Universidade Federal de Uberlândia
Centro de Ciências Exatas e Tecnologia
Pós-Graduação em Engenharia Elétrica

Tese:

Novos Coeficientes Wavelets Baseados em Intervalos Musicais para Análise de Timbres de Instrumentos Acústicos.

João Cândido Lima Dovicchi

Outubro
- 1999 -

Universidade Federal de Uberlândia
Centro de Ciências Exatas e Tecnologia
Departamento de Engenharia Elétrica

Tese:

Novos Coeficientes Wavelets Baseados em Intervalos Musicais para Análise de Timbres de Instrumentos Acústicos.

Tese apresentada por João Cândido Lima Dovicchi à Universidade Federal de Uberlândia (UFU), para obtenção do Título de Doutor em Engenharia Elétrica, aprovada em 27 de outubro de 1999 pela banca examinadora: Professor Doutor Gilberto Arantes Carrijo (orientador), Professor Doutor Luciano Vieira Lima (co-orientador), Professor Doutor João Bosco da Mota Alves (UFSC), Professor Doutor Amilcar Zani Netto (USP), Professor Doutor Calimério Augusto Soares Netto (UFU).

Novos Coeficientes Wavelets Baseados em Intervalos Musicais para Análise de Timbres de Instrumentos Acústicos.

João Cândido Lima Dovicchi

Tese apresentada por João Cândido Lima Dovicchi à Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica.

Prof. Gilberto Arantes Carrijo, Dr.
Orientador

Prof. Darizon Alves de Andrade, Dr.
Coordenador
Curso de Pós-Graduação

Ao Marcelo *in Memoriam*

Agradecimentos

Agradeço àqueles que contribuíram para a ampliação do meu horizonte. Ao fazê-lo, também contribuíram para a ampliação da minha ignorância, pois descobri quão pouco sei.

Ao **Departamento de Música e Artes Cênicas** e à **Coordenação dos Cursos de Música** da Universidade Federal de Uberlândia, pelo apoio recebido. Ao Sr. Miguel Correia Lima, que, graciosamente, facilitou o meu acesso à Rede Internet através de sua empresa: **Internet Triângulo Provider**, onde grande parte do levantamento bibliográfico deste projeto foi realizado. Ao Núcleo de Processamento de Dados da UFU (NUPRO), ao Centro Nacional de Processamento de Alto Desempenho de Minas Gerais e do Nordeste (CENAPAD/MG e CENAPAD/NE) Ao Prof. Henrique Barbosa Leite pelos constantes esclarecimentos sobre programação e utilização do Linux. Ao Prof. Lawrence Fritts, do Electronic Music Studios da University of Iowa, Iowa City, pela gentil permissão de utilização dos sons gravados por sua equipe. Ao Prof. Waldenor Barros de Moraes Filho pela leitura crítica e revisão do texto. À Profª Drª Célia Zorzo Barcelos pelo auxílio inestimável na revisão matemática. A todos aqueles que me incentivaram e me ajudaram nesse caminho, principalmente, a minha mãe.

Conteúdo

Lista de Tabelas

Lista de Figuras

"The verie essence and, as it were, springheade and origine of all musiche is the verie pleasaunte sounde which the trees of the forest do when they growe."

EDGARD ALAN POE (IN: AL AARAAF)

Resumo

Neste trabalho são apresentados dois novos conjuntos de coeficientes para a equação de dilação, com a finalidade de gerar filtros *Wavelets* para a análise de sinais de instrumentos musicais. Tais coeficientes foram calculados com base nos fatores do intervalo cromático e foram denominados **Chroma4** e **Chroma6**. Os estudos com as duas funções transformadas *Wavelets* obtidas, mostraram que é possível recuperar o sinal original a partir dos sinais transformados, com grande precisão nos resultados, provando sua eficiência na análise de sinais sonoros de instrumentos musicais. A combinação das transformadas localizadas de Fourier e os filtros *Wavelets* demonstraram que o modelo proposto possibilita uma análise de multiresolução facilitando a identificação de freqüências específicas e a localização de transientes nas regiões de interesse no sinal.

Processamento Digital De Sinal, Wavelets, Eletroacústica, Música.

Abstract

This work presents two new sets of coefficients of the dilation equation for the construction of *Wavelet* filters to be applied in musical instruments sounds analysis. These coefficients were obtained from the chromatic interval factors and named here as **Chroma4** and **Chroma6**. Studies with those both *Wavelets* transform functions showed that it is possible to rebuild the original signal from the transformed ones with great precision, with results that prooved to be efficient in musical instruments sounds analysis. Combining short time Fourier transform and the two new *Wavelet* filters one notice that the proposed model yelds good results in a multiresolution analysis and helps specific frequencies identification besides transient localization at the signal's portion of interest.

DIGITAL SIGNAL PROCESSING, WAVELETS, ELECTROACOUSTICS, MUSIC.

Capítulo 1

Introdução

"Não será nosso dever encontrar meios sinfônicos de expressar nosso tempo, meios que evoquem o progresso, o arrojo e as vitórias dos dias de hoje? O século do avião merece sua própria música."
Claude Debussy

O principal foco deste trabalho é apresentar o resultado das pesquisas feitas com novos coeficientes da equação de dilação, comumente denominados coeficientes *Wavelets*, por meio da combinação de duas ferramentas analíticas com a finalidade de obter melhores resultados na análise de timbres de instrumentos musicais acústicos.

A pesquisa de métodos de síntese de timbres que reproduzam com maior fidelidade os instrumentos acústicos depende de métodos de análise que facilitem o cálculo de funções de transferência e que permitam, aos sintetizadores, produzir sons cada vez mais próximos do som natural. Além disso, ferramentas de análise mais poderosas podem permitir a criação de timbres ainda não explorados que venham a ser utilizados na composição de música eletroacústica.

1.1 Praeludium

O interesse pela música eletrônica não é recente. Busoni[13], já havia escrito sobre ela em 1907, na época um vago sonho, em seu *Esboço de uma Nova Estética da Música* (ampliado e republicado em 1910). Inspirada no *telharmonium*[1] do cientista canadense Taddheus Cahill, a profecia de Busoni nao tardaria a acontecer. As décadas de 20 e 30 presenciaram o surgimento de inúmeros meios de produção de ondas sonoras por intermédio de aparelhos eletro-eletrônicos. Leon Theremin criava seu *aetherophone*[2] ou *theremin* enquanto surgiram as "ondas martenot" do francês Maurice Martenot e o *trautonium*[3] do alemão Friedrich Trautwein. Mas foi apenas na década de 50 que o americano Bob Moog criou o sintetizador, que inicialmente utilizava controladores análogos, passando depois para digitais, em teclados convencionais ou outros tipos de controladores. Assim disse Bob Moog[70] na abertura da III Convenção Anual CyberArts[4]:

> *"When people think of a synthesizer, they think generally of one being controlled by a keyboard — specially a synthesizer used in vernacular or commer-*

[1] Instrumento analógico provido de chaves que gerava sons por meio de impulsos elétricos, criado em 1906.

[2] Aparelho provido de antenas que gera sons a partir de interferência

[3] Tipo de teclado eletrônico

[4] "Quando as pessoas pensam em um sintetizador, pensam geralmente em um controlado por um teclado — especialmente os sintetizadores usados na música comercial ou vernacular. Nos primeiros dias da *'Moog music'*, nós desenhamos sintetizadores análogos modulares que podiam ser controlados não apenas por teclados, mas por uma série de outros controladores, muitos deles com capacidade de alterações contínuas. (trad. do autor)

> *cial music. This says more about our vernacular music than it does about the technology. In early days of 'Moog Music', we designed modular analog synthesizers that could be controlled not only by keyboards but by a series of other controllers, most of which were capable of continous change."*

Hoje, os computadores controlam módulos de som (sintetizadores) através de um protocolo de comunicação chamado MIDI (*Music Instrument Digital Interface*) inclusive com a utilização de parâmetros complexos para a expressão da envoltória do sinal sonoro [37].

Uma das personalidades mais importantes da música eletrônica, foi Leon Theremin. Lev Termen, seu nome russo, acreditava que a fusão da física com a música poderia mudar o panorama artístico e definir uma nova estética através da aplicação da tecnologia às artes[36]. Graduado em física e violoncelista amador, Theremin inventou, na década de 20, um instrumento musical eletrônico baseado em dois osciladores de rádio-freqüência, sintonizados na mesma freqüência e que, com os movimentos da mão em torno de uma antena, criavam efeitos de interferência na freqüência fixa de outro oscilador, denominados efeitos *heterodyning*. O instrumento logo despertou o interesse da comunidade artística e, em 1924, o primeiro concerto para theremin e orquestra de Andrei Pashchenko[5] foi executado em Leningrado. Entre 1924 e 1930, Theremin se apresentou em grandes salas de concerto, executando obras originais para o instrumento e transcrições de Offenbach, Scriabin, Schubert e outros, com grande sucesso. A música entrava na era eletrônica.

---

[5] "Mistérios Sinfônicos" para orquestra e aetherophone de Andrei Pashchenko, 1923.

1.2 Análise e Síntese (Intermezzo)

A síntese digital de áudio tem sido um dos principais objetos de pesquisa na área de processamento de sinais. Inúmeros algoritmos de filtros, métodos de síntese, análise e conversão analógico-digitais (A/D) têm surgido recentemente. Os sofisticados circuitos de processamento de som (placas de som) associados aos velozes processadores e a programas específicos, permitem o processamento de sons digitalizados com grande acuidade e facilidade. A constante busca de novos timbres passou a ser viável através das possibilidades criadas pelo advento dos velozes computadores, com grande capacidade de memória, que permitiram o aparecimento de complexos algoritmos matemáticos para o cálculo de funções descritivas do fenômeno acústico. Tais algoritmos são conhecidos hoje como Síntese Áudio Digital (*Digital Audio Synthesis*).

Os principais métodos de síntese digital se resumem em:

1. Síntese Aditiva ou *Additive Synthesis*;
2. Síntese por Freqüência Modulada ou *FM Synthesis*;
3. Síntese por Predição Linear ou *Linear Prediction Synthesis*;
4. Síntese baseada na Teoria Granular ou *Theory of Granular Synthesis*.

Na Síntese Aditiva, obtém-se a representação discreta do sinal no domínio do tempo, através da amostragem de um sinal sonoro. Com a análise destes sinais discretos obtêm-se dados que caracterizam um determinado sinal, que pode ser armazenado ou transmitido para posterior reprodução. A transformação desses dados para o domínio da freqüência é de grande importância para a análise do sinal. Tal transformação do domínio do tempo para o domínio da freqüência é relativamente simples, usando-se métodos matemáticos computacionais disponíveis e amplamente estudados. Para isso, podemos utilizar a transformada de Fourier, transformada de Laplace, transformada Z, transformada Q, dentre outras[31, 82, 107]. Uma das vantagens da aplicação de transformadas ortogonais, em processamento de sinais digitais, é a compressão de dados, além da obtenção do espectro do sinal que pode ser utilizado na análise e ressíntese do mesmo sinal.

A aplicação do método de Síntese Aditiva tem evoluído desde a década de 70. Vários autores discutem, detalhadamente, a produção de sons musicais digitais sintetizados de alta qualidade, abordando os problemas da amostragem de dados em cada estágio do processo [31, 72, 90]. Alguns autores têm pesquisado outras abordagens matemáticas para o tratamento das séries de harmônicos, e sua aplicação na síntese de sons musicais [65, 66, 97]. Em seu trabalho, Marar e Moreira abordam a síntese aditiva, bem como apresentam alguns aspectos básicos, fundamentais para a utilização de computadores em síntese de sons musicais, descrevendo um método através de funções básicas não senoidais, provenientes da transformada de Karhunen-Loève [65, 66].

Algumas classes de sinais sonoros possuem certas particularidades que dificultam a reconstrução precisa do sinal através das informações espectrais. Nesse caso, sinais que apresentam componentes que variam com o tempo de duração do sinal, envolvendo freqüência, amplitude e fase, exigem um tratamento dinâmico e grande demanda de potência aritmética do processador e/ou da quantidade de memória. Isso acontece, principalmente, na síntese de sons musicais. A necessidade de simplificar a geração de sons musicais tem levado à pesquisa de métodos de síntese aditiva, através de funções não senoidais [54, 97]. Mais recentemente, o método de *cross-correlation* da transformada da Constante-Q (*Constant-Q Transform*) também vem sendo investigado para aplicação em síntese sonora [10, 12].

A técnica de síntese por Freqüência Modulada (FM) possibilita um controle temporal mais simples sobre a banda do espectro onde as freqüências podem estar relacionadas de maneiras variadas. A técnica desenvolvida por Chowning [18], consiste na modulação da freqüência de uma determinada portadora senoidal através de um sinal de mesma freqüência ou de freqüência correlata. O espectro resultante contém componentes de amplitudes descritas por uma função Bessel de primeira ordem, $J_n(I)$, onde n é o n-ésimo harmônico e I é o índice de modulação[6]. A equação que define uma onda FM com pico de amplitude A onde portadora e sinal são senoidais é:

$$a = A \cdot \sin(\alpha t + I \cdot \sin \beta t) \tag{1.1}$$

onde a = a amplitude instantânea da portadora modulada, A = amplitude do pico da freqüência modulada, α = a freqüência da portadora em rad/seg, β = a freqüência do sinal modulante em rad/sec e I = índice de modulação [18].

Sounders mostrou as vantagens na utilização de ondas FM triangulares sobre as senoides de Chowning e sobre outros métodos de síntese, no que concerne à velocidade resultante da simplicidade computacional [96]. Derivações mais complexas deste método foram descritas por outros autores [61, 91].

---

[6] Razão entre o desvio da freqüência do pico da portadora e a freqüência do sinal

Recentemente, novos métodos de síntese digital têm sido descritos. O método de predição linear, também conhecido como *Linear Prediction Synthesis*, deriva do método LPC (*Linear Predictive Coding*) já amplamente utilizado na transmissão em telecomunicações, e vem sendo pesquisado para sua utilização na síntese digital do sinal musical [2, 15, 59, 75]. A solução do problema da qualidade na ressíntese de timbres sonoros, usando-se a técnica LPC tem sido estudada para aplicação em Música e Computação, mas ainda de forma insipiente [59]. Na realidade, as técnicas de LPC têm sido usadas na análise e ressíntese de voz. Entretanto, os resultados não têm sido satisfatórios devido ao uso de banda limitada do sinal de pulso para modelar os componentes harmônicos. A técnica de LPC consiste na procura de séries no domínio do tempo de um filtro n-polar de resposta de impulso infinito (IIR[7]), cujos coeficientes se adaptem aos formandos do sinal sonoro. Os sinais são então computados para a recuperação dos sinais originais. Embora essa técnica não tenha sido criada para a obtenção de resultados de alta qualidade, para a utilização em música, alguns trabalhos têm mostrado a possibilidade de aperfeiçoá-la para este fim [2].

A síntese de espectros sonoros por métodos não-lineares tem sido descrita desde a década de 40 [44]. Entretanto, a partir da década de 80, os métodos baseados na Teoria da Síntese Granular têm sido mais minusciosamente pesquisados [87], recebendo contribuições a partir da Teoria do Caos [60] e de outras teorias de sistemas dinâmicos não-lineares [35, 94]. Manzolli [63, 64] usou a teoria de fractais (*Fracwave*) para produção de sons com características dinâmicas através do controle paramétrico de mapas não-lineares simples, o que permite a geração de novos timbres e a construção de estruturas sônicas a partir de

[7]Infinite Impulse Response

um nível atômico, mas o método ainda não se mostrou aplicável à análise ou síntese de sons de instrumentos musicais.

Ultimamente tem havido um grande interesse no estudo de outras funções que possam, à maneira das transformadas, ser utilizadas na síntese do som. Entre elas, as transformadas *Wavelets* têm despertado muita atenção dos cientistas que procuram novos métodos de análise e síntese de sinais [29, 69]. Do mesmo modo que as funções transformadas de Fourier (FFT[8]) [25], as transformadas discretas *Wavelet* (DWT[9]) são operações lineares em um vetor cujo tamanho é uma potência de dois, transformando-o em outro vetor, numericamente diferente, de mesmo tamanho. Os algoritmos das operações com as *Wavelets* apresentam uma grande eficiência e rapidez de processamento e, além disso, por serem funções ortogonais são inversíveis como as FFT.

Enquanto, para a FFT, o domínio das bases é um conjunto de funções trigonométricas bastante conhecidas (senos e cossenos), o domínio das *Wavelets* tem bases um pouco mais complexas, chamadas *Wavelet mãe*. Evidentemente, pode ser encontrada uma infinidade de bases para o espaço das funções, mas o grande interesse pelas *Wavelets* está no fato de serem localizadas no espaço da função — diferentemente dos senos e cossenos — e localizadas em escalas específicas do domínio da freqüência, como as bases de Fourier. Outra diferença entre as bases da FFT e das *Wavelets* é que não existe um único conjunto de bases, como no caso da FFT, mas um conjunto infinito de bases. Essa versatilidade permite que as funções *Wavelets* sejam escolhidas conforme o conjunto a que se destinam,

[8]Fast Fourier Transform
[9]Discrete Wavelet Transform

ou seja, quão compactamente se encontram localizadas no espaço da função, ou quão esparsas no domínio *Wavelet*.

A síntese sonora de timbres de instrumentos musicais depende, basicamente, de dois fatores: do conjunto de parciais harmônicos utilizados, e do envelope sonoro, ou seja, transientes de ataque, decaimento, nível de sustentação e término[10] (ADSR[11]). O envelope sonoro é tão importante que, quando o som de um instrumento é tocado de um tape em reverso ou quando o ataque é removido por um processo de gravação ou corte, mesmo os mais experientes músicos têm dificuldade em reconhecer a fonte sonora [92]. Os sons musicais podem ser classificados em periódicos, ou quase periódicos (harmônicos), e não-periódicos (enarmônicos), embora não se tenha, naturalmente, sons totalmente periódicos ou totalmente aperiódicos. No caso dos instrumentos musicais, a grande maioria é de natureza periódica, como por exemplo as cordas, os sopros e alguns instrumentos de percussão (piano, vibrafone e similares). Os instrumentos de percussão — à exceção dos melódicos — produzem sons de natureza aperiódica.

Devido à natureza quase-periódica do som dos instrumentos melódicos, seus sons podem ser descritos como uma somatória de um número finito de senóides, mas como não são perfeitamente periódicos, as amplitudes e freqüências de cada harmônico podem variar independentemente.

Cada instrumento possue um envelope sonoro característico não só nos transientes de ADSR, como também na altura do som emitido. Por exemplo, a figura 1.1 representa os

[10] Moorer[72] considera apenas *attack*, *steady-state* e *release*. Entretanto adota-se uma nomenclatura mais moderna, como a especificada para o General Standard (GS) MIDI.

[11] do inglês Attack, Decay, Sustain e Release

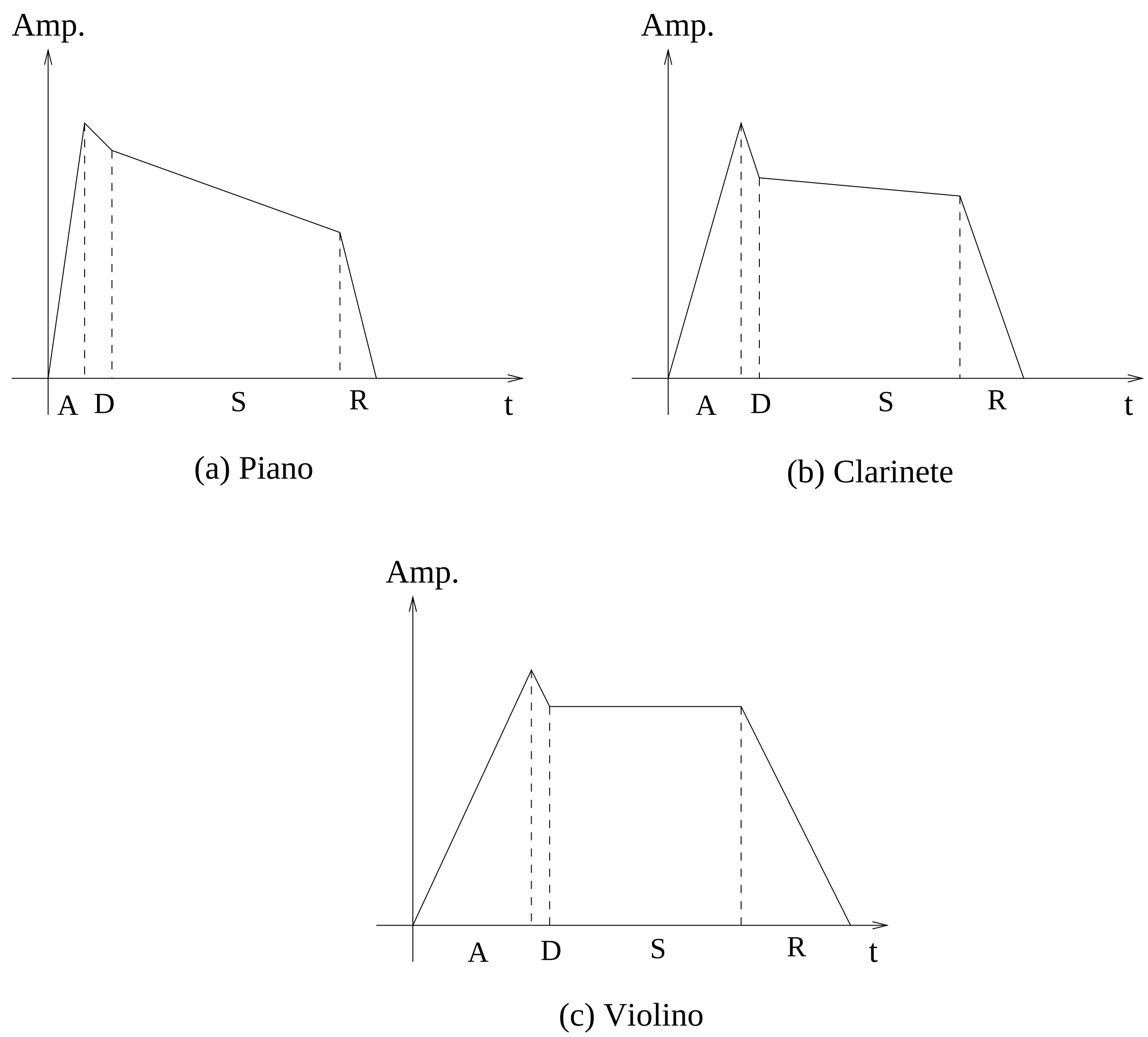

Figura 1.1: Envelopes sonoros de (A) piano, (B) clarinete e (C) violino, emitindo uma nota lá (440.2 Hz).

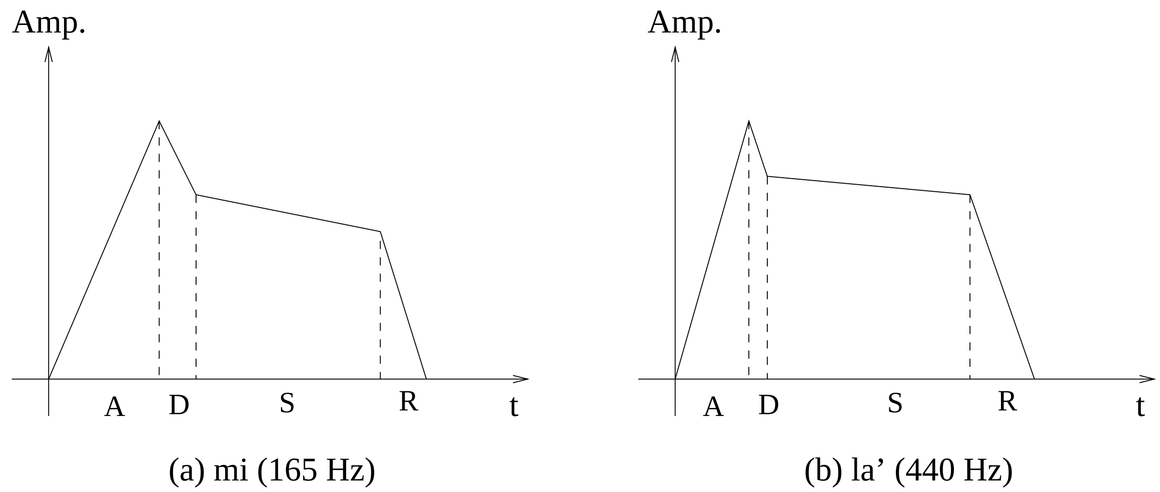

Figura 1.2: Envelopes sonoros de clarinete emitindo uma nota mi (165.0 Hz) e uma nota lá (440.2 Hz).

transientes de uma nota lá (440.2 Hz) emitida pelo piano, clarinete e violino; a figura 1.2 representa uma nota mi (165.0 Hz) e uma nota lá (440.2 Hz) no clarinete.

O espectro de cada freqüência de um instrumento também é um fator importante na qualidade tímbrica. O timbre de uma nota em um determinado instrumento é dado pela amplitude relativa de parciais específicos. Em um clarinete, por exemplo, os harmônicos ímpares (múltiplos da freqüência fundamental de 3, 5, 7, 9, etc.) têm amplitude relativa maior que os pares (cf. tabela C.14), enquanto no violino acontece o inverso; já os metais, apresentam um espectro que contém praticamente todos os harmônicos da série. As características espectrais de um instrumento também dependem da altura da fundamental, ou seja, o espectro varia de nota para nota. O Fagote, por exemplo, chega a apresentar apenas dois harmônicos em uma nota dó (523 Hz) enquanto apresenta dez harmônicos na nota mi (82 Hz) (cf. tabela C.18 no apêndice C.1). A figura 1.3 faz uma comparação entre

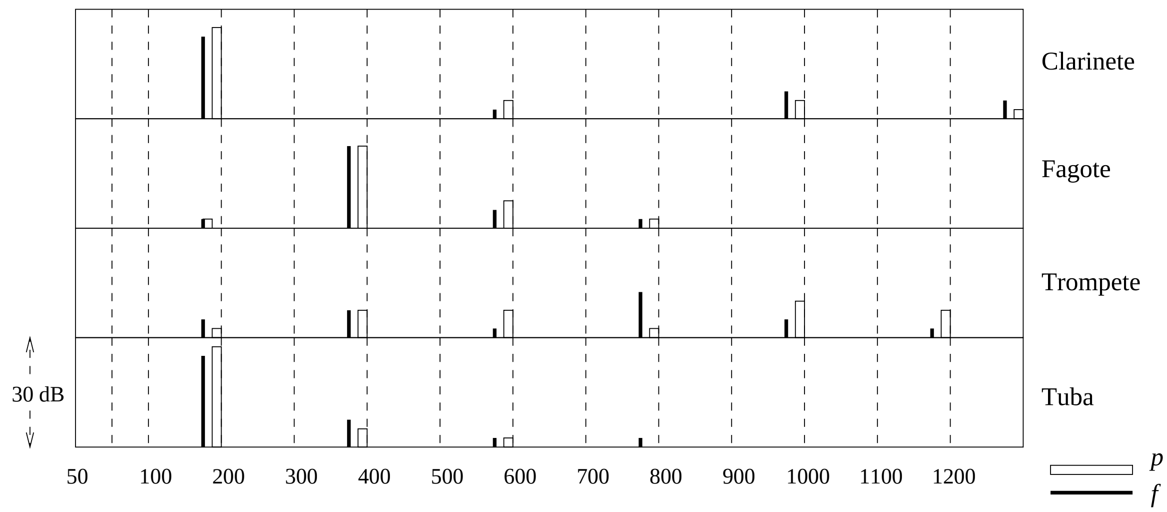

Figura 1.3: Espectro de alguns instrumentos tocando uma mesma nota em *forte* e em *piano* (freqüências em Hz).

os espectros de alguns instrumentos tocando uma mesma nota (Sol 194 Hz).

Ainda, um aspecto dos instrumentos de cordas é a diferença dos espectros de cada corda — uma mesma nota tocada em cordas diferentes apresentam espectros diferentes. Assim, os métodos de análise dos timbres de instrumentos acústicos são de grande importância, não só para a compreensão de suas características, mas também para o cálculo de funções de transferência na síntese de timbres para instrumentos eletrônicos. Este trabalho propõe um novo modelo para a análise de timbres, que poderá abrir novas perspectivas na compreensão desses fenômenos acústicos.

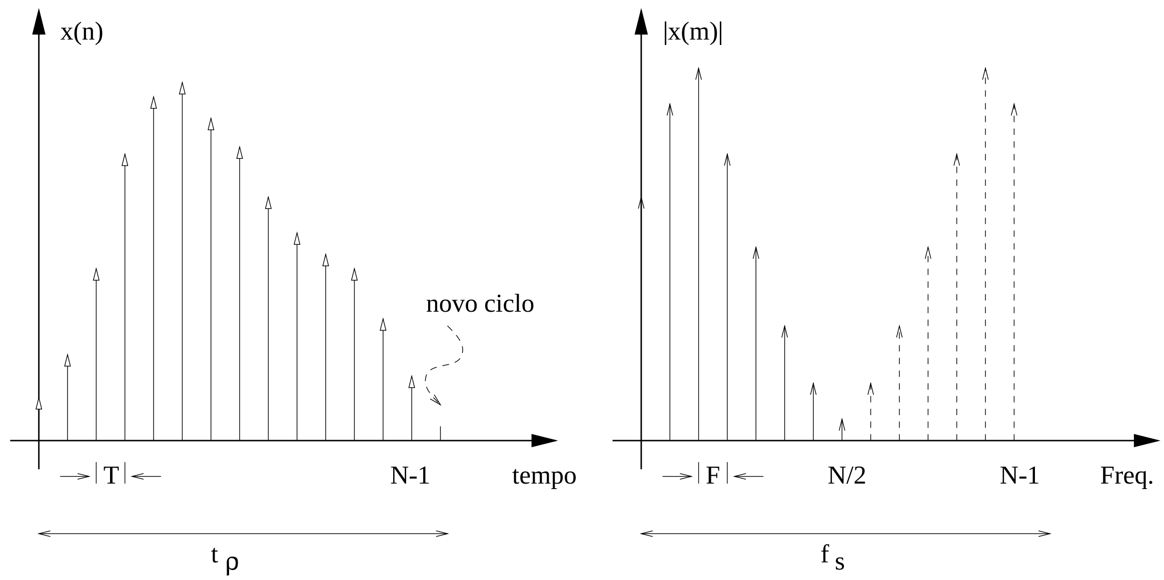

Figura 1.4: Amostragem em função do tempo e seu espectro (DFT). Modif. de Kuc [57].

1.3 Transformadas (Fughe)

1.3.1 Fourier

A essência do cálculo da Transformada Rápida de Fourier (FFT) [25] é uma série de operações matemáticas conhecidas como Transformada Discreta de Fourier (DFT[12]) que é um conjunto m de variáveis no domínio da freqüência a partir de um conjunto n de amostras no domínio do tempo. A figura 1.4 ilustra um sinal $x(n)$ com N amostras em intervalos de T segundos. Para n variando de 0 a $N-1$, a duração do sinal é $t_\rho = N \cdot T$.

A DFT de $x(n)$ é definida pela soma finita:

[12] *Discrete Fourier Transform*

$$X(m) = \frac{1}{N} \sum_{n=0}^{n-1} x(n) \cdot W^{n \cdot m} \tag{1.2}$$

onde:

$$W = e \cdot \frac{-i2\pi}{N} \tag{1.3}$$

A função $X(m)$ gera m variáveis no domínio da freqüência com incremento $F = 1/t_\rho$. Para $x(n)$ real com N amostras, um único espectro pode ser computado para $N/2$ pontos da freqüência. Na verdade, $X(m)$ é uma função periódica em m com N pontos em cada período, mas apenas $N/2$ são únicos.

Observando a definição da DFT, verifica-se que são necessárias cerca de N multiplicações e adições complexas para computar o espectro de cada m em particular e, se forem calculados $N/2$ componentes espectrais, o número de computações para o cálculo de todo o espectro é da ordem de N^2. Como foi demonstrado por Cooley e Tukey[25] pode-se calcular esta transformada com um número de processos computacionais da ordem de $N \cdot log_2(N)$, o que poupa um grande esforço computacional. Este método é conhecido como Transformada Rápida de Fourier (FFT[13]).

Os algoritmos de DFT funcionam melhor quando o número de pontos da amostra é uma potência inteira de 2, ou seja: $N = 2^k$, onde k é um inteiro positivo. Um programa que utiliza DFT com a finalidade de análise espectral, apresenta certas particularidades na relação entre a DFT e a transformada contínua de Fourier. Deve-se considerar que o

[13]Fast Fourier Transform

tratamento matemático considera o sinal como se ele fosse periódico, embora, na realidade, o sinal pode não ser periódico no sentido matemático estrito.

Um algoritmo para o cálculo da DFT deve levar em consideração alguns fatores básicos. Se tomarmos $N = 2^k$, para k inteiro positivo, amostras em um período, e se T é o incremento entre cada amostra, então o período $t_\rho = 2^k \cdot T$. O espectro obtido da DFT, também será periódico e conterá 2^k componentes espectrais. Entretanto, se a amostragem em função do tempo for real, pode ser demonstrado que metade dos componentes são coincidentes, logo apenas $N/2$ componentes espectrais complexos são significativos. Tais componentes são incrementados de $F = 1/t_\rho$; $m = 0$ corresponde à componente DC, $m = 1$ é a fundamental, $m = 2$ é o segundo harmônico, etc.. Para se evitar a sobreposição espectral, a taxa de amostragem deve ser maior ou igual ao dobro da maior freqüência do espectro, ou seja, se a maior freqüência for f_m, o máximo intervalo entre as amostras deve satisfazer:

$$T < \frac{1}{2 \cdot f_m}, \tag{1.4}$$

de acordo com o **Teorema de Niquist** [57].

1.3.2 Wavelets

A análise de dados de acordo com escalas variáveis no domínio do tempo e da freqüência é a idéia básica da utillização da teoria das *Wavelets*. Conforme cita A. Graps [47], alguns pesquisadores acham que a utilização de *Wavelets* abre uma nova perspectiva na análise de dados. De 15 anos para cá, a Teoria das *Wavelets* tem despertado enorme interesse em todas as áreas. Na realidade, a teoria matemática em que é baseada não é nova, remontando mais de 80 anos [27].

As *Wavelets* são funções matemáticas que ampliam intervalos de dados, separando-os em diferentes componentes de freqüência, permitindo a análise de cada componente em sua escala correspondente. Uma função *Wavelet* adequada a um conjunto de dados permite uma representação esparsa desse conjunto. Isso torna as *Wavelets* uma excelente ferramenta de compressão de dados. Outros campos de aplicação das *Wavelets*[29, 47] incluem astronomia, acústica, física nuclear, processamento de imagem e de som, música, fractais, turbulência, etc., o que tem despertado um grande interesse dos cientistas.

Os trabalhos de A. Grossman and J. Morlet [51], no final da década de 70, foram os responsáveis por esse interesse de engenheiros e matemáticos na construção de bases para análise clássica de harmônicos e de outros espaços de funções. Na busca de integrais quadráticas do tipo $(ax + b)$, chegaram ao problema de construção de uma base para $L^2(\mathbf{R})$[14] a partir de um conjunto discreto de $(ax + b)$.

Grossmann e Morlet demonstraram que se $\psi(t) \in \Re$, seus elementos podem ser usados como se fossem uma base ortogonal (cf. equação 1.5) , ou seja, cada sinal de energia finita pode ser representado como uma combinação linear de $\psi_{(a,b)}(t)$ e seus coeficientes são o produto escalar $\int_{-\infty}^{\infty} f(t)\psi_{(a,b)}(t) \cdot dt$. Assim, o produto escalar mede a flutuação do sinal $f(t)$, em torno do ponto b, na escala dada por $a > 0$.

[14] $L^2(R)$, chamado de espaço de Hilbert, é um conjunto das sucessões $x = (\lambda_k)$ de números complexos λ, tal que: $\lambda = (\alpha + i\beta), \qquad \alpha, \beta \in \Re$, de somas quadráticas absolutas, tal que: $\sum_1^{\infty} |\lambda_k| < \infty$.

$$\psi_{(a,b)}(t) = \frac{1}{\sqrt{a}} \cdot \psi\left(\frac{t-b}{a}\right), \qquad a > 0, \qquad b \in \Re. \tag{1.5}$$

Y. Meyer [68] encontrou uma função "suave" especial, que ele denominou *Wavelet*, cuja oscilação, acima e abaixo do eixo do x, tendia a zero no infinito, e cuja órbita discreta formava uma base de Hilbert para $L^2(\mathbf{R})$, e uma base incondicional para todos os espaços comuns de Banach [15] das funções em $\mathbf{R}$. P. G. Lemarié [58] usou isto para provar certos fatos básicos da álgebra dos operadores de Calderón-Zygmund.

Recentemente, I. Daubechies [30] encontrou um algoritmo para construir outras *Wavelets* para os espaços de funções particulares, incluindo *Wavelets* compactas ortonormais [16].

A vantagem das *Wavelets* sobre os métodos tradicionais de Fourier, é que a base das funções de Fourier são depentes da freqüência mas não do tempo, ou seja, pequenas alterações no domínio da freqüência produzem alterações em todo o domínio do tempo. As *Wavelets* são dependentes de ambos os domínios, da freqüência (via dilação)[17] e do tempo (via translação)[18] o que é uma vantagem em diversos casos. Por muitas décadas,

[15] conjunto das aplicações lieares contínuas em um espaço normado L^2, tal que:

$$\Lambda(L) = \mathbf{T} : L \to L.$$

[16] Base ortonormal é uma base ortogonal em que os vetores são unitários, tal que sua norma é unitária, ou seja:

$$\|a_i\| = 1 \qquad \text{ou} \qquad \sqrt{a_i \times a_i} = \|a_i\|$$

[17] Função que toma um vetor e retorna outro vetor, múltiplo do vetor dado: $\overrightarrow{v} \to \lambda\overrightarrow{v}$, onde λ é um escalar qualquer e $\overrightarrow{v}$ um vetor de qualquer espaço vetorial.

[18] Função que soma um vetor constante $\overrightarrow{k}$ a um vetor dado: $\overrightarrow{v} \to \overrightarrow{k} + \overrightarrow{v}$, onde $\overrightarrow{v}$ e $\overrightarrow{k}$ são vetores de qualquer espaço vetorial.

os cientistas procuraram funções mais apropriadas do que senos e cossenos para análise de sinais. As bases das funções de Fourier são impróprias para o tratamento local de dados, pois são séries infinitas e não se adaptam à análise de dados descontínuos. As *Wavelets* não só se prestam à aproximação de funções finitas, como também servem para análise de dados descontínuos.

As *Wavelets* satisfazem certas propriedades específicas. Uma delas é a propriedade de integrar a zero, ondulando acima e abaixo do eixo x, outras são propriedades técnicas que asseguram o cálculo rápido e fácil da transformada direta e inversa. Assim como senos e cossenos são bases na análise de Fourier, as *Wavelets* são usadas como bases na representação de outras funções [103].

Existem vários tipos de *Wavelets*, dentre as quais a mais simples é a *Wavelet de Haar*:

$$\psi_{jk}(x) = \mathrm{constante} \cdot \psi(2^{\mathrm{j}}\mathrm{x} - \mathrm{k}), \tag{1.6}$$

onde ψ define uma base ortogonal em $L^2(\mathbf{R})$, ou seja, cada elemento em $L^2(\mathbf{R})$ pode ser representado por uma combinação linear, que pode ser infinita, dessas bases. Gootman e Wickerhauser [46] apresentam condições necessárias e suficientes para que funções em $L^2(\mathbf{R})$ suportem transformadas compactas de Fourier para gerar bases ortogonais por meio de translações de inteiros e dilações quadráticas.

Uma vez estabelecida uma base *Wavelet* $\psi(x)$ — chamada *Wavelet mãe* — pode-se estabelecer bases de translação e dilação da *Wavelet mãe*, tal que: $\left\{\psi\left(\frac{x-b}{a}\right), (a,b) \in \Re^{+} \times \Re\right\}$, onde os valores de a e b devem ser selecionados convenientemente[19], na definição da base, tal que: $a = 2^{-j}$ e $b = k \cdot 2^{j}$, onde k e j são inteiros. A escolha de a e b proporcionará uma base esparsa e, além disso, possibilitará a análise com múltiplas resoluções no processamento de sinais. Assim, o grande mérito das *Wavelets* está no fato de que a análise no domínio do tempo é feita através de uma contração[20], enquanto a análise no domínio da freqüência se é feita por uma dilação[21] dos espaços da função.

Existem duas classes de *Wavelets*: "tempo-escala" (*time-scale Wavelets*) e "tempo-freqüência" (*time-frequency Wavelets*). As *Wavelets* "tempo-escala" são denominadas *Wavelets* de Grossmann-Morlet e são baseadas na análise de um sinal em uma vasta gama de escalas; as *Wavelets* "tempo-freqüência" são chamadas de *Wavelets* de Gabor e são fragmentos de um determinado sinal. Entretanto, para cada aplicação pode ser estabelecida uma *Wavelet mãe* apropriada, porque o sinal original — ou função — pode ser representado em termos de uma expansão *Wavelet* que usa os coeficientes em uma combinação das suas funções correspondentes.

Atualmente, os programas para a computação de transformadas *Wavelets* podem ser encontrados em C e FORTRAN[22].

[19] Note que se $(a, b) \in \Re^{+} \times \Re$, a é um valor positivo, enquanto b é um real qualquer, e o espaço é o produto cartesiano de a e b.

[20] Versão de alta freqüência da *Wavelet mãe*.

[21] Versão de baixa freqüência da *Wavelet mãe*.

[22] Essas rotinas foram escritas por Mac A. Cody (`http://www.dfw.net/ mcody`), e adaptadas para o *GNU C Compiler* (gcc).

1.3.3 Fourier *versus* Wavelets (Coda)

Basicamente, tanto a transformada discreta de Fourier (DFT), quanto a transformada discreta *Wavelet* (DWT) são operadores lineares em um vetor que contém $\log_2 n$ elementos e o transforma em um vetor $\{(1, 2^n)|n \in \Re\}$. Assim, as propriedades das matrizes envolvidas no cálculo das duas transformadas são semelhantes: tanto a matriz da transformada inversa da DFT como a da DWT, são as transpostas das originais.

Ambas as transformadas são rotações, no espaço de uma função, de um domínio para outro. A DFT usa funções senóides e cossenóides como bases e a DWT usa bases mais complexas chamadas *Wavelet mãe*, *Wavelets* ou *Wavelets* analíticas. Estas bases são localizadas no domínio da freqüência e podem ser analisadas através de espectros[23] e de escalogramas[24] que permitem isolar determinada freqüência e calcular distribuições de potência.

Entretanto, uma das diferenças principais entre a DFT e a DWT está no fato de que uma função *Wavelet* isolada é espacialmente localizada no domínio do tempo e da freqüência, enquanto as senóides e cossenóides de Fourier não o são. Essa característica confere às funções *Wavelets*, a propriedade de tornarem-se esparsas quando transformadas de um domínio para outro, possibilitando, desta forma, sua aplicação na remoção de ruídos de uma série no domínio do tempo [73].

Analisando a figura 1.5 nota-se a diferença entre as bases das funções de Fourier (A) e *Wavelets* (B) no plano tempo-freqüência. No caso da FFT a janela é quadrada e o

[23] descrição da potência em determinado intervalo de freqüência.
[24] plano de representação tempo-freqüência

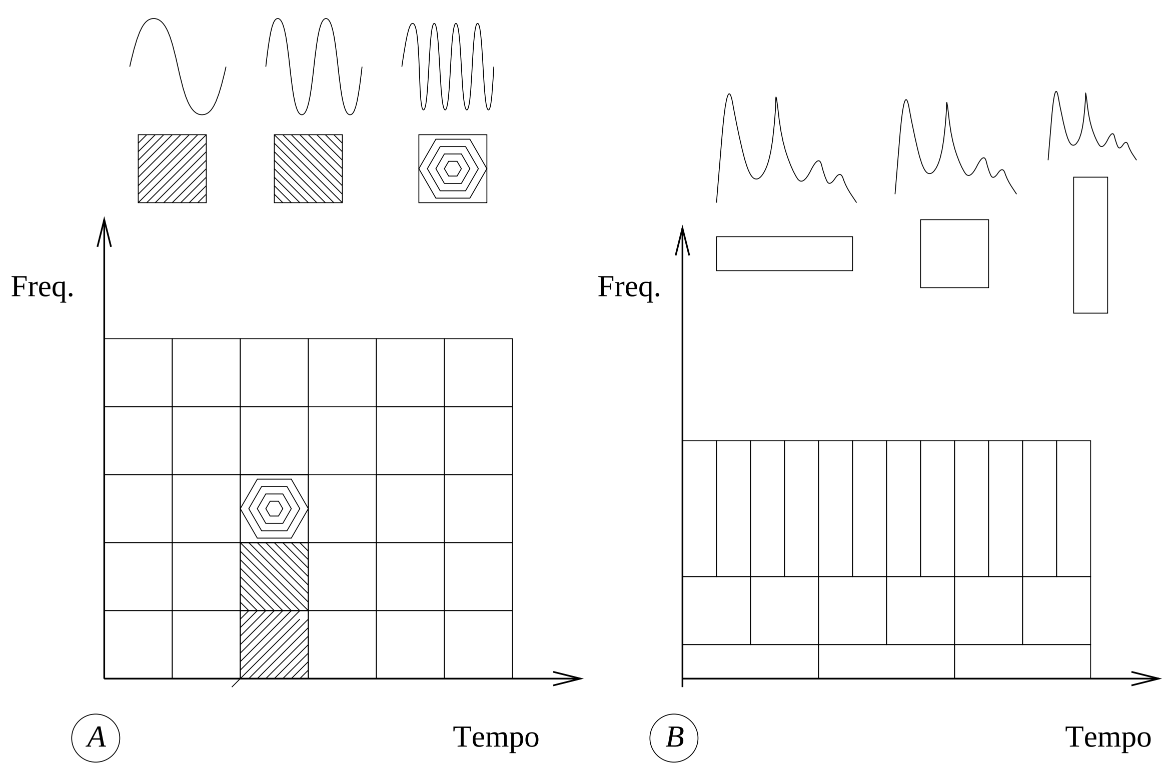

Figura 1.5: Diagrama do plano tempo-freqüência. (A) representação da base de Fourier. (B) representação de bases *Wavelet*.

truncamento da senóide ou cossenóide é feito para caber na janela. Como uma única janela é usada, a resolução é a mesma independentemente da localização no plano tempo-freqüência. A vantagem da transformada *Wavelet* é que a janela varia, podendo-se obter uma análise mais detalhada da freqüência ou evidenciar das descontinuidades do sinal.

Capítulo 2

Fundamentos Teóricos

Este capítulo apresenta algumas considerações matemáticas que fundamentam o modelo proposto. Em detrimento de uma compreensão mais clara e objetiva, evitou-se, tanto quanto possível, o refinamento matemático que pode ser encontrado nas referências citadas.

O que se pretende, neste trabalho é apresentar novos coeficientes, baseados em intervalos cromáticos de subdivisão da oitava[1], analisando os resultados através da combinação da transformada localizada de Fourier (Short Time Fourier Transform, ou STFT) na análise de sinais musicais e sua comparação com outros coeficientes de transformadas *Wavelet*.

Sabe-se que um sinal pode ser representado como uma série infinita de senos e cossenos, chamada expansão de Fourier. A desvantagem da expansão de Fourier, entretanto, está no fato de que existe apenas o espectro de freqüências sem qualquer informação de quando cada freqüência ocorre.

[1]Em música, a oitava é subdividida em 12 freqüências, á razão de $2^{1/12}$, que se denomina intervalo cromático.

As bases *Wavelets* podem ser utilizadas para a decomposição e reconstrução de um conjunto de dados. Assim como senos e cossenos são bases para a análise de *Fourier*, *Wavelets* podem ser usadas como bases de outras funções. Como já foi visto no capítulo 1, existem algumas diferenças fundamentais entre *Wavelets* e o sistema trigonométrico de senos e cossenos. Enquanto senos e cossenos são funções localizadas no domínio da freqüência mas não no domínio do tempo, as *Wavelets* são localizadas em ambos os domínios. Além disso, existe apenas um sistema trigonométrico, enquanto existem diversos conjuntos de bases *Wavelets* que podem ser gerados a partir de translação e dilação de uma determinada base específica denominada *Wavelet mãe*.

Embora a idéia principal de uma representação em ambos os domínios seja a possibilidade de analisar cada janela[2], com a finalidade de obter informações sobre os componentes de freqüência, quando um segmento do sinal é selecionado e transformado para o domínio da freqüência, os resultados não são coerentes com o sinal original. O sinal transformado pode ser bem localizado no tempo, mas as freqüências se encontram misturadas e acontece uma perda de resolução na sua reconstrução. Isto ocorre, porque a secção do sinal é, na realidade, uma convolução entre o sinal e a janela de secção e a solução do problema depende de como se faz a partição.

Valens [102], em sua tese, aponta uma solução baseada no princípio de incerteza de Heisenberg (cf. tb. [105]) que, em termos de processamento de sinal, declara que[3]:

[2]Neste trabalho, o termo "janela" (*window*) será utilizado para denotar um determinado segmento de um sinal, em qualquer domínio.

[3]"é impossível saber a freqüência exata e o tempo que ela ocorre em um sinal. Em outras palavras, um sinal pode, simplesmente, não ser representado como um ponto no espaço tempo-freqüência. O prinípio da incerteza mostra que é muito importante como o sinal é seccionado." (trad. do autor)

> "It is impossible to know the exact frequency and the exact time of occurrence of this frequency in a signal. In other words, a signal can simply not be represented as a point in time-frequency space. The uncertainty principle shows that it is very important how one cuts the signal"

A transformada *Wavelet* é, provavelmente, a mais recente solução para as limitações da transformada de Fourier, na qual o uso de um "janelamento" modular e escalonável resolve o problema da localização das áreas de interesse no sinal. A representação das funções transformadas *Wavelets* (escalogramas) pode ser comparada às representações das notas musicais em uma partitura [4], o que permite uma representação muito mais compacta quando comparada às transformadas trigonométricas (p. ex. Transformadas de *Fourier*). Assim, as *Wavelets* podem ser utilizadas com eficiência na análise, síntese e compressão de sinais, imagens e outras análises numéricas.

2.1 Fundamentos Matemáticos

A fundamentação matemática que se segue encontra-se descrita com maiores detalhes e profundidade por diversos autores [28, 34, 98, 101, 104]. Numa tentativa de tornar os conceitos matemáticos complexos que envolvem a teoria das *Wavelets* mais acessíveis, optou-se por um tratamento menos formal, incluindo-se alguns conceitos prévios que foram considerados necessários para a compreensão dos algoritmos descritos no capítulo 3. Tais conceitos já são axiomas das teorias matemáticas modernas.

[4] Ver capítulo 5 do livro de Meyer[69] sobre a interpretação de Pierre Boulez da Flauta Mágica de Mozart.

2.1.1 Espaços de Hilbert

O *Espaço de Hilbert* é definido como um subconjunto **H** de um conjunto de números complexos, que satisfaz as seguintes propriedades[50]:

1. **H** é um espaço vetorial, ou seja, os vetores em **H** podem ser somados ou multiplicados por escalares complexos conforme as regras usuais da aritmética de vetores.

2. **H** tem um *produto interno*, ou seja, quaisquer vetores x e y em **H** podem ser multiplicados de forma a produzir um escalar representado por $< x, y >$. O *produto interno* do vetor x por si mesmo ($< x, x >$) é um escalar real positivo, cuja raíz quadrada é a norma de x ($\|x\|$).

$$\sqrt{< x, x >} = \|x\| \tag{2.1}$$

3. Dados dois vetores x e y em **H**, a medida $\|x - y\|$ pertence ao espaço, o que faz com que **H** tenha uma estrutura espacial mensurável.

4. **H** tem uma medida espacial completa no que diz respeito à distância definida pela sua norma ($\|H\|$), ou seja, a distância entre os vetores x_n e x_m tende a zero, quando $n \to \infty$ e $m \to \infty$ e x_n tende a x em **H**. Em outras palavras, se $\|x_n - x_m\| \to 0$, então $\|x_n - x\| \to 0$ para qualquer x em **H**.

Dados dois vetores x e y em um *Espaço de Hilbert* **H**, diz-se que são ortogonais se seu *produto interno* é zero, ou seja,

$$< x, y >= 0, \tag{2.2}$$

o conjunto de vetores $\{x_i\}$, $i \in A, A = \{1, 2, 3, \ldots\}$, é dito que é *ortogonal* se quaisquer dois de seus elementos são ortogonais. Tal conjunto será *ortonormal* se seus vetores tiverem norma igual a 1.

Uma *base ortonormal* de **H** é um conjunto ortonormal e pode ser demonstrado que qualquer vetor em **H** pode ser representado como o limite das combinação linear da base ortonormal. O número de elementos da base se denomina a *dimensão* de **H**. No apêndice A.1 encontram-se alguns exmplos de espaços de Hilbert.

2.1.2 Bases de Hilbert e Séries de Fourier

Os coeficientes de *Fourier* de um vetor x em relação a um conjunto ortonormal $\{x_i\}$ é definido como:

$$\hat{x}(i) =< x, x_i > , \tag{2.3}$$

onde $\hat{x}$ é uma função complexa definida no conjunto de índice A:

$$\hat{x} : A \to C \tag{2.4}$$

A série:

$$\sum_i \hat{x}(i)x_i \tag{2.5}$$

é chamada de *Série de Fourier* de x. Se $\{x_i\}$ for uma base ortonormal, pode-se escrever:

$$x = \sum_i \hat{x}(i)x_i \tag{2.6}$$

Subespaços e Projeções Ortogonais

Um *subespaço fechado* de um espaço de Hilbert ($\mathbf{H}$) é um subconjunto $S \subset H$ que satisfaz as seguintes propriedades:

1. Se x e y forem dois vetores em S, e α e β forem escalares arbitrários, então:

$$\alpha x + \beta x \in S \tag{2.7}$$

2. Se $x_n \in S$ e o $\lim_{n\to\infty} x_n = x$ então $x \in S$.

Dado um vetor $x \in H$ e um subespaço fechado $S \subset H$, demonstra-se que existe um e apenas um vetor $p \in S$, próximo de x e um vetor $q = x - p$ tal que $q \perp S$, chamado de *projeção ortogonal* de x em S, e $x = p + q$, se $p \in S$ e $q \perp S$.

Se $\{x_i\}_{i\in A}$ é um conjunto ortonormal de vetores, S é um subspaço fechado em extensão a este conjunto, x é um vetor em $\mathbf{H}$ e p é a sua projeção ortogonal em S, então pode-se escrever:

$$p = \sum_i \hat{x}(i) x_i \tag{2.8}$$

Se $x \in S$ então $p = x$ e, ainda, se $\{x_i\}_{i \in A}$ é uma base ortonormal, então $x = p$ para qualquer $x \in H$, logo:

$$x = \sum_i \hat{x}(i) x_i \tag{2.9}$$

Entretanto, como $\{x_i\}_{i \in A}$ não é o máximo conjunto de vetores ortonormais, a equação acima só é válida para os vetores $x \in S$.

Propriedades

Seja $\{u_\alpha\}$ uma base ortogonal, então:

1. Qualquer vetor x em $\mathbf{H}$ pode ser representado como uma série de *Fourier*. (cf. equação 2.9)

2. Dados dois vetores x e y em $\mathbf{H}$, pode-se escrever:

$$< x, y >= \sum_i \hat{x}(i) \overline{\hat{y}(i)} \tag{2.10}$$

e também,

$$\|x\|^2 = \sum_i |\hat{x}(i)|^2 \tag{2.11}$$

3. Se $x = \sum_i c_i x_i$ para alguma seqüência de números complexos c_i então $c_i = \hat{x}(i)$.

Se $L^2(A)$ é um conjunto complexo de elementos, cuja soma quadrática é finita (cf. apêndice A.1), a correspondência $x \longmapsto \hat{x}$ é biunívoca, é um operador de $\mathbf{H}$ em $L^2(A)$ e preserva o produto interno.

2.1.3 Bases Canônicas

Seja

$$
\begin{aligned}
e_1 &= (1, 0, 0, \ldots, 0) \\
e_2 &= (0, 1, 0, \ldots, 0) \\
e_3 &= (0, 0, 1, \ldots, 0) \\
&\vdots \\
e_n &= (0, 0, 0, \ldots, 1),
\end{aligned}
$$

o conjunto $\{e_k\}$, onde $1 \leq k \leq n$, forma uma base ortogonal $e = (e_1, e_2, \ldots, e_n)$ chamada *base canônica* (cf. apêndice A.1).

No caso da série de *Fourier*, os coeficientes do vetor $x = [x_1, x_2, x_3, \cdots, x_n]$ são seus componentes, ou seja:

$$
\begin{array}{rcccl}
\hat{x}(1) & = & < x, e_1 > & = & x_1 \\
\hat{x}(2) & = & < x, e_2 > & = & x_2 \\
\hat{x}(3) & = & < x, e_3 > & = & x_3 \\
 & \vdots & & \vdots & \\
\hat{x}(n) & = & < x, e_n > & = & x_n
\end{array}
$$

2.2 Transformada Discreta de Fourier

Seja $\omega = e^{-i2\pi/n}$ e k um inteiro tal que $0 \leq k \leq n-1$ e

$$
u_k = \frac{1}{\sqrt{n}}\left(1, \omega^k, \omega^{2k}, \cdots, \omega^{(n-1)k}\right), \tag{2.12}
$$

sendo$\{u_k\}$ é uma base ortogonal tem-se que:

$$
\hat{x}(k) =< x, u_k >= \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \omega^{-jk} \tag{2.13}
$$

e $\hat{x}$ é chamada *transformada discreta de Fourier*, onde x é dado pela equação inversa:

$$
x = \sum_{j=0}^{n-1} \hat{x}(j) u_j \tag{2.14}
$$

que pode ser escrita como:

$$x_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{j=n-1} x_j \omega^{jk} \tag{2.15}$$

2.2.1 A série de Fourier Clássica

Seja:

$$e_k(t) = e^{ikt} \tag{2.16}$$

para cada k inteiro (positivo ou negativo), e_k constitui um conjunto ortonormal em $\mathbf{L}^2\,[0, 2\pi]$ (cf. apêndice A.1).

Além disso, pode ser demonstrado que este conjunto forma uma base trigonométrica ortonormal, cujos coeficientes da função $f \in L^2[0, 2\pi]$ são dados por:

$$\hat{f}(k) =< f, e_k >= \frac{1}{2\pi} \int_0^{2\pi} f(t) e^{-ikt} dt \tag{2.17}$$

A série de Fourier de f é dada, então, por:

$$f(t) = \sum_{-\infty}^{+\infty} \hat{f}(k) e^{-ikt} \tag{2.18}$$

cuja convergência é o limite da norma de $\mathbf{L}^2$, ou:

$$\lim_{n \to \infty} = |f - \sum_{-n}^{+n} \hat{f}(k) e_k\|_2 = 0 \tag{2.19}$$

Esta é a série clássica de Fourier para uma função periódica em 2π.

2.2.2 A Transformada Clássica de Fourier

A transformada clássica de uma função f não periódica é definida como:

$$\hat{f}(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(t)e^{-ikt}dt \tag{2.20}$$

onde k é uma variável real contínua. A Transformada Inversa é dada por:

$$f(t) = \sqrt{2\pi} \int_{-\infty}^{+\infty} \hat{f}(k)e^{ikt}dk \tag{2.21}$$

A transformada de Fourier é normalmente definida para o espaço $\mathbf{L}^1[-\infty, +\infty]$, ou seja, é requisito básico que:

$$\int_{-\infty}^{+\infty} |f(t)|dt < \infty \tag{2.22}$$

Isto garante que $\hat{f}(k)$ satisfaça as seguintes propriedades:

1. a função $\hat{f}(k)$ é definida para todas as freqüências k;
2. $\hat{f}(k)$ é uma função contínua de k;
3. $\hat{f}(k)$ tende a zero quando $k \to \pm\infty$;

Entretanto, demonstra-se que a Transformada de Fourier pode ser aplicada ao espaço $\mathbf{L}^2[-\infty, +\infty]$, ou seja, sinais analógicos de energia finita. A única diferença é que, neste caso, a transformada pode não ser contínua ou pode não tender a zero para as freqüências próximas a $\pm\infty$, ou ainda podem não ser definidas para valores isolados de k em freqüências zero[50].

A Transformada de Fourier no espaço $\mathbf{L}^2[-\infty, +\infty]$ deve satisfazer as seguintes propriedades[5]:

1. É uma transformação unívoca de $\mathbf{L}^2[-\infty, +\infty]$ em si mesma;

2. Preserva as normas da função, ou seja:

$$\int_{-\infty}^{+\infty} |f(t)|^2 dt = \int_{-\infty}^{+\infty} |\hat{f}(k)|^2 dk \tag{2.23}$$

3. Preserva os produto interno[6], ou seja:

$$\int_{-\infty}^{+\infty} f(t)\overline{g(t)} dt = \int_{-\infty}^{+\infty} \hat{f}(k)\overline{\hat{g}(k)} dk. \tag{2.24}$$

2.3 Wavelets

2.3.1 A *Wavelet* de Haar

A *Wavelet* com os mais simples coeficientes no espaço $\mathbf{L}^2[-\infty, +\infty]$ é denominada *Wavelet* de Haar [28] que pode ser definida como uma função ψ onde:

$$\psi_k^j(t) = 2^{j/2}\psi(2^j t - k) \tag{2.25}$$

[5]**Nota:** A Transformada de Fourier em $\mathbf{L}^2[-\infty, +\infty]$ não pode ser considerada como uma base ortonormal, porque as funções e^{ikt} não constituem uma base ortonormal já que não possuem norma finita em $\mathbf{L}^2$ e também em $\mathbf{L}^2[-\infty, +\infty]$.

[6]Esta última propriedade chama-se **Identidade de Parseval**.

Wavelets **Ortonormais**

Dada uma função $\psi \in \mathbf{L}^2[-\infty, +\infty]$, é dita que é ortonormal se as funções ψ_k^j formarem uma base ortonormal[24]. Os índices j e k desta base s˜o inteiros representados pelo conjunto $\mathbf{Z}\times\mathbf{Z}$. O fator $2^{j/2}$ garante que $\|\psi_k^j\|^2 = \|\psi\|^2$ e, portanto, é requisito que $\|\psi\|^2 = 1$.

Dada uma função $f \in \mathbf{L}^2[-\infty, +\infty]$ os *coeficientes wavelets* de f são seus coeficientes de Fourier na base ψ_k^j, ou seja, os escalares

$$c_k^j =< f, \psi_k^j >= \int_{-\infty}^{+\infty} f(t)\overline{\psi_k^j(t)}dt \tag{2.26}$$

e, portanto,

$$f(t) = \sum_{(j,k)\in Z\times Z} c_k^j \psi_k^j(t). \tag{2.27}$$

Esta série é a *série wavelet* de f.

Wavelet de Haar

A *Wavelet* de Haar é a função

$$\psi(t) = \begin{cases} +1 & 0 \le t < \frac{1}{2} \\ -1 & \frac{1}{2} \le t < 1 \end{cases}$$

A função

$$\phi(t) = \begin{cases} 1 & 0 \leq t < 1 \\ 0 & t < 0 \qquad t \geq 1 \end{cases}$$

é chamada de *função escalonante* de $\psi(t)$ e pode-se observar que:

$$\phi(t) = \phi(2t) + \phi(2t-1) = \frac{1}{\sqrt{2}}\phi_0^1(t) + \frac{1}{\sqrt{2}}\phi_1^1(t) \tag{2.28}$$

e

$$\psi(t) = \phi(2t) - \phi(2t-1) = \frac{1}{\sqrt{2}}\phi_0^1(t) - \frac{1}{\sqrt{2}}\phi_1^1(t). \tag{2.29}$$

A equação 2.28 é a *equação de dilação* do sistema de Haar e os coeficientes h_k do **filtro wavelet** são:

$$h_0 = \frac{1}{\sqrt{2}} \qquad\qquad h_1 = \frac{1}{\sqrt{2}}.$$

Para provar que as funções ψ_k^j formam uma base ortonormal, tem-se que demonstrar que:

1. São mutualmente ortogonais, ou seja:

$$\int_{-\infty}^{+\infty} \psi_k^j(t)\psi_m^l(t)dt = 0 \tag{2.30}$$

sempre que $(j,k) \neq (l,m)$.

2. Que têm norma unitária, ou seja, para qualquer índices (j,k)

$$\int_{-\infty}^{+\infty} |\psi_k^j(t)|^2 dt = 1. \tag{2.31}$$

3. Que qualquer função $f \in \mathbf{L}^2[-\infty,+\infty]$ pode ser representada pela série wavelet, ou seja:

$$f(t) = \sum_{(j,k)\in Z\times Z} c_k^j \psi_k^j(t). \tag{2.32}$$

Os itens (1) e (2) podem ser facilmente verificados e o item 3 pode ser demonstrando pelo seguinte teorema[7]:

Teorema 2.3.1 *Qualquer função $f \in \mathbf{L}^2[-\infty,+\infty]$ pode ser representada pela wavelet de Haar (cf. equação 2.32).*

Corolário 2.3.1 *Seja a função f uma função característica no intervalo diádico*

$$f(t) = \begin{cases} 1 & 0 \leq t < 1 \\ 0 & t < 0 \qquad t \geq 1 \end{cases},$$

pode-se escrever

$$f(t) = \frac{1}{2}f(t/2) + \frac{1}{2}\psi(t/2) = \frac{1}{2}f(t/2) + \frac{1}{\sqrt{2}}\psi_0^{-1}(t). \tag{2.33}$$

[7] O detalhamento da demonstração desse teorema se encontra no livro de Ingrid Daubechies [28]

Fazendo-se a substituição de t por $t/2$

$$f(t/2) = \frac{1}{2}f(t/4) + \frac{1}{2}\psi(t/4) = \frac{1}{2}f(t/4) + \psi_0^{-2}(t) \tag{2.34}$$

e substituindo-se na expressão anterior, tem-se:

$$f(t) = \frac{1}{4}f(t/4) + (\frac{1}{\sqrt{2}})^2\psi_0^{-2} + (\frac{1}{\sqrt{2}})\psi_0^{-1}. \tag{2.35}$$

Repedindo-se a substituição até o n-ésimo fator, tem-se:

$$f(t) = (\frac{1}{2})^n f(t/2^n) + (\frac{1}{\sqrt{2}})^n\psi_0^{-n} + \cdots + (\frac{1}{\sqrt{2}})\psi_0^{-1}. \tag{2.36}$$

Corolário 2.3.2 *Seja o erro $\varepsilon_n(t) = (\frac{1}{2})^n f(t/2^n)$ observa-se que este torna-se muito pequeno conforme n aumenta, ou seja:*

$$\|\varepsilon_n\|_2 = 2^{-n/2} \longmapsto 0 \qquad \text{e, portanto,}$$

$$f(t) = \sum_{n=1}^{\infty}(\frac{1}{\sqrt{2}})^n\psi_0^{-n}(t) \tag{2.37}$$

Corolário 2.3.3 *Seja $f \in \mathbf{L}^2[-\infty, +\infty]$ tal que seu valor seja 0, fora do intervalo dádico $[0, 1]$, f pode ser aproximada a menos de um erro, conforme*

$$f(t) = \sum_{k=0}^{2^n-1} s_k\phi_k^n(t) + \varepsilon'_n(t) \tag{2.38}$$

onde $\|\varepsilon'_n\|_2 \longmapsto 0$, *quando* $n \longmapsto \infty$.

Demonstração 2.3.1 *A função* ϕ, *bem como as funções* ϕ_k^j *podem ser expressas como somas de séries de Haar, portanto* f *pode ser aproximada, a menos de um erro, por intermédio de combinações lineares de funções de Haar. Isto implica que* f *pode ser expressa através de séries de Haar.*

Seja $f \in \mathbf{L}^2[-\infty, +\infty]$ *arbitrária, pode-se aproximá-la por meio de outra função que tome valores zero fora do mesmo intervalo diádico, ou seja:*

$$f(t) = f_n(t) + \varepsilon''_n(t) \tag{2.39}$$

onde $\|\varepsilon''_n\|_2 \longmapsto 0$ *quando* $n \longmapsto \infty$.

Assim, sendo que f_n *pode ser expressa como série de Haar, segue que o mesmo tem que ser verdadeiro para* f, *completando a demonstração do teorema 2.3.1.*

Cálculo dos Coeficientes de Haar

Seja ψ a transformada *Wavelet* de Haar e ϕ a função escalonante, nota-se que:

$$\phi_k^j(t) = \frac{1}{\sqrt{2}}\phi_{2k}^{j+1}(t) + \frac{1}{\sqrt{2}}\phi_{2k+1}^{j+1}(t) \tag{2.40}$$

e

$$\psi_k^j(t) = \frac{1}{\sqrt{2}}\phi_{2k}^{j+1}(t) - \frac{1}{\sqrt{2}}\phi_{2k+1}^{j+1}(t) \tag{2.41}$$

Dada uma função $f \in \mathbf{L}^2[-\infty, +\infty]$ pode-se calcular seus coeficientes de Haar c_k^j, desde que, para um inteiro n, sejam dados valores amostrados s_k, ou seja, "médias" de f

$$s_k^n = 2^{n/2} \int_{k/2^n}^{(k+1)/2^n} f(t)dt = \int_{-\infty}^{+\infty} f(t)\phi_k^n(t)dt, \tag{2.42}$$

onde $k = 0, \pm 1, \pm 2, \ldots$. Se definirmos valores de $j < n$ para todos os inteiros k, podemos calcular

$$s_k^j = \int_{-\infty}^{+\infty} f(t)\phi_k^n(t)dt \qquad \text{e} \tag{2.43}$$

$$c_k^j = \int_{-\infty}^{+\infty} f(t)\psi_k^n(t)dt. \tag{2.44}$$

Pelas equações 2.40 e 2.41, é possível escrever:

$$s_k^j = \frac{1}{\sqrt{2}}(s_{2k}^{j+1} + s_{2k+1}^{j+1}) \qquad \text{e} \qquad \mathrm{c_k^j} = \frac{1}{\sqrt{2}}(\mathrm{s_{2k}^{j+1} - s_{2k+1}^{j+1}}) \tag{2.45}$$

Assim, por meio de várias iterações, a partir de um conjunto de amostras

$\ldots, s_{-1}^n, s_0^n, s_1^n, \ldots$ calcula-se

$\ldots, s_{-1}^{n-1}, s_0^{n-1}, s_1^{n-1}, \ldots$

e $\ldots, c_{-1}^{n-1}, c_0^{n-1}, c_1^{n-1}, \ldots,$

então, calcula-se

$\ldots, s_{-1}^{n-2}, s_0^{n-2}, s_1^{n-2}, \ldots$ e $\ldots, c_{-1}^{n-2}, c_0^{n-2}, c_1^{n-2}, \ldots,$

e assim, sucessivamente, até que se tenha atingido o número desejado de coeficientes Haar a ser calculado [33].

Este algoritmo é denominado *algoritmo em pirâmide* [14] e pode ser representado esquematicamente por:

$$\begin{array}{ccccccccc} s_k^n \rightarrow & s_k^{n-1} \rightarrow & s_k^{n-2} \rightarrow & s_k^{n-3} \rightarrow & \cdots \\ & \searrow c_k^{n-1} & \searrow c_k^{n-2} & \searrow c_k^{n-3} & \searrow \cdots \end{array}$$

2.3.2 Análise de Multiresolução

A *análise de multiresolução* [28] via uma função escalonante ϕ é uma seqüência crescente de subespaços $\mathbf{V}_j$ de $\mathbf{L}^2[-\infty,+\infty]$ próximos, tal que:

$$\cdots V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots$$

que satisfaz as seguintes propriedades:

1. *Densidade*, ou seja, $\cup_j V_j$ é densa em $\mathbf{L}^2[-\infty,+\infty]$.

2. *Separação*, ou seja, $\cap_j V_j = 0$.

3. *Escalabilidade*, ou seja, $f(t) \in V_j \Longleftrightarrow f(2t) \in V_{j+1}$ para todos os inteiros j e funções f arbitrárias.

4. *Ortonormalidade*, ou seja, as funções $\phi(t-k)$, para $k = 0, \pm 1, \pm 2, \ldots$ forma uma base ortonormal de V_0.

A análise de multiresolução $\{V_j\}$ é dita gerada pela função escalonante ϕ, no sentido de que para cada j o subespaço V_j é gerado por funções ϕ_k^j, para $k = 0, \pm1, \pm2, \ldots$.

Das propriedades 3 e 4 segue que $\{\phi_k^j\}$ é base ortonormal de V_j para cada $j = 0, \pm1, \pm2, \ldots$. E, considerando que $\phi \in V_0 \subset V_1$, pode-se escrever:

$$\phi(t) = \sum_{-\infty}^{+\infty} h_k \phi_k^1(t) \qquad \text{ou} \tag{2.46}$$

$$\phi(t) = \sqrt{2} \sum_{-\infty}^{+\infty} h_k \phi(2t - k). \tag{2.47}$$

A equação 2.47 é chamada de *Equação de Dilação*, e os coeficientes h_k são os *coeficientes do filtro wavelet* da análise de multiresolução[8].

A transformada de Fourier da *Equação de Dilação* pode ser escrita como:

$$\hat{\phi}(\xi) = \left(\frac{1}{\sqrt{2}} \sum_{-\infty}^{+\infty} h_k e^{-ik\xi/2} \right) \hat{\phi}\left(\frac{\xi}{2}\right). \tag{2.48}$$

Coeficientes Wavelets

Os coeficientes de filtros wavelets devem satisfazer a equação

$$\sum_{-\infty}^{+\infty} |h_k|^2 = 1, \tag{2.49}$$

posto que $\|\phi\| = 1$ [69]. onde h_k pode ser escrito como

[8] A caracterização das funções ϕ (*Funções Escalonantes*) é complicada, enquanto a caracterização das seqüências de números complexos que formam os filtros *Wavelets* é mais fácil. Uma vez conhecidos os coeficientes *Wavelets* a função escalonante ϕ pode ser construída por meio da equação 2.47.

$$h_k = \int_{-\infty}^{+\infty} \phi(t)\overline{\phi_k^1(t)}dt \tag{2.50}$$
$$= \sqrt{2}\int_{-\infty}^{+\infty} \phi(t)\overline{\phi(2t-k)}dt \tag{2.51}$$

A ortogonalidade de ϕ, ou seja,

$$\int_{-\infty}^{+\infty} \phi(t)\overline{\phi(t-k)}dt = \delta(k,0), \tag{2.52}$$

onde δ é o símbolo de Kronecker[9], pode ser expressa em termos da equação de dilação.

$$\sum_m \sum_n h_m \overline{h_n} 2 \int_{-\infty}^{+\infty} \phi(2t-m)\overline{\phi(2t-2k-n)}dt = \delta(k,0)c. \tag{2.53}$$

A equação 2.53 pode ser escrita em termos da $\sum_m$ como:

$$\sum_m h_m \overline{h_{m+2k}} = \delta(k,0), \tag{2.54}$$

onde k é um inteiro arbitrário[10].

Se assumirmos que ϕ é integrável e

$$I = \int_{-\infty}^{+\infty} \phi(t)dt \neq 0 \tag{2.55}$$

pode-se integrar ambos os lados da equação de dilação, logo

[9] Uma função é dita possuir um δ de Kronecker quando assume apenas os valores 0 e 1.
[10] Observe que $\sum_m |h_m|^2 = 1$ é um caso especial onde $k = 0$.

$$\begin{aligned} \int_{-\infty}^{+\infty} \phi(t)dt &= \sqrt{2}\sum_{-\infty}^{+\infty} h_k \int_{-\infty}^{+\infty} \phi(2t-k)dt \\ &= \frac{1}{\sqrt{2}}\sum_{-\infty}^{+\infty} h_k \int_{-\infty}^{+\infty} \phi(s)ds \end{aligned}. \tag{2.56}$$

Assim, igualando-se o primeiro e terceiro termos da equação 2.56 e dividindo por I (cf. equação 2.55), temos:

$$\sum_{-\infty}^{+\infty} h_k = \sqrt{2}. \tag{2.57}$$

Nota 2.3.1 *É interessante observar que a soma dos coeficientes Wavelets corresponde à metade do conjunto de intervalos da oitava, correspondendo exatamente ao intervalo de quarta aumentada, ou sexto intervalo cromático a partir de uma freqüência fundamental ($2^{6/12} = \sqrt{2}$).*

Propriedades das funções escalonantes

A função ϕ que resolve a equação de dilação pode ser escrita como:

$$\phi(t) = \sqrt{2}\sum_{-\infty}^{+\infty} h(k)\phi(2t-k) \tag{2.58}$$

que no "espaço da freqüência[11]" é:

$$\hat{\phi}(\xi) = \left(\frac{1}{\sqrt{2}}\sum_{-\infty}^{+\infty} h(k)e^{-ik\xi/2}\right)\hat{\phi}(\xi/2). \tag{2.59}$$

Pode-se descrever uma função $P(\xi)$ tal que:

[11] Note que não se pode falar em domínio da freqüência neste caso.

$$P(\xi) = \frac{1}{\sqrt{2}} \sum_{-\infty}^{+\infty} h(k) e^{-ik\xi} \tag{2.60}$$

que em uma função periódica em 2π é chamada de símbolo do filtro $\{h_k\}$ e pode-se, então, escrever a equação 2.59 como:

$$\hat{\phi}(\xi) = P(\xi/2)\hat{\phi}(\xi/2), \tag{2.61}$$

de onde pode-se escrever:

$$\begin{aligned}
\hat{\phi}(\xi/2) &= P(\xi/4)\hat{\phi}(\xi/4) \\
\hat{\phi}(\xi/4) &= P(\xi/8)\hat{\phi}(\xi/8) \\
\vdots \;\; &\vdots \;\; \vdots \\
\hat{\phi}(\xi/2^{n-1}) &= P(\xi/2^n)\hat{\phi}(\xi/2^n)
\end{aligned}$$

e, portanto:

$$\hat{\phi}(\xi) = \hat{\phi}(\xi/2^n) \prod_{k=1}^{n} P(\xi/2^k). \tag{2.62}$$

Podemos, agora, enunciar dois corolários:

Corolário 2.3.4 *A função escalonante ϕ é integrável.*

Corolário 2.3.5 *A função escalonante $\phi \in \Re$*

Como conseqüência do corolário 2.3.4 pode-se dizer que $\hat{\phi}$ é contínua e que:

$$\int_{-\infty}^{+\infty} \phi(t)dt = \sqrt{2\pi}\hat{\phi}(0) \neq 0.$$

Conforme Daubechies [28], se a integral de $\phi(t)dt$ é diferente de zero, implica que $P(0) = 1$ e, por causa da continuidade de $\hat{\phi}$, pode-se limitar a equação 2.62 a:

$$\hat{\phi}(\xi) = \hat{\phi}(0) \prod_{k=1}^{\infty} P(\xi/2^k). \tag{2.63}$$

O coroário 2.3.5 pode ser usado para demonstrar que $\hat{\phi}(0) = \pm\frac{1}{\sqrt{2}}$ e, generalizando, $\hat{\phi}(0) > 0$.

Se é possível expressar a ortogonalidade da função escalonante $\phi(t)$ por $\phi(t-k)$ no "espaço da freqüência", pode-se escrever (usando-se a identidade de Parseval):

$$\begin{aligned} \delta(k,0) &= \int_{-\infty}^{+\infty} \phi(t)\overline{\phi(t-k)}dt \\ &= \int_{-\infty}^{+\infty} \hat{\phi}(\xi)e^{ik\xi}\overline{\hat{\phi}(\xi)}d\xi \\ &= \int_{-\infty}^{+\infty} |\hat{\phi}(\xi)|^2 e^{ik\xi}d\xi \end{aligned}$$

Decompondo-se a última integral numa soma de integrais no intervalo $[0, 2\pi]$ e mudando-se as variáveis, pode-se escrever:

$$\begin{aligned}
\delta(k,0) &= \sum_m \int_{2\pi m}^{2\pi(m+1)} |\hat{\phi}(\xi)|^2 e^{ik\xi} d\xi \\
&= \sum_m \int_0^{2\pi} |\hat{\phi}(\mu + 2\pi m)|^2 e^{ik\mu} d\mu \\
&= \int_0^{2\pi} \left(\sum_m |\hat{\phi}(\mu + 2\pi m)|^2 e^{ik\mu} \right) d\mu
\end{aligned}$$

onde $\sum_m |\hat{\phi}(\mu + 2\pi m)|^2$ tem o mesmo coeficiente de Fourier que $1/2\pi$, logo:

$$\sum_m |\hat{\phi}(\mu + 2\pi m)|^2 = \frac{1}{2\pi} \tag{2.64}$$

A soma na equação 2.64 pode ser decomposta em duas somas, uma sobre os índices ímpares e outra sobre os índices pares.

$$\sum_{\text{pares}} |\hat{\phi}(\mu + 2\pi m)|^2 + \sum_{\text{impares}} |\hat{\phi}(\mu + 2\pi m)|^2 = \frac{1}{2\pi} \tag{2.65}$$

Substituindo a equação 2.59, usando $\eta = \mu/2$ e usando a periodicidade de P, tem-se:

$$\sum_n |P(\eta)|^2 |\hat{\phi}(\eta + 2\pi n)|^2 + \sum_n |P(\eta + \pi)|^2 |\hat{\phi}(\eta + \pi + 2\pi n)|^2 = \frac{1}{2\pi}.$$

Tomando-se os fatores comuns e usando a equação 2.64, pode-se escrever:

$$|P(\eta)|^2 \sum_n |\hat{\phi}(\eta + 2\pi n)|^2 + |P(\eta + \pi)|^2 \sum_n |\hat{\phi}(\eta + \pi + 2\pi n)|^2$$

$$
\begin{aligned}
&= |P(\eta)|^2 \left(\frac{1}{2\pi}\right) + |P(\eta+\pi)|^2 \left(\frac{1}{2\pi}\right) \\
&= \left(\frac{1}{2\pi}\right)
\end{aligned}
$$

e, portanto,

$$|P(\eta)|^2 + |P(\eta+\pi)|^2 = 1, \tag{2.66}$$

onde $|P(0)|^2 + |P(\pi)|^2 = 1$ é um caso especial da equação 2.66.

Considerando-se que $P(0) = 1$ implica que $P(\pi) = 0$ e, portanto, da equação 2.62 tem-se que:

$$
\begin{aligned}
\hat{\phi}(2\pi) &= \hat{\phi}(0)P(\pi)P(\pi/2)\cdots = 0 \\
\hat{\phi}(4\pi) &= \hat{\phi}(0)P(2\pi)P(\pi)\cdots = 0 \\
\vdots \quad & \quad \vdots
\end{aligned}
$$

onde $\hat{\phi}(2\pi m) = 0$ para todos os inteiros m. A equação 2.64 implica que $\hat{\phi}(0) = 1/\sqrt{2\pi}$, que substituído na equação 2.62 produz:

$$\hat{\phi}(\xi) = \left(\frac{1}{\sqrt{(2\pi)}}\right) \prod_{k=1}^{\infty} P(\xi/2^k). \tag{2.67}$$

2.4 Teorema Fundamental

Suponha que $\{V_j\}$ seja um vetor de multiresolução com uma função escalonante ϕ e seja W_j o complemento ortogonal de V_j em V_{j+1} que pode ser escrito como:

$$V_{j+1} = V_j \bigoplus W_j.$$

A definição de análise de multiresolução implica que $V_{j+1} = \oplus_{-\infty}^{j} W_l$ e $\mathbf{L}^2[-\infty, +\infty] = \oplus_{-\infty}^{+\infty} W_j$, onde todos os subespaços W_j são mutualmente ortogonais. Os espaços V_j e W_j são chamados de **direção** e **detalhe** da análise de multiresolução, respectivamente. Assim, pode-se dizer que existe uma *Wavelet* ortonormal ψ tal que para cada inteiro j, a função $\left\{\psi_k^j\right\}$ forma uma base ortonormal W_j, o que pode ser demonstrado.

Teorema 2.4.1 *Dada uma análise de multiresolução $\{V_j\}$ com uma função escalonante ϕ, existe uma base ortonormal ψ que gera o detalhe da análise de multiresolução. Se a equação de dilação for*

$$\phi(t) = \sum_{-\infty}^{+\infty} h_k \sqrt{2} \phi(2t - k)$$

então a função ψ pode ser escrita como:

$$\psi(t) = \sum_{-\infty}^{+\infty} (-1)^k \overline{h_{1-k}} \sqrt{2} \phi(2t - k).$$

Prova 2.4.1 *Para qualquer inteiro j é possível demonstrar que $\left\{\psi_k^j\right\}_{k \in Z}$ constitui uma* *base ortonormal de W_j, isto prova o Teorema 2.4.1.*

Para a demonstração do teorema 2.4.1 veja apêndice A.2.

2.5 Suporte Compacto

Diz-se que uma função $\phi(t)$ tem suporte compacto no intervalo $[0, N]$, quando $\phi(2t-k)$ tem suporte compacto em $[k/2, N/2 + k/2]$. Assim, existem finitos k onde a intersecção destes dois intervalos não é vazia e, desde que

$$h_k = \sqrt{2}\int_0^N \phi(t)\phi(2t-k)dt,$$

é possível dizer que tem-se um número finito de coeficientes $h_k \neq 0$. Por meio do teorema de Paley-Wiener [28] prova-se duas importantes propriedades:

Suporte da função escalonante

Teorema 2.5.1 *Uma função escalonante ϕ tem suporte compacto em $[0, N]$ se, e somente se, $h_k = 0$ para $k < 0$ e $k > N$.*

Prova 2.5.1 *A prova deste teorema encontra-se na página 176 de "Ten Lectures on Wavelets" de Ingrid Daubechies [28].*

Propriedade da aproximação

Seja p um inteiro, diz-se que um filtro wavelet h_k satisfaz uma condição de aproximação $A(p)$, de ordem p em π, se $P(\xi) = 0$ e $P^{(m)}(\pi) = 0$ para $m = 0, 1, \cdots, p-1$, onde P^m é a m-ésima derivada de P (cf. equação 2.60). Assim, pode-se escrever que:

$$\sum_{k=0}^{N} (-1)^k k^m h_k = 0. \tag{2.68}$$

Capítulo 3

Material e Método

3.1 Material

Os experimentos foram realizados no Laboratório de Música Eletroacústica do Departamento de Música e Artes Cênicas (DEMAC), Núcleo de Processamento de Dados (NUPRO) da Universidade Federal de Uberlândia (UFU) e nos computadores do Centro Nacional de Processamento de Alto Desempenho do Nordeste (CENAPAD/NE) e de Minas Gerais (CENAPAD/MG) através de processamento local e remoto, utilizando-se os recursos da Rede Nacional de Pesquisa/REDEMINAS (RNP/REDEMINAS) e Empresa Brasileira de Telecomunicações (EMBRATEL), através de um link de 1 MB cedido pela empresa Coligui Ltda. de Uberlândia.

Os equipamentos utilizados foram computadores de última geração, salvo nos casos em que não eram necessários tais recursos, quando foram utilizados computadores PCs.

3.1.1 Equipamentos e Programas

Descrição e especificação dos equipamentos utilizados e lotados no Laboratório de Música Eletroacústica da UFU:

1. Computador PC 486 DX2/50, 32Mb de memória RAM e 760Mb de Disco Rígido, com sistema operacional Linux, Kernel 1.2.13 atualizado para 2.0.30;

2. Computador PC/IBM^{TM}modelo P75, 16Mb de memória RAM e 2.1Gb de Disco Rígido, com sistema operacional Linux, Kernel 2.0.30;

3. Computador PC/IBM^{TM}modelo P75, 16Mb de memória RAM e 540Mb e 1.2Gb de Disco Rígido, com sistema operacional Linux, Kernel 2.0.32;

4. Estação RISC6000/IBM^{TM}, 64Mb de memória RAM e 4.2Gb de Disco Rígido, com sistema operacional AIX/IBM^{TM}versão 4.2.

Descrição e especificação dos equipamentos utilizados e lotados no Núcleo de Processamento de Dados da UFU:

1. Estação RISC6000/IBM^{TM}duplo processador, 128Mb de memória RAM e 8.4Gb de Disco Rígido, com sistema operacional AIX/IBM^{TM}versão 4.2;

2. Cluster de RISCs SP3/IBM^{TM}com 8 processadores paralelos, 256Mb de memória RAM e 15Gb de Disco Rígido, com sistema operacional AIX/IBM^{TM}versão 4.2;

3. Supercomputador IBM^{TM}modelo 3090, atualizado para 9672, com sistema operacional VM/CMS.

Descrição e especificação dos equipamentos utilizados e lotados no CENAPAD/NE e CENAPAD/MG:

1. Cluster de RISCs SP2 e SP3/IBM com 4 ou 8 processadores paralelos, 256Mb de memória RAM e 20Gb de Disco Rígido, com sistema operacional AIX/IBM^{TM}versão 4.0 ou superior.

Descrição de equipamento pessoal utilizado:

1. Computador PC Pentium 133MHz, 32Mb de memória RAM e 2.1 de Disco Rígido, com sistema operacional Linux, Kernel 2.0.36;

2. Computador Portátil 486 DX4/100, 8Mb de memória e 540Mb de Disco Rígido, sistema operacional LINUX, kernel 2.0.32.

Compiladores utilizados:

1. Compilador GCC da *Free Software Foundation* (FSF/GNU) versão 2.7.2.1 ou superior, para compilação dos códigos em C/C++;

2. Compilador CC para o AIX/IBM^{TM}, para compilação dos códigos em C/C++;

3. Compilador FORTRAN77 da *Free Software Foundation* (FSF/GNU), versão 1.4h, para compilação dos códigos em FORTRAN [20];

4. Compilador FORTRAN77 da IBM^{TM}para o Mainframe 9672 (cf. item anterior);

5. *Saphire Sound Compiler*, versão 1.8, Jim Finnis[1], para compilação de amostras sintetizadas em testes com padrões conhecidos de freqüências;

6. *Sound Compact Compiler*, versão 1.0, João Dovicchi, para gerar padrões conhecidos de freqüências [38];

Outros *softwares* utilizados:

1. *X Window System Wavelet Packet Laboratory* (xwpl), versão 1.3[2];

2. Code: **THE FAST WAVELET TRANSFORM** by Mac A. Cody [20];

3. *DSPLab Version 1.0 Beta*[3];

4. **Numerical Recipes Software, FORTRAN code for Daubechies Wavelets**[4];

5. *XMusic: a GUI for Barry Vercoe's CSound*, de George Drapeau[5];

6. *MiXViews version 1.21* (mxv)[6];

3.1.2 Amostras de Instrumentos Acústicos

As amostras de instrumentos acústicos foram gentilmente cedidas pelo Prof. Lawrence Fritts[7] da Universidade de Iowa, no formato AIFF, gravadas em 16 bits linear (cf. fichas

[1] `white@elf.dircom.co.uk`

[2] `ftp.pascal.math.yale.edu/pub/software/xwpl`

[3] `riley@jchisholm.phys.dal.ca`

[4] Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, 1992.

[5] `drapeau@jessica.stanford.edu`

[6] Douglas Scott `MixViews@create.ucsb.edu`

[7] Electronic Music Studios `lawrence-frits@uiowa.edu`

Instr.	nota	freq. (Hz)	din.	num. de amostras	tempo de duração	Ampl. Máx.
flauta	dó	260.0	mf	111.291	2,5236	2,087
(vibrato)	lá	440.0	mf	114.448	2,5952	3,716
flauta	dó	260.0	mf	118.613	2,6896	0,582
(sem vibr.)	lá	440.0	mf	112.641	2,5542	3,318
oboé	dó	260.0	mf	61.937	1,4044	3,900
(sem vibr.)	lá	440.0	mf	70.100	1,5896	3,064
clarinete	dó	260.0	mf	106.681	2,4191	3,906
(Si bemol)	lá	440.0	mf	112.766	2,5570	3,888
Fagote	dó	65.0	mf	100.140	2,2707	10,941
	lá	110.0	mf	102.518	2,3147	7,928
Trompa	dó	260.0	mf	114.297	2.5918	5.067
(Fá)	lá	440.0	mf	110.979	2.5165	11.341

Tabela 3.1: Especificação dos timbres utilizados para teste dos coeficientes. Amplitude máxima em valores absolutos, taxa de amostragem 44,1 kHz e tempo de duração em segundos.

técnicas no apêndice C.1). Para o teste dos coeficientes, as freqüências foram separadas e segmentadas em 2^n amostras tanto no transiente de ataque como no estado estacionário.

Após vários estudos com diversos formatos, optou-se por um que produzisse melhores resultados com menor esforço computacional para a análise do modelo proposto. Assim, as amostras foram transformadas para 8 bits com a finalidade de facilitar a análise numérica e espectrográfica, usando-se o *software* MiXViews. As amostras e seus respectivos transientes encontram-se descritas na tabela 3.1.

3.2 Método

3.2.1 Descrição

Com a finalidade de estudar o comportamento do modelo proposto, foram escolhidos apenas algumas freqüências de alguns instrumentos de sopros. Neste caso, sons de flauta transversal (com e sem vibrato), oboé, clarinete em si bemol , fagote e trompa em fá foram estudados em seus transientes de ataque e de estado estacionário. O teste do modelo foi levado a termo por meio de:

1. Transformada *Wavelet* direta e inversa;

2. Reconstrução do sinal e refinamento dos segmentos de passa-baixa (cf. apêndice B.4);

3. Cálculo da função limiar[8] do sinal original e do sinal reconstruído (cf. apêndice B.4);

4. Pesquisa da particao adequada do sinal, por intermédio do estudo dos segmentos passa-baixa do filtros *Wavelets*;

5. Pesquisa de coeficientes apropriados[9] e a montagem de uma biblioteca de funções para esta finalidade (cf. apêndice B.5);

6. Estudo da eficiência das conversões entre os domínios do tempo e da freqüência, em cada escala de tempo-freqüência sem perda significativa da precisão dos resultados;

---

[8] *Thresholding*

[9] Bases apropriadas para a análise de timbres, modeladas a partir de diferentes intervalos de freqüência.

7. Estudo da eficácia da combinação da análise de Fourier com a análise *Wavelet* dos sinais musicais [39],

com a finalidade de verificar a importância da localização do espectro de freqüências em cada transiente do envelope sonoro.

3.2.2 Construção de *Wavelets* Ortonormais

A construção de *Wavelets* ortonormais que sejam compactas, ou seja, localizadas no tempo pode ser feita através da análise de multiresolução, descrita anteriormente, e posterior associação das funções escalonantes com os coeficientes do filtro *Wavelet*. Para isto, estabeleceu-se as equações algébricas necessárias de tal forma que o modelo satisfaça as condições estabelecidas pelo filtro. Isso inclui condições que garantam o suporte compacto e determinado grau de aproximação pelas *Wavelets* associadas. (cf. Secção 2.5)

O processo escolhido foi o seguinte:

1. Solução das equações com a finalidade de obter um filtro *Wavelet* finito;

2. Como o filtro determina a equação de dilação, foram preditas, por seu intermédio, funções escalonantes plausíveis.

3. A equação de dilação foi resolvida iterativamente, ou foi calculada uma transformada de Fourier da função escalonante e tomada sua transformada inversa;

4. Através da função escalonante e do filtro *Wavelet* obtido, foi predita uma *Wavelet* ortonormal conforme o Teorema Fundamental (cf. teorema 2.4.1);

5. Assim, a função escalonante foi testada e, conseqüentemente, verificou-se a ortonormalidade *Wavelet* prevista.

Os coeficientes dos filtros *Wavelets* associados às funções escalonantes com suporte no intervalo $[0, N]$ são números $h_0, h_1, \ldots, h_n$ que satisfazem as seguintes condições:

1. A soma dos coeficientes é igual a $\sqrt{2}$

$$\sum_{k=0}^{n} h_k = \sqrt{2}. \tag{3.1}$$

2. A soma dos quadrados dos coeficientes é iqual a 1

$$\sum_{k=0}^{n} |h_k|^2 = 1. \tag{3.2}$$

Além disso, deve ser satisfeita a condição:

$$\sum_{k=0}^{N} h_k h_{k+2l} = \delta(1, 0). \tag{3.3}$$

Nota 3.2.1 *Embora a equação 3.3 deva ser satisfeita para qualquer inteiro k, considera-se apenas os valores entre $-N/2$ e $+N/2$, uma vez que para os outros valores ela será reduzida a uma soma de zeros Em compensação, os filtros Wavelets terão que satisfazer a condição de aproximação $A(p)$ para um inteiro p, tal que:*

$$\sum_{k=0}^{n} (-1)^k k^l h_k = 0, \tag{3.4}$$

onde k é um inteiro e $0 \leq k < p-1$.

3.2.3 Algoritmos

O cálculo dos coeficientes foi feito de acordo com as equações paramétricas de Pollen [80], também descritas por Cody [20] e por Daubechies [28]. O cálculo dos parâmetros de espaço[10] para um sistema de coeficientes h_n tal que $-2 \leq n \leq 3$, pode ser obtido por:

$$\begin{aligned}
h_{-2} &= \frac{(1+\cos\alpha+\sin\alpha)\cdot(1-\cos\beta-\sin\beta)+2\sin\beta\cos\alpha}{4} \\
h_{-1} &= \frac{(1-\cos\alpha+\sin\alpha)\cdot(1+\cos\beta-\sin\beta)-2\sin\beta\cos\alpha}{4} \\
h_0 &= \frac{1+\cos(\alpha-\beta)+\sin(\alpha-\beta)}{2} \\
h_1 &= \frac{1+\cos(\alpha-\beta)-\sin(\alpha-\beta)}{2} \\
h_2 &= 1-h_{-2}-h_0 \\
h_3 &= 1-h_{-1}-h_1
\end{aligned}$$

onde $\alpha \geq -\pi$ e $\beta < \pi$. Para isso, foi desenvolvido um programa em C que calcula os coeficientes a partir dos dados fornecidos (cf. apêndice B.1).

Os coeficientes obtidos foram usados para gerar funções escalonantes (filtros passa-baixa) e funções *Wavelets* (filtros passa-alta). Os parâmetros escalonantes correspondem diretamente aos coeficientes *Wavelet*, enquanto os filtros *Wavelets* são formados por ordem reversa dos coeficientes obtidos, com os sinais invertidos. Os filtros são representações das redefinições de funções escalonantes e *Wavelets*, de forma recursiva, e as aproximações

---

[10] Um parâmetro de espaço é um conjunto de soluções que mapeiam um conjunto de funções em uma ou mais variáveis ou parâmetros.

foram feitas através de repetidas convoluções entre o filtro de passa-baixa e as funções escalonantes e os coeficientes[11]. Tal algoritmo foi descrito por Burt e Adelson [14]. A convolução de uma etapa da transformada *Wavelet* encontra-se na subrotina listada no apêndice B.3.

3.2.4 Procedimentos

Os sinais foram convertidos a partir de um arquivo de áudio no formato RIFF/WAVE para um arquivo contendo o vetor de dados, utilizando-se um programa em C, feito para este fim (cf. apêndice B.2), e seccionados em suas porções de ataque e estado estacionário com 4096 amostras, cada um.

O conjunto de 4 coeficientes foi obtido através das equações paramétricas de Pollen [80], fazendo-se $\alpha = 2^{1/12}$ e $\beta = 0$. O conjunto de 6 coeficientes foi otido de forma semelhante, fazendo-se $\alpha = 2^{1/12}$ e $\beta = 2^{2/12} - \alpha$. Sendo que, α representa o fator cromático da escala "bem-temperada" e β a diferença intervalar. Por se tratarem de valores derivados do intervalo cromático, sugere-se a denominação de **Chroma4** e **Chroma6** para os coeficientes, respectivamente.

Usando-se o *MixViews* (mxv) foi calculada a transformada de Fourier de cada segmento selecionado, para comparação posterior com os resultados obtidos nos arquivos tratados com as transformadas *Wavelets*. Posteriormente, foram procedidas as análises usando-se os coeficientes de Haar, Daubechies4, Daubechies6, Coiflet4 e Coiflet6, cujos resultados foram comparados com os obtidos pelos coeficientes **Chroma4** e **Chroma6**.

[11] Este processo é conhecido como algoritmo em pirâmide ou em cascata.

Usando-se um algoritmo semelhante ao descrito por Mallat [62], cada segmento da transformada *Wavelet* foi separado para a obtenção de várias secções. Cada secção foi dividida ao meio ($N/2$) separando-se o segmento de passa-alta e passa-baixa, estudados separadamente. O algoritmo de Mallat usa uma função escalonante para separar os segmentos de passa-alta em determinado ponto, sem que o filtro tenha, necessariamente, que chegar ao final da transformação (até $N = 1$). Neste trabalho, os segmentos de passa-alta foram eliminados um a um até que o restante do sinal transformado pudesse render a recuperação do sinal original sem perda significativa de suas características. Em cada partição, tomou-se sempre a metade da transformada ou o segmento de passa-baixa. (cf. figura 3.1)

Através do algoritmo do refinamento descrito no apêndice B.4, foram feitos estudos de várias secções de passa-baixa dos filtros *Wavelets* aplicados, com a finalidade de analisar a fidelidade dos resultados e a eficiência dos coeficientes, na recuperação do sinal quando comparados com o espectro original. Esta técnica consiste em separar as transformadas *Wavelets* Discretas (DWT[12]) usando-se o processo em pirâmide, semelhante ao descrito por Burt e Adelson[14].

O processo consiste na convolução dos dados amostrados com uma base *Wavelet* ortonormal que produz dois segmentos de dados como em um filtro de impulso de resposta finita (FIR[13]). Assim, os coeficientes *Wavelets* são usados como filtros passa-baixa e passa-alta complementares que, à semelhança de um filtro FIR podem ser descritos como:

[12] Discrete Wavelet Transform
[13] Finite Impulse Response

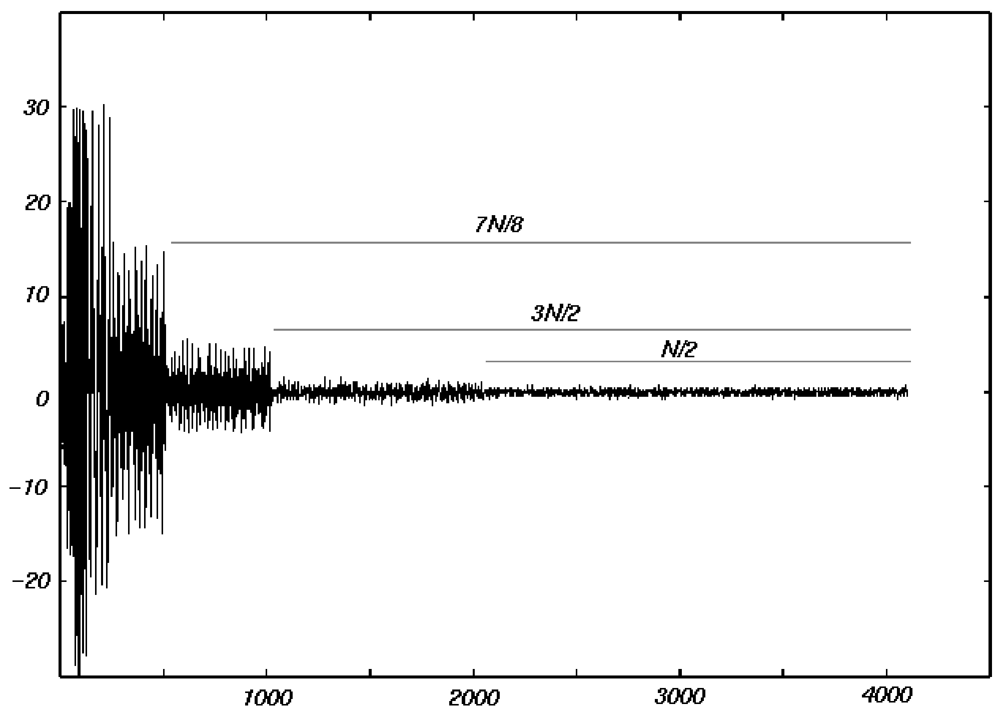

Figura 3.1: Exemplo da partição do sinal transformado, usando o próprio segmento de passa-baixa como função escalonante.

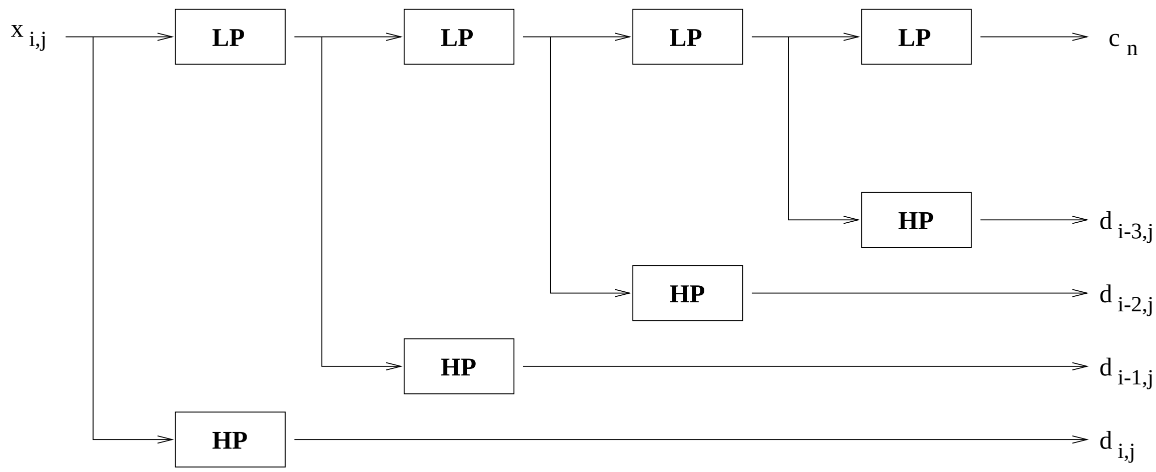

Figura 3.2: Algoritmo em pirâmide da transformada *Wavelet* direta e a convolução repetida da amostra com os coeficientes. LP = *Low-Pass* e HP = *High-Pass*.

$$y_k = \sum_{t=-\infty}^{\infty} c_{(k-t)} \cdot x_t$$

onde y_k é a saída produzida de um sinal x_k convoluto com $c_{(k-t)}$. O novo conjunto é novamente separado em dois novos segmentos de passa-alta e passa-baixa. Cada vez, o segmento tinha $N/2$ elementos, ou seja, era dimidiado em relação ao conjunto anterior. Tal operação pode ser, sucessivamente, repetida até que se atinja o intervalo de amostragem. Para N operações, faz-se necessário um conjunto de 2^N amostras (cf. figura 3.2).

Através do algoritmo inverso, cada segmento passa-baixa pode ser usado para recuperar o sinal, mesmo sendo desprezado o segmento de passa-alta. Este algoritmo é denominado de **refinamento** [28] e o resultado obtido foi comparado com o sinal original.

Capítulo 4

Resultados

Os resultados apresentados aqui foram obtidos pelos métodos descritos no capítulo anterior. Os arquivos originais foram convertidos[1] de 16 bits linear para 8 bits linear pelo MixViews e foi calculada sua transformada de Fourier. As amplitudes de cada formante do espectro obtido por uma transformada localizada de Fourier (STFT[2]), usando-se um janelamento de 100 amostras em secções do ataque e estado estacionário dos sinais originais, encontram-se no apêndice C. Todos os arquivos utilizados foram obtidos por gravação direta do instrumento, com taxa de amostragem de 44.1 kHz, e os detalhes dos arquivos originais encontram-se no apêndice C.1[3]

[1]A conversão foi feita para maior praticidade, uma vez que nos testes do modelo proposto o que interessava era a coerência dos valores numéricos.

[2]Short Time Fourier Transform

[3]Os tempos t_n, nas tabelas, referem-se às magnitudes absolutas do espectro obtido numa janela de 100 amostras em torno de $n \times 11.4625$ ms para $n = \{1 \ldots 5\}$.

4.1 Transformadas Diretas e Inversas

Os estudos das transformadas *Wavelets* diretas e inversas dos sinais, usando-se os coeficientes **Chroma4** e **Chroma6**, foram feitos para medir sua aplicabilidade no processo de filtragem. Na verdade, a ortonormalidade dos coeficientes, conforme descrição nos capítulos 2 e 3, e seu suporte compacto já apontava para os resultados obtidos, entretanto a qualidade dos resultados dependiam de testes mais formais. Assim, os estudos apresentaram resultados extremamente consistentes, quando comparados com os sinais originais. Como nota-se na tabela 4.1, as transformadas de Fourier do sinal original e do sinal tratado são extremamente semelhantes.

Sabe-se que os timbres dos instrumentos de sopros mais graves, como no caso do fagote, apresentam várias características próprias. O que dificulta sua análise, não apenas a sobreposição[4] causada pela pequena diferença intervalar entre seus harmônicos, mas também o fato das freqüências mais proeminentes se localizarem em outras faixas bastante diferentes da fundamental [92]. Exatamente por estas características, estes sinais são ideais para testar a consistência do modelo. No caso do fagote, em notas mais graves, a energia da fundamental e dos primeiros parciais é bastante inferior à dos formantes em torno da freqüência de 500 Hz[5]. A figura 4.1 mostra como as freqüências em torno de 500 Hz têm intensidade bem maior que os primeiros harmônicos.

Se compararmos os espectros obtidos por STFT do timbre original do Fagote e os

---

[4] *Overlapping*

[5] Alguns sons do fagote como a nota com freqüência de 130 tem a fundamental com magnitude zero. O som que se percebe é produzido no ouvido interno onde a fundamental é localizada pela diferença entre os parciais existentes no timbre [43, 92, 95].

sinal	**freqüência**	t_1	t_2	t_3	t_4	t_5
original	441.43	0.6996	3.3319	5.7855	6.2912	5.7680
Chroma4	441.43	0.6996	3.3321	5.7860	6.2918	5.7701
Chroma6	441.43	0.6988	3.3283	5.7787	6.2824	5.7616
original	882.86	0.2538	1.2277	2.1117	2.2623	2.0957
Chroma4	882.86	0.2535	1.2264	2.1091	2.2591	2.0941
Chroma6	882.86	0.2542	1.2278	2.1094	2.2582	2.0922
original	1324.29	0.0632	0.3432	0.6033	0.6418	0.6201
Chroma4	1324.29	0.0630	0.3427	0.6028	0.6413	0.6182
Chroma6	1324.29	0.0638	0.3451	0.6057	0.6441	0.6211
original	1765.72	0.0407	0.2009	0.3366	0.3471	0.3383
Chroma4	1765.72	0.0410	0.2016	0.3375	0.3484	0.3355
Chroma6	1765.72	0.0406	0.2008	0.3367	0.3476	0.3385
original	2207.15	0.0320	0.1362	0.2057	0.2035	0.1956
Chroma4	2207.15	0.0322	0.1363	0.2052	0.2030	0.1916
Chroma6	2207.15	0.0319	0.1357	0.2042	0.2030	0.1922
original	2648.58	0.0082	0.0281	0.0364	0.0377	0.0451
Chroma4	2648.58	0.0085	0.0297	0.0387	0.0399	0.0451
Chroma6	2648.58	0.0083	0.0286	0.0364	0.0369	0.0421
original	3090.01	0.0074	0.0250	0.0284	0.0220	0.0169
Chroma4	3090.01	0.0074	0.0263	0.0306	0.0237	0.0188
Chroma6	3090.01	0.0075	0.0254	0.0274	0.0225	0.0154
original	3531.45	0.0011	0.0089	0.0156	0.0148	0.0182
Chroma4	3531.45	0.0016	0.0097	0.0162	0.0158	0.0211
Chroma6	3531.45	0.0012	0.0102	0.0166	0.0136	0.0218
original	3972.88	0.0036	0.0108	0.0088	0.0049	0.0083
Chroma4	3972.88	0.0037	0.0125	0.0109	0.0058	0.0085
Chroma6	3972.88	0.0039	0.0117	0.0094	0.0067	0.0056
original	4414.31	0.0017	0.0021	0.0008	0.0007	0.0024
Chroma4	4414.31	0.0016	0.0018	0.0012	0.0006	0.0046
Chroma6	4414.31	0.0013	0.0021	0.0012	0.0008	0.0045
original	5297.17	0.0013	0.0018	0.0017	0.0024	0.0053
Chroma4	5297.17	0.0011	0.0019	0.0022	0.0022	0.0048
Chroma6	5297.17	0.0013	0.0015	0.0001	0.0020	0.0046
original	5738.60	0.0024	0.0047	0.0032	0.0060	0.0069
Chroma4	5738.60	0.0025	0.0056	0.0033	0.0065	0.0091
Chroma6	5738.60	0.0027	0.0063	0.0045	0.0074	0.0085
original	6180.03	0.0012	0.0024	0.0015	0.0012	0.0024
Chroma4	6180.03	0.0010	0.0030	0.0023	0.0014	0.0033
Chroma6	6180.03	0.0014	0.0031	0.0009	0.0015	0.0037

Tabela 4.1: Comparação do espectro do som de uma Trompa em fá tocando Lá ($\sim$ 440 Hz). Amostra do estado estacionário. Máxima Magnitude do espectro no som original = 6.2920 em 441.4 Hz.

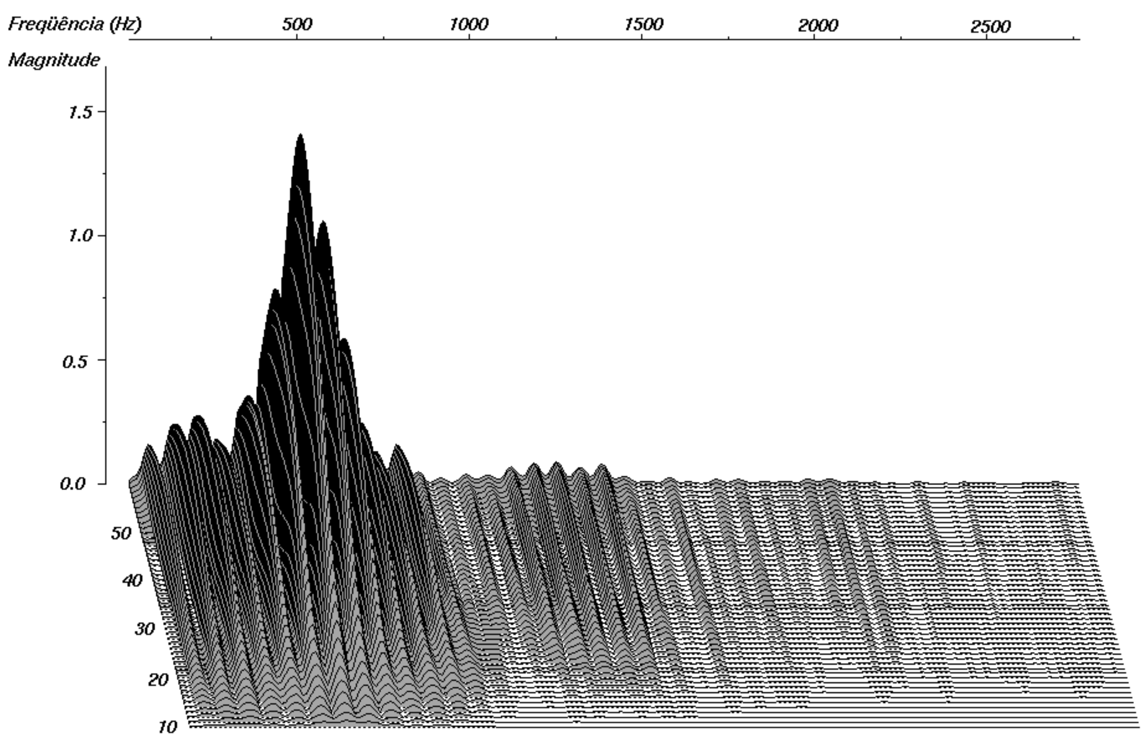

Figura 4.1: Espectro do som original de fagote tocando Dó ($\sim$ 64.60 Hz). Amostra do ataque.

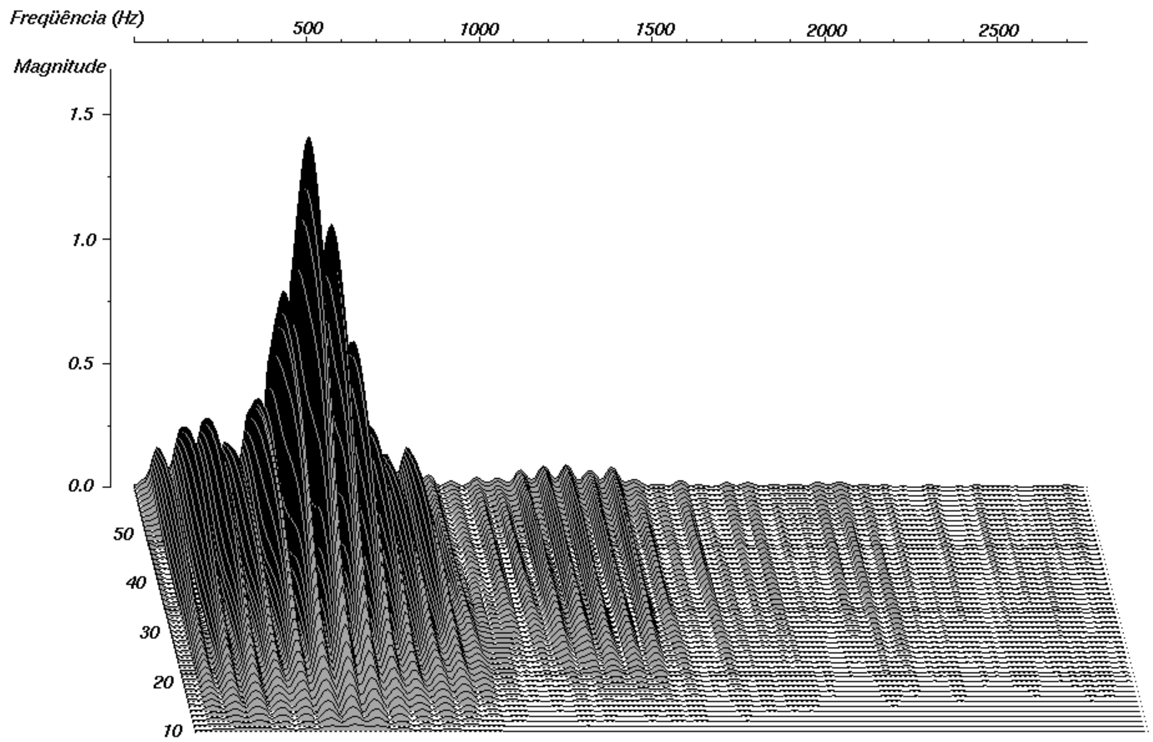

Figura 4.2: Espectro do som tratado com o coeficiente Chroma4 de fagote tocando Dó 2 ($\sim$ 64.60 Hz). Amostra do ataque.

tratados com os novos coeficientes, quase não se pode notar diferença, tal como visto nas figuras 4.1, 4.2 e 4.3.

Quando os resultados das freqüências de maior intensidade do fagote são comparados (cf. tabela 4.2), pôde-se observar uma pequena alteração na reconstrução do sinal em torno da região de maior energia dos formantes, principalmente, quando se utiliza os coeficientes **Chroma6** na sua transformação. Entretanto, tal variação pode ser desprezada já que não existem alterações perceptíveis no resultado sonoro.

No caso do som mais grave da flauta (Dó), apesar do elevado ruído não periódico causado pelo fluxo de ar durante a emissão da nota, a reconstrução do sinal a partir

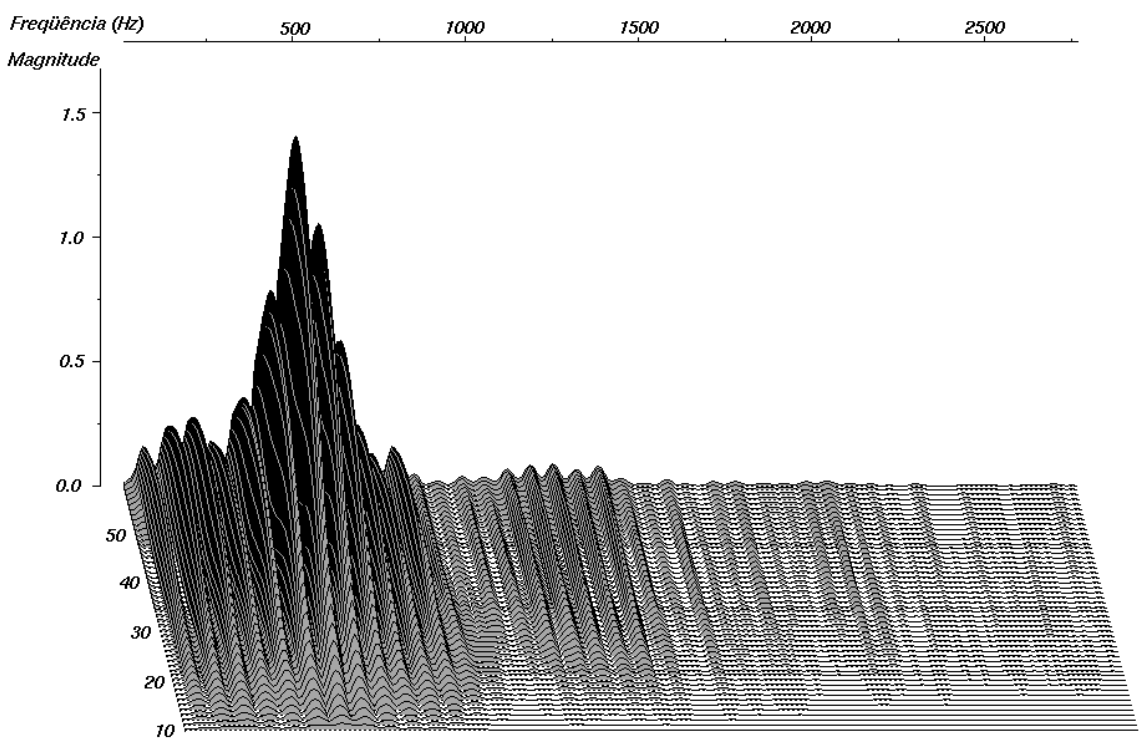

Figura 4.3: Espectro do som tratado com o coeficiente Chroma6 de fagote tocando Dó 2 ($\sim$ 64.60 Hz). Amostra do ataque.

sinal	**freqüência**	t_1	t_2	t_3	t_4	t_5
original	323.00	0.0150	0.1672	0.4218	0.5365	0.4915
Chroma4	323.00	0.0150	0.1674	0.4225	0.5377	0.4926
Chroma6	323.00	0.0147	0.1662	0.4206	0.5349	0.4910
original	387.60	0.0225	0.2622	0.6790	0.8621	0.8048
Chroma4	387.60	0.0225	0.2623	0.6799	0.8631	0.8054
Chroma6	387.60	0.0222	0.2605	**0.6768**	**0.8591**	**0.8028**
original	452.20	0.0266	0.3308	0.9116	1.1945	1.2027
Chroma4	452.20	0.0266	0.3308	0.9116	1.1938	1.2025
Chroma6	452.20	0.0262	**0.3286**	**0.9083**	**1.1899**	**1.1980**
original	516.80	0.0213	0.2661	0.6794	0.7578	0.8655
Chroma4	516.80	0.0213	0.2659	0.6783	0.7561	0.8652
Chroma6	516.80	0.0210	0.2642	**0.6768**	**0.7556**	**0.8619**
original	581.40	0.0116	0.1458	0.3637	0.4034	0.4793
Chroma4	581.40	0.0116	0.1454	0.3627	0.4031	0.4793
Chroma6	581.40	0.0115	0.1455	0.3635	0.4017	**0.4770**

Tabela 4.2: Comparação das freqüências mais significativas do espectro do som de um fagote tocando uma nota Dó ($\sim$ 64.60 Hz). Amostra do ataque. Máxima Magnitude no espectro do som original = 1.6760 em 463.0 Hz. Em negrito estão destacados os valores de magnitude com diferencial $\delta \geq 0.002$.

das transformadas **Chroma4** e **Chroma6** não apresentou diferença em relação ao sinal original (cf. tabela 4.3).

Como pode ser observado na tabela 4.4, a comparação entre os coeficientes de Haar, Coiflet4, Daubechies4 e Chroma4, mostrou maior proximidade dos valores de magnitude entre o espectro do som original e do sinal transformado, usando-se os coeficientes **Chroma4**, embora o resultado não tenha nenhuma influência na qualidade do timbre, uma vez que não se pode perceber nenhuma diferença auditiva[6].

A tabela 4.5 mostra os resultados da comparação entre os coeficientes de Haar, Coiflet6, Daubechies6 e Chroma6, na qual nota-se resultado semelhante ao da tabela 4.4.

4.2 Refinamento

Para a aplicação do algoritmo de refinamento (cf. apêndice B.4), cada etapa da convolução para o cálculo da transformada foi selecionada e separada, conforme ilustra a figura 3.1. O estudo do refinamento mostrou que os coeficientes **Chroma4** e **Chroma6** apresentaram resultados satisfatórios e que podem ser aplicados, também, em algoritmos de compressão de sinais, de forma semelhante às *wave packets* descritas por Wickerhauser [106] e que será objeto de estudo futuro.

O modelo mostrou ser capaz de recuperar o sinal com fidelidade ao sinal original, até com o segmento passa-baixa com $N/4$ elementos. Se forem utilizadas recursões maiores, o sinal poderá ser recuperado desde que adequadamente tratado com outro modelo para

[6]Posteriormente, a utilização destes coeficientes para a localização determinadas freqüências será estudada, devendo ser objeto de análise mais minuciosa.

sinal	**freqüência**	t_1	t_2	t_3	t_4	t_5
original	258.40	0.0016	0.0235	0.0845	0.1587	0.1863
Chroma4	258.40	0.0016	0.0235	0.0845	0.1587	0.1865
Chroma6	258.40	0.0016	0.0234	0.0840	0.1571	0.1841
original	516.80	0.0014	0.0163	0.0555	0.1103	0.1379
Chroma4	516.80	0.0014	0.0163	0.0555	0.1103	0.1380
Chroma6	516.80	0.0014	0.0163	0.0552	0.1093	0.1365
original	775.20	0.0013	0.0140	0.0284	0.0207	0.0134
Chroma4	775.20	0.0013	0.0140	0.0284	0.0207	0.0135
Chroma6	775.20	0.0013	0.0140	0.0285	0.0208	0.0129
original	1033.59	0.0011	0.0124	0.0346	0.0503	0.0543
Chroma4	1033.59	0.0011	0.0124	0.0346	0.0503	0.0543
Chroma6	1033.59	0.0011	0.0123	0.0344	0.0504	0.0542
original	1291.99	0.0009	0.0036	0.0036	0.0090	0.0137
Chroma4	1291.99	0.0009	0.0036	0.0036	0.0090	0.0137
Chroma6	1291.99	0.0009	0.0037	0.0038	0.0090	0.0128
original	1550.39	0.0008	0.0052	0.0056	0.0085	0.0249
Chroma4	1550.39	0.0008	0.0052	0.0056	0.0085	0.0248
Chroma6	1550.39	0.0008	0.0052	0.0057	0.0077	0.0235
original	1808.79	0.0007	0.0018	0.0155	0.0397	0.0545
Chroma4	1808.79	0.0007	0.0018	0.0155	0.0397	0.0545
Chroma6	1808.79	0.0007	0.0019	0.0154	0.0394	0.0542
original	2067.19	0.0007	0.0046	0.0104	0.0063	0.0057
Chroma4	2067.19	0.0007	0.0046	0.0104	0.0063	0.0056
Chroma6	2067.19	0.0007	0.0045	0.0104	0.0067	0.0045
original	2325.59	0.0007	0.0030	0.0081	0.0171	0.0154
Chroma4	2325.59	0.0007	0.0030	0.0081	0.0171	0.0155
Chroma6	2325.59	0.0007	0.0031	0.0084	0.0178	0.0163
original	2583.98	0.0007	0.0049	0.0082	0.0046	0.0104
Chroma4	2583.98	0.0007	0.0049	0.0082	0.0046	0.0103
Chroma6	2583.98	0.0007	0.0049	0.0083	0.0041	0.0089
original	2842.38	0.0007	0.0036	0.0050	0.0142	0.0130
Chroma4	2842.38	0.0007	0.0036	0.0050	0.0142	0.0130
Chroma6	2842.38	0.0007	0.0035	0.0047	0.0139	0.0128
original	3100.78	0.0007	0.0027	0.0040	0.0069	0.0115
Chroma4	3100.78	0.0007	0.0027	0.0040	0.0069	0.0114
Chroma6	3100.78	0.0007	0.0029	0.0041	0.0070	0.0117
original	3359.18	0.0006	0.0025	0.0016	0.0025	0.0018
Chroma4	3359.18	0.0006	0.0025	0.0016	0.0025	0.0019
Chroma6	3359.18	0.0006	0.0024	0.0014	0.0027	0.0018

Tabela 4.3: Comparação do espectro do som de uma Flauta tocando Dó ($\sim$ 260 Hz). Amostra do ataque. Máxima Magnitude do espectro no som original = 0.1889 em 258.4 Hz.

sinal	**freqüência**	t_1	t_2	t_3	t_4	t_5
original	258.40	0.0378	0.1797	0.3127	0.3423	0.3118
Haar	258.40	0.0378	0.1797	0.3123	0.3413	0.3107
Coif4	258.40	0.0374	0.1787	0.3121	0.3419	0.3093
Daub4	258.40	0.0375	0.1788	0.3111	0.3413	0.3119
Chroma4	258.40	0.0379	0.1802	0.3135	0.3431	0.3124
original	516.80	0.0445	0.2138	0.3684	0.3985	0.3680
Haar	516.80	0.0445	0.2138	0.3680	0.3974	0.3671
Coif4	516.80	0.0444	0.2139	0.3675	0.3978	0.3697
Daub4	516.80	0.0450	0.2149	0.3686	0.3973	0.3670
Chroma4	516.80	0.0446	0.2146	0.3696	0.3995	0.3690
original	775.20	0.0520	0.2381	0.3934	0.4183	0.3910
Haar	775.20	0.0520	0.2381	0.3930	0.4172	0.3903
Coif4	775.20	0.0521	0.2384	0.3937	0.4177	0.3891
Daub4	775.20	0.0516	0.2369	0.3917	0.4170	0.3900
Chroma4	775.20	0.0520	0.2386	0.3945	0.4192	0.3919
original	1033.59	0.0483	0.2271	0.3694	0.3864	0.3609
Haar	1033.59	0.0483	0.2271	0.3691	0.3856	0.3605
Coif4	1033.59	0.0481	0.2263	0.3694	0.3874	0.3607
Daub4	1033.59	0.0488	0.2281	0.3708	0.3886	0.3603
Chroma4	1033.59	0.0484	0.2276	0.3701	0.3869	0.3609
original	1291.99	0.1453	0.6508	0.9910	0.9955	0.9739
Haar	1291.99	0.1453	0.6508	0.9908	0.9948	0.9739
Coif4	1291.99	0.1437	0.6446	0.9821	0.9874	0.9661
Daub4	1291.99	0.1443	0.6462	0.9828	0.9860	0.9661
Chroma4	1291.99	0.1452	0.6501	0.9901	0.9954	0.9732
original	1550.39	0.0403	0.1853	0.2633	0.2470	0.2559
Haar	1550.39	0.0403	0.1853	0.2632	0.2466	0.2556
Coif4	1550.39	0.0399	0.1847	0.2625	0.2467	0.2579
Daub4	1550.39	0.0404	0.1857	0.2634	0.2460	0.2541
Chroma4	1550.39	0.0400	0.1842	0.2619	0.2464	0.2552

Tabela 4.4: Comparação dos coeficientes de Haar, Coiflet4, Daubechies4 e Chroma4 do som de um Oboé tocando Dó ($\sim$ 260 Hz). Amostra do ataque. Máxima Magnitude do espectro no som original = 1.4063 em 1300 Hz.

sinal	**freqüência**	t_1	t_2	t_3	t_4	t_5
original	258.40	0.0378	0.1797	0.3127	0.3423	0.3118
Haar	258.40	0.0378	0.1797	0.3123	0.3413	0.3107
Coif6	258.40	0.0372	0.1778	0.3106	0.3408	0.3110
Daub6	258.40	0.0372	0.1778	0.3105	0.3404	0.3103
Chroma6	258.40	0.0377	0.1798	0.3131	0.3423	0.3102
original	516.80	0.0445	0.2138	0.3684	0.3985	0.3680
Haar	516.80	0.0445	0.2138	0.3680	0.3974	0.3671
Coif6	516.80	0.0445	0.2138	0.3670	0.3969	0.3673
Daub6	516.80	0.0442	0.2132	0.3688	0.3989	0.3666
Chroma6	516.80	0.0437	0.2120	0.3660	0.3961	0.3668
original	775.20	0.0520	0.2381	0.3934	0.4183	0.3910
Haar	775.20	0.0520	0.2381	0.3930	0.4172	0.3903
Coif6	775.20	0.0516	0.2367	0.3925	0.4178	0.3907
Daub6	775.20	0.0523	0.2383	0.3929	0.4184	0.3908
Chroma6	775.20	0.0519	0.2380	0.3940	0.4198	0.3924
original	1033.59	0.0483	0.2271	0.3694	0.3864	0.3609
Haar	1033.59	0.0483	0.2271	0.3691	0.3856	0.3605
Coif6	1033.59	0.0478	0.2261	0.3696	0.3872	0.3611
Daub6	1033.59	0.0482	0.2262	0.3682	0.3852	0.3601
Chroma6	1033.59	0.0485	0.2277	0.3692	0.3870	0.3617
original	1291.99	0.1453	0.6508	0.9910	0.9955	0.9739
Haar	1291.99	0.1453	0.6508	0.9908	0.9948	0.9739
Coif6	1291.99	0.1435	0.6433	0.9813	0.9877	0.9658
Daub6	1291.99	0.1440	0.6451	0.9821	0.9883	0.9655
Chroma6	1291.99	0.1440	0.6457	0.9835	0.9891	0.9663
original	1550.39	0.0403	0.1853	0.2633	0.2470	0.2559
Haar	1550.39	0.0403	0.1853	0.2632	0.2466	0.2556
Coif6	1550.39	0.0404	0.1851	0.2629	0.2473	0.2555
Daub6	1550.39	0.0405	0.1847	0.2605	0.2455	0.2547
Chroma6	1550.39	0.0402	0.1841	0.2621	0.2468	0.2535

Tabela 4.5: Comparação dos coeficientes de Haar, Coiflet6, Daubechies6 e Chroma6 do som de um Oboé tocando Dó ($\sim$ 260 Hz). Amostra do ataque. Máxima Magnitude do espectro no som original = 1.4063 em 1300 Hz.

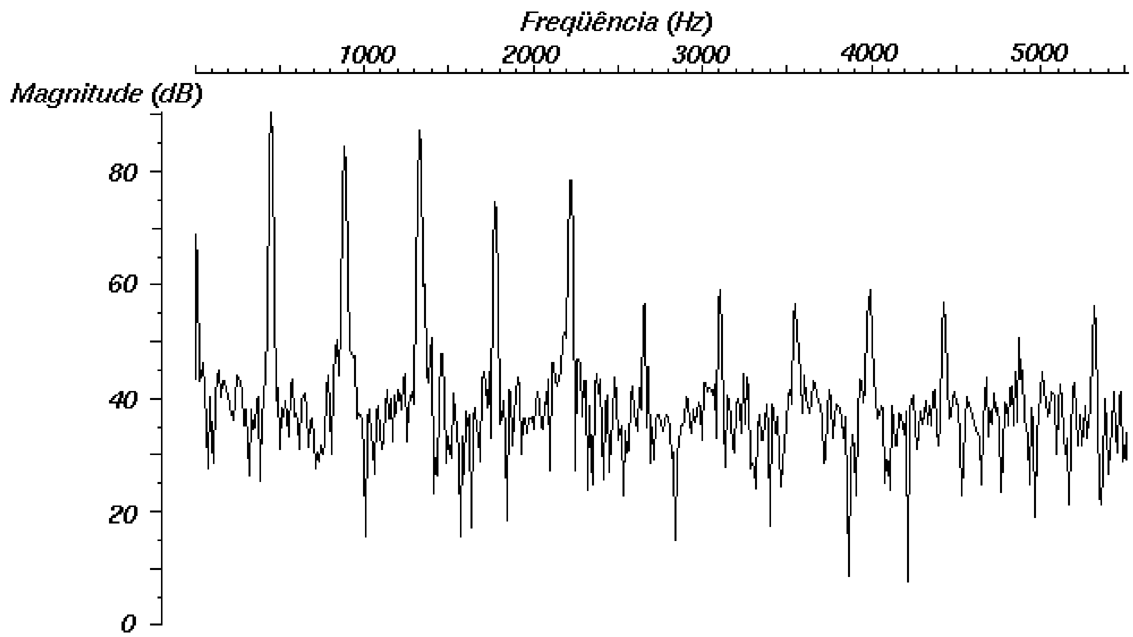

Figura 4.4: Espectro do som Lá de flauta transversal ($\sim$ 441.4 Hz). Amostra do estado estacionário, $t = 45.8$ ms.

a eliminar as altas freqüências indesejáveis que são ressaltadas ou adicionadas de acordo com a centróide[7] do sinal.

A figura 4.4 ilustra o espectro do estado estacionário da flauta transversal e as figuras 4.5, 4.6 e 4.7 mostram o espectro do sinal recuperado a partir dos segmentos de passa-baixa $N/2, N/4$ e $N/8$.

Comparando-se as figuras 4.5 e 4.6, com a figura 4.4, pôde-se observar apenas alterações muito sutis e que não afetam o som sintetizado, mantendo todas as características princi-

[7] A centróide de um timbre ́definida como

$$c = \frac{\sum_0^n f_n A_n}{\sum_0^n A_n}$$

onde f_n é freqüência de cada harmônico e A_n sua respectiva magnitude absoluta.

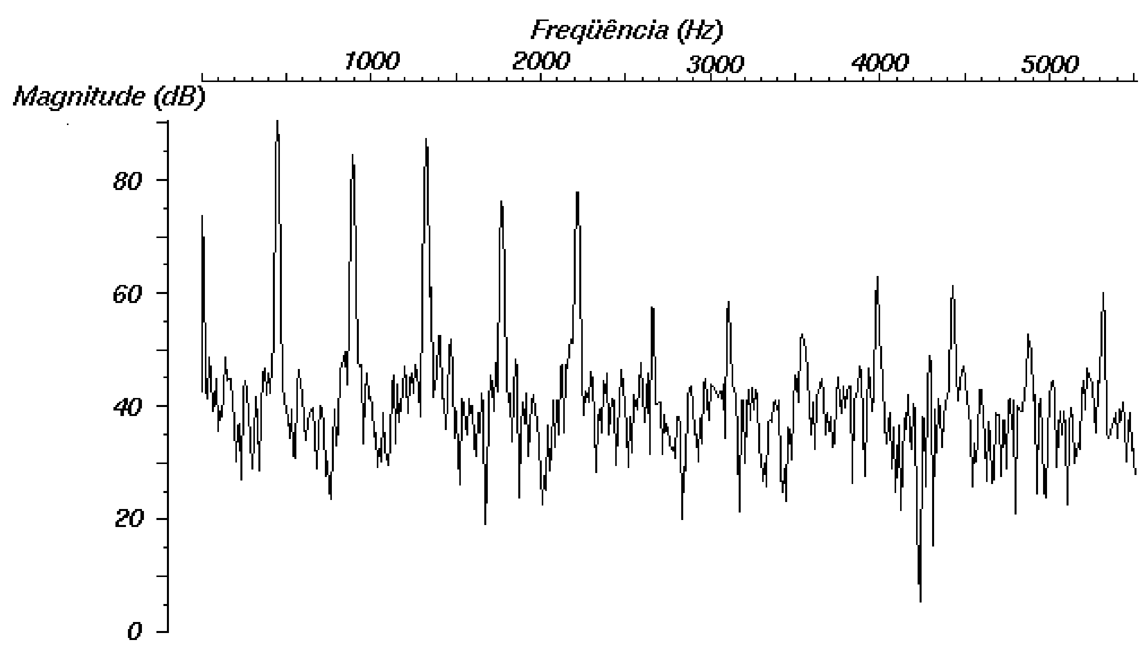

Figura 4.5: Espectro do refinamento do primeiro segmento da transformada *Wavelet* $(N/2)$ do som Lá de flauta transversal ($\sim$ 441.4 Hz). Amostra do estado estacionário, $t = 45.8$ ms.

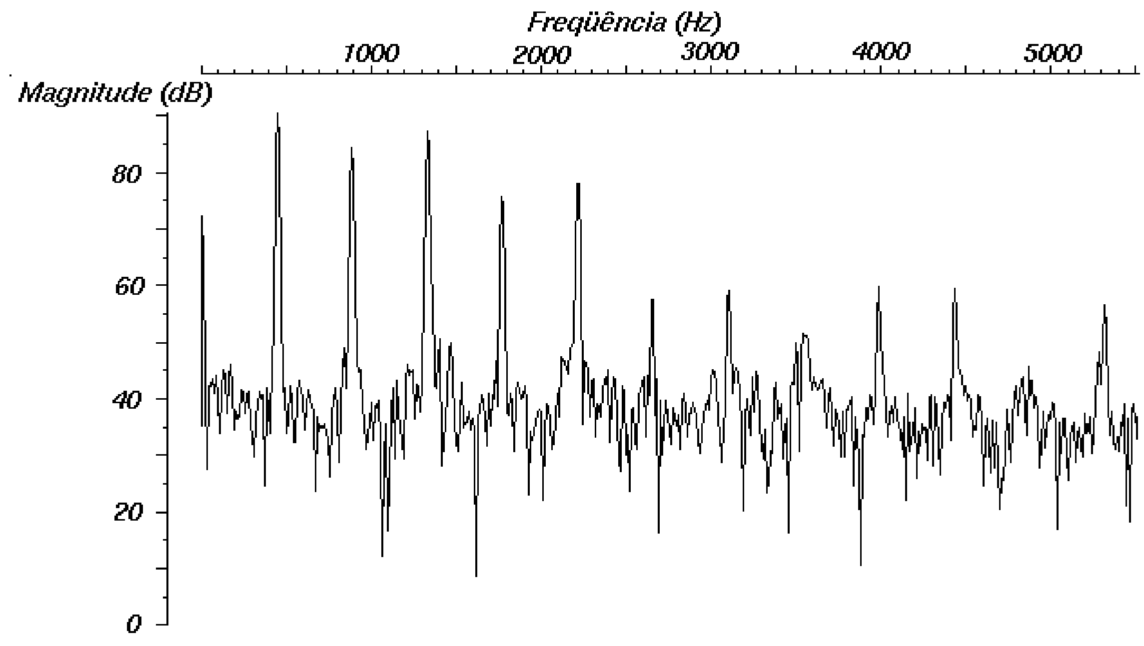

Figura 4.6: Espectro do refinamento do segundo segmento da transformada *Wavelet* ($N/4$) do som Lá de flauta transversal ($\sim$ 441.4 Hz). Amostra do estado estacionário, $t = 45.8$ ms.

pais do sinal original. Entretanto, quando comparadas a figura 4.7 e a figura 4.4 notou-se a interferência da região da centróide ($\sim$ 1013 Hz) na região de $\sim$ 4000 Hz. Tal interferência será objeto de estudo futuro para que se possa determinar a sua possível aplicação na identificação de instrumentos acústicos.

A explicação para o que ocorre está no fato de que, quando se efetua o partição do sinal transformado, cada convolução é calculada com metade do sinal original e, quando se aplica o algoritmo de refinamento com uma janela menor, os elementos dessa janela são ressaltados nos segmentos seguintes. Quanto menor a janela, maior a interferência obtida. Assim, o modelo está limitado à informação contida naquele segmento passa-baixa. Entretanto, o tempo de computação necessário para a obtenção da transformada é

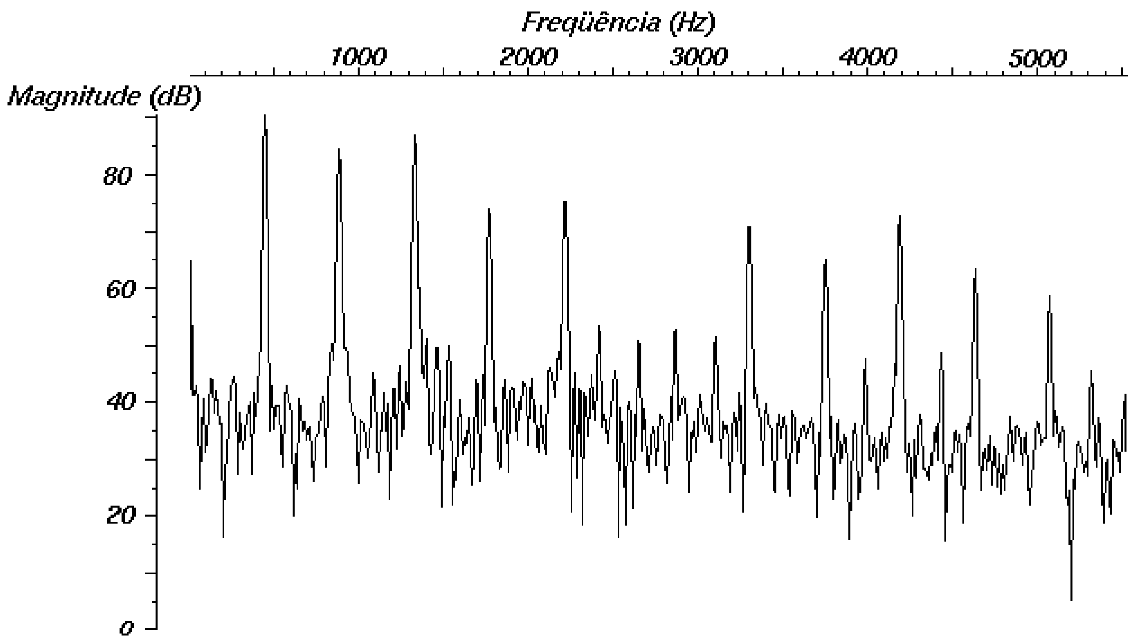

Figura 4.7: Espectro do refinamento do terceiro segmento da transformada *Wavelet* ($N/8$) do som Lá de flauta transversal ($\sim$ 441.4 Hz). Amostra do estado estacionário, $t = 45.8$ ms.

bem menor, uma vez que não se faz necessária a aplicação do algoritmo até a dimidiação final do escalograma.

O estudo com sons de outros instrumentos e com outros coeficientes *Wavelets* mostraram resultados semelhantes. As figuras 4.8 até 4.13 apresentam alguns exemplos comparativos dos refinamentos com os coeficientes **Chroma4** e **Chroma6**, a partir do sinal de outros timbres de instrumentos (por exemplo, clarinete e trompa).

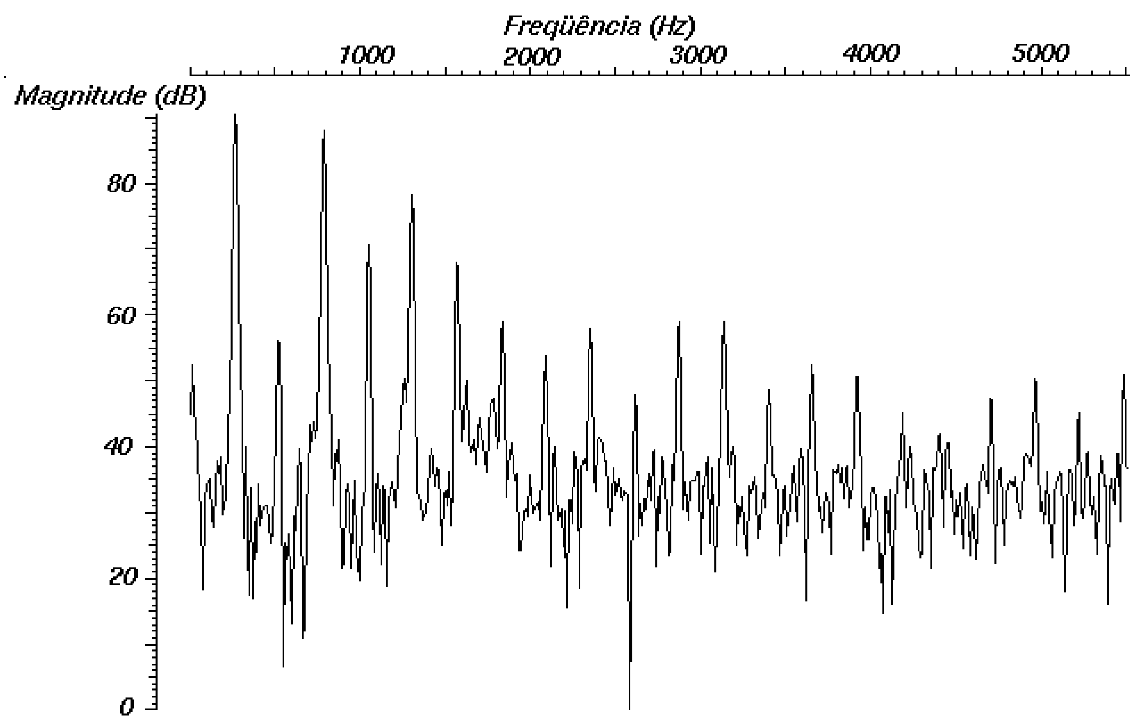

Figura 4.8: Espectro do som Dó do clarinete em Si bemol ($\sim$ 258.4 Hz). Amostra do estado estacionário, $t = 45.8$ ms.

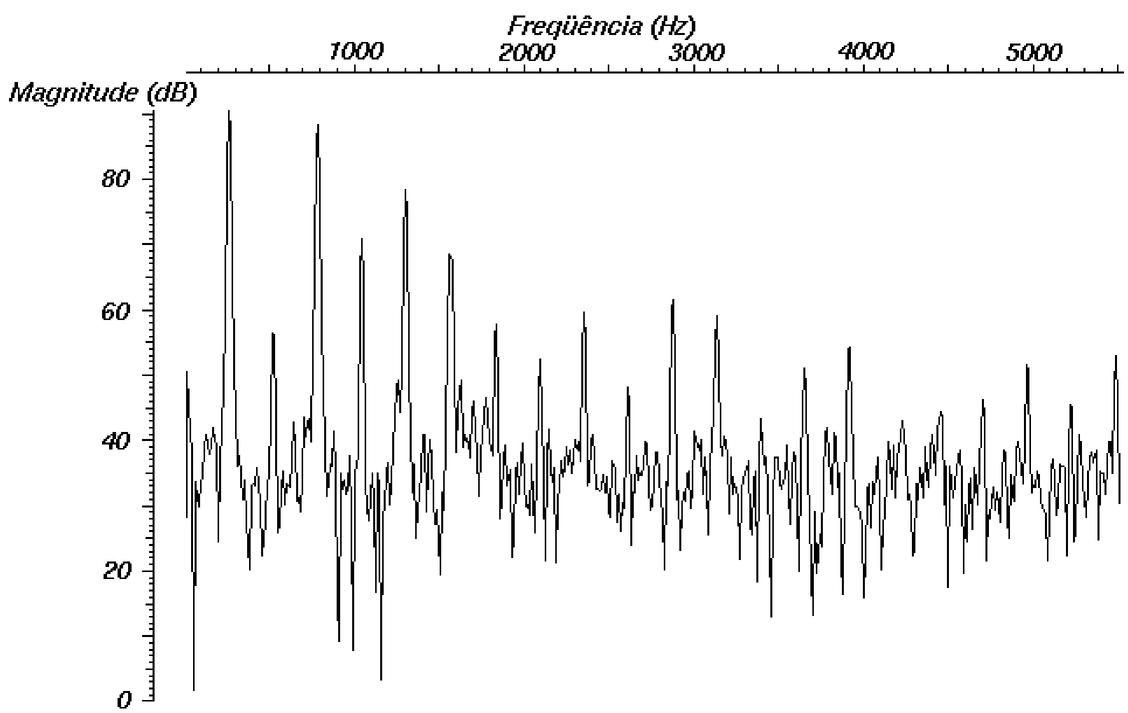

Figura 4.9: Espectro do refinamento ($N/2$) da transformada *Wavelet*, coeficientes **Chroma4**, do som Dó do clarinete em Si bemol ($\sim$ 258.4 Hz). Amostra do estado estacionário, $t = 45.8$ ms.

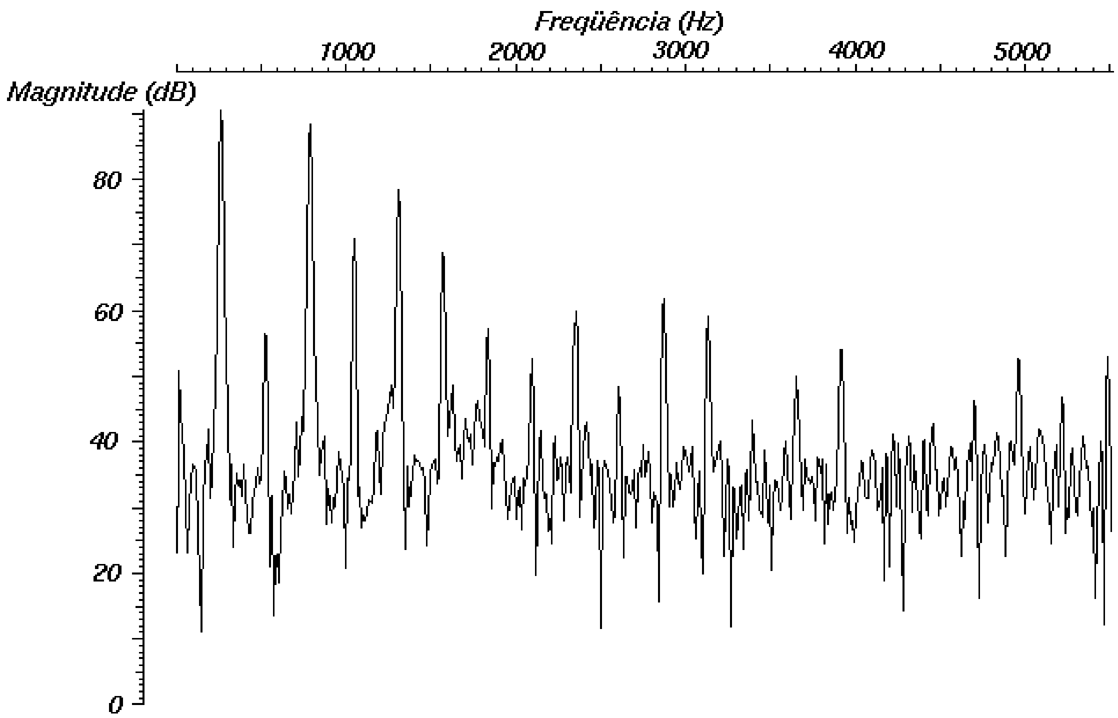

Figura 4.10: Espectro do refinamento ($N/2$) da transformada *Wavelet*, coeficientes **Chroma6**, do som Dó do clarinete em Si bemol (~ 258.4 Hz). Amostra do estado estacionário, $t = 45.8$ ms.

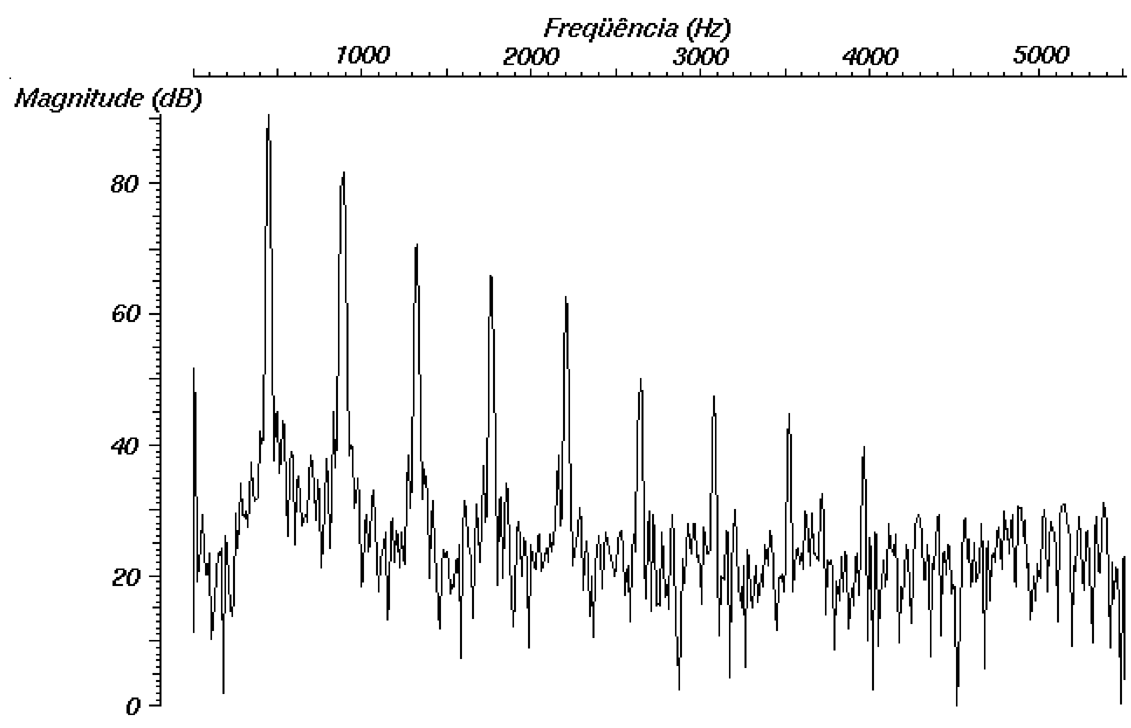

Figura 4.11: Espectro do som Lá da trompa em Fá ($\sim$ 441.4 Hz). Amostra do estado estacionário, $t = 45.8$ ms.

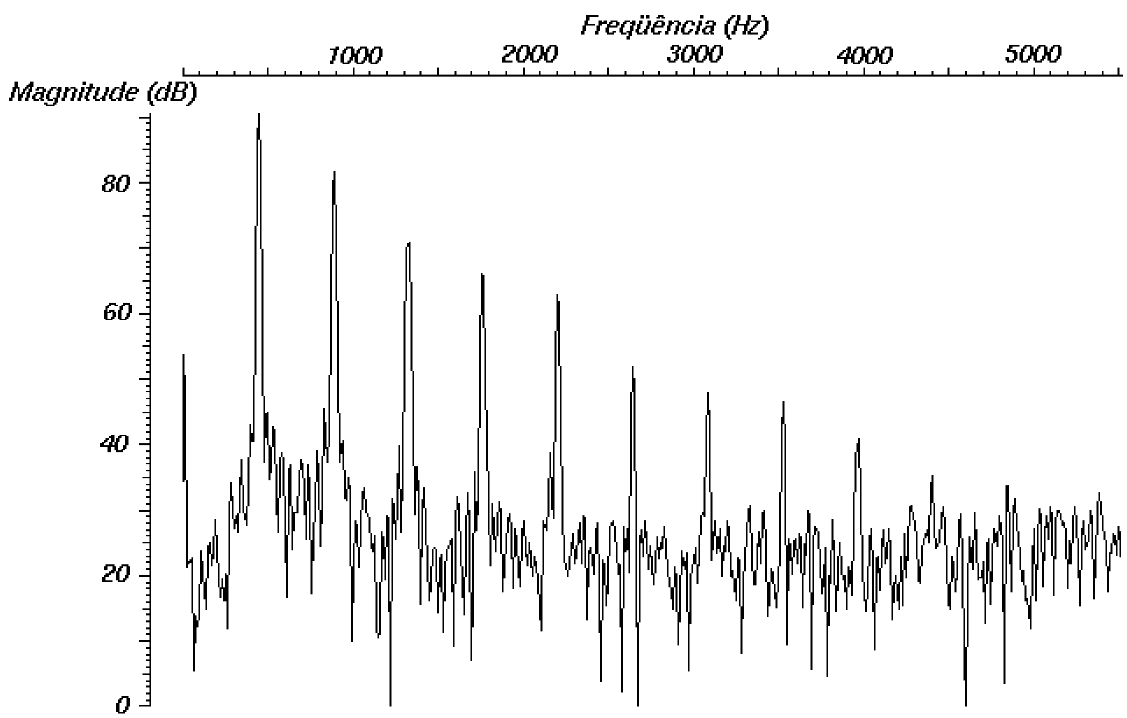

Figura 4.12: Espectro do refinamento $(N/2)$ da transformada *Wavelet*, coeficientes **Chroma4**, do som Lá da trompa em fá ($\sim$ 441.4 Hz). Amostra do estado estacionário, $t = 45.8$ ms.

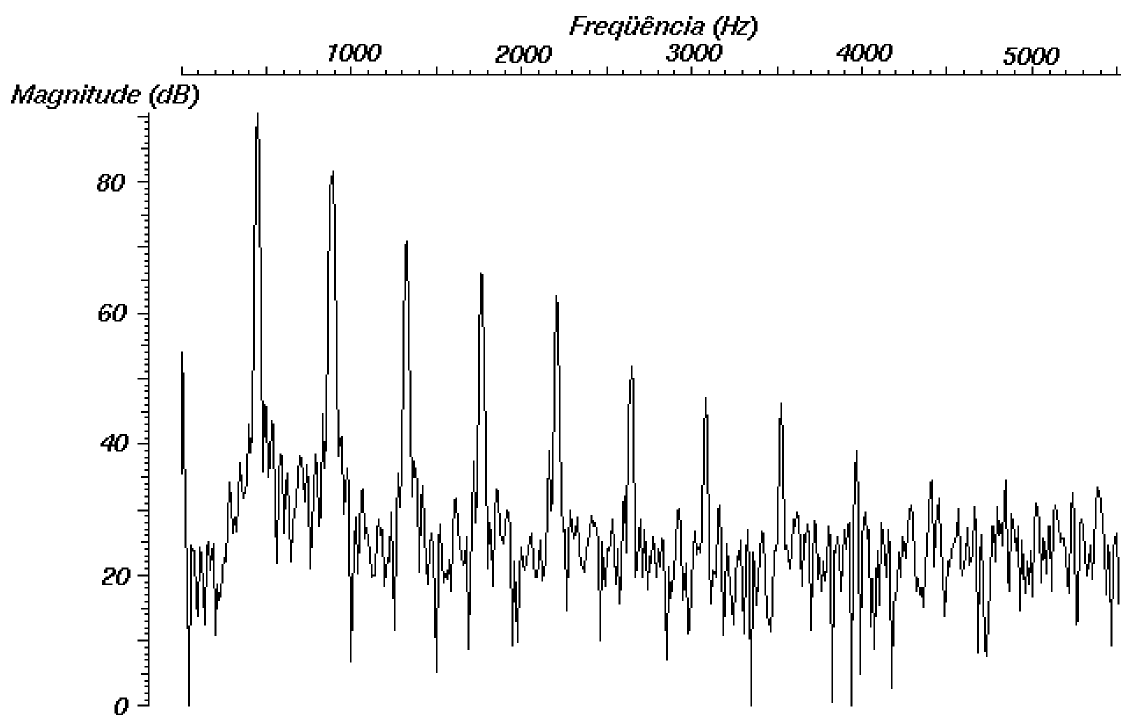

Figura 4.13: Espectro do refinamento ($N/2$) da transformada *Wavelet*, usando **Chroma6**, do som Lá da trompa em fá ($\sim$ 441.4 Hz). Amostra do estado estacionário, $t = 45.8$ ms.

4.3 Discussão dos Resultados

Apesar de muitas tentativas [11], a procura de métodos ideais de análise e reconstrução de sinais sonoros musicais não tem tido muito sucesso nos últimos anos. De acordo com Grey, "sem o controle separado de cada formante, não se pode garantir a natureza quase periódica na simulação dos sons orquestrais" [48]. Sabe-se que diferentes fatores determinam a qualidade do timbre de um instrumento. Primeiro o ataque ou o começo da nota que, como já foi mencionado, é uma característica crítica; segundo, outra importante característica tímbrica é a maneira como o som desaparece após ter sido emitido ou continua enquanto sustentado. Isso é uma das justificativas para a escolha do transiente de ataque dos instrumentos para testar o modelo proposto.

É necessário, além disso, fazer o estudo em janelas definidas de acordo com o sinal, bem como a pesquisa de coeficientes que, efetivamente, auxiliem na análise dessas janelas. Assim, como citado anteriormente, a diferença entre as bases da FFT e das *Wavelets* está no fato de que não existe um único conjunto de bases *Wavelets*, mas um conjunto infinito de bases, o que possibilita a escolha de coeficientes adequados quanto à sua localização compacta no espaço da função, dependendo de quão esparsos são no domínio *Wavelet*.

A escolha das bases de translação e dilação para o cálculo destes coeficientes pode permitir a localização de freqüências não componentes do espectro harmônico, mas que caracterizam o som de determinados instrumentos musicais [42].

A análise de Fourier dos escalogramas de timbres de instrumentos musicais revelou-se válida uma vez que, conforme as propriedades da aproximação $A(p)$ (cf. secção 2.5), verificou-se as seguintes conseqüências, já previstas na equação 2.68 [28]:

1. A translação de $\phi(t-k)$ reproduziu localmente as potências $t^0, t^1, \ldots, t^{p-1}$;

2. As funções refinadas puderam ser aproximadas com erro de ordem p para cada escala $h = 2^{-j}$, ou seja:

$$\|f(t) - \sum_k a_k \phi(t-k)\| \leq C \cdot 2^{-jp} \|f^{(p)}(t)\|; \tag{4.1}$$

3. Os primeiros p momentos da *Wavelet* ψ resultaram em zeros, já que:

$$\int_{-\infty}^{+\infty} t^m \psi(t) dt = 0 \tag{4.2}$$

   para $m = 0, 1, 2, \ldots, p-1$;

4. Os coeficientes *Wavelets* $\{c_k^j\}$ de função refinada tiveram uma dimidiação de ordem 2^{-p}, conforme previsto por:

$$2^{-j/2} c_k^j = \left| \int_{-\infty}^{+\infty} f(t) \psi(2^j t - k) dt \right| \leq C \cdot 2^{-jp}; \tag{4.3}$$

5. As funções escalonantes e *Wavelets* se tornam mais refinadas conforme p aumentou. Este refinamento se deu-se na ordem de $p/2$, aproximadamente (cf. apêndice B.4).

Capítulo 5

Conclusão

O principal escôpo deste trabalho foi determinar novos coeficientes aplicáveis à análise de timbres de instrumentos acústicos, a partir de subdivisões dos intervalos musicais, desenvolvendo os algoritmos para estudar sua viabilidade e eficácia na transformação, no refinamento do sinal e na localização de freqüências que os possam descrever.

Foram determinados dois conjuntos de coeficientes, aqui denominados **Chroma4** e **Chroma6** que foram estudados em suas capacidades de agirem como filtros *Wavelets* para decomposição e reconstrução de vários tipos de sinais, em diversas regiões de interesse. Também foi estudada a sua capacidade de reconstruir o sinal a partir dos segmentos passa-baixa em vários níveis de dimidiação, mostrando-se, assim, sua viabilidade na compressão dos dados.

Os testes foram realizados comparando-se os resultados dos espectros obtidos por transformadas localizadas de Fourier dos sinais originais e dos sinais recuperados, após seu tratamento com os coeficientes *Wavelets* estudados.

Uma vez que a soma dos coeficientes $h_k = \sqrt{2}$ corresponde ao intervalo de quarta aumentada[1] e, portanto, metade da oitava (1º harmônico), outros coeficientes podem ser descritos dentro deste intervalo. Os testes com os subconjuntos relativos ao intervalo cromático e sua diferença revelaram-se consistentes como pôde ser observado pelos resultados apresentados.

5.1 Validade dos Coeficientes

A validade dos coeficientes **Chroma4** e **Chroma6** depende de respeitarem as propriedades das *Wavelets*. Como visto no capítulo 2, $\psi(t)$ deve satisfazer a equação 5.1.

$$\int \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < +\infty \tag{5.1}$$

onde $\Psi(\omega)$ é a transformada de Fourier de $\psi(t)$. Assim, para validar os coeficientes, a transformada de Fourier de $\psi(t)$ deve desaparecer na freqüência zero, o que Valens [102] chama de "condição de admissibilidade". Isto leva à conclusão de que o valor médio da *Wavelet* no domínio do tempo deve ser zero,

$$\int \psi(t) dt = 0, \tag{5.2}$$

e, como nota Valens [102]:

> "e ainda deve oscilar, ou seja, $\psi(t)$ deve ser uma onda."

[1]Note que a quarta aumentada é o sexto intervalo cromático a partir de uma determinada freqüência, logo $2^{6/12} = 2^{1/2}$.

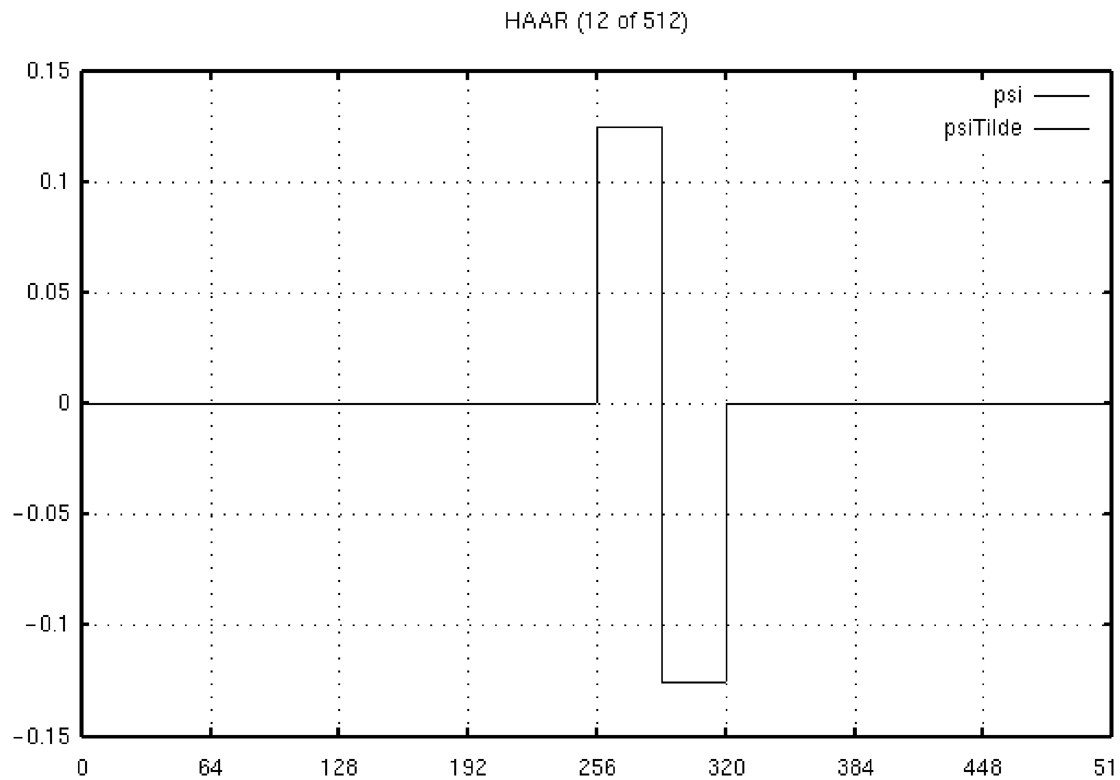

Figura 5.1: ψ dos coeficientes de Haar.

Assim, como pode-se ver na figura 5.1, os coeficientes de Haar satisfazem a equação 5.2, bem como os coeficients Daubechies4 e Daubechies6 mostrados nas figuras 5.2 e 5.3, respectivamente. Da mesma maneira, pode-se mostrar que os coeficientes **Chroma4** e **Chroma6** também satisfazem a condição da equação 5.2 e, portanto são válidos.

As figuras 5.4 e 5.5 mostram os valores de $\psi(t)$ para os coeficientes **Chroma4** e **Chroma6**.

De acordo com os resultados apresentados no capítulo 4, observa-se que há aplicabilidade e consistência dos coeficientes em sinais diferentes. Em todos eles, tanto o processo de transformação como a recuperação e o refinamento do sinal mostraram resultados bastante satisfatórios, o que viabiliza a utilização do modelo proposto para a análise de sinais

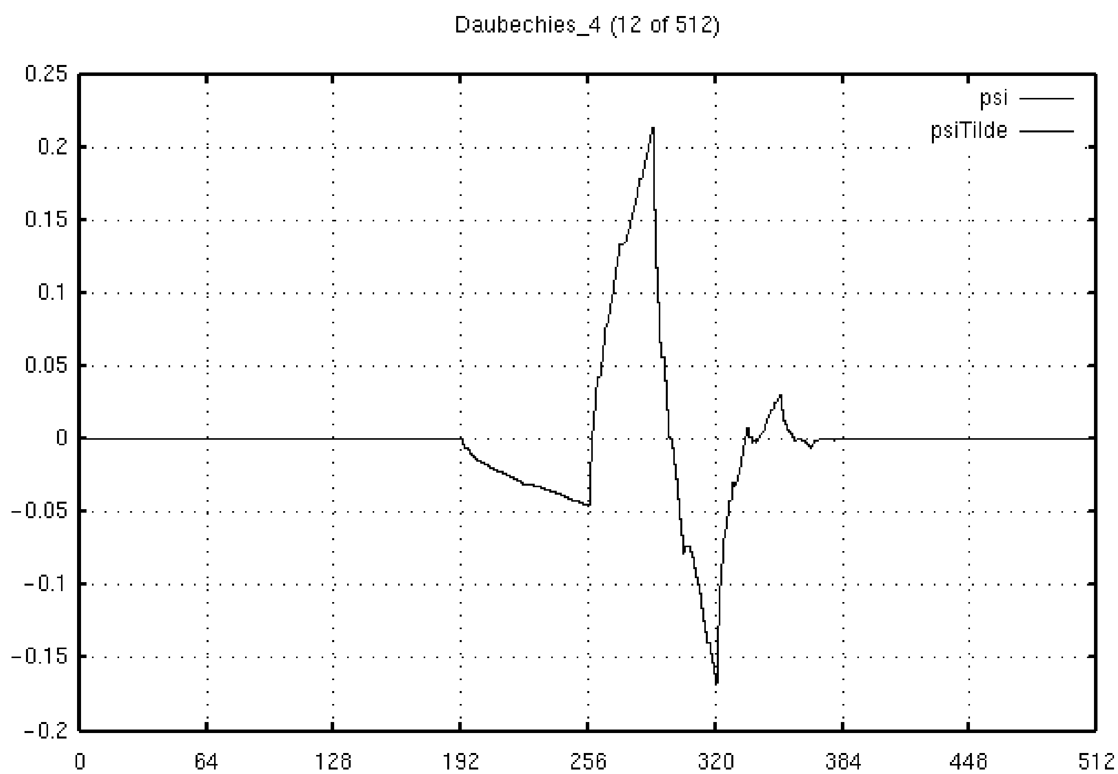

Figura 5.2: ψ dos coeficientes Daubechies4.

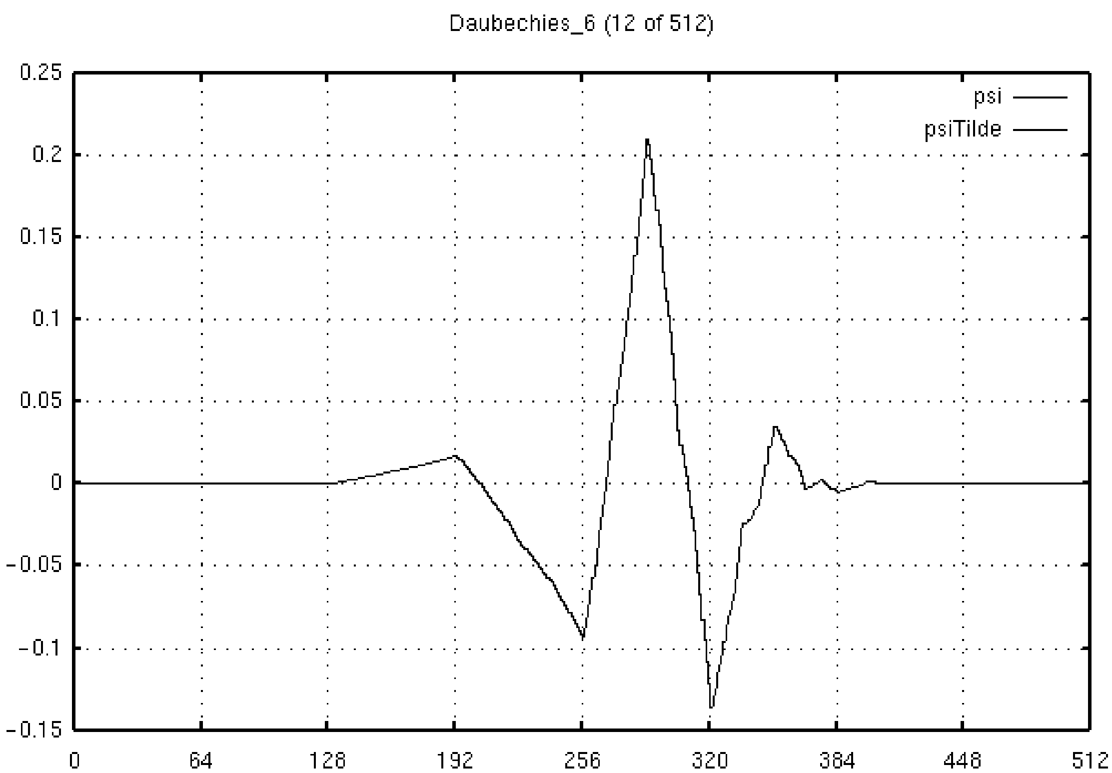

Figura 5.3: ψ dos coeficientes Daubechies6.

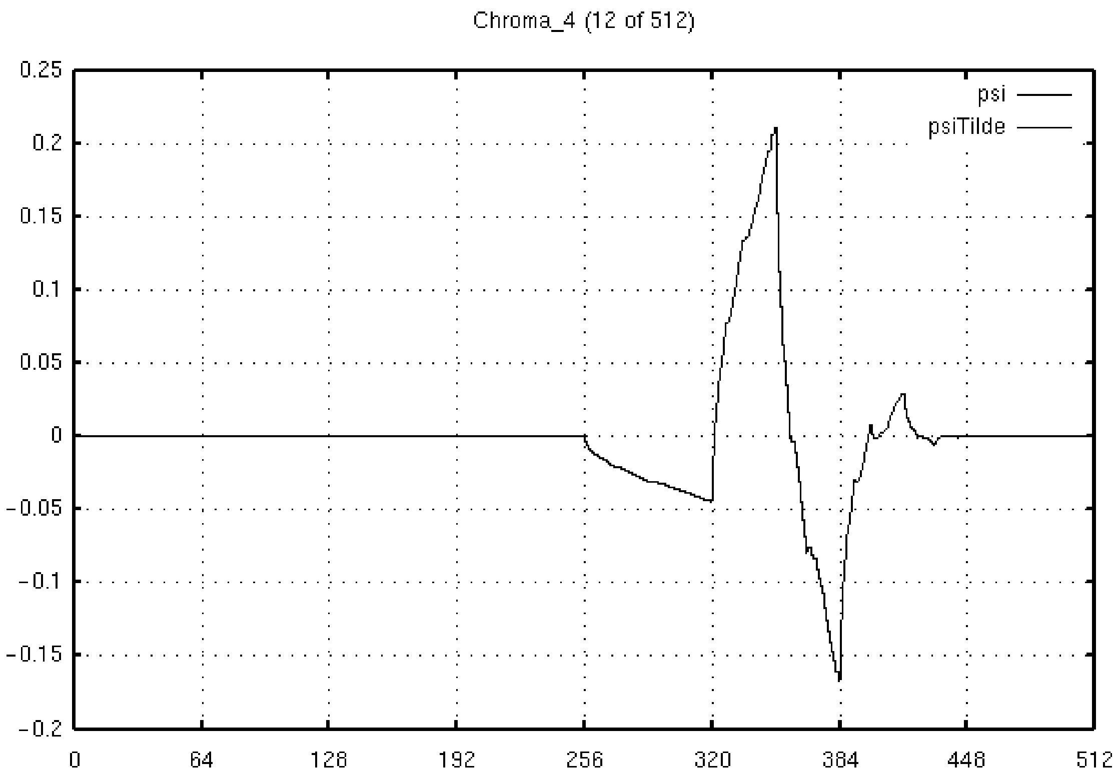

Figura 5.4: ψ dos coeficientes Chroma4.

sonoros de instrumentos musicais.

Pode-se afirmar que o modelo funciona com eficácia não apenas na transformada *Wavelet* direta, inversa e refinamento do sinal, mas também na identificação das freqüências características dos sons de instrumentos acústicos.

5.2 Coda

Não foi propósito deste trabalho fechar a questão sobre coeficientes aplicáveis à análise de sinais musicais. Pelo contrário, são abertas novas perspectivas que ainda devem ser estudadas em trabalhos de teses e dissertações, quais sejam:

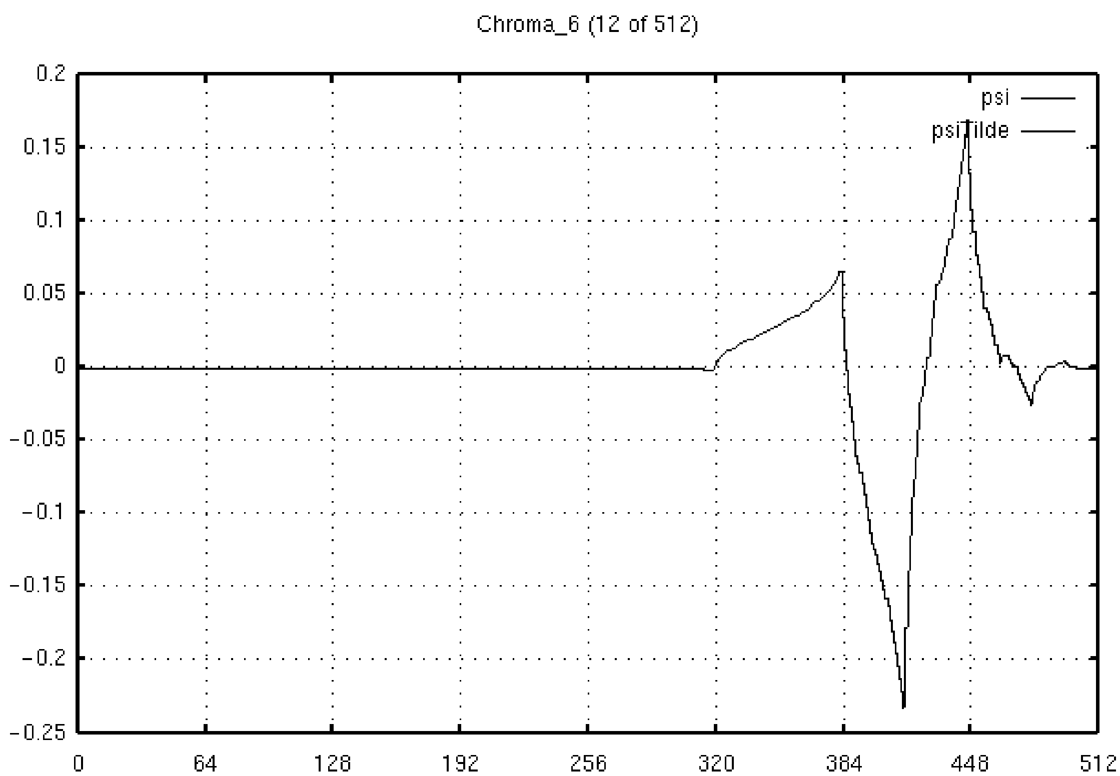

Figura 5.5: ψ dos coeficientes Chroma6.

1. Aplicação do modelo na identificação de freqüências não harmônicas para caracterização de instrumentos, na tentativa de captar a "assinatura" do sinal;

2. Outros modelos, utilizando os fatores dos intervalos musicais na identificação de timbres e de acordes entre vários instrumentos para aplicação na computação em música;

3. Utilização deste tipo de modelo para reconhecimento de freqüência na conversão de arquivos de áudio para protocolos MIDI;

4. Utilização de modelos semelhantes para estudo do deslocamento da centróide do sinal com a finalidade de localizar freqüências específicas.

Conclui-se, desta forma, que o trabalho realizado neste estudo termina por apresentar um novo modelo válido para análise de timbres de instrumentos musicais.

Finis laus Deo

Bibliografia

[1] ADELSON, E. H.; SIMONCELLI, E. e HINGORANI, R. "Orthogonal Pyramid Transforms for Image Coding", in: **Visual Communications and Image Processing II**, `ftp://ftp.cis.upenn.edu/pub/eero/adelson87.ps.Z`, *Proc. SPIE*, Vol. 845, pp. 50-58, 1987.

[2] AGUIAR, C. "Modelling the Excitation Function to Improve Quality in LPC's Resynthesis." **Anais do XIV Congresso da Sociedade Brasileira de Computação**, I Simpósio Brasileiro de Computação e Música - Caxambú - MG, 1994.

[3] ANGELO, JR., E. J. "A Tutorial Introduction to Digital Filtering." *The Bell TechnicalJournal*, Vol. 60, n^o 7, pp. 1499–1546, 1981.

[4] ANTONINI, M.; BARLAUD, M.; MATHIEU, P. e DAUBECHIES, I. "Image Coding Using Wavelet Transforms", *IEEE Trans. on Image Processing*, Vol. 1 N^o 2, 1992.

[5] ARISTÓTELES. (384–322 A.C.) **Physica**, trad. R. P. Hardie e R. K. Gaye, 1952, ISBN: 0-852229-163-9, Univ. of Chicago Press, Chicago, Vol. 2, Cap. 2, p. 270 ff, 1952.

[6] ARISTÓTELES. (384–322 A.C.) **Metaphysica**, trad. W. D. Ross, 1952, ISBN: 0-

852229-163-9, Univ. of Chicago Press, Chicago, Vol. 11, Cap. 6, p. 591; Vol. 12 Cap. 8, p. 603, 1952.

[7] AUSHER, P.; WEISS, G. e WICHERHAUSER, M. V. "Local Sine and Cosine Bases of Coifman and Meyer and the Construction of Smooth Wavelets." in: **Wavelets — a Tutorial**, C. K. Chui (Ed.), New York: Academic Press, Inc., pp. 237–256, 1991

[8] BEYLKIN, G.; COIFMAN, R. R. e ROKHLIN, V. "Fast Wavelet Transforms and Numerical Algorithms I", *Comm. Pure Appl. Math*, v. 44, (`ftp://pascal.math.yale.edu/pub/wavelets/papers/bcr.tex`)

[9] BRIGHAM, E. O. **The Fast Fourier Transforms**, Prentice Hall, Englewood Cliffs, NJ, 1975.

[10] BROWN, J. C. "Calculations of a Q Constant Spectral Transform." *J. Acoust. Soc. Am.*, vol. 89, n^o 1, (Jan. 1991), pp. 425–434, 1991.

[11] BROWN, J. C. "Musical Fundamental Frequency Tracking using a Pattern Recognition Method." *J. Acoust. Soc. Am.*, vol. 92, n^o 3, (Sep. 1992), pp. 1394–1402, 1992

[12] BROWN, J. C. e PUCKETTE, M. S. "An Efficient Algorithm for the Calculation of a Constant Q Transform." *J. Acoust. Soc. Am.*, vol. 92, n^o 5, (Nov. 1992), pp. 2698–2701, 1992.

[13] BUSONI, F. **Sketch of a New Aesthetic of Music**, New York, 1911

[14] BURT, J. e ADELSON, E. R. The Laplacian Pyramid as a Compact Image Code. *IEEE Trans. Comm.*, COM-31, pp. 532-540, 1983.

[15] CANN, R. An "Analysis/Synthesis Tutorial", In: **Foundation of Computer Music**, Curtis Roads & John Strawn (Eds.), Cambridge, Mass.: MIT Press, pp. 114–144, 1988.

[16] CHEN, C. S.; HUO, K. S. "Karhunen-Loéve Method for Data Compression and Speech Synthesis", *IEEE Proceedings-I*, vol. 138 n^o 5, pp. 377–380, 1991.

[17] CHOI, A. "On Improvement of the Real-Time Performance of Two Fundamental Frequency Recognition Algorithms." in: **Anais do XIV Congresso da SBC**, I Simpósio Brasileiro de Computação e Música, Caxambú, MG, 1994.

[18] CHOWNING, J. M. "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation." *J. Audio Eng. Soc.*, vol. 21 n^o 7, pp. 526–534, 1973.

[19] CHOWNING, J., e BRISTOW, D. **FM Theory & Applications, By Musicians For Musicians.** Tokyo: Yamaha Music Foundation, 195 pp., 1986.

[20] CODY, M. A. "The Wavelet Transform Beyond Fourier Transforms." *Dr. Dobb's Journal*, M & T Publishing Inc., April, 12 pp., 1992.

[21] CODY, M. A. "A Wavelet Analyser. An Alternative to FFT-based Spectrum Analyser." *Dr. Dobb's Journal*, M & T Publishing Inc., April, 9 pp., 1993.

[22] CODY, M. A. "The Wavelet Packed Transform Extending the Wavelet Transform." *Dr. Dobb's Journal*, M & T Publishing Inc., April, 9 pp., 1994.

[23] COIFMAN, R. R.; MEYER, Y. e WICKERHAUSER, M. V. "Wavelet Analysis and Signal Processing." *Tech. Rep.*, Yale University, New Haven, `ftp://ceres.math.yale.edu/pub/`, 29 pp., sem data.

[24] COIFMAN, R. R. e WICKERHAUSER, M. V. "Wavelets and Adapted Waveform Analysis." *Tech. Rep.*, Yale University, New Haven, `ftp://ceres.math.yale.edu/pub/`, 33 pp., sem data.

[25] COOLEY, J. W.; TUKEY, J. W. "An Algorithm for the Machine Calculation of Complex Fourier Series." *Math of Computation*, vol. 19, pp. 297–301, 1965.

[26] COPERNICUS, N. (1543). **De revolutionibus orbitum coelestium**, Nüremberg, trad. Charles Glenn Wallis, 1952, University of Chicago Press, Chicago, Livro I, p. 510, 1952.

[27] DAUBECHIES, I. "Orthonormal Bases of Compactly Supported Wavelets." *Comm. Pure Appl. Math.*, Vol. 41, pp.906–966, 1988.

[28] DAUBECHIES, I. **Ten Lectures on Wavelets**, CBMS-NFS Regional Conf. Series in Appl. Math., Vol 61. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[29] DAUBECHIES, I. "Where do Wavelets Come From? — A Personal Point of View." *Proc. of IEEE*, vol. 84, no. 4, 1996.

[30] DAUBECHIES, I. "Discrete series of coherent states and their use in signal analysis." *Tech. Rep.*, Courant Institute of Mathematical Sciences, New York, sem data.

[31] DeFATTA, D. J.; LUCAS, J. G.; HODKISS, W. S. **Digital Signal Processing: A System Design Approach**, New york: John Willey & Sons, 1988.

[32] DE LYON, B. "Ondelettes Orthogonales et Biorthogonales." *Tech. Rep.*, Inst. de Recherche Informatique et Syst. Alatoires (IRISA), Publ. Int. nº 732, 34 pp., 1996.

[33] DE LYON, B. e JUDITSKY, A. "On the Computation of Wavelet Coefficients." *Tech. Rep.*, Inst. de Recherche Informatique et Syst. Alatoires (IRISA), Publ. Int. nº 856, 31 pp., 1997.

[34] DeVORE, R. e LUCIER, B. J. "Wavelets." *Acta Numerica* (no prelo), `http://www.amara.com/`, sem data.

[35] DI SCIPIO, A.; PRIGNANO, I. "The Synthesis of Complex Sonic Events by Functional Iterations." in: **Anais do XIV Congresso da SBC**, I Simpósio Brasileiro de Computação e Música, Caxambú, MG, pp. 15–20, 1994.

[36] DOERSCHUK, R. L. "The Life and Legacy of Leon Theremin." *Keyboard*, vol. 20, nº 2, pp.48–68, 1994.

[37] DOVICCHI, J. C. "Altering Digital Sound Output Through MIDI's Non Registered Parameter Number (NRPN): A Case Directed Research on Synthesized Winds and Strings Envelope." **Anais do XIV Congresso da SBC**, I Simpósio Brasileiro de Computação e Música, Caxambú - MG, 8 pp., 1994.

[38] DOVICCHI, J. C. "Compilação Acústica através de Ray-Tracing." in: **II Encontro Internacional de Música Eletroacústica**, Univ. de Brasília, Brasília, 8 pp., 1997.

[39] DOVICCHI, J. C.; CARRIJO, G., LIMA, L. "Fourier Transforms and Wavelets: a Combined Technique for Musical Signal Analysis." *Proceedings of the Conference on Mathematical and Computational Methods in Music, Diderot'99.*, Universidade de Viena, Viena, 13 pp., 1999.

[40] DOVICCHI, J. C.; CARRIJO, G., LIMA, L. "Music Based Coefficients for Wavelets

Transform: a Contribution to Musical Signal Analysis." aceito na 108ª Convenção da Audio Eng. Soc., **108th AES Convention**, Paris, Fev. 2000.

[41] EDWARDS, T. "Discrete Wavelet Transform: Theory and Implementation." *Tech. Rep.*, Stanford University, 1991.

[42] EVANGELISTA, G., "Pitch Synchronous Wavelet Representations of Speech and Musical Signals." *IEEE Trans. on Signal Proc.*, 41 (12), 3313-3330, 1993.

[43] FLETCHER, H. "The Physical Criterion for Determining the Pitch of a Musical Tone." *Phys. Rev.*, XXIII, 1924.

[44] GABOR, D. "Acoustical Quanta and the Theory of Hearing." *Nature*, 159 (4044), pp. 591–594, 1947.

[45] GOEL, P. e VIDACOVIĆ "Wavelet Transformation as Diversity Enhancers." *Tech. Rep.*, Duke University, 21 pp., `brani@isds.duke.edu`, sem data.

[46] GOOTMAN, E. C. e WIRCKENHAUSER, M. V. "Elementary Wavelets." *Tech. Rep.*, Department of Mathematics, University of Georgia, Athens, Georgia, sem data.

[47] GRAPS, A. "An Introduction to Wavelets." *IEEE Computational Science and Engineering*, vol. 2, nº 2, Summer 95, 1995.

[48] GREY, J. M. **An Exploration of Musical Timbre.** Ph.D. Diss., Dept. Psychology, Stanford Univ., Stanford, CA, 1975.

[49] GRIFFITHS, P. **A Música Moderna: Uma História Concisa e Ilustrada de Debussy a Boulez**, Rio de Janeiro: Jorge Zahar Ed., 1987.

[50] GROSS, L. **Harmonic Analysis on Hilbert Space**, Am. Math. Soc., AMS, 62 pp., 1963

[51] GROSSMAN, A. e MORLET, J. "Decomposition of Hardy functions into square-integrable wavelets of constant shape." *SIAM J. Math. Anal.*, vol. 15, pp. 723–736, 1984.

[52] HEIL, C. E. e Walnut, D. F. "Continuous and Discrete Wavelet Transform." *SIAM Rev.*, 31, pp. 628-666, 1989.

[53] HUSMANN, H. **Einleitung in die Musikwissenschaft**, Heinrichshofen's Verlag, Wilhelmshaven, pp. 13–92, 1975.

[54] HUTCHINS, B. A. "Aplications of real-time Hadamard Transform Network to Sound Synthesis." *Journal of Audio Engeneering Society*, vol. 23 *nº* 7, sept. 1975, pp. 558–562, 1975.

[55] KEPLER, J. (1620). **Epitomes Astronomiae Copernicanae Liber Quartus**, Linz am Donau, trad. Charles Glenn Wallis, 1952, Univ. of Chicago Press, Chicago, Livro IV, pp. 846–933, 1952.

[56] KEPLER, J. **Harmonices Mundi Libri**, trad. Charles Glenn Wallis, 1952, Univ. of Chicago Press, Chicago, pp. 1016–1050, 1952.

[57] KUC, R. **Introduction to Digital Signal Processing.**, McGraw Hill Book Co., New York, 474 pp., 1982.

[58] LEMARIÉ, P. G. "Continuité sur les espaces de Besov des opérateurs définis par les intégrales singulières." **Annales de l'Institute Fourier**, *appud*: Mallat, S.G.

(1989). *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, 1989.

[59] LANSKY, P. "Compositional Application of Linear Predictive Coding." In: **Current Directions in Computer Music Research**. ed. Max V Mathews and John Pierce, Mass.: MIT Press, p. 5–8, 1989.

[60] LAUTERBORN, W.; PARLITZ, U. "Methods of Chaos Physics and their Application to Acoustics" *J. Acoust. Soc. Am.*, vol. 84 n^o 6, pp. 1975–1993, 1988.

[61] LEBRUN, M. "A Derivation of Spectrum of FM with a Complex Modulating Wave." *Computer Music Journal*, vol. 1 n^o 4, pp. 51–52, 1977.

[62] MALLAT, S. A "Theory for Multiresolution Signal Decomposition: The Wavelet Representation." *IEEE Trans. Pattern Anal. Machine Intelligence*, 11, pp. 674-693, 1989.

[63] MANZOLLI, J. **Non-Linear Dynamics and Fractals as a Model for Sound Synthesis**, PhD diss. , University of Nottingham, England, 1993.

[64] MANZOLLI, J. "FracWave: Non-Linear Dynamics as Timbral Constructs." **Anais do I Simpósio Brasileiro de Música e Computação**, XIV Congresso da SBC, Caxambú, MG., pp. 45–51, 1994.

[65] MARAR, J. F. **Utilização da Transformada Karhunen-Loéve em Síntese de Tons Musicais**. Diss. Mestrado, USP, São Carlos, 1992.

[66] MARAR, J. F.; MOREIRA, E. S. "Síntese Aditiva de Tons Musicais através da Transformada de Karhunen-Loéve." **Anais do I Simpósio Brasileiro de Música e Computação**, XIV Congresso da SBC, Caxambú, MG., pp. 9–14, 1994.

[67] MATHEWS, M. V.; KOHOUT, J. "Eletronic simulation of violin." *J. Acoust. Soc. Amer.*, vol. 53, pp. 1620–1626, 1973.

[68] MEYER, Y. "Principe d'incertitude, bases hilbertiennes et algèbres d'operateurs." *Séminaire Bourbaki*, vol. 662, 1986.

[69] MEYER, Y. **Wavelets Algorithms & Aplications**, trad. Robert D. Ryan, 2^a Ed. (1994), Philadelphia: SIAM - Society for Industrial and Applied Mathematics, 1993.

[70] MOOG, R. "Oppening Conference." in: **III Anual CyberArts Convention**, *appud.* Mark Vail, "Controller Odities." *Keyboard*, vol. 19, n^o 3, pp. 44–49, (Mar. 1993), 1992.

[71] MOORER, J. A. "The Synthesis of Complex Audio Spectra by Means of Discrete Summation Formulae." *J. Audio Eng. Soc.*, vol. 24, n^o 9, pp. 717–727, Nov. 76, 1976.

[72] MOORER, J. A. "Signal Processing Aspects of Computer Music: A Survey." *Proceedings of the IEEE*, vol. 65 n^o 8, August 1977, pp. 1108–1137, 1977.

[73] PERRIER, V.; PHILIPOVITCH, T. e BASDEVANT, C. "Wavelet Spectra Compared to Fourier Spectra." *Tech. Rep.*, Lab. de Météorologie Dynamique, Paris, sem data.

[74] PESQUET, J-C. "Time Invariant Orthonormal Wavelet Representation." *IEEE Transactions on Signal Processing*, Vol. 41, n^o 8, 1964–1970, 1996.

[75] PETERSEN, T. L. "Vocal Tract Modulation of Instrumental Sounds by Digital Filtering." In: **Proceedings of the Second Annual Music Computation Conference**, J. Beauchamp and J. Melby (eds.), Part I: Synthesys Techniques. Urbana: Office of Continuing Education, University of Illinois, 1975.

[76] PIELEMEIER, W. J.; WAKEFIELD, G.H.; e SIMONI, M. H. “Time-Frequency Analysis of Musical Signals.” *Proceedings of IEEE*, vol. 84, n^o 9, pp. 1216–1230, 1996

[77] PLATÃO. (428–348 A.C.) **The Dialogues of Plato: Gorgias.**, trad. Benjamin Jowett, 1952, Univ. of Chicago Press, Chicago, p. 245, 1952.

[78] PLATÃO. (428–348 A.C.) **The Dialogues of Plato: Republic**. trad. Benjamin Jowett, 1952, Univ. of Chicago Press, Chicago, Vol. 7, pp. 391–398, 1952.

[79] PLATÃO. (428–348 A.C.) **The Dialogues of Plato: Laws**, trad. Benjamin Jowett, 1952, Univ. of Chicago Press, Chicago, Vol 7, pp. 728–730, Vol. 12, pp. 797–798, 1952.

[80] POLLEN, D. “Parametrization of Compactly Supported Wavelets.” *Tech. Rep.*, Aware Inc., 1989

[81] PRINZ, P. “Calculating the Dual Gabor Window for General Sampling Sets.” *IEEE Transaction on Signal Processing*, Vol. 44, n^o 8, pp. 2078–2082, 1996.

[82] PROAKIS, J. G. e MANOLAKIS, D. G. **Introduction to Digital Signal Processing**, Macmillan Publ. Co., New York, 1988.

[83] PTOLEMAEUS, C. (100-178 A.D.). **The Almagest**, trad. R. Catesby Taliaferro, 1952, Univ. of Chicago Press, Chicago, Livro I, pp. 5–6, 1952.

[84] RABINER, L. R. “Techniques for Designing Finite Impulse-Response Digital Filters.” *IEEE Trans. Commun. Technol.*, vol. COM-19, pp. 188–195, apr. 1991.

[85] REISSEL, L-M. “Multiresolution Geometric Algorithms Using Wavelets I: Representa-

tion for Parametric Curves and Surfaces." *Tech. Rep.* University of Britsh Columbia, TR 93-17, 33 pp., 1993.

[86] RISSET, J. C. "Computer Study of Trumpet Tones." in: **Bell Laboratories Report**, Murray Hill, Bell Labs, New Jersey, 1966.

[87] ROADS, C. "Granular Synthesis of Sound." in: **Foundations of Computer Music**, C. Roads & J. Strawn (eds.), Cambridge, Mass.: MIT Press, pp. 145–159, 1988.

[88] ROADS, C.; STRAWN, J. (EDS.) **Foundations of Computer Music**, Cambridge, Mass.: MIT Press, 1988.

[89] SCHAFFER, R. W. **Echo Removal by Discrete Generalized Linear Filtering**, PhD Diss., MIT, Cambridge, MA, MIT RLE Tech. Rep. 466, 1969.

[90] SCHAFFER, R. W.; RABINER, L. R. "Digital Signal Processing Approach to Interpolation." *Proceedings of the IEEE*, vol. 61 *n^o^* 6, June 1973, pp. 692–702, 1973.

[91] SCHOTTSTAEDT, B. "The Simulation of Natural Instruments Tones Using Frequancy Modulation with a Complex Modulating Wave." *Computer Music Journal*, vol. 1 n^o 4, pp. 46–50, 1977.

[92] SEASHORE, C. E. **Psychology of Music**, New York: Dover, republicação da edição de 1938 da McGraw Hill Co. Inc., 408 pp., 1967

[93] SELESNICK, I. W. e BURRUS, C. S. "Automatic Generation of Prime Lenght FFT Programs." *Tech. Rep.*, Rice University, Houston, TX, 25 pp., 1994.

[94] SMALLEY, D. "Spectro-morphology and Structuring Processes. in: **The Language**

of Electroacustic Music, S. Emmerson (ed.), London: McMillan Press Ltd., pp. 61–63, 1986.

[95] SNOW, W. B. “Change of Pitch with Loudness at Low Frequencies.” *J. Acoust. Soc. Am.*, 8, pp. 14–19, 1936

[96] SOUNDERS, S. “Improved FM Audio Synthesis Methods for Real-Time Digital Music Generation.” *Computer Journal of Music*, vol. 1 n^o 1, pp. 53–55, 1977

[97] STAPLETON, J. C.; BASS, S. C. “Synthesis of Musical Tones Based on the Karhunen-Love Transform.” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36 n^o 3, March 1988, pp. 305–319, 1988.

[98] STRANG, G. “Wavelets and Dilation Equations: a Brief Introduction.” *SIAM Rev.*, 31, pp. 613–627, 1989.

[99] STRINTZIS, M. G. “Optimal Biorthogonal Wavelet Bases for Signal Decomposition.” *IEEE Transaction on Signal Processing*, Vol. 44, n^o 6, pp. 1406–1417, 1996.

[100] SWELDENS, W. e PIESSENS, R. “Wavelet Sampling Techniques.” in: **Joint Statistical Meetings**, S. Francisco, (separata), 11 pp., 1993.

[101] SWELDENS, W. **The Construction and Application of Wavelets in Numerical Analysis.**, Tese, Columbia University, 1995.

[102] VALENS, C. **The Fast Lifting Wavelet Transform and its Application to Maritime Radar Image Processing.**, Tese, Technishe Universiteit Eindhover, Holanda, 1998.

[103] VIDAKOVIĆ, B.; MÜLLER, P. "Wavelets for Kids: a Tutorial Introduction." **Tech. Report**, Institute of Statistics and Decision Sciences, Duke University, Durham.

[104] WICHERHAUSER, M. V. "Lectures on Wavelet Packet Algorithms." *Tech. Rep.*, `ftp://ceres.math.yale.edu/pub/`, Dept. Math. University of Washington, St. Louis, 75 pp., 1991.

[105] WICKERHAUSER, M. V. "Time Localization for Wavelet Transform." *Tech. Rep.*, `ftp://ceres.math.yale.edu/pub/`, Dept. Math. University of Washington, St. Louis, 18 pp., 1994

[106] WICKERHAUSER, M. V. "Acoustic Signal Compression with Wave Packets." *Tech. Rep.*, `ftp://ceres.math.yale.edu/pub/`, Dept. Math., Yale University, New Haven, 15 pp. sem data.

[107] WINOGRAD, S. On "Computing the Discrete Fourier Transform." *Math. Comp.* 32., 1978

Apêndice A

Exemplos Matemáticos

A.1 Exemplos de Espaços de Hilbert

1. Espaço C^n de *Hilbert*.

   Seja C^n uma n-upla de números complexos, α um escalar, $x = \{x_1, x_2, \cdots, x_n\}$ e $y = \{y_1, y_2, \cdots, y_n\}$, então:

$$(x + y) = (x_1 + y_1, x_2 + y_2, \cdots, x_n + y_n) \tag{A.1}$$

$$\alpha x = (\alpha x_1, \alpha x_2, \cdots, \alpha x_n) \tag{A.2}$$

$$< x, y >= (x_1\overline{y_1} + x_2\overline{y_2} + \cdots + x_n\overline{y_n}) \tag{A.3}$$

$$\|x\| = (|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2) \tag{A.4}$$

2. Espaço $L^2[0, 2\pi]$ de *Hilbert*.

   Seja um conjunto de funções de *Lebesgue* complexas f, no intervalo de $[0, 2\pi]$, tal que:

$$\int_0^{2\pi} |f(t)|^2 dt < \infty, \tag{A.5}$$

   se f e g pertencem a $L^2[0, 2\pi]$, e α é um escalar:

$$(f + g)(t) = f(t) + g(t) \tag{A.6}$$

$$(\alpha f)(t) = \alpha f(t) \tag{A.7}$$

$$< f, g >= \frac{1}{2\pi} \int_0^{2\pi} f(t)\overline{g(t)} dt \tag{A.8}$$

$$\|f\| = \sqrt{\frac{1}{2\pi} \int_0^{2\pi} |f(t)|^2 dt} \tag{A.9}$$

3. Espaço $L^2[-\infty, +\infty]$ de *Hilbert*.

Seja um conjunto de funções de *Lebesgue* complexas f, no intervalo de $[-\infty, +\infty]$, tal que:

$$\int_{-\infty}^{+\infty} |f(t)|^2 dt < \infty, \tag{A.10}$$

se f e g pertencem a $L^2[-\infty, +\infty]$, e α é um escalar:

$$(f+g)(t) = f(t) + g(t) \tag{A.11}$$

$$(\alpha f)(t) = \alpha f(t) \tag{A.12}$$

$$< f, g >= \int_{-\infty}^{+\infty} f(t)\overline{g(t)}dt \tag{A.13}$$

$$\|f\| = \sqrt{\int_{-\infty}^{+\infty} |f(t)|^2 dt} \tag{A.14}$$

4. Espaço $L^2[S]$ de *Hilbert.*

Seja A um conjunto arbitrário de números complexos de soma quadrática, $\alpha \longmapsto c_\alpha$, tal que:

$$\sum_{\alpha} |c_\alpha|^2 < \infty \tag{A.15}$$

A.2 Demonstração do Teorema 2.4.1

Para demonstrar o Teorema Fundamental pode-se dividí-lo em três Lemas, e considerando o caso $j = 0$ sem perda de generalidade.

Lema A.2.1 *A função $\phi(t)$ forma um sistema ortonormal em V_l.*

Demonstração A.2.1 *Pela definição de ψ pode-se escrever:*

$$\begin{aligned} \psi(t-l) &= \sum_k (-1)^k \overline{h_{1-k}} \sqrt{2} \phi(2t - 2l - k) \\ &= \sum_n (-1)^{n-2l} \overline{h_{1+2l-n}} \sqrt{2} \phi(2t - n) \end{aligned}$$

onde $n = 2l + k$. Consequentemente, para quaisquer dois inteiros l e m,

$$\left\langle \psi_m^0, \psi_l^0 \right\rangle = \sum_n h1 + 2l - n\overline{h1 + 2l - n}$$

Fazendo-se $u = 1 + 2l - n$ e $s = l - m$ e lembrando as condições estabelecidas para os coeficientes do filtro Wavelet (veja 2.3.2), pode-se escrever:

$$\left\langle \psi_m^0, \psi_l^0 \right\rangle = \sum_u h_u \overline{h_{u+2s}} = \delta(0, s) = \delta(m, l)$$

Lema A.2.2 *Sejam as finções $\phi(t)$ e $\psi(t)$ sistemas ortonormais em V_l, os coeficientes de $\phi(t)$ e $\psi(t)$ são ortogonais.*

Demonstração A.2.2 *Usando-se a identidade de Parseval e as representações de $\phi(t)$ e $\psi(t)$ em termos de $\left\{\phi_k^l(t)\right\}$ de V_l, podemos escrever:*

$$\begin{aligned}
\left\langle \phi_l^0 \psi_m^0 \right\rangle &= \sum_n (-1)^n h_{n-2l} h_{1+2m-n} \\
&= \sum_u (-1)^u h_u h_{1-u+2s} \\
&= \sum_{u \geq s+1} (-1)^u h_u h_{1-u+2s} + \sum_{u \leq s} (-1)^u h_u h_{1-u+2s} \\
&= 0
\end{aligned}$$

onde $u = n = 2l$ e $s = m - l$.

Lema A.2.3 *Os coeficientes de $\phi(t)$ e $\psi(t)$ formam, também, uma base de um sistema ortonormal de V_l.*

Demonstração A.2.3 *Para demonstrar isto, basta provar que cada ϕ_n^l é a soma de suas séries de Fourier relativas aos coeficientes de ϕ e ψ.*

A prova, no caso $n = 0$ pode ser generalizada. Tomando-se a equação de dilação e a definição de ψ temos:

$$\phi(t-k) = \sum_l h_l \sqrt{2} \phi(2t - 2k - l) \qquad \text{e}$$

$$\psi(t-k) = \sum_l (-1)^l \overline{h_{1-l}} \sqrt{2} \phi(2t - 2k - l) \qquad \text{, logo}$$

$$\left\langle \phi(t-k), \sqrt{2}\phi(2t) \right\rangle = h_{-2k} \qquad \text{e}$$

$$\left\langle \psi(t-k), \sqrt{2}\phi(2t) \right\rangle = (-1)^{-2k}\overline{h_{1+2k}} = \overline{h_{1+2k}}.$$

Além disso, a série de Fourier de $\phi(2t)$ é dada por:

$$\frac{1}{\sqrt{2}}\sum_k \overline{h_{-2k}}\phi(t-k) + \frac{1}{\sqrt{2}}\sum_k h_{1+2k}\psi(t-k) \qquad \text{ou}$$

$$\sum_k\sum_l \overline{h_{-2k}}\phi(2t-2k-l) + \sum_k\sum_l (-1)^l h_{1+2k}\overline{h_{1-l}}\phi(2t-2k-l) \qquad \text{ou, ainda,}$$

$$\sum_u \phi(2t-u)\sum_{2k+l=u}\overline{h_{-2k}}h_l + \sum_u \phi(2t-u)\sum_{2k+l=u}(-1)^l h_{1+2k}\overline{h_{1-l}}.$$

e tomarmos $F(u)$ e $G(u)$ tal que:

$$\begin{aligned} F(u) &= \sum_{2k+l=u}\overline{h_{-2k}}h_l \\ &= \sum_{\gamma \mathrm{par}}\overline{h_\gamma}h_{\gamma+u} \end{aligned}$$

$$\begin{aligned} G(u) &= \sum_{2k+l=u} (-1)^l h_{1+2k} \overline{h_{1-l}} \\ &= \sum_{\gamma \text{impar}} (-1)^u h_\gamma \overline{h_{\gamma-u}} \\ &= \sum_{\delta+u \text{impar}} (-1)^u h_{\delta+u} \overline{h_\delta}, \end{aligned} \tag{A.16}$$

a série de Fourier pode ser escrita como:

$$\sum_u (F(u) + G(u))\, \phi(2t-u).$$

Tomando-se u separadamente, ímpar ou par, no caso de u ímpar, tem-se $F(u)+G(u) = 0$ *e, no caso de u par, tem-se:*

$$F(u) + G(u) = \sum_\beta h_\beta \overline{h_{\beta+u}},$$

que assumirá valores 1 *ou* 0 *dependendo de* $u = 0$ *ou* $u \neq 0$.

Consequentemente,

$$\sum_u (F(u) + G(u))\, \phi(2t-u) = \phi(2t),$$

o que completa a prova do teorema 2.4.1.

Apêndice B

Listagem dos Programas

Os programas, rotinas e algoritmos foram desenvolvidos para esta tese e podem ser obtidos gratuitamente de `dovicchi@demac.ufu.br`. As bibliotecas em C dos filtros *Wavelet* foram construídas a partir de coeficiente publicados em diversas fontes [1, 4, 8, 28, 62, 85].

B.1 Programa de Cálculo de Coeficientes *Wavelet*

```
/*
TITULO:         wcoef.c - Calcula os coeficientes wavelet a partir de alfa e
                beta fornecidos.
DATA:           13/09/98
DESCRICAO:      A partir dos valores de alfa e beta conhecidos fornecidos
                o programa calcula os valores de a(-2), a(-1), a(0), a(1),
                a(2) e a(3) que sao os coeficientes wavelet da funcao
                escalonante (passa-baixa) e, a partir dai, calcula o filtro
                wavelet correspondente.
NOME:           wcoef.c
COMPILADOR:     Qualquer compilador C (ANSI C).
ATENCAO:        Desenvolvido em ANSI C, por Joao Dovicchi, Depto. de Musica
                da Universidade Federal de Uberlandia <dovicchi@demac.ufu.br>.
REFERENCIAS:    - I. Daubechies "Ten Lectures on Wavelets",
```

```
                - Kernighan, B. W. "The C Programming Language", New Jersey:
                  Prentice-Hall,Inc., 2nd. Ed, 1988.
                - Plauger, P.J. "The Standard C Library", New Jersey:
                  Prentice-Hall, Inc., 1992.
*/

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <math.h>


FILE *ofp; /* arquivo de saida (output file pointer)*/

int main (int argc, char *argv[]){
        int i;
        double pi;
        double alfa, beta, soma_an, soma_wn, sqr_sum_wn;
        double a[6],w[6];
        char entr[16];
        pi = 4.0*atan(1.0);
        printf("Alfa >= -pi\n");
        printf("Entre o valor de alfa: ");
        scanf("%s", &entr);
        alfa = atof(entr);
        printf("\nBeta < pi");
        printf("\nEntre o valor de beta: ");
        scanf("%s", &entr);
        beta = atof(entr);
        if( alfa < -pi ){
                printf("\n\n *** Valor incorreto de Alfa *** \n");
                exit(1);
        }
        if( beta >= pi ){
                printf("\n\n *** Valor incorreto de Beta *** \n");
                exit(1);
        }

        printf("\nCalculando os valores dos coeficientes para alfa= %2.32f",alfa);
        printf(" e beta= %2.32f\n", beta);
        a[0] = ((1.0 + cos(alfa) + sin(alfa)) * (1.0 - cos(beta) - sin(beta)) +
2.0 * sin(beta) * cos(alfa))/4.0;
        a[1] = ((1.0 - cos(alfa) + sin(alfa)) * (1.0 + cos(beta) - sin(beta)) -
2.0 * sin(beta) * cos(alfa))/4.0;
        a[2] = (1.0 + cos(alfa - beta) + sin(alfa - beta))/2.0;
```

```
        a[3] = (1.0 + cos(alfa - beta) - sin(alfa - beta))/2.0;
        a[4] = 1.0 - a[0] - a[2];
        a[5] = 1.0 - a[1] - a[3];

        printf("\na_(-2) = %2.32f\ta_(-1) = %2.32f\na_(0) = %2.32f\ta_(1) = %2.32f\n", a[0],a[1],a[2],a[3]);
        printf("a_(2) = %2.32f\ta_(3) = %2.32f\n", a[4], a[5]);

        soma_an = a[0]+a[1]+a[2]+a[3]+a[4]+a[5];

        printf("\n\\sum_{a_{-2}}^{a^3} = %2.32f\n", soma_an);

        soma_wn =0.0;
        for( i=0; i<6; i++){
                w[i] = (sqrt(2.0)/2.0) * a[i];
                printf("\nw(%d) = %2.32f", i, w[i]);
                soma_wn = soma_wn + w[i];
        }
        printf("\n\\sum_{n=0}^{n=5} w(n) = %2.32f\n", soma_wn);

        sqr_sum_wn= 0.0;

        for( i=0;i<6;i++){
                sqr_sum_wn = sqr_sum_wn + (w[i] * w[i]);
        }

        printf("\n\\sum_{n=0}^{n=5} |w(n)|^2 = %2.32f\n", sqr_sum_wn);

        return 1;
}
```

B.2 Programa de conversão de arquivos de áudio

```
/*
TITULO:         wd.c - Separa o cabecalho dos dados em um arquivo ".wav".
DATA:           07/08/95
DESCRICAO:      Retira o cabecalho dos dados em um arquivo ".wav", e grava os
                dados transformados em decimais no arquivo plot.dat.
SISTEMA:        Qualquer um.
NOME:           wavdec.c
COMPILADOR:     Qualquer compilador C (ANSI C).
REFERENCIAS:    - Kernighan, B. W. "The C Programming Language", New Jersey:
                  Prentice-Hall,Inc., 2nd. Ed, 1988.
                - Plauger, P.J. "The Standard C Library", New Jersey:
                  Prentice-Hall, Inc., 1992.
```

```
                - Ridge, P. M. et all "Sound Blaster: O Livro Oficial", Sao
                  Paulo, Macron Books, 1994.
*/

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>

FILE *sfp; /* arquivo wave (sample file pointer)*/
FILE *ofp; /* arquivo de saida (output file pointer)*/

int main (int argc, char *argv[])
{

long i;
double ch;

if ( ( sfp = fopen(*++argv,"r") ) != NULL)
        {
                for (i=0;i<44;i++) ch=fgetc(sfp);
                i = 1;
                ch = (double) fgetc(sfp);
                ofp = fopen ("wd.dat", "w");
                do
                        {
                        if(ch < 0) ch+=128.; else ch-=128.;
                        if(ch == -128.) ch = 0.;
                                fprintf(ofp,"%f\n", ch);
                                ch = (double) fgetc(sfp);
                                i++;
                        } while (! feof(sfp));
                        fclose(ofp);
        } else printf(" Erro de leitura do arquivo.");
return 1;
}
```

B.3 Subrotina de convolução de uma transformada *Wavelet*

```
/* wfltr_convolve -- faz uma convolucao de uma transformada wavelet */
static void wfltr_convolve(
```

```
        waveletfilter *wfltr,   /* in: filtro de convolucao */
        bool isFwd,             /* in: TRUE <=> transformada direta */
        TYPE_ARRAY *aIn,        /* in: dado de entrada */
        int incA,               /* in: espacamento de elem. em aIn[] e aXf[] */
        int n,                  /* in: tam. de  aIn e aXf */
        TYPE_ARRAY *aXf)        /* out: dado de saida (OK se == aIn) */
{
        int nDiv2 = n / 2;
        int iA, iHtilde, iGtilde, j, jH, jG, i;
        double sum, aInVal;
        bool flip;

        /*
         *      De acordo com Daubechies:
         *
         *      H eh chamado de "analyzing smoothing filter".
         *      G de "analyzing detail filter".
         *      Htilde de "reconstruction smoothing filter".
         *      Gtilde de "reconstruction detail filter".
         */

        /* O "reconstruction detail filter" eh a imagem do
         * "analysis smoothing filter".
         */
        int nGtilde = wfltr->nH;

        /* O "analysis detail filter" eh a imagem do
         * "reconstruction smoothing filter".
         */
        int nG = wfltr->nHtilde;

        if (isFwd) {
        /*
         *      A analise de um unico passo pode ser descrita como:
         *              aTmp1D[0..n/2-1] = H * a (smooth)
         *              aTmp1D[n/2..n-1] = G * a (detail)
         */
                for (i = 0; i < nDiv2; i++) {
                /*
                 *      aTmp1D[0..nDiv2-1] contem os componentes "smooth".
                 */
                        sum = 0.0;
                        for (jH = 0; jH < wfltr->nH; jH++) {
                /*
                 *      Cada linha de H eh incremento de 2 da anterior.
                 *      Se assumirmos que os dados sao periodicos podemos
```

```
         *      aparar os valores de aIn[] se necessario.
         *      Se tivermos mais amostras do coeficientes H entao
         *      podemos aparar os Hs tambem.
         */
                        iA = MOD(2 * i + jH - wfltr->offH, n);
                        sum += wfltr->cH[jH] * aIn[incA * iA];
                }
                aTmp1D[i] = sum;

        /*
         *      aTmp1D[nDiv2..n-1] contem os componentes de detalhe.
         */
                sum = 0.0;
        /*
         *      imsG <= 0 <=> "flip" inicial
         */
                flip = (wfltr->imsG <= 0);
                for (jG = 0; jG < nG; jG++) {
        /*
         *      Podemos construir os coeficientes G 'in loco' a partir
         *      dos coeficientes H~.
         *
         *      Como H, cada linha de G eh incremento de 2 da anterior
         *      e aqui tambem podemos aparar os coeficientes de G.
         */
                iA = MOD(2 * i + jG - wfltr->offG, n);
                if (flip)
                   sum -= wfltr->cHtilde[nG - 1 - jG] * aIn[incA * iA];
                else
                   sum += wfltr->cHtilde[nG - 1 - jG] * aIn[incA * iA];
                        flip = !flip;
                }
                aTmp1D[nDiv2 + i] = sum;
        }
} else {
        /*
         *      A transformada inversa pode ser feita eficientemente
         *      se a reconstrucao for feita em cada passo:
         *
         *      aTmp1D = (H~)^t * aIn[incA * (0..n/2-1)]
         *              + (G~)^t * aIn[incA * (n/2..n-1)]
         *
         *      onde x^t eh a transposta de x.
         */
        for (i = 0; i < n; i++)
                aTmp1D[i] = 0.0;        /* isso eh necessario */
```

```
            for (j = 0; j < nDiv2; j++) {
                    aInVal = aIn[incA * j];
                    for (iHtilde = 0; iHtilde < wfltr->nHtilde; iHtilde++) {                /*
             *      Cada linha de H~ eh incremento de 2 da anterior.
             */
                    iA = MOD(2 * j + iHtilde - wfltr->offHtilde, n);
                    aTmp1D[iA] += wfltr->cHtilde[iHtilde] * aInVal;
                    }
            /*
             *      imsGtilde <= 0 <=> "flip" inicial.
             */
                    flip = (wfltr->imsGtilde <= 0);
                    aInVal = aIn[incA * (j + nDiv2)];
                    for (iGtilde = 0; iGtilde < nGtilde; iGtilde++) {
            /*
             *      Como com H~, podemos aparar os coeficientes de G~
             *      que sao imagem de H.
             *
             *      Do mesmo modo se os dados sao periodicos podemos aparar
             *      os valores de aIn[], se necessario, e se tivermos
             *      mais amostras que coeficientes G~, podemos apara-los
             *      tambem.
             */
                 iA = MOD(2 * j + iGtilde - wfltr->offGtilde, n);
    if (flip)
                    aTmp1D[iA] -= wfltr->cH[nGtilde - 1 - iGtilde] * aInVal;
                 else
                    aTmp1D[iA] += wfltr->cH[nGtilde - 1 - iGtilde] * aInVal;
                            flip = !flip;
                    }
            }
        }
        for (i = 0; i < n; i++)
                aXf[incA * i] = aTmp1D[i];

        return;
}
```

B.4 Refinamento (*smooth* e *threshold*)

```
/* wrefine_step -- faz uma iteracao de refinamento da wavelet */
static void wrefine_step(
        waveletfilter *wfltr,
```

```
        TYPE_ARRAY *a,
        int n)
{
        int nDiv2 = n / 2;
        int i, iHtilde, j, iA;
        TYPE_ARRAY *aNew;

        (void) MALLOC_LINTOK(aNew, n, TYPE_ARRAY);
        for (i = 0; i < n; i++)
                aNew[i] = 0.0;

/*
 *      Refinamento identico a transformada wavelet inversa (veja wxfrm.c)
 *      com os elementos G~= zero.
 *      Cada passo da reconstrucao pode ser descrito como:
 *
 *              aNew[0..n-1] = Htilde^t * a[0..n/2-1]
 *
 *      onde (H~)^t eh a transposta de H~.
 */
        for (j = 0; j < nDiv2; j++) {
                for (iHtilde = 0; iHtilde < wfltr->nHtilde; iHtilde++) {
                /*
                 *      Cada linha de H~ eh incrementado de 2 em relacao ao
                 *      anterior;
                 */
                        iA = MOD(2 * j + iHtilde - wfltr->offHtilde, n);
                        aNew[iA] += wfltr->cHtilde[iHtilde] * a[j];
                }
        }
        for (i = 0; i < n; i++)
                a[i] = aNew[i];

        FREE_LINTOK(aNew);

        return;
}
```

B.5 Biblioteca de Filtros e Coeficientes

```
/*
 *  Este arquivo contem as definicoes de todos os filtros utilizados em
 *  minha tese de doutorado e que foram utilizados para analise,
```

```
 *  reconstrucao e interpolacao. As fontes sao citadas, sendo a principal
 *  delas o livro "Ten Lectures on Wavelets" de Ingrid Daubechies. Alguns
 *  coeficientes foram corrigidos por Bob Lewis <bobl@cs.ubc.ca> e grande
 *  parte deste codigo foi escrito por ele e modificado por mim.
 *
 *  Alguns filtros tem H = Htilde, conforme o programa original. Outros
 *  foram alterados de acordo com a necessidade e os resultados obtidos
 *  em minha tese.
 */

#include "local.h"

#if !defined(M_SQRT2)
#define M_SQRT2     1.41421356237309504880168872420969808
#endif

/*
 * -------------------------------------------------------------------
 * Adelson-Simoncelli-Hingorani filter
 *
 * Fonte: Adelson, Simoncelli, Hingorani, "Orthogonal Pyramid Transforms
 *   for Image Coding", Visual Communications and Image Processing II,
 *   Proc. SPIE Vol. 845, pp. 50-58.
 */
static double cHAdelsonSimoncelliHingorani[] = {
M_SQRT2 * 0.01995,
M_SQRT2 * -0.04271,
M_SQRT2 * -0.05224,
M_SQRT2 * 0.29271,
M_SQRT2 * 0.56458,
M_SQRT2 * 0.29271,
M_SQRT2 * -0.05224,
M_SQRT2 * -0.04271,
M_SQRT2 * 0.01995
};
waveletfilter wfltrAdelsonSimoncelliHingorani = {
"AdelsonSimoncelliHingorani",
cHAdelsonSimoncelliHingorani,
N_ELEM(cHAdelsonSimoncelliHingorani),
cHAdelsonSimoncelliHingorani,
N_ELEM(cHAdelsonSimoncelliHingorani),
/* offsets of H, G, Htilde, and Gtilde, respectively */
4, 3, 4, 3,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};
```

```
/*
 *  ------------------------------------------------------------
 *  Antonini et al.  filters
 *
 *  Fonte: M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies,
 *   "Image Coding Using Wavelet Transforms", IEEE Trans. on Image
 *   Processing, Vol. 1 No. 2, April 1992
 *
 *    (Table II - Spline Variant with Less Dissimilar Lengths)
 *
 *   Contributed by Brendt Wohlberg (brendt@dip1.ee.uct.ac.za).
 */

static double cHAntoniniBarlaudMatheiuDaubechies_4[] = {
M_SQRT2 *  0.026749,
M_SQRT2 * -0.016864,
M_SQRT2 * -0.078223,
M_SQRT2 *  0.266864,
M_SQRT2 *  0.602949,
M_SQRT2 *  0.266864,
M_SQRT2 * -0.078223,
M_SQRT2 * -0.016864,
M_SQRT2 *  0.026749
};
static double cHtildeAntoniniBarlaudMatheiuDaubechies_4[] = {
M_SQRT2 * -0.045636,
M_SQRT2 * -0.028772,
M_SQRT2 *  0.295636,
M_SQRT2 *  0.557543,
M_SQRT2 *  0.295636,
M_SQRT2 * -0.028772,
M_SQRT2 * -0.045636
};
waveletfilter wfltrAntoniniBarlaudMatheiuDaubechies_4_4 = {
"AntoniniBarlaudMatheiuDaubechies_4_4",
cHAntoniniBarlaudMatheiuDaubechies_4,
N_ELEM(cHAntoniniBarlaudMatheiuDaubechies_4),
cHtildeAntoniniBarlaudMatheiuDaubechies_4,
N_ELEM(cHtildeAntoniniBarlaudMatheiuDaubechies_4),
/* offsets of H, G, Htilde, and Gtilde, respectively */
4, 2, 3, 3,
/* initial mirror signs for G and Gtilde, respectively */
-1, 1
};
```

```
/*
 * ----------------------------------------------------------------
 * Battle-Lemarie filter
 *
 * Fonte: Mallat, "A Theory for Multiresolution Signal Decomposition: The
 *   Wavelet Representation", IEEE PAMI, v. 11, no. 7, 674-693, Table 1
 */

static double cHBattleLemarie[] = {
M_SQRT2 * -0.002,
M_SQRT2 * -0.003,
M_SQRT2 *  0.006,
M_SQRT2 *  0.006,
M_SQRT2 * -0.013,
M_SQRT2 * -0.012,
M_SQRT2 *  0.030,  /*  5 e 6 (sinal alterado por Bob Lewis no */
M_SQRT2 *  0.023,  /*  programa original e diferem do artigo de */
M_SQRT2 * -0.078,  /*  Mallat */
M_SQRT2 * -0.035,
M_SQRT2 *  0.307,
M_SQRT2 *  0.542,
M_SQRT2 *  0.307,
M_SQRT2 * -0.035,
M_SQRT2 * -0.078,
M_SQRT2 *  0.023,
M_SQRT2 *  0.030,
M_SQRT2 * -0.012,
M_SQRT2 * -0.013,
M_SQRT2 *  0.006,
M_SQRT2 *  0.006,
M_SQRT2 * -0.003,
M_SQRT2 * -0.002
};

waveletfilter wfltrBattleLemarie = {
"BattleLemarie",
cHBattleLemarie,
N_ELEM(cHBattleLemarie),
cHBattleLemarie,
N_ELEM(cHBattleLemarie),
/* offsets of H, G, Htilde, and Gtilde, respectively */
11, 10, 11, 10,
/* initial mirror signs for G and Gtilde, respectively */
-1, -1
};
```

```
/*
 * ----------------------------------------------------------------
 * Burt-Adelson filter
 *
 * Fonte: Daubechies, I. "Ten Lectures on Wavelets", Table 8.4
 */
static double cHBurtAdelson[] = {
M_SQRT2 * -1.0 / 20.0,
M_SQRT2 *  5.0 / 20.0,
M_SQRT2 * 12.0 / 20.0,
M_SQRT2 *  5.0 / 20.0,
M_SQRT2 * -1.0 / 20.0
};
static double cHtildeBurtAdelson[] = {
M_SQRT2 *  -3.0 / 280.0,
M_SQRT2 * -15.0 / 280.0,
M_SQRT2 *  73.0 / 280.0,
M_SQRT2 * 170.0 / 280.0,
M_SQRT2 *  73.0 / 280.0,
M_SQRT2 * -15.0 / 280.0,
M_SQRT2 *  -3.0 / 280.0
};
waveletfilter wfltrBurtAdelson = {
"BurtAdelson",
cHBurtAdelson,
N_ELEM(cHBurtAdelson),
cHtildeBurtAdelson,
N_ELEM(cHtildeBurtAdelson),
/* offsets of H, G, Htilde, and Gtilde, respectively */
2, 2, 3, 1,
/* initial mirror signs for G and Gtilde, respectively */
-1, 1
};

/*
 * ----------------------------------------------------------------
 * Coiflet filters
 *
 * Fonte: Beylkin, Coifman, and Rokhlin "Fast Wavelet Transforms and
 *   Numerical Algorithms I", Comm. Pure Appl. Math, v. 44, Appendix A
 */

#ifndef SQRT15
#define SQRT15 3.87298334620741688517927
#endif
```

```
static double cHCoiflet_2[] = {
M_SQRT2 * (SQRT15 - 3) / 32.0,
M_SQRT2 * (1 - SQRT15) / 32.0,
M_SQRT2 * (6 - 2 * SQRT15) / 32.0,
M_SQRT2 * (2 * SQRT15 + 6) / 32.0,
M_SQRT2 * (SQRT15 + 13) / 32.0,
M_SQRT2 * (9 - SQRT15) / 32.0
};
waveletfilter wfltrCoiflet_2 = {
"Coiflet_2",
cHCoiflet_2,
N_ELEM(cHCoiflet_2),
cHCoiflet_2,
N_ELEM(cHCoiflet_2),
/* offsets of H, G, Htilde, and Gtilde, respectively */
4, 0, 4, 0,
/* initial mirror signs for G and Gtilde, respectively */
-1, -1
};

static double cHCoiflet_4[] = {
 0.0011945726958388,
-0.01284557955324,
 0.024804330519353,
 0.050023519962135,
-0.15535722285996,
-0.071638282295294,
 0.57046500145033,
 0.75033630585287,
 0.28061165190244,
-0.0074103835186718,
-0.014611552521451,
-0.0013587990591632
};
waveletfilter wfltrCoiflet_4 = {
"Coiflet_4",
cHCoiflet_4,
N_ELEM(cHCoiflet_4),
cHCoiflet_4,
N_ELEM(cHCoiflet_4),
/* offsets of H, G, Htilde, and Gtilde, respectively */
7, 3, 7, 3,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};
```

```
static double cHCoiflet_6[] = {
-0.0016918510194918,
-0.00348787621998426,
 0.019191160680044,
 0.021671094636352,
-0.098507213321468,
-0.056997424478478,
 0.45678712217269,
 0.78931940900416,
 0.38055713085151,
-0.070438748794943,
-0.056514193868065,
 0.036409962612716,
 0.0087601307091635,
-0.011194759273835,
-0.0019213354141368,
 0.0020413809772660,
 0.00044583039753204,
-0.00021625727664696
};
waveletfilter wfltrCoiflet_6 = {
"Coiflet_6",
cHCoiflet_6,
N_ELEM(cHCoiflet_6),
cHCoiflet_6,
N_ELEM(cHCoiflet_6),
/* offsets of H, G, Htilde, and Gtilde, respectively */
7, 9, 7, 9,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};

/*
 * ---------------------------------------------------------------
 * Daubechies filters
 *
 * Fonte: Daubechies, I. "Ten Lectures on Wavelets", Table 6.1
 */

#ifndef SQRT3
#define SQRT3 1.7320508075688772935274 5
#endif

static double cHDaubechies_4[] = {
M_SQRT2 * (1 + SQRT3) / 8.0,
M_SQRT2 * (3 + SQRT3) / 8.0,
```

```
M_SQRT2 * (3 - SQRT3) / 8.0,
M_SQRT2 * (1 - SQRT3) / 8.0
};
waveletfilter wfltrDaubechies_4 = {
"Daubechies_4",
cHDaubechies_4,
N_ELEM(cHDaubechies_4),
cHDaubechies_4,
N_ELEM(cHDaubechies_4),
/* offsets of H, G, Htilde, and Gtilde, respectively */
1, 1, 1, 1,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};

static double cHDaubechies_6[] = {
 0.332670552950,
 0.806891509311,
 0.459877502118,
-0.135011020010,
-0.085441273882,
 0.035226291882
};
waveletfilter wfltrDaubechies_6 = {
"Daubechies_6",
cHDaubechies_6,
N_ELEM(cHDaubechies_6),
cHDaubechies_6,
N_ELEM(cHDaubechies_6),
/* offsets of H, G, Htilde, and Gtilde, respectively */
1, 3, 1, 3,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};

static double cHDaubechies_8[] = {
 0.230377813309,
     0.714846570553,
 0.6308807667930,
-0.027983769417,
-0.187034811719,
 0.030841381836,
 0.032883011667,
-0.010597401785
};
waveletfilter wfltrDaubechies_8 = {
```

```
"Daubechies_8",
cHDaubechies_8,
N_ELEM(cHDaubechies_8),
cHDaubechies_8,
N_ELEM(cHDaubechies_8),
/* offsets of H, G, Htilde, and Gtilde, respectively */
1, 5, 1, 5,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};

static double cHDaubechies_10[] = {
 0.1601023979741929,
 0.6038292697971895,
 0.7243085284377726,
 0.1384281459013203,
-0.2422948870663823,
-0.0322448695846381,
 0.0775714938400459,
-0.0062414902127983,
-0.0125807519990820,
 0.0033357252854738
};
waveletfilter wfltrDaubechies_10 = {
"Daubechies_10",
cHDaubechies_10,
N_ELEM(cHDaubechies_10),
cHDaubechies_10,
N_ELEM(cHDaubechies_10),
/* offsets of H, G, Htilde, and Gtilde, respectively */
1, 7, 1, 7,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};

static double cHDaubechies_12[] = {
 0.1115407433501095,
 0.4946238903984533,
 0.7511339080210959,
 0.3152503517091982,
-0.2262646939654400,
-0.1297668675672625,
 0.0975016055873225,
 0.0275228655303053,
-0.0315820393184862,
 0.0005538422011614,
```

```
 0.0047772575119455,
-0.0010773010853085
};
waveletfilter wfltrDaubechies_12 = {
"Daubechies_12",
cHDaubechies_12,
N_ELEM(cHDaubechies_12),
cHDaubechies_12,
N_ELEM(cHDaubechies_12),
/* offsets of H, G, Htilde, and Gtilde, respectively */
1, 9, 1, 9,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};

static double cHDaubechies_20[] = {
 0.026670057901,
 0.188176800078,
 0.527201188932,
 0.688459039454,
 0.281172343661,
-0.249846424327,
-0.195946274377,
 0.127369340336,
 0.093057364604,
-0.071394147166,
-0.029457536822,
 0.033212674059,
 0.003606553567,
-0.010733175483,
 0.001395351747,
 0.001992405295,
-0.000685856695,
-0.000116466855,
 0.000093588670,
-0.000013264203
};
waveletfilter wfltrDaubechies_20 = {
"Daubechies_20",
cHDaubechies_20,
N_ELEM(cHDaubechies_20),
cHDaubechies_20,
N_ELEM(cHDaubechies_20),
/* offsets of H, G, Htilde, and Gtilde, respectively */
2, 16, 2, 16,
/* initial mirror signs for G and Gtilde, respectively */
```

```
1, 1
};

/*
 * ---------------------------------------------------------------
 * Haar filter
 *
 * Fonte: Daubechies, I "Ten Lectures on Wavelets", p. 10.
 */

static double cHHaar[] = {
M_SQRT2 * 1.0 / 2.0,
M_SQRT2 * 1.0 / 2.0
};

waveletfilter wfltrHaar = {
"Haar",
cHHaar,
N_ELEM(cHHaar),
cHHaar,
N_ELEM(cHHaar),
/* offsets of H, G, Htilde, and Gtilde, respectively */
0, 0, 0, 0,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};

/*
 * ---------------------------------------------------------------
 * Chroma filters.
 * Coeficientes originais a partir de fatores do intervalo cromatico.
 */

/* alfa=1.059463 (2^1/12), beta = 0 */
static double cHChroma_4[] = {
0.488877,
0.834893,
0.218230,
-0.127786
};
waveletfilter wfltrChroma_4 = {
"Chroma_4",
cHChroma_4,
N_ELEM(cHChroma_4),
cHChroma_4,
N_ELEM(cHChroma_4),
```

```
/* offsets of H, G, Htilde, and Gtilde, respectively */
0, 0, 0, 0,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};

static double cHChroma_6[] = {/* actually Chroma_6 */
/* alfa= raiz 12 de 2 e beta= raiz 12 de 2^6:
alfa= 1.059463 beta = 0.062999 (dif. entre dois int. cromaticos) */
-0.014561067937305388272917205938488,
0.46211056360837471368441242702829,
0.84245747811804405635882631031564,
0.24880228939824242107015095371025,
-0.12078962899419119769728325763936,
-0.0038060718200696457176601672500 7

};


waveletfilter wfltrChroma_6 = {
"Chroma_6",
cHChroma_6,
N_ELEM(cHChroma_6),
cHChroma_6,
N_ELEM(cHChroma_6),
/* offsets of H, G, Htilde, and Gtilde, respectively */
0, 0, 0, 0,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};

/*
 * -----------------------------------------------------------------
 * Pseudocoiflet filter
 * Fonte: Reissell, "Multiresolution Geometric Algorithms Using Wavelets I:
 *   Representation for Parametric Curves and Surfaces", UBC TR 93-17,
 *   p. 33
 */

static double cHPseudocoiflet_4[] = {
M_SQRT2 *  -1.0 / 512.0,
  0.0,
M_SQRT2 *  18.0 / 512.0,
M_SQRT2 * -16.0 / 512.0,
M_SQRT2 * -63.0 / 512.0,
M_SQRT2 * 144.0 / 512.0,
```

```
M_SQRT2 * 348.0 / 512.0,
M_SQRT2 * 144.0 / 512.0,
M_SQRT2 * -63.0 / 512.0,
M_SQRT2 * -16.0 / 512.0,
M_SQRT2 *  18.0 / 512.0,
  0.0,
M_SQRT2 *  -1.0 / 512.0
};
static double cHtildePseudocoiflet_4[] = {
M_SQRT2 *  -1.0 / 32.0,
  0.0,
M_SQRT2 *   9.0 / 32.0,
M_SQRT2 *  16.0 / 32.0,
M_SQRT2 *   9.0 / 32.0,
  0.0,
M_SQRT2 *  -1.0 / 32.0
};
waveletfilter wfltrPseudocoiflet_4_4 = {
"Pseudocoiflet_4_4",
cHPseudocoiflet_4,
N_ELEM(cHPseudocoiflet_4),
cHtildePseudocoiflet_4,
N_ELEM(cHtildePseudocoiflet_4),
/* offsets of H, G, Htilde, and Gtilde, respectively */
6, 2, 3, 5,
/* initial mirror signs for G and Gtilde, respectively */
-1, 1
};

/*
 * -----------------------------------------------------------------
 * Spline filters
 *
 * Fonte: Daubechies, I. "Ten Lectures on Wavelets", Table 8.2
 * (corrected by Bob Lewis)
 */
static double cHSpline_2[] = {
M_SQRT2 * -0.125,
M_SQRT2 *  0.25,
M_SQRT2 *  0.75,
M_SQRT2 *  0.25,
M_SQRT2 * -0.125
};
static double cHSpline_3[] = {
M_SQRT2 *  3.0 / 64.0,
M_SQRT2 * -9.0 / 64.0,
```

```
M_SQRT2 * -7.0 / 64.0,
M_SQRT2 * 45.0 / 64.0,
M_SQRT2 * 45.0 / 64.0,
M_SQRT2 * -7.0 / 64.0,
M_SQRT2 * -9.0 / 64.0,
M_SQRT2 *  3.0 / 64.0
};
static double cHSpline_4[] = {
M_SQRT2 *   3.0 / 128.0,
M_SQRT2 *  -6.0 / 128.0,
M_SQRT2 * -16.0 / 128.0,
M_SQRT2 *  38.0 / 128.0,
M_SQRT2 *  90.0 / 128.0,
M_SQRT2 *  38.0 / 128.0,
M_SQRT2 * -16.0 / 128.0,
M_SQRT2 *  -6.0 / 128.0,
M_SQRT2 *   3.0 / 128.0
};
static double cHSpline_5[] = {
M_SQRT2 *  -5.0 / 512.0,
M_SQRT2 *  15.0 / 512.0,
M_SQRT2 *  19.0 / 512.0,
M_SQRT2 * -97.0 / 512.0,
M_SQRT2 * -26.0 / 512.0,
M_SQRT2 * 350.0 / 512.0,
M_SQRT2 * 350.0 / 512.0,
M_SQRT2 * -26.0 / 512.0,
M_SQRT2 * -97.0 / 512.0,
M_SQRT2 *  19.0 / 512.0,
M_SQRT2 *  15.0 / 512.0,
M_SQRT2 *  -5.0 / 512.0
};
static double cHSpline_7[] = {
M_SQRT2 *   -35.0 / 16384.0,
M_SQRT2 *  -105.0 / 16384.0,
M_SQRT2 *  -195.0 / 16384.0,
M_SQRT2 *   865.0 / 16384.0,
M_SQRT2 *   363.0 / 16384.0, /* correcao Bob Lewis (336 no texto) */
M_SQRT2 * -3489.0 / 16384.0,
M_SQRT2 *  -307.0 / 16384.0,
M_SQRT2 * 11025.0 / 16384.0,
M_SQRT2 * 11025.0 / 16384.0,
M_SQRT2 *  -307.0 / 16384.0,
M_SQRT2 * -3489.0 / 16384.0,
M_SQRT2 *   363.0 / 16384.0, /* idem */
M_SQRT2 *   865.0 / 16384.0,
```

```
M_SQRT2 *  -195.0 / 16384.0,
M_SQRT2 *  -105.0 / 16384.0,
M_SQRT2 *   -35.0 / 16384.0
};

static double cHtildeSpline_2[] = {
M_SQRT2 * 1.0 / 4.0,
M_SQRT2 * 2.0 / 4.0,
M_SQRT2 * 1.0 / 4.0
};
static double cHtildeSpline_3[] = {
M_SQRT2 * 1.0 / 8.0,
M_SQRT2 * 3.0 / 8.0,
M_SQRT2 * 3.0 / 8.0,
M_SQRT2 * 1.0 / 8.0
};

waveletfilter wfltrSpline_2_2 = {
"Spline_2_2",
cHSpline_2,
N_ELEM(cHSpline_2),
cHtildeSpline_2,
N_ELEM(cHtildeSpline_2),
/* offsets of H, G, Htilde, and Gtilde, respectively */
2, 0, 1, 1,
/* initial mirror signs for G and Gtilde, respectively */
-1, 1
};
waveletfilter wfltrSpline_2_4 = {
"Spline_2_4",
cHSpline_4,
N_ELEM(cHSpline_4),
cHtildeSpline_2,
N_ELEM(cHtildeSpline_2),
/* offsets of H, G, Htilde, and Gtilde, respectively */
4, 0, 1, 3,
/* initial mirror signs for G and Gtilde, respectively */
-1, 1
};
waveletfilter wfltrSpline_3_3 = {
"Spline_3_3",
cHSpline_3,
N_ELEM(cHSpline_3),
cHtildeSpline_3,
N_ELEM(cHtildeSpline_3),
/* offsets of H, G, Htilde, and Gtilde, respectively */
```

```
3, 1, 1, 3,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};
waveletfilter wfltrSpline_3_5 = {
"Spline_3_5",
cHSpline_5,
N_ELEM(cHSpline_5),
cHtildeSpline_3,
N_ELEM(cHtildeSpline_3),
/* offsets of H, G, Htilde, and Gtilde, respectively */
5, 1, 1, 5,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};
waveletfilter wfltrSpline_3_7 = {
"Spline_3_7",
cHSpline_7,
N_ELEM(cHSpline_7),
cHtildeSpline_3,
N_ELEM(cHtildeSpline_3),
/* offsets of H, G, Htilde, and Gtilde, respectively */
7, 1, 1, 7,
/* initial mirror signs for G and Gtilde, respectively */
1, 1
};

/*
 * Array de todos os filtros.
 */
waveletfilter *wfltrList[] = {
&wfltrAdelsonSimoncelliHingorani,
&wfltrAntoniniBarlaudMatheiuDaubechies_4_4,
&wfltrBattleLemarie,
&wfltrBurtAdelson,
&wfltrCoiflet_2,
&wfltrCoiflet_4,
&wfltrCoiflet_6,
&wfltrDaubechies_4,
&wfltrDaubechies_6,
&wfltrDaubechies_8,
&wfltrDaubechies_10,
&wfltrDaubechies_12,
&wfltrDaubechies_20,
&wfltrHaar,
&wfltrChroma_4,
```

```
&wfltrChroma_6,
&wfltrPseudocoiflet_4_4,
&wfltrSpline_2_2,
&wfltrSpline_2_4,
&wfltrSpline_3_3,
&wfltrSpline_3_5,
&wfltrSpline_3_7,
NULL
};

/*
 * -----------------------------------------------------------------
 * wfltr_exchange -- troca os componentes H/H~ G/G~ de um filtro
 * -----------------------------------------------------------------
 */
void wfltr_exchange(
waveletfilter *wfltrNorm, /* entra: filtro original */
waveletfilter *wfltrRev) /* sai: filtro reverso */
                                       /*(ok se == wfltrNorm) */
{
int iSwap;
double *pDSwap;

/* Esta funcao so funciona para filtros biortogonais. Filtros ortogonais
 * permanecem os mesmos.
 */

iSwap = wfltrNorm->offH;
wfltrRev->offH = wfltrNorm->offHtilde;
wfltrRev->offHtilde = iSwap;

iSwap = wfltrNorm->nH;
wfltrRev->nH = wfltrNorm->nHtilde;
wfltrRev->nHtilde = iSwap;

iSwap = wfltrNorm->offG;
wfltrRev->offG = wfltrNorm->offGtilde;
wfltrRev->offGtilde = iSwap;

iSwap = wfltrNorm->imsG;
wfltrRev->imsG = wfltrNorm->imsGtilde;
wfltrRev->imsGtilde = iSwap;

pDSwap = wfltrNorm->cH;
wfltrRev->cH = wfltrNorm->cHtilde;
wfltrRev->cHtilde = pDSwap;
```

```
return;
}

/*
 * wfltr_select -- Selecao do filtro pelo nome com "*" (praticidade).
 */
waveletfilter *wfltr_select(
char *wcpat,
int *nMatch)
{
int i;
waveletfilter *wfltr = NULL;

(*nMatch) = 0;
for (i = 0; wfltrList[i] != NULL; i++) {
if (wildmat(wfltrList[i]->name, wcpat)) {
wfltr = wfltrList[i];
(*nMatch)++;
}
}
return ( (*nMatch) == 1 ? wfltr : NULL );
}
```

B.6 Algoritmo principal de transformação

```
/*
 *  Este arquivo descreve prototypes. (cf. wrefine_t.c)
 */

/* Minima ordem permitida para o filtro wavelet */
#define MIN_ORDER 2

/* Transformadas multidimensionais suportam um maximo dessas dimensoes */
#define MXN_D 32

static void wfltr_convolve _PROTO((waveletfilter *wfltr, bool isFwd,
TYPE_ARRAY *aIn, int incA, int n, TYPE_ARRAY *aXf));
static void wxfrm_1d_varstep _PROTO((TYPE_ARRAY *a, int incA, int nA, bool isFwd,
waveletfilter *wfltr, TYPE_ARRAY *aXf));
static void wxfrm_nd_nonstd _PROTO((TYPE_ARRAY *a, int nA[], int nD,
bool isFwd, waveletfilter *wfltr));
static void wxfrm_nd_std _PROTO((TYPE_ARRAY *a, int nA[], int nD,
bool isFwd, waveletfilter *wfltr));
```

```
static TYPE_ARRAY *aTmp1D = NULL;

/* wfltr_convolve -- faz uma convolucao de uma transformada wavelet */
static void wfltr_convolve(
waveletfilter *wfltr, /* in: filtro de convolucao */
bool isFwd, /* in: TRUE <=> transformada direta */
TYPE_ARRAY *aIn, /* in: dado de entrada */
int incA, /* in: espacamento de elem. em aIn[] e aXf[] */
int n, /* in: tam. de  aIn e aXf */
TYPE_ARRAY *aXf) /* out: dado de saida (OK se == aIn) */
{
int nDiv2 = n / 2;
int iA, iHtilde, iGtilde, j, jH, jG, i;
double sum, aInVal;
bool flip;

/*
 * De acordo com Daubechies:
 *
 * H eh chamado de "analyzing smoothing filter".
 * G de "analyzing detail filter".
 * Htilde de "reconstruction smoothing filter".
 * Gtilde de "reconstruction detail filter".
 */

/* O "reconstruction detail filter" eh a imagem do
 * "analysis smoothing filter".
 */
int nGtilde = wfltr->nH;

/* O "analysis detail filter" eh a imagem do
 * "reconstruction smoothing filter".
 */
int nG = wfltr->nHtilde;

if (isFwd) {
/*
 *  A analise de um unico passo pode ser descrita como:
 * aTmp1D[0..n/2-1] = H * a (smooth)
 * aTmp1D[n/2..n-1] = G * a (detail)
 */
for (i = 0; i < nDiv2; i++) {
/*
 * aTmp1D[0..nDiv2-1] contem os componentes "smooth".
 */
```

```
sum = 0.0;
for (jH = 0; jH < wfltr->nH; jH++) {
/*
 *  Cada linha de H eh incremento de 2 da anterior.
 *  Se assumirmos que os dados sao periodicos podemos
 *  aparar os valores de aIn[] se necessario.
 *  Se tivermos mais amostras do coeficientes H entao
 *  podemos aparar os Hs tambem.
 */
iA = MOD(2 * i + jH - wfltr->offH, n);
sum += wfltr->cH[jH] * aIn[incA * iA];
}
aTmp1D[i] = sum;

/*
 * aTmp1D[nDiv2..n-1] contem os componentes de detalhe.
 */
sum = 0.0;
/*
 * imsG <= 0 <=> "flip" inicial
 */
flip = (wfltr->imsG <= 0);
for (jG = 0; jG < nG; jG++) {
/*
 *  Podemos construir os coeficientes G 'in loco' a partir
 *  dos coeficientes H~.
 *
 * Como H, cada linha de G eh incremento de 2 da anterior
 * e aqui tambem podemos aparar os coeficientes de G.
 */
iA = MOD(2 * i + jG - wfltr->offG, n);
if (flip)
sum -= wfltr->cHtilde[nG - 1 - jG] * aIn[incA * iA];
else
sum += wfltr->cHtilde[nG - 1 - jG] * aIn[incA * iA];
flip = !flip;
}
aTmp1D[nDiv2 + i] = sum;
}
} else {
/*
 *  A transformada inversa pode ser feita eficientemente
 *  se a reconstrucao for feita em cada passo:
 *
 * aTmp1D = (H~)^t * aIn[incA * (0..n/2-1)]
 * + (G~)^t * aIn[incA * (n/2..n-1)]
```

```
 *
 * onde x^t eh a transposta de x.
 */
for (i = 0; i < n; i++)
aTmp1D[i] = 0.0; /* isso eh necessario */
for (j = 0; j < nDiv2; j++) {
aInVal = aIn[incA * j];
for (iHtilde = 0; iHtilde < wfltr->nHtilde; iHtilde++) {
/*
 * Cada linha de H~ eh incremento de 2 da anterior.
 */
iA = MOD(2 * j + iHtilde - wfltr->offHtilde, n);
aTmp1D[iA] += wfltr->cHtilde[iHtilde] * aInVal;
}
/*
 * imsGtilde <= 0 <=> "flip" inicial.
 */
flip = (wfltr->imsGtilde <= 0);
aInVal = aIn[incA * (j + nDiv2)];
for (iGtilde = 0; iGtilde < nGtilde; iGtilde++) {
/*
 *  Como com H~, podemos aparar os coeficientes de G~
 *  que sao imagem de H.
 *
 * Do mesmo modo se os dados sao periodicos podemos aparar
 * os valores de aIn[], se necessario, e se tivermos
 * mais amostras que coeficientes G~, podemos apara-los
 * tambem.
 */
iA = MOD(2 * j + iGtilde - wfltr->offGtilde, n);
if (flip)
aTmp1D[iA] -= wfltr->cH[nGtilde - 1 - iGtilde] * aInVal;
else
aTmp1D[iA] += wfltr->cH[nGtilde - 1 - iGtilde] * aInVal;
flip = !flip;
}
}
}
for (i = 0; i < n; i++)
aXf[incA * i] = aTmp1D[i];

return;
}

/* wxfrm_1d_varstep -- Transf. Wavelet discreta 1-dimensional de varios passos*/
static void wxfrm_1d_varstep(
```

```
TYPE_ARRAY *aIn, /* in: array original */
int incA, /* in: espacamento dos elem. em aIn[] e aXf[] */
int nA, /* in: tam. de a (tem que ser potencia de 2) */
bool isFwd, /* in: TRUE <=> transformada direta*/
waveletfilter *wfltr, /* in: filtro wavelet usado*/
TYPE_ARRAY *aXf) /* out: array tramsformado */
{
int iA;

if (nA < MIN_ORDER)
return;

if (isFwd) {
/*
 *  Comecando da maior para menor ordem.
 */
wfltr_convolve(wfltr, isFwd, aIn, incA, nA, aXf);
for (iA = nA / 2; iA >= MIN_ORDER; iA /= 2)
wfltr_convolve(wfltr, isFwd, aXf, incA, iA, aXf);
} else {
/*
 * Da menor para a maior.
 */
if (aXf != aIn) {
for (iA = 0; iA < nA; iA++)
aXf[incA * iA] = aIn[incA * iA];
}
for (iA = MIN_ORDER; iA <= nA; iA *= 2)
wfltr_convolve(wfltr, isFwd, aXf, incA, iA, aXf);
}

return;
}

/* wxfrm_[fd]a1d -- DWT 1-dimensional */
void FUNC_1D(
TYPE_ARRAY *a, /* in: array original*/
int nA, /* in: tam. de  a (potencia de 2) */
bool isFwd, /* in: TRUE <=> transformada direta */
waveletfilter *wfltr, /* in: filtro wavelet usado */
TYPE_ARRAY *aXf) /* out: array transformado */
{
assert(aTmp1D == NULL);

(void) MALLOC_LINTOK(aTmp1D, nA, TYPE_ARRAY);
wxfrm_1d_varstep(a, 1, nA, isFwd, wfltr, aXf);
```

```
FREE_LINTOK(aTmp1D);
aTmp1D = NULL;

return;
}

/* wxfrm_[fd]and -- DWT n-dimensional */
void FUNC_ND(
TYPE_ARRAY *aIn, /* in: dados originais */
int nA[], /* in: tam. de cada dim. de a (potencia de 2) */
int nD, /* in: num. de dimensoes (tam. de nA[]) */
bool isFwd, /* in: TRUE <=> transformada direta */
bool isStd, /* in: TRUE <=> base padrao */
waveletfilter *wfltr, /* in: filtro wavelet usado */
TYPE_ARRAY *aXf) /* out: dados transformados (ok se == a) */
{
int k, d, nDMax, nATot;
int lg2nA;

if (nD < 1 || MXN_D < nD) {
(void) fprintf(stderr,
"number of dimensions must be between 1 and %d -- exiting\n",
MXN_D);
exit(1);
}

for (d = 0; d < nD; d++) {
for (lg2nA = 0; (1 << lg2nA) < nA[d]; lg2nA++)
continue;
if ((1 << lg2nA) != nA[d]) {
(void) fprintf(stderr,
"size of dimension #%d (= %d) is not a power of 2 -- exiting\n", d, nA[d]);
exit(1);
}
}

nDMax = 1;
nATot = 1;
for (d = 0; d < nD; d++) {
if (nA[d] > nDMax)
nDMax = nA[d];
nATot *= nA[d];
}

assert(aTmp1D == NULL);
(void) MALLOC_LINTOK(aTmp1D, nDMax, TYPE_ARRAY);
```

```
/*
 *  aXf[] inicialmente igual a aIn[] eh modificada 'in loco'
 */
if (aIn != aXf) {
for (k = 0; k < nATot; k++)
aXf[k] = aIn[k];
}

if (isStd)
wxfrm_nd_std(aXf, nA, nD, isFwd, wfltr);
else
wxfrm_nd_nonstd(aXf, nA, nD, isFwd, wfltr);

FREE_LINTOK(aTmp1D);
aTmp1D = NULL;

return;
}

/* wxfrm_nd_nonstd -- transformada wavelet n-dimensional nao padrao*/
static void wxfrm_nd_nonstd(
TYPE_ARRAY *a, /* in: dado original/final (modif. 'in loco') */
int nA[], /* in: tam. de cada dim. de a (potencia de 2) */
int nD, /* in: num. de dimensoes (tam. de nA[]) */
bool isFwd, /* in: TRUE <=> transformada direta */
waveletfilter *wfltr) /* in: filtro wavelet usado*/
{
int d, d0, dMax, nConv, iConv, iConvDecoded;
int iA, nATot, nBTot;
int nARev[MXN_D], nB[MXN_D], incA[MXN_D];
bool stretchOk[MXN_D];

/*
 *  A matriz a declarada:
 * TYPE_ARRAY a[n][m];
 * em nA declarada:
 * int nA[] = { n, m };
 * Entretanto, por conveniencia, refere-se aa forma nARev[] (que
 * eh a reversa de nA[]) porque nARev[0] eh o indice de variacao
 * mais rapida no mapeamento de a[]. Isto poderia ser melhor
 * implementado em linguagem funcional.
 */
for (d = 0; d < nD; d++)
nARev[d] = nA[nD - 1 - d];
```

```
/*
 * A[] eh a matriz original.  B[] eh uma submatriz de A[] cuja
 * transformada estah sendo computada num dado tempo.
 */
nATot = 1;
for (d = 0; d < nD; d++) {
incA[d] = nATot;
nATot *= nARev[d];
}

nBTot = nATot;
if (isFwd) {
/* inicialmente, B eh equivalente a A */
for (d = 0; d < nD; d++)
nB[d] = nARev[d];
/*
 * loop principal (escala descentente)
 */
while (nBTot > 1) {
for (d0 = 0; d0 < nD; d0++) {
if (nB[d0] <= 1)
continue;
nConv = nBTot / nB[d0]; /* # of convolucoes */
for (iConv = 0; iConv < nConv; iConv++) {
iA = 0;
iConvDecoded = iConv;
for (d = 0; d < nD; d++) {
if (d != d0) {
iA += incA[d] * (iConvDecoded % nB[d]);
iConvDecoded /= nB[d];
}
}
wfltr_convolve(wfltr, TRUE, &a[iA], incA[d0], nB[d0], &a[iA]);
}
}
nBTot = 1;
for (d = 0; d < nD; d++) {
if (nB[d] > 1) {
nB[d] /= 2;
nBTot *= nB[d];
}
}
}
} else {
/* inicialmente, B eh uma submatriz 1x1x..x1 */
for (d = 0; d < nD; d++)
```

```
nB[d] = 1;
nBTot = 1;
/*
 * loop principal (escala ascendente)
 */
while (nBTot < nATot) {
dMax = 0;
for (d = 1; d < nD; d++) {
if (nARev[d] * nB[dMax] >= nARev[dMax] * nB[d])
dMax = d;
}
for (d = 0; d < nD; d++)
stretchOk[d] = (nB[d] > 1 || (nARev[d] * nB[dMax] >= nARev[dMax] * nB[d]));

for (d0 = nD - 1; d0 >= 0; d0--) {
if (!stretchOk[d0])
continue;
nB[d0] *= 2;
nBTot *= 2;
}

for (d0 = nD - 1; d0 >= 0; d0--) {
if (!stretchOk[d0])
continue;
nConv = nBTot / nB[d0];
for (iConv = 0; iConv < nConv; iConv++) {
iA = 0;
iConvDecoded = iConv;
for (d = 0; d < nD; d++) {
if (d != d0 && stretchOk[d]) {
iA += incA[d] * (iConvDecoded % nB[d]);
iConvDecoded /= nB[d];
}
}
wfltr_convolve(wfltr, FALSE, &a[iA], incA[d0], nB[d0], &a[iA]);
}
}
}
}
return;
}

/* wxfrm_nd_std -- transformada wavelet n-dimensional padrao */
static void wxfrm_nd_std(
TYPE_ARRAY *a, /* in: dado original/final (modif. 'in loco') */
int nA[], /* in: tam de cada dim de a (Potencia de 2) */
```

```
int nD, /* in: num. de dimensoes (tam. de nA[]) */
bool isFwd, /* in: TRUE <=> transformada direta */
waveletfilter *wfltr) /* in: filtro wavelet usado */
{
int incA = 1;
int incNext, d, i1, i2, nATot;
int nARev[MXN_D];

/*
 *  Veja comentario acima sobre a inversao da matriz 'a'.
 */
for (d = 0; d < nD; d++)
nARev[d] = nA[nD - 1 - d];

nATot = 1;
for (d = 0; d < nD; d++)
nATot *= nARev[d];

/*
 * loop principal sobre as dimensoes
 */
for (d = 0; d < nD; d++) {
incNext = nARev[d] * incA;
if (nARev[d] >= MIN_ORDER) {
for (i2 = 0; i2 < nATot; i2 += incNext) {
for (i1 = 0; i1 < incA; i1++) {
wxfrm_1d_varstep(&a[i1 + i2], incA, nARev[d], isFwd, wfltr,
&a[i1 + i2]);
}
}
}
incA = incNext;
}
return;
}
```

Apêndice C

Tabelas de Instrumentos e Créditos dos Sons

C.1 Fichas Técnicas

As tabelas abaixo apresentam 13 harmônicos das freqüências Dó ($\sim$ 260 Hz) e Lá ($\sim$ 440 Hz) dos instrumentos de sopros estudados. No caso do Fagote, foram utilizados as freqüências Dó ($\sim$ 65Hz) e Lá ($\sim$ 110 Hz). Os tempos t_n, nas tabelas, se referem às magnitudes absolutas do espectro obtido numa janela de 100 amostras em torno de $n \times 11.4625$ ms para $n = \{1 \ldots 5\}$.

As gravações foram feitas em microfones Neumann KM 84 acoplados a um *mixer* Mackie 1402-VLZ e um gravador DAT Panasonic SV-3800 DAT Recorder, transferidas para o Sound Designer II via uma placa Automedia III em um Power Macintosh 8500. As notas foram executadas cromaticamente, com duração de aproximadamente 2 segundos, seguidas do som ambiente.

Flauta Transversal sem Vibrato

Detalhes Técnicos:

Performer: Sonja Feig

Instrumento:

Data da Gravação: 20 de dezembro de 1997

Local da Gravação: Wendell Johnson Speech and Hearing Center, University of Iowa.

Altura do Microfone:

Altura do Instrumento:

Distância do Microfone:

Sons Utilizados

Dó

1. Taxa de Amostragem: 44.1 kHz
2. Duração: 2.689637 seg
3. Nr. Amostras: 118613
4. Formato: 16 bits linear RIFF/WAV
5. Mx. Ampl.: 0.852

Lá

freqüência	t_1	t_2	t_3	t_4	t_5
258.40	0.0016	0.0235	0.0845	0.1587	0.1863
516.80	0.0014	0.0163	0.0555	0.1103	0.1379
775.20	0.0013	0.0140	0.0284	0.0207	0.0134
1033.59	0.0011	0.0124	0.0346	0.0503	0.0543
1291.99	0.0009	0.0036	0.0036	0.0090	0.0137
1550.39	0.0008	0.0052	0.0056	0.0085	0.0249
1808.79	0.0007	0.0018	0.0155	0.0397	0.0545
2067.19	0.0007	0.0046	0.0104	0.0063	0.0057
2325.59	0.0007	0.0030	0.0081	0.0171	0.0154
2583.98	0.0007	0.0049	0.0082	0.0046	0.0104
2842.38	0.0007	0.0036	0.0050	0.0142	0.0130
3100.78	0.0007	0.0027	0.0040	0.0069	0.0115
3359.18	0.0006	0.0025	0.0016	0.0025	0.0018

Tabela C.1: Espectro do ataque da nota Dó ($\sim$ 360 Hz) tocada por uma flauta transversal. (sem vibrato)

1. Taxa de Amostragem: 44.1 kHz
2. Duração: 2.554218 seg
3. Nr. Amostras: 112641
4. Formato: 16 bits linear RIFF/WAV
5. Mx. Ampl.: 3.318

freqüência	t_1	t_2	t_3	t_4	t_5
258.40	0.0252	0.1183	0.2034	0.2220	0.2092
516.80	0.0314	0.1447	0.2438	0.2584	0.2386
775.20	0.0051	0.0208	0.0333	0.0382	0.0379
1033.59	0.0079	0.0401	0.0728	0.0837	0.0827
1291.99	0.0044	0.0199	0.0315	0.0280	0.0196
1550.39	0.0087	0.0384	0.0585	0.0580	0.0618
1808.79	0.0099	0.0413	0.0631	0.0607	0.0573
2067.19	0.0057	0.0219	0.0308	0.0259	0.0202
2325.59	0.0042	0.0132	0.0169	0.0137	0.0102
2583.98	0.0042	0.0168	0.0192	0.0128	0.0111
2842.38	0.0026	0.0076	0.0053	0.0005	0.0007
3100.78	0.0016	0.0047	0.0049	0.0059	0.0062
3359.18	0.0009	0.0023	0.0046	0.0034	0.0005

Tabela C.2: Espectro do estado estacionário da nota Dó ($\sim$ 360 Hz) tocada por uma flauta transversal. (sem vibrato)

freqüência	t_1	t_2	t_3	t_4	t_5
441.43	0.0065	0.0905	0.3106	0.5389	0.6064
882.86	0.0043	0.0423	0.1283	0.1971	0.2109
1324.29	0.0041	0.0401	0.0940	0.0828	0.1057
1765.72	0.0019	0.0142	0.0499	0.0936	0.1020
2207.15	0.0017	0.0171	0.0406	0.0332	0.0368
2648.58	0.0003	0.0032	0.0133	0.0107	0.0056
3090.01	0.0008	0.0073	0.0102	0.0051	0.0085
3531.45	0.0008	0.0019	0.0047	0.0070	0.0065
3972.88	0.0007	0.0048	0.0100	0.0082	0.0094
4414.31	0.0007	0.0037	0.0057	0.0046	0.0029
5297.17	0.0003	0.0018	0.0032	0.0025	0.0009
5738.60	0.0001	0.0006	0.0021	0.0038	0.0029
6180.03	0.0002	0.0024	0.0038	0.0046	0.0054

Tabela C.3: Espectro do ataque da nota Lá ($\sim$ 440 Hz) tocada por uma flauta transversal. (sem vibrato)

freqüência	t_1	t_2	t_3	t_4	t_5
441.43	0.1116	0.5333	0.9346	1.0253	0.9314
882.86	0.0618	0.2761	0.4668	0.5099	0.4576
1324.29	0.0865	0.4058	0.6756	0.7123	0.6739
1765.72	0.0241	0.0982	0.1581	0.1676	0.1401
2207.15	0.0358	0.1525	0.2358	0.2615	0.2634
2648.58	0.0039	0.0139	0.0192	0.0223	0.0271
3090.01	0.0038	0.0140	0.0204	0.0189	0.0197
3531.45	0.0029	0.0130	0.0193	0.0158	0.0151
3972.88	0.0028	0.0116	0.0138	0.0144	0.0221
4414.31	0.0017	0.0059	0.0073	0.0097	0.0115
5297.17	0.0012	0.0036	0.0055	0.0080	0.0152
5738.60	0.0004	0.0031	0.0096	0.0121	0.0095
6180.03	0.0005	0.0007	0.0028	0.0049	0.0020

Tabela C.4: Espectro do estado estacionário da nota Lá ($\sim$ 440 Hz) tocada por uma flauta transversal. (sem vibrato)

Flauta Transversal com Vibrato

Detalhes Técnicos:

Performer: Sonja Feig

Instrumento:

Data da Gravação: 20 de dezembro de 1997

Local da Gravação: Wendell Johnson Speech and Hearing Center, University of Iowa.

Altura do Microfone:

Altura do Instrumento:

Distância do Microfone:

Dó

1. Taxa de Amostragem: 44.1 kHz

2. Duração: 2.523605 seg
3. Nr. Amostras: 111291
4. Formato: 16 bits linear RIFF/WAV
5. Mx. Ampl.: 2.087

Lá

1. Taxa de Amostragem: 44.1 kHz
2. Duração: 2.595193 seg
3. Nr. Amostras: 114448
4. Formato: 16 bits linear RIFF/WAV
5. Mx. Ampl.: 3.716

freqüência	t_1	t_2	t_3	t_4	t_5
258.40	0.0326	0.1404	0.2143	0.1985	0.1749
516.80	0.0334	0.1325	0.1692	0.1119	0.1063
775.20	0.0037	0.0139	0.0167	0.0118	0.0083
1033.59	0.0021	0.0037	0.0073	0.0106	0.0053
1291.99	0.0057	0.0166	0.0131	0.0058	0.0106
1550.39	0.0020	0.0064	0.0084	0.0114	0.0116
1808.79	0.0024	0.0076	0.0066	0.0087	0.0179
2067.19	0.0016	0.0083	0.0075	0.0060	0.0027
2325.59	0.0024	0.0080	0.0055	0.0034	0.0052
2583.98	0.0017	0.0059	0.0060	0.0007	0.0058
2842.38	0.0014	0.0071	0.0087	0.0039	0.0018
3100.78	0.0029	0.0073	0.0038	0.0006	0.0050
3359.18	0.0022	0.0055	0.0039	0.0021	0.0026

Tabela C.5: Espectro do ataque da nota Dó ($\sim$ 360 Hz) tocada por uma flauta transversal. (com vibrato)

freqüência	t_1	t_2	t_3	t_4	t_5
258.40	0.0326	0.1404	0.2143	0.1985	0.1749
516.80	0.0334	0.1325	0.1692	0.1119	0.1063
775.20	0.0037	0.0139	0.0167	0.0118	0.0083
1033.59	0.0021	0.0037	0.0073	0.0106	0.0053
1291.99	0.0057	0.0166	0.0131	0.0058	0.0106
1550.39	0.0020	0.0064	0.0084	0.0114	0.0116
1808.79	0.0024	0.0076	0.0066	0.0087	0.0179
2067.19	0.0016	0.0083	0.0075	0.0060	0.0027
2325.59	0.0024	0.0080	0.0055	0.0034	0.0052
2583.98	0.0017	0.0059	0.0060	0.0007	0.0058
2842.38	0.0014	0.0071	0.0087	0.0039	0.0018
3100.78	0.0029	0.0073	0.0038	0.0006	0.0050
3359.18	0.0022	0.0055	0.0039	0.0021	0.0026

Tabela C.6: Espectro do estado estacionário da nota Dó ($\sim$ 360 Hz) tocada por uma flauta transversal. (com vibrato)

freqüência	t_1	t_2	t_3	t_4	t_5
441.43	0.0234	0.2596	0.7663	1.1699	1.2232
882.86	0.0026	0.0322	0.1115	0.1763	0.1925
1324.29	0.0022	0.0771	0.3151	0.4604	0.4163
1765.72	0.0014	0.0244	0.1065	0.1999	0.2957
2207.15	0.0016	0.0086	0.0543	0.0768	0.0735
2648.58	0.0007	0.0052	0.0117	0.0115	0.0290
3090.01	0.0006	0.0051	0.0130	0.0127	0.0070
3531.45	0.0007	0.0012	0.0036	0.0030	0.0125
3972.88	0.0020	0.0099	0.0129	0.0069	0.0033
4414.31	0.0014	0.0042	0.0045	0.0019	0.0077
5297.17	0.0005	0.0022	0.0061	0.0063	0.0065
5738.60	0.0005	0.0031	0.0070	0.0081	0.0069
6180.03	0.0002	0.0010	0.0033	0.0033	0.0038

Tabela C.7: Espectro do ataque da nota Lá ($\sim$ 440 Hz) tocada por uma flauta transversal. (com vibrato)

freqüência	t_1	t_2	t_3	t_4	t_5
441.43	0.1664	0.7361	1.1492	1.1892	1.1910
882.86	0.0314	0.1336	0.1843	0.2347	0.3475
1324.29	0.0339	0.1260	0.1150	0.1300	0.2358
1765.72	0.0607	0.1941	0.1368	0.1799	0.4246
2207.15	0.0066	0.0284	0.0187	0.0209	0.0571
2648.58	0.0047	0.0131	0.0114	0.0157	0.0189
3090.01	0.0017	0.0017	0.0037	0.0099	0.0166
3531.45	0.0019	0.0020	0.0024	0.0054	0.0060
3972.88	0.0033	0.0051	0.0007	0.0104	0.0201
4414.31	0.0019	0.0043	0.0047	0.0062	0.0061
5297.17	0.0018	0.0045	0.0022	0.0014	0.0003
5738.60	0.0010	0.0035	0.0041	0.0028	0.0065
6180.03	0.0016	0.0042	0.0021	0.0018	0.0041

Tabela C.8: Espectro do estado estacionário da nota Lá ($\sim$ 360 Hz) tocada por uma flauta transversal. (com vibrato)

Oboé

Detalhes Técnicos:

Performer: Dr. Kira Wharton

Instrumento: Howarth 55 (1996) full conservatory, 3rd octave key

Data da Gravação: 23 de julho de 1997

Local da Gravação: Wendell Johnson Speech and Hearing Center, University of Iowa.

Altura do Microfone: 32”

Altura do Instrumento: 22”

Distância do Microfone: 36”

Dó

1. Taxa de Amostragem: 44.1 kHz
2. Duração: 1.404467 seg
3. Nr. Amostras: 61937
4. Formato: 16 bits linear RIFF/WAV
5. Mx. Ampl.: 3.900

Lá

1. Taxa de Amostragem: 44.1 kHz
2. Duração: 1.589569 seg
3. Nr. Amostras: 70100

freqüência	t_1	t_2	t_3	t_4	t_5
258.40	0.0031	0.0394	0.1425	0.2629	0.2889
516.80	0.0033	0.0609	0.2259	0.3665	0.3586
775.20	0.0034	0.0741	0.2655	0.3911	0.3642
1033.59	0.0033	0.0771	0.2819	0.3961	0.3387
1291.99	0.0033	0.1431	0.6213	0.8972	0.8053
1550.39	0.0021	0.0482	0.1903	0.2272	0.1835
1808.79	0.0015	0.0265	0.0929	0.0874	0.0442
2067.19	0.0012	0.0108	0.0237	0.0111	0.0074
2325.59	0.0007	0.0015	0.0218	0.0438	0.0338
2583.98	0.0004	0.0042	0.0270	0.0356	0.0220
2842.38	0.0006	0.0067	0.0205	0.0177	0.0159
3100.78	0.0002	0.0035	0.0139	0.0150	0.0031
3359.18	0.0001	0.0052	0.0128	0.0082	0.0033

Tabela C.9: Espectro do ataque da nota Dó ($\sim$ 360 Hz) tocada por um oboé.

4. Formato: 16 bits linear RIFF/WAV

5. Mx. Ampl.: 3.064

freqüência	t_1	t_2	t_3	t_4	t_5
258.40	0.0378	0.1797	0.3127	0.3423	0.3118
516.80	0.0445	0.2138	0.3684	0.3985	0.3680
775.20	0.0520	0.2381	0.3934	0.4183	0.3910
1033.59	0.0483	0.2271	0.3694	0.3864	0.3609
1291.99	0.1453	0.6508	0.9910	0.9955	0.9739
1550.39	0.0403	0.1853	0.2633	0.2470	0.2559
1808.79	0.0191	0.0745	0.0968	0.0900	0.0934
2067.19	0.0065	0.0217	0.0213	0.0149	0.0148
2325.59	0.0028	0.0136	0.0168	0.0130	0.0131
2583.98	0.0077	0.0180	0.0151	0.0125	0.0191
2842.38	0.0041	0.0150	0.0104	0.0067	0.0097
3100.78	0.0037	0.0087	0.0031	0.0020	0.0008
3359.18	0.0018	0.0076	0.0065	0.0026	0.0030

Tabela C.10: Espectro do estado estacionário da nota Dó ($\sim$ 360 Hz) tocada por um oboé.

freqüência	t_1	t_2	t_3	t_4	t_5
441.43	0.0084	0.0593	0.1235	0.1464	0.1326
882.86	0.0150	0.1235	0.2758	0.3366	0.3062
1324.29	0.0263	0.2586	0.6206	0.7852	0.7191
1765.72	0.0143	0.1780	0.4739	0.6330	0.5884
2207.15	0.0040	0.0347	0.0819	0.1043	0.0969
2648.58	0.0007	0.0161	0.0507	0.0685	0.0612
3090.01	0.0003	0.0149	0.0433	0.0563	0.0507
3531.45	0.0024	0.0198	0.0449	0.0513	0.0440
3972.88	0.0008	0.0026	0.0046	0.0077	0.0076
4414.31	0.0018	0.0146	0.0332	0.0377	0.0312
5297.17	0.0004	0.0031	0.0084	0.0114	0.0110
5738.60	0.0008	0.0044	0.0116	0.0143	0.0106
6180.03	0.0006	0.0024	0.0058	0.0085	0.0088

Tabela C.11: Espectro do ataque da nota Lá ($\sim$ 440 Hz) tocada por um oboé.

freqüência	t_1	t_2	t_3	t_4	t_5
441.43	0.0086	0.0662	0.1352	0.1563	0.1508
882.86	0.0386	0.2181	0.4074	0.4541	0.4212
1324.29	0.1501	0.7125	1.2522	1.3819	1.2581
1765.72	0.1398	0.6600	1.1493	1.2588	1.1428
2207.15	0.0140	0.0625	0.1087	0.1219	0.1146
2648.58	0.0091	0.0416	0.0721	0.0802	0.0730
3090.01	0.0140	0.0678	0.1214	0.1353	0.1209
3531.45	0.0185	0.0865	0.1521	0.1662	0.1391
3972.88	0.0068	0.0283	0.0443	0.0407	0.0291
4414.31	0.0106	0.0413	0.0585	0.0588	0.0560
5297.17	0.0021	0.0090	0.0121	0.0121	0.0137
5738.60	0.0001	0.0012	0.0033	0.0072	0.0091
6180.03	0.0008	0.0043	0.0071	0.0031	0.0090

Tabela C.12: Espectro do estado estacionário da nota Lá ($\sim$ 440 Hz) tocada por um oboé.

Clarinete em Si bemol

Detalhes Técnicos:

Performer: Christine Bello

Instrumento:

Data da Gravação:

Local da Gravação: Wendell Johnson Speech and Hearing Center, University of Iowa.

Altura do Microfone: 32”

Altura do Instrumento: 22”

Distância do Microfone: 36”

Dó

1. Taxa de Amostragem: 44.1 kHz

2. Duração: 2.419070 seg

freqüência	t_1	t_2	t_3	t_4	t_5
258.40	0.0092	0.1391	0.5718	1.1436	1.3610
516.80	0.0008	0.0034	0.0138	0.0253	0.0268
775.20	0.0058	0.0406	0.0111	0.2807	0.5526
1033.59	0.0008	0.0027	0.0060	0.0354	0.0676
1291.99	0.0028	0.0147	0.0134	0.0361	0.0729
1550.39	0.0009	0.0037	0.0046	0.0077	0.0260
1808.79	0.0005	0.0025	0.0004	0.0048	0.0081
2067.19	0.0006	0.0017	0.0017	0.0056	0.0079
2325.59	0.0003	0.0021	0.0045	0.0062	0.0057
2583.98	0.0003	0.0012	0.0014	0.0024	0.0040
2842.38	0.0008	0.0052	0.0062	0.0046	0.0053
3100.78	0.0001	0.0023	0.0049	0.0044	0.0070
3359.18	0.0007	0.0034	0.0054	0.0051	0.0061

Tabela C.13: Espectro do ataque da nota Dó ($\sim$ 360 Hz) tocada por um clarinete em si bemol.

3. Nr. Amostras: 106681

4. Formato: 16 bits linear RIFF/WAV

5. Mx. Ampl.: 3.906

Lá

1. Taxa de Amostragem: 44.1 kHz

2. Duração: 2.557052 seg

3. Nr. Amostras: 112766

4. Formato: 16 bits linear RIFF/WAV

5. Mx. Ampl.: 3.888

freqüência	t_1	t_2	t_3	t_4	t_5
258.40	0.1812	0.8825	1.5371	1.6727	1.5387
516.80	0.0130	0.0365	0.0416	0.0320	0.0295
775.20	0.1358	0.6193	0.9537	0.9636	0.9596
1033.59	0.0095	0.0670	0.1082	0.0951	0.1076
1291.99	0.0383	0.1481	0.1837	0.1500	0.1776
1550.39	0.0102	0.0512	0.0527	0.0323	0.0398
1808.79	0.0078	0.0208	0.0115	0.0058	0.0090
2067.19	0.0061	0.0163	0.0114	0.0038	0.0044
2325.59	0.0055	0.0166	0.0084	0.0034	0.0076
2583.98	0.0033	0.0060	0.0029	0.0003	0.0022
2842.38	0.0051	0.0035	0.0029	0.0028	0.0065
3100.78	0.0034	0.0075	0.0028	0.0024	0.0057
3359.18	0.0041	0.0086	0.0042	0.0022	0.0025

Tabela C.14: Espectro do estado estacionário da nota Dó ($\sim$ 360 Hz) tocada por um clarinete em si bemol.

freqüência	t_1	t_2	t_3	t_4	t_5
441.43	0.0081	0.1324	0.5370	1.1210	1.4273
882.86	0.0007	0.0091	0.0574	0.1656	0.2770
1324.29	0.0036	0.0207	0.0318	0.0963	0.1828
1765.72	0.0005	0.0062	0.0121	0.0239	0.0658
2207.15	0.0006	0.0056	0.0183	0.0322	0.0444
2648.58	0.0005	0.0031	0.0038	0.0102	0.0213
3090.01	0.0001	0.0039	0.0090	0.0023	0.0127
3531.45	0.0005	0.0025	0.0029	0.0102	0.0183
3972.88	0.0003	0.0019	0.0052	0.0078	0.0128
4414.31	0.0006	0.0022	0.0051	0.0021	0.0104
5297.17	0.0002	0.0024	0.0059	0.0066	0.0060
5738.60	0.0004	0.0012	0.0014	0.0034	0.0024
6180.03	0.0002	0.0018	0.0036	0.0057	0.0063

Tabela C.15: Espectro do ataque da nota Lá ($\sim$ 440 Hz) tocada por um clarinete em si bemol.

freqüência	t_1	t_2	t_3	t_4	t_5
441.43	0.2418	1.1562	2.0102	2.1846	1.9893
882.86	0.0803	0.3748	0.6347	0.6678	0.5983
1324.29	0.0483	0.2353	0.3975	0.4241	0.3875
1765.72	0.0154	0.0725	0.1219	0.1358	0.1337
2207.15	0.0114	0.0520	0.0768	0.0825	0.0820
2648.58	0.0083	0.0264	0.0306	0.0372	0.0367
3090.01	0.0079	0.0251	0.0269	0.0211	0.0240
3531.45	0.0015	0.0038	0.0049	0.0034	0.0060
3972.88	0.0025	0.0059	0.0061	0.0013	0.0036
4414.31	0.0016	0.0060	0.0053	0.0033	0.0038
5297.17	0.0018	0.0050	0.0028	0.0040	0.0035
5738.60	0.0021	0.0040	0.0029	0.0044	0.0051
6180.03	0.0005	0.0014	0.0008	0.0028	0.0027

Tabela C.16: Espectro do estado estacionário da nota Lá ($\sim$ 440 Hz) tocada por um clarinete em si bemol.

Fagote

Detalhes Técnicos:

Performer: Dr. Max Wharton

Instrumento: Fox 601 #23385, Bocal - Heckel cc2 unplated

Data da Gravação: 23 de julho de 1997

Local da Gravação: Wendell Johnson Speech and Hearing Center, University of Iowa.

Altura do Microfone: 54"

Altura do Instrumento: 48"

Distância do Microfone: 54"

Dó

1. Taxa de Amostragem: 44.1 kHz

2. Duração: 2.270748 seg

3. Nr. Amostras: 100140

4. Formato: 16 bits linear RIFF/WAV

5. Mx. Ampl.: 10.941

Lá

1. Taxa de Amostragem: 44.1 kHz

2. Duração: 2.314671 seg

3. Nr. Amostras: 102518

4. Formato: 16 bits linear RIFF/WAV

5. Mx. Ampl.: 7.928

freqüência	t_1	t_2	t_3	t_4	t_5
64.60	0.0091	0.0625	0.1632	0.2560	0.2749
129.20	0.0142	0.1124	0.2911	0.4233	0.4152
193.80	0.0178	0.1533	0.3564	0.4614	0.4507
258.40	0.0141	0.1276	0.2876	0.3514	0.3298
323.00	0.0150	0.1672	0.4218	0.5365	0.4915
387.60	0.0225	0.2622	0.6790	0.8621	0.8048
452.20	0.0266	0.3308	0.9116	1.1945	1.2027
516.80	0.0213	0.2661	0.6794	0.7578	0.8655
581.40	0.0116	0.1458	0.3637	0.4034	0.4793
646.00	0.0062	0.0870	0.2043	0.2016	0.2413
710.60	0.0060	0.0430	0.0999	0.1066	0.1223
775.20	0.0069	0.0387	0.1032	0.0948	0.1194
839.79	0.0043	0.0240	0.0602	0.0220	0.0291

Tabela C.17: Espectro do ataque da nota Dó ($\sim$ 65 Hz) tocada por um fagote.

freqüência	t_1	t_2	t_3	t_4	t_5
64.60	0.0376	0.2105	0.3851	0.4331	0.3969
129.20	0.0584	0.3020	0.5220	0.5635	0.5149
193.80	0.0610	0.3255	0.5585	0.5912	0.5542
258.40	0.0591	0.2687	0.4124	0.4139	0.3923
323.00	0.0793	0.3645	0.5482	0.5585	0.5408
387.60	0.1351	0.5993	0.8406	0.8193	0.8308
452.20	0.2040	0.8920	1.1500	1.0510	1.1361
516.80	0.1802	0.7084	0.7925	0.6482	0.7993
581.40	0.1207	0.4248	0.3893	0.3042	0.4016
646.00	0.0671	0.2239	0.1629	0.1113	0.1701
710.60	0.0339	0.0961	0.0543	0.0427	0.0684
775.20	0.0280	0.0869	0.0424	0.0161	0.0351
839.79	0.0071	0.0047	0.0062	0.0040	0.0073

Tabela C.18: Espectro do estado estacionário da nota Dó ($\sim$ 65 Hz) tocada por um fagote.

freqüência	t_1	t_2	t_3	t_4	t_5
107.67	0.0178	0.0999	0.2010	0.2492	0.2320
215.33	0.0291	0.1560	0.2771	0.3036	0.2917
323.00	0.0322	0.1580	0.2541	0.2465	0.2534
441.43	0.1148	0.8184	1.7332	2.0797	1.9163
549.10	0.1037	0.8089	1.8002	2.2609	2.2204
656.76	0.0187	0.0925	0.1443	0.1546	0.1412
775.20	0.0092	0.0878	0.2753	0.4625	0.5050
882.86	0.0066	0.0679	0.1950	0.2880	0.2842
990.53	0.0029	0.0272	0.0809	0.1286	0.1415
1098.19	0.0017	0.0065	0.0333	0.0533	0.0533
1205.86	0.0056	0.0442	0.0800	0.0801	0.0931
1324.29	0.0053	0.0589	0.1609	0.2303	0.2284
1431.96	0.0019	0.0216	0.0585	0.0833	0.0864

Tabela C.19: Espectro do ataque da nota Lá ($\sim$ 110 Hz) tocada por um fagote.

freqüência	t_1	t_2	t_3	t_4	t_5
107.67	0.0390	0.1783	0.3017	0.3245	0.3074
215.33	0.0612	0.2453	0.3691	0.3812	0.3804
323.00	0.0809	0.2638	0.3166	0.3018	0.3345
441.43	0.3188	1.4966	2.5850	2.8157	2.5392
549.10	0.3534	1.6631	2.7537	2.9043	2.6751
656.76	0.0659	0.1681	0.1589	0.1588	0.0828
775.20	0.0642	0.3508	0.6457	0.7194	0.6193
882.86	0.0447	0.2396	0.4297	0.4685	0.4625
990.53	0.0249	0.1225	0.1976	0.2037	0.1698
1098.19	0.0118	0.0572	0.0838	0.0816	0.1056
1205.86	0.0237	0.0903	0.1087	0.0932	0.0968
1324.29	0.0439	0.1970	0.3205	0.3344	0.3024
1431.96	0.0188	0.0769	0.1075	0.0970	0.0797

Tabela C.20: Espectro do estado estacionário da nota Lá ($\sim$ 110 Hz) tocada por um fagote.

Trompa

Detalhes Técnicos:

Performer: Jessica Wiskus

Instrumento: French Horn

Data da Gravação: 23 de junho de 1997

Local da Gravação: Wendell Johnson Speech and Hearing Center, University of Iowa.

Altura do Microfone:

Altura do Instrumento:

Distância do Microfone:

Dó

1. Taxa de Amostragem: 44.1 kHz
2. Duração: 2.591769 seg
3. Nr. Amostras: 114297
4. Formato: 16 bits linear RIFF/WAV
5. Mx. Ampl.: 5.067

Lá

1. Taxa de Amostragem: 44.1 kHz
2. Duração: 2.516531 seg

freqüência	t_1	t_2	t_3	t_4	t_5
258.40	0.0259	0.2508	0.7071	1.0638	1.1018
516.80	0.0023	0.0898	0.4383	0.7561	0.8412
775.20	0.0048	0.0459	0.1371	0.1992	0.2046
1033.59	0.0035	0.0182	0.0359	0.0474	0.0362
1291.99	0.0022	0.0034	0.0048	0.0041	0.0060
1550.39	0.0023	0.0104	0.0147	0.0115	0.0094
1808.79	0.0023	0.0080	0.0068	0.0097	0.0111
2067.19	0.0010	0.0047	0.0073	0.0093	0.0066
2325.59	0.0008	0.0016	0.0033	0.0031	0.0050
2583.98	0.0010	0.0036	0.0036	0.0017	0.0033
2842.38	0.0006	0.0041	0.0055	0.0040	0.0026
3100.78	0.0007	0.0025	0.0031	0.0034	0.0035
3359.18	0.0004	0.0015	0.0035	0.0027	0.0027

Tabela C.21: Espectro do ataque da nota Dó ($\sim$ 360 Hz) tocada por uma trompa em fá.

3. Nr. Amostras: 110979

4. Formato: 16 bits linear RIFF/WAV

5. Mx. Ampl.: 11.341

freqüência	t_1	t_2	t_3	t_4	t_5
258.40	0.1906	0.8386	1.3937	1.4963	1.4078
516.80	0.2217	0.9470	1.4801	1.5304	1.4878
775.20	0.0988	0.3741	0.5051	0.4908	0.5043
1033.59	0.0391	0.1310	0.1424	0.1244	0.1458
1291.99	0.0196	0.0590	0.0520	0.0341	0.0509
1550.39	0.0056	0.0149	0.0101	0.0030	0.0137
1808.79	0.0039	0.0104	0.0069	0.0047	0.0122
2067.19	0.0013	0.0021	0.0017	0.0019	0.0037
2325.59	0.0016	0.0031	0.0032	0.0014	0.0033
2583.98	0.0019	0.0032	0.0016	0.0011	0.0032
2842.38	0.0003	0.0013	0.0016	0.0014	0.0018
3100.78	0.0003	0.0010	0.0008	0.0009	0.0027
3359.18	0.0013	0.0027	0.0015	0.0005	0.0023

Tabela C.22: Espectro do estado estacionário da nota Dó ($\sim$ 360 Hz) tocada por uma trompa em fá.

freqüência	t_1	t_2	t_3	t_4	t_5
441.43	0.2833	1.9953	4.0977	4.8266	4.6959
882.86	0.0681	0.5571	1.1486	1.3162	1.3871
1324.29	0.0133	0.1217	0.2097	0.2162	0.3059
1765.72	0.0052	0.0576	0.0709	0.0623	0.1415
2207.15	0.0027	0.0306	0.0219	0.0145	0.0606
2648.58	0.0008	0.0074	0.0042	0.0033	0.0193
3090.01	0.0001	0.0028	0.0055	0.0033	0.0075
3531.45	0.0020	0.0090	0.0122	0.0061	0.0051
3972.88	0.0004	0.0020	0.0050	0.0051	0.0057
4414.31	0.0005	0.0007	0.0008	0.0027	0.0061
5297.17	0.0009	0.0031	0.0029	0.0029	0.0068
5738.60	0.0008	0.0024	0.0039	0.0051	0.0067
6180.03	0.0006	0.0026	0.0032	0.0029	0.0007

Tabela C.23: Espectro do ataque da nota Lá ($\sim$ 440 Hz) tocada por uma trompa em fá.

freqüência	t_1	t_2	t_3	t_4	t_5
441.43	0.6996	3.3319	5.7855	6.2912	5.7680
882.86	0.2538	1.2277	2.1117	2.2623	2.0957
1324.29	0.0632	0.3432	0.6033	0.6418	0.6201
1765.72	0.0407	0.2009	0.3366	0.3471	0.3383
2207.15	0.0320	0.1362	0.2057	0.2035	0.1956
2648.58	0.0082	0.0281	0.0364	0.0377	0.0451
3090.01	0.0074	0.0250	0.0284	0.0220	0.0169
3531.45	0.0011	0.0089	0.0156	0.0148	0.0182
3972.88	0.0036	0.0108	0.0088	0.0049	0.0083
4414.31	0.0017	0.0021	0.0008	0.0007	0.0024
5297.17	0.0013	0.0018	0.0017	0.0024	0.0053
5738.60	0.0024	0.0047	0.0032	0.0060	0.0069
6180.03	0.0012	0.0024	0.0015	0.0012	0.0024

Tabela C.24: Espectro do estado estacionário da nota Lá ($\sim$ 440 Hz) tocada por uma trompa em fá.