

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

JONATAN ELIAS GOEBEL

PROJETO DE DIPLOMAÇÃO
TRANSFORMADA WAVELET PACKET

Porto Alegre
2009

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

TRANSFORMADA WAVELET PACKET.
APLICADA NA PREDIÇÃO DE FALHAS EM VÁLVULAS.

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Marcelo Soares Lubaszewski

CO-ORIENTADOR: Luiz Fernando Gonçalves

Porto Alegre

2009

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

JONATAN ELAIS GOEBEL

**TRANSFORMADA WAVELET PACKET
APLICADA NA PREDIÇÃO DE FALHAS EM VÁLVULAS.**

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de “Projeto de Diplomação”, do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Marcelo Soares Lubaszewski, UFRGS

Formação Institut National Polytechnique de Grenoble.

Banca Examinadora:

Prof. Dr. Marcelo Soares Lubaszewski, UFRGS

Doutor pela (Instituição onde obteve o título – Cidade, País)

Prof. Dr. Arturo Suman Bretas, UFRGS

Doutor pela Virginia Polytechnic Institute and State University (VT), Blacksburg, Estados Unidos

Prof. Dr. Alexandre Sanfelice Bazanella, UFRGS

Doutor pela Universidade Federal de Santa Catarina (Doutorado Sanduíche - UCSB, Santa Barbara, Califórnia)

Porto Alegre, Dezembro de 2009.

DEDICATÓRIA

Dedico este trabalho aos meus pais, Celi e Gilmar Goebel, em especial pela dedicação e apoio em todos os momentos difíceis. Pelo suporte e total apoio as minhas decisões e objetivos traçados.

Também as minhas irmãs Débora e Taciane Goebel que se fizeram presente durante toda a minha formação, as vezes em discussões mas muitas outras com total apoio e colaboração.

Aos meus avós que estiveram sempre preocupados com seu neto, e não perdiam a oportunidade de me mimar quando nos encontrávamos.

Em especial a memória de minha avó Elisa Capra Goebel e bisavó Ida Huff.

AGRADECIMENTOS

Aos pais que deram todo o suporte mesmo a distância, estando sempre do meu lado me apoiando.

As minhas irmãs, que mesmo morando no mesmo apartamento tiveram entenderam a minha ausência nos momentos mais difíceis.

Aos meus avós e bisavós, principalmente aqueles que não pude me despedir devido aos compromisso com a universidade.

Aos colegas da faculdade pelo seu auxílio nas tarefas desenvolvidas durante o curso e apoio na revisão deste trabalho.

Aos colegas do trabalho que sempre me motivaram e me apoiaram quando não podia dar prioridade para o mesmo.

À Universidade, professores, funcionários que propiciaram um excelente ambiente de aprendizagem.

A AIESEC que completou minha formação com conhecimento e experiência em gestão de equipes.

Também a todos os meus amigos de ensino médio que estiveram presentes principalmente no início do curso.

RESUMO

A manutenção representa um enorme custo nas indústrias de um modo geral, e manutenções não programadas ou desnecessárias representam um prejuízo significativo. Os métodos de manutenção estão evoluindo e disponibilizando cada vez mais informações ao usuário para que ele possa ter o controle de seus equipamentos. Métodos proativos estão sendo desenvolvidos para atuar sobre os equipamentos de modo a evitar seu desgaste e disponibilizar informações sobre o estado do equipamento para que as manutenções sejam agendadas e realizadas somente quando realmente for necessário. O sistema de manutenção proativa que está sendo desenvolvido na UFRGS se baseia em analisar as informações do equipamento através da transformada *Wavelet Packet* e alimentar uma rede neural que determinara o estado do equipamento.

Este trabalho implementa o cálculo da energia do sinal em bandas de frequência através da transformada *wavelet packet* que é o sinal de entrada na rede neural do sistema. Ao final do período do projeto foi implementado o cálculo de um coeficiente da árvore *wavelet packet*, restando apenas a implementação da máquina de controle deste bloco para efetuar todo o cálculo e os resultados obtidos através da simulação deste bloco conferem com a resposta do algoritmo implementado no Matlab.

Palavras-chaves: Engenharia Elétrica. Manutenção. Transformada Wavelet Packet. VHDL.

ABSTRACT

Maintenance is a huge cost in the industry in general, and unscheduled maintenance or unnecessary represent a significant loss. The maintenance methods are evolving and increasingly providing information to the User so that he can take control of their equipment. Proactive methods are being developed to work on equipment in order to prevent wear and provide information about the state of the equipment so that maintenance is scheduled and performed only when absolutely necessary. The system of proactive maintenance that is being developed at UFRGS is based on analysis of information from the device through Wavelet Packet Transform and feed a neural network that determined the status of the equipment.

This work implements the calculation of the signal energy into frequency bands by wavelet packet transform that is the input signal in the neural network system. At the end of the project was implemented the calculation of a coefficient of wavelet packet tree, leaving only the implementation of the control machine of this block to make all the calculation and the results obtained by simulating this block grant to the response of the algorithm implemented in Matlab.

Keywords: Electrical Engineering. Maintenance. Wavelet Packet Transform. VHDL.

SUMÁRIO

DEDICATÓRIA	4
RESUMO.....	6
ABSTRACT	7
SUMÁRIO.....	8
LISTA DE ILUSTRAÇÕES	10
LISTA DE TABELAS.....	11
LISTA DE ABREVIATURAS.....	12
1. INTRODUÇÃO.....	13
1.1. Métodos De Manutenção	14
1.2. Predição e Automação.....	15
2. TRANSFORMADAS.....	16
2.1. Transformada de Fourier	16
2.2. Transformada de Gabor.....	17
3. TRANSFORMADA WAVELET.....	19
3.1. Transformada de Haar	19
3.2. Transforma Wavelet Contínua.....	20
3.3. Transformada Wavelet Discreta.....	23
3.4. Análise de Mallat.....	24
3.5. Transformada Wavelet Packet.....	27
3.6. Aplicações Das Wavelets.....	29
4. DESENVOLVIMENTO E IMPLEMENTAÇÃO.....	32
4.1. Algoritmo	35
4.2. Especificação e Hardware Utilizado.....	36
4.3. Arquitetura.....	36
4.4. Filtros Passa Alta e Passa Baixa.....	38

4.5. Cálculo do Coeficiente	40
4.6. Cálculo dos Coeficientes de Um Nível	42
4.7. Árvore Wavelet Packet	42
4.8. Energia Do Sinal.....	43
4.9. Mapa de Memória.....	45
5. RESULTADOS	46
6. CONCLUSÕES E TRABALHOS FUTUROS.....	47
7. REFERÊNCIAS	48
APÊNDICE A.	50

LISTA DE ILUSTRAÇÕES

FIGURA 1 - ANÁLISE DE UM SINAL ATRAVÉS DA STFT.....	18
FIGURA 2 - DILATAÇÃO E TRANSLAÇÃO DA <i>WAVELET</i> MÃE NA TRANSFORMADA DE HAAR.	20
FIGURA 3 - DIFERENÇA ENTRE AS FUNÇÕES DA STFT E DA CWT.....	22
FIGURA 4 - ANÁLISE DE UM SINAL ATRAVÉS DA TRANSFORMADA <i>WAVELET</i>	22
FIGURA 5 - REPRESENTAÇÃO DOS SUBESPAÇOS V_n	24
FIGURA 6 - DECOMPOSIÇÃO DA DWT MULTIRESSOLUÇÃO.....	25
FIGURA 7 - FILTRAGEM E DECIMAÇÃO DO SINAL.	26
FIGURA 8 - DECOMPOSIÇÃO EM SUB-BANDAS <i>WAVELET DISCRETA</i>	26
FIGURA 9 - ÁRVORE <i>WAVELET PACKET</i> ($N_j = N2^j$).....	27
FIGURA 10 - EXEMPLO DA ÁRVORE <i>WAVELET</i> E FUNÇÃO CUSTO.	28
FIGURA 11 - REPRESENTAÇÃO DAS DAUBECHIES DE 4, 12 E 20 TAPS E SEU ESPECTRO EM FREQÜÊNCIA.....	32
FIGURA 12 - MATRIZ <i>WAVELET</i> PARA A TRANSFORMADA DE HAAR.[23].....	33
FIGURA 13 - GENERALIZAÇÃO DA MATRIZ <i>WAVELET</i> . [23].....	34
FIGURA 14 - ARQUITETURA DESENVOLVIDA PARA A <i>WAVELET PACKET</i> EM VHDL.	38
FIGURA 15 - ARQUITETURA DO FILTRO ADAPTATIVO.	39
FIGURA 16 - ARQUITETURA DO BLOCO DE CONTROLE DE UM COEFICIENTE.....	41
FIGURA 17 - REPRESENTAÇÃO DA MEMÓRIA NO CÁLCULO DOS COEFICIENTES.	41
FIGURA 18 - ARRANJO DA MEMÓRIA NO CÁLCULO DE UM NÍVEL DA ÁRVORE <i>WAVELET PACKET</i>	42
FIGURA 19 - ARQUITETURA PARA O CÁLCULO DA ENERGIA.....	45

LISTA DE TABELAS.

TABELA 1 - COEFICIENTES DA DAUBECHIES 6.....	40
TABELA 2 - MAPA DA MEMÓRIA.	45
TABELA 3 - RESULTADO DA SIMULAÇÃO.....	46

LISTA DE ABREVIATURAS.

CWT – Continuous Wavelet Transform.

DWPT – Discrete Wavelet Packet Transform.

DWT – Discrete Wavelet Transform.

EZW - Embedded Zerotrees of Wavelets Transform.

FT – Fourier Transform.

FBI – Federal Bureau of Investigation.

LE – Logical Elements.

MRA – Multiresolution analysis.

QF – Quadrature Filter.

RPA - Recursive Pyramid Algorithm.

SNR – Signal to Noise Ratio.

STFT – Short Term Fourier Transform.

WT – Wavelet Transform.

1. INTRODUÇÃO

Em 2007 a Associação Brasileira de Manutenção, Abraman, através do artigo “Quanto custa e como é a manutenção no Brasil/2007” apontou que o custo com manutenção no país representa 3,89% do PIB, totalizando 56 bilhões de reais anuais, e que aproximadamente 4% do faturamento bruto das empresas é gasto em ações de manutenção, revelando que parte significativa do lucro pode estar sendo consumida neste setor [1].

Um dos maiores problemas da manutenção é que em grande parte dos casos é necessário parar o processo produtivo da indústria, por isso essas manutenções são realizadas em horários cuidadosamente selecionados, e uma manutenção não programada pode representar grandes prejuízos.

Para evitar essas situações muitas empresas realizam manutenções periódicas, para garantir que todos os equipamentos estejam em condições de operação. Porém este tipo de manutenção, chamada de preditiva, gera manutenções desnecessárias, muitas vezes chegando a representar um custo maior do que o risco de uma paralisação emergencial.

Para solucionar este problema está sendo desenvolvido na UFRGS um sistema de manutenção proativa, que analisa as informações de utilização do equipamento, neste caso para manutenção de válvulas, e realiza uma detecção do estado atual do equipamento, assim como a sua probabilidade de falha.

Este sistema está sendo desenvolvido através da técnica de redes neurais artificiais, e a informação obtida do equipamento é analisada através da *Discret Wavelet Packet Transform* (DWPT), que será o ponto de estudo neste trabalho.

1.1. MÉTODOS DE MANUTENÇÃO

Hoje existem basicamente quatro formas de se realizar a manutenção de equipamentos industriais.

A primeira é a manutenção corretiva que se baseia em utilizar o equipamento ao máximo sem realizar qualquer tipo de manutenção, por isto é o método mais barato, porém este tipo de manutenção aumenta a probabilidade de falhas em horários de alta produção, fato que gera uma parada em horários não programados causando grandes prejuízos.

Para evitar estas paradas foi criado o sistema de manutenção preventiva, que se baseia em realizar a manutenção e eventual trocas de peças com uma periodicidade planejada, evitando assim que os equipamentos cheguem ao ponto de falha. Mesmo esta sendo considerada uma forma de manutenção mais econômica por não gerar grandes danos no processo produtivo, ela gera um certo desperdício de recursos pois realiza manutenções e trocas de peças desnecessárias.

A terceira forma de manutenção é a preditiva, ela monitora os equipamentos continuamente de tal forma que seja possível prever uma falha. Porém este processo deve ser acompanhado de uma mão-de-obra especializada, e deve ser implementado também algum sistema de monitoramento, como análises termográficas e de vibração, que acaba elevando o seu custo.

Para atacar estes problemas foi criado o sistema de manutenção proativa, que semelhantemente ao sistema preditivo monitora os equipamentos, mas disponibilizando informações sobre as condições atuais do equipamento, até mesmo seu tempo de vida e qual a probabilidade ocorrer alguma falha no futuro próximo. Dessa forma não é necessária uma interpretação dos dados disponíveis para o usuário. Este método de manutenção também pode automatizar o processo de tal forma que permita o sistema de monitoramento agir automaticamente sobre o equipamento em casos de emergência.

1.2. PREDIÇÃO E AUTOMAÇÃO

Neste projeto será utilizada a técnica de redes neurais artificiais para realizar a predição do estado da máquina. Para que este sistema funcione é necessário calibrar a rede neural com dados reais que são obtidos com ensaios em laboratório de várias condições de estresse ou mesmo em condições normais de operação.

A rede neural artificial será alimentada pela energia do sinal de torque da válvula e representada em bandas de frequência, por isto é importante a utilização de ferramentas matemáticas que tenham como representar este sinal no domínio da frequência.

Como este projeto trata de predição, não basta saber quais as frequências estão atuando no sinal, é necessário ter o conhecimento de em quais momentos elas atuaram, de tal forma que seja possível identificar se foi uma situação de estresse ou se é uma condição de operação que pode representar algum defeito.

Através destas informações a rede neural artificial determinará qual o estado atual do equipamento, e o quão próximo ela está de uma falha de acordo com as leituras realizadas nos ensaios, com isto ela é capaz de atuar diretamente ou passar um relatório com informações mais precisas sobre a condição atual do equipamento para o usuário.

2. TRANSFORMADAS.

Em geral, para realizar a análise de um sistema parte-se de um modelo matemático. Este modelo representará os diversos fenômenos que ocorrem dentro do sistema, o objetivo desta modelagem é permitir o estudo de como o sistema se comporta com diferentes entradas, como ele se comportará caso haja alguma falha, e se sua resposta é realmente a esperada para a aplicação em questão.

Apesar de essas informações serem importantes no domínio do tempo, sua análise neste domínio pode ser extremamente complexa, além de que uma resposta no tempo pode mascarar informações importantes, como resposta ao ruído e outras perturbações que um sistema pode sofrer.

Uma forma de conseguir simplificar o problema, ou simplesmente obter uma informação crítica de um sistema é a utilização de funções “transformadas”. Estas funções geram a representação de um sinal ou sistema em outro domínio de espaço que seja conveniente para a aplicação, usualmente o domínio da frequência.

2.1. TRANSFORMADA DE FOURIER

A Transformada de Fourier (FT) foi desenvolvida por Jean Baptiste Joseph Fourier (1768 - 1830) e é amplamente utilizada em análise de sinais. A FT se baseia em representar um sinal por uma série de senóides de diferentes frequências, podendo assim dizer que é uma técnica que manipula o sinal a partir de uma base de tempo, para uma base de frequência, ela é obtida através da equação 2.1.1, onde $x(t)$ é o sinal a ser analisado, f é a frequência e $X(f)$ é a FT de $x(t)$ [2].

$$X(f) = F[x(t)] = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt \quad (2.1.1)$$

A análise de Fourier só se aplica a um sinal estacionário, já que uma vez realizada a FT, qualquer referência temporal é perdida, pois nesta análise o sinal é decomposto em funções senos e cossenos igualmente estacionários. Dessa forma para realizar a FT em sinais não estacionários é necessário considerá-lo como estacionário e estender sua análise de $-\infty$ até $+\infty$, de tal forma que a identificação do sinal com o tempo é perdida. Porém esta perda de referência temporal pode ser um problema, já que muitas vezes não basta saber qual frequência é dominante, mas também em quais momentos elas atuam, já que a grande maioria dos sinais na natureza são dinâmicos.

2.2. TRANSFORMADA DE GABOR.

Em 1946 Dennis Gabor adaptou a transformada de Fourier para analisar apenas uma parte do sinal no tempo, uma técnica chamada de *windowing* (janelamento) do sinal. A adaptação de Gabor, conhecida como *Short Time Fourier Transform* (STFT), coloca o sinal em uma função de duas dimensões, tempo e frequência, conforme a equação 2.1 [3].

$$STFT(\tau, f) = \int x(t)g(t - \tau)e^{-j2\pi ft} dt \quad (2.1)$$

Esta função pode ser descrita como a Transformada de Fourier do sinal $x(t)$, previamente “janelada” por uma função $g(t)$ ao redor do tempo τ . Este método assume estacionariedade do sinal dentro desta janela limitada $g(t)$. A sua análise está representada na figura 1, onde a figura (a) representa um sinal qualquer, e (b) a sua STFT [3].

Entretanto, a STFT possui um compromisso entre a resolução no tempo e na frequência na análise do sinal. Esta transformada fornece a informação de quando e em qual frequência ocorreu um evento no sinal, porém esta informação tem precisão limitada pelo tamanho da janela, sendo que uma vez escolhido o tamanho da janela, este será o mesmo para todas as frequências. Uma janela muito estreita resulta em uma boa resolução temporal. Mas

um trecho estreito (janela estreita) tem associado no espectro uma função com lóbulo principal largo, o que reduz a resolução entre frequências contíguas [3].

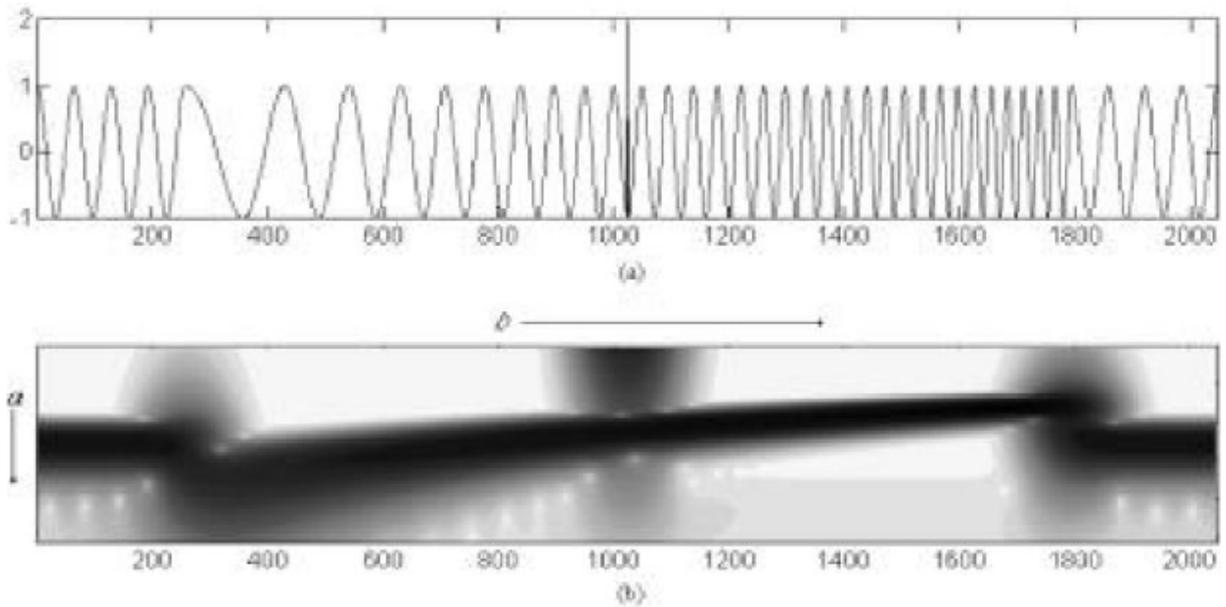


Figura 1 - Análise de um sinal através da STFT.

A resolução no tempo e em frequência não pode ser arbitrariamente pequena, e seu produto fica limitado pela inequação de Heisenberg:

$$\Delta t \cdot \Delta f \geq \frac{1}{4\pi} \quad (2.2)$$

onde Δt e Δf são a resolução no tempo e na frequência respectivamente [3].

Para aprimorar esta técnica, se fazia necessário alguma forma de adaptar o tamanho da janela de acordo com a frequência, permitindo que se tenha uma janela temporal curta em altas frequências e uma maior nas baixas frequências, a solução encontrada foi a análise através da transformada *wavelet*.

3. TRANSFORMADA WAVELET.

A primeira transformada *wavelet* (WT) surgiu antes mesmo da STFT, publicada por Alfréd Haar em 1910, ela é a mais simples das *wavelets*, porém a sua aplicação não era vantajosa, pois era composta por um espaço Hilbertiano de funções não contínuas, dessa forma não era uma boa solução para ser usado em sinais contínuos.

Em 1975 George Zweig descobriu a transformada *wavelet* contínua que posteriormente foi formulada por Jean Morlet em 1982, e ficou conhecida como *Continuous Wavelet Transform* (CWT).

3.1. TRANSFORMADA DE HAAR.

Apesar da transformada *wavelet* não ser uma evolução da transformada de Haar propriamente dita, e dela ter sido descoberta anteriormente a transformada *wavelet*, podemos considerar ela como sendo uma transformada *wavelet* onde a *wavelet* mãe é definida pela equação 3.1, e pode facilmente ser utilizada para a compreensão de uma forma geral de todas as transformadas *wavelets* [4].

$$\Psi(x) = \begin{cases} 1, & [0, \frac{1}{2}) \\ -1, & [\frac{1}{2}, 1) \end{cases} \quad (3.1)$$

A dilatação e a translação da *wavelet* mãe, através da equação 3.2, forma uma base ortogonal no espaço $L^2(R)$, dessa forma ela pode representar qualquer elemento do espaço $L^2(R)$ como uma combinação linear dessas funções bases [4].

$$\Psi_{jk}(x) = \text{const.} \Psi(2^j x - k) \quad (3.2)$$

É possível verificar a ortogonalidade dessas funções através da equação 3.3.

$$\int \Psi_{jk} \Psi_{j'k'} = 0 \quad (3.3)$$

Esta só não será satisfeita quando $j = j'$ e $k = k'$, e neste caso teremos a energia do sinal como resultado da equação.

A constante que torna esta base ortonormal é $2^{j/2}$, e pode ser verificada pela equação 3.4.

$$1 = \int (\text{const.} \Psi(2^j x - k))^2 dx = \text{const}^2 2^{-j} \int (\Psi(x))^2 dx = \text{const}^2 \cdot 2^{-j} \quad (3.4)$$

Assim obtemos a equação geral das funções base da transformada de Haar:

$$\Psi_{j,k}(t) = \sqrt{2^j} \cdot \Psi(2^j x - k) \quad (3.5)$$

As funções $\Psi_{10}, \Psi_{11}, \Psi_{20}, \Psi_{22}, \Psi_{23}$ estão retratadas na figura 2. O conjunto $\{\Psi_{jk}, j \in \mathbb{Z}, k \in \mathbb{Z}\}$ define uma base ortogonal do espaço L^2 .

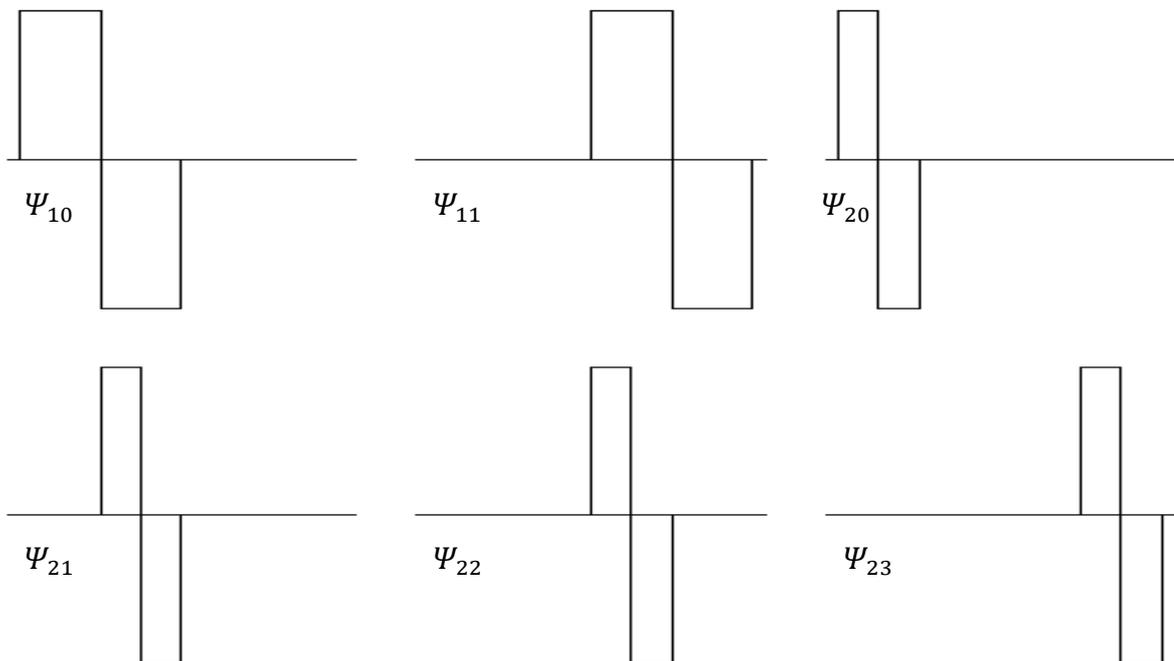


Figura 2 - Dilatação e translação da *wavelet* mãe na transformada de Haar.

3.2. TRANSFORMA WAVELET CONTÍNUA.

A análise de Wavelet foi proposta com o intuito de superar a limitação de janela fixa que existe na STFT, utilizando uma técnica de janelamento variável. A resolução Δt e Δf

variam no plano tempo-freqüência de forma a obter uma análise multiresolução. Essa análise é igual a de um banco de filtros, e a resolução temporal deve aumentar com a freqüência central da análise dos filtros, por isso foi proposto que a resolução em freqüência (Δf) seja proporcional a f de forma que:

$$\frac{\Delta f}{f} = c \quad (3.6)$$

Onde c é o inverso do fator de qualidade do filtro ($Q = \frac{\omega_0}{B}$) [3].

A análise do banco de filtros é composta por filtros passa-banda, com fator de qualidade constante.

A análise por *Wavelet* usa uma janela temporal pequena onde se deseja mais informações de alta freqüência, e janelas temporais maiores onde se deseja informações de baixa freqüência.

Na transformada *Wavelet* é introduzida a notação de escala como uma alternativa a notação de freqüência, desta forma em uma transformada *Wavelet* temos a representação do sinal em um plano tempo-escala, enquanto na STFT tínhamos a representação em um plano tempo-freqüência[3].

A CWT é definida como um espaço de funções ortonormais, denominadas baixas *wavelets*, as quais foram uma base de funções da transformada *wavelet*. Assim a CWT é definida como o produto interno na forma:

$$CWT_{f(a,b)} = \langle f(t), \Psi_{a,b}(t) \rangle = \int f(t) \cdot \Psi_{a,b}(t) \cdot dt \quad (3.7)$$

Onde $\Psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \Psi\left(\frac{t-b}{a}\right)$, a é o fator de escala e b a variável de deslocamento[3].

A CWT mede a similaridade entre o sinal e a função básica $\Psi_{a,b}(t)$. As *wavelets* $\Psi_{a,b}(t)$ são versões escalonadas pelo parâmetro a e transladadas pelo parâmetro b da função básica $\Psi(t)$ ou *wavelet* mãe.

A transformada wavelet contínua inversa pode ser obtida pela seguinte equação:

$$f(t) = \frac{1}{c_h} \iint CWT(a, b) \cdot \Psi_{a,b}(t) \frac{da \cdot db}{a^2} \quad (3.8)$$

Da equação 3.8, c_h é uma constante que depende de $\Psi(t)$. A equação 3.9 é chamada de condição de admissibilidade, que garante a existência da Transformada Inversa e satisfaz a Resolução de Identidade [3]:

$$c_h = \int \frac{|H(w)|^2}{|w|} dw < +\infty \quad (3.9)$$

A figura 3 apresenta a diferença entre as formas de ondas utilizadas na transformada de Gabor (ou STFT) e na WT, e a figura 4 representa a análise do mesmo sinal analisado pela figura 3 através da STFT, agora analisado por uma *wavelet* [5].

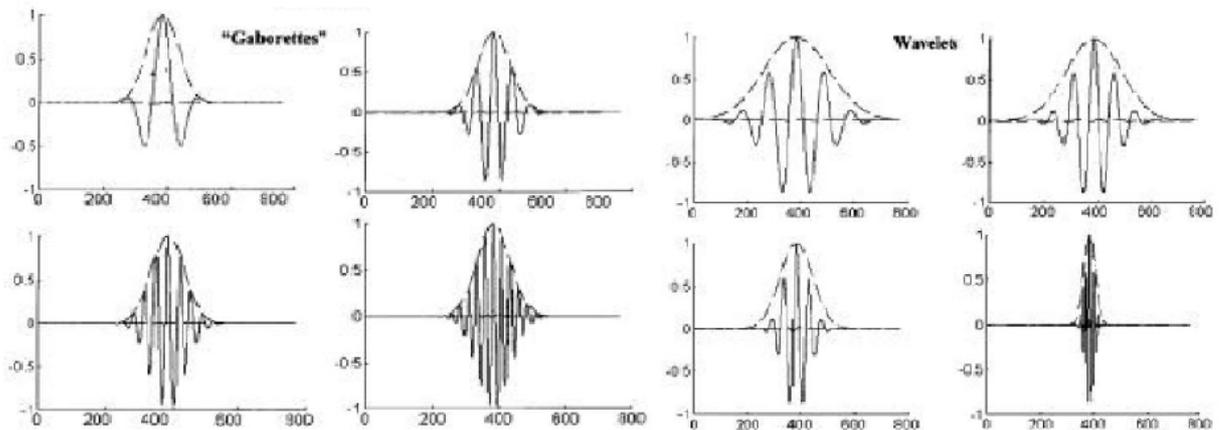


Figura 3 - Diferença entre as funções da STFT e da CWT.

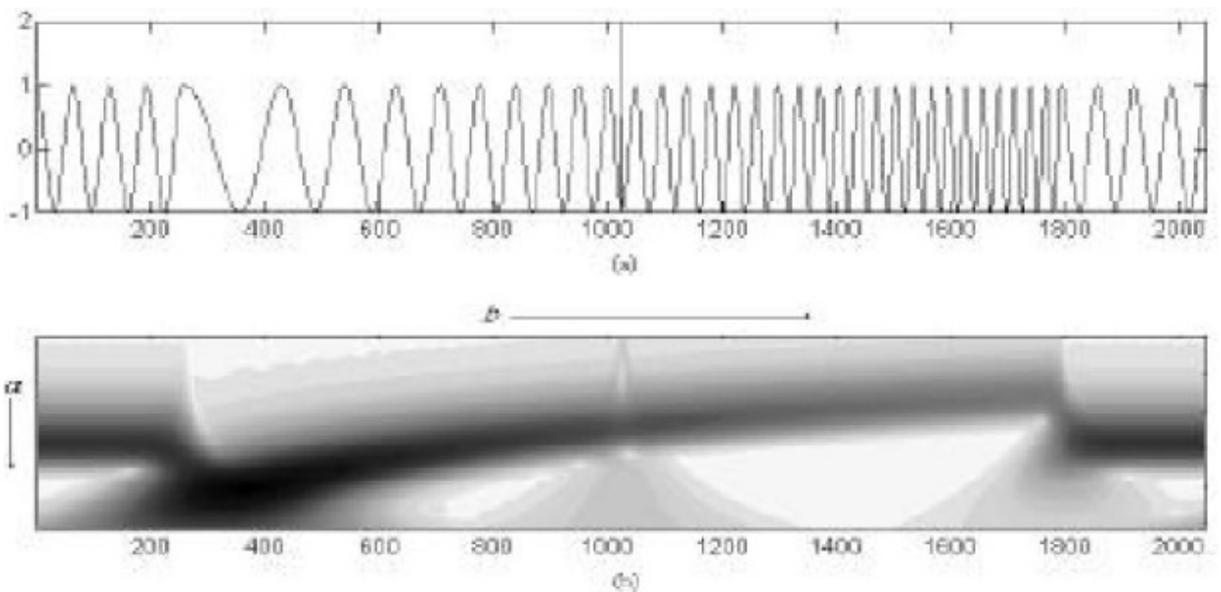


Figura 4 - Análise de um sinal através da transformada *wavelet*.

3.3. TRANSFORMADA WAVELET DISCRETA.

Uma forma de simplificar o cálculo da CWT é usar parâmetros de deslocamento e escala discretos. A esta análise dá-se o nome de *Discret Wavelet Transform* (DWT).

A transformada Wavelet Discreta de uma função $f(t)$ é definida como:

$$DWT(m, p) = d_{m,p} = \int_{-\infty}^{\infty} f(t) \cdot \Psi_{m,p} dt \quad (3.10)$$

onde $\Psi_{m,p}$ forma uma base de funções *wavelets*. Criadas a partir da função *wavelet* mãe $\Psi_{a,b}(t)$, mediante parâmetros de escala e translação discretos (m e p , respectivamente).

Assim $\Psi_{a,b}(t)$ é dado por:

$$\Psi_{m,p} = \frac{1}{\sqrt{a_0^m}} \Psi\left(\frac{t}{a_0^m} - p \cdot b_0\right) = \frac{1}{\sqrt{a_0^m}} \Psi\left(\frac{t - p \cdot a_0^m \cdot b_0}{a_0^m}\right) \quad (3.11)$$

onde a_0 e b_0 são constantes e m e p pertencem ao conjunto dos inteiros [3].

A reconstrução do sinal $f(t)$ através dos coeficientes *wavelets* exige uma escolha muito especial de $\Psi(t)$. Daubechies provou que a condição necessária e suficiente para a reconstrução do sinal dos coeficientes *wavelets* é que a energia de estar entre dois limites positivos:

$$A\|f\|^2 \leq \sum_{m,p} |\langle f, \Psi_{m,p} \rangle|^2 \leq B\|f\|^2 \quad (3.12)$$

Se $A = B$, a energia da WT é proporcional a energia do sinal e as *wavelets* se comportam como bases ortogonais [3].

A reconstrução do sinal é dada pela equação:

$$f(t) = \frac{2}{A+B} \sum_m \sum_p c_{m,p} \Psi_{m,p}(t) \quad (3.13)$$

3.4. ANALISE DE MALLAT.

A análise Multiresolução (MRA) é o método de concepção da maioria das transformadas *wavelets* discretas, e foi introduzido por Stephane Mallat e Yves Meyer em 1988/89.

A análise Multiresolução em um espaço $L^2(\mathbb{R})$ consiste em uma seqüência de subespaços $V_0, V_1, V_2, \dots, V_n$ de tal forma que $V_{n-1} \subset V_n$, a figura 5 faz uma representação gráfica destes espaços [3].

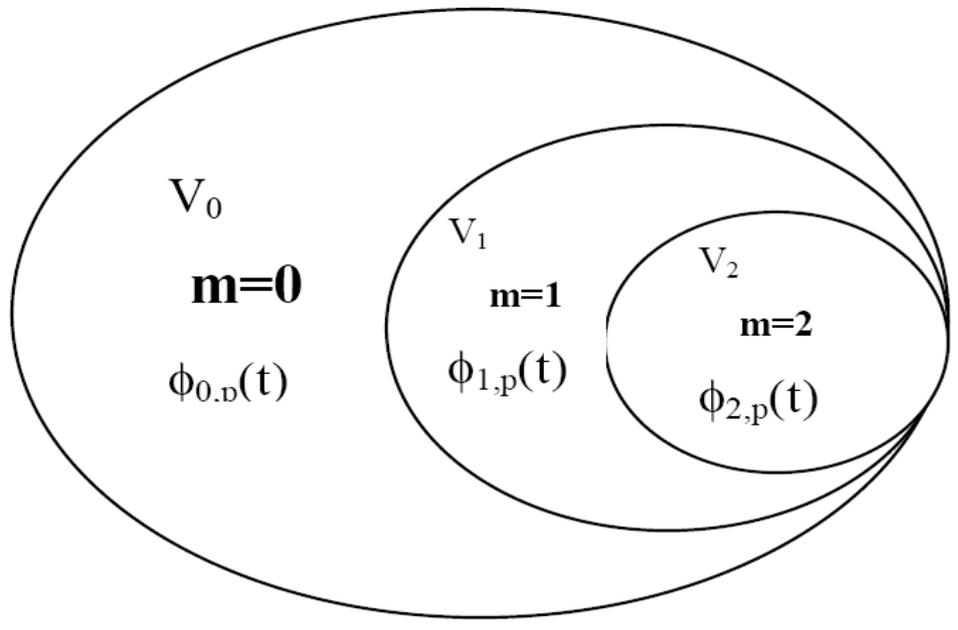


Figura 5 - Representação dos subespaços V_n .

Se o espaço V_j é gerado através da função $\phi_{j,k}(x)$ e o espaço V_{j+1} é gerado através da função $\phi_{j+1,k}(x)$ onde $\phi_{j+1,k}(x) = \sqrt{2}\phi_{j,k}(2x)$ então temos que:

$$f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}, j \in \mathbb{Z} \quad (3.14)$$

Dessa forma podemos dizer que os espaços V_j e V_{j+1} são similares, e como $V_0 \subset V_1$, qualquer função em V_j pode ser escrita como uma combinação linear das funções bases $\sqrt{2}\phi(2x)$ de V_1 . Em particular:

$$\phi(x) = \sum_k h(k)\sqrt{2}\phi(2x - k) \quad (3.15)$$

Onde os coeficientes $h(k)$ são definidos como $\langle \phi(x), \sqrt{2}\phi(2x - k) \rangle$. Considere agora o complemento ortogonal W_j de V_j para V_{j+1} (i.e, $V_{j+1} = V_j \oplus W_j$), da mesma forma que para V_1 , temos que $\{\sqrt{2}\psi(2x - k), k \in \mathbb{Z}\}$ forma uma base ortonormal para o subespaço W_1 .

Para uma dada função $f \in L^2(\mathbb{R})$ é possível encontrar um N tal que $f_N \in V_N$ aproxima f com uma precisão pré-atribuída. Se $g_i \in W_i$ e $f_i \in V_i$, então

$$f_N = f_{N-1} + g_{N-1} = \sum_{i=1}^M g_{N-i} + f_{N-i} \quad (3.16)$$

Assim é definida a função $\phi_{j,k}(x)$ como sendo a função de escalonamento e a função $\psi_{j,k}(x)$ como sendo a função *wavelet*. A primeira é relacionada a um filtro passa-baixas, e os coeficientes da transformada obtidos através dela representam o conteúdo do sinal, ou uma aproximação “borrada”, e a segunda é relacionada a um filtro passa-altas e seus coeficientes representam as bordas, ou detalhamento (figura 6) [3].

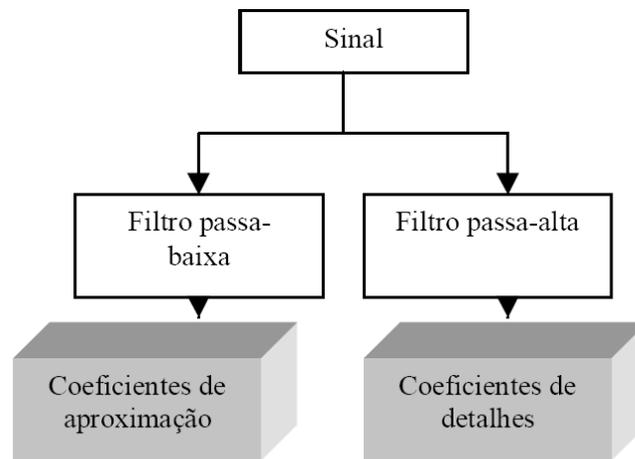


Figura 6 - Decomposição da DWT multiresolução.

Os filtros utilizados para realizar o cálculo da DWT através da análise de Mallat, são chamados de *Quadrature Filters* (QF), e tem como característica filtrar e decimar o sinal. decimação é uma operação que reduz a amostragem do sinal, isto é, ela descarta algumas amostras, decimar por um fator de 2, significa que para cada amostra, uma é descartada, o processo de filtragem com a decimação esta representado na figura 7, onde cD representa os coeficientes de detalhamento e Ca os coeficientes de aproximação [3].

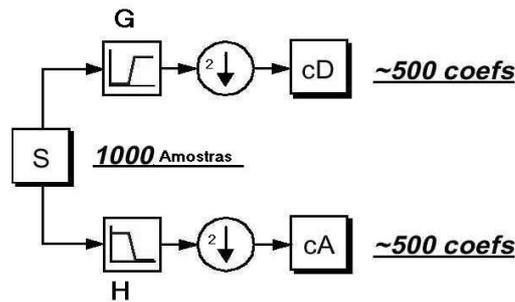


Figura 7 - Filtragem e decimação do sinal.

A relação entre estes filtros e as funções de escalonamento e *wavelet* são representadas pela seguintes equações[4].

$$h(p - 2k) = \langle \phi_{m-1,p}, \phi_{m,k} \rangle = \frac{1}{\sqrt{2}} \int \phi(t - p) \cdot \phi\left(\frac{t}{2} - k\right) dt \quad (3.17)$$

$$g(p - 2k) = \langle \phi_{m-1,p}, \psi_{m,k} \rangle = \frac{1}{\sqrt{2}} \int \phi(t - p) \cdot \psi\left(\frac{t}{2} - k\right) dt \quad (3.18)$$

Desta forma é possível calcular os coeficientes de um nível para outro utilizando filtros digitais (QF), e desenvolver algoritmos rápidos recursivamente. Estas relações estão nas equações 3.17 e 3.18, e estas são constantes em qualquer nível de dilatação, ou seja, não é necessário recalculá-las para cada novo nível de dilatação.

Se realizarmos esta operação de modo iterativo, com sucessivas decomposições, teremos a decomposição do sinal em sub-bandas, dessa forma obtém-se a árvore de decomposição da *wavelet*. Esta decomposição pode ser obtida de maneira eficiente como mostra a figura 8 [3].

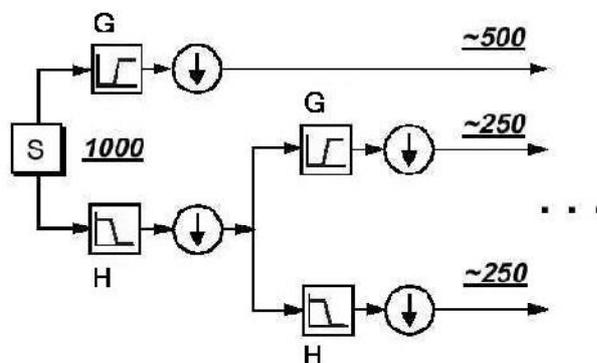


Figura 8 - Decomposição em sub-bandas *Wavelet Discreta*.

3.5. TRANSFORMADA WAVELET PACKET.

Como visto na seção anterior a implementação da DWT pode ser feita de maneira bastante eficiente através da análise multiresolução dividindo o sinal em coeficientes de aproximação e detalhamento como visto na figura 8, porém apenas os coeficientes de aproximação são novamente divididos em coeficientes de detalhamento e aproximação. Já na *Wavelet Packet* os coeficientes de detalhe também são divididos, decompondo o sinal em forma de uma árvore binária, chamada árvore *Wavelet Packet*. Dessa forma obtemos o sinal dividido em bandas de frequência com diferentes resoluções, como representado na figura 9. onde $G(k/N_j)$ e $H(k/N_j)$, representam os QF, e $C_{j,k}$ os coeficientes da matriz wavelet.

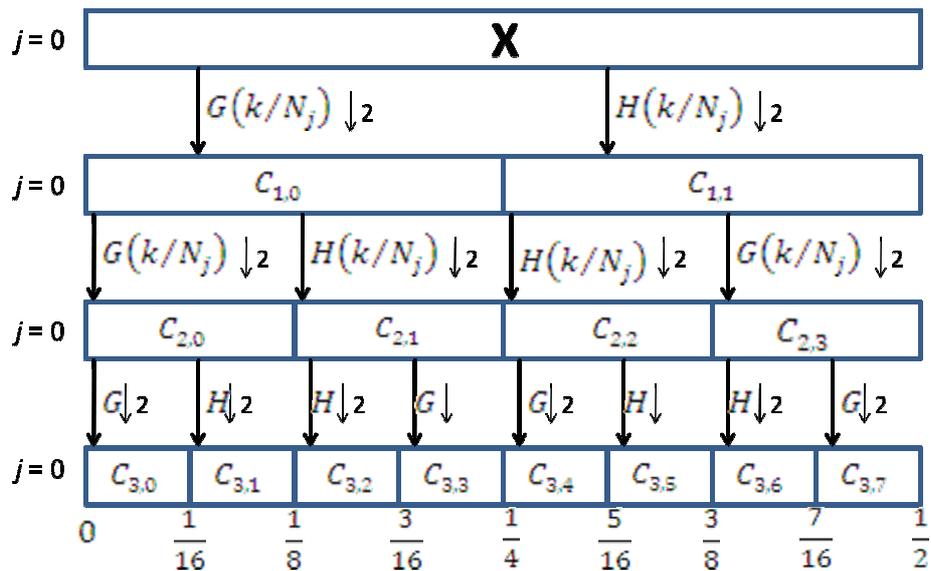


Figura 9 - Árvore wavelet packet ($N_j = N/2^j$).

3.5.1. BASE WAVELET.

A base *wavelet* é o conjunto da matriz wavelet onde as funções *wavelet* e de escalonamento são diferentes de zero. Esta é a base necessária para recuperar o sinal, porém muitas vezes não é necessário recuperar exatamente o sinal, precisamos apenas de uma

aproximação do sinal, neste caso podemos escolher uma base *wavelet* que despreze alguns valores para que se tenha uma matriz *wavelet* reduzida. Para definirmos qual a melhor base para representarmos o sinal escolhemos uma função custo [6].

Um exemplo de função custo é a *threshold*. No caso de sinais discretos esta função conta o número de valores em um nó da árvore *wavelet packet* cujo valor absoluto é maior que um limiar ε . A figura 10 b) representa o custo da matriz *wavelet* representada pela figura 10 a) para um limiar $\varepsilon = 1$ [6].

Para definirmos a melhor base somamos os custo dos ramos de cada nó da árvore, começando pelos nós inferiores, se esta soma for maior ou igual, então esta será a melhor base, caso contrário passamos para o nó acima indo em direção a raiz da árvore, os ramos rasurados na figura 10 b) representam a melhor base para esta função custo [6].

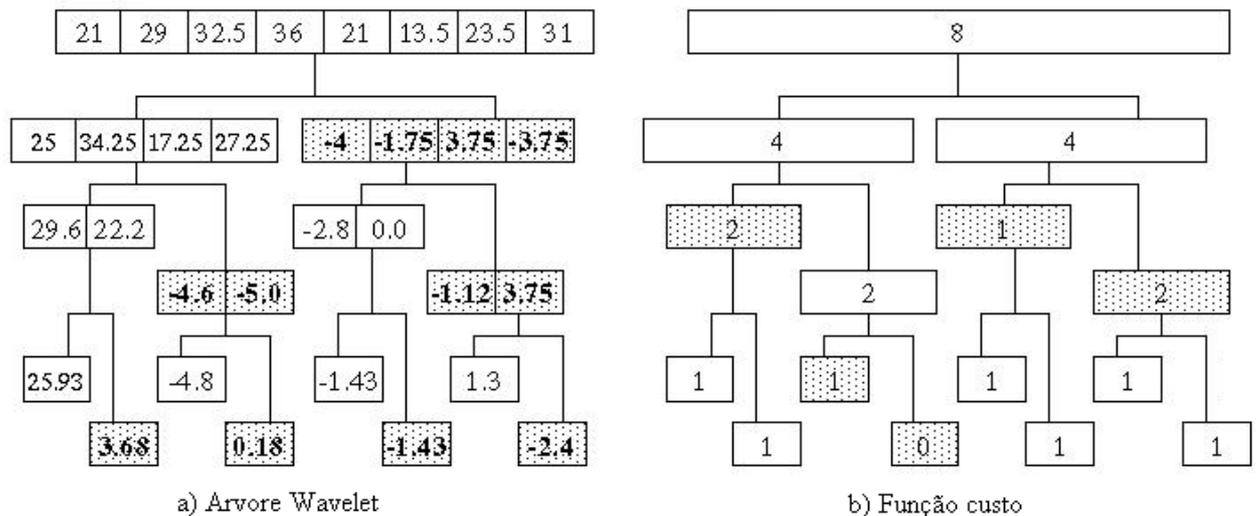


Figura 10 - Exemplo da árvore *wavelet* e função custo.

Existem outras formas de funções custo, como por exemplo, a de Entropia, que se baseia na equação da entropia de Shannon-Weaver, e do logaritmo da Energia.

3.6. APLICAÇÕES DAS WAVELETS.

As wavelets são utilizadas de diversas formas na área de análise de sinais. Porém elas vão muito além da análise espectral de um sinal, como a *Wavelet Mãe* pode ser qualquer função que atenda aos critérios citados no capítulo 3, a WT pode ser ajustada de acordo com sua aplicação.

Dessa forma as *wavelets* estão ganhando espaço em diversas áreas. Truchetet e Lalignant escreveram um artigo comentando as principais aplicações desta ferramenta principalmente em ambiente industrial, que mostra o quanto ela está se difundindo entre todas as áreas da engenharia [7].

3.6.1. COMPRESSÃO DE DADOS.

A transformada *wavelet packet* pode ser utilizada na compressão de dados, som, e imagem, tanto para serem guardados como para serem transmitidos, podendo realizar essas compressões com ou sem perdas de acordo com a aplicação.

Os métodos de compressão através das transformadas *wavelets* são adequados para representação de transientes, tais como sons de percussão em componentes de frequência de áudio, ou componentes de alta frequência em imagens bidimensionais, por exemplo, uma imagem das estrelas em céu noturno. Isto significa que os elementos transitórios de um sinal de dados podem ser representados por uma menor quantidade de informação do que seria o caso com alguma outra transformada [7].

As *wavelets* são utilizadas na compressão de imagens no formato JPEG2000 e através da transformada EZW (*Embedded Zerotrees of Wavelets Transform*) cuja principal característica é uma alta compressão da imagem com mínima perda. Hoje a transformada *wavelet* é utilizada pelo FBI (*Federal Bureau of Investigation*) para a compressão e análise de impressões digitais [8].

3.6.2. ENERGIA DE SINAIS.

A aplicação das WTs esta direcionada para aplicação digital, porém esta é uma ferramenta extremamente poderosa para extrair informação de energia e potência de sinais reais e não estacionários.

Os métodos tradicionais para medir a tensão e corrente em sistemas elétricos, por exemplo, são realizados tanto no domínio do tempo quanto no domínio da frequência através da transformada de Fourier. A aproximação no domínio do tempo é o mais preciso e a aproximação no domínio da frequência permite a análise em bandas de frequências individuais e a determinação de distorções harmônicas, mas ela sofre com a necessidade de se ter um sinal estacionário, e como as formas de ondas são estacionárias por natureza, existe uma perda de informação temporal [9].

As wavelets se provarão um ferramenta de processamento de sinais poderosa na análise de sistemas de potencia transientes. A vantagem dela é que preserva tanto a relação temporal como a de frequência relacionada.

3.6.3. TESTES NÃO DESTRUTIVOS (NON-DESTRUCTIVE TEST)

Testes não destrutivos são realizados para verificar o estado de um componente qualquer sem causar algum dano a este, como exemplo é possível citar o teste em cabos de aço através da fuga do fluxo magnético aplicando uma série de impulsos de diferentes magnitude e duração, dependendo das dimensões do cabo. A transformada wavelet mostra ter uma precisão significativamente maior que os outros métodos devido a relação SNR (Signal Noise Ratio) do sinal recuperado através da transformada de Haar [7].

Através desse método é proposta também uma ferramenta em tempo real de monitoramento das condições de equipamentos. Idéia que é o centro deste trabalho, esta

análise consiste em calcular a DWPT do sinal e selecionar os principais componentes, então uma rede neural artificial é utilizada para extrair a informação monitorada e indicar o estado atual do equipamento, podendo assim ser utilizado para realizar previsões de falhas.

3.6.4. ANÁLISE DE PROCESSOS ESTOCÁSTICOS.

As wavelets também são extremamente úteis quando nos referimos a probabilidade e estatísticas de séries numéricas, sendo bastante utilizada em Economia. Ela é utilizada para estimar o risco de investimentos baseado na variância do mercado e na covariância entre o mercado e um portfólio [10].

Outro exemplo é a aplicação na modelagem da velocidade do vento em um local de destino a partir da velocidade e direção do vento de referência através de séries temporais. A técnica proposta transforma as séries temporais em uma representação por transformada wavelet packet e utiliza métodos de modelagem estatísticos para identificar quais wavelet packets são úteis para modelar a série [11].

4. DESENVOLVIMENTO E IMPLEMENTAÇÃO

Este trabalho está sendo desenvolvido para a implementação da transformada *wavelet packet* em VHDL, que irá extrair a energia do sinal em bandas de frequência sem perder a referência temporal do sistema, e assim alimentar uma rede neural para realizar a predição de falhas nos equipamentos.

Para isto foi escolhida a transformada Daubechies 6. A família de transformada Daubechies foi introduzida pela matemática Belga Ingrid Daubechies em 1988, esta é uma família de *wavelets* ortogonais cuja a ordem N (Daubechies N) representa o número de coeficientes dos filtros passa-altas e passa-baixas necessários para calcular a DWPT.

A Figura 11 representa as funções de escalonamento e funções *wavelets* de três Daubechies, e sua respectiva representação no domínio da frequência através da transformada de Fourier. Através destes gráficos é possível visualizar como os filtros de Daubechies se aproximam dos filtros passa-altas e passa-baixas e formam um conjunto de funções ortogonais.

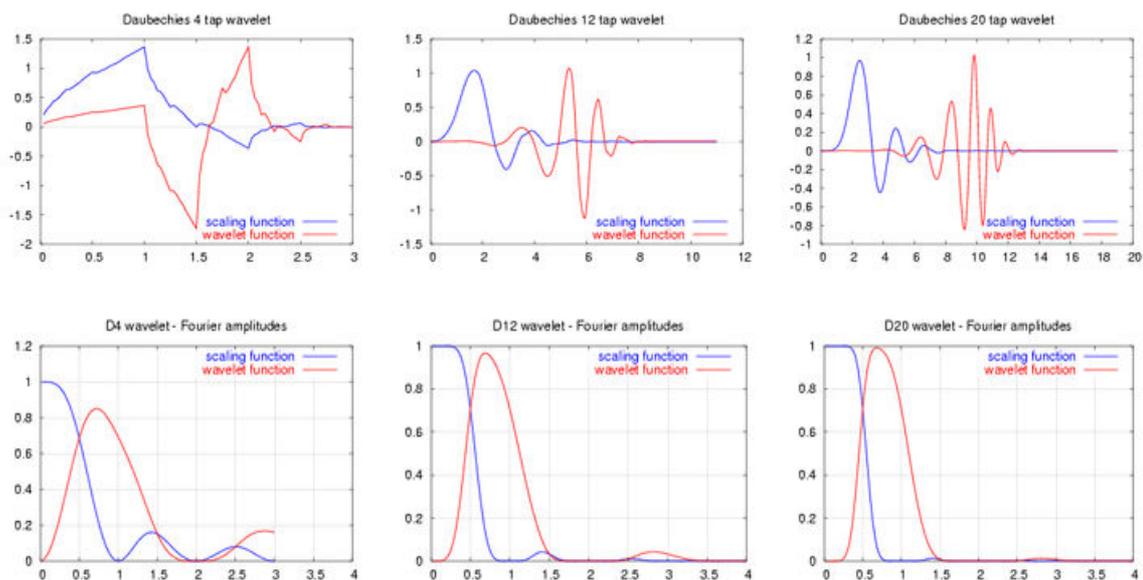


Figura 11 - Representação das Daubechies de 4, 12 e 20 taps e seu espectro em frequência.

Observa-se que as equações 4.1b e 4.1c e as equações 4.1d, 4.1e representam a mesma informação, x_1 e x_2 respectivamente, representando assim um excesso de informação, que é descartado através do processo de decimação do sinal filtrado.

As transformadas de Daubechies 2 é a própria transformada de Haar, apenas aumentado o número de coeficiente da função *wavelet* que na transformada de Haar são $[h_0, h_1] = \left[\frac{1}{2}, \frac{1}{2}\right]$. Assim toda a família Daubechies é calculada desta forma, apenas expandindo o número de coeficiente dos filtros.

A Figura 13 representa a generalização da matriz *wavelet* para transformada com filtro de ordem $N = L + 1$, já com o processo de decimação.

For a filter of order $L + 1$ the wavelet transformation matrix has the form:

$$W_N = \begin{pmatrix} h_L & h_{L-1} & h_{L-2} & h_{L-3} & \cdots & h_1 & h_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & h_L & h_{L-1} & \cdots & h_3 & h_2 & h_1 & h_0 & \cdots & 0 & 0 \\ & & & & \ddots & & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & h_L & h_{L-1} & h_{L-2} & h_{L-3} & \cdots & 0 & 0 \\ & & & & \ddots & & & & & & & \\ h_{L-2} & h_{L-3} & h_{L-4} & h_{L-5} & \cdots & & & & & \cdots & h_L & h_{L-1} \\ \hline g_L & g_{L-1} & g_{L-2} & g_{L-3} & \cdots & g_1 & g_0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & g_L & g_{L-1} & \cdots & g_3 & g_2 & g_1 & g_0 & \cdots & 0 & 0 \\ & & & & \ddots & & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & g_L & g_{L-1} & g_{L-2} & g_{L-3} & \cdots & 0 & 0 \\ & & & & \ddots & & & & & & & \\ n_{L-2} & g_{L-3} & g_{L-4} & g_{L-5} & \cdots & & & & & \cdots & g_L & g_{L-1} \end{pmatrix}$$

Figura 13 - Generalização da matriz *wavelet*. [23]

Os coeficiente da função de escalonamento são calculados a partir dos coeficientes da função *wavelet* da seguinte forma [23]:

$$g_k = (-1)^{k-1} h_k \quad (4.2)$$

4.1. ALGORITMO

Escolhida a transformada wavelet, foi utilizado o Matlab para simular a sua resposta e desenvolver um algoritmo de possível implementação em VHDL.

O *software* Matlab possui um pacote de transformadas *wavelets*, e este foi usado como referência para verificar a resposta do algoritmo a ser implementado, chegando ao seguinte resultado.

```

begin
  lx = Lenght(Signal)
  C[1, lx] = Signal
  for ( m = 0 to mMax )
    for ( p = 0 to 2m - 1 )
      X = C[lxp, lxp + lx]
      for ( n = 1 to  $\frac{\text{lenght}(X)}{2}$  )
        a[n] =  $\sum_{i=0}^{N-1} h_i X[2n - i]$ 
        d[n] =  $\sum_{i=0}^{N-1} g_i X[2n - i]$ 
      end for
      if p is odd
        C[lxp, lxp + lx] = [ a d ]
      else
        C[lxp, lxp + lx] = [ d a ]
      end if
    end for;
    lx =  $\frac{lx}{2}$ 
  end for;
  Out = C2
end

```

Aonde m representa o nível da árvore *wavelet packet* que esta sendo processado no momento, m_{Max} o número máximo de níveis na árvore, p o coeficiente da árvore que esta sendo calculado, e as equações 4.3a e 4.3b representam os filtros passa-baixas e passa-altas da transformada *wavelet* já com o processo de decimação, e h_i e g_i os coeficientes da transformada de Daubechies.

$$a[n] = \sum_{i=0}^{N-1} h_i X[2n - i] \quad (4.3a)$$

$$d[n] = \sum_{i=0}^{N-1} g_i X[2n - i] \quad (4.3b)$$

4.2. ESPECIFICAÇÃO E HARDWARE UTILIZADO.

Para esta aplicação será calculada a árvore *wavelet packet* até o sexto nível, porém o módulo não será fixo, ele deve ser maleável de modo a ser utilizado em outras aplicações, dessa forma os coeficientes dos filtros e o número de níveis a serem calculados serão armazenados em memória e serão utilizados pela máquina de controle para determinar quantos coeficientes deverão ser calculados.

A precisão dos cálculos deve ser em ponto flutuante de 32 bits e todas as operações matemáticas serão realizadas utilizando as bibliotecas da Xilinx.

O FPGA utilizado será o Virtex-II XC2VP30 FF896 e a memória utilizada é uma Kingston KVR266x64C25/512, que é o hardware disponível para desenvolvimento no laboratório.

4.3. ARQUITETURA.

Sandro Vilela da Silva [24], implementou a DWT para aplicação na compressão de imagens no formato JPEG2000, e sugeriu algumas formas construtivas.

A primeira foi a RPA (*Recursive Pyramid Algorithm*), que é comumente utilizada para a implementação da DWT. Este algoritmo permite a computação dos coeficientes de saída de todos os níveis da árvore *wavelet* de ordem superior a primeira de modo intercalado com a computação dos coeficientes de saída da primeira oitava. Deste modo o tempo total de computação de todas as oitavas é igual ao tempo de computação da primeira oitava, o que é muito vantajoso no caso da DWT uma vez que a árvore não é simétrica, a cada oitava é reduzido o número de coeficientes a serem calculados. Outra vantagem do uso do RPA é a redução da quantidade de memória para o armazenamento dos dados para a computação de todas as oitavas.

No caso da *DWPT* todos os níveis da árvore *wavelet packet* possuem o mesmo número de amostras, de tal forma que o tempo de processamento para elas é o mesmo podendo assim calcular cada nível de uma vez sem que isto signifique uma perda de processamento.

A arquitetura que será implementada neste trabalho é semelhante a arquitetura de filtros paralelos apresentada por Sandro Vilela da Silva, que utiliza dois filtros em paralelos para calcular os coeficientes *wavelet* e de escalonamento ao mesmo tempo de forma recursiva. Este método apresenta muitas vantagens em sua implementação por ser de fácil expansão apesar de ter uma máquina de controle complexa, de tal forma que para aumentar o número de níveis da árvore, o número de elementos lógicos (LE) consumidos do FPGA é quase constante, havendo um aumento mínimo gerado por alterações na máquina de controle [24].

Como a maior preocupação neste projeto é a área que a *DWPT* irá ocupar no FPGA será implementado apenas um filtro adaptativo, que será utilizado tanto para o filtro passa baixas como para o passa alta, fato que aumentará consideravelmente o tempo de processamento já que os coeficientes não serão calculado em paralelos, e sim um após o outro, porém permite que todo o processo de filtragem seja realizado com apenas um bloco multiplicador e um somador.

Com a arquitetura definida, representada na figura 14, foi estabelecida uma ordem para o desenvolvimento do projeto.

1. Desenvolver o Filtro utilizando apenas um multiplicador e um somador.
2. Tornar o filtro adaptativo para os coeficientes h_i e g_i .
3. Desenvolver a máquina de controle dos filtros para o cálculo dos coeficientes de nível 1 de uma entrada de 2^n amostras (aqui inclui-se a decimação).
4. Desenvolver a máquina de controle para o cálculo de todos os coeficientes de um nível.

5. Desenvolver a máquina de controle para o cálculo de todos os níveis da *wavelet packet*.
6. Desenvolver o cálculo da energia dos coeficientes da *wavelet packet* em bandas de frequência.

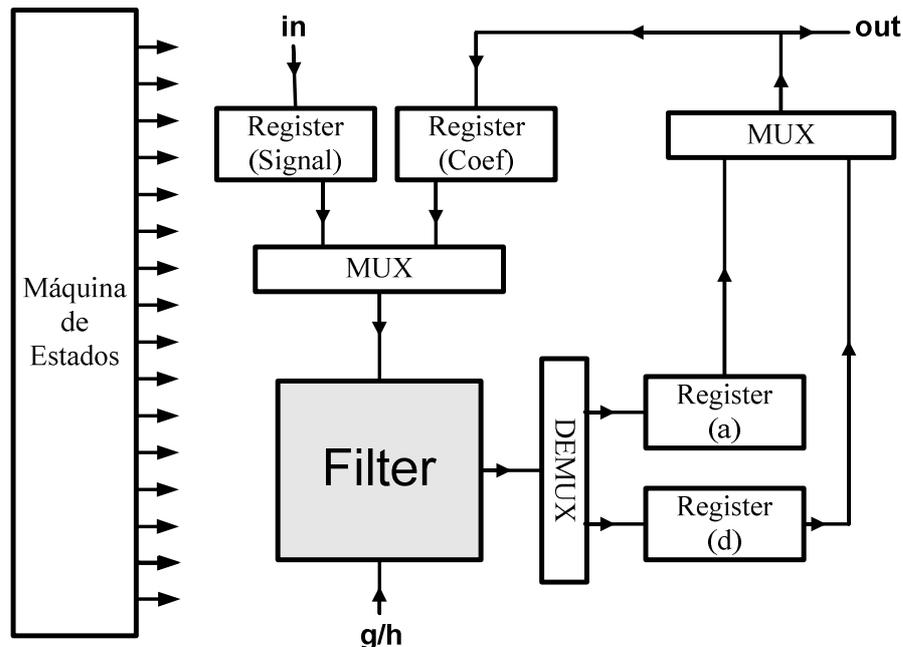


Figura 14 - Arquitetura desenvolvida para a *Wavelet Packet* em VHDL.

4.4. FILTROS PASSA ALTAS E PASSA BAIXAS.

A figura 14 representa a arquitetura utilizada na implementação do filtro adaptativo, o filtro terá três entradas e uma saída, além das entradas e saídas de controle “*start*”, “*reset*”, “*clk*”, “*load*”, “*done*”. A entrada de dados “*In*” na figura 15, é na verdade um ponteiro para a primeira posição na memória onde se encontram os valores a serem filtrados, já que para implementar a Daubechies 6 temos um filtro de mesma ordem.

A entrada “*g/h*” é uma chave seletora para escolher se serão utilizados os coeficientes do filtro passa alta ou do passa baixa, a entrada “*Taps*” indica quantos passos serão realizados pelo filtro, e deve ser diferente da ordem do filtro apenas nas primeiras amostras a serem filtradas, e a única saída “*Out*” é o resultado em ponto flutuante do filtro.

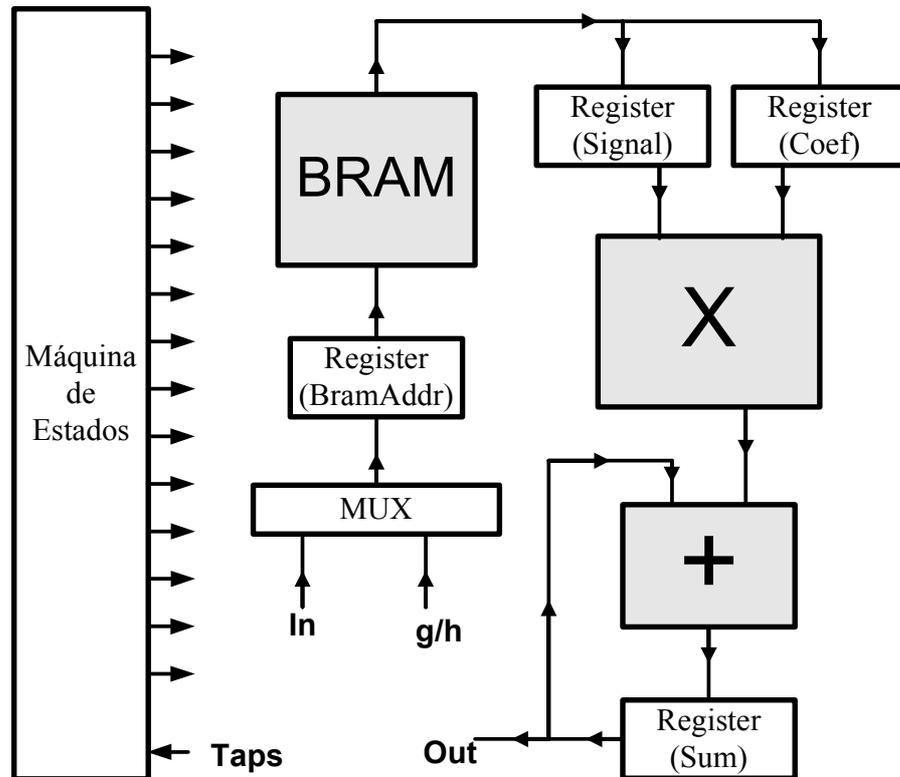


Figura 15 - Arquitetura do filtro adaptativo.

O fato de a entrada ser um ponteiro e não o dado a ser computado se faz necessário para reduzir o número de portas de entrada e manter um padrão na leitura dos dados a serem computados. Para a saída este padrão não é vantajoso, pois é apenas um vetor de dado e seria necessário mais um vetor de entrada para apontar o local de memória no qual ele deveria ser escrito.

Esta estrutura de entradas e saídas reduzidas facilita o controle deste bloco pelo laço principal, e também permite que este laço tenha total controle sobre a região da memória que será ocupada.

A tabela 1 contém os coeficientes dos filtros da Daubechies que serão armazenados em memória para realizar o processo de filtragem do sinal.

Tabela 1 - Coeficientes da Daubechies 6.

Coef	Decimal	Hexa	Coef	Decimal	Hexa
g_0	0.332670552950	3EAA53CB	h_0	0.035226291882	3D104972
g_1	0.806891509311	3F4E9071	h_1	0.085441273882	BDAEFBD6
g_2	0.459877502118	3EEB7510	h_2	-0.135011020010	BE0A4054
g_3	-0.135011020010	BE0A4054	h_3	-0.459877502118	3EEB7510
g_4	-0.085441273882	BDAEFBD6	h_4	0.806891509311	3F4E9071
g_5	0.035226291882	3D104972	h_5	-0.332670552950	3EAA53CB

4.5. CÁLCULO DO COEFICIENTE.

Esta é basicamente uma máquina de controle, este módulo deve filtrar os sinais de entrada com ambos os filtros, passa-altas e passa-baixas, e salvar o seu resultado na memória. Este bloco possui, além das entradas e saídas padrão, quatro entradas, que são um ponteiro para a posição na memória em que se inicia o vetor a ser filtrado, um para a posição onde ele deve escrever a saída, a quantidade de amostras que este vetor possui, que é o mesmo para a entrada e saída, e o índice deste coeficiente no nível ‘ p ’ da árvore *wavelet packet*, já que a ordem de saída do coeficiente *wavelet* e de escalonamento são intercalados.

Como este bloco escreve a saída diretamente na memória, ele não possui nenhuma saída além da saída de controle “*Done*”. A arquitetura utilizada está representada na figura 16.

Para garantir que o processo de decimação não cause nenhum efeito indesejável ao sinal, como perda de informação, é necessário que o coeficiente tenha um número par de amostras e como em cada nível da árvore *wavelet packet* temos o dobro de coeficientes do nível superior é conveniente que o sinal de entrada tenha um número de amostras de 2^n .

Dessa forma a entrada “*Coef Length*” fica limitada a exponencial de 2. É importante salientar que isto não é um requisito, porém permite que o módulo se adapte mais facilmente a valores diferentes de níveis e de ordem da *wavelet packet*, já que evita que o número de amostras em cada coeficiente seja ímpar em qualquer nível.

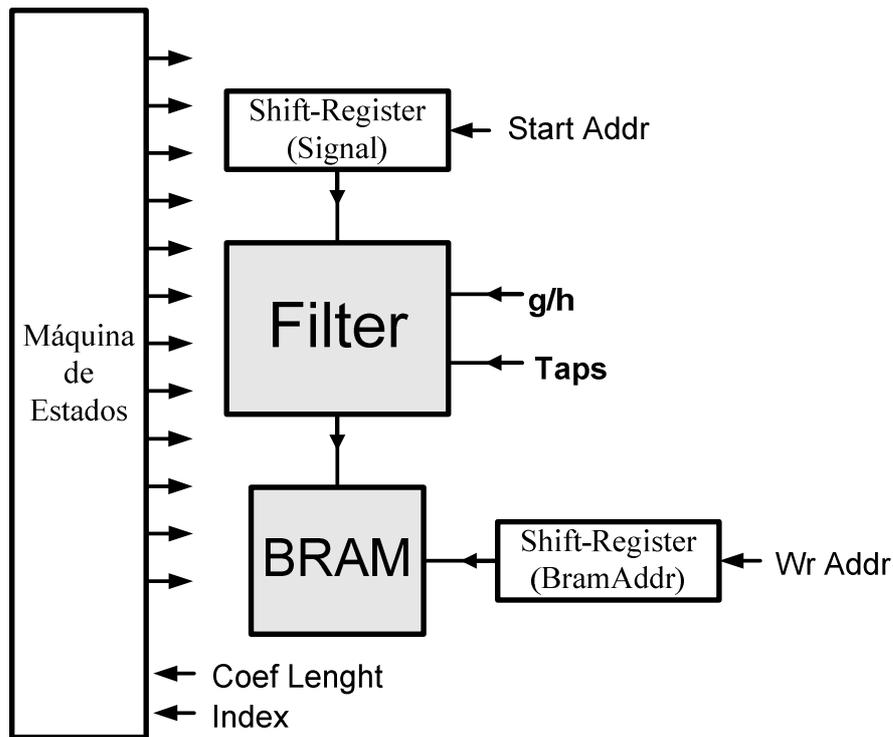


Figura 16 - Arquitetura do bloco de controle de um coeficiente.

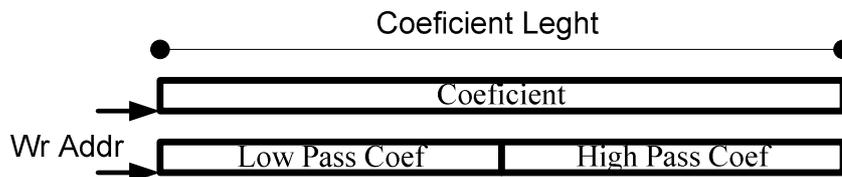


Figura 17 - Representação da memória no cálculo dos coeficientes.

A figura 17 mostra como será a utilização da memória neste bloco, de forma que faz necessário ocupar duas vezes o número de amostras que temos em um coeficiente em memória para realizarmos toda a operação de filtragem do sinal. Como primeiro nível da

árvore *wavelet packet* é calculado com todo o sinal de entrada, então se faz necessário ter disponível duas vezes o tamanho do sinal em memória.

4.6. CÁLCULO DOS COEFICIENTES DE UM NÍVEL.

Da mesma forma do bloco anterior, este é um bloco de controle. Assim obtêm-se entradas e saídas idênticas, porém atuarão diferente na máquina de controle, pode-se então utilizar a figura 15 para representar a arquitetura utilizada neste bloco do projeto.

A entrada “*index*”, aqui vai se referir ao nível da árvore *wavelet packet* que estamos calculando, que vai determinar principalmente quantos coeficientes de “*Coef Length*” deverão ser processados, note que esta entrada é a mesma para ambos os blocos.

A organização da memória, tanto com os valores de entrada como com os valores de saída, esta representado na figura 18. A função principal deste bloco é garantir que as saídas dos coeficientes fiquem na ordem representada, para que no final não seja necessário reordenar estes valores para obter a energia nas bandas de frequência desejadas.



Figura 18 - Arranjo da memória no cálculo de um nível da árvore *wavelet packet*.

4.7. ÁRVORE WAVELET PACKET.

Novamente tem-se apenas um bloco de controle, estes três últimos poderiam ser executados juntamente, porém a divisão em blocos facilita a implementação e de certa forma diminui a probabilidade de erros aparecerem durante o desenvolvimento.

Este bloco será responsável por armazenar o sinal de entrada em memória para que seja possível realizar o processamento com um grande número de amostras, e efetuar o

cálculo dos coeficientes por quantas vezes for necessário para obtermos a resolução em frequência desejada. Neste projeto é necessário termos a representação do sinal em 20 bandas de frequência ($2^m > 20$), então é necessário que se tenha pelo menos um $m = 5$, que representa o número de níveis menos um, na árvore *wavelet packet*, já que o primeiro nível da árvore é o próprio sinal.

Como a saída do sistema não é exatamente a árvore completa, porém a energia do sinal em bandas de frequência, não se faz necessário armazenar todos os coeficientes calculados, dessa forma a região de memória do sinal que está sendo processado pode ser considerada livre no passo seguinte, então esta região será reservada para guardar a saída do próximo passo, de tal forma que para executar todo o processamento do sinal, seja necessário duas vezes o número de amostras que o sinal de entrada possui.

4.8. ENERGIA DO SINAL.

Como comentado no capítulo 2, as transformadas são utilizadas como ferramentas na análise de sinais, elas permitem a visualização de um mesmo sinal por vários ângulos, de forma que seja possível obter informações que não seria possível no domínio do tempo. Como toda ferramenta, a transformada *wavelet packet* apenas facilita o trabalho, e a partir dela é possível chegar ao resultado, que neste caso é a representação do sinal em bandas de frequência.

4.8.1. TEOREMA DE PARSEVAL.

O teorema de Parseval demonstra uma aplicação da transformada de Fourier relacionando a energia transportada por um sinal pela sua transformada de Fourier [25]. A energia de um sinal é definida por:

$$W = \int_{-\infty}^{+\infty} f^2(t)dt \quad (4.4)$$

onde W é a energia do sinal $f(t)$.

Ao aplicar uma transformada com uma base ortonormal, como é o caso da transformada de Fourier, a relação de energia do sinal é mantida, dessa forma o teorema de Parseval afirma que podemos obter a energia do sinal a partir de sua FT através da seguinte equação:

$$W = \frac{1}{2\pi} \int_{-\infty}^{+\infty} |F(\omega)|^2 d\omega \quad (4.5)$$

e podemos obter a energia de uma banda limitada de frequência da seguinte forma:

$$W = \frac{1}{\pi} \int_{\omega_1}^{\omega_2} |F(\omega)|^2 d\omega \quad (4.6)$$

onde a banda largura de banda B é determinada por $\omega_2 - \omega_1$.

Apesar de não aplicar a FT, esta sendo utilizada a DWPT para fazer uma representação do sinal no domínio da frequência de maneira equivalente, porém sem perder a referência temporal. Desta forma pode-se utilizar o mesmo teorema de Parseval para obter a energia do sinal, e como os coeficientes da árvore *wavelet packet* já representam o sinal em bandas de frequência, basta calcular a energia de cada coeficiente.

4.8.2. CÁLCULO DA ENERGIA

Apesar dos coeficientes já representarem a saída em bandas de frequência, um coeficiente pode ter mais de uma amostra, dessa forma continua sendo necessário integrar cada coeficiente para obter a sua energia.

A arquitetura de *hardware* definida para realizar este processo esta representada na figura 19, onde “*Steps*” representa o número de amostras que deverão ser integradas, “*In*” é o endereço da primeira amostra do coeficiente, e “*Out*” é a energia do coeficiente.

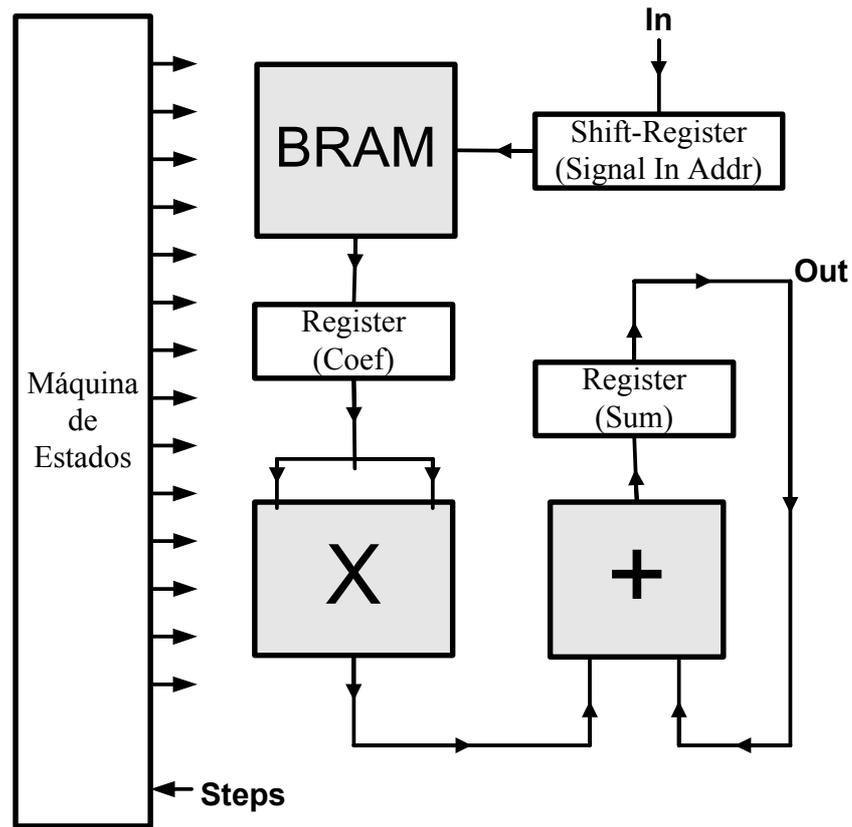


Figura 19 - Arquitetura para o cálculo da energia.

4.9. MAPA DE MEMÓRIA.

Para evitar o uso de registradores que são limitados dentro do FPGA, todos os valores são mapeados em memória. Dessa forma foi montada a estrutura representada na tabela 2:

Tabela 2 - Mapa da memória.

0x0000	Signal Lenght
0x0001	m (Wavelet Level)
0x0002	p (number of coefficients)
0x0010..0x016	H Coefficients
0x0020..0x0026	G Coefficients
0x0030..0x2FFF	Signal In
0x2030..0x3FFF	Energy Out

5. RESULTADOS.

No período do trabalho foi possível completar o código em VHDL para o cálculo de um coeficiente da árvore *wavelet packet*, restando desenvolver o código para a máquina de controle que utiliza esta função para gerar a árvore completa e calcular a energia do sinal.

A simulação realizada para o bloco já finalizado encontra-se no Apêndice A. Para realizar esta simulação foi escolhido um vetor qualquer de 16 amostras, sendo ele [9 1 8 1 7 1 6 1 5 1 4 1 3 1 2 1], e seu resultado foi comparado com o obtido no Matlab.

Através da simulação (Apêndice A) pode-se montar o mapa de memória da saída do sistema representado na tabela 3, e então comparar ela com a matriz de saída da função *wavelet packet* criada no Matlab calculada apenas para um nível.

Tabela 3 - Resultado da Simulação.

Endereço	Valor em float	Valor	Endereço	Valor em float	Valor
0x2037	0xBF3BD6BA	-0.7337452	0x203F	0x40DDBD42	6.9293528
0x2036	0x4056C1DD	3.3555825	0x203E	0x408E51CC	4.4474850
0x2035	0x40D906BD	6.7820725	0x203D	0x4085B2DA	4.1780825
0x2034	0x40C2661E	6.0749655	0x203C	0x405E2476	3.4709754
0x2033	0x40ABC580	5.3678590	0x203B	0x4030E00C	2.7636747
0x2032	0x409524E2	4.6607523	0x203A	0x4003A1FD	2.0567620
0x2031	0x407D0888	3.9536457	0x2039	0x3FACC181	1.3496553
0x2030	0x404FC74A	3.2465386	0x2038	0x3F247E0C	0.6425483

Com estes resultados concluí-se que o algoritmo implementado esta funcionando perfeitamente.

6. CONCLUSÕES E TRABALHOS FUTUROS.

Foi visto neste trabalho que a Transformada Wavelet possui uma ampla área de atuação, sendo aplicada fortemente em processamento de sinais e imagens tanto na computação como na indústria.

A sua implementação baseada na análise de Mallat simplifica o hardware, permitindo que ela seja processada em cascata utilizando sempre o mesmo bloco operacional, o que reduz os requerimentos de hardware, e até mesmo seu algoritmo para implementação em software fica simplificado.

A vantagem da PWT de manter a referência temporal e ter uma saída já distribuída em bandas de frequência facilita a análise da energia do sinal e permite sua utilização em sinais não estacionários, e por estas características esta sendo bastante utilizada para realizar simulações e previsões de eventos naturais.

Como discorrido neste trabalho a DWT e a PWT possuem diversas aplicações que podem ser utilizadas como fonte de estudo para outros trabalhos. Computacionalmente ela pode ser facilmente implementada, podendo ser realizado um trabalho focado na sua aplicação, como por exemplo, análise de sinais sonoros. Já para uma aplicação em hardware, a sua implementação consome um tempo muito grande, fato que pode tornar o trabalho limitado a própria transformada *Wavelet*.

7. REFERÊNCIAS

- [1] Abraman. “Quanto custa e como é a manutenção no Brasil/2007”. www.brasilengenharia.com.br, Novembro de 2007.
- [2] Haykin, Simon; Veen, Barry Van. ”Sinais e Sistemas”. Bookman, pp 163-246, 2002.
- [3] Parraga, Adriane. “Aplicação da Transformada Wavelet Packet na Análise e Classificação de Sinais de Vozes Patológicas”. Universidade Federal do Rio Grande do Sul. pp 44-69. 2002.
- [4] Vidakovic, Brani; Mueller, Petter. “Wavelets For Kids, A Torial Introduction.”. Dake University.
- [5] Cesar, Roberto M.-Jr. “Transformadas Lineares: Fourier, Wavelets”. Departamento de Comunicação – IME – USP.
- [6] Kaplan, Ian. “Wavelets and Signal Processing”. http://www.bearcave.com/misl/misl_tech/wavelets/index.html. 2003
- [7] F. Truchetet, O. Laligant. “Wavelets in industrial applications: a review”. Universidade de Bourgogne, França.
- [8] Topiwala , Pankaj N.” Wavelet Image and Video Compression”. Springer Netherlands. 2008.
- [9] Yokoyama, Nobuaki; Kawasaki Zen-Ichiro; Hamid, Efrina Yanti Hamid. “Rms and Power Measurements: A Wavelet Packet Transform Approach”.
- [10] D. Percival, A. Walden, Wavelet methods for time series analysis, Cambridge University Press, Cambridge, UK, 2002
- [11] G.P. Nason, T. Sapatinas, ”Wavelet packet transfer function modelling of nonstationary time series”, *Statistics and Computing*, **12**(1), pp. 45-56, 2002
- [12] Carrier, Pierre. “Wavelets: algorithms and applications”. University of Minnesota. 2009

- [13] Da Silva, Sandro Vilela. “Projeto de uma Arquitetura Dedicada à compressão de imagens no Padrão JPEG2000”. Universidade Federal do Rio Grande do Sul, pp. 50-60, 2005.
- [14] Sadiku, Matthew N. O.; Alexander, Charles K. “Fundamentos de Circuitos Elétricos”. Bookman, pp. 712-715, 2003.
- [15] Cesar, Roberto M. Jr. “Transformadas Lineares: Fourier, Wavelets”. Departamento de computação – IME – USP.
- [16] Dragotti, Pier Luigi; Vetterli, Martin. “Wavelet Footprints: Theory, Algorithms, and Applications”. IEEE Transactions on Signal Processing, Vol. 51, No. 5, May 2003.
- [17] Mahmoud, Mohamed I.; Dessouky, Moawad I. M.; Deyab, Salah, Elfouly, Fatma H. “Comparison between Haar and Daubechies Wavelet Transformions on FPGA Technology”. World Academy of Science, 2007.
- [18] V. Herrero, J. Cerdà, R. Gadea, M. Martínez, A. Sebastià. “Implementation of 1-D Daubechies Wavelet Transform on FPGA”. Universidad Politécnica de Valencia.

APÊNDICE A.

Nas figuras a seguir encontra-se a simulação do código VHDL já implementado.

