



Przedmiot: Dedykowane algorytmy diagnostyki medycznej.

ST\_ana

Temat projektu: Detekcja symptomów choroby niedokrwiennej na podstawie analizy interwału ST

Spis treści:

1.	ABSTRAKT .....	2
2.	WSTĘP .....	3
3.	KONCEPCJA PROPONOWANEGO ROZWIĄZANIA.....	6
3.1	ALGORYTM ZNAJDUJĄCY LINIĘ IZOELEKTRYCZNA.....	7
3.2	DETEKCJA PUNKTÓW J i ST.....	7
3.3	OBLICZANIE PARAMETRÓW DIAGNOSTYCZNYCH ODCINKA ST.....	8
3.4	DETEKCJA EPIZODÓW.....	8
3.5	ZAPISYWANIE WYNIKÓW.....	9
4.	REZULTATY I WNIOSKI.....	9
5.	PODSUMOWANIE .....	11
6.	LITERATURA.....	13
9.	DODATEK A. OPIS INFORMATYCZNY PROCEDUR.....	14

Wykonali: Jarosław Kardynia, Michał Piela

V rok IB\_EIM

konsultant: *dr hab. inż. Piotr Augustyniak*

Kraków, styczeń 2011.

## 1. Abstrakt

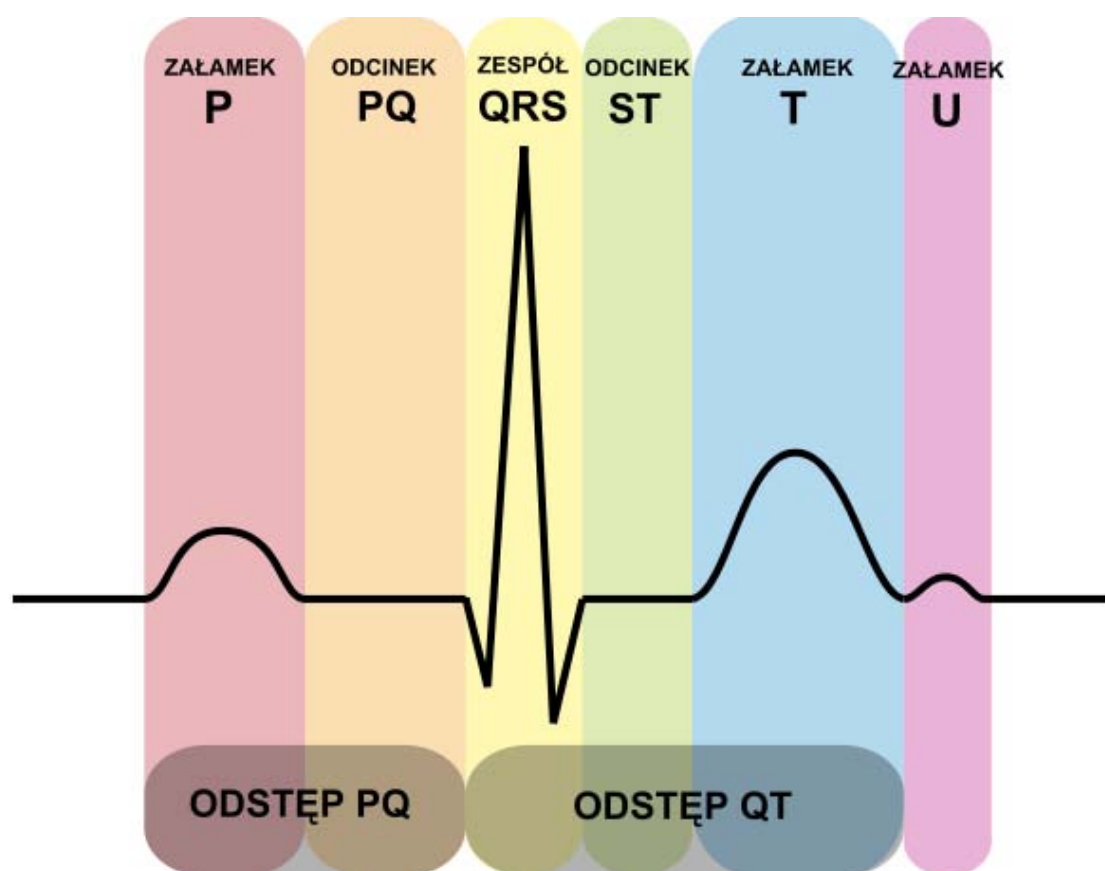
Projekt omawia zagadnienie automatycznej interpretacji elektrokardiogramu zarejestrowanego w komputerze. Celem pracy jest automatyczna analiza odcinka ST zarejestrowanego elektrokardiogramu oraz detekcja nieprawidłowości w jego obszarze. Program docelowo wspomaga lekarza, przedstawiając mu listę miejsc w sygnale, na które powinien zwrócić szczególną uwagę. Algorytm wyszukuje charakterystyczne, istotne diagnostycznie punkty przebiegu, które w następnych etapach są przedmiotem szczegółowej analizy. Ilościowa ocena maksymalnych obniżień i podniesień odcinka ST pozwala na wykrycie epizodów. Odbywa się ona po uprzednim wyznaczeniu amplitudy linii izoelektrycznej, względem której przeprowadzane są obliczenia. Każdy z wykrytych epizodów posiada szereg parametrów opisujących sytuację w jakiej wystąpił. Aplikacja spełnia wszystkie postawione założenia, odnajduje miejsca zmian patologicznych w elektrokardiogramie w obszarze odcinka ST.

## 2. Wstęp

Elektrokardiografia jest jedną z podstawowych metod diagnostycznych chorób układu krążenia a przede wszystkim serca. Polega ona na rejestracji czynności elektrycznej mięśnia sercowego. Do tego celu wykorzystuje się szereg elektrod przymocowanych do klatki piersiowej oraz kończyn pacjenta. Podczas skurczu serca dochodzi do wzbudzenia impulsu elektrycznego w mięśniu sercowym co może być rejestrowane na powierzchni skóry. Wartość powstałej różnicy potencjałów jest ściśle związana z fazą skurczu a wszelkie odstępstwa od wzorcowego (uznanego za prawidłowe) elektrokardiogramu może wskazywać na chorobę[1].

Elektrokardiogram składa się z: linii izoelektrycznej, załameków, odcinków i odstępów. Linia izoelektryczna rejestrowana jest w czasie, gdy w sercu nie ma żadnych pobudzeń. Załamki to wychylenia od linii izoelektrycznej. Odcinki oznaczają czas trwania linii izoelektrycznej pomiędzy załamekami a odstęp to łączny czas trwania odcinków i sąsiadującego załamka[1].

Celem niniejszego projektu jest stworzenie systemu automatycznej analizy jednego z najważniejszych, spośród elementów diagnostycznych elektrokardiogramu - odcinka ST. W diagnostyce chorób serca ważna jest ocena uniesienia, obniżenie i nachylenia odcinka ST. Na rysunku 1 przedstawiona typowy elektrokardiogram z wyszczególnionym odcinkiem ST.



Rys.1. Ukazano typowy elektrokardiogram z zaznaczonymi elementami diagnostycznymi. Kolorem zielonym zaznaczony jest odcinek ST. [2]

Odcinek ST związany jest z okresem depolaryzacji komór.

Prawidłowy odcinek ST trwa od 20 do 120 ms. Łączy on zespół QRS z załamkiem T. Zaczyna się w punkcie J, czyli na połączeniu zespołu QRS i odcinka ST a kończy się na początku załamka T. Jednakże zazwyczaj trudno jest określić dokładnie miejsce końca odcinka ST. Związek między odcinkiem RT i załamkiem T powinien być badany razem. Typowy odcinek ST trwa 80 ms i powinien leżeć na poziomie linii izoelektrycznej, która wyznaczana jest na odcinku PR i TP.[4]

Monitorowanie stanu odcinka ST jest istotne z punktu widzenia zapobiegania niedokrwieniu oraz wczesnego wykrywania sygnałów sugerujących wystąpienie patologicznych zmian w przyszłości. Jest to ważne szczególnie ze względu na to, że większość epizodów niedokrwienia przebiega bez bólowo i bezobjawowo, co czyni je trudnymi do wykrycia bez ciągłego monitorowania. Obniżenie bądź podniesienie poziomu załamka ST wynika ze zmian zachodzących w fazie repolaryzacji komórek mięśnia roboczego komór. Ich przyczyną jest brak równowagi jonowej, będący z kolei bezpośrednim następstwem niedotlenienia. Nieprawidłowe ze względu na odcinek ST wykresy sygnału EKG przedstawiono na rysunku 2.



Rys.2. Sygnał EKG. Po lewej - w pierwszej dobie po zawale, po prawej – w drugiej dobie .[3]

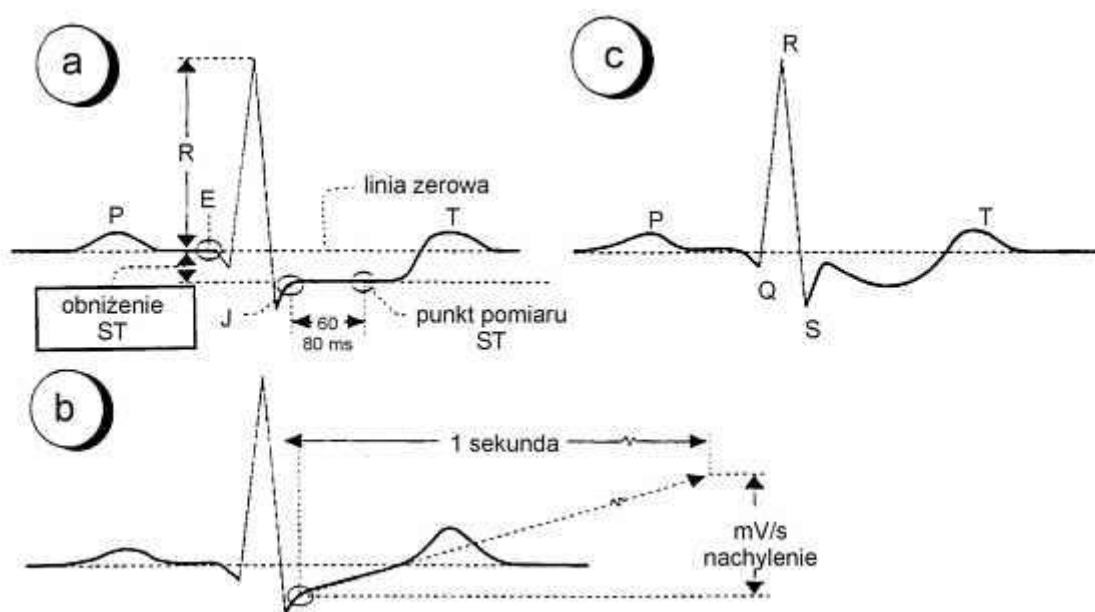
Podwyższenie odcinka ST ma diagnostyczne znaczenie jeśli pionowa odległość pomiędzy zapisem EKG i linią izoelektryczną w punkcie 40ms po punkcie J wynosi co najmniej 0.1 mV (1mm) w odprowadzeniu kończynowym lub 0.2 mV (2mm) w odprowadzeniu przedsercowym. Ten pomiar jest fałszywie pozytywny w 15-20% przypadków i fałszywie negatywny w 20-30% przypadków [5]. Na uniesienie mogą mieć wpływ różne czynniki, takie jak: ostre pełnościenne niedokrwienie (świeży zawał serca), dławica Prinzmetala, kardiomiopatia tako-tsubo, tętniak pozawałowy lewej komory, tętniak rozwarstwiający aorty i wiele innych. Jednakże najważniejszą przyczyną uniesienia odcinka ST jest zawał mięśnia sercowego[1].

Obniżenie odcinka ST jest znaczące jeśli wynosi 1 mm w V5-V6 lub 1.5 mm w AVF lub III odprowadzeniu. W badaniu stresu farmakologicznego obniżenie odcinka ST o co najmniej 1mm po podaniu adenozyne wskazuje na odwracalne niedokrwienie, a podczas próby wysiłkowej wymagane jest 2mm obniżenie aby sklasyfikować odwracalne niedokrwienie. Obniżenie odcinka

ST często wskazuje na niedokrwienie mięśnia sercowego, którego główną przyczyną jest zwężenie światła tętnicy wieńcowej. Innymi przykładami chorób, które mogą zostać zdiagnozowane na podstawie obniżenia odcinka ST są np. podwładziowe niedotlenienie lub nawet zawał[6].

Nachylenie odcinka ST jednoznacznie wskazuje na chorobę wieńcową. Przebieg skośny w dół wskazuje na niedokrwienie, natomiast skośny w górę jest uznawany za zmianę fizjologiczną podczas wysiłku. [7]

Na rysunku 3 przedstawiono punkty pomiarowe i sposób pomiaru odcinka ST niezbędne podczas diagnozy.



Rys.3. Punkty pomiarowe EKG związane z odcinkiem ST. [7]

Rysunek 3 a i c przedstawia zmiany w elektrokardiogramie wskazujące na zaburzenia w pracy serca podczas próby wysiłkowej. Rysunek 3 b przedstawia sposób obliczania nachylenia odcinka ST. [7]

Automatyczna analiza sygnału ma na celu umożliwić jego interpretację bez udziału człowieka. Aby to osiągnąć tworzy się oprogramowanie rejestratorów EKG, które implementuje specyficzne algorytmy odpowiedzialne za ekstrakcję wymaganych cech sygnału oraz porównanie ich z prawidłowymi wartościami mieszczącymi się w normach.

Sygnal przygotowany do analizy odcinków ST powinien być pozbawiony wszystkich pobudzeń pochodzenia komorowego. Tylko pozostałe pobudzenia są analizowane pod względem zmian zachodzących w obrębie odcinka ST. Eliminacja tych pobudzeń odbywa się we wcześniejszej fazie projektu, podczas klasyfikacji zespołów QRS. Proces ten jest konieczny ze względu na to, że repolaryzacja w przypadku pobudzeń pochodzenia innego niż zatokowe przebiega odmiennie.

Etapy analizy:

- a) wydzielenie obszaru załamka ST, jego początku i końca
- b) pomiar nachylenia odcinka ST w odniesieniu do poziomu oraz jego odchylenia względem linii izoelektrycznej.

Program powinien posiadać algorytm uśredniający wartości odcinka ST na podstawie wielu przebiegów, co umożliwi odrzucanie błędnych pomiarów a tym samym zmniejszenie ilości fałszywych alarmów. Błędy pomiaru mogą powstawać w wyniku zmian amplitudy spowodowanej jakością kontaktu elektrod ze skórą oraz grubością tkanki tłuszczowej w miejscu przyłożenia elektrody.

Określanie potencjalnych miejsc wystąpienia niedokrwienia może być ułatwione poprzez próbę analizy wszystkich 12 odprowadzeń. Szczegóły istotne z punktu widzenia diagnostycznego mogą być zauważalne tylko na niektórych z nich, a całkiem niewidoczne na innych.

Początek odcinka ST przyjmuje się w punkcie J, czyli po zakończeniu zespołu QRS. Ponieważ zarówno punkt J oraz punkt końca odcinka ST (przypadający na początek załamka T) jest trudny do zlokalizowania, przyjmuje się (o ile interwał RJ jest stały), że punkt J przypada na 45ms po szczycie załamka R a koniec odcinka ST na 105ms po szczycie załamka R. [8] Rozległość obszar załamka ST jest zależna od szybkości rytmu serca, toteż innym bardziej zaawansowanym rozwiązaniem jest obliczenie początku i końca odcinka ST (odstępy czasowe od szczytu załamka R) korzystając ze wzoru uwzględniającego wartość rytmu serca [8]. W niniejszym projekcie zdecydowano się przyjąć właśnie to rozwiązanie. Zostanie ono szerzej opisane w dalszej części.

### 3. Koncepcja proponowanego rozwiązania

Wszystkie procedury, mające na celu analizę odcinka ST w celu otrzymania parametrów diagnostycznych są przeprowadzane tylko dla pobudzeń serca, mających swoje źródło w węzle zatokowo-przedsionkowym. Wszystkie okresy zawierające zespoły QRS innych klas morfologicznych są pomijane. Algorytm, przed przystąpieniem do właściwych obliczeń sprawdza wartość zmiennej MorFT zapisanej w wejściowej strukturze QRS\_ATTR. Jeżeli jej wartość jest inna niż 0 (źródło SV), następuje przejście do kolejnego okresu sygnału.

### 3.1 Algorytm znajdujący linię izoelektryczną

Prawidłowe wyznaczenie parametrów odcinka ST tj. jego nachylenia oraz odchylenia wymaga znajomości wartości poziomu punktu BL, znajdującego się na linii izoelektrycznej tuż przed wystąpieniem zespołu QRS. Plik wejściowy, będący bazą danych niezbędnych do wykonania obliczeń związanych z odcinkiem ST dostarcza informacji o numerze próbki punktu detekcji QRS. Numer tej próbki jest wykorzystywany do poszukiwania wstecz ostatniego punktu należącego do linii izoelektrycznej (punkt BL). Algorytm sprawdza wartości pochodnych w kolejnych fragmentach przebiegu począwszy od punktu detekcji do punktu oddalonego wstecz o ok. 100 ms (maksymalny czas trwania odcinka PQ). W każdej iteracji sprawdzane są kolejne trzy punkty analizowanego fragmentu sygnału. Wykrycie różnic między tymi trzema punktami, na poziomie nie większym niż wartość tolerancji jest równoznaczne z detekcją linii izoelektrycznej oraz punktu BL. W przypadku, gdy kryteria założonej tolerancji nie są spełnione, algorytm zaczyna poszukiwania od początku po uprzedniej inkrementacji tej wartości. Początkowa wartość tolerancji wynosi  $2\mu\text{V}$ . Ilość punktów sprawdzanych w poszukiwaniu linii przyjęto jako trzy po to, aby spełnić kryterium minimalnej długości dla fragmentu sygnału mającego pełnić rolę linii izoelektrycznej. Trzy kolejne próbki, przy próbkowaniu 128 Hz opisują ok. 30ms odcinek czasowy elektrokardiogramu.

### 3.2 Detekcja punktów J i ST

Odnalezienie numerów próbek odpowiadających punktom J oraz ST wymaga wykorzystania wzorów obliczających ich czasowe odległości względem maksimum zespołu QRS. W algorytmie wykorzystano wzory wyznaczające:

- a) Czas pomiędzy maksimum R oraz punktem J

$$t_{R-J} = 40 + 0,948\sqrt{t_{R-R}} \text{ [ms]}$$

- b) Czas pomiędzy maksimum R oraz punktem ST

$$t_{R-ST} = 40 + 2,21\sqrt{t_{R-R}} \text{ [ms]}$$

Wartość interwału między uderzeniowego z danego okresu sygnału  $t_{R-R}$ , pobierana jest ze struktury QRS\_QTTR ze zmiennej localRR. Wyznaczone wartości czasów stanowią podstawę do obliczenia numerów próbek punktów ST i J. Ponadto wymagane jest również odczytanie wartości poprawki punktu detekcji względem maksimum R (maksimum R i punkt detekcji to dwa różne punkty). Korekta zapisana jest w strukturze pod nazwą Amax. Bezwzględne numery próbek punktów J i ST wyznaczone są na podstawie znajomości czasów  $t_{R-J}$ ,  $t_{R-ST}$ , oraz częstotliwości próbkowania 128 Hz:

$$nrProbkiST = nrProbkiR + (t_{R-ST}[\text{s}] + Amax) * 128\text{Hz}$$

$$nrProbkiJ = nrProbkiR + (t_{R-J}[s] + Amax) * 128Hz$$

Znając numery próbek tych punktów możliwe jest odczytywanie wartości amplitudy sygnału w tych miejscach oraz obliczanie parametrów odcinka ST.

### 3.3 Obliczanie parametrów diagnostycznych odcinka ST

Podstawowymi parametrami diagnostycznymi obliczanymi przez algorytm jest maksymalne obniżenie lub podniesienie odcinka ST oraz jego nachylenie. W celu wyznaczenia maksymalnego odchylenia wyznacza się wartości amplitud punktów ST i J względem linii izoelektrycznej. Następnie porównując wartości bezwzględne tych liczb otrzymuje się największe odchylenie. Nachylenie odcinka ST jest aproksymowane prostą przechodzącą przez analizowane punkty. Opisana jest ona wzorem  $y=ax$ , gdzie współczynnik kierunkowy  $a$  stanowi poszukiwane nachylenie. W celu wyznaczenia współczynnika kierunkowego w pierwszym etapie na podstawie znajomości numerów próbek na osi poziomej obliczana jest długość odcinka ST [ms]. Iloraz różnic amplitud oraz długości odcinka stanowi współczynnik kierunkowy prostej, a tym samym jej nachylenie:

$$wsp.kierunkowy = \frac{s(ST) - s(J)}{długość\ ST}$$

$s(ST)$  – amplituda sygnału w punkcie ST

$s(J)$  – amplituda sygnału w punkcie J

$długość\ ST$  - wyznaczana jako różnica  $t_{R-ST}$  i  $t_{R-J}$

Algorytm uwzględnia wszystkie przypadki położenia punktów ST i J względem linii izoelektrycznej, zwracając w wyniku zarówno dodatnie i ujemne wartości odchylenia i nachyleń. Obliczone parametry umieszczane są w tablicy char str[2\*3], w taki sposób, że na pierwszych dwóch pozycjach znajdują się dane dla kanału pierwszego, na dwóch kolejnych dla drugiego, ostatnie dwie pozycje przypisane są kanałowi trzeciemu.

### 3.4 Detekcja epizodów

Epizod ST jest zgłaszany przez program w przypadku wystąpienia obniżenia bądź podniesienia odcinka o 1mm tj, 100 $\mu$ V względem linii izoelektrycznej. Detekcja epizodu jest możliwa więc, już na etapie wyznaczania podstawowych parametrów diagnostycznych. Rozpoznanie nieprawidłowości rozpoczyna proces obliczania i zbierania informacji, dających większe rozeznanie schorzenia z nim związanego. Następuje odczytanie czasu poprzedniego epizodu, który jest wraz z każdą nową detekcją nadpisywany do przygotowanej zmiennej. Pozwala to na określanie odstępów



czasowych od poprzednich epizodów. Czas od ostatniego epizodu określany jest jako różnica czasowa punktu J obecnego odcinka ST (w którym epizod nastąpił) oraz punktu ST poprzedniego odcinka z epizodem. Ponadto sprawdzany jest numer zespołu QRS, po którym nastąpiły nieprawidłowości oraz współczynniki HR w okresie obecnym, następnym oraz poprzedzającym epizod. Wszystkie te wartości są odczytywane ze struktury QRS\_ATTR. Algorytm uwzględnia zmiany we wszystkich trzech kanałach, sprawdzając kolejno wartości w tablicy char str[2\*3]. Sprawia to, że epizody niezauważalne w części kanałów nie są pomijane, lecz właściwie wykrywane na podstawie danych z innego kanału.

### **3.5 Zapisywanie wyników**

Do pliku wyjściowego zapisywane są obliczone parametry diagnostyczne oraz pomocnicze wyznaczone we wcześniejszym etapie algorytmu. Wypisywane są tylko informacje dla okresów zawierających epizody, pozostałe traktowane jako prawidłowe są pomijane w pliku wyjściowym. Dane ułożone są w kolumnach oddzielonych znakiem spacji, każda z nich jest przypisana do jednego rodzaju danych.

Plik zawiera następujące informacje (wypisane wg numerów kolumn):

- 1) Numer kanału, w którym wystąpił epizod
- 2) Numer pierwszego zespołu QRS, w którym epizod wystąpił
- 3) Czas trwania epizodu [100ms]
- 4) Odstęp od poprzedniego epizodu
- 5) Maksymalna wartość obniżenia, podniesienia
- 6) Wartość nachylenia ST ( $100\mu\text{V/ms}$ )
- 7) Numer epizodu
- 8) Wartość HR przed rozpoczęciem epizodu
- 9) Wartość HR w trakcie epizodu
- 10) Wartość HR po zakończeniu epizodu

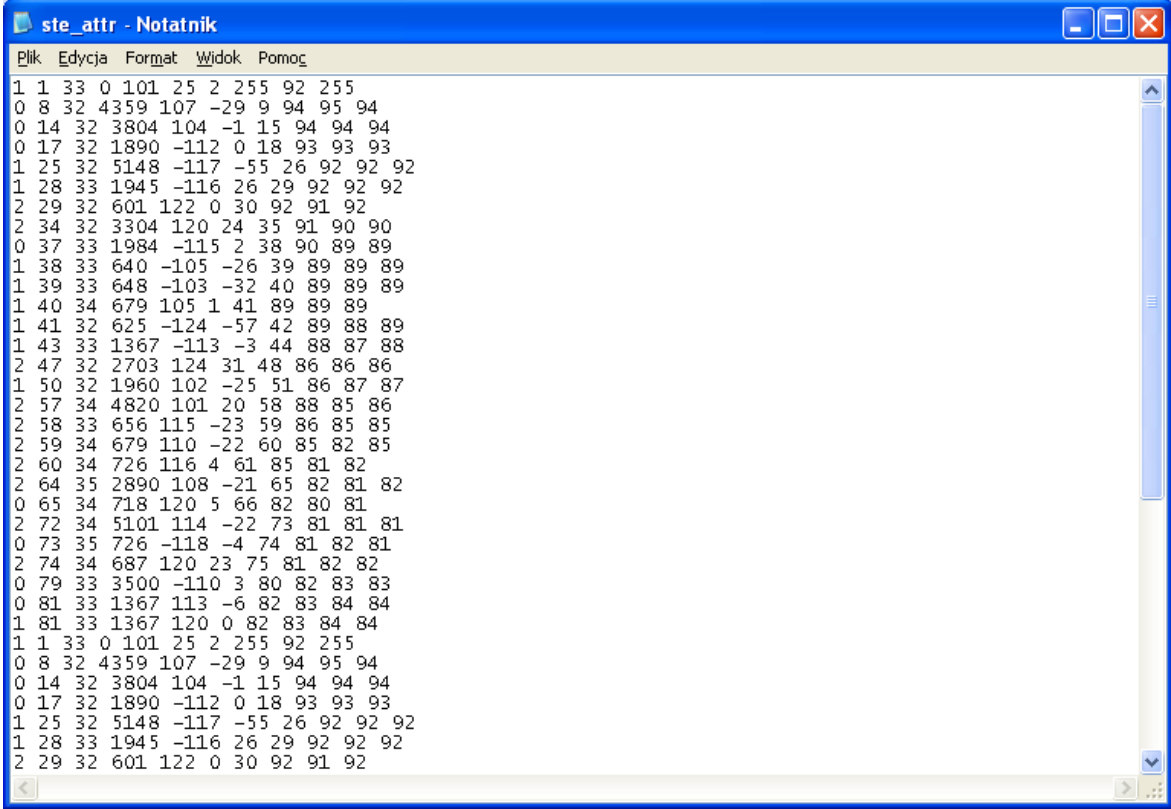
## **4. Rezultaty i wnioski**

Aplikacja prawidłowo analizuje odcinek ST oraz wykrywa epizody zgodnie z postawionymi założeniami.

Kluczowym elementem algorytmu jest prawidłowe rozpoznanie linii izoelektrycznej. Ewentualne błędy, niedokładności rzutują w konsekwencji na dokładność i wiarygodność obliczanych w kolejnych etapach parametrów diagnostycznych. Wynika to z tego, że ich wartości są obliczane w

odniesieniu do wartości amplitudy linii izoelektrycznej. 95% wykrytych w sygnale poziomów linii ma amplitudę bezwzględną równą w przybliżeniu  $120\mu\text{V}$ . W pozostałych przypadkach zaobserwowano znaczne odchylenia, przekraczające nawet wartości  $200\mu\text{V}$  lub będące bliskie  $0\mu\text{V}$ . Przyczyny takiej sytuacji mogą mieć różnorodne źródła, związane nie tylko z algorytmem, ale również z charakterem sygnału wybranego do analizy. Detekcja trzech sąsiednich punktów, posiadających zbliżone amplitudy, należących jednak do zespołu QRS lub załamka P może w efekcie dać fałszywe rozpoznanie linii izoelektrycznej. Zmienione amplitudy linii mogą świadczyć również o jakości kontaktu elektrod ze skórą. Taką przyczynę nieprawidłowości należy jednak odrzucić ze względu na to, że poziom linii zmienia się skokowo. Inną przyczyną może być charakter sygnału, wykazujący cechy migotania komór, co obawia się falowaniem całego przebiegu. Badanie pochodnej linii izoelektrycznej wykazuje jednak małe prawdopodobieństwo takiego przypadku.

Przykładowy plik wynikowy przedstawiony został na rysunku 4:



```
ste_attr - Notatnik
Plik  Edycja  Format  Widok  Pomoc
1 1 33 0 101 25 2 255 92 255
0 8 32 4359 107 -29 9 94 95 94
0 14 32 3804 104 -1 15 94 94 94
0 17 32 1890 -112 0 18 93 93 93
1 25 32 5148 -117 -55 26 92 92 92
1 28 33 1945 -116 26 29 92 92 92
2 29 32 601 122 0 30 92 91 92
2 34 32 3304 120 24 35 91 90 90
0 37 33 1984 -115 2 38 90 89 89
1 38 33 640 -105 -26 39 89 89 89
1 39 33 648 -103 -32 40 89 89 89
1 40 34 679 105 1 41 89 89 89
1 41 32 625 -124 -57 42 89 88 89
1 43 33 1367 -113 -3 44 88 87 88
2 47 32 2703 124 31 48 86 86 86
1 50 32 1960 102 -25 51 86 87 87
2 57 34 4820 101 20 58 88 85 86
2 58 33 656 115 -23 59 86 85 85
2 59 34 679 110 -22 60 85 82 85
2 60 34 726 116 4 61 85 81 82
2 64 35 2890 108 -21 65 82 81 82
0 65 34 718 120 5 66 82 80 81
2 72 34 5101 114 -22 73 81 81 81
0 73 35 726 -118 -4 74 81 82 81
2 74 34 687 120 23 75 81 82 82
0 79 33 3500 -110 3 80 82 83 83
0 81 33 1367 113 -6 82 83 84 84
1 81 33 1367 120 0 82 83 84 84
1 1 33 0 101 25 2 255 92 255
0 8 32 4359 107 -29 9 94 95 94
0 14 32 3804 104 -1 15 94 94 94
0 17 32 1890 -112 0 18 93 93 93
1 25 32 5148 -117 -55 26 92 92 92
1 28 33 1945 -116 26 29 92 92 92
2 29 32 601 122 0 30 92 91 92
```

Rys. 4 Przykładowy plik wynikowy

Czasy odcinków ST wyznaczone w aplikacji przyjmują wartości ok. 32ms. Zawiera się to w przedziale długości tych odcinków obserwowanych w praktyce lekarskiej (20-120ms).

Można zaobserwować, że możliwość detekcji epizodów we wszystkich trzech kanałach jest bardzo użyteczną opcją, która ma duże znaczenie w diagnozowaniu chorób. W pliku wynikowym zaledwie 1 % przypadków widocznych jest we wszystkich 3 kanałach. Zdecydowana większość (ponad 90% epizodów) jest dostrzegalna tylko w jednym z kanałów. Inaczej przedstawia się rozkład ilości

epizodów wykrytych w kanale o konkretnym numerze. Na podstawie danych wynikowych oszacowano procentowy rozkład, który przedstawiono w Tabeli x.

Tabela 1

Numer kanału	Ilość wykrytych epizodów [%]
1	27
2	38
3	35

Tabela 1 pokazuje, że rozkład ilości epizodów w różnych kanałach jest zbliżony, porównywalny. Wybierając do analizy tylko jeden kanał można odnaleźć zaledwie ok. 1/3 wszystkich nieprawidłowości, co stanowiłoby bardzo duże obniżenie jakości analizy i diagnozy.

W wynikach zauważono błędy w wartościach HR, które są podawane w pliku wyjściowym jako informacje dodatkowe dotyczące epizodu. Można zaobserwować je na rysunku 4 przedstawiającym przykładowy plik wynikowy (pierwszy wiersz danych). Błędy te nie powstają w wyniku działania algorytmu, są przepisywane ze struktury wejściowej do tego projektu. Dokładność obliczanych w tym projekcie położeń punktów J oraz ST również zależy od poprawności punktów detekcji i wartości korekt (również odczytywanych ze struktury wejściowej).

Dodatkowym testem, potwierdzającym wszystkie błędy i odchyłki może być wyświetlenie fragmentów sygnału wraz z zaznaczonymi punktami charakterystycznymi oraz wizualne określenie ich poprawności. Innym rozwiązaniem może być testowanie sygnału z bazy CSE w następujących etapach:

- przyjęcie wyników wejściowych obliczonych przez najlepsze algorytmy
- wykonanie wszystkich operacji analizy i detekcji przy użyciu algorytmu opisanego w tym projekcie
- przeprowadzenie analizy i detekcji z wykorzystaniem algorytmów najlepszych
- porównanie wyników końcowych obu algorytmów

## 5. Podsumowanie

Aplikacja spełnia wszystkie wymagania postawione przed przystąpieniem do prac związanych z implementacją algorytmów. Osiągnięciem jest to, że w sposób logiczny i spójny analizuje odcinek ST zarejestrowanego elektrokardiogramu, dając konstruktywne wyniki. Na ich podstawie możliwe jest wyciągnięcie szeregu informacji o stanie zdrowia pacjenta oraz danych statystycznych wspomagających i uogólniających problem detekcji patologicznych zmian. Przedstawiony algorytm nie został przetestowany pod względem spełniania kryteriów normy IEC 60601-2-51.

Jego poprawność jest zatem tylko subiektywną oceną autorów projektu. Sprawia to, że nie może być stosowany do celów diagnostycznych w szpitalach i innych instytucjach medycznych. Pomimo tego stanowi doskonałą próbę i sposobność do zdobycia doświadczenia w zakresie automatycznej interpretacji EKG. Wszystkie informacje zdobyte w trakcie rozwiązywania problemów projektowych oraz implementacji algorytmów poszerzają świadomość trudności i wyzwań stojących przed osobą realizującą oprogramowanie z tej dziedziny. Mogą stanowić dobry wstęp do rozwijania i implementowania podobnych projektów w przyszłości.

## 6. Literatura

- [1] Lek. Bartłomiej Kiszka, *Przyczyny uniesienia odcinka ST w elektrokardiogramie*, 2008
- [2] <http://pl.wikipedia.org/wiki/Elektrokardiografia>
- [3] Joanna Grabska-Chrzastowska, *EKG, układ krążenia*
- [4] [http://en.wikipedia.org/wiki/ST\\_interval](http://en.wikipedia.org/wiki/ST_interval)
- [5] [http://en.wikipedia.org/wiki/ST\\_elevation](http://en.wikipedia.org/wiki/ST_elevation)
- [6] [http://en.wikipedia.org/wiki/ST\\_depression](http://en.wikipedia.org/wiki/ST_depression)
- [7] <http://www.czytelniamedyczna.pl/2892,elektrokardiograficzne-proby-wysilkowe-u-pacjentow-z-choroba-niedokrwienna-serca.html>
- [8] P. Augustyniak, *Przetwarzanie sygnałów elektrodiagnostycznych*

## 7. DODATEK A. Opis informatyczny procedur

```
/*
 * przechowuje dane wejsciowe takie jak
 * sygnał i atrybuty dotyczące QRS, uzyskane
 * we wcześniejszym etapie analizy sygnału
 * a wykorzystywane obecnie. Przechowuje długość
 * tablic wczytanych danych oraz sygnału a także
 * ilość wykrytych epizodów.
 *
 * Tworzy globalną zmienną rejestr, która jest uzupełniana
 * w trakcie prowadzonej w czasie działania programu analizy
 * sygnału.
 */
struct STRegister{
    unsigned int dataLength; /* dlugosc tablicy z atrybutami QRS */
    unsigned int signalLength; /* dlugosc tablicy z sygnałem */
    QRS_ATTR *data; /*tablica z atrybutami QRS */
    unsigned char *signal; /* tablica z danymi sygnału */
    int epizodsAmount; /*ilosc epizodow */
} rejestr;

/*****
 *
 * void wczytajAtrybutyQRS
 *
 * Przeznaczenie:
 * Wczytuje atrybuty QRS, które zawierają wynik
 * analizy sygnału z wcześniejszej fazy projektu
 *
 * Argumenty:
 * I char * filename - nazwa pliku do wczytania
 * I/O STRegister rej - struktura rejestru
 *
 * Zwraca: void
 *
 * Używane funkcje: fopen, printf, rewind, fgets, fgetc, fclose
 *
 * Używane zmienne: FILE* file, int znak, long iloscDanych, int EOF,
 * QRS_ATTR data[], int iloscSpacji, char lineBuffer[80],
 * char pojedynczaDana[10], int iteratorPojDanej, char c,
 *
 *
 * Uwagi: brak
 *
 * Autor: Jarosław Kardynia i Michał Piela
 *
 * Ostatnia modyfikacja: 19 stycznia 2011r.
 */
void wczytajAtrybutyQRS(char * filename, STRegister rej){
    FILE* file;

    if ((file=fopen(filename, "r"))==NULL) {
        printf ("Nie mogę otworzyć pliku sygnału.\n");
        exit(1);
    }

    int znak; // przechowuje pojedynczy znak wczytane z pliku
    long iloscDanych=0; // ilosc znalezionych linii w pliku

    while((znak = fgetc( file ) ) != EOF) { //sprawdza ilosc linii (moze zostac
        //rozpoznane o 1 linie za malo
        //gdy plik nie konczy sie nowa pusta
        //linią)
        if((char)znak == '\n'){
            iloscDanych = iloscDanych +1;
        }
    }
}
```

```

rej.dataLength = iloscDanych; // zapisuje wynik w globalnym rejestrze

rewind(file);

QRS_ATTR data[iloscDanych]; // alokuje pamiec dla odpowiedniej ilości danych
int iloscSpacji; // okresla numer slowa w zdaniu w danej linii (np. 0
                // oznacza ze analizowane jest pierwsze slowo)
char lineBuffer[80]; // bufor na pojedyncza linie
char pojedynczaDana[10]; // bufor na pojedyncza liczbe
int iteratorPojDanej;

char c; // przechowuje pojedynczy znak

for(int iterator=0; iterator<10; iterator=iterator+1){//przygotowanie bufora
                                                    // pojedynczej danej
    pojedynczaDana[iterator] = ' ';
    if(iterator == 9){
        pojedynczaDana[iterator] = '\\0';
    }
}

for(int i=0; i < rej.dataLength; i = i+1){ // pobiera kolejno nową linie
    fgets(lineBuffer,80,file);

    //analizuje poj. linie
    for(int j=0, iloscSpacji=0, iteratorPojDanej=0; ;j = j+1){
        c = pojedynczaDana[iteratorPojDanej] = lineBuffer[j];

        if(c == '\\0'){ // koniec linii
            break;
        }
        if(c == ' '){ // koniec wyrazu
            if(iloscSpacji==0){ //DetPt

                data[i].DetPt = atoi(pojedynczaDana);
            }else if(iloscSpacji ==1){ //AMax
                data[i].AMax = (char)atoi(pojedynczaDana);
            }else if(iloscSpacji ==2){ //localRR
                data[i].localRR = (short) atoi(pojedynczaDana);
            }else if(iloscSpacji ==3){ //HR
                data[i].HR = (unsigned char)atoi(pojedynczaDana);
            }else if(iloscSpacji ==4){ //MorfT
                data[i].MorfT = (char)atoi(pojedynczaDana);
            }else if(iloscSpacji ==5){ //cl
                data[i].cl = (short)atoi(pojedynczaDana);
            }else if(iloscSpacji ==6){ //str[0]
                data[i].str[0] = (char)atoi(pojedynczaDana);
            }else if(iloscSpacji ==7){ //str[1]
                data[i].str[1] = (char)atoi(pojedynczaDana);
            }else if(iloscSpacji ==8){ //str[2]
                data[i].str[2] = (char)atoi(pojedynczaDana);
            }else if(iloscSpacji ==9){ //str[3]
                data[i].str[3] = (char)atoi(pojedynczaDana);
            }else if(iloscSpacji ==10){ //str[4]
                data[i].str[4] = (char)atoi(pojedynczaDana);
            }else if(iloscSpacji ==11){ //str[5]
                data[i].str[5] = (char)atoi(pojedynczaDana);
            }
        }

        iloscSpacji = iloscSpacji+1;
        iteratorPojDanej= 0;

        //czyszczenie bufora pojedynczej danej
        for(int iterator=0; iterator<10; iterator=iterator+1){
            pojedynczaDana[iterator] = ' ';
            if(iterator == 9){
                pojedynczaDana[iterator] = '\\0';
            }
        }

        continue;
    }
}

```

```

        }
        iteratorPojDanej = iteratorPojDanej+1;

    }
    rej.data = data; // przypisanie wczytanych danych do globalnego rejestru
    fclose (file);
}

/*****
*
*         void wczytajSygnal
*
* Przeznaczenie:
*         Wczytuje sygnał
*
* Argumenty:
*         I char * filename - nazwa pliku do wczytania
*         I/O STRegister rej - struktura rejestru
*
* Zwraca: void
*
* Używane funkcje: fopen, printf, fgetc, rewind, fgets, fclose
*
* Używane zmienne: FILE* file, long iloscDanych, int EOF, unsigned char sig[],
*
* Uwagi: brak
*
* Autor: Jarosław Kardynia i Michał Piela
*
* Ostatnia modyfikacja: 19 stycznia 2011r.
*/
void wczytajSygnal(char * filename, STRegister rej){
    FILE* file;

    if ((file=fopen(filename, "r"))==NULL) {
        printf ("Nie mogę otworzyć pliku sygnału.\n");
        exit(1);
    }
    long iloscDanych=0;
    while(fgetc( file ) != EOF ){ //sprawdza ilosc danych
        iloscDanych = iloscDanych +1;
    }
    rewind(file);
    unsigned char sig[iloscDanych];

    for(int i=0; i < iloscDanych; i = i+1){ // wczytuje dane
        sig[i] = (unsigned char)fgetc(file);
    }
    fclose (file);

    //zapis do struktury
    rej.signal = sig;
    rej.signalLength = iloscDanych;
    printf("%i, ilosc", iloscDanych);
}

/*****
* int[] znajdzIzolynie
*
* szukanie lini izoelektrycznej
* poszukiwania zaczynają się od punktu detekcji QRS wstecz w okresie czasu
* przypadającego na maksymalną wartosc odcinka PQ = 100ms (PQ=40-100ms)
*/
int* znajdzIzolynie(STRegister rej,int i){
    int nrOstatniejProbkiPQ;
    int liniaIzo[3];
    int znaleziono=0;
    int punktDetekcjiQRS= rej.data[i].DetPt;
    int tolerancjaPredkosciLiniIz=2;
    int nrProbkiKoncowejPoszukiwania=punktDetekcjiQRS-0.1*120;

```



```

while (1){
    for(int m= punktDetekcjiQRS;m>nrProbkiKoncowejPoszukiwania+1;m=m-1){
        if ((rej.signal[m*3]-rej.signal[(m-1)*3])<tolerancjaPredkosciLiniIz
&&(rej.signal[(m-1)*3]-rej.signal[(m-2)*3])<tolerancjaPredkosciLiniIz){
            nrOstatniejProbkiPQ=m;
            for(int chan=0;chan<3;chan=chan+1){
                liniaIzo[chan]=rej.signal[m*(chan+1)];
            }
            znaleziono=1;
            break;
        }
    }
    if (znaleziono==1) {break;}
    else {tolerancjaPredkosciLiniIz=tolerancjaPredkosciLiniIz+1;}
}
return liniaIzo;
}

```

```

/*****

```

```

* void analizaST

```

```

*

```

```

* Przeznaczenie:

```

```

* Dokonuje analizy sygnału EKG pod kątem znalezienia
* nieprawidłowości w odcinku ST. Znajduje miejsca
* obniżenia, podwyższenia i nachylenia odcinka ST
* korzystając z danych uzyskanych z analizy sygnału
* z wcześniejszej fazy projektu.
*

```

```

* Argumenty:

```

```

* I stAttr tab[] - przechowuje wynik analizy
* I/O STRegister rej - struktura rejestru

```

```

*

```

```

* Zwraca: void

```

```

*

```

```

* Używane funkcje: znajdzIzolinie,

```

```

*

```

```

* Używane zmienne: int i,j,k; int RJtime;int RSTtime;int punktJ;int punktST;
* unsigned int DJtime;unsigned int DSTtime;int nrProbkaST;
* int nrProbkaJ;int STlength;int nrProbkiST_epizodu;
* unsigned int tolerancja;int epizodsAmount;stAttr s,
* int liniaIzo[], int amplitudaST; int amplitudaJ;
*

```

```

* Uwagi: brak

```

```

*

```

```

* Autor: Jarosław Kardynia i Michał Piela

```

```

*

```

```

* Ostatnia modyfikacja: 19 stycznia 2011r.

```

```

*/

```

```

void analizaST(stAttr tab[], STRegister rej){

```

```

    int i,j,k;
    int RJtime;
    int RSTtime;
    int punktJ;
    int punktST;
    unsigned int DJtime;
    unsigned int DSTtime;
    int nrProbkaST;
    int nrProbkaJ;
    int STlength;
    int nrProbkiST_epizodu=0;
    unsigned int tolerancja=100;
    int epizodsAmount=0;

```

```

    stAttr s;
    //stAttr tabStr[rejestr.dataLength];
    for(i =0, k=0; i < 90; i = i+1) //rejestr.dataLength

```

```

{
    int *liniaIzo = znajdzIzolynie(rejestr,i);
    int punktDetekcjiQRS= rej.data[i].DetPt;
    RJtime=40+0.948*sqrt(rej.data[i].localRR);
    RSTtime=40+2.21*sqrt(rej.data[i].localRR);
    DJtime=RJtime+rej.data[i].AMax; //w ms
    DSTtime=RSTtime+rej.data[i].AMax; // w ms
    nrProbkaST=punktDetekcjiQRS+DSTtime*128/1000;

    nrProbkaJ=punktDetekcjiQRS+DJtime*128/1000;
    STlength=RSTtime-RJtime;

    //TYLKO pochodzenia nadkomorowego
    if(rej.data[i].Morft==0)
    {
        //amplitud punktów ST i J względem lini izoel.
        int amplitudaST;
        int amplitudaJ;
        for (int kanal=0;kanal<3;kanal=kanal+1){

            amplitudaST=rej.signal[nrProbkaST*(kanal+1)]-liniaIzo[kanal];
            amplitudaJ=rej.signal[nrProbkaJ*(kanal+1)]-liniaIzo[kanal];

            //maksymalne obniżenie, podniesienie

            if(abs(amplitudaST)>abs(amplitudaJ)) {s.str[kanal*2]=(char)amplitudaST;}
            else {s.str[kanal*2]=(char)amplitudaJ;}

            //nachylenie
            s.str[kanal*2+1]=(char)((amplitudaST-amplitudaJ)/(nrProbkaST- nrProbkaJ));
            //numer kanału
            s.chan[kanal]=kanal;

            //WARTOSCI WSPOLNE DLA WSZYSTKICH KANALOW (OBLICZANE TYLKO RAZ)
            if (kanal==2){
                //wartosci HR

                if (i==0) s.HRBefore=0;
                else s.HRBefore=rej.data[i-1].HR;
                s.HRAfter=rej.data[i+1].HR;
                s.HRatMxD=rej.data[i].HR;

                //czas trwania epizodu
                s.Dur=STlength;

                //odstep od epizodu poprzedniego
                If (abs((int)s.str[0])>100 || abs((int)s.str[2])>100 || abs((int)s.str[4])>100)
                {
                    if(nrProbkiST_epizodu==0) {
                        nrProbkiST_epizodu=nrProbkaST;
                        s.Sep=0;}
                    else{
                        s.Sep=(nrProbkaJ-nrProbkiST_epizodu)*1000/128;
                        nrProbkiST_epizodu=nrProbkaST;}
                }
                else{s.Sep=0;}

                //Numer zespolu QRS przed epizodem
                s.StartQrsNr=i;
            }
        }
    }
}
//ednif
tab[k]=s;
k=k+1;
}
}

```