [11] [Online]. Available: <https://www.draw.io/>.
- [12] „PhysioNet,” [Online]. Available: <https://www.physionet.org>. [Data uzyskania dostępu: 30 11 2017].
- [13] „The PTB Diagnostic ECG Database,” [Online]. Available: <https://www.physionet.org/physiobank/database/ptbdb/>. [Data uzyskania dostępu: 30 11 2017].
- [14] M. L. f. C. Physiology, „Native Python WFDB package,” [Online]. Available: <https://github.com/MIT-LCP/wfdb-python>. [Data uzyskania dostępu: 30 11 2017].
- [15] B. Milner, „A real-time QRS detection algorithm for the Arduino platform - based on the Pan-Tompkins Algorithm,” [Online]. Available: https://github.com/blakeMilner/real_time_QRS_detection. [Data uzyskania dostępu: 30 11 2017].
- [16] F. Uysal, „QRS Detection Using Pan-Tompkins algorithm,” [Online]. Available: <https://archive.cnx.org/contents/611d4152-cf7f-4344-b56d-15775f92fbac@1/qrs-detection-using-pan-tompkins-algorithm>. [Data uzyskania dostępu: 30 11 2017].
- [17] D. Wedekind, „MATLAB - Pan-Tompkins QRS Detector,” [Online]. Available: <https://github.com/danielwedekind/qrsdetector>. [Data uzyskania dostępu: 30 11 2017].
- [18] A. B. Downey, Think DSP, O'REILLY, 2016.

[19] S. W. Smith, The Scientist & Engineer's Guide to Digital Signal Processing, California Technical Pub, 1997.

WYKAZ RYSUNKÓW

Rys. 2.1. Wykres zmiany potencjału błony komórkowej mięśnia sercowego [1]	11
Rys. 2.2. Odprowadzenia kończynowe Einthovena i Goldbergera [3]	12
Rys. 2.3. Umieszczeni odprowadzeń Wilsona na klatce piersiowej [3]	12
Rys. 2.4. Charakterystyczne załamki EKG [5]	14
Rys. 3.1. Diagram przypadków użycia utworzony za pomocą [11].....	24
Rys. 4.1. Diagram klas utworzony za pomocą [11]	25
Rys. 4.2. Diagram klas utworzony za pomocą [11]	28
Rys. 5.1. Ekran powitalny aplikacji	34
Rys. 5.2. Ekran po wczytaniu sygnału EKG	35
Rys. 5.3. Ekran aplikacji przedstawiający wyniki detekcji załamków PQRS.....	36
Rys. 6.1. Sygnał EKG z dodanym komponentem o dużej amplitudzie i niskiej częstotliwości.....	41

WYKAZ LISTINGÓW

Listing 5.1. Inicjalizowanie zmiennych sygnału.....	30
Listing 5.2. Funkcje preprocessingu	31
Listing 5.3. Implementacja algorytmu Pan-Tompkins	32
Listing 5.4. Etap inicjalizacji parametrów w funkcji implementującej algorytm Laguna-Thakor.....	33
Listing 5.5. Etap inicjalizacji parametrów w funkcji implementującej algorytm Laguna-Thakor.....	34
Listing 6.1. Kod generujący biały szum.....	41

WYKAZ TABEL

Tabela 6.1. Scenariusz testowy nr 1	37
Tabela 6.2. Scenariusz testowy nr 2	37
Tabela 6.3. Scenariusz testowy nr 3	38
Tabela 6.4. Scenariusz testowy nr 4	38
Tabela 6.5. Skuteczność detekcji załamków.....	39
Tabela 6.6. Skuteczność detekcji załamków po wprowadzeniu białego szumu	40