

**AKADEMIA GÓRNICZO – HUTNICZA
im. Stanisława Staszica w Krakowie**

**Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki
Katedra Automatyki**



PRACA DYPLOMOWA

Kierunek studiów :

Systemy Automatyki i Aparatury Medycznej

Temat pracy dyplomowej:

*Metody kompresji EKG z wykorzystywaniem
dekompozycji czasowo - częstotliwościowej*

Wykonawca:

Konrad Heksel

Promotor:

Dr inż. Piotr Augustyniak

Ocena:

KRAKÓW 2001

Spis treści

1. WPROWADZENIE	2
2. TRANSFORMACJA FALKOWA	5
2.1. INFORMACJE OGÓLNE	5
2.2. ANALIZA WIELOROZDZIELCZA	7
2.3. FILTRACJA PASMOWA SYGNAŁU	11
2.4. REKONSTRUKCJA.....	16
2.5. ZASADA ZACHOWANIA ENERGII.....	18
3. FALKI	19
3.1. FUNKCJA HAAR'A.....	19
3.2. FALKA DAUBECHIES.....	19
3.3. FALKI COIFLETS	20
3.4. FALKI BIORTOGONALNE	20
3.5. PORÓWNANIE PARAMETRÓW FALEK	21
4. SYGNAŁ EKG	22
4.1. ZJAWISKA ELEKTRYCZNE W POJEDYNCZEJ KOMÓRCE MIĘŚNIA SERCOWEGO	22
4.2. AKTYWNOŚĆ ELEKTRYCZNA SERCA, A PRZEBIEG EKG	23
4.3. REJESTRACJA PRACY SERCA	26
4.4. REPREZENTACJA FALKOWA SYGNAŁU EKG	29
5. OPIS PROGRAMU	30
5.1. BUDOWA APLIKACJI.....	30
5.2. INSTRUKCJA UŻYTKOWNIKA	34
6. DOŚWIADCZENIA	38
6.1. DOŚWIADCZENIA WPROWADZAJĄCE, DOBÓR PROGU OBCINANIA.....	38
6.2. DOBÓR POZIOMU DEKOMPOZYCJI	46
6.3. DOBÓR FALKI.....	50
6.4. WPLYW DŁUGOŚCI FALKI NA JAKOŚĆ KOMPRESJI	62
6.5. ZNALEZIENIE OPTYMALNEJ FALKI.....	65
6.6. PROBLEM BRZEGOWY	68
6.7. ZASTOSOWANIE DRUGIEGO PROGU	80
7. PODSUMOWANIE	87
8. LITERATURA	88
9. ZAŁĄCZNIK	89
9.1. FUNKCJA „_CONVOLUTE”	89
9.2. FUNKCJA „_ADDEXTENSIONS”	89
9.3. FUNKCJA „_DWT”	91
9.4. FUNKCJA „_INVERTDWT”	93
9.5. FUNKCJA „_EXTRACTSIGNAL”	95
9.6. FUNKCJA „_UPCOEF”	96
9.7. FUNKCJA „_TRUNCATECOEFF”	97
9.8. FUNKCJA „_SETWAVELETFACE”	99

1. Wprowadzenie

Dokoła nas znajdują się sygnały, które niosą informację o otaczających nas zjawiskach. Źródłem emisji owych sygnałów są między innymi mowa ludzka, wibracje pracujących maszyn, drgania sejsmiczne, przebiegi medyczne EKG i EEG, sygnały urządzeń elektrycznych, układów elektronicznych oraz wiele innych mających olbrzymie znaczenie w życiu człowieka. Potrzeba odczytywania i interpretacji wszelkiego rodzaju zjawisk skłoniła do tworzenia coraz bardziej zaawansowanych metod ich analizy, opisu i szczegółowej obróbki (kodowania, kompresji, odszumiania, itp.).

W wielu wypadkach informacja niesiona przez obserwowane sygnały charakteryzuje się nadmiarowością. Wynika ona głównie z korelacji między sąsiednimi próbkami mierzonego sygnału. Właściwość ta pozwala z dużym prawdopodobieństwem przewidzieć wartość jednej próbki na podstawie wartości sąsiadujących z nią próbek. Problem efektywnego opisu sygnału wiąże się także z jego charakterem. Pod tym względem sygnały można podzielić na dwie zasadnicze grupy: stacjonarne (deterministyczne), czyli niezmiennie w czasie, ich wartość można wyznaczyć w dowolnej chwili czasu oraz sygnały niestacjonarne (stochastyczne) – zmieniające swój charakter z upływem czasu, pozwalające określić wartość sygnału jedynie z pewnym prawdopodobieństwem.

Spośród metod opisu sygnałów bardzo użytecznym narzędziem okazały się metody transformacyjne pozwalające przedstawić sygnał w dziedzinie częstotliwości i zbadać jego właściwości widmowe. Analizy częstotliwościowe polegają na przedstawieniu badanego sygnału za pomocą kombinacji liniowej pewnych bazowych sygnałów elementarnych, które zazwyczaj charakteryzują się ortogonalnością względem siebie.

$$x(t) = \sum_k a_k g_k(t)$$

gdzie:

$x(t)$ – analizowany sygnał

$g_k(t)$ – 'k'-ty sygnał bazowy

$a_k(t)$ – współczynnik określający intensywność występowania w badanym sygnale bazy $g_k(t)$, innymi słowy wyznacza stopień podobieństwa między tymi sygnałami

$$a_k = \int x(t) g_k^*(t) dt$$

$g_k^*(t)$ oznacza wartość sprzężoną do $g_k(t)$.

Ideą metod częstotliwościowych jest minimalizacja niezerowych współczynników a_k , czyli minimalizacja informacji potrzebnej do opisu sygnału. Zatem bardzo ważnym czynnikiem jest wybór odpowiedniego zestawu funkcji bazowych. Im kształt funkcji bazowych będzie bardziej dopasowany do analizowanego sygnału, tym mniejsza ilość tych funkcji wystarczy do jego aproksymacji. W efekcie pociąga to za sobą większość zerowych współczynników a_k oraz stosunkowo małą liczbę „istotnych” współczynników reprezentujących badany przebieg. Opierając się na powyższym sygnały stacjonarne powinny być analizowane funkcjami bazowymi o charakterze stacjonarnym. Takie są np. funkcje sinus i cosinus doskonale zlokalizowane w dziedzinie częstotliwości (nieskończony nośnik) – nadają się one do opisu

przebiegów okresowych. Natomiast dla sygnałów niestacjonarnych należy wybrać funkcje o dobrej lokalizacji przestrzennej (czasowej), czyli o skończonym (selektywnym) nośniku.

Sygnały EKG, których analizie poświęcona jest niniejsza praca zaliczają się do przebiegów niestacjonarnych. Do ich opisu zostanie wykorzystana transformacja falkowa. Reprezentacja falkowa sygnału jest funkcją dwu zmiennych, stanowiącą kombinację liniową skalowalnej (parametr m) i przesuwalnej (parametr n) funkcji zwanej falką – matką ψ (mother wavelet):

$$f(x) = \sum_m \sum_n d_{m,n} \psi(2^{-m}x - n)$$

Współczynniki $d_{m,n}$ oblicza się analogicznie jak w przypadku transformaty Fouriera:

$$d_{m,n} = 2^m \int f(x) \psi(2^{-m}x - n) dx$$

Wzór można interpretować również jako filtrację sygnału analizowanego $f(x)$ przez sygnał analizujący $\psi(x,m,n)$.

Falka, na podstawie której przy pomocy skalowania i translacji tworzy się zespół ortogonalnych funkcji bazowych, charakteryzuje specyficznymi właściwościami: dobrą lokalizacją w czasie (przestrzeni), czyli zwartym nośnikiem funkcji oraz ograniczonym pasmem w dziedzinie częstotliwości. Dobre umiejscowienie i selektywność w dziedzinie czasu powoduje, że falki doskonale nadają się do aproksymacji sygnałów zmiennych w czasie. Inną istotną cechą reprezentacji falkowej jest to, iż zawiera ona w sobie informacje o sygnale na różnych poziomach szczegółowości. Własność ta, określana mianem hierarchicznej rozdzielczości, jest szczególnie użyteczna w sieciach komputerowych, gdyż umożliwia w naturalny sposób progresywną transmisję, polegającą na przesyłaniu najpierw bardzo zgrubnego sygnału, a następnie dosyłaniu jego szczegółów. Poważnym atutem zastosowania transformacji falkowej do analizy sygnałów jest prostota obliczeniowa. Praktyczne wykorzystanie tego mechanizmu sprowadza się do obliczania splotów dwóch wektorów, z których jeden reprezentuje wartości próbek sygnału badanego, a drugi jest kilku lub kilkunastoelementowym wektorem reprezentującym filtr cyfrowy. Wspomniane własności falki powodują, że stała się ona „naturalnym” aproksymatorem sygnałów fizycznych. Podczas transformacji znaczna część współczynników $d_{m,n}$ jest bardzo mała w porównaniu do pozostałych i może być przyjęta za zero bez zauważalnego zniekształcenia aproksymowanej funkcji. Fakt ten decyduje o dużym znaczeniu transformaty w kompresji sygnałów i obrazów. Także w dziedzinie filtracji szumów falki mają olbrzymie praktyczne zastosowanie. Odpowiednie dobranie „falki matki” (służącej do generacji zespołu falek bazowych) zgodnie z charakterem badanego sygnału, pozwala precyzyjnie wyodrębnić właściwy sygnał spośród przypadkowych zakłóceń, które po transformacji będą reprezentowane przez współczynniki o małych wartościach, a te z kolei zostaną usunięte przez funkcję progową.

Transformacja falkowa jest stosunkowo nowym i ciągle jeszcze odkrywaniem narzędziem do obróbki sygnałów. Dobra aproksymacja oraz niski nakład obliczeniowy to główne atuty, które spowodowały, że już teraz reprezentacja falkowa jest coraz szerzej wykorzystywana w dziedzinach aplikacji takich jak: przetwarzanie sygnałów i obrazów, analiza numeryczna, astronomia, chemia oraz wielu innych.

Niniejsza praca ma na celu praktyczne zbadanie właściwości transformacji falkowej, będącej bardzo dobrze zapowiadającym się przedstawicielem metod czasowo – częstotliwościowych dekompozycji sygnałów. Szczegółowy nacisk położony został na zastosowanie przetwarzania falkowego w kompresji przebiegów EKG. W celu umożliwienia

przeprowadzenia stosownych badań napisana została aplikacja softwarowa wyposażona w mechanizm obliczeniowy transformaty falkowej (prostej i odwrotnej) oraz zawierająca obszerny zbiór falek o różnych właściwościach. Program ten operuje na bazie rzeczywistych sygnałów EKG obrazujących zróżnicowane przypadki pracy serca (pacjentów zdrowych i chorych). Niniejsza praca skupia się głównie na praktycznych badaniach nad zagadnieniem transformacji falkowej. Z tego powodu rozważania teoretyczne zostały ograniczone do postaci pozwalającej na ogólne wprowadzenie czytelnika w opracowywane zagadnienia, nie wnikając w „zawiłości” matematyczne wyprowadzeń i dowodów przytaczanych wzorów i zależności. Z drugiej strony duży nacisk położono na praktyczną realizację zagadnienia, prezentując szereg doświadczeń poruszających istotne zagadnienia i problemy związane z tą tematyką.

Zawartość niniejszej pracy przedstawia się następująco:

Rozdział drugi zawiera teoretyczne podstawy zagadnienia transformacji falkowej oraz podaje algorytmy rozkładu sygnału na współczynniki falkowe. Są tutaj przedstawione dwa klasyczne podejścia – analiza wielorozdzielcza oraz filtracja pasmowa sygnału.

W rozdziale trzecim ujęta jest krótka charakterystyka poszczególnych rodzin falek: Haar, Daubechies, Coiflets, Symlets i Biorthogonal.

W rozdziale czwartym zamieszczono ogólny opis sygnału EKG i jego najważniejszych właściwości.

Rozdział piąty zawiera szczegółowy opis aplikacji: wymagania systemowe, funkcjonalność oraz charakterystykę poszczególnych składowych programu. Znajduje się tutaj dokładna instrukcja użytkownika opisująca sposób posługiwania programem.

Szósty rozdział prezentuje przeprowadzone doświadczenia ułożone w pięć bloków tematycznych. Są to kolejno: dobór progu kompresji, poziomu dekompozycji i rodzaju falki, zagadnienie problemu brzegowego oraz użycie drugiego progu (koncepcja progowania czasowego). Przykłady te zostały wzbogacone dużą ilością ilustracji pozwalających na wizualną ocenę wyników i jakości otrzymywanych przebiegów. Każde doświadczenie zostało sfinalizowane wyczerpującymi wnioskami i refleksjami autora.

Zakończenie zawiera ogólne podsumowanie całej pracy oraz zwraca uwagę na celowość kontynuacji pewnych zagadnień dotyczących transformacji falkowej.

W załączniku znajdują się wydruki ważniejszych procedur wykorzystanych w programie wraz z opisem ich działania.

2. Transformacja falkowa

2.1. Informacje ogólne

Ciągła transformata falkowa sygnału $x(t)$ jest zdefiniowana w następujący sposób w dziedzinie czasu i częstotliwości:

$$CWT_x^T(t, a) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} x(\tau) \gamma^* \left(\frac{\tau - t}{a} \right) d\tau \quad (1a)$$

$$CWT_x^F(t, a) = \sqrt{|a|} \int_{-\infty}^{+\infty} X(f) A^*(af) e^{j2\pi ft} df \quad (1b)$$

Zaś transformacja do niej odwrotna jest określona zależnością:

$$x(t) = \frac{2}{C_y} \int_0^{+\infty} \left[\int_{-\infty}^{+\infty} CWT(\tau, a) g \left(\frac{t - \tau}{a} \right) d\tau \right] \frac{da}{a^2} \quad (2)$$

gdzie:

$$C_y = \int_{-\infty}^{+\infty} \frac{|\Gamma(f)|^2}{|f|} df < \infty, \quad \Gamma(0) = 0$$

Funkcja $g(t)$ oznacza falkę syntezującą, $\gamma(t)$ dualną do niej falkę analizującą, zaś $G(f)$ i $\Gamma(f)$ są ich widmami Fouriera.

Równanie (1a) reprezentuje filtrację sygnału analizowanego $x(t)$ przez sygnał analizujący $\gamma(t)$, przeskalowany w dziedzinie czasu współczynnikiem a („rozciągany” dla $a > 1$ oraz „ściskany” dla $a < 1$). Natomiast równoważne mu równanie (1b) przedstawia odwrotne przekształcenie Fouriera iloczynu widma sygnału i przeskalowanego widma falki $\gamma(t)$.

Ciągła czasowo-częstotliwościowa reprezentacja falkowa zawiera wiele nadmiarowych informacji. W wyniku spróbkowania jej parametrów, tzn. czasu t i współczynnika skali a , otrzymuje się współczynniki szeregu falkowego, analogiczne do współczynników szeregu Fouriera. Po podstawieniu:

$$t = n2^{-m}, \quad a = 2^{-m}$$

Uzyskuje się diadyczny szereg falkowy. Znalazł on obecnie najszersze praktyczne zastosowanie i nosi nazwę dyskretnej transformacji falkowej (DWT). Sygnał przedstawiony za pomocą DWT ma postać podwójnej sumy:

$$x(t) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} d_{m,n} g_{m,n}(t), \quad (3)$$

Gdzie współczynniki $d_{m,n}$ oblicza się z:

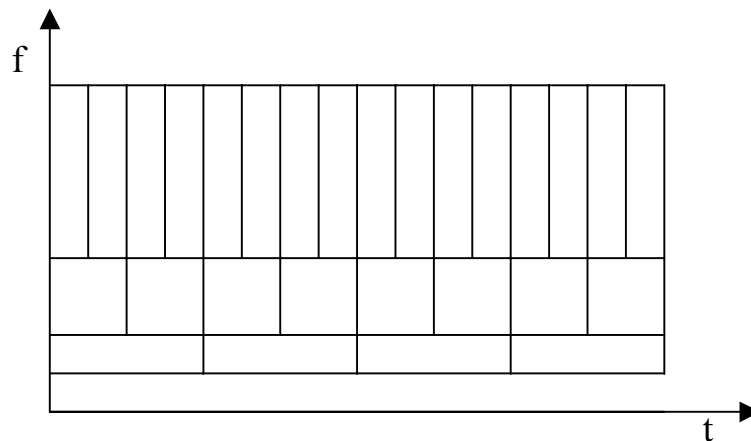
$$d_{m,n} = \int x(t) \gamma_{m,n}^*(t) dt,$$

A funkcje bazowe $g_{m,n}$ ze skalowania i translacji „falki matki”:

$$g_{m,n}(t) = 2^{-m/2} \gamma(2^{-m}t - n)$$

Funkcja $\gamma_{m,n}(t)$ jest funkcją dualną do $g_{m,n}(t)$. Przy założeniu ortogonalności bazy falkowej $\gamma_{m,n}(t) = g_{m,n}(t)$.

Cechą charakterystyczną dla dyskretnej transformacji falkowej jest stałość podstawowej komórki czasowo-częstotliwościowej tzw. atomu TF. Kiedy parametr skali m zwiększa się o 1, falki stają się dwa razy krótsze, ich widmo częstotliwościowe jest dwa razy szersze, czyli pole atomu TF pozostaje niezmiennie. W wyniku tego DWT realizuje schemat dekompozycji czasowo-częstotliwościowej przedstawiony na rysunku.



Rys.2.1.1 Rozmieszczenie atomów TF w szachownicy czasowo – częstotliwościowej dekompozycji falkowej

2.2. Analiza wielorozdzielcza

Transformacja falkowa jest ściśle związana z teorią analizy wielorozdzielczej. W analizie tej rozpatruje się ciąg przestrzeni funkcyjnych, w których dane sygnały cyfrowe aproksymowane są z różną dokładnością i wyrażane za pomocą różnej liczby funkcji bazowych. Owe przestrzenie funkcyjne, oznaczane przez V_m , $m \in \mathbb{Z}$, powinny spełniać następujące założenia:

- 1) $\dots \subset V_2 \subset V_1 \subset V_0 \dots$
- 2) $\bigcap_{m \in \mathbb{Z}} V_m = \{0\}$, $\overline{\bigcup_{m \in \mathbb{Z}} V_m} = L^2(\mathbb{R})$
- 3) $f(x) \in V_m$ wtedy i tylko wtedy gdy $f(2^m x) \in V_0$
- 4) jeżeli $f(x) \in V_0$ to $f(x-n) \in V_0$ dla wszystkich $n \in \mathbb{Z}$

Pierwszy z aksjomatów wyraża wzajemne zawieranie się kolejnych przestrzeni. Oznacza to, że przestrzeń V_{m-1} zawiera wszystkie funkcje z przestrzeni V_m oraz pewne dodatkowe funkcje, których w V_m nie ma. Aksjomat drugi świadczy o tym, iż nie istnieje funkcja, która należałaby do wszystkich przestrzeni i że domknięcie wszystkich przestrzeni daje przestrzeń funkcji o „ograniczonej” energii. Warunek trzeci wyraża fakt, że wszystkie przestrzenie są skalowaną wersją jednej przestrzeni (aspekt wielorozdzielczy). Czwarte założenie stwierdza natomiast iż dowolna funkcja po operacji translacji pozostaje w tej samej przestrzeni funkcyjnej, do której należała pierwotnie.

Dla przestrzeni o podanych wyżej własnościach możemy zdefiniować funkcję $\varphi(x) \in V_0$ taką, że $\varphi_{0,n}(x) = \varphi(x - n)$ stanowi bazę ortonormalną przestrzeni V_0 . Funkcja $\varphi(x)$ nazywana jest funkcją skalującą i odgrywa kluczową rolę w transformacji falkowej.

Przykładowo sygnał $x(t)$ może być przedstawiony w przestrzeni V_m na m -tym poziomie aproksymacji przy pomocy sumy poprzesuwaných funkcji bazowych $\varphi_{m,n}(t)$.

$$x(t) \approx \sum_n c_{m,n} \varphi(2^{-m}t - n), \quad c_{m,n} = \int x(t) \varphi^*(2^{-m}t - n) dt$$

Ten sam sygnał można również można z dwukrotnie mniejszą rozdzielczością w przestrzeni V_{m+1} (gorsza aproksymacja – mniej szczegółowa).

$$x(t) \approx \sum_n c_{m+1,n} \varphi(2^{-(m+1)}t - n) = \sum_n c_{m+1,n} \varphi\left(\frac{2^{-m}t}{2} - n\right)$$

$$c_{m+1,n} = \int x(t) \varphi^*\left(\frac{2^{-m}t}{2} - n\right) dt$$

Przy przejściu z przestrzeni V_m (bardziej szczegółowej) do przestrzeni V_{m+1} (mniej szczegółowej) część informacji zostaje utracona. Odrzucone szczegóły można umownie zdefiniować w przestrzeni W_{m+1} , która stanowi dopełnienie przestrzeni V_{m+1} do V_m , czyli:

$$V_m = V_{m+1} + W_{m+1}$$

Jeżeli przestrzeń V_{m+1} jest rozpinana przez funkcje $\varphi_{m+1,n}(t)$, to w przypadku przestrzeni W_{m+1} rolę tę pełnią funkcje $\psi_{m+1,n}(x)$, nazywane falkami i są one ortogonalne do $\varphi_{m+1,n}(t)$, a wspólnie natomiast rozpinają przestrzeń V_m . Wynika z tego fakt, że bez straty informacji sygnał z poziomu 'm' (przestrzeni V_m) można opisać za pomocą funkcji bazowych z przestrzeni niższego poziomu V_{m+1} i W_{m+1} :

$$x(t) \approx \sum_n c_{m,n} \varphi(2^{-m} t - n) = \sum_n c_{m+1,n} \varphi(2^{-(m+1)} t - n) + \sum_n d_{m+1,n} \psi(2^{-(m+1)} t - n)$$

Podobnie, jak powyżej, przestrzeń V_{m+1} można zastąpić za pomocą V_{m+2} i W_{m+2} . Analogicznie można definiować kolejne przestrzenie aproksymujące V oraz ich dopełnienia (do wyższych w hierarchii poziomów) W . W efekcie otrzymujemy drzewiastą strukturę przestrzeni z kolejnych poziomów:

$$\begin{aligned} V_m &= V_{m+k} + W_{m+k} + W_{m+k-1} + \dots + W_{m+2} + W_{m+1} \\ V_m &= V_{m+k} + \sum_{\rho=m+1}^{m+k} W_{\rho} \end{aligned} \quad (4)$$

Zależność (4) jest fundamentalnym równaniem wykorzystywanym w dekompozycji falkowej. Pozwala zapisać sygnał na różnych poziomach szczegółowości za pomocą zespołu wzajemnie ortogonalnych funkcji bazowych: falek $\psi(t)$ oraz funkcji skalujących $\varphi(t)$:

$$x(t) = \sum_n c_{m_k,n} \varphi(2^{-m_k} t - n) + \sum_{m=m_0}^{m_k} \sum_n d_{m,n} \psi(2^{-m} t - n) \quad (5)$$

Gdzie współczynniki:

$$c_{m_k,n} = \int x(t) \varphi^*(2^{-m_k} t - n) dt \quad (5a)$$

reprezentują aproksymację sygnału $x(t)$ na k -tym poziomie, natomiast współczynniki:

$$d_{m,n} = \int x(t) \psi^*(2^{-m} t - n) dt \quad (5b)$$

stanowią detale (szczegóły) z kolejnych poziomów przestrzeni funkcyjnych W .

Na funkcję skalującą oraz falek narzuca się następujące warunki:

$$\int \varphi(t)dt = 1, \quad \int \psi(t)dt = 0 \quad (6)$$

Są one niezwykle istotne w przypadku falki. Oznaczają zwarty nośnik funkcji $\psi(t)$, czyli wartość zerową w całym zakresie czasowym, za wyjątkiem kilku skupionych punktów różnych od zera.

Otrzymanie współczynników dekompozycji falkowej $c_{m,n}$ i $d_{m,n}$ na podstawie definicji (wzory 5a i 5b) jest dosyć kłopotliwe, dlatego w praktycznej realizacji numerycznej dyskretnej transformacji falkowej nie wykorzystuje się falek (i funkcji skalujących) tylko związane z nimi filtry. Zagadnienie to wyjaśnić można dwoma drogami: za pomocą metod cyfrowego przetwarzania sygnałów oraz teorii wielorozdzielczej aproksymacji sygnału.

Oparcie się na teorii wielorozdzielczej aproksymacji sygnału pozwoli wyprowadzić rekurencyjną zależność między współczynnikami dekompozycji falkowej $c_{m,n}$ i $d_{m,n}$ z dwóch kolejnych poziomów rozdzielczości. Punktem wyjścia jest zależność między przestrzeniami funkcyjnymi:

$$V_m = V_{m+1} + W_{m+1} \quad (7)$$

Czyli jeżeli mamy pewną przestrzeń funkcyjną aproksymacji V_0 oraz przestrzeń szczegółów W_0 to możemy zdefiniować przestrzeń „z wyższego poziomu” V_{-1} , która będzie „nadprzestrzenią” V_0 i W_0 . Skoro więc:

$$V_0 \subset V_{-1}, \quad W_0 \subset V_{-1}$$

to każdą funkcję należącą do przestrzeni V_0 i W_0 możemy opisać za pomocą funkcji bazowych przestrzeni V_{-1} . Ponieważ funkcja skalująca $\varphi_0(t)$ należy do V_0 , a falka $\psi_0(t)$ należy do W_0 oraz $\{\varphi_{-1,n}(t), n \in \mathbb{Z}\}$ jest bazą ortonormalną w V_{-1} to:

$$\varphi(2^0 t) = \sum_n h(n)\varphi(2^1 t - n) \quad (8a)$$

$$\psi(2^0 t) = \sum_n g(n)\varphi(2^1 t - n) \quad (8b)$$

Na podstawie (8a) wynika, że funkcja skalująca na poziomie niższym jest liniową kombinacją kilku funkcji skalujących z poziomu wyższego. Ponieważ na poziomie niższym aproksymacja jest mniej szczegółowa (bardziej niskoczęstotliwościowa), to współczynniki wagowe $h(n)$ pełnią rolę filtru dolnoprzepustowego. Analogicznie falka na poziomie niższym jest liniową kombinacją kilku funkcji skalujących z poziomu wyższego (8b), tym razem jednak reprezentuje szczegóły. Zatem współczynniki $g(n)$ reprezentują filtr górnoprzepustowy.

Znając zależności między funkcjami bazowymi z dwóch kolejnych poziomów przestrzeni funkcyjnych (8a), (8b) oraz wykorzystując równania definicyjne współczynników $c_{m,n}$ (5a) i

$d_{m,n}$ (5b) można wyprowadzić zależność pomiędzy współczynnikami rozwinięcia na tych poziomach:

$$c_m(n) = \sum_k h(n - 2k)c_{m-1}(k) \quad (9a)$$

$$d_m(n) = \sum_k g(n - 2k)c_{m-1}(k) \quad (9b)$$

Są to kluczowe wzory definiujące związek pomiędzy współczynnikami z dwóch kolejnych poziomów rozwinięcia falkowego. Właśnie te zależności mają obecnie zastosowanie w praktycznych algorytmach transformacji falkowych sygnałów. Mają one swoją praktyczną interpretację: Współczynniki $c_{m,n}$ poziomu niższego otrzymuje się drogą filtracji dolnoprzepustowej współczynników $c_{m-1,n}$ z poziomu wyższego za pomocą filtru $h(n)$ oraz następującej po niej decymacji rzędu drugiego tzw. „downsamplingu” (usunięcie co drugiej wartości), zaś współczynniki $d_{m,n}$ w wyniku filtracji górnoprzepustowej tych samych współczynników $c_{m-1,n}$ za pomocą filtru $g(n)$ i analogicznej decymacji. Sam proces filtracji, lub inaczej splotu dwóch sygnałów zdefiniowany jest następująco:

$$c'_m(n) = \sum_k h(n - k)c_{m-1}(k)$$

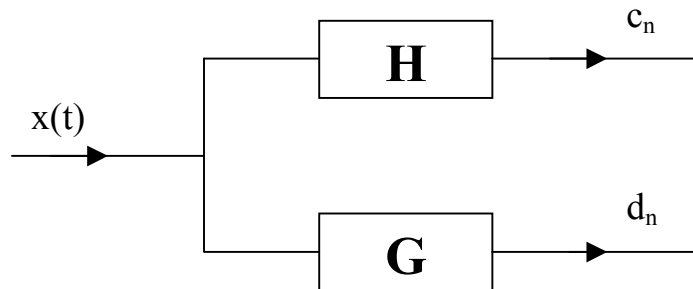
Decymacja sygnału jest natomiast zrealizowana przez:

$$c_m(n) = c'_m(2n)$$

Wzór (9) jest zwięźlejszym zapisem dwóch powyższych operacji.

2.3. Filtracja pasmowa sygnału

Wykorzystując metody cyfrowego przetwarzania sygnałów do wyznaczenia współczynników $c_{m,n}$ i $d_{m,n}$ uwzględnia się fakt, iż funkcje skalujące $\varphi(t)$ posiadają widmo częstotliwościowe odpowiadające filtrowi dolnoprzepustowemu, natomiast widmo falki $\psi(t)$ przypomina widmo filtra pasmowoprzepustowego (górnoprzepustowego). W tym celu należy zaprojektować odpowiednie filtry, których pasma przepuszczania będą odpowiadały widmom funkcji bazowych ($\varphi(t)$ i $\psi(t)$). Przefiltrowanie sygnału $x(t)$ przez zespół „filtrów bazowych” da w efekcie ciąg współczynników $c_{m,n}$ i $d_{m,n}$. Pojedynczy cykl filtracji (na jednym poziomie dekompozycji) jest zaprezentowany na rysunku:



Rys.2.3.1 Filtracja sygnału $x(t)$ filtry dolnoprzepustowym H i górnoprzepustowym G

H jest filtrem dolnoprzepustowym o transmitancji $H(\omega)$ odpowiadającej funkcji $\varphi(t)$, natomiast G jest filtrem górnoprzepustowym o transmitancji $G(\omega)$ odpowiadającym funkcji $\psi(t)$. Pojedynczy etap filtracji przedstawiony na rysunku dokonuje rozszczepienia sygnału na dwa sygnały: dolnopasmowy i górnopasmowy, czyli wyodrębnia „szkielet” i szczegóły sygnału. Z numerycznego punktu widzenia operacja filtracji sygnału nie przedstawia żadnych problemów. W dziedzinie częstotliwości transformata sygnału na wyjściu jest równa iloczynowi transmitancji sygnału wejściowego i transformaty filtru:

$$C(\omega) = H(\omega) * X(\omega)$$

$$D(\omega) = G(\omega) * X(\omega)$$

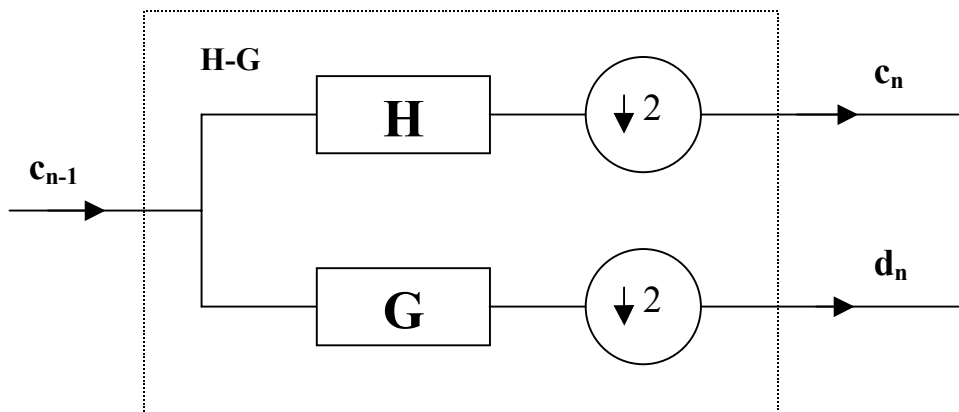
W dziedzinie czasu natomiast operacja ta jest splotem (konwolucją) sygnału wejściowego i odpowiedzi impulsowej filtru $h(n)$, $g(n)$ (sygnałem otrzymanym przy wymuszeniu filtru „delta diraca” $\delta(t)$):

$$c(n) = \sum_k h(n-k)x(k) \quad (10a)$$

$$d(n) = \sum_k g(n-k)x(k) \quad (10b)$$

Powyższe zależności są analogiczne do wzorów (9a), (9b) wyprowadzonych w oparciu o teorię wielorozdzielczej aproksymacji sygnału. Brak w nich jedynie mechanizmu decymacji, czyli usuwania co drugiej próbki.

Po uwzględnieniu decymatorów i założeniu, że sygnał wejściowy $x(t)$ jest równy współczynnikom aproksymacji z wyższego poziomu $c(n)$, postać zależności (10a) i (10b) jest już identyczna ze wzorami (9a), (9b) wyprowadzonymi na podstawie teorii wielorozdzielczej analizy sygnałów. Natomiast układ filtracji z rys.2.3.1 zmieni postać na przedstawioną na rys.2.3.2 i będzie realizował proces pojedynczego poziomu dekompozycji polegającego na uzyskaniu współczynników niższego poziomu na podstawie współczynników z wyższego poziomu.



Rys.2.3.2 Pojedynczy etap dekompozycji falkowej

Należy zauważyć, że dzięki operacji „downsamplingu” powyższy algorytm nie wprowadza zmian co do długości obrabianego sygnału. Sam proces filtracji spowodowałby, że długość sygnału c_n oraz długość d_n byłaby równa długości sygnału c_{n-1} , co dwukrotnie zwiększyłoby ilość danych potrzebnych do magazynowania tej samej informacji. Usunięcie co drugiej próbki nie wprowadza żadnych zniekształceń przenoszonej informacji (umożliwia pełną rekonstrukcję sygnału) oraz pozwala zachować pierwotną długość sygnału.

Przedstawiony powyżej układ filtracji i „downsamplingu” stanowi pojedynczy poziom dekompozycji falkowej (oznaczony „H-G”), która w swojej idei ma za zadanie rozłożyć sygnał na wiele sygnałów niosących informacje o sygnale pierwotnym na różnych poziomach szczegółowości. Ogólny schemat obliczania transformaty falkowej przedstawia się według poniższego algorytmu:

Sygnał poddawany transformacji falkowej zawierający N próbek wygodnie jest zapisać za pomocą wektora (indeks N za nawiasem oznacza długość wektora):

$$c_0 = [c_{0,0} \quad c_{0,1} \quad \dots \quad c_{0,N-1}]_N$$

Po pierwszym kroku dekompozycji transformata falkowa tego sygnału jest wektorem o tej samej długości tzn. zawiera $N/2$ współczynników $c_{1,k}$ i $N/2$ współczynników $d_{1,k}$.

$$[c_{1,0} \quad c_{1,1} \quad \dots \quad c_{1,N/2-1}]_{N/2} [d_{1,0} \quad d_{1,1} \quad \dots \quad d_{1,N/2-1}]_{N/2}$$

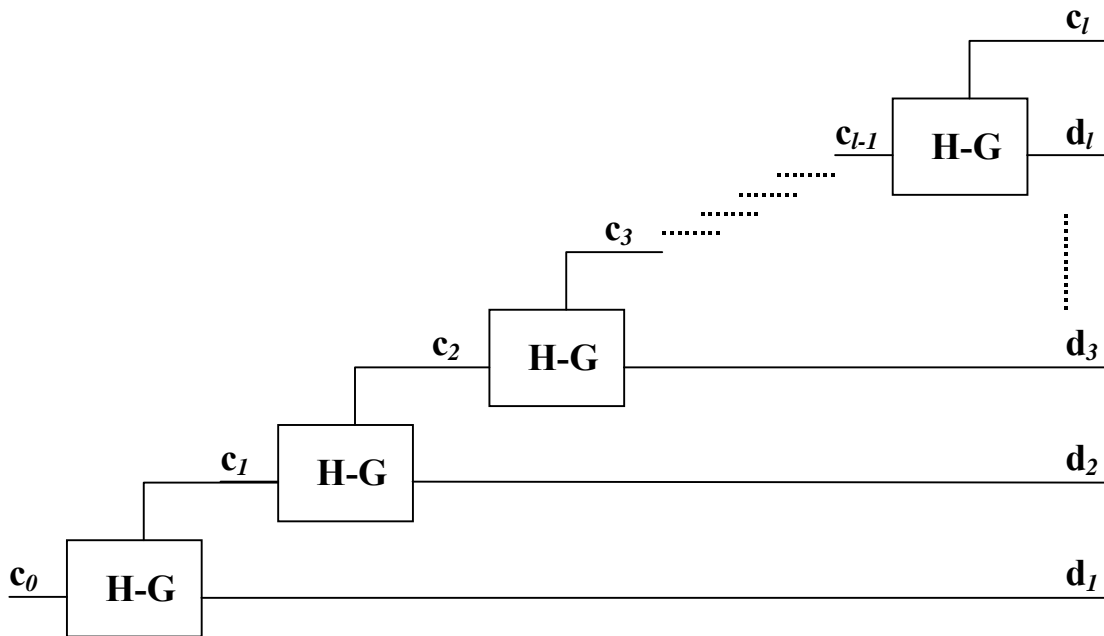
W następnej iteracji transformata falkowa będzie miała elementy:

$$[c_{2,n}]_{N/4}, [d_{2,n}]_{N/4}, [d_{1,n}]_{N/2}$$

Po l -krotnym zastosowaniu tej procedury otrzymuje się transformatę falkową wektora $[c_{0,n}]_N$ w postaci wektora

$$[c_{l,n}]_{N/(2^l)}, [d_{l,n}]_{N/(2^l)}, \dots, [d_{2,n}]_{N/4}, [d_{1,n}]_{N/2}$$

Jeżeli N jest potęgą 2 to l może osiągnąć wartość $\log_2 N$. Na szczycie piramidy będzie pojedynczy współczynnik reprezentujący aproksymację (od funkcji skalującej), poniżej zaś same współczynniki reprezentujące szczegóły (od falek).



Rys.2.3.3 Schemat piramidy wielopoziomowej dekompozycji falkowej

Kolejne wartości współczynników wyznacza się z zależności:

$$c_{m,n} = \sum_k h_k c_{m-1,2n-k}, \quad d_{m,n} = \sum_k g_k c_{m-1,2n-k}$$

$h_k = [h_{k,0}, h_{k,1}, \dots, h_{k,L-1}]$ jest odpowiedzią impulsową filtra dolnoprzepustowego,
 $g_k = [g_{k,0}, g_{k,1}, \dots, g_{k,L-1}]$ jest odpowiedzią impulsową filtra górnoprzepustowego,
 $n = 0, 1, \dots, N/2-1$
 $\max(0, 2n-L+1) \leq k \leq \min(2n, L-1)$

Użyte wyżej filtry muszą charakteryzować się określonymi właściwościami:

- 1) muszą mieć skończoną odpowiedź impulsową FIR (ang. finite impulse response),
- 2) powinny zapewniać dokładną rekonstrukcję sygnału, tj. spełniać równanie

$$c_{m-1,n} = \sum_k [h_{2k-n}c_{m,k} + g_{2k-n}d_{m,k}]$$

gdzie:

$$h_k = [h_{k,L-1}, h_{k,L-2}, \dots, h_{k,1}, h_{k,0}],$$

$$g_k = [g_{k,L-1}, g_{k,L-2}, \dots, g_{k,1}, g_{k,0}],$$

$$n = 0, 1, \dots, N-1$$

$$n/2 \leq k \leq \min(N/2-1, n+L-1)$$

- 3) powinny być filtrami liniowo-fazowymi (ang. linear-phase), tj. takimi, których transmitancja wyraża się wzorem:

$$H(e^{j\omega}) = A(\omega) \exp[-j\Phi(\omega)], \quad \Phi(\omega) = [k\pi + (L-1)\omega]/2,$$

$$k=0 \text{ dla filtru symetrycznego (} h_{L-n-1} = h_n, \quad n=0, 1, \dots, L-1 \text{)}$$

$$k=1 \text{ dla filtru antysymetrycznego (} h_{L-n-1} = -h_n, \quad n=0, 1, \dots, L-1 \text{)}$$

W rzeczywistości wszystkie trzy warunki spełnione są tylko dla tzw. falki Haar'a, która jest przypadkiem trywialnym i nie ma w praktyce większego zastosowania. Projektowane obecnie filtry spełniają dwa pierwsze warunki i są to tzw. filtry „kwadraturowe” (FIR-QMF, ang. Finite Impulse Response – Quadrature Mirror Filter). Jest to zespół czterech filtrów (dwóch stosowanych do analizy - h , g i dwóch do syntezy - h' , g') ściśle powiązanych ze sobą i odpowiadających jednej falce.

Z praktycznego punktu widzenia projektowanie falek zdeterminowane jest stworzeniem zespołu filtrów kwadraturowych o żądanych właściwościach, z których jednoznacznie wynika kształt falki. Same filtry natomiast tworzy się według poniższego algorytmu:

1. ustala się odpowiedź impulsową filtru dolnoprzepustowego analizy h (pochodzącego od funkcji skalującej) spełniającą następujące ograniczenia:

$$\sum_{n=0}^{L-1} (h_n)^2 = 1,$$

$$\sum_{n=0}^{L-1} h_{2k+n} h_n = 0$$

Pierwsze ograniczenie wynika z „unormowania” odpowiedzi filtru (równości norm funkcji skalujących dla różnych rozdzielczości). Często przekształca się je do prostszej postaci:

$$\sum_{n=0}^{L-1} h_n = \sqrt{2} \tag{11}$$

Drugi warunek wynika natomiast z ortogonalności translacji funkcji skalującej. Wymnożenie filtru przez „siebie” z przesunięciem o parzystą ilość próbek daje wartość zerową.

Powyższe dwa ograniczenia z reguły nie określają jednoznacznie postaci filtru, lecz pozostawiają jeszcze kilka stopni swobody, które wykorzystuje się w celu uzyskania żądanych, specyficznych właściwości falki.

2. Pozostałe zaś filtry: górnoprzepustowy analizy g i obydwa filtry syntezy h' i g' zdefiniowane są za pomocą filtru h :

$$g_n = (-1)^n h_{L-1-n},$$

$$h'_n = h_{L-1-n},$$

$$g'_n = g_{L-1-n} = (-1)^{n+1} h_n$$

Czyli, jeżeli zaprojektowany filtr h ma odpowiedź impulsową:

$$(h_n) = (h_0, h_1, h_2, h_3)$$

to postać pozostałych filtrów będzie następująca:

$$(g_n) = (h_3, -h_2, h_1, -h_0)$$

$$(h'_n) = (h_3, h_2, h_1, h_0)$$

$$(g'_n) = (-h_0, h_1, -h_2, h_3)$$

2.4. Rekonstrukcja

Rekonstrukcja sygnału jest procesem odwrotnym do dekompozycji. Stanowi wyodrębnienie sygnału pierwotnego z szeregu współczynników falkowych otrzymanych w wyniku dekompozycji. Do rekonstrukcji wykorzystuje się dwa filtry syntezy z zestawu filtrów kwadraturowych: dolnoprzepustowy h^{\cdot} i górnoprzepustowy g^{\cdot} .

Pojedynczy cykl procesu rekonstrukcji odtwarza sygnał zawierający się w przestrzeni V_m z sygnału wygładzonego z przestrzeni V_{m+1} i z sygnału szczegółowego należącego do przestrzeni W_{m+1} . Odbywa się to zgodnie ze wzorem:

$$\begin{aligned} c_{m,n} &= \sum_k [h^{\cdot}_{2k-n} c_{m+1,k} + g^{\cdot}_{2k-n} d_{m+1,k}] \\ &= \sum_k h^{\cdot}_{2k-n} c_{m+1,k} + \sum_k g^{\cdot}_{2k-n} d_{m+1,k} \end{aligned} \quad (12)$$

Realizacja numeryczna wzoru (12) jest dosyć kłopotliwa, dlatego też w praktyce stosuje się przekształcenie według poniższego algorytmu:

$$\begin{aligned} \sum_k h^{\cdot}_{2k-n} c_{m+1,k} &= \sum_k h^{\cdot}_{k-n} c^{\cdot}_{m+1,k} \\ \text{gdzie: } c^{\cdot}_m &= c_{2m}. \end{aligned} \quad (13)$$

$$\begin{aligned} \sum_k g^{\cdot}_{2k-n} d_{m+1,k} &= \sum_k g^{\cdot}_{k-n} d^{\cdot}_{m+1,k} \\ \text{gdzie: } d^{\cdot}_m &= d_{2m} \end{aligned}$$

wobec czego wzór (12) zmieni się do postaci:

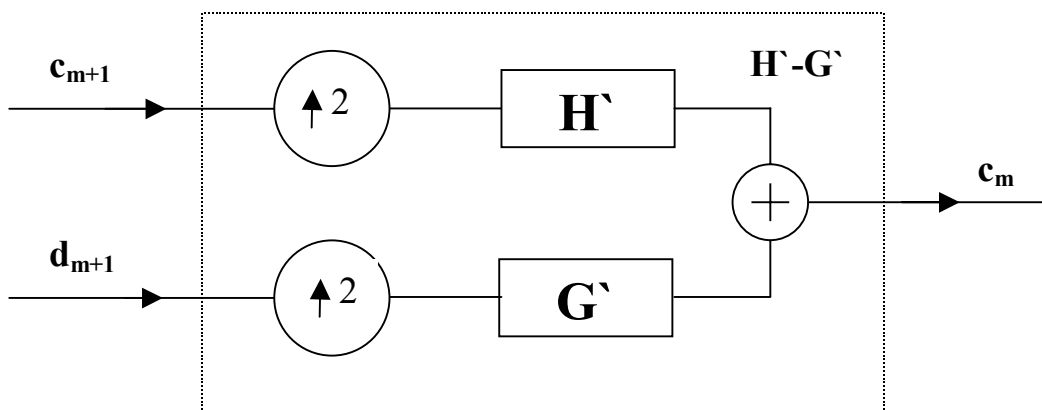
$$c_{m,n} = \sum_k h^{\cdot}_{k-n} c^{\cdot}_{m+1,k} + \sum_k g^{\cdot}_{k-n} d^{\cdot}_{m+1,k} \quad (14)$$

Zamiast sumowania po parzystych indeksach odpowiedzi impulsowej filtru i oryginalnego sygnału (jak sugeruje wzór (12)), powyższe sumowanie rozciąga się na wszystkie elementy filtru i sygnału przekształconego poprzez dopróbkowanie za pomocą tzw. ekspanderów. Powodują one dwukrotne wydłużenie sygnału wstawiając pomiędzy oryginalnymi próbkami dodatkowych próbki zerowe:

$$\begin{aligned} c^{\cdot}_{m+1,k} &= (c_{m+1,0}, 0, c_{m+1,1}, 0, c_{m+1,2}, 0, c_{m+1,3}, 0, c_{m+1,4}, \dots) \\ d^{\cdot}_{m+1,k} &= (d_{m+1,0}, 0, d_{m+1,1}, 0, d_{m+1,2}, 0, d_{m+1,3}, 0, d_{m+1,4}, \dots) \end{aligned} \quad (15)$$

Wyrażenie (14) jest równoważne definicji rekonstrukcji falkowej (12). Określa ono praktyczny algorytm obliczania próbek sygnału z współczynników falkowych.

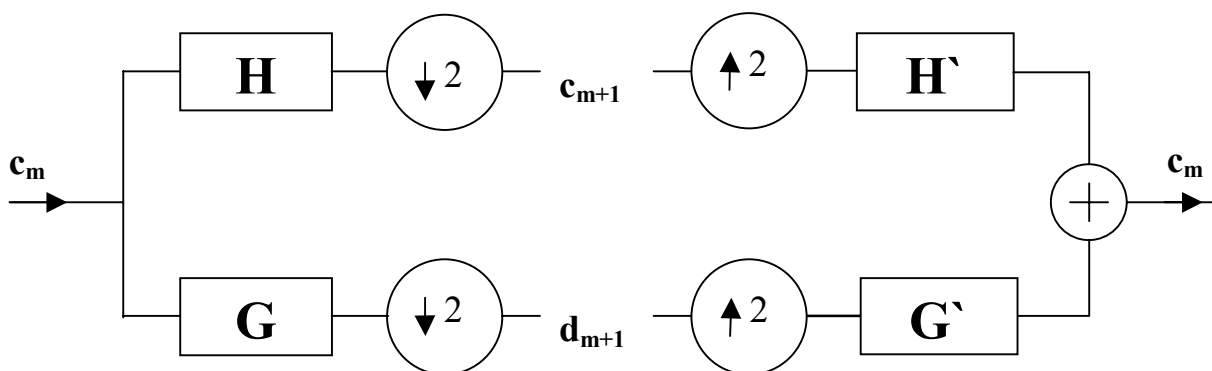
Schemat pojedynczego poziomu rekonstrukcji przedstawia poniższy rysunek:



Rys.2.4.1 Pojedynczy etap rekonstrukcji sygnału

Obydwa sygnały: wygładzony c_{m+1} i reprezentujący szczegóły d_{m+1} są podawane na tzw. ekspandery (oznaczone kółkiem ze strzałką w górę i 2), które dokonują procesu „upsamplingu”, czyli dwukrotnego wydłużenia sygnału poprzez dołożenie między wszystkimi próbkami sygnału pierwotnego próbek zerowych. Powstaną wtedy ciągi próbek mające postać (15). Następnie otrzymane sygnały są filtrowane odpowiednio filrami syntezy dolnoprzepustowym h' i górnoprzepustowym g' , po czym z kolei następuje operacja sumowania. Na wyjściu sumatora pojawia się zrekonstruowany sygnał c_m należący do przestrzeni V_m .

Na podstawie dotychczasowych rozważań wynika, że transformacja falkowa jest transformacją bezstratną, tzn. można sygnał poddać procesowi dekompozycji, a następnie współczynniki falkowe (będące próbkami sygnałów z przestrzeni o niższych rozdzielczościach) poddać procesowi rekonstrukcji, by w efekcie otrzymać sygnał identyczny z oryginalnym. Pojedynczy poziom dekompozycji i rekonstrukcji falkowej jest przedstawiony na rysunku:



Rys.2.4.2 Pojedynczy etap dekompozycji i rekonstrukcji pokazujące prostą i odwrotną transformację falkową bez straty informacji

2.5. Zasada zachowania energii

Transformacja falkowa jest przekształceniem bezstratnym. Podczas przejścia sygnału z dziedziny czasu do postaci współczynników falkowych cała energia zostaje zachowana. Właściwość tę precyzyjnie określa wzór:

$$\sum_{n=0}^{N-1} |s(n)|^2 = \frac{1}{N} \sum_{v=-N/2+1}^{N/2} |\tilde{s}(v)|^2 = N^2 |S_{0,0}|^2 + \sum_{j=-p}^{-1} \sum_{k=0}^{2^j N-1} |S_{j,k}|^2$$

gdzie:

$s(n)$ – wartość n -tej próbki sygnału

N – długość sygnału

$S_{j,k}$ – k -ty współczynnik detali z j -tego poziomu dekompozycji

$S_{0,0}$ – współczynnik aproksymacji stojący na piramidzie współczynników falkowych po p -krotnej dekompozycji falkowej (tuż obok odpowiadającego mu współczynnika detali $S_{-p,0}$)

j – indeks skalowania

k – indeks translacji

Zależność ta definiuje zasadę zachowania energii sygnału podczas transformacji falkowej prostej (dekompozycji) i odwrotnej (rekonstrukcji). Mówi ona o tym, że cała energia sygnału będąca sumą kwadratów wartości wszystkich próbek jest równa sumie kwadratów wszystkich współczynników uzyskanych w procesie dekompozycji falkowej.

3. Falki

3.1. Funkcja Haar'a

Wykorzystywane w praktyce falki nie mają prostych postaci analitycznych. Stosunkowo nieskomplikowaną falką jest funkcja Haar'a. Nie ma ona jednak większego zastosowania praktycznego, obecnie wykorzystuje się ją raczej do celów dydaktycznych. Jej definicja wyraża się następująco:

$$\psi(x) = \begin{cases} 1 & 0 \leq x < 0,5 \\ -1 & 0,5 \leq x < 1 \\ 0 & \end{cases}$$

Jest jedyną falką, która posiadając własność ortogonalności, posiada również oś antysymetrii. Oznacza to, że odpowiadający jej filtr nie wprowadza nieliniowego przesunięcia fazowego pomiędzy sygnałem na wejściu, a sygnałem na wyjściu. Daje to między innymi możliwość kaskadowego łączenia filtrów bez konieczności kompensacji fazy sygnału.

3.2. Falka Daubechies

Falka Daubechies jest obecnie jedną z najpopularniejszych i najchętniej wykorzystywanych falek. Jej główne atuty to zwarty nośnik o szerokości $2p-1$, dokładna aproksymacja funkcji stałych i liniowych oraz stosunkowo prosta postać.

Współczynniki odpowiedzi impulsowej filtrów generujących falkę 'db2' o nośniku od 0 do 3 (h_0, h_1, h_2, h_3) można wyznaczyć przy pomocy zależności:

Z warunku (11) współczynniki muszą spełniać:

$$h_0 + h_1 + h_2 + h_3 = \sqrt{2}$$

Z warunku ortogonalności wynika:

$$h_0 h_2 + h_1 h_3 = 0$$

Otrzymano układ dwóch równań z czterema niewiadomymi. Aby „wypełnić” pozostałe dwa stopnie swobody należy stworzyć stowarzyszony z ciągiem (h_n) ciąg (g_n) , który ma postać $(h_3, -h_2, h_1, -h_0)$. Przy założeniu, że będzie on ortogonalny do ciągu stałego $(1, 1, 1, 1)$ oraz do ciągu liniowego $(1, 2, 3, 4)$ otrzymamy brakujące dwa równania:

$$h_3 - h_2 + h_1 - h_0 = 0$$

$$h_3 - 2h_2 + 3h_1 - 4h_0 = 0$$

Rozwiązaniem tych równań są współczynniki odpowiedzi impulsowych filtrów kwadraturowych generujące falę 'db2':

$$h_0 = \frac{1 + \sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3 + \sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3 - \sqrt{3}}{4\sqrt{2}}, \quad h_3 = \frac{1 - \sqrt{3}}{4\sqrt{2}}$$

3.3. Falki Coiflets

Falka 'Coiflets' posiada stosunkowo małą asymetrię spośród ortogonalnych falek o zwartym nośniku. Skonstruowano je przy założeniu, że mają posiadać znikające momenty ich funkcji skalujących:

$$\int x^l \varphi(x) dx = 0, \quad \text{dla } l = 1, 2, \dots, p - 1$$

Nośnik falki 'Coiflets' ma szerokość $3p-1$.

3.4. Falki Biortogonalne

W falkach 'biortogonalnych' rozwiązany został problem asymetrii. Opisana wcześniej dekompozycja i rekonstrukcja falkowa charakteryzowała się tym, że wszystkie cztery filtry pochodziły od jednego filtru H . Filtr ten był filtrem niesymetrycznym i niesymetryczna była odpowiadająca mu falka. Filtry niesymetryczne wprowadzają nieliniowe przesunięcie fazowe pomiędzy sygnałem na wejściu i sygnałem na wyjściu, co w przypadku pewnych zastosowań może być wadą.

Tradycyjne podejście do projektowania filtrów zostało uogólnione na przypadek dwóch różnych (niezależnych) filtrów dolnoprzepustowych analizy (dekompozycji) H i syntezy (rekonstrukcji) H' , w taki sposób, że filtry mogą być symetryczne i tym samym o liniowym przesunięciu fazowym. Osiąga się to za cenę rezygnacji z ortogonalności falek odpowiadających filtrowi H , (równoważnej warunkowi $\sum_n h_{2k+n}h_n=0$) na rzecz tzw. biortogonalności, określonej warunkiem:

$$\sum_n h'_n h_{2k+n} = 0$$

W konsekwencji przetwarzany sygnał wyrażany jest poprzez transformatę falkową jako:

$$f = \sum_m \sum_n \langle f, \psi_{m,n} \rangle \psi'_{m,n}$$

Oznacza to, że analiza sygnału dokonywana jest w jednej bazie falkowej, natomiast synteza w drugiej. Obie bazy nie są ortogonalne, ale pary falek, jedna z bazy analizy, druga z bazy syntezy, są względem siebie ortogonalne. Filtry górnoprzepustowe oblicza się ze wzoru:

$$g'_n = (-1)^{n+1} h_{-n+1},$$

$$g_n = (-1)^{n+1} h'_{-n+1}$$

Filtry analizy h i g mogą mieć jedną z następujących postaci:

- obydwa filtry są symetryczne i nieparzystej długości i długości ich różnią się o liczbę stanowiącą iloczyn liczby nieparzystej i 2
- jeden filtr jest symetryczny a drugi niesymetryczny; długości obydwu są parzyste i takie same lub różnią się o liczbę stanowiącą iloczyn liczby parzystej i 2
- jeden filtr ma długość nieparzystą, drugi parzystą; obydwa mają wszystkie zera na okręgu jednostkowym; obydwa są symetryczne lub jeden jest symetryczny a drugi niesymetryczny.

3.5. Porównanie parametrów falek

W poniższej tabeli zebrano podstawowe informacje na temat poszczególnych rodzin falek.

Rodzina	Haar	Daubechies	Symlets	Coiflets	Biorthogonal
Skrócona nazwa	'haar'	'db'	'sym'	'coif'	'bior'
Index	-	N	N	N	N_d, N_r
Przykład	haar	db2	sym4	coif3	bior3.7
Ortogonalność	tak	tak	tak	tak	nie
Biortogonalność	tak	tak	tak	tak	tak
Długość filtru	2	2N	2N	6N	$\max(2N_d, 2N_r)+2$
Symetria	tak	silna niesymetria	niewielka niesymetria	niewielka niesymetria	tak
Moment zaniku falki	1	N	N	2N	$N_r - 1$
Moment zaniku funkcji skalującej	-	-	-	2N-1	-

Tab.3.5.1 Podstawowe parametry falek

4. Sygnał EKG

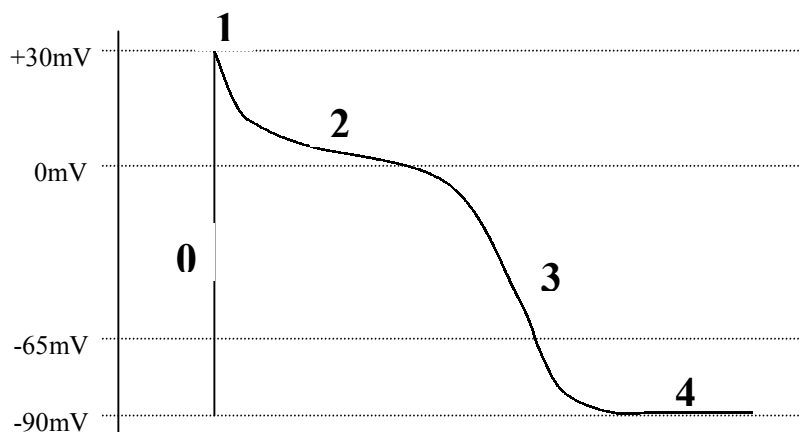
Niniejszy rozdział ma na celu przedstawienie powiązania między sygnałem EKG, będącym obiektem przetwarzania falkowego, a zjawiskami fizycznymi zachodzącym w sercu. Na podstawie elektrycznych właściwości i zachowań komórki mięśnia sercowego wyjaśniona jest zasada działania pracy serca. Zamieszczone zostały podstawowe wiadomości z zakresu elektrokardiografii określające sposób pomiaru akcji serca oraz opisujące kolejne składowe przebiegu EKG.

4.1. Zjawiska elektryczne w pojedynczej komórce mięśnia sercowego

Mięsień serca będącym przebiegiem badań elektrokardiografii osłonięty jest przez tkanki i przez to niedostępny do bezpośredniej obserwacji. Wiadomości o aktywności elektrycznej mięśnia sercowego można czerpać za pomocą badania pola elektrycznego, wzbudzanego przez serce i towarzyszącego cyklowi jego pracy. Podobnie jak w innych komórkach ustroju, procesom życiowym włókien mięśniowych serca towarzyszą zmiany napięć elektrycznych. Źródłem prądu elektrycznego w komórce mięśnia sercowego jest *potencjał dyfuzyjny*, który powstaje w wyniku istnienia gradientu stężenia jonów znajdujących się po obu stronach półprzepuszczalnej błony komórkowej. Potencjał dyfuzyjny zależy nie tylko od gradientu stężenia danych jonów, lecz także od przepuszczalności błony komórkowej dla danego typu jonu. Stężenie potasu K^+ wewnątrz komórki mięśnia serca jest o wiele wyższe, niż za zewnątrz, a stężenie sodu Na^+ w komórce jest niższe, niż na zewnątrz.

W stanie spoczynku błona komórkowa jest względnie przepuszczalna dla potasu, natomiast nieprzepuszczalna dla sodu. Z tego powodu siła elektromotoryczna komórki pozostającej w spoczynku jest wywołana istnieniem potencjału dyfuzyjnego potasu, zaś gradient stężenia sodu nie odgrywa większej roli. Wnętrze pozostającej w spoczynku komórki wykazuje potencjał ujemny, rzędu $-90mV$, w odniesieniu do otoczenia komórki. Jest to tzw. *spoczynkowy potencjał błonowy*.

Jeżeli komórka zostanie pobudzona bodźcem o odpowiedniej amplitudzie potencjał wewnątrz komórki wzrasta. Gdy jego wartość przekroczy tzw. potencjał progowy wynoszący ok. $-65mV$ następuje szybka (ok. 1-2ms) depolaryzacja wnętrza komórki, zablokowanie kanałów sodowo-potasowych i wzrost potencjału do ok. $+30mV$. Tak powstałą zmianę potencjału błonowego nazywa się *impulsem czynnościowym*. Jego kształt pokazany jest na rysunku 4.1.1.



Rys.4.1.1 Potencjał pojedynczego włókna mięśnia sercowego podczas pobudzenia

Impuls ten składa się z pięciu charakterystycznych faz:

Faza 0 – to faza *szybkiej depolaryzacji*. Pobudzenie elektryczne, zmniejszające ujemny potencjał błonowy do wartości -65mV , prowadzi do nagłego wzrostu przepuszczalności błony komórkowej dla sodu. Szybki dośrodkowy prąd sodowy powoduje szybką depolaryzację większości komórek mięśnia sercowego podczas, gdy wpływ stężenia potasu jest ograniczony. Jednocześnie od wartości potencjału błonowego w granicach od -55 do -60mV rozpoczyna się napływ wapnia do wnętrza komórki. Wolny dośrodkowy prąd wapniowy zapoczątkowuje skurcz włókna mięśniowego.

Faza 1 – faza wstępnej *szybkiej repolaryzacji* – zależy od napływu jonów chloru do wnętrza komórki.

Faza 2 – to okres *powolnej repolaryzacji*. W tym czasie dośrodkowe i odśrodkowe prądy jonowe: napływ wapnia i rozpoczynający się wypływ potasu równoważą się, a potencjał błony pozostaje we względnej stabilizacji. Faza ta jest cechą specyficzną komórek mięśnia sercowego, gdyż trwa ok. 100ms. Prawie do końca tej fazy komórka jest w stanie tzw. refrakcji względnej, czyli nie reaguje w tym czasie na żadne, nawet bardzo silne pobudzenia. Okres ten jest głównym powodem ograniczenia maksymalnej częstotliwości skurczów serca.

Faza 3 – jest okresem *szybkiej repolaryzacji*, w którym dominuje odśrodkowy prąd potasowy, prowadzący do szybkiego narastania ujemnego ładunku wnętrza komórki. Od wartości około -65mV znowu zmniejsza się przepuszczalność błony komórkowej dla sodu.

Faza 4 – to faza *spoczynku*. Między wnętrzem komórki, a jej powierzchnią panuje spoczynkowy potencjał błonowy wynoszący -90mV .

Czas trwania potencjału czynnościowego w roboczym włóknie mięśnia komór wynosi około 300ms, w tym faza szybkiej depolaryzacji kilka milisekund.

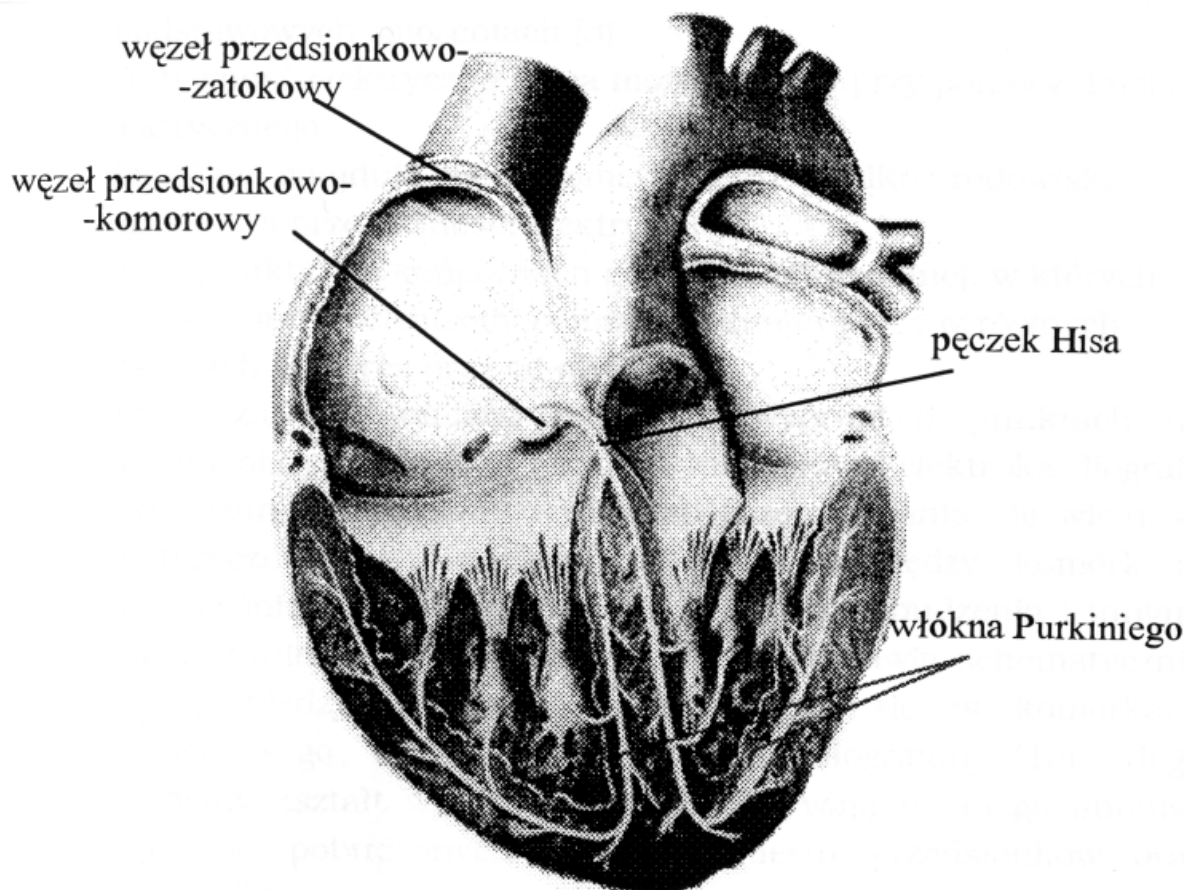
4.2. Aktywność elektryczna serca, a przebieg EKG

Pobudzone elektrycznie włókna mięśnia serca wytwarzają zmieniające się stale w czasie trwania cyklu pracy serca pole elektryczne. Kolejność ich pobudzenia oraz ilość czynnych elektrycznie włókien w danym momencie czasu kształtuje potencjał i kierunek pola elektrycznego. Genezę załamków elektrokardiogramu rozpatrywać więc należy w zestawieniu z anatomicznymi strukturami, które są elektrycznie aktywne w okresie zapisu określonego załamka.

Włókna mięśniowe serca obejmowane pobudzeniem elektrycznym są dwóch rodzajów:

1. włókna kurczliwe
2. włókna przewodzące

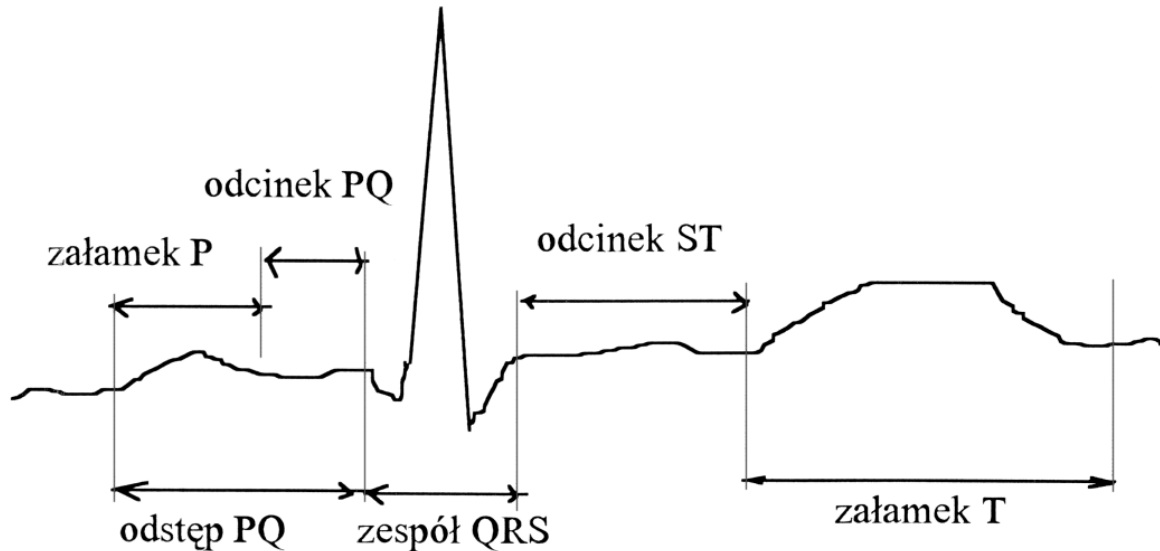
Włókna kurczliwe w grubych warstwach tworzą mięsień przedsionków, przegród i komór. Włókna przewodzące układają się w pasmach lub grupują się w postaci węzłów, tworząc układ przewodzący serca. Do układu przewodzącego należy: węzeł zatokowy (Keith-Flacka), węzeł przedsionkowo-komorowy A-V (Tawary), wspólny pęczek Hisa, lewa i prawa odnoga pęczka Hisa oraz włókna Purkiniego (rys.4.2.1). Pobudzenie elektryczne rozszerza się na komórki sąsiadujące z komórką pobudzoną, w ten sposób w powtarzającym się porządku przemieszcza się fala pobudzenia przez wszystkie komórki mięśnia sercowego.



Rys.4.2.1 Rozmieszczenie komórek rozrusznikowych w sercu

Pobudzenie serca rozpoczyna się we włóknach górnej części węzła zatokowego, który leży w prawym przedsionku w pobliżu ujścia żyły głównej górnej. Węzeł zatokowy posiada najwyższą częstotliwość swego automatyzmu wyzwalania pobudzenia, wskutek czego rytm tu wyzwalany jest rytmem prowadzącym. Leżące niżej ośrodki zdolne do wyzwalania rytmu, jak węzeł Tawary i pęczek Hisa, cechuje wolniejszy automatyzm. W skutek tego, w przypadkach fizjologicznie prawidłowych powyższe włókna mięśniowe ulegają szybszemu pobudzeniu płynącemu z węzła zatokowego, nim zdążą wyładować się własnym pobudzeniem.

Z węzła zatokowego fala pobudzenia przechodzi do otaczającego mięśnia prawego przedsionka i szerzy się przez jego przednią ścianę ku lewemu przedsionkowi. W dalszym ciągu obejmuje uszko, a później tylną ścianę lewego przedsionka. W ten sposób pierwszą strukturą anatomiczną pobudzoną jest prawy przedsionek. Końcowy okres pobudzenia prawego przedsionka nakłada się już rozpoczęty proces aktywacji lewego przedsionka. W pobudzeniu przedsionków można więc wyróżnić dwa okresy depolaryzacji prawego i lewego przedsionka. Efektem tego procesu jest załamek P obserwowany na krzywej elektrokardiografu (rys.4.2.2).



Rys.4.2.2 Krzywa EKG

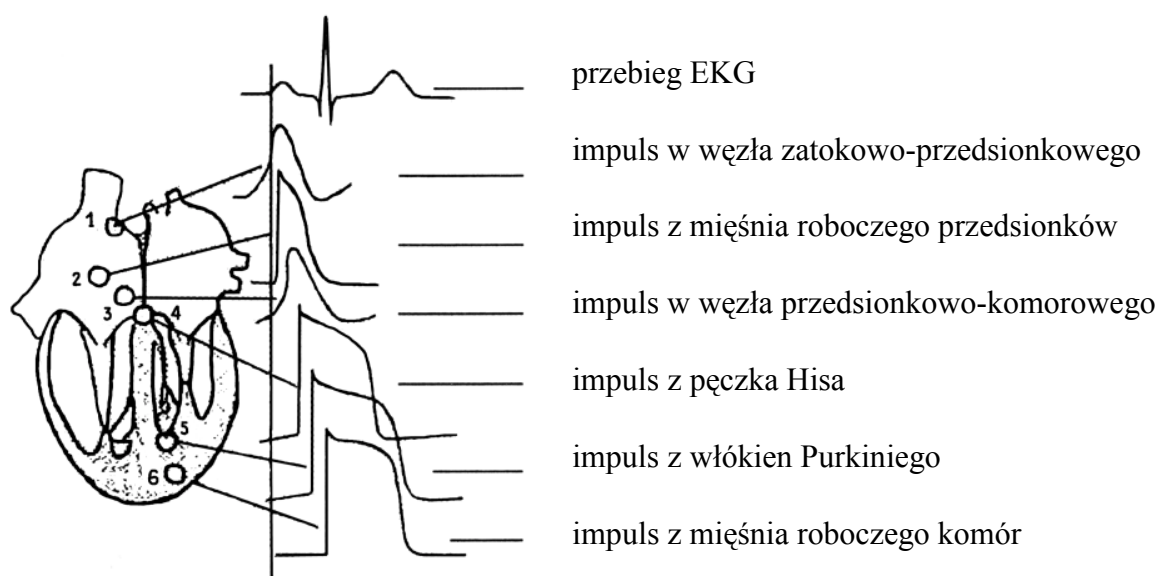
Po depolaryzacji obu przedsionków krzywa EKG przebiega poziomo w linii izoelektrycznej, tworząc płaski odcinek P-Q. W okresie tym fala pobudzenia szerzy się w dół, obejmuje węzeł przedsionkowo-komorowy, wspólny pęczek Hisa oraz lewą i prawą odnogę pęczka Hisa, układ włókien Purkiniego po lewej stronie przegrody, później po prawej stronie przegrody międzykomorowej. Pobudzenie elektryczne układu przewodzącego nie uwidacznia się na krzywej elektrokardiograficznej z powodu zbyt niskiego potencjału elektrycznego towarzyszącemu temu zjawisku. Początek pobudzenia węzła A-V występuje ze znacznym opóźnieniem i dopiero po upływie około jednej dziesiątej części sekundy po załamku P wykryto aktywność elektryczną we włóknach mięśniowych węzła Tawary. Ta właściwość opóźnienia fali pobudzenia na jej drodze do komór stanowi ważny mechanizm ochronny. Broni on komory przed zbyt wielką częstotliwością ich pobudzania w przypadku zwielokrotnienia bodźców napływających od strony przedsionków. Ze względu na wcześniejsze rozkrzewienie włókien Purkiniego od strony lewej odnogi pęczka Hisa fala pobudzenia najwcześniej spotyka się z kurczliwym mięśniem serca po lewej stronie przegrody międzykomorowej w środkowej części jej ściany. Pobudzenie masy mięśniowej przegrody międzykomorowej jest więc najwcześniej ujawnionym objawem akcji mięśnia komór, którego wyrazem jest zespół QRS.

Po wystąpieniu załamka Q fala pobudzenia elektrycznego, która osiągnęła już włókna Purkiniego w podwierzdziowej warstwie lewej i prawej komory, szerzy się przez całą grubość mięśnia komór od strony jam serca ku powierzchni zewnętrznej. Fala pobudzenia elektrycznego szybciej wygasa w prawej komorze z powodu krótkiej drogi przez jej cienki mięsień. Fala pobudzenia przepływająca przez znacznie grubszą masę mięśniową lewej komory wzbudza znacznie silniejszy, bo około 10-krotnie wyższy potencjał, wskutek czego siły elektromotoryczne w prawej komorze można pominąć. Wektor wypadkowy tego etapu depolaryzacji ma niemal identyczny przebieg jak wektor depolaryzacji ściany samej lewej komory. Wektor ten zapisuje się jako wysoki załamek R. Kierunek tego najsilniejszego wektora depolaryzacji mięśnia komór przesądza o kierunku średniej osi elektrycznej.

Repolaryzacja przedsionków odbywa się torem wytyczonym przez proces depolaryzacji. Zaczyna się w trakcie załamka P i trwa do początku odcinka S-T. Kierunek wychylenia jest w tym przypadku przeciwny do wychylenia załamka P. Ponieważ jednak siły repolaryzacji przedsionków są słabe, wychylenie to jest najczęściej niewidoczne na elektrokardiogramie, niekiedy powodując obniżenie odcinka P-R.

Odcinek S-T, następujący po zespole QRS, odpowiada okresowi powolnej repolaryzacji mięśnia komór. Napięcia wywołane w tym okresie są za małe dla odchylenia odcinka ST od linii izoelektrycznej. Okresowi szybkiej repolaryzacji mięśnia sercowego odpowiada natomiast odcinek T. Repolaryzacja mięśnia sercowego nie jest procesem analogicznym do repolaryzacji izolowanego włókna mięśniowego, ponieważ nie przebiega zgodnie z kierunkiem procesu depolaryzacji. Szereg różnic w szybkości i nasileniu procesów metabolicznych poszczególnych warstw serca sprawia, że w większości odprowadzeń załamek T jest dodatni.

Cały sygnał EKG jest wynikiem zsumowania wszystkich potencjałów wygenerowanych w cyklu pracy serca. Na rysunku 4.2.3 pokazano w sposób ideowy nałożenie się najważniejszych impulsów z komórek bodźcotwórczych oraz mięśni roboczych tworzących razem przebieg EKG rejestrowany przez elektrokardiograf.



Rys.4.2.3 Powstawanie krzywej EKG

4.3. Rejestracja pracy serca

Kształt sygnału EKG odczytywanego przez elektrokardiograf zależy przede wszystkim od układu pomiarowego, czyli sposobu podłączenia elektrod do pacjenta. Rutynowe badanie elektrokardiograficzne obejmuje 12 odprowadzeń, na które składają się trzy różne sposoby pomiaru pracy serca. Są to: trzy odprowadzenia kończynowe dwubiegunowe, trzy odprowadzenia kończynowe jednobiegunowe i sześć odprowadzeń jednobiegunowych przedsercowych.

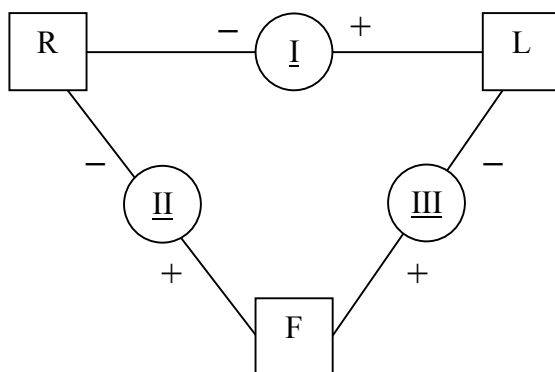
Odprowadzenia kończynowe dwubiegunowe nazywane są także odprowadzeniami standardowymi lub klasycznymi. Tworzą one układ trójkąta równobocznego postulowanego przez Einthovena. Odprowadzenia te oznaczają się cyframi rzymskimi I,II,III. Schemat odprowadzeń kończynowych dwubiegunowych przedstawia rys.4.3.1.

Odprowadzenie 'I' tworzą elektrody umieszczone na lewym przedramieniu i prawym przedramieniu.

Odprowadzenie 'II' łączy lewe podudzie z prawym przedramieniem.

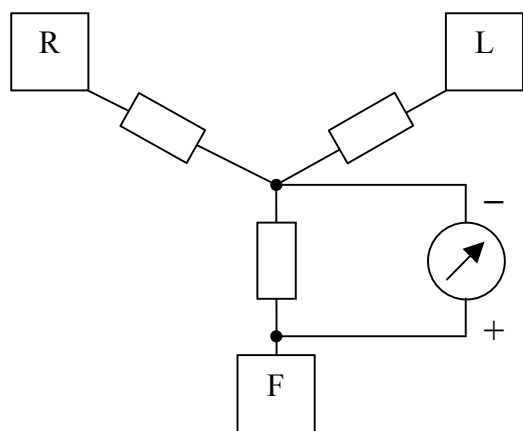
Odprowadzenie 'III' łączy lewe podudzie z lewym przedramieniem.

Odprowadzenia te rejestrują różnicę potencjałów między okolicami ciała mniej więcej jednakowo oddalonymi od serca. Są one bardzo dogodne z praktycznego punktu widzenia, ponieważ w większości ludzi zdrowych wychylenia załamków w zapisach z tych odprowadzeń skierowane są do góry. Cechuje je duża prostota, jednak założenia pomiarowe dalece odbiegają od rzeczywistych warunków.

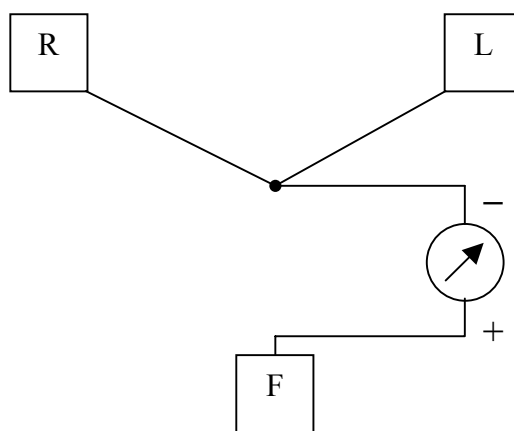


Rys.4.3.1 Odprowadzenia Einthovena (R - prawa ręka, L – lewa ręka, F – lewa noga)

Odprowadzenia jednobiegunowe uzyskuje się w ten sposób, że elektrodę badającą (czynną) umieszcza się w określonym punkcie powierzchni ciała, natomiast drugi biegun odprowadzenia podłącza się do specjalnie skonstruowanego wyjścia o potencjale zerowym (central terminal). W wersji Wilsona stanowi ono połączenie trzech elektrod kończynowych z tym, że do każdego przewodnika dołącza się opór równy 5000Ω . Uzyskane w ten sposób odprowadzenia oznacza się literą V. Umieszczając elektrodę badającą na prawym przedramieniu, lewym przedramieniu, bądź lewym podudziu uzyskuje się odprowadzenia kończynowe: VR, VL, VF (rys.4.3.2). Zapisy te cechują się bardzo niską amplitudą załamków. Stąd Goldberger wprowadził modyfikację tego systemu do postaci stanowiącej połączenie dwóch elektrod kończynowych bez włączania dodatkowych oporów (rys.4.3.3). Odprowadzenia jednobiegunowe rejestrowane przy użyciu elektrody Goldbergera określa się symbolami aVR, aVL i aVF ('a' od słowa 'augmented' - zwiększony).



Rys.4.3.2 Odprowadzenie kończynowe jednobiegunowe (VF) według Wilsona



Rys.4.3.3 Odprowadzenie kończynowe jednobiegunowe (aVF) w modyfikacji Goldbergera

Podczas, gdy odprowadzenia kończynowe reprezentują niejako rzut siły elektromotorycznej serca na płaszczyznę czołową, odprowadzenia przedsercowe, czyli piersiowe, odzwierciedlają zjawiska elektryczne w płaszczyźnie poziomej. Najczęściej w pomiarach stosuje się sześć standartowych punktów przyłożenia elektrod na klatce piersiowej:

Punkt 1 – w IV międzyżebżu przy prawym brzegu mostka.

Punkt 2 – w IV międzyżebżu przy lewym brzegu mostka.

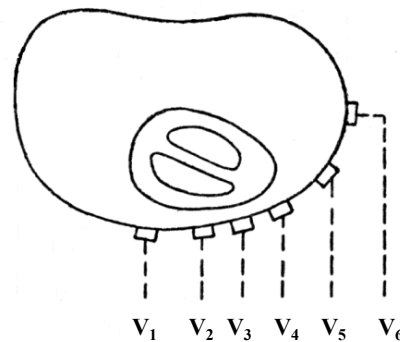
Punkt 3 – w połowie odległości między punktem 2 i 4.

Punkt 4 – w V międzyżebżu w linii środkowej obojczykowej.

Punkt 5 – w przedniej lewej linii pachowej, w miejscu przecięcia jej przez prostopadłą linię poprowadzoną od punktu 4.

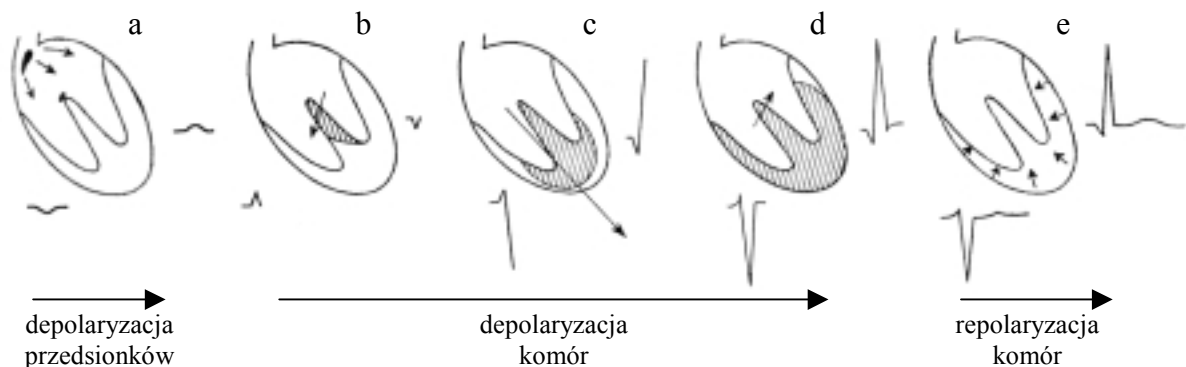
Punkt 6 – w środkowej lewej linii pachowej, w miejscu przecięcia jej przez prostopadłą linię poprowadzoną od punktu 4.

Odprowadzenia przedsercowe oznacza się symbolami od V_1 do V_6 .



Rys.4.3.4 Punkty przyłożenia elektrod w odprowadzeniach przedsercowych – rzut na płaszczyznę poziomą (widok z góry pacjenta)

Na rysunku 4.3.5 zobrazowany został schemat powstawania krzywej EKG w zależności od miejsca przyłożenia elektrod elektrokardiografu. Zasadę tę najłatwiej zaobserwować w odprowadzeniach przedsercowych, rejestrujących rzut sił elektromotorycznych na płaszczyznę poziomą. Rysunek pokazuje przebiegi zarejestrowane przez odprowadzenia z nad prawej (V_1) i lewej komory (V_6).



Rys.4.3.5 Kolejne fazy powstawania załamków w sygnale EKG w odprowadzeniach z nad prawej i lewej komory.

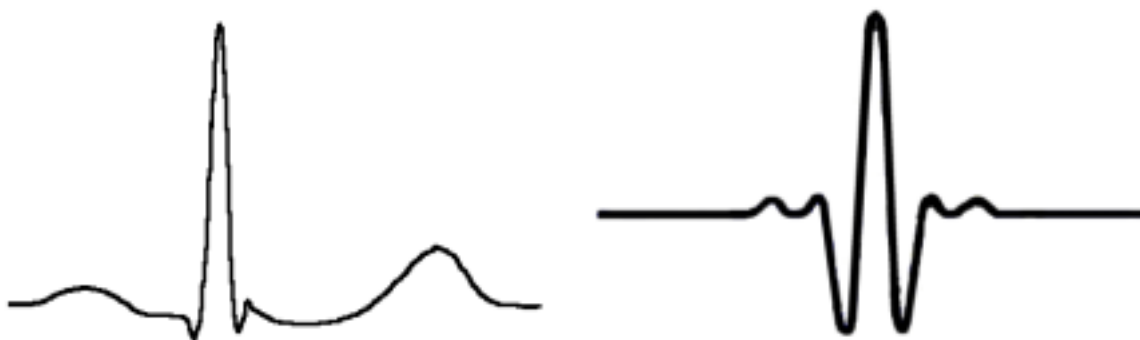
a – depolaryzacja przedsionków

b,c,d – depolaryzacja komór w czasie odpowiednio 0,01 , 0,04 i 0,07 sekundy od rozpoczęcia depolaryzacji

e – repolaryzacja przedsionków

4.4. Reprezentacja falkowa sygnału EKG

Poszukiwanie optymalnej reprezentacji sygnału EKG polega na znalezieniu takiego odwzorowania, które zapewniłoby jak największą minimalizację danych potrzebnych do zapisania informacji o tym sygnale. Generalnie o jakości reprezentacji decyduje podobieństwo właściwości między sygnałem analizowanym (EKG) i analizującym (sygnały bazowe). Transformacja falkowa wydaje się być dobrym narzędziem do zapisu elektrokardiogramu z tego względu, że funkcja bazowa, czyli 'falka matka' posiada bardzo dobrą lokalizację w czasie, zachowując dobrą selektywność (lokalizację w dziedzinie częstotliwości). Sygnał EKG ma charakter niestacjonarny. Na stały przebieg o amplitudzie izolinii w regularnych odstępach czasu nałożone są załamki P, QRS i T. Ta cecha decyduje o dużym znaczeniu lokalizacji czasowej sygnału analizującego. Na rysunku 5.4.1. są pokazane obydwie sygnały: EKG i przykładowej falki. Widać, że ich charakter jest bardzo podobny, z czego można wnioskować dużą efektywność opisu przebiegu EKG za pomocą falek.

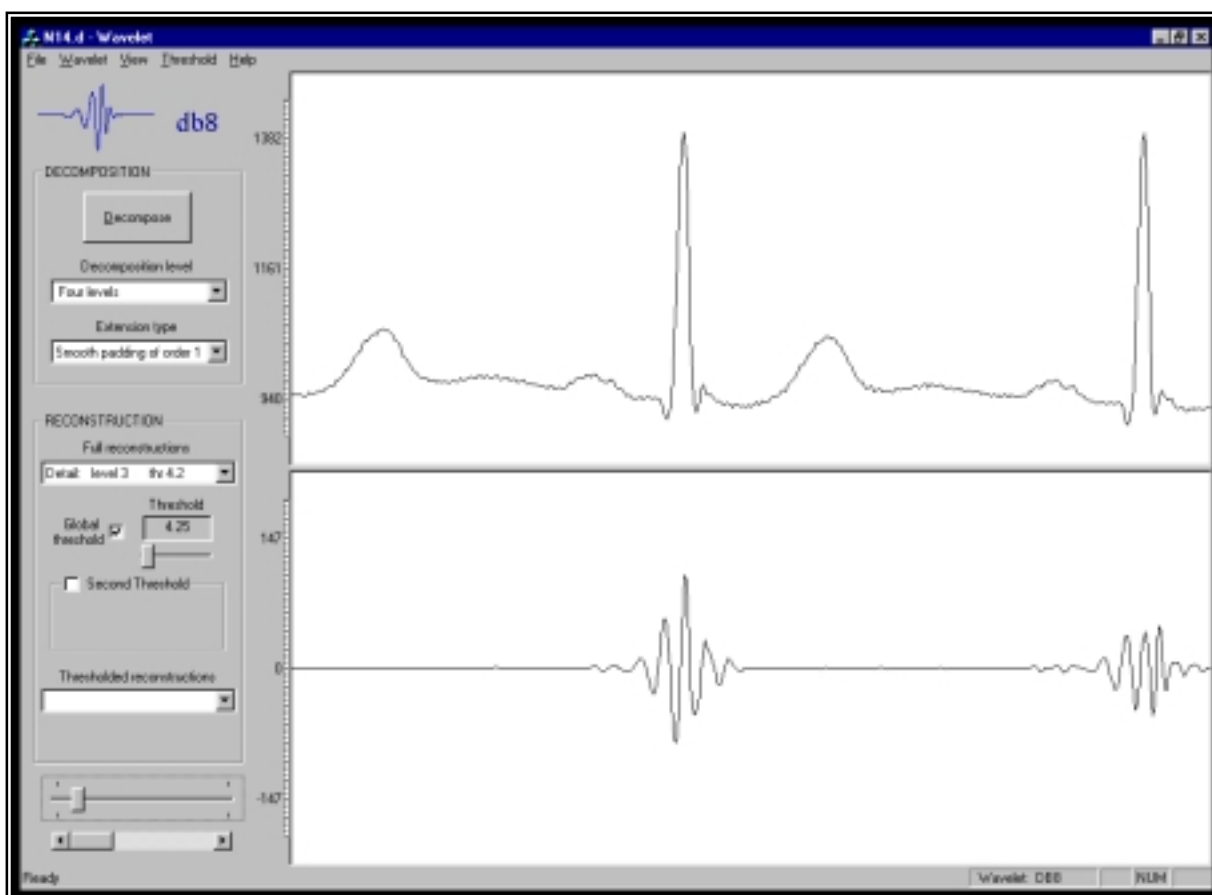


Rys.5.4.1 Z lewej strony kształt typowego przebiegu EKG, z prawej falka 'bior5.5'

5. Opis programu

5.1. Budowa aplikacji

Program składa się z czterech podstawowych modułów: pulpitu kontrolnego, dwóch okien i menu (rys.5.1.1).

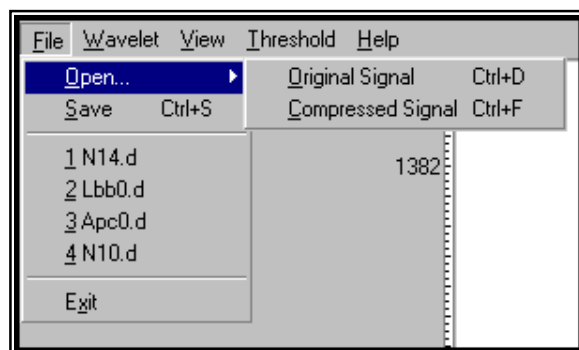


Rys.5.1.1 Ogólny widok programu

Menu umieszczone jest w lewym górnym rogu pod listwą programu. Pozwala na załadowanie danych do pamięci programu oraz wybór niektórych opcji pracy. Dostępnych jest pięć podmenu:

File – zapewnia operacje plikowe i zawiera odpowiednio (rys.5.1.2):

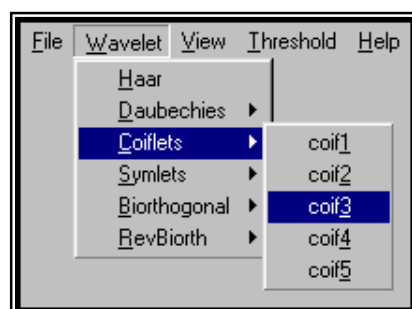
- **Open/Original Signal** – powoduje załadowanie spróbkowanego przebiegu EKG zapisanych w postaci ciągu kolejnych próbek z plików o rozszerzeniu *.d oraz *.txt.
- **Open/Compressed Signal** – powoduje załadowanie skompresowanego sygnału EKG w postaci ciągu współczynników falkowych z plików o rozszerzeniu *.flk.
- **Save** – zapisuje skompresowany sygnał EKG do pliku *.flk.
- **1...4** – ładuje sygnały EKG z ostatnio otwieranych plików
- **Exit** – wyjście z programu



Rys.5.1.2 Widok menu 'File'

Wavelet – udostępnia wybór falki wykorzystanej do dekompozycji falkowej (rys.5.1.3)

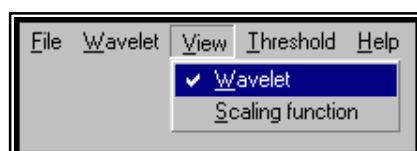
- **Haar** – falka Haar'a
- **Daubechies** – rodzina falek Daubechies oznaczanych symbolem 'DBx'
- **Coiflets** – rodzina falek Coiflets oznaczanych symbolem 'COIFx'
- **Symlets** – rodzina falek Symlets oznaczanych symbolem 'SYMx'
- **Biorthogonal** - rodzina falek Biorthogonal oznaczanych symbolem 'BIORx.x'
- **Revbiorth** - rodzina falek Reverse Biorthogonal oznaczanych symbolem 'RBIORx.x'



Rys.5.1.3 Widok menu 'Wavelet'

View – zawiera opcje wyboru wyświetlania kształtu falki lub funkcji skalującej w górnej części pulpitu sterującego, zaraz pod menu (rys.5.1.4).

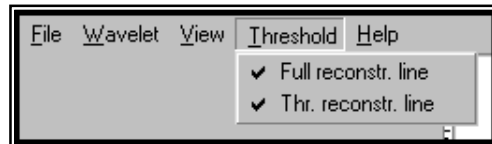
- **Wavelet** – zaznaczenie powoduje wyświetlanie falki
- **Scaling function** – zaznaczenie powoduje wyświetlanie funkcji skalującej



Rys.5.1.4 Widok menu 'View'

Threshold – udostępnia opcje wyświetlania linii progu obcinania współczynników falkowych (rys.5.1.5).

- **Full reconstr. line** – uaktywnia rysowanie linii progu podczas wyświetlania szczegółów w pełni zrekonstruowanego sygnału
- **Thr. reconstr. line** – uaktywnia rysowanie linii progu w przypadku rekonstrukcji progowej (tzn. gdy rekonstrukcja przeprowadzana była wyłącznie z współczynników o większej wartości niż wartość progu)



Rys.5.1.5 Widok menu 'Threshold'

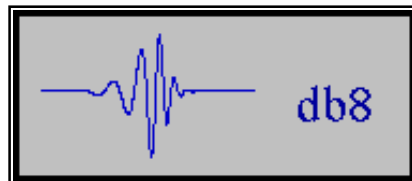
Help – podmenu pomocy

- **About program** – zawiera informacje o programie

Dwa centralne okna służą do wyświetlania przebiegów EKG. Okno górne wyświetla oryginalne przebiegi, natomiast dolne wyświetla przebiegi zrekonstruowane ze współczynników falkowych uzyskanych w wyniku dekompozycji falkowej.

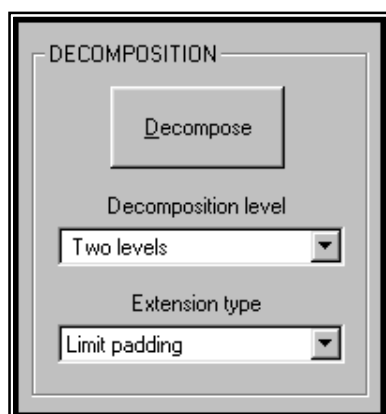
Pulpit kontrolny umieszczony jest z lewej strony. Zawiera większość narzędzi potrzebnych do sterowania programem. Podzielony jest on na cztery odrębne obszary.

1. Na górze wyświetlana jest postać bieżącej falki wybranej z menu oraz jej oznaczenie (rys.5.1.6).

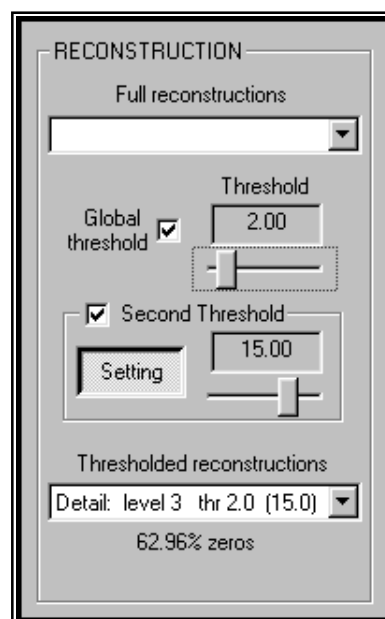


Rys.5.1.6 Falka 'Daubeches 8'

2. Poniżej znajduje się obszar „DECOMPOSITION” służący do sterowania dekompozycją sygnałów. Zawiera dwie rozwijane listwy wyboru odpowiednio poziomu dekompozycji („Decomposition level”) oraz rodzaju rozszerzenia sygnału („Extension type” - decydującego o stopniu eliminacji błędu brzegowego), a także przycisk uruchomienia procesu dekompozycji („Decompose”).



Rys.5.1.6 Panel 'Decomposition'



Rys.5.1.7 Panel 'Reconstruction'

3. Obszar „RECONSTRUCTION” udostępnia narzędzia sterowania rekonstrukcją sygnału. Dostępne są dwie rozwijane listwy: „Full reconstructions” oraz „Thresholded reconstructions” włączające rekonstrukcję sygnału i automatyczne wyświetlenie otrzymanego przebiegu. Pierwsza pozycja dokonuje pełnej rekonstrukcji wykorzystując wszystkie współczynniki falkowe, druga natomiast tzw. progowej rekonstrukcji zerując uprzednio współczynniki mniejsze od wartości progowej. W obrębie tego obszaru znajdują się także narzędzia do konfiguracji wartości prog. „Global threshold” jest polem wyboru, którego zaznaczenie powoduje nastawianie takiego samego progu dla wszystkich poziomów dekompozycji. Obok z prawej strony jest suwak („Threshold”) do regulacji poziomu progę oraz okienko wyświetlające nastawioną wartość. Aktualny poziom progę pokazują także dwie niebieskie linie na przebiegu zrekonstruowanego przebiegu (o ile zaznaczona jest pozycja w menu THRESHOLD). Opcja „Second threshold” umożliwia włączenie drugiego progę rekonstrukcji. Po jego wybraniu stają się dostępne przycisk „Settings” oraz analogiczny suwak z okienkiem do ustawienia wartości progę drugiego. Wciśnięcie przycisku „Settings” powoduje wyszarzenie górnego okna wyświetlania, w którym następnie za pomocą myszy można ustawić czasowy zakres ważności progę drugiego. Zwolnienie tego przycisku przywróci poprzedni wygląd okna, przy czym ustawienia zakresu drugiego progę zostaną zapamiętane.
4. Na dole pulpitu kontrolnego znajdują się dwa suwaki. Górny służy do skalowania (rozszerzania) przebiegów, natomiast dolny do zmiany pozycji (przesuwania).

Pasek statusu znajduje się na dole widoku programu. W przypadku najechania kursorem myszy na któryś z obiektów pulpitu lub pozycję w menu, z lewej strony paska statusu znajdzie się podpowiedź o przeznaczeniu danego elementu.

5.2. Instrukcja użytkownika

Program działa poprawnie pod systemem operacyjnym Windows 95/98. W skład aplikacji wchodzi pliki 'Wavelet.exe', 'rar.exe' oraz folder 'baza' z plikami zawierającymi przebiegi EKG. Program uruchamia się za pomocą 'Wavelet.exe'. Aplikacja pozwala przeprowadzić kompresję falkową sygnałów EKG. Umożliwia praktyczne poznanie procesów dekompozycji i rekonstrukcji. Dostępność regulacji wielu parametrów daje pole do wielu eksperymentów i lepszego zrozumienia zagadnienia.

Aplikacja po uruchomieniu ma postać opisaną w punkcie 5.1. Pracę z programem rozpoczyna się od załadowania oryginalnego **przebiegu EKG** (tzn. takiego, którego kolejne wartości reprezentują wartości sygnału spróbkowanego w czasie). Zestaw oryginalnych przebiegów znajduje się w plikach tekstowych z rozszerzeniem '*.d' umieszczonych w folderze 'baza'. Przebieg należy wczytać poprzez wybranie z menu 'File/Open.../Original Signal' lub wciśnięcie klawisza skrótu 'Ctrl+D'. Pojawi się okno dialogowe z listą dostępnych plików. Wybranie jednego z nich spowoduje zamknięcie okna i załadowanie przebiegu, który następnie zostanie narysowany w górnym oknie wyświetlania programu. W menu 'File' znajdują się także zazwyczaj cztery pola typu 'Recent files' zawierające nazwy ostatnio używanych plików. Przez wybranie odpowiedniej pozycji można również załadować żądany sygnał.

Pierwszym krokiem po wczytaniu przebiegu jest przeprowadzenie jego dekompozycji. Przed uruchomieniem samego procesu dekompozycji można ustawić trzy istotne parametry: rodzaj falki dostępny w menu 'Wavelet', poziom dekompozycji (listwa rozwijana 'Decomposition level') oraz sposób rozszerzenia sygnału (listwa 'Extension type'). **Wybór falki** decyduje oczywiście o jakości dekompozycji. Generalnie, im bardziej postać falki jest zbliżona do kształtu obrabianego sygnału tym kompresja jest lepsza. Dokładniejsza analiza właściwości ważniejszych falek dokonana jest w rozdziale 3. **Poziom dekompozycji** jest krotnością jednego etapu procesu dekompozycji. Oznacza to, że po uruchomieniu n-poziomowej dekompozycji sygnału proces ten dokona n-krotnej analizy sygnału na różnych stopniach szczegółowości (najpierw wyodrębni wysokoczęstotliwościowe cechy sygnału, potem rekurencyjnie będzie analizował właściwości sygnału w coraz niższych przedziałach częstotliwości). **Sposób rozszerzenia sygnału** związany jest ze zniekształceniami brzegowymi dekompozycji falkowej. Operacja splotu, na której bazuje proces cyfrowej dekompozycji falkowej DWT wymaga uzupełnienia sygnału dodatkowymi próbkami na obu końcach. Wybór parametru 'Extension type' określa sposób doboru wartości tych próbek. Dokładniejszy opis algorytmów wypełnienia sygnału zamieszczony jest w dalszej części pracy. Po ustaleniu parametrów dekompozycji należy wybrać przycisk 'Decompose' lub klawisz skrótu 'Alt+D', aby uruchomić proces dekompozycji. Operacja ta będzie trwała do kilku sekund w zależności od mocy obliczeniowej komputera i oczywiście od wybranych parametrów, a w szczególności od poziomu dekompozycji i długości falki (falki o większych indeksach są dłuższe i wymagają więcej obliczeń przy operacji splotu). Na czas operacji dekompozycji falkowej kursor myszy zamienia się w klepsydrę i gdy z powrotem przybierze pierwotny kształt oznacza to, że zdekomponowany sygnał w postaci współczynników falkowych jest w pamięci komputera.

Następnym krokiem jest rekonstrukcja sygnału.

Na początek warto skoncentrować się nad **pełną rekonstrukcją sygnału** (czyli procesem odwrotnym do dekompozycji) z zestawem danych wejściowych będących współczynnikami otrzymanymi w wyniku dekompozycji. W tym celu należy otworzyć listwę 'Full reconstructions' i wybrać pozycję 'Reconstructed signal'. Automatycznie zostanie wywołana rekonstrukcja sygnału i po chwili (porównywalnej z czasem dekompozycji) zrekonstruowany

przebieg zostanie odrysowany w dolnym oknie. Obydwa przebiegi (oryginalny i zrekonstruowany) można rozciągać i przesuwac w celu zaobserwowania wizualnych podobieństw i różnic między nimi świadczących o jakości transformacji falkowej. Po rekonstrukcją całego sygnału możliwe jest zrekonstruowanie samych współczynników detali z pierwszego, drugiego i kolejnych poziomów dekompozycji, a także współczynników aproksymacji pozostałych po odrzuceniu detali. W zależności od ilości ustawionych poziomów dekompozycji po otwarciu listwy 'Full reconstructions' pod pozycją 'Reconstructed signal' pojawią się 'Detail: level 1', 'Detail: level 2', ... 'Detail: level n' oraz na końcu 'Aproximation: level n'. Po wybraniu 'Detail: level 1' podobnie jak w poprzednim przypadku nastąpi automatyczna rekonstrukcja, lecz tym razem samych współczynników detali poziomu pierwszego i otrzymany sygnał zostanie narysowany w dolnym oknie. Na podstawie widocznego przebiegu można oszacować zawartość sygnału o najwyższej częstotliwości (oczywiście częstotliwości w rozumieniu falkowym – gdzie podstawą sygnału jest konkretna falka, a nie sinusoida) w sygnale oryginalnym. Tutaj pomocne mogą się okazać linijki umieszczone po lewej stronie okien wyświetlania pokazujące wartość amplitudy. Tą operację można powtórzyć dla kolejnych poziomów detali wybierając następne pozycje w 'Reconstructed signal'. Wybranie ostatniej pozycji 'Aproximation: level n' spowoduje zrekonstruowanie samych współczynników aproksymacji i narysowany zostanie przebieg, który faktycznie będzie wynikiem odjęcia od sygnału oryginalnego wszystkich zrekonstruowanych sygnałów detali. W tym momencie także warto porównać obydwie przebiegi.

Po rekonstrukcji któregoś z poziomów detali na przebiegu pojawią się dwie niebieskie linie. Są to linie dodatnia i ujemna pokazujące bezwzględną wartość progów. **Próg** jest wartością, do której w przypadku kompresji są porównywane wszystkie współczynniki danego poziomu. Jeżeli bezwzględna wartość współczynnika jest mniejsza od progów to jest on eliminowany (zerowany). Do regulacji progów służy suwak oznaczony 'Threshold' wraz z okienkiem wyświetlającym jego aktualną wartość. Aktualna wartość progów ma również odzwierciedlenie w przesuwających się niebieskich liniach na wykresie dających rozeznanie, które ze współczynników zostaną usunięte. Należy jednak mieć na względzie to, że wykres nie przedstawia ciągu współczynników, lecz przebieg uzyskany w wyniku ich rekonstrukcji. Charakter obydwu ciągów pozostaje jednak ten sam, dlatego celowe jest umieszczanie linii progowych na przebiegu. Wyświetlanie linii progowych można także w razie potrzeby wyłączyć. W tym celu należy odznaczyć w menu 'Threshold' pozycję 'Full reconstr. line'. Druga pozycja 'Thr. reconstr. line' ma to samo zastosowanie, ale przy wyświetlaniu tzw. rekonstrukcji progowej, która będzie omówiona w dalszej kolejności. Można zauważyć, że zaraz po rekonstrukcji detali wartość progów ma już wstępną wartość. Jest to wartość domyślna obliczana bezpośrednio po dekompozycji sygnału jako wartość optymalna będąca kompromisem między maksymalną kompresją sygnału, a minimalnymi zniekształceniami. Jest to wartość ustalona dla wszystkich poziomów detali jednocześnie. Użytkownik ma możliwość nastawienia progów dla każdego poziomu indywidualnie, jak również jednorazowego ustawienia globalnego progów dla wszystkich poziomów jednocześnie – w tym celu należy zaznaczyć opcję 'Global threshold'. Wartości progów detali dla każdego poziomu są także wypisane w menu listwy 'Full reconstructions' na odpowiadającym ich pozycjach. Jest to szczególnie wygodne, gdy progi są ustawiane niezależnie różnych poziomów, gdyż przy suwaku 'Threshold' wypisana wartość dotyczy tylko progów aktualnie wyświetlanego poziomu.

Po wybraniu żądanej wartości progów można obejrzeć efekt wycięcia „nieistotnych” współczynników poprzez rekonstrukcję za pomocą drugiej listwy '**Thresholded reconstructions**'. Działanie jej jest analogiczne do listwy 'Full reconstructions' z tą różnicą, że przed samym procesem rekonstrukcji wszystkie współczynniki są porównywane z

odpowiadającym im progom i w przypadku mniejszej amplitudy od progu zerowane. W tym miejscu wydaje się być celowe porównanie otrzymanych przebiegów z sygnałem oryginalnym oraz z przebiegami otrzymanymi z pełnej rekonstrukcji. Po rekonstrukcji z obcinaniem progowym pod listwą pojawia się zawartość procentowa współczynników wyzerowanych np. „82,50 zeros”. Dotyczy to oczywiście współczynników z danego poziomu. Tylko w przypadku pozycji ‘Reconstructed signal’ procent wyzerowanych współczynników dotyczy całego sygnału (wszystkich współczynników otrzymanych w procesie dekompozycji). Można zauważyć, że podczas rekonstrukcji współczynników aproksymacji (pozycja ‘Aproximation: level n’ w ‘Thresholded reconstructions’) procent wyzerowanych współczynników wynosi zero. Dzieje się tak dlatego, że współczynniki aproksymacji stanowią niejako szkielet całego sygnału i z założenia nie ingeruje się w ich wartości. Tak więc sygnał zrekonstruowany ze współczynników aproksymacji będzie zawsze taki sam niezależnie, czy zostanie stworzony za pomocą ‘Thresholded reconstructions’, czy ‘Full reconstructions’. Na uwagę zasługuje jeszcze przypadek rekonstrukcji całego sygnału ‘Reconstructed signal’ przy pomocy ‘Thresholded reconstructions’. Pod procentową zawartością obciętych współczynników znajduje się zapis typu „PRD=0,425%”. Chodzi tu o różnicę średniokwadratową pomiędzy sygnałem oryginalnym, a otrzymanym w wyniku kompresji falkowej (dekompozycji, zerowania współczynników i rekonstrukcji). **Wartość PRD** jest miarą podobieństwa pomiędzy tymi sygnałami, im jest ona mniejsza tym sygnały są do siebie bardziej podobne. PRD wylicza się ze wzoru:

$$PRD = \sqrt{\frac{\sum_{i=1}^n [X_{org}(i) - X_{rec}(i)]^2}{\sum_{i=1}^n [X_{org}(i)]^2}} * 100\%$$

PRD – różnica średniokwadratowa między sygnałem oryginalnym, a zrekonstruowanym

$X_{org}(i)$ – amplituda i -tej próbki oryginalnego sygnału

$X_{rec}(i)$ – amplituda i -tej próbki zrekonstruowanego sygnału

Obydwie wartości: wyzerowane współczynniki oraz PRD są wyznacznikiem efektywności algorytmu falkowego. Procent wyzerowanych współczynników mówi o stopniu kompresji sygnału, czyli eliminacji zbędnych informacji, natomiast PRD o stopniu zachowania informacji zawartych w sygnale. Idealna kompresja polega na takim dobraniu progów, aby zachować odpowiedni kompromis między tymi wartościami.

Aplikacja ta ma także możliwość zastosowania **drugiego progu** obcinania współczynników. Koncepcję tą nasunął sam charakter sygnałów EKG, z którego można wyznaczyć odcinki istotniejsze z punktu widzenia diagnozy lekarskiej, jakimi są załamki P, QRS i T oraz te mniej ważne, jak odcinek pomiędzy załamkiem T i P. Dlatego też kompresja będzie skuteczniejsza, jeżeli w mniej istotnych miejscach przebiegu będzie można zastosować oddzielne większe progowanie. Opcję drugiego progu włącza się poprzez zaznaczenie bloku ‘Second threshold’, w wyniku czego w polu tym pojawi się przycisk ‘Settings’ oraz analogiczny do poprzedniego progu suwak do ustawiania wartości progu wraz z okienkiem do wypisania ustawionej wartości. Przycisk ‘Settings’ służy do ustalenia zakresów, w których będzie obowiązywał drugi próg. Ustalenie tych zakresów najwygodniejsze jest, gdy oryginalny sygnał jest wczytany, zdekomponowany i zrekonstruowany jest któryś z

poziomów detali. Wtedy w górnym oknie znajduje się sygnał oryginalny, a w dolnym zrekonstruowane współczynniki detali. Należy zaznaczyć opcję 'Second threshold', a następnie włączyć 'Settings', co spowoduje wyszarzenie górnego okna. Warto także zmienić wartość drugiego progu tak, aby różnił się od progu pierwszego np. dwukrotnie. Spowoduje to, że zmiany zakresu progów będą widoczne w dolnym oknie. Zakresy drugiego progu wybiera się myszą w górnym oknie. Robi się to przyciskając lewy przycisk myszy na początku zakresu, przesuwając kursor myszy z wciśniętym przyciskiem i puszczając przycisk na końcu zakresu. W zaznaczonym polu, które jest teraz zakresem progu drugiego tło staje się ciemniejsze, natomiast niebieska linia w dolnym oknie zmienia amplitudę na ustawioną dla tego progu. W podobny sposób można wyznaczyć następne zakresy. Położenie zakresów można korygować poprzez „złapanie” lewym przyciskiem myszy linii brzegowej zakresu i przesunięcie jej w żądane miejsce. Można także usunąć niepotrzebny zakres wciskając prawy przycisk myszy w obrębie tego zakresu i wybranie z wysuniętego menu kontekstowego pozycji 'Delete Bracket'. Wybranie 'Delete All' spowoduje usunięcie wszystkich przedziałów drugiego progu. Podczas ustalania zakresów w razie potrzeby można odświeżyć zawartość obu okien przez naciśnięcie prawego przycisku myszy w dolnym oknie i wybranie 'Refresh'. Spowoduje to ponowne odrysowanie przebiegów, linii progowych i zakresów według ustawionych wartości. Dla precyzji warto rozciągnąć przebiegi w oknach za pomocą odpowiednich suwaków i w powiększeniu dokładnie ustalić granice pomiędzy progami. Po ustaleniu zakresów dla drugiego progu zaleca się zwolnić przycisk 'Settings', aby nie zaciemniać oryginalnego przebiegu. Górne okno powróci do pierwotnego wyglądu, przy czym ustawienia drugiego progu pozostaną zapamiętane. W razie potrzeby można z powrotem wybrać 'Settings' i powrócić do ustawień w celu skorygowania zakresów.

Po wybraniu satysfakcjonującego progu jest możliwość **zapisania** skompresowanego sygnału w postaci współczynników falkowych na dysku. W tym celu należy z menu 'File' wybrać pozycję 'Save' lub przycisnąć kombinację szybkiego dostępu 'Ctrl+S'. Otworzy się okno dialogowe z ustawioną ścieżką dostępu do katalogu 'baza'. W oknie trzeba wprowadzić nazwę pliku i potwierdzić przyciskiem 'Zapisz' lub 'Save'. Powstanie plik z rozszerzeniem '*.flk' zawierający rodzaj falki, ilość poziomów dekompozycji, tablicę zawierającą liczbę współczynników każdego poziomu oraz tablice samych współczynników falkowych. Dane te są dodatkowo pakowane programem archiwizującym 'rar.exe'. Zrobione jest to po to, aby stopień kompresji sygnału miał swoje odbicie w objętości pliku. Sama kompresja falkowa nie usuwa nieistotnych współczynników, tylko je zeruje, zaś z punktu widzenia przechowywania informacji zarówno liczba '1263,234' jak i '0' zajmują dokładnie po cztery bajty (dotyczy to platform 32 bitowych jakimi są 'Windows 95OSR2/98'). Zatem dodatkowe upakowanie danych umożliwia fizyczne odzwierciedlenie skutków wynikających z wyboru zastosowanego poziomu progu.

Istnieje również możliwość odczytania skompresowanego sygnału z pliku dykowego. Z menu 'File' należy wybrać 'Open.../Compressed Signal' lub wcisnąć 'Ctrl +F' i następnie z otwartego okna zaznaczyć żądany plik. Po wczytaniu pliku nastąpi jego automatyczna rekonstrukcja analogiczna do wybrania z listwy 'Thresholded reconstructions' pozycji 'Reconstructed signal'. Zrekonstruowany sygnał zostanie narysowany w dolnym oknie, natomiast pod listwą 'Thresholded reconstructions' zostanie wypisany procent zerowych współczynników przebiegu.

6. Doświadczenia

Poniższy rozdział ma za zadanie przedstawić praktyczną realizację kompresji falkowej. Doświadczenia są przeprowadzone na bazie sygnałów EKG zmierzonych za pomocą odprowadzeń 'MLII' z częstotliwością próbkowania 360Hz oraz z kwantem amplitudy napięcia $LSB=5\mu V$. Zamieszczone poniżej przykłady są poparciem teoretycznych rozważań. Mają na celu pokazanie ogólnych właściwości metod transformacyjnych kompresji, jak również szczególnych cech charakteryzujących transformatę falkową.

6.1. Doświadczenia wprowadzające, dobór progu obcinania

Doświadczenie prezentuje podstawowe czynności obsługi programu, takie jak: wczytanie oryginalnego przebiegu, dobranie parametrów dekompozycji i rekonstrukcji falkowej oraz uruchomienie tych procesów. Koncentruje się przede wszystkim na doborze progu rekonstrukcji i jego wpływie na efektywność kompresji. Jakość kompresji falkowej mierzona jest za pomocą dwóch wskaźników. Pierwszy z nich pokazuje liczbę usuniętych (wyzerowanych) współczynników, czyli reprezentuje stopień kompresji. Drugi natomiast jest wskaźnikiem PRD - różnicy średniokwadratowej pomiędzy sygnałem oryginalnym, a skompresowanym i reprezentuje stopień podobieństwa pomiędzy tymi sygnałami. Bardzo ważna jest także subiektywna ocena jakości kompresji, dlatego zamieszczono rysunki z otrzymanymi przebiegami dla wizualnego porównania efektów zmian progu rekonstrukcji. Jednym z głównych powodów stosowania kompresji sygnałów w praktyce jest potrzeba minimalizacji zasobów wykorzystywanych do archiwizacji tych danych, czyli popularnie mówiąc "ilości miejsca zajmowanego na dysku". W programie dostępna jest opcja 'Save' umożliwiająca zapisanie skompresowanego sygnału w postaci współczynników falkowych do pliku o rozszerzeniu '*.flk'. Porównanie pojemności tych plików jest praktycznym wyznacznikiem stopnia kompresji sygnału.

Do ćwiczenia wybrano sygnał 'N0.d'. Pozostawiono domyślne parametry dekompozycji tzn. falkę 'db2', drugi poziom dekompozycji oraz rodzaj rozszerzenia sygnału ustawiony na wartość 'Limit padding'. Po wciśnięciu 'Decompose' sygnał został przetransformowany do postaci współczynników falkowych reprezentujących dziedzinę „czas-skala”. Aby obejrzeć efekt przetwarzania falkowego dokonano kolejno rekonstrukcji współczynników każdego poziomu oddzielnie oraz wszystkich razem transformując cały sygnał z powrotem do dziedziny czasu. Uczyniono to poprzez wybranie z listwy 'Full reconstructions' dostępnych pozycji:

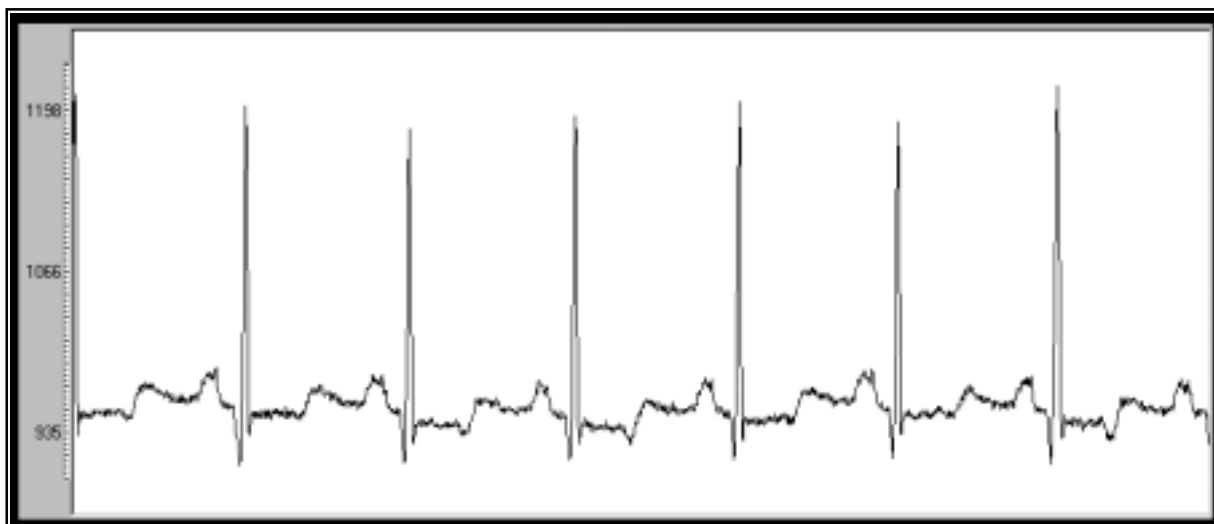
'Reconstructed signal' – rekonstrukcja całego sygnału

'Detail: level 1' – rekonstrukcja szczegółów poziomu pierwszego

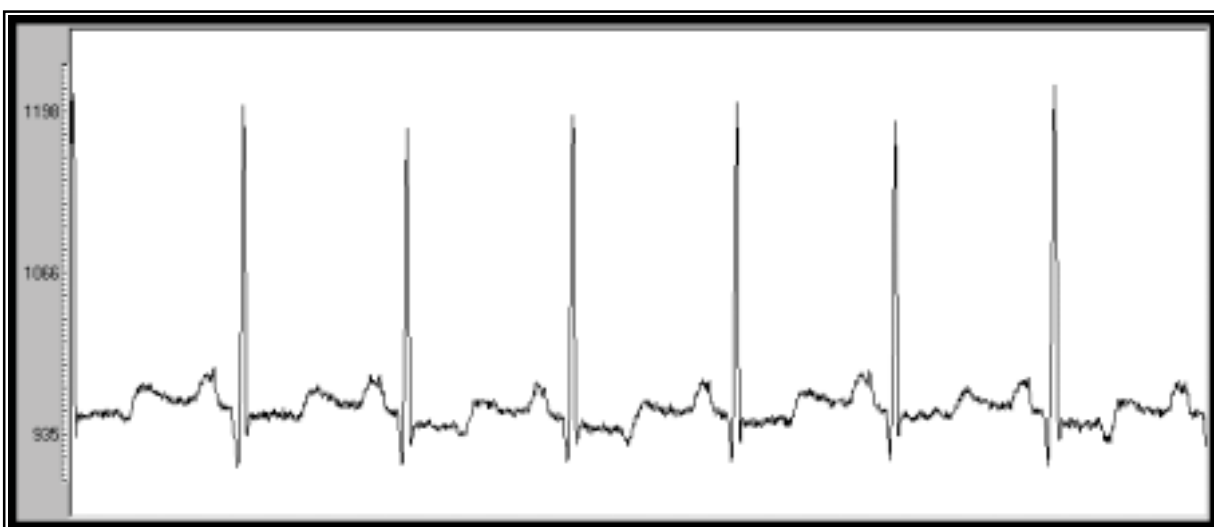
'Detail: level 2' – rekonstrukcja szczegółów poziomu drugiego

'Aproximation: level 2' – rekonstrukcja aproksymacji poziomu drugiego

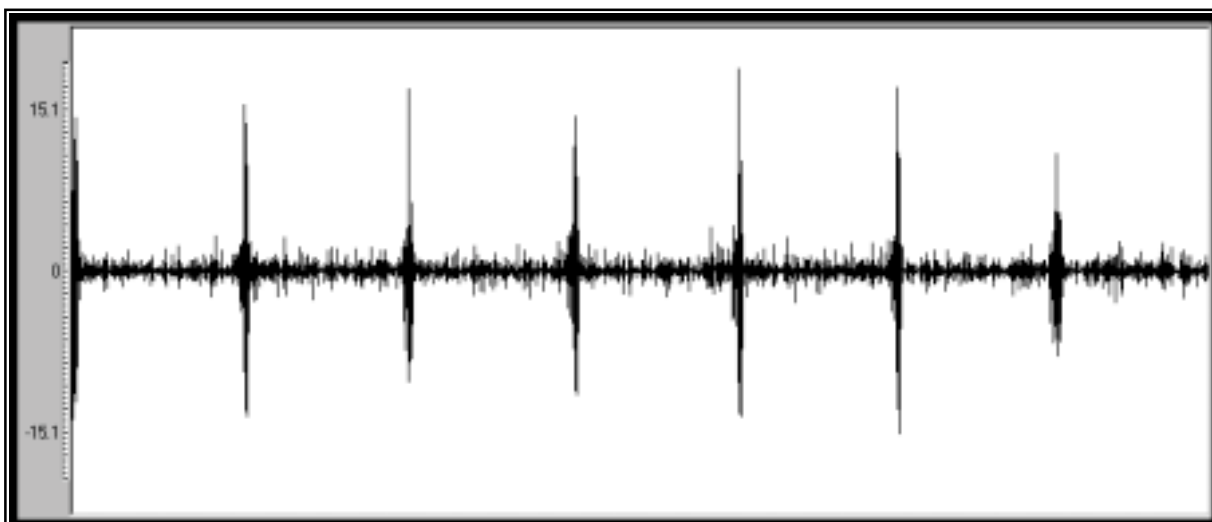
Dla większej przejrzystości przebiegów wyłączono opcję rysowania linii progów przez odznaczenie pozycji 'Full reconstr. line' w menu 'Threshold'. Sygnał oryginalny oraz przebiegi uzyskane w wyniku rekonstrukcji zamieszczone są na rysunkach 6.1.1 – 6.1.5.



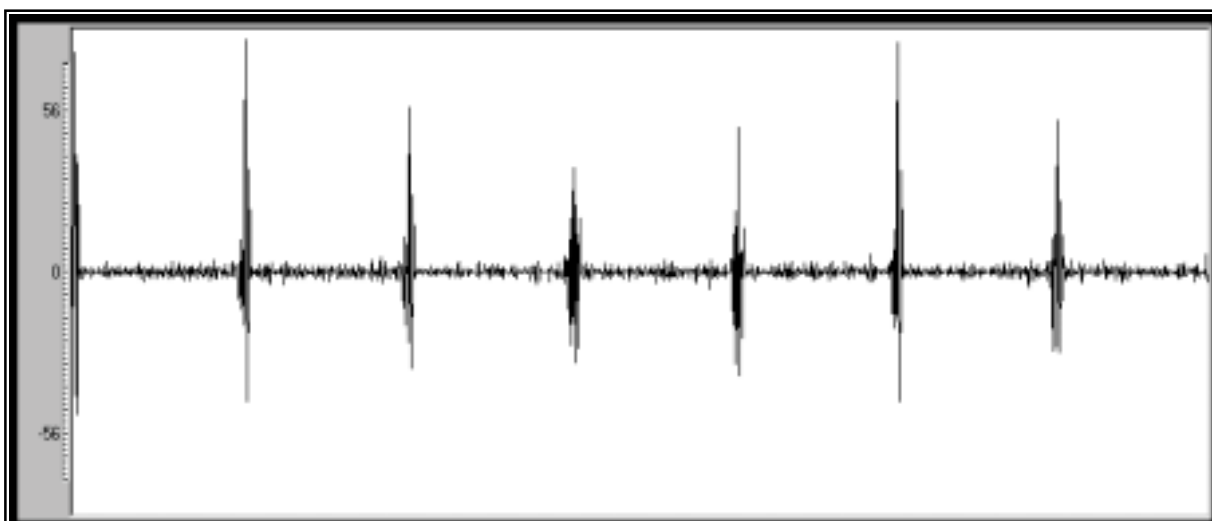
Rys. 6.1.1 Sygnał oryginalny 'N0'



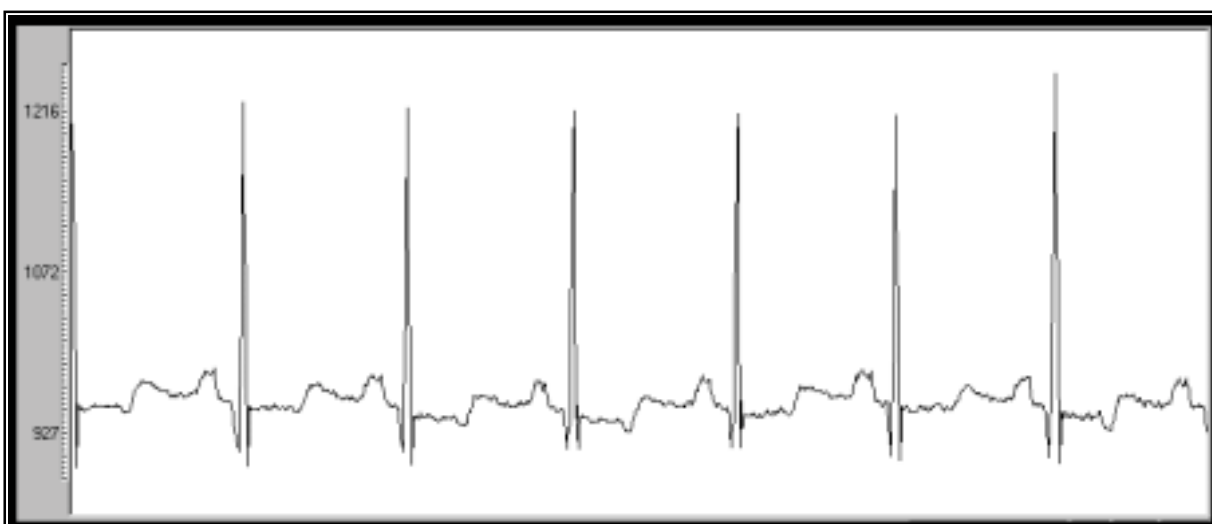
Rys. 6.1.2 Sygnał zrekonstruowany



Rys. 6.1.3 Szczegóły – poziom pierwszy



Rys. 6.1.4 Szczegóły – poziom drugi



Rys. 6.1.5 Aproksymacja – poziom drugi

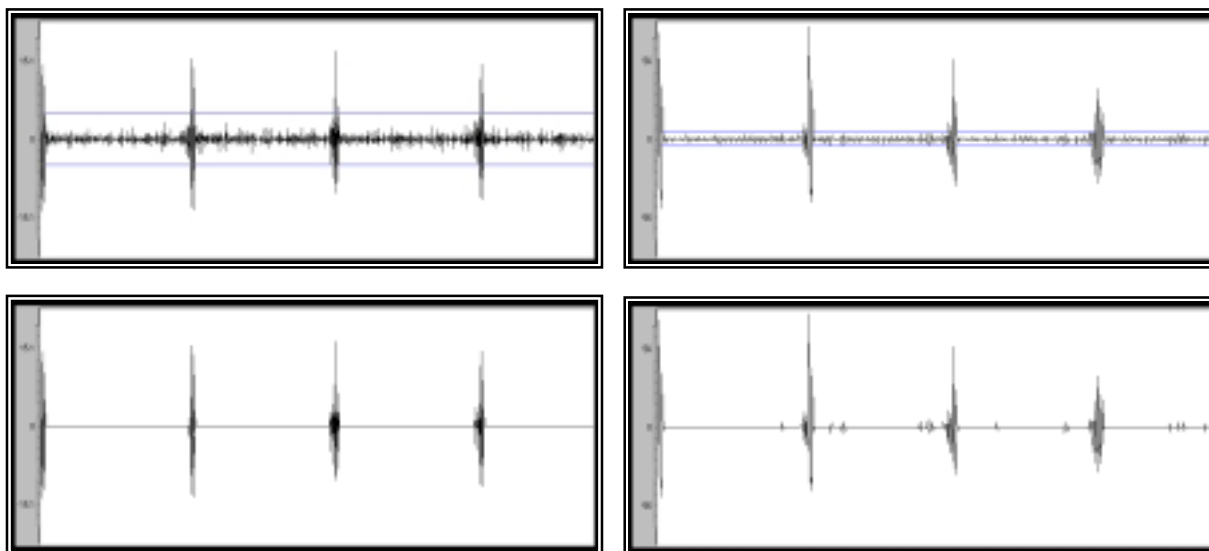
Pierwszym wnioskiem, jaki się nasunął po obejrzeniu otrzymanych przebiegów jest bezstratność transformacji falkowej. Można to stwierdzić przez porównanie sygnału oryginalnego i zrekonstruowanego ze współczynników falkowych. Obydwa przebiegi są identyczne. Wynika z tego, że przejście z dziedziny czasu do przestrzeni czasowo-częstotliwościowej reprezentowanej przez współczynniki falkowe odbywa się z zachowaniem całej energii sygnału. Podobnie też odwrotna transformacja, czyli rekonstrukcja powoduje przekazanie całej informacji z powrotem do czasowej reprezentacji sygnału.

Na podstawie zamieszczonych rysunków można zaobserwować dokładnie, w jaki sposób dwupoziomowa dekompozycja podzieliła sygnał na trzy przedziały częstotliwościowe. Przebieg 'Detail: level 1' reprezentuje składowe sygnału o najwyższych częstotliwościach. Został on otrzymany w wyniku filtracji górnoprzepustowej na pierwszym poziomie dekompozycji (przy użyciu najkrótszej falki). Widać, że amplituda tego sygnału utrzymuje się praktycznie na poziomie $\pm 2\mu V$ z wyjątkiem załamka QRS, gdzie odchylenia zwiększają się do ok. $\pm 15\mu V$. 'Detail: level 2' zawiera składowe o średnich częstotliwościach. Powstał w wyniku filtracji dolnoprzepustowej na poziomie pierwszym i górnoprzepustowej na poziomie drugim, na którym długość falki jest dwukrotnie większa. Charakter otrzymanego sygnału jest podobny, jak w poprzednim przypadku. Przebieg ma jedynie większe amplitudy: $\pm 50\mu V$ na załampakach QRS oraz $\pm 3\mu V$ w pozostałej części sygnału. Sygnał 'Aproximation: level 2' został otrzymany w wyniku dwukrotnej filtracji filtrami dolnoprzepustowymi i reprezentuje składowe sygnału oryginalnego o najniższej częstotliwości. Ma on podobny kształt do sygnału oryginalnego, jednakże generalnie jest bardziej wygładzony.

Tutaj nasuwa się następny wniosek potwierdzający teoretyczne rozważania: sygnał zrekonstruowany, czyli 'Reconstructed signal' jest sumą cząstkowych sygnałów 'Detail: level 1', 'Detail: level 2' i 'Aproximation: level 2'. Zatem skoro sygnał zrekonstruowany jest równy sygnałowi oryginalnemu (wniosek pierwszy), to kształt 'Aproximation: level 2' różni się od sygnału oryginalnego dokładnie wartościami próbek przebiegów detali pierwszego i drugiego poziomu.

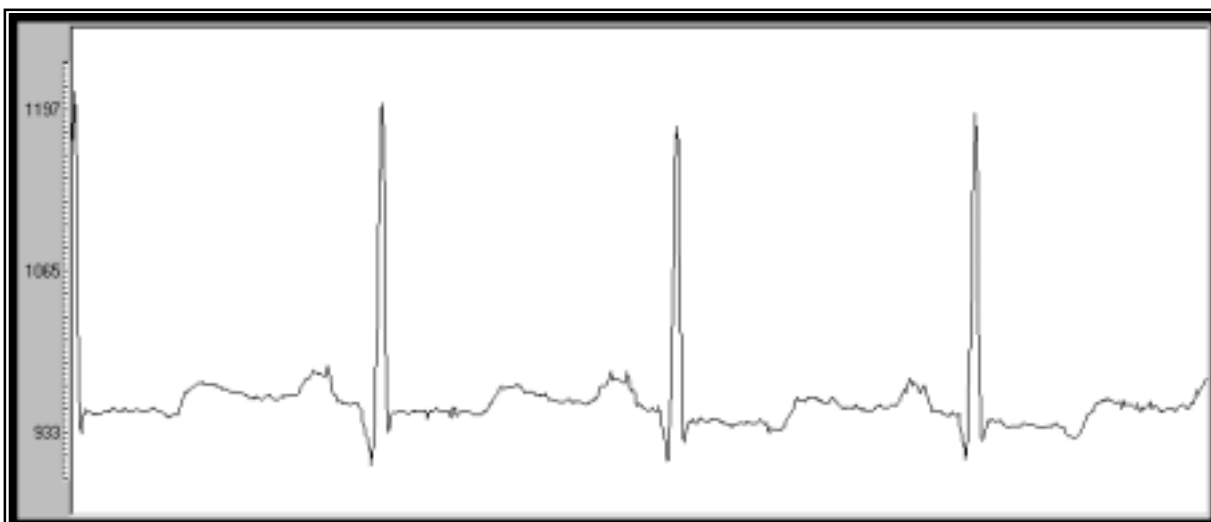
Warto tutaj jeszcze wspomnieć o sygnale EKG, jako obiekcie przetwarzania falkowego. Na podstawie przebiegów detali z dwóch poziomów widać, że energia sygnału (z pominięciem załampaków QRS) równomiernie rozkłada się na całej jego długości. Amplituda obydwu składowych wydzielonych z sygnału jest w przybliżeniu jednakowa: wynosi $\pm 2\mu V$ dla pierwszego poziomu i $\pm 3\mu V$ dla drugiego. Natomiast załamki QRS, które w sygnale oryginalnym są praktycznie szpilkami o amplitudzie $250\mu V$ magazynują większą energię sygnału. Objawia się to zwiększoną amplitudą detali na obydwu poziomach. W szczególności na drugim poziomie stosunek amplitud na załamku QRS i poza nim wynosi $50\mu V/3\mu V$, czyli ponad 16-krotny.

Dalszy etap doświadczenia ma za zadanie ocenić wpływ doboru progu obcinania na jakość przetwarzania falkowego. Po każdym procesie dekompozycji jest on automatycznie dobierany. Wartość progu obliczany jest na podstawie współczynników detali z pierwszego poziomu. W naszym wypadku próg ten wyniósł 5,04. Pozostawiając jego wartość nie zmienioną dokonano rekonstrukcji sygnału za pomocą drugiej listwy 'Thresholded reconstructions'. Efekt wyzerowania współczynników detali, których bezwzględna wartość nie przekraczała wartości progów pokazuje rysunek 6.1.6. Dla porównania dołączono także sygnały z pełnej rekonstrukcji z włączonymi liniami pokazującymi wartość progu. Zamieszczone przebiegi zostały dwukrotnie rozciągnięte, przez co łatwiej jest zaobserwować ich charakterystyczne cechy.

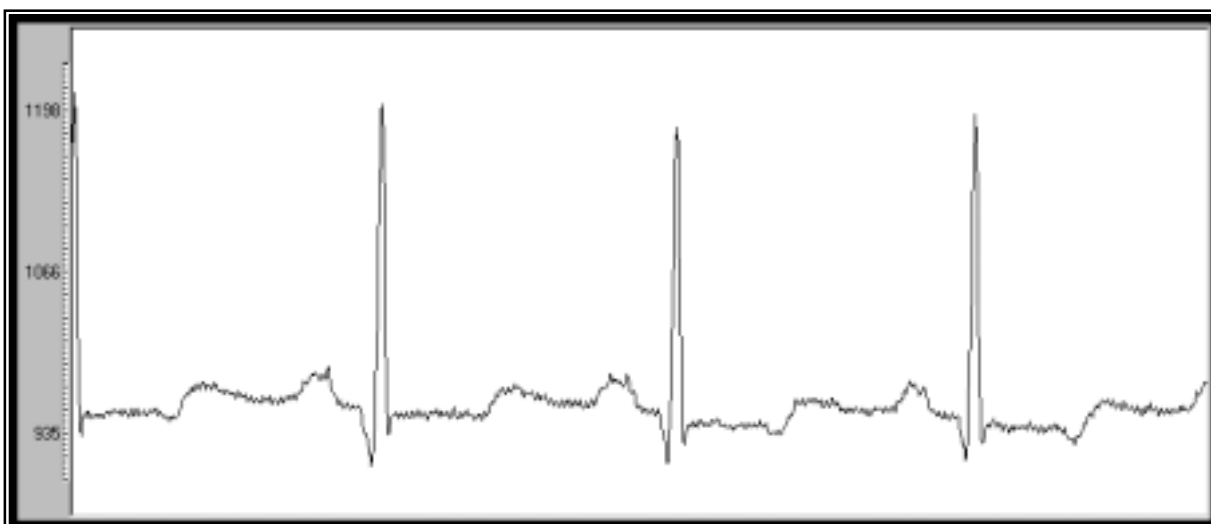


Rys. 6.1.6 Sygnały rekonstrukcji współczynników detali pierwszego (na lewo) i drugiego (na prawo) poziomu. U góry zaznaczony jest próg kompresji. U dołu sygnały po wyzerowaniu współczynników.

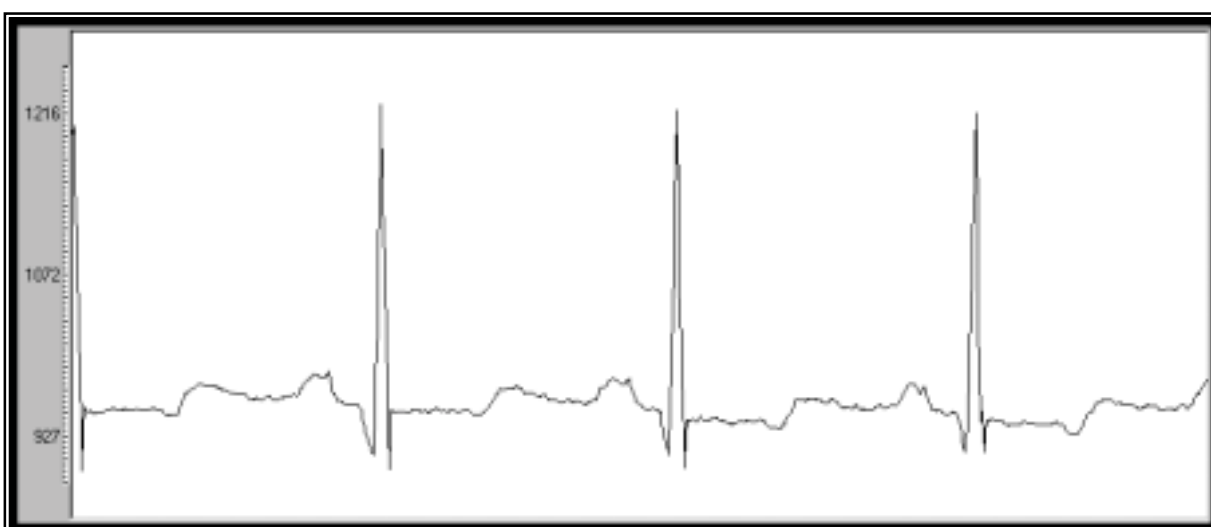
Na rysunkach tych widać, że obcięciu uległy praktycznie wszystkie współczynniki spoza obszarów załamek QRS, które z kolei zachowały się prawie nie zmienione. Na poziomie pierwszym zostało wyzerowanych 95,71% współczynników, natomiast na drugim 87,35%. Rysunku 'Aproximation: level 2' z 'Thresholded reconstructions' nie zamieszczono ponieważ współczynników aproksymacji nie obcina się, gdyż stanowią one niejako „nośną” całego sygnału. Łatwo się domyślić więc, że przebieg ten jest identyczny z jego odpowiednikiem z 'Full reconstructions'. Na koniec warto zobaczyć finalny efekt kompresji falkowej z progowaniem, czyli rekonstrukcję całego sygnału po operacji progowania. Uruchamia się ją poprzez wybór pozycji 'Reconstructed signal' z listwy 'Thresholded reconstructions'. Przebieg ten jest pokazany na rys. 6.1.7.



Rys. 6.1.7 Sygnał zrekonstruowany po wyzerowaniu współczynników



Rys. 6.1.8 Sygnał oryginalny



Rys. 6.1.9 Sygnał zrekonstruowany ze współczynników aproksymacji

Widać, że przebieg odtworzony ze skompresowanych współczynników (rys.6.1.7) jest praktycznie taki sam jak sygnał oryginalny, tylko bardziej wygładzony. W sumie usuniętych zostało 69,65% z pośród wszystkich współczynników, przy stracie jakości PRD wynoszącej 1,539%. Dla oszacowania jakości rekonstrukcji załączono jeszcze dwa rysunki rozciągniętych sygnałów: oryginalnego (rys.6.1.8) oraz zrekonstruowanego ze współczynników aproksymacji (rys.6.1.9). Przy takiej ilości wyzerowanych współczynników (95,71% na poziomie pierwszym i 87,35% na drugim) można by się spodziewać, że sygnał finalny będzie bardziej podobny do sygnału aproksymowanego, niż do oryginalnego. W istocie jest inaczej. Widać, że zostały w nim usunięte drobne oscylacje, podobnie jak w aproksymacji. Jednak w odróżnieniu do aproksymacji zachowany został charakter sygnału oryginalnego. W powiększeniu można zauważyć zniekształcenia konturów w sygnale aproksymowanym, szczególnie na załamek QRS. W sygnale zrekonstruowanym załamki te są gładkie (to widać dopiero w jeszcze większym powiększeniu). Dzięki niewielkiej ilości zachowanych współczynników z pierwszego i drugiego poziomu detali finalny sygnał zachował wierność kształtów sygnału oryginalnego.

Uzyskany sygnał zapisano następnie na dysku pod nazwą N0_thr5.flk. Należy pamiętać o tym, że na dysk zapisywane są współczynniki falkowe po operacji progowania. Sama rekonstrukcja jest pomocna przy doborze progu, ponieważ można wtedy wizualnie ocenić skutki tego wyboru. Przy tak ustalonym progu (5,04) plik ma 2354 bajty objętości. Powyższy eksperyment powtórzono jeszcze trzykrotnie. Dla tego samego sygnału zmieniono próg na wartość 0 i zapisano pod nazwą N0_thr0.flk, potem na wartość 1 i zapisano pod nazwą N0_thr1.flk oraz próg 20 nazywając sygnał N0_thr20.flk. Rezultaty wynikłe z doborów progów zamieszczono w tabeli 6.1.1.

Sygnał N0.d				
falka: db2				
poziom dekompozycji: 'two levels'				
rodzaj wypełnienia: 'Limit padding'				
Próg kompresji:	0	1	5,04	20
Zera – poziom pierwszy:	0%	56,0%	95,71%	99,71%
Zera – poziom drugi:	0%	21,01%	87,35%	95,33%
Zera – ogółem:	0%	33,22%	69,65%	73,65%
PRD:	0%	0,335%	1,539%	2,534%
Nazwa pliku:	N0_thr0.flk	N0_thr1.flk	N0_thr5.flk	N0_thr20.flk
Wielkość pliku:	7144 bajty	5322 bajty	2354 bajty	1927 bajtów

Tab.6.1.1 Parametry sygnałów przy zastosowaniu różnych progów kompresji

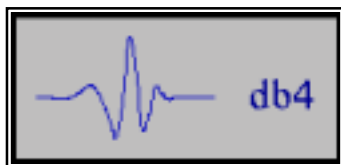
Wyniki zgromadzone w tabeli potwierdzają fakt, że ze wzrostem progu obcinania zwiększa się liczba zer, wzrasta wskaźnik PRD określający stopień zniekształceń w stosunku do sygnału oryginalnego oraz maleje objętość pliku potrzebnego do archiwizacji skompresowanego sygnału. Podczas nastawiania najkorzystniejszej wartości progu należy się

kierować tymi trzema wartościami oraz oczywiście wizualną obserwacją, bo jeśli chodzi o informacje zawarte w sygnale EKG to tylko specjalista może określić, które dane są istotne, a które można odrzucić. Na podstawie kształtów otrzymanych przebiegów można stwierdzić, że najkorzystniejszy próg, to ten wyznaczony automatycznie po dekompozycji o wartości 5,04. Z pośród zaprezentowanych progów charakteryzuje się najkorzystniejszym stosunkiem stopnia kompresji do jakości otrzymanego sygnału. W przypadku progu 1 jakość sygnału była praktycznie idealna, co pokazuje minimalny współczynnik PRD (0,335%), lecz stopień kompresji niewystarczający. Sygnał w rzeczywistości zawierał dużo informacji (próbek) nie mających istotnego znaczenia dla całości sygnału. Przeciwnością tego przypadku jest sygnał otrzymany z zastosowania progu 20. Cechuje go bardzo wydajna kompresja. Rozmiar pliku jest prawie cztery razy mniejszy w stosunku do rozmiaru pliku przy zerowej kompresji, lecz niewiele mniejszy od sygnału skompresowanego progiem równym 5,04. Widać to również na podstawie wartości wyzerowanych współczynników, gdyż to od nich bezpośrednio wynika objętość pliku. Jednak z drugiej strony tak wysoki próg spowodował wycięcie współczynników istotnych dla sygnału, który przez to charakteryzuje się dużymi zniekształceniami: duża wartość PRD. Sam kształt zrekonstruowanego sygnału miał kształt praktycznie taki jak sygnału aproksymowanego (tzn. całkowicie pozbawionego szczegółów).

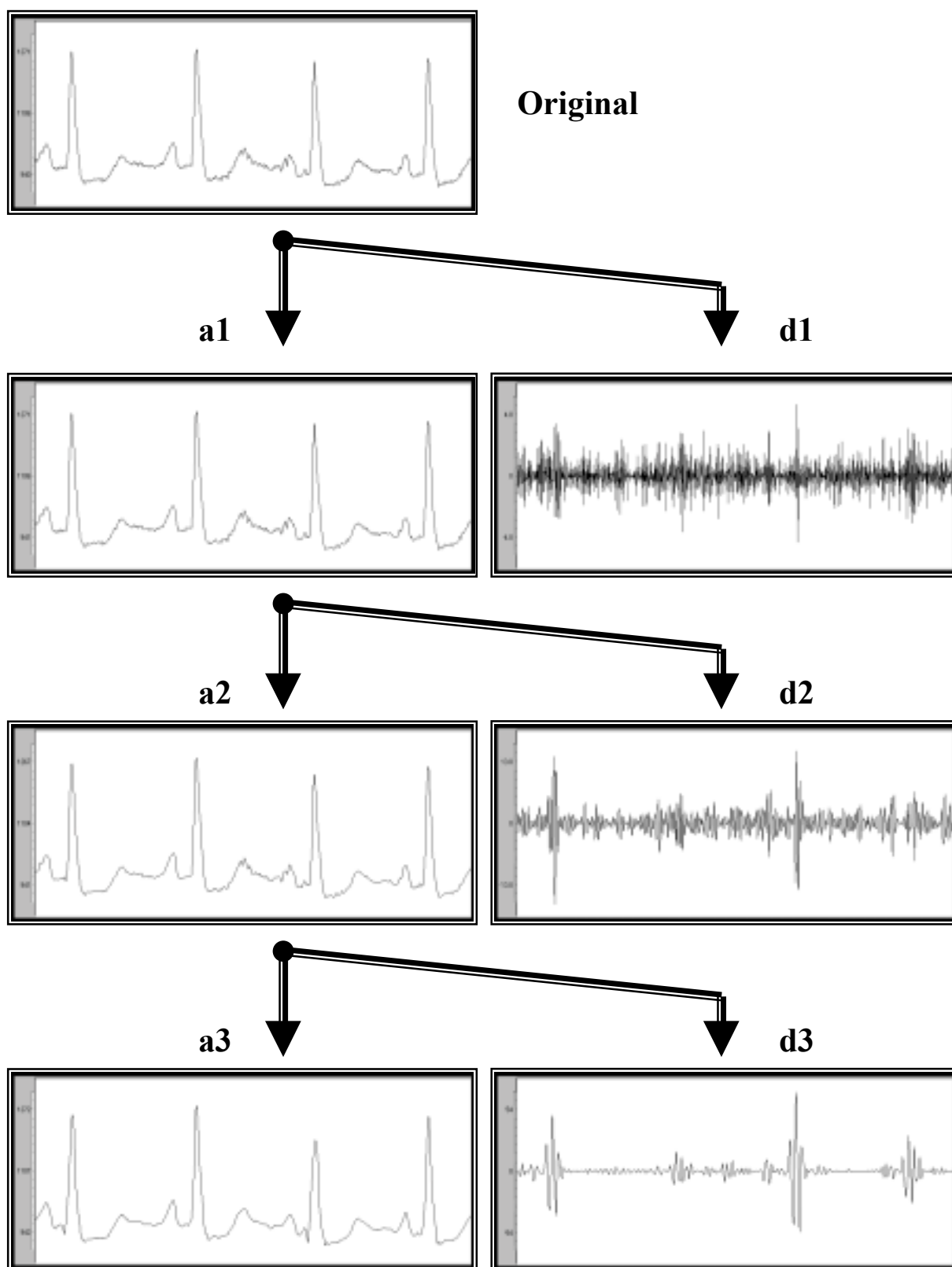
6.2. Dobór poziomu dekompozycji

Doświadczenie ma za zadanie ocenić wpływ doboru poziomu dekompozycji na kompresję falkową oraz dla wybranego sygnału wybrać jego optymalną wartość. W ćwiczeniu zastosowano sygnał 'N18.d' oraz następujące ustawienia programu: falka 'db4', typ wypełnienia 'Limit padding' oraz poziom dekompozycji kolejno od 1 do 6.

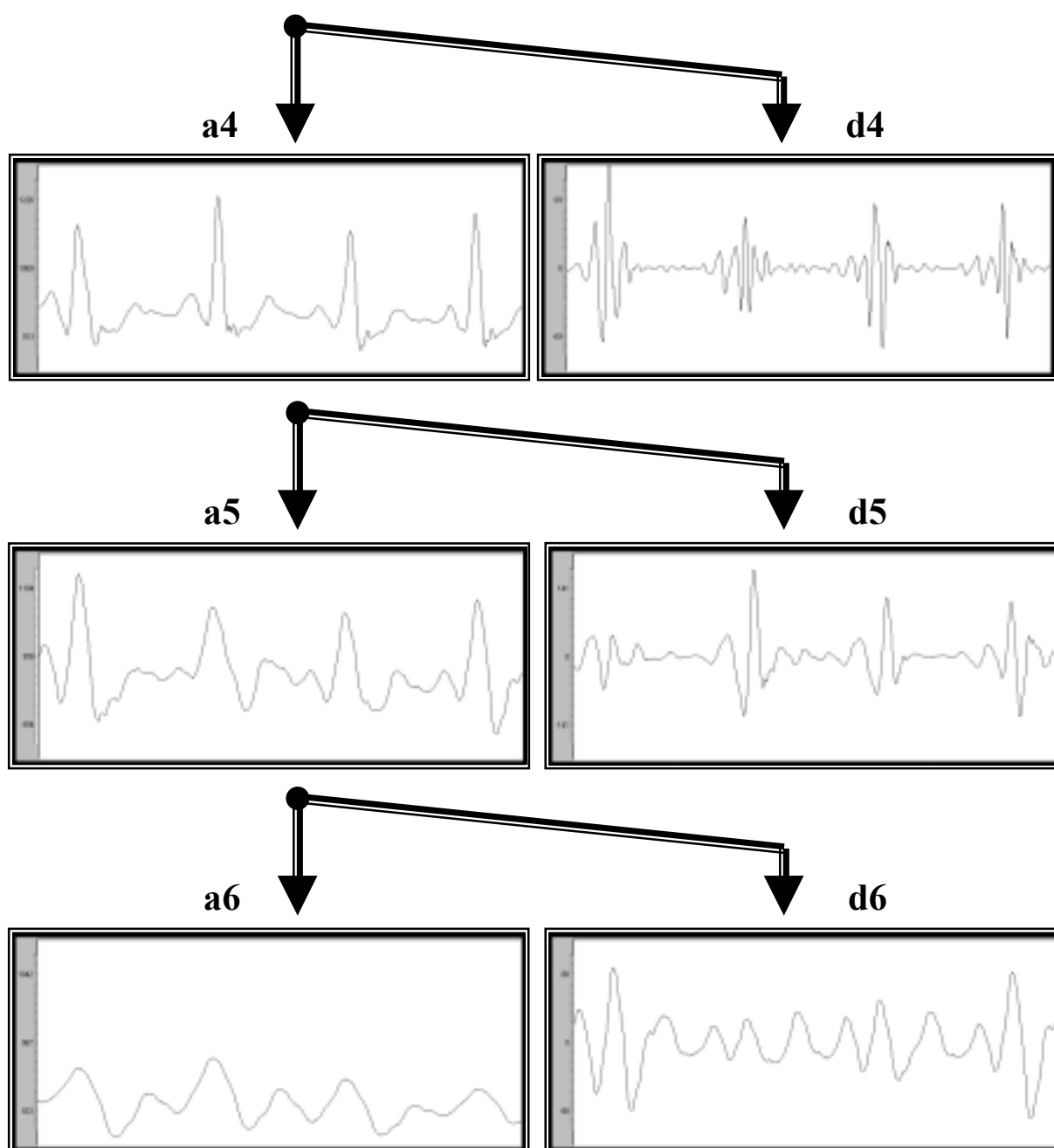
Na początku należy uzmysłowić sobie, że każdy kolejny poziom dekompozycji powoduje zdekomponowanie aktualnego sygnału aproksymacji na kolejne dwa sygnały aproksymacji i detali poziomu wyższego. Oczywiście sygnał w tym kontekście znaczy zestaw współczynników falkowych, gdyż w procesie dekompozycji tylko na nich się operuje. Dekomponując sygnał oryginalny na aproksymacje i szczegóły poziomu pierwszego zakłada się niejawnie, że jest on sygnałem aproksymacji z nieistniejącego w rzeczywistości niższego poziomu. W ten sposób dekompozycja każdego poziomu wygląda tak samo i przeprowadza się ją na zasadzie rekurencji. Zaznaczyć należy także, że dekomponuje się tylko i wyłącznie współczynniki aproksymacji, współczynników detali w „klasycznej” dekompozycji falkowej nie rusza się. Znaczy to, że współczynniki szczegółów poziomu pierwszego uzyskane przy pomocy jednopoziomowej dekompozycji będą takie same jak uzyskane poprzez wielopoziomową dekompozycję niezależnie od liczby poziomów. Dekompozycja falkowa sygnału tworzy tak zwane drzewo dekompozycji, które składa się z 'n' sygnałów detali kolejnych poziomów od 1 do n i jednego sygnału aproksymacji na poziomie n-tym. Celem tego doświadczenia jest określenie jakie to drzewo ma być duże, czyli od którego punktu każdy następny rozkład będzie tylko zbędnym nakładem obliczeń. Dla załadowanego sygnału 'N18.d' drzewo dekompozycji ma wygląd jak na rysunku 6.2.1. Drzewo jest sześciopoziomowe, gdyż tyle poziomów dekompozycji udostępnia program. Aby uzyskać sygnały (współczynniki) aproksymacji zdekomponowano oddzielnie sygnał oryginalny za każdym razem z innym ustawieniem poziomu dekompozycji. Dla sygnałów detali wystarczy dekompozycja na szóstym poziomie. Warto jeszcze zaznaczyć, że przebiegi widoczne na rysunkach to sygnały uzyskane w wyniku rekonstrukcji danych współczynników falkowych.



Rys. 6.2.2 Falka Daubechies 4



Rys. 6.2.1a Drzewo dekompozycji falkowej do poziomu szóstego dla sygnału 'N18.d'
(c.d. na drugiej stronie)



Rys. 6.2.1b Drzewo dekompozycji falkowej do poziomu szóstego dla sygnału 'N18.d'

Generalnie, z punktu widzenia kompresji najbardziej interesujące są przebiegi szczegółów (d1-d6) z tego względu, że tylko na nich można operować (eliminować „zbędne” współczynniki). Sytuacja idealna jest wtedy, gdy na danym poziomie szczegółów cała energia sygnału kumuluje się na kilku współczynnikach, to znaczy kiedy mała część współczynników ma duże amplitudy, a pozostałe małe. Wtedy usunięcie tych „nieistotnych” współczynników spowoduje dużą redukcję danych przy zachowaniu istoty całego sygnału. Zatem na początek warto przyrzeć się bliżej charakterowi oryginalnego sygnału oraz kształcie falki (przedstawionej na rys.6.2.2), ponieważ ich wzajemna korelacja (podobieństwo) decyduje o

wielkości współczynników. Widać, że sygnał szczególnie dużo informacji gromadzi na załawkach QRS. Są to duże szpilki składające się ze składowych wysokoczęstotliwościowych (ostre załamki R) oraz ze średnich częstotliwości. Widać to na zamieszczonych przebiegach. Po odjęciu od sygnału dwóch pierwszych przebiegów detali 'd1' i 'd2' wierzchołki załawków R znacznie się „wytepiły”, szczególnie to jest widoczne na pierwszym i trzecim załamku 'a3' otrzymanym po odjęciu 'd2' i 'd3' od 'a1'. Kontury QRS-ów widoczne są aż do poziomu piątego – 'a5'. Widać dokładnie, jak w sygnałach 'a3', 'a4' i 'a5' falka próbuje dopasować się do kształtu QRS. Warto zaobserwować, jak falka z przebiegu 'd5' prawie idealnie wpasowała się w załamek P i RS drugiego QRS-a z sygnału aproksymacji. Jeśli chodzi o pozostałą część sygnału, czyli załamek P, T i odcinek między kolejnymi QRS-ami, to jest on raczej wolnozmienny. Aż do czwartego poziomu detali amplitudy sygnału są niewielkie w tych miejscach, dopiero na wyższych poziomach współczynniki te odgrywają duże znaczenie dla sygnału.

Powracając do zagadnienia optymalnej ilości poziomów dekompozycji, na podstawie obserwacji zamieszczonych przebiegów wynika, że najkorzystniejsza byłaby dekompozycja do czwartego (ewentualnie piątego) poziomu. Pierwszy poziom detali ma tak małą amplitudę (nie przekraczająca $2\mu\text{V}$), że praktycznie cały można potraktować jako szum. Na poziomach drugim, trzecim i czwartym widać coraz większy wzrost amplitudy detali w rejonach QRS-ów przy niezmienniej amplitudzie w pozostałych częściach sygnałów (czyli to o co nam chodzi). Na piątym poziomie ten stosunek amplitud znowu zaczyna maleć, by w końcu na szóstym poziomie zrównać się. Należy też mieć świadomość jaki jest rozkład ilościowy współczynników między poziomami. Na pierwszy poziom detali przypada 50% wszystkich współczynników, na drugi 25%, na trzeci 12,5%, czwarty 6%, piąty 3%, a na szósty zaledwie 1,5%. Pozostałe współczynniki przypadają sygnałowi aproksymacji i w tym wypadku będzie to 1,5%. Tak więc w rzeczywistości powyżej trzeciego poziomu nie zauważa się praktycznych zmian efektów kompresji. Natomiast jeśli chodzi o sam proces dekompozycji i rekonstrukcji, to poziom dekompozycji nie ma znaczenia, jeśli nie wycina się współczynników. Sama transformacja falkowa jest bezstratna. Zatem algorytm ustalania optymalnego poziomu dekompozycji może wyglądać następująco:

- wstępna dekompozycja do sześciu poziomów
- odznaczenie globalnego progu
- znalezienie dla każdego poziomu optymalnego progu
- w przypadku wyższych poziomów, jeżeli ustalony próg spowoduje wyzerowanie zbyt małej ilości ogólnych współczynników należy z niego zrezygnować i ustawić na zero
- ostatecznie ustawić poziom dekompozycji na wartość, dla której dla najwyższego poziomu ustawiona została niezerowa wartość

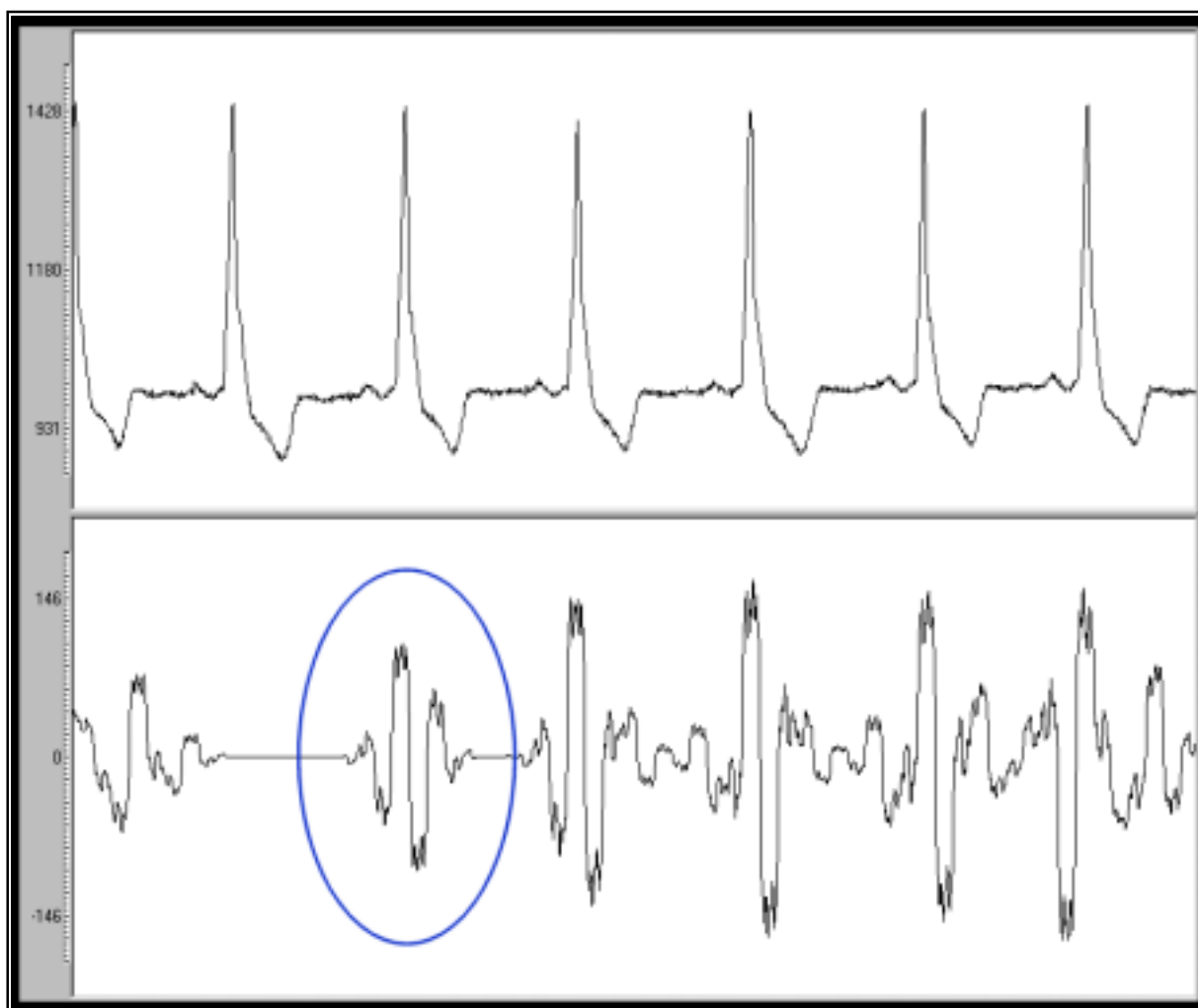
W tym sposobie ustalania progu chodzi o to, że jeżeli na danym poziomie nie usuwa się współczynników to nie ma sensu dekomponować sygnału na tym poziomie. W prezentowanym przykładzie kompresja na piątym i szóstym poziomie nie przyniosła wymiernych korzyści, więc zrezygnowano z tych progów. W efekcie zdekomponowano sygnał do czwartego poziomu, po czym nieznacznie wyregulowano jeszcze wartości progów dla każdego z poziomów indywidualnie. Dla wartości progów 5,5 dla pierwszego, 7,0 – drugiego, 6,5 – trzeciego oraz 8,0 dla czwartego poziomu otrzymano 83,70% zerowych współczynników, co odpowiada kompresji 5,5 raza przy PRD równym 2,231%.

6.3. Dobór falki

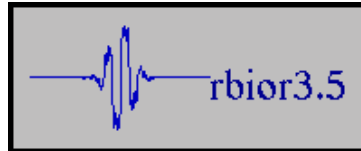
Znalezienie właściwej falki do kompresji sygnału uwarunkowane jest przede wszystkim kształtem i właściwościami sygnału. Chodzi o to, aby tak dobrać bazę falkową (zbudowaną na podstawie szukanej falki-matki), żeby znaleźć możliwie silną korelację sygnału z jak najmniejszą liczbą falek bazowych. W programie dostępny jest dosyć duży wybór falek. Są między innymi falki ortogonalne (Haar, Daubechies), o zmniejszonej niesymetrii (Symlets) oraz biortogonalne (Biorthogonal, Reverse biorthogonal). Nie znając jednak dokładnie matematycznych właściwości sygnału trudno jest arbitralnie dobrać właściwą falke. Zwykle robi się to empirycznie szukając podobieństwa między falką, a sygnałem. Dekompozycja falkowa jest niczym innym, jak „rozłożeniem” sygnału na zespół falek bazowych, zatem pożądane jest, aby charakter tych sygnałów był jak najbardziej zbliżony.

Poniżej zamieszczono kilka przykładów, których celem jest pokazanie, że każdy współczynnik w zdekomponowanym sygnale odpowiada za jedną falke w jego oryginale (lub inaczej w zrekonstruowanym sygnale).

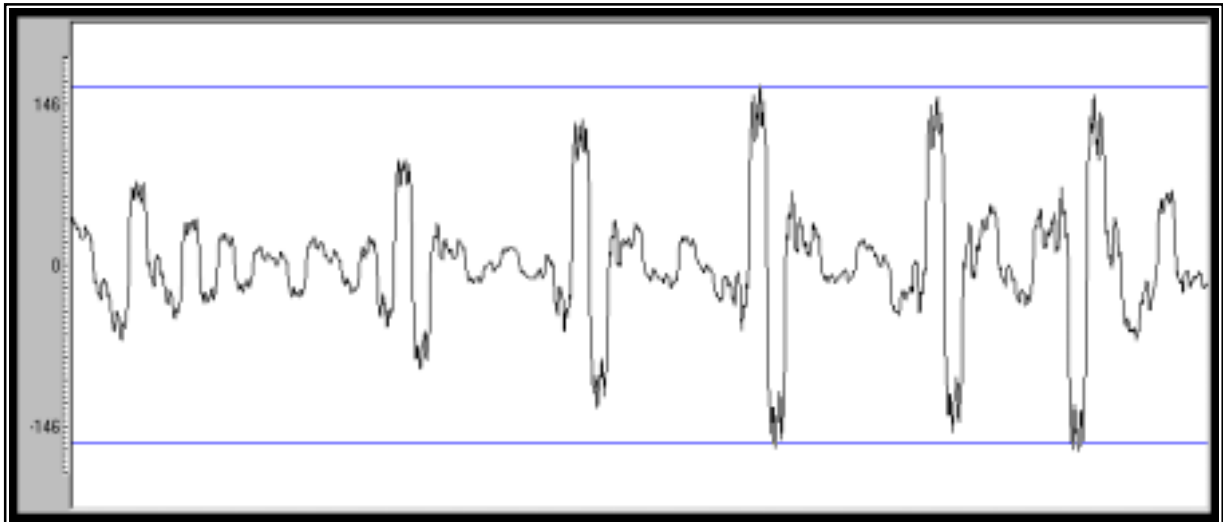
W programie załadowano sygnał ‘Lbb2.d’ oraz wybrano falke ‘rbior3.5’ (rys.6.3.1b). Dla standardowego ustawienia typu rozszerzenia zdekomponowano sygnał do szóstego poziomu włącznie. Następnie zrekonstruowano szósty próg detali i po ustawieniu maksymalnego progu $160\mu\text{V}$ ponownie zrekonstruowano ten poziom przy pomocy ‘Thresholded reconstructions’. Otrzymany przebieg włącznie z oryginalnym sygnałem zaprezentowano na rys. 6.3.1a.



Rys.6.3.1a U góry sygnał oryginalny ‘Lbb2.d’. Na dole rekonstrukcja szóstego poziomu z progiem kompresji $160\mu\text{V}$. W zaznaczonym miejscu widać wyodrębnioną pojedynczą falke ‘rbior3.5’.



Rys.6.3.1b Falka 'Reverse biorthogonal 3.5'

Rys.6.3.1c Rekonstrukcja współczynników szóstego poziomu bez obcinania. Poziome linie wskazują amplitudę progu kompresji $160\mu\text{V}$.

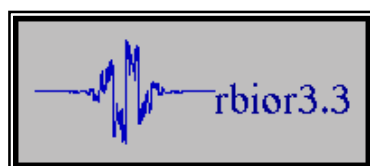
Rysunek 6.3.1 w istocie dowodzi, że jest możliwe wyodrębnienie ze zrekonstruowanego sygnału detali „zaburzeń”, które będą miały taki sam kształt jak falka. Przebieg ten może nie zupełnie się wydawać podobny do falki na rys.6.3.1b, ponieważ jest odwrócony. Efekt ten spowodowany jest ujemnym znakiem współczynnika, z którego powstała rekonstrukcja.

Proces dekompozycji przekształca sygnał w uporządkowany ciąg współczynników falkowych. Zakładając, że sygnał ma 2048 próbek to po pierwszym poziomie dekompozycji stworzonych zostanie 1024 współczynników detali pierwszego poziomu. Każdy z tych współczynników reprezentuje falkę o amplitudzie proporcjonalnej do swojej wartości i zasięgu równym długości filtrów związanych z daną falką. Na przykład dla falki 'Daubechies 2' filtry mają długość czterech próbek, zatem na pierwszym poziomie każdy współczynnik koreluje z czterema próbkami sygnału oryginalnego. Faktycznie więc na pierwszym poziomie mamy do czynienia wysokimi częstotliwościami. Ta sama sytuacja dzieje się na pozostałych poziomach, tylko z każdym następnym poziomem falki się rozszerzają i obejmują swoim zasięgiem dwukrotnie większy obszar i częstotliwość reprezentacji sygnału maleje. Filtry falki 'rbior3.5' mają długość dwunastu próbek, czyli na pierwszym poziomie zasięg falki obejmuje 12 próbek sygnału, na drugim 24, trzecim 48, czwartym 96, piątym 192, i w końcu na szóstym 384 próbki. Zgadza się to z zamieszczonym rysunkiem bo widać, że zaznaczona falka zajmuje prawie jedną piątą długości sygnału. Wycięcie wszystkich współczynników w pobliżu (tego, którego falkę oglądamy) oraz duża szerokość falki (dużo próbek do jej wykreślenia) spowodowały, że tak doskonale się wyeksponowała na rysunku. Warto jeszcze zauważyć, że amplituda falki ma ok. $100\mu\text{V}$, podczas gdy próg obcinania ustawiony został na $160\mu\text{V}$. Faktyczna amplituda współczynnika odpowiadającego falki była bez wątpienia wyższa. Kolejne etapy rekonstrukcji sprawiły, że końcowa amplituda falki okazała się mniejsza.

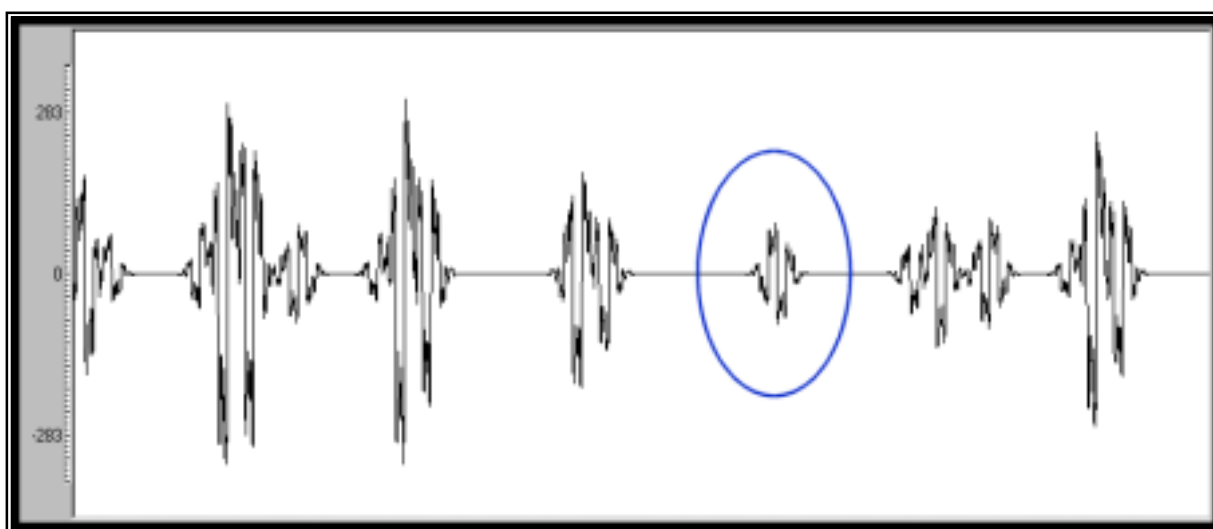
Dodatkowo na rysunku 6.3.1c zamieszczono rekonstrukcję detali szóstego poziomu bez obcinania współczynników z włączoną linią wskazującą wartość progu. Jest to rekonstrukcja

wszystkich 64 współczynników (dla sygnału o 2048 próbkach tyle przypada na szósty poziom) o różnej amplitudzie.

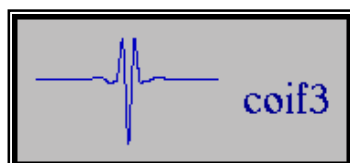
Poniżej zaprezentowano jeszcze kilka przykładów wyodrębnienia falki z sygnału dla innych typów falek.



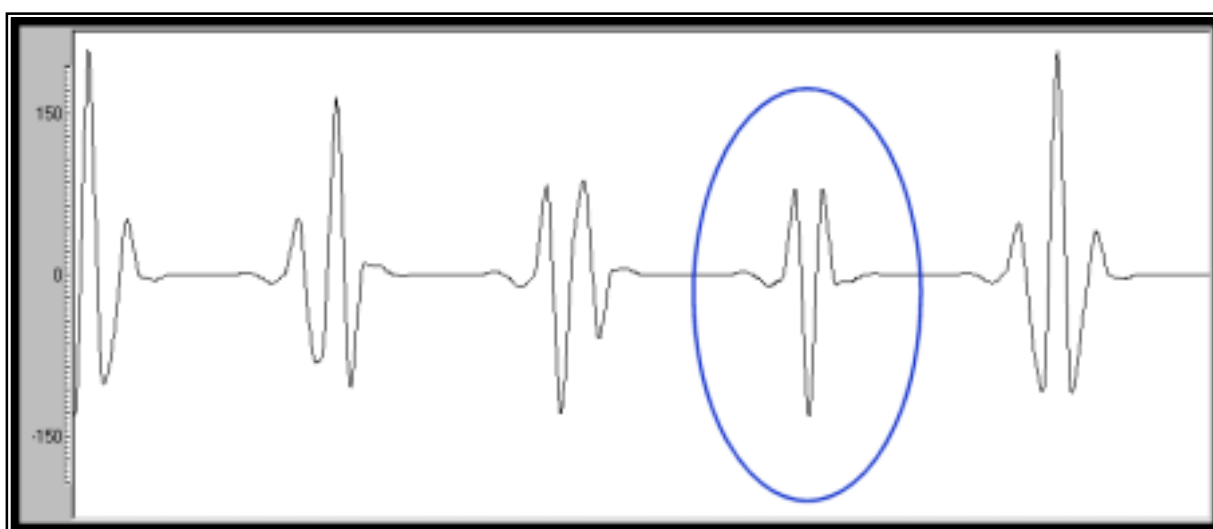
Rys.6.3.2a Falka 'Reverse biorthogonal 3.3'



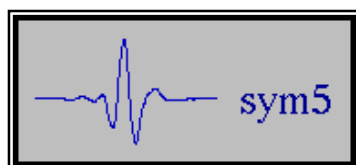
Rys.6.3.2b Rekonstrukcja współczynników piątego poziomu z progiem kompresji $160\mu\text{V}$. Zaznaczona falka jest odwrócona w stosunku do 'rbior3.3', czyli zbudowana została ze współczynnika ujemnego.



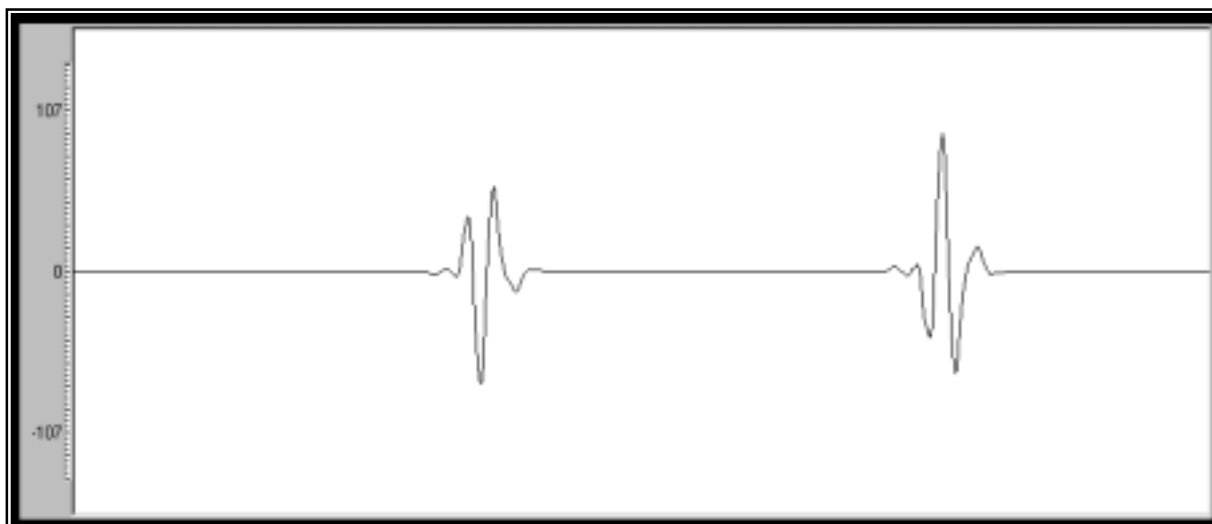
Rys.6.3.3a Falka 'Coiflets 3'



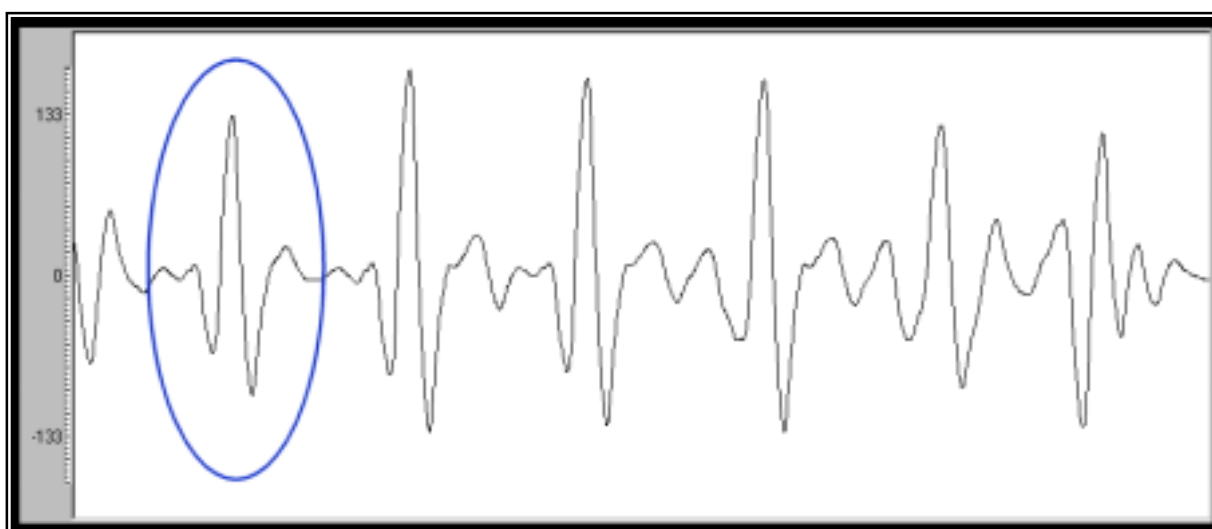
Rys.6.3.3b Rekonstrukcja współczynników piątego poziomu z progiem kompresji $160\mu\text{V}$. Współczynnik dodatni.



Rys.6.3.4a Falka 'Symlets 5'



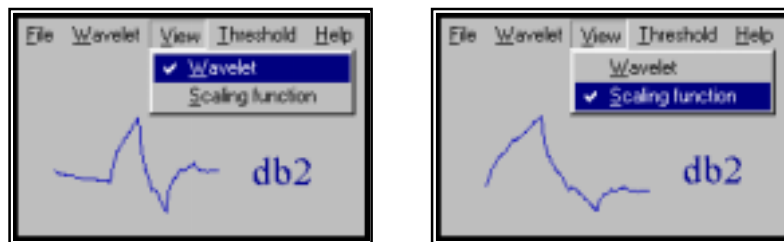
Rys.6.3.4b Rekonstrukcja współczynników czwartego poziomu z progiem kompresji $160\mu\text{V}$. Przebieg rozciągnięty. Z lewej strony widać falkę ze współczynnika ujemnego, z prawej dodatniego.



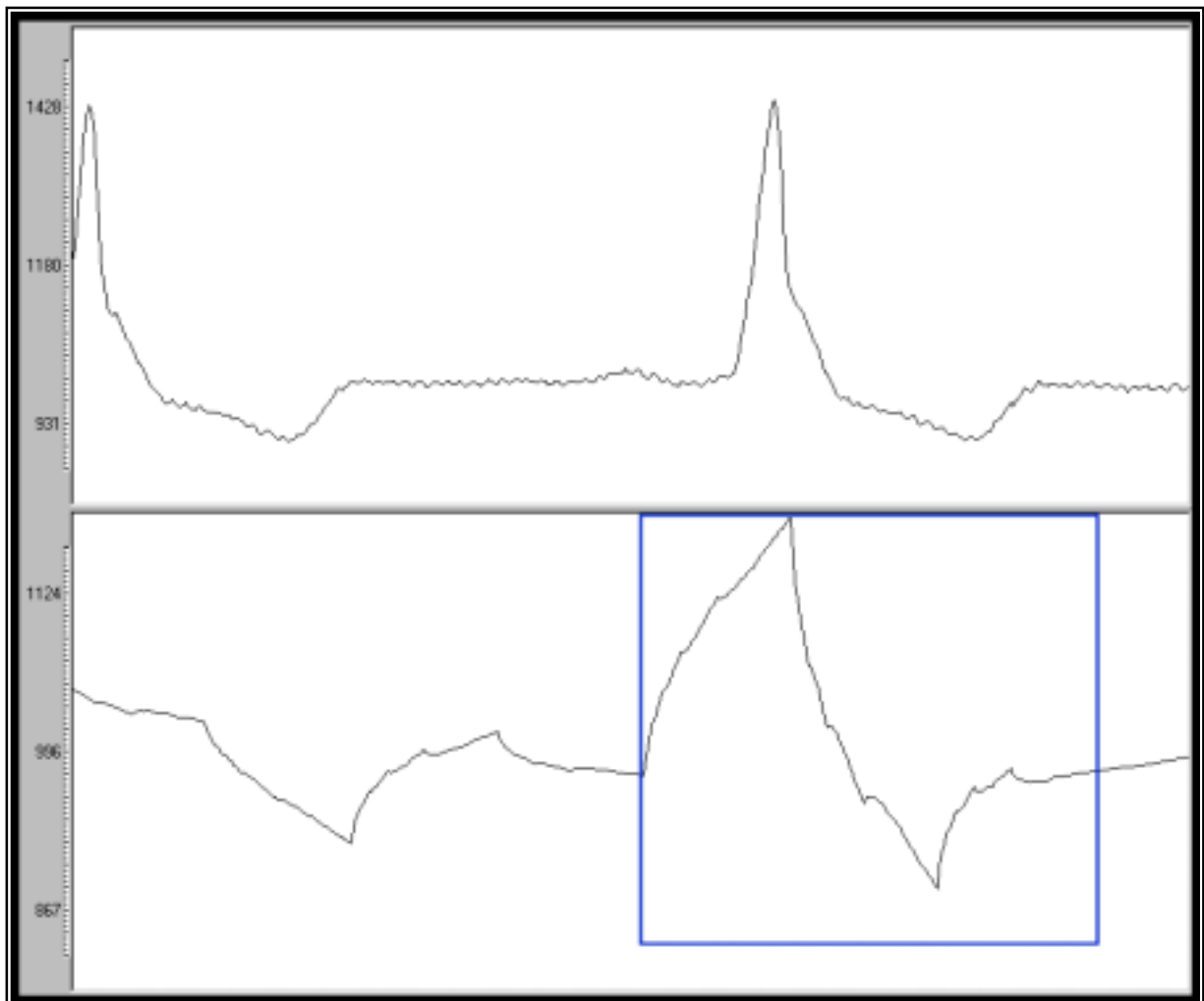
Rys.6.3.4c Rekonstrukcja współczynników szóstego poziomu z progiem kompresji $160\mu\text{V}$.

Skoro wiadomo, co reprezentują sygnały zrekonstruowanych detali, to pozostaje jeszcze pytanie: czym są przebiegi aproksymacji. W analizie wielorozdzielczej, na której oparta jest transformata falkowa kluczową rolę odgrywa tak zwana „funkcja skalująca” ściśle związana z falką danego typu. Jest ona odpowiedzialna za transformacje sygnału na różne poziomy rozdzielczości. Sama transformacja za pomocą funkcji skalującej jest stratnym przekształceniem, ponieważ wygładza sygnał. Jest ona odpowiednikiem dolnoprzepustowej filtracji sygnału. Wiadomo jednak, że transformata falkowa jest bezstratna. Tak jest, ponieważ w przekształceniu wykorzystuje zarówno bazę funkcji skalujących jak również bazę falek, które zachowują informację o szczegółach odrzuconą przez skalowanie. Zatem należałoby

przypuszczać, że sygnały aproksymacji składają się ze złożenia wielu funkcji skalujących. Rysunek 6.3.5 pokazuje, że tak w istocie jest. Generalnie trudno jest pokazać funkcję skalującą w sygnale. W sygnale aproksymacji nie można normalnie usuwać współczynników, więc aby zobaczyć niezniekształcony obraz trzeba znaleźć jeden duży współczynnik otoczony wieloma o bardzo małej amplitudzie. Zaznaczona funkcja skalująca może mylić się z falką, jednak sygnały te mają jedną charakterystyczną rzecz odróżniającą je w sygnale. W przypadku 'Daubechies 2' amplituda falki jest w przybliżeniu jednakowa zarówno po ujemnej, jak i dodatniej stronie, natomiast amplituda funkcji skalującej po stronie dodatniej jest około cztery razy większa, niż po ujemnej. Biorąc pod uwagę fakt, że wyodrębniony sygnał na rysunku jest trochę zniekształcony przez sąsiednie współczynniki nie ma wątpliwości, iż przedstawia funkcję skalującą.



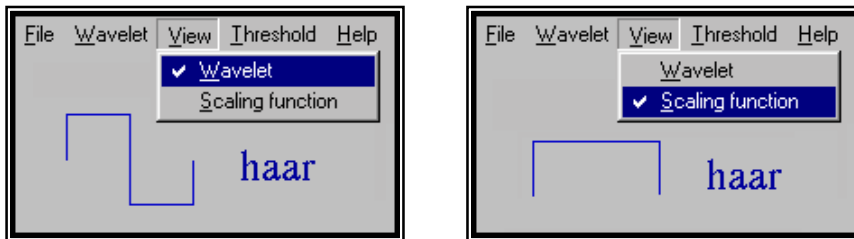
Rys.6.3.5a Z lewej falka 'db2', z prawej funkcja skalująca 'db2'



Rys. 6.3.5b U góry sygnał oryginalny 'Lbb2.d'. U dołu aproksymacja szóstego poziomu. Z sygnału wyodrębniona funkcja skalująca.

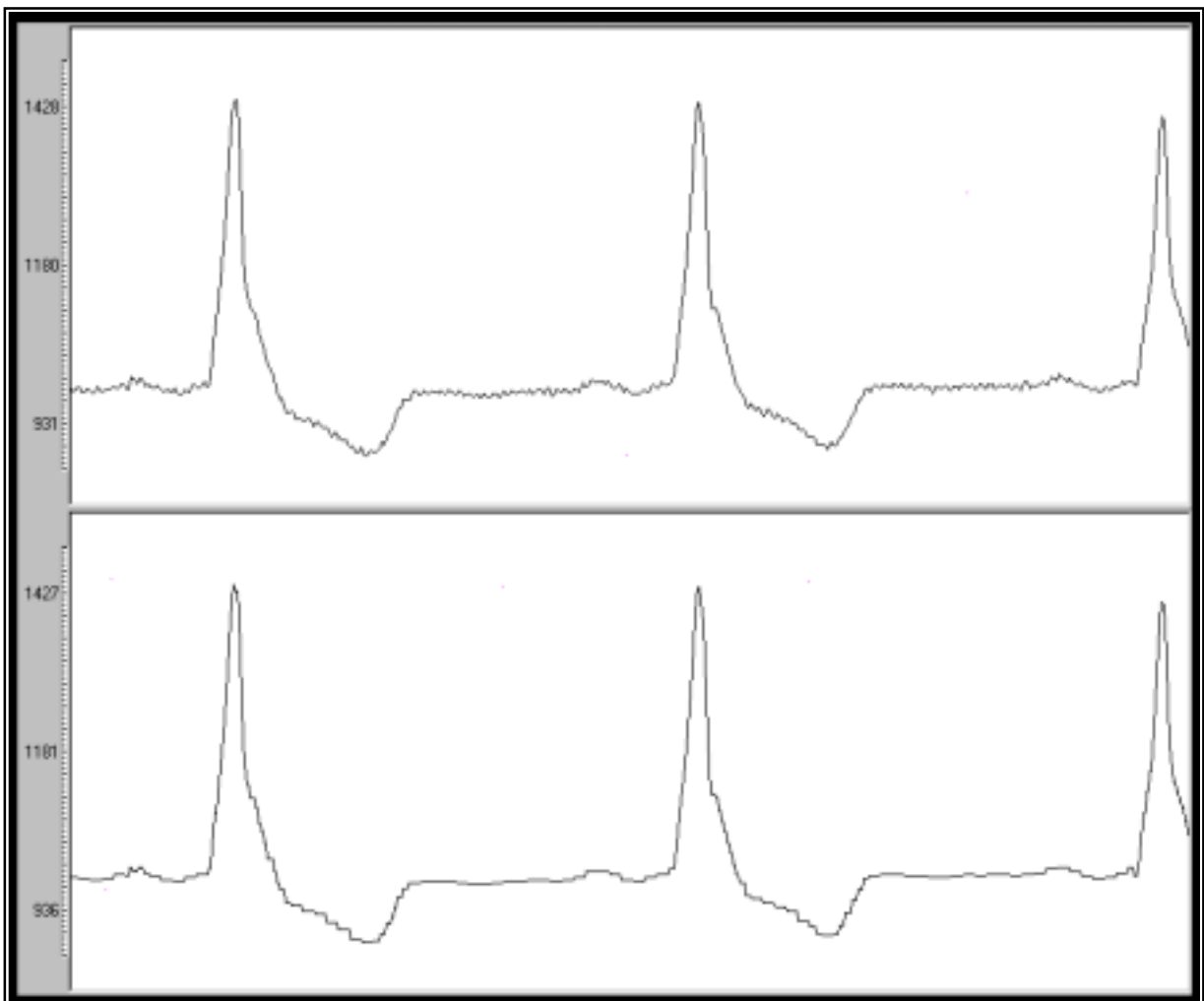
Mając praktyczną wiedzę i wyobrażenie, czym jest ciąg współczynników falkowych zdekomponowanego sygnału, oraz co reprezentują sobą sygnały rekonstrukcji aproksymacji i detali warto zastanowić się nad konsekwencjami doboru konkretnej falki do kompresji sygnału.

Najprostszą, w zasadzie trywialną i nadającą się do każdego zastosowania jest falka „Haar’a” (jednocześnie „Daubechies 1”). Ma ona kształt prostokątny, dlatego też nawet intuicyjnie można za jej pomocą opisać każdy przebieg (rys.6.3.6).

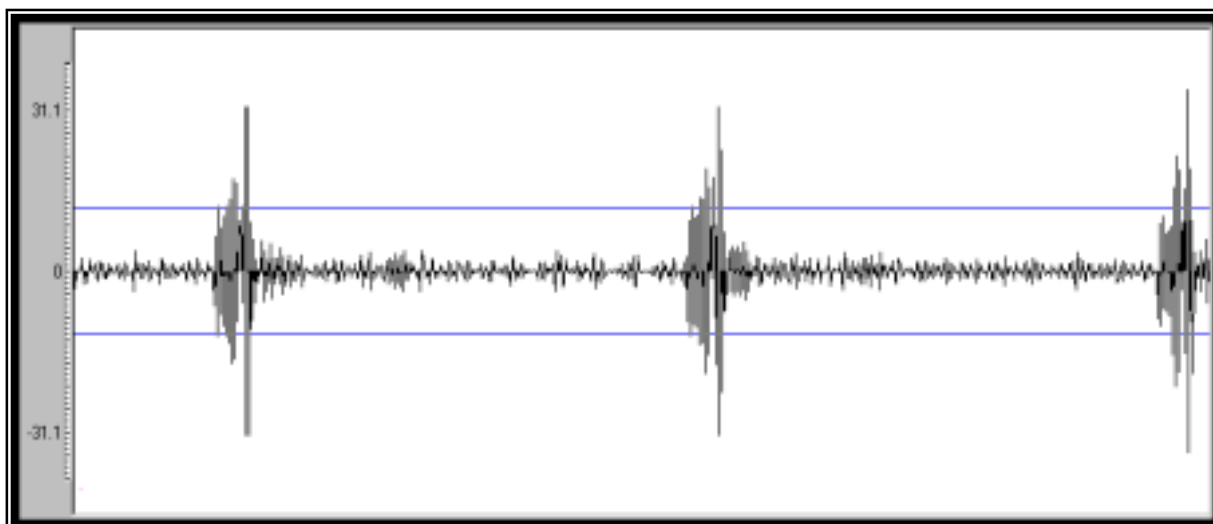


Rys.6.3.6a Falka i funkcja skalująca ‘Haar’

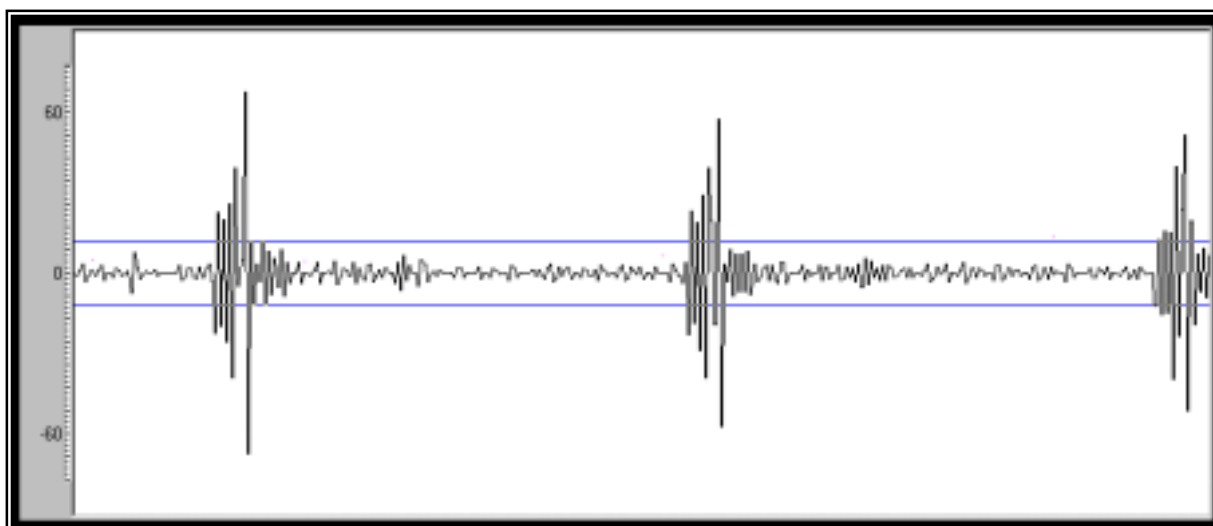
Do doświadczenia ponownie wykorzystano sygnał ‘Lbb2.d’. Ustawiono trzy poziomy dekompozycji, rozszerzenie typu ‘Limit padding’. Sygnał rozciągnięto na trzy widoczne QRS-y.



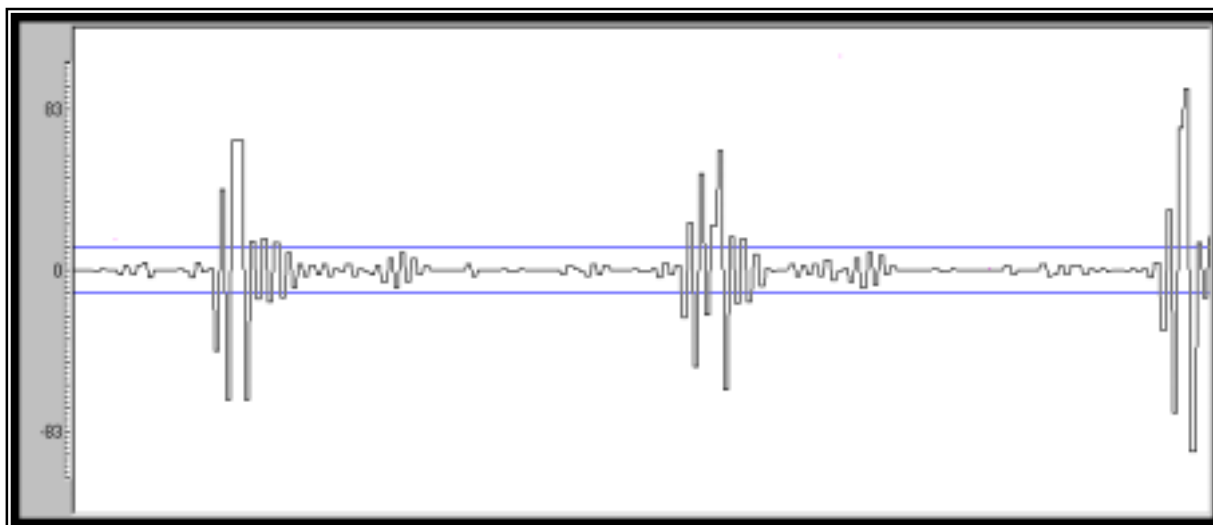
Rys.6.3.6b Sygnał oryginalny ‘Lbb2.d’ (u góry) oraz rekonstrukcja z progiem 12,28 μ V.



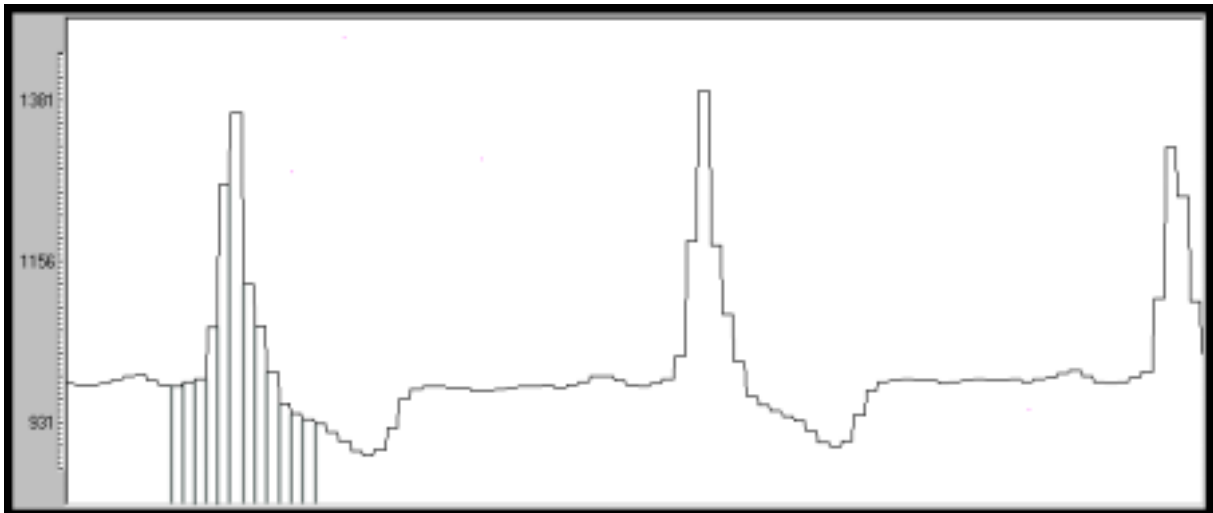
Rys.6.3.6c Rekonstrukcja pierwszego poziomu detali sygnału 'Lbb2.d'. Zaznaczony próg 12,28 μ V.



Rys.6.3.6d Rekonstrukcja drugiego poziomu detali sygnału 'Lbb2.d'.



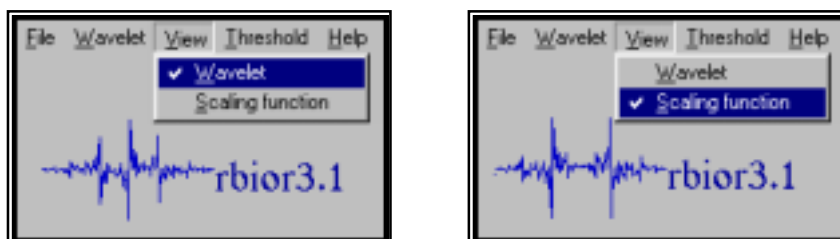
Rys.6.3.6e Rekonstrukcja trzeciego poziomu detali sygnału 'Lbb2.d'.



Rys.6.3.6f Rekonstrukcja trzeciego poziomu aproksymacji sygnału 'Lbb2.d'. Na pierwszym QRS zaznaczono linie odgradzające poszczególne funkcje skalujące.

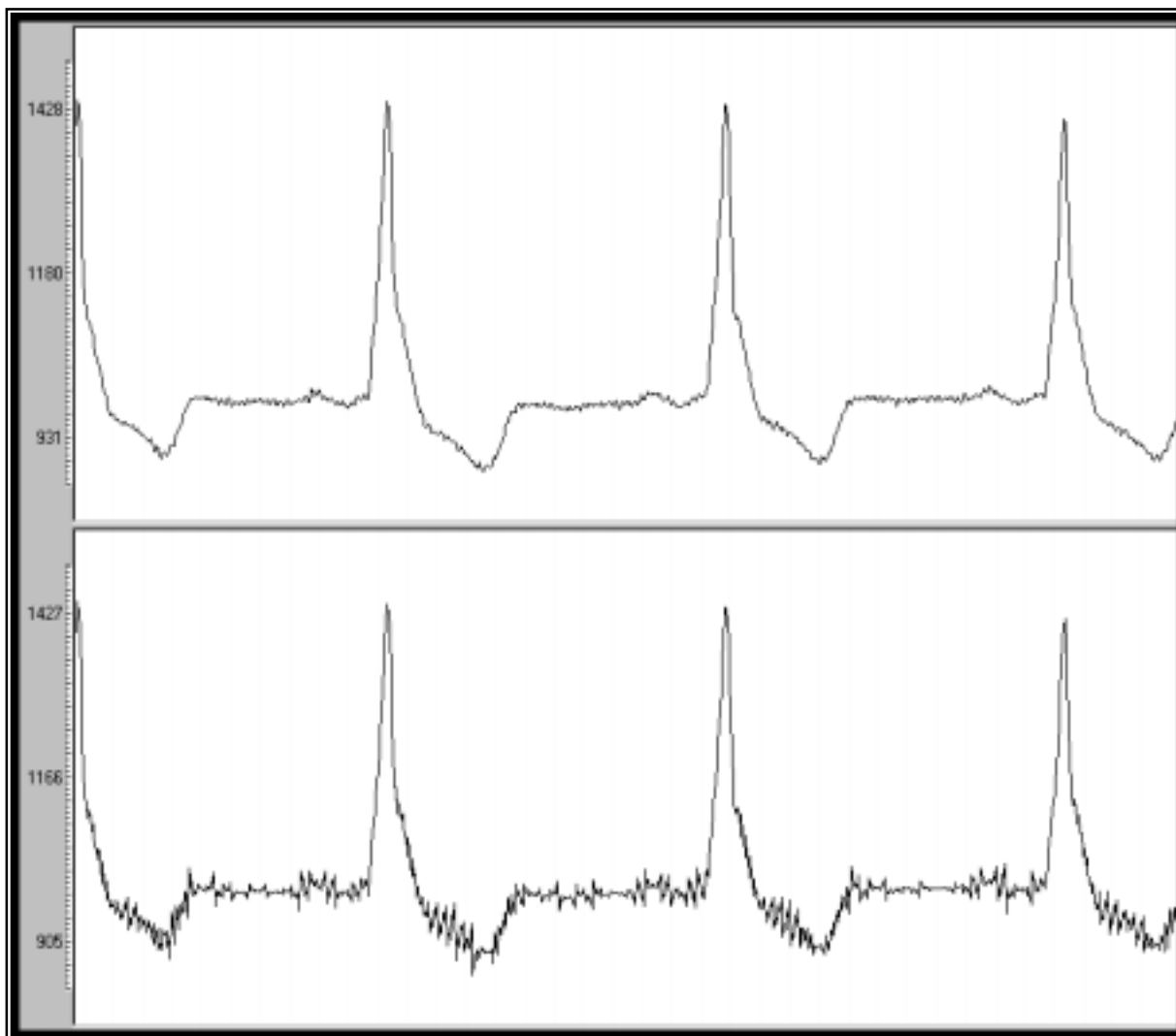
Na zamieszczonych rysunkach rekonstrukcji detali (rys.6.3.6c-d-e) widać złożenie falek Haar'a o różnych amplitudach. Szczególnie widoczne to jest na poziomie trzecim reprezentującym niższe częstotliwości, na którym więcej punktów wchodzi w skład pojedynczej falki. Aproksymacja (rys.6.3.6f) składa się ze „słupów” poukładanych jeden przy drugim, którymi de facto są funkcje skalujące. Zostało to dodatkowo zaznaczone za pomocą pionowych linii na pierwszym QRS. Zastosowanie domyślnego progu równego $12,28\mu\text{V}$ jednakowego dla wszystkich poziomów zaowocowało kompresją równą 77,83% wyciętych współczynników. Spowodowało to jednak dosyć duże straty w sygnale $\text{PRD}=3,271\%$. Widać na rysunku 6.3.6b, że na wykresie rekonstrukcji pojawiły się schodki, szczególnie wyraźne na załawkach P i T, ponieważ w tych obszarach amplituda współczynników detali była na tyle mała, że zostały one usunięte. Sumując, rozkład sygnału na tego rodzaju sygnały bazowe, jakie tworzy falka Haar'a jest intuicyjne i co również jest dosyć ważne nakład obliczeniowy jest niski, ponieważ związane z falką filtry mają prostą postać analityczną (są to dwuliczbowe wektory o wartościach 0 lub 1). Niestety, jakość tej kompresji jest raczej niezadowolająca. Wysoka wartość wskaźnika PRD jest rezultatem małej korelacji falki z sygnałem. Można było się zresztą domyśleć, że prostokątne kształty falek nie będą efektywnie aproksymowały sygnału.

Następny przykład prezentuje przypadek niedopasowania falki z sygnałem. Falka wykorzystana w doświadczeniu to 'Reverse biorthogonal 3.1' (rys.6.3.7a), sygnał wciąż pozostaje ten sam – 'Lbb2.d'. Poziom dekompozycji - czwarty.

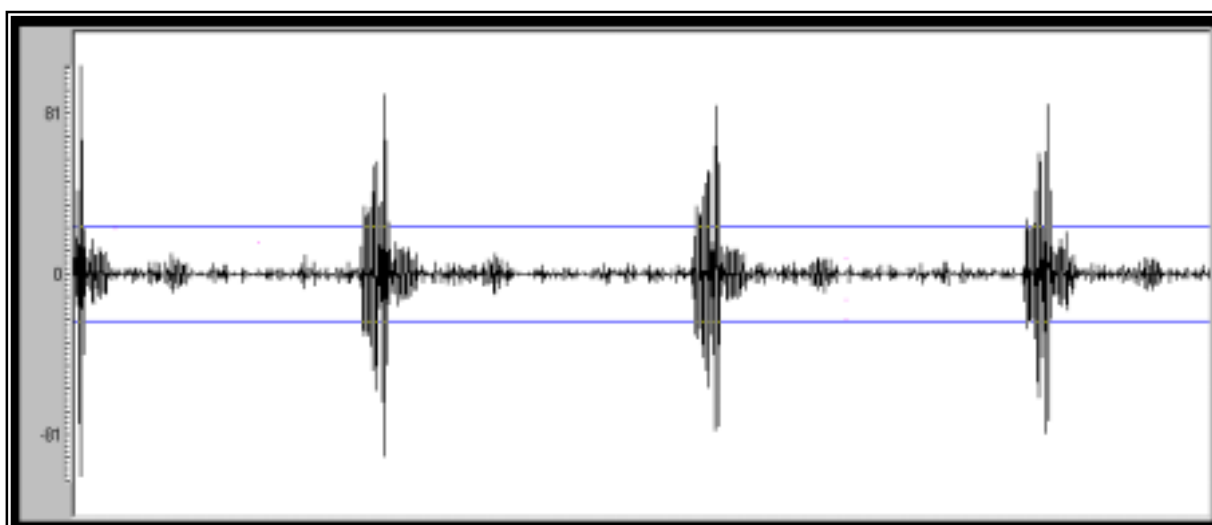


Rys.6.3.7a Falka i funkcja skalująca 'Reverse biorthogonal 3.1'

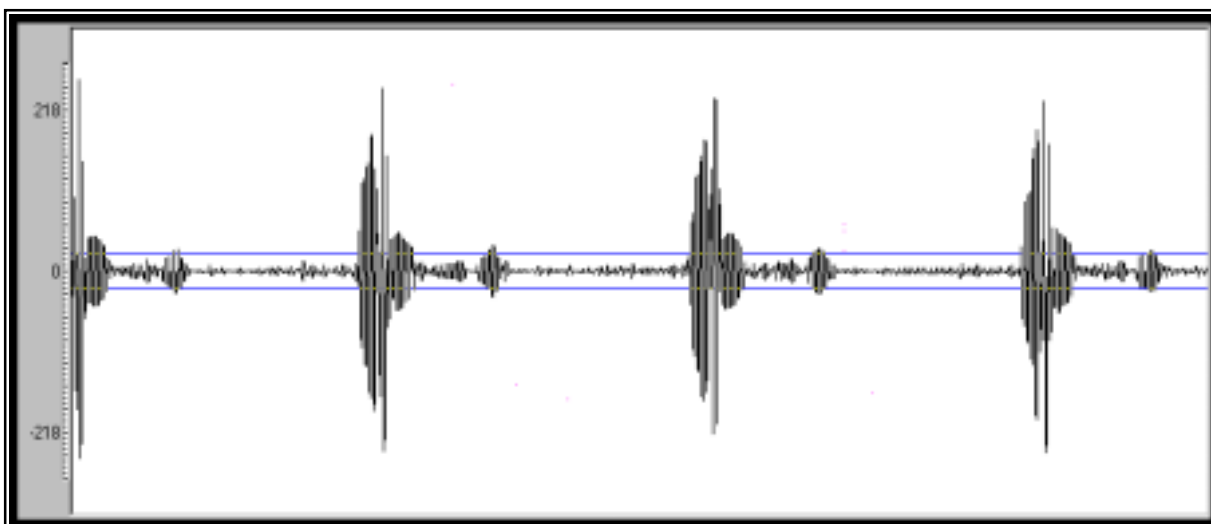
Kolejne rekonstrukcje sygnału zamieszczono na rysunkach 6.3.7b-g.



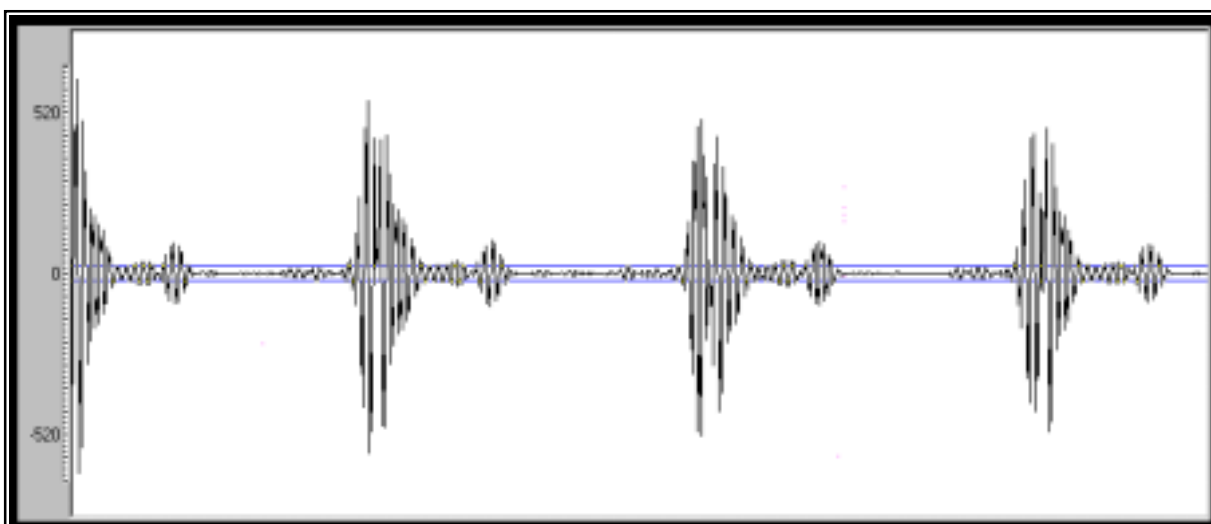
Rys.6.3.7b Sygnał oryginalny 'Lbb2.d' (u góry) oraz rekonstrukcja z progiem $24,56\mu\text{V}$ (falka 'rbior3.1').



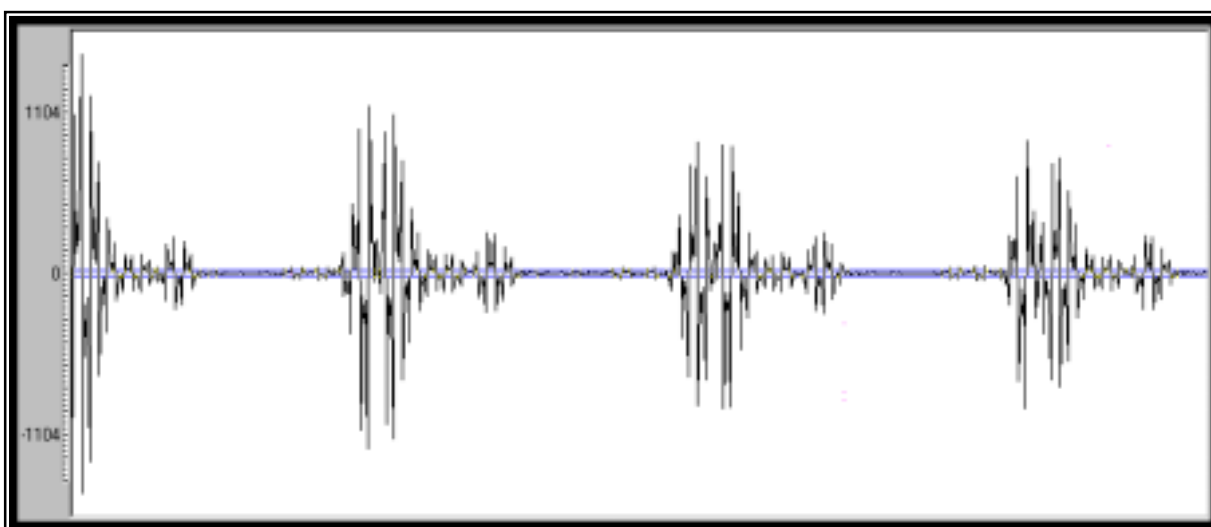
Rys.6.3.7c Rekonstrukcja pierwszego poziomu detali sygnału 'Lbb2.d' przy użyciu falki 'rbior3.1'. Zaznaczony próg $24,56\mu\text{V}$.



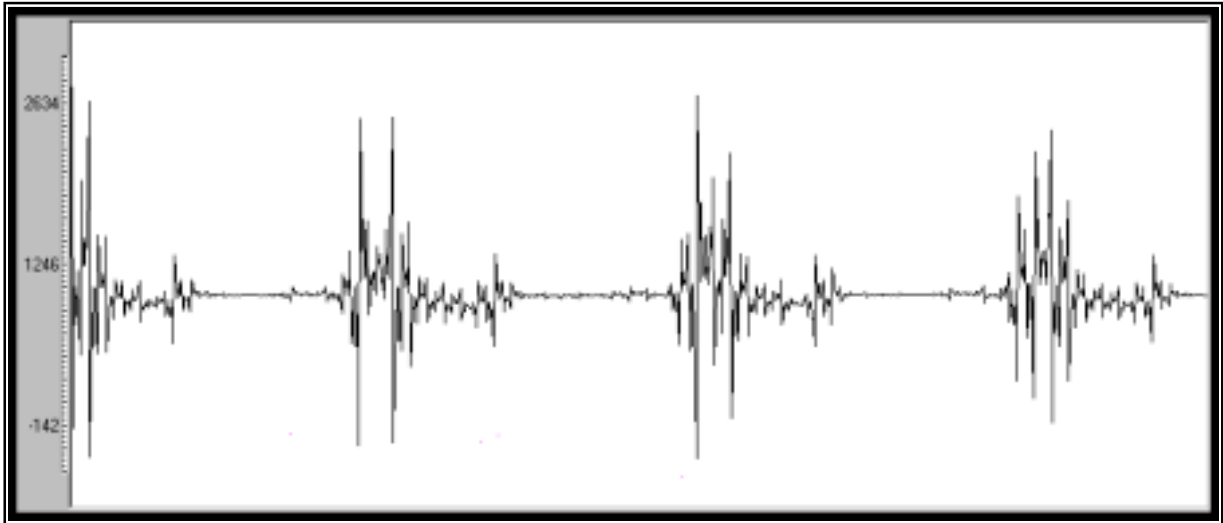
Rys.6.3.7d Rekonstrukcja drugiego poziomu detali sygnału 'Lbb2.d' przy użyciu falki 'rbior3.1'



Rys.6.3.7e Rekonstrukcja trzeciego poziomu detali sygnału 'Lbb2.d' przy użyciu falki 'rbior3.1'



Rys.6.3.7f Rekonstrukcja czwartego poziomu detali sygnału 'Lbb2.d' przy użyciu falki 'rbior3.1'



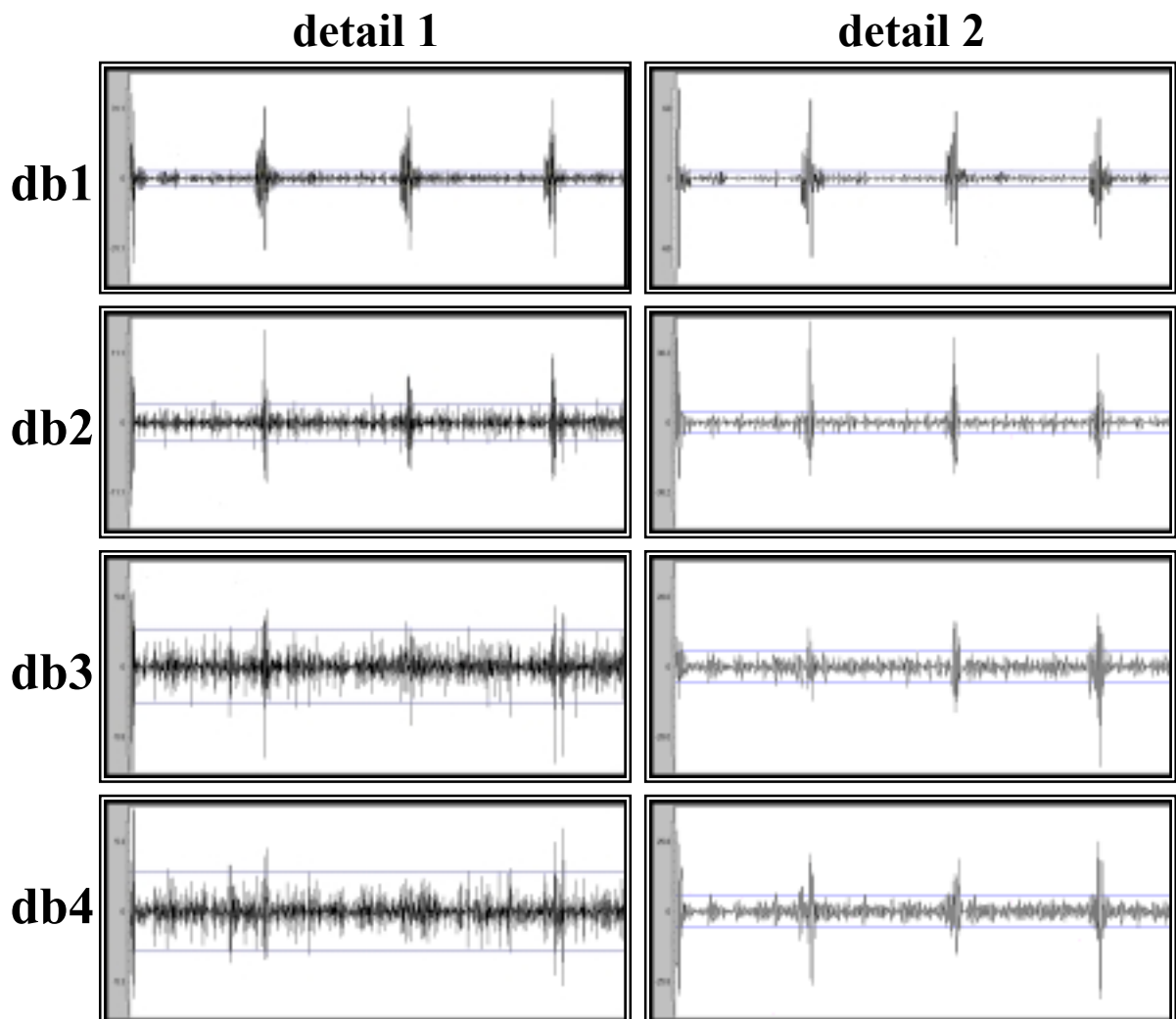
Rys.6.3.7g Rekonstrukcja czwartego poziomu aproksymacji sygnału 'Lbb2.d' przy użyciu falki 'rbior3.1'.

Na podstawie analizy przebiegów widać, że charakter falki znacznie różni się od sygnału. Jej kształt ma postać sygnału zróżniczkowanego i prawdopodobnie byłaby dobrym narzędziem kompresji dla sygnałów o takim właśnie charakterze. Jednak pomimo jej „nietypowego” kształtu należy zaznaczyć, że jak wszystkie falki tworzy ona ortonormalną bazę w przestrzeni, za pomocą której można opisać każdy przebieg. Również w przypadku tego sygnału dekompozycja do przestrzeni współczynników (skala-czas), a następnie powrotna rekonstrukcja odbyła się bez straty jakości. Przebiegu tej rekonstrukcji nie zamieszczono, ponieważ ma ona kształt identyczny z oryginalnym. Opis naszego sygnału nie jest efektywny z punktu widzenia rozkładu energii pomiędzy współczynniki falkowe. Z powodu małego podobieństwa falki z sygnałem praktycznie żaden z sygnałów bazowych nie mógł samodzielnie reprezentować większej części sygnału. Każdy fragment sygnału opisywany jest przez wiele współczynników o zbliżonej wadze (amplitudzie). Sprawia to, że praktycznie każdy współczynnik w zapisie falkowym jest ważny i odjęcie części z nich (istota kompresji) spowoduje duże straty w jakości sygnału. Taką sytuację pokazuje rysunek 6.3.7b. Cały sygnał z wyjątkiem QRS-ów jest silnie zniekształcony, ponieważ w tych właśnie miejscach współczynniki nie przekraczały wartości progu i zostały wyzerowane. Wartość progu jest ona dosyć duża i wynosi $24,56\mu\text{V}$. Została ona jednak dobrana na podstawie amplitudy współczynników pierwszego poziomu, które dochodzi nawet do $80\mu\text{V}$. Warto zwrócić uwagę na bardzo dużą amplitudę przebiegów kolejnych poziomów. Detale poziomu drugiego - $\pm 218\mu\text{V}$, poziomu trzeciego - $\pm 520\mu\text{V}$ i czwartego $\pm 1104\mu\text{V}$. Są to bardzo duże wartości biorąc pod uwagę, że sygnał oryginalny nie przekracza $1430\mu\text{V}$. Najbardziej jednak „niezwykłym” przebiegiem jest sygnał aproksymacji poziomu czwartego. W ogóle nie przypomina on wyglądem sygnału oryginalnego i ma amplitudę – bagatela $(2634+142)\mu\text{V}!!!$, co jest wartością prawie dwukrotnie większą od oryginału, a biorąc pod uwagę odległości maksimum od minimum stosunek ten wzrasta do 5,5 razy. Widać, że sam kształt funkcji skalującej jest dość nietypowy. Jej amplituda jest taka sama zarówno od strony ujemnej, jak i dodatniej. Na przebiegu aproksymacji można bez trudu wyodrębnić pojedyncze funkcje skalujące.

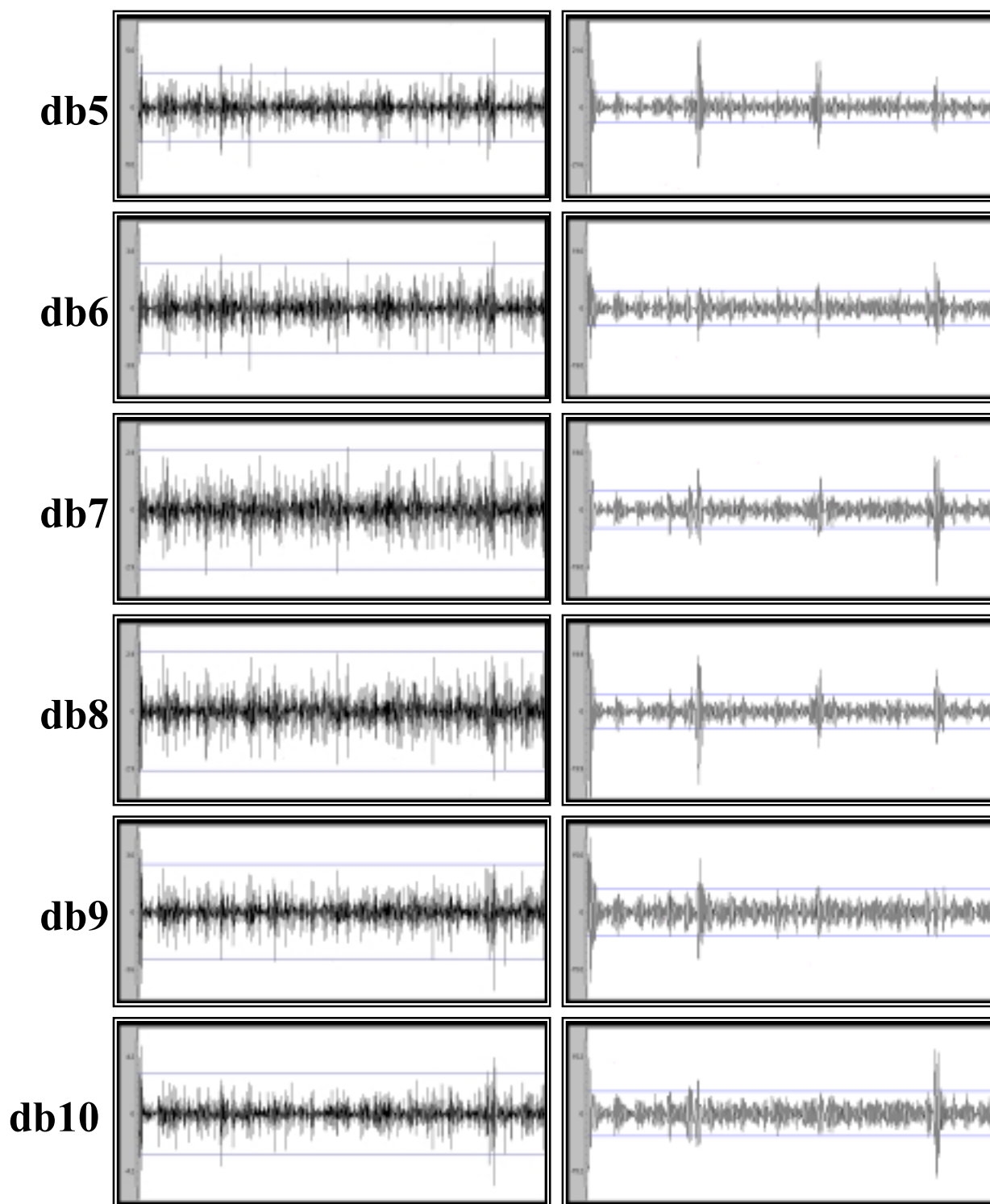
Rezultaty kompresji przy zastosowanym progu są następujące: 92,68% wyzerowanych współczynników pierwszego poziomu detali, 82,1% z drugiego poziomu, 68,6% z trzeciego oraz 43,08% z czwartego. W sumie efektywność kompresji osiągnęła 78,03%, czyli podobnie jak w poprzednim doświadczeniu. Jednak bardzo duża wartość PRD=12,415% potwierdza zbyt duży spadek jakości sygnału. Jest to czterokrotnie więcej w stosunku do poprzedniego doświadczenia. Aby otrzymać zadowalającą jakość sygnału wymagane było zmniejszenie progu do wartości $6\mu\text{V}$. Dało to spadek PRD do wartości 2,795% i pociągnęło również obniżenie stopnia kompresji do 50,02% wyzerowanych współczynników.

6.4. Wpływ długości falki na jakość kompresji

Niniejsze ćwiczenie skupia się nad zagadnieniem doboru długości falki. Próbuje określić, jaki wpływ ma to na jakość kompresji. W doświadczeniu założono, że ogólna jakość kompresji zależy od jakości współczynników pierwszego i drugiego poziomu detali. Można przyjąć takie założenie, ponieważ współczynniki te stanowią 75% ogółu wszystkich współczynników (50% - pierwszy i 25% - drugi). Do ćwiczenia wykorzystano przebieg 'Lbb2.d', dwa poziomy dekompozycji i rozszerzenie 'Limit padding'. Przeprowadzono dekompozycję i rekonstrukcję kolejno dla wszystkich falek rodziny 'Daubechies' od 'db1' do 'db10'. Przebiegi rekonstrukcji współczynników detali pierwszego i drugiego poziomu dla każdej falki przedstawiono na rysunku 6.3.8. Dodatkowo dla łatwiejszego określenia amplitudy sygnałów włączono linie progów, które dla wszystkich pierwszych poziomów wskazują wartość $3\mu\text{V}$, natomiast dla drugich $6\mu\text{V}$.



Rys. 6.4.1a Przebiegi detali pierwszego i drugiego poziomu dla różnych długości falek grupy 'Daubechies'



Rys.6.4.1b Przebiegi detali pierwszego i drugiego poziomu dla różnych długości falek grupy 'Daubechies'

Przedstawione przebiegi mają jedną wspólną cechę. Łączy je jednakowa amplituda poza obszarem QRS niezależnie od długości falek. Dla pierwszego poziomu można przyjąć, że jest to $2\text{-}3\mu\text{V}$, a dla drugiego $5\text{-}6\mu\text{V}$. Wynika z tego, że w omawianym obszarze długość falek nie ma większego wpływu na kompresję. Zupełnie inna sprawa jest w zakresie załamków QRS. Dla krótkich falek 'db1' i 'db2' amplituda QRS-ów kilkukrotnie przewyższa amplitudę pozostałej części sygnału zarówno na pierwszym i drugim poziomie. Z powodu swej wysokiej

amplitudy współczynniki te odgrywają istotną rolę w sygnale i nie mogą zostać wycięte w procesie kompresji, co znacznie obniża jej skuteczność. Przy wzroście długości falki ta różnica amplitud zaczyna się zacierać. Właściwie już od 'db4', czy 'db5' w sygnale rekonstrukcji poziomu pierwszego trudno jest wyodrębnić obszary QRS. Podobnie też dla drugiego poziomu począwszy od 'db5' stosunek rozpatrywanych amplitud zaczyna się stabilizować. Wynika z tego, że dla krótszych falek opis załamków QRS wymaga wykorzystania również składowych wysokoczęstotliwościowych podczas, gdy kształt dłuższych falek pozwala na pełną analizę za pomocą tylko niższych częstotliwości. W skrócie mówiąc jedna dłuższa falka z poziomu wyższego potrafi opisać taki sam fragment informacji z załamka QRS jak kilka krótszych falek z wyższego poziomu. Na podstawie tych spostrzeżeń należałoby się spodziewać, że jakość kompresji będzie najniższa dla falek 'db1' i 'db2', natomiast w przypadku dłuższych falek jakość będzie porównywalna. Tak też faktycznie jest. Tabela 6.3.8 pokazuje wyniki kompresji sygnału 'Lbb2.d' dla progu $3\mu\text{V}$ dla poziomu pierwszego oraz $6\mu\text{V}$ dla drugiego (globalny próg wyłączony) dla kolejnych falek 'Daubechies'. W tabeli obok falek zamieszczono ich długość, czyli długość wektorów reprezentujących odpowiednie filtry. Poza tym jest liczba określająca procent wyzerowanych współczynników odpowiednio: dla poziomu pierwszego, drugiego i ogółu wszystkich współczynników, a także wskaźnik PRD.

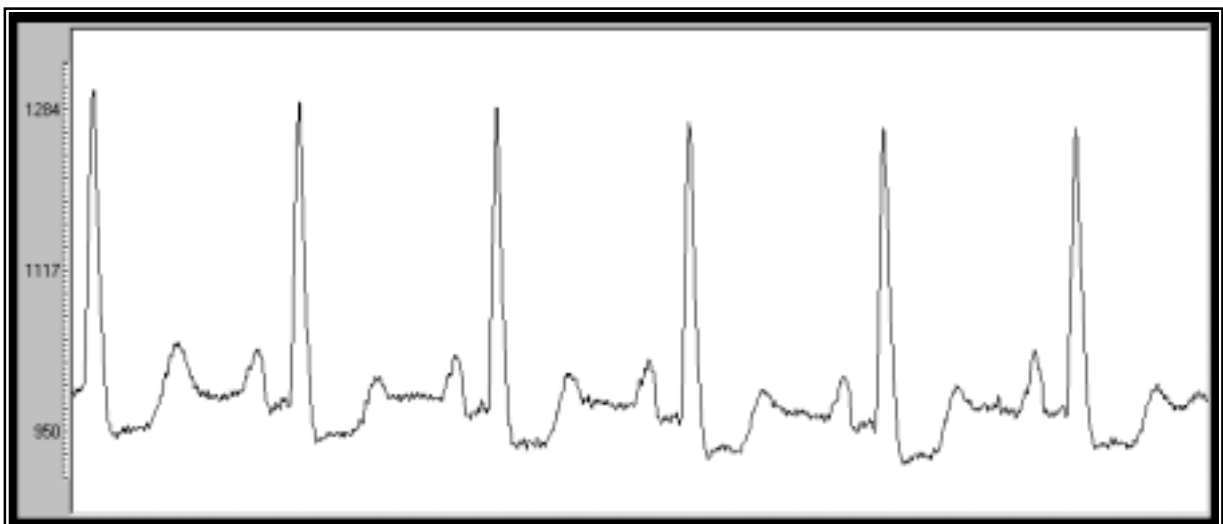
Sygnal Lbb2.d					
poziom dekompozycji: 'two levels'					
próg kompresji:					
- poziom pierwszy $3\mu\text{V}$					
- poziom drugi $6\mu\text{V}$					
rodzaj wypełnienia: 'Limit padding'					
Rodzaj falki	Długość falki	Liczba zer: poziom 1	Liczba zer: poziom 2	Liczba zer: ogółem	PRD
db1	2	73,63%	70,90%	54,54%	1,651%
db2	4	87,32%	79,18%	63,42%	1,619%
db3	6	91,42%	80,39%	65,76%	1,732%
db4	8	93,67%	76,79%	65,94%	1,669%
db5	10	99,42%	75,48%	68,46%	1,716%
db6	12	95,63%	77,12%	66,94%	1,680%
db7	14	96,21%	76,20%	66,99%	1,687%
db8	16	96,31%	77,25%	67,26%	1,669%
db9	18	96,71%	76,72%	67,31%	1,676%
db10	20	96,71%	75,29%	66,91%	1,656%

Tab.6.4.2 Rezultaty kompresji sygnału 'Lbb2.d' przy zastosowaniu różnych długości falek

Generalnie falki od 'db4' do 'db10' uzyskały dosyć wysoką i stabilną jakość kompresji. Liczba zerowych współczynników waha się w granicach 66-68%, zaś PRD ok. 1,66%. Dla 'db5' PRD jest trochę wyższe (1,732%), ale dlatego, że miało najwięcej zer (68,46%). Zdecydowanie najgorszą falką okazała się najkrótsza i najprostsza falka 'db1' (czyli falka Haar'a opisywana w punkcie 6.3.6) ze stopniem kompresji 54,54%. Falka 'db2' osiągnęła większą wartość 63,42%, natomiast 'db3' pod względem kompresji wypadła nie gorzej od „dłuższych” falek, jednak dość dobrą kompresję osiągnęła kosztem najwyższej wartości PRD (1,732%). Reasumując, na podstawie przeprowadzonych obserwacji i pomiarów wynika, że do kompresji falkowej badanego sygnału 'Lbb2.d' lepsze rezultaty osiąga się stosując dłuższe falki (o wyższych numerach).

6.5. Znalezienie optymalnej falki

Doświadczenie ma na celu wybranie falki, dla której kompresja będzie najskuteczniejsza. Kryterium wyboru jest stosunek liczby zer do wskaźnika PRD. Doświadczenie zostało wykonane dla różnych opcji dekompozycji: dwupoziomowej, trypoziomowej i czteropoziomowej. Dla każdej z nich został ustalony wskaźnik PRD, dla którego szukano maksymalnej odsetki wyzerowanych współczynników tak, aby PRD nie został przekroczony. Wyniki zamieszczono w tabeli 6.5.2, zaś sygnał 'N17.d' wykorzystany w doświadczeniu zaprezentowano na rys.6.5.1.

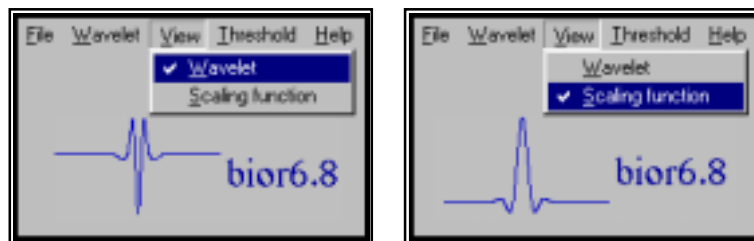


Rys.6.5.1 Sygnał oryginalny 'N17.d'

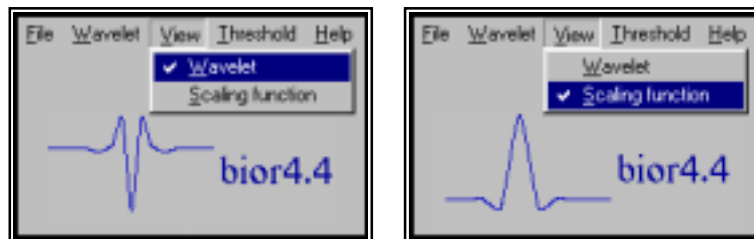
<p align="center">Sygnal N17.d wypełnienie: 'Limit padding' globalny próg</p>								
dekompozycja: dwa poziomy PRD = 1,500%			dekompozycja: trzy poziomy PRD=1,500%			dekompozycja: cztery poziomy PRD=1,500%		
Falka	Próg	Zera	Falka	Próg	Zera	Falka	Próg	Zera
haar	4,25	59,28%	haar	4,00	62,60%	haar	4,00	63,96%
db2	4,25	66,59%	db2	3,94	71,29%	db2	3,81	72,10%
db4	5,00	70,31%	db4	4,25	75,92%	db4	4,19	76,86%
db6	5,19	70,71%	db6	4,56	76,62%	db6	4,31	77,86%
db8	5,00	70,34%	db8	4,38	75,57%	db8	4,31	76,55%
db10	5,31	70,94%	db10	4,56	76,51%	db10	4,50	77,18%
coif1	4,94	68,19%	coif1	4,25	72,97%	coif1	4,19	74,04%
coif3	5,31	70,63%	coif3	4,56	77,00%	coif3	4,44	77,94%
coif5	5,31	70,67%	coif5	4,63	77,12%	coif5	4,50	77,98%
sym2	4,25	66,59%	sym2	3,94	71,29%	sym2	3,81	72,10%
sym4	5,06	70,11%	sym4	4,38	76,31%	sym4	4,25	77,53%
sym6	4,94	70,47%	sym6	4,44	76,29%	sym6	4,25	77,37%
sym8	5,25	70,78%	sym8	4,69	76,72%	sym8	4,50	78,12%
bior1.3	4,50	58,32%	bior1.3	4,00	61,67%	bior1.3	4,00	62,91%
bior1.5	4,25	58,87%	bior1.5	4,00	62,79%	bior1.5	4,00	64,25%
bior2.2	4,81	69,21%	bior2.2	4,44	74,58%	bior2.2	4,25	76,22%
bior2.6	5,19	69,50%	bior2.6	4,63	75,01%	bior2.6	4,56	76,55%
bior2.8	4,81	69,13%	bior2.8	4,44	74,28%	bior2.8	4,25	75,99%
bior3.1	4,13	68,19%	bior3.1	3,81	71,48%	bior3.1	3,69	72,19%
bior3.5	5,31	70,47%	bior3.5	4,94	76,00%	bior3.5	4,75	77,37%
bior3.7	4,88	69,48%	bior3.7	4,44	74,52%	bior3.7	4,38	75,79%
bior3.9	5,69	70,74%	bior3.9	5,00	76,08%	bior3.9	4,94	77,56%
bior4.4	5,13	70,83%	bior4.4	4,25	77,22%	bior4.4	4,13	78,52%
bior5.5	4,75	70,08%	bior5.5	3,88	75,52%	bior5.5	3,81	77,08%
bior6.8	5,19	70,96%	bior6.8	4,44	77,62%	bior6.8	4,31	78,88%
rbior1.3	4,69	69,36%	rbior1.3	3,94	74,62%	rbior1.3	3,94	76,03%
rbior1.5	4,63	69,77%	rbior1.5	4,06	75,68%	rbior1.5	3,94	76,84%
rbior2.2	4,75	64,30%	rbior2.2	4,00	66,96%	rbior2.2	3,88	66,92%
rbior2.6	5,13	69,45%	rbior2.6	4,31	74,72%	rbior2.6	4,13	76,07%
rbior2.8	4,81	69,71%	rbior2.8	4,13	75,05%	rbior2.8	3,94	75,85%
rbior3.1	4,50	40,82%	rbior3.1	4,00	40,97%	rbior3.1	3,50	36,85%
rbior3.5	4,50	65,06%	rbior3.5	3,69	66,38%	rbior3.5	3,56	66,32%
rbior3.7	4,44	65,67%	rbior3.7	3,56	67,59%	rbior3.7	3,44	67,39%
rbior3.9	4,75	67,29%	rbior3.9	3,75	70,19%	rbior3.9	3,63	70,49%
rbior4.4	5,01	69,57%	rbior4.4	4,31	74,90%	rbior4.4	4,19	75,78%
rbior5.5	4,56	58,54%	rbior5.5	4,13	73,64%	rbior5.5	4,06	74,40%
rbior6.8	5,13	70,38%	rbior6.8	4,38	76,57%	rbior6.8	4,25	77,51%

Tab.6.5.2 Parametry kompresji sygnału 'N17.d' dla różnych falek i poziomów dekompozycji

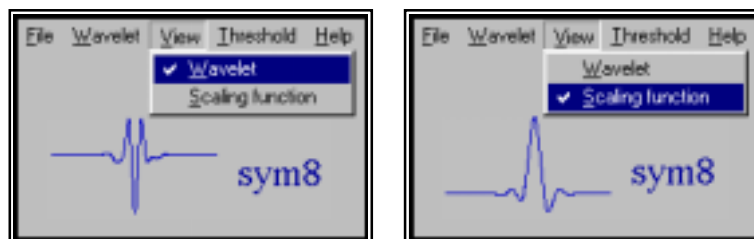
Na podstawie tabeli 6.5.2 wybrano faleki, które uzyskały najlepszą i najgorszą wydajność kompresji przy wskaźniku PRD równym 1,500%. Generalnie poziom dekompozycji nie miał znaczenia przy wyborze falek, ponieważ wraz ze wzrostem poziomu dekompozycji wzrastał stopień kompresji, ale praktycznie równomiernie dla wszystkich falek. Spośród wszystkich zamieszczonych w tabeli falek zdecydowana większość (ok.2/3 ogółu) uzyskała dobrą lub bardzo dobrą jakość kompresji pomiędzy 70 a 80 procent zerowych współczynników. Trzy najskuteczniejsze faleki to: 'bior6.8' – 78,88% (rys.6.5.3) 'bior4.4' – 78,52% (rys.6.5.4) i 'sym8' – 78,12% (rys.6.5.5). Zamieszczone tu wartości procentowe kompresji dotyczą dekompozycji cztero-poziomowej. Cechą wspólną tych falek jest znaczna długość oraz symetria (biortogonalne) lub prawie symetria (Symlets). Zdecydowanie najgorszą faleką okazała się 'rbior3.1' (rys.6.3.7a) z powodu charakteru bardzo odbiegającego od sygnału (opisywana w punkcie 6.3.3). Poza tym generalnie niższy stopień kompresji uzyskały faleki krótkie: 'haar' – 63,96% (rys.6.3.6a), 'bior1.3' – 62,91% (rys.6.5.6), 'bior1.5' – 64,25% (rys.6.5.7).



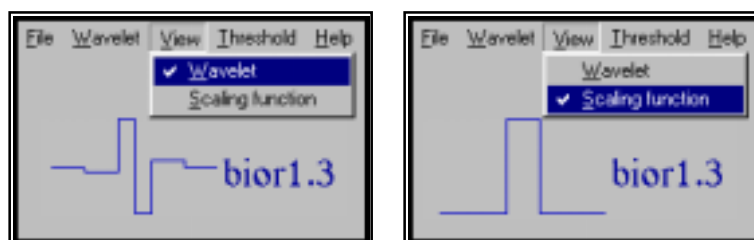
Rys.6.5.3 Faleka i funkcja skalująca 'Reverse biorthogonal 6.8' – dobra kompresja



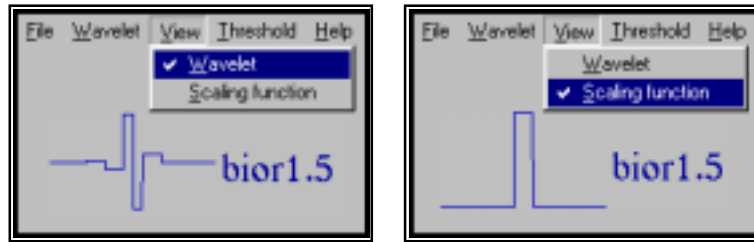
Rys.6.5.4 Faleka i funkcja skalująca 'Reverse biorthogonal 4.4' – dobra kompresja



Rys.6.5.5 Faleka i funkcja skalująca 'Symlets 8' – dobra kompresja



Rys.6.5.6 Faleka i funkcja skalująca 'Reverse biorthogonal 1.3' – słaba kompresja



Rys.6.5.7 Falka i funkcja skalująca 'Reverse biorthogonal 1.5' – słaba kompresja

6.6. Problem brzegowy

Dekompozycja falkowa jest w praktyce równoważna filtracji sygnału przez odpowiedni zestaw filtrów. Proces ten w dziedzinie czasu nierozdzielnie jest związany z operacją splotu, który wyraża się wzorem:

$$w(k) = \sum_j s(j) * f(k+1-j)$$

Gdzie:

w(x) – kolejne próbki sygnału wyjściowego

s(x) – kolejne próbki sygnału wejściowego

f(x) – kolejne próbki filtru

k, j = 1, 2, ...

Powyższy wzór jest użyteczny w środowiskach programistycznych indeksujących wektory od „jednego”, np. w Matlabie. Aplikacja ta została napisana w języku „C++”, który indeksację wektorów zaczyna od „zera”. Zatem powyższy wzór należy przekształcić do postaci:

$$w(k) = \sum_j s(j) * f(k-j)$$

Gdzie:

k, j = 0, 1, 2, ...

k – indeks próbek sygnału wyjściowego

j – indeks próbek sygnału wejściowego

Jeżeli „J” jest długością sygnału wejściowego i „I” długością filtru to długość sygnału wyjściowego wynosi:

$K=J+M-1$

zaś:

$k = 0, 1, 2, \dots, K-1$

$j = 0, 1, 2, \dots, J-1$

Przy tak zdefiniowanej operacji podczas obliczania początkowych i końcowych wartości pojawia się problem brakujących próbek sygnału wejściowego.

Jeżeli sygnał wejściowy ma postać:

s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Zaś filtr:

f_0	f_1	f_2
-------	-------	-------

To wartości kolejnych próbek wyjścia operacji splotu będą miały wartości:

$$\begin{aligned}
 w_0 &= f_0 s_0 \\
 w_1 &= f_0 s_1 + f_1 s_0 \\
 w_2 &= f_0 s_2 + f_1 s_1 + f_2 s_0 \\
 w_3 &= f_0 s_3 + f_1 s_2 + f_2 s_1 \\
 w_4 &= f_0 s_4 + f_1 s_3 + f_2 s_2 \\
 w_5 &= f_0 s_5 + f_1 s_4 + f_2 s_3 \\
 &\dots \text{itd}
 \end{aligned}$$

Widać, że wartości próbek w_0 i w_1 są zniekształcone, ponieważ nie zależą od wszystkich współczynników filtru. Z przodu sygnału brakuje dwóch próbek, które wypełniłyby brakującą informację. Podobna sytuacja istnieje również na końcu sygnału. Ilość zniekształconych współczynników jest uzależniona od długości filtru i jest równa długości filtru minus jeden z obu stron sygnału. Aby zminimalizować te zniekształcenia uzupełnia się brakujące próbki sygnału wejściowego, a następnie po operacji splotu usuwa się nadwyżkę próbek (dokładnie tyle, ile zostało dodanych) z sygnału wyjściowego. Schemat postępowania wygląda następująco:

Sygnał wejściowy:

s_{-2}	s_{-1}	s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
----------	----------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Filtr:

f_0	f_1	f_2
-------	-------	-------

Wartości kolejnych próbek wyjścia:

$$\begin{aligned}
 \cancel{w_{-2}} &= \cancel{f_0 s_{-2}} \\
 \cancel{w_{-1}} &= \cancel{f_0 s_{-1} + f_1 s_{-2}} \\
 w_0 &= f_0 s_0 + f_1 s_{-1} + f_2 s_{-2} \\
 w_1 &= f_0 s_1 + f_1 s_0 + f_2 s_{-1} \\
 w_2 &= f_0 s_2 + f_1 s_1 + f_2 s_0 \\
 w_3 &= f_0 s_3 + f_1 s_2 + f_2 s_1 \\
 w_4 &= f_0 s_4 + f_1 s_3 + f_2 s_2 \\
 w_5 &= f_0 s_5 + f_1 s_4 + f_2 s_3 \\
 &\dots \text{itd}
 \end{aligned}$$

Dla filtru o długości „I” należy dodać I-1 próbek na początku sygnału oraz I-1 na końcu.

Gdy już wiadomo, jak rozszerzyć sygnał pozostaje jeszcze kwestia doboru wartości dodatkowych próbek. W programie zastosowano pięć różnych kryteriów:

1. „Zero padding” – wypełnianie zerami. W praktyce jest to równoznaczne z tym, jakby w ogóle nie było realizowane rozszerzanie sygnału.

$$s_{-1} = 0$$

$$s_{-2} = 0$$

2. „Limit padding” – wypełnianie wartością brzegowej próbki

$$s_{-1} = s_0$$

$$s_{-2} = s_0$$

3. „Smooth padding of order 1” – wypełnianie według przyrostu między dwoma brzegowymi próbkami

jeśli $a = s_1 - s_0$

to:

$$s_{-1} = s_0 - a$$

$$s_{-2} = s_{-1} - a$$

4. „Symetric padding” – odbicie symetryczne sygnału

$$s_{-1} = s_0$$

$$s_{-2} = s_1$$

5. „Periodic padding” – założenie okresowości sygnału i pobranie wartości próbek z sąsiedniego okresu

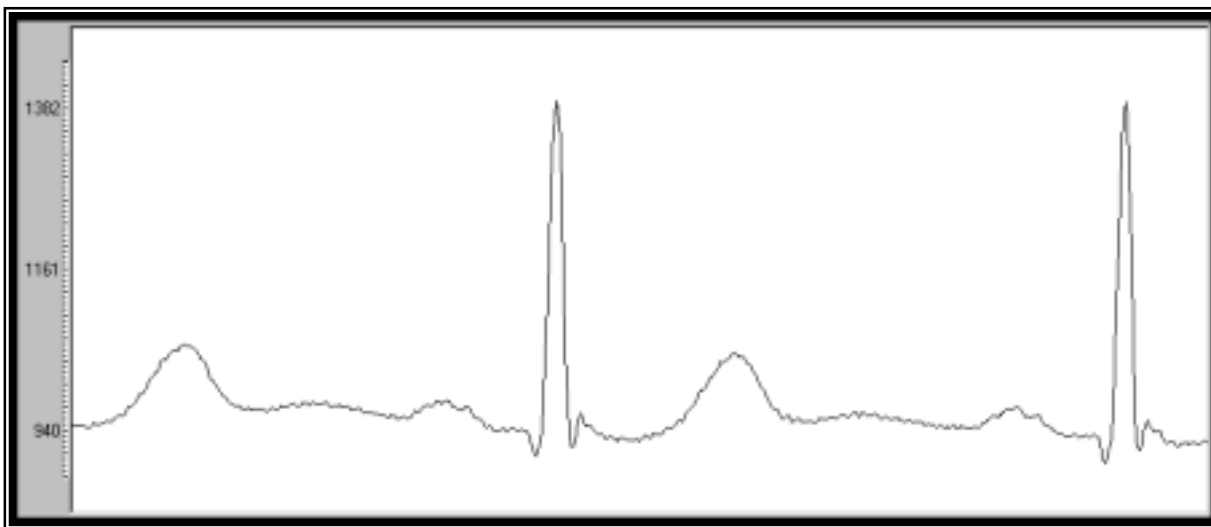
jeżeli okres wynosi N próbek to:

$$s_{-1} = s_{N-1}$$

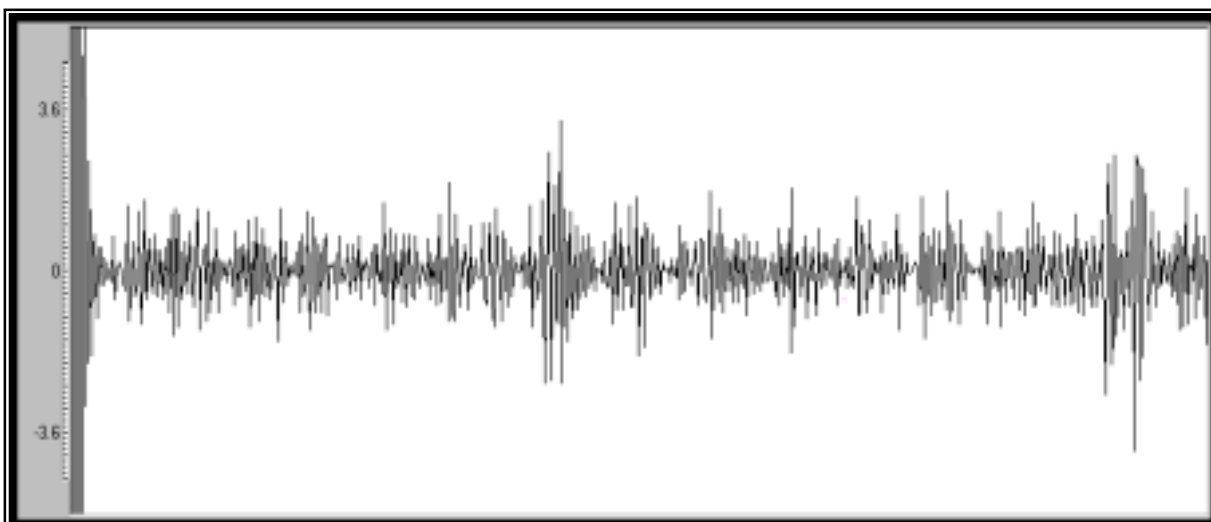
$$s_{-2} = s_{N-2}$$

O ile cztery pierwsze kryteria dają się łatwo zastosować do sygnałów EKG, to z ostatnim kryterium jest problem, ponieważ EKG jest sygnałem niestacjonarnym, co wyklucza okresowość. Można jednak opierając się na powtarzalności sygnału EKG wymuszonej rytmem serca założyć, że sygnał jest okresowy o okresie T-T (równym odległości pomiędzy sąsiednimi załamkami T). W ten sposób można wypełnić brakujące próbki wartościami próbek z sąsiedniego QRS. Znajdowanie następnego QRS-a oparte jest na przeszukiwaniu sygnału w celu znalezienia kilku sąsiednich próbek, których odległość (różnica średniokwadratowa) od próbek na skraju sygnału będzie minimalna. Dokładniejszy opis funkcji realizującej algorytm przeszukiwania zamieszczony jest w załączniku.

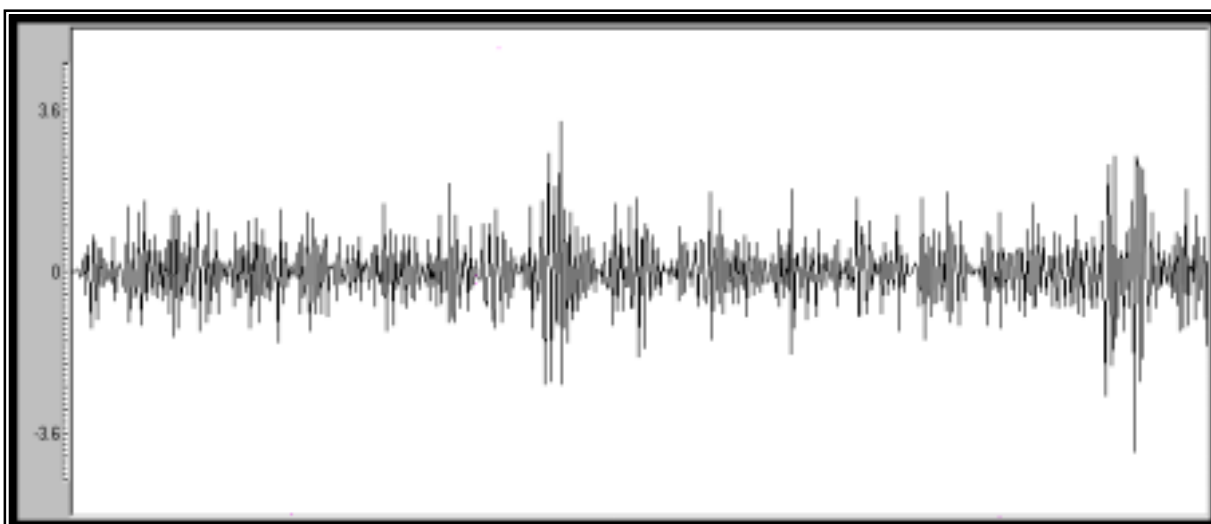
Poniższe doświadczenie ma na celu sprawdzenie jaki wpływ na kompresję ma wybór rodzaju rozszerzenia sygnału. Sygnałem testowym jest 'N14.d', zaś wykorzystana falka to 'db8'. Poziom kompresji ustawiony na „Four levels”. Poniżej na rysunkach przedstawiony został sygnał oryginalny oraz kolejne przebiegi będące wynikiem czteropoziomowej dekompozycji falkowej dla pięciu różnych typów wypełnienia rozszerzonego sygnału.



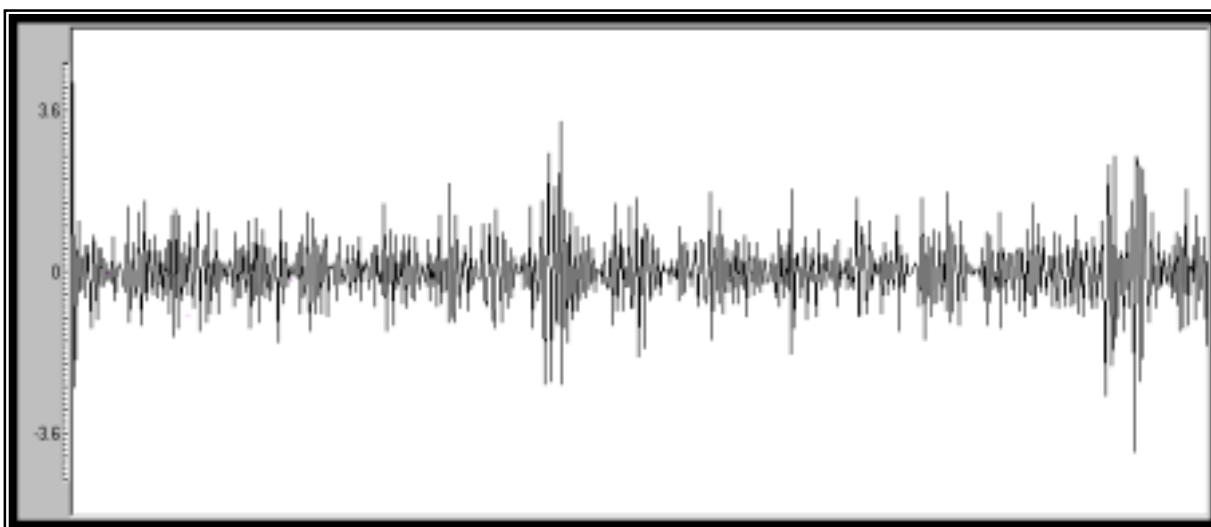
Rys.6.6.1 Sygnał oryginalny 'N14.d' rozszerzony



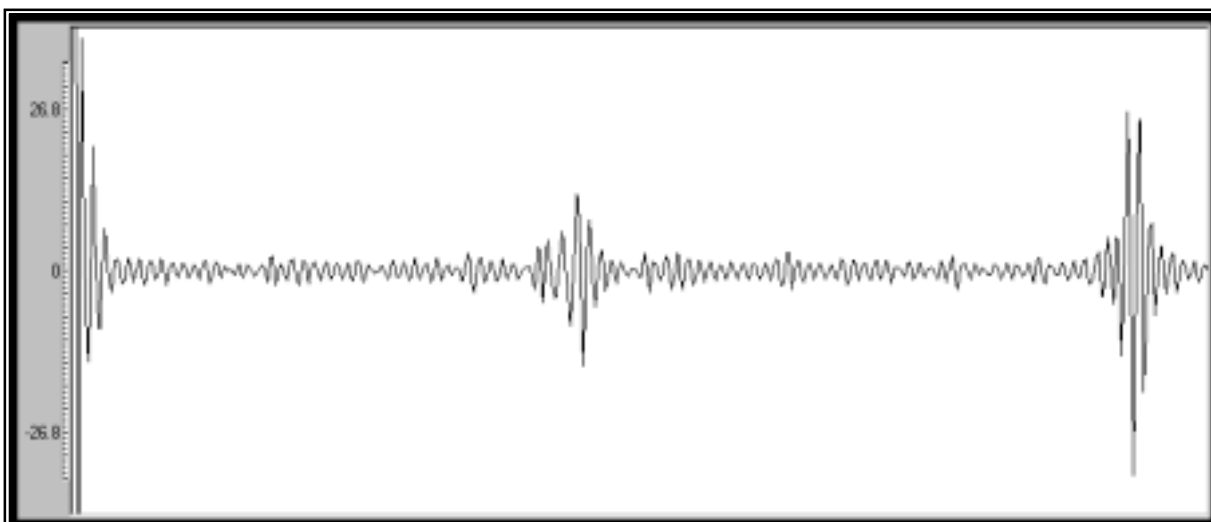
Rys.6.6.2a 'Zero padding' – pierwszy poziom detali



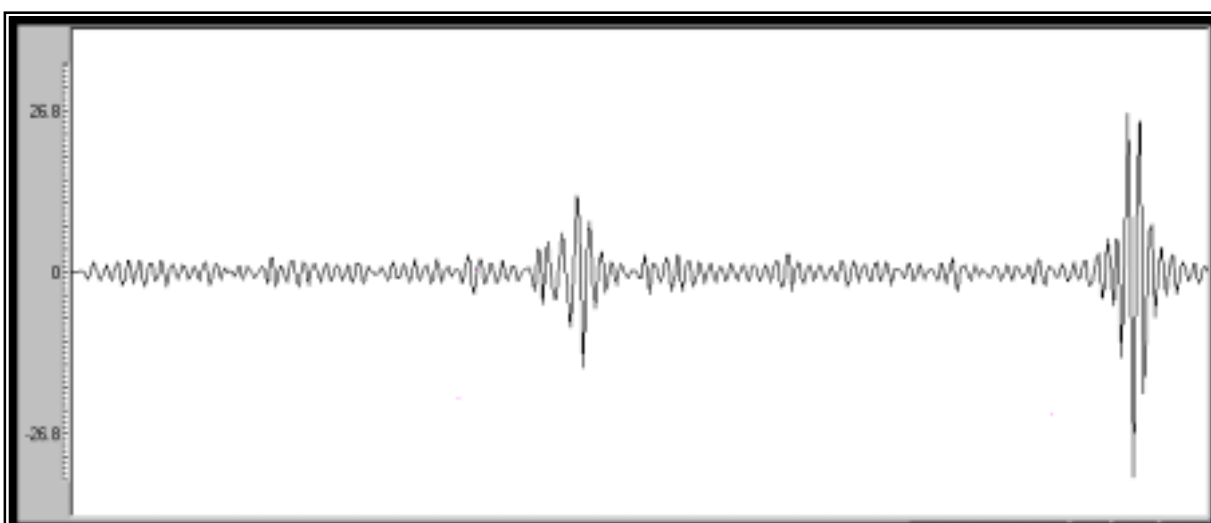
Rys.6.6.3a 'Limit padding' – pierwszy poziom detali



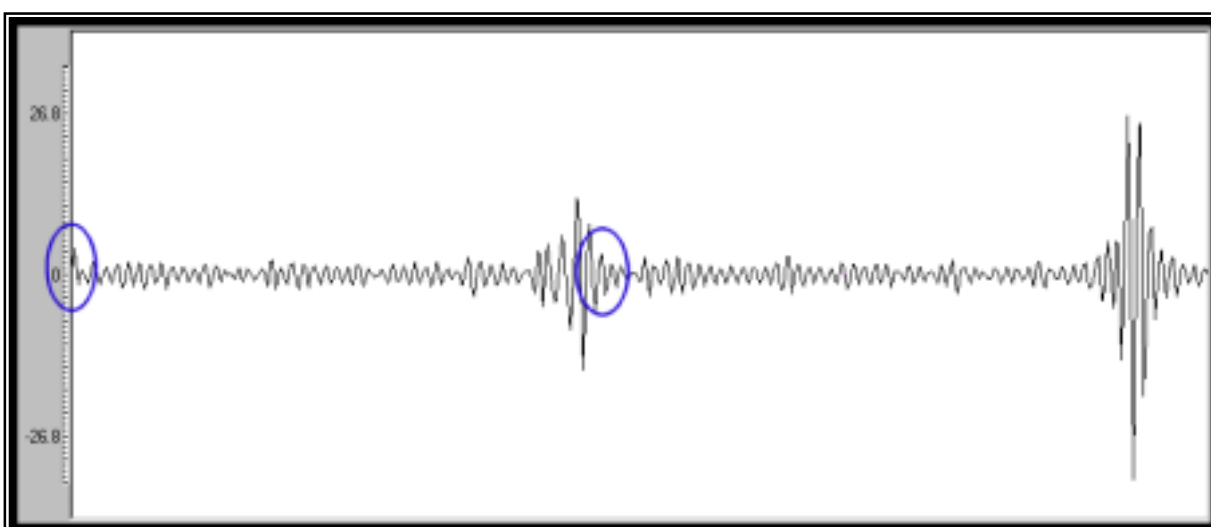
Rys.6.6.4a 'Periodic padding' – pierwszy poziom detali



Rys.6.6.2b 'Zero padding' – drugi poziom detali



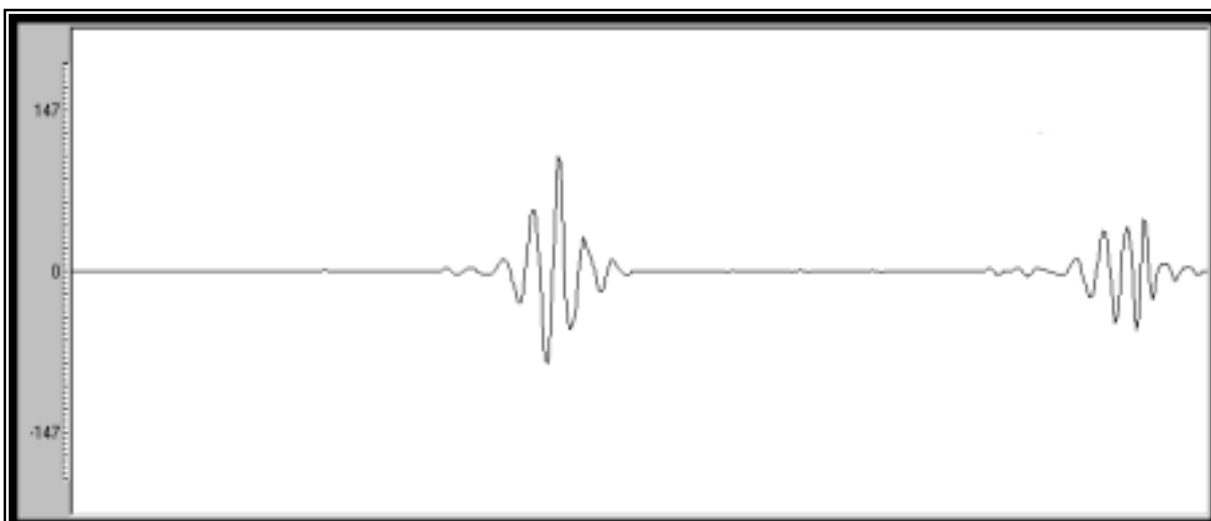
Rys.6.6.3b 'Limit padding' – drugi poziom detali



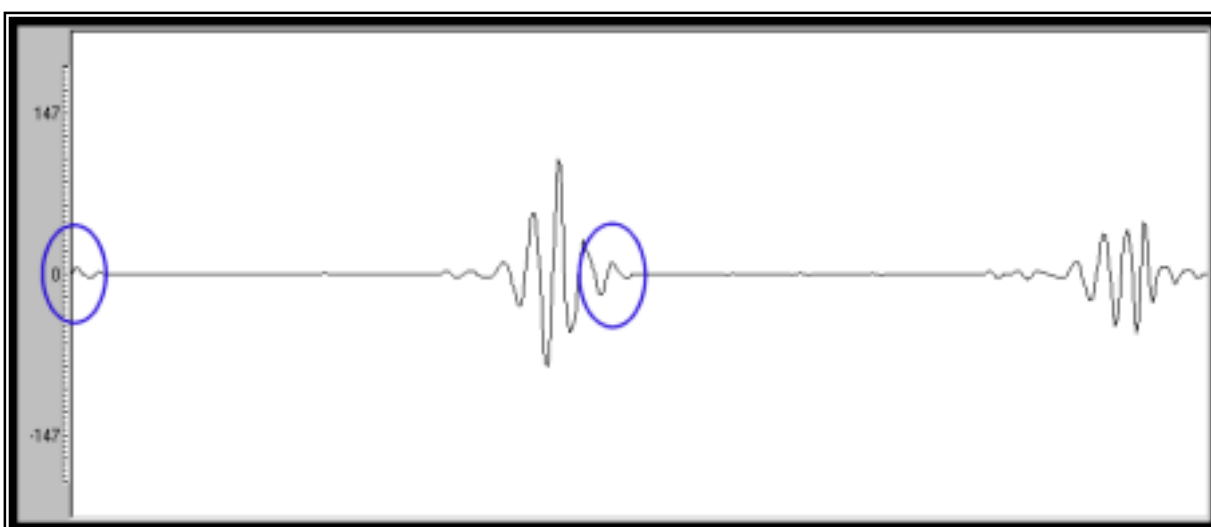
Rys.6.6.4b 'Periodic padding' – drugi poziom detali



Rys.6.6.2c 'Zero padding' – trzeci poziom detali



Rys.6.6.3c 'Limit padding' – trzeci poziom detali



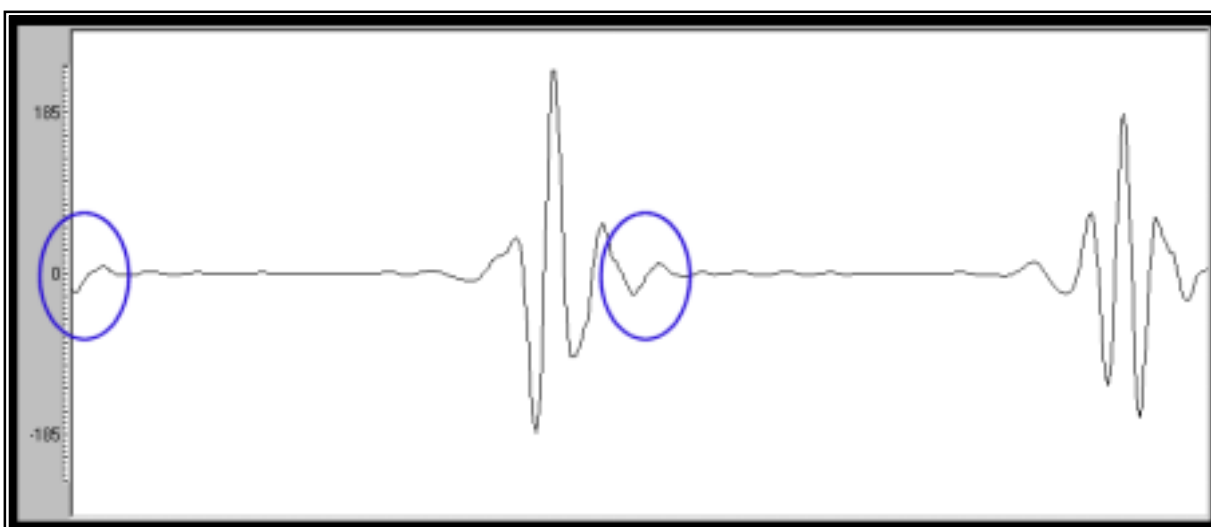
Rys.6.6.4c 'Periodic padding' – trzeci poziom detali



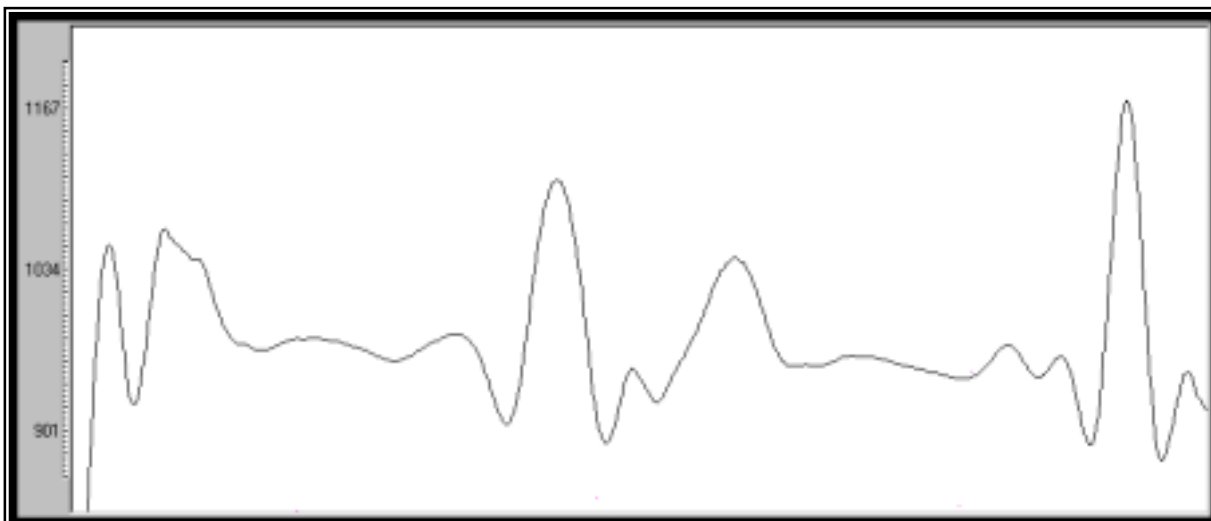
Rys.6.6.2d 'Zero padding' – czwarty poziom detali



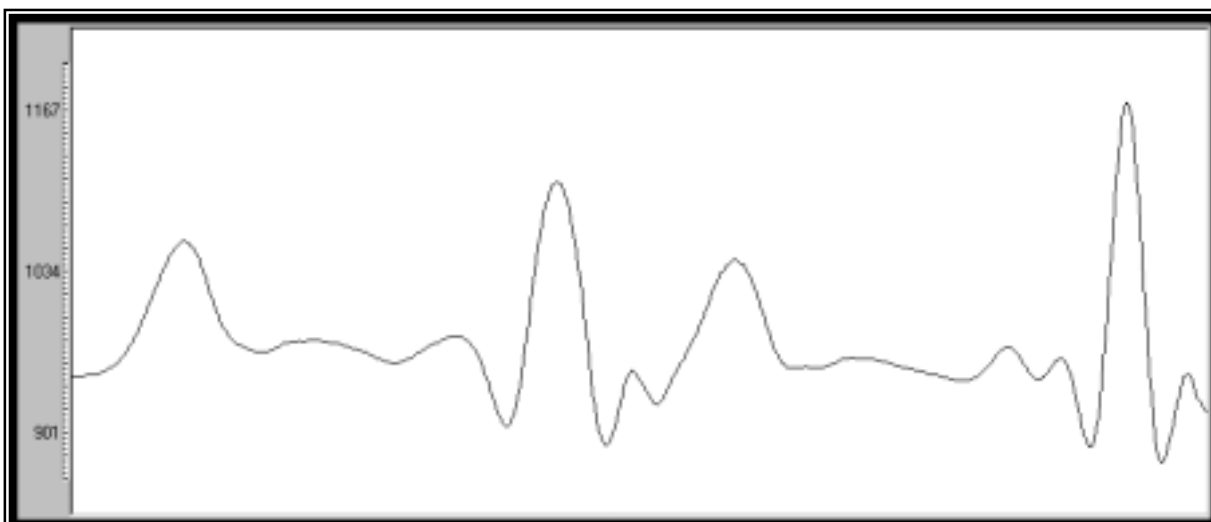
Rys.6.6.3d 'Limit padding' – czwarty poziom detali



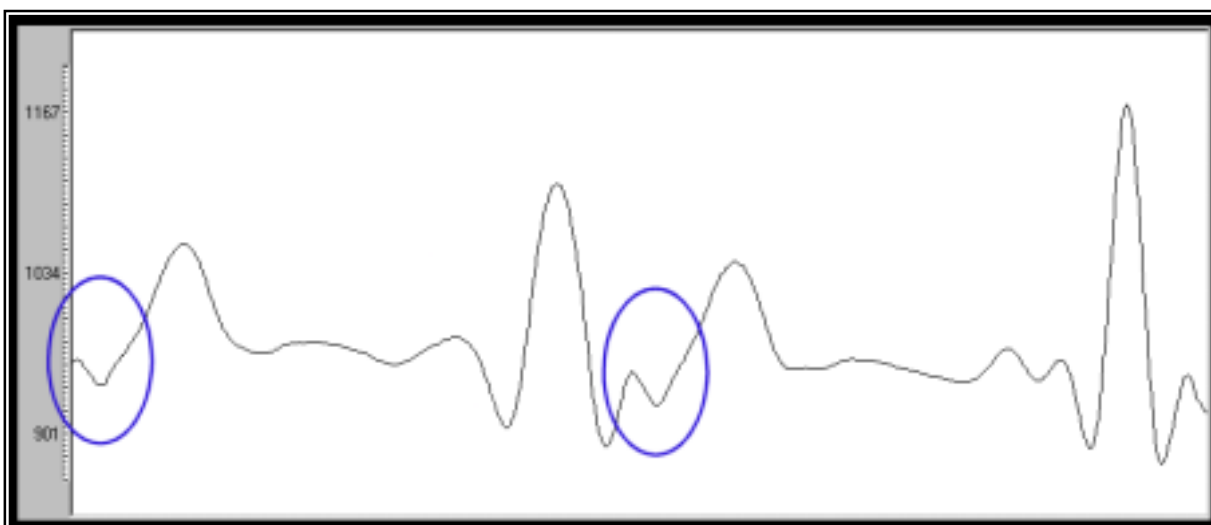
Rys.6.6.4d 'Periodic padding' – czwarty poziom detali



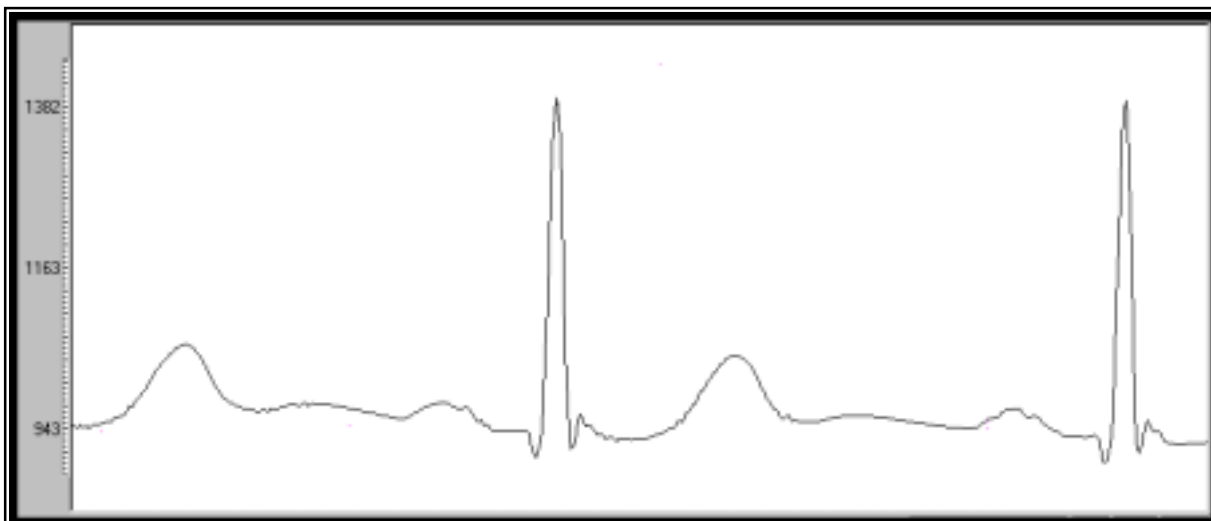
Rys.6.6.2e 'Zero padding' – czwarty poziom aproksymacji



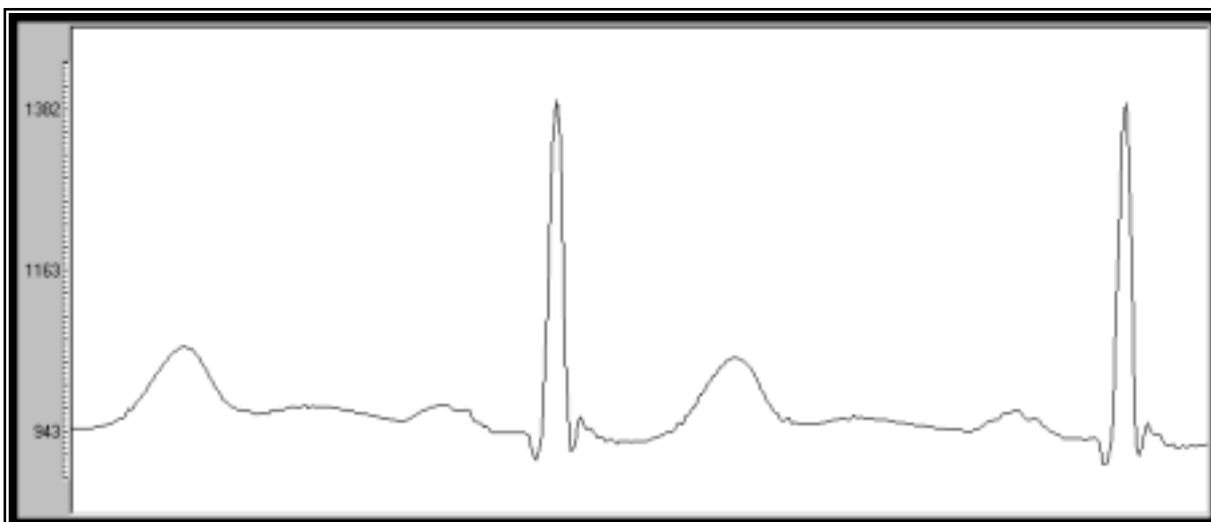
Rys.6.6.3e 'Limit padding' – czwarty poziom aproksymacji



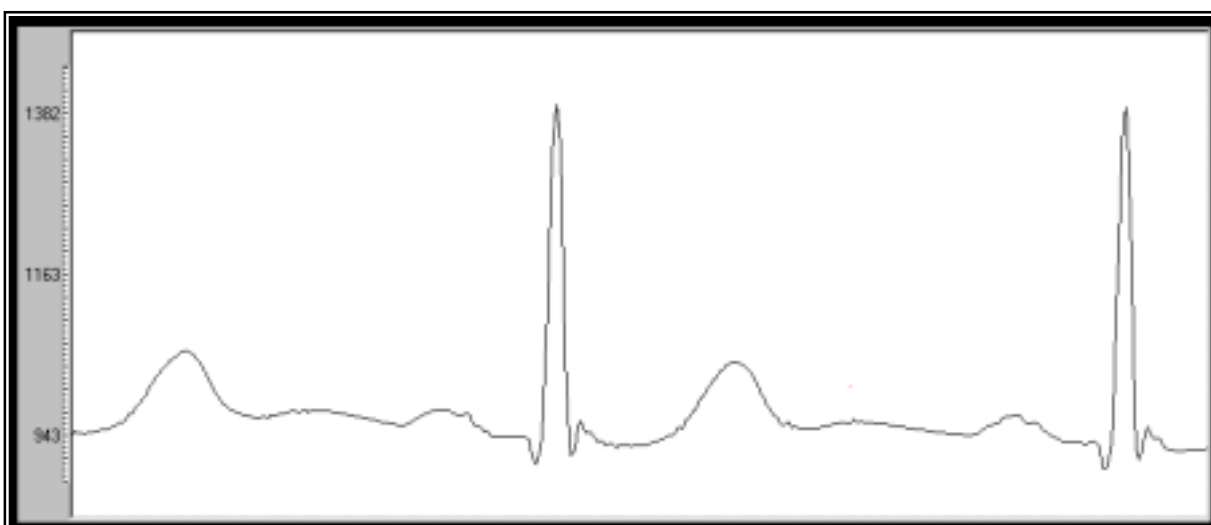
Rys.6.6.4e 'Periodic padding' – czwarty poziom aproksymacji



Rys.6.6.2f 'Zero padding' – rekonstrukcja z progiem równym $4,38\mu\text{V}$. Zawartość zer w sygnale 81,35%, PRD = 1,548%.



Rys.6.6.3f 'Limit padding' – rekonstrukcja z progiem równym $4,25\mu\text{V}$. Zawartość zer w sygnale 83,48%, PRD = 1,537%.



Rys.6.6.4f 'Periodic padding' – rekonstrukcja z progiem równym $4,32\mu\text{V}$. Zawartość zer w sygnale

Po dokładnej obserwacji otrzymanych przebiegów nasunęły się następujące wnioski:

- Wybór typu rozszerzenia nie ma wpływu na jakość pełnej rekonstrukcji. Za każdym razem sygnał zrekonstruowany był identyczny z oryginalnym. Z tego też powodu przebiegów tych nie przedstawiono na rysunkach. Potwierdza to teoretyczne rozważania o bezstratności transformacji falkowej. Zmiana typu rozszerzenia wpływa natomiast na rozkład energii pomiędzy poszczególnymi poziomami dekompozycji. Widać, że początki przebiegów detali i aproksymacji (z lewej strony) różnią się między sobą.
- Wypełnienie współczynników sposobem „Zero padding” wprowadziło duże zakłócenie do sygnału. Widać na przebiegu sygnału oryginalnego, że początkowe próbki mają wartość około $950\mu\text{V}$. Nagły spadek do wartości zero spowodował, że w tym miejscu w sygnale pojawiły się składowe wysokoczęstotliwościowe o bardzo dużej amplitudzie. Jest to dokładnie pokazane na przebiegu detali poziomu pierwszego (rys.6.6.2a). Na pozostałych poziomach detali i na przebiegu aproksymacji także widać efekt działania skoku, co jest oczywiste, ponieważ skok jednostkowy zajmuje teoretycznie nieskończone pasmo częstotliwości (rys.6.6.2b-e). Z powodu dużych amplitud współczynników detali na początku sygnału kompresja tego sygnału była mniej optymalna od pozostałych przykładów. Procent wyzerowanych współczynników wyniósł 81,35%, natomiast PRD 1,548%. Przypadek ten, jak wcześniej wspomniano, jest równoznaczny z tym, jakby w ogóle nie była stosowana procedura rozszerzenia sygnału, co łatwo można zauważyć przez podstawienie wartości zerowych przy obliczaniu początkowych próbek splotu w zamieszczonych powyżej obliczeniach. Jest on więc dobrym przykładem potwierdzającym użyteczność procedury rozszerzania sygnału.
- „Limit padding”, „Smooth padding of order 1” oraz „Symetric padding” w naszym przypadku zachowywały się tak samo. Można zauważyć, że sygnał jest stały na początku sygnału. To spowodowało, że próbki wygenerowane przez te trzy algorytmy są praktycznie takie same. Mogą one ewentualnie różnić się podczas wypełniania rozszerzonego sygnału na wyższych poziomach, mimo to jednak nie miało to znaczącego wpływu na finalny kształt poszczególnych poziomów sygnałów. Z tego powodu te trzy algorytmy wypełnienia są rozpatrywane razem, także rysunki 6.6.3 dotyczą zarówno „Limit padding”, jak i dwóch pozostałych. Wypełnienie za pomocą tych sposobów sprawiło, że w kolejnych poziomach dekompozycji sygnał pozostał stały, czego efektem stała się pomijalnie mała amplituda współczynników detali w tym miejscu. Pociąga to za sobą, że na początku sygnałów poszczególnych rekonstrukcji poziomów detali przebiegi są płaskie (rys. 6.6.3b-e), a sygnał aproksymacji ma w tym miejscu łagodny kształt. Wydaje się więc oczywiste, że kompresja w tym przypadku była najskuteczniejsza: 83,48% wyzerowanych współczynników oraz PRD = 1,537%.
- Bardzo ciekawe efekty zostały uzyskane w wyniku zastosowania wypełnienia typu „Periodic padding”. W rekonstrukcjach kolejnych poziomów detali i aproksymacji widać dodatkowe zaburzenie na początku sygnału wywołane wygenerowanymi współczynnikami. Te dodatkowe współczynniki to część załamka S „zapożyczonego” z sąsiedniego QRS. W zamieszczonych przebiegach rekonstrukcji (rys.6.6.4a-e), szczególnie wyższych poziomów detali (trzeciego i czwartego) oraz aproksymacji widać, że kształt początku sygnału ma charakter taki jak w następnym QRS. Gdyby oryginalnie sygnał był dłuższy, prawdopodobnie właśnie taki kształt miałyby poszczególne rekonstrukcje w tym miejscu. Ten sposób generacji „brakujących próbek” jest lepszy od wcześniejszych. Tamte algorytmy wypełniały sygnał na podstawie brzegowych wartości sygnału („Limit padding”, „Smooth padding of order 1” oraz „Symetric padding”) lub też w ogóle od niego nie zależały („Zero padding”). „Periodic padding” w przeciwieństwie do swoich poprzedników dobierał wartości próbek kierując się

charakterem całego sygnału. Porównując przebiegi detali czwartego poziomu „Zero padding”(6.6.2d) i „Periodic padding” (6.6.4d) można by stwierdzić, że ten pierwszy również wygenerował na początku kształt podobny do tego z sąsiedniego QRS-a, jest on nawet dużo wyraźniejszy. Jednak po bliższej analizie wynika, że kształt ten nie pochodzi od QRS-a, lecz od wspomnianego wyżej skoku od wartości 0 do 950. Przekonyującym dowodem na to jest fakt, że odległość tego kształtu od następnego QRS jest mniejsza od normalnej odległości między QRS-ami (odległości „T-T”). Kompresja uzyskana przy pomocy metody „Periodic padding” pozwoliła na wyzerowanie 81,96% współczynników przy PRD równym 1,545%. Efektywność kompresji w porównaniu do „Limit padding” jest mniejsza, lecz jest to zrozumiałe biorąc pod uwagę fakt obecności części załamka QRS na początku sygnału.

- Porównując sygnały rekonstrukcji progowej dla kolejnych typów wypełnienia sygnału oraz liczbowe wartości wyzerowanych współczynników i wskaźnika PRD wynika, że z punktu widzenia kompresji wybór rodzaju wypełnienia nie ma większego znaczenia. Widać, że różnice na początkach sygnałów rekonstrukcji (rys.6.6.2f, 6.6.3f oraz 6.6.4f) są bardzo symboliczne. W bardzo dużym powiększeniu można zauważyć dopiero faktyczne różnice między tymi sygnałami.
- Na wielkość zniekształceń brzegowych ma wpływ długość falki. Im dłuższe są filtry zbudowane z falek, tym więcej współczynników trzeba wygenerować i wszystkie one będą miały wpływ na dekompozycję. Tak więc wybór sposobu wypełnienia staje się istotniejszy. W zaprezentowanym przykładzie wykorzystano dosyć długą falę ‘db8’ (długość filtrów wynosi 16). Zakłada się jednak, że długość sygnału jednowymiarowego jest dużo większa od długości falki, zatem ewentualne zniekształcenia na brzegach sygnału nie zniekształcają praktycznie w ogóle informacji niesionej przez ten sygnał. Problem zniekształceń brzegowych nabiera dopiero większego znaczenia w przypadku kompresji falkowej obrazów dwuwymiarowych, gdzie wymiary obrazu mają ograniczoną ilość pikseli i wtedy wybór długości falki pod tym kątem jest bardzo istotna. Na koniec wykonano jeszcze raz to samo doświadczenie wykorzystując krótszą falę ‘db2’(długość filtrów wynosi 4). Otrzymane wyniki kompresji dla porównania zamieszczono w tabeli 6.6.5. Na ich podstawie widać, że faktycznie krótsze falki mają mniejszy wpływ na zniekształcenia brzegowe, przez co dobór rodzaju wypełnienia rozszerzonego sygnału jest mniej istotny.

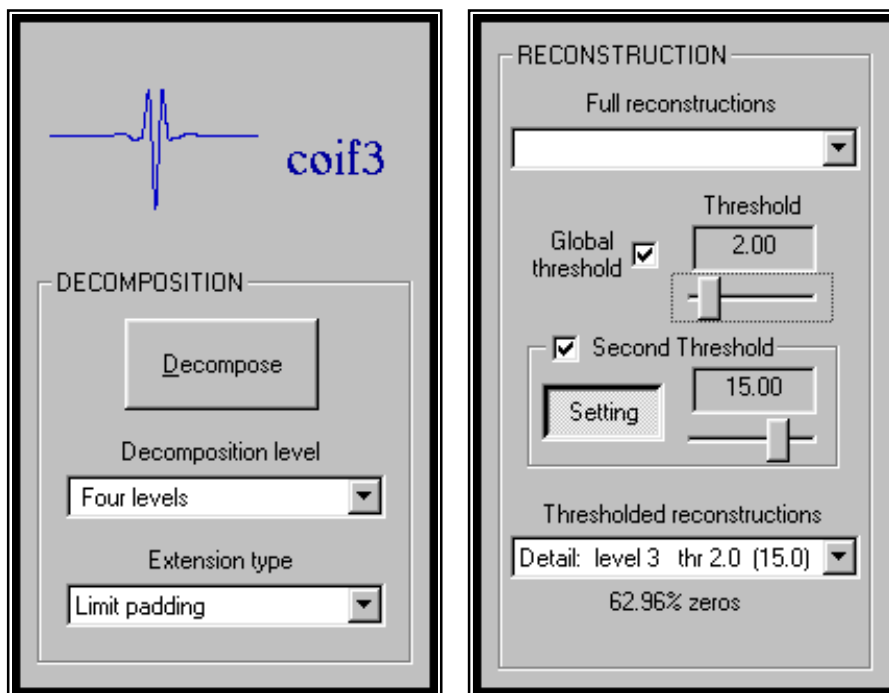
Sygnal Lbb2.d poziom dekompozycji: ‘four levels’				
Rodzaj wypełnienia	Falka ‘Daubechies 8’		Falka ‘Daubechies 2’	
	Zera	PRD	Zera	PRD
‘Zero padding’	81,35%	1,458%	82,35%	1,826%
‘Limit padding’	83,48%	1,537%	82,98%	1,827%
‘Smooth padding’	83,48%	1,537%	82,98%	1,827%
‘Symetric padding’	83,48%	1,537%	82,98%	1,827%
‘Periodic padding’	81,96%	1,545%	82,69%	1,828%

Tab.6.6.5 Efektywność kompresji w zależności od rodzaju wypełnienia rozszerzonego sygnału

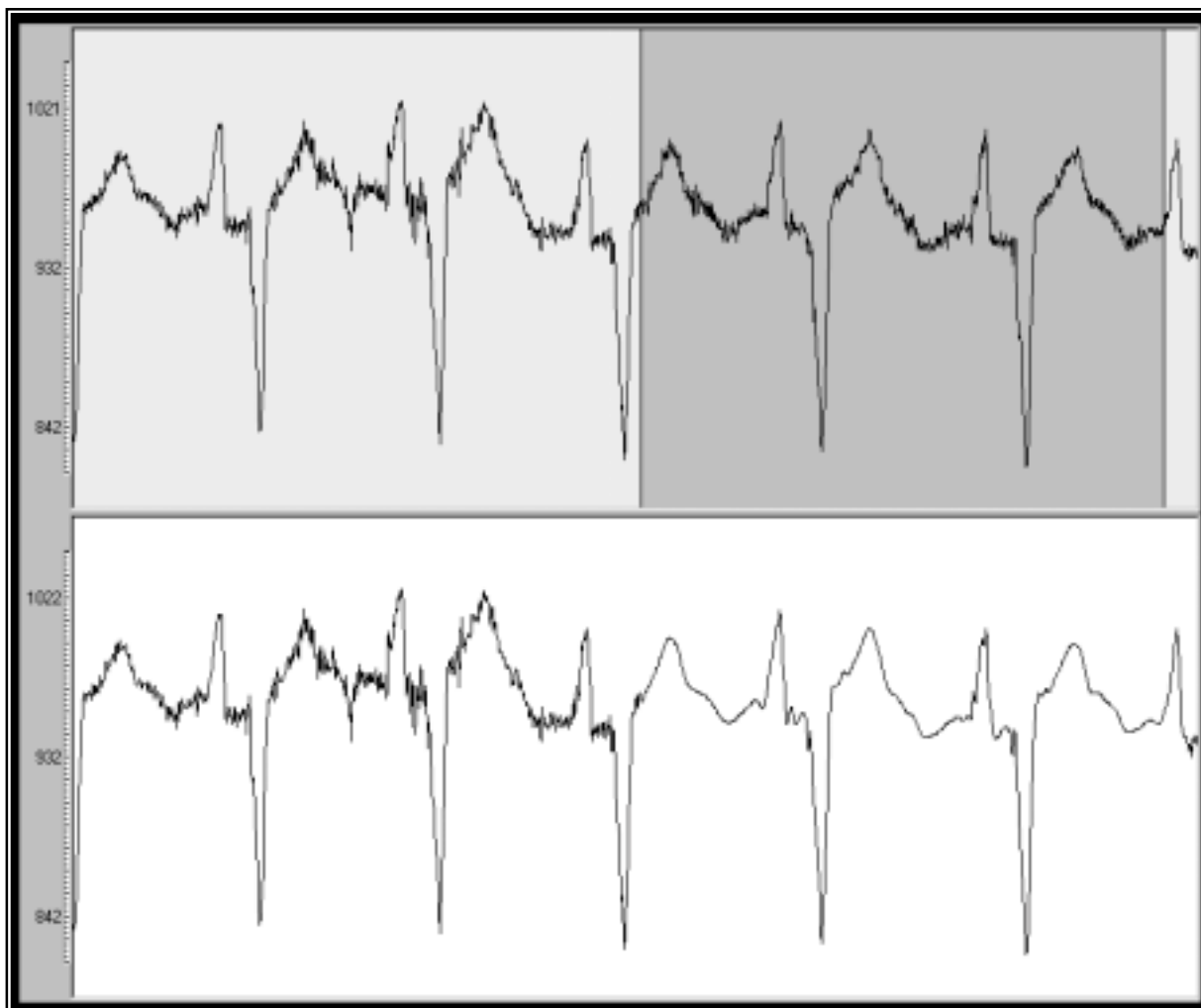
6.7. Zastosowanie drugiego progu

W przypadku, gdy informacja niesiona przez sygnał nie jest tak samo ważna w różnych przedziałach czasowych bardzo przydatnym narzędziem może okazać się progowanie zależne od czasu. W sygnale EKG możemy wyróżnić dwie fazy: fazę pracy serca oraz fazę spoczynku. Pierwsza faza obejmuje skurcz przedsionków (załamek P), skurcz komór (załamki QRS) oraz rozkurcz komór (załamek T), zaś druga faza zawiera odcinek między T i P. Zakładając, że druga faza jest mniej istotna z punktu widzenia diagnostyki lekarskiej to koncepcja stosowania progowania czasowego ma również odniesienie w sygnałach EKG. W programie został zrealizowany mechanizm skokowej zmiany progu. Jest to wprawdzie trywialna forma progowania czasowego, jednak doskonale pokazująca zalety tej idei. Dodatkowo zaletą tego rozwiązania jest prosty interfejs ustawiania wartości i zakresów drugiego progu za pomocą myszy w górnym oknie.

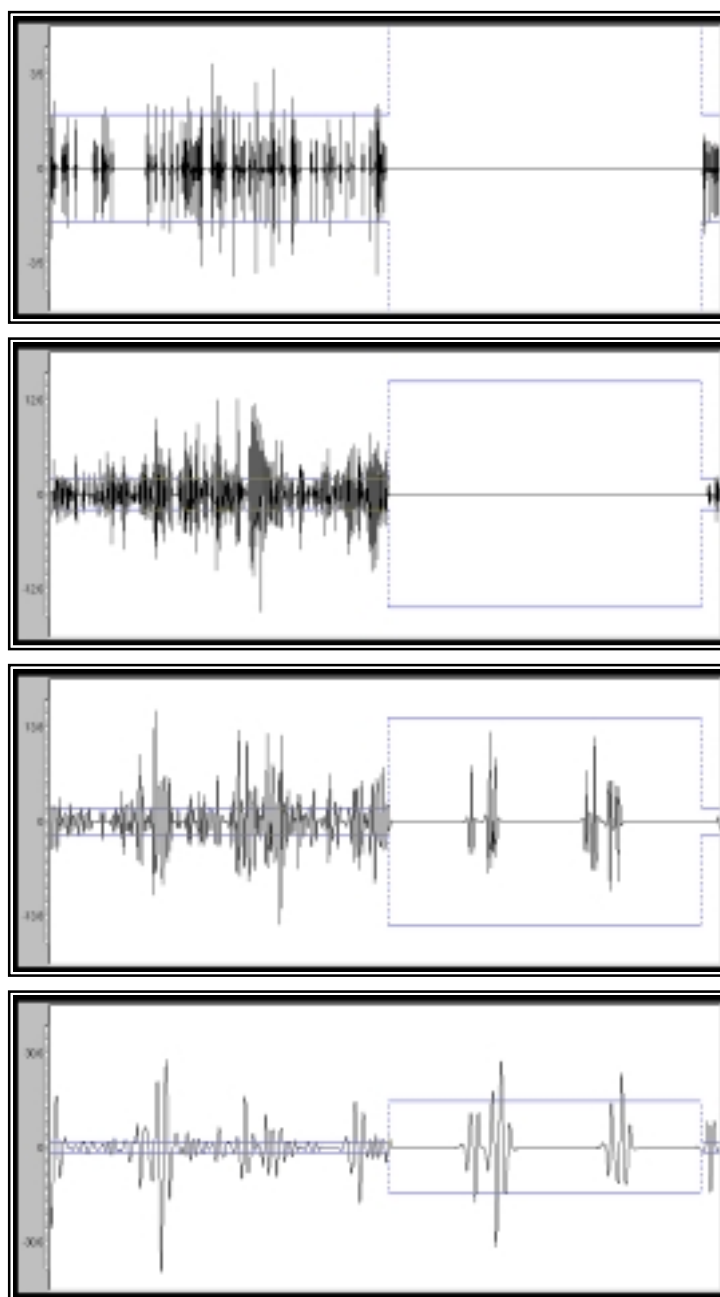
Pierwsze doświadczenie ma za zadanie pokazanie konsekwencji zastosowania drugiego progu. Do prezentacji wybrany został sygnał 'N3.d', falka 'coif.3', poziom dekompozycji: czwarty oraz rozszerzenie 'Limit padding'. Po przeprowadzeniu dekompozycji zaznaczono 'Second level', a następnie wciśnięto przycisk 'Settings'. Spowodowało to nieznaczące wyszarzenie górnego okna. Następnie za pomocą myszy metodą „przeciągnij i upuść” zaznaczono obszar od połowy do końca okna. Obszar ten po dalszym wyszarzeniu stał się zakresem drugiego progu. Aby ustawić amplitudę progów zrekonstruowano jeden z poziomów detali. Pozostawiając zaznaczony 'Global threshold' ustawiono amplitudy progów odpowiednio: pierwszy $2\mu\text{V}$ oraz drugi $15\mu\text{V}$. Przykładowy widok pulpitu przedstawiono na rys.6.7.1. Dla tych ustawień przeprowadzono progową rekonstrukcję całego sygnału. Otrzymany efekt przedstawiono na rys.6.7.2a. Na rysunku tym widać wyraźnie, że w zakresie ważności drugiego progu zrekonstruowany sygnał stał się bardzo wygładzony. Amplituda $15\mu\text{V}$ zlikwidowała w tym obszarze składowe o wyższych częstotliwościach. W celu pokazania rezultatów działania tak skonfigurowanego progu na poszczególne poziomy zamieszczono dodatkowo progowe rekonstrukcje detali (rys.6.7.2b).



Rys.6.7.1 Widok ustawień pulpitu dekompozycji i rekonstrukcji

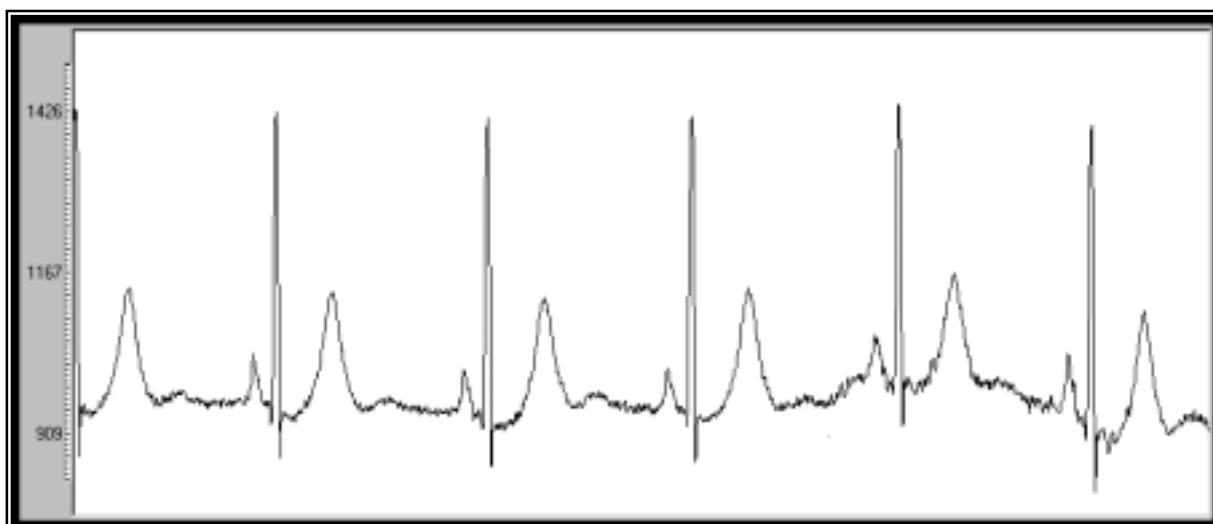


Rys.6.7.2a Kompresja sygnału 'N3.d' z wykorzystaniem drugiego progu



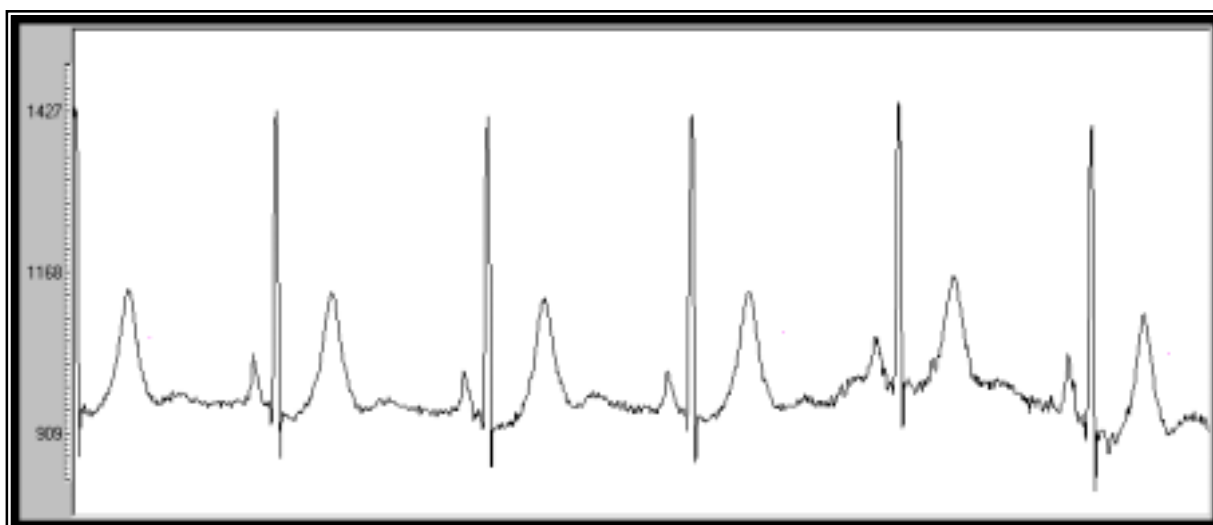
Rys.6.7.2b Rekonstrukcje progowe poszczególnych poziomów detali z wykorzystaniem drugiego progu. Na górze jest pierwszy poziom, na dole czwarty. Wartość pierwszego progu $2\mu\text{V}$, drugiego $15\mu\text{V}$

Następne doświadczenie pokazuje praktyczne możliwości, jakie daje progowanie czasowe. Przeprowadzono je z wykorzystaniem sygnału 'N9.d' (rys.6.7.3) oraz falki 'bior3.3'. Ustawione parametry to 'Five levels' dla poziomu dekompozycji i 'Limit padding' dla wypełnienia próbek rozszerzeń sygnału. Ze wstępnej analizy sygnału wynika, że w obrębie załamek P, QRS i T ma on dosyć ostre kształty, prawdopodobnie zawierające istotne informacje o pracy serca. Przeciwnie, odcinek między T i P ma charakter wolnozmienny z nałożoną składową wysokoczęstotliwościową o małej amplitudzie, którą w zasadzie można uznać za szum i próbować wyeliminować.

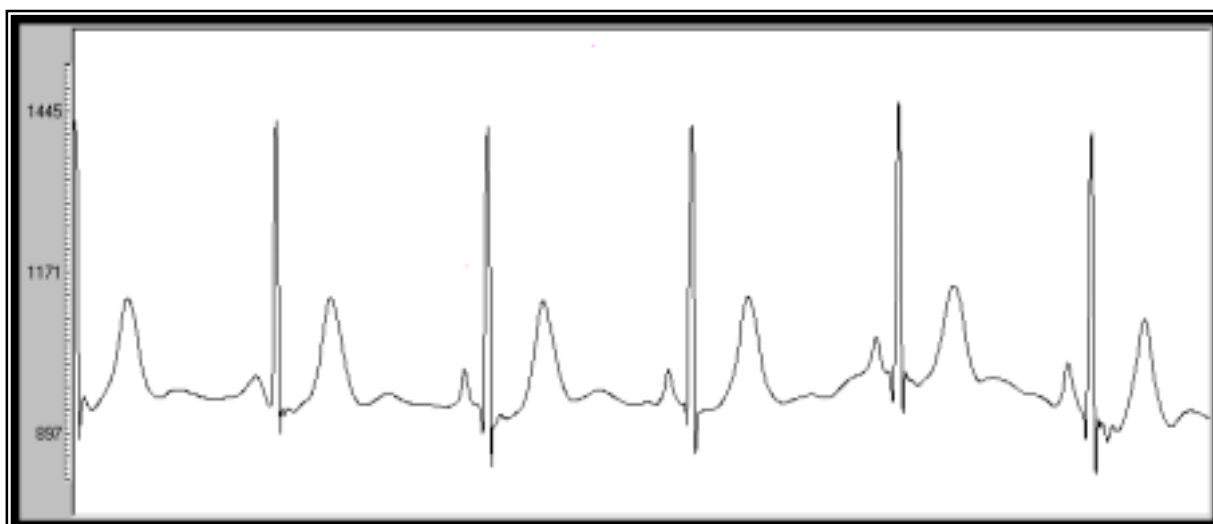


Rys.6.7.3 Sygnał oryginalny 'N9.d'

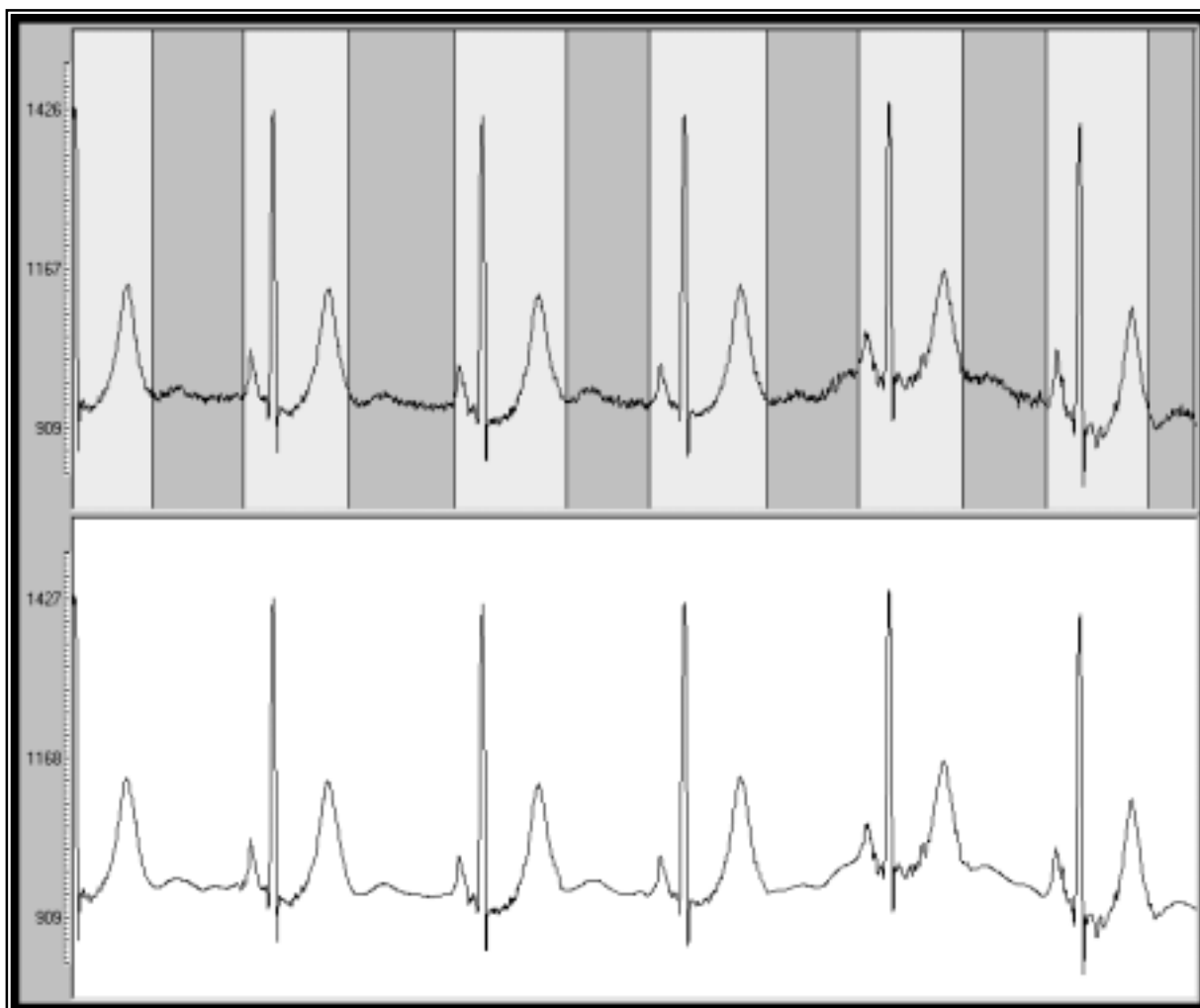
Na początek przeprowadzono kompresję z wykorzystaniem jednego progu tak, aby uzyskać zadowalającą jakość sygnału. Po dekompozycji wartość progu została automatycznie ustawiona na $4,09\mu\text{V}$. Wartość tę zwiększono do $6\mu\text{V}$ i tak skompresowane współczynniki zrekonstruowano otrzymując sygnał przedstawiony na rys.6.7.4.

Rys.6.7.4 Sygnał 'N9.d' zrekonstruowany dla progu $6\mu\text{V}$. Zawartość zer wynosi 73,80%, PRD = 2,056%.

Widać, że otrzymany przebieg zachował wszystkie istotne szczegóły z sygnału oryginalnego przy jednocześnie dość wysokim stopniu kompresji wynoszącym 73,80%. Z drugiej strony jednak odcinki między załamkami T, a P pozostały ciągle zaszumione. Zatem sygnał skompresowany nadal zawierał zbędne (pasożytnicze) informacje, które magazynują przeciw energii. Próba zwiększenia wartości progu tak, aby wyeliminować niepotrzebne składowe spowodowała również utratę ważnych informacji z obszaru załamek P i QRS. Przypadek ten obrazuje rysunek 6.7.5. Całkowitą eliminację szumów osiągnięto dla progu $50\mu\text{V}$. Widać, że wspomniane odcinki nabrały pożądanego, łagodnego kształtu, jednak zbyt wysoki próg zniekształcił ostre w oryginale załamki P (w szczególności pierwszy z lewej).



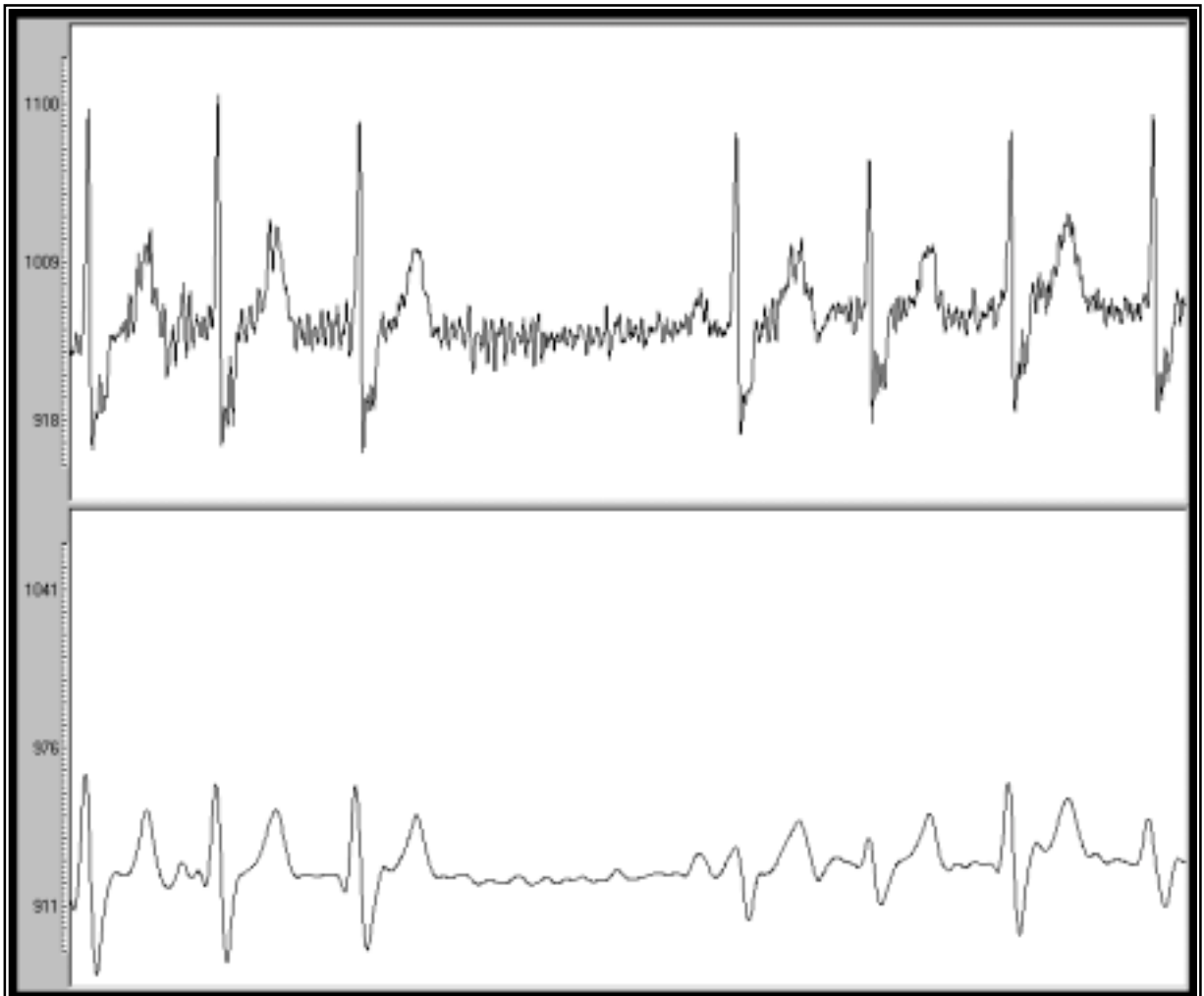
Rys.6.7.5 Sygnał 'N9.d' zrekonstruowany dla progu $50\mu\text{V}$. Zawartość zer wynosi 93,27%, PRD = 7,373%. Duże zniekształcenia załameków P.



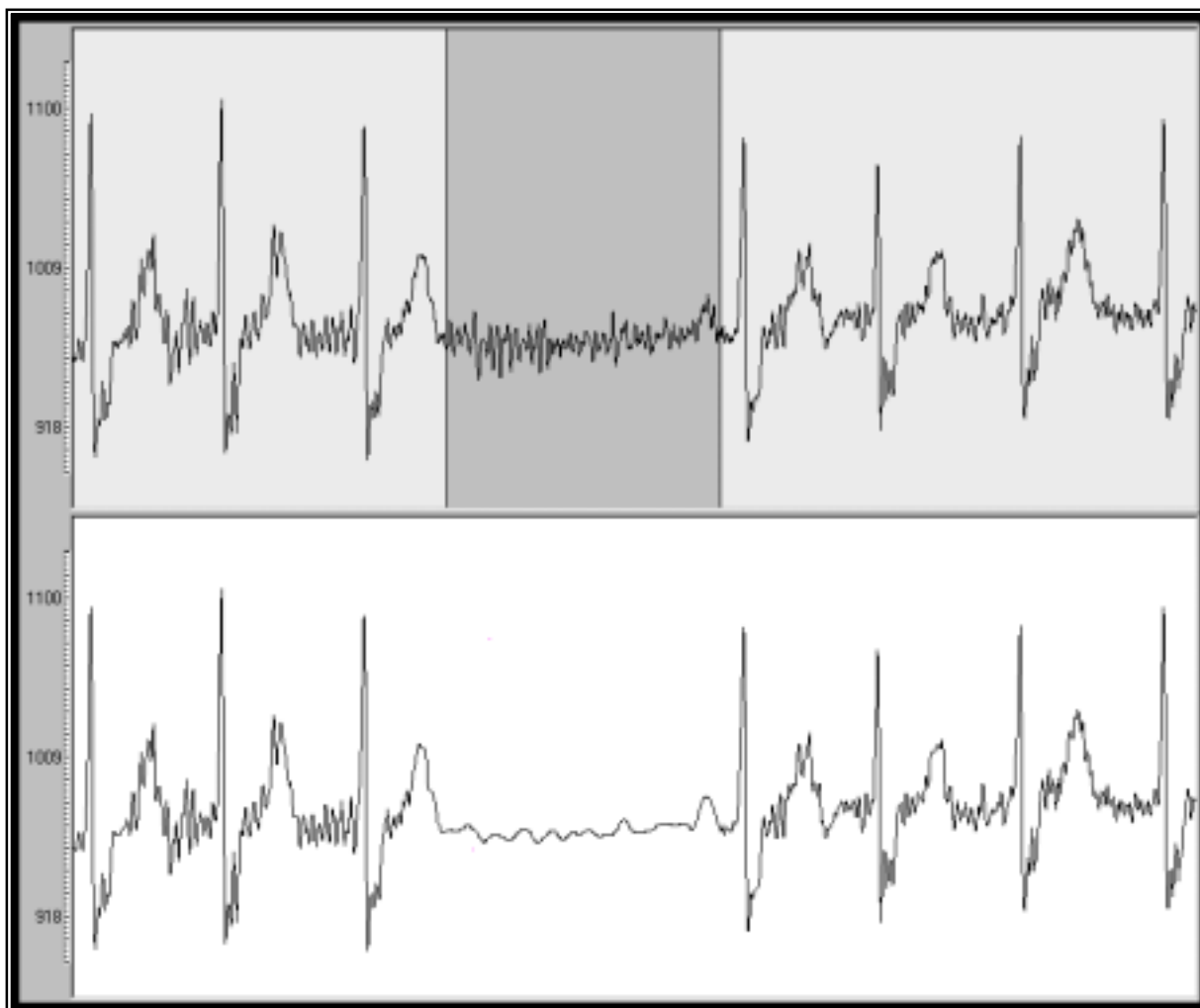
Rys.6.7.6 Ustawienia drugiego progu (u góry). Ciemniejszy kolor oznacza obszar ważności drugiego progu. Na dole zrekonstruowany sygnał dla progów ustalonych na $6\mu\text{V}$ dla pierwszego i $50\mu\text{V}$ dla drugiego progu. Zera – 82,64%, PRD – 3,929%.

Rozwiązaniem dla tego rodzaju konfliktu jest zastosowanie drugiego progu. Zakresy aktywności drugiego progu ustawiono, jak na rysunku 6.7.6 (u góry). Dla pierwszego progu ustalono wartość $6\mu\text{V}$, natomiast dla drugiego $50\mu\text{V}$. Uzyskany efekt zaprezentowano w dolnej części rys.6.7.6. W tym przypadku otrzymano pożądaną jakość kompresji – zachowanie istotnych informacji oraz redukcję zbędnych szczegółów. Metoda ta pozwoliła osiągnąć także ponadprzeciętny współczynnik kompresji 82,64%.

Sygnał wykorzystany w następnym doświadczeniu to 'Apc0.d' charakteryzujący się brakiem jednego cyklu serca. W środku sygnału brak załamków P, QRS i T (rys.6.7.7). W zasadzie jedyną istotną informacją w tym miejscu jest sam fakt braku załamków i czas trwania tego stanu. Szczegółowy rozkład składowych poszczególnych częstotliwości jest w tym przypadku raczej bez znaczenia. Do analizy sygnału wykorzystano tą samą co poprzednio falkę 'bior3.3', cztery poziomy dekompozycji i rozszerzenie 'Limit padding'. W dolnej części tego rysunku przedstawiono także rekonstrukcję współczynników czwartego poziomu aproksymacji. Okazuje się, że w miejscu braku QRS sam sygnał aproksymacji zawiera wystarczającą ilość informacji, zatem można tutaj zmienić próg na większy i pozbyć się wszystkich współczynników detali.



Rys.6.7.7 U góry sygnał oryginalny 'Apc0.d', u dołu zrekonstruowany sygnał współczynników aproksymacji czwartego poziomu z wykorzystaniem falki 'bior3.3'



Rys.6.7.8 U góry ustawienia zakresu drugiego progu (kolor ciemniejszy), u dołu zrekonstruowany sygnał dla wartości pierwszego progu $3,08\mu V$ oraz drugiego $80\mu V$

Na rysunku 6.7.8 pokazano ustawienia ważności drugiego progu. Ustawiono dla niego maksymalną amplitudę $80\mu V$, pozostawiając domyślną amplitudę progu pierwszego – $3,08\mu V$. Tak wysoki próg drugiego progu spowodował usunięcie wszystkich współczynników z zaznaczonego zakresu. Widać, że w tym miejscu sygnał zrekonstruowany jest identyczny z sygnałem aproksymacji. Stopień kompresji tego sygnału wyniósł $74,78\%$ wyzerowanych współczynników, natomiast PRD ustaliło się na $3,304\%$.

7. Podsumowanie

Celem niniejszej pracy była analiza zastosowania kompresji falkowej do sygnałów EKG. Założenie to zostało spełnione. W pierwszych rozdziałach zamieszczono dokładną i ujętą w przejrzysty sposób charakterystykę transformacji falkowej. Wyjaśniono między innymi zasadę obliczania drzewa dekompozycji oraz sposób rekonstruowania sygnału.

Praca ta została wzbogacona programem, w którym zaimplementowano powyższe algorytmy. Program umożliwia praktyczną realizację przetwarzania falkowego na dołączonej do aplikacji bazie rzeczywistych sygnałów EKG. Niewątpliwą zaletą programu jest jego aspekt edukacyjny. Intuicyjny oraz prosty w obsłudze interfejs, dostępność wielu opcji i ustawień parametrów kompresji falkowej, a także bogaty zestaw falek pozwala na lepsze zrozumienie tematu poprzez możliwość doświadczalnej weryfikacji konkretnych zagadnień. Ważnym atutem jest również użyty język programowania „C++”, który zapewnia szybkie i niezawodne działanie programu.

W rozdziale szóstym pracy zaprezentowano przykładowe ćwiczenia z programem, kładąc nacisk na możliwie klarowne przedstawienie właściwości falek. Omówione zostały takie aspekty, jak wpływ rodzaju falki na jakość kompresji oraz wpływ progu obcinania, czy poziomu dekompozycji. Warto zwrócić uwagę na doświadczenie „6.6. Problem brzegowy” dotyczące zagadnienia brzegowego związanego z operacją splotu wykorzystywaną w procedurach obliczania dekompozycji i rekonstrukcji falkowej. Omówiono w nim kilka klasycznych metod korekcji tego problemu oraz podano jeden algorytm zwany ‘Periodic padding’ będący koncepcją własną autora. Metoda zapewnia bardzo dobrą jakość przetwarzanego sygnału. Dobieranie brakujących próbek na podstawie wartości z sąsiedniego QRS spowodowało naturalny rozkład energii sygnału na kolejnych poziomach dekompozycji i zachowanie oryginalnego kształtu przebiegu po kompresji.

Bardzo interesujące jest poruszone w rozdziale „6.7. Zastosowanie drugiego progu” zagadnienie progowania zależnego od czasu. Jest to także rozwiązanie zaproponowane przez autora, normalnie nie występujące w literaturze. W momencie, gdy znany jest charakter przetwarzanego sygnału można świadomie wyodrębnić bardziej, lub mniej interesujące fragmenty sygnału i na tej podstawie określić odpowiadające im wartości progów. Sposób ten pozwala na optymalną kompresję zachowującą wyłącznie istotne cechy sygnału. Dodatkowo bogaty interfejs graficzny programu pozwala na łatwe i wygodne ustawienie zarówno zakresów i wartości progów.

Autor starał się możliwie wyczerpująco przedstawić zagadnienia transformacji falkowej, kładąc nacisk na praktyczną realizację przetwarzania sygnałów EKG. Większość z tych tematów wartych jest, by poświęcić im jeszcze więcej uwagi. W szczególności dobór właściwej falki, czy dwa wyżej wspomniane problemy są bardzo ciekawe i wciąż do końca nie wyjaśnione, w związku z czym zasługujące na dalszą kontynuację. Załączona aplikacja zwiększa atrakcyjność niniejszej pracy, dając możliwość samodzielnego pogłębiania wiedzy. Program ten może służyć także jako cenna pomoc w ćwiczeniach laboratoryjnych, stanowiąc praktyczne poparcie podczas omawiania zagadnień transformacji falkowej.

Generalnie program jest dokończoną i w pełni funkcjonującą aplikacją. Mimo to istnieje oczywiście możliwość udoskonalania istniejących i dodawania nowych procedur wraz z postępem wiedzy na temat falek. Dlatego też w załączniku zamieszczono wydruki istotniejszych procedur wraz z opisem w celu ułatwienia zrozumienia i weryfikacji zastosowanych algorytmów.

8. Literatura

- [1] Jan T. Białasiewicz – *Falki i aproksymacje* – WNT Warszawa 2000
- [2] W. Rakowski – *Metoda falkowa kompresji obrazów cyfrowych* – ????????
- [3] W. Rakowski – *Reprezentacja falkowa obrazów cyfrowych – teoria i zastosowania* – ????
- [4] T. Zieliński – *Analiza czasowo-częstotliwościowa sygnałów biomedycznych* – II Sympozjum Modelowanie i Pomiary w Medycynie Krynica 2000
- [5] Randy K. Young – *Wavelet theory and its applications* – Kluwer Academic Publishers 1993
- [6] M. Kowalewski, D. Bochen, M. Urban – *Ocena przydatności analizy falkowej w interpretacji zmienności rytmu serca* – referat
- [7] K. Daqrouq – *Zastosowanie dyskretnej transformaty wavelet do obróbki sygnału EKG* – referat
- [8] B. Bradie – *Wavelet Packet-Based Compression of Single Lead ECG* – IEEE Transactions on Biomedical Engineering, vol. 43, no. 5. May 1996
- [8] G. Antonioli – *EEG Data Compression Techniques* – IEEE Transactions on Biomedical Engineering, vol. 44, no. 2. February 1997
- [9] R. Lyons – *Wprowadzenie do cyfrowego przetwarzania sygnałów* - ????
- [10] Szabatin – *Teoria sygnałów* - ????
- [11] M. Pasko, J. Walczak – *Teoria sygnałów* – Wydawnictwo Politechniki Śląskiej Gliwice 1999
- [12] S. Bober, B. Dąbrowska, A. Dąbrowski – *Elektrokardiografia praktyczna* – Państwowy Zakład Wydawnictw Lekarskich Warszawa 1978
- [13] B. I A. Dąbrowscy – *Podręcznik elektrokardiografii* – Państwowy Zakład Wydawnictw Lekarskich Warszawa 1993
- [14] J. Kwoczyński – *Elektrokardiografia* – Państwowy Zakład Wydawnictw Lekarskich Warszawa 1972
- [15] J. Grębosz – *Symfonia C++* – Oficyna Kallimach Kraków 1996
- [16] J. Grębosz – *Pasja C++* – Oficyna Kallimach Kraków 1997
- [17] Charles Petzold – *Programowanie Windows* – Wydawnictwo RM Warszawa 1999
- [18] M. J. Young – *Visual C++ 6* – Sybex 1998
- [19] MATLAB – *Wavelet Toolbox User's Guide* – The MathWorks, Inc., Natick, MA 1997

9. Załącznik

W niniejszym załączniku zamieszczone są wydruki ważniejszych procedur programu.

9.1. Funkcja „_Convolute”

Funkcja oblicza spłot dwóch wektorów.

Parametry wejściowe:

- signal – wektor sygnału
- l_signal – długość wektora sygnału
- filter – wektor filtru
- l_filter – długość wektora filtru

Parametry wyjściowe:

- result – wektor zawierający wynik operacji spłotu

```
void CEkgDoc::_Convolute(float signal[], int l_signal, float filter[], int l_filter, float result[])
{
    int l_result=l_signal+l_filter-1;           // długość wektora wynikowego
    for(int r=0; r<l_result; r++)              // jeden obieg tej pętli oblicza wartość jednego punktu wektora
    {
        result[r]=0;                          // 'result[]'
        for(int f=0; f<l_filter; f++)          // poszukiwanie par do mnożenia po wektorze filtru
            for(int s=0; s<l_signal; s++)      // i po wektorze sygnału
            {
                if(r==f+s)                    // warunek sparowania punktów do pomnożenia
                {
                    result[r]+= filter[f] * signal[s]; // operacja mnożenia
                    break;                      // wyjście do pętli po wektorze filtru
                }
            }
    }
    return;
}
```

9.2. Funkcja „_AddExtensions”

Funkcja rozszerza wektor i wypełnia dodane punkty na podstawie pięciu różnych algorytmów: ZERO, LIMIT, SMOOTH, SYMETRIC, PERIODIC. Wektor ten jest rozszerzany na podstawie długości aktualnie ustawionego filtru.

Parametry wejściowe:

- Przebieg – wektor sygnału przeznaczonego do rozszerzenia
- PrzebiegDl – długość wektora sygnału

Parametry wyjściowe:

- Przebieg – wektor rozszerzony znajduje się w tym samym miejscu w pamięci

```
void CEkgDoc::_AddExtensions(float Przebieg[], int PrzebiegDl)
{
    float *wsk_left=Przebieg;
    float *wsk_right=&Przebieg[PrzebiegDl-1];
}
```

```

switch(ExtensionType)
{
case ZERO: { // rozszerzenie sygnału wypełniane jest zerami
    for(int i=0;i<FilterLength-1;i++)
    {
        wsk_left--;
        *(wsk_left)=0;
        wsk_right++;
        *(wsk_right)=0;
    }
    break;}
case LIMIT: { // wypełnienie wartościami granicznymi
    float left = *wsk_left;
    float right= *wsk_right;
    for(int i=0;i<FilterLength-1;i++)
    {
        wsk_left--;
        *(wsk_left)=left;
        wsk_right++;
        *(wsk_right)=right;
    }
    break;}
case SMOOTH: { // wypełnienie punktami tworzącymi styczną do krawędzi wykresu
    float left_difference= *wsk_left - *(wsk_left+1);
    float right_difference= *wsk_right - *(wsk_right-1);
    for(int i=0;i<FilterLength-1;i++)
    {
        wsk_left--;
        *wsk_left= *(wsk_left+1) + left_difference;
        wsk_right++;
        *wsk_right=*(wsk_right-1) + right_difference;
    }
    break;}
case SYMETRIC: { // wypełnienie symetrycznymi wartościami wzgl. krawędzi wykresu
    for(int i=0;i<FilterLength-1;i++)
    {
        *(wsk_left-1-i) = *(wsk_left+i);
        *(wsk_right+1+i)= *(wsk_right-i);
    }
    break;}
case PERIODIC: { // założenie okresowości sygnału o okresie R-R i wypełnienie wartościami
                // z następnego (poprzedniego) okresu

    int start_i=PrzebiegDI/30; // wyznaczenie wartości kroków przeszukiwania na podstawie
    int end_i=PrzebiegDI/2; // długości całego sygnału
    int step_y=PrzebiegDI/200;
    if(step_y==0) step_y=1;
    int counter_max=PrzebiegDI/5;

    double MinDifference=100000, difference;
    int best_sample=0;
    int counter=0;

    // przeszukiwanie z lewej strony (na początku sygnału)
    for(int i=start_i; i<end_i;i++)
    {
        difference=0;

```

```

for(int j=0; j<10;j++)
    // szukanie największego podobieństwa do sąsiedniego QRS na podstawie minimalnej
    // różnicy średniokwadratowej
    difference+= fabs((double)( *(wsk_left+i*j*step_y)- *(wsk_left+j*step_y) ));
if(MinDifference>difference)
{
    MinDifference=difference;
    best_sample=i;
    counter=0;
}
else
{
    counter++;
    if(counter>counter_max) break;
}
}
// wypełnienie próbek z lewej strony
for(i=0;i<FilterLength-1;i++)
    *(wsk_left-1-i) = *(wsk_left-1-i+best_sample);

MinDifference=100000;
counter=0;

// przeszukiwanie z prawej strony (na końcu sygnału)
for(i=start_i; i<end_i;i++)
{
    difference=0;
    for(int j=0; j<10;j++)
        difference+= fabs((double)( *(wsk_right-i-j*step_y)- *(wsk_right-j*step_y) ));
    if(MinDifference>difference)
    {
        MinDifference=difference;
        best_sample=i;
        counter=0;
    }
    else
    {
        counter++;
        if(counter>counter_max) break;
    }
}
// wypełnienie próbek z prawej strony
for(i=0;i<FilterLength-1;i++)
    *(wsk_right+1+i) = *(wsk_right+1+i-best_sample);

break;}
}
}

```

9.3. Funkcja „_Dwt”

Funkcja wykonuje wielopoziomą dekompozycję falkową sygnału. Korzysta z funkcji „_AddExtensions” oraz „_Convolute”.

Parametry wejściowe:

- Przebieg – wektor sygnału przeznaczonego do dekompozycji
- PrzebiegDl – długość wektora sygnału

- LevelDekomp – ilość poziomów dekompozycji

Parametry wyjściowe:

- DataResult – tablica zawierająca wektory detali kolejno od pierwszego do n-tego poziomu i na końcu wektor aproksymacji n-tego poziomu (gdzie $n = \text{LevelDekomp}$)
- LengthResult – tablica zawierająca długości wektorów w tablicy powyższej umieszczone w takiej samej kolejności, jak wyżej

```
void CEkgDoc::_Dwt(float DataResult[], int LengthResult[], int *Przebieg, int PrzebiegDI, int LevelDekomp)
{
    CWaitCursor Klepsydra;      // wyświetla kursor klepsydry

                                // tablica wykorzystywana do obliczeń konwolucji
                                // tutaj jest umieszczany wynik konwolucji
    float *calculation_ext = new float[PrzebiegDI+3*(FilterLength-1)];
                                // wskaźnik pokazujący na powyższy sygnał z wyłączeniem rozszerzenia
    float *calculation=calculation_ext+(FilterLength-1);

                                // przechowuje sygnał poddawany procesowi konwolucji
    float *signal_ext = new float[PrzebiegDI+2*(FilterLength-1)];
                                // wskaźnik pokazujący na powyższy sygnał z wyłączeniem rozszerzenia
    float *signal=signal_ext+(FilterLength-1);

                                // wskaźnik pokazujący, w które miejsce DataResult[] kopiować efekt
                                // jednego poziomu dekompozycji falkowej
    float *result_addr=DataResult;
                                // wskaźnik pokazujący, w które miejsce LengthResult[] kopiować
                                // długość sygnału będącego wynikiem jednego poziomu dekompozycji
    int *l_result_addr=LengthResult;
    *l_result_addr=PrzebiegDI;      // pierwsze miejsce zawiera dl. oryginalnego sygnału

                                // skopiowanie oryginalnego przebiegu
    for(int i=0; i<PrzebiegDI; i++)
        *(signal+i)= (float)*(Przebieg +i);

    for (i=1; i<=LevelDekomp; i++)      // kolejne poziomy procesu dekompozycji
    {
        int signal_length= *l_result_addr;
        int signal_ext_length= *l_result_addr+2*(FilterLength-1);

        _AddExtensions( signal, signal_length);

                                // konwolucja filtrem górnoprzepustowym
        _Convolute( signal_ext, signal_ext_length, HiF_D, FilterLength, calculation_ext);
                                // dalej interesuje nas sygnał bez rozszerzeń o długości
        int length= *l_result_addr+(FilterLength-1);      // o tyle próbek konwolucja wydłuża sygnał
        length/=2;      // po downsamplingu
                                // kopiowanie co drugiej próbki (czyli nieparzyste) do bufora przeznaczenia
        for(int j=0; j<length; j++)
            *(result_addr+j)= *(calculation+(2*j+1));
        *(++l_result_addr)=length;      // skopiowanie dl. sygnału do tablicy 'LengthResult'
        result_addr+=length;      // przesunięcie wskaźnika o tą długość w 'DataResult'

                                // konwolucja filtrem dolnoprzepustowym
    }
}
```

```

_Convolute( signal_ext, signal_ext_length, LoF_D, FilterLength, calculation_ext);
if(i==LevelDekomp) // ostatni poziom dekompozycji
{
    for(int j=0; j<length; j++)
        *(result_addr+j)= *(calculation+(2*j+1)); // do docelowej tablicy

    *(++l_result_addr)=length; // skopiowanie dl. sygnału do tablicy 'LengthResult'
}
else
{
    for(int j=0; j<length; j++)
        *(signal+j)= *(calculation+(2*j+1));
}
}
delete [] calculation_ext; // zwolnienie pamięci wykorzystywanej przez tablice robocze
delete [] signal_ext;
}

```

9.4. Funkcja „_InvertDwt”

Funkcja wykonuje wielopoziomą rekonstrukcję falkową sygnału. Korzysta z funkcji „_UpCoef”.

Parametry wejściowe:

- DataIn – tablica zawierająca wektory współczynników szczegółów kolejno od pierwszego do n-tego poziomu i na końcu wektor współczynników aproksymacji n-tego poziomu (gdzie $n = \text{LevelDekomp}$)
- LengthIn – tablica zawierająca długości wektorów w ‘DataIn’
- LevelDekomp – ilość poziomów dekompozycji

Parametry wyjściowe:

- PrzebiegOut – wektor będący wynikiem rekonstrukcji falkowej współczynników zamieszczonych w ‘DataIn’

```

void CEkgDoc::_InvertDwt(float DataIn[], int LengthIn[], float *PrzebiegOut, int LevelDekomp)
{
    CWaitCursor Klepsydra; // wyświetla kursor klepsydry

    // przechowuje Approximation przed konwolucja
    float *dataApr = new float[LengthIn[1]];
    // przechowuje Approximation po konwolucji
    float *calcApr = new float[2*LengthIn[1]-1+(FilterLength-1)];
    // przechowuje Detail przed konwolucją
    float *dataDet;
    // przechowuje Detail po konwolucji
    float *calcDet = new float[2*LengthIn[1]-1+(FilterLength-1)];

    // służy do pokazywania na długość danej tablicy współczynników
    int *length=&LengthIn[LevelDekomp+1]; // ustawienie na koniec tablicy (czyli dl. wsp. aproksymacji)

    float *wsk;

```

```

wsk=DataIn;

for(int i=1; i<=LevelDekomp; i++)
    wsk+=LengthIn[i]; // pokazuje na współczynniki aproksymacji
for(i=0; i< *length; i++)
    *(dataApr+i) = *(wsk+i); // skopiowanie współczynników do tablicy 'dataApr'

dataDet=DataIn;
for(i=1; i<LevelDekomp; i++)
    dataDet+=LengthIn[i]; // pokazuje na współczynniki detali

for(int poziom=LevelDekomp; poziom>=1; poziom--)
{
    length--; // wskaźnik pokazuje na odpowiednią dl. poziomu detali
                // wydobycie konturów
    _UpCoef(dataApr, *length, calcApr, *(length-1), APROXIM);
                // oraz detali
    _UpCoef(dataDet, *length, calcDet, *(length-1), DETAIL);

                // skrócenie sygnałów po konwolucjach aby miały dl. jak ten z poziomu wyżej
    int shift= ( ( 2* (*length) + FilterLength-2 ) - *(length-1) )/2;
    float *wskApr=calcApr+shift;
    float *wskDet=calcDet+shift;

    if(poziom==1) // ostatni poziom kopiowany jest na wyjście
    {
        MinMaxExtr[0]=10000;
        MinMaxExtr[1]=0;

        for(i=0; i< *(length-1); i++)
        {
            *(PrzebiegOut+i)= *(wskApr+i) + *(wskDet+i);
            if(i>200 && i<1500){ // znajduje najmniejszą
                if(*(PrzebiegOut+i)<MinMaxExtr[0]) MinMaxExtr[0]=*(PrzebiegOut+i); // i największą wartość sygnału
            }
            if(*(PrzebiegOut+i)>MinMaxExtr[1]) MinMaxExtr[1]=*(PrzebiegOut+i);}
        }
        DisplayMultiplay=1;
    }
    else
    {
        // sygnał aproksymacji z poziomu wyżej powstaje z sumy dwóch poniższych
        for(i=0; i< *(length-1); i++)
            *(dataApr+i)= *(wskApr+i) + *(wskDet+i);

        // sygnał detali z poziomu wyżej jest dostarczony
        // przesunięcie jedynie wskaźnika tak, aby pokazywał na zadane współczynniki detali
        dataDet-= *(length-1);
    }
}
delete [] dataApr;
delete [] calcApr;
delete [] calcDet;
}

```

9.5. Funkcja „_ExtractSignal”

Funkcja wykonuje rekonstrukcję falkową współczynników z pojedynczego poziomu dekompozycji sygnału. Rekonstruuje wybrany poziom detali lub aproksymacji. Korzysta z funkcji „_UpCoef”.

Parametry wejściowe:

- DataIn – tablica zawierająca wektory współczynników szczegółów kolejno od pierwszego do n-tego poziomu i na końcu wektor współczynników aproksymacji n-tego poziomu (gdzie $n = \text{LevelDekomp}$)
- LengthIn – tablica zawierająca długości wektorów w ‘DataIn’
- LevelDekomp – ilość poziomów dekompozycji
- LevelNumber – poziom, którego dotyczy rekonstrukcja

Parametry wyjściowe:

- PrzebiegOut – wektor będący wynikiem rekonstrukcji falkowej danego poziomu współczynników zamieszczonych w ‘DataIn’

```
void CEkgDoc::_ExtractSignal(float DataIn[], int LengthIn[], float *PrzebiegOut, int LevelDekomp, int LevelNumber)
```

```
{
    CWaitCursor Klepsydra; // wyświetla kursor klepsydry

    // dwa robocze bufory do obliczania rekonstrukcji
    float *buffor_1 = new float[2*LengthIn[1]-1+(FilterLength-1)];
    float *buffor_2 = new float[2*LengthIn[1]-1+(FilterLength-1)];
    // wskaźniki do poruszania się po tych buforach

    float *dataWsk = buffor_1;
    float *calcWsk = buffor_2;

    dataWsk=DataIn;
    for(int i=1; i<LevelNumber; i++)
        dataWsk+=LengthIn[i];

    int *length;
    int poziom_dek;
    filter_filtr_type;

    if(LevelNumber>LevelDekomp) // rekonstrukcja dotyczy współczynników aproksymacji
    {
        filtr_type=APROXIM;
        length=&LengthIn[LevelDekomp]; // ustawienie na ostatni poziom współczynników detali
        poziom_dek=LevelDekomp;
    }
    else
    {
        filtr_type=DETAIL;
        length=&LengthIn[LevelNumber]; // ustawienie na odpowiedni poziom współczynników detali
        poziom_dek=LevelNumber;
    }
}
```



```

int licznik=0;

// pętla dokonująca rekonstrukcji na kolejnych poziomach
for(int poziom=poziom_dek; poziom>=1; poziom--)
{
    _UpCoef(dataWsk, *length, calcWsk, *(length-1), filtr_type);

    int shift= ( ( 2* (*length) + FilterLength-2 ) - *(length-1) )/2;
    dataWsk=calcWsk+shift;

    if(licznik%2) calcWsk=buffer_2;
    else calcWsk=buffer_1;

    licznik++;
    length--;
    filtr_type=APROXIM;
}

MinMaxExtr[0]=10000;
MinMaxExtr[1]=0;
// poszukiwanie maksimum i minimum otrzymanego przebiegu
for(i=0; i<LengthIn[0]; i++)
{
    *(PrzebiegOut+i)=*(dataWsk+i);
    if(i>200 && i<1500){
        if(*(dataWsk+i)<MinMaxExtr[0]) MinMaxExtr[0]=*(dataWsk+i); // znajduje najmniejsza
        if(*(dataWsk+i)>MinMaxExtr[1]) MinMaxExtr[1]=*(dataWsk+i);} // i najw. wart sygnalu
    }

    // na podstawie maksimum i minimum wyznaczenie współczynnika podziału
    // wykorzystywanego przy rysowaniu otrzymanego przebiegu
float height=MinMaxExtr[1]-MinMaxExtr[0];
DisplayMultiplay=1;
if(height<160) DisplayMultiplay=2;
if(height<80) DisplayMultiplay=4;
if(height<40) DisplayMultiplay=8;
if(height<20) DisplayMultiplay=16;

delete [] buffer_1;
delete [] buffer_2;
}

```

9.6. Funkcja „_UpCoef”

Funkcja realizuje proces ‘upsamplingu’, czyli dwukrotnego zwiększenia długości sygnału poprzez wstawienia między wszystkimi próbkami próbek zerowych. Funkcja wykonuje również operację splotu z odpowiednim filtrem. Korzysta z funkcji „_Convolute”.

Parametry wejściowe:

- DataIn – wektor sygnału wejściowego
- LengthIn – długość wektora ‘DataIn’
- Filtr_type – rodzaj filtra użytego do operacji splotu z sygnałem ‘DataIn’

Parametry wyjściowe:

- DataOut – wektor wyjściowy otrzymany po operacji ‘upsamplingu’ i ‘konwolucji’
- LengthOut – długość wektora wyjściowego

```
void CEkgDoc::_UpCoef(float DataIn[], int LengthIn, float DataOut[], int LengthOut, filter_filtr_type)
{
    float *upsampled = new float[2*LengthIn];    // wykorzystana będzie dl. [2*LengthIn-1]

    for(int i=0; i<LengthIn; i++)                // operacja upsamplingu
    {
        *(upsampled+2*i)=DataIn[i];
        *(upsampled+2*i+1)=0;
    }

    switch(filtr_type)                           // splot (filtracja) właściwym filtrem
    {
    case APROXIM:
        _Convolute( upsampled, 2*LengthIn-1, LoF_R, FilterLength, DataOut);
        break;
    case DETAIL:
        _Convolute( upsampled, 2*LengthIn-1, HiF_R, FilterLength, DataOut);
        break;
    }
    delete [] upsampled;
}
}
```

9.7. Funkcja „_TruncateCoeff”

Funkcja zeruje współczynniki falkowe o wartości niższej, niż ustalony próg podczas procesu kompresji.

Parametry wejściowe:

- brak

Parametry wyjściowe:

- zwraca zawartość procentową wyzerowanych współczynników spośród wszystkich wykorzystywanych w procesie rekonstrukcji

```
float CEkgDoc::_TruncateCoeff()
{
    float *source_wsk=CoeffNoTrunc;             // wskaźnik na bufor źródłowy
    float *dest_wsk=ComprEkg.SignalData;       // wskaźnik na bufor docelowy
    int i,j;
    float tr_number=0;                          // ilość obciętych współczynników

    if(SecondThreshold)                        // jeśli aktywne dwa progi obcinania
    {
        int *ThrChange = new int[NumberOfBrackets*2+1]; // liczba zmian progu obcinania
        int NextThrChange=0;
        float _threshold;
        BOOL thr2_flag=0;
    }
}
```

```

for(i=1; i<=ComprEkg.DecLevel; i++) // pętla po poziomach dekompozycji
{
    thr2_flag=0;
    _threshold=ThresholdTab[i];
    NextThrChange=0;

    for(j=0; j<NumberOfBrackets; j++) // wylicza położenie krawędzi przedziałów
        // rozgraniczających pierwszy i drugi próg
        // na kolejnych poziomach dekompozycji sygnału
        {
            ThrChange[2*j]=Bracket[j+1].a*ComprEkg.SignalLength[i]/QrsLength;
            ThrChange[2*j+1]=Bracket[j+1].b*ComprEkg.SignalLength[i]/QrsLength;
        }
    ThrChange[NumberOfBrackets*2]=ComprEkg.SignalLength[i]; // ostatni (koniec sygnału)

    for(j=0; j<ComprEkg.SignalLength[i]; j++) // pętla po współczynnikach w obrębie jednego
        // poziomu dekompozycji
        {
            if(j>=ThrChange[NextThrChange])
            {
                thr2_flag=!thr2_flag;
                if(thr2_flag) _threshold=Threshold_2Tab[i];
                else _threshold=ThresholdTab[i];
                NextThrChange++;
            }

            if( *source_wsk> _threshold || *source_wsk< -_threshold )
                *dest_wsk= *source_wsk;
            else
            {
                *dest_wsk=0;
                if(ActualLevel==0) // gdy pełna rekonstrukcja zliczane wszystkie współczynniki
                    tr_number++;
                else if(i==ActualLevel) // zliczane tylko dla konkretnego poziomu detali
                    tr_number++;
            }

            source_wsk++;
            dest_wsk++;
        }
    }

    delete [] ThrChange;
}
else // aktywny jeden próg obcinania
{
    // obcinamy współczynniki detali
    for(i=1; i<=ComprEkg.DecLevel; i++)
    {
        for(j=0; j<ComprEkg.SignalLength[i]; j++)
        {
            if( *source_wsk>ThresholdTab[i] || *source_wsk< -ThresholdTab[i] )
                *dest_wsk= *source_wsk;
            else
            {
                *dest_wsk=0;
                if(ActualLevel==0) // gdy pełna rekonstrukcja zliczane wszystkie współczynniki
                    tr_number++;
                else if(i==ActualLevel) // zliczane tylko dla konkretnego poziomu detali
                    tr_number++;
            }
        }
    }
}

```

```

        tr_number++;
    }

    source_wsk++;
    dest_wsk++;
}
}

// współczynniki aproksymacji nie są zerowane
for(j=0; j<ComprEkg.SignalLength[ComprEkg.DecLevel+1]; j++)
{
    *dest_wsk= *source_wsk;
    source_wsk++;
    dest_wsk++;
}

int length=0;
if(ActualLevel==0) // zliczanie wszystkich współczynników
    for(i=1; i<=ComprEkg.DecLevel+1; i++)
        length+=ComprEkg.SignalLength[i];
else
    length=ComprEkg.SignalLength[ActualLevel]; // tylko współczynniki konkretnego poziomu

tr_number/=length; // dzielenie współczynników obciętych przez wszystkie
// w obrębie interesującego poziomu

return tr_number*100; // zwracany jest stosunek w procentach (zawartość procentowa zer)
}

```

9.8. Funkcja „_SetWaveletFace”

Funkcja na podstawie wartości filtrów kwadraturowych oblicza odpowiadający im kształt falki lub funkcji skalującej i rysuje na pulpicie kontrolnym programu.

Parametry wejściowe:

- brak

Parametry wyjściowe:

- brak

```

void CEkgDoc::_SetWaveletFace()
{
    CWaitCursor Klepsydra; // wyświetla kursor klepsydry
    int L = FilterLength;
    int BufLength= (((L*2+L-2)*2+L-2)*2+L-2)*2+L-2;

    float *buffor_1 = new float[BufLength]; // bufor na obliczenia
    float *dataWsk = buffor_1; // wskaźnik do poruszania się po buforze
    float *calcWsk = WaveletFace; // 'WaveletFace' jest docelowym buforem
    // przechowującym dane do odrysowania

    int i;
    if(Wavelet_view) // decyduje o wyświetlaniu falki lub funkcji skalującej
        for(i=0; i<FilterLength; i++)
            *(dataWsk+i)=HiF_R[i]; // dla falki podstawiany jest filtr górnoprzepustowy rekonstrukcji
}

```

```
else
    for(i=0; i<FilterLength; i++)
        *(dataWsk+i)=LoF_R[i];    // dla funkcji skalującej filtr dolnoprzepustowy rekonstrukcji

int length_source;
int length_dest=FilterLength;

int licznik=0;

for(i=0; i<5; i++)                // pięciokrotna rekonstrukcja współczynników odpowiedniego filtru
                                    // pozwoli uzyskać żądany kształt
{
    length_source=length_dest;
    length_dest=length_dest*2+L-2;

    _UpCoef(dataWsk, length_source, calcWsk, length_dest, APROXIM);

    dataWsk=calcWsk;

    if(licznik%2) calcWsk=WaveletFace;
    else        calcWsk=buffer_1;

    licznik++;
}

delete [] buffer_1;

                                    // unieważnienie obszaru na pulpicie sterowania pozwoli na odrysowanie
                                    // kształtu falki przechowywanego w 'WaveletFace'
CMainFrame * pFrame = (CMainFrame *) (AfxGetApp()->m_pMainWnd);
if(!pFrame) return;
CRect rect(0,0,200,80);
pFrame->m_wndDlgBar.InvalidateRect(&rect);
}
```