MAC0317/MAC5920

Introdução ao Processamento de Sinais Digitais

Seção 6.5: Bancos de Filtros para Sinais Finitos

Seção 6.5.1: Estratégias de extensão

Para estender a ideia de bancos de filtros para sinais finitos, adotaremos a estratégia de estender o sinal finito para o contexto bi-infinito.

Seja $x \in C^N$. Queremos representá-lo através de um sinal $\tilde{x} \in L^2(Z)$. Algumas opções são:

1. *extensão com zeros*

$$\tilde{x}_k = \begin{cases} x_k, & k \in \{0, 1, \ldots, N-1\} \\ 0, & \text{c.c.} \end{cases}$$

1. *extensão periódica*

$$\tilde{x}_k = x_{k \bmod N}, \quad \forall k \in Z$$

1. *espelhamento + extensão periódica*

A extensão periódica do tipo 2 corresponde ao sinal

$$\ldots, x_{N-2}, x_{N-1}, x_0, x_1, \ldots, x_{N-2}, x_{N-1}, x_0, x_1, \cdots$$

ao passo que a extensão do tipo 3 corresponde a

$$\ldots, x_3, x_2, x_1, x_0, x_0, x_1, \ldots, x_{N-2}, x_{N-1}, x_{N-1}, x_{N-2}, \cdots$$

e é também chamada de *extensão simétrica de meio-ponto* pois o espelhamento ocorre entre duas amostras. Existe também a *extensão simétrica de ponto inteiro*:

$$\ldots, x_3, x_2, x_1, x_0, x_1, \ldots, x_{N-2}, x_{N-1}, x_{N-2}, \cdots$$

Observe que nenhuma dessas extensões geram sinais de energia finita (em $L^2(Z)$). Será necessário fazer uma espécie de *gambiarra* para permitir o uso daquela teoria.

Extensões pseudo-periódicas

As extensões 2 e 3 acima podem ser feitas um número finito de vezes, para garantir que $\tilde{x} \in L^2(Z)$. A título de exemplo, desenvolveremos a DWT em C^N a partir da extensão periódica (caso 2 acima). Trabalharemos com extensões que chamaremos de *pseudo-periódicas*: para q "grande o suficiente" definiremos:

$$\tilde{x}_k = \begin{cases} x_{k \bmod N}, & -qN \leq k \leq \ + qN \\ 0, & \text{c.c.} \end{cases}$$

Análise da extensão periódica

Por sua maior simplicidade, consideraremos a extensão periódica sem reflexões simétricas na análise do banco de filtros.

Teorema 6.5.1: Seja $x \in C^N$, e considere um banco de filtros $\{l_a, h_a, l_s, h_s\}$ onde todos os filtros tem N ou menos coeficientes. Se interpretarmos esses filtros como elementos de C^N tomando os índices módulo N, e se tomarmos todas as convoluções como convoluções circulares em C^N, então a transformação

$$x \mapsto X = (X_l, X_h) \qquad \left(C^N \mapsto C^N\right)$$

é inversível, e sua inversa é dada por

$$X \mapsto x = l_s * U(X_l) + h_s U(X_h) \qquad \left(C^N \mapsto C^N\right).$$

Prova: Mostraremos que a utilização da transformada baseada em um banco de filtros com reconstrução perfeita no sinal $\tilde{x} \in L^2(Z)$ equivale à mesma sequência de operações em C^N a partir de $x \in C^N$, interpretando as convoluções como convoluções circulares. Pela propriedade de reconstrução perfeita em $L^2(Z)$, temos que os sinais

$$\tilde{X}_l = D(\tilde{x} * l_a),$$

$$\tilde{X}_h = D(\tilde{x} * h_a),$$

onde $*$ é a convolução (linear) em $L^2(Z)$, permitem a reconstrução de $\tilde{x}$ através da expressão

$$\tilde{x} = U(\tilde{X}_l) * l_s + U(\tilde{X}_h) * h_s.$$

Dado $x \in C^N$, podemos definir

$$X_l = D(x * l_a) \in C^{N/2}$$

onde $*$ é a convolução circular

$$(x * l_a)_n = \sum_{k=0}^{N-1} (l_a)_k x_{n-k},$$

e $D \in R^{\frac{N}{2} \times N}$ é a matriz tal que $D_{i,2i} = 1$ e $D_{i,j} = 0$ caso contrário:

Na etapa de ressíntese, vamos combinar X_l e X_h através da expressão

$$U(X_l) * l_s + U(X_h) * h_s = \hat{x}$$

e $U = D^T \in R^{N \times \frac{N}{2}}$.

Queremos mostrar que $\hat{x}$ (versão ressintetizada em domínio finito) é idêntico a x.

Para fazer isso, vamos observar as correspondências entre X_l e $\tilde{X}_l$ e entre X_h e $\tilde{X}_h$. Note inicialmente que as convoluções linear (em Z) e circular (em C^N) coincidem nas amostras "centrais" $n = 0, 1, \ldots, N - 1$:

$$
\begin{aligned}
(\tilde{x} * l_a)_n &= \sum_{-\infty}^{\infty} (l_a)_k \tilde{x}_{n-k} && \text{(convolução linear)} \\
&= \sum_{k=0}^{N-1} (l_a)_k \tilde{x}_{n-k} \\
&= \sum_{k=0}^{N-1} (l_a)_k x_{(n-k \,\%\, N)} \\
&= (x * l_a)_n && \text{(convolução circular)}
\end{aligned}
$$

pois $(l_a)_k = 0$ se $k \notin \{0, 1, \ldots, N - 1\}$ e $\tilde{x}_{n-k} = x_{n-k}$ para $n, k \in \{0, 1, \ldots, N - 1\}$.

Um argumento semelhante mostra que $(x * h_a)_n = (\tilde{x} * h_a)_n$ para $n = 0, 1, \ldots, N - 1$.

A subamostragem D preservará apenas as componentes de índice par, tanto no contexto infinito quanto no finito. Assim, a partir do conhecimento de $X_l, X_h \in C^{N/2}$, podemos recuperar a informação completa de $\tilde{X}_l, \tilde{X}_h \in L^2(Z)$, sendo que

$$(\tilde{X}_l)_{n+kN/2} = (X_l)_n,$$

$$(\tilde{X}_h)_{n+kN/2} = (X_h)_n, \qquad \text{para } n = 0, 1, \ldots, \frac{N}{2} - 1, \ \forall k \in Z.$$

Notando que as convoluções com os filtros l_s, h_s de ressíntese possuem a mesma propriedade em relação às convoluções, podemos ver que, para $n = 0, 1, \ldots, N - 1$, o sinal $\hat{x} \in C^N$ reconstruído pela aplicação do banco de filtros em C^N coincide com o sinal $\tilde{x} \in L^2(Z)$ perfeitamente reconstruído em $L^2(Z)$ e portanto com o sinal finito original $x \in C^N$:

$$
\begin{aligned}
\hat{x}_n &= [U(X_l) * l_s + U(X_h) * h_s]_n && \text{(convolução circular em } C^N\text{)} \\
&= [U(X_l) * l_s]_n + [U(X_h) * h_s]_n \\
&= [U(\tilde{X}_l) * l_s]_n + [U(\tilde{X}_h) * h_s]_n && \text{(convolução linear em } L^2(Z)\text{)} \\
&= [U(\tilde{X}_l) * l_s + U(\tilde{X}_h) * h_s]_n \\
&= \tilde{x}_n && \text{(reconstrução perfeita em } L^2(Z)\text{)} \\
&= x_n
\end{aligned}
$$

Exemplo 6.7: considere a DWT com o banco de filtros de Haar ortogonalizado, que satisfaz $(l_s)_k = (l_a)_{-k}, \forall k$ e $(h_s)_k = (h_a)_{-k}, \forall k$, aplicada a um vetor $x \in C^4$:

Assim

$$X = W_4^a \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

onde

Exemplo 6.8

Na ressíntese, temos $x = U(X_l) * l_s + U(X_h) * h_s$:

ou seja,

$$x = W_4^s \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

onde

Observe que:

- $W_4^s = (W_4^a)^T$
- *todas as colunas (e linhas) de W_4^s e W_4^a são ortogonais (2 a 2)*

Logo

$$W_4^s W_4^a = I \qquad \text{(pois a IDWT é a inversa da DWT)}$$

$$W_4^a (W_4^a)^T = I \qquad \text{(ortonormalidade do banco de filtros)}$$

Seção 6.5.3: Formulação Matricial da DWT em $\mathbb{C}^N$

DWT (Análise): dado $x \in \mathbb{C}^N$,

IDWT (Síntese): dado $X \in C^N$,

$$\begin{aligned} x &= U(X_l) * l_s + U(X_h) * h_s \\ &= M_{l_s} U X_l + M_{h_s} U X_h \\ &= \begin{bmatrix} M_{l_s} U & M_{h_s} U \end{bmatrix} \begin{bmatrix} X_l \\ X_h \end{bmatrix} \\ &= W_N^s X \end{aligned}$$

Se um banco de filtros possui a propriedade

$$(l_s)_k = (l_a)_{-k}, \forall k$$
$$(h_s)_k = (h_a)_{-k}, \forall k$$

sendo denominados *filtros adjuntos* (ou revertidos no tempo), temos (exercício 4.9)

$$M_{l_s} = (M_{l_a})^T$$
$$M_{h_s} = (M_{h_a})^T.$$

Neste caso,

$$
\begin{aligned}
W^s_N &= \begin{bmatrix} M_{l_s}U & M_{h_s}U \end{bmatrix} \\
&= \begin{bmatrix} (M_{l_a})^T U^T & (M_{h_a})^T U^T \end{bmatrix} \\
&= \begin{bmatrix} (M_{l_a}U)^T & (M_{h_a}U)^T \end{bmatrix} \\
&= \begin{bmatrix} M_{l_a}U \\ M_{h_a}U \end{bmatrix}^T \\
&= (W^a_N)^T
\end{aligned}
$$

Seção 6.5.4: Formulação Matricial da DWT de Vários Níveis

DWT de 2 níveis: lembrando do diagrama

$$x \mapsto \begin{bmatrix} x_l \\ x_h \end{bmatrix} \mapsto \begin{bmatrix} \begin{bmatrix} x_{ll} \\ x_{lh} \end{bmatrix} \\ x_h \end{bmatrix}$$

podemos escrever

DWT de 3 níveis

$$x \mapsto \begin{bmatrix} x_l \\ x_h \end{bmatrix} \mapsto \begin{bmatrix} x_{ll} \\ x_{lh} \\ x_h \end{bmatrix} \mapsto \begin{bmatrix} x_{lll} \\ x_{llh} \\ x_{lh} \\ x_h \end{bmatrix}$$

Custo de implementação da DWT de vários níveis

É fácil ver que o custo de implementação da DWT, independentemente do número de níveis, é sempre **linear**, se considerarmos que os filtros têm uma quantidade pequena $O(1)$ de coeficientes.

Isso ocorre pois cada estágio terá custo linear (limitado por αL) na quantidade L de coeficientes processados, e essa quantidade será $L = N$ no 1° estágio, $L = N/2$ no 2° estágio, e assim por diante, logo o custo total será limitado por

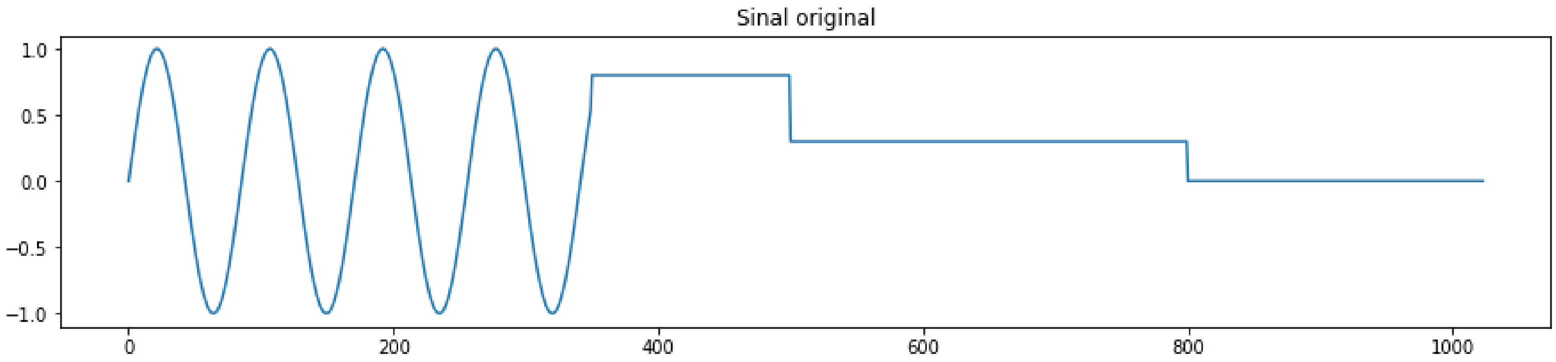
$$\alpha\left(N + \frac{N}{2} + \frac{N}{4} + \cdots\right) \leq \alpha 2N.$$

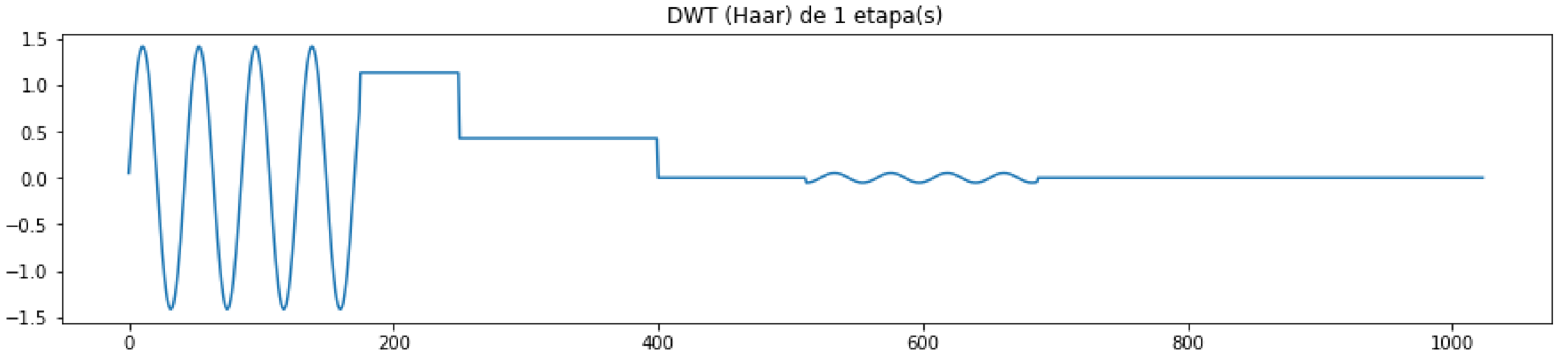
Exemplo 6.9

Voltamos ao sinal do exemplo 6.2, para inspecionar as transformadas de vários estágios:

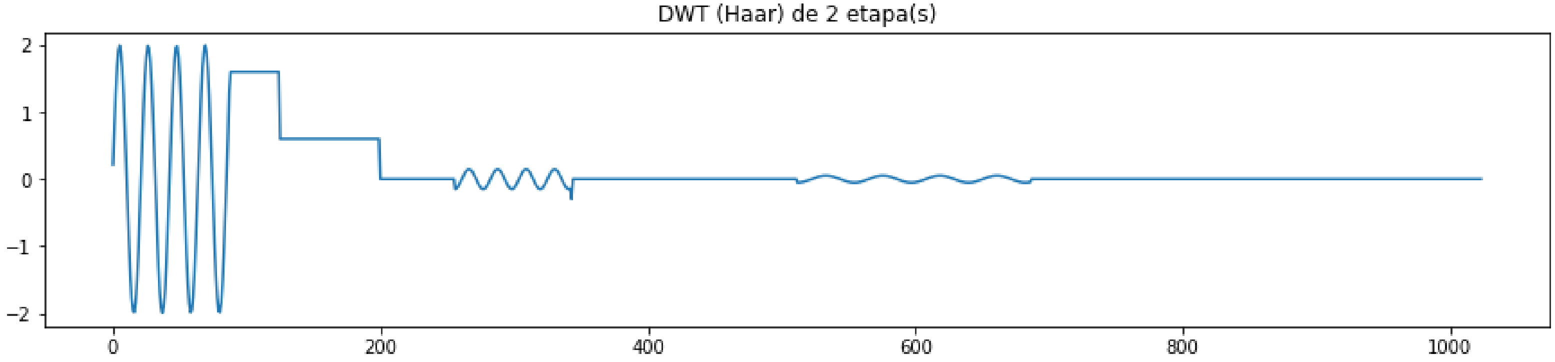
$$
\begin{array}{lllllll}
X^{(1)} = & (& X_{l}, & & & X_{h} &) \\
X^{(2)} = & (& X_{ll}, & & X_{lh}, & X_{h} &) \\
X^{(3)} = & (& X_{lll}, & X_{llh}, & X_{lh}, & X_{h} &) \\
X^{(4)} = & (& X_{llll}, \quad X_{lllh}, & X_{llh}, & X_{lh}, & X_{h} &)
\end{array}
$$

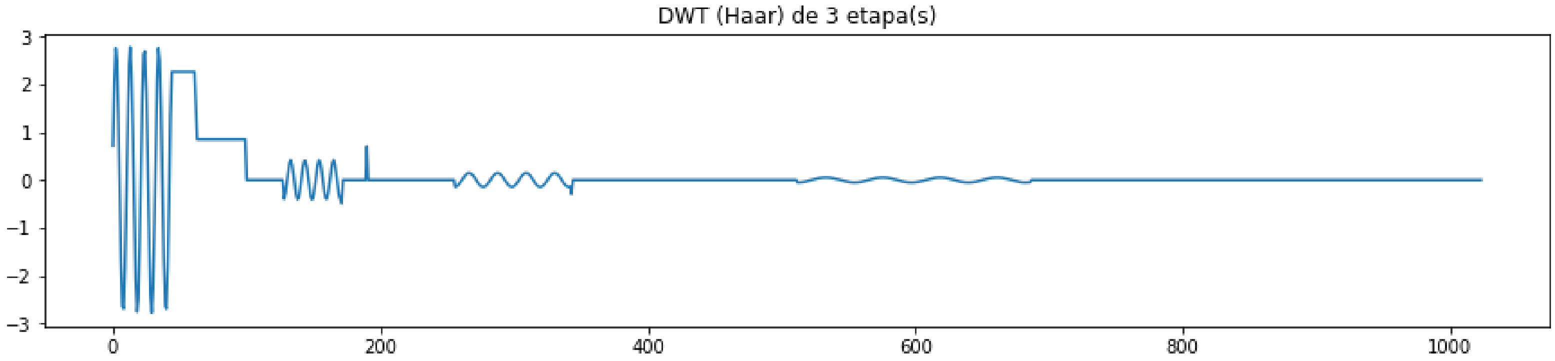
In [3]:
```
plotdwt(0)
plotdwt(1)
```





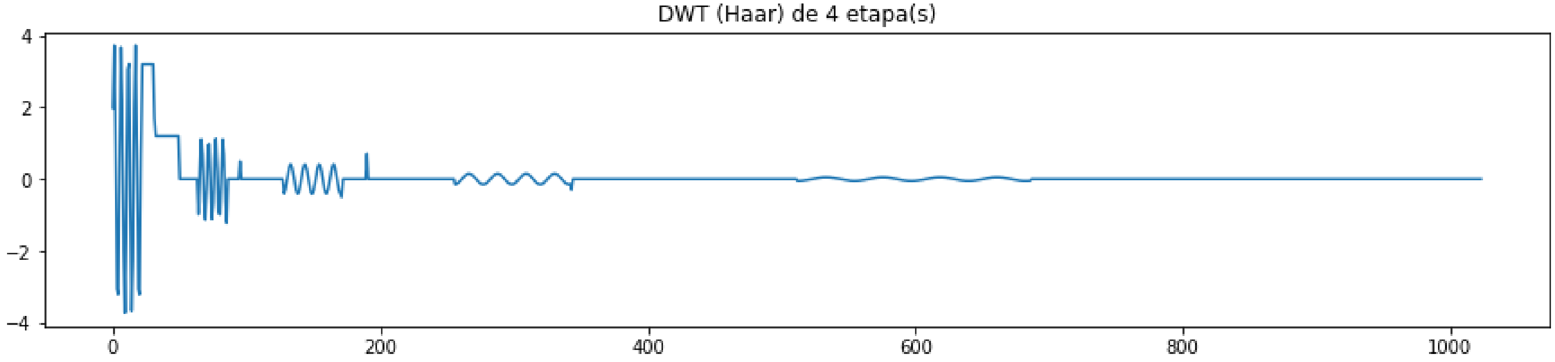
In [4]:
```
plotdwt(2)
plotdwt(3)
```

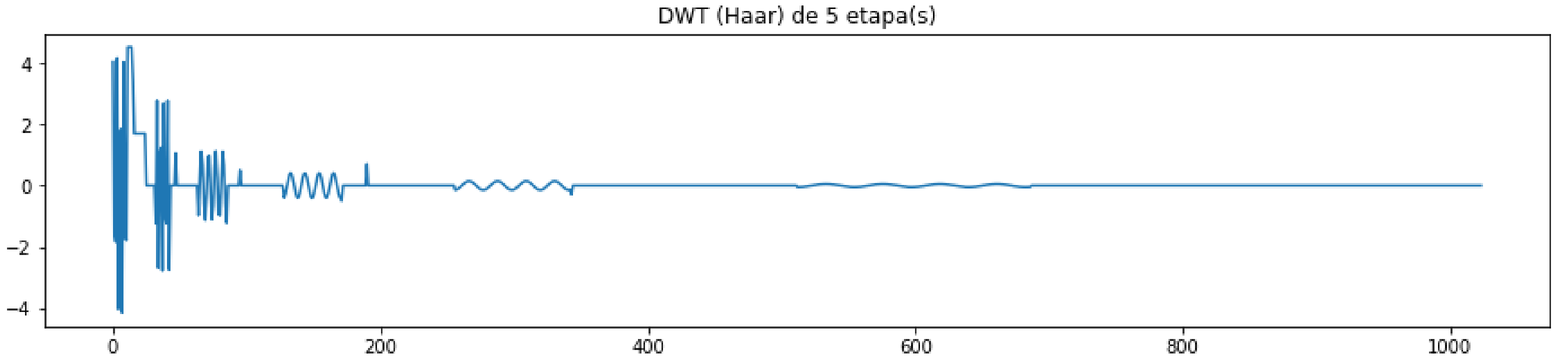




In [5]:

```
plotdwt(4)
plotdwt(5)
```





Reconstrução da DWT por camadas

No primeiro nível da DWT, temos

$$
\begin{aligned}
x &= W_N^s \begin{pmatrix} X_l \\ X_h \end{pmatrix} \\
&= W_N^s \begin{pmatrix} X_l \\ 0 \end{pmatrix} + W_N^s \begin{pmatrix} 0 \\ X_h \end{pmatrix} \\
&= \alpha_1(x) + \delta_1(x)
\end{aligned}
$$

onde α_1 é a reconstrução usando *apenas* os coeficientes de aproximação (como se X_h fosse $= 0$), e δ_1 a reconstrução usando *apenas* os coeficientes de detalhes (como se X_l fosse $= 0$). Chamamos as reconstruções α_1 e δ_1 de aproximação e detalhes de 1ª

Considere agora a DWT de segunda ordem. Podemos escrever

$$
\begin{aligned}
x &= W_N^{s,(2)}\left(\begin{pmatrix} x_{ll} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ x_{lh} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ x_h \end{pmatrix}\right) \\
&= W_N^{s,(2)}\begin{pmatrix} x_{ll} \\ 0 \\ 0 \end{pmatrix} + W_N^{s,(2)}\begin{pmatrix} 0 \\ x_{lh} \\ 0 \end{pmatrix} + W_N^{s}\begin{pmatrix} 0 \\ 0 \\ x_h \end{pmatrix} \\
&= \alpha_2(x) + \delta_2(x) + \delta_1(x)
\end{aligned}
$$

No caso da DWT de terceira ordem, temos

$$
\begin{aligned}
x &= W_N^{s,(3)} \begin{pmatrix} X_{lll} \\ 0 \\ 0 \\ 0 \end{pmatrix} + W_N^{s,(3)} \begin{pmatrix} 0 \\ X_{llh} \\ 0 \\ 0 \end{pmatrix} + W_N^{s,(2)} \begin{pmatrix} 0 \\ 0 \\ X_{lh} \\ 0 \end{pmatrix} + W_N^{s} \begin{pmatrix} 0 \\ 0 \\ 0 \\ X_h \end{pmatrix} \\
&= \alpha_3(x) + \delta_3(x) + \delta_2(x) + \delta_1(x)
\end{aligned}
$$

e em geral

$$\alpha_r(x) = \alpha_{r+1}(x) + \delta_{r+1}(x)$$

ou

$$\alpha_{r+1} = \alpha_r(x) - \delta_{r+1}(x)$$

Figuras 6.14 e 6.15

Neste exemplo veremos mais uma vez as reconstruções do sinal do exemplo 6.2, usando apenas os coeficientes de aproximação de ordem M.

In [7]:

```
interactive(plot_reconstruction, family=r,)
```