

Politechnika Śląska  
Wydział Automatyki, Elektroniki i Informatyki  
Kierunek Informatyka

Rozprawa doktorska

Wielorozdzielcza reprezentacja  
obrazów realizowana według  
schematu liftingu z aproksymacją  
metodą najmniejszych kwadratów  
w bloku predykcji

Marcin Jaromin

Promotor: prof. dr hab. inż. Konrad Wojciechowski

Gliwice, 2010

Panu Profesorowi Konradowi Wojciechowskiemu  
za życzliwość, wsparcie merytoryczne  
oraz cenne uwagi udzielane podczas pisania pracy  
składam serdeczne podziękowania

Pracę dedykuję mojej żonie i dzieciom  
za ich cierpliwość, miłość i wyrozumiałość  
w czasie zgłębiania przeze mnie tematyki falkowej,  
jak i podczas dalszych badań i analiz,  
które doprowadziły do powstania niniejszej rozprawy

---

# Spis treści

---

<b>Spis treści</b>	<b>3</b>
<b>I Wprowadzenie</b>	<b>7</b>
1 Cel i zakres pracy	9
<b>II Analiza falkowa ortogonalnych funkcji ciągłych</b>	<b>17</b>
2 Wstęp do przestrzeni ortogonalnych	19
3 Falki ortogonalne i równania definiujące	23
4 Falki o nośniku zwartym	27
4.1 Stopnie swobody i niezmienniczość względem translacji . . . . .	28
4.2 Momenty znikające i falki Daubechies . . . . .	29
5 Obliczanie współczynników filtra dolno- i górnoprzepustowego	31
6 Wyznaczanie funkcji skalującej o zwartym nośniku	33
6.1 Algorytm Stranga . . . . .	33
6.2 Macierze przekształceń - postać ogólna . . . . .	37
6.3 Implementacja algorytmu Stranga . . . . .	39
6.3.1 Przygotowywanie danych . . . . .	39
6.3.2 Realizacja algorytmu Stranga . . . . .	39
6.4 Wykreślanie funkcji skalującej . . . . .	40
6.4.1 Przygotowanie wyników . . . . .	40
6.4.2 Graficzna prezentacja wyników . . . . .	41
7 Wyznaczanie funkcji falkowej o zwartym nośniku	43
8 Przybliżone wyznaczanie wartości funkcji skalującej	45
8.1 Algorytm kaskadowy - wyznaczanie wartości w dziedzinie czasu . . . . .	45
8.1.1 Zbieżność algorytmu kaskadowego . . . . .	45
8.1.2 Wyznaczanie funkcji skalującej metodą iteracyjną . . . . .	48
8.2 Algorytm kaskadowy - wyznaczanie wartości w dziedzinie częstotliwościowej . . . . .	51



<b>III Analiza wielorozdzielcza z wykorzystaniem przestrzeni ortogonalnych</b>	<b>55</b>
<b>9 Analiza wielorozdzielcza</b>	<b>57</b>
9.1 Skalowanie funkcji . . . . .	57
9.2 Własności analizy wielorozdzielczej . . . . .	57
9.3 Falkowa analiza wielorozdzielcza . . . . .	60
9.3.1 Aproksymacja za pomocą falek o nośniku zwartym . . . . .	60
9.3.2 Aproksymacja falkowa funkcją Haara . . . . .	61
9.4 Aproksymacja Haara - implementacja . . . . .	64
9.5 Algorytm Mallata . . . . .	66
9.6 Obliczanie przybliżonej wartości początkowej współczynników falkowych . . . . .	69
9.7 Dwuwymiarowa analiza wielorozdzielcza . . . . .	70
<b>10 Dyskretna transformata falkowa (DWT)</b>	<b>73</b>
<b>IV Analiza wielorozdzielcza z wykorzystaniem przestrzeni biortogonalnych</b>	<b>79</b>
<b>11 Własności biortogonalnych baz falkowych</b>	<b>81</b>
<b>12 Reprezentacja polifazowa</b>	<b>85</b>
<b>13 Schemat liftingu</b>	<b>89</b>
13.1 Wyznaczanie nowych biortogonalnych filtrów falkowych . . . . .	89
13.2 Algorytm Euklidesa . . . . .	92
13.3 Faktoryzacja za pomocą schematu liftingu . . . . .	94
13.4 Przykład realizacji schematu liftingu . . . . .	95
13.4.1 Analiza biortogonalnych falek CDF(2,2) (Cohen-Daubechies-Feauveau) . . . . .	95
13.4.2 Faktoryzacja falek Haara . . . . .	99
13.4.3 Rotacja Givensa . . . . .	100
13.4.4 Skalowanie . . . . .	101
13.5 Biortogonalne falki CDF . . . . .	101
13.6 Interpolacja funkcji za pomocą biortogonalnej transformaty falkowej . . . . .	103
13.7 Wyznaczanie współczynników predykcji dla filtrów interpolowanych . . . . .	107
13.8 Implementacja interpolacji funkcji z wykorzystaniem schematu liftingu . . . . .	109
13.8.1 Predyktor rzędu I - Haara . . . . .	110
13.8.2 Predyktor rzędu II - funkcja liniowa . . . . .	111
13.8.3 Predyktor rzędu IV - wielomian stopnia trzeciego . . . . .	112
13.8.4 Predyktor rzędu IV - interpolacja funkcji $\sin(x)$ wielomianem stopnia trzeciego . . . . .	113
13.9 Wyznaczanie współczynników uaktualnienia dla funkcji periodycznych . . . . .	115
<b>V Schemat liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów w bloku predykcji</b>	<b>119</b>
<b>14 Aproksymacja funkcji metodą najmniejszych kwadratów w bloku predykcji</b>	<b>121</b>
14.1 Wektor predykcji określany za pomocą aproksymacji wielomianowej funkcji . . . . .	124
14.2 Wektor predykcji określany za pomocą interpolacji wielomianowej funkcji . . . . .	125
14.3 Problem złego uwarunkowania macierzy współczynników układu . . . . .	126
14.4 Funkcje bazowe - wielomiany ortogonalne . . . . .	127

14.4.1	Własności wielomianów czynnikowych . . . . .	127
14.4.2	Wielomiany ortogonalne . . . . .	128
14.4.3	Wielomiany ortogonalne Grama . . . . .	130
14.5	Aproksymacja trygonometryczna . . . . .	133
14.6	Implementacja aproksymacji MNK w bloku predykcji z wykorzystaniem baz ortogonalnych . . . . .	135
14.6.1	Implementacja aproksymacji MNK w bloku predykcji za pomocą wielomianów ortogonalnych Grama . . . . .	136
14.6.2	Implementacja aproksymacji MNK w bloku predykcji za pomocą wielomianów trygonometrycznych . . . . .	139
<b>15</b>	<b>Relacja pomiędzy dyskretną transformatą falkową a transformatą realizowaną przy użyciu schematu liftingu</b>	<b>143</b>
<b>16</b>	<b>Zastosowanie schematu liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów w bloku predykcji</b>	<b>151</b>
<b>VI</b>	<b>Analiza obrazu za pomocą schematu liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów w bloku predykcji</b>	<b>157</b>
<b>17</b>	<b>Schemat liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów</b>	<b>159</b>
17.1	Redukcja szumu obrazu . . . . .	160
17.2	Usuwanie nieciągłości obrazu . . . . .	164
17.3	Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty dla celów kompresji obrazu . . . . .	171
17.3.1	Model przestrzeni kolorów RGB . . . . .	172
17.3.2	Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty modelu RGB . . . . .	176
17.3.3	Model przestrzeni kolorów YCbCr . . . . .	181
17.3.4	Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty modelu RGB w przestrzeni YCbCr . . . . .	184
<b>VII</b>	<b>Wnioski końcowe</b>	<b>191</b>
<b>18</b>	<b>Podsumowanie</b>	<b>193</b>
<b>19</b>	<b>Kierunki dalszych badań</b>	<b>197</b>
<b>VIII</b>	<b>Dodatki</b>	<b>199</b>
	<b>Bibliografia</b>	<b>201</b>
	<b>Spis rysunków</b>	<b>206</b>
	<b>Spis tablic</b>	<b>210</b>
	<b>Lista algorytmów</b>	<b>211</b>



Część I

Wprowadzenie



## Rozdział 1

---

# Cel i zakres pracy

---

Analiza wielorozdzielcza wykorzystująca metody falkowe zajmuje istotną rolę w cyfrowej analizie i przetwarzaniu funkcji jedno lub wielowymiarowych.

Przekształcenia tego typu ze względu na swoją atrakcyjność i możliwość dopasowania do zbioru próbek badanego układu znalazły swoje zastosowanie m. in. podczas badania nieciągłości funkcji, usuwania szumu czy też w algorytmach związanych z kompresją obrazów [3].

W 1992 roku Tom Hopper z FBI's Criminal Justice Information Services Division oraz Jonathan Bradley i Chris Brislawn z Los Alamos National Laboratory opracowali metodę zastosowania falek do kompresji odcisków palców.

Zmniejszanie pojemności danych oparli o zaawansowane metody progresywnego kwantowania współczynników wykorzystując zależności przestrzenne pomiędzy współczynnikami falkowej reprezentacji obrazu [18].

Pozwoliło to na znaczną oszczędność przestrzeni dyskowej w celu przechowywania ponad 200 milionów zdjęć. [3, 4, 10].

Innym praktycznym przykładem jest film „Toy Story 2”, wyprodukowany w 1995 roku przez Pixar Studios, gdzie duża część brył została zabarwiona w przestrzeniach podpasmowych, co z matematycznego punktu widzenia jest równoważne zastosowaniu falek[34].

Najbardziej znanym, a jednocześnie najczęściej używanym formatem kompresji z zastosowaniem technik falkowych jest format JPEG2000 wprowadzony przez International Standards Organization w roku 1999[34]. Format ten został wprowadzony jako uzupełnienie istniejących, bardzo popularnych technik kompresji JPEG, wykorzystując do celów przetwarzania obrazu jednowymiarową transformatę falkową z filtrami biortogonalnymi CDF(5,3) i CDF(97)[7].

Filtry CDF (rozd. 13.5) zrealizowane za pomocą schematu liftingu, pozwalają na zaprojektowanie odwracalnych transformacji całkowitoliczbowych za pomocą biortogonalnych współczynników falkowych. W konsekwencji błąd rekonstrukcji jest praktycznie zerowy, a wyniki przekształceń zbliżone są do operacji zmiennoprzecinkowych.

Tego typu transformata falkowa ma swoje zastosowanie również w zapisach wideo w kompresji różnicowej kolejnych ramek obrazu [39, 43].

Powyższe praktyczne zastosowanie, jak i zainteresowanie autora dotyczące aspektów falkowych, miało decydujący wpływ na tematykę zawartą w dalszych częściach rozprawy.



Rysunek 1.1: Przykład wykorzystania przekształceń falkowych

Jednym z celów pracy było zaprezentowanie w sposób chronologiczny powstawanie teorii falkowej, zaczynając od przestrzeni ortogonalnych ciągłych, jak i dyskretnych, kończąc na przestrzeni biortogonalnych realizowanych za pomocą schematu liftingu.

W części II została przeprowadzona elementarna analiza falkowa funkcji ciągłych. Poruszone są w niej kwestie ortogonalności funkcji bazowych jak i konsekwencje stosowania ich podczas dalszej analizy.

Wstępne własności przestrzeni ortogonalnych przedstawione za pomocą szeregu Fouriera uwidaczniają problem tzw. nieskończonej energii (rozdz. 2).

Starając się eliminować ten problem, pokazany został ortogonalny układ falkowy, który w dalszym etapie został szczegółowo omówiony (rozdz. 3).

Łącznie z omówieniem ortogonalności układu wprowadzone zostały falki o nośniku zwartym, pozwalając na uzyskanie funkcji falkowych z filtrami o skończonej odpowiedzi impulsowej (rozdz. 4)[75].

Korzystając z powyższej własności, a zarazem narzucając dodatkowe ograniczenia na stopnie swobody filtrów dolno i górnoprzepustowych za pomocą tzw. momentów znikających, zdefiniowano i omówiono dodatkowe własności falek Daubechies (rozdz. 4.2) o maksymalnej liczbie momentów dla danego nośnika.

Rozdziały części II opisują także metody uzyskiwania wartości współczynników filtrów funkcji falkowych i skalujących w sposób dokładny za pomocą algorytmu Stranga dla punktów binarnych (rozdz. 6.1) jak i techniki przybliżone otrzymywane w sposób iteracyjny (rozdz. 8).

Część III jest kontynuacją części II która opisuje podstawowe własności analizy wielorozdzielczej, z wykorzystaniem technik falkowych, w której istotną rolę odgrywa szczegółowo omówiony algorytm Mallata (rozdz. 9.5).

W rozprawie nakreślony został sposób wyznaczania współczynników falkowych i skalujących (rozdz. 9.6), pomijając przy tym pracochłonną i obciążoną dużym błędem zaokrąglenia - operację całkowania.

Ostatni rozdział części III (rozdz. 10) stanowiąc niejako przejście z przestrzeni ciągłych do jej odpowiednika w przestrzeni dyskretnej, prezentuje dyskretną transformatę falkową dla przestrzeni ortogonalnych.

Dyskretyzacja transformaty jest konsekwencją algorytmu Mallata pozwalającego na stosunkowo proste rozdzielanie funkcji dyskretnej na jej reprezentację zgrubną jak i szczegółową. Dzięki takiej zależności dyskretną transformatę falkową można uogólnić na jej odpowiednik w przestrzeniach biortogonalnych.

Wstępem do wyżej wymienionych przestrzeni i ich baz falkowych jest część IV która prezentuje analizę wielorozdzielczą w oparciu o tzw. macierz polifazową (rozdz. 12).

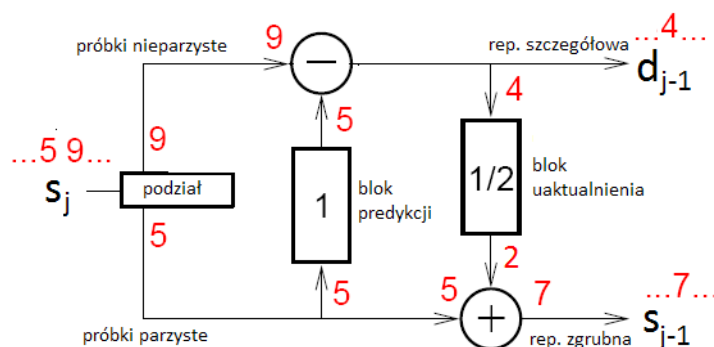
Opisana jest w niej faktoryzacja macierzy przy wykorzystaniu algorytmu Euklidesa (rozdz. 13.2) który pozwala doprowadzić reprezentację polifazową do tzw. postaci predykcyjnej wykorzystującą technikę schematu liftingu.

Schemat ten umożliwia wyodrębnić biortogonalną reprezentację zgrubną i szczegółową wykorzy-

stując zależność pomiędzy parzystymi i nieparzystymi próbkami podlegającymi analizie (rozd. 13).

Podstawową wersją schematu liftingu [72] jest konstrukcja przestrzeni aproksymacji i detali dokonując wstępnego podziału próbek na elementy parzyste i nieparzyste, a następnie przeprowadzając na nich transformację za pomocą dwóch przekształceń zwanych odpowiednio

- blokiem predykcji\*
- blokiem uaktualnienia



Rysunek 1.2: Przykład realizacji schemat liftingu

W bloku predykcji opisanym m. in. w rozdz. 13.6 szacowane są nowe próbki nieparzyste na podstawie próbek parzystych, które po wyliczeniu odejmuje się od oryginalnych próbek nieparzystych wstępnego podziału otrzymując przestrzeń detali  $d$  tzw. reprezentację szczegółową.

Kolejny blok jest operacją uaktualnienia, który jest poniekąd etapem odwrotnym, rekompensującym utratę połowy współczynników aproksymacji podczas wstępnego podziału. Modyfikacja wartości próbek parzystych odbywa się na zasadzie dodania do nich wartości powstałej z liniowej kombinacji współczynników detali tworząc reprezentację zgrubną. Operacja uaktualnienia ma na celu normalizację układu w taki sposób, aby wartość średnia współczynników aproksymacji miała tę samą wartość co wartość średnia próbek przed transformacją.

W ostatnim rozdziale części IV jako wkład własny, autor opisał i udowodnił swoje spostrzeżenia dotyczące zależności wzajemnej współczynników wektora predykcji jak i uaktualnienia dla funkcji periodycznych. Wykazują one, że suma współczynników wektora predykcji jest równa 1, zaś dla uaktualnienia wynosi ona  $\frac{1}{2}$  (rozd. 13.9).

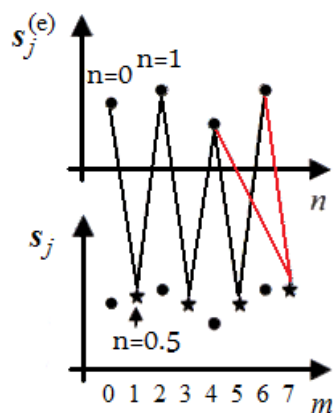
W artykułach [61, 60] dla celów wyznaczania nowych próbek nieparzystych w bloku predykcji Wim Sweldens zaproponował technikę interpolacji wielomianowej, gdzie współczynniki wielomianu oblicza się na podstawie próbek parzystych stanowiące węzły wielomianu interpolacyjnego (rozd. 13.6).

Obliczając próbki detali, nowe wartości próbek nieparzystych przyjmowane są jako wartości punktów należące do funkcji interpolującej, leżące równoodległe pomiędzy próbkami parzystymi.

Na rys. 1.3 przedstawiony jest przykład szacowania próbek nieparzystych ( $\star$ ) z interpolacją wielomianową podczas operacji predykcji, z wykorzystaniem dwóch sąsiadujących, próbek parzystych ( $\bullet$ ). Numer rzędu predyktora określa liczbę próbek używanych w celu wyznaczenia funkcji interpolującej (w opisywanym przypadku rząd predyktora ma wartość II).

\*Pojęcie **predykcji** ma swoje uzasadnienie w terminologii cyfrowego przetwarzania szeregów czasowych tzw. sygnałów. Określa ono przewidywanie wartości nowych próbek nieparzystych na podstawie znajomości próbek parzystych. W matematyce termin określany jest zazwyczaj szacowaniem, jednakże ze względu na aspekty historyczne i rozpowszechnione nazewnictwo określenie **predykcja** będzie używana w dalszej części pracy.





Rysunek 1.3: Blok predykcji dla predyktora rzędu II

Ponieważ, zgodnie z przykładem, wielomianem interpolującym dla dwóch punktów jest wielomian stopnia 1 (funkcja liniowa), w konsekwencji otrzymuje się ogólną zależność, że rząd predyktora jest zawsze o jeden większy od stopnia wielomianu interpolowanego.

Charakterystyczną cechą takiego podejścia, jest to, że do estymacji próbek nieparzystych dla warunków brzegowych, można wykorzystać zależność próbek parzystych najbliższej położonej szacowanej próbki (rozd. 13.6). W klasycznym podejściu do filtracji próbek przyjmuje się okresowy charakter zestawu próbek lub dokłada się próbki na krańcach dziedziny wykorzystując zasadę lustrzanego odbicia.

#### ? SzacowaniePróbekNieparzystych

SzacowaniePróbekNieparzystych[funkcja, zakres, zakresWidoczności]  
 Funkcja prezentująca zastosowanie interpolacji i  
 aproksymacji metoda najmniejszych kwadratów w bloku predykcji schematu liftingu

```
funkcja[x_] := x + RandomReal[0.2] - 0.1;
zakres = {0, 2, 0.2};
zakresWidoczności = {{0, 0.8}, {0.2, 0.8}};
```

Rysunek 1.4: Szacowanie próbek nieparzystych - dane

Podczas wieloletnich badań nad analizą falkową i jej implementacją komputerową dla funkcji jedno i dwuwymiarowych autor zauważył, że funkcja interpolująca nie zawsze pozwala wyliczyć w sposób zadowalający, z odpowiednio zadaną dokładnością, nowe próbki nieparzyste. Duże rozbieżności pomiędzy wartościami próbek nieparzystych funkcji oryginalnej a wartościami próbek szacowanych prowadzą do dużego zaszumienia reprezentacji szczegółów, powodując utratę istotnych informacji (1.5 III).

W pracy, w części V, jako alternatywę dla tego typu rozwiązania, została zaproponowana autorska realizacja bloku predykcji za pomocą aproksymacji metodą najmniejszych kwadratów (w skrócie MNK), która stanowi jednocześnie istotę pracy własnej (rozd. 14).

Aby zobrazować różnicę, pomiędzy aproksymacją MNK a interpolacją na rys. 1.5 zaprezentowane zostały 4 wykresy, przy czym pierwszy z nich (rys. 1.5 I) przedstawia zbiór wszystkich próbek biorących udział w transformacji z wykorzystaniem zarówno interpolacji, jak i aproksymacji funkcji, gdzie

- Punkty ● — czerwone określają próbki parzyste służące do wyznaczenia funkcji interpolującej, czy też aproksymującej podczas bloku predykcji

- Punkty ●—niebieskie reprezentują próbki nieparzyste wstępnego podziału, biorące udział w wyznaczaniu reprezentacji szczegółowej
- Punkty ○—zielone wyznaczają próbki nieparzyste oszacowane w bloku predykcji za pomocą aproksymacji metodą najmniejszych kwadratów (regresji prostej)
- Punkty ○—niebieskie określają próbki nieparzyste oszacowane w bloku predykcji za pomocą interpolacji wielomianowej
- Funkcja wielomianowa  $f(x)$  opisana kolorem niebieskim przedstawia funkcję interpolującą
- Funkcja liniowa  $g(x)$  wykreślona w kolorze zielonym reprezentuje funkcję aproksymacyjną
- Czerwone pionowe linie łączące próbki parzyste z funkcją aproksymującą określają błąd aproksymacji względem punktów parzystych

W celu przejrzystości i opisu istoty problemu w bloku predykcji na rys. 1.5 II zaprezentowany został fragment grupy próbek podlegających analizie.

Jak wynika z reprezentacji graficznej, wyznaczenie nowych próbek nieparzystych (○—niebieskie) za pomocą funkcji interpolującej nie jest gwarancją lepszego oszacowania ich względem próbek nieparzystych (●—niebieskie), oryginalnych (rys. 1.5 II, III).

Na podstawie omawianego przykładu można zauważyć, że wyznaczone nowe próbki za pomocą funkcji aproksymującej (○—zielone) są obciążone mniejszym błędem szacunkowym (predykcji) względem próbek oryginalnych (rys. 1.5 II, IV).

Reasumując, użycie wielomianów interpolujących nie zawsze prowadzi do znalezienia optymalnego rozwiązania problemu.

Ponadto, aproksymacja ze względu na swoje właściwości dotyczące wyznaczania linii trendu bez ograniczenia interpolacyjnego (funkcja aproksymacyjna nie musi przechodzić przez wszystkie węzły badanego układu), pozwala na znalezienie funkcji ( $g(x)$  z rys. 1.5) o znacznie niższym stopniu aniżeli wielomian interpolacyjny ( $f(x)$  z rys. 1.5).

Dodatkową zaletą jest to, że funkcja aproksymacyjna nie musi być wielomianem.

**Część V**, nawiązująca do autorskiego pomysłu wyznaczania predyktora za pomocą aproksymacji MNK, ukazuje również problem tzw. złego uwarunkowania macierzy współczynników układu z jednoczesną propozycją eliminacji problemu (rozdz. 14.3 - 14.5).

W części tej zobrazowana została także relacja pomiędzy klasyczną transformacją falkową a jej realizacją za pomocą schematu liftingu.

Przedstawia ona sposób wyznaczania filtrów górno- i dolnoprzepustowych przy pomocy bloków predykcji i uaktualnienia wraz z przykładowym wykreśleniem funkcji skalującej i falkowej.

Ostatni rozdział omawianej części rozprawy jest rozdziałem podsumowującym analizę funkcji jednowymiarowej, prezentujący sposób wykorzystania schematu liftingu z wykorzystaniem aproksymacji MNK m.in. do celów odszumiania funkcji czy też badania jej nieciągłości.

**Część VI** pracy poświęcona badaniom i analizie obrazu podczas redukcji szumu (rozdz. 17.1), usuwaniu nieciągłości (rozdz. 17.2) i badaniu wrażliwości schematu liftingu z wykorzystaniem aproksymacji MNK na zerowanie współczynników transformaty dla celów kompresji obrazu (rozdz. 17.3).

Podczas badania wrażliwości na zerowanie współczynników przeprowadzono analizę eliminacji progowej modelu przestrzennego kolorów RGB jak i YCbCr z wykazaniem ich różnic w celach optymalizacyjnych (rozdz. 17.3).

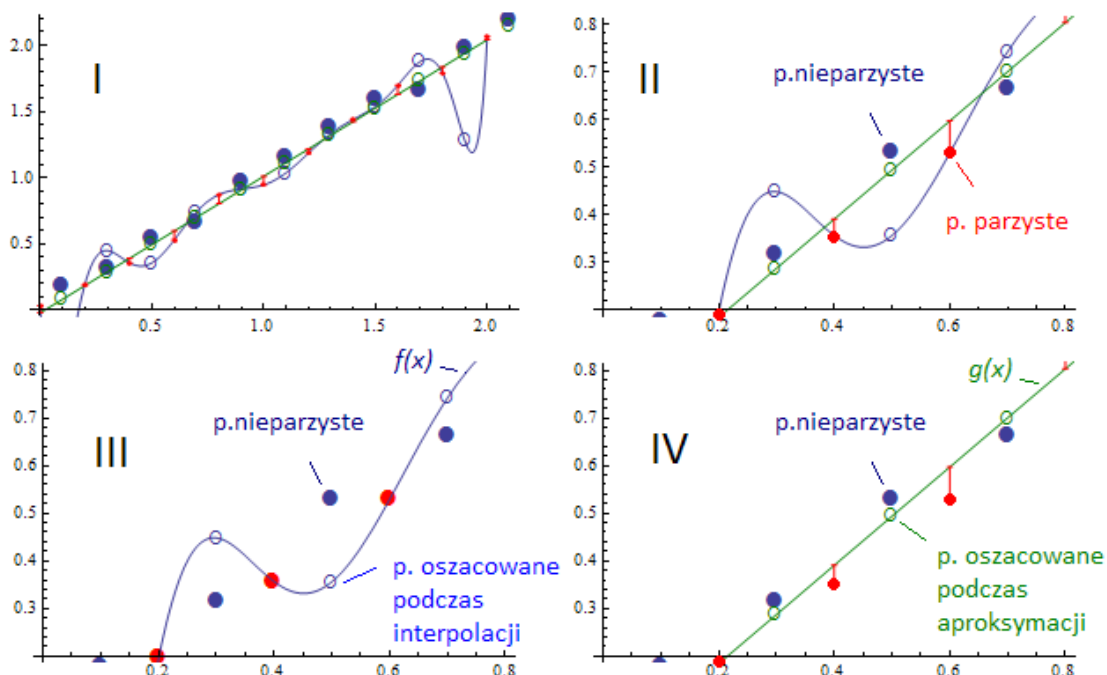
**Część VII** jest podsumowaniem rozprawy doktorskiej, określająca obszar zastosowań metod falkowych w oparciu o schemat liftingu.

Prezentuje ona analizę wyników uzyskanych w pracy, potwierdzając poprawność postawionych tez, opisanych w części I, proponując dalsze drogi rozwoju dla wielorozdzielczej analizy funkcji za pomocą

SzacowaniePróbekNieparzystych[funkcja, zakres, zakresWidoczności]

$f(x) = 0.03233307 - 38.2425x + 523.773x^2 - 2785.24x^3 + 7877.17x^4 - 13270.8x^5 + 13989.6x^6 - 9319.91x^7 + 3813.1x^8 - 874.655x^9 + 86.1139x^{10}$  //funkcja interpolująca

$g(x) = -0.0221575 + 1.03199x$  //funkcja aproksymująca



Rysunek 1.5: Szacowanie próbek nieparzystych - implementacja

schematu liftingu wykorzystującego aproksymację metodą najmniejszych kwadratów w bloku predykcji.

Do celów badań i prezentacji został wykorzystany zbiór metod i procedur (wymienionych w rozdz. 19) stworzonych a zarazem oprogramowanych przez autora za pomocą uniwersalnego środowiska do modelowania matematycznego Mathematica firmy Wolfram.

Zbiór funkcji został pogrupowany i przygotowany do dystrybucji, stanowiąc w pełni działający i kompleksowy pakiet narzędzia Mathematica do analizy schematu liftingu z wykorzystaniem aproksymacji MNK.

W pracy sformułowano następujące tezy:

### Teza 1:

Możliwe jest zdefiniowanie nowej biortogonalnej transformaty falkowej w przypadku której, w bloku predykcji schematu liftingu, zamiast interpolacji wielomianowej zaproponowanej przez Wima Sweldensa [61, 60] użyta jest aproksymacja metodą najmniejszych kwadratów.

### Teza 2:

Nowa transformata falkowa spełniająca wszystkie własności analizy wielorozdzielczej, posiada również korzystną cechę, którą jest możliwość wykorzystania funkcji aproksymującej w bloku predykcji o rzędzie niższym bądź równym funkcji interpolującej.

Poprawność stawianych tez autor zaprezentował we wnioskach końcowych części VII odwołując się jednocześnie do odpowiednich fragmentów rozprawy doktorskiej.

W części VIII, ostatniej zamieszczona została bibliografia wraz ze spisami rysunków, tabel i algorytmów wykorzystywanych w pracy.



## Część II

# Analiza falkowa ortogonalnych funkcji ciągłych



## Rozdział 2

---

# Wstęp do przestrzeni ortogonalnych

---

Joseph Fourier na podstawie badań dotyczących strumienia przepływu ciepła zauważył, że przepływ podobnie jak dźwięk można przedstawić jako sumę prostych funkcji sinusoidalnych o stałych częstotliwościach i amplitudach.

Wykazał, że każdą funkcję harmoniczną spełniającą warunki Dirichleta\* można przedstawić w postaci dyskretnej [44, 5]

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \sin(nx) \quad (2.1)$$

Mnożąc obustronnie wyrażenie 2.1 przez  $\cos(nx)$ , a następnie całkując przez  $\int_{-\pi}^{\pi}$  otrzymał ogólny wzór na współczynniki  $a_n$  określające amplitudę składowych  $\cos(nx)$

$$a_n = \langle f(x), \cos(nx) \rangle = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos(nx), \quad n = 0, 1, 2, 3, \dots \quad (2.2)$$

Analogicznie mnożąc przez  $\sin(mx)$  i całkując przez  $\int_{-\pi}^{\pi}$  wyprowadził wzór na współczynniki  $b_n$  tj.

$$b_n = \langle f(x), \sin(nx) \rangle = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin(nx), \quad n = 1, 2, 3, \dots \quad (2.3)$$

Aby rozszerzyć dziedzinę badanej funkcji wystarczy zmienić granicę całkowania  $x \rightarrow \frac{2\pi}{T}t$ , wtedy

---

\*Jeżeli  $f : \mathbb{R} \rightarrow \mathbb{R}$  jest funkcją okresową o okresie  $T$  spełnia warunki Dirichleta

1. Funkcja  $f$  jest bezwzględnie całkowalna, tzn.  $\int_{-\frac{T}{2}}^{\frac{T}{2}} |f(x)| dx < \infty$
  2. Funkcja w przedziale jednego okresu posiada skończoną liczbę maksimów lokalnych i minimów lokalnych
  3. Funkcja w przedziale jednego okresu posiada skończoną liczbę punktów nieciągłości pierwszego rodzaju
- wtedy  $f$  posiada reprezentację w postaci szeregu Fouriera



to postać ogólna badanej funkcji prezentuje się jako liniowa kombinacja funkcji harmoniczych na przedziale  $\langle -\frac{T}{2}, \frac{T}{2} \rangle$ .

Zatem

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(n\frac{2\pi}{T}t\right) + \sum_{n=1}^{\infty} b_n \sin\left(n\frac{2\pi}{T}t\right), \quad (2.4)$$

gdzie

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \cos\left(n\frac{2\pi}{T}t\right) dt, \quad n = 0, 1, 2, 3, \dots \quad (2.5)$$

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \sin\left(n\frac{2\pi}{T}t\right) dt, \quad n = 1, 2, 3, \dots \quad (2.6)$$

Powyższy szereg nazywamy szeregiem Fouriera.

W praktyce często stosowany jest inny zapis zależności 2.4, ze względu na jej fizyczną zależność.

Przyjmuje się, że  $\omega = \frac{2\pi}{T}$ , gdzie  $\omega$  nosi nazwę pulsacji lub częstości kołowej funkcji  $x(t)$  co w konsekwencji prowadzi do wzoru

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + \sum_{n=1}^{\infty} b_n \sin(n\omega t), \quad (2.7)$$

gdzie

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \cos(n\omega t) dt, \quad n = 0, 1, 2, 3, \dots \quad (2.8)$$

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) \sin(n\omega t) dt, \quad n = 1, 2, 3, \dots \quad (2.9)$$

Współczynniki  $a_n$  nazywane są często współczynnikami widma parzystego, zaś  $b_n$  - współczynnikami widma nieparzystego.

Aby zunifikować zapis współczynników  $a_n$  i  $b_n$  do jednego zapisu  $c_n$ , postać trygonometryczną można przedstawić za pomocą zespolonej postaci wykładniczej.

Wykorzystując wzory Eulera

$$\sin(n\omega t) = \frac{e^{in\omega t} - e^{-in\omega t}}{2i} \quad (2.10)$$

$$\cos(n\omega t) = \frac{e^{in\omega t} + e^{-in\omega t}}{2} \quad (2.11)$$

szereg Fouriera można zapisać w postaci [47]

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega t} \quad (2.12)$$

gdzie  $c_n$  - nazywamy amplitudą zespoloną i wyraża się wzorem

$$c_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x(t) e^{-in\omega t} dt \quad (2.13)$$

Zależności pomiędzy współczynnikami postaci trygonometrycznej, a postacią wykładniczą można zobrazować w poniższy sposób

$$\begin{aligned} a_n &= c_n + c_{-n} &= c_n + c_n^* &= 2\Re(c_n) \\ b_n &= j(c_n - c_{-n}) &= i(c_n - c_n^*) &= -2\Im(c_n) \\ c_n &= c_{-n}^* \end{aligned} \quad (2.14)$$

Weryfikując własności szeregu można zauważyć, że funkcje bazowe t.j  $\frac{1}{2}$ ,  $\cos(\omega t)$ ,  $\dots$ ,  $\cos(n\omega t)$ ,  $\sin(\omega t)$ ,  $\dots$ ,  $\sin(n\omega t)$  spełniają układ ortogonalny, gdzie iloczyn skalarny  $f \circ g = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(t)g(t)dt$ .

W rozdziale tym został opisany szereg Fouriera, ponieważ jest on niejako prekursorem szeregów ortogonalnych otwierającym nowe horyzonty rozwoju analizy wielorozdzielczej z analizą falkową włącznie [25, 26, 68].

Szereg Fouriera ma niestety swoje wady, funkcja sinusoidalna wraz ze stałą amplitudą narzuca pewne ograniczenia związane z aproksymacją funkcji badanej ze względu na jej nieskończoną energię. Wykorzystywanie ortogonalnych funkcji falkowych do celów aproksymacyjnych, opisywanych w następnym rozdziale, pozwala na eliminację tego problemu.



## Rozdział 3

---

# Falki ortogonalne i równania definiujące

---

Falki ortogonalne to rodziny funkcji zbioru liczb rzeczywistych, z których każda jest wyprowadzona z funkcji tzw. falki matki za pomocą przesunięcia i skalowania.

$$\psi_{mn}(x) = 2^{\frac{m}{2}} \psi(2^m x - n) \quad (3.1)$$

Funkcje te oprócz powyższej własności posiadają skończoną energię skupioną wobec punktu dzięki czemu eliminują problem funkcji sinusoidalnych.

Analiza falkowa spełnia dodatkowo 5 podstawowych postulatów [75]:

### Postulat 1

Istnieje ciąg przestrzeni  $V_M, V_{M+1}, V_{M+2}, \dots, V_m, \dots$  które należą do *przestrzeni aproksymacji* wzajemnie zanurzonych w  $L^2(\mathbb{R})$ , zwanych analizą wielorozdzielczą, gdzie

$$V_M \subset V_{M+1} \subset \dots \subset V_m \subset \dots \subset L^2(\mathbb{R}), \quad M \in \mathbb{Z} \quad (3.2)$$

$$\lim_{m \rightarrow -\infty} V_m = \{0\}$$

$$\overline{\lim_{m \rightarrow \infty} V_m} = L^2(\mathbb{R})$$

gdzie każda z przestrzeni  $V_m$  posiada bazę ortonormalną  $\{v_{mn}(x)\}_{n \in \mathbb{Z}}$ , to znaczy

$$V_m = \{v_{mn}(x), \quad n \in \mathbb{Z}\} \quad (3.3)$$

Dzięki powyższej własności można zdefiniować wielostopniową aproksymację dowolnej funkcji  $F(x) \in L^2(\mathbb{R})$ .

$$F(x; m) = \sum_{n \in \mathbb{Z}} a_{mn}^F v_{mn}(x), \quad m = M, M + 1, \dots \quad (3.4)$$

gdzie

$$a_{mn}^F = \int_{-\infty}^{\infty} F(x) v_{mn}(x) dx$$

Precyzja aproksymacji rośnie wraz ze wzrostem  $m$ .

### Postulat 2

Istnieje ciąg  $W_M, W_{M+1}, \dots, W_m$  przestrzeni detali takich, że [29]

$$W_m \perp V_m \quad \wedge \quad V_{m+1} = V_m \oplus W_m \quad (3.5)$$

dla  $m \in M, M + 1, \dots$  gdzie każda z przestrzeni  $W_m$  posiada bazę ortonormalną  $\{w_{mn}(x)_{n \in \mathbb{Z}}\}$ , to znaczy

$$W_m = \{w_{mn}(x), \quad n \in \mathbb{Z}\} \quad (3.6)$$

Z powyższego postulatu wynika, że:

$$F(x; m) = F(x; m - 1) + \sum_{n \in \mathbb{Z}} b_{m-1,n}^F w_{m-1,n}(x) \quad (3.7)$$

gdzie

$$b_{m-1,n}^F = \int_{-\infty}^{\infty} F(x) w_{m-1,n}(x) dx$$

### Postulat 3

Ortonormalne funkcje bazowe  $\{v_{mn}(x)_{n \in \mathbb{Z}}\}$  i  $\{w_{mn}(x)_{n \in \mathbb{Z}}\}$  dla kolejnych przestrzeni  $V_m$  i  $W_m$  otrzymuje się w sposób rekurencyjny według następującej zasady

$$\begin{aligned}
 v_{mn}(x) &= \sqrt{2}v_{m-1,n}(2x) \\
 w_{mn}(x) &= \sqrt{2}w_{m-1,n}(2x)
 \end{aligned}
 \tag{3.8}$$

#### Postulat 4

Przyjmując

$$\begin{aligned}
 v_{0n}(x) &= \varphi(x - n) \\
 w_{0n}(x) &= \psi(x - n)
 \end{aligned}
 \tag{3.9}$$

z wykorzystaniem postulatu 3 można wyznaczyć wszystkie funkcje bazowe powstające z przekształcenia i skalowania funkcji podstawowych  $\varphi(x)$  i  $\psi(x)$  tj.

$$\begin{aligned}
 \varphi_{mn}(x) &= 2^{\frac{m}{2}}\varphi(2^m x - n) \\
 \psi_{mn}(x) &= 2^{\frac{m}{2}}\psi(2^m x - n)
 \end{aligned}
 \tag{3.10}$$

#### Postulat 5

Funkcja  $\varphi(x)$  nazywa się funkcją skalującą (falką ojcem), zaś  $\psi(x)$ - falką (falką matką).

Funkcja skalująca spełnia warunek normalizacji

$$\int_{-\infty}^{\infty} \varphi(x) = 1
 \tag{3.11}$$

zaś funkcja falkowa - warunek oscylacji, pozwalając na oscylację funkcji wokół osi  $y = 0$

$$\int_{-\infty}^{\infty} \psi(x) = 0
 \tag{3.12}$$

Dzięki zdefiniowanym postulatом można określić następujące **równania definiujące** wykorzystywane w analizie falkowej tj.

$$\begin{aligned}
 \varphi(x) &= \sqrt{2} \sum_{n \in \mathbb{Z}} h_n \varphi(2x - n) \\
 \psi(x) &= \sqrt{2} \sum_{n \in \mathbb{Z}} g_n \varphi(2x - n)
 \end{aligned}
 \tag{3.13}$$

gdzie współczynniki filtrów dolnoprzepustowych  $\{h_n\}$  jak i górnoprzepustowych  $\{g_n\}$  powinny spełniać następujące równania

$$\begin{cases} \sum_{n \in \mathbb{Z}} h_n &= \sqrt{2} \\ \sum_{n \in \mathbb{Z}} h_n^2 &= 1 \\ \sum_{n \in \mathbb{Z}} h_n h_{n-2k} &= 0, \quad k \neq 0, \quad k \in \mathbb{Z} \end{cases} \quad (3.14)$$

oraz

$$\begin{cases} \sum_{n \in \mathbb{Z}} g_n &= 0 \\ \sum_{n \in \mathbb{Z}} g_n^2 &= 1 \\ \sum_{n \in \mathbb{Z}} g_n g_{n-2k} &= 0, \quad k \in \mathbb{Z} \setminus \{0\} \end{cases} \quad (3.15)$$

Ponieważ na podstawie powyższych zależności otrzymuje się dodatkowy warunek wykazujący korelację pomiędzy filtrami [26]

$$\sum_{n \in \mathbb{Z}} g_n h_{n-2k} = 0, \quad k \in \mathbb{Z} \quad (3.16)$$

w konsekwencji prowadzi to do definicji tzw. filtrów lustrzanych wyrażonych równaniem

$$g_n = (-1)^n h_{-n+1} \quad (3.17)$$

Własność ta pozwala na proste otrzymywanie wartości współczynników funkcji skalującej ze współczynników funkcji falkowej i na odwrót.

Otrzymane współczynniki filtrów pozwalają definiować filtry których reprezentacją graficzną są falki o ograniczonym, zwartym nośniku tj. przyjmujące niezerowe wartości dla przedziału zwartego.

Falki takie, o nośniku zwartym zapewniają analizę wielorozdzielczą z wykorzystaniem nośnika o skończonej energii.

## Rozdział 4

---

# Falki o nośniku zwartym

---

Falki o nośniku zwartym są to falki które na przedziale  $\langle s_1, s_2 \rangle$  posiadają niezerowe współczynniki, natomiast poza przedziałem wszystkie współczynniki przyjmują wartość zero.

Nośnik funkcji oznacza się jako  $\text{supp } f$ , natomiast najmniejszy domknięty odcinek  $\langle s_1, s_2 \rangle$  zawierający nośnik jako  $\text{Supp } f$  - nośnikiem zwartym.

Innymi słowy funkcję o nośniku zwartym nazywa się funkcję, gdzie [52]

$$\forall x \notin \langle s_1, s_2 \rangle f(x) = 0 \quad (4.1)$$

W aproksymacji wielorozdzielczej najczęstsze zastosowanie posiadają funkcje falkowe o zwartym nośniku ze względu na swoją całą energię skondensowaną na skończonym przedziale.

Dzięki takiej własności można aproksymować funkcję  $L^2(\mathbb{R})$  tzw. funkcjami skalującymi zwięzając i przesuwając odpowiednio ich nośnik.

Jeżeli funkcja skalująca posiada zwarty nośnik  $\text{Supp } \varphi(x) = \langle s_1, s_2 \rangle$ , wtedy to  $\text{Supp } \varphi(2x - n) = \langle \frac{s_1+n}{2}, \frac{s_2+n}{2} \rangle$ .

Dzięki tej własności równanie skalujące dla funkcji skalującej  $\varphi(x)$  wynosi odpowiednio [75]:

$$\varphi(x) = \sqrt{2} \sum_{n=s_1}^{s_2} h_n \varphi(2x - n) \quad (4.2)$$

gdzie liczba współczynników filtru dolnoprzepustowego  $\{h_n\}$  wynosi odpowiednio  $S = s_2 - s_1 + 1$ . Na podstawie powyższej zależności 4.2 równania definiujące wyrażają się w poniższy sposób [75]

$$\begin{cases} \sum_{n=s_1}^{s_2} h_n & = \sqrt{2} \\ \sum_{n=s_1}^{s_2} h_n^2 & = 1 \\ \sum_{n=s_1+2k}^{s_2} h_n h_{n-2k} & = 0 \quad k = 1, 2, \dots, \frac{S}{2} - 1 \wedge h_{s_1}, h_{s_2} \neq 0 \end{cases} \quad (4.3)$$



oraz

$$\begin{cases} \sum_{n=-s_2+1}^{-s_1+1} g_n & = 0 \\ \sum_{n=-s_2+1}^{-s_1+1} g_n^2 & = 1 \\ \sum_{n=-s_2+1}^{-s_1+1} g_n g_{n-2k} & = 0, \quad k \in \mathbb{Z} \setminus \{0\} \wedge g_{-s_2+1}, g_{-s_1+1} \neq 0 \end{cases} \quad (4.4)$$

Ponieważ równanie falkowe przyjmuje postać

$$\psi(x) = \sqrt{2} \sum_{n=-s_2+1}^{-s_1+1} g_n \varphi(2x - n) \quad (4.5)$$

lub zgodnie z zasadą odbicia równoważnie

$$\psi(x) = \sqrt{2} \sum_{n=-s_2+1}^{-s_1+1} (-1)^n h_{-n+1} \varphi(2x - n), \quad (4.6)$$

wtedy to można bezpośrednio wyznaczyć nośnik falki  $\psi(x)$  dla odpowiadającej jej funkcji skalującej  $\varphi(x)$  tj.

$$\begin{aligned} \text{Supp } \varphi(2x - n) &= \left\langle \frac{s_1+n_{\psi, \min}}{2}, \frac{s_2+n_{\psi, \max}}{2} \right\rangle \\ &= \left\langle \frac{s_1-s_2+1}{2}, \frac{s_2-s_1+1}{2} \right\rangle \\ &= \left\langle \frac{1}{2} - \frac{s_2-s_1}{2}, \frac{1}{2} + \frac{s_2-s_1}{2} \right\rangle \\ &= \left\langle \frac{1}{2} - \frac{s}{2}, \frac{1}{2} + \frac{s}{2} \right\rangle \\ &= \langle t_1, t_2 \rangle \end{aligned} \quad (4.7)$$

## 4.1 Stopnie swobody i niezmienniczość względem translacji

Analizując powyższe rozważania można zauważyć, że istnieje  $S = s_2 - s_1 + 1$  współczynników filtrów, natomiast układy 4.3 i 4.4 posiadają po  $1 + 1 + (\frac{S}{2} - 1) = \frac{S}{2} + 1$  równań [70].

Otrzymuje się zatem  $\frac{S}{2} - 1$  stopni swobody, dzięki którym można projektować falki o zadanych własnościach.

Ze względu na przesunięcie funkcji falkowych funkcja falkowa ma charakter niezmienniczy tj. posiada te same wartości współczynników  $\{h_n\}$ . Dzięki tej własności dla funkcji o nośniku  $\langle s_1, s_2 \rangle$  można przyjąć nośnik  $\langle 0, s \rangle$ , gdzie  $s = s_2 - s_1$ , zaś równania 4.3 przyjmują standardową postać

$$\begin{cases} \sum_{n=0}^s h_n & = \sqrt{2} \\ \sum_{n=0}^s h_n^2 & = 1 \\ \sum_{n=2k}^s h_n h_{n-2k} & = 0 \quad k = 1, 2, \dots, \frac{s}{2} - 1 \end{cases} \quad (4.8)$$

## 4.2 Momenty znikające i falki Daubechies

Aby wypełnić tzw. lukę w dowolności kształtu funkcji falkowych, ze względu na  $\frac{s}{2} - 1$  niezdefiniowanych równań, można przyjąć, że badana falka jest ortogonalna względem wielomianu stopnia  $r_\psi$ .

Konsekwencją tego jest  $r_\psi$  znikających momentów dla falki o nośniku  $\langle t_1, t_2 \rangle$  [19, 75] tj.

$$\int_{t_1}^{t_2} x^k \psi(x) dx = 0, \quad k = 0, 1, 2, \dots, r_\psi \quad (4.9)$$

Na podstawie powyższego założenia 4.9 otrzymuje się dodatkową zależność

$$\sum_{n=-s+1}^1 g_n n^k = 0, \quad k = 0, 1, \dots, r_\psi \quad (4.10)$$

lub równoważnie

$$\sum_{n=-s+1}^1 (-1)^n h_{-n+1} n^k = 0, \quad k = 0, 1, \dots, r_\psi \quad (4.11)$$

gdzie  $s = s_2 - s_1$  natomiast  $g_n = (-1)^n h_{-n+1}$ .

Zatem, dzięki dodatkowej własności 4.10 można stworzyć falki które będą całkowicie zdeterminowane przez powyższe własności.

Ponieważ liczba momentów znikających  $r_\psi \leq \frac{s}{2} - 1 = \frac{s-1}{2}$ , tak więc  $r_\psi = \frac{s-1}{2}$  jest maksymalną liczbą momentów dla danej długości nośnika [75].

Analogicznie, dla zadanej długości nośnika  $s \geq 2r_\psi + 1$  wynika, że maksymalna długość nośnika  $s = 2r_\psi + 1$  jest zawsze nieparzysta, zatem współczynniki falkowe muszą być liczbą parzystą.

Taka klasa falek która spełnia własność najkrótszego nośnika względem danej liczby momentów znikających jest tzw. klasą falek Daubechies o numerach  $p = 1, 2, \dots$  gdzie  $p = \frac{s+1}{2}$  ( $s = 2p - 1$ ) [53].

Falki Daubechies oznaczane są odpowiednio  $db1, db2, \dots$  gdzie pierwsze dwie litery pochodzą od nazwiska twórcy - Ingrid Daubechies, zaś liczba z prawej strony symbolu określa numer  $p$  falki.



## Rozdział 5

---

# Obliczanie współczynników filtra dolno- i górnoprzepustowego

---

Dla celów prezentacji można rozważyć klasę falek *db2* zatem  $p = 2$ , gdzie  $s = 2p - 1 = 3$ .

Z długości  $s = 3$  wynika, że falka posiada cztery współczynniki  $h_0, h_1, h_2$  i  $h_3$ , ponadto liczba momentów znikających  $r_\psi = \frac{s-1}{2} = p - 1 = 1$ .

Układ równań prezentuje się zatem następująco

$$\begin{cases} h_0 + h_1 + h_2 + h_3 = \sqrt{2} \\ h_0^2 + h_1^2 + h_2^2 + h_3^2 = 1 \\ h_0 h_2 + h_1 h_3 = 0 \\ -1h_0 + 0h_1 + 1h_2 - 2h_3 = 0 \end{cases}$$

Podczas przekształceń algebraicznych dla powyższego układu otrzymuje się wielomian stopnia IV.

W tej sytuacji do celów obliczeniowych można użyć przekształceń Cardano [41] pozwalających na uzyskanie wyniku dokładnego, gdzie rozwiązaniem równania są współczynniki filtra dolnoprzepustowego

$$\begin{cases} h_0 = \frac{1}{8}(\sqrt{2} + \sqrt{6}) = \frac{1+\sqrt{3}}{4\sqrt{2}} \approx 0.48296 \\ h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}} \approx 0.83652 \\ h_2 = \frac{1}{4}\sqrt{6-3\sqrt{3}} = \frac{3-\sqrt{3}}{4\sqrt{2}} \approx 0.22414 \\ h_3 = \frac{1}{8}(\sqrt{2} - \sqrt{6}) = \frac{1-\sqrt{3}}{4\sqrt{2}} \approx -0.12941 \end{cases}$$

Korzystając z zależności  $g_n = (-1)^n h_{-n+1}$  współczynniki filtra górnoprzepustowego wynoszą odpowiednio

$$\begin{cases} g_{-2} = \frac{1-\sqrt{3}}{4\sqrt{2}} \approx 0.12941 \\ g_{-1} = -\frac{3-\sqrt{3}}{4\sqrt{2}} \approx 0.22414 \\ g_0 = \frac{3+\sqrt{3}}{4\sqrt{2}} \approx 0.83652 \\ g_1 = -\frac{1+\sqrt{3}}{4\sqrt{2}} \approx -0.48296 \end{cases}$$

Obliczenia współczynników nie zawsze można wykonać w sposób dokładny, symboliczny - spowodowane jest to dużym stopniem skomplikowania czy też brakiem odpowiednich metod wyliczających pierwiastki wielomianów stopnia wyższego aniżeli IV.

Z powyższego powodu dla nieliniowych układów równań stosowane są zazwyczaj metody przybliżone, numeryczne - wykorzystujące np. zasadę minimalizacji funkcji celu tj.

$$\left(\sum_{n=0}^s h_n^2 - 1\right)^2 \rightarrow \min_{\{h_n\}_{n=0}^s}$$

z dodatkowymi ograniczeniami

$$\begin{cases} \sum_{n=0}^s h_n - \sqrt{2} = 0 \\ \sum_{n=0}^s h_{n-r} h_n = 0, \quad r = 2, 4, \dots, s-1 \\ h_0, h_s \neq 0 \end{cases}$$

## Rozdział 6

---

# Wyznaczanie funkcji skalującej o zwartym nośniku

---

### 6.1 Algorytm Stranga

Wykorzystując uprzednio obliczone wartości filtra dolnoprzepustowego dla celów prezentacji można wyznaczyć wartości funkcji skalującej.

Funkcję skalującą można przedstawić w postaci uwikłanej

$$\tilde{\varphi}(x) = \sqrt{2} \sum_{n=s_1}^{s_2} h_n \tilde{\varphi}(2x - n).$$

lub w jej zestandaryzowanej postaci

$$\varphi(x) = \sqrt{2} \sum_{n=0}^s h_n \varphi(2x - n)$$

ze względu na charakter niezmienniczy współczynników  $h_n$  z dokładnością do translacji  $t \in \mathbb{Z}$  [53].

Aby wyznaczyć wartości funkcji skalującej dla przykładu z rozdziału 5 ( $s = 3$ ) otrzymuje się następujący układ równań dla kolejnych liczb całkowitych zawartych w nośniku  $\langle 0, 3 \rangle$

$$\begin{cases} \varphi(0) = \sqrt{2}h_0\varphi(0) \\ \varphi(1) = \sqrt{2}h_0\varphi(2) + \sqrt{2}h_1\varphi(1) + \sqrt{2}h_2\varphi(0) \\ \varphi(2) = \sqrt{2}h_1\varphi(3) + \sqrt{2}h_2\varphi(2) + \sqrt{2}h_3\varphi(1) \\ \varphi(3) = \sqrt{2}h_3\varphi(3) \end{cases}$$

gdzie  $\varphi(x) = 0$  dla  $x \notin \langle 0, 3 \rangle$ .

Wiedząc, że  $\varphi(0) = \varphi(3) = 0$  i podstawiając  $c_n = \sqrt{2}h_n$ , wyznacza się zredukowaną postać

$$\begin{cases} \varphi(1) = c_1\varphi(1) + c_0\varphi(2) \\ \varphi(2) = c_3\varphi(1) + c_2\varphi(2) \end{cases}$$

$$\begin{bmatrix} \varphi(1) \\ \varphi(2) \end{bmatrix} = \begin{bmatrix} c_1 & c_0 \\ c_3 & c_2 \end{bmatrix} \begin{bmatrix} \varphi(1) \\ \varphi(2) \end{bmatrix}$$

Podczas analizy układu można zauważyć interesującą własność, że  $\sum_{n \in \mathbb{Z}} c_{2n} = \sum_{n \in \mathbb{Z}} c_{2n+1} = 1$ , wynika to bezpośrednio z zależności równań falkowych [54]

$$\sum_{n \in \mathbb{Z}} h_{2n} = \sum_{n \in \mathbb{Z}} h_{2n+1} = \frac{1}{\sqrt{2}} \quad (6.1)$$

$$\sum_{n \in \mathbb{Z}} g_{2n} = -\sum_{n \in \mathbb{Z}} g_{2n+1} = \frac{1}{\sqrt{2}} \quad (6.2)$$

Otrzymany układ posiada niezerowe rozwiązanie, ponieważ macierz  $A' - I$  jest macierzą osobliwą ( $|A' - I| = 0$ ), gdzie

$$A' = \begin{bmatrix} c_1 & c_0 \\ c_3 & c_2 \end{bmatrix}$$

zaś liczba  $\lambda = 1$  jest wartością własną macierzy  $A'$ .

Aby uzyskać zdeterminowane, jednoznaczne rozwiązanie można nałożyć na układ dodatkowe ograniczenie.

W opisywanym przypadku jest to tzw. normalizacja wartości funkcji skalującej

$$\sum_{n=0}^s \varphi(n) = 1 \quad (6.3)$$

W konsekwencji powyższe założenie normalizujące pozwala otrzymać następujące wartości

$$\begin{cases} \varphi(0) = 0 \\ \varphi(1) = \frac{1+\sqrt{3}}{2} \\ \varphi(2) = \frac{1-\sqrt{3}}{2} \\ \varphi(3) = 0 \end{cases}.$$

Rozwiązanie dla argumentów niecałkowitych wyznacza się odpowiednio

$$\begin{cases} \varphi(\frac{1}{2}) = c_0\varphi(1) + c_1\varphi(0) \\ \varphi(\frac{3}{2}) = c_0\varphi(3) + c_1\varphi(2) + c_2\varphi(1) + c_3\varphi(0) \\ \varphi(\frac{5}{2}) = c_2\varphi(3) + c_3\varphi(2) \end{cases}$$

Równoważny zapis w formie macierzowej  $Y = BX$  (rugując przy tym  $\varphi(3)$  gdzie  $\varphi(3) = 0$ ) przedstawia się następująco

$$\begin{bmatrix} \varphi(\frac{1}{2}) \\ \varphi(\frac{3}{2}) \\ \varphi(\frac{5}{2}) \end{bmatrix} = \begin{bmatrix} c_1 & c_0 & 0 \\ c_3 & c_2 & c_1 \\ 0 & 0 & c_3 \end{bmatrix} \begin{bmatrix} \varphi(0) \\ \varphi(1) \\ \varphi(2) \end{bmatrix}$$

Zatem, wyznaczając kolejne wartości metodą połowienia (bisekcji) dla konkretnych argumentów otrzymuje się poniższe równania macierzowe

$$\begin{bmatrix} \varphi(\frac{1}{4}) \\ \varphi(\frac{5}{4}) \\ \varphi(\frac{9}{4}) \end{bmatrix} = \begin{bmatrix} c_0 & 0 & 0 \\ c_2 & c_1 & c_0 \\ 0 & c_3 & c_2 \end{bmatrix} \begin{bmatrix} \varphi(\frac{1}{2}) \\ \varphi(\frac{3}{2}) \\ \varphi(\frac{5}{2}) \end{bmatrix} \quad (6.4)$$

$$\begin{bmatrix} \varphi(\frac{3}{4}) \\ \varphi(\frac{7}{4}) \\ \varphi(\frac{11}{4}) \end{bmatrix} = \begin{bmatrix} c_1 & c_0 & 0 \\ c_3 & c_2 & c_1 \\ 0 & 0 & c_3 \end{bmatrix} \begin{bmatrix} \varphi(\frac{1}{2}) \\ \varphi(\frac{3}{2}) \\ \varphi(\frac{5}{2}) \end{bmatrix} \quad (6.5)$$



gdzie [53]

$$\mathbf{A} = \begin{bmatrix} c_0 & 0 & 0 \\ c_2 & c_1 & c_0 \\ 0 & c_3 & c_2 \end{bmatrix} \quad (6.6)$$

$$\mathbf{B} = \begin{bmatrix} c_1 & c_0 & 0 \\ c_3 & c_2 & c_1 \\ 0 & 0 & c_3 \end{bmatrix}$$

Wykorzystując powyższą metodę dla coraz to mniejszego podziału wyznacza się ogólną zależność

$$\begin{bmatrix} \varphi(\frac{m}{2}) \\ \varphi(\frac{m}{2} + 1) \\ \varphi(\frac{m}{2} + 2) \end{bmatrix} = \begin{bmatrix} c_0 & 0 & 0 \\ c_2 & c_1 & c_0 \\ 0 & c_3 & c_2 \end{bmatrix} \begin{bmatrix} \varphi(m) \\ \varphi(m + 1) \\ \varphi(m + 2) \end{bmatrix} \quad (6.7)$$

$$\begin{bmatrix} \varphi(\frac{3}{2}m) \\ \varphi(\frac{3}{2}m + 1) \\ \varphi(\frac{3}{2}m + 2) \end{bmatrix} = \begin{bmatrix} c_1 & c_0 & 0 \\ c_3 & c_2 & c_1 \\ 0 & 0 & c_3 \end{bmatrix} \begin{bmatrix} \varphi(m) \\ \varphi(m + 1) \\ \varphi(m + 2) \end{bmatrix}$$

gdzie  $m \in \frac{1}{2^N}$  i  $n \in \mathbb{N}$ .

Otrzymana reguła dla tzw. punktów binarnych nazywana jest często **algorytmem Stranga**, który ze względu na powyższą formułę spełnia następującą własność [53, 54].

Niech

$$\varphi(\mathbf{m}) = \begin{bmatrix} \varphi(m) \\ \varphi(m + 1) \\ \vdots \\ \varphi(m + s - 1) \end{bmatrix} \quad (6.8)$$

wtedy to powyższy wektor wartości dla funkcji skalującej oblicza się według następującego wzoru [75]

$$\varphi(\mathbf{m}) = \prod_{n=1}^l [(1 - m_n)] \mathbf{A} + m_n \mathbf{B} \varphi(\mathbf{0}) \quad (6.9)$$

gdzie  $l$  jest długością mantysy reprezentowanej w postaci binarnej,  $m_n$  natomiast jest wartością poszczególnych cyfry ( $m_n \in \{0, 1\}$ ) dla  $n$ -tej pozycji mantysy.

Dla przykładu, aby obliczyć wartość dla argumentu z siatki binarnej  $\frac{53}{128} = 0.414063_{10} = 0.0110101_2$  uzyskuje się następujący wzór

$$\varphi\left(\frac{53}{128}\right) = \varphi(0.414063_{10}) = \varphi(0.0110101_2) = \begin{bmatrix} \varphi(0.414063_{10}) \\ \varphi(1.414063_{10}) \\ \varphi(2.414063_{10}) \end{bmatrix}$$

gdzie

$$\begin{bmatrix} \varphi(0.414063_{10}) \\ \varphi(1.414063_{10}) \\ \varphi(2.414063_{10}) \end{bmatrix} = \mathbf{ABBABAB} \begin{bmatrix} \varphi(0) \\ \varphi(1) \\ \varphi(2) \end{bmatrix} \\ = \begin{bmatrix} \frac{844+477\sqrt{3}}{2048} \\ \frac{92+35\sqrt{3}}{1024} \\ \frac{1020-547\sqrt{3}}{2048} \end{bmatrix}$$

## 6.2 Macierze przekształceń - postać ogólna

Weryfikując postać macierzy przekształceń która wykorzystuje algorytm Stranga podczas obliczania wartości funkcji dla argumentów binarnych można otrzymać istotną regularność. Macierze te są niesymetrycznymi macierzami taśmowymi z przesunięciem segmentowym o elementach

$$\mathbf{C} = \begin{bmatrix} c_1 & c_0 \\ c_3 & c_2 \\ \vdots & \vdots \\ c_{s-2} & c_{s-3} \\ c_s & c_{s-1} \end{bmatrix} \quad (6.10)$$

gdzie na mocy reguły 6.1 suma elementów każdej kolumny wynosi 1. Macierze przekształcenia wynoszą zatem odpowiednio [54, 75]

$$\mathbf{A} = \begin{bmatrix} c_0 & 0 & 0 & \dots & 0 & 0 \\ c_2 & c_1 & c_0 & \dots & 0 & 0 \\ \vdots & c_3 & c_2 & \dots & \vdots & \vdots \\ c_{s-3} & \vdots & \vdots & \dots & \vdots & \vdots \\ c_{s-1} & c_{s-2} & c_{s-3} & \dots & c_1 & c_0 \\ & c_s & c_{s-1} & & c_3 & c_2 \\ 0 & 0 & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & & c_{s-2} & c_{s-3} \\ 0 & 0 & 0 & \dots & c_s & c_{s-1} \end{bmatrix} \quad (6.11)$$

i

$$\mathbf{B} = \begin{bmatrix} c_1 & c_0 & 0 & 0 & \dots & 0 \\ c_3 & c_2 & c_1 & c_0 & \dots & 0 \\ \vdots & \vdots & c_3 & c_2 & \dots & \vdots \\ c_{s-2} & c_{s-3} & \vdots & \vdots & \dots & \vdots \\ c_s & c_{s-1} & c_{s-2} & c_{s-3} & \dots & c_1 \\ & & c_s & c_{s-1} & & c_3 \\ 0 & 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & & c_{s-2} \\ 0 & 0 & 0 & 0 & \dots & c_s \end{bmatrix} \quad (6.12)$$

## 6.3 Implementacja algorytmu Stranga

### 6.3.1 Przygotowywanie danych

Do celów obliczeniowych wymagana jest znajomość współczynników filtru  $h_n$ , postaci macierzy  $A$  i  $B$ , jak również wektora własnego  $\varphi(0)$ .

```

h[0] = (1 + sqrt[3]) / (4 sqrt[2]);
h[1] = (3 + sqrt[3]) / (4 sqrt[2]);
h[2] = (3 - sqrt[3]) / (4 sqrt[2]);
h[3] = (1 - sqrt[3]) / (4 sqrt[2]);

c[0] = sqrt[2] h[0];
c[1] = sqrt[2] h[1];
c[2] = sqrt[2] h[2];
c[3] = sqrt[2] h[3];

A = {{c[0], 0, 0}, {c[2], c[1], c[0]}, {0, c[3], c[2]}};
B = {{c[1], c[0], 0}, {c[3], c[2], c[1]}, {0, 0, c[3]}};

phi0 = {0, (sqrt[3] + 1) / 2, (-sqrt[3] + 1) / 2};

```

Rysunek 6.1: Algorytm Stranga - przygotowywanie danych

### 6.3.2 Realizacja algorytmu Stranga

```

MatrixForm[binary[53/128, A, B, phi0]]

```

MatrixForm=

$$\begin{pmatrix} 0.414063 & 1.41406 & 2.41406 \\ \frac{844+477\sqrt{3}}{2048} & \frac{92+35\sqrt{3}}{1024} & \frac{1020-547\sqrt{3}}{2048} \end{pmatrix}$$

Rysunek 6.2: Algorytm Stranga - przykładowy wynik dla  $\varphi(\frac{53}{128}) = \varphi(0.414063)$ .

---

**Algorytm 6.1** Algorytm Stranga - implementacja
 

---

```

binary[liczba_, A_, B_, phi0_] := Module[{a, dlugosc, bin, Cpom, Result},

  (* tworzenie liczby binarnej w postaci kolejnych elementow w tablicy
     np: 0,0111 -> {{1,1,1},-1}*)

  a = RealDigits[N[liczba], 2];

  dlugosc = 0;

  For[i = 1, i <= Length[a[[1]]], i++,
    If[a[[1, i]] == 1, dlugosc = i - a[[2]], 1]
  ];

  (* tworzenie wlasciwej postaci kolejnych elementow w tablicy np: 0,0111 -> {0,1,1,1}*)
  bin = Table[If[i <= -a[[2]], 0, a[[1, i + a[[2]]]]], {i, 1, dlugosc}];

  (* wartosc poczatkowa *)
  Cpom = phi0;

  (* algorytm Stranga *)
  For[i = Length[bin], i >= 1, i--,
    If[bin[[i]] == 0, Cpom = A.Cpom, Cpom = B.Cpom]
  ];

  (* prezentacja wyniku w prostszej postaci *)
  Cpom = FullSimplify[Cpom];

  (* Tablica wynikow *)
  result = {Table[N[liczba] + i, {i, 0, Length[Cpom] - 1}], Cpom};

  Return[result];

]

```

---

## 6.4 Wykreślanie funkcji skalującej

### 6.4.1 Przygotowanie wyników

Aby wykreślić funkcję skalującą należy przygotować zbiór argumentów binarnych  $m$  dla których będą wyliczane wartości funkcji.

Dla celów prezentacji argumenty zostały wytypowane z zakresu  $\{0, \frac{1}{256}, \frac{2}{256}, \dots, \frac{255}{256}\}$ .

Zatem dla **db2**, ze względu na nośnik  $s = 3$  uzyskuje się zbiór argumentów  $\{0, \frac{1}{256}, \frac{2}{256}, \dots, \frac{255}{256}, 1, 1\frac{1}{256}, 1\frac{2}{256}, \dots, 1\frac{255}{256}, 2, 2\frac{1}{256}, 2\frac{2}{256}, \dots, 2\frac{255}{256}\}$ .

**Algorytm 6.2** Algorytm Stranga - wykreślanie funkcji skalującej

---

```

concat::usage = "funkcja do składania dwóch tabel"
concat[list1_, list2_] :=
  Return[{list1[[1]]~Join~list2[[1]], list1[[2]]~Join~list2[[2]]};

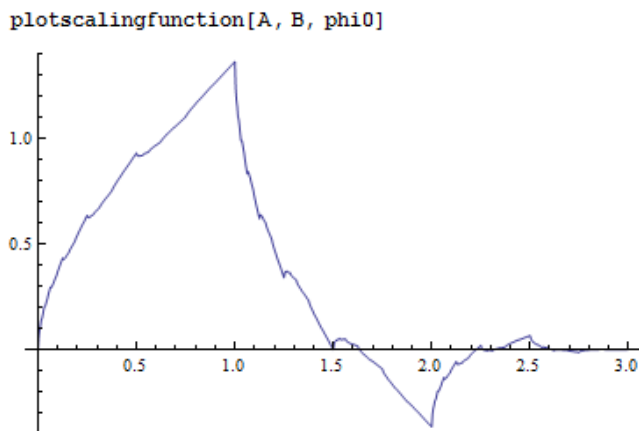
plotscalingfunction::usage = "wykreślanie funkcji skalującej"
plotscalingfunction[A_, B_, phi0_] := Module[{arraypom, j},
  arraypom = {{0}, {0}};
  (* generowanie zbioru binarnych wartości argumentów {0, 1/256, ..., 255/256}*)
  For[j = 0, j < 1, j += N[1/256],
    arraypom = concat[arraypom, binary[j, A, B, phi0]];
  ];
  (* stworzenie posortowanej listy*)

  ListPlot[Sort[Table[{arraypom[[1, i]], arraypom[[2, i]]}, {i, 1, Length[arraypom[[1]]}],
    #1[[1]] < #2[[1]] &, Joined -> True]
]

```

---

## 6.4.2 Graficzna prezentacja wyników

Rysunek 6.3: Wykres funkcji skalującej **db2** uzyskany za pomocą algorytmu Stranga



## Rozdział 7

---

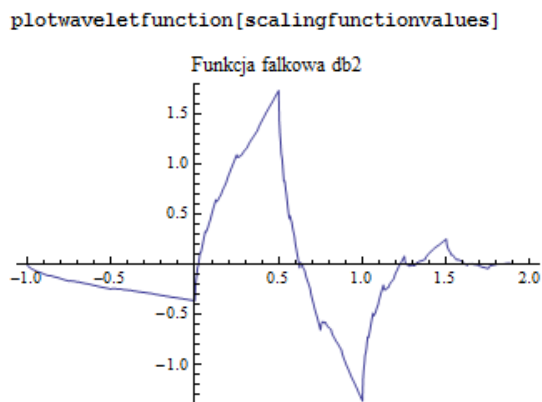
# Wyznaczanie funkcji falkowej o zwartym nośniku

---

Analogicznie jak w przypadku funkcji skalującej, dla wykreślenia funkcji falkowej wyznacza się wartości funkcji skalującej dzięki zależnościom uzyskiwanym na mocy wzoru [70]:

$$\psi(x) = \sqrt{2} \sum_{n=-s+1}^1 (-1)^n h_{-n+1} \varphi(2x - n) \quad (7.1)$$

które pozwalają obliczyć kolejne wartości funkcji falkowej i zobrazować ją za pomocą przykładowej implementacji 7.1.



Rysunek 7.1: Wykres funkcji falkowej **db2** uzyskany za pomocą algorytmu Stranga



---

**Algorytm 7.1** Algorytm Stranga - wykreślanie funkcji falkowej
 

---

```

scalingfunctionvalues::usage = "argumenty i wartosci funkcji skalujacej"
scalingfunctionvalues := Module[{arraypom, j},
  arraypom = {{}, {}};
  For[j = 0, j < 1, j += N[1/256],
    arraypom = concat[arraypom, binary[j, A, B, phi0]];
  ];
  Return[arraypom]
];

phivalue::usage = "znajdywanie w tablicy wartosci funkcji skalujacej dla danego argumentu"
phivalue[x_, array_] := Module[{licznik, dlugoscarray},
  licznik = 1;

  dlugoscarray = Length[array[[1]]];
  While[(licznik < dlugoscarray) && (x != array[[1, licznik]]), licznik++];
  If[licznik < dlugoscarray, Return[array[[2, licznik]]], Return[0]
  ];
];

falka::usage = "obliczanie wartosci funkcji falkowej"
falka[x_, s_, arraypom_] := Module[{n},
  Return[FullSimplify[Sqrt[2]  $\sum_{n=-s+1}^1 (-1)^{nh[-n+1]}$  phivalue[2 x - n, arraypom]]];
];

plotwaveletfunction::usage =
  "przygotowanie listy, sortowanie i wykreslenie funkcji falkowej"
plotwaveletfunction[arraypom_] :=
  ListPlot[Sort[Table[{arraypom[[1, i]] - 1, falka[arraypom[[1, i]] - 1, 3, arraypom]},
    {i, 1, Length[arraypom[[1]]}], #1[[1]] < #2[[1]] &, Joined -> True,
  PlotLabel -> "Funkcja falkowa db2"]

```

---

## Rozdział 8

---

# Przybliżone wyznaczanie wartości funkcji skalującej

---

Oprócz dokładnych metod obliczania wartości funkcji skalujących i w konsekwencji funkcji falkowych, istnieją metody przybliżone które w sposób iteracyjny pozwalają uzyskać wartości z ustaloną dokładnością.

Jednymi z takich metod są tzw. algorytmy kaskadowe które, ze względu na swoją zbieżność, pozwalają obliczać wartości funkcji skalujących zarówno w dziedzinie czasu, jak i częstotliwości[5].

### 8.1 Algorytm kaskadowy - wyznaczanie wartości w dziedzinie czasu

#### 8.1.1 Zbieżność algorytmu kaskadowego

Wartości funkcji rekurencyjnej można obliczyć na podstawie iteracji za pomocą filtru dolnoprzepustowego.

Punktem startowym jest przypisanie  $\varphi^{(0)}(t) = \text{box}(t)^*$ , zaś kolejne iteracje przebiegają według wzoru [54]

$$\varphi^{(i+1)}(t) = \sqrt{2} \sum_n h_n \varphi^{(i)}(2t - n) \quad (8.1)$$

Algorytm wykorzystywany jest dla funkcji ciągłych, gdzie podczas kolejnych przybliżeń funkcja przyjmuje postać schodkową.

Długość przedziału zaś, która przyjmuje wartość stałą po każdej iteracji, określa wartość  $2^{-i}$  (gdzie  $i$  jest numerem kolejnej iteracji).

Jeżeli  $\varphi^{(i)}(t)$  zbiega się granicznie do  $\varphi(t)$ , wtedy to granica funkcji spełnia równanie dylatacyjne 8.1.

Aby zobrazować zagadnienie zbieżności można posłużyć się następującymi przykładami:

---

$$*\text{box}(t) = \begin{cases} 1 & \text{dla } t \in \langle 0, 1 \rangle \\ 0 & \text{dla } t \notin \langle 0, 1 \rangle \end{cases}$$

**Przykład 1** Dla przykładu można przyjąć funkcję, gdzie

$$\varphi^{(1)}(t) = \frac{4}{3}\varphi^{(0)}(2t) + \frac{2}{3}\varphi^{(0)}(2t-1)$$

Pierwsza iteracja funkcją schodkową posiada wysokości o wartościach  $\frac{4}{3}$  i  $\frac{2}{3}$ .

Iteracja następną tworzy postać

$$\begin{aligned}\varphi^{(2)}(t) &= \frac{4}{3}\varphi^{(1)}(2t) + \frac{2}{3}\varphi^{(1)}(2t-1) \\ &= \frac{16}{9}\varphi^{(0)}(4t) + \frac{8}{9}\varphi^{(0)}(4t-1) + \frac{8}{9}\varphi^{(0)}(4t-2) + \frac{4}{9}\varphi^{(0)}(4t-3)\end{aligned}$$

uzyskując kolejne wysokości  $\frac{16}{9}$ ,  $\frac{8}{9}$ ,  $\frac{8}{9}$  i  $\frac{4}{9}$ .

Analizując następne kroki można zauważyć, że maksymalna wysokość dla kolejnych iteracji wynosi  $(\frac{4}{3})^i$ , gdzie  $i$  jest indeksem kolejnej iteracji.

Ponieważ szereg powstały dla kolejnych powtórzeń jest rozbieżny, nie można zatem użyć go do celów aproksymacyjnych.

Weryfikując ten sam filtr dolnoprzepustowy w dziedzinie częstotliwości, gdzie

$$H(z) = \frac{2}{3} + \frac{1}{3}z^{-1}, \quad z = e^{-i\omega t}$$

dla kolejnej iteracji, filtr przyjmuje wartość

$$H^{(1)}(z) = H(z)H(z^2) = \frac{4}{9} + \frac{2}{9}z^{-1} + \frac{2}{9}z^{-2} + \frac{1}{9}z^{-3}.$$

Uogólniając, wzór dla  $i$ -tej iteracji, otrzymuje się następującą zależność

$$H^{(i)}(z) = \prod_{k=0}^i H(z^{2^k}).$$

**Przykład 2** Innym przykładem zbieżności jest funkcja

$$\varphi^{(1)}(t) = \varphi^{(0)}(2t) + \varphi^{(0)}(2t-1)$$

gdzie

$$H(z) = \frac{1}{2} + \frac{1}{2}z^{-1}$$

Przechodząc przez następne iteracje otrzymuje się tą samą funkcję, ponieważ suma dwukrotnie zwężonych funkcji  $box(t)$  i przesuniętych o 1 jest oryginalną funkcją  $box(t)$ .

$$\varphi(t) = \varphi(2t) + \varphi(2t - 1) \quad (8.2)$$

Po  $i$ -tej iteracji  $H^{(i)}(z)$  posiada  $2^i$  współczynników których nośniki wynoszą odpowiednio  $2^{-i}$ .

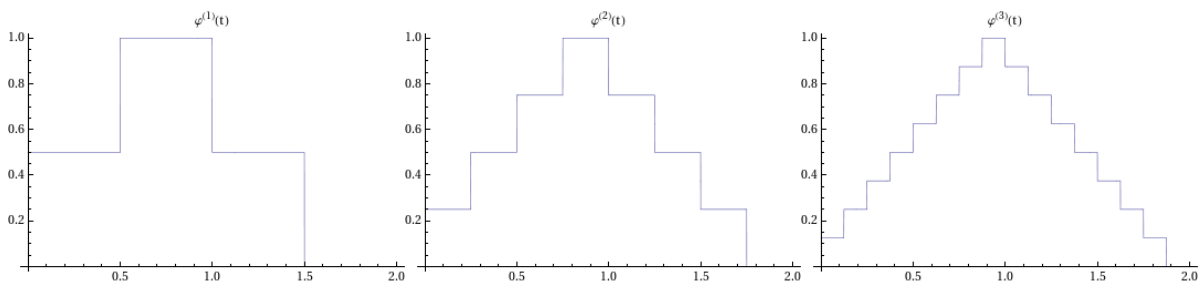
**Przykład 3** Ostatnim modelem obrazującym przypadek zbieżności jest iteracja

$$\varphi^{(1)}(t) = \frac{1}{2}\varphi^{(0)}(2t) + \varphi^{(0)}(2t - 1) + \frac{1}{2}\varphi^{(0)}(2t - 2).$$

przy czym

$$H(z) = \frac{1}{4} + \frac{1}{2}z^{-1} + \frac{1}{4}z^{-2}$$

Funkcja ta dla wartości  $\varphi(1)$  jest zbieżna do 1, zaś dla  $\varphi(2) \rightarrow 0$ .



Rysunek 8.1: Przykład zbieżności metody iteracyjnej

Badając powyższe przykłady możemy wyszczególnić następujące zależności [54] :

1. Wartość  $H(z) = \frac{2}{3} + \frac{1}{3}z^{-1}$  dla  $z = -1$  jest różna od 0 ( $\omega = \pi$ ). Iteracja nie jest zbieżna.
2. Wartość  $H(z) = \frac{1}{2} + \frac{1}{2}z^{-1}$  dla  $z = -1$  jest równa 0. Iteracja jest zbieżna.
3. Wartość  $H(z) = \frac{1}{4} + \frac{1}{2}z^{-1} + \frac{1}{4}z^{-2}$  dla  $z = -1$  ma podwójne miejsce zerowe. Iteracja zbieżna.

Wartość 0 dla  $z = -1$  nie gwarantuje tutaj zbieżności funkcji  $\varphi^i(t)$ , jednakże jest to warunek konieczny aby uzyskać zbieżność [54].

### 8.1.2 Wyznaczanie funkcji skalującej metodą iteracyjną

Do wyznaczenia funkcji skalującej metodą iteracyjną można wykorzystać współczynniki filtra **db2** które wynoszą odpowiednio

$$\begin{cases} h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}} \\ h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}} \\ h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}} \\ h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}} \end{cases}$$

gdzie  $H(z) = \sqrt{2}(h_0 + h_1z^{-1} + h_2z^{-2} + h_3z^{-3}) = \frac{1+\sqrt{3}}{4} + \frac{3+\sqrt{3}}{4}z^{-1} + \frac{3-\sqrt{3}}{4}z^{-2} + \frac{1-\sqrt{3}}{4}z^{-3}$ .

Ponieważ  $H(-1) = \frac{1+\sqrt{3}}{4} - \frac{3+\sqrt{3}}{4} + \frac{3-\sqrt{3}}{4} - \frac{1-\sqrt{3}}{4} = 0$  warunek konieczny zbieżności został spełniony.

---

#### Algorytm 8.1 Wyznaczanie funkcji skalującej metodą iteracyjną

---

```

box[x_] := 1 /; x >= 0 && x <= 1;
box[x_] := 0;

h[0] = (1 + Sqrt[3]) / (4 Sqrt[2]);
h[1] = (3 + Sqrt[3]) / (4 Sqrt[2]);
h[2] = (3 - Sqrt[3]) / (4 Sqrt[2]);
h[3] = (1 - Sqrt[3]) / (4 Sqrt[2]);

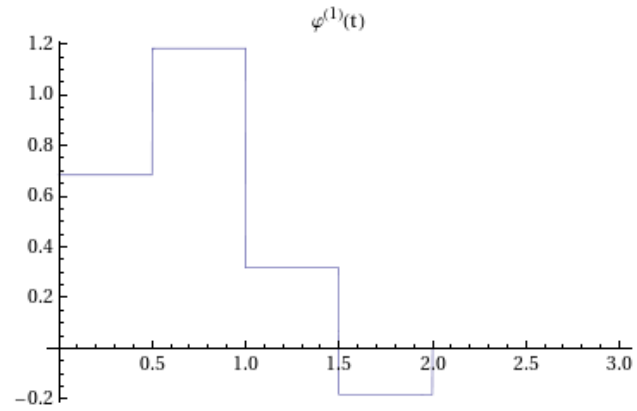
scalingfunctionbybox::usage = "Wyznaczanie iteracyjne funkcji skalujacej";
scalingfunctionbybox[x_, n_] := Module[{},
  If[n == 0,
    Return[Sqrt[2] * (h[0] * box[2 x] + h[1] * box[2 x - 1] + h[2] * box[2 x - 2] + h[3] * box[2 x - 3])],
    Return[Sqrt[2] * (h[0] * scalingfunctionbybox[2 x, n - 1] + h[1] * scalingfunctionbybox[2 x - 1, n - 1] +
      h[2] * scalingfunctionbybox[2 x - 2, n - 1] + h[3] * scalingfunctionbybox[2 x - 3, n - 1])]
  ];
];

```

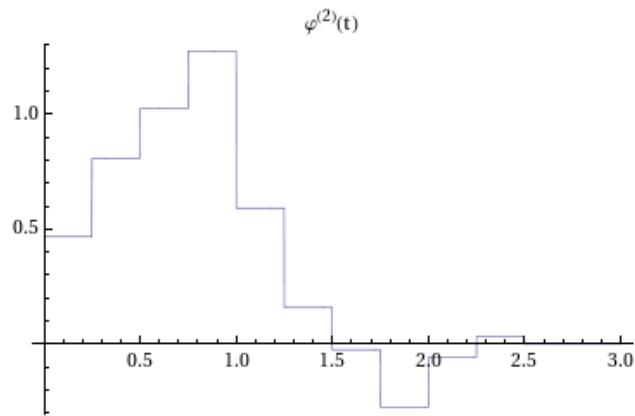
---

Poniżej zaprezentowane zostały wyniki dla kolejnych iteracji.

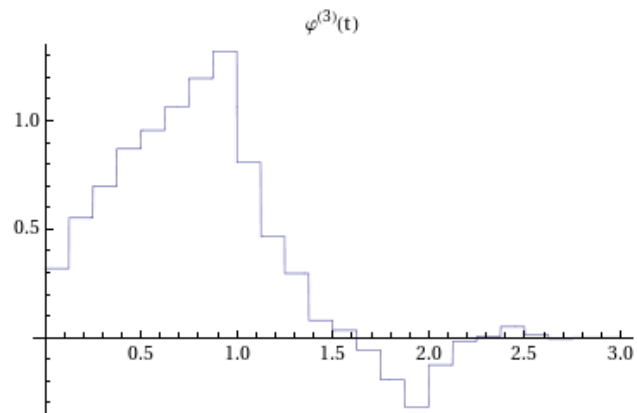
```
Plot[scalingfunctionbybox[x, 0], {x, 0, 3}, PlotLabel -> "φ(1)(t)"]
```



```
Plot[scalingfunctionbybox[x, 1], {x, 0, 3}, PlotLabel -> "φ(2)(t)"]
```

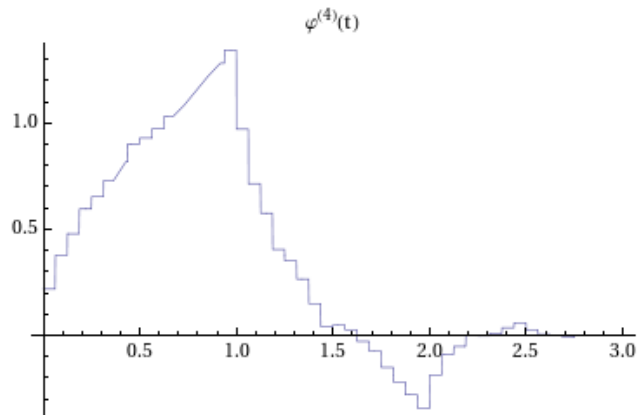


```
Plot[scalingfunctionbybox[x, 2], {x, 0, 3}, PlotLabel -> "φ(3)(t)"]
```

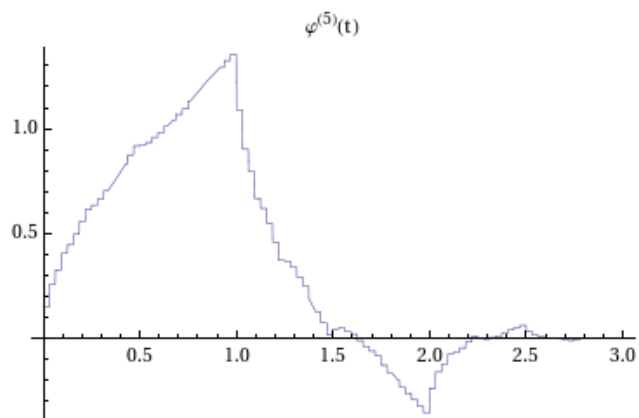


Rysunek 8.2: Funkcja skalująca **db2** uzyskana metodą iteracyjną dla kolejnych iteracji cz.I

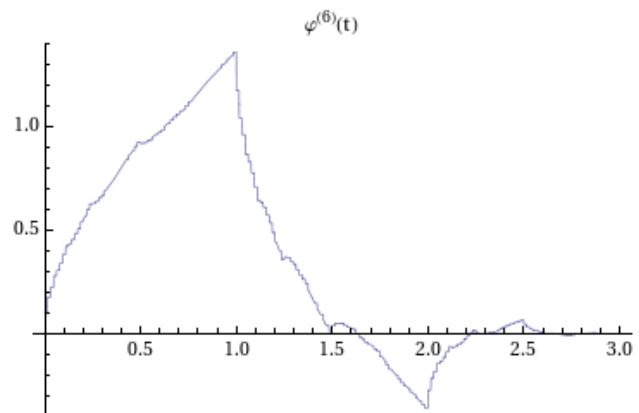
```
Plot[scalingfunctionbybox[x, 3], {x, 0, 3}, PlotLabel -> " $\varphi^{(4)}(t)$ "]
```



```
Plot[scalingfunctionbybox[x, 4], {x, 0, 3}, PlotLabel -> " $\varphi^{(5)}(t)$ "]
```



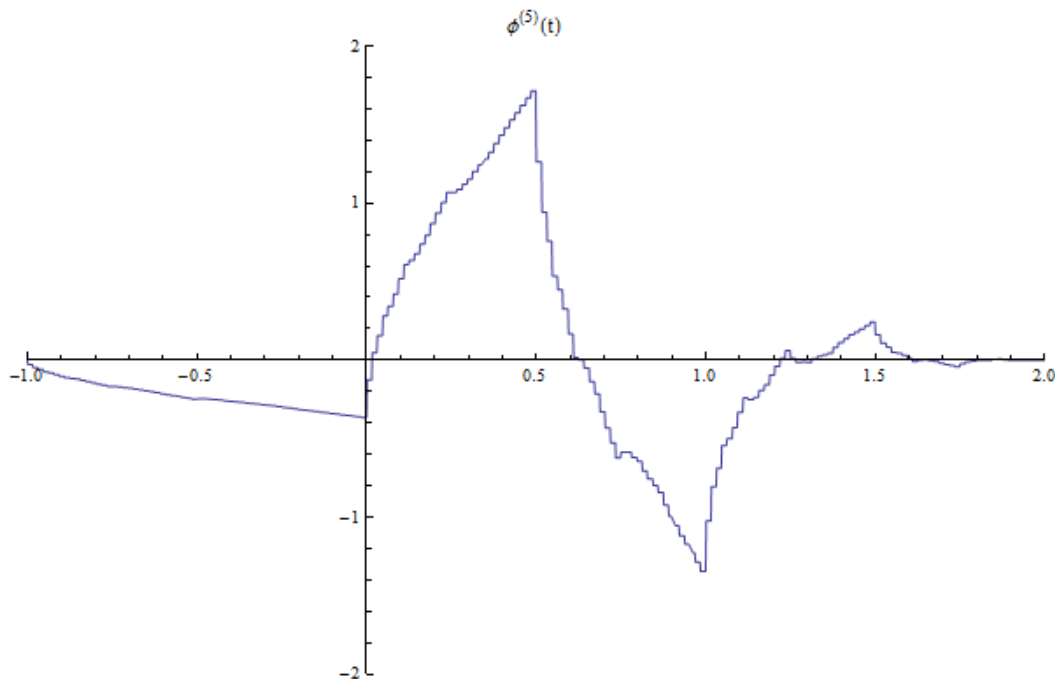
```
Plot[scalingfunctionbybox[x, 5], {x, 0, 3}, PlotLabel -> " $\varphi^{(6)}(t)$ "]
```



Rysunek 8.3: Funkcja skalująca **db2** uzyskana metodą iteracyjną dla kolejnych iteracji cz. II

Wykorzystując powyższy algorytm iteracyjny 8.1 można uzyskać przybliżony wykres funkcji falkowej

```
waveletsbybox::usage = "iteracyjne wyznaczenie funkcji falkowej";
waveletsbybox[x_, i_] :=
  Module[{s, n}, s = 3; Return[Sqrt[2] Sum[(-1)^n h[-n+1] scalingfunctionbybox[2 x - n, i]]];
Plot[waveletsbybox[x, 4], {x, -2, 3}, PlotRange -> {{-1, 2}, {-2, 2}}, PlotLabel -> "φ(5)(t)"]
```



Rysunek 8.4: Funkcja falkowa **db2** uzyskana metodą iteracyjną dla  $i=5$

## 8.2 Algorytm kaskadowy - wyznaczenie wartości w dziedzinie częstotliwościowej

Ponieważ równanie funkcji skalującej wyraża się wzorem

$$\varphi(x) = \sqrt{2} \sum_{n=s_1}^{s_2} h_n \varphi(2x - n)$$

przekształcając obustronnie powyższy wzór za pomocą transformaty Fouriera, otrzymuje się kolejno



$$\begin{aligned}
\Phi(\omega) &= \int_{-\infty}^{\infty} \varphi(x) e^{-i\omega x} dx \\
&= \sqrt{2} \sum_{n=s_1}^{s_2} h_n \int_{-\infty}^{\infty} \varphi(2x - n) e^{-i\omega x} dx = \\
&= \frac{\sqrt{2}}{2} \sum_{n=s_1}^{s_2} h_n \int_{-\infty}^{\infty} \varphi(t) e^{-i\omega(\frac{t+n}{2})} dt = \\
&= \frac{\sqrt{2}}{2} \sum_{n=s_1}^{s_2} h_n e^{-\frac{i\omega n}{2}} \int_{-\infty}^{\infty} \varphi(t) e^{-\frac{i\omega t}{2}} dt
\end{aligned}$$

zatem

$$\Phi(\omega) = \frac{\sqrt{2}}{2} \sum_{n=s_1}^{s_2} h_n e^{-\frac{i\omega n}{2}} \Phi\left(\frac{\omega}{2}\right). \quad (8.3)$$

Dzięki zależności 8.3 uzyskuje się kaskadowy algorytm na obliczanie wartości funkcji skalującej w dziedzinie częstotliwości:

$$\begin{aligned}
\Phi(\omega) &= \frac{\sqrt{2}}{2} \sum_{n=s_1}^{s_2} h_n e^{-\frac{i\omega n}{2}} \Phi\left(\frac{\omega}{2}\right) \\
\Phi\left(\frac{\omega}{2}\right) &= \frac{\sqrt{2}}{2} \sum_{n=s_1}^{s_2} h_n e^{-\frac{i\omega n}{2^2}} \Phi\left(\frac{\omega}{2^2}\right) \\
\vdots &= \vdots \\
\Phi\left(\frac{\omega}{2^{k-1}}\right) &= \frac{\sqrt{2}}{2} \sum_{n=s_1}^{s_2} h_n e^{-\frac{i\omega n}{2^k}} \Phi\left(\frac{\omega}{2^k}\right)
\end{aligned}$$

skąd, stosując metodę podstawiania dla powyższego układu otrzymuje się

$$\Phi(\omega) = \left(\frac{\sqrt{2}}{2}\right)^l \Phi\left(\frac{\omega}{2^l}\right) \prod_{k=1}^l \sum_{n=s_1}^{s_2} h_n e^{-\frac{i\omega n}{2^k}} \quad (8.4)$$

Do celów praktycznych można wykorzystać wzór w postaci granicznej [6, 5, 53, 54], gdzie

$$\Phi(\omega) = \lim_{l \rightarrow \infty} \left(\frac{\sqrt{2}}{2}\right)^l \Phi\left(\frac{\omega}{2^l}\right) \prod_{k=1}^l \sum_{n=s_1}^{s_2} h_n e^{-\frac{i\omega n}{2^k}} = \Phi(0) \prod_{k=1}^{\infty} \frac{\sqrt{2}}{2} \sum_{n=s_1}^{s_2} h_n e^{-\frac{i\omega n}{2^k}}.$$

Wiedząc, że

$$\Phi(0) = \int_{-\infty}^{\infty} \varphi(x) e^{-i0x} dx = \int_{-\infty}^{\infty} \varphi(x) dx = 1$$

postać funkcji skalującej w dziedzinie częstotliwości, w konsekwencji wyraża się wzorem

$$\Phi(\omega) = \prod_{k=1}^{\infty} \frac{\sqrt{2}}{2} \sum_{n=s_1}^{s_2} h_n e^{-\frac{i\omega n}{2^k}} \quad (8.5)$$

gdzie stopień precyzji określa się liczbą iteracji  $k$ .



## Część III

# Analiza wielorozdzielcza z wykorzystaniem przestrzeni ortogonalnych



## Rozdział 9

---

# Analiza wielorozdzielcza

---

### 9.1 Skalowanie funkcji

Analiza wielorozdzielcza za pomocą falek ortogonalnych jest ściśle związana analizą i syntezą funkcji badanej przy użyciu falek  $\varphi_{mn}$  i  $\psi_{mn}$  [2, 70] otrzymywanych poprzez translację i dylatację tzw. funkcji podstawowych gdzie  $\varphi(x)$  i  $\psi(x)$ .

Funkcja  $\varphi(x)$  jest funkcją skalującą (falką - ojcem) wykorzystywaną w celu otrzymania reprezentacji zgrubnej, zaś  $\psi(x)$  - funkcją falkową (falką - matką) biorącą udział w wyznaczaniu reprezentacji szczegółowej, gdzie

$$\begin{aligned}\psi_{mn}(t) &= 2^{\frac{m}{2}} \psi(2^m x - n) \\ \varphi_{mn}(t) &= 2^{\frac{m}{2}} \varphi(2^m x - n)\end{aligned}\tag{9.1}$$

Dla falek o nośnikach zwartych, ze względu na skalowanie, nośniki przyjmują postać [5]:

$$\begin{aligned}Supp \varphi_{mn}(x) &= \left[ \frac{s_1+n}{2^m}, \frac{s_2+n}{2^m} \right] \\ Supp \psi_{mn}(x) &= \left[ \frac{t_1+n}{2^m}, \frac{t_2+n}{2^m} \right]\end{aligned}\tag{9.2}$$

Ponieważ przy wzroście skali falki nośnik staje się coraz to węższy, można zauważyć, że przy dużej skali  $m$  nośniki są skupione wokół punktów centralnych falek

$$\begin{aligned}x_{\varphi_{mn}} &= \frac{\frac{s_1+s_2}{2} + n}{2^m} \\ x_{\psi_{mn}} &= \frac{\frac{t_1+t_2}{2} + n}{2^m} = \frac{\frac{1}{2} + n}{2^m}\end{aligned}\tag{9.3}$$

### 9.2 Własności analizy wielorozdzielczej

Falkowe metody wielorozdzielcze są szczególnie przydatne do analiz czasowo-częstotliwościowych szeregów niestacjonarnych.

W analizie zgodnie z postulatami 1 i 2 z rozdziału 3 przyjmuje się, że istnieją ciągi  $V_M, V_{M-1}, \dots, V_m, \dots$  (gdzie  $m \in \mathbb{Z}$ ) przestrzeni wzajemnie zanurzonych w  $L^2(\mathbb{R})$  [14, 24, 51, 55] takich, że :

$$V_M \subset V_{M+1} \subset \dots \subset V_m \subset \dots \subset L^2(\mathbb{R}) \quad (9.4)$$

gdzie

$$\lim_{m \rightarrow -\infty} V_m = \{0\} \quad (9.5)$$

$$\overline{\lim_{m \rightarrow \infty} V_m} = L^2(\mathbb{R}) \quad (9.6)$$

Każda z przestrzeni  $V_m$  posiada bazę ortonormalną  $\{v_{mn}(x)\}_{n \in \mathbb{Z}}$ , gdzie

$$V_m = \text{span}\{v_{mn}(x), n \in \mathbb{Z}\}. \quad (9.7)$$

zaś, odpowiadające im przestrzenie dopełnień (detali)  $W_m$  - bazę  $\{w_{mn}(x)\}_{n \in \mathbb{Z}}$  tj.

$$W_m = \text{span}\{w_{mn}(x), n \in \mathbb{Z}\}. \quad (9.8)$$

Ponadto, przestrzenie  $V_i$  i  $W_j$ , gdzie  $i, j \in \mathbb{Z}$ , są przestrzeniami wzajemnie ortogonalnymi

$$V_i \perp W_j. \quad (9.9)$$

Dzięki powyższym własnościom wynika, że analiza wielorozdzielcza jest ściśle związana z tzw. transformatą falkową. Polega ona na wielopoziomowej, zdekomponowanej reprezentacji funkcji w postaci szczegółowej i zgrubnej. Przy czym, każda następną reprezentacją zgrubną na poziomie wyższym może być przedstawiona jako suma reprezentacji szczegółowej i zgrubnej na poziomie niższym [13, 17] tj.

$$V_{j+1} = V_j \oplus W_j \quad (9.10)$$

gdzie  $W_j$  jest dopełnieniem  $V_{j+1}$  w  $V_j$ .  
W konsekwencji otrzymuje się [73, 3, 40]

$$V_j = V_k \oplus W_k \oplus \dots \oplus W_{j-1} = V_k \oplus \bigoplus_{i=k}^{j-1} W_i \quad (9.11)$$

i

$$L^2(\mathbb{R}) = V_k \oplus \bigoplus_{i=k}^{\infty} W_i \quad (9.12)$$

Powyższe wyrażenia pozwalają na przejście z niższego poziomu rozdzielczości do wyższego, uwydatniając przy tym detale przestrzeni, niewidoczne w przestrzeni o niższym poziomie szczegółowości.

Funkcję aproksymowaną na poziomie  $j$ -tym definiuje się często na dwa sposoby [72].

W pierwszym przypadku jest to **model zagregowany**

$$F(x; m) = \sum_{j \in \mathbb{Z}} \alpha_{mj} \varphi_{mj}(x) \quad (9.13)$$

lub w przypadku drugim, na odpowiadający mu **model zdekomponowany**

$$F(x; m) = \sum_{j \in \mathbb{Z}} \alpha_{kj} \varphi_{kj}(x) + \sum_{i=k}^{m-1} \sum_{j \in \mathbb{Z}} \beta_{ij} \psi_{ij}(x) \quad (9.14)$$

gdzie w sytuacji granicznej model ten można wyrazić wzorem

$$F(x) = \sum_{j=-\infty}^{\infty} \alpha_{kj} \varphi_{kj}(x) + \sum_{i=k}^{\infty} \sum_{j=-\infty}^{\infty} \beta_{ij} \psi_{ij}(x). \quad (9.15)$$

Współczynniki  $\alpha_{kj}$  i  $\beta_{ij}$  dzięki własności ortonormalności funkcji skalujących i falkowych wyznacza się jako następujące iloczyny skalarne

$$\begin{aligned} \alpha_{kj} &= \int_{-\infty}^{\infty} F(x) \varphi_{kj} dx \\ \beta_{ij} &= \int_{-\infty}^{\infty} F(x) \psi_{ij} dx \end{aligned} \quad (9.16)$$

Analizując model zagregowany można zauważyć, że podczas aproksymacji traci on informację o szczegółach ponieważ wykorzystuje do przybliżeń funkcji badanej jedynie współczynniki reprezentacji zgrubej. Model zdekomponowany, w przeciwieństwie do zagregowanego, analizuje funkcję na  $m$ -tym poziomie rozdzielczości, co też pozwala na szczegółową jej analizę na wielu poziomach. Dzięki tej własności, model zdekomponowany jest w praktyce częściej wykorzystywany podczas dekompozycji funkcji ze względu na dodatkowy nośnik informacji szczegółowej [14].



### 9.3 Falkowa analiza wielorozdzielcza

Dla funkcji  $F(x)$  najlepszą aproksymacją w skali  $m$  jest rzut ortogonalny funkcji na podprzestrzeń  $V_k$  oznaczany jako  $F(x, m)$  [11, 3, 70, 16], gdzie

$$F(x; m) = \sum_{j \in \mathbb{Z}} \alpha_{kj} \varphi_{kj}(x) + \sum_{i=k}^{m-1} \sum_{j \in \mathbb{Z}} \beta_{ij} \psi_{ij}(x) \quad (9.17)$$

Aproksymator  $F(x; m)$  jest zatem, skończonym szeregiem falkowym ograniczonym do odpowiedniej liczby członów, uzależnionych od stopnia aproksymacji  $m$ .

Ponieważ

$$F(x) = \sum_{j=-\infty}^{\infty} \alpha_{kj} \varphi_{kj}(x) + \sum_{i=k}^{\infty} \sum_{j=-\infty}^{\infty} \beta_{ij} \psi_{ij}(x)$$

można go zapisać w postaci równoważnej jako

$$F(x) = F(x; m) + \sum_{i=m}^{\infty} \sum_{j=-\infty}^{\infty} \beta_{ij} \psi_{ij}(x) \quad (9.18)$$

gdzie błąd aproksymacji wyraża się wzorem

$$\delta(F; x; m) = F(x) - F(x; m) = \sum_{i=m}^{\infty} \sum_{j=-\infty}^{\infty} \beta_{ij} \psi_{ij}(x) \quad (9.19)$$

Wynika z tego, że model zdekomponowany pozwala na iteracyjne szacowanie aproksymacji falkowej poprzez dodawanie kolejnych współczynników przestrzeni detali tj.

$$F(x; m+1) = F(x; m) + \sum_{j=-\infty}^{\infty} \beta_{mj} \psi_{mj}(x) \quad (9.20)$$

#### 9.3.1 Aproksymacja za pomocą falek o nośniku zwartym

Falki o zwartym nośniku pozwalają aproksymować funkcję dla każdego argumentu funkcji aproksymowanej za pomocą lokalnych sum o skończonej liczbie składników (postaci zagregowanej) tj. [75]

$$F(x; m) = \sum_{j=n_{\min}(\varphi, x, m)}^{n_{\max}(\varphi, x, m)} \alpha_{mj} \varphi_{mj}(x) \quad (9.21)$$

lub równoważnie w postaci zdekomponowanej

$$F(x; m) = \sum_{j=n_{min}(\varphi, x, k)}^{n_{max}(\varphi, x, k)} \alpha_{kj} \varphi_{kj}(x) + \sum_{i=k}^{m-1} \sum_{j=n_{min}(\psi, x, k)}^{n_{max}(\psi, x, k)} \beta_{ij} \psi_{ij}(x) \quad (9.22)$$

gdzie granicami sumowania są

$$\begin{aligned} n_{min}(\varphi, x, k) &= \lceil 2^k x - s_2 \rceil \\ n_{max}(\varphi, x, k) &= \lfloor 2^k x - s_1 \rfloor \end{aligned} \quad (9.23)$$

oraz

$$\begin{aligned} n_{min}(\psi, x, k) &= \lceil 2^k x - t_2 \rceil \\ n_{max}(\psi, x, k) &= \lfloor 2^k x - t_1 \rfloor \end{aligned} \quad (9.24)$$

Wykorzystując powyższą zależność, wzory na współczynniki falkowe dla modelu zagregowanego wynoszą odpowiednio, dla postaci zagregowanej

$$\alpha_{ij} = \int_{\frac{s_1+j}{2^i}}^{\frac{s_2+j}{2^i}} F(x) \varphi_{ij}(x) \quad (9.25)$$

zaś dla modelu zdekomponowanego

$$\begin{aligned} \alpha_{kj} &= \int_{\frac{s_1+j}{2^k}}^{\frac{s_2+j}{2^k}} F(x) \varphi_{kj}(x) \\ \beta_{ij} &= \int_{\frac{t_1+j}{2^i}}^{\frac{t_2+j}{2^i}} F(x) \psi_{ij}(x) \end{aligned} \quad (9.26)$$

### 9.3.2 Aproksymacja falkowa funkcją Haara

Aby zobrazować analizę wielorozdzielczą za pomocą przestrzeni ortogonalnych i zwartych do tego celu można wykorzystać tzw. funkcje Haara.

Dla potrzeb przykładu, funkcja która będzie podlegała aproksymacji będzie miała postać

$$F(x) = \begin{cases} \sin(x) \cos^2(x) & \text{dla } x \in \langle -\pi, \pi \rangle \\ 0 & \text{dla } x \notin \langle -\pi, \pi \rangle \end{cases} \quad (9.27)$$

Ponieważ funkcja aproksymacyjna jest funkcją Haara [55]

$$H(x) = \begin{cases} 1 & \text{dla } x \in \langle 0, \frac{1}{2} \rangle \\ -1 & \text{dla } x \in \langle \frac{1}{2}, 1 \rangle \\ 0 & \text{dla } x \notin \langle 0, 1 \rangle \end{cases} \quad (9.28)$$

rodzina funkcji falkowych  $H_{ij}(x) = 2^{\frac{i}{2}} H(2^i x - j)$  wyraża się wzorem [8, 73, 37]

$$H_{ij}(x) = \begin{cases} 2^{\frac{i}{2}} & \text{dla } x \in \langle j2^{-i}, (j + \frac{1}{2})2^{-i} \rangle \\ -2^{\frac{i}{2}} & \text{dla } x \in \langle (j + \frac{1}{2})2^{-i}, (j + 1)2^{-i} \rangle \\ 0 & \text{dla } x \notin \langle j2^{-i}, (j + 1)2^{-i} \rangle \end{cases} \quad (9.29)$$

W związku z tym, że funkcja 9.27 będzie aproksymowana na przedziale  $\langle -2^{-p}, 2^{-p} \rangle$  wystarczy przyjąć, że poziom  $p$  jest równy  $-2[3]$ .

Poziom ten jest wystarczający, ponieważ funkcja aproksymowana przyjmuje niezerowe wartości jedynie w przedziale  $\langle -\pi, \pi \rangle$ .

Analizując funkcje falkowe gdzie nośnikiem jest przedział

$$\text{Supp } H_{ij}(x) = \langle j2^{-i}, (j + 1)2^{-i} \rangle \quad (9.30)$$

można zauważyć, że dla  $i < p$  przedział  $\langle 0, 2^{-p} \rangle$  zawarty jest w lewej połowie nośników funkcji  $H_{j0}$ , natomiast przedział  $\langle -2^{-p}, 0 \rangle$  - w prawej połowie nośników  $H_{j,-1}$  [73].

Dzięki tej zależności w prosty sposób uzyskuje się aproksymację  $F(x)$  za pomocą aproksymatora  $F(x; p)$ .

Ponieważ każdy poziom rozdzielczości  $i$  (gdzie  $i < p$ ), ze względu na zerowe wartości pozostałych współczynników zależy tylko od współczynników  $\beta_{i,0}$  i  $\beta_{i,-1}$  takich, że

$$\beta_{i,0} = \int_0^{2^{-p}} 2^{\frac{i}{2}} F(x) dx$$

oraz

$$\beta_{i,-1} = \int_{-2^{-p}}^0 2^{\frac{i}{2}} F(x) dx,$$

aproksymator można przedstawić w formie

$$F(x; m) = F(x; p) + \sum_{i=p}^{m-1} \sum_{j=-2^{i-p}}^{2^{i-p}-1} \beta_{ij} H_{ij}(x),$$

gdzie

$$F(x; p) = \begin{cases} \sum_{i < p} \beta_{i,0} H_{i,0} & \text{dla } x \in \langle 0, 2^{-p} \rangle \\ \sum_{i < p} \beta_{i,-1} H_{i,-1} & \text{dla } x \in \langle -2^{-p}, 0 \rangle \end{cases}.$$

W konsekwencji aproksymacja na poziomie  $p$

$$F(x; p) = \begin{cases} \sum_{i < p} (\int_0^{2^{-p}} 2^{\frac{i}{2}} F(x) dx) 2^{\frac{i}{2}} = \int_0^{2^{-p}} F(x) dx \sum_{i < p} 2^i & \text{dla } x \in \langle 0, 2^{-p} \rangle \\ \sum_{i < p} (\int_{-2^{-p}}^0 2^{\frac{i}{2}} F(x) dx) 2^{\frac{i}{2}} = \int_{-2^{-p}}^0 F(x) dx \sum_{i < p} 2^i & \text{dla } x \in \langle -2^{-p}, 0 \rangle \end{cases}$$

upraszcza się do postaci[73]

$$F(x; p) = \begin{cases} 2^p \int_0^{2^{-p}} F(x) dx & \text{dla } x \in \langle 0, 2^{-p} \rangle \\ 2^p \int_{-2^{-p}}^0 F(x) dx & \text{dla } x \in \langle -2^{-p}, 0 \rangle \end{cases} \quad (9.31)$$

## 9.4 Aproksymacja Haara - implementacja

---

### Algorytm 9.1 Aproksymacja Haara

---

```

calka::usage = "Obliczanie calki metoda trapezow";
calka[f_, a_, b_, n_] :=
  Sum[N[(f[i] + f[i + (b - a) / n]) / 2 * (b - a) / n], {i, a, b - (b - a) / n, (b - a) / n}];

(* liczba podzialu przedzialu calkowania *)
iteracja = 200;

f::usage = "Definicja funkcji aproksymowanej";
f[x_] := 0;
f[x_] := Sin[x] * Cos[x]^2 /; x >= -Pi && x < Pi;

H::usage = "Definicja falki Haara";
H[x_] := 1 /; x >= 0 && x < 1/2;
H[x_] := -1 /; x >= 1/2 && x < 1;
H[x_] := 0;

Hjk::usage = "Definicja rodziny falek";
Hjk[x_, j_, k_] := 2^(j/2) H[2^j x - k];

Fp::usage = "Wartosc funkcji na p-tym poziomie";
Fp[f_, x_, p_] := 0 /; x < -2^(-p) || x > 2^(-p);
Fp[f_, x_, p_] := 2^p * calka[f, -2^(-p), 0, iteracja] /; x >= -2^(-p) && x < 0;
Fp[f_, x_, p_] := 2^p * calka[f, 0, 2^(-p), iteracja] /; x >= 0 && x < 2^(-p);

iloczynfunkcji::usage = "Funkcja pomocnicza podczas obliczania wspolczynnika";
iloczynfunkcji[f_, x_, j_, k_] := N[f[x] Hjk[x, j, k]];

bij::usage = "Algorytm na obliczanie wspolczynnika bij";
bij[f_, a_, b_, i_, j_, n_] :=
  Sum[N[1/2^n (iloczynfunkcji[f, x, i, j] + iloczynfunkcji[f, x + (b - a)/n, i, j]) (b - a)],
  {x, a, b - (b - a)/n, (b - a)/n}];

funkcjaaproksymowana::usage = "Obliczanie wartosci funkcji aproksymowanej";
funkcjaaproksymowana[f_, x_, p_, m_] :=
  Fp[f, x, p] + Sum[Sum[2^(i-p)-1 (bij[f, -2^(-p), 2^(-p), i, j, iteracja] * Hjk[x, i, j])],
  i=p, m-1];

przestrzendetali::usage = "Obliczanie wartosci funkcji detali o szczegolach poziomie m+1";
przestrzendetali[f_, x_, p_, m_] :=
  Module[{i = m}, Sum[2^(i-p)-1 (bij[f, -2^(-p), 2^(-p), i, j, iteracja] * Hjk[x, i, j])],
  j=2^(i-p)];

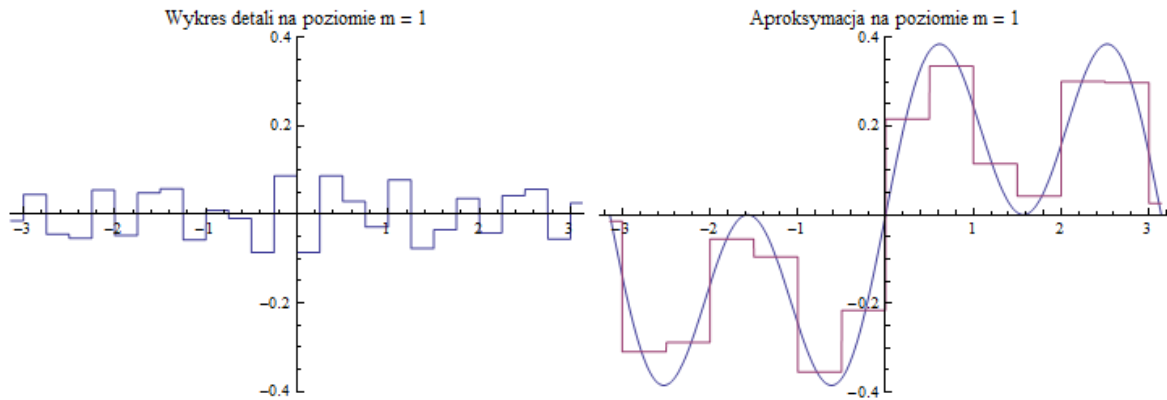
```

---

```

TableForm[
  {{Plot[przestrzendetali[f, x, -2, 1], {x, -Pi, Pi}, PlotRange -> {{-Pi, Pi}, {-0.4, 0.4}},
    PlotLabel -> "Wykres detali na poziomie m = 1",
    Plot[{f[x], funkcjaaproxymowana[f, x, -2, 1]}, {x, -Pi, Pi}, PlotLabel -> "Aproxymacja na poziomie m = 1"]}]}

```

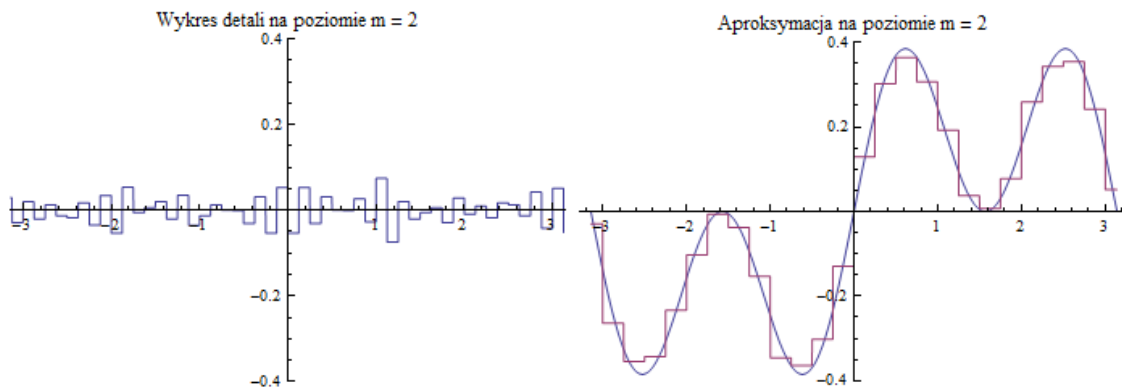


Rysunek 9.1: Aproxymacja Haara na poziomie rozdzielczości I

```

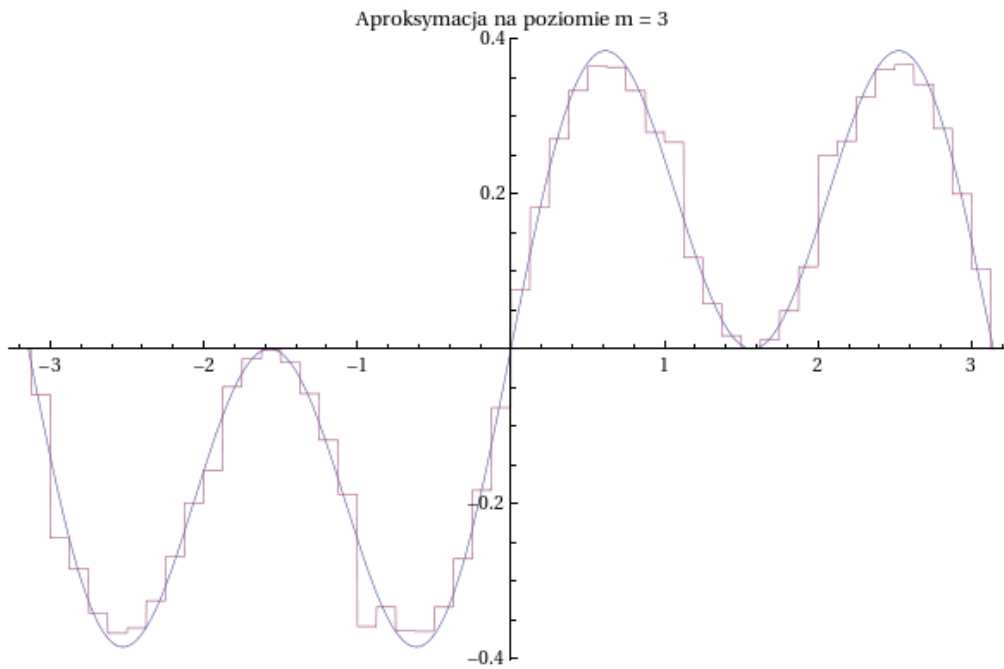
TableForm[
  {{Plot[przestrzendetali[f, x, -2, 2], {x, -Pi, Pi}, PlotRange -> {{-Pi, Pi}, {-0.4, 0.4}},
    PlotLabel -> "Wykres detali na poziomie m = 2",
    Plot[{f[x], funkcjaaproxymowana[f, x, -2, 2]}, {x, -Pi, Pi}, PlotLabel -> "Aproxymacja na poziomie m = 2"]}]}

```



Rysunek 9.2: Aproxymacja Haara na poziomie rozdzielczości II

```
Plot[{f[x], funkcjaaproxymowana[f, x, -2, 3]}, {x, -Pi, Pi}, PlotLabel -> "Aproxymacja na poziomie m = 3"]
```



Rysunek 9.3: Aproxymacja Haara na poziomie rozdzielczości III

## 9.5 Algorytm Mallata

Analiza wielorozdzielcza z wykorzystaniem aproksymacji falkowej dla nośników zwartych, wiąże się z wykonaniem transformaty falkowej dla każdego współczynnika falkowego osobno. Konsekwencją tego, jest duży koszt obliczeniowy wymagający rachunku całkowego, gdzie funkcją podcałkową jest iloczyn funkcji aproksymowanej z funkcją skalującą, czy też falkową dla danego modelu zdekomponowanego. Istnieje jednak metoda która pozwala na znaczne zmniejszenie stopnia złożoności obliczeniowej. Metoda ta pozwala wyliczyć w sposób bezpośredni, współczynniki poziomów niższych na podstawie liniowej zależności współczynników falkowych stopnia wyższego [75].

Korzystając z wyrażeń

$$\varphi(x) = \sqrt{2} \sum_{n=s_1}^{s_2} h_n \varphi(2x - n)$$

$$\psi(x) = \sqrt{2} \sum_{n=-s_2+1}^{-s_1+1} g_n \varphi(2x - n)$$

i podstawiając w miejsce  $x \rightarrow 2^m x$ , otrzymuje się

$$\varphi(2^m x) = \sqrt{2} \sum_{n=s_1}^{s_2} h_n \varphi(2^{m+1} x - n)$$

$$\psi(2^m x) = \sqrt{2} \sum_{n=-s_2+1}^{-s_1+1} g_n \varphi(2^{m+1} x - n)$$

lub równoważnie

$$\begin{aligned}\varphi_{m0}(x) &= \sum_{n=s_1}^{s_2} h_n \varphi_{m+1,n}(x) \\ \psi_{m0}(x) &= \sum_{n=-s_2+1}^{-s_1+1} g_n \varphi_{m+1,n}(x)\end{aligned}\tag{9.32}$$

co wynika bezpośrednio z postulatu 3 i 4 opisanego w rozdziale 3.

Podstawiając w miejsce argumentu  $x$  odpowiednio  $x - \frac{n_\varphi(m)}{2^m}$  i  $x - \frac{n_\psi(m)}{2^m}$  wyrażenie 9.32 przyjmuje poniższą postać

$$\begin{aligned}\varphi_{m0}\left(x - \frac{n_\varphi(m)}{2^m}\right) &= \sum_{n=s_1}^{s_2} h_n \varphi_{m+1,n}\left(x - \frac{n_\varphi(m)}{2^m}\right) \\ \psi_{m0}\left(x - \frac{n_\psi(m)}{2^m}\right) &= \sum_{n=-s_2+1}^{-s_1+1} g_n \varphi_{m+1,n}\left(x - \frac{n_\psi(m)}{2^m}\right)\end{aligned}\tag{9.33}$$

Na podstawie przekształceń

$$\begin{aligned}\varphi_{m0}(x) &= 2^{\frac{m}{2}} \varphi(2^m x) \\ \varphi_{m0}\left(x - \frac{n}{2^m}\right) &= 2^{\frac{m}{2}} \varphi\left(2^m \left(x - \frac{n}{2^m}\right)\right) \\ &= 2^{\frac{m}{2}} \varphi(2^m x - n)\end{aligned}$$

można wyznaczyć odpowiednio zależności funkcji skalujących, gdzie

$$\begin{aligned}\varphi_{mn}(x) &= 2^{\frac{m}{2}} \varphi(2^m x - n) \\ \varphi_{mn}(x) &= \varphi_{m0}\left(x - \frac{n}{2^m}\right)\end{aligned}\tag{9.34}$$

Analogicznie dla funkcji falkowych otrzymuje się

$$\begin{aligned}\psi_{mn}(x) &= 2^{\frac{m}{2}} \psi(2^m x - n) \\ \psi_{mn}(x) &= \psi_{m0}\left(x - \frac{n}{2^m}\right)\end{aligned}\tag{9.35}$$

Na bazie wyznaczonych równań 9.34 i 9.35, podstawiając je do wyrażenia 9.33 oblicza się związki [75]



$$\begin{aligned}
\varphi_{mn_{\varphi(m)}}(x) &= \sum_{n=s_1}^{s_2} h_n 2^{\frac{m+1}{2}} \varphi(2^{m+1}(x - \frac{n_{\varphi(m)}}{2^m}) - n) \\
&= \sum_{n=s_1}^{s_2} h_n 2^{\frac{m+1}{2}} \varphi(2^{m+1}x - (n + 2n_{\varphi(m)})) \\
\psi_{mn_{\psi(m)}}(x) &= \sum_{n=-s_2+1}^{-s_1+1} g_n 2^{\frac{m+1}{2}} \varphi(2^{m+1}(x - \frac{n_{\psi(m)}}{2^m}) - n) \\
&= \sum_{n=-s_2+1}^{-s_1+1} g_n 2^{\frac{m+1}{2}} \varphi(2^{m+1}x - (n + 2n_{\psi(m)}))
\end{aligned}$$

lub równoważnie

$$\begin{aligned}
\varphi_{mn_{\varphi(m)}}(x) &= \sum_{n=s_1}^{s_2} h_n \varphi_{m+1, n+2n_{\varphi(m)}}(x) \\
\psi_{mn_{\psi(m)}}(x) &= \sum_{n=-s_2+1}^{-s_1+1} g_n \varphi_{m+1, n+2n_{\psi(m)}}(x)
\end{aligned} \tag{9.36}$$

W końcowym etapie, mnożąc obustronnie równania przez  $F(x)$  i następnie całkując otrzymuje się wyrażenia

$$\begin{aligned}
\underbrace{\int_{-\infty}^{\infty} F(x) \varphi_{mn_{\varphi(m)}}(x) dx}_{\alpha_{mn_{\varphi(m)}}} &= \sum_{n=s_1}^{s_2} h_n \underbrace{\int_{-\infty}^{\infty} F(x) \varphi_{m+1, n+2n_{\varphi(m)}}(x) dx}_{\alpha_{m+1, n+2n_{\varphi(m)}}} \\
\underbrace{\int_{-\infty}^{\infty} F(x) \psi_{mn_{\psi(m)}}(x) dx}_{\beta_{mn_{\psi(m)}}} &= \sum_{n=-s_2+1}^{-s_1+1} g_n \underbrace{\int_{-\infty}^{\infty} F(x) \varphi_{m+1, n+2n_{\psi(m)}}(x) dx}_{\alpha_{m+1, n+2n_{\psi(m)}}}
\end{aligned} \tag{9.37}$$

Dzięki powyższej formule uzyskuje się metodę wyznaczania współczynników falkowych  $\alpha_{mn}$  i  $\beta_{mn}$ , które prowadzą bezpośrednio do **algorytmu Mallata** [70, 3, 30]

$$\begin{aligned}
\alpha_{mn_{\varphi(m)}} &= \sum_{n=s_1}^{s_2} h_n \alpha_{m+1, n+2n_{\varphi(m)}} \\
\beta_{mn_{\psi(m)}} &= \sum_{n=-s_2+1}^{-s_1+1} g_n \alpha_{m+1, n+2n_{\psi(m)}}
\end{aligned} \tag{9.38}$$

Analizując wzory algorytmu można zauważyć, że znając współczynniki  $\alpha_{m+1, n}$  można w prosty sposób wyznaczyć współczynniki  $\alpha_{mn_{\varphi(m)}}$  i  $\beta_{mn_{\varphi(m)}}$ . Analogicznie, znając  $\alpha_{mn_{\varphi(m)}}$  otrzymuje się, w sposób kaskadowy, kolejne współczynniki  $\alpha_{m-1, n_{\varphi(m-1)}}$  i  $\beta_{m-1, n_{\varphi(m-1)}}$ .

Postępując w analogiczny sposób z kolejnymi współczynnikami niższego poziomu, w sposób iteracyjny wylicza się współczynniki szukanego poziomu aproksymacji  $k$  tj.  $\alpha_{k, n_{\varphi(k)}}$  i  $\beta_{k, n_{\varphi(k)}}$ .

Reprezentacja graficzna powyższej zależności przedstawia się następująco:

$$\begin{array}{ccccccc}
\alpha_{m+1, n} & \implies & \alpha_{m, n_{\varphi(m)}} & \implies & \alpha_{m-1, n_{\varphi(m-1)}} & \implies & \dots \implies \alpha_{k+1, n_{\varphi(k+1)}} \implies \alpha_{k, n_{\varphi(k)}} \\
\downarrow & & \downarrow & & \downarrow & & \downarrow \\
\beta_{m, n_{\psi(m)}} & & \beta_{m-1, n_{\psi(m-1)}} & & \beta_{m-2, n_{\psi(m-2)}} & & \beta_{k, n_{\psi(k)}}
\end{array} \tag{9.39}$$

## 9.6 Obliczanie przybliżonej wartości początkowej współczynników falkowych

Aby otrzymać wartość początkową współczynników falkowych  $\alpha_{Mn}$  w sposób dokładny wyznacza się ją według wzoru [3]

$$\alpha_{Mn} = \int_{\frac{s_1+n}{2^M}}^{\frac{s_2+n}{2^M}} F(x)\varphi_{Mn}(x)dx \quad (9.40)$$

Ze względu na minimalizację błędu obliczeniowego dla małej skali  $M$ , algorytm Mallata wymaga korzystania z powyższego wzoru, który w konsekwencji wiąże się to z dużym nakładem obliczeniowym.

W przypadku wysokiej skali  $M$  obliczenia można w znacznym stopniu uprościć, ponieważ nośnik falek  $Supp \varphi_{Mn}(x) = \langle \frac{s_1+n}{2^M}, \frac{s_2+n}{2^M} \rangle$  jest silnie skoncentrowany wokół punktu

$$x_{\varphi, Mn} = \frac{\frac{s_1+s_2}{2} + n}{2^M} = \frac{s_1 + s_2}{2^{M+1}} + \frac{n}{2^M}, \quad (9.41)$$

Przyjmując, że funkcja  $F(x)$  dla dużego  $M$  na przedziale  $\langle \frac{s_1+n}{2^M}, \frac{s_2+n}{2^M} \rangle$  ma wartość stałą,  $F(x)$  wynosi w przybliżeniu  $F(\frac{n}{2^M})$ . Wynika to z wartości granicznej  $\lim_{M \rightarrow \infty} \frac{s_1+s_2}{2^{M+1}} = 0$ .

Tak więc, podstawiając do wzoru

$$\begin{aligned} \alpha_{Mn} &= \int_{\frac{s_1+n}{2^M}}^{\frac{s_2+n}{2^M}} F(x)\varphi_{Mn}(x)dx \\ &\approx F\left(\frac{n}{2^M}\right) \int_{\frac{s_1+n}{2^M}}^{\frac{s_2+n}{2^M}} \varphi_{Mn}(x)dx \\ &= F\left(\frac{n}{2^M}\right) \int_{\frac{s_1+n}{2^M}}^{\frac{s_2+n}{2^M}} 2^{\frac{M}{2}} \varphi(2^M x - n)dx, \end{aligned} \quad (9.42)$$

stosując metodę podstawiania  $2^M x - n = t \Rightarrow dx = 2^{-M} dt$  otrzymuje się

$$\alpha_{Mn} = 2^{\frac{M}{2}} 2^{-M} F\left(\frac{n}{2^M}\right) \underbrace{\int_{s_1}^{s_2} \varphi(t)dt}_1 = 2^{-\frac{M}{2}} F\left(\frac{n}{2^M}\right) \quad (9.43)$$

Zatem, przybliżona wartość współczynników falkowych wyraża się wzorem [75]

$$\alpha_{Mn} \approx 2^{-\frac{M}{2}} F\left(\frac{n}{2^M}\right). \quad (9.44)$$

Reasumując, dzięki wysokiemu poziomowi rozdzielczości  $M$  można wyznaczyć współczynniki początkowe bez wykorzystywania operacji splotu i całkowania, co też eliminuje znacznie koszt obliczeniowy dla początkowych wartości współczynników falkowych.

## 9.7 Dwuwymiarowa analiza wielorozdzielcza

Reprezentacja falkowa w prosty sposób może być uogólniona na wymiar stopnia drugiego, dzięki czemu ma ona swoje przełożenie podczas analizy obrazu.

Jeżeli badana funkcja posiada skończoną energię  $F(x, y) \in L^2(\mathbb{R}^2)$ , gdzie wielorozdzielcza aproksymacja  $L^2(\mathbb{R}^2)$  jest ciągiem podprzestrzeni analizy wielorozdzielczej jednowymiarowej uogólnionej na przypadek dwuwymiarowy, wtedy to przestrzenie  $\mathbf{V}_j$  definiuje się następująco [35, 70]:

$$\begin{aligned} \mathbf{V}_0 &= V_0 \otimes V_0 \\ F(x, y) \in \mathbf{V}_0 &\iff F(2^j x, 2^j y) \in \mathbf{V}_j \end{aligned} \tag{9.45}$$

Przestrzenie  $\mathbf{V}_j$  tworzą w  $L^2(\mathbb{R}^2)$  ciąg spełniający warunki:

$$\mathbf{V}_M \subset \mathbf{V}_{M+1} \subset \dots \subset \mathbf{V}_m \subset \dots \subset L^2(\mathbb{R}^2) \tag{9.46}$$

gdzie

$$\lim_{m \rightarrow -\infty} \mathbf{V}_m = \{0\} \tag{9.47}$$

$$\overline{\lim_{m \rightarrow \infty} \mathbf{V}_m} = L^2(\mathbb{R}^2) \tag{9.48}$$

Ponieważ funkcje  $\varphi(\bullet - n)$ ,  $n \in \mathbb{Z}$  tworzą bazę ortonormalną w  $V_0$ , funkcje utworzone przez ich iloczyn stanowią bazę ortonormalną w  $\mathbf{V}_0$  gdzie

$$\Phi_{0,mn}(x, y) = \varphi(x - n)\varphi(x - m), \quad n, m \in \mathbb{Z}$$

Ponadto, dla przestrzeni  $\mathbf{V}_j$  bazę ortonormalną stanowią funkcje

$$\begin{aligned} \Phi_{j,mn}(x, y) &= \varphi_{jn}(x)\varphi_{jm}(x) \\ &= 2^j \Phi(2^j x - n, 2^j y - m) \quad n, m \in \mathbb{Z} \end{aligned} \tag{9.49}$$

Analogicznie, jak dla przypadku jednowymiarowego dla każdego  $j \in \mathbb{Z}$  istnieje przestrzeń komplementarna  $\mathbf{W}_j$  która jest dopełnieniem  $\mathbf{V}_j$  w  $\mathbf{V}_{j+1}$  [52, 70] tj.:

$$\begin{aligned}
\mathbf{V}_{j+1} &= V_{j+1} \otimes V_{j+1} = (V_j \oplus W_j) \otimes (V_j \oplus W_j) \\
&= V_j \otimes V_j \oplus [(W_j \otimes V_j) \oplus (V_j \otimes W_j) \oplus (W_j \otimes W_j)] \\
&= \mathbf{V}_j \oplus \mathbf{W}_j
\end{aligned} \tag{9.50}$$

Analizując powyższe wyrażenie można zauważyć, że przestrzeń szczegółów składa się z trzech członów z bazami  $\psi_{jn}\varphi_{jm}$ ,  $\varphi_{jn}\psi_{jm}$  i  $\psi_{jn}\psi_{jm}$  [52].

Reasumując, przestrzeń detali składa się z trzech falek

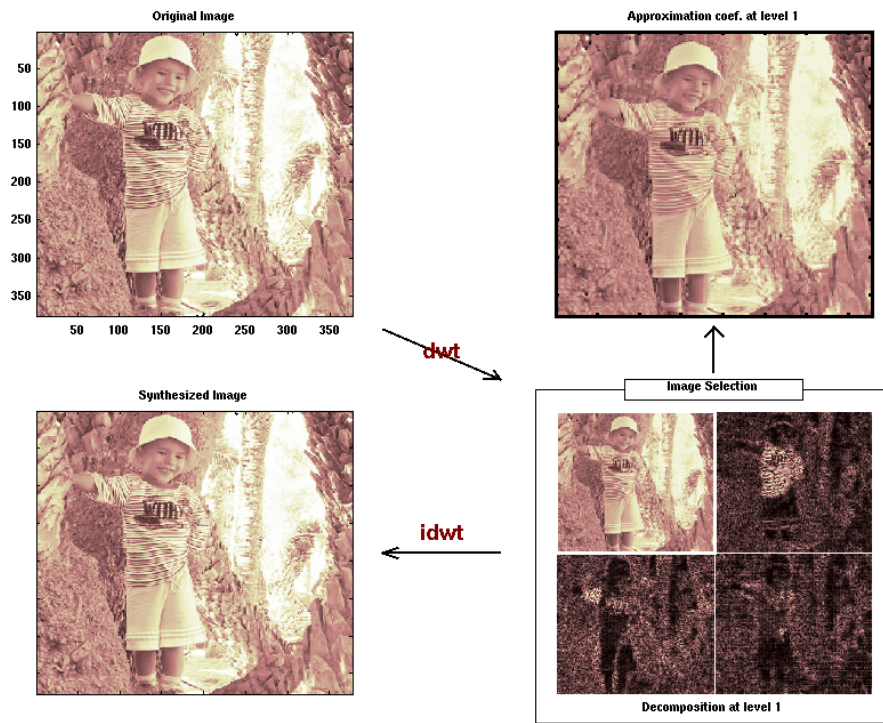
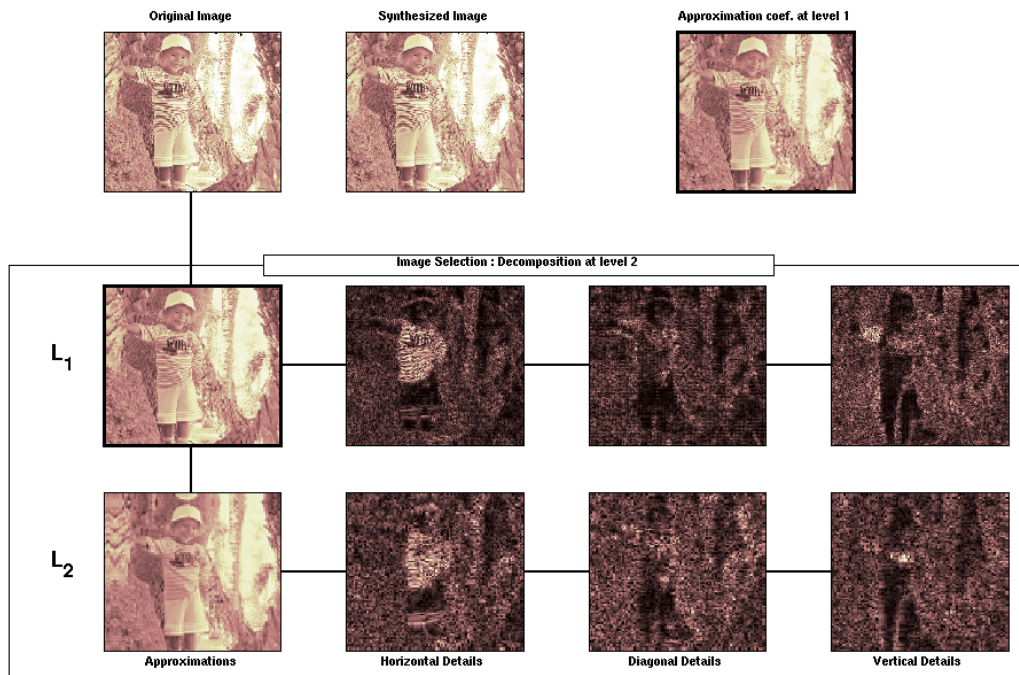
$$\begin{aligned}
\Psi^h(x, y) &= \varphi(x)\psi(y) \\
\Psi^v(x, y) &= \psi(x)\varphi(y) \\
\Psi^d(x, y) &= \psi(x)\psi(y)
\end{aligned} \tag{9.51}$$

gdzie falekę  $\Psi^h(x, y)$  nazywa się faleką poziomą,  $\Psi^v(x, y)$  - faleką pionową, a  $\Psi^d(x, y)$  - faleką diagonalną.

Aproksymacja obrazu (dwuwymiarowa reprezentacja) na poziomie  $\mathbf{V}_{j+1}$  składa się z aproksymacji zgrubnej  $\mathbf{V}_j$  i przestrzeni detali  $\mathbf{W}_j$  reprezentujący trzy składowe: poziomą, pionową i diagonalną. Dla składowej poziomej detalu uwypuklone są krawędzie poziome obrazu, dla pionowej - krawędzie pionowe, zaś dla diagonalnej wyróżnione są diagonalne. Charakterystyka tych składowych uzasadnia przyjęte nazwy.

Dla obrazu oryginalnego o liczbie elementów  $N \times N$  każda ze składowych dla procesu dekompozycji i części aproksymacji zgrubnej posiada  $\frac{N}{2} \times \frac{N}{2}$  elementów. Wynika z tego, że aproksymacja zgrubna reprezentuje obraz o rozmiarze jednej czwartej obrazu oryginalnego (rys. 9.4).

Dekompozycja obrazu na poziomie wyższym może być przedstawiona także w postaci struktury drzewa (rys.9.5), gdzie pierwszym obrazem (od lewej) jest tzw. aproksymacja zgrubna, natomiast następne to przestrzenie szczegółów (pozioma, diagonalna i pionowa).

Rysunek 9.4: Schemat falkowej dekompozycji obrazu za pomocą falki **db2**

Rysunek 9.5: Wizualizacja falkowej dekompozycji obrazu w postaci struktury drzewa

## Rozdział 10

---

# Dyskretna transformata falkowa (DWT)

---

Wykorzystanie analitycznych wzorów ciągłej transformaty falkowej w praktyce wiąże się zazwyczaj z dużą złożonością obliczeniową, co też może być przyczyną, że obliczenia stają się niemożliwe nawet dla stosunkowo prostych funkcji.

Alternatywą dla tego typu rozwiązania może być tzw. dyskretyzacja transformaty falkowej, a co się z tym wiąże dyskretyzacja funkcji badanej, poddając ją zabiegowi próbkowania w dziedzinie czasu.

Odpowiednio gęste próbkowanie, umożliwia rozkład funkcji badanej na jej reprezentację zgrubną, jak i szczegółową dając w konsekwencji wynik który jest mocno zbliżony, praktycznie tożsamościowo równy względem analizy w przestrzeni ciągłej, uzyskany znacznie mniejszym kosztem obliczeniowym.

Pomysł dyskretnej transformaty falkowej oparty jest na algorytmie Mallata opisanym w rozdziale 9.5 za pomocą wzoru 9.38.

Za pomocą transformaty funkcja przepuszczana jest przez filtr połówkowy dolnoprzepustowy  $h$  z funkcją przenoszenia  $H(f) = 1$  dla  $f < \frac{f_c}{2}$ , i  $H(f) = 0$  dla  $f > \frac{f_c}{2}$ , gdzie  $f_c$  jest częstotliwością Nyquista\* ( $f_c = \frac{1}{2}\Delta t = \frac{1}{2}$ ) [33].

Działanie takiego filtru jest dane przez konwolucję [54]:

$$y(i) = \sum_k h^*(2i - k)y(k) \quad (10.1)$$

gdzie filtr usuwa wszystkie próbki należące do częstotliwości powyżej połowy częstotliwości Nyquista.

Jeżeli na funkcji zostanie przeprowadzona operacja splotu za pomocą połówkowego filtru górno-przepustowego  $g$ , wtedy to otrzymuje się funkcję która przenosi informację o szczegółach funkcji aproksymowanej na poziom o jeden stopień wyżej.

Operacja taka wyraża się wzorem:

---

\*Częstotliwość Nyquista jest to maksymalna częstotliwość składowych widmowych funkcji poddawanej procesowi próbkowania, które mogą zostać odtworzone z ciągu próbek bez zniekształceń. Składowe widmowe o częstotliwościach wyższych od częstotliwości Nyquista ulegają podczas próbkowania nałożeniu na składowe o innych częstotliwościach (zjawisko aliasingu), co powoduje, że nie można ich już poprawnie odtworzyć[33].

$$x(i) = \sum_k g^*(2i - k)y(k) \quad (10.2)$$

W konsekwencji, analizę funkcji można przedstawić za pomocą macierzy konwolucji, jako [5]

$$\begin{bmatrix} y_n \\ x_n \end{bmatrix} = \begin{bmatrix} \mathbf{H}^* \\ \mathbf{G}^* \end{bmatrix} y_{n+1} \quad (10.3)$$

lub równoważnie

$$y_{n+1} = \begin{bmatrix} \mathbf{H} & \mathbf{G} \end{bmatrix} \begin{bmatrix} y_n \\ x_n \end{bmatrix} \quad (10.4)$$

gdzie [53, 54]

$$\mathbf{H}^* = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & & & & & & \\ & & h_0 & h_1 & h_2 & h_3 & & & & \\ & & & & h_0 & h_1 & h_2 & h_3 & & \\ h_2 & h_3 & & & & & h_0 & h_1 & & \end{bmatrix} \quad (10.5)$$

i

$$\mathbf{G}^* = \begin{bmatrix} h_3 & -h_2 & h_1 & -h_0 & & & & & & \\ & & h_3 & -h_2 & h_1 & -h_0 & & & & \\ & & & & h_3 & -h_2 & h_1 & -h_0 & & \\ h_1 & -h_0 & & & & & h_3 & -h_2 & & \end{bmatrix} \quad (10.6)$$

zaś macierze filtrów spełniają własności  $\mathbf{H}\mathbf{H}^* = \mathbf{I}$ ,  $\mathbf{G}\mathbf{G}^* = \mathbf{I}$  i  $\mathbf{H}\mathbf{G}^* = \mathbf{G}\mathbf{H}^* = \mathbf{0}$ .

Aby określić rzut ortogonalny dowolnej funkcji  $f(t)$  na podprzestrzeń ortogonalną  $V_0$ , zgodnie ze wzorem 9.16, wymaga to obliczenia wektora iloczynów skalarnych  $y(n) = \langle \varphi(t - n), f(t) \rangle$ ,  $n \in \mathbb{Z}$ .

W sytuacji jednak, gdy rozdzielczość przestrzeni jest wystarczająco duża w stosunku do funkcji wejściowej  $f(t)$ , funkcja ta może ulec próbkowaniu. Wynika to stąd, że falka  $\varphi(t)$  jest dolnoprzepustową odpowiedzią impulsową o jednostkowej całce [6].

Dodatkowo, gdy funkcja  $f(t)$  jest gładka, zaś funkcja  $\varphi(t)$  jest dobrze tłumiona, korzystając z powyższych rozważań można zatem przyjąć przybliżenie [3]

$$y(n) \approx f(n) \quad (10.7)$$

które pozwala na zmniejszenie w znacznym stopniu złożoności obliczeniowej.

Przykładem takiej aproksymacji może być analiza wektora  $\mathbf{y} = [8 \ 5 \ 7 \ 2]$  za pomocą falki Haara ( $h_0 = h_1 = \frac{1}{\sqrt{2}}$ ).

Jeżeli funkcja skalująca przyjmuje postać  $\varphi_{0l}(k) = \delta_{k-l}$  wtedy to wektor  $\mathbf{y}$  można zapisać jako

$$\mathbf{y} = 8 \begin{bmatrix} \varphi_{00}(0) \\ \varphi_{00}(1) \\ \varphi_{00}(2) \\ \varphi_{00}(3) \end{bmatrix} + 5 \begin{bmatrix} \varphi_{01}(0) \\ \varphi_{01}(1) \\ \varphi_{01}(2) \\ \varphi_{01}(3) \end{bmatrix} + 7 \begin{bmatrix} \varphi_{02}(0) \\ \varphi_{02}(1) \\ \varphi_{02}(2) \\ \varphi_{02}(3) \end{bmatrix} + 2 \begin{bmatrix} \varphi_{03}(0) \\ \varphi_{03}(1) \\ \varphi_{03}(2) \\ \varphi_{03}(3) \end{bmatrix}. \quad (10.8)$$

Następnym etapem jest zdefiniowanie macierzy konwolucji, gdzie

$$\mathbf{H}_2^* = \begin{bmatrix} h_0 & h_1 & & \\ & & h_0 & h_1 \end{bmatrix} \quad (10.9)$$

$$\mathbf{G}_2^* = \begin{bmatrix} h_1 & -h_0 & & \\ & & h_1 & -h_0 \end{bmatrix} \quad (10.10)$$

wtedy to równanie analizy wielorozdzielczej na poziomie 2 przyjmuje postać

$$\begin{bmatrix} y_{01} \\ y_{11} \\ x_{01} \\ x_{11} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & & \\ & & 1 & 1 \\ 1 & -1 & & \\ & & & 1 & -1 \end{bmatrix} \begin{bmatrix} 8 \\ 5 \\ 7 \\ 2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 13 \\ 9 \\ 3 \\ 5 \end{bmatrix}.$$

Ostatnie dwa współczynniki należą do przestrzeni detali, stąd też są wykorzystywane przy syntezie funkcji aproksymowanej.

Ostatni krok aproksymacji to podział funkcji na współczynniki  $y_{02}$  i  $x_{02}$ .

$$\begin{bmatrix} y_{02} \\ x_{02} \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 13 \\ 9 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 22 \\ 4 \end{bmatrix} = \begin{bmatrix} 11 \\ 2 \end{bmatrix}$$

tak więc

$$\begin{bmatrix} y_{02} \\ x_{02} \\ x_{01} \\ x_{11} \end{bmatrix} = \begin{bmatrix} 11 \\ 2 \\ 3 \\ 5 \end{bmatrix}.$$

W celu rekonstrukcji funkcji można odwrócić proces, wykorzystując do tego celu własności analizy wielorozdzielczej.

Aproksymowana funkcja może być wyrażona zatem jako kombinacja liniowa funkcji, gdzie [5]



$$\mathbf{y} = \frac{13}{\sqrt{2}}\varphi_{-1,0} + \frac{9}{\sqrt{2}}\varphi_{-1,1} + \frac{3}{\sqrt{2}}\psi_{-1,0} + \frac{5}{\sqrt{2}}\psi_{-1,1} \quad (10.11)$$

lub

$$\mathbf{y} = 11\varphi_{-2,0} + 2\psi_{-2,0} + \frac{3}{\sqrt{2}}\psi_{-1,0} + \frac{5}{\sqrt{2}}\psi_{-1,1} \quad (10.12)$$

podczas gdy funkcja skalująca Haara

$$\varphi_{mn} = \begin{cases} 2^{\frac{m}{2}} & \text{dla } x \in \langle n2^{-m}, (n+1)2^{-m} \rangle \\ 0 & \text{dla } x \notin \langle n2^{-m}, (n+1)2^{-m} \rangle \end{cases}$$

i funkcja falkowa Haara

$$\psi_{mn} = \begin{cases} 2^{\frac{m}{2}} & \text{dla } x \in \langle n2^{-m}, (n+\frac{1}{2})2^{-m} \rangle \\ -2^{\frac{m}{2}} & \text{dla } x \in \langle (n+\frac{1}{2})2^{-m}, (n+1)2^{-m} \rangle \\ 0 & \text{dla } x \notin \langle n2^{-m}, (n+1)2^{-m} \rangle \end{cases}$$

Analizując powyższy zapis, macierz konwolucji prezentuje się w postaci [5, 53]

$$\begin{bmatrix} y_{02} \\ x_{02} \\ x_{01} \\ x_{11} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & & \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & & \\ & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & & \\ & & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} y_{00} \\ y_{10} \\ y_{20} \\ y_{30} \end{bmatrix} \quad (10.13)$$

lub równoważnie

$$\begin{aligned} \begin{bmatrix} y_{02} \\ x_{02} \\ x_{01} \\ x_{11} \end{bmatrix} &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & & \\ 1 & -1 & & \\ & & \sqrt{2} & \\ & & & \sqrt{2} \end{bmatrix} \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & & \\ & 1 & 1 & \\ 1 & -1 & & \\ & & 1 & -1 \end{bmatrix} \begin{bmatrix} y_{00} \\ y_{10} \\ y_{20} \\ y_{30} \end{bmatrix} = \\ &= \frac{1}{2} \begin{bmatrix} 1 & 1 & & \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & & \\ & & \sqrt{2} & -\sqrt{2} \end{bmatrix} \begin{bmatrix} y_{00} \\ y_{10} \\ y_{20} \\ y_{30} \end{bmatrix} \end{aligned} \quad (10.14)$$

Wynika z tego, że podczas jednego mnożenia macierzowego z poziomu 0 można uzyskać analizę funkcji na poziomie 2.

Stosunkowo prosta analiza funkcji pozwala na szybkie przekształcenie jej na podprzestrzeń aproksymacji i reprezentacji szczegółowej. Operacja za pomocą macierzy konwolucji narzuca jednak pewny narzut niepotrzebnych operacji mnożenia przez zero.

W praktyce dla przestrzeni dyskretnej jako reprezentację zgrubną przyjmuje się wartości współczynników przy funkcjach skalujących, zaś przestrzeń detali jest wynikiem wartości współczynników przy funkcjach falkowych aproksymatora  $f(x; m)$  zdefiniowanej we wzorze 9.17.

Wynika to stąd, że im współczynniki falkowe są bliższe zeru tym lepsze jest dopasowanie funkcji falkowych do badanej funkcji.

Operacja dyskretnej transformaty falkowej dla przestrzeni ortogonalnej ze względu na swoją ortogonalność nie pozwala często na elastyczne dopasowanie się falki do analizowanych próbek. Takie złe dopasowanie falki powoduje, że reprezentacja szczegółowa posiada współczynniki bardzo zaszumione, zaś reprezentacja zgrubna ma tendencję do odchylenia względem charakterystyki funkcji.

Jednym ze sposobów rozwiązania problemu, było zaproponowanie przez Wima Sweldensa nowej metody konstruowania falek, zwanej schematem liftingu [58]. Schemat ten polega na nowej koncepcji tworzenia **falek biortogonalnych** pozwalających na dopasowywanie się tzw. banku filtrów do badanej funkcji.

Podczas konstrukcji falek biortogonalnych, za pomocą schematu liftingu, nie korzysta się z algebraicznych operacji translacji i przesunięć jednej funkcji podstawowej. Falki te mogą korzystać z wielu funkcji, ale musi zostać zachowana zależność, żeby bazowa funkcja jednego poziomu stanowiła skończoną, liniową kombinację funkcji wyższego poziomu rozdzielczości. Gwarantuje to możliwość zdefiniowania zagnieżdżonych przestrzeni koniecznych do opisanego analizy wielorozdzielczej [61].



## Część IV

# Analiza wielorozdzielcza z wykorzystaniem przestrzeni biortogonalnych



## Rozdział 11

---

# Własności biortogonalnych baz falkowych

---

Rekonstrukcja funkcji na podstawie reprezentacji falkowej powinna wykorzystywać falki o stosunkowo gładkiej charakterystyce. Ze względu na dużą złożoność obliczeniową falka jest efektywna wtedy, gdy posiada jak najkrótszy nośnik. Jednakże, im krótszy nośnik, tym bardziej zanika gładkość funkcji. Zatem, podczas optymalizacji funkcji, należy uwzględniać dwie powyższe cechy falki.

Dosyć istotną cechą dla analizy funkcji jest liniowa faza filtrów o skończonej odpowiedzi impulsowej (SOI), które pozwalają na niewykorzystywanie kompensacji fazy na poszczególnych stopniach dekompozycji.

Falki Daubechies posiadające, dla określonej długości nośnika, maksymalną liczbę znikających momentów są optymalne w sensie kryterium ortogonalności i dokładności aproksymacji. Nie posiadają one jednak banku filtrów o liniowej fazie [71].

W celu uzyskania liniowości fazy banku filtrów, a co za tym idzie - symetryczności względem punktu  $x_0$ , do analizy wykorzystuje się funkcje bazowe liniowo niezależne, z wyłączeniem warunku ortogonalności.

Bazy te zamiast ortogonalności zachowują cechę biortogonalności, pozwalającą wykorzystywać własności analizy wielorozdzielczej.

Oprócz rodziny funkcji skalujących  $\{\varphi(x - n)\}_{n \in \mathbb{Z}}$ , rozpiętej na przestrzeni  $V_0$ , do celów dekompozycji i syntezy funkcji badanej wykorzystuje się dualną bazę Riesz podprzestrzeni  $\tilde{V}_0$  rodziny funkcji  $\{\tilde{\varphi}(x - n)\}_{n \in \mathbb{Z}}$ .

Ponadto, jeżeli bazy  $\{\varphi_{mn}\}_{n \in \mathbb{Z}}$  i  $\{\tilde{\varphi}_{mn}\}_{n \in \mathbb{Z}}$  są bazami Riesz  $V_m$  i  $\tilde{V}_m$ , wtedy to falki  $\{\psi_{mn}\}_{n \in \mathbb{Z}}$  i  $\{\tilde{\psi}_{mn}\}_{n \in \mathbb{Z}}$  są bazami dualnych podprzestrzeni detali  $W_m$  i  $\tilde{W}_m$ , gdzie  $V_{m+1} = V_m \oplus W_m$  i  $\tilde{V}_{m+1} = \tilde{V}_m \oplus \tilde{W}_m$ .

Biortogonalność charakteryzuje się tym, że przestrzeń detali  $W_m$  nie jest ortogonalna do  $V_m$  lecz do jej odpowiednika w bazie dualnej  $\tilde{V}_m$ .

Analogicznie przestrzeń detali  $\tilde{W}_m$  jest ortogonalna do rzutu na podprzestrzeń  $V_m$  [49].

Równaniami funkcji skalującej  $\varphi(x)$ , funkcji falkowej  $\psi(x)$  i ich dualnych odpowiedników  $\tilde{\varphi}(x)$  i  $\tilde{\psi}(x)$  są [12, 31, 66]:

$$\varphi(x) = \sqrt{2} \sum_k h_k \varphi(2x - k) \quad (11.1)$$

$$\psi(x) = \sqrt{2} \sum_k g_k \psi(2x - k) \quad (11.2)$$

$$\tilde{\varphi}(x) = \sqrt{2} \sum_k \tilde{h}_k \tilde{\varphi}(2x - k) \quad (11.3)$$

$$\tilde{\psi}(x) = \sqrt{2} \sum_k \tilde{g}_k \tilde{\psi}(2x - k) \quad (11.4)$$

gdzie współczynniki  $h_k, g_k, \tilde{h}_k$  i  $\tilde{g}_k$  reprezentują kolejne, poszczególne składowe współczynniki filtrów  $h, g, \tilde{h}$  i  $\tilde{g}$ .

Ponadto, filtry dolnoprzepustowe w dziedzinie czasu spełniają zależność[45]

$$\sum_{k \in \mathbb{Z}} h_k \tilde{h}_{k-2n} = \delta_n \quad (11.5)$$

zaś cecha biortogonalności uogólnia względem przestrzeni ortogonalnych, zasadę filtrów lustrzanych za pomocą wzorów

$$\tilde{g}_n = (-1)^n h_{1-n} \quad (11.6)$$

$$g_n = (-1)^n \tilde{h}_{1-n}. \quad (11.7)$$

Funkcje bazowe dla baz biortogonalnych definiowane są analogicznie jak dla przypadku baz ortogonalnych:

$$\varphi_{jk}(x) = 2^{\frac{j}{2}} \varphi(2^j x - k) \quad (11.8)$$

$$\psi_{jk}(x) = 2^{\frac{j}{2}} \psi(2^j x - k)$$

Wynika z tego, że aproksymacja funkcji na poziomie rozdzielczości  $m$  dla modelu zagregowanego wyraża się wzorem

$$f(x; m) = \sum_{j \in \mathbb{Z}} \alpha_{mj} \varphi_{mj}(x) \quad (11.9)$$

natomiast model zdekomponowany posiada formę

$$f(x; m) = \sum_{j \in \mathbb{Z}} \alpha_{kj} \varphi_{kj}(x) + \sum_{i=k}^{m-1} \sum_{j \in \mathbb{Z}} \beta_{ij} \psi_{ij}(x) \quad (11.10)$$

Współczynniki falkowe wylicza się za pomocą iloczynu skalarnego funkcji aproksymowanej z funkcjami skalującymi i falkowymi przestrzeni dualnej [74] tj.

$$\alpha_{j,l} = \langle f, \tilde{\varphi}_{j,l} \rangle = \int_{-\infty}^{\infty} f(x) \tilde{\varphi}_{j,l}(x) dx \quad (11.11)$$

$$\beta_{j,l} = \langle f, \tilde{\psi}_{j,l} \rangle = \int_{-\infty}^{\infty} f(x) \tilde{\psi}_{j,l}(x) dx \quad (11.12)$$

Schemat Mallata opisany w rozdziale 9.5 dla przestrzeni ortogonalnych można w analogiczny sposób przenieść na przestrzeń biortogonalną. Algorytm ten pozwala na iteracyjne obliczenie współczynników poziomu niższego znając współczynniki falkowe poziomu wyższego i odpowiadające im współczynniki filtrów dualnych [65]:

$$\alpha_{j,l} = \sqrt{2} \sum_k \tilde{h}_{k-2l} \alpha_{j+1,k} \quad (11.13)$$

$$\beta_{j,k} = \sqrt{2} \sum_k \tilde{g}_{k-2l} \alpha_{j+1,k}. \quad (11.14)$$

Aby uzyskać stabilne i zupełne biortogonalne bazy falkowe o nośniku zwartym dla filtrów dolnoprzepustowych  $h$  i  $\tilde{h}$ , filtry te muszą spełniać **warunek doskonałej rekonstrukcji** w dziedzinie częstotliwości [57]

$$\tilde{H}^*(\omega)H(\omega) + \tilde{H}^*(\omega + \pi)H(\omega + \pi) = 2 \quad (11.15)$$

lub równoważnie za pomocą transformaty  $Z$  dla przestrzeni dyskretnej

$$\begin{aligned} h(z)\tilde{h}(z^{-1}) + g(z)\tilde{g}(z^{-1}) &= 2 \\ h(z)\tilde{h}(-z^{-1}) + g(z)\tilde{g}(-z^{-1}) &= 0 \end{aligned} \quad (11.16)$$

Jeżeli filtry  $\tilde{h}_n$  o współczynnikach niezerowych dla  $\tilde{N}_1 \leq n \leq \tilde{N}_2$  i  $h_n$  w przypadku niezerowych współczynników dla  $N_1 \leq n \leq N_2$  spełniają warunek doskonałej rekonstrukcji (11.15), wtedy do odpowiadające im funkcje skalujące posiadają skończony nośnik

$$\text{Supp } \tilde{\varphi}(x) = \langle \tilde{N}_1, \tilde{N}_2 \rangle \quad (11.17)$$

$$\text{Supp } \varphi(x) = \langle N_1, N_2 \rangle \quad (11.18)$$



zaś nośniki falki syntezy i analizy wyrażają się w sposób następujący [45]

$$\text{Supp } \tilde{\psi}(x) = \left\langle \frac{\tilde{N}_1 - N_2 + 1}{2}, \frac{\tilde{N}_2 - N_1 + 1}{2} \right\rangle \quad (11.19)$$

$$\text{Supp } \psi(x) = \left\langle \frac{N_1 - \tilde{N}_2 + 1}{2}, \frac{N_2 - \tilde{N}_1 + 1}{2} \right\rangle \quad (11.20)$$

Konstrukcja biortogonalnych falek o nośniku zwartym pozwala na otrzymanie charakterystyk falek o charakterze symetrycznym lub antysymetrycznym.

Skojarzone filtry które spełniają ponadto warunek doskonałej rekonstrukcji (11.15) uzyskują liniową fazę, dzięki którym nie trzeba wykorzystywać kompensacji fazy na poszczególnych stopniach dekompozycji.

Jeżeli  $\tilde{\psi}$  i  $\psi$  dla falek biortogonalnych posiadają  $\tilde{p}$  i  $p$  momentów znikających, rozmiar ich nośnika wynosi co najmniej  $\tilde{p} + p + 1$ .

Takie falki które posiadają minimalny nośnik noszą nazwę CDF od nazwisk ich twórców: Cohena, Daubechies i Feauveau[6].

## Rozdział 12

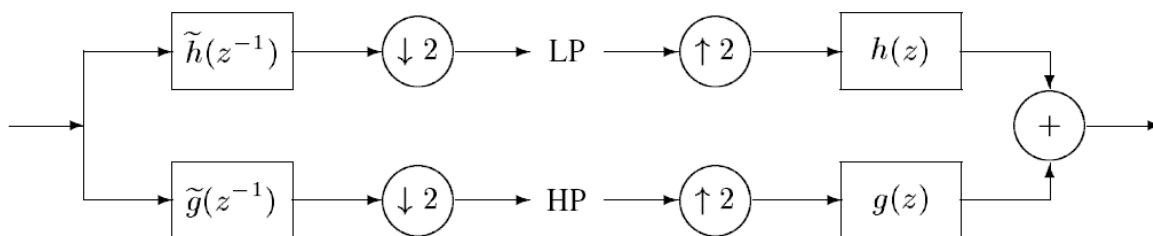
# Reprezentacja polifazowa

Wielorozdzielcza analiza falkowa dla dyskretnej przestrzeni biortogonalnej, analogicznie jak w przypadku przestrzeni ortogonalnych, do filtracji funkcji używa schematu dwukanałowej dekompozycji z wykorzystaniem filtrów górno- i dolnoprzepustowych zgodnie z rys. 12.1.

Istotną zaletą tego typu rozwiązania jest tzw. jawność współczynników filtru analizy  $\tilde{h}$ ,  $\tilde{g}$  i syntezy  $h$ ,  $g$ , dzięki którym można otrzymać bezpośrednio wyrażenia funkcji falkowych i skalujących zgodnie z równaniami definiującymi opisanymi w rozdziale 3 tj.

$$\begin{aligned}\varphi(x) &= \sqrt{2} \sum_{n \in \mathbb{Z}} h_n \varphi(2x - n) \\ \psi(x) &= \sqrt{2} \sum_{n \in \mathbb{Z}} g_n \varphi(2x - n)\end{aligned}\tag{12.1}$$

Podczas rekonstrukcji i dekompozycji układu nie wykorzystuje się żadnych korelacji pomiędzy reprezentacją zgrubną a reprezentacją detali, natomiast filtracja realizowana jest za pomocą splotu, w dziedzinie czasu, lub za pomocą iloczynu w dziedzinie częstotliwościowej za pomocą dyskretnej transformacji Fouriera.



Rysunek 12.1: Schemat dwukanałowej dekompozycji i rekonstrukcji

Analizując dyskretną transformatę falkową można zauważyć, że podczas przetwarzania funkcji, w pierwszej kolejności, podlega ona filtracji dolno i górnoprzepustowej a następnie decymacji  $\downarrow 2$  (w przypadku analizy) lub procesowi uzupełniania - ekspandera  $\uparrow 2$  (podczas syntezy). Decymacja jest stosowana ze względu na eliminację nakładania się widm pasm powstałych w wyniku filtracji (aliasing), który powoduje wzmocnienie niektórych obszarów funkcji [35].

Ponieważ ekspander i decymacja powodują nadmiar obliczeniowy, stąd też aby zmniejszyć złożoność obliczeniową układu Wim Sweldens zauważył [60], że podczas analizy funkcji wystarczy ograniczyć się do własności próbek parzystych otrzymanych po transformacji.

Analiza tego typu podejścia, prowadzi w konsekwencji do pojęcia macierzy polifazowej, którą otrzymuje się za pomocą elementarnych przekształceń biortogonalnych filtrów górno i dolnoprzepustowych z wykorzystaniem transformacji  $Z^*$ .

Aby określić powyższą macierz, definiuje się następujące własności.

Jeżeli filtr  $h$  jest typu SOI (skończoną odpowiedzią impulsową), gdzie nośnikiem funkcji skalującej jest przedział  $\langle k_1, k_2 \rangle$  dla  $k_i \in \mathbb{Z}$ , wtedy to przy pomocy wielomianu Laurenta można określić filtr dolnoprzepustowy w postaci [67]

$$\tilde{h}(z) = \sum_{k=k_1}^{k_2} \tilde{h}_k z^{-k}. \quad (12.2)$$

Stopień wielomianu Laurenta  $h$  definiuje się wtedy jako  $|\tilde{h}| = k_2 - k_1$ , zaś długość filtru (nośnika) jako  $|\tilde{h}| + 1$ .

Dla zmniejszenia złożoności obliczeniowej wykorzystuje się zbiór współczynników falkowych  $\tilde{h} = \{\tilde{h}_k\}$  ograniczony do zbioru próbek parzystych  $\tilde{h}^s = \{\tilde{h}_{2k}\}$  [15] takich, że [64]

$$\tilde{h}^s(z) = \tilde{h}_e(z) = \sum_k \tilde{h}_{2k} z^{-k}. \quad (12.3)$$

gdzie  $\tilde{h}_e(z^2)$  jest parzystą częścią  $\tilde{h}(z)$  tj.

$$\tilde{h}_e(z^2) = \frac{\tilde{h}(z) + \tilde{h}(-z)}{2} \quad (12.4)$$

Analogicznie, dla współczynników nieparzystych filtr przyjmuje postać

$$\tilde{h}_o(z) = \sum_k \tilde{h}_{2k+1} z^{-k} \quad (12.5)$$

gdzie

$$\tilde{h}_o(z^2) = \frac{z}{2} (\tilde{h}(z) - \tilde{h}(-z)). \quad (12.6)$$

Funkcję pasma dolnoprzepustowego (współczynniki reprezentacji zgrubnej) można zatem wyrazić za pomocą wzoru

$$s(z) = \tilde{h}(z)x(z) = \tilde{h}_e(z)x_e(z) + z^{-1}\tilde{h}_o(z)x_o(z). \quad (12.7)$$

---

\*W dyskretnej transformacji falkowej najczęściej wykorzystuje się odpowiednik transformacji Laplace'a dla przestrzeni dyskretnej, czyli transformację  $Z$ , gdzie filtr  $h(z) = \sum_n h_n z^{-n}$ ,  $z = e^{i\omega}$  i  $n \in \mathbb{N}$  [45].

zaś jego odpowiednik w paśmie górnoprzepustowym (współczynniki reprezentacji szczegółów) jako

$$d(z) = \tilde{g}(z)x(z) = \tilde{g}_e(z)x_e(z) + z^{-1}\tilde{g}_o(z)x_o(z). \quad (12.8)$$

Ponieważ bank filtrów procesu dekompozycji opisuje się postacią macierzową

$$\begin{bmatrix} s(z) \\ d(z) \end{bmatrix} = \begin{bmatrix} \tilde{h}(z) \\ \tilde{g}(z) \end{bmatrix} x(z), \quad (12.9)$$

wykorzystując wyrażenia 12.7 i 12.8 można ją przedstawić za pomocą formy [64]

$$\begin{bmatrix} s(z) \\ d(z) \end{bmatrix} = \begin{bmatrix} \tilde{h}_e(z) & \tilde{h}_o(z) \\ \tilde{g}_e(z) & \tilde{g}_o(z) \end{bmatrix} \begin{bmatrix} x_e(z) \\ z^{-1}x_o(z) \end{bmatrix} \quad (12.10)$$

Powyższą macierz banku filtrów nazywa się polifazową i oznacza się ją jako [15]

$$\tilde{\mathbf{P}}(z) = \begin{bmatrix} \tilde{h}_e(z) & \tilde{h}_o(z) \\ \tilde{g}_e(z) & \tilde{g}_o(z) \end{bmatrix}. \quad (12.11)$$

Macierz ta jest charakterystyką pewnego zestawu filtrów który odpowiada za transformację fal-kową. W przypadku gdy macierz polifazowa jest macierzą jednostkową, transformacja dzieli zbiór próbek jedynie na ich zbiory parzyste i nieparzyste.

Znając **warunek doskonałej rekonstrukcji** (11.16) i definiując tzw. modulacyjną macierz jako

$$\mathbf{M}(z) = \begin{bmatrix} h(z) & h(-z) \\ g(z) & g(-z) \end{bmatrix}. \quad (12.12)$$

można zdefiniować warunek doskonałej rekonstrukcji w formie macierzowej [15]

$$\tilde{\mathbf{M}}(z^{-1})^T \mathbf{M}(z) = 2\mathbf{I} \quad (12.13)$$

gdzie dualna macierz jest macierzą z „tyldą” dla każdego elementu macierzy.

W przypadku procesu syntezy analogiczne uproszczenia (12.10) prowadzą do następującej dualnej macierzy polifazowej:

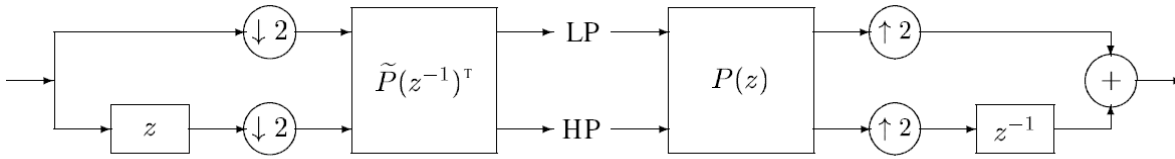
$$\mathbf{P}(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix}. \quad (12.14)$$

Powyższe zależności określają zatem relację pomiędzy macierzą polifazową a macierzą modulatoryjną, gdzie

$$\mathbf{P}(z^2)^T = \frac{1}{2} \mathbf{M}(z) \begin{bmatrix} 1 & z \\ 1 & -z \end{bmatrix} \quad (12.15)$$

Własność doskonałej rekonstrukcji można zatem zapisać za pomocą macierzy polifazowych uwzględniający opóźnienie w wyniku filtracji

$$\mathbf{P}(z) \tilde{\mathbf{P}}(z^{-1})^T = \mathbf{I}. \quad (12.16)$$



Rysunek 12.2: Schemat analizy i syntezy za pomocą reprezentacji polifazowej

Jeżeli macierz polifazowa będzie odwracalna, wtedy [45]

$$\tilde{\mathbf{P}}(z^{-1})^T = \mathbf{P}(z)^{-1} = \frac{1}{h_e(z)g_o(z) - h_o(z)g_e(z)} \begin{bmatrix} g_o(z) & -g_e(z) \\ -h_o(z) & h_e(z) \end{bmatrix}. \quad (12.17)$$

Jeśli dodatkowo wyznacznik macierzy  $\det \mathbf{P}(z) = h_e(z)g_o(z) - h_o(z)g_e(z)$  będzie równy 1, wtedy para filtrów  $(h, g)$  nazywa się komplementarną i pozwala bezpośrednio określić zależności filtrów

$$\begin{aligned} \tilde{h}_e(z) &= g_o(z^{-1}) \\ \tilde{h}_o(z) &= -g_e(z^{-1}) \\ \tilde{g}_e(z) &= -h_o(z^{-1}) \\ \tilde{g}_o(z) &= h_e(z^{-1}) \end{aligned} \quad (12.18)$$

Wykorzystując powyższe własności, filtry analizy i syntezy prezentują się w sposób następujący

$$\begin{aligned} \tilde{h}(z) &= -z^{-1}g(-z^{-1}) \\ \tilde{g}(z) &= -z^{-1}h(-z^{-1}) \end{aligned} \quad (12.19)$$

## Rozdział 13

---

# Schemat liftingu

---

Konsekwencją reprezentacji polifazowej jest metoda liftingu która przekształca macierz polifazową na postać iloczynu macierzy górno i dolnotrójkątnych z jedynkami na przekątnych.

Transformacja zbudowana na schemacie liftingu zawiera iteracyjnie powtarzane bloki predykcji i uaktualnienia (dualny i prymalny lifting).

Bloki te modyfikują próbki nieparzyste wykorzystując liniowy model predykcji z kilku sąsiednich próbek parzystych, a następnie uaktualniają wartości próbek parzystych, aby zachować wartość średnią oryginalnego zbioru próbek.

Schemat liftingu jest techniką wprowadzoną przez Wima Sweldensa, która służy zarówno projektowaniu falek biortogonalnych jak i jest sposobem realizacji biortogonalnej transformacji falkowej [9, 60]. Podczas procesu syntezy lub dekompozycji funkcji wykorzystuje się tzw. bank filtrów, czyli zespół skończonej liczby filtrów, którego filtry są używane w sposób oddzielny do osobnych próbek tej samej funkcji. Dużą zaletą jest to, że cała konstrukcja odbywa się w dziedzinie czasu, nie wykorzystując do tego celu przekształceń Fouriera. Schemat liftingu zapewnia odwracalność transformacji falkowej oraz posiada wystarczająco dużo stopni swobody, aby konstruować przekształcenia zgodnie z przyjętymi założeniami tj. wymagana regularność, liczba momentów, kształt widma, lokalizacja w dziedzinie czasu.

W schemacie liftingu wyróżnia się dwie operacje: lifting dualny (blok predykcji) i lifting prymalny (blok uaktualnienia), dzięki którym można bezpośrednio wpływać na właściwości transformaty falkowej [23, 31].

Podczas prymalnego liftingu następuje zamiana filtrów górnoprzepustowych na nowe, powodując zmianę dualnej funkcji skalującej zachowując przy tym ich komplementarność, opisaną w kolejnym rozdziale.

Lifting dualny w sposób analogiczny zamienia filtr dolnoprzepustowy na jego nowy odpowiednik [38].

### 13.1 Wyznaczanie nowych biortogonalnych filtrów falkowych

Przekształcenie macierzy polifazowej na iloczyn macierzy górno i dolnotrójkątnych z jedynkami na przekątnej pozwala na wyodrębnienie operacji liftingu prymalnego i dualnego z zachowaniem komplementarności filtrów falkowych.

Ponadto, jeżeli para filtrów syntezy  $(h, g)$  jest komplementarna (wyznacznik macierzy polifazowej wynosi 1) wtedy to para filtrów analizy  $(\tilde{h}, \tilde{g})$  jest także komplementarna.

Powyższe własności wynikają bezpośrednio z poniższych twierdzeń:

**Twierdzenie 13.1:** (Prymalny lifting) Jeżeli istnieje komplementarna para filtrów  $(h, g)$ , wtedy nowy filtr  $g^{new}$ , o skończonej odpowiedzi impulsowej i komplementarny do  $h$ , przyjmuje formę [15]:

$$g^{new}(z) = g(z) + h(z)s(z^2) \quad (13.1)$$

gdzie  $s(z)$  jest wielomianem Laurenta.

Analogicznie, jeżeli filtr  $g^{new}$  przyjmuje formę 13.1 wtedy  $g^{new}$  jest komplementarny do  $h$ .

**Dowód:** Współczynniki  $h(z)s(z^2)$  macierzy polifazowej przyjmują postać  $h_e(z)s(z)$  dla części parzystej i  $h_o(z)s(z)$  dla części nieparzystej

$$\begin{aligned} \mathbf{P}^{new}(z) &= \begin{bmatrix} h_e(z) & g_e^{new}(z) \\ h_o(z) & g_o^{new}(z) \end{bmatrix} = \begin{bmatrix} h_e(z) & g_e(z) + h_e(z)s(z) \\ h_o(z) & g_o(z) + h_o(z)s(z) \end{bmatrix} \\ &= \begin{bmatrix} h_e & g_e \\ h_o & g_o \end{bmatrix} \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix} = \mathbf{P}(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix} \end{aligned} \quad (13.2)$$

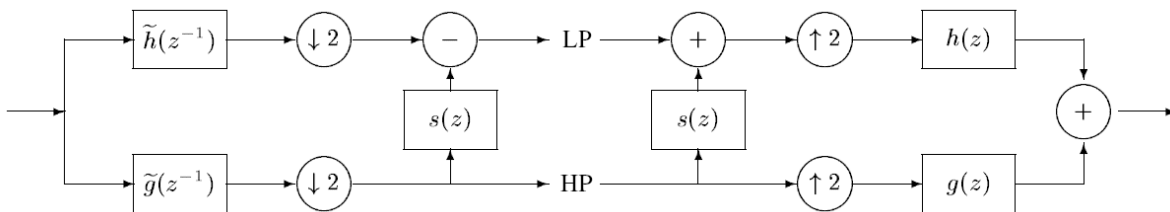
Operacja nie zmienia zatem wyznacznika macierzy [15]  $\square$

Na mocy twierdzenia 13.1 i relacji pomiędzy filtrami dolnoprzepustowymi  $h$  i  $\tilde{h}$  dla dualnej macierzy polifazowej otrzymuje się:

$$\tilde{\mathbf{P}}^{new}(z) = \tilde{\mathbf{P}}(z) \begin{bmatrix} 1 & 0 \\ -s(z^{-1}) & 1 \end{bmatrix} \quad (13.3)$$

Wynika z tego, że nowy filtr analizy  $\tilde{h}$  jest wyrażony wzorem [60]

$$\tilde{h}^{new}(z) = \tilde{h}(z) - \tilde{g}(z)s(z^{-2}) \quad (13.4)$$



Rysunek 13.1: Liftingu prymalny - blok uaktualnienia

**Twierdzenie 13.2:** (Dualny lifting) Jeżeli istnieje komplementarna para filtrów  $(h, g)$ , wtedy nowy filtr  $h^{new}$  o skończonej odpowiedzi impulsowej i komplementarny do  $g$  przyjmuje formę [15]:

$$h^{new}(z) = h(z) + g(z)t(z^2) \quad (13.5)$$

gdzie  $t(z)$  jest wielomianem Laurenta.

Analogicznie, jeżeli filtr  $h^{new}$  przyjmuje formę 13.5 wtedy  $h^{new}$  jest komplementarny do  $g$ .

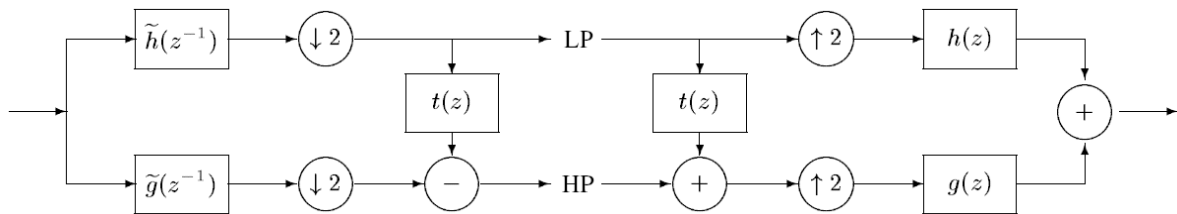
**Dowód:** Współczynniki  $g(z)t(z^2)$  macierzy polifazowej przyjmują postać  $g_e(z)t(z)$  dla części parzystej i  $g_o(z)t(z)$  dla części nieparzystej [64]

$$\begin{aligned} \mathbf{P}^{new}(z) &= \begin{bmatrix} h_e^{new}(z) & g_e(z) \\ h_o^{new}(z) & g_o(z) \end{bmatrix} = \begin{bmatrix} h_e(z) + g_e(z)t(z) & g_e(z) \\ h_o(z) + g_o(z)t(z) & g_o(z) \end{bmatrix} \\ &= \begin{bmatrix} h_e & g_e \\ h_o & g_o \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t(z) & 1 \end{bmatrix} = \mathbf{P}(z) \begin{bmatrix} 1 & 0 \\ t(z) & 1 \end{bmatrix} \end{aligned} \quad (13.6)$$

Powyższa operacja, podobnie jak w przypadku poprzednim nie zmienia wyznacznika macierzy [15]□

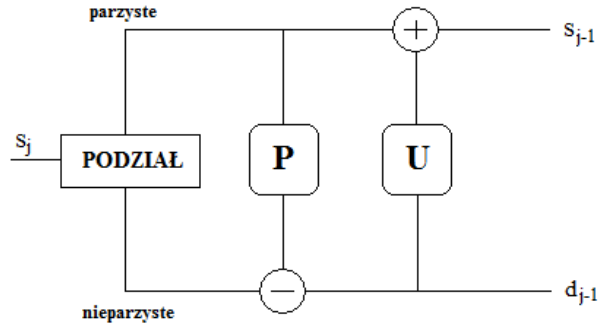
Dualny lifting w sposób analogiczny definiuje także nowy filtr górnoprzepustowy analizy  $\tilde{g}$  [60]

$$\tilde{g}^{new}(z) = \tilde{g}(z) - \tilde{h}(z)\tilde{t}(z^{-2}) \quad (13.7)$$



Rysunek 13.2: Lifting dualny - blok predykcji





Rysunek 13.3: Bloki schematu liftingu

## 13.2 Algorytm Euklidesa

Faktoryzacja macierzy polifazowych, czyli przekształcenie do postaci iloczynowej macierzy górno i dolnotrójkątnych jedynek na przekątnych, realizowana jest za pomocą algorytmu Euklidesa [15] dotyczącego znajdowania największego wspólnego dzielnika NWD dla dwóch wielomianów Laurenta. Główną różnicą pomiędzy NWD dla liczb całkowitych i dla wielomianów Laurenta jest to, że dla liczb całkowitych otrzymuje się unikalne rozwiązanie, zaś dla wielomianów można otrzymać ich wiele.

Aby znaleźć NWD dla dwóch wielomianów Laurenta  $a(z)$  i  $b(z) \neq 0$ , gdzie stopień wielomianu  $|a(z)| \geq |b(z)|$ , przyjmuje się, że  $a_o(z) = a(z)$  i  $b_o(z) = b(z)$ .

Iterację wykonuje się zaczynając od  $i = 0$  [15] dla

$$a_{i+1}(z) = b_i(z) \quad (13.8)$$

$$b_{i+1}(z) = a_i(z) \text{ modulo } b_i(z) \quad (13.9)$$

W kolejnych krokach zmniejsza się stopień wielomianu do indeksu  $n$  kiedy to reszta z dzielenia (modulo)  $b_n(z) = 0$ . Rozwiązaniem jest wtedy  $NWD(a(z), b(z)) = a_n(z)$ .

W przypadku gdy  $|b_{i+1}(z)| < |b_i(z)|$  można odnaleźć takie  $m$ , że  $|b_m(z)| = 0$ . Algorytm kończy się dla  $n = m + 1$  gdzie liczba kroków jest ograniczona dla  $n \leq |b(z)| + 1$ .

Przyjmując, że  $q_{i+1}(z) = \lfloor a_i(z)/b_i(z) \rfloor$  algorytm przyjmuje formę macierzową [67]

$$\begin{bmatrix} a_n(z) \\ 0 \end{bmatrix} = \prod_{i=n}^1 \begin{bmatrix} 0 & 1 \\ 1 & -q_i(z) \end{bmatrix} \begin{bmatrix} a(z) \\ b(z) \end{bmatrix} \quad (13.10)$$

lub postać odwrotną

$$\begin{bmatrix} a(z) \\ b(z) \end{bmatrix} = \prod_{i=1}^n \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_n(z) \\ 0 \end{bmatrix} \quad (13.11)$$

gdzie  $NWD(a(z), b(z)) = a_n(z)$ , a  $q_i(z)$  jest ilorazem całkowitym wielomianów  $a_{i-1}(z)$  i  $b_{i-1}(z)$  [15].

**Przykład:**

Mając dwa wielomiany Laurenta  $a(z) = z^{-1} + 6 + z$  i  $b(z) = 4 + 4z$  wyznacza się ich iloraz.

$$\begin{array}{r} (z^{-1} + 6 + z) : (4 + 4z) = \frac{1}{4}z^{-1} + \frac{5}{4} \\ -z^{-1} - 1 \\ \hline 5 + z \\ -5 - 5z \\ \hline -4z \end{array}$$

gdzie wynik można napisać jako  $z^{-1} + 6 + z = (\frac{1}{4}z^{-1} + \frac{5}{4})(4 + 4z) - 4z$ .

Innym rozwiązaniem może być

$$\begin{array}{r} (z^{-1} + z + 6) : (4 + 4z) = \frac{1}{4}z^{-1} + \frac{1}{4} \\ -z^{-1} - 1 \\ \hline z + 5 \\ -z - 1 \\ \hline 4 \end{array}$$

wtedy to  $z^{-1} + 6 + z = (\frac{1}{4}z^{-1} + \frac{1}{4})(4 + 4z) + 4$  gdzie :

$$\begin{aligned} a_1(z) &= 4 + 4z \\ b_1(z) &= 4 \\ q_1(z) &= \frac{1}{4}z^{-1} + \frac{1}{4} \end{aligned}$$

Postępując analogicznie dla  $n = 2$ , otrzymuje się

$$\begin{aligned} a_2(z) &= 4 \\ b_2(z) &= 0 \\ q_2(z) &= \frac{a_1(z)}{b_1(z)} = 1 + z \end{aligned}$$

Ponieważ  $b_2(z) = 0$  dlatego też wyrażenie można napisać w postaci

$$\begin{bmatrix} z^{-1} + 6 + z \\ 4 + 4z \end{bmatrix} = \begin{bmatrix} \frac{1}{4}z^{-1} + \frac{1}{4} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 + z & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 0 \end{bmatrix} \quad (13.12)$$

gdzie liczba kroków wynosi  $n = |b(z)| + 1 = 2$ .

### 13.3 Faktoryzacja za pomocą schematu liftingu

Jeżeli para filtrów  $(h, g)$  jest komplementarna ( $\det \mathbf{P}(z) = 1$ ) wtedy to współczynniki  $h_e(z)$  i  $h_o(z)$  można doprowadzić się do postaci iloczynowej macierzy zgodnej z algorytmem Euklidesa [64].

Ponieważ iloraz dwóch wielomianów Laurenta nie jest unikalny, dlatego też można zawsze znaleźć takie rozwiązanie, że  $NWD(h_e(z), h_o(z))$  będzie liczbą stałą równą  $K$  taką, że [67]

$$\begin{bmatrix} h_e(z) \\ h_o(z) \end{bmatrix} = \prod_{i=1}^n \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K \\ 0 \end{bmatrix} \quad (13.13)$$

Jeżeli  $|h_o(z)| > |h_e(z)|$  wtedy to pierwszy iloraz  $q_1(z) = 0$ .

W rozważaniach można przyjąć, że  $n$  jest parzyste, w przeciwnym przypadku, jeżeli  $n$  będzie nieparzyste, należy pomnożyć  $h(z)$  przez czynnik  $z$ , zaś  $g(z)$  przez  $-z^{-1}$ , ponieważ taka operacja nie zmienia wyznacznika macierzy polifazowej i pozwala otrzymać parzystą liczbę mnożeń.

Mając filtr  $h(z)$  określa się komplementarny filtr  $g^0$  taki, że

$$\mathbf{P}^0(z) = \begin{bmatrix} h_e(z) & g_e^0(z) \\ h_o(z) & g_o^0(z) \end{bmatrix} = \prod_{i=1}^n \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix} \quad (13.14)$$

Wiedząc, że

$$\begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & q_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_i(z) & 1 \end{bmatrix} \quad (13.15)$$

równanie 13.14 można zapisać w postaci

$$\mathbf{P}^0(z) = \prod_{i=1}^{\frac{n}{2}} \begin{bmatrix} 1 & q_{2i-1}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i}(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix} \quad (13.16)$$

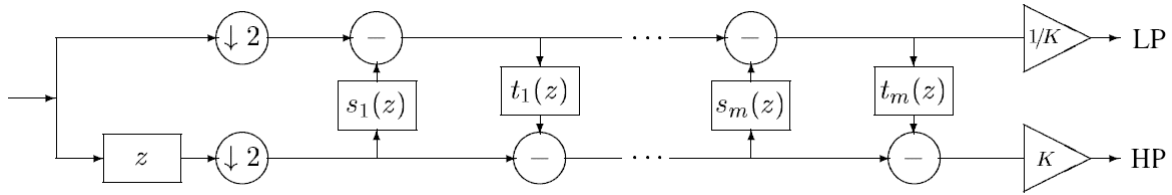
Ponieważ na mocy twierdzenia 13.1 za pomocą filtru  $g^0$  można zawsze uzyskać filtr  $g$  za pomocą operacji

$$\mathbf{P}(z) = \mathbf{P}^0(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}. \quad (13.17)$$

implikacją powyższych własności jest następujące twierdzenie [15]:

**Twierdzenie 13.3:** Dla każdej komplementarnej pary filtrów  $(h, g)$  istnieją takie wielomiany Laurenta  $s_i(z)$  i  $t_i(z)$  dla  $K \neq 0$ , gdzie [45]

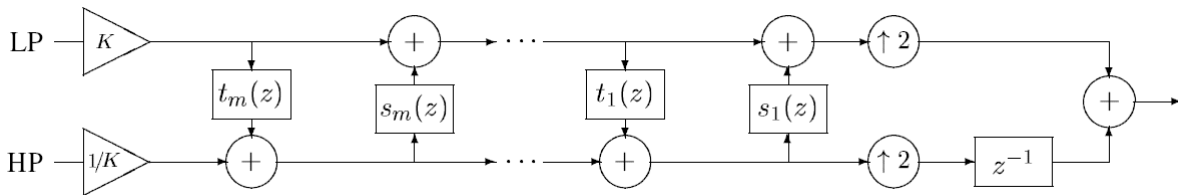
$$\mathbf{P}(z) = \prod_{i=1}^m \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix}. \quad (13.18)$$



Rysunek 13.4: Schemat liftingu ze skalowaniem - analiza funkcji

Dualną macierz polifazową otrzymuje się w sposób analogiczny:

$$\tilde{\mathbf{P}}(z) = \prod_{i=1}^m \begin{bmatrix} 1 & 0 \\ -s_i(z^{-1}) & 1 \end{bmatrix} \begin{bmatrix} 1 & -t_i(z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{K} & 0 \\ 0 & K \end{bmatrix}. \quad (13.19)$$



Rysunek 13.5: Schemat liftingu ze skalowaniem - synteza funkcji

## 13.4 Przykład realizacji schematu liftingu

### 13.4.1 Analiza biortogonalnych falek CDF(2,2) (Cohen-Daubechies-Feauveau)

Dla falek CDF(2,2)\* para filtrów analizy przyjmuje postać [67, 64]:

$$\begin{aligned} \tilde{h}(z) &= -\frac{1}{8}z^{-2} + \frac{1}{4}z^{-1} + \frac{3}{4} + \frac{1}{4}z - \frac{1}{8}z^2 \\ \tilde{g}(z) &= \frac{1}{4}z^{-2} - \frac{1}{2}z^{-1} + \frac{1}{4} \end{aligned} \quad (13.20)$$

W pierwszym kroku tworzy się podział filtrów na część parzystą i nieparzystą zgodnie ze wzorami 12.7 i 12.8

$$\begin{aligned} \tilde{h}(z) &= \underbrace{\left\{-\frac{1}{8}z^{-2} + \frac{3}{4} - \frac{1}{8}z^2\right\}}_{\tilde{h}_e(z^2)} + z^{-1} \underbrace{\left\{\frac{1}{4} + \frac{1}{4}z^2\right\}}_{\tilde{h}_o(z^2)} \\ \tilde{g}(z) &= \underbrace{\left\{\frac{1}{4}z^{-2} + \frac{1}{4}\right\}}_{\tilde{g}_e(z^2)} + z^{-1} \underbrace{\left\{-\frac{1}{2}\right\}}_{\tilde{g}_o(z^2)} \end{aligned}$$

\*Liczby w nawiasie informują o liczbie momentów znikających dla falki prymalnej i dualnej. W opisywanym przypadku  $\tilde{N} = 2$  i  $N = 2$ .

Ponieważ

$$\tilde{h}_e(z) = -\frac{1}{8}z^{-1} + \frac{3}{4} - \frac{1}{8}z$$

$$\tilde{h}_o(z) = \frac{1}{4} + \frac{1}{4}z$$

$$\tilde{g}_e(z) = \frac{1}{4}z^{-1} + \frac{1}{4}$$

$$\tilde{g}_o(z) = -\frac{1}{2}$$

macierz polifazowa przyjmuje postać:

$$\tilde{\mathbf{P}}(z) = \begin{bmatrix} -\frac{1}{8}z^{-1} + \frac{3}{4} - \frac{1}{8}z & \frac{1}{4} + \frac{1}{4}z \\ \frac{1}{4}z^{-1} + \frac{1}{4} & -\frac{1}{2} \end{bmatrix}$$

Wyznacznik macierzy  $\det \tilde{\mathbf{P}}(z) = -\frac{1}{2}$  wykazuje brak komplementarności pary filtrów, z tej przyczyny należy zastosować skalowanie zgodnie ze wzorem 12.18.

Dzięki przeskalowaniu uzyskuje się w prosty sposób współczynniki macierzy polifazowej

$$\mathbf{P}(z) = [\tilde{\mathbf{P}}(z^{-1})^T]^{-1} = \begin{bmatrix} 1 & \frac{1}{2} + \frac{1}{2}z \\ \frac{1}{2}z^{-1} + \frac{1}{2} & \frac{1}{4}z^{-1} - \frac{3}{2} + \frac{1}{4}z \end{bmatrix}$$

```
P[z_] := {Podzial[ht], Podzial[gt]};
MatrixForm[P[z]]
```

MatrixForm=

$$\begin{pmatrix} \frac{3}{4} - \frac{1}{8z} - \frac{z}{8} & \frac{1}{4} + \frac{z}{4} \\ \frac{1}{4} + \frac{1}{4z} & -\frac{1}{2} \end{pmatrix}$$

Rysunek 13.6: Macierz polifazowa dla CDF(2,2)

**Algorytm 13.1** Wyznaczanie macierzy polifazowej

---

```

ht[z_] := -1/8 z^-2 + 1/4 z^-1 + 3/4 + 1/4 z - 1/8 z^2;
gt[z_] := 1/4 z^-2 - 1/2 z^-1 + 1/4;

Podzial::usage = "Kreowanie macierzy polifazowej";
Podzial[ht_] := Block[{hte = 0, hto = 0, i, c, wsp, z, minpow = 0, maxpow = 0},
  For[i = -30, i < 30, i++,
    c = ExpandAll[ht[z] / z^i];
    wsp[i] = ExpandAll[c - Integrate[D[c, z], z]];
    If[wsp[i] ≠ 0, If[minpow == 0, minpow = i]; maxpow = i]
  ]

  For[i = minpow, i ≤ maxpow, i++,
    If[Mod[i, 2] == 0,
      hto += wsp[i] * z^i,
      hte += wsp[i] * z^(i+1)
    ]
  ];

  hto = hto /. z → Sqrt[z];
  hte = hte /. z → Sqrt[z];

  Return[{hto, hte}]
];

```

---

Ponieważ macierz polifazowa wyraża się wzorem

$$\mathbf{P}(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix}$$

filtry syntezy wyznacza się w sposób bezpośredni

$$\begin{aligned}
 h(z) &= h_e(z^2) + z^{-1}h_o(z^2) \\
 &= 1 + z^{-1}\left(\frac{1}{2}z^{-2} + \frac{1}{2}\right) \\
 &= \frac{1}{2}z^{-3} + \frac{1}{2}z^{-1} + 1 \\
 g(z) &= g_e(z^2) + z^{-1}g_o(z^2) \\
 &= \frac{1}{2} + \frac{1}{2}z^2 + z^{-1}\left(\frac{1}{4}z^{-2} - \frac{3}{2} + \frac{1}{4}z^2\right) \\
 &= \frac{1}{4}z^{-3} - \frac{3}{2}z^{-1} + \frac{1}{2} + \frac{1}{4}z + \frac{1}{2}z^2
 \end{aligned}$$

Oby otrzymać kolejne bloki schematu liftingu, macierz polifazową poddaje się procesowi faktoryzacji za pomocą algorytmu Euklidesa.

Wykorzystując zależność [65]

$$\begin{aligned}
\tilde{\mathbf{P}}^{new}(z) &= \tilde{\mathbf{P}}(z) \begin{bmatrix} 1 & 0 \\ -s(z^{-1}) & 1 \end{bmatrix} \\
&= \begin{bmatrix} -\frac{1}{8}z^{-1} + \frac{3}{4} - \frac{1}{8}z & \frac{1}{4} + \frac{1}{4}z \\ \frac{1}{4}z^{-1} + \frac{1}{4} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -s(z^{-1}) & 1 \end{bmatrix} \\
&= \begin{bmatrix} \tilde{h}_e^{new}(z) & \frac{1}{4} + \frac{1}{4}z \\ \tilde{g}_e^{new}(z) & -\frac{1}{2} \end{bmatrix}
\end{aligned}$$

wyznacza się układ równań w postaci

$$\begin{cases} \tilde{h}_e^{new}(z) = (-\frac{1}{8}z^{-1} + \frac{3}{4} - \frac{1}{8}z) - (\frac{1}{4} + \frac{1}{4}z)s(z^{-1}) \\ \tilde{g}_e^{new}(z) = (\frac{1}{4}z^{-1} + \frac{1}{4}) + \frac{1}{2}s(z^{-1}) \end{cases}$$

lub w przekształconej formie

$$\begin{cases} -\frac{1}{8}z^{-1} + \frac{3}{4} - \frac{1}{8}z = (\frac{1}{4} + \frac{1}{4}z)s(z^{-1}) + \tilde{h}_e^{new}(z) \\ \frac{1}{4}z^{-1} + \frac{1}{4} = -\frac{1}{2}s(z^{-1}) + \tilde{g}_e^{new}(z) \end{cases}$$

Za pomocą niejednoznaczności algorytmu Euklidesa opisanego w rozdziale 13.2 otrzymuje się trzy rozwiązania dla równania

$$-\frac{1}{8}z^{-1} + \frac{3}{4} - \frac{1}{8}z = \begin{cases} (-\frac{1}{2}z^{-1} + \frac{7}{2}) (\frac{1}{4} + \frac{1}{4}z) - z \\ (-\frac{1}{2}z^{-1} - \frac{1}{2}) (\frac{1}{4} + \frac{1}{4}z) + 1 \\ (\frac{7}{2}z^{-1} - \frac{1}{2}) (\frac{1}{4} + \frac{1}{4}z) - z^{-1} \end{cases} \quad (13.21)$$

Ponieważ drugie rozwiązanie posiada symetryczne współczynniki, ze względów subiektywnych, jako rozwiązanie można przyjąć<sup>†</sup>

$$\begin{cases} s(z^{-1}) = -\frac{1}{2}z^{-1} - \frac{1}{2} \\ \tilde{h}_e^{new}(z) = 1 \\ \tilde{g}_e^{new}(z) = (\frac{1}{4}z^{-1} + \frac{1}{4}) + \frac{1}{2}s(z^{-1}) \\ = 0 \end{cases}$$

<sup>†</sup>Rozwiązaniem może być jakiegokolwiek z rozwiązań 13.21

Podsumowując obliczenia, rozwiązaniem przykładu jest iloczyn macierzowy, gdzie

$$\begin{aligned}\tilde{\mathbf{P}}(z) &= \tilde{\mathbf{P}}^{new}(z) \begin{bmatrix} 1 & 0 \\ s(z^{-1}) & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & \frac{1}{4} + \frac{1}{4}z \\ 0 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2}z^{-1} - \frac{1}{2} & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{4} + \frac{1}{4}z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{2}z^{-1} - \frac{1}{2} & 1 \end{bmatrix}\end{aligned}$$

Z formy iloczynu macierzowego uzyskuje się w sposób bezpośredni kolejne bloki schematu liftingu przedstawione w tabeli 13.1.

Operacja	Algorytm
Podział	$s_i \leftarrow x_{2i}$
	$d_i \leftarrow x_{2i+1}$
Lifting dualny	$d_i \leftarrow d_i - \frac{1}{2}(s_i + s_{i+1})$
Lifting prymalny	$s_i \leftarrow s_i + \frac{1}{4}(d_{i-1} + d_i)$
Skalowanie	$s_i \leftarrow -\frac{1}{2}s_i$

Tablica 13.1: Schemat liftingu dla CDF(2,2)

### 13.4.2 Faktoryzacja falek Haara

Innym przykładem realizacji analizy funkcji za pomocą schematu liftingu jest układ z nieunormowaną postacią filtrów Haara, gdzie

$$\begin{aligned}h(z) &= 1 + z^{-1} \\ g(z) &= -\frac{1}{2} + \frac{1}{2}z^{-1}\end{aligned}\tag{13.22}$$

Macierz polifazowa przyjmuje formę

$$\mathbf{P}(z) = \begin{bmatrix} 1 & -\frac{1}{2} \\ 1 & \frac{1}{2} \end{bmatrix}$$

Ponieważ

$$\tilde{\mathbf{P}}(z) = [\mathbf{P}(z^{-1})^{-1}]^T = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -1 & 1 \end{bmatrix},$$

filtry analizy mają postać  $\tilde{h}(z) = \frac{1}{2}z^{-1} + \frac{1}{2}$  i  $\tilde{g}(z) = z^{-1} - 1$ .

Przeprowadzając faktoryzację dla  $\mathbf{P}(z)$  rozwiązaniem jest postać iloczynowa, gdzie



$$\begin{aligned} \mathbf{P}(z) &= \begin{bmatrix} 1 & -\frac{1}{2} \\ 1 & \frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & g_e^{new}(z) \\ 1 & g_o^{new}(z) \end{bmatrix} \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & 1 \end{bmatrix}. \end{aligned}$$

Operacja analizy	Algorytm	Operacja syntezy	Algorytm
Podział	$s_i \leftarrow x_{2i}$	Lifting prymalny	$s_i \leftarrow s_i - \frac{1}{2}d_i$
	$d_i \leftarrow x_{2i+1}$	Lifting dualny	$d_i \leftarrow d_i + s_i$
Lifting dualny	$d_i \leftarrow d_i - s_i$	Łączenie	$x_{2i+1} = d_i$
Lifting prymalny	$s_i \leftarrow s_i + \frac{1}{2}d_i$		$x_{2i} = s_i$

Tablica 13.2: Schemat liftingu dla nieznormalizowanej falki Haara

### 13.4.3 Rotacja Givensa

Innym przykładem dla celów faktoryzacji jest dwuwymiarowa macierz rotacji [63, 15]

$$G = \begin{bmatrix} \cos(z) & -\sin(z) \\ \sin(z) & \cos(z) \end{bmatrix} \quad (13.23)$$

która jest macierz polifazową  $\mathbf{P}(z)$  o filtrach komplementarnych  $h(z) = \sin(z)z^{-1} + \cos(z)$  i  $g(z) = \cos(z)z^{-1} - \sin(z)$ , gdzie  $z \neq \frac{\pi}{2} + k\pi$ .

Postać iloczynowa macierzy wyraża w sposób następujący

$$\begin{bmatrix} \cos(z) & -\sin(z) \\ \sin(z) & \cos(z) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \operatorname{tg}(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -\sin(z)\cos(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(z) & 0 \\ 0 & \operatorname{sec}(z) \end{bmatrix} \quad (13.24)$$

lub równoważne (dla  $z \neq k\frac{\pi}{2}$ , gdzie  $k \in \mathbb{Z}$ )

$$\begin{bmatrix} \cos(z) & -\sin(z) \\ \sin(z) & \cos(z) \end{bmatrix} = \begin{bmatrix} 1 & \operatorname{ctg}(z) - \operatorname{cosec}(z) \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \operatorname{ctg}(z) - \operatorname{cosec}(z) \\ 0 & 1 \end{bmatrix} \quad (13.25)$$

Reprezentacja schematu liftingu dla rozwiązania 13.25 przedstawia się następująco:

Operacja syntezy	Algorytm
Lifting prymalny	$s_i \leftarrow s_i + (\operatorname{ctg}(z) - \operatorname{cosec}(z))d_i$
Lifting dualny	$d_i \leftarrow d_i + \sin(z)s_i$
Lifting prymalny	$s_i \leftarrow s_i + (\operatorname{ctg}(z) - \operatorname{cosec}(z))d_i$
Łączenie	$x_{2i-1} = d_i$
	$x_{2i} = s_i$

Tablica 13.3: Schemat liftingu dla rotacji Givensa

Złożenie trzech macierzy w geometrii oznacza, że każdy obrót dwuwymiarowy można zapisać jako złożenie trzech rzutów.

Ponadto, dla każdego komplementarnych filtrów używając transformacji obrotu nie traci się ich komplementarności.

#### 13.4.4 Skalowanie

Aby uniknąć skalowania jako ostatniego kroku liftingu można zastosować algorytm Euklidesa dla macierzy skalującej w postaci [63, 15]

$$\begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix} = \begin{bmatrix} 1 & K - K^2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{K} & 1 \end{bmatrix} \begin{bmatrix} 1 & K - 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

lub też jego innej formie

$$\begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 - \frac{1}{K} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ K & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{K^2} - \frac{1}{K} \\ 0 & 1 \end{bmatrix}.$$

Takie rozwiązanie jest istotnym elementem gdy skalowanie nie jest operacją odwracalną na przykład w przypadku transformacji falkowej na wartościach całkowitych.

### 13.5 Biortogonalne falki CDF

W bieżącym rozdziale, zaprezentowane zostało zestawienie biortogonalnych falek CDF, czyli falek biortogonalnych o maksymalnej liczbie momentów znikających  $N$  i  $\tilde{N}$  dla danej długości nośnika.

Parametry falek zostały pogrupowane i przedstawione w formie tabeli z informacjami o wartościach ich filtrów, jak i również blokach liftingu [65, 66, 64] wykorzystywanych podczas transformacji falkowej.

$N$	$\tilde{N}$	Parametry definiujące
1	x	$\tilde{g}(z) = \frac{1}{2}z^{-1} - \frac{1}{2}$ $t_1(z) = -1$ $K_1 = 1$
1	1	$\tilde{h}(z) = \frac{1}{2}z^{-1} + \frac{1}{2}$ $s_1(z) = \frac{1}{2}$ $K_2 = \frac{1}{2}$
1	3	$\tilde{h}(z) = -\frac{1}{16}z^{-3} + \frac{1}{16}z^{-2} + \frac{1}{2}z^{-1} + \frac{1}{2} + \frac{1}{16} - \frac{1}{16}z^2$ $s_1(z) = -\frac{1}{16}z^{-1} + \frac{1}{2} + \frac{1}{16}z$ $K_2 = \frac{1}{2}$
1	5	$\tilde{h}(z) = \frac{3}{256}z^{-5} - \frac{3}{256}z^{-4} - \frac{11}{128}z^{-3} + \frac{11}{128}z^{-2} + \frac{1}{2}z^{-1} + \frac{1}{2} + \frac{11}{128}z - \frac{11}{128}z^2 - \frac{3}{256}z^3 + \frac{3}{256}z^4$ $s_1(z) = \frac{3}{256}z^{-2} - \frac{11}{128}z^{-1} + \frac{1}{2} + \frac{11}{128}z - \frac{3}{256}z^2$ $K_2 = \frac{1}{2}$

Tablica 13.4: Dekompozycja dla filtrów analizy biortogonalnych falek CDF(1,x) z  $N = 1$  i  $\tilde{N} = 1, 3, 5$  znikającymi momentami

$N$	$\tilde{N}$	Parametry definiujące
2	x	$\tilde{g}(z) = \frac{1}{4}z^{-2} - \frac{1}{2}z^{-1} + \frac{1}{4}$ $t_1(z) = -\frac{1}{2}z^{-1} - \frac{1}{2}$ $K_1 = 1$
2	2	$\tilde{h}(z) = -\frac{1}{8}z^{-2} + \frac{1}{4}z^{-1} + \frac{3}{4} + \frac{1}{4}z - \frac{1}{8}z^2$ $s_1(z) = \frac{1}{4} + \frac{1}{4}z$ $K_2 = -\frac{1}{2}$
2	4	$\tilde{h}(z) = \frac{3}{128}z^{-4} - \frac{3}{64}z^{-3} - \frac{1}{8}z^{-2} + \frac{19}{64}z^{-1} + \frac{45}{64} + \frac{19}{64}z - \frac{1}{8}z^2 - \frac{3}{64}z^3 + \frac{3}{128}z^4$ $s_1(z) = -\frac{3}{64}z^{-1} + \frac{19}{64} + \frac{19}{64}z - \frac{3}{64}z^2$ $K_2 = -\frac{1}{2}$
2	6	$\tilde{h}(z) = -\frac{5}{1024}z^{-6} + \frac{5}{512}z^{-5} + \frac{17}{512}z^{-4} - \frac{39}{512}z^{-3} - \frac{123}{1024}z^{-2} + \frac{81}{256}z^{-1} + \frac{175}{256}$ $+ \frac{81}{256}z - \frac{123}{1024}z^2 - \frac{39}{512}z^3 + \frac{17}{512}z^4 + \frac{5}{512}z^5 - \frac{5}{1024}z^6$ $s_1(z) = \frac{5}{512}z^{-2} - \frac{39}{512}z^{-1} + \frac{81}{256} + \frac{81}{256}z - \frac{39}{512}z^2 + \frac{5}{512}z^3$ $K_2 = -\frac{1}{2}$

Tablica 13.5: Dekompozycja dla filtrów analizy biortogonalnych falek CDF(2,x) z  $N = 2$  i  $\tilde{N} = 2, 4, 6$  znikającymi momentami

$N$	$\tilde{N}$	Parametry definiujące
4	x	$\tilde{g}(z) = -\frac{1}{16}z^{-3} + \frac{1}{4}z^{-2} - \frac{3}{8}z^{-1} + \frac{1}{4} - \frac{1}{16}z$ $t_1(z) = 0$ $s_1(z) = -\frac{1}{4} - \frac{1}{4}z$ $t_2(z) = -z^{-1} - 1$ $K_1 = 2$
4	2	$\tilde{h}(z) = \frac{3}{32}z^{-3} - \frac{3}{8}z^{-2} + \frac{5}{32}z^{-1} + \frac{5}{4} + \frac{5}{32}z - \frac{3}{8}z^2 + \frac{3}{32}z^3$ $s_2(z) = \frac{3}{16} + \frac{3}{16}z$ $K_2 = -\frac{1}{4}$
4	4	$\tilde{h}(z) = -\frac{5}{256}z^{-5} + \frac{5}{64}z^{-4} - \frac{1}{256}z^{-3} - \frac{3}{8}z^{-2} + \frac{35}{128}z^{-1} + \frac{35}{32}$ $+ \frac{35}{128}z - \frac{3}{8}z^2 - \frac{1}{256}z^3 + \frac{5}{64}z^4 - \frac{5}{256}z^5$ $s_2(z) = -\frac{5}{128}z^{-1} + \frac{29}{128} + \frac{29}{128}z - \frac{5}{128}z^2$ $K_2 = -\frac{1}{4}$
4	6	$\tilde{h}(z) = \frac{35}{8192}z^{-7} - \frac{35}{2048}z^{-6} - \frac{55}{8192}z^{-5} + \frac{115}{1024}z^{-4} - \frac{557}{8192}z^{-3} - \frac{733}{2048}z^{-2} + \frac{2625}{8192}z^{-1} + \frac{525}{512}$ $+ \frac{2625}{8192}z - \frac{733}{2048}z^2 - \frac{557}{8192}z^3 + \frac{115}{1024}z^4 - \frac{55}{8192}z^5 - \frac{35}{2048}z^6 + \frac{35}{8192}z^7$ $s_2(z) = \frac{35}{4096}z^{-2} - \frac{265}{4096}z^{-1} + \frac{499}{2048} + \frac{499}{2048}z - \frac{265}{4096}z^2 + \frac{35}{4096}z^3$ $K_2 = -\frac{1}{4}$

Tablica 13.6: Dekompozycja dla filtrów analizy biortogonalnych falek CDF(4,x) z  $N = 4$  i  $\tilde{N} = 2, 4, 6$  znikającymi momentami

## 13.6 Interpolacja funkcji za pomocą biortogonalnej transformaty falkowej

Istnieje klasa transformacji interpolacyjnych która ze względu na dużą efektywność dekompozycji danych wraz z prostą realizacją algorytmiczną jest często wykorzystywana w schematach kompresji stratno-bezstratnej.

Filtry implementujące transformację interpolacyjną nazywane są filtrami interpolacyjnymi, zaś symetryczne falki skojarzone z interpolacyjnymi funkcjami skalującymi - falkami interpolacyjnymi[61, 45].

Dla tego typu filtrów, faktoryzacja macierzy polifazowej syntezy układu prowadzi do dwóch bloków schematu liftingu [45]:

$$\mathbf{P}(z) = \begin{bmatrix} 1 & g_e(z) \\ h_o(z) & 1 + h_o(z)g_e(z) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ h_o(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & g_e(z) \\ 0 & 1 \end{bmatrix} \quad (13.26)$$

co też implikuje  $h_e(z) = 1$ .

Podczas interpolacji w najprostszym przypadku można przyjąć, że wartość próbki nieparzystej  $s_i^{(o)}$  jest taka sama jak wartość próbki parzystej  $s_i^{(e)}$  (rys. 13.7a). Stąd też, jeżeli blok predykcji  $\mathbf{P}$  jest liniowym filtrem nieprzyczynowym SOI (skończoną odpowiedzią impulsową) wtedy to można go opisać za pomocą jednego niezerowego współczynnika predykcji  $p_0$ , tzw. predyktora rzędu I, gdzie [42, 72]

$$d_{i-1} = s_i^{(o)} - P(s_i^{(e)}) = s_i^{(o)} - p_0 s_i^{(e)} \quad (13.27)$$

$$s_{i-1} = s_i^{(e)} + U(d_{i-1}) \quad (13.28)$$

Zakładając analogicznie, że uaktualnienie jest liniową nieprzyczynową kombinacją współczynników  $d_{j-1}$  wtedy

$$\begin{aligned}
 s_{i-1} &= s_j^{(e)} + u_0 d_{j-1} \\
 &= s_j^{(e)} + u_0 (s_i^{(o)} - p_0 s_i^{(e)}) . \\
 &= (1 - u_0 p_0) s_j^{(e)} + u_0 s_j^{(o)}
 \end{aligned} \tag{13.29}$$

Jedną z cech schematu liftingu jest to, że wartość średnia na różnych poziomach rozdzielczości musi mieć tą samą wartość co funkcja oryginalna [31, 65].

Ponieważ wartość średnia dla poziomu  $i$  wynosi

$$\bar{s}_i = \frac{1}{N} \sum_{n=0}^{N-1} s_i(n) = \frac{1}{N} \left( \sum_{n=0}^{\frac{N}{2}-1} s_i^{(e)}(n) + \sum_{n=0}^{\frac{N}{2}-1} s_i^{(o)}(n) \right) \tag{13.30}$$

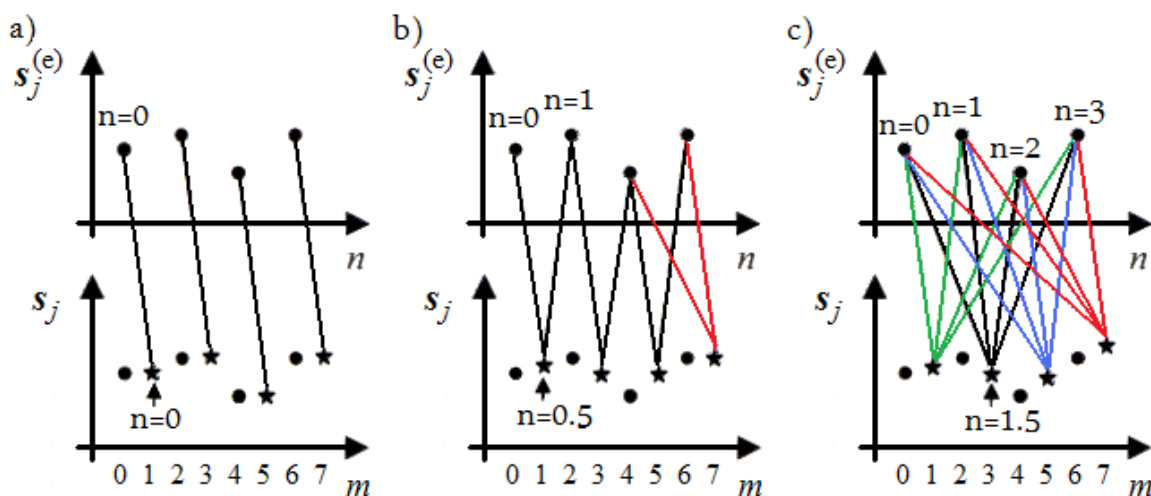
zaś dla  $i - 1$  z uwzględnieniem wzoru 13.28

$$\bar{s}_{i-1} = \frac{2}{N} \sum_{n=0}^{\frac{N}{2}-1} s_{i-1}(n) = \frac{1}{N} \left( \sum_{n=0}^{\frac{N}{2}-1} (2 - 2u_0 p_0) s_j^{(e)} + \sum_{n=0}^{\frac{N}{2}-1} 2u_0 s_j^{(o)} \right) \tag{13.31}$$

na mocy równości  $\bar{s}_i = \bar{s}_{i-1}$  wartość  $u_0 = \frac{1}{2}$  i  $p_0 = 1$ .

Powyższy przykład realizuje transformatę falkową z wykorzystaniem nieunormowanej funkcji Haara. Jest to jedyny przypadek ortogonalnej bazy który może być realizowany za pomocą schematu predykcji i uaktualnienia.

Błąd predykcji w powyższym przypadku wynosi zero wtedy, gdy dwa kolejne współczynniki posiadają te same wartości.



Rysunek 13.7: Przykłady interpolacji funkcji z predyktorem rzędu a) I, b) II i c) IV [72]

Istnieją interpolacje wyższego rzędu<sup>‡</sup> gdzie dla bloku predykcji stosuje się więcej aniżeli jedną próbkę parzystą [61, 72].

Predyktor drugiego rzędu (rys. 13.7b) szacuje zatem, próbkę nieparzystą na podstawie sąsiadujących z nią dwóch próbek parzystych.

W tym przypadku, błąd predykcji wynosi zero gdy nieparzysta próbka jest zależna liniowo od próbek parzystych tzn. wszystkie trzy próbki leżą na jednej prostej  $s(n) = a_1n + a_0$ .

Analogicznie, dla predyktora rzędu IV do obliczenia próbki nieparzystej wykorzystuje się cztery próbki parzyste, dwie leżące kolejno po lewej i dwie po prawej stronie próbki nieparzystej (rys. 13.7c - linia czarna).

Dla powyższego przypadku, wartość próbki określa się na podstawie średniej ważonej próbek i ściśle określonych wag  $p_0, p_1, p_2, p_3$  będących współczynnikami wektora predykcji.

Jeżeli analizowany fragment funkcji jest wielomianem co najwyżej trzeciego stopnia  $s(n) = a_3n^3 + a_2n^2 + a_1n + a_0$  dla  $n = 0, 1, 2, 3$  wówczas oszacowana wartość próbki środkowej wynosi  $s(1.5)$  [72].

Podczas obliczeń można zauważyć istotną zależność, że stopień wielomianu interpolującego wykorzystywanego podczas operacji predykcji jest o 1 mniejszy od rzędu predyktora czyli liczby próbek wykorzystywanych podczas bloku predykcji.

Analiza przypadków granicznych dla predyktorów wyższych rzędów pozwala zobrazować problem deficytu próbek parzystych wykorzystywanych podczas szacowania próbek nieparzystych. Niedobór próbek z początku dziedziny funkcji znajduje się po lewej stronie próbki nieparzystej, zaś po stronie prawej dla próbek z końca dziedziny funkcji badanej.

Rozwiązaniem takim jest estymacja nowych próbek nieparzystych na bazie dostępnych najbliższych próbek z odpowiednio zmodyfikowanymi wagami [72].

W klasycznym zespole filtrów takie podejście do problemu nie jest możliwe, stosuje się wtedy zasadę dokładania próbek na zasadzie odbicia lustrzanego lub wykorzystuje się okresowy charakter funkcji.

Przyjmując zasadę periodyzacji próbek można wykazać, że dla wektorów predykcji i uaktualnienia przy zachowaniu tej samej długości wektora zachodzi zależność ich współczynników  $u_j = \frac{p_j}{2}$ .

Przykładem takim może być predyktor rzędu II gdzie  $|\vec{p}| = 2$ .

<sup>‡</sup>Rzędem predyktora nazywa się jego długość tzn. liczbę próbek wykorzystywaną podczas szacowania próbek nieparzystych.

Stosując interpolację z wykorzystaniem predyktora rzędu II, próbkę nieparzystą szacuje się za pomocą dwóch sąsiadujących próbek parzystych.

Interpolacja odbywa się wtedy za pomocą funkcji liniowej, gdzie przy zachowaniu równomierności próbek [58] otrzymuje się

$$\tilde{s}_{i-1}^{(o)}(n) = \frac{s_i^{(e)}(n-1) + s_i^{(e)}(n)}{2} = \frac{1}{2}s_i^{(e)}(n-1) + \frac{1}{2}s_i^{(e)}(n) \quad (13.32)$$

stąd  $p_0 = \frac{1}{2}$  i  $p_1 = \frac{1}{2}$ .

Ponieważ współczynniki detalu wylicza się z opóźnieniem zgodnie z zasadą "leniwej" falki [61], gdzie

$$d_{i-1}(n-1) = s_i^{(o)}(n-1) - \tilde{s}_{i-1}^{(o)}(n) = s_i^{(o)}(n-1) - \frac{1}{2}s_i^{(e)}(n-1) - \frac{1}{2}s_i^{(e)}(n), \quad (13.33)$$

operację uaktualnienia wyraża się następująco [72]

$$s_{i-1}(n) = s_i^{(e)}(n) + u_0 d_{j-1}(n-1) + u_1 d_{j-1}(n) \quad (13.34)$$

Po podstawieniu wzoru 13.33 do 13.34 wyznacza się

$$s_{i-1}(n) = s_i^{(e)}(n) + u_0 \left( s_i^{(o)}(n-1) - \frac{1}{2}s_i^{(e)}(n-1) - \frac{1}{2}s_i^{(e)}(n) \right) + u_1 \left( s_i^{(o)}(n) - \frac{1}{2}s_i^{(e)}(n) - \frac{1}{2}s_i^{(e)}(n+1) \right). \quad (13.35)$$

Na mocy równości wartości średniej gdzie  $\bar{s}_i = \bar{s}_{i-1}$  tj

$$\frac{1}{N} \sum_{n=0}^{N-1} s_i(n) = \frac{2}{N} \sum_{n=0}^{\frac{N}{2}-1} s_{i-1}(n) \quad (13.36)$$

oblicza się regułę

$$\begin{aligned}
 \sum_{n=0}^{N-1} s_i(n) &= 2 \sum_{n=0}^{\frac{N}{2}-1} s_{i-1}(n) \\
 \sum_{n=0}^{\frac{N}{2}-1} [s_i^{(e)}(n) + s_i^{(o)}(n)] &= 2 \sum_{n=0}^{\frac{N}{2}-1} [s_i^{(e)}(n) + u_0(s_i^{(o)}(n-1) - \frac{1}{2}s_i^{(e)}(n-1) - \frac{1}{2}s_i^{(e)}(n))] \\
 &\quad + 2 \sum_{n=0}^{\frac{N}{2}-1} [u_1(s_i^{(o)}(n) - \frac{1}{2}s_i^{(e)}(n) - \frac{1}{2}s_i^{(e)}(n+1))] \\
 \sum_{n=0}^{\frac{N}{2}-1} [s_i^{(e)}(n) + s_i^{(o)}(n)] &= 2 \sum_{n=0}^{\frac{N}{2}-1} (1 - \frac{1}{2}u_0 - \frac{1}{2}u_0 - \frac{1}{2}u_1 - \frac{1}{2}u_1) s_i^{(e)}(n) \\
 &\quad + 2 \sum_{n=0}^{\frac{N}{2}-1} (u_0 + u_1) s_i^{(o)}(n) \\
 \sum_{n=0}^{\frac{N}{2}-1} [s_i^{(e)}(n) + s_i^{(o)}(n)] &= \sum_{n=0}^{\frac{N}{2}-1} \underbrace{[2(1 - u_0 - u_1)]}_1 s_i^{(e)}(n) + \underbrace{[2(u_0 + u_1)]}_1 s_i^{(o)}(n)
 \end{aligned} \tag{13.37}$$

Poprzez porównanie obu stron równości 13.37 otrzymuje się zależność  $u_0 + u_1 = \frac{1}{2}$ .

Ponieważ dla spełnienia symetrii można przyjąć [58], że  $u_0 = u_1 = \frac{1}{4}$ , ze współczynników wektora predykcji i wektora uaktualnienia wynika zależność  $\vec{U} = \frac{\vec{P}}{2}$ .

### 13.7 Wyznaczanie współczynników predykcji dla filtrów interpolowanych

Realizacja interpolacji wielomianowej, zaproponowana przez Wima Sweldensa[58], za pomocą schematu liftingu odbywa się przy pomocy parzystej liczby próbek funkcji o liczbie  $N$ . Aby wyznaczyć nieparzyste próbki  $s_i^{(o)}$  na podstawie próbek parzystych  $s_i^{(e)}$  konstruuje się wielomian stopnia  $N - 1$  który lokalnie opisuje charakterystykę funkcji [60]

$$s_j^{(e)}(n) = a_{N-1}n^{N-1} + a_{N-2}n^{N-2} + \dots + a_1n + a_0 \tag{13.38}$$

gdzie wartości  $n$  są argumentami próbek parzystych tj.

$$\left\{ \begin{array}{l} s_j^{(e)}(n_1) = a_{N-1}n_1^{N-1} + a_{N-2}n_1^{N-2} + \dots + a_1n_1 + a_0 \\ s_j^{(e)}(n_2) = a_{N-1}n_2^{N-1} + a_{N-2}n_2^{N-2} + \dots + a_1n_2 + a_0 \\ \vdots \\ s_j^{(e)}(n_N) = a_{N-1}n_N^{N-1} + a_{N-2}n_N^{N-2} + \dots + a_1n_N + a_0 \end{array} \right. \tag{13.39}$$

Równorzędna forma macierzowa przedstawia się następująco



$$\begin{bmatrix} s_j^{(e)}(n_1) \\ s_j^{(e)}(n_2) \\ \vdots \\ s_j^{(e)}(n_N) \end{bmatrix} = \begin{bmatrix} n_1^{N-1} & n_1^{N-2} & \dots & 1 \\ n_2^{N-1} & n_2^{N-2} & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ n_N^{N-1} & n_N^{N-2} & \dots & 1 \end{bmatrix} \begin{bmatrix} a_{N-1} \\ a_{N-2} \\ \vdots \\ a_0 \end{bmatrix} = \mathbf{B}_{N \times N} \begin{bmatrix} a_{N-1} \\ a_{N-2} \\ \vdots \\ a_0 \end{bmatrix}. \quad (13.40)$$

Dzięki rachunkom macierzowym i znajomości wartości funkcji dla kolejnych próbek można obliczyć kolejne współczynniki wielomianu interpolacyjnego, gdzie:

$$\begin{bmatrix} a_{N-1} \\ a_{N-2} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} n_1^{N-1} & \dots & 1 \\ \vdots & \ddots & \vdots \\ n_N^{N-1} & \dots & 1 \end{bmatrix}^{-1} \begin{bmatrix} s_j^{(e)}(n_1) \\ s_j^{(e)}(n_2) \\ \vdots \\ s_j^{(e)}(n_N) \end{bmatrix} = \mathbf{B}_{N \times N}^{-1} \begin{bmatrix} s_j^{(e)}(n_1) \\ s_j^{(e)}(n_2) \\ \vdots \\ s_j^{(e)}(n_N) \end{bmatrix}. \quad (13.41)$$

Aby otrzymać zinterpolowaną wartość  $s_j(n)$  dla próbki nieparzystej leżącej pomiędzy próbkami parzystymi oblicza się ją na podstawie równania

$$s_j(n) = \begin{bmatrix} n^{N-1} & n^{N-2} & \dots & 1 \end{bmatrix} \begin{bmatrix} a_{N-1} \\ a_{N-2} \\ \vdots \\ a_0 \end{bmatrix}. \quad (13.42)$$

Podstawiając równanie 13.41 do równania 13.42 otrzymuje się wzór na interpolacyjną wersję bloku predykcji w schemacie liftingu [72]

$$s_j(n) = \underbrace{\begin{bmatrix} n^{N-1} & \dots & 1 \\ \vdots & \ddots & \vdots \\ n_N^{N-1} & \dots & 1 \end{bmatrix}^{-1}}_{\mathbf{P}} \begin{bmatrix} s_j^{(e)}(n_1) \\ s_j^{(e)}(n_2) \\ \vdots \\ s_j^{(e)}(n_N) \end{bmatrix} \quad (13.43)$$

lub równoważnie

$$s_j(n) = \mathbf{p} \bullet \begin{bmatrix} s_j^{(e)}(n_1) \\ s_j^{(e)}(n_2) \\ \vdots \\ s_j^{(e)}(n_N) \end{bmatrix} \quad (13.44)$$

gdzie

$$\mathbf{p} = [ n^{N-1} \quad n^{N-2} \quad \dots \quad 1 ] \begin{bmatrix} n_1^{N-1} & \dots & 1 \\ \vdots & \ddots & \vdots \\ n_N^{N-1} & \dots & 1 \end{bmatrix}^{-1}. \quad (13.45)$$

Wyznaczone równanie jednoznacznie potwierdza, że dla interpolacji wielomianowej wyznaczona wartość nowej próbki nieparzystej na podstawie próbek parzystych jest średnią ważoną wartości próbek parzystych z wagami wektora  $\mathbf{p}$ .

## 13.8 Implementacja interpolacji funkcji z wykorzystaniem schematu liftingu

Biortogonalna transformata falkowa w swojej interpolacyjnej wersji pozwala na uzyskanie odpowiedniej gładkości charakterystyki częstotliwościowej w przypadku gdy funkcja  $s_j$  jest wielomianem stopnia  $N - 1$ . Określa ona wtedy liczbę znikających momentów które są równe liczbie rzędu predyktora.

Właściwość taka zwana *wielomianowym usuwaniem* jest gwarancją dokładności predykcji, gdzie  $d_{j-1} = 0$ .

## 13.8.1 Predyktor rzędu I - Haara

```
<< SchematLiftinguZAproksymacjaMNK`
(* Rząd musi być conajwyżej 2 razy mniejszy od długości tablicy danych *)
rzad = 1;
tablicaDanych = {1, 1, 2, 2, 4, 4, 3, 3};

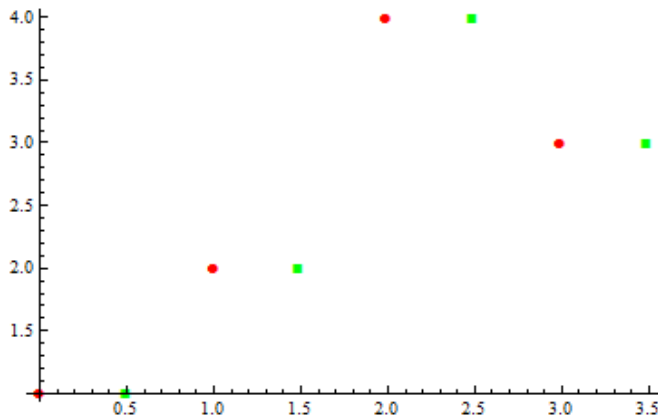
SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "predykcja"]

p =  $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ 

SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "detal"]

d =  $\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$ 

SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "wykres"]
Interpolacja wielomianowa sygnału z predyktorem rzędu 1
```



Rysunek 13.8: Interpolacja wielomianowa - predyktor rzędu I

Przekształcenie Haara jest to jedyny przykład ortogonalnej transformacji falkowej którą można odtworzyć za pomocą schematu pojedynczej predykcji i uaktualnienia. Interpolacje kolejnych rzędów doprowadzają do powstania biortogonalnych systemów falkowych [66, 59].

Funkcja przedstawiona w postaci  $\circ$  reprezentuje próbki parzyste, natomiast próbki o symbolu  $\square$  są próbkami nieparzystymi szacowanymi podczas operacji predykcji.

Macierz jednokolumnowa  $p$  oznacza wektor wag predykcji dla kolejnych przewidywań próbek, gdzie dla każdej operacji wyliczania próbki nieparzystej jednoelementowy predyktor ma wartość  $p_0 = 1$  (podrozdział 13.6).

W sytuacji gdy funkcja interpolowana jest funkcją Haara, wtedy to różnica  $d$  (współczynniki detalu) pomiędzy wartością próbki nieparzystej oryginalnej a wartością szacowanej w bloku predykcji wynosi 0, co wskazuje na zerowy błąd predykcji i perfekcyjną dokładność (rys. 13.8).

### 13.8.2 Predyktor rzędu II - funkcja liniowa

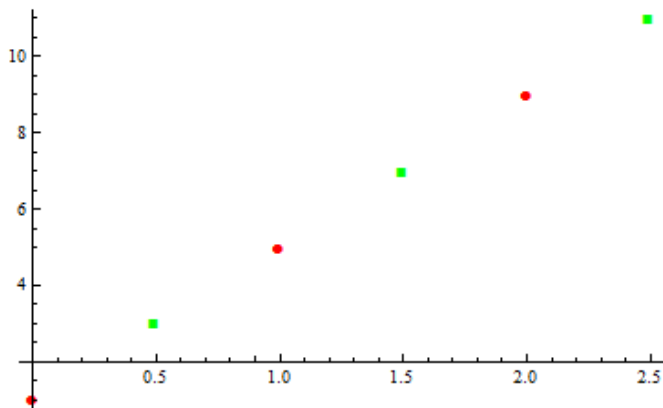
Innym przykładem predykcji jest predyktor rzędu II, który wykorzystuje interpolację liniową względem funkcji badanej.

```
<< SchematLiftinguZApryksymacjaMNK`
(* Rząd musi być conajwyżej 2 razy mniejszy od długości tablicy danych *)
rzad = 2;
tablicaDanych = Table[2 x - 1, {x, 1, 6}];

SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "predykacja"]
p =  $\begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \\ -0.5 & 1.5 \end{pmatrix}$ 

SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "deta"]
d =  $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ 

SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "wykres"]
Interpolacja wielomianowa sygnału z predyktorem rzędu 2
```



Rysunek 13.9: Interpolacja wielomianowa - predyktor rzędu II

Do analizy można wykorzystać go w sytuacji, gdy próbka nieparzysta jest rozmieszczona na odcinkach które powstają poprzez połączenie sąsiadujących z nią próbek parzystych. Wtedy to można zagwarantować maksymalną dokładność, a co się z tym wiąże minimalny błąd predykcji.

Wektor predykcji w tej sytuacji (podrozdział 13.6) wynosi  $\mathbf{p} = [0.5 \ 0.5]$  (rys. 13.7b).

Obliczenia dla przypadku granicznego tj. dla ostatniej próbki nieparzystej szacowane są z wagą  $\mathbf{p} = [-1.5 \ -0.5]$ , ponieważ do jej obliczenia nie są brane wyłącznie sąsiednie próbki parzyste.

Próbka nieparzysta jest próbką ostatnią, zatem dla potrzeb estymacji wykorzystywane są dwie próbki parzyste znajdujące się w najbliższym otoczeniu, czyli po jej lewej stronie (13.7b, kolor czerwony).

### 13.8.3 Predyktor rzędu IV - wielomian stopnia trzeciego

Predyktor rzędu IV który interpoluje układ za pomocą wielomianu stopnia trzeciego. W analizie predykcyjnej stosuje się predyktory rzędów parzystych (za wyjątkiem rzędu I), ze względu na symetryczny rozkład sąsiednich próbek parzystych po obu stronach próbek nieparzystych.

Rząd predyktora jest zawsze o jeden większy od rzędu wielomianu interpolującego ponieważ do interpolacji wielomianowej za pomocą schematu liftingu z predyktorem rzędu  $n + 1$  wykorzystuje się wielomian stopnia  $n$ .

Z analizy wynika, że wektor predykcji, gdzie  $\mathbf{p} = [-0.0625 \ 0.5625 \ 0.5625 \ -0.0625]$ , jest używany w sytuacji gdy do wyznaczania próbki nieparzystej która w swoim sąsiedztwie posiada symetrycznie po dwie próbki nieparzyste z każdej ze stron.

Predyktor  $\mathbf{p} = [0.3125 \ 0.9375 \ -0.3125 \ 0.0625]$  użyty jest (rys.13.7c, kolor zielony) tylko raz na początku, gdzie do wyznaczania próbki stosuje się jedyną próbkę parzystą po jej lewej stronie i trzy po prawej.

Wektor  $\mathbf{p} = [0.0625 \ -0.03125 \ 0.9375 \ 0.3125]$  natomiast, służy do oszacowania przedostatniej próbki nieparzystej z użyciem trzech próbek parzystych po lewej stronie i jednej po stronie prawej (rys.13.7c, kolor niebieski).

W sytuacji skrajnej, dla ostatniej próbki, stosuje się predyktor z wagami  $\mathbf{p} = [-0.3125 \ 1.3125 \ -2.1875 \ 2.1875]$ , gdzie w relacje z predyktorem wchodzi wartości najbliższych czterech próbek parzystych leżących po lewej stronie próbki nieparzystej (rys.13.7c, kolor czerwony).

```

<< SchematLiftinguZAproksymacjaMNK`

(* Rząd musi być conajwyżej 2 razy mniejszy od długości tablicy danych *)

rzad = 4;
tablicaDanych = Table[2 x^2 - x - 1, {x, 1, 12}];

SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "predykcja"]

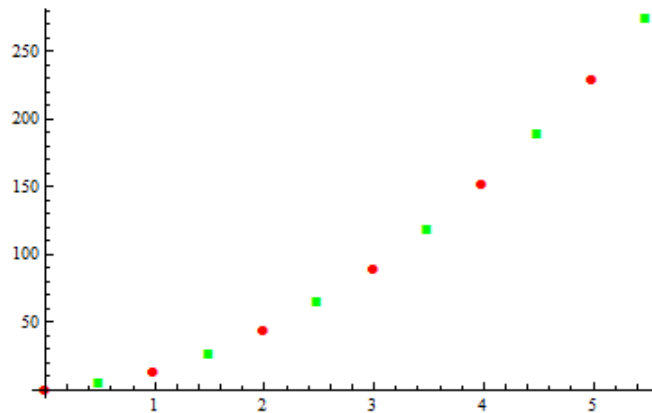
P = 
$$\begin{pmatrix} 0.3125 & 0.9375 & -0.3125 & 0.0625 \\ -0.0625 & 0.5625 & 0.5625 & -0.0625 \\ -0.0625 & 0.5625 & 0.5625 & -0.0625 \\ -0.0625 & 0.5625 & 0.5625 & -0.0625 \\ 0.0625 & -0.3125 & 0.9375 & 0.3125 \\ -0.3125 & 1.3125 & -2.1875 & 2.1875 \end{pmatrix}$$


SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "detal"]

d = 
$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$


SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "wykres"]
Interpolacja wielomianowa sygnału z predyktorem rzędu 4

```



Rysunek 13.10: Interpolacja wielomianowa - predyktor rzędu IV

#### 13.8.4 Predyktor rzędu IV - interpolacja funkcji $\sin(x)$ wielomianem stopnia trzeciego

Podczas analizy funkcji sinusoidalnej można zauważyć, że wektor predykcji interpolacji wielomianowej jest niezależny od funkcji interpolowanej. Współczynniki predykcji dla danego rzędu są niezmiennicze, niezależne od funkcji badanej.

```
<< SchematLiftinguZAproksymacjaMKN`
(* Rząd musi być conajwyżej 2 razy mniejszy od długości tablicy danych *)
rzad = 4;
tablicaDanych = Table[Sin[x], {x, 1, 12}];

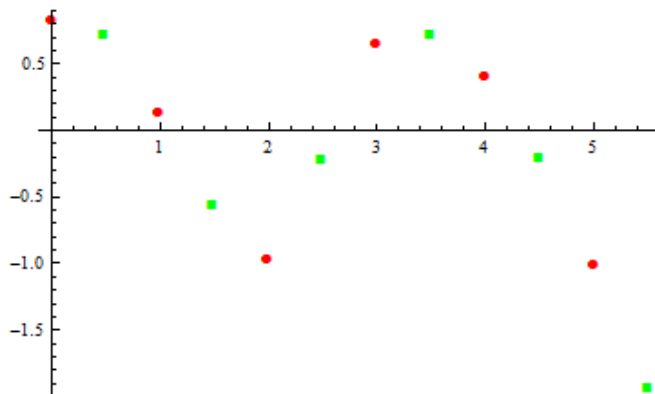
SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "predykcja"]

p = 
$$\begin{pmatrix} 0.3125 & 0.9375 & -0.3125 & 0.0625 \\ -0.0625 & 0.5625 & 0.5625 & -0.0625 \\ -0.0625 & 0.5625 & 0.5625 & -0.0625 \\ -0.0625 & 0.5625 & 0.5625 & -0.0625 \\ 0.0625 & -0.3125 & 0.9375 & 0.3125 \\ -0.3125 & 1.3125 & -2.1875 & 2.1875 \end{pmatrix}$$


SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "detal"]

d = 
$$\begin{pmatrix} 0.173312 \\ -0.203134 \\ -0.0749981 \\ 0.265554 \\ -0.352644 \\ 1.39046 \end{pmatrix}$$


SchematLiftinguInterpolacjaFunkcji[tablicaDanych, rzad, "wykres"]
Interpolacja wielomianowa sygnału z predyktorem rzędu 4
```



Rysunek 13.11: Interpolacja wielomianowa - funkcja  $\sin(x)$

Badając powyższą funkcję trygonometryczną uzyskuje się niezerowy błąd predykcji. Wynika to z tego to, że funkcja stopnia trzeciego nie jest optymalna dla badanej funkcji. Ostatnia duża wartość współczynnika błędu predykcji spowodowana jest tym, że do oszacowania ostatniej próbki, należącej do funkcji trygonometrycznej stosuje się interpolację wielomianową stopnia trzeciego z wykorzystaniem czterech próbek parzystych leżących po lewej stronie próbki nieparzystej.

Prowadzi to do błędnej estymacji próbki otrzymując w konsekwencji duży błąd predykcji.

Przypadek ten obrazuje nieoptymalne dopasowanie predyktora do badanej funkcji.

### 13.9 Wyznaczanie współczynników uaktualnienia dla funkcji periodycznych

Schemat liftingu w implementacji zaproponowanej przez Wima Sweldensa oprócz jednego bloku predykcji, posiada również dodatkowy blok uaktualnienia, odpowiedzialny za to aby wartość średnia funkcji oryginalnej była równa średniej wartości próbek reprezentacji zgrubnej, gdzie

$$\begin{aligned}
 \bar{s}_i &= \bar{s}_{i-1} \\
 \frac{1}{N} \sum_{m=0}^{N-1} s_i(m) &= \frac{2}{N} \sum_{n=0}^{\frac{N}{2}-1} s_{i-1}(n) \\
 \frac{1}{N} \sum_{n=0}^{\frac{N}{2}-1} (s_i(2n) + s_i(2n+1)) &= \frac{2}{N} \sum_{n=0}^{\frac{N}{2}-1} s_{i-1}(n) \\
 \frac{1}{N} \sum_{n=0}^{\frac{N}{2}-1} (s_i^{(e)}(n) + s_i^{(o)}(n)) &= \frac{2}{N} \sum_{n=0}^{\frac{N}{2}-1} s_{i-1}(n) \\
 \sum_{n=0}^{\frac{N}{2}-1} (s_i^{(e)}(n) + s_i^{(o)}(n)) &= 2 \sum_{n=0}^{\frac{N}{2}-1} s_{i-1}(n)
 \end{aligned} \tag{13.46}$$

Wyznaczając wektor współczynników uaktualnienia przyjmuje się, że wartości współczynników detalu  $d_{i-1}$  i współczynników aproksymacji  $s_{i-1}$  na  $i-1$ -szym poziomie rozdzielczości, zgodnie z zasadą liftingu określa się wzorem [42, 50]

$$\begin{aligned}
 d_{i-1}(n-1) &= s_i^{(o)}(n-1) - P\{s_i^{(e)}(n-k)\} \\
 s_{i-1}(n) &= s_i^{(e)}(n) + U\{d_{i-1}(n-l)\}
 \end{aligned} \tag{13.47}$$

gdzie indeks  $i$  określa poziom analizy wielorozdzielczej, zaś  $k$  i  $l$  są indeksami kolejnych próbek wchodzących w skład operacji predykcji  $P$  i uaktualnienia  $U$ .

Korzystając z zależności liniowej współczynników parzystych od nieparzystych oblicza się wyrażenia w sposób następujący

$$\begin{aligned}
 d_{i-1}(n-1) &= s_i^{(o)}(n-1) - \sum_{k=0}^{\dim \vec{P}-1} [p_k s_i^{(e)}(n-k)] \\
 s_{i-1}(n) &= s_i^{(e)}(n) + \sum_{l=0}^{\dim \vec{U}-1} [u_l d_{i-1}(n-l)] \\
 &= s_i^{(e)}(n) + \sum_{l=0}^{\dim \vec{U}-1} [u_l (s_i^{(o)}(n-l) + \\
 &\quad - \sum_{k=0}^{\dim \vec{P}-1} p_k s_i^{(e)}(n-k-l+1))]
 \end{aligned} \tag{13.48}$$

gdzie  $\dim \vec{P}$  i  $\dim \vec{U}$  są wymiarami wektorów współczynników predykcji i uaktualnienia. Podstawiając wzór 13.48 do 13.46 otrzymuje się równanie



$$\begin{aligned}
2 \sum_{n=0}^{\frac{N}{2}-1} s_{i-1}(n) &= 2 \sum_{n=0}^{\frac{N}{2}-1} [s_i^{(e)}(n) + \sum_{l=0}^{\dim \vec{U}-1} [u_l [s_i^{(o)}(n-l) + \\
&\quad - \sum_{k=0}^{\dim \vec{P}-1} p_k s_i^{(e)}(n-k-l+1)]]] \\
&= 2 \sum_{n=0}^{\frac{N}{2}-1} [s_i^{(e)}(n) + \sum_{l=0}^{\dim \vec{U}-1} u_l s_i^{(o)}(n-l) + \\
&\quad - \sum_{l=0}^{\dim \vec{U}-1} u_l \sum_{k=0}^{\dim \vec{P}-1} p_k s_i^{(e)}(n-k-l+1)] \tag{13.49} \\
&= 2 \sum_{n=0}^{\frac{N}{2}-1} s_i^{(e)}(n) \\
&\quad - 2 \sum_{l=0}^{\dim \vec{U}-1} u_l \sum_{k=0}^{\dim \vec{P}-1} p_k \sum_{n=0}^{\frac{N}{2}-1} s_i^{(e)}(n-k-l+1) + \\
&\quad + 2 \sum_{l=0}^{\dim \vec{U}-1} u_l \sum_{n=0}^{\frac{N}{2}-1} s_i^{(o)}(n-l)
\end{aligned}$$

Przyjmując okresowy charakter funkcji dla warunków granicznych, otrzymuje się zależności

$$\begin{aligned}
\sum_{n=0}^{\frac{N}{2}-1} s_i^{(e)}(n) &= \sum_{n=0}^{\frac{N}{2}-1} s_i^{(e)}(n-k-l+1) \\
\sum_{n=0}^{\frac{N}{2}-1} s_i^{(o)}(n) &= \sum_{n=0}^{\frac{N}{2}-1} s_i^{(o)}(n-l)
\end{aligned} \tag{13.50}$$

co wynika z faktu, że

$$\begin{aligned}
s_i^{(e)}(n-k-l+1) &= s_i^{(e)}((n-k-l+1) \text{ modulo } \frac{N}{2}) \\
s_i^{(o)}(n-l) &= s_i^{(o)}((n-l) \text{ modulo } \frac{N}{2})
\end{aligned} \tag{13.51}$$

W konsekwencji równanie upraszcza się do postaci

$$\begin{aligned}
2 \sum_{n=0}^{\frac{N}{2}-1} s_{i-1}(n) &= 2 \sum_{n=0}^{\frac{N}{2}-1} s_i^{(e)}(n) - 2 \sum_{l=0}^{\dim \vec{U}-1} u_l \sum_{k=0}^{\dim \vec{P}-1} p_k \sum_{n=0}^{\frac{N}{2}-1} s_i^{(e)}(n) \\
&\quad + 2 \sum_{l=0}^{\dim \vec{U}-1} u_l \sum_{n=0}^{\frac{N}{2}-1} s_i^{(o)}(n) \\
&= \sum_{n=0}^{\frac{N}{2}-1} [(2 - 2 \sum_{l=0}^{\dim \vec{U}-1} u_l \sum_{k=0}^{\dim \vec{P}-1} p_k) s_i^{(e)}(n) + (2 \sum_{l=0}^{\dim \vec{U}-1} u_l) s_i^{(o)}(n)]
\end{aligned}$$

Wykorzystując równość średniej wartości próbek na każdym poziomie rozdzielczości można wykazać relacje

$$\begin{aligned} \sum_{n=0}^{N-1} s_i(n) &= 2 \sum_{n=0}^{\frac{N}{2}-1} s_{i-1}(n) \\ \sum_{n=0}^{\frac{N}{2}-1} (s_i^{(e)}(n) + s_i^{(o)}(n)) &= 2 \sum_{n=0}^{\frac{N}{2}-1} s_{i-1}(n) \\ \sum_{n=0}^{\frac{N}{2}-1} (s_i^{(e)}(n) + s_i^{(o)}(n)) &= \sum_{n=0}^{\frac{N}{2}-1} \left[ \underbrace{\left( 2 - 2 \sum_{l=0}^{\dim \vec{U}-1} u_l \sum_{k=0}^{\dim \vec{P}-1} p_k \right)}_1 s_i^{(e)}(n) + \underbrace{\left( 2 \sum_{l=0}^{\dim \vec{U}-1} u_l \right)}_1 s_i^{(o)}(n) \right] \end{aligned}$$

Porównując lewą i prawą stronę wyrażenia wraz z wartościami współczynników przy  $s_i^{(e)}$  i  $s_i^{(o)}$  powyższy wzór wyznacza regułę

$$\begin{cases} 2 - 2 \sum_{l=0}^{\dim \vec{U}-1} u_l \sum_{k=0}^{\dim \vec{P}-1} p_k = 1 \\ 2 \sum_{l=0}^{\dim \vec{U}-1} u_l = 1 \end{cases} \quad (13.52)$$

co w konsekwencji doprowadza do ogólnej własności współczynników wektorów predykcji i uaktualnienia dla funkcji periodycznych

$$\begin{cases} \sum_{k=0}^{\dim \vec{P}-1} p_k = 1 \\ \sum_{l=0}^{\dim \vec{U}-1} u_l = \frac{1}{2} \end{cases} \quad (13.53)$$

Przykładem zależności 13.53 mogą być wyniki uzyskane w podrozdziale 13.8.3 dla predyktora rzędu IV gdzie

$$\vec{P} = \{-0.0625, 0.5625, 0.5625, -0.0625\} \quad (13.54)$$

Zgodnie z własnością predyktora suma jego współczynników wynosi 1.

Ponieważ suma współczynników uaktualnienia wynosi  $\frac{1}{2}$ , zatem dla tej samej długości wektorów  $\vec{P}$  i  $\vec{U}$  można przyjąć  $\vec{U} = \frac{\vec{P}}{2}$ , czyli

$$\vec{U} = \{-0.03125, 0.28125, 0.28125, -0.03125\}. \quad (13.55)$$

Weryfikując wzór 13.53 wnioskuje się, że długość wektora uaktualnienia może być inna aniżeli długość wektora predykcji, ważne aby była zachowana wartość sumy współczynników. Powyższy wektor uaktualnienia  $\vec{U}$  może być wykorzystywany zatem na przykład dla predyktorów innych rzędów. Tak więc, jego wybór może być uzależniony jedynie od charakteru funkcji aproksymowanej.

W przypadku gdy operacja predykcji dla badanej funkcji potrafi wyliczyć nowe próbki nieparzyste w sposób idealnie pokrywający się z oryginalnymi, wtedy to zmiana wektora  $\vec{U}$  nie ma najmniejszego

wpływu na zmianę charakteru funkcji aproksymacji zgrubnej. Ponieważ przestrzeń detalu jest w takiej sytuacji funkcją zerową, zatem operacja uaktualnienia znika i schemat liftingu jest oparty jedynie na operacji predykcji.

## Część V

Schemat liftingu z wykorzystaniem  
aproksymacji metodą najmniejszych  
kwadratów w bloku predykcji



# Aproksymacja funkcji metodą najmniejszych kwadratów w bloku predykcji

---

Aproksymacja funkcji jest procesem który pozwala na uzyskanie rozwiązań przybliżonych na podstawie funkcji znanych, dzięki którym można uzyskać rozwiązanie bliskie rozwiązaniu dokładnemu, uzyskując przy tym pewną założoną regularność. Stosuje się ją najczęściej, gdy nie można odnaleźć funkcji analitycznej która wyznaczałaby precyzyjnie przebieg zmienności funkcji.

Jednym z typów aproksymacji jest aproksymacja metodą najmniejszych kwadratów (MNK) która pozwala na przybliżenie funkcji aproksymowanych za pomocą funkcji będącą kombinacją liniową funkcji bazowych.

Przybliżenie funkcji inną funkcją może powodować pojawienie się błędów zwanych błędami aproksymacji. Przyczyną tego jest m. in. brak wymagań, aby funkcja aproksymująca przechodziła przez jakikolwiek punkt należący do aproksymowanej funkcji.

Ponieważ funkcja aproksymująca może wygładzać funkcję aproksymowaną, zatem jej gładkość przekłada się także na jej różniczkowalność.

W rozdziale tym, jako wartość dodana, został zaproponowany autorski pomysł realizacji schematu liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów.

Pozwala to na szacowanie nieparzystych próbek funkcji w bloku predykcji za pomocą wielomianów stopni znacznie niższych aniżeli stopień wielomianu interpolacyjnego opisanego przez Wima Sweldensa.

Takie rozwiązanie nie tylko umożliwia przybliżanie funkcji za pomocą wielomianów algebraicznych, lecz także za pomocą dowolnych funkcji bazowych.

Przybliżenie funkcji w sensie aproksymacji MNK w bloku predykcji rozumiane jest jako minimalizacja pewnej funkcji błędu  $f_j$  na  $j$ -tym poziomie rozdzielczości zwaną funkcją celu

$$f_j = \sqrt{\sum_{i=1}^N w(n_i) [s_j^{(e)}(n_i) - \sum_{k=1}^M a_{M-k} \Psi_{M-k}(n_i)]^2} \rightarrow \min \quad (14.1)$$

gdzie  $w(\cdot)$  jest funkcją wagową ( $w(n_i) \geq 0$ ),  $s_j^{(e)}(\cdot)$  - próbkami parzystymi funkcji aproksymowanej na  $j$ -tym poziomie rozdzielczości, zaś  $\sum_k a_k \Psi_k(\cdot)$  jest liniową kombinacją funkcji bazowych  $\Psi(\cdot)$  [27, 28].

Minimalizacja funkcji celu odbywa się na zasadzie obliczenia układu równań powstałego z wyznaczenia kolejnych pochodnych wyrażenia podpierwiastkowego funkcji 14.1 po wszystkich współczynnikach  $a_j$ .

Wyliczone pochodne przyrównuje się do zera uzyskując układ  $M$  równań z  $M$  niewiadomymi, gdzie

$$\begin{aligned} \frac{\delta f}{\delta a_{M-1}} &= -2 \sum_{i=1}^N w(n_i) [s_j^{(e)}(n_i) - \sum_{k=1}^M a_{M-k} \Psi_{M-k}(n_i)] \Psi_{M-1}(n_i) = 0 \\ \frac{\delta f}{\delta a_{M-2}} &= -2 \sum_{i=1}^N w(n_i) [s_j^{(e)}(n_i) - \sum_{k=1}^M a_{M-k} \Psi_{M-k}(n_i)] \Psi_{M-2}(n_i) = 0 \\ &\vdots \\ \frac{\delta f}{\delta a_0} &= -2 \sum_{i=1}^N w(n_i) [s_j^{(e)}(n_i) - \sum_{k=1}^M a_{M-k} \Psi_{M-k}(n_i)] \Psi_0(n_i) = 0 \end{aligned} \quad (14.2)$$

co w konsekwencji prowadzi do postaci

$$\begin{aligned} \sum_{k=1}^M a_{M-k} \sum_{i=1}^N w(n_i) \Psi_{M-k}(n_i) \Psi_{M-1}(n_i) &= \sum_{i=1}^N w(n_i) s_j^{(e)}(n_i) \Psi_{M-1}(n_i) \\ \sum_{k=1}^M a_{M-k} \sum_{i=1}^N w(n_i) \Psi_{M-k}(n_i) \Psi_{M-2}(n_i) &= \sum_{i=1}^N w(n_i) s_j^{(e)}(n_i) \Psi_{M-2}(n_i) \\ &\vdots \\ \sum_{k=1}^M a_{M-k} \sum_{i=1}^N w(n_i) \Psi_{M-k}(n_i) \Psi_0(n_i) &= \sum_{i=1}^N w(n_i) s_j^{(e)}(n_i) \Psi_0(n_i) \end{aligned} \quad (14.3)$$

Powyższy zapis można przedstawić w postaci macierzowej

$$\mathbf{B}_{M \times M} \begin{bmatrix} a_{M-1} \\ a_{M-2} \\ \vdots \\ a_0 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N w(n_i) s_j^{(e)}(n_i) \Psi_{M-1}(n_i) \\ \sum_{i=1}^N w(n_i) s_j^{(e)}(n_i) \Psi_{M-2}(n_i) \\ \vdots \\ \sum_{i=1}^N w(n_i) s_j^{(e)}(n_i) \Psi_0(n_i) \end{bmatrix} \quad (14.4)$$

gdzie

$$\mathbf{B}_{M \times M} = \begin{bmatrix} \sum_{i=1}^N w(n_i) \Psi_{M-1}(n_i) \Psi_{M-1}(n_i) & \cdots & \sum_{i=1}^N w(n_i) \Psi_0(n_i) \Psi_{M-1}(n_i) \\ \sum_{i=1}^N w(n_i) \Psi_{M-1}(n_i) \Psi_{M-2}(n_i) & \cdots & \sum_{i=1}^N w(n_i) \Psi_0(n_i) \Psi_{M-2}(n_i) \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^N w(n_i) \Psi_{M-1}(n_i) \Psi_0(n_i) & \cdots & \sum_{i=1}^N w(n_i) \Psi_0(n_i) \Psi_0(n_i) \end{bmatrix}. \quad (14.5)$$

Rozdzielając prawostronne sumy wyrażenia 14.4 otrzymuje się układ postaci

$$\mathbf{B}_{M \times M} \cdot \begin{bmatrix} a_{M-1} \\ a_{M-2} \\ \vdots \\ a_0 \end{bmatrix} = \underbrace{\Psi_{M \times N} \cdot \mathbf{W}_{N \times N}}_{\mathbf{W}_{N \times N}} \cdot \begin{bmatrix} s_j^{(e)}(n_1) \\ s_j^{(e)}(n_2) \\ \vdots \\ s_j^{(e)}(n_N) \end{bmatrix} \quad (14.6)$$

gdzie

$$\Psi_{M \times N} = \begin{bmatrix} \Psi_{M-1}(n_1) & \Psi_{M-1}(n_2) & \cdots & \Psi_{M-1}(n_N) \\ \Psi_{M-2}(n_1) & \Psi_{M-2}(n_2) & & \Psi_{M-2}(n_N) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0(n_1) & \Psi_0(n_2) & \cdots & \Psi_0(n_N) \end{bmatrix}. \quad (14.7)$$

Postępując analogicznie jak w przypadku interpolacji (13.41), wektor współczynników kombinacji liniowej funkcji bazowej przedstawia się następująco

$$\begin{bmatrix} a_{M-1} \\ a_{M-2} \\ \vdots \\ a_0 \end{bmatrix} = \mathbf{B}_{M \times M}^{-1} \cdot \Psi_{M \times N} \cdot \mathbf{W}_{N \times N} \cdot \begin{bmatrix} s_j^{(e)}(n_1) \\ s_j^{(e)}(n_2) \\ \vdots \\ s_j^{(e)}(n_N) \end{bmatrix}. \quad (14.8)$$

Ponieważ szukaną, aproksymowaną wartość funkcji  $s_j^{(e)}$  w dowolnym punkcie  $n$  leżącym pomiędzy węzłami  $n_1, n_2, \dots, n_N$  oblicza się na podstawie



$$s_j^{(e)}(n) = \begin{bmatrix} \Psi_{M-1}(n) & \Psi_{M-2}(n) & \cdots & \Psi_0(n) \end{bmatrix} \begin{bmatrix} a_{M-1} \\ a_{M-2} \\ \vdots \\ a_0 \end{bmatrix} \quad (14.9)$$

tak więc dzięki zależności 14.8 predykcyjna postać aproksymacji funkcji wyraża się wzorem

$$s_j(n) = \begin{bmatrix} \Psi_{M-1}(n) & \cdots & \Psi_0(n) \end{bmatrix} \cdot \mathbf{B}_{M \times M}^{-1} \cdot \Psi_{M \times N} \cdot \mathbf{W}_{N \times N} \cdot \begin{bmatrix} s_j^{(e)}(n_1) \\ s_j^{(e)}(n_2) \\ \vdots \\ s_j^{(e)}(n_N) \end{bmatrix}. \quad (14.10)$$

lub w postaci predykcyjnej

$$s_j(n) = \mathbf{p} \bullet \begin{bmatrix} s_j^{(e)}(n_1) \\ s_j^{(e)}(n_2) \\ \vdots \\ s_j^{(e)}(n_N) \end{bmatrix} \quad (14.11)$$

gdzie

$$\mathbf{p} = \begin{bmatrix} \Psi_{M-1}(n) & \Psi_{M-2}(n) & \cdots & \Psi_0(n) \end{bmatrix} \cdot \mathbf{B}_{M \times M}^{-1} \cdot \Psi_{M \times N} \cdot \mathbf{W}_{N \times N} \quad (14.12)$$

## 14.1 Wektor predykcji określany za pomocą aproksymacji wielomianowej funkcji

Aproksymacja wielomianowa funkcji formułowana dla funkcji określonej na dyskretnym zbiorze argumentów jest aproksymacją metodą najmniejszych kwadratów, gdzie funkcje bazowe wynoszą odpowiednio  $\Psi_i(n) = n^i$  natomiast wagi  $w_i(n) = 1$  ( $\mathbf{W}_{N \times N} = \mathbf{I}_{N \times N}$ ).

Wtedy to

$$\mathbf{p} = [ n^{M-1} \quad n^{M-2} \quad \dots \quad 1 ] \cdot \mathbf{B}_{M \times M}^{-1} \cdot \Psi_{M \times N} \quad (14.13)$$

gdzie

$$\mathbf{B}_{M \times M}^{-1} \cdot \Psi_{M \times N} = \begin{bmatrix} \sum_{i=1}^N n_i^{2M-2} & \dots & \sum_{i=1}^N n_i^{M-1} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^N n_i^{M-1} & \dots & \sum_{i=1}^N 1 \end{bmatrix}^{-1} \begin{bmatrix} n_1^{M-1} & \dots & n_N^{M-1} \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} \quad (14.14)$$

## 14.2 Wektor predykcji określany za pomocą interpolacji wielomianowej funkcji

Jeżeli liczba funkcji bazowych  $M$  jest równa liczbie próbek parzystych  $N$  wykorzystywanych do aproksymacji ( $M = N$ ), wtedy to predyktor przyjmuje postać

$$\mathbf{p} = [ n^{N-1} \quad n^{N-2} \quad \dots \quad 1 ] \cdot \mathbf{B}_{N \times N}^{-1} \cdot \Psi_{N \times N} \quad (14.15)$$

Analizując iloczyn macierzy kwadratowych

$$\mathbf{B}_{N \times N}^{-1} \cdot \Psi_{N \times N} = \begin{bmatrix} \sum_{i=1}^N n_i^{2N-2} & \dots & \sum_{i=1}^N n_i^{N-1} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^N n_i^{N-1} & \dots & \sum_{i=1}^N 1 \end{bmatrix}^{-1} \begin{bmatrix} n_1^{N-1} & \dots & n_N^{N-1} \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} \quad (14.16)$$

wykorzystując dodatkowo własności przekształceń macierzy odwrotnych  $[AB]^{-1} = B^{-1}A^{-1}$  dla macierzy  $\mathbf{B}_{N \times N}^{-1}$  takich, że

$$\begin{aligned}
 \mathbf{B}_{N \times N}^{-1} &= \begin{bmatrix} \sum_{i=1}^N n_i^{2N-2} & \sum_{i=1}^N n_i^{2N-3} & \cdots & \sum_{i=1}^N n_i^{N-1} \\ \sum_{i=1}^N n_i^{2N-3} & \sum_{i=1}^N n_i^{2N-4} & \cdots & \sum_{i=1}^N n_i^{N-2} \\ \vdots & \ddots & \ddots & \vdots \\ \sum_{i=1}^N n_i^{N-1} & \sum_{i=1}^N n_i^{N-2} & \cdots & \sum_{i=1}^N 1 \end{bmatrix}^{-1} = \\
 &= \begin{bmatrix} n_1^{N-1} & n_2^{N-1} & \cdots & n_N^{N-1} \\ n_1^{N-2} & n_2^{N-2} & \cdots & n_N^{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} n_1^{N-1} & n_1^{N-2} & \cdots & 1 \\ n_2^{N-1} & n_2^{N-2} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ n_N^{N-1} & n_N^{N-2} & \cdots & 1 \end{bmatrix}^{-1} = \\
 &= \begin{bmatrix} n_1^{N-1} & n_1^{N-2} & \cdots & 1 \\ n_2^{N-1} & n_2^{N-2} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ n_N^{N-1} & n_N^{N-2} & \cdots & 1 \end{bmatrix}^{-1} \begin{bmatrix} n_1^{N-1} & n_2^{N-1} & \cdots & n_N^{N-1} \\ n_1^{N-2} & n_2^{N-2} & \cdots & n_N^{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}^{-1}.
 \end{aligned} \tag{14.17}$$

Po redukcji otrzymuje się uproszczoną formę iloczynu macierzy, gdzie

$$\mathbf{B}_{N \times N}^{-1} \cdot \Psi_{N \times N} = \begin{bmatrix} n_1^{N-1} & n_1^{N-2} & \cdots & 1 \\ n_2^{N-1} & n_2^{N-2} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ n_N^{N-1} & n_N^{N-2} & \cdots & 1 \end{bmatrix}^{-1} \tag{14.18}$$

Powyższy predyktor aproksymacji jest dokładnym odwzorowaniem predyktora interpolacyjnego otrzymanym w przekształceniu 13.45[61, 27, 28].

### 14.3 Problem złego uwarunkowania macierzy współczynników układu

W analizie sygnałów cyfrowych wartości na zbiorze dyskretnym są często ustalane w sposób empiryczny, obarczone błędami aproksymacji - wielomian aproksymujący powinien zatem posiadać stopień na tyle niski, aby wygładzał losowe błędy wynikające na przykład z pomiarów.

Aproksymacja metodą najmniejszych kwadratów w bloku predykcji odbywa się na zasadzie wyznaczenia minimalnej wartości funkcji celu  $f_j$  na  $j$ -tym poziomie rozdzielczości wykorzystując wartości odwrotnej macierzy  $\mathbf{B}_{M \times M}^{-1}$ .

Postępowanie takie posiada istotną wadę, które może prowadzić do tak zwanego złego uwarunkowania macierzy współczynników układu  $\mathbf{B}_{M \times M}$  ze wzoru 14.5.

Aby zobrazować problem złego uwarunkowania macierzy można przyjąć, że współczynniki  $\mathbf{B}_{M \times M}$  posiadają argumenty leżące równomiernie na przedziale  $\langle 0, 1 \rangle$ . Współczynniki te dla dużych wartości  $N$  można w przybliżeniu wyliczyć za pomocą rachunku całkowego tj.

$$\sum_{i=1}^N n_i^k \approx N \int_0^1 n^k dn = N \frac{n^{k+1}}{k+1} \Big|_0^1 = \frac{N}{k+1} \quad (14.19)$$

gdzie  $k \in \langle 0, 2M - 2 \rangle$ .

Tak więc, postać macierzy współczynników układu  $\mathbf{B}_{M \times M}$  wyraża się w postaci [27, 69]

$$\mathbf{B}_{M \times M} = N \begin{bmatrix} \frac{1}{2M-1} & \frac{1}{2M-2} & \cdots & \frac{1}{M} \\ \frac{1}{2M-2} & \frac{1}{2M-3} & \cdots & \frac{1}{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{M} & \frac{1}{M-1} & \cdots & 1 \end{bmatrix} \quad (14.20)$$

Wyliczając macierz odwrotną  $\mathbf{B}_{M \times M}^{-1}$  można zauważyć, że np.: dla  $M \geq 12$  macierz ta posiada elementy rzędu  $4 \cdot 10^{22}$ . Podczas rozwiązywania układu przy pomocy komputera zaokrąglenia są na tyle duże, że obliczenia tracą sens.

Reasumując, aproksymacja z funkcjami bazowymi  $n^i$  jest wtedy pozbawiona sensu za wyjątkiem małego  $M$ .

Aby zoptymalizować obliczenia dla aproksymacji wielomianami wyższych rzędów, można zastosować jedną z metod układu normalnego. Do tego celu wykorzystuje się przekształcenia Householdera, metody Goluba lub inną metodę rozkładu macierzy na iloczyn  $QR$  [48, 69].

Inną opcją jest zastosowanie bazy wielomianów ortogonalnych pozwalających uzyskać rozwiązanie bez stosowania przekształceń macierzy odwrotnej, ze względu na diagonalną macierz  $\mathbf{B}_{M \times M}$ .

## 14.4 Funkcje bazowe - wielomiany ortogonalne

Aby rozważyć przypadek zastosowania wielomianów ortogonalnych w aproksymacji metodą liftingu należy wprowadzić pojęcie wielomianu czynnikowego.

### 14.4.1 Własności wielomianów czynnikowych

**Definicja.** *Wielomianem czynnikowym stopnia  $n$  nazywa się iloczyn  $n$  czynników z których pierwszy jest równy  $x$ , zaś każdy następny jest mniejszy od swojego poprzednika o stałą liczbę  $h$ , gdzie [48, 36]*

$$x_h^{[n]} = x(x-h)(x-2h) \dots (x-(n-1)h) \quad (14.21)$$

Ponadto przyjmuje się, że dla  $h = 1$  oznacza się  $x_h^{[n]} \equiv x^{[n]}$  zaś  $x^{[0]} = 1$ .

Badając różnicę rzędu pierwszego dla skończonego wielomianu czynnikowego otrzymuje się

$$\begin{aligned}
 \Delta x_h^{[n]} &= (x+h)_h^{[n]} - x_h^{[n]} = \\
 &= (x+h)x_h^{[n-1]} - x_h^{[n]} \\
 &= (x+h)x_h^{[n-1]} - x_h^{[n-1]}(x - (n-1)h) \\
 &= x_h^{[n-1]}(x+h - x + nh - h) = nhx_h^{[n-1]}
 \end{aligned} \tag{14.22}$$

Analogicznie dla różnic rzędu  $n$  wzór ogólny wynosi

$$\Delta^k x_h^{[n]} = \Delta(\Delta^{k-1} x_h^{[n]}) = \frac{n!}{(n-k)!} h^k x_h^{[n-k]} \tag{14.23}$$

gdzie  $k = 1, 2, \dots, n$ .

Dzięki powyższej analogii wzór na sumę skończoną przyjmuje postać [36]

$$\begin{aligned}
 S_N &= \sum_{i=0}^{N-1} x_h^{[n]}(i) \\
 &= \frac{1}{h(n+1)} \sum_{i=0}^{N-1} \Delta x_h^{[n+1]}(i) \\
 &= \frac{1}{h(n+1)} \{x_h^{[n+1]}(1) - x_h^{[n+1]}(0) + x_h^{[n+1]}(2) - x_h^{[n+1]}(1) + \dots + x_h^{[n+1]}(N) - x_h^{[n+1]}(N-1)\} \\
 &= \frac{x_h^{[n+1]}(N) - x_h^{[n+1]}(0)}{h(n+1)}
 \end{aligned} \tag{14.24}$$

### 14.4.2 Wielomiany ortogonalne

Wielomiany ortogonalne posiadają następujące własności:

**Definicja:** Funkcje  $P(x)$  i  $Q(x)$  nazywamy *ortogonalnymi* na zbiorze punktów  $X = \{x_0, x_1, \dots, x_n\}$  jeśli  $\sum_{i=0}^n P(x_i)Q(x_i) = 0$  [62].

Jeżeli funkcje  $\{P_k(x)\}$  między sobą są parami ortogonalne, wtedy taki układ nazywamy *ortogonalnym* i zachodzi zależność [69]

$$\sum_{i=0}^n P_j(x_i)P_k(x_i) = 0 \tag{14.25}$$

Aby wymusić, żeby funkcja  $P(x)$  nie we wszystkich punktach dziedziny posiadała miejsca zerowe przyjmuje się, że

$$\sum_{i=0}^n P_j^2(x_i) > 0. \quad (14.26)$$

Ponieważ wielomiany  $P_j(x)$ , gdzie  $(j = 0, 1, 2, 3, \dots, m)$  są liniowo niezależne, zatem dowolny wielomian  $W_m(x)$  można przedstawić za pomocą liniowej kombinacji układu ortogonalnego [69]

$$W_m(x) = a_0 P_0(x) + a_1 P_1(x) + \dots + a_m P_m(x) \quad (14.27)$$

gdzie współczynniki  $a_k$  oblicza się poprzez dzielenie wielomianu  $W_m(x)$  przez kolejne wielomiany ortogonalne  $P_k(x)$  wykorzystując algorytm Euklidesa tj.

$$\begin{aligned} \frac{Q_m(x)}{P_m(x)} &= b_m + \frac{R_{m-1}(x)}{P_m(x)} \\ \frac{R_{m-1}(x)}{P_{m-1}(x)} &= b_{m-1} + \frac{R_{m-2}(x)}{P_{m-1}(x)} \\ &\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \cdot \\ \frac{R_1(x)}{P_1(x)} &= b_1 + \frac{R_0(x)}{P_1(x)} \\ \frac{R_0(x)}{P_0(x)} &= b_0 \end{aligned} \quad (14.28)$$

Innym sposobem obliczania współczynników  $a_k$  jest wykorzystanie warunku ortogonalności. Wtedy to, mnożąc obustronnie równanie 14.27 przez  $P_k(x)$  i sumując po wszystkich punktach  $x_0, x_1, \dots, x_n$  wyznacza się  $m + 1$  równań dla  $k = 0, 1, 2, \dots, m$ , gdzie

$$\begin{aligned} \sum_{i=0}^n W_m(x_i) P_k(x_i) &= a_0 \sum_{i=0}^n P_0(x_i) P_k(x_i) + a_1 \sum_{i=0}^n P_1(x_i) P_k(x_i) \\ &+ \dots + a_k \sum_{i=0}^n P_k^2(x_i) + \dots + a_m \sum_{i=0}^n P_m(x_i) P_k(x_i) \end{aligned} \quad (14.29)$$

Na podstawie warunków ortogonalności 14.25 i 14.26 otrzymuje się  $m + 1$  równań

$$\sum_{i=0}^n W_m(x_i) P_k(x_i) = a_k \sum_{i=0}^n P_k^2(x_i) \quad (14.30)$$

z których współczynniki  $a_k$  wynoszą odpowiednio

$$a_k = \frac{\sum_{i=0}^n W_m(x_i) P_k(x_i)}{\sum_{i=0}^n P_k^2(x_i)} = \frac{c_k}{s_k}. \quad (14.31)$$

Na bazie równania 14.27 wylicza się bezpośrednio wielomian aproksymacyjny [62]

$$W_m(x) = \sum_{i=0}^m \frac{c_i}{s_i} P_i(x). \quad (14.32)$$

Funkcja celu 14.1 dotycząca odchylenia standardowego wyraża się wzorem

$$\begin{aligned} s^2 &= \sum_{j=0}^n [W_m(x_j) - \sum_{k=0}^m a_k P_k(x_j)]^2 \\ &= \sum_{j=0}^n [W_m^2(x_j) - 2 \sum_{j=0}^n W_m(x_j) \sum_{k=0}^m a_k P_k(x_j) + \sum_{j=0}^n (\sum_{k=0}^m a_k P_k(x_j))^2] \\ &= \sum_{j=0}^n [W_m^2(x_j) - 2 \sum_{j=0}^n W_m(x_j) \sum_{k=0}^m a_k P_k(x_j) + \\ &+ 2 \sum_{k=0}^m \sum_{i=0}^{k-1} a_k a_i \sum_{j=0}^n P_k(x_j) P_i(x_j) + \sum_{k=0}^m a_k^2 \sum_{j=0}^n P_k^2(x_j)] \\ &= \sum_{j=0}^n W_m^2(x_j) + \sum_{k=0}^m [a_k^2 \sum_{j=0}^n P_k^2(x_j) - 2a_k \sum_{j=0}^n W_m(x_j) P_k(x_j)] \\ &= \sum_{j=0}^n W_m^2(x_j) + \sum_{k=0}^m [a_k^2 s_k - 2a_k c_k] \\ &= \sum_{j=0}^n W_m^2(x_j) + \sum_{k=0}^m s_k (a_k - \frac{c_k}{s_k}) - \sum_{k=0}^m \frac{c_k^2}{s_k}. \end{aligned} \quad (14.33)$$

Ponieważ pierwszy i ostatni człon wyrażenia 14.33 nie zależy od współczynników  $a_k$ , tak więc odchylenie standardowe jest najmniejsze w przypadku gdy drugi człon znika. Zdarzenie to zachodzi w przypadku gdy

$$a_k = \frac{c_k}{s_k} = \frac{\sum_{i=0}^n W_m(x_i) P_k(x_i)}{\sum_{i=0}^n P_k^2(x_i)}. \quad (14.34)$$

Odchylenie standardowe, a tym samym funkcja celu wynosi wtedy [62]

$$s = \sqrt{\sum_{j=0}^n W_m^2(x_j) - \sum_{k=0}^m \frac{[\sum_{i=0}^n W_m(x_i) P_k(x_i)]^2}{\sum_{i=0}^n P_k^2(x_i)}} \quad (14.35)$$

Metoda aproksymacji w oparciu o wielomiany ortogonalne pozwala wyeliminować problem złego uwarunkowania macierzy współczynników, który jest główną przyczyną błędów zaokrągleń [21, 22].

Ponieważ macierz  $\mathbf{B}_{M \times M}$  jest wtedy macierzą diagonalną, odpowiadająca jej macierz odwrotna  $\mathbf{B}_{M \times M}^{-1}$  jest macierzą tego samego typu o współczynnikach na przekątnej  $\frac{1}{b_{ii}}$ . Zależność ta w istotny sposób redukuje złożoność obliczeniową w konsekwencji zmniejszając błędy obliczeniowe.

### 14.4.3 Wielomiany ortogonalne Grama

Szczególnym przypadkiem wielomianów ortogonalnych są wielomiany Grama [69]. Wielomiany te wykorzystują układ  $n + 1$  równoodległych punktów  $x = x_0, x_1, \dots, x_n$  różniących się między sobą o stałe  $h$  gdzie  $x_i = x_0 + ih$  dla  $i = 0, 1, 2, \dots, n$ .

Do analizy funkcji dla punktów  $i = 0, 1, 2, \dots, n$  wystarczy zatem skorzystać z przekształcenia liniowego  $i = \frac{x_i - x_0}{h}$ .

Budując układ wielomianów ortogonalnych

$$P_{0n}, P_{1n}, P_{2n}, \dots, P_{mn} \quad (14.36)$$

stopni od 0 do  $m$  gdzie  $m \leq n$  na zbiorze punktów  $i = 0, 1, 2, \dots, m$  takich, że dla  $j \neq k$

$$\sum_{i=0}^n P_{jn}(i)P_{kn}(i) = 0 \quad (14.37)$$

należy przyjąć, że [62]

$$P_{kn}(0) = 1 \text{ dla } k = 0, 1, \dots, m. \quad (14.38)$$

Ponadto, wielomiany  $P_{kn}(t)$  przyjmują następującą postać uogólnioną potęgowo

$$P_{kn}(i) = 1 + a_1 i^{[1]} + a_2 i^{[2]} + \dots + a_k i^{[k]} \quad (14.39)$$

gdzie

$$i^{[j]} = i(i-1)(i-2)\dots(i-j+1) \quad (14.40)$$

nazywane są *potęgami uogólnionymi* [48, 36].

Wiedząc, że dowolny wielomian  $P_{jn}$  można wyrazić za pomocą wzoru

$$P_{jn}(i) = \sum_{j=0}^k a_i (i+j)^{[j]} \quad (14.41)$$

aby zachodził wzór 14.37 warunkiem wystarczającym jest, aby każdy z czynników  $(i+j)^{[j]}$  był ortogonalny względem  $P_{kn}(i)$  tj. [69]

$$\sum_{i=0}^n (i+j)^{[j]} P_{kn}(i) = 0 \quad (14.42)$$

dla  $j = 0, 1, 2, \dots, k-1$ .

Podstawiając wzór 14.39 do 14.42 otrzymuje się wyrażenie

$$\sum_{i=0}^n (i+j)^{[j]} P_{kn}(i) = \sum_{i=0}^n (i+j)^{[j]} + a_1 \sum_{i=0}^n (i+j)^{[j+1]} + \dots + a_k \sum_{i=0}^n (i+j)^{[j+k]} = 0. \quad (14.43)$$

Ponieważ zgodnie ze wzorem 14.24 i własnością  $j^{[k+1]} = 0$  dla  $k \geq j$  suma skończona wynosi [36]

$$\sum_{i=0}^n (i+j)^k = \frac{(n+j+1)^{[k+1]} - j^{[k+1]}}{k+1} = \frac{(n+j+1)^{[k+1]}}{k+1}. \quad (14.44)$$



postać 14.43 redukuje się do wyrażenia

$$\frac{(n+j+1)^{[j+1]}}{j+1} + \frac{(n+j+1)^{[j+2]}}{j+2} a_1 + \dots + \frac{(n+j+1)^{[j+1+k]}}{j+1+k} a_k = 0, \quad (14.45)$$

gdzie, dzieląc obustronnie równanie przez  $\frac{(n+j+1)^{[j+1]}}{j+1}$  otrzymuje się szereg postaci

$$\frac{1}{j+1} + \frac{n^{[1]}}{j+2} a_1 + \dots + \frac{n^{[k]}}{j+1+k} a_k = 0. \quad (14.46)$$

Dla  $n^{[k]} a_k = b_k$  układ równań złożony z  $k$  niewiadomych  $a_1, a_2, \dots, a_k$  i  $k$  równań można przedstawić jako

$$\frac{1}{j+1} + \frac{b_1}{j+2} + \dots + \frac{b_k}{j+1+k} = 0. \quad (14.47)$$

Obliczając współczynniki  $b_s$  można skorzystać z rachunku macierzowego lub wykonać obliczenia przy pomocy operacji podstawienia. Wyliczenie realizuje się podstawiając z prawej strony równania iloraz wielomianu w postaci iloczynowej, gdzie dla każdego  $j = 0, 1, 2, \dots, k-1$  wartość licznika wynosi 0, dzieląc go przez wspólny mianownik członów lewej strony równania tj [36].

$$\frac{1}{j+1} + \frac{b_1}{j+2} + \dots + \frac{b_k}{j+k+1} = \frac{c_k j^{[k]}}{(j+1+k)^{[k+1]}}. \quad (14.48)$$

Mnożąc obustronnie równanie przez  $j+1$  i wstawiając za  $j$  wartość  $(-1)$  otrzymuje się

$$1 = \frac{c_k (-1) \cdot (-2) \cdot \dots \cdot (-k)}{1 \cdot 2 \cdot \dots \cdot k} \quad (14.49)$$

gdzie w konsekwencji

$$c_k = (-1)^k. \quad (14.50)$$

Podstawiając do równania 14.48 wyliczony współczynnik  $c_k$  można w sposób analogiczny wyliczyć kolejne współczynniki  $b_s$ , tzn. mnożąc równanie

$$\frac{1}{j+1} + \frac{b_1}{j+2} + \dots + \frac{b_s}{j+1+s} + \dots + \frac{b_k}{j+1+k} = \frac{(-1)^k j^{[k]}}{(j+1+k)^{[k+1]}} \quad (14.51)$$

przez  $j+1+s$  i podstawiając za  $j = -(1+s)$ .

Ponieważ współczynnik  $b_s$  wynosi

$$\begin{aligned} b_s &= \frac{(-1)^k j^{[k]}}{(j+1+k) \cdot (j+k) \cdot (j+k-1) \cdot \dots \cdot (j+2+s) \cdot (j+s) \cdot \dots \cdot (j+2) \cdot (j+1)} \Big|_{j=-(1+s)} = \\ &= \frac{(-1)^{2k} \cdot (k+s)^{[k]}}{(k-s)! \cdot (-1)^k \cdot s!} = (-1)^k \cdot \frac{(k+s)^{[k]}}{(k-s)! s!} \end{aligned} \quad (14.52)$$

gdzie

$$\frac{(k+s)^{[k]}}{(k-s)!s!} = \frac{(k+s)^{[k]}s!}{s!(k-s)!s!} = \frac{k!}{s!(k-s)!} \cdot \frac{(k+s)!}{s!k!} = \binom{k}{s} \cdot \binom{k+s}{s}, \quad (14.53)$$

wzór po redukcji przyjmuje wartość

$$b_s = (-1)^k \cdot \binom{k}{s} \cdot \binom{k+s}{s}. \quad (14.54)$$

Korzystając z obliczonych współczynników  $b_s$  i właściwości  $a_s = \frac{b_s}{n^{[s]}}$  można określić końcowy wzór wielomianów ortogonalnych Grama, które wyrażają się równaniem [69]

$$P_{kn}(i) = 1 - \binom{k}{1} \cdot \binom{k+1}{1} \cdot \frac{i}{n} + \binom{k}{2} \cdot \binom{k+2}{2} \cdot \frac{i^{[2]}}{n^{[2]}} - \dots + (-1)^k \cdot \binom{k}{k} \cdot \binom{k+k}{k} \cdot \frac{i^{[k]}}{n^{[k]}} \quad (14.55)$$

lub w postaci symbolicznej [36, 69]

$$P_{kn}(i) = \sum_{s=0}^k (-1)^s \cdot \binom{k}{s} \cdot \binom{k+s}{s} \cdot \frac{i^{[s]}}{n^{[s]}}, \quad k = 0, 1, 2, \dots, m. \quad (14.56)$$

Szereg 14.56 określa postać ogólną wielomianów Grama stopni  $k = 0, 1, 2, \dots, m$  dla  $n+1$  węzłów gdzie  $i = 0, 1, \dots, n$  ( $i = \frac{x_i - x_0}{h}$ ).

Wykorzystując wzór ogólny 14.56 na mocy równania 14.32 określa się wzór aproksymacyjny Grama

$$W_m(x) = \sum_{k=0}^m \frac{c_k}{s_k} P_{k,n}\left(\frac{x-x_0}{h}\right) \quad \text{dla } m \leq n, \quad (14.57)$$

gdzie

$$c_k = \sum_{i=0}^n W_m(x_i) P_{kn}(x_i) \quad (14.58)$$

i

$$s_k = \sum_{i=0}^n P_{kn}^2\left(\frac{x_i - x_0}{h}\right) = \frac{(n+k+1)^{[k+1]}}{(2k+1) \cdot n^{[k]}}. \quad (14.59)$$

Istotną cechą omawianych wielomianów jest to, że podstawiając za  $P_{kn}(\cdot) \rightarrow \alpha_k P_{kn}(\cdot)$  otrzymuje się ten sam wielomian  $W_m(\cdot)$ . Dzięki takiej własności można mnożyć dowolny wielomian Grama przez dowolną liczbę różną od zera uzyskując przy tym całkowite wartości wielomianów w odpowiednich węzłach aproksymacji.

## 14.5 Aproksymacja trygonometryczna

Podczas badania zjawisk czasu rzeczywistego można spotykać się z sytuacją, że analizowana funkcja jest funkcją sinusoidalną. Stąd też zamiast aproksymowania ją wielomianami algebraicznymi do tego celu wygodniej jest użyć wielomianów trygonometrycznych [46, 69].

Do rozważań można przyjąć, że punkty pomiarowe są równoodległe na przedziale  $\langle t_0, t_{2L-1} \rangle$  tj.

$$x_i = \frac{\pi k}{L}, \text{ gdzie } k = \frac{t_i - t_0}{h}, i = 0, 1, 2, \dots, 2L - 1 \quad (14.60)$$

ponadto ortogonalnymi funkcjami układu bazowego jest zbiór

$$1, \sin(x_i), \cos(x_i), \sin(2x_i), \cos(2x_i), \dots, \sin(jx_i), \cos(jx_i). \quad (14.61)$$

Układ ten posiada własność ortogonalności nie tylko na przedziale ciągłym  $\langle 0, 2\pi \rangle$  lecz także na przedziale dyskretnym [69, 36], gdzie

$$\sum_{i=0}^{2L-1} \cos(kx_i)\cos(lx_i) = \begin{cases} 0 & k \neq l \\ L & k = l \neq 0 \\ 2L & k = l = 0 \end{cases}, \quad (14.62)$$

$$\sum_{i=0}^{2L-1} \sin(kx_i)\sin(lx_i) = \begin{cases} 0 & k \neq l \\ L & k = l \neq 0 \\ 0 & k = l = 0 \end{cases}, \quad (14.63)$$

$$\sum_{i=0}^{2L-1} \cos(kx_i)\sin(lx_i) = 0 \quad (14.64)$$

dla  $k, l \in \mathbb{Z} \cap \langle 0, L \rangle$ .

Ze względu na przybliżenie funkcji

$$y_n = a_0 + \sum_{j=1}^m (a_j \cos(jx) + b_j \sin(jx)) \quad (14.65)$$

macierz  $\mathbf{B}_{M \times M}$  można zdefiniować jako

$$\begin{aligned}
 \mathbf{B}_{M \times M} &= \begin{bmatrix} \sum_{i=0}^{2L-1} 1 \cdot 1 & \sum_{i=0}^{2L-1} 1 \cdot \cos(x) & \dots & \sum_{i=0}^{2L-1} 1 \cdot \sin(mx) \\ \sum_{i=0}^{2L-1} \cos(x) \cdot 1 & \sum_{i=0}^{2L-1} \cos(x)\cos(x) & \dots & \sum_{i=0}^{2L-1} \cos(x)\sin(mx) \\ \vdots & \vdots & & \vdots \\ \sum_{i=0}^{2L-1} \cos(mx) \cdot 1 & \sum_{i=0}^{2L-1} \cos(mx)\cos(x) & \dots & \sum_{i=0}^{2L-1} \cos(mx)\sin(mx) \\ \sum_{i=0}^{2L-1} \sin(x) \cdot 1 & \sum_{i=0}^{2L-1} \sin(x)\cos(x) & \dots & \sum_{i=0}^{2L-1} \sin(x)\sin(mx) \\ \vdots & \vdots & & \vdots \\ \sum_{i=0}^{2L-1} \sin(mx) \cdot 1 & \sum_{i=0}^{2L-1} \sin(mx)\cos(x) & \dots & \sum_{i=0}^{2L-1} \sin(mx)\sin(mx) \end{bmatrix} \\
 &= \begin{bmatrix} 2L & 0 & \dots & 0 \\ 0 & L & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & L \end{bmatrix}.
 \end{aligned} \tag{14.66}$$

Ponieważ przybliżony wzór 14.65 jest częściowym szeregiem Fouriera, aproksymację wyznacza się na podstawie minimalizacji funkcji celu

$$f = \sum_{i=0}^{2L-1} [s(x_i) - y_m(x_i)]^2. \tag{14.67}$$

W przypadku pesymistycznym szereg przybliży funkcję dla dużej liczby współczynników  $a_j$  i  $b_j$ , natomiast dla przypadków optymistycznych funkcji okresowych, szereg ogranicza się do ich kilku niezerowych wartości.

## 14.6 Implementacja aproksymacji MNK w bloku predykcji z wykorzystaniem baz ortogonalnych

Aproksymacja funkcji w przypadku próbek uzyskiwanych empirycznie lub z pewnym pomiarem błędu jest w większości przypadków metodą zalecaną podczas analizy funkcji. Powodem tym jest wygładzanie charakterystyk których zaszumienie było spowodowane na przykład błędem pomiarowym. Dzięki tej metodzie w kolejnych etapach analizy wielorozdzielczej można ukryć szczegóły które nie są istotne dla konkretnych badań. Inną istotną zaletą jest aproksymowanie funkcji wielomianami o stopniu niższym aniżeli podczas interpolacji. Pozwala ona także na wykorzystanie do tego celu szerokiej grupy funkcji bazowych.

Aby zmniejszyć złożoność obliczeniową, w celu wyliczenia wektora predykcji można wykorzystać bazy ortogonalne, w konsekwencji minimalizując błędy przybliżeń.

Ponieważ macierz  $\mathbf{B}_{M \times M}$  wykorzystywana w schemacie liftingu dla baz ortogonalnych jest macierzą diagonalną, zatem obniża to znacznie koszt obliczeń podczas operacji przekształcania w/w macierzy do macierzy odwrotnej ( $b_{ii}^{-1} = \frac{1}{b_{ii}}, b_{ij}^{-1} = 0$  dla  $i \neq j$ ).

### 14.6.1 Implementacja aproksymacji MNK w bloku predykcji za pomocą wielomianów ortogonalnych Grama

```
<< SchematLiftinguZ Aproksymacja MNK >
```

```
tablicaDanych = { 1.3 , 1.245 , 1.095 , 0.855 , 0.514 , 0.037 , -0.6 , -1.295 , -1.767 , -1.914 };
zakresArgumentow = { 0 , 2.7 };
```

```
? SchematLiftingu MNK Funkcji Z Wielomianem Ortogonalnym
```

```
SchematLiftingu MNK Funkcji Z Wielomianem Ortogonalnym [zakresArgumentow, tablicaDanych, stopienWielomianu Ortogonalnego, typPrezentacji, numer]
SchematLiftingu MNK Funkcji Z Wielomianem Ortogonalnym [zakresArgumentow, tablicaDanych, stopienWielomianu Ortogonalnego, typPrezentacji]
Prezentacja danych schematu liftingu z aproksymacja wielomianowa funkcji jednowymiarowej za
pomocą wielomianów ortogonalnych z trybem wyświetlania (PostacMacierzowa, Wykres, WzorPostacOgolna, WzorPostacWielomianowa, Predykcja)
```

Rysunek 14.1: Aproksymacja funkcji za pomocą wielomianów ortogonalnych - przygotowanie danych

Metoda `SchematLiftingu MNK Funkcji Z Wielomianem Ortogonalnym` posiada 5 argumentów wejściowych z których pierwszy `zakresArgumentow` jest dwuelementową tablicą określającą granice przedziału zbioru argumentów badanych próbek, następny parametr `tablicaDanych` jest tablicą wartości próbek funkcji badanej. Trzeci argument `stopienWielomianu Ortogonalnego` określa stopień wielomianu wykorzystywany do celów aproksymacji, zaś `typPrezentacji` jest parametrem określającym sposób prezentacji danych na wyjściu. Ostatnia zmienna `numer` jest wartością opcjonalną wykorzystywaną w przypadku określania wektora predykcji funkcji badanej dla konkretnej próbki nieparzystej.

$$\text{SchematLiftingu MNK Funkcji Z Wielomianem Ortogonalnym [zakresArgumentow, tablicaDanych, 5, "PostacMacierzowa"]}$$

$$\begin{pmatrix} -0.053 \\ 1.82187 \\ -0.340773 \\ -0.203925 \\ 0.0913531 \\ -0.0125154 \end{pmatrix} = \begin{pmatrix} \frac{1}{10} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{27}{110} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{3}{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{147}{715} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{81}{715} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{3}{65} \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{7}{9} & \frac{5}{9} & \frac{1}{3} & \frac{1}{9} & -\frac{1}{9} & -\frac{1}{3} & -\frac{5}{9} & -\frac{7}{9} & -1 \\ 1 & \frac{1}{3} & -\frac{1}{6} & -\frac{1}{2} & -\frac{2}{3} & -\frac{2}{3} & -\frac{1}{2} & -\frac{1}{6} & \frac{1}{3} & 1 \\ 1 & -\frac{1}{3} & -\frac{5}{6} & -\frac{31}{42} & -\frac{2}{7} & \frac{2}{7} & \frac{31}{42} & \frac{5}{6} & \frac{1}{3} & -1 \\ 1 & -\frac{11}{9} & -\frac{17}{18} & \frac{1}{6} & 1 & 1 & \frac{1}{6} & -\frac{17}{18} & -\frac{11}{9} & 1 \\ 1 & -\frac{7}{3} & \frac{1}{6} & \frac{11}{6} & 1 & -1 & -\frac{11}{6} & -\frac{1}{6} & \frac{7}{3} & -1 \end{pmatrix} \cdot \begin{pmatrix} 1.3 \\ 1.245 \\ 1.095 \\ 0.855 \\ 0.514 \\ 0.037 \\ -0.6 \\ -1.295 \\ -1.767 \\ -1.914 \end{pmatrix}$$

Rysunek 14.2: Wielomian ortogonalny - współczynniki aproksymacji

W przypadku gdy `typPrezentacji` posiada wartość „`PostacMacierzowa`” metoda zwraca reprezentację współczynników funkcji bazowych  $\mathbf{A}_{M \times 1}$  w postaci iloczynu macierzy zgodnie ze wzorem 14.8. Ponieważ waga  $w_{ij} = 1$ , wyznacza to  $\mathbf{W}_{N \times N} = \mathbf{I}_{N \times N}$  gdzie wektor współczynników wielomianu  $\mathbf{A}_{M \times 1} = \mathbf{B}_{M \times M}^{-1} \cdot \Psi_{M \times N} \cdot \text{Signal}_{N \times 1}$ .

```
SchematLiftinguMNFunkcjiZWielomianemOrtogonalnym [zakresArgumentow, tablicaDanych,
5, "WzorPostacOgolna"]
W(x) = -0.053 P0,9[3.33333 x] + 1.82187 P1,9[3.33333 x] - 0.340773 P2,9[3.33333 x] -
0.203925 P3,9[3.33333 x] + 0.0913531 P4,9[3.33333 x] - 0.0125154 P5,9[3.33333 x]
```

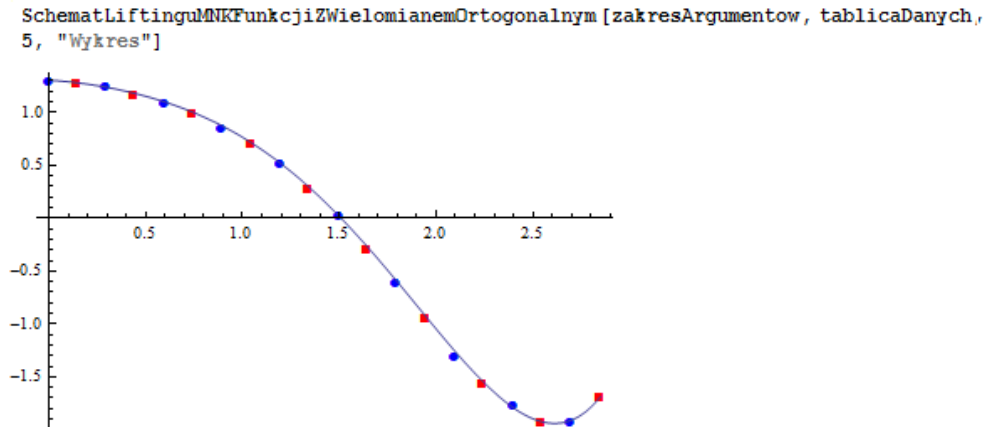
Rysunek 14.3: Wielomian ortogonalny - postać ogólna

Argument **typPrezentacji** równy „WzorPostacOgolna” obrazuje równanie wielomianowe funkcji aproksymującej w postaci sumy iloczynów współczynników i funkcji Grama, które są ortogonalnymi wielomianami bazowymi funkcji aproksymującej.

```
SchematLiftinguMNFunkcjiZWielomianemOrtogonalnym [zakresArgumentow, tablicaDanych,
5, "WzorPostacWielomianowa"]
W(x) = 1.30301 - 0.100607 x - 0.459942 x2 + 0.254789 x3 - 0.318347 x4 + 0.0858394 x5
```

Rysunek 14.4: Wielomian ortogonalny - postać wielomianowa

Dla wartości „WzorPostacWielomianowa” wynik metody jest postacią algebraiczną funkcji wielomianowej doprowadzonej do najprostszej postaci.



Rysunek 14.5: Wielomian ortogonalny - wykres predykcji

Typ „Wykres” pozwala na prezentację wyników w postaci graficznej, gdzie na wykresie można wyodrębnić trzy informacje. Próbkę przedstawioną w postaci  $\circ$  reprezentują funkcję przekazywaną w argumencie **tablicaDanych**, jest to tzw. funkcja próbek parzystych. Próbkę reprezentowaną za pomocą symbolu  $\square$  są próbkami nieparzystymi, szacowanymi w operacji predykcji, zaś funkcja ciągła jest jego funkcją aproksymującą. Jeżeli próbki parzyste i szacowane nieparzyste, pokrywają się z funkcją aproksymującą - implikuje to, że funkcja detali jest funkcją zerową (oszacowane próbki nieparzyste, posiadają te same wartości co próbki nieparzyste funkcji aproksymowanej) zatem wektor uaktualnienia nie ma wpływu na aktualizację próbek parzystych (reprezentację zgrubną).

$$\text{SchematLiftinguMKNKFunkcjiZWielomianemOrtogonalnym}[\text{zakresArgumentow}, \text{tablicaDanych}, \\ 5, \text{"Predykcja"}] \\ \left\{ \frac{5}{128}, -\frac{15}{128}, \frac{3}{256}, \frac{55}{256}, \frac{45}{128}, \frac{45}{128}, \frac{55}{256}, \frac{3}{256}, -\frac{15}{128}, \frac{5}{128} \right\}$$

Rysunek 14.6: Wielomian ortogonalny - współczynniki predykcji

Opisywana procedura **SchematLiftinguMKNKFunkcjiZWielomianemOrtogonalnym** dla typu prezentacji „**Predykcja**” w sytuacji gdy nie jest zdefiniowany ostatni argument **numer** zwraca predyktor dla próbki nieparzystej leżącej symetrycznie w środku dziedziny badanej funkcji pomiędzy  $\frac{n}{2}$  próbką parzystą, a próbką  $(\frac{n}{2} + 1)$ .

Wynik prezentowany jest w postaci tablicy.

$$\text{SchematLiftinguMKNKFunkcjiZWielomianemOrtogonalnym}[\text{zakresArgumentow}, \text{tablicaDanych}, \\ 5, \text{"Predykcja"}, 1 + 1/2] \\ \left\{ \frac{34863}{91520}, \frac{56003}{91520}, \frac{108763}{549120}, -\frac{2209}{16640}, -\frac{1149}{7040}, \frac{1061}{274560}, \frac{1913}{14080}, \frac{11129}{183040}, -\frac{39079}{274560}, \frac{4287}{91520} \right\}$$

Rysunek 14.7: Wielomian ortogonalny - współczynniki predykcji dla próbki nieparzystej numer 1.5

W przypadku gdy oprócz parametru **typPrezentacji** o wartości „**Predykcja**” zostanie zdefiniowany dodatkowy parametr **numer**, wtedy to zostanie wyświetlona tablica predyktora dla wartości przewidywanej numerem próbki *numer*.

Przykładowo, gdy wartość parametru *numer* =  $1\frac{1}{2}$  wtedy na mocy predyktora określa się wartość wektora predykcji dla szacowanej nieparzystej próbki leżącej pomiędzy 1 a 2 próbką parzystą funkcji badanej.

### 14.6.2 Implementacja aproksymacji MNK w bloku predykcji za pomocą wielomianów trygonometrycznych

```
<< SchematLiftinguZAproksymacjaMNK`
```

```
signal = Table[{i, 2 - Sin[i] + 3 Cos[3 i]}, {i, 0, 7 Pi/4, Pi/4}];
```

```
? SchematLiftinguMNKFunkcjiZWielomianemTrygonometrycznym
```

```
SchematLiftinguMNKFunkcjiZWielomianemTrygonometrycznym[wektorDanych, stopienWielomianuTrygonometrycznego, typPrezentacji, numer]
SchematLiftinguMNKFunkcjiZWielomianemTrygonometrycznym[wektorDanych, stopienWielomianuTrygonometrycznego, typPrezentacji]
Prezentacja danych schematu liftingu z aproksymacja trygonometryczna
w bloku predykcji z trybem wyswietlania (PostacMacierzowa, Wykres, WzorPostacOgolna, Predykcja)
```

Rysunek 14.8: Aproksymacja funkcji schematem liftingu za pomocą wielomianów trygonometrycznych - przygotowane danych

Metoda **SchematLiftinguMNKFunkcjiZWielomianemTrygonometrycznym** posiada 4 argumenty wejściowe z których pierwszy **wektorDanych** jest tablicą wartości próbek funkcji badanej, zaś **stopienWielomianuTrygonometrycznego** jest argumentem określającym stopień wielomianu wykorzystywany do celów aproksymacji. Stopień wielomianu trygonometrycznego  $k$  rozdzielony jest na jeden współczynnik stały,  $\frac{k-1}{2}$  współczynników cosinusów  $\{cos(x), \dots, cos(\frac{k-1}{2}x)\}$  i  $\frac{k-1}{2}$  współczynników sinusów  $\{sin(x), \dots, sin(\frac{k-1}{2}x)\}$ , zatem parametr **stopienWielomianuTrygonometrycznego** musi być liczbą nieparzystą.

Argument **typPrezentacji** jest parametrem określającym sposób prezentacji danych na wyjściu. Ostatni argument **numer** jest argumentem opcjonalnym który analogicznie jak w przypadku metody **SchematLiftinguMNKFunkcjiZWielomianemOrtogonalnym**, wykorzystywany jest jedynie w przypadku określania wektora predykcji funkcji badanej dla konkretnej próbki nieparzystej.

```
SchematLiftinguMNKFunkcjiZWielomianemTrygonometrycznym [signal, 7, "PostacMacierzowa"]
```

$$\begin{pmatrix} 2. \\ 0. \\ 0. \\ 3. \\ -1. \\ 0. \\ 0. \end{pmatrix} = \begin{pmatrix} 0.125 & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0.25 & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0.25 & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0.25 & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0.25 & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0.25 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0.25 \end{pmatrix} \cdot \begin{pmatrix} 1. & 1. & 1. & 1. & 1. & 1. & 1. \\ 1. & 0.707107 & 0. & -0.707107 & -1. & -0.707107 & 0. & 0.707107 \\ 1. & 0. & -1. & 0. & 1. & 0. & -1. & 0. \\ 1. & -0.707107 & 0. & 0.707107 & -1. & 0.707107 & 0. & -0.707107 \\ 0. & 0.707107 & 1. & 0.707107 & 0. & -0.707107 & -1. & -0.707107 \\ 0. & 1. & 0. & -1. & 0. & 1. & 0. & -1. \\ 0. & 0.707107 & -1. & 0.707107 & 0. & -0.707107 & 1. & -0.707107 \end{pmatrix} \cdot \begin{pmatrix} 5. \\ -0.828427 \\ 1. \\ 3.41421 \\ -1. \\ 4.82843 \\ 3. \\ 0.585786 \end{pmatrix}$$

Rysunek 14.9: Wielomian trygonometryczny - współczynniki aproksymacji



W przypadku gdy **typPrezentacji** posiada wartość „**PostacMacierzowa**”, metoda zwraca reprezentację współczynników funkcji bazowych  $\mathbf{A}_{M \times 1}$  w postaci iloczynu macierzy zgodnie ze wzorem 14.8.

Ponieważ waga  $w_{ij} = 1$ , wyznacza to  $\mathbf{W}_{N \times N} = \mathbf{I}_{N \times N}$  zatem  $\mathbf{A}_{M \times 1} = \mathbf{B}_{M \times M}^{-1} \cdot \Psi_{M \times N} \cdot \mathbf{Signal}_{N \times 1}$ .

Analizując  $\mathbf{B}_{M \times M}^{-1}$  na podstawie przykładu z rys. 14.9 i komórki  $b_{[1,1]}^{-1} = \frac{1}{2L}$  można wywnioskować liczbę próbek badanych  $2L$ , tak więc dla powyższego przypadku ich liczba wynosi  $(0.125)^{-1} = (\frac{1}{8})^{-1} = 8$ .

Ponieważ stopień wielomianu trygonometrycznego  $k$  jest nieparzysty, zaś  $\max\{k\} = 2L - 1$  stopień wielomianu równy 7 jest stopniem maksymalnym funkcji aproksymującej.

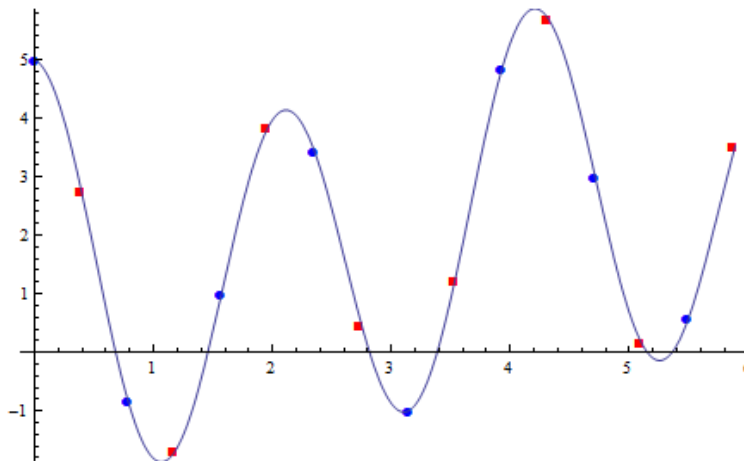
```
SchematLiftinguMNFunkcjiZWielomianemTrygonometrycznym [signal, 7, "WzorPostacOgolna"]
```

```
W(t) = 3. cos(3 t) - 1. sin(t) + 2.
```

Rysunek 14.10: Wielomian trygonometryczny - postać ogólna

Argument **typPrezentacji** równy „**WzorPostacOgolna**” prezentuje równanie trygonometryczne funkcji aproksymującej w postaci sumy iloczynów współczynników i funkcji bazowych sinusoidalnych.

```
Show[ SchematLiftinguMNFunkcjiZWielomianemTrygonometrycznym [signal, 7, "Wykres" ]]
```



Rysunek 14.11: Wielomian trygonometryczny - wykres

Typ „**Wykres**” pozwala na prezentację wyników w postaci graficznej, gdzie na wykresie można wyodrębnić trzy informacje. Próbki przedstawione w postaci  $\circ$  reprezentują wartość próbek przekazywanych w argumencie **signal**, jest to tzw. funkcja próbek parzystych. Próbki reprezentowane za pomocą symbolu  $\square$ , są próbkami szacowanymi podczas operacji predykcji, zaś funkcja ciągła jest funkcją trygonometryczną aproksymującą badane próbki. Jeżeli próbki parzyste i szacowane nieparzyste pokrywają się z funkcją aproksymującą, trygonometryczną - implikuje to, że reprezentacja detali jest funkcją zerową (szacowane próbki nieparzyste pokrywają się z oryginalnymi próbkami nieparzystymi funkcji). Analogicznie jak w przypadku wielomianów ortogonalnych opisanych w rozdziale poprzednim, prowadzi to do zaniku bloku uaktualnienia.

```
N[SchematLiftinguMNFunkcjiZWielomianemTrygonometrycznym [signal, 7, "Predykcja"]]  
{-0.024864, 0.0835223, -0.187076, 0.628417, 0.628417, -0.187076, 0.0835223, -0.024864}
```

Rysunek 14.12: Wielomian trygonometryczny - predykcja

Typ prezentacji „**Predykcja**” w sytuacji gdy nie jest zdefiniowany ostatni argument **numer**, zwraca, w postaci tablicy, predyktor dla próbki nieparzystej leżącej symetrycznie w środku dziedziny funkcji. pomiędzy  $\frac{n}{2}$  próbką a próbką  $(\frac{n}{2} + 1)$ . Rezultat dla typu „**Predykcja**” wyliczany jest dla próbek o  $n$  parzystym, gdzie  $n$  jest liczbą próbek analizowanego zakresu dziedziny funkcji.

```
N[SchematLiftinguMNFunkcjiZWielomianemTrygonometrycznym [signal, 7, "Predykcja", 3 + 1/2]]  
{0.0835223, -0.187076, 0.628417, 0.628417, -0.187076, 0.0835223, -0.024864, -0.024864}
```

Rysunek 14.13: Wielomian trygonometryczny - współczynniki predykcji dla próbki nieparzystej numer 3.5

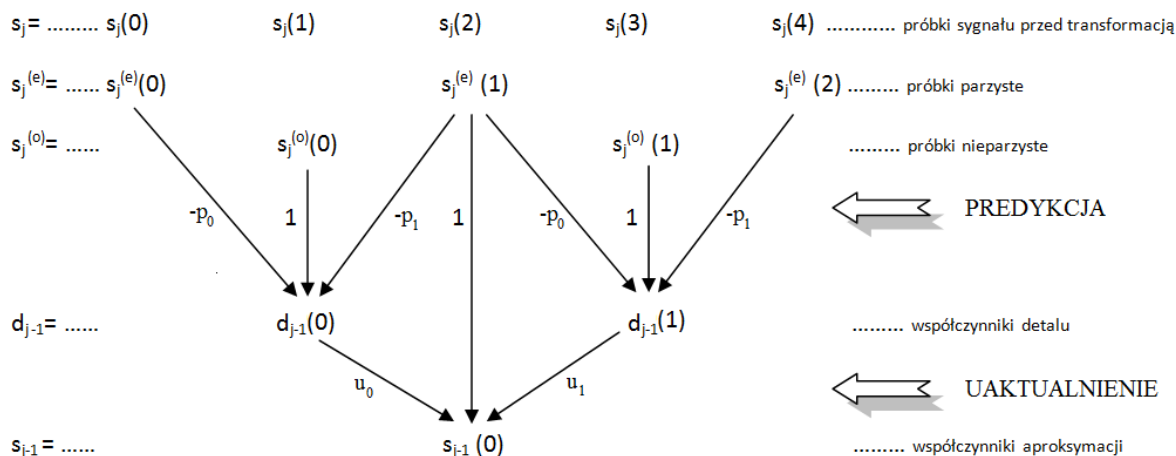
W przypadku gdy dla typu **typPrezentacji** o wartości „**Predykcja**” zostanie zdefiniowany argument **numer**, wtedy to zostanie wyliczony wektor predykcji dla szacowanej nowej próbki nieparzystej z numerem o wartości *numer* .

Dla przykładu, gdy  $numer = 3\frac{1}{2}$  zgodnie z założeniem funkcja **SchematLiftinguMNFunkcjiZWielomianemTrygonometrycznym** wylicza wartości współczynników wektora predykcji dla nowej próbki leżącej pomiędzy 3 a 4 próbką parzystą funkcji badanej.



# Relacja pomiędzy dyskretną transformatą falkową a transformatą realizowaną przy użyciu schematu liftingu

W klasycznej, dyskretniej teorii falkowej do wyznaczania współczynników aproksymacji podczas dekompozycji funkcji, wykorzystuje się bank filtru dolnoprzepustowego  $\tilde{h}$  wraz z decymacją  $\downarrow 2$  funkcji. Analogicznie, współczynniki przestrzeni detali otrzymuje się za pomocą filtracji górnoprzepustowej z wykorzystaniem filtru  $\tilde{g}$  i decymacji  $\downarrow 2$  tej samej funkcji zgodnie ze schematem 12.1. Zależność pomiędzy klasyczną teorią falkową a jej predykcyjnym odpowiednikiem można zaprezentować za pomocą schematu liftingu z wektorem predykcji  $\mathbf{p} = [p_0, p_1]$  i wektorem uaktualnienia  $\mathbf{u} = [u_0, u_1]$  [72, 56]



Rysunek 15.1: Zależność pomiędzy transformatą klasyczną, a transformatą realizowaną za pomocą schematu liftingu

Schemat 15.1 pozwala w prosty sposób otrzymać biortogonalny system falkowy. Wagi filtrów

oblicza się za pomocą sumowania dróg przejść dla każdej próbki analizowanej funkcji.

Współczynniki filtru górnoprzepustowego analizy funkcji otrzymuje się w sposób jawny, poprzez analizę wag gałęzi łączących próbki parzyste  $s_j^{(e)}$  i nieparzyste  $s_j^{(o)}$  ze współczynnikami detalu  $d_{j-1}$ .

$$\begin{cases} \tilde{g}_0 = -p_0 \\ \tilde{g}_1 = 1 \\ \tilde{g}_2 = -p_1 \end{cases}, \quad (15.1)$$

natomiast współczynniki filtru dolnoprzepustowego  $\tilde{h}$  ze względu na operację opuszczania (predykcji) i podnoszenia (uaktualnienia) na zasadzie sumowania dróg przejścia od próbek  $s_j^{(e)}$  i  $s_j^{(o)}$  zaczynając, a kończąc na współczynnikach aproksymacji  $s_{j-1}$  tj.

$$\begin{cases} \tilde{h}_0 = -p_0 \cdot u_0 \\ \tilde{h}_1 = 1 \cdot u_0 \\ \tilde{h}_2 = -p_1 \cdot u_0 + 1 - p_0 \cdot u_1 \\ \tilde{h}_3 = 1 \cdot u_1 \\ \tilde{h}_4 = -p_1 \cdot u_1 \end{cases} \quad (15.2)$$

Przykładem realizacji filtrów za pomocą schematu liftingu może być schemat liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów i funkcją aproksymującą  $f(x) = a_0 + a_1x$ .

Aproksymacja na podstawie analizy dwóch próbek parzystych wyznacza wektory  $\mathbf{p} = [0.5; 0.5]$  i  $\mathbf{u} = [0.25; 0.25]$  (rozdz. 13.8.2).

Współczynniki filtru dolnoprzepustowego  $\tilde{h}$  dekompozycji układu wynoszą wówczas

$$\begin{cases} \tilde{h}_0 = -0.5 \cdot 0.25 & = -0.125 \\ \tilde{h}_1 = 1 \cdot 0.25 & = 0.25 \\ \tilde{h}_2 = (-0.5) \cdot 0.25 + 1 + (-0.5) \cdot 0.25 & = 0.75 \\ \tilde{h}_3 = 1 \cdot 0.25 & = 0.25 \\ \tilde{h}_4 = (-0.5) \cdot 0.25 & = -0.125 \end{cases} \quad (15.3)$$

natomiast filtr górnoprzepustowy przyjmuje wartości

$$\begin{cases} \tilde{g}_0 = -0.5 \\ \tilde{g}_1 = 1 \\ \tilde{g}_2 = -0.5 \end{cases}. \quad (15.4)$$

Schemat liftingu na podstawie współczynników predykcji i uaktualnienia determinuje współczynniki filtrów analizy jak i współczynniki filtrów syntezy.

Znając schemat przejścia współczynników predykcyjnych do współczynników falkowych, a co się z tym wiąże filtry analizy, filtry dualne - syntezy układu uzyskuje się w sposób następujący

1. Po wyznaczeniu wektorów predykcji  $\mathbf{p}$  i uaktualnienia  $\mathbf{u}$  przyjmuje się, że wektory te posiadają wartości przeciwne tj.  $-\mathbf{p}$  i  $-\mathbf{u}$ .
2. W dalszej kolejności współczynniki filtrów podlegają wymianie z  $\tilde{h}$  do  $g$  i  $\tilde{g}$  do  $h$ .
3. Ostatnim krokiem jest odwrócenie kolejności występowania współczynników filtrów.

Powyższą zależność wyznacza filtr górnoprzepustowy syntezy w postaci

$$\left\{ \begin{array}{l} g_0 = \tilde{h}_4 = -0.125 \\ g_1 = -\tilde{h}_3 = -0.25 \\ g_2 = \tilde{h}_2 = 0.75 \\ g_3 = -\tilde{h}_1 = -0.25 \\ g_4 = \tilde{h}_0 = -0.125 \end{array} \right. \quad (15.5)$$

natomiast filtr dolnoprzepustowy prezentuje się następująco

$$\left\{ \begin{array}{l} h_0 = p_0 = -\tilde{g}_2 = 0.5 \\ h_1 = \tilde{g}_1 = 1 \\ h_2 = p_1 = -\tilde{g}_0 = 0.5 \end{array} \right. \quad (15.6)$$

Schemat zależności współczynników predykcji i uaktualnienia nie zawsze jest symetryczny jak w przypadku schematu 15.1, jednakże pozwala zawsze i w sposób jednoznaczny otrzymać filtry biortogonalne dla danej operacji predykcji i uaktualnienia[72]. Sytuacja odwrotna nie daje takiej gwarancji. Aby z pary filtrów otrzymać wektor predykcji i uaktualnienia należy doprowadzić dwukanałowy schemat dekompozycji do reprezentacji polifazowej. Następnie, macierz polifazową przekształca się za pomocą dzielenia wielomianów Laurenta (rozdz. 13.2) do postaci iloczynowej 13.18 z której wyodrębnia się bloki predykcji i uaktualnienia[58].

Ponieważ operacja dzielenia wielomianów Laurenta jest niejednoznaczna, faktoryzacja zespołu filtrów falkowych do postaci predykcyjnej może mieć więcej niż jedno rozwiązanie (rozdz. 13.4), ponadto może wymagać więcej niż jedną parę operacji predykcji i uaktualnienia.

Poniżej zaprezentowano przykład wyliczenia (rys. 15.3) filtrów górno i dolnoprzepustowych jak i sposób wykreślenia funkcji falkowych i skalujących(rys. 15.4, rys. 15.5) na podstawie uprzednio wyliczonych wektorów predykcji i uaktualnienia (rys. 15.2).

```
<< SchematLiftinguZAproksymacjaMNK`
```

```
?PredyktorAproksymacji
```

```
PredyktorAproksymacji[zakresArgumentow, tablicaDanych, PhiBaza, Waga, Rząd, tryb]
```

```
Wyliczenie danych wektora predykcji schematu liftingu z
```

```
trybem wyświetlania (predykcja, detal, predykcjaLeniwa, detalLeniwy, współczynnikiBazy)
```

```
rzad = 2;
```

```
waga[x_] := 1;
```

```
baza[x_] = {1, x};
```

```
sygnal = {1, 2, 3, 4};
```

```
zakres = {0, Pi / 4};
```

```
predict = PredyktorAproksymacji[zakres, sygnal, baza, waga, rzad, "predykcja"][[1]];
update = predict / 2;
```

```
{predict, update} // TableForm
```

```
TableForm=
```

```
0.5 0.5
```

```
0.25 0.25
```

Rysunek 15.2: Wyliczanie wartości współczynników wektora predykcji i uaktualnienia o tej samej długości

### ? WyznaczanieWspolczynnikowFiltruZWektorowPredykcjiUaktualnienia

WyznaczanieWspolczynnikowFiltruZWektorowPredykcjiUaktualnienia[wektorPredykcji,  
wektorUaktualnienia, typWyjsciuowy, przesunieciePredyktora, przesuniecieUaktualnienia]  
WyznaczanieWspolczynnikowFiltruZWektorowPredykcjiUaktualnienia[ wektorPredykcji, wektorUaktualnienia, typWyjsciuowy]  
Funkcja wyznaczajaca za pomoca wspolczynnikow predykcji i uaktualnienia wspolczynniki filtru  
postaci tablicowej analizy (typWyjsciuowy:wspolAnalizaFiltrTab) i syntezy (typWyjsciuowy:wspolSyntezaFiltrTab)

```
WyznaczanieWspolczynnikowFiltruZWektorowPredykcjiUaktualnienia[predict, update,  
"wspolAnalizaFiltrTab"] // TableForm
```

```
TableForm=  
-0.125    0.25    0.75    0.25    -0.125  
-0.5      1      -0.5
```

```
WyznaczanieWspolczynnikowFiltruZWektorowPredykcjiUaktualnienia[predict, update,  
"wspolSyntezaFiltrTab"] // TableForm
```

```
TableForm=  
0.5      1      0.5  
-0.125  -0.25  0.75  -0.25  -0.125
```

Rysunek 15.3: Wylizanie wspolczynnikow filtru analizy  $\tilde{h}$ ,  $\tilde{g}$  i syntezy  $h$ ,  $g$



## ?PrezentacjaFunkcjiFalkowych

PrezentacjaFunkcjiFalkowych[wektorPredykcji, wektorUaktualnienia, typFunkcji, liczbaIteracji, zakres]

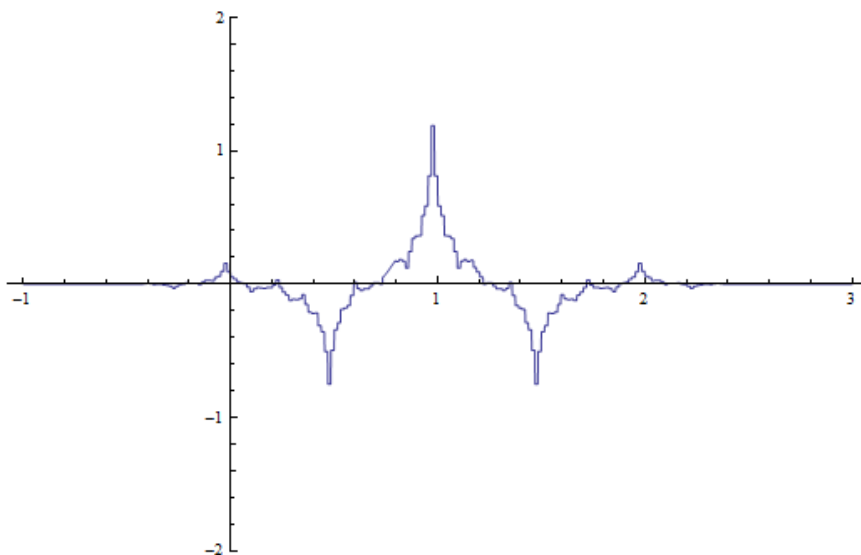
Wykresianie funkcji skalujących i falkowych za pomocą współczynników predykcji (predict) i

współczynników uaktualnienia (update). Argument typFunkcji określa typ funkcji wykreslanej (funkcjaSkalujacaAnalizy,

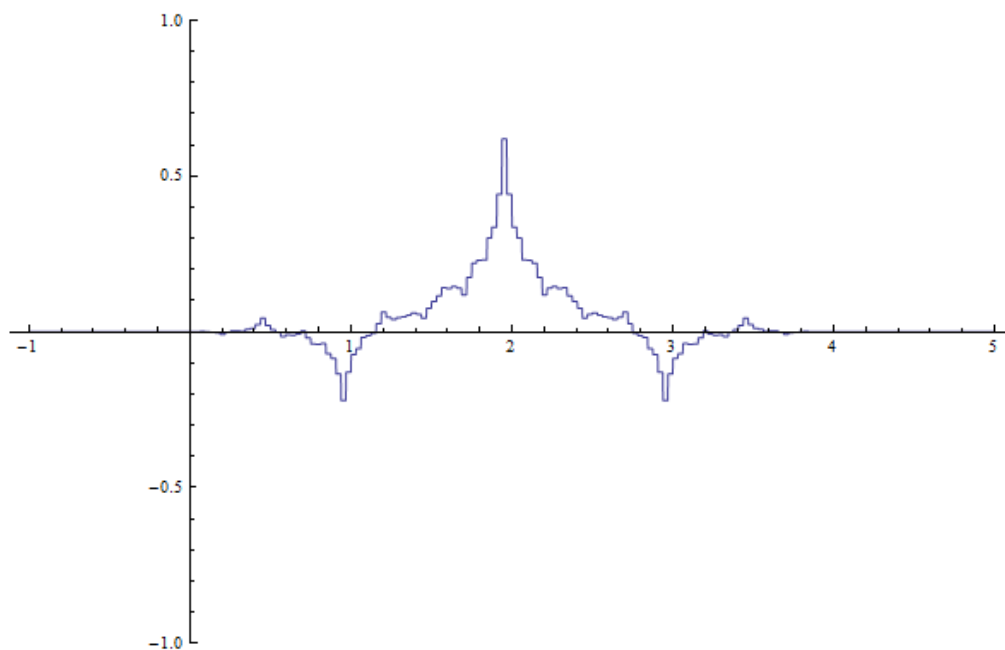
funkcjaFalkowaAnalizy, funkcjaSkalujacaSyntezy, funkcjaFalkowaSyntezy), liczbaIteracji – liczbe iteracji

wykorzystywana do celow przyblizen funkcji, zakres – tablica zakresu ukladu wspolrzednych {{xmin,xmax},{ymin,ymax}}.

```
PrezentacjaFunkcjiFalkowych[predict, update, "funkcjaFalkowaAnalizy", 4,
{{-1, 3}, {-2, 2}}]
```

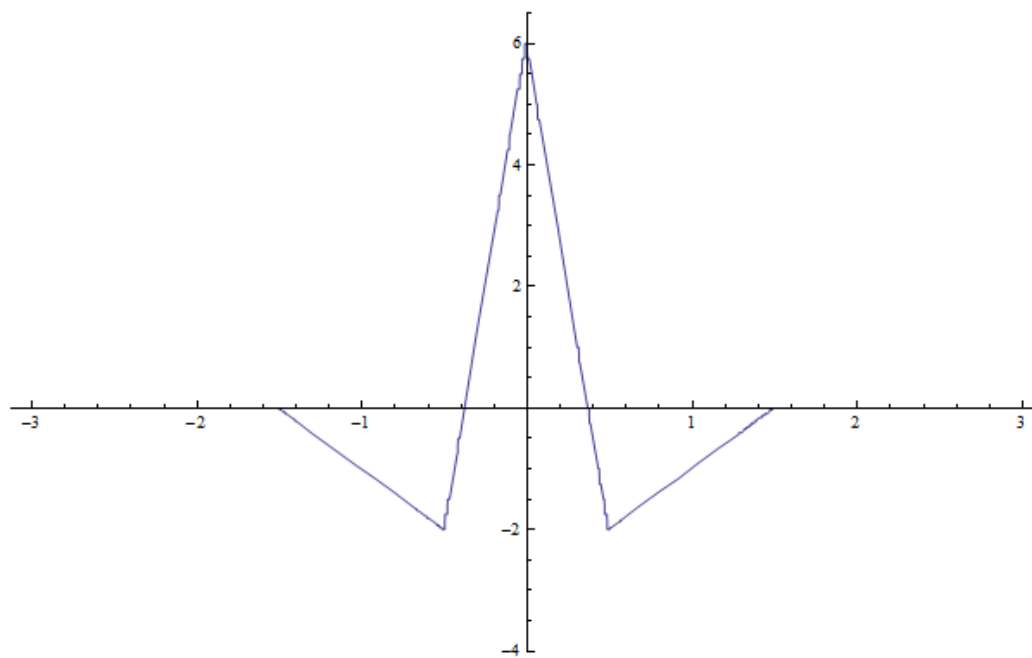


```
PrezentacjaFunkcjiFalkowych[predict, update, "funkcjaSkalujacaAnalizy", 4,
{{-1, 5}, {-1, 1}}]
```

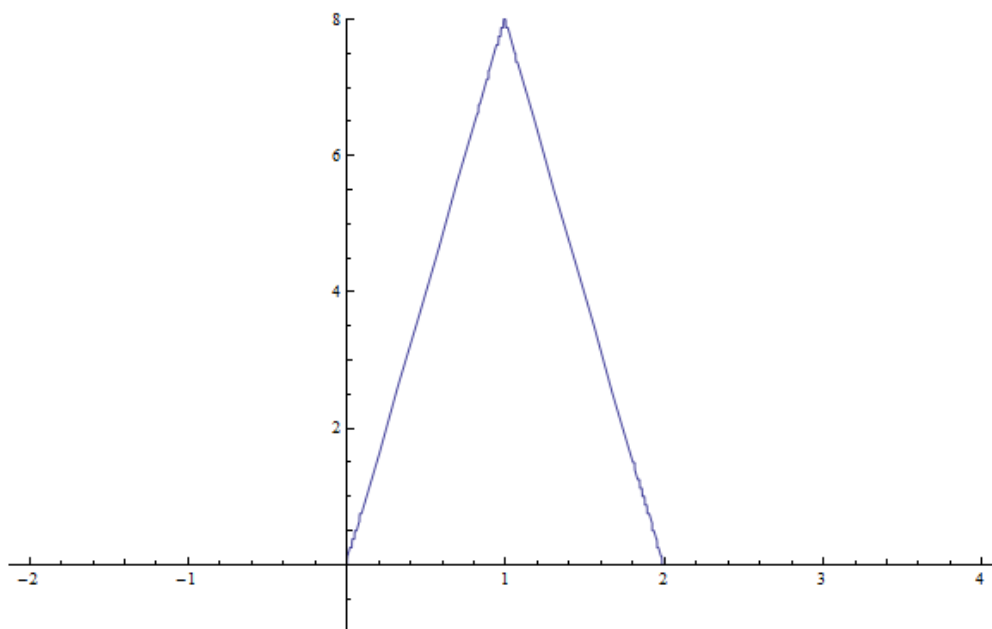


Rysunek 15.4: Analiza funkcji - graficzna reprezentacja funkcji falkowej i skalującej

```
PrezentacjaFunkcjiFalkowych[predict, update, "funkcjaFalkowaSyntezy", 4,  
{-3, 3}, {-4, 6.5}]
```



```
PrezentacjaFunkcjiFalkowych[predict, update, "funkcjaSkalujacaSyntezy", 5,  
{-2, 4}, {-1, 8}]
```



Rysunek 15.5: Synteza funkcji - graficzna reprezentacja funkcji falkowej i skalującej



# Zastosowanie schematu liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów w bloku predykcji

---

Transformata falkowa z wykorzystaniem aproksymacji metodą najmniejszych kwadratów (MNK) umożliwia zdekomponować funkcję na ciąg detali które przechowują informację o szczegółach funkcji dla danego poziomu rozdzielczości. Poprzez sukcesywne usuwanie szczegółów otrzymuje się aproksymację funkcji zachowując jej kształt. Pełna informacja o funkcji oryginalnej zachowana jest w zbiorze złożonym ze zbiorów wszystkich detali jego rozwinięcia i zbioru aproksymacji najniższego poziomu rozdzielczości.

Dzięki operacji predykcji, przestrzeń detali można wykorzystać do weryfikacji funkcji pod względem jej regularności, ciągłości lub usuwania szumu badanej funkcji. Analiza wielorozdzielcza z wykorzystaniem falkowej dekompozycji funkcji pozwala m. in. na sprawdzenie w dziedzinie czasu, w którym momencie zaszła zmiana charakteru funkcji [8, 53].

Przykład z rys.16.2 ilustruje przebieg funkcji który na wykresie prezentuje się jako funkcja harmoniczna. W rzeczywistości funkcja ta posiada trzy punkty nieciągłości występujące w postaci skokowej z przesunięciem o długości 0.01 dla argumentów -5, 0 i 5(rys.16.1).

```
f[x_] := Sin[x - 0.01] /; x < -5;  
f[x_] := Sin[x] /; (x ≥ -5 && x < 0);  
f[x_] := Sin[x + 0.01] /; (x ≥ 0 && x < 5);  
f[x_] := Sin[x + 0.02] /; x ≥ 5;
```

Rysunek 16.1: Wykrywanie nieciągłości funkcji - dane

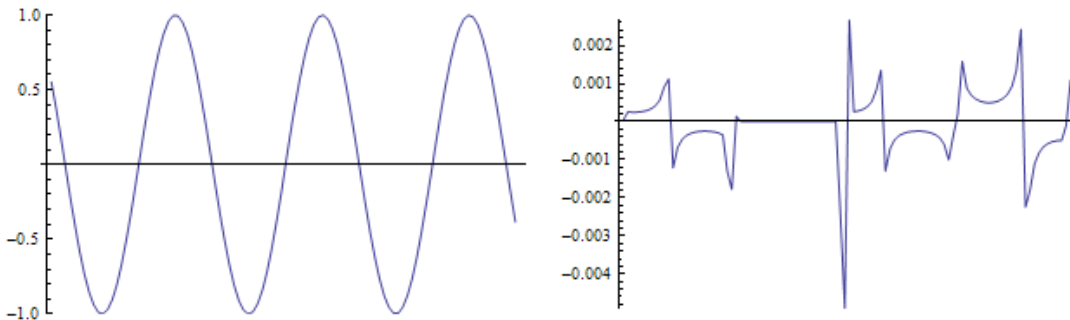
? SchematLiftinguMNFunkcjiPrezentacja

SchematLiftinguMNFunkcjiPrezentacja[zakresArgumentow, tablicaDanych, PhiBaza, Waga, Rząd, tryb]  
 Prezentacja danych schematu liftingu z aproksymacja MNK w bloku  
 predykcji z trybem wyswietlania (detal, aproksymacja, oryginal, probkiParzyste, probkiNieparzyste)

(\* Rząd musi być conajwyżej 2 razy mniejszy od długości tablicy danych \*)

```
Rząd = 4;
Waga[x_] := 1;
PhiBaza[x_] := {1, Sin[x]};
zakresArgumentow = {-10, 10};
tablicaDanych = Table[f[x], {x, -10, 10, 0.1}];
```

```
GraphicsRow[
  {SchematLiftinguMNFunkcjiPrezentacja[zakresArgumentow, tablicaDanych, PhiBaza, Waga, Rząd,
    "aproksymacja"],
    SchematLiftinguMNFunkcjiPrezentacja[zakresArgumentow, tablicaDanych, PhiBaza, Waga, Rząd, "detal"]}];
```



Rysunek 16.2: Wykrywanie nieciągłości funkcji (reprezentacja zgrubna, reprezentacja szczegółów)

Istotną rzeczą podczas wykrywania nieciągłości funkcji jest dobór odpowiedniego wektora bazy aproksymacji. Przyjmując bazę funkcji aproksymującej  $[1, \sin(x)]$  (rys. 16.2), można zauważyć, że charakterystyka reprezentacji detali niesie z sobą informację o zakresie gdzie funkcja aproksymowana pokrywa się z funkcją aproksymującą (przedział z funkcją zerową reprezentacji szczegółów).

Wynika z tego, że funkcja skalująca na wyższym poziomie rozdzielczości jest silnie skorelowana z funkcją w dziedzinie  $\leq -5, 0$ .

Optymalne dopasowanie funkcji aproksymującej do funkcji badanej dla wybranego zakresu dziedziny, nie zawsze pozwala na analizę konkretnego problemu.

Dla powyższej bazy funkcji aproksymującej (rys. 16.2), wykres detali nie zwraca na przykład jednoznacznej informacji dotyczącej nieciągłości na całym badanym przedziale.

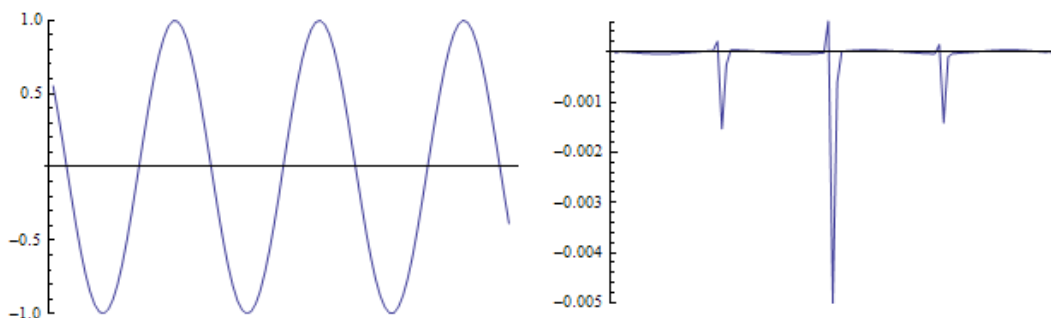
Rozwiązaniem problemu może być takie dopasowanie wektora bazy funkcji aproksymującej, aby mogła ona z dobrym przybliżeniem aproksymować funkcję dla całego zakresu (rys. 16.3).

Spowoduje to, że reprezentacja detali wykaże punkty nieciągłości, nie przekazując jednocześnie danych o charakterze funkcji w poszczególnych przedziałach dziedziny.

Przykładem takim może być zastosowanie interpolacji w bloku predykcji schematu liftingu z wykorzystaniem wielomianu stopnia 3-go i rzędu predykcji stopnia 4-go (rys. 16.3).

```
(* Rząd musi być conajwyżej 2 razy mniejszy od długości tablicy danych *)
Rzad = 4;
Waga[x_] := 1;
PhiBaza[x_] := {1, x, x^2, x^3};
zakresArgumentow = {-10, 10};
tablicaDanych = Table[f[x], {x, -10, 10, 0.1}];

GraphicsRow[
  {SchematLiftinguMNKFunkcjiPrezentacja[zakresArgumentow, tablicaDanych, PhiBaza, Waga,
    Rzad, "aproksymacja"], SchematLiftinguMNKFunkcjiPrezentacja[zakresArgumentow,
    tablicaDanych, PhiBaza, Waga, Rzad, "detal"]}
]
```



Rysunek 16.3: Wykrywanie nieciągłości funkcji

Innym przykładem zastosowania schematu liftingu z wykorzystaniem aproksymacji MNK, może być badanie zmiany charakteru funkcji dla której na przykład w danej chwili następuje gwałtowna zmiana częstotliwości.

Analiza takiego przypadku w praktyce może mieć zastosowanie na przykład podczas badania ruchów sejsmicznych lub poprawności działania turbin podczas chłodzenia łopatek powietrzem[3].

Zgodnie z analizą zmian charakteru funkcji w przestrzeni detali z rys.16.5 można wywnioskować, podobnie jak w przypadku nieciągłości, niedopasowanie funkcji skalującej do przebiegu funkcji w punkcie zmiany.

```
h[x_] := Sin[x] /; x < 0;
h[x_] := -Cos[2 (x - Pi / 2)] /; x ≥ 0;
```

Rysunek 16.4: Wykrywanie zmiany charakterystyki funkcji I - dane

```
(* Rząd musi być conajwyżej 2 razy mniejszy od długości tablicy danych *)
```

```
Rzad = 4;
```

```
Waga[x_] := 1;
```

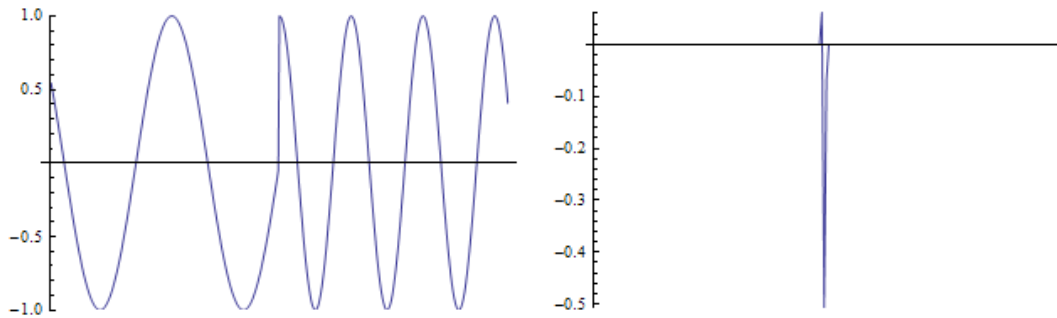
```
PhiBaza[x_] := {1, x, x^2, x^3};
```

```
tablicaDanych = Table[h[x], {x, -10, 10, 0.05}];
```

```
zakresArgumentow = {-10, 10};
```

```
GraphicsRow[
```

```
{SchematLiftinguMNFunkcjiPrezentacja[zakresArgumentow, tablicaDanych, PhiBaza, Waga,
  Rzad, "oryginal"], SchematLiftinguMNFunkcjiPrezentacja[zakresArgumentow, tablicaDanych,
  PhiBaza, Waga, Rzad, "detal"]}]
```



Rysunek 16.5: Wykrywanie zmiany charakteru funkcji I

Zastosowana aproksymacja MNK do celów przekształceń, w przeciwieństwie do interpolacji, pozwala na zmianę długości wektora bazy  $\Psi$  funkcji aproksymującej bez zmiany rzędu predyktora tzn. liczby próbek podlegających aproksymacji. Konsekwencją tego jest możliwość dopasowania wielomianu, o znacznie mniejszym stopniu aniżeli rząd wektora predykcji do funkcji badanej.

Wykrywanie zmiany charakteru funkcji z rys. 16.5 umożliwia zidentyfikowanie obszaru w którym następuje zmiana charakteru funkcji badanej, jednakże nie pozwala zweryfikować czy dana funkcja odzyskuje stan poprzedni.

Rozwiązaniem tego typu problemu jest takie dopasowanie funkcji aproksymowanej do funkcji aproksymującej która pozwoli na wyodrębnienie informacji o dziedzinie w której funkcja posiada inny charakter aniżeli funkcja przewidywana.

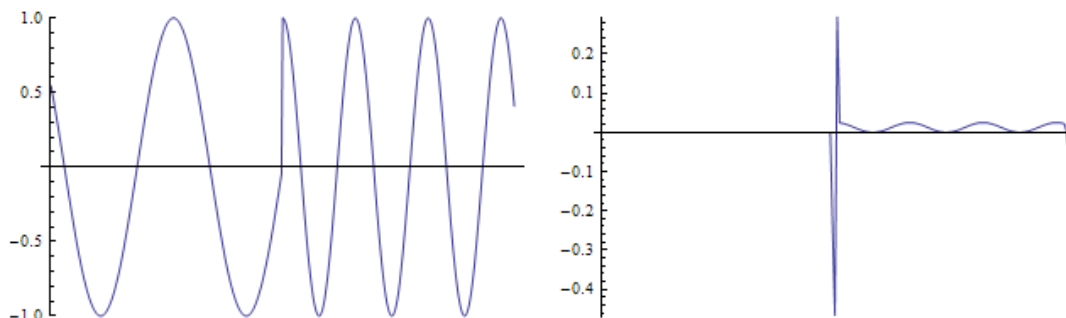
Przykład z rys. 16.6 ustala czy funkcja po przekroczeniu punktu zmiany powróciła do stanu początkowego, o pierwotnym kształcie.

```

Rzad = 4;
Waga[x_] := 1;
PhiBaza[x_] := {1, Sin[x]};
tablicaDanych = Table[h[x], {x, -10, 10, 0.05}];
zakresArgumentow = {-10, 10};

GraphicsRow[
  {SchematLiftinguMNKFunkcjiPrezentacja[zakresArgumentow, tablicaDanych, PhiBaza, Waga,
    Rzad, "oryginal"], SchematLiftinguMNKFunkcjiPrezentacja[zakresArgumentow, tablicaDanych,
    PhiBaza, Waga, Rzad, "detal"]}

```



Rysunek 16.6: Wykrywanie zmiany charakteru funkcji II

Często w praktyce na przykład podczas badania szeregów czasowych, próbki funkcji otrzymuje się w sposób empiryczny, konsekwencją tego są tzw. błędy pomiarowe które doprowadzają nierzadko do zaszumienia, zniekształcenia wykresu funkcji [33].

Dzięki aproksymacji, funkcja może być poddana również wygładzeniu co też może mieć znaczny wpływ na poprawienie poziomu regularności czy różniczkowalności funkcji [3].

Zbiór wykresów z rys. 16.8 prezentuje zdekomponowany wykres funkcji sinusoidalnej o znacznym, losowym zaszumieniu (zdefiniowany na rys.16.7).

Odpowiednie dopasowanie funkcji aproksymowanej do funkcji docelowej (niezaszumionej) pozwala na uzyskanie charakterystyki funkcji o kształcie znacznie zbliżonym czy też dokładnym do kształtu funkcji docelowej.

```

(* Rzad musi być conajwyżej 2 razy mniejszy od długości tablicy danych *)
Rzad = 4;
Waga[x_] := 1;
PhiBaza[x_] := {1, Sin[x]};
tablicaDanych = Table[Sin[x] + RandomReal[{-0.1, 0.1}], {x, -10, 10, 0.01}];
zakresDanych = {-10, 10};

```

Rysunek 16.7: Usuwanie szumu funkcji - dane



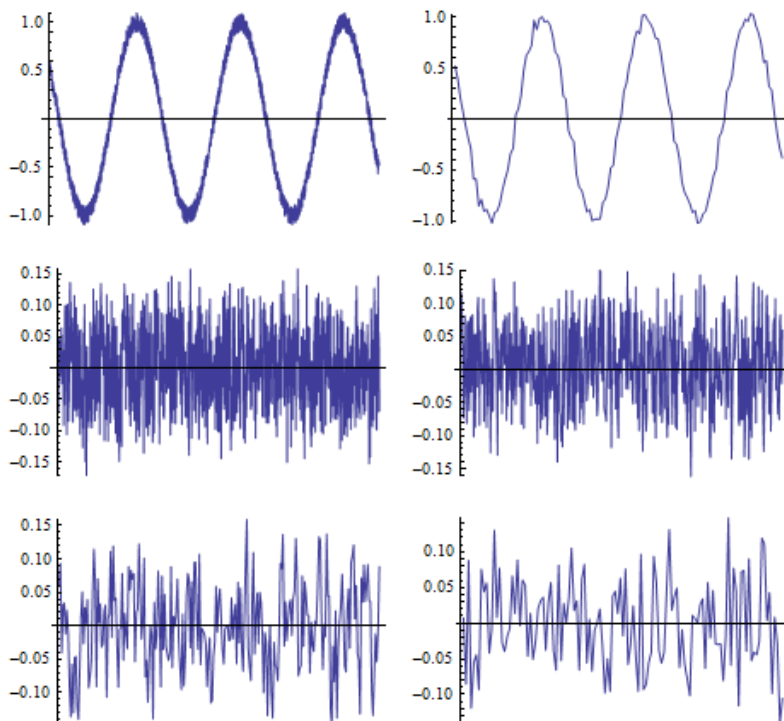
```

tablicaDanych2 = tablicaDanych;

For[i = 1, i ≤ 4,
  detal[i] = SchematLiftinguMNFunkcjiPrezentacja[zakresDanych, tablicaDanych2,
    PhiBaza, Waga, Rzad, "detal"];
  tablicaDanych2 = SchematLiftinguMNFunkcjiPrezentacja[zakresDanych,
    tablicaDanych2, PhiBaza, Waga, Rzad, "aproksymacja"];
  i++;
];

GraphicsGrid[
  {{SchematLiftinguMNFunkcjiPrezentacja[zakresDanych, tablicaDanych, PhiBaza,
    Waga, Rzad, "oryginal"],
    SchematLiftinguMNFunkcjiPrezentacja[zakresDanych, tablicaDanych2, PhiBaza,
    Waga, Rzad, "oryginal"]},
  {detal[1], detal[2]}, {detal[3], detal[4]}}]

```



Rysunek 16.8: Usuwanie szumu funkcji

## Część VI

Analiza obrazu za pomocą schematu  
liftingu z wykorzystaniem aproksymacji  
metodą najmniejszych kwadratów  
w bloku predykcji



## Rozdział 17

---

# Schemat liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów

---

W poprzedniej części rozprawy, schemat liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów (MNK) został ograniczony do przestrzeni jednowymiarowej m.in. ze względu na wnikliwą analizę i zobrazowanie zależności pomiędzy jej dyskretną i predykcyjną implementacją transformacji falkowej. W części bieżącej operacja ta zostanie przeniesiona na wymiar o jeden wyższy tzn. na obszar dwuwymiarowy w celu analizy obrazu.



Rysunek 17.1: Analiza wielorozdzielcza obrazu

Analizę wielorozdzielczą dwuwymiarowych przestrzeni dyskretnych realizuje się za pomocą dwu-etapowej transformacji jednowymiarowej.

W pierwszym etapie dokonuje się transformacji na każdym wierszu analizowanego obrazu otrzymując w konsekwencji dwie reprezentacje: zgrubną i szczegółową.

Drugim etapem jest analogiczna operacja na kolumnach dla obrazów powstałych w etapie pierwszym.

Wynikiem takiego przekształcenia są cztery obrazy: jedna reprezentacja zgrubna i trzy reprezentacje przestrzeni detali (pozioma, pionowa i diagonalna)[3, 55, 57].

Na rysunku 17.1 można zauważyć ścisłą zależność nazw przestrzeni szczegółów obrazu od jej graficznej realizacji. Reprezentacja detali pozioma (prawy górny obraz) uwypukla krawędzie poziome obrazu oryginalnego, w reprezentacji pionowej (lewy dolny obraz) - wyróżnione są krawędzie pionowe, zaś dla diagonalnej - szczegóły występujące w kierunku przekątnej obrazu.

Dekompozycja obrazu za pomocą analizy wielorozdzielczej ma swoje zastosowanie m.in. podczas redukcji szumu, wykrywaniu nieciągłości czy też badaniu wrażliwości na zerowanie współczynników transformaty która może być wykorzystana podczas kompresji obrazu [3].

Wymienione typy operacji wykorzystywane zostaną podczas dalszych badań dotyczących przekształceń falkowych przestrzeni dwuwymiarowych za pomocą schematu liftingu z wykorzystaniem aproksymacji MNK w bloku predykcji.

## 17.1 Redukcja szumu obrazu

Efektywną operację redukcji szumu można przyjąć w sytuacji, gdy znana jest charakterystyka funkcji badanej, pozwalając optymalnie dopasować wektor funkcji bazowych zwany wektorem bazy, wchodzący w skład funkcji aproksymującej.

Redukcję szumu realizuje się w następujących etapach:

- dekompozycja obrazu
- eliminacja progowa współczynników
- rekonstrukcji obrazu

Najczęstszymi metodami eliminacji progowej współczynników wykorzystywanymi w analizie wielorozdzielczej są [3]

- eliminacja progowa twarda
- eliminacja progowa miękka

**Eliminacja progowa twarda** (wzór 17.1) polega na zerowaniu tych współczynników których wartość bezwzględna jest mniejsza lub równa ustalonej wartości zwanej wartością progową  $\delta$ , nie zmieniając przy tym pozostałych wartości.

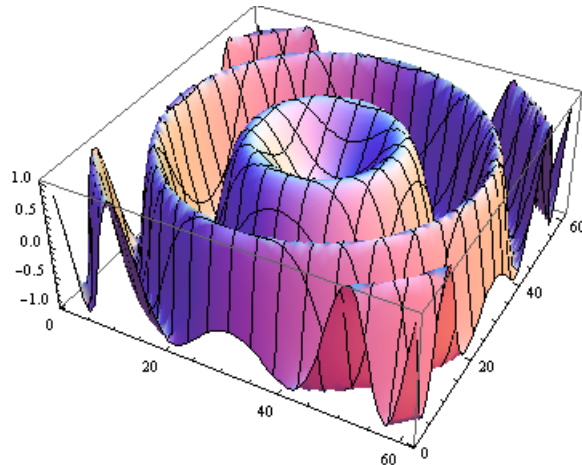
$$y(t) = \begin{cases} x(t), & |x(t)| > \delta \\ 0, & |x(t)| \leq \delta \end{cases} \quad (17.1)$$

**Eliminacja progowa miękka** (wzór 17.2) wykorzystuje własności eliminacji progowej twardej, zmniejszając jednocześnie wartości bezwzględne pozostałych współczynników o zadaną wartość progową  $\delta$ .

$$y(t) = \begin{cases} \text{sign}(x(t)) (|x(t)| - \delta), & |x(t)| > \delta \\ 0, & |x(t)| \leq \delta \end{cases} \quad (17.2)$$

Schemat liftingu z wykorzystaniem aproksymacji MNK, w przeciwieństwie do interpolacji, pozwala na przybliżanie obrazu za pomocą funkcji aproksymujących których węzły aproksymacji nie muszą pokrywać się z wartościami aproksymowanego obrazu.

W przypadku gdy dane wartości funkcji są odszumiane, wymagane jest aby kształt funkcji odszumionej wykazywał się większą regularnością aniżeli funkcja zaszumiona. Do tego celu nie wskazane zazwyczaj jest aby funkcja aproksymująca przebiegała przez wszystkie wartości funkcji aproksymowanej, ponadto powinna je efektywnie przybliżać do funkcji o prognozowanym kształcie.



Rysunek 17.2: Fragment funkcji  $\sin(x^2 + y^2)$

```
funkcja = Table[N[Sin[x^2] Cos[y^2] + Sin[y^2] Cos[x^2]], {x, -Pi, Pi, Pi/30}, {y, -Pi, Pi, Pi/30}];
funkcjaOdszumiana = Table[funkcja[[i, j]] + RandomReal[{-0.2, 0.2}], {x, -Pi, Pi, Pi/30}, {y, -Pi, Pi, Pi/30}];
Rzad = 8;
Waga[x_] := 1;
PhiBaza1[x_] = {1, Cos[x^2], Sin[x^2]};
PhiBaza2[x_] = {1, x, x^2, x^3, x^4, x^5, x^6, x^7};
PhiBaza3[x_] = {1, x, x^2};
zakresArgumentow = {-Pi, Pi};

dane = SchematLiftinguMNKObrazu[zakresArgumentow, funkcjaOdszumiana, PhiBaza, Waga, Rzad, 2, "miekka"];
dane2 = SchematLiftinguMNKObrazu[zakresArgumentow, funkcjaOdszumiana, PhiBaza2, Waga, Rzad, 2, "miekka"];
dane3 = SchematLiftinguMNKObrazu[zakresArgumentow, funkcjaOdszumiana, PhiBaza3, Waga, Rzad, 2, "miekka"];
```

Rysunek 17.3: Redukcja szumu obrazu - przygotowanie danych

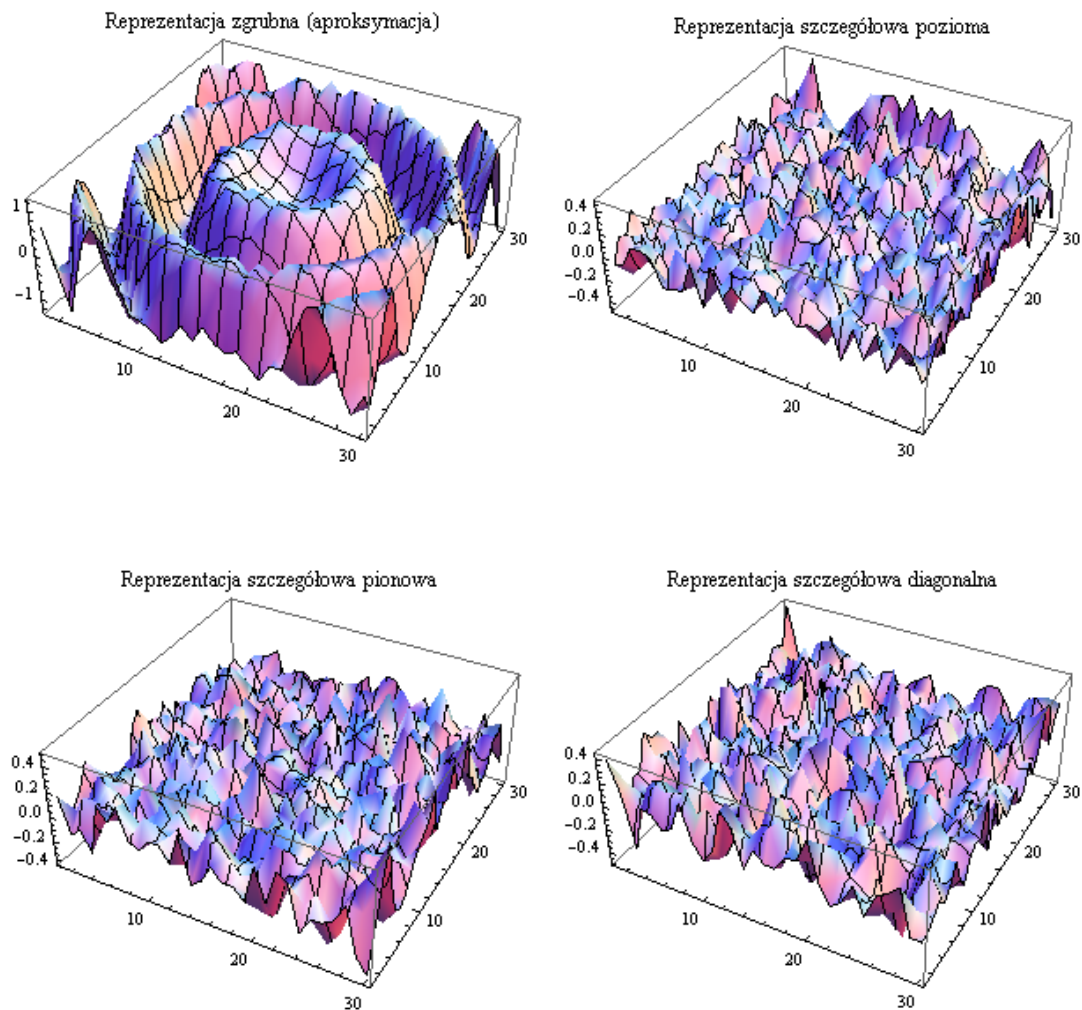
Do celów badań i analizy obrazów wykorzystywana została aproksymacja funkcji  $\sin(x^2 + y^2) \equiv \sin(x^2)\cos(y^2) + \cos(x^2)\sin(y^2)$  przedstawiona na rys. 17.2.

Parametry aproksymacji zostały określone na rys. 17.3 gdzie **zakresArgumentow** określa przedział argumentów dla wartości danych **funkcjaOdszumiana** podlegających aproksymacji.

**PhiBaza1**, **PhiBaza2** i **PhiBaza3** wyznaczają funkcje bazowe funkcji aproksymującej, **Waga** - wagę funkcji aproksymującej, zaś **Rzad** opisuje rząd predyktora (liczbę punktów wykorzystywanych do predykcji).

Wielkość podziałki dla zakresu argumentów wyliczana jest na podstawie liczby próbek (wartości) wchodzących w skład aproksymacji.

```
analiza = AnalizaObrazuSchematemLiftinguMKNK[zakresArgumentow, funkcjaOdszumiana, PhiBaza1,
Waga, Rzad, 0.0, "miękki"];
GraphicsGrid[
{{ListPlot3D[analiza[[1]], PlotLabel -> "Reprezentacja zgrubna (aproksymacja)"},
ListPlot3D[analiza[[2]], PlotLabel -> "Reprezentacja szczegółowa pozioma"}},
{ListPlot3D[analiza[[3]], PlotLabel -> "Reprezentacja szczegółowa pionowa"},
ListPlot3D[analiza[[4]], PlotLabel -> "Reprezentacja szczegółowa diagonalna"}]}
```



Rysunek 17.4: Przykład analizy wielorozdzielczej obrazu

Pierwszym etapem podczas usuwania szumu jest dekompozycja obrazu która dzieli obraz na cztery części: jedną reprezentację aproksymacji i trzy reprezentacje szczegółów (rys. 17.4).

Następnie wykonuje się eliminacja progową z określonym progiem i typem eliminacji gdzie dla omawianego przykładu został ustalony próg 2.0.

Wartość progu 2.0 powoduje, że wszystkie próbki obrazów detali są zerowane, ponieważ wartości  $\sin(x^2 + y^2)$  należą do przedziału  $< -1, 1 >$ .

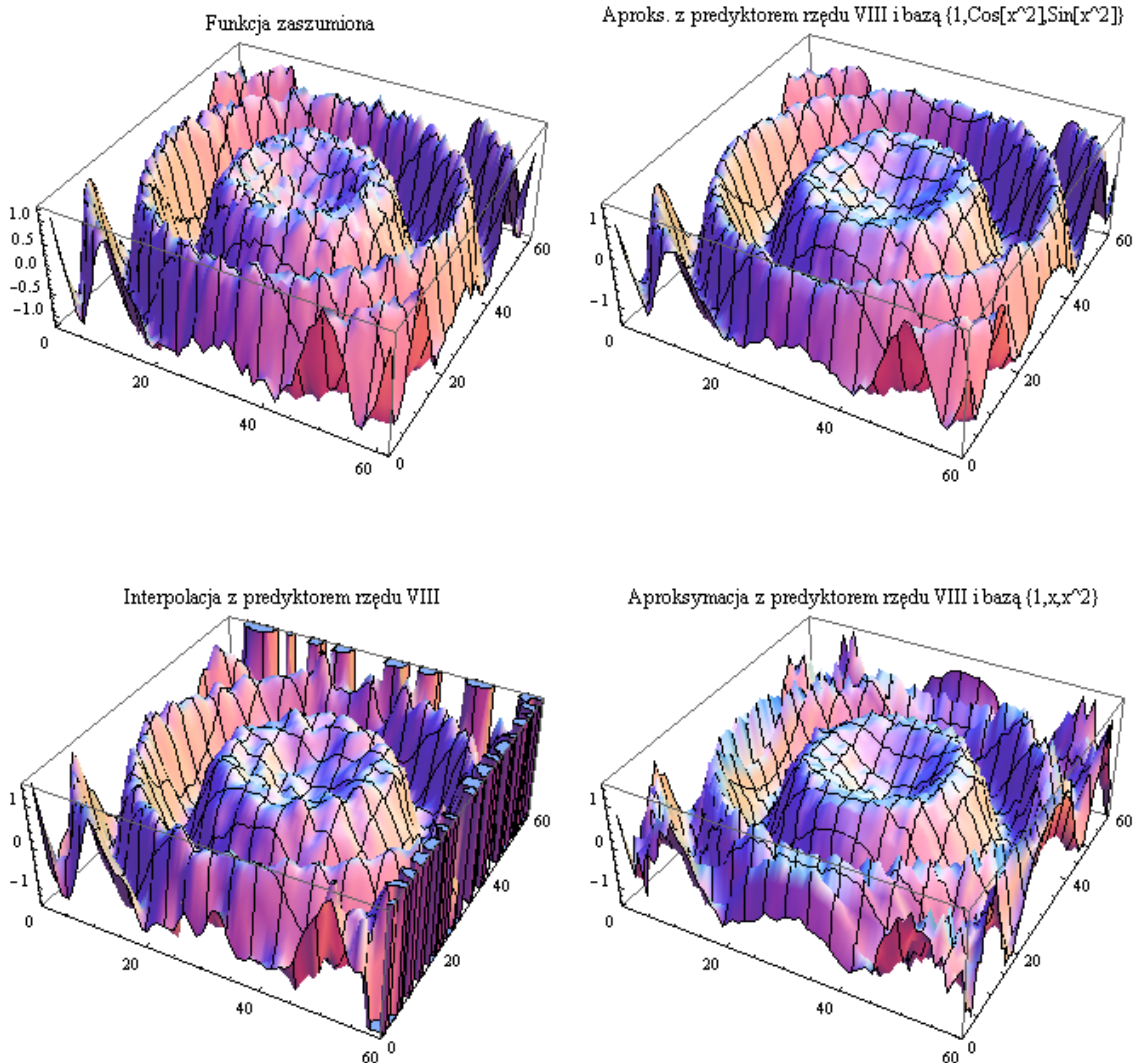
Ze względu na zerowanie wszystkich próbek wybór typu eliminacji progowej nie ma wpływu na



wynik eliminacji progowej.

Na rys. 17.5 zostało zaprezentowane odszumianie obrazu z efektywnością uzależnioną od doboru wektora bazy funkcji aproksymującej.

```
GraphicsGrid[
  {{ListPlot3D[funkcjaOdszumiana, PlotLabel -> "Funkcja zaszumiona" ],
    ListPlot3D[dane, PlotRange -> {{0, 60}, {0, 60}, {-1.5, 1.5}},
      PlotLabel -> "Aproks. z predyktorem rzędu VIII i bazą {1,Cos[x^2],Sin[x^2]}",
    ListPlot3D[dane2, PlotRange -> {{0, 60}, {0, 60}, {-1.5, 1.5}}, PlotLabel -> "Interpolacja z predyktorem rzędu VIII" ],
    ListPlot3D[dane3, PlotRange -> {{0, 60}, {0, 60}, {-1.5, 1.5}},
      PlotLabel -> "Aproksymacja z predyktorem rzędu VIII i bazą {1,x,x^2}" ]}}
```



Rysunek 17.5: Porównanie wyników podczas usuwania szumu obrazu

Na podstawie otrzymanych wyników (rys. 17.6, 17.7) można wywnioskować, że aproksymacja bazą  $[1, \cos(x^2), \sin(x^2)]$  (**PhiBaza1**) stanowi najlepsze, spośród wybranych baz, przybliżenie funkcji niezaszumionej  $\sin(x^2 + y^2)$  powodując jej znaczne wygładzenie, zwiększając jednocześnie stopień regularności (rys. 17.5).

Według zastosowanego wskaźnika efektywności podczas usuwania szumu jakim jest wariancja pró-

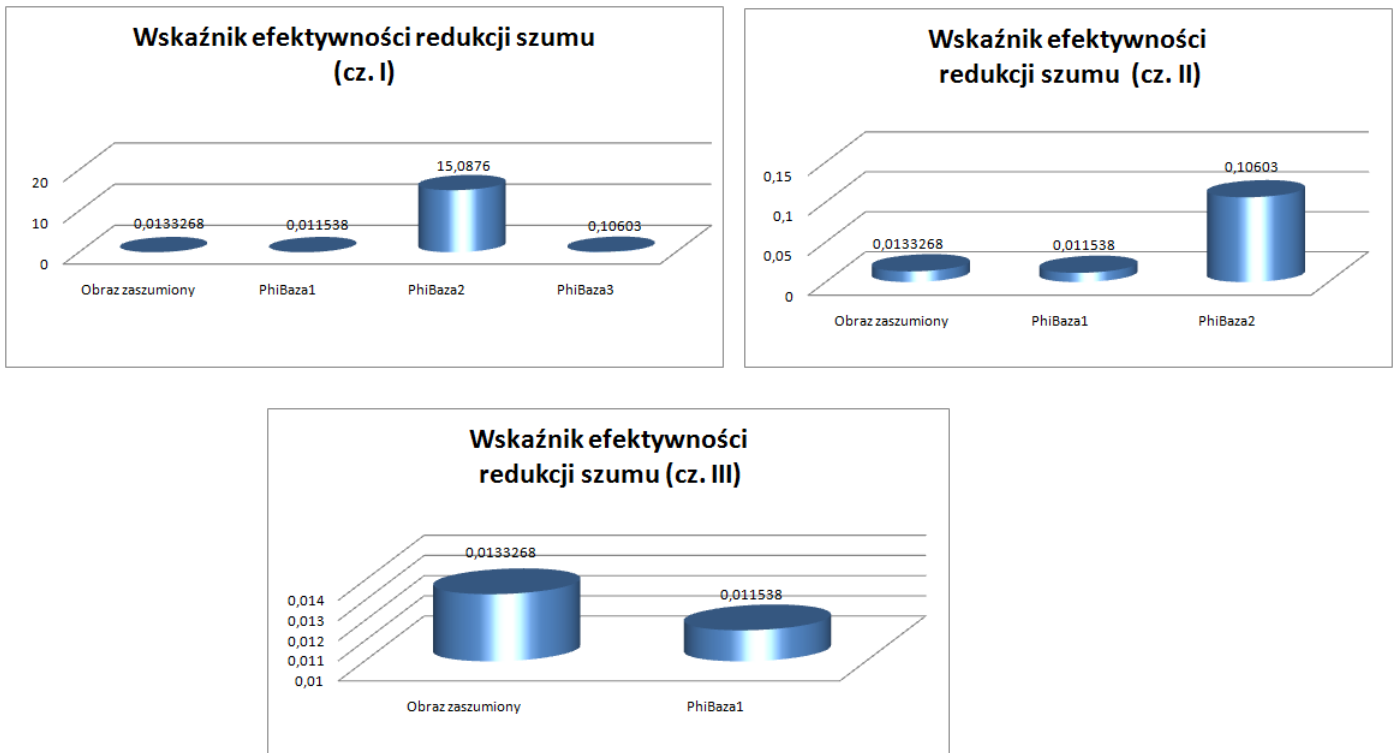


bek funkcji odsumowanej względem funkcji niezasumowanej wynika (rys. 17.6), że operacja aproksymacji dla bazy **PhiBaza1** usunęła szum z 0.0133268 do 0.011538, natomiast dwie pozostałe znacznie zwiększyły stopień zaszumienia do 15.0876 w przypadku interpolacji z predyktorem rzędu VIII (**PhiBaza2**) i 0.10603 w przypadku aproksymacji trójmianem kwadratowym (**PhiBaza3**).

Analizując rezultat (rys. 17.6, 17.7) otrzymuje się wniosek, że efektywność usuwania szumu nie wzrasta wraz ze wzrostem stopnia wielomianu aproksymującego. Optymalizacja redukcji szumu polega przede wszystkim na dopasowaniu wektora bazy funkcji aproksymującej do kształtu funkcji docelowej, niezasumowanej.

```
{MiaraEfektywnosciObrazu[funkcjaOdszumiana, funkcja], MiaraEfektywnosciObrazu[dane, funkcja]},
{MiaraEfektywnosciObrazu[dane2, funkcja], MiaraEfektywnosciObrazu[dane3, funkcja]} // TableForm
//TableForm=
0.0133268 0.011538
15.0876 0.10603
```

Rysunek 17.6: Wskaźnik efektywności redukcji szumu



Rysunek 17.7: Wskaźnik efektywności redukcji szumu - postać graficzna

## 17.2 Usuwanie nieciągłości obrazu

W analizie funkcji ciągłych wyróżnia się dwa rodzaje nieciągłości [20]

- nieciągłość pierwszego rodzaju - typu luka, określa się ją jako nieciągłość pierwszego rodzaju jeśli granica funkcji  $f$  w punkcie  $c$  jest różna od  $f(c)$
- nieciągłość drugiego rodzaju występuje wtedy, jeśli nie istnieje co najmniej jedna jednostronna granica lub co najmniej jedna jednostronna granica jest nieskończona

W celu eliminacji, analizy czy też wykrycia skoków spowodowanych na przykład błędami pomiarowymi lub zakłóceniami można wykorzystać filtrację wykorzystywaną w transformacie falkowej przy użyciu aproksymacji MNK.

W prezentacji wykorzystany zostanie obraz którego charakterystykę można przedstawić wzorem określonym w środowisku Mathematica jako *funkcjaNieciaglosci[x\_,y\_]* (rys. 17.8).

```
Clear[funkcjaNieciaglosci];
funkcjaNieciaglosci[x_, y_] := Sin[x] Cos[y] /; x^2 + y^2 > 1 && x^2 + y^2 < 1.3;
funkcjaNieciaglosci[x_, y_] := Sin[x] Cos[y] * 0.8;
daneFunkcjaNieciaglosci = Table[funkcjaNieciaglosci[x, y], {x, -Pi, Pi, Pi/30}, {y, -Pi, Pi, Pi/30}];

Rzad = 8;
Waga[x_] := 1;
PhiBaza1[x_] = {Sin[x], Cos[x]};
PhiBaza2[x_] = {1, x};
PhiBaza3[x_] = {1, x, x^2, x^3, x^4, x^5, x^6, x^7};

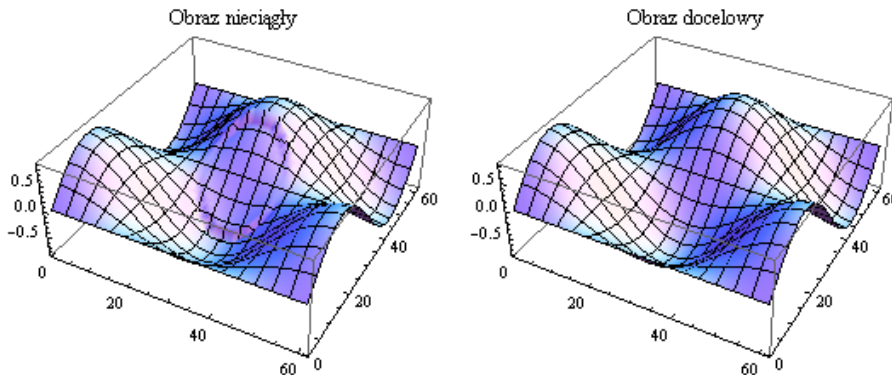
zakresArgumentow = {-Pi, Pi};
```

Rysunek 17.8: Przygotowanie danych w celu analizy nieciągłości obrazu

Do analizy zostały wybrane trzy wektory bazowe funkcji aproksymującej:

1. **baza trygonometryczna (PhiBaza1)**- wytypowana ze względu na sinusoidalną charakterystykę funkcji
2. **baza liniowa (PhiBaza2)**- wytypowana z powodu małej złożoności obliczeniowej
3. **baza wielomianowa stopnia 7 (PhiBaza3)** - interpolacja wielomianowa z predyktorem rzędu VIII

```
bezNieciaglosci = Table[Sin[x] Cos[y] 0.8, {x, -Pi, Pi, Pi/30}, {y, -Pi, Pi, Pi/30}];
GraphicsRow[{{ListPlot3D[daneFunkcjaNieciaglosci, PlotLabel -> "Obraz nieciągły"],
ListPlot3D[bezNieciaglosci, PlotLabel -> "Obraz docelowy"]}]
```



Rysunek 17.9: Porównanie obrazu nieciągłego z jego wygładzonym odpowiednikiem

Analizowany obraz nieciągły (rys. 17.9) prezentuje funkcję trójwymiarową  $f(x, y) = 0.8\sin(x)\cos(y)$  która dla ograniczonego obszaru, dziedziny walca cylindrycznego  $1 < x^2 + y^2 < 1.3$  przyjmuje wartość  $f(x, y) = \sin(x)\cos(y)$ .

Dwuwymiarowa funkcja w pierwszym przykładzie zostanie poddana transformacji na dwa odrębne sposoby, za pomocą filtracji po wierszach w pierwszym przypadku i filtracji po kolumnach w przypadku drugim.

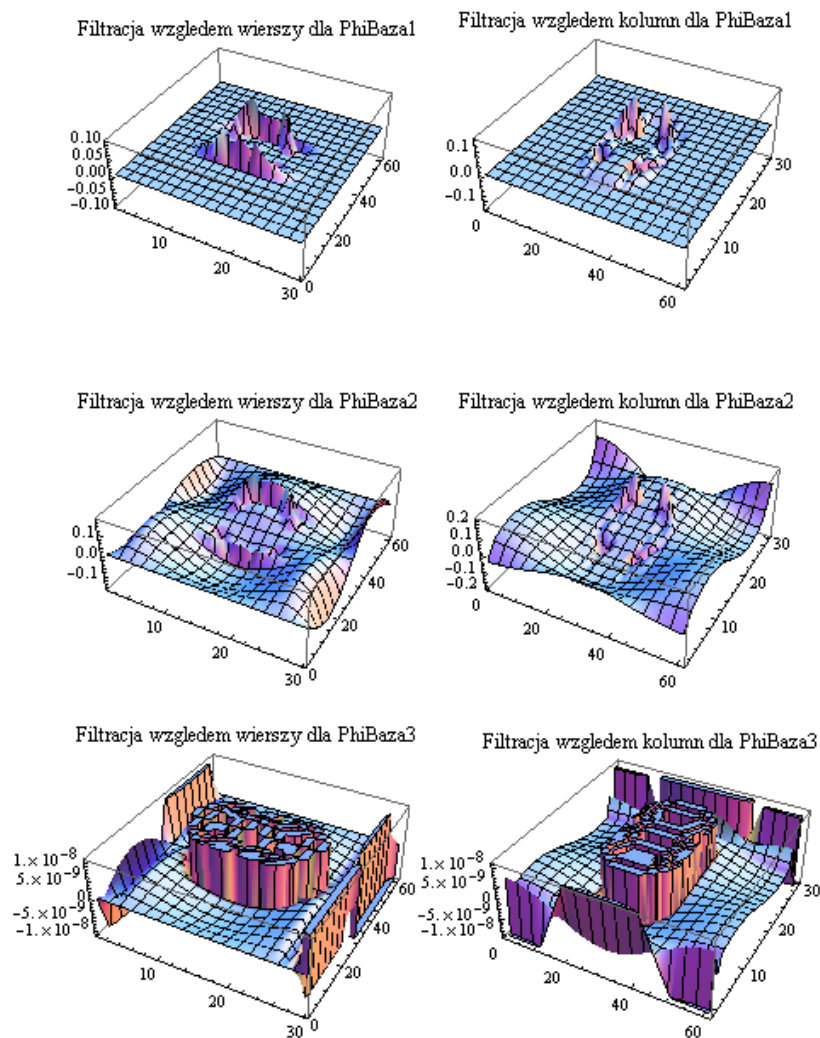
```
filtracjaWierszy1 = FiltracjaObrazuSchematemLiftinguMKNK[zakresArgumentow, daneFunkcjaNieciaglosci,
PhiBaza1, Waga, Rzad, 0, "miekka", "wiersze"];
filtracjaKolumn1 = FiltracjaObrazuSchematemLiftinguMKNK[zakresArgumentow, daneFunkcjaNieciaglosci,
PhiBaza1, Waga, Rzad, 0, "miekka", "kolumny"];
filtracjaWierszy2 = FiltracjaObrazuSchematemLiftinguMKNK[zakresArgumentow, daneFunkcjaNieciaglosci,
PhiBaza2, Waga, Rzad, 0, "miekka", "wiersze"];
filtracjaKolumn2 = FiltracjaObrazuSchematemLiftinguMKNK[zakresArgumentow, daneFunkcjaNieciaglosci,
PhiBaza2, Waga, Rzad, 0, "miekka", "kolumny"];
filtracjaWierszy3 = FiltracjaObrazuSchematemLiftinguMKNK[zakresArgumentow, daneFunkcjaNieciaglosci,
PhiBaza3, Waga, Rzad, 0, "miekka", "wiersze"];
filtracjaKolumn3 = FiltracjaObrazuSchematemLiftinguMKNK[zakresArgumentow, daneFunkcjaNieciaglosci,
PhiBaza3, Waga, Rzad, 0, "miekka", "kolumny"];
```

Rysunek 17.10: Filtracja danych - przygotowanie wyników

Do celów prezentacji dla każdego typu wektora bazy funkcji aproksymującej wybrana została reprezentacja przestrzeni detali, ponieważ ona jest nośnikiem informacji o nieciągłości obrazu.

Na rysunku 17.11 można zauważyć, że dla bazy trygonometrycznej **PhiBaza1** filtr niesie ze sobą w dużym stopniu informację o lokalizacji zaburzeń obrazu. Dla pozostałych baz (**PhiBaza2**, **PhiBaza3**), operacja filtracji przekazała obraz z którego można nie tylko wyodrębnić zmiany zachodzące w obrębie walca cylindrycznego, lecz także dodatkowe zafalowania w innych obszarach wykresu. Szczególne zaszumienie można zauważyć na krańcach przedziałów otrzymanych z wykorzystaniem interpolacji (**PhiBaza3**).

```
GraphicsGrid[
  {{ListPlot3D[filtracjaWierszy1[[2]], PlotLabel -> "Filtracja wzgledem wierszy dla PhiBaza1"],
    ListPlot3D[filtracjaKolumn1[[2]], PlotLabel -> "Filtracja wzgledem kolumn dla PhiBaza1"]},
  {ListPlot3D[filtracjaWierszy2[[2]], PlotLabel -> "Filtracja wzgledem wierszy dla PhiBaza2"],
    ListPlot3D[filtracjaKolumn2[[2]], PlotLabel -> "Filtracja wzgledem kolumn dla PhiBaza2"]},
  {ListPlot3D[filtracjaWierszy3[[2]], PlotLabel -> "Filtracja wzgledem wierszy dla PhiBaza3"],
    ListPlot3D[filtracjaKolumn3[[2]], PlotLabel -> "Filtracja wzgledem kolumn dla PhiBaza3"]}] ]
```



Rysunek 17.11: Filtracja danych - prezentacja wyników

Powyższa analiza pozwala ustalić analogiczny wniosek, jak w przypadku redukcji szumu obrazu, że dla celów wykrycia nieciągłości nie tyle istotny jest stopień wielomianu aproksymującego co odpowiedni dobór bazy względem charakterystyki przestrzennej obrazu.

W drugim przykładzie, obraz poddany zostanie filtracji zgodnie z zasadą analizy wielorozdzielczej funkcji dwuwymiarowej (rozdz. 9.7) dekomponując go na jeden obraz zgrubny i trzy obrazy z informacją o detalach.

Uzyskując reprezentację szczegółową w postaci trzech członów można zauważyć, że w celu optymalnej reprezentacji, nie w każdym przypadku otrzymuje się charakterystykę płaską dla dziedziny poza obszarem walca cylindrycznego.

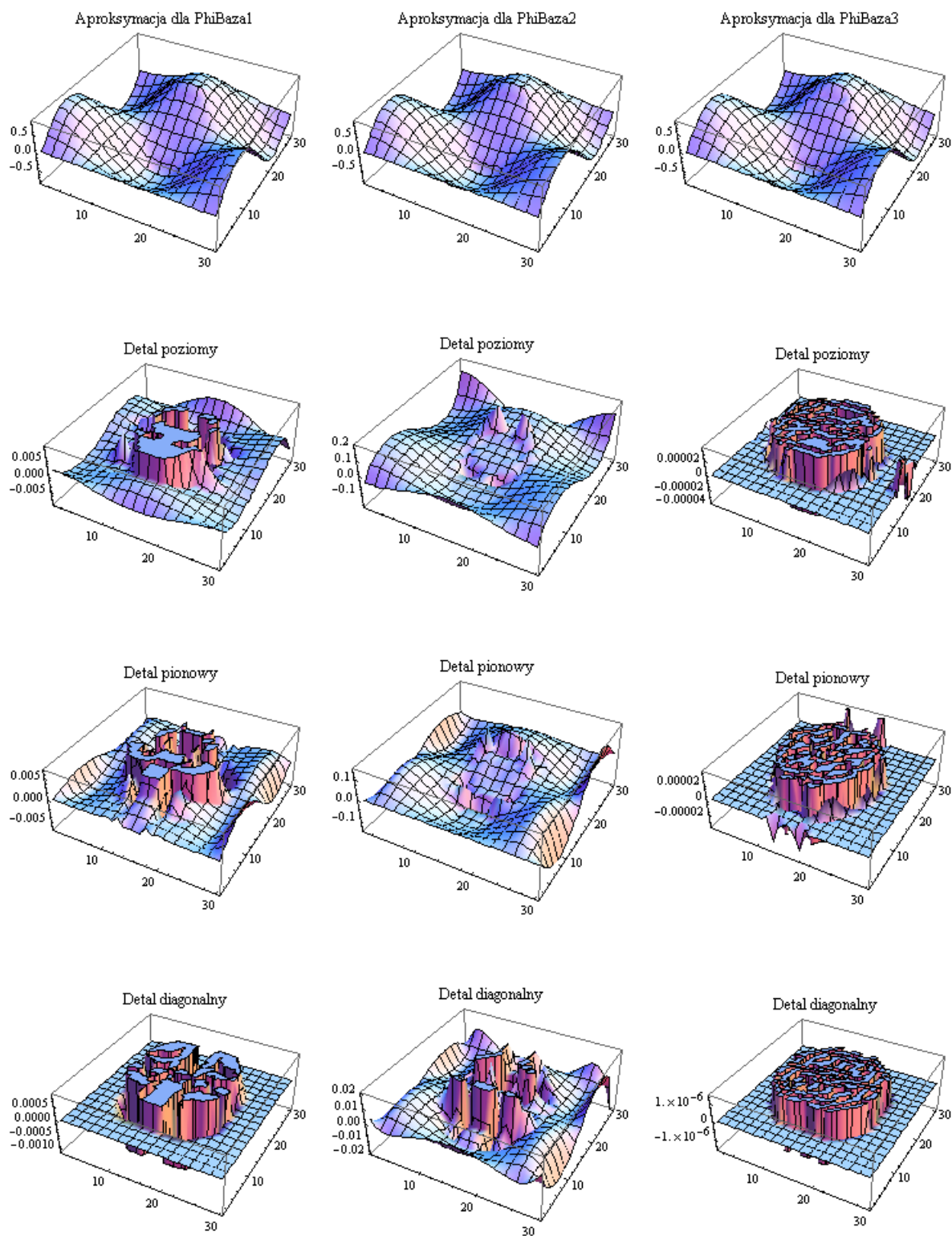
Przyczyną tego jest poddawanie obrazu dwukrotnemu zabiegowi filtracji, gdzie filtracja w drugim etapie odbywa się już na wcześniejszej transformacji obrazu z dodatkowymi zaburzeniami po aproksymacji obrazu.

Dla bazy **PhiBaza1** obraz płaski poddawany jest transformacji za pomocą bazy sinusoidalnej, powodując jego dodatkowe zafalowanie (rys. 17.13). Stąd też do celów analizy należy brać pod uwagę wypadkową trzech przestrzeni detali przechowujących informacje o szczegółach obrazu.

```
analiza1 = AnalizaObrazuSchematemLiftinguMKNK[zakresArgumentow, daneFunkcjaNieciaglosci, PhiBaza1,
Waga, Rzad, 0, "miekka"];
analiza2 = AnalizaObrazuSchematemLiftinguMKNK[zakresArgumentow, daneFunkcjaNieciaglosci, PhiBaza2,
Waga, Rzad, 0, "miekka"];
analiza3 = AnalizaObrazuSchematemLiftinguMKNK[zakresArgumentow, daneFunkcjaNieciaglosci, PhiBaza3,
Waga, Rzad, 0, "miekka"];
```

```
GraphicsGrid[
{{ListPlot3D[analiza1[[1]], PlotLabel -> "Aproksymacja dla PhiBaza1"},
ListPlot3D[analiza2[[1]], PlotLabel -> "Aproksymacja dla PhiBaza2"},
ListPlot3D[analiza3[[1]], PlotLabel -> "Aproksymacja dla PhiBaza3"}},
{ListPlot3D[analiza1[[2]], PlotLabel -> "Detal poziomy"},
ListPlot3D[analiza2[[2]], PlotLabel -> "Detal poziomy"},
ListPlot3D[analiza3[[2]], PlotLabel -> "Detal poziomy"}},
{ListPlot3D[analiza1[[3]], PlotLabel -> "Detal pionowy"},
ListPlot3D[analiza2[[3]], PlotLabel -> "Detal pionowy"},
ListPlot3D[analiza3[[3]], PlotLabel -> "Detal pionowy"}},
{ListPlot3D[analiza1[[4]], PlotLabel -> "Detal diagonalny"},
ListPlot3D[analiza2[[4]], PlotLabel -> "Detal diagonalny"},
ListPlot3D[analiza3[[4]], PlotLabel -> "Detal diagonalny"}
} ]
```

Rysunek 17.12: Analiza obrazu - przygotowanie wyników



Rysunek 17.13: Analiza danych - prezentacja wyników



W celu efektywnej oceny usuwania nieciągłości obrazu można przeprowadzić aproksymację z eliminacją progową o wartości 2.0 (rys. 17.14), która ze względu na wartości mieszące się pomiędzy  $< -1, 1 >$  wyzeruje wszystkie próbki ze wszystkich reprezentacji detali.

```

aproxymacja1 = SchematLiftinguMKNKObrazu[zakresArgumentow, daneFunkcjaNieciaglosci, PhiBaza1, Waga,
Rzad, 2, "miekka"];
aproxymacja2 = SchematLiftinguMKNKObrazu[zakresArgumentow, daneFunkcjaNieciaglosci, PhiBaza2, Waga,
Rzad, 2, "miekka"];
aproxymacja3 = SchematLiftinguMKNKObrazu[zakresArgumentow, daneFunkcjaNieciaglosci, PhiBaza3, Waga,
Rzad, 2, "miekka"];

```

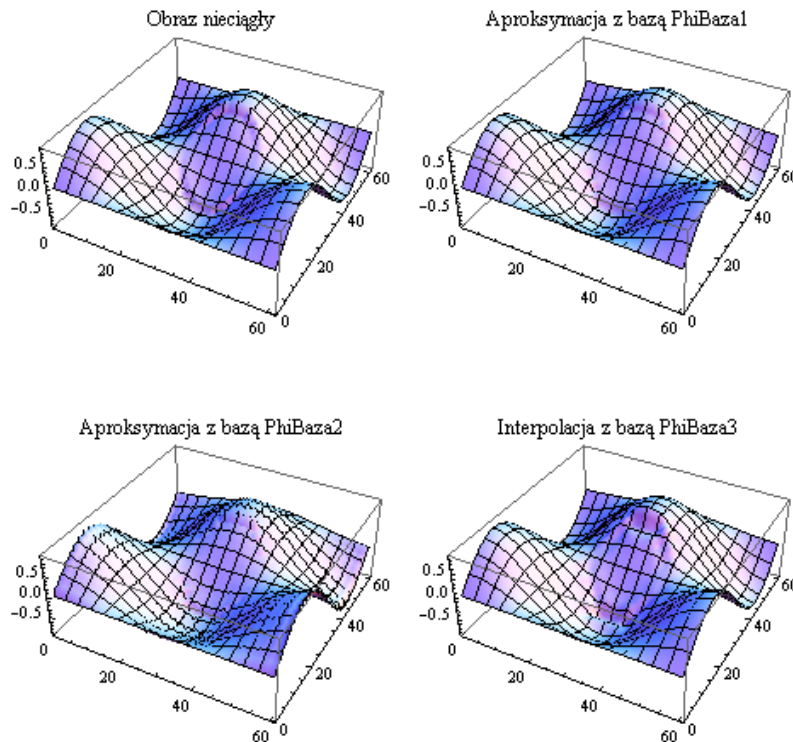
Rysunek 17.14: Redukcja nieciągłości - przygotowanie wyników

Analizując w sposób graficzny (rys. 17.15) można stwierdzić, że baza trygonometryczna **PhiBaza1** obrazuje najlepsze wygładzenie charakterystyki, natomiast aproksymacja bazą liniową **PhiBaza2** spowodowała znaczne zaszumienie obrazu.

```

GraphicsGrid[{{ListPlot3D[daneFunkcjaNieciaglosci, PlotLabel -> "Obraz nieciągły"],
ListPlot3D[aproxymacja1, PlotLabel -> "Aproksymacja z bazą PhiBaza1"]},
{ListPlot3D[aproxymacja2, PlotLabel -> "Aproksymacja z bazą PhiBaza2"] ,
ListPlot3D[aproxymacja3, PlotLabel -> "Interpolacja z bazą PhiBaza3"]}
}]

```



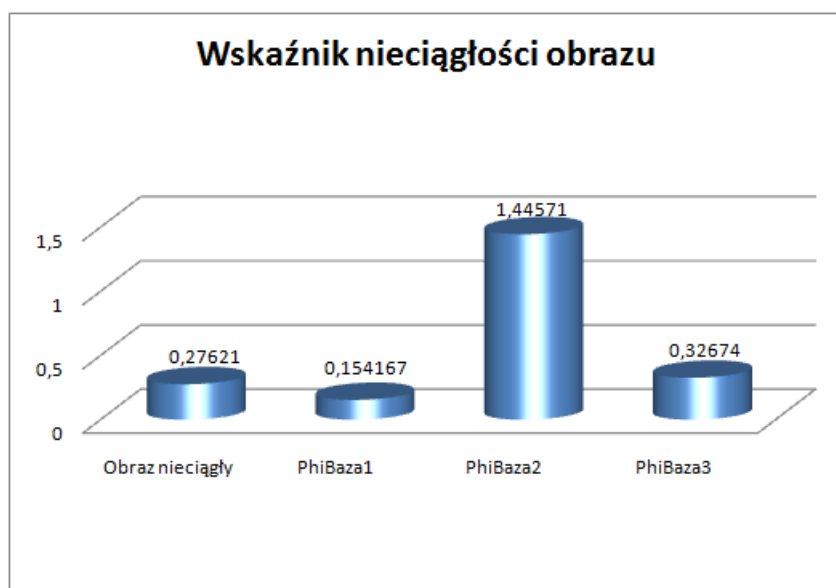
Rysunek 17.15: Redukcja nieciągłości - prezentacja wyników

W celu obiektywnej oceny operacji eliminacji nieciągłości można się posłużyć wskaźnikiem nieciągłości obrazu (rys. 17.16) który jest wariancją pomiędzy próbkami obrazu docelowego a jego

aproxymacją.

```
{MiaraEfektywnosciObrazu[bezNieciaglosci, daneFunkcjaNieciaglosci],
  MiaraEfektywnosciObrazu[bezNieciaglosci, aproxymacja1]},
{MiaraEfektywnosciObrazu[bezNieciaglosci, aproxymacja2],
  MiaraEfektywnosciObrazu[bezNieciaglosci, aproxymacja3]}} * 1000 // TableForm
TableForm=
0.27621 0.154167
1.44571 0.32674
```

Rysunek 17.16: Wskaźnik nieciągłości obrazu - prezentacja wyników



Rysunek 17.17: Wskaźnik nieciągłości obrazu - prezentacja graficzna

Otrzymany wynik\* pokazuje (rys. 17.17), że z pośród badanych baz - jedynie baza trygonometryczna (**PhiBaza1**) spowodowała redukcję nieciągłości obniżając wartość wskaźnika nieciągłości obrazu.

Interpolacja wykorzystana w bloku predykcji, według zastosowanej oceny, spowodowała dodatkowe zaszumienie obrazu.

### 17.3 Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty dla celów kompresji obrazu

Kompresja zgodnie z definicją to odwracalny lub nieodwracalny proces redukcji długości reprezentacji danych[45].

Tego typu transformację dla obrazów z wykorzystaniem schematu liftingu dokonuje się za pomocą rozwinięć falkowych na cztery składowe ze znacznie zmniejszoną liczbą współczynników.

Procedura kompresji realizowana jest analogicznie jak w przypadku redukcji szumu czy też eliminacji nieciągłości za pomocą trzech faz [3]:

\*Ze względu na bardzo małe wartości wskaźnika eliminacji nieciągłości uzyskanego dla badanego obrazu, wyniki zostały pomnożone przez współczynnik skalujący  $k = 1000$  dla celów prezentacji danych



- analizy obrazu
- eliminacji progowej współczynników obrazu
- syntezy obrazu

Eliminacja progowa w tym przypadku dopuszcza dwa podejścia [3]:

- za pomocą ustalonego globalnego progu lub współczynnika jakości kompresji zerując współczynniki poniżej zadanego progu
- wizualnie dobierając próg na każdym poziomie rozwinięcia

Do celów badawczych zostanie wykorzystana technika badania wrażliwości schematu liftingu z wykorzystaniem interpolacji wielomianowej i aproksymacji metodą najmniejszych kwadratów na zerowanie współczynników transformaty w szczególności współczynników reprezentujących szczegóły obrazu.

```
kubus = N[Import["kubuswavelets.bmp", "Data"] / 255, 15];
mati = N[Import["mati.jpg", "Data"] / 255, 15];
lena = N[Import["lenaM.bmp", "Data"] / 255, 15];
barbara = N[Import["barbaraM.bmp", "Data"] / 255, 15];
dom = N[Import["domM.bmp", "Data"] / 255, 15];
```

Rysunek 17.18: Pobranie obrazów do postaci tablic dwuwymiarowych w celach analizy

### 17.3.1 Model przestrzeni kolorów RGB

```
Rzad = 8;
Waga[x_] := 1;
PhiBazal[x_] = {1, Sin[x], Cos[x]};
zakresArgumentow = {-Pi, Pi};
```

Rysunek 17.19: Parametry aproksymacji metodą najmniejszych kwadratów

RGBSchematLiftinguMNObrazuPrezentacja [zakresArgumentow,  
kubus, PhiBaza1, Waga, Rzad, 0, "miekki"]



Rysunek 17.20: Wielorozdzielcza analiza obrazu formatu RGB

Przeprowadzając aproksymację dla formatu RGB można zauważyć istotną zależność pomiędzy rozdzielczością a jakością obrazu po aproksymacji (rys. 17.21,17.22,17.23)..

Im niższa rozdzielczość tym większa stopień pogorszenia jakości obrazu podczas operacji transformacji falkowej.

Powodem tym jest wyższy gradient zmiany kolorów dla sąsiednich próbek w przypadku niższych rozdzielczości, powodując przy tym większe błędy aproksymacji obrazu.

W przypadku aproksymacji dla większych rozdzielczości obraz wykazuje gładzsze przejście (mniejsze skoki) pomiędzy kolorami (rys. 17.30).

Graficzna prezentacja próbek obrazu z rys.17.21 w postaci przestrzennej ukazująca zmiany kolorów formatu RGB w całej dziedzinie obrazu (rys. 17.24) pozwala na zobrazowanie problemu.

RGBSchematLiftinguMKNKObrazuPrezentacja [zakresArgumentow, lena, PhiBaza1, Waga,  
Rzad, 1, "twarda"]



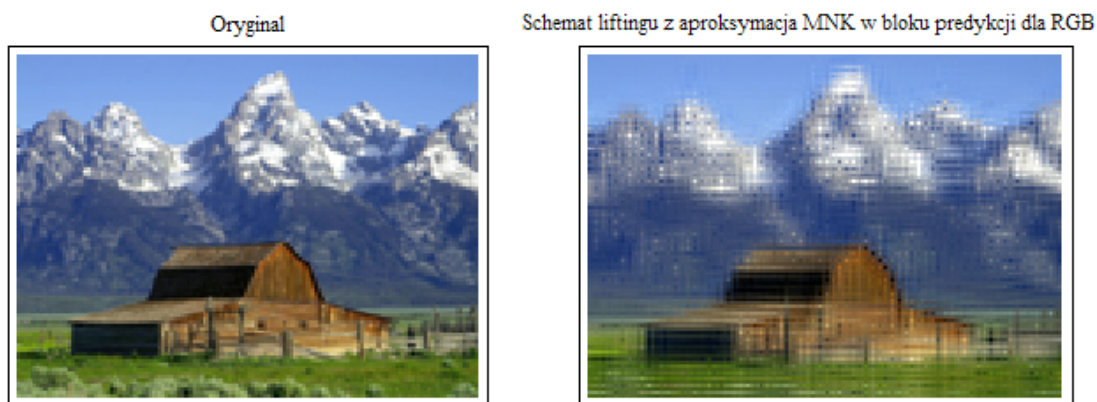
Rysunek 17.21: Aproksymacja obrazu MNK w bloku predykcji dla modelu RGB - I

RGBSchematLiftinguMKNKObrazuPrezentacja [zakresArgumentow, barbara, PhiBaza1, Waga,  
Rzad, 1, "twarda"]

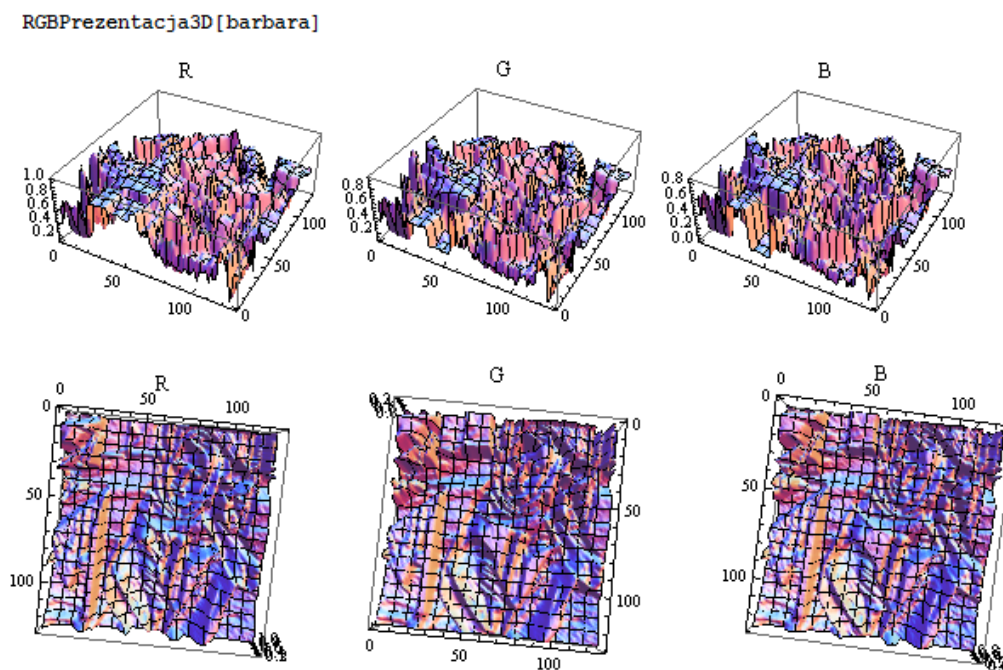


Rysunek 17.22: Aproksymacja obrazu MNK w bloku predykcji dla modelu RGB - II

RGBSchematLiftinguMNKObrazuPrezentacja[zakresArgumentow, dom, PhiBaza1, Waga,  
Rzad, 1, "twarda"]



Rysunek 17.23: Aproksymacja obrazu MNK w bloku predykcji dla modelu RGB - III



Rysunek 17.24: Graficzna prezentacja zmian kolorów składowych formatu RGB

RGBPrezentacja [barbara]



Rysunek 17.25: Składowe obrazu w formacie RGB

### 17.3.2 Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty modelu RGB

Do badania obrazu i oceny wrażliwości schematu liftingu na zerowanie współczynników transformaty zostanie wykorzystany, podobnie jak w przypadku redukcji szumu i badania nieciągłości, wskaźnik wariancji próbek obrazu aproksymowanego względem obrazu oryginalnego.

Porównanie wyników odbędzie się będzie za pomocą transformacji obrazu oryginalnego z rys. 17.23 dla trzech różnych wektorów baz : trygonometrycznej (**PhiBaza1**), liniowej (**PhiBaza2**) i interpolacyjnej(**PhiBaza3**) z wykorzystaniem predyktora rzędu VIII (rys. 17.26).

```
Rzad = 8;
Waga[x_] := 1;
PhiBaza1[x_] = {1, Sin[x], Cos[x]};
PhiBaza2[x_] = {1, x};
PhiBaza3[x_] = {1, x, x^2, x^3, x^4, x^5, x^6, x^7};
zakresArgumentow = {- Pi, Pi};
```

Rysunek 17.26: Przygotowanie parametrów aproksymacji do analizy

W przypadku transformacji falkowej obrazu za pomocą aproksymacji MNK w przestrzeni RGB można zauważyć znaczne zaszumienie obrazów dla bazy trygonometrycznej, jak i liniowej (rys. 17.27,17.28).

Interpolacja techniką predykcyjną wykazuje w sposób wizualny znacznie mniejsze zaszumienie dla części środkowej, jednakże prawy i dolny obszar przy krawędzi obrazu utracił w dużym stopniu informację spowodowaną dużym zaszumieniem próbek, co zostało zobrazowane na rys. 17.24. Zniekształcenie w sytuacji granicznej spowodowane jest niedopasowaniem bazy funkcji aproksymującej względem obrazu. Przypadek taki zaprezentowany został w podrozdziale 13.8.4 kiedy to funkcja sinusoidalna była poddawana interpolacji wielomianowej za pomocą predyktora rzędu IV.

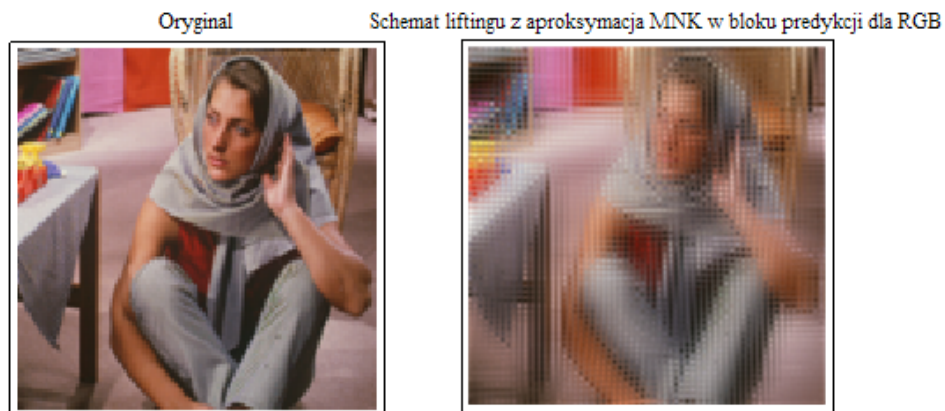


RGBSchematLiftinguMKNKObrazuPrezentacja[zakresArgumentow, barbara, PhiBaza1, Waga,  
Rzad, 1.0, "miekka"]



Rysunek 17.27: Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB z wykorzystaniem bazy trygonometrycznej

RGBSchematLiftinguMKNKObrazuPrezentacja[zakresArgumentow, barbara, PhiBaza2, Waga,  
Rzad, 1.0, "miekka"]



Rysunek 17.28: Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB z wykorzystaniem bazy liniowej

```
RGBSchematLiftinguMKNKObrazuPrezentacja[zakresArgumentow, barbara, PhiBaza3, Waga,  
Rzad, 1.0, "miekkka"]
```



Rysunek 17.29: Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB z wykorzystaniem bazy interpolacyjnej

```
RGBSchematLiftinguMKNKObrazuPrezentacja[zakresArgumentow, mati, PhiBaza1, Waga, Rzad, 1, "twarda"]
```



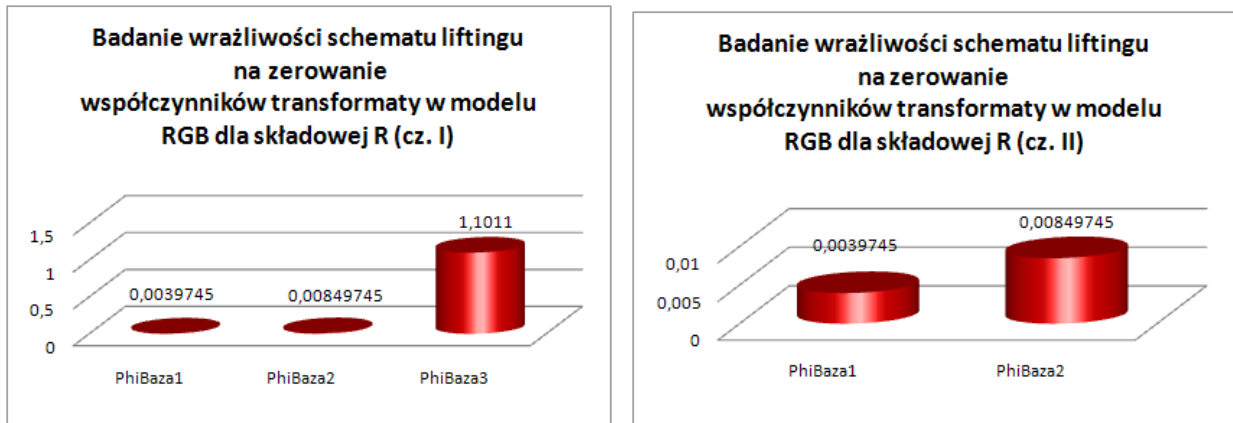
Rysunek 17.30: Zerowanie współczynników reprezentacji szczegółowej modelu RGB

```

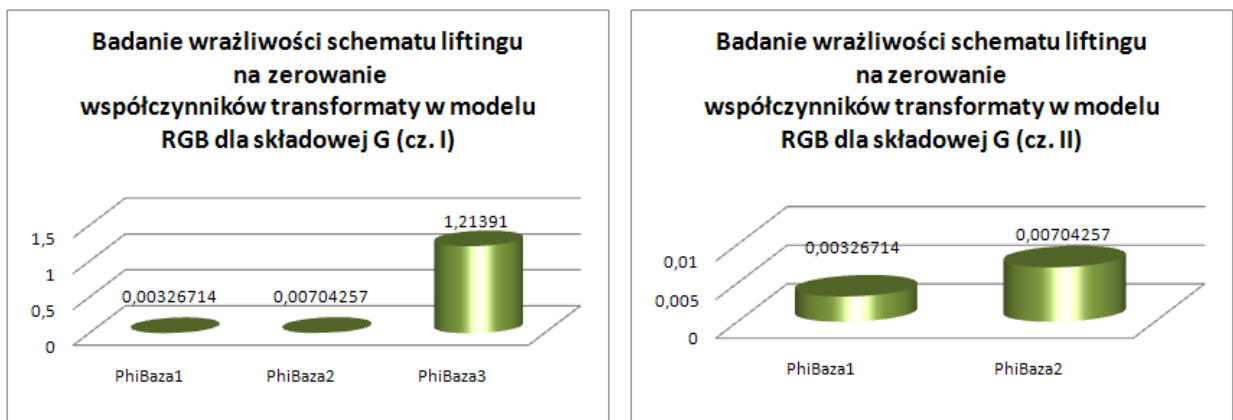
RGBAproksymacja1 = RGBSchematLiftinguMKNKObrazu[zakresArgumentow, barbara, PhiBaza1,
    Waga, Rzad, 1.0, "miekka"];
RGBAproksymacja2 = RGBSchematLiftinguMKNKObrazu[zakresArgumentow, barbara, PhiBaza2,
    Waga, Rzad, 1.0, "miekka"];
RGBAproksymacja3 = RGBSchematLiftinguMKNKObrazu[zakresArgumentow, barbara, PhiBaza3,
    Waga, Rzad, 1.0, "miekka"];

{MiaraEfektywnosciObrazu[RGBAproksymacja1, barbara],
    MiaraEfektywnosciObrazu[RGBAproksymacja2, barbara],
    MiaraEfektywnosciObrazu[RGBAproksymacja3, barbara]} // TableForm
0.0039745  0.00326714  0.00377999
0.00849745  0.00704257  0.00813898
1.1011     1.21391    2.44816
    
```

Rysunek 17.31: Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB

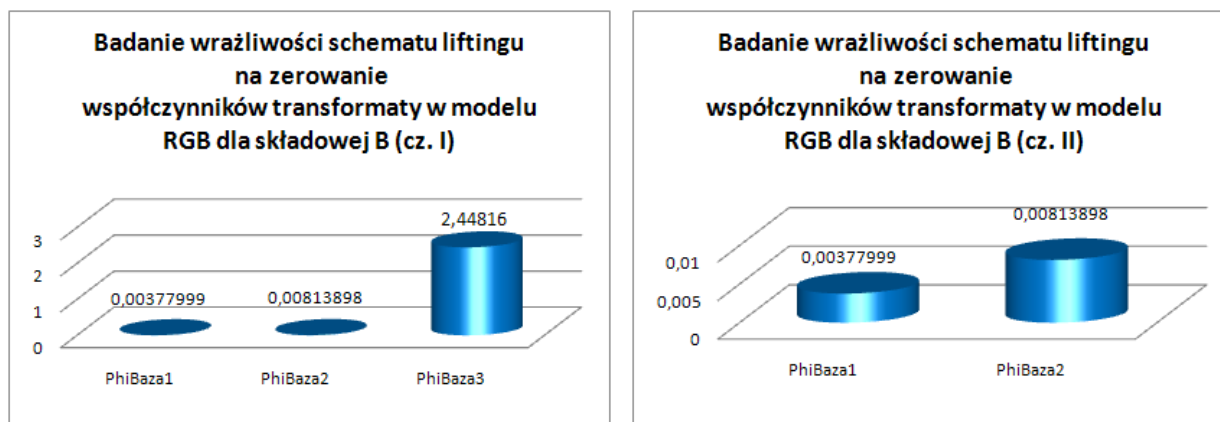


Rysunek 17.32: Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB dla składowej R



Rysunek 17.33: Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB dla składowej G



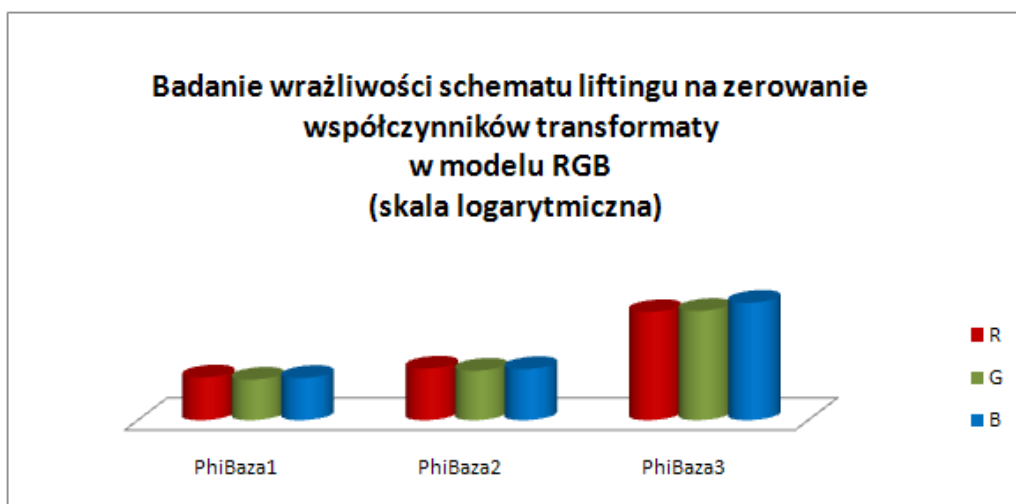


Rysunek 17.34: Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB dla składowej B

Wyniki przedstawione na rys. 17.31-17.35, przedstawiają trzy oceny z których każda wyznacza odrębną miarę dla osobnych składowych formatu RGB, dla każdego badanego wektora bazy funkcji aproksymujących.

Analizując w sposób wizualny wynik transformacji funkcji z wykorzystaniem interpolacji (**PhiBaza3**) można zauważyć, że charakteryzuje się ona stosunkowo mniejszym zaszumieniem aniżeli wynik transformacji dla baz **PhiBaza1** i **PhiBaza2**.

Utrata informacji na granicach obrazu z zastosowaniem filtrów interpolacyjnych wpływa jednak na radykalny wzrost wartości wariancji dla wszystkich wartości próbek poszczególnych składowych obrazu RGB względem obrazu oryginalnego, powodując jego znaczące pogorszenie wartości wskaźnika wrażliwości schematu liftingu na zerowanie współczynników transformaty.



Rysunek 17.35: Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB - postać graficzna

Zgodnie z otrzymanymi wynikami, transformacja obrazu za pomocą aproksymacji MNK z wykorzystaniem bazy trygonometrycznej (**PhiBaza1**) dla formatu RGB wykazała najlepszy wskaźnik wrażliwości pomimo, że jej analiza wizualna nie pozwala jednoznacznie stwierdzić optymalnego rozwiązania. Przyczyną tego jest jego równomierne pogorszenie jakości analizowanego obrazu które

nie powoduje tak dużych zniekształceń obrazu jak na przykład w przypadku lokalnych, granicznych zaburzeń bazy interpolacyjnej (**PhiBaza3**).

### 17.3.3 Model przestrzeni kolorów YCbCr

Ze względu na redundancję danych przestrzeni kolorów RGB (rys. 17.25) zastosowanie aproksymacji MNK techniką biortogonalnej transformaty falkowej w celu uzyskania rozwiązań przybliżonych nie jest dobrym rozwiązaniem.

W konsekwencji, podczas aproksymacji w modelu RGB tracona jest ważna informacja o luminancji obrazu.

Ponieważ oko ludzkie jest bardziej podatne na informacje o jasności (luminancji) obrazu niż na informację o kolorze (chrominancji) pozwala to przeniesienie analizy na inny model przestrzeni kolorów w celu polepszenia jakości obrazu po transformacji [1].

Operację transformacji falkowej można przeprowadzić w przestrzeni kolorów YCbCr gdzie dane o luminancji przechowywane lub transmitowane są w oryginalnej postaci, zaś dane o chrominancji (Cb, Cr) podlegają transformacji, dzięki czemu mogą być przesyłane wykorzystując mniejszą szerokość pasma.

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.0813124 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

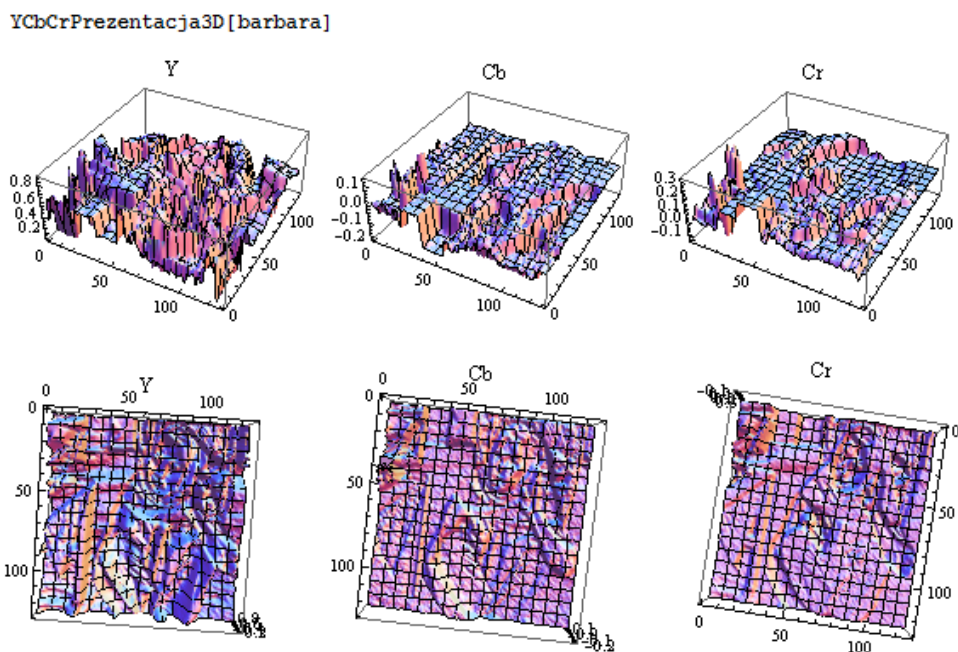
Rysunek 17.36: Macierz przekształcenia przestrzeni RGB do YCbCr [1]

Format YCbCr nie jest bezwzględna przestrzenią kolorów jest tylko metodą na opisanie informacji przekazywanych w formacie RGB, zatem nowe wartości próbek otrzymuje się na podstawie przekształcenia z rys. 17.36.

YCbCrPrezentacja [barbara]



Rysunek 17.37: Podział obrazu na składowe YCbCr



Rysunek 17.38: Graficzna prezentacja zmian kolorów składowych formatu YCbCr

Analizując podział na składowe YCbCr (17.37,17.38) widać istotną różnicę pomiędzy składową luminancji a składowymi chrominancji koloru czerwonego i niebieskiego. Składowa luminancji uwypukla ważne cechy obrazu pod względem konturów i jasności, natomiast składowe chrominancji obrazują w sposób bardziej regularny informacje o rozkładzie kolorów: niebieskiego i czerwonego.

Charakterystyka przestrzenna pozwala w sposób wizualny zobrazować dlaczego do operacji eliminacji progowej wykorzystuje się tylko składowe chrominancji (rys. 17.25). Na wykresie można zauważyć, że składowe chrominancji cechują się o wiele większą regularnością, aniżeli składowa luminancji.

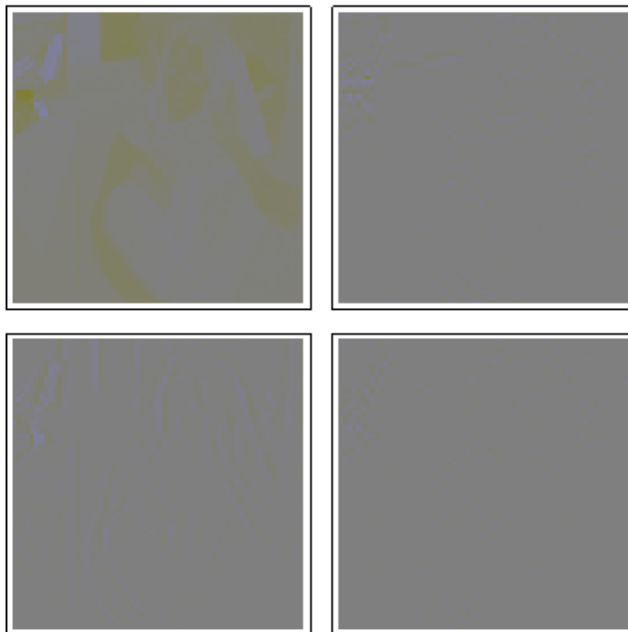
Składowe Cb i Cr pozwalają, ze względu na swoją gładką charakterystykę i małe wartości przestrzenne, na efektywniejszą aproksymację bez utraty istotnych szczegółów (rys. 17.39,17.40,17.41).

```
YCbCrAnalizaObrazuSchematemLiftinguMNKPrezentacja[zakresArgumentow,  
Y, PhiBaza1, Waga, Rzad, 0.0, "miekk", "Y"]
```



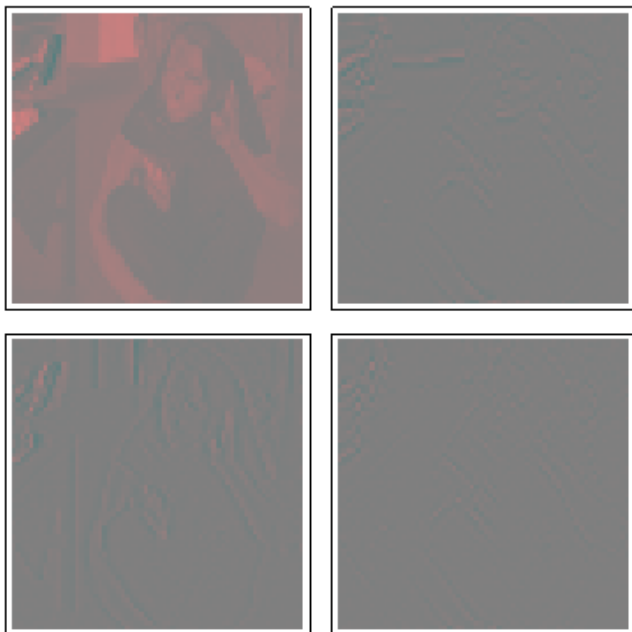
Rysunek 17.39: Analiza składowej Y obrazu za pomocą schematu liftingu w wykorzystaniem aproksymacji MNK (składowa nie podlega aproksymacji podczas transformacji w modelu YCbCr)

```
YCbCrAnalizaObrazuSchematemLiftinguMNKPrezentacja[zakresArgumentow,  
Cb, PhiBaza1, Waga, Rzad, 0.0, "miekk", "Cb"]
```



Rysunek 17.40: Analiza składowej Cb obrazu za pomocą schematu liftingu w wykorzystaniem aproksymacji MNK

```
YCbCrAnalizaObrazuSchematemLiftinguMKNPrezentacja[zakresArgumentow,  
Cr, PhiBaza1, Waga, Rzad, 0.0, "miekka", "Cr"]
```



Rysunek 17.41: Analiza składowej Cr obrazu za pomocą schematu liftingu w wykorzystaniu aproksymacji MNK

#### 17.3.4 Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty modelu RGB w przestrzeni YCbCr

Biorąc pod uwagę transformację falkową w formacie YCbCr realizowaną jest w sposób zbliżony jak w przypadku formatu RGB. Jedyną różnicą polega na pominięciu, podczas eliminacji współczynników, składowej luminancji Y na którą oko ludzkie jest najbardziej czułe. Eliminacji progowej współczynników podlegają jedynie składowe chrominancji które zgodnie z rys. 17.38 wykazują znacznie większą regularność charakterystyki obrazu.

YCbCrSchematLiftinguMKNKObrazuPrezentacja[zakresArgumentow, barbara, PhiBaza1,  
Waga, Rząd, 1.0, "miekka"]



Rysunek 17.42: Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowych Cb i Cr z wykorzystaniem bazy trygonometrycznej

YCbCrSchematLiftinguMKNKObrazuPrezentacja[zakresArgumentow, barbara, PhiBaza2,  
Waga, Rząd, 1.0, "miekka"]



Rysunek 17.43: Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowych Cb i Cr z wykorzystaniem bazy liniowej



```
YCbCrSchematLiftinguMKNKObrazuPrezentacja[zakresArgumentow, barbara, PhiBaza3,
Waga, Rząd, 1.0, "miekkka"]
```



Rysunek 17.44: Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowych Cb i Cr z wykorzystaniem bazy interpolacyjnej

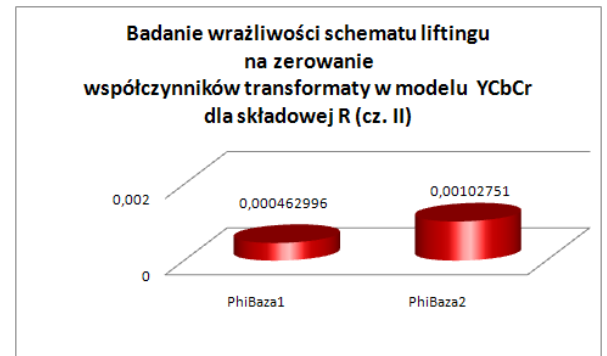
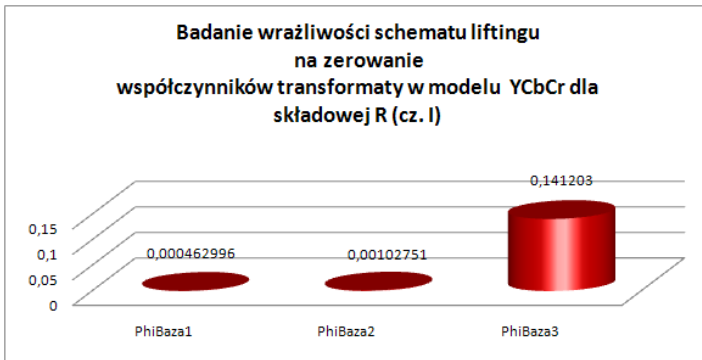
Postępując analogicznie jak w przypadku badania wrażliwości transformacji falkowej na zerowanie współczynników dla modelu RGB można zauważyć, że użycie interpolacji wielomianowej (**PhiBaza3**) do celu transformacji spowodowało otrzymanie w sposób graficzny, obrazu o najgorszej jakości. Spowodowane to było dużymi błędami aproksymacji na krańcach dziedziny badanego układu.

```
YCbCrAproksymacja1 = YCbCrSchematLiftinguMKNKObrazu[zakresArgumentow, barbara,
PhiBaza1, Waga, Rząd, 1.0, "miekkka"];
YCbCrAproksymacja2 = YCbCrSchematLiftinguMKNKObrazu[zakresArgumentow, barbara,
PhiBaza2, Waga, Rząd, 1.0, "miekkka"];
YCbCrAproksymacja3 = YCbCrSchematLiftinguMKNKObrazu[zakresArgumentow, barbara,
PhiBaza3, Waga, Rząd, 1.0, "miekkka"];

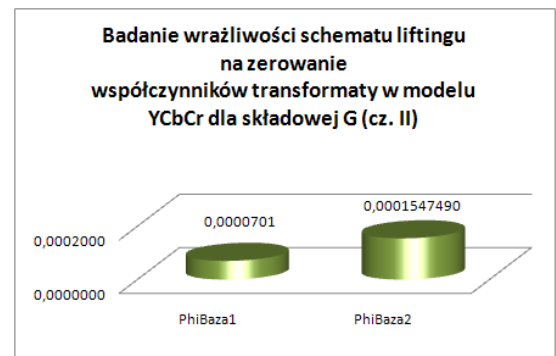
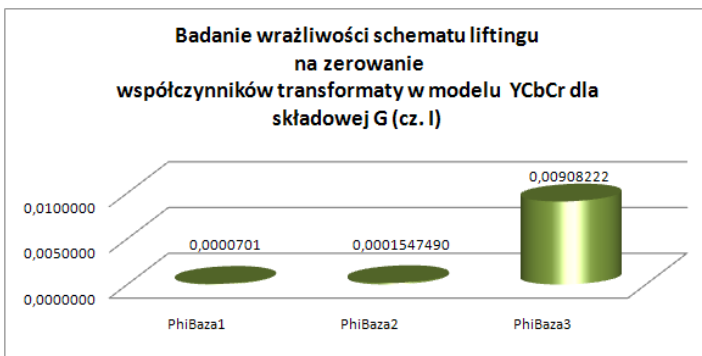
{MiaraEfektywnosciObrazu[YCbCrAproksymacja1, barbara],
MiaraEfektywnosciObrazu[YCbCrAproksymacja2, barbara],
MiaraEfektywnosciObrazu[YCbCrAproksymacja3, barbara]} // TableForm

0.000462996 0.0000701036 0.000519782
0.00102751 0.000154749 0.00103264
0.141203 0.00908222 0.490307
```

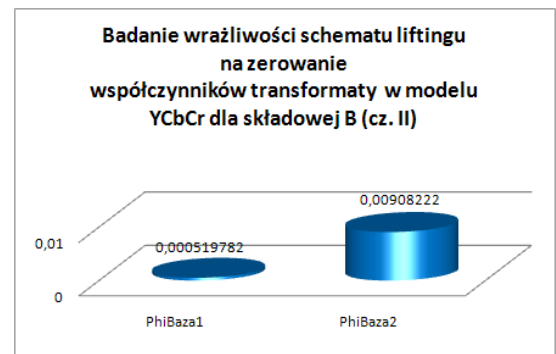
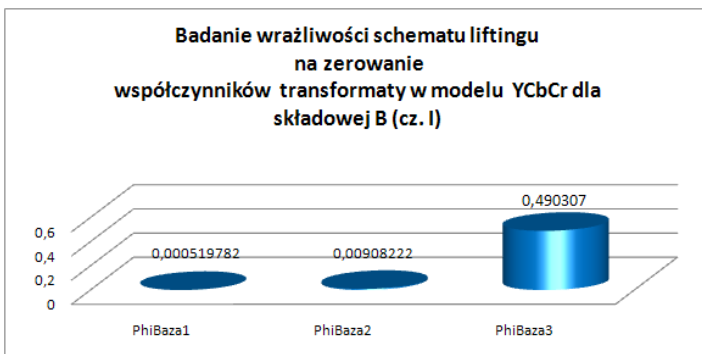
Rysunek 17.45: Miara oceny efektywności kompresji obrazu RGB w przestrzeni modelu YCbCr



Rysunek 17.46: Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowej R

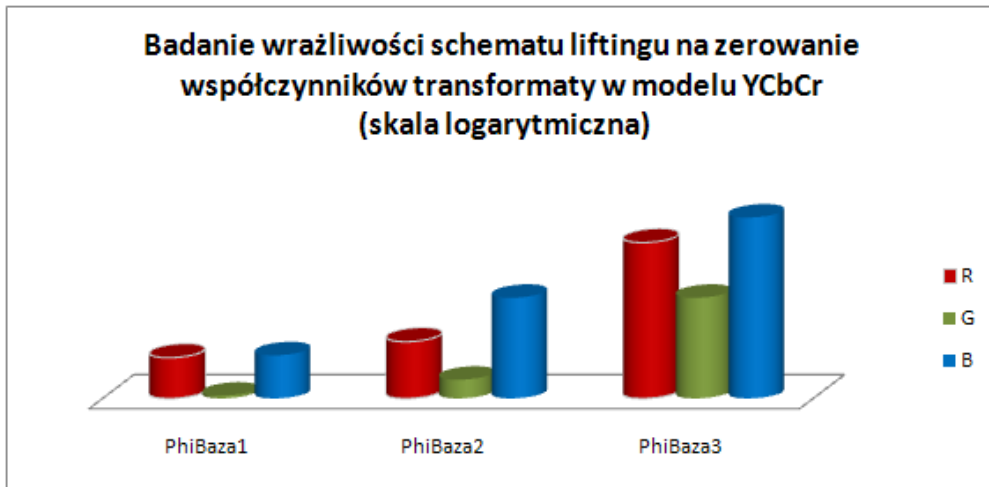


Rysunek 17.47: Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowej G



Rysunek 17.48: Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowej B





Rysunek 17.49: Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr - postać graficzna

Według rezultatów modelu YCbCr uzyskanych dla badanych trzech baz funkcji aproksymującej (rys. 17.26, 17.49) najniższe wartości wariancji dla przetransformowanych poszczególnych składowych RGB, a co się z tym wiąże najlepszą wartość wskaźnika wrażliwości względem badanego obrazu, wykazuje aproksymacja metodą najmniejszych kwadratów w bloku predykcji z wykorzystaniem bazy trygonometrycznej (**PhiBaza1**).

Analiza przestrzennego rozkładu kolorów z rys. 17.38 i jego tendencja do obszarowej sinusoidalnej charakterystyki potwierdza optymalność w doborze trygonometrycznej bazy dla celów biortogonalnej transformacji falkowej z użyciem aproksymacji metodą najmniejszych kwadratów .

Poniżej zostały zaprezentowane przykładowe obrazy przed i po transformacji z zastosowaniem bazy trygonometrycznej.

```
YCbCrSchematLiftinguMKNKobrazuPrezentacja[zakresArgumentow, lena, PhiBaza1, Waga, Rząd, 1, "twarda"]
```



Rysunek 17.50: Zerowanie współczynników reprezentacji szczegółowej - przykład I

YCbCrSchematLiftinguMKNKObrazuPrezentacja[zakresArgumentow, kubus, PhiBaza1, Waga, Rzad, 1, "twarda"]



Rysunek 17.51: Zerowanie współczynników reprezentacji szczegółowej - przykład II

YCbCrSchematLiftinguMKNKObrazuPrezentacja[zakresArgumentow, mati, PhiBaza1, Waga, Rzad, 1, "twarda"]



Rysunek 17.52: Zerowanie współczynników reprezentacji szczegółowej - przykład III



## Część VII

# Wnioski końcowe



## Rozdział 18

---

# Podsumowanie

---

Schemat liftingu jest narzędziem konstrukcji biortogonalnych systemów falkowych [58, 60], który pozwala na dopasowywanie funkcji aproksymującej do charakterystyki funkcji aproksymowanej.

Zależność reprezentacji zgrubej od reprezentacji szczegółowej podczas dekompozycji funkcji, za pomocą bloku predykcji i uaktualnienia umożliwia rekonstrukcję funkcji bez potrzeby znajomości filtrów dolno- i górnoprzepustowych które są niezbędne w przypadku klasycznej realizacji transformaty falkowej.

W pracy została zaprojektowana, zaimplementowana i wykorzystana do badań technika schematu liftingu z autorskim pomysłem wykorzystania aproksymacji metodą najmniejszych kwadratów (MNK) w bloku predykcji (część V).

Narzędzie to, w przeciwieństwie do interpolacyjnej wersji zaproponowanej przez Wima Sweldensa [60, 58], pozwala na szacowanie nieparzystych próbek estymowanych za pomocą funkcji nie przechodzących przez węzły tj. parzyste próbki funkcji aproksymowanej (rozd. 1, rozdz. 14).

Dzięki powyższej własności funkcja aproksymująca może być nie tylko wielomianem rzędu o wiele niższego co funkcja interpolująca, lecz także nie musi charakteryzować się postacią wielomianową.

Ponadto, w pracy

- zaprezentowano m.in. sposoby uzyskiwania funkcji falkowych i skalujących za pomocą metod dokładnych jak i przybliżonych wraz z ich prezentacją graficzną (rozd. 6-8)
- wykazano zależność pomiędzy dyskretną, klasyczną transformatą falkową a transformatą realizowaną za pomocą schematu liftingu (rozd. 15)
- opisano związek pomiędzy współczynnikami wektora predykcji, jak i wektora uaktualnienia schematu liftingu dla funkcji okresowych (rozd. 13.9)
- dla potrzeb uzasadnienia i prezentacji wyników, wywodzących się z rozważań teoretycznych, zaimplementowano kompleksowy pakiet << *SchematLiftinguZAproksymacjaMNK* >>.
  - zbiór funkcji został napisany z wykorzystaniem narzędzia do obliczeń symbolicznych Mathematica firmy Wolfram
  - w pakiecie zostało zaimplementowanych 45 procedur i funkcji które pozwalają na wnikliwą analizę funkcji jedno- i dwuwymiarowych za pomocą schematu liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów w bloku predykcji.

```
<< SchematLiftinguZAproksymacjaMnk`
Length[Names["SchematLiftinguZAproksymacjaMnk`*"]]
45
? SchematLiftinguZAproksymacjaMnk` *
```

▼ SchematLiftinguZAproksymacjaMnk`

AnalizaFunkcjiPredictUpdate	RGBToYCbCr
AnalizaFunkcjiEliminacjaProgowaSzumuPredictUpdate	SchematLiftinguInterpolacjaFunkcji
AnalizaObrazuSchematemLiftinguMnk	SchematLiftinguMnkFunkcji
CbConvertToImageData	SchematLiftinguMnkFunkcjiPrezentacja
CbFormat	SchematLiftinguMnkFunkcjiZWielomianemOrtogonalnym
CrConvertToImageData	SchematLiftinguMnkFunkcjiZWielomianemTrygonometrycznym
CrFormat	SchematLiftinguMnkObrazu
EliminacjaProgowaSzumu	SyntezaFunkcji
FiltracjaObrazuSchematemLiftinguMnk	SyntezaFunkcjiPredictUpdate
MacierzPrzekształceniaRGBToYCbCr	SyntezaObrazuSchematemLiftinguMnk
MiaraEfektywnosciObrazu	WyznaczanieWspolczynnikaWektorowPredykcjiUaktualnienia
NormalizacjaFunkcji	YCbCr
NormalizacjaObrazu	YCbCrAnalizaObrazuSchematemLiftinguMnkPrezentacja
PredyktorAproksymacji	YCbCrConvertToImageData
PrezentacjaFunkcjiFalkowych	YCbCrFormat
RedukcjaSzumu	YCbCrPrezentacja
RGBAnalizaObrazuSchematemLiftinguMnk	YCbCrPrezentacja3D
RGBAnalizaObrazuSchematemLiftinguMnkPrezentacja	YCbCrSchematLiftinguMnkObrazu
RGBFormat	YCbCrSchematLiftinguMnkObrazuPrezentacja
RGBPrezentacja	YCbCrToRGB
RGBPrezentacja3D	YConvertToImageData
RGBSchematLiftinguMnkObrazu	YFormat
RGBSchematLiftinguMnkObrazuPrezentacja	

Rysunek 18.1: Lista procedur i funkcji pakietu `<< SchematLiftinguZAproksymacjaMnk``.

- przeprowadzono analizę otrzymanych wyników dotyczących bloku predykcji schematu liftingu z aproksymacją MNK. Wskazano możliwości optymalizacji obliczeń ze względu na złe uwarunkowanie macierzy współczynników wykorzystując do tego celu wielomiany ortogonalne z wyszczególnieniem wielomianów Grama i wielomiany trygonometryczne (rozdz. 14.3 - 14.5)
- przeprowadzono eksperymenty dla przykładowych jednowymiarowych danych w celu redukcji szumów, wykrywania nieciągłości i zmiany charakteru funkcji jednowymiarowych (rozdz. 16)
- przeprowadzono analizę i ocenę obrazu podczas redukcji szumów, wykrywaniu nieciągłości i badaniu wrażliwości schematu liftingu z wykorzystaniem aproksymacji MNK na zerowanie współczynników transformaty w szczególności współczynników reprezentujących szczegóły (rozdz. 17).

W części I zostały postawione następujące tezy:

1. Możliwe jest zdefiniowanie nowej biortogonalnej transformaty falkowej w przypadku której w bloku predykcji schematu liftingu zamiast interpolacji wielomianowej zaproponowanej przez Wima Sweldensa [61, 60] użyta jest aproksymacja metodą najmniejszych kwadratów.
2. Nowa transformata falkowa spełniająca wszystkie własności analizy wielorozdzielczej, posiada również korzystną cechę, którą jest możliwość wykorzystania funkcji aproksymującej w bloku predykcji o rzędzie niższym bądź równym funkcji interpolującej.

**Ad. 1.** Otrzymanie schematu liftingu z klasycznego schematu dwukanałowej dekompozycji i rekonstrukcji (rys. 15.4), opisanego szczegółowo w rozdz. 12-13, nie jest jednoznaczne i nie zawsze możliwe.

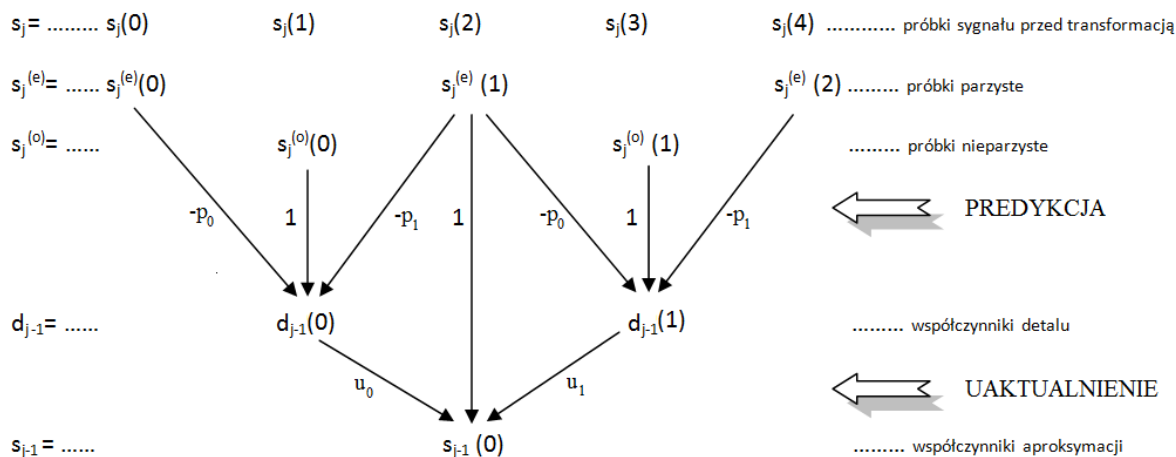
Brak rozwiązania może wynikać z braku istnienia stałej wartości wyznacznika macierzy polifazowej  $\mathbf{P}(z)$ , uniemożliwiając określenie zależności biortogonalnych filtrów falkowych (rozdz. 12).

Niejednoznaczność natomiast jest konsekwencją wykorzystywania algorytmu Euklidesa przy użyciu wielomianów Laurenta podczas obliczeń macierzowych (rozdz. 13.2).

Obliczenia te przekształcają macierz polifazową do iloczynu macierzy dolno i górnotrójkątnych z jedynkami na przekątnych, będących odpowiednio algebraiczną reprezentacją bloków predykcji i uaktualnienia.

$$\mathbf{P}(z) = \prod_{i=1}^m \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix}$$

W przeciwnym kierunku (zgodnym z postawioną tezą 1), sytuacja wygląda zupełnie inaczej, ponieważ schemat liftingu można przekształcić zawsze i w sposób jednoznaczny do postaci klasycznego schematu dwukanałowego z wykorzystaniem filtrów biortogonalnych.



Rysunek 18.2: Przykład zależności pomiędzy transformatą klasyczną, a transformatą realizowaną za pomocą schematu liftingu z predyktorem  $p = [p_1, p_2]$  i wektorem uaktualnienia  $u = [u_1, u_2]$ .

Realizacja tego typu przekształcenia została zaprezentowana i omówiona w rozdz. 15, na podstawie relacji pomiędzy dyskretną transformatą falkową a transformatą realizowaną przy użyciu schematu liftingu.

Można zatem, wywnioskować, że schemat liftingu jest szybszą obliczeniowo implementacją transformaty falkowej pozwalającą jednocześnie na prostą realizację transformaty odwrotnej.



Użycie aproksymacji metodą najmniejszych kwadratów w bloku predykcji (rozd. 14) pozwala, drogą przekształceń macierzowych, uzyskać alternatywną do wersji interpolacyjnej (rozd. 13.7) postać predykcijną.

Nowe próbki nieparzyste uzyskiwane drogą aproksymacji (rozd. 14), wyliczane są w ten sam sposób jak w przypadku interpolacji (rozd. 13.6.1) tj. za pomocą iloczynu skalarnego wektora predykcji z wektorem próbek parzystych.

Wektory predykcji dla tego samego rzędu predyktora (liczba próbek parzystych wykorzystywanych w bloku predykcji), podczas aproksymacji i interpolacji różnią się jednak między sobą wartościami współczynników.

Powodem tego jest inny dobór wektora bazy i rzędu wykorzystywanego predyktora schematu liftingu.

Wyjątkiem jest sytuacja gdy wielomian aproksymujący jest taki sam jak wielomian interpolujący, wtedy to wektory są ze sobą tożsamościowo równe (rozd. 14.2).

Reasumując, obliczenia z wykorzystaniem wektora predykcji uzyskiwanego za pomocą aproksymacji metodą najmniejszych kwadratów (wzór 14.11) w sposób analogiczny do jego interpolacyjnego odpowiednika (wzór 13.44), jak również własność schematu liftingu dotycząca relacji pomiędzy klasyczną transformatą falkową a jej predykcijną implementacją pozwalają potwierdzić poprawność postawionej tezy 1.

**Ad. 2.** Uzasadnieniem tezy 2 są przykłady przedstawione w rozdz. 16 (część V) i rozdz. 17 (część VI), jak również powszechnie znane fakty z teorii aproksymacji i interpolacji [69, 62].

Proponowana przez autora implementacja schematu liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów (MNK) w bloku predykcji (rozd. 14), ze względu na elastyczność dopasowania do badanego układu, ma swoje zastosowanie dla celów separacji funkcji jednowymiarowej czy też obrazu od jego szumu.

Wykorzystanie dodatkowo eliminacji progowej (rozd. 17.1) podczas wyżej wymienionej transformacji pozwala na wykorzystanie jej narzędzia dla potrzeb usuwania nieciągłości układu (rozd. 17.2).

Przykładem takim jest funkcja z rys. 16.8, gdzie po eliminacji progowej i rekonstrukcji można otrzymać funkcję harmoniczną o zadanej regularności.

Podczas dekompozycji obrazów redukcja współczynników z zadaniem progiem umożliwia wygładzić dwuwymiarową funkcję, eliminując lub minimalizując reprezentację szczegółową na zadanym poziomie rozdzielczości (rozd. 17.2).

Ponieważ przestrzeń detali wyodrębnia cechy układu nadające się do oceny samopodobieństwa funkcji i ich własności fraktalnych, dwuwymiarowa analiza pozwala na wykorzystanie jej do rozróżniania tekstur i analizy fraktalnej.

Aproksymacja MNK techniką predykcijną, ze względu na swoje zalety wykorzystywane podczas odsumowania funkcji opisane w rozdziałach VI i V, może mieć zastosowanie m.in. w geofizyce, astronomii, biologii, a także w ekonomii. Jej cechy mogłoby być przydatne podczas badań nad wstrząsami sejsmicznymi w seismologii lub podczas analizy zaburzeń ruchu płynów w hydrodynamicie. Własności tego typu aproksymacji pozwalają również na użycie jej w analizie mowy i w sterowaniu procesami przemysłowymi.

Popularność tego typu rozwiązań ciągle rośnie, szczególnie w obrębie analizy obrazów dynamicznych, które w dobie technologii internetowej i telewizji cyfrowej mają tendencję do gwałtownego rozwoju. W przyszłości zatem można spodziewać się ich dalszego rozwoju m.in. w kierunku powstawania optymalnych algorytmów dla tzw. kodeków [32], pozwalając dopasowywać jakość wyświetlanego obrazu do szybkości komputera, bądź transferu danych poprzez sieć Internet.

## Rozdział 19

---

# Kierunki dalszych badań

---

Na bazie rozważań teoretycznych i otrzymanych wyników podczas zrealizowanych badań przy użyciu schematu liftingu z wykorzystaniem aproksymacji metodą najmniejszych kwadratów w bloku predykcji, można wyodrębnić przykładowe kierunki dalszych badań

- rozszerzenie schematu liftingu z aproksymacją metodą najmniejszych kwadratów na analizę funkcji jednowymiarowych lub obrazów z nierównomiernie rozłożonymi próbkami w dziedzinie czasu.
- zmniejszenie złożoności obliczeniowej przy pomocy ortogonalnych baz funkcji aproksymującej dla próbek nierównomiernie rozłożonych.
- klasyfikacja wektorów predykcji lub wektorów bazy funkcji aproksymujących pod kątem dopasowania do badanych układów.
- przeniesienie schematu liftingu z aproksymacją metodą najmniejszych kwadratów na przypadek  $n$ -wymiarowych rozmaitości (np.: usuwanie nieciągłości rozkładu ciśnienia w zadanym obszarze przestrzeni).
- wykrywanie istotnych cech badanego obiektu pod względem geometrycznym lub w celu analizy innych przestrzeni barw.
- rozróżnianie tekstur i analizy fraktalnej w celu oceny samopodobieństwa funkcji i ich własności fraktalnych

Powyższe przykłady badań stanowią zaledwie część obszarów w których można przeprowadzać doświadczenia z zakresu analizy wielorozdzielczej wykorzystując aproksymację metodą najmniejszych kwadratów.

Kierunki realizacji nowych narzędzi i ich użycie uzależnione jest m.in. od zapotrzebowania i rozwoju różnorodnych dziedzin techniki na przykład technologii internetowych i sposobu reprezentacji obrazów. Schemat liftingu z jego aproksymacyjną implementacją, ze względu na elastyczne dopasowanie do badanego układu, może umożliwić rozwój coraz to lepszych narzędzi analizy wielorozdzielczej.

Badania tego typu rozwiązań pozwolą m. in. na poprawę jakości danych graficznych poddawanych aproksymacji z możliwością przywrócenia danych do postaci przed transformacją.



Część VIII

Dodatki



---

# Bibliografia

---

- [1] *A novel skin color model in YCbCr color space and its application to human face detection*, volume 1, 2002. [cytowanie na str. 181, 208]
- [2] Richard Baraniuk. Optimal tree approximation with wavelets. In *Wavelet Applications in Signal and Image Processing VII, volume 3813 of SPIE Proceedings*, pages 196–207, 1999. [cytowanie na str. 57]
- [3] Jan T. Białasiewicz. *Falki i aproksymacje*. Wydawnictwo Naukowo-Techniczne, Warszawa, 2000. [cytowanie na str. 9, 58, 60, 62, 68, 69, 74, 153, 155, 160, 171, 172]
- [4] Andrew G. Bruce, Andrew G. Bruce, Hong ye Gao, and Hong ye Gao. Wavelet analysis. *IEEE Spectrum*, 33:26–35, 1996. [cytowanie na str. 9]
- [5] Adhemar Bultheel. Wavelets with applications in signal and image processing. Technical report, Department of Computer Science, K.U.Leuven, 2006. [cytowanie na str. 19, 45, 52, 57, 74, 75, 76]
- [6] C. S. Burrus and Ramesh A. Gopinath. Wavelet transforms and filter banks. In *In Wavelets: A Tutorial in Theory and Applications*, pages 603–654. Academic Press, 1992. [cytowanie na str. 52, 74, 84]
- [7] Jian Liub Chengyi Xionga, Jinwen Tianb. Low complexity reconfigurable architecture for the 5/3 and 9/7 discrete wavelet transform. pages 303–308, 2006. [cytowanie na str. 9]
- [8] Chang Chiann, Chang Chiann, Pedro A. Morettin, and Pedro A. Morettin. A wavelet analysis for time series. *J. Nonparametr. Statist.*, 10:1–46, 1998. [cytowanie na str. 62, 151]
- [9] Christos Chrysa and Antonio Ortega. Minimum memory implementations of the lifting scheme. In *In Proceedings of SPIE, International Symposium on Optical Science and Technology*, 2000. [cytowanie na str. 89]
- [10] Barry A. Cipra. Wavelet Applications Come to the Fore. Technical report, SIAM News (Mathematics that Counts), 1993. [cytowanie na str. 9]
- [11] Albert Cohen. Wavelets: the mathematical background. In *Proc. IEEE*, pages 514–522, 1996. [cytowanie na str. 60]
- [12] Stephan Dahlke and Angela Kunoth. Biorthogonal wavelets and multigrid. In *in: Adaptive Methods | Algorithms, Theory and Applications, Proceedings of the 9th GAMM Seminar, W. Hackbusch, G. Wittum (eds.), NNFM Series*, pages 99–119, 1994. [cytowanie na str. 81]

- [13] Wolfgang Dahmen. Stability of multiscale transformations. *J. Fourier Anal. Appl.*, 2:341–361, 1996. [cytowanie na str. 58]
- [14] Wolfgang Dahmen, Angela Kunoth, and Karsten Urban. Wavelets in numerical analysis and their quantitative properties. pages 93–130. Press, 1997. [cytowanie na str. 58, 59]
- [15] Ingrid Daubechies and Wim Sweldens. Factoring wavelet transforms into lifting steps. Technical report, 1998. [cytowanie na str. 86, 87, 90, 91, 92, 93, 94, 100, 101]
- [16] Ronald A. Devore and R. A. Devore. Constructive approximation. *Acta Numerica*, 7:51–150, 1993. [cytowanie na str. 60]
- [17] Ronald A. Devore and Bradley J. Lucier. Wavelets. *Acta Numerica*, 1:1–56, 1992. [cytowanie na str. 58]
- [18] V. N. Dvornychenko, V. N. Dvornychenko, Carlos M. Gutierrez, William A. Jeffrey, Michael D. Garris, Michael D. Garris, and Michael D. Garris. Summary of nist latent fingerprint testing workshop, nistir 7377, november 2006, [http://fingerprint.nist.gov/latent/ir\\_7377.pdf](http://fingerprint.nist.gov/latent/ir_7377.pdf). [cytowanie na str. 9]
- [19] A. N. Fedorova and M. G. Zeitlin. Wavelets in optimization and approximations. *Math. and Comp. in Simulation*, 46:527–534, 1998. [cytowanie na str. 29]
- [20] Grigorij M. Fichtenholz. *Rachunek różniczkowy i całkowy*, volume 1. 12 edition, 2002. [cytowanie na str. 164]
- [21] Bernd Fischer and J Urgen Prestin. Wavelets based on orthogonal polynomials. *Math. Comp.*, 66:1593–1618, 1997. [cytowanie na str. 130]
- [22] Bernd Fischer and Woula Themistoclakis. Orthogonal polynomial wavelets. *Numer. Algorithms*, 30:37–58, 2001. [cytowanie na str. 130]
- [23] Masafumi Fujii and Wolfgang J. R. Hoefer. Applications of biorthogonal interpolating wavelets to the galerkin scheme of time dependent maxwell's equations. *IEEE Microwave and Wireless Components Letters*, 11:22–24, 2001. [cytowanie na str. 89]
- [24] A. C. Gilbert. A comparison of multiresolution and classical one-dimensional homogenization schemes. *Appl. Comp. Har. Anal.*, 5:1–35, 1998. [cytowanie na str. 58]
- [25] Carnell Hampton, Zac Ziegler, and Tim Persons. Basic wavelet transforms. [OnLine] <http://citeseer.ist.psu.edu/hampton98basic.html>, 1998. [cytowanie na str. 21]
- [26] Christopher E. Heil, David, and F. Walnut. Continuous and discrete wavelet transforms. *SIAM Review*, 31:628–666, 1989. [cytowanie na str. 21, 26]
- [27] Marcin Jaromin. Approximation of signals by predict wavelet transform. volume 5337/2009 of *Lecture Notes in Computer Science*, pages 184–197. Springer Berlin / Heidelberg, 2009. Computer Vision and Graphics, International Conference, ICCVG 2008, Warsaw, Poland, November 10-12, 2008 Revised Papers. [cytowanie na str. 121, 126, 127]
- [28] Marcin Jaromin. Aproksymacja sygnału na podstawie schematu liftingu. volume 30 of *Informatyka*, pages 5–23. Silesian University of Technology Press, 2009. zeszyty naukowe Politechniki Śląskiej. [cytowanie na str. 121, 126]

- [29] Bjorn Jawerth and Wim Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Rev.*, 36:377–412, 1994. [cytowanie na str. 24]
- [30] A. Juditsky, Q. Zhang, B. Delyon, P. y. Glorennec, and A. Benveniste. Wavelets in identification. In *in SYSID'94, 10th IFAC Symposium on System Identification*, pages 27–48, 1994. [cytowanie na str. 68]
- [31] Tanja Karp and Alfred Mertins. Lifting schemes for biorthogonal modulated filter banks. In *In Proc. International Conference on Digital Signal Processing*, 1997. [cytowanie na str. 81, 89, 104]
- [32] Lin Luo, Jin Li, Shipeng Li, Zhenquan Zhuang, and Ya qin Zhang. Motion compensated lifting wavelet and its application in video coding. In *in: Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 481–484, 2001. [cytowanie na str. 196]
- [33] Richard G. Lyons. *Understanding digital signal processing*. Addison Wesley Longman, Inc., 1997. [cytowanie na str. 73, 155]
- [34] Dana Mackenzie. *Wavelets: Seeing the Forest and the Trees*. Technical report, National Academy of Sciences, 2001. [cytowanie na str. 9]
- [35] Stephane G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989. [cytowanie na str. 70, 85]
- [36] B.P. Demidowicz, I.A. Maron. *Metody numeryczne cz. I. Analiza, algebra, metody Monte Carlo*. Wydawnictwo Naukowe PWN, Warszawa, 1965. [cytowanie na str. 127, 128, 131, 132, 133, 134]
- [37] Pedro A. Morettin. From Fourier to wavelet analysis of time series. In *In Proceedings in Computational Statistics*, pages 111–122. Physica-Verlag, 1996. [cytowanie na str. 62]
- [38] Ramesh Neelamani and C. Sidney Burrus. *Wavelet transforms via lifting*. Rice Universty, Houston, 2007. [cytowanie na str. 89]
- [39] Truong Q. Nguyen. A tutorial on filter banks and wavelets. In *Proc. IEEE International Conference on Digital Signal Processing*, Cypress, June 1995. [cytowanie na str. 9]
- [40] J. E. Odegard, R. A. Gopinath, and C. S. Burrus. Optimal wavelets for signal decomposition and the existence of scale limited signals. In *In Proc. Int. Conf. Acoust., Speech, Signal Processing*. IEEE, 1992. [cytowanie na str. 58]
- [41] Victor Y. Pan. Solving a polynomial equation: Some history and recent progress. *SIAM Rev.*, 39(2):187–220, 1997. [cytowanie na str. 31]
- [42] Grégoire Pau, Christophe Tillier, and Béatrice Pesquet-popescu. Optimization of the predict operator in lifting-based motion compensated temporal filtering. In *in Proc. of Visual Communications and Image Processing*, 2004. [cytowanie na str. 103, 115]
- [43] F. Payan, Y. Boulfani, and M. Antonini. Temporal lifting scheme for the compression of animated sequences of meshes. In *Proceedings of IEEE International Workshop VLBV 2005*, Sardinia, Italy, september 2005. [cytowanie na str. 9]
- [44] Jurgen Prestin and Kathi Selig. Interpolatory and orthonormal trigonometric wavelets. In *J. Zeevi and R. Coifman*, pages 201–255. Academic Press, 1998. [cytowanie na str. 19]



- [45] Artur Przelaskowski. *Kompresja danych. Podstawy, metody bezstratne, kodery obrazów*. BTC, Warszawa, 2005. [cytowanie na str. 82, 84, 86, 88, 94, 103, 171]
- [46] Rong qing Jia. Using the refinement equation for the construction of pre-wavelets ii. pages 209–246. Academic Press, 1991. [cytowanie na str. 133]
- [47] Rong qing Jia and Jianzhong Wang. Stability and linear independence associated with wavelet decompositions. *Proc. Amer. Math. Soc.*, 117:1115–1124, 1993. [cytowanie na str. 21]
- [48] Anthony Ralston. *A first course in numerical analysis*. McGraw-Hill, Inc., New York, 1965. [cytowanie na str. 127, 131]
- [49] Peter Schröder and Wim Sweldens. Spherical wavelets: efficiently representing functions on the sphere. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 161–172, New York, NY, USA, 1995. ACM. [cytowanie na str. 81]
- [50] Jo Simoens, Jo Simoens, Jo Simoens, Stefan Vandewalle, Stefan Vandewalle, and Stefan Vandewalle. On the stability of wavelet bases in the lifting scheme. Technical report, Department of Computer Science, K.U.Leuven, 2000. [cytowanie na str. 115]
- [51] Pramila Srinivasan and Leah H. Jamieson. Wavelets based on splines: an applications. *SPIE*, 2825:46–4, 1996. [cytowanie na str. 58]
- [52] David Stanhill and Yehoshua Y. Zeevi. Two-dimensional orthogonal wavelets with vanishing moments. *IEEE Trans. Signal Processing*, 46:2579–2590, 1995. [cytowanie na str. 27, 70, 71]
- [53] Gilbert Strang. Wavelet transforms versus fourier transforms. *BULL.AMER.MATH.SOC*, 28:288, 1993. [cytowanie na str. 29, 33, 36, 52, 74, 76, 151]
- [54] Gilbert Strang and Truong Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, MA, USA, 1996. [cytowanie na str. 34, 36, 37, 45, 47, 52, 73, 74]
- [55] Vasily Strela and A. Vogan. Multiwavelets: Theory and applications. Department of Mathematics and Computer Science, Drexel University, Philadelphia, 1996. [cytowanie na str. 58, 62, 160]
- [56] W. Sweldens. Wavelets: What next? *Proc. IEEE*, 84(4):680–685, 1996. [cytowanie na str. 143]
- [57] Wim Sweldens. Wavelet sampling techniques. In *In 1993 Proceedings of the Statistical Computing Section*, pages 20–29, 1993. [cytowanie na str. 83, 160]
- [58] Wim Sweldens. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In *in Wavelet Applications in Signal and Image Processing III*, pages 68–79. SPIE conference, 1995. [cytowanie na str. 77, 106, 107, 145, 193]
- [59] Wim Sweldens. Wavelets and the lifting scheme: A 5 minute tour. *Z. Angew. Math. Mech*, 76:41–44, 1996. [cytowanie na str. 110]
- [60] Wim Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29:511–546, 1997. [cytowanie na str. 11, 14, 86, 89, 90, 91, 107, 193, 195]
- [61] Wim Sweldens and Peter Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, pages 15–87. ACM SIGGRAPH Course Notes, 1996. [cytowanie na str. 11, 14, 77, 103, 105, 106, 126, 195]

- [62] B.P. Demidowicz, I.A. Maron, E.Z. Szuwałowa. *Metody numeryczne cz. II. Przybliżanie funkcji, równania różniczkowe i całkowe*. Wydawnictwo Naukowe PWN, Warszawa, 1965. [cytowanie na str. 128, 130, 131, 196]
- [63] Trac D. Tran. Linear phase perfect reconstruction filter bank: lattice structure, design, and application in image coding. *IEEE Trans. on Signal Processing*, 48:133–147, 1998. [cytowanie na str. 100, 101]
- [64] G. Uytterhoeven, F. Van Wulpen, M. Jansen, D. Roose, and A. Bultheel. WAILI: A software library for image processing using integer wavelet transforms. In K.M. Hanson, editor, *Medical Imaging 1998: Image Processing*, volume 3338 of *Proc. SPIE Int. Soc. Opt. Eng.*, pages 1490–1501. International Society for Optical Engineering, 1998. [cytowanie na str. 86, 87, 91, 94, 95, 101]
- [65] Geert Uytterhoeven. *Wavelets: Software and applications* [ph.d. thesis]. Belgium, 1999. [cytowanie na str. 83, 97, 101, 104]
- [66] Geert Uytterhoeven, Geert Uytterhoeven, Dirk Roose, Dirk Roose, Adhemar Bultheel, and Adhemar Bultheel. Wavelet based interactive video communication and image database consulting wavelet transforms using the lifting scheme. Technical report, Department of Computer Science, K.U.Leuven, 1997. [cytowanie na str. 81, 101, 110]
- [67] C. Valens. The fast lifting wavelet transform. In *A Really Friendly Guide to Wavelets*. PolyValens, 2004. [cytowanie na str. 86, 92, 94, 95]
- [68] M. Victor Wickerhauser. *The wavelets and the applications*. Department of mathematics, Washington University in St. Louis, 2007. [cytowanie na str. 21]
- [69] Z. Fortuna, B. Macukow, J. Wąsowski. *Metody numeryczne*. Wydawnictwo Naukowo-Techniczne, Warszawa, 1998. [cytowanie na str. 127, 128, 129, 130, 131, 133, 134, 196]
- [70] Przemysław Wojtaszczyk. *Teoria falek*. Wydawnictwo Naukowe PWN, Warszawa, 2000. [cytowanie na str. 28, 43, 57, 60, 68, 70]
- [71] Xiao-Ping Zhang, M. Desai, and Ying-Ning Peng. On the symmetry of orthogonal complex filter banks and wavelets. *IEEE Trans. on Signal Proc*, 47:1039–1048, 1999. [cytowanie na str. 81]
- [72] Tomasz P. Zieliński. *Cyfrowe przetwarzanie sygnałów. Od teorii do zastosowań*. Wydawnictwo Komunikacji i Łączności, Warszawa, 2007. [cytowanie na str. 11, 59, 103, 105, 106, 108, 143, 145, 206]
- [73] Radosław Grzymkowski, Adam Zielonka. *Zastosowanie teorii falek w zagadnieniach brzegowych*. Wydawnictwo Pracowni Komputerowej J. Skalmierskiego, Gliwice, 2004. [cytowanie na str. 58, 62, 63]
- [74] Rob A. Zuidwijk. Directional and time-scale wavelet analysis. *SIAM Journal on Mathematical Analysis*, 31:416–430, 2000. [cytowanie na str. 83]
- [75] Przemysław Śliwiński Zygmunt Hasiewicz. *Falki ortogonalne o zwartym nośniku. Zastosowanie do nieparametrycznej identyfikacji systemów*. EXIT, Warszawa, 2005. [cytowanie na str. 10, 23, 27, 29, 36, 37, 60, 66, 67, 69]

---

# Spis rysunków

---

1.1	Przykład wykorzystania przekształceń falkowych . . . . .	10
1.2	Przykład realizacji schemat liftingu . . . . .	11
1.3	Blok predykcji dla predyktora rzędu II . . . . .	12
1.4	Szacowanie próbek nieparzystych - dane . . . . .	12
1.5	Szacowanie próbek nieparzystych - implementacja . . . . .	14
6.1	Algorytm Stranga - przygotowywanie danych . . . . .	39
6.2	Algorytm Stranga - przykładowy wynik dla $\varphi(\frac{53}{128}) = \varphi(0.414063)$ . . . . .	39
6.3	Wykres funkcji skalującej <b>db2</b> uzyskany za pomocą algorytmu Stranga . . . . .	41
7.1	Wykres funkcji falkowej <b>db2</b> uzyskany za pomocą algorytmu Stranga . . . . .	43
8.1	Przykład zbieżności metody iteracyjnej . . . . .	47
8.2	Funkcja skalująca <b>db2</b> uzyskana metodą iteracyjną dla kolejnych iteracji cz.I . . . . .	49
8.3	Funkcja skalująca <b>db2</b> uzyskana metodą iteracyjną dla kolejnych iteracji cz. II . . . . .	50
8.4	Funkcja falkowa <b>db2</b> uzyskana metodą iteracyjną dla $i=5$ . . . . .	51
9.1	Aproksymacja Haara na poziomie rozdzielczości I . . . . .	65
9.2	Aproksymacja Haara na poziomie rozdzielczości II . . . . .	65
9.3	Aproksymacja Haara na poziomie rozdzielczości III . . . . .	66
9.4	Schemat falkowej dekompozycji obrazu za pomocą falki <b>db2</b> . . . . .	72
9.5	Wizualizacja falkowej dekompozycji obrazu w postaci struktury drzewa . . . . .	72
12.1	Schemat dwukanałowej dekompozycji i rekonstrukcji . . . . .	85
12.2	Schemat analizy i syntezy za pomocą reprezentacji polifazowej . . . . .	88
13.1	Liftingu prymalny - blok uaktualnienia . . . . .	90
13.2	Lifting dualny - blok predykcji . . . . .	91
13.3	Bloki schematu liftingu . . . . .	92
13.4	Schemat liftingu ze skalowaniem - analiza funkcji . . . . .	95
13.5	Schemat liftingu ze skalowaniem - synteza funkcji . . . . .	95
13.6	Macierz polifazowa dla CDF(2,2) . . . . .	96
13.7	Przykłady interpolacji funkcji z predyktorem rzędu a) I, b) II i c)IV [72] . . . . .	105
13.8	Interpolacja wielomianowa - predyktor rzędu I . . . . .	110
13.9	Interpolacja wielomianowa - predyktor rzędu II . . . . .	111
13.10	Interpolacja wielomianowa - predyktor rzędu IV . . . . .	113
13.11	Interpolacja wielomianowa - funkcja $\sin(x)$ . . . . .	114

14.1	Aproksymacja funkcji za pomocą wielomianów ortogonalnych - przygotowanie danych	136
14.2	Wielomian ortogonalny - współczynniki aproksymacji	136
14.3	Wielomian ortogonalny - postać ogólna	137
14.4	Wielomian ortogonalny - postać wielomianowa	137
14.5	Wielomian ortogonalny - wykres predykcji	137
14.6	Wielomian ortogonalny - współczynniki predykcji	138
14.7	Wielomian ortogonalny - współczynniki predykcji dla próbki nieparzystej numer 1.5	138
14.8	Aproksymacja funkcji schematem liftingu za pomocą wielomianów trygonometrycznych - przygotowane dane	139
14.9	Wielomian trygonometryczny - współczynniki aproksymacji	139
14.10	Wielomian trygonometryczny - postać ogólna	140
14.11	Wielomian trygonometryczny - wykres	140
14.12	Wielomian trygonometryczny - predykcja	141
14.13	Wielomian trygonometryczny - współczynniki predykcji dla próbki nieparzystej numer 3.5	141
15.1	Zależność pomiędzy transformatą klasyczną, a transformatą realizowaną za pomocą schematu liftingu	143
15.2	Wyliczanie wartości współczynników wektora predykcji i uaktualnienia o tej samej długości	146
15.3	Wyliczanie współczynników filtrów analizy $\tilde{h}$ , $\tilde{g}$ i syntezy $h$ , $g$	147
15.4	Analiza funkcji - graficzna reprezentacja funkcji falkowej i skalującej	148
15.5	Synteza funkcji - graficzna reprezentacja funkcji falkowej i skalującej	149
16.1	Wykrywanie nieciągłości funkcji - dane	151
16.2	Wykrywanie nieciągłości funkcji (reprezentacja zgrubna, reprezentacja szczegółów)	152
16.3	Wykrywanie nieciągłości funkcji	153
16.4	Wykrywanie zmiany charakterystyki funkcji I - dane	153
16.5	Wykrywanie zmiany charakteru funkcji I	154
16.6	Wykrywanie zmiany charakteru funkcji II	155
16.7	Usuwanie szumu funkcji - dane	155
16.8	Usuwanie szumu funkcji	156
17.1	Analiza wielorozdzielcza obrazu	159
17.2	Fragment funkcji $\sin(x^2 + y^2)$	161
17.3	Redukcja szumu obrazu - przygotowanie danych	161
17.4	Przykład analizy wielorozdzielczej obrazu	162
17.5	Porównanie wyników podczas usuwania szumu obrazu	163
17.6	Wskaźnik efektywności redukcji szumu	164
17.7	Wskaźnik efektywności redukcji szumu - postać graficzna	164
17.8	Przygotowanie danych w celu analizy nieciągłości obrazu	165
17.9	Porównanie obrazu nieciągłego z jego wygładzonym odpowiednikiem	166
17.10	Filtracja danych - przygotowanie wyników	166
17.11	Filtracja danych - prezentacja wyników	167
17.12	Analiza obrazu - przygotowanie wyników	168
17.13	Analiza danych - prezentacja wyników	169
17.14	Redukcja nieciągłości - przygotowanie wyników	170
17.15	Redukcja nieciągłości - prezentacja wyników	170
17.16	Wskaźnik nieciągłości obrazu - prezentacja wyników	171
17.17	Wskaźnik nieciągłości obrazu - prezentacja graficzna	171

17.18	Pobranie obrazów do postaci tablic dwuwymiarowych w celach analizy . . . . .	172
17.19	Parametry aproksymacji metodą najmniejszych kwadratów . . . . .	172
17.20	Wielorozdzielcza analiza obrazu formatu RGB . . . . .	173
17.21	Aproksymacja obrazu MNK w bloku predykcji dla modelu RGB - I . . . . .	174
17.22	Aproksymacja obrazu MNK w bloku predykcji dla modelu RGB - II . . . . .	174
17.23	Aproksymacja obrazu MNK w bloku predykcji dla modelu RGB - III . . . . .	175
17.24	Graficzna prezentacja zmian kolorów składowych formatu RGB . . . . .	175
17.25	Składowe obrazu w formacie RGB . . . . .	176
17.26	Przygotowanie parametrów aproksymacji do analizy . . . . .	176
17.27	Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB z wykorzystaniem bazy trygonometrycznej . . . . .	177
17.28	Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB z wykorzystaniem bazy liniowej . . . . .	177
17.29	Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB z wykorzystaniem bazy interpolacyjnej . . . . .	178
17.30	Zerowanie współczynników reprezentacji szczegółowej modelu RGB . . . . .	178
17.31	Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB . . . . .	179
17.32	Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB dla składowej R . . . . .	179
17.33	Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB dla składowej G . . . . .	179
17.34	Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB dla składowej B . . . . .	180
17.35	Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu RGB - postać graficzna . . . . .	180
17.36	Macierz przekształcenia przestrzeni RGB do YCbCr [1] . . . . .	181
17.37	Podział obrazu na składowe YCbCr . . . . .	181
17.38	Graficzna prezentacja zmian kolorów składowych formatu YCbCr . . . . .	182
17.39	Analiza składowej Y obrazu za pomocą schematu liftingu w wykorzystaniem aproksymacji MNK (składowa nie podlega aproksymacji podczas transformacji w modelu YCbCr) . . . . .	183
17.40	Analiza składowej Cb obrazu za pomocą schematu liftingu w wykorzystaniem aproksymacji MNK . . . . .	183
17.41	Analiza składowej Cr obrazu za pomocą schematu liftingu w wykorzystaniem aproksymacji MNK . . . . .	184
17.42	Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowych Cb i Cr z wykorzystaniem bazy trygonometrycznej . . . . .	185
17.43	Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowych Cb i Cr z wykorzystaniem bazy liniowej . . . . .	185
17.44	Badanie wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowych Cb i Cr z wykorzystaniem bazy interpolacyjnej . . . . .	186
17.45	Miara oceny efektywności kompresji obrazu RGB w przestrzeni modelu YCbCr . . . . .	186
17.46	Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowej R . . . . .	187
17.47	Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowej G . . . . .	187
17.48	Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr dla składowej B . . . . .	187

17.49	Wskaźnik wrażliwości schematu liftingu na zerowanie współczynników transformaty w modelu YCbCr - postać graficzna . . . . .	188
17.50	Zerowanie współczynników reprezentacji szczegółowej - przykład I . . . . .	188
17.51	Zerowanie współczynników reprezentacji szczegółowej - przykład II . . . . .	189
17.52	Zerowanie współczynników reprezentacji szczegółowej - przykład III . . . . .	189
18.1	Lista procedur i funkcji pakietu << <i>SchematLiftinguZAproksymacjaMNK</i> >>. . . . .	194
18.2	Przykład zależności pomiędzy transformatą klasyczną, a transformatą realizowaną za pomocą schematu liftingu z predyktorem $p = [p_1, p_2]$ i wektorem uaktualnienia $u = [u_1, u_2]$ . . . . .	195

---

# Spis tablic

---

13.1 Schemat liftingu dla CDF(2,2) . . . . .	99
13.2 Schemat liftingu dla nieznormalizowanej falki Haara . . . . .	100
13.3 Schemat liftingu dla rotacji Givensa . . . . .	100
13.4 Dekompozycja dla filtrów analizy biortogonalnych falek CDF(1,x) z $N = 1$ i $\tilde{N} =$ 1, 3, 5 znikającymi momentami . . . . .	101
13.5 Dekompozycja dla filtrów analizy biortogonalnych falek CDF(2,x) z $N = 2$ i $\tilde{N} =$ 2, 4, 6 znikającymi momentami . . . . .	102
13.6 Dekompozycja dla filtrów analizy biortogonalnych falek CDF(4,x) z $N = 4$ i $\tilde{N} =$ 2, 4, 6 znikającymi momentami . . . . .	103

---

# Lista algorytmów

---

6.1	Algorytm Stranga - implementacja . . . . .	40
6.2	Algorytm Stranga - wykreślanie funkcji skalującej . . . . .	41
7.1	Algorytm Stranga - wykreślanie funkcji falkowej . . . . .	44
8.1	Wyznaczanie funkcji skalującej metodą iteracyjną . . . . .	48
9.1	Aproksymacja Haara . . . . .	64
13.1	Wyznaczanie macierzy polifazowej . . . . .	97