
9 Identyfikacja systemów za pomocą falek – metody obliczeniowe

Przemysław Śliwiński

9.1 Wstęp

Badania nad zastosowaniem falek do identyfikacji charakterystyk nieliniowych w systemach dynamicznych o strukturze blokowej przy nieparametrycznej informacji wstępnej zaowocowały dotychczas szeregiem prac ustalających własności statystyczne (zbieżność i szybkość zbieżności do poszukiwanej nieliniowości) opartych o nie algorytmów (zob. [44, 27, 28, 29, 31, 30, 49] i por. wstęp do rozdziału 8.). Porównując algorytmy falkowe z algorytmami z wcześniejszych prac (zob. np. [19, 21, 20, 22, 37, 43, 42, 48]), w których do tego zadania używano funkcji jądrowych bądź klasycznych szeregów ortogonalnych (Fouriera, Hermite’a, Laguerra i Legendre’a) stwierdzono, m.in., że łączą one w sobie zalety:

- algorytmów jądrowych (dobre własności lokalizujące, przydatne w przypadku nieregularnych i nieciągłych nieliniowości), oraz
- algorytmów ortogonalnych (brak potrzeby przechowywania pomiarów po wyznaczeniu współczynników empirycznych rozwinięcia).

W niniejszym rozdziale przedmiotem rozważań są zagadnienia związane z efektywną implementacją falkowych algorytmów identyfikacji. Pokażemy, jak wykorzystując specyficzne własności funkcji falkowych uzyskać algorytmy obliczeniowe posiadające, charakterystyczną dla algorytmów falkowych z innych dziedzin, szybkość i prostotę, zachowując przy tym korzystne własności statystyczne algorytmów „teoretycznych” przedstawione w rozdziale 8.

Wyjątkowa popularność ortogonalnych funkcji falkowych o zwartym nośniku, jaką można obecnie obserwować w rozmaitych dziedzinach nauki, np. w samej matematyce: w aproksymacji, [14], [32, 33], w numerycznych algorytmach rozwiązywania cząstkowych równań różniczkowych, [10], w algebrze liniowej, [47], w statystyce, [4], [41], czy też szczególnie, śledząc ich zastosowania w przetwarzaniu sygnałów: w kompresji obrazów¹ i filmów, [1, 2], [38], w kompresji dźwięku, [9], odszumianiu danych, [16, 17, 15, 18], [26], analizie danych biomedycznych [55], wydaje się mieć wspólne przyczyny, wśród których, jako najważniejsze można wymienić dwie:

¹Prominentnym przykładem jest tutaj nowy standard kompresji obrazów bitowych JPEG2000 (<http://www.jpeg.org/jpeg2000/>) – następcą popularnego standardu JPEG, zob. [53].

1. Oszczędność falkowej reprezentacji szerokiej klasy sygnałów i funkcji.
Za pomocą niewielkiej liczby współczynników rozwinięcia falkowego można wyrazić sygnały i funkcje gładkie, jak i posiadające nieciągłości [14], [39], [10].
2. Prostota i szybkość algorytmów numerycznych przekształcających sygnały (funkcje) w ciągi współczynników falkowych [39], [12].
Złożoność obliczeniowa szybkiego przekształcenia falkowego (*Fast Wavelet Transform, FWT*) jest rzędu $\mathcal{O}(N)$, a zatem jest nawet mniejsza niż złożoność algorytmu FFT (*Fast Fourier Transform, FFT*), dla którego – przypomnijmy – jest ona rzędu $\mathcal{O}(N \log N)$ (zob. np. [51]).

Uzyskanie takich własności okupione zostało jednakże brakiem jawnych wzorów na funkcje falkowe: nie posiadają one – poza znanym już wcześniej systemem ortogonalnym Haara [24] – zwartej postaci, a definiowane są jedynie *implicite*, poprzez tak zwane *równania skalujące* (*scaling equations* lub *dilation equations*)

$$\varphi(x) = \sqrt{2} \sum_l h_l \varphi(2x-l) \quad \text{oraz} \quad \psi(x) = \sqrt{2} \sum_l g_l \varphi(2x-l) \quad (9.1)$$

i odpowiednie zbiory współczynników $\{h_l\}$ i $\{g_l = (-1)^l h_{1-l}\}$ [11, 40, 39]; zob. Dodatek A.

Okazuje się jednak, że w wymienionych wyżej zastosowaniach brak jawnej postaci funkcji falkowych nie odgrywa znaczącej roli, ponieważ w odpowiadających im metodach obliczeniowych, opartych zwykle o szybkie przekształcenie falkowe, wykorzystuje się tylko znajomość współczynników $\{h_l\}$ i $\{g_l\}$ – stanowi on natomiast istotną przeszkodę w bezpośrednim zastosowaniu algorytmów falkowych przedstawionych i badanych w rozdziale 8. Potrzeba skonstruowania odpowiednich wersji algorytmów obliczeniowych wynika tu, w szczególności, z następujących przyczyn:

1. W rozważanym zadaniu identyfikacji zakłada się losowy charakter sygnału wejściowego oraz istnienie gęstości prawdopodobieństwa tego sygnału. W konsekwencji, zmierzony sygnał wejściowy może przybierać dowolne wartości rzeczywiste.
2. Znane metody obliczeniowe pozwalają wyznaczyć dokładne wartości funkcji falkowych jedynie w punktach binarnych (o skończonym rozwinięciu dwójkowym; tj. w punktach $2^{-H}i$, gdzie H, i są całkowite i skończone)²; zob. [11, 50] i Dodatek B.

W rozdziale prezentujemy proste rozwiązanie tego problemu oparte na następującym spostrzeżeniu:

²Warto przypomnieć, że zbiór ten jest podzbiorem właściwym zbioru liczb wymiernych \mathcal{Q} . Przykładem liczby o nieskończonym rozwinięciu dwójkowym jest ułamek dziesiętny $\frac{1}{10}$.

Możliwe do wyznaczenia wartości funkcji falkowych w punktach binarnych możemy potraktować jako „próbki” tych funkcji i stabelaryzować, a następnie, posługując się interpolacją, przybliżać wartości funkcji falkowych w pozostałych punktach.

9.2 Szybki algorytm obliczeniowy

Konstruowanie obliczeniowych odpowiedników algorytmów uzyskanych na drodze rozważań teoretycznych ma na celu takie ich zmodyfikowanie, aby obliczenia wynikające z tych rozważań mogły zostać wykonane (por. np. [5, 34, 45, 46]):

- w jak najkrótszym czasie
- z użyciem jak najmniejszej ilości pamięci, oraz
- w taki sposób, by wynik obliczeń był najbliższy teoretycznemu.

W tej części rozdziału proponujemy algorytm obliczeniowy dla falkowego algorytmu identyfikacji (8.15)-(8.16) i (8.19) z rozdziału 8. wykorzystując szybkie przekształcenie falkowe do wyznaczenia współczynników empirycznych i zbadamy go pod kątem pierwszych dwóch kryteriów.³

Algorytm. Algorytm składa się z następujących kroków (kroki 1-4, służące do wyznaczenia współczynników modelu, wykonywane są tylko raz dla posiadanego zbioru pomiarów):

1. Wyznaczenie skali aproksymacji K modelu dla danej liczby pomiarów N :

$$K(N) = \left\lfloor \frac{1}{2^{\lambda_g \psi + 1}} \log_2 N \right\rfloor \quad \text{albo} \quad K(N) = \left\lfloor \frac{1}{3} \log_2 N \right\rfloor \quad (9.2)$$

gdzie $\lambda_g \psi = \min \{ \nu_R, \nu_f, r_\psi + 1 \}$ (por. (8.24) i (8.25) w rozdziale 8).⁴

2. Wyznaczenie granic indeksów n współczynników modelu:

$$\begin{aligned} n_{\min}(\varphi, M) &= \lfloor 2^M c - s_2 \rfloor + 1, & n_{\min}(\psi, m) &= \lfloor 2^m c - t_2 \rfloor + 1 \\ n_{\max}(\varphi, M) &= \lceil 2^M d - s_1 \rceil - 1, & n_{\max}(\psi, m) &= \lceil 2^m d - t_1 \rceil - 1 \end{aligned} \quad (9.3)$$

gdzie $c = \min_{k=1, \dots, N} \{x_k\}$ oraz $d = \max_{k=1, \dots, N} \{x_k\}$ i gdzie $\lfloor \cdot \rfloor$ i $\lceil \cdot \rceil$ oznaczają funkcje podłoga i sufit (ang. *floor and ceiling functions*; por. np. [35]), natomiast s_1, s_2 oraz t_1, t_2 to, odpowiednio, lewe i prawe granice nośników funkcji skalujących φ i falek ψ .

³Zagadnieniom związanym z trzecim z kryteriów poświęcona jest następna część pracy; zob. rozdz. 9.3 na str. 108.

⁴Pierwszy ze wzorów ma znaczenie teoretyczne i jest wykorzystywany do analizy własności algorytmów falkowych. Drugi posiada istotny walor praktyczny, ponieważ do wyznaczenia skali K wymaga jedynie znajomości liczby pomiarów N ; por. [30, 49].

3. Obliczenie pomocniczych współczynników falkowych $\hat{\alpha}_{K_n}, \hat{a}_{K_n}$ na podstawie pomiarów $\{(x_k, y_k)\}_{k=1}^N$

$$\left. \begin{aligned} \hat{\alpha}_{K_n+} &= \varphi_{K_n}(x_k) \cdot y_k \\ \hat{a}_{K_n+} &= \varphi_{K_n}(x_k) \end{aligned} \right\} \text{ dla } n = n_{\min}(x_k; \varphi, K), \dots, n_{\max}(x_k; \varphi, K) \quad (9.4)$$

gdzie zakresy indeksów n „aktywnych” współczynników wyznaczane są dla kolejnych wartości x_k według wzorów (por. (9.3)):

$$\begin{aligned} n_{\min}(x_k; \varphi, M) &= \lfloor 2^M x_k - s_2 \rfloor + 1 \\ n_{\max}(x_k; \varphi, M) &= \lceil 2^M x_k - s_1 \rceil - 1 \end{aligned} \quad (9.5)$$

4. Obliczenie współczynników modelu w oparciu o szybkie przekształcenie falkowe (por. (9.1))

$$\begin{aligned} \hat{\beta}_{mn} &= \sqrt{2} \sum_l g_l \hat{\alpha}_{m+1, 2n+l}, & \hat{\alpha}_{mn} &= \sqrt{2} \sum_l h_l \hat{\alpha}_{m+1, 2n+l} \\ \hat{b}_{mn} &= \sqrt{2} \sum_l g_l \hat{a}_{m+1, 2n+l}, & \hat{a}_{mn} &= \sqrt{2} \sum_l h_l \hat{a}_{m+1, 2n+l} \end{aligned} \quad (9.6)$$

5. Obliczenie wartości modelu dla wybranego argumentu x_0

$$\hat{R}(x_0; K) = \frac{\sum_{n=n_{\min}(x_0; \varphi, M)}^{n_{\max}(x_0; \varphi, M)} \hat{\alpha}_{Mn} \varphi_{Mn}(x_0) + \sum_{m=M}^{K-1} \sum_{n=n_{\min}(x_0; \psi, m)}^{n_{\max}(x_0; \psi, m)} \hat{\beta}_{mn} \psi_{mn}(x_0)}{\sum_{n=n_{\min}(x_0; \varphi, M)}^{n_{\max}(x_0; \varphi, M)} \hat{a}_{Mn} \varphi_{Mn}(x_0) + \sum_{m=M}^{K-1} \sum_{n=n_{\min}(x_0; \psi, m)}^{n_{\max}(x_0; \psi, m)} \hat{b}_{mn} \psi_{mn}(x_0)} \quad (9.7)$$

gdzie (por. (9.5))

$$\begin{aligned} n_{\min}(x_0; \psi, m) &= \lfloor 2^m x_0 - t_2 \rfloor + 1 \\ n_{\max}(x_0; \psi, m) &= \lceil 2^m x_0 - t_1 \rceil - 1 \end{aligned}$$

9.2.1 Własności obliczeniowe algorytmu szybkiego

Złożoność obliczeniowa. Wyznaczenie skali K ze wzorów w (9.2) ma oczywiście stałą złożoność, tj. $\mathcal{O}(1)$. Z kolei, wyznaczenie granic indeksów w (9.3) wymaga $\mathcal{O}(N)$ operacji porównania.

Przyjmując (chwilowo), że koszt obliczenia wartości funkcji falkowej jest stały, rozważmy koszt obliczenia współczynników pomocniczych $\hat{\alpha}_{K_n}$. Ponieważ (ze względu na zwartość nośników funkcji falkowych) każda z N par pomiarów bierze udział w $n_{\max}(x_k; \varphi, K) - n_{\min}(x_k; \varphi, K) = \mathcal{O}(1)$ obliczeniach, to łatwo można obliczyć, że liczba operacji **opms**⁵ potrzebna do wyznaczenia tych współczynników jest rzędu $\mathcal{O}(N)$.

Zauważmy teraz, że dla malejących skal m maleje także (w przybliżeniu dwukrotnie, zob. wzory (9.8)-(9.9)) liczba współczynników $\hat{\beta}_{mn}$ i \hat{b}_{mn} modelu. Ponieważ liczba niezerowych współczynników $\{h_l\}$ i $\{g_l\}$ jest skończona

⁵Skrót **opms** oznacza jedną ze skalnych operacji arytmetycznych $\{+, -, \cdot, \div, \sqrt{\cdot}\}$; por. [34].

i ustalona dla wybranej rodziny falek o zwartym nośniku – zob. przykład dla falek Daubechies w Dodatku A – to łatwo z kolei obliczyć, że operacje w (9.6), wykonywane są

$$\mathcal{O} \left(2 \sum_{m=M}^{K-1} 2^m \right) = \mathcal{O} (2 (2^{K-1} + 2^{K-2} + \dots + 2^{M+1} + 2^M)) = \mathcal{O} (2^K) \text{ razy,}$$

Uwzględniając teraz, że wyznaczenie wartości modelu w wybranym punkcie x_0 ma złożoność $\mathcal{O}(K)$, otrzymujemy łącznie

$$\mathcal{Z}(K) = \mathcal{O}(1) + \mathcal{O}(N) + \mathcal{O}(N) + \mathcal{O}(2^K) + \mathcal{O}(K) = \mathcal{O}(N)$$

dla K ze wzorów w (9.2) i przy dowolnym $\lambda_{g\psi} > 0$.⁶

Rozmiar zajmowanej pamięci. W oparciu o granice sumowań w (9.3) możemy znaleźć liczbę współczynników empirycznych w modelu falkowym – a stąd rząd rozmiaru pamięci potrzebnej do ich przechowania. Oznaczmy w tym celu przez $\mathcal{P}(m)$ liczbę współczynników w poszczególnych skalach m . Ponieważ

$$\begin{aligned} \mathcal{P}(m) &= n_{\max}(\varphi, m) - n_{\min}(\varphi, m) + 1 = \\ &= n_{\max}(\psi, m) - n_{\min}(\psi, m) + 1 \\ &= \lceil 2^m d - t_1 - 1 \rceil - \lfloor 2^m c - t_2 + 1 \rfloor + 1, \end{aligned} \quad (9.8)$$

stąd, korzystając z oszacowania ($t = t_2 - t_1 = s_2 - s_1$)

$$\begin{aligned} \mathcal{P}(m) &< (2^m d - t_1 + 1) - (2^m c - t_2 - 1) - 1 = 2^m (d - c) + t + 1 \\ &= \mathcal{O}(2^m) \end{aligned} \quad (9.9)$$

otrzymujemy odpowiednie oszacowanie liczby współczynników \mathcal{P}_K dla całego estymatora

$$\begin{aligned} \mathcal{P}(K) &= 2 \left(\mathcal{P}(M) + \sum_{m=M}^{K-1} \mathcal{P}(m) \right) < \\ &< 2 \left[2^K (d - c) + 2(K - M + 1)(t + 1) \right] = \mathcal{O}(2^K) \end{aligned} \quad (9.10)$$

Algorytm oparty o szybkie przekształcenie falkowe posiada zatem następujące właściwości:

- + Złożoność obliczeniowa $\mathcal{Z}(K) = \mathcal{O}(N)$ jest równa co do rzędu złożoności oryginalnego algorytmu FWT i jest niezależna od gładkości identyfikowanej nieliniowości μ (i funkcji gęstości f).
- + Do jego realizacji można wykorzystać gotowe pakiety oprogramowania (MatLAB, Mathematica) implementujące algorytm FWT.

⁶Uzyskana złożoność $\mathcal{O}(N)$ dotyczy wyznaczenia wartości modelu "za pierwszym razem". Kolejne wartości – już w oparciu o obliczone współczynniki modelu – wyznaczone są zatem z mniejszą złożonością $\mathcal{Z}(K) = \mathcal{O}(K) = \mathcal{O}(\log N)$, dla K ze wzorów (9.2).

- Wymaga dodatkowej pamięci dla przechowania pośrednich współczynników empirycznych $\hat{\alpha}_{mn}$ dla każdej skali $m = K - 1, \dots, M^7$.

Przykład 9.1 Wykorzystanie praktycznej reguły doboru skali K ze wzoru (9.2) pozwala uzyskać oszacowanie ilości potrzebnej pamięci rzędu $\mathcal{P}(K) = \mathcal{O}(N^{1/3})$ oraz złożoność równą $\mathcal{Z}(K) = \mathcal{O}(N)$.

Porównanie wydajności algorytmu w stosunku do algorytmu opartego wprost na wzorach (8.15)-(8.16) i (8.19) z rozdziału 8. prezentujemy na Rys. 9.1. Jak można zaobserwować, szybkość algorytmu z FWT może być nawet dziesięciokrotnie większa.

9.3 Algorytm uproszczony

Powyższa analiza nie uwzględnia kosztu wyznaczenia wartości funkcji falkowych, który - ze względu na wspomniane już na wstępie specyficzne własności funkcji falkowych - jest zmienny i zależy od wartości argumentu, a znane procedury pozwalają wyznaczać te wartości jedynie w punktach o skończonej reprezentacji dwójkowej; zob. Dodatek B. Uwzględniając te ograniczenia, proponujemy poniżej algorytm, w którym funkcje falkowe zastąpione zostaną ich interpolacjami. Pokażemy, że taki zabieg pozwala:

- Utrzymać ogólną postać algorytmów teoretycznych (por. (9.4)-(9.7)).
- Uzyskać tę samą złożoność obliczeniową jaką wyznaczyliśmy powyżej dla oryginalnych algorytmów z funkcjami falkowymi.
- Zachować (przy dodatkowych założeniach dotyczących dokładności interpolacji) korzystne własności statystyczne algorytmów oryginalnych⁸.

9.3.1 Konstrukcja interpolacji funkcji falkowych

Ideę algorytmu i jego własności zaprezentujemy na przykładzie interpolacji pierwszego rzędu (tj. za pomocą funkcji prostych - odcinkami stałych); zob. np. [54]. Interpolację taką można zdefiniować formalnie jako⁹

$$\bar{\varphi}(x; H) = \varphi\left(\frac{\lfloor 2^H x \rfloor}{2^H}\right) \quad (9.12)$$

⁷Od tej wady wolna jest wersja algorytmu FWT oparta o tzw. algorytm retuszujący (ang. *lifting scheme*) zaproponowany przez Sweldensa w [12]. Algorytm ten pozwala bowiem obliczyć transformatę *in situ*.

⁸A zatem spełnić trzecie z wymagań stawianych algorytmom obliczeniowym; por. str. 105.

⁹Dokładność interpolacji można poprawić stosując wzór

$$\bar{\varphi}^H(x) = \varphi\left(\frac{\lfloor 2^H x + 1/2 \rfloor}{2^H}\right) \quad (9.11)$$

ponieważ wartości $\varphi(x)$ przybliżane są tu przez wartości dla najbliższego sąsiada.

skąd wynika, że wartości funkcji falkowych φ w dowolnym punkcie x są przybliżane za pomocą wartości tych funkcji w najbliższym (z lewej strony) punkcie leżącym na siatce binarnej 2^{-H} . Interpolacje przeskalowanych i przesuniętych wersji funkcji φ_{mn} otrzymujemy korzystając ze zmodyfikowanej formuły

$$\bar{\varphi}_{mn}(x; H) = \varphi \left(\frac{\lfloor 2^{H+m} \cdot 2^m x \rfloor}{2^{H+m}} - n \right) \quad (9.13)$$

dla której zachodzi ta korzystna własność, że wraz ze wzrostem skali m rośnie jednocześnie dokładność interpolacji. Interpolacje falek ψ i ich przeskalowanych i przesuniętych wersji ψ_{mn} otrzymujemy natychmiast z $\bar{\varphi}_{mn}(x; H)$, korzystając z ogólnej zależności w (9.1).

Uwaga 9.1 Notacja zastosowana w (9.12)-(9.13) – dogodna podczas badania własności algorytmów – odpowiada również prostej i efektywnej implementacji opartej o wcześniej obliczone i przechowywane w pamięci wartości funkcji φ w punktach siatki binarnej 2^{-H} .

Uproszczony algorytm obliczeniowy otrzymujemy przez zastąpienie oryginalnych funkcji falkowych φ_{Mn} i ψ_{mn} we wzorach (9.4)-(9.7), ich interpolacjami $\bar{\varphi}_{Mn}(x; H)$ i $\bar{\psi}_{mn}(x; H)$.

9.3.2 Własności statystyczne algorytmu uproszczonego

Zastosowanie interpolacji w miejsce oryginalnych funkcji falkowych powoduje pojawienie się dodatkowego błędu. Z następującego oszacowania tego błędu (podobne oszacowanie zachodzi falek ψ_{mn} i ich interpolacji $\bar{\psi}_{mn}(x; H)$)

$$\begin{aligned} \varphi_{mn}(x) - \bar{\varphi}_{mn}(x; H) &= 2^{\frac{m}{2}} \left[\varphi(2^m x - n) - \varphi \left(\frac{\lfloor 2^{H+m} \cdot 2^m x \rfloor}{2^{H+m}} - n \right) \right] \\ &\leq 2^{\frac{m}{2}} c \left| 2^m x - \frac{\lfloor 2^{H+m} \cdot 2^m x \rfloor}{2^{H+m}} \right|^\eta = 2^{\frac{m}{2}} c \left| \frac{2^{2m} x - \lfloor 2^{H+m} \cdot 2^m x \rfloor}{2^{H+m}} \right|^\eta \\ &= \mathcal{O} \left(2^{\frac{m}{2}} \cdot 2^{-\eta(H+m)} \right), \quad c > 0, \eta = \min \{ \rho, 1 \} \end{aligned}$$

gdzie ρ jest wykładnikiem Höldera funkcji skalującej, wynika, że im większa skala interpolacji H , tym dokładniej $\bar{\varphi}(x; H)$ przybliży oryginalną funkcję skalującą φ . Można dalej pokazać, że dla uproszczonego algorytmu obliczeniowego zachodzi następujące oszacowanie (por. (8.27) w rozdziale 8.)

$$\bar{R}(x_0; K, H) = R(x_0) + \mathcal{O} \left(\left(2^{-2\lambda_g \psi K} + 2^{-2\eta H} + \frac{2^K}{N} \right)^{1/2} \right) \quad (9.14)$$

w którym kolejne składowe to oszacowania błędu aproksymacji, błędu interpolacji oraz błędu wariancji. Aby zatem zmniejszyć wpływ błędu interpolacji na zachowanie się algorytmu obliczeniowego, należy przyjąć jak największą wartość H . Wiąże się to jednak albo ze znacznym wzrostem złożoności obliczeniowej

(jeśli odpowiednie wartości funkcji falkowych są obliczane *on-line*)¹⁰, albo z istotnym zwiększeniem się rozmiaru potrzebnej pamięci (gdy wyznaczone wartości przechowujemy w tablicy)¹¹. Okazuje się, że można tak dobrać wartość skali H tak, aby – nie zmieniając asymptotycznych własności algorytmu teoretycznego – w jak najmniejszym stopniu zużyć zasoby komputera. W tym celu wystarczy zauważyć, że:

Rząd błędu algorytmu $\bar{R}(x; K, H)$ nie ulegnie zmianie (a w konsekwencji, zachowane zostają asymptotyczne własności algorytmu teoretycznego $\bar{R}(x; K)$) jeśli skalę interpolacji H dobierzemy względem skali K , tak aby zachodziła nierówność

$$2^{-2\lambda_g \psi K} \geq 2^{-2\eta H}. \quad (9.15)$$

Na podstawie tego spostrzeżenia można zaproponować prostą regułę doboru skali interpolacji

$$H = \left\lceil \frac{\lambda_g \psi}{\eta} K \right\rceil \quad (9.16)$$

dzięki której oszacowanie w (9.14) staje się, co do rzędu, równe oszacowaniu błędu algorytmu teoretycznego (8.27) z rozdziału 8. Reguła ta jednakże, podobnie jak pierwsza reguła doboru skali K w (9.2), ma charakter teoretyczny, ponieważ zwykle wartość parametru $\lambda_g \psi$ jest nieznana. Alternatywną jej wersją, jest poniższa

$$H' = \left\lceil \frac{r_\psi + 1}{\eta} K \right\rceil \quad (9.17)$$

gdzie $r_\psi + 1$ jest liczbą znikających momentów falki ψ (por. Tabela 8.1 w rozdziale 8). Reguła ta ma walor **praktyczny**, bo korzystamy w niej jedynie ze znanej skali K i dostępnych informacji o własnościach funkcji falkowych, a ponadto jest **bezpieczna**, ponieważ niezależnie od trafności doboru rodziny falkowej, nie pogarszamy własności algorytmu obliczeniowego względem własności jego teoretycznego odpowiednika. Zauważmy bowiem, że jeśli $r_\psi + 1 > \min\{\nu_R, \nu_f\}$, to faktyczny rząd błędu aproksymacji wynosi $\mathcal{O}(2^{-2\min\{\nu_R, \nu_f\}K})$ i rząd błędu interpolacji, równy $\mathcal{O}(2^{-2(r_\psi+1)K})$, jest wówczas mniejszy. Jeśli natomiast zachodzi odwrotna sytuacja (tj. $r_\psi + 1 < \min\{\nu_R, \nu_f\}$), to oba błędy są tego samego rzędu $\mathcal{O}(2^{-2(r_\psi+1)K})$.

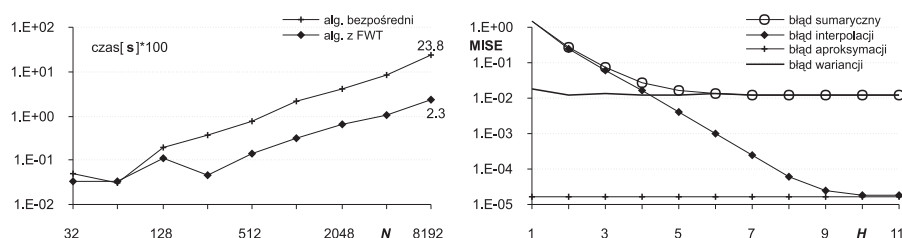
Przykład 9.2 Przyjmując, że skala K została dobrana według reguły praktycznej z (9.2), otrzymujemy natychmiast, że (por. (9.15))

$$H = \left\lceil \frac{1}{3} \log_2 N \right\rceil \quad (9.18)$$

Wyjątek stanowi rodzina falek Daubechies o numerze 2, dla której – ponieważ $\eta = 0.55 < 1$ – reguła praktyczna ma postać $H = \lceil 2/3 \log_2 N \rceil$.

¹⁰Do wyznaczenia wartości funkcji falkowej dla argumentu o rozwinięciu dwójkowym o H cyfrach, wymagana liczba obliczeń jest rzędu $\mathcal{O}(2^H)$; zob. Dodatek B.

¹¹Do zapamiętania interpolacji funkcji falkowych z rozdzielczością 2^{-H} , ilość potrzebnej pamięci jest rzędu $\mathcal{O}(2^H)$.



Rysunek 9.1. Porównanie szybkości działania algorytmów (po lewej). Porównanie błędów składowych w algorytmie uproszczonym (po prawej)

9.4 Zakończenie

Wykazaliśmy, że (dzięki zaprojektowanym uproszczonym algorytmom obliczeniowym) algorytmy identyfikacji oparte o funkcje falkowe łączą w sobie korzystne cechy statystyczne jak i numeryczne. W szczególności zatem:

- Dziedziczą one po algorytmach teoretycznych optymalną szybkość zbieżności.
- Posiadają optymalną złożoność obliczeniową.

W ramach dalszych prac badawczych planuje się:

- Opracowanie i ustalenie własności tych algorytmów, opartych na falek bi-ortogonalnych i semi-ortogonalnych, [52, 8], i wykorzystujących funkcje interpolujące wyższych rzędów, [6, 7].
- Opracowanie i zbadanie algorytmów semi-rekurencyjnych
- Udostępnienie implementacji algorytmów w sieci Internet w postaci *usług sieciowych* (ang. Web Services), programów źródłowych w języku C++ oraz jako skrypty dla środowiska MatLAB.

Na zakończenie warto podkreślić, że w literaturze, opisywane w rozdziale problemy obliczeniowe dotyczące funkcji falkowych w dowolnych (losowych) punktach nie znalazły jak dotąd pełnego rozwiązania (w odniesieniu do zadania identyfikacji nieliniowości). Znane podobne wyniki koncentrują się jedynie na wyznaczeniu falkowych współczynników empirycznych; zob. np. prace: [13], [3], [25], [26], [36], [23].

Dodatek A: Współczynniki falek Daubechies

Współczynniki $\{h_i\}_{i=0}^{2p-1}$ funkcji falkowych Daubechies o numerze p można szczególnie łatwo wyznaczyć z następujących warunków (zob. np. [45, str. 592] i [51,

str. 298])

$$\begin{aligned} \text{Ortonormalność :} & \quad \begin{cases} \sum_{t=0}^{2p-1} h_t h_{t+2n} = \delta_{0,n}, & n = 0, \dots, p-1 \end{cases} \\ \text{Znikające momenty :} & \quad \begin{cases} \sum_{l=0}^{2p-1} (-1)^l h_{(2p-1)-l} \cdot l^r = 0, & r = 0, \dots, p-1 \end{cases} \end{aligned}$$

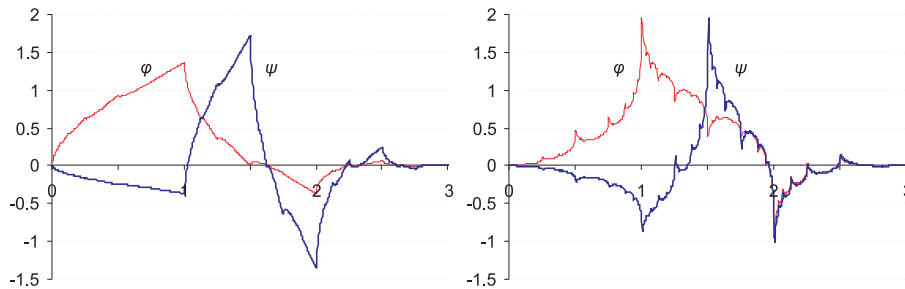
które tworzą, dla każdego p , układ $2p$ równań o $2p$ niewiadomych h_l .

Przykład 9.3 Dla $p = 2$ układ równań przyjmuje następującą postać:

$$\begin{cases} h_0^2 + h_1^2 + h_2^2 + h_3^2 = 1 & : (n = 0) \\ h_0 h_2 + h_1 h_3 = 0 & : (n = 1) \\ h_3 - h_2 + h_1 - h_0 = 0 & : (r = 0) \\ 0h_3 - 1h_2 + 2h_1 - 3h_0 = 0 & : (r = 1) \end{cases} \quad (9.19)$$

którego rozwiązaniem są współczynniki

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



Rysunek 9.2. Wykresy funkcji falkowych: klasyczna para funkcji falkowych Daubechies o numerze $p = 2$ (i $r_\psi = 1$ znikających momentach) (**po lewej**). Przykład egzotycznej pary funkcji falkowych, dla której nie jest spełnione jedynie ostatnie równanie w układzie (9.19) – stąd $r_\psi = 0$ (**po prawej**)

Dodatek B: Obliczanie wartości falek

W pracy Stranga (zob. [51] i por. np. [11]) zaproponowano następujący algorytm wyznaczania wartości funkcji skalujących w punktach binarnych¹². W algorytmie tym zakłada się, że wartości funkcji skalującej dla argumentów całkowitych ($n = 0, 1, \dots, s$), są znane. W oparciu o wzór (9.1) można zatem wyznaczyć

¹²Przykładowa implementacja tego algorytmu dla środowiska MatLAB znajduje się na stronie: <http://web.mit.edu/1.130/WebDocs/1.130/Software/Examples/phivals.m>.

wartości w punktach $\frac{n}{2}$, a sukcesywnie jego stosowanie pozwala obliczyć wartości funkcji skalującej φ dla dowolnych argumentów $b, b \in [0, 1)$, leżących na siatce binarnej 2^{-H} , $H \geq 1$.

Niech teraz

$$\varphi(b) = [\varphi(b) \quad \varphi(b+1) \quad \dots \quad \varphi(b+s-1)]^T$$

oznacza wektor wartości funkcji skalującej w punktach $b, b+1, \dots, b+s-1$. Wartości w tych punktach obliczane są za pomocą wzoru

$$\varphi(b) = \prod_{h=1}^H [(1-b_h)\mathbf{A} + b_h\mathbf{B}] \varphi(0)$$

gdzie $b_h \in \{0, 1\}$ są kolejnymi wyrazami rozwinięcia dwójkowego argumentu b , a \mathbf{A} i \mathbf{B} są macierzami kwadratowymi o rozmiarze $s \times s$, których elementy zdefiniowane są za pomocą współczynników $\{h_l\}$:

$$\begin{aligned} \mathbf{A} &= [a_{ij}] = h_{2i-j} & , \quad i, j = 0, 1, \dots, s-1 \\ \mathbf{B} &= [b_{ij}] = h_{2i-j+1} \end{aligned}$$

Przykład 9.4 Dla falek Daubechies o numerze 2, $s = 3$. Macierze \mathbf{A} i \mathbf{B} są równe

$$\mathbf{A} = \begin{bmatrix} h_0 & 0 & 0 \\ h_2 & h_1 & h_0 \\ 0 & h_3 & h_2 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} h_1 & h_0 & 0 \\ h_3 & h_2 & h_1 \\ 0 & 0 & h_3 \end{bmatrix}$$

natomiast wektor wartości funkcji skalującej w punktach całkowitych jest równy

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

Niech $b = 9/16 = 0.5625_{10} = 0.1001_2$, wówczas

$$\varphi(0.5625) = \varphi(0.1001) = \mathbf{BAAB}\varphi(0) = \begin{bmatrix} \frac{63+33\sqrt{3}}{512} & \frac{5-\sqrt{3}}{256} & \frac{55-31\sqrt{3}}{512} \end{bmatrix}^T.$$

9.5 Bibliografia

- [1] Special issue on JPEG2000 standard. *IEEE Signal Processing Magazine*, 18, 2001.
- [2] Special issue on JPEG 2000 standard. *Signal Processing: Image Communication*, 17, 2002.
- [3] A. Antoniadis, G. Grégoire, and P. Vial. Random design wavelet curve smoothing. *Statistics and Probability Letters*, 35:225–232, 1997.
- [4] A. Antoniadis and G. Oppenheim. *Wavelets and Statistics*. Springer-Verlag, New York, 1995.
- [5] A. Björk and G. Dahlquist. *Metody numeryczne*. Państwowe Wydawnictwo Naukowe, Warszawa, 1983.

-
- [6] T. Blu, P. Thévenaz, and M. Unser. Complete parameterization of piecewise-polynomial interpolation kernels. *IEEE Transactions on Image Processing*, 12(11):1297–1309, November 2003.
 - [7] T. Blu, P. Thévenaz, and M. Unser. Linear interpolation revitalized. *IEEE Transactions on Image Processing*, 13(5):710–719, May 2004.
 - [8] T. Blu and M. Unser. Wavelets, fractals, and radial basis functions. *IEEE Transactions on Signal Processing*, 50(3):543–553, March 2002.
 - [9] C. K. Chui. *Wavelets: A mathematical tool for signal processing*. SIAM, Philadelphia, 1997.
 - [10] A. Cohen. *Numerical analysis of wavelets methods*. Studies in Mathematics and Its Applications. Elsevier, Amsterdam, 2003.
 - [11] I. Daubechies. *Ten Lectures on Wavelets*. SIAM Edition, Philadelphia, 1992.
 - [12] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.
 - [13] B. Delyon and A. Juditsky. Estimating wavelet coefficients. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, pages 151–168. Springer-Verlag, New York, 1995.
 - [14] R. A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.
 - [15] D. L. Donoho. De-noising by soft thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995.
 - [16] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, 1994.
 - [17] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90:1200–1224, 1995.
 - [18] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard. Density estimation by wavelet thresholding. *Annals of Statistics*, 24:508–539, 1996.
 - [19] W. Greblicki. Nonparametric system identification by orthogonal series. *Problems of Control and Information Theory*, 8:67–73, 1979.
 - [20] W. Greblicki. Nonparametric orthogonal series identification of Hammerstein systems. *International Journal of Systems Science*, 20:2355–2367, 1989.
 - [21] W. Greblicki and M. Pawlak. Hammerstein system identification by nonparametric regression estimation. *International Journal of Control*, 45:343–354, 1987.
 - [22] W. Greblicki and M. Pawlak. Nonparametric recovering nonlinearities in block oriented systems with the help of Laguerre polynomials. *Control – Theory and Advanced Technology*, 10:771–791, 1994.
 - [23] L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer-Verlag, New York, 2002.
 - [24] A. Haar. Zur Theorie der Orthogonalen Funktionen-Systeme. *Annals of Mathematics*, 69, 1910.
 - [25] P. Hall and B. Turlach. Interpolation methods for nonlinear wavelet regression with irregularly spaced design. *The Annals of Statistics*, 25(5):1912–1925, 1997.
 - [26] W. Härdle, G. Kerkyacharian, D. Picard, and A. Tsybakov. *Wavelets, Approximation, and Statistical Applications*. Springer-Verlag, New York, 1998.

- [27] Z. Hasiewicz. Hammerstein system identification by the Haar multiresolution approximation. *International Journal of Adaptive Control and Signal Processing*, 13:697–717, 1999.
- [28] Z. Hasiewicz. Modular neural networks for non-linearity recovering by the Haar approximation. *Neural Networks*, 13:1107–1133, 2000.
- [29] Z. Hasiewicz. Non-parametric estimation of non-linearity in a cascade time series system by multiscale approximation. *Signal Processing*, 81:791–807, 2001.
- [30] Z. Hasiewicz, M. Pawlak, and P. Śliwiński. Non-parametric identification of nonlinearities in block-oriented complex systems by orthogonal wavelets with compact support. Przyjęte do publikacji w: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 2005.
- [31] Z. Hasiewicz and P. Śliwiński. Identification of non-linear characteristics of a class of block-oriented non-linear systems via Daubechies wavelet-based models. *International Journal of Systems Science*, 14:1121–1144, 2002.
- [32] S. Jaffard. Functions with prescribed Hölder exponent. *Applied and Computational Harmonic Analysis*, 2:400–401, 1995.
- [33] S. Jaffard and Y. Meyer. Wavelet methods for pointwise regularity and local oscillations of functions. *Memoirs of the A.M.S.*, 123, 1996.
- [34] A. Kielbasiński and H. Schwetlick. *Numeryczna algebra liniowa*. Wydawnictwa Naukowo Techniczne, Warszawa, 1992.
- [35] D. E. Knuth. *The Art of Computer Programming. Volume 1. Fundamental algorithms*. Addison-Wesley Pub Co, 1998.
- [36] A. Kovacz and B. W. Silverman. Extending the scope of wavelet regression methods by coefficient-dependent thresholding. *Journal of the American Statistical Association*, 95:172–183, 2000.
- [37] A. Krzyżak. Identification of discrete Hammerstein systems by the Fourier series regression estimate. *International Journal of Systems Science*, 20:1729–1744, 1989.
- [38] J. Li. Image compression: The mathematics of JPEG 2000. *Modern Signal Processing*, 46:185–221, 2003.
- [39] S. G. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998.
- [40] Y. Meyer. *Wavelets and operators*. Cambridge University Press, Cambridge, 1992.
- [41] R. Ogden. *Essential Wavelets For Statistical Applications and Data Analysis*. Birkhäuser, Boston, 1997.
- [42] E. Rafajłowicz. Nonparametric orthogonal series estimators of regression: A class attaining the optimal convergence rate in l_2 . *Statistic & Probability Letters*, 5:219–224, 1987.
- [43] M. Pawlak. On the series expansion approach to the identification of Hammerstein systems. *IEEE Transactions on Automatic Control*, 36:763–767, 1991.
- [44] M. Pawlak and Z. Hasiewicz. Nonlinear system identification by the Haar multiresolution analysis. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 45:945–961, 1998.

-
- [45] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1993.
- [46] A. Ralston. *Wstęp do analizy numerycznej*. Państwowe Wydawnictwo Naukowe, Warszawa, 1971.
- [47] M. B. Ruskai, G. Beylin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer, and L. Rafael. *Wavelets and Their Applications*. Jones and Barlett, Boston, 1992.
- [48] L. Rutkowski. On system identification by nonparametric function fitting. *IEEE Transactions on Automatic Control*, 27:225–227, 1982.
- [49] P. Śliwiński and Z. Hasiewicz. Computational algorithms for multiscale identification of nonlinearities in Hammerstein systems with random inputs. Przyjęte do publikacji w: *IEEE Transactions on Signal Processing*, 2005.
- [50] G. Strang. Wavelets and dilation equations: A brief introduction. *SIAM Review*, 31:614–627, 1989.
- [51] G. Strang. Wavelet transforms vs. Fourier transforms. *Bulletin of American Mathematical Society*, 28:288–305, 1993.
- [52] W. Sweldens. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In A. F. Laine and M. Unser, editors, *Wavelet Applications in Signal and Image Processing III*, pages 68–79. Proc. SPIE 2569, 1995.
- [53] D. Taubman and M. Marcellin. *JPEG2000. Image Compression Fundamentals, Standards and Practice*, volume 642 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, 2002.
- [54] M. Unser. Splines. a perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, pages 22–38, 1999.
- [55] M. Unser and A. Aldroubi. A review of wavelets in biomedical applications. *Proceedings of the IEEE*, 84:626–638, 1996.