Universidade de São Paulo
Instituto de Física de São Carlos
Departamento de Física e Informática

Conversão de Voz Baseada na Transformada
Wavelet

Lucimar Sasso Vieira

São Carlos - SP - Brasil
Março de 2007

Universidade de São Paulo
Instituto de Física de São Carlos
Departamento de Física e Informática

Conversão de Voz Baseada na Transformada
Wavelet

Lucimar Sasso Vieira

Dissertação apresentada ao Instituto de Física de São Carlos da Universidade de São Paulo, como parte dos requisitos necessários à obtenção do título de Mestre em Ciências - Física Aplicada - Opção Física Computacional.

Orientador: Prof. Dr. Rodrigo Capobianco Guido

São Carlos - SP - Brasil
Março de 2007

”Lutar para ser admirado pelos outros é tolice. Realize sonhos com a naturalidade de um rio que sabe por onde corre e não para que alguém o aplauda”...

Roberto Shinyashiki

”Dedico esta dissertação ao meu orientador,
Rodrigo Capobianco Guido, pela atenção e
paciência diante das dificuldades encontradas
durante a realização deste trabalho”.

Agradecimentos

Gostaria de agradecer:

- a paciência recebida de toda a minha família, durante as incansáveis noites em que, sem dormir, eu necessitava de apoio ;
- pela fraternidade, que sempre percebi nas faces dos meus amigos ;
- e aos meus queridos professores e funcionários do IFSC, pelo apoio fundamental dado durante a realização deste trabalho.

Sumário

1	Introdução e Motivação	12
2	Revisão da Literatura	14
2.1	Conceitos Elementares de Processamento de Sinais	14
2.1.1	Processamento de sinais: analógico X digital	14
2.1.2	Sinais e sistemas	15
2.1.3	Filtros digitais	16
2.1.4	Teorema da convolução	18
2.1.5	Resposta ao impulso de um filtro digital	18
2.1.6	Teorema da amostragem	19
2.1.7	<i>Aliasing</i>	19
2.1.8	Transformada Discreta de Fourier	19
2.1.9	Transformada Z	19
2.1.10	Função de transferência	20
2.1.11	A Transformada de Fourier de Tempo Reduzido (STFT)	20
2.2	A Transformada <i>wavelet</i> Discreta (DWT)	20
2.2.1	Cálculo da DWT	25
2.2.2	Cálculo da DWT inversa (IDWT)	26
2.2.3	Momentos Nulos	26
2.2.4	Famílias de Transformadas <i>Wavelet</i>	27
2.3	Princípios Elementares de Redes Neurais Artificiais (RNAs)	29
2.4	Os Sistemas Geradores de Voz	32
2.4.1	O sistema bio-físico de produção de voz	32
2.4.2	Síntese de voz	33
2.5	Interpretação de Voz	34
2.5.1	O Sistema bio-físico de interpretação de fala	35
2.5.2	Reconhecimento de fala por intermédio computacional	36
2.5.3	O <i>cepstrum</i>	37
2.6	Algoritmos para <i>Voice Morphing</i>	38

3	Descrição do Sistema Proposto	41
3.1	A Arquitetura do sistema proposto	41
3.2	O Algoritmo de <i>voice morphing</i>	41
4	Testes e Resultados	46
4.1	Testes de Convergência	46
4.2	Testes de comparação dos resultados com distância Euclidiana . .	47
4.3	Testes de comparação dos resultados via critério perceptual	48
5	Conclusões e Trabalhos Futuros	55
	Apêndice I - Coeficientes dos filtros <i>wavelet</i> utilizados nas experiências.	57
	Apêndice II - Código fonte do algoritmo conversor em linguagem C++.	69
	Apêndice III - Publicações durante o mestrado.	112
	Referências Bibliográficas	114

Lista de Tabelas

2.1	Características das famílias de <i>wavelets</i> utilizadas no presente trabalho, incluindo a quantidade de momentos da função <i>wavelet</i> . . .	27
2.2	As 25 bandas críticas do sistema auditivo humano.	37
3.1	Escala Bark (B.S.), banda crítica correspondente (B.C.), banda crítica proposta (B.P.) e índice da amostra (I.A.) da DWT que leva a tais frequências, levando em conta que cada <i>frame</i> do sinal tem 256 amostras e foi amostrado a uma taxa de 16000 amostras por segundo.	44
4.1	Parte superior: Convergência do algoritmo para converter a sentença <i>sa1.wav</i> do diretório <i>/timit/test/dr1/mdab0</i> na <i>sa1.wav</i> do diretório <i>/timit/test/dr1/mjsw0</i> , onde ambos os locutores são do sexo masculino (M → M) ; Parte inferior: idem para converter a sentença <i>sa1.wav</i> do diretório <i>/timit/test/dr1/faks0</i> na <i>sa1.wav</i> do diretório <i>/timit/test/dr1/fadac1</i> , onde ambos os locutores são do sexo feminino (F → F); Legenda: DWT: base <i>wavelet</i> , SW: suporte correspondente dos filtros , QI: quantidade máxima de iterações para atingir um erro de $< 10^{-3}$ para cada amostra do sinal.	49
4.2	Parte superior: Convergência do algoritmo para converter a sentença <i>sa1.wav</i> do diretório <i>/timit/test/dr1/mwbt0</i> na <i>sa1.wav</i> do diretório <i>/timit/test/dr1/fjem0</i> , onde o <i>source speaker</i> é do sexo masculino e o <i>target speaker</i> é do sexo feminino (M → F) ; Parte inferior: idem para converter a sentença <i>sa1.wav</i> do diretório <i>/timit/test/dr1/fjem0</i> na <i>sa1.wav</i> do diretório <i>/timit/test/dr1/mwbt0</i> , onde <i>source speaker</i> é do sexo feminino e o <i>target speaker</i> é do sexo masculino (F → M); Legenda: DWT: base <i>wavelet</i> , SW: suporte correspondente dos filtros , QI: quantidade máxima de iterações para atingir um erro de $< 10^{-3}$ para cada amostra do sinal.	50

- 4.3 Distância Euclidiana entre a sentença *sa1.wav* do diretório */timit/ test/ dr1/ mdab0* e a *sa1.wav* do diretório */timit/ test/ dr1/ mjsw0*, onde o *source speaker* e o *target speaker* são do sexo masculino ($M \rightarrow M$) ; Parte inferior: idem para as sentenças *sa1.wav* do diretório */timit/ test/ dr1/ faks0* e *sa1.wav* do diretório */timit/ test/ dr1/ fadac1*, respectivamente, onde o *source speaker* e o *target speaker* são do sexo feminino ($F \rightarrow F$); Legenda: DWT: base *wavelet* , SW: suporte correspondente dos filtros , DI: distância Euclidiana entre os sinais (original e convertido). 51
- 4.4 Distância Euclidiana entre a sentença *sa1.wav* do diretório */timit/ test/ dr1/ mwbt0* e a *sa1.wav* do diretório */timit/ test/ dr1/ fjem0*, onde o *source speaker* é do sexo masculino e o *target speaker* é do sexo feminino ($M \rightarrow F$) ; Parte inferior: idem para as sentenças *sa1.wav* do diretório */timit/ test/ dr1/ fjem0* e *sa1.wav* do diretório */timit/ test/ dr1/ mwbt0*, respectivamente, ou seja, o *source speaker* é do sexo feminino e o *target speaker* é do sexo masculino ($F \rightarrow M$); Legenda: DWT: base *wavelet* , SW: suporte correspondente dos filtros , DI: distância Euclidiana entre os sinais (original e convertido). 52
- 4.5 Distância Euclidiana entre a sentença *sa1.wav* do diretório */timit/ test/ dr1/ mdab0* e a *sa1.wav* do diretório */timit/ test/ dr1/ mjsw0*, onde o *source speaker* e o *target speaker* são do sexo masculino ($M \rightarrow M$) ; Parte inferior: idem para as sentenças *sa1.wav* do diretório */timit/ test/ dr1/ faks0* e *sa1.wav* do diretório */timit/ test/ dr1/ fadac1*, respectivamente, onde o *source speaker* e o *target speaker* são do sexo feminino ($F \rightarrow F$); Legenda: DWT: base *wavelet* , SW: suporte correspondente dos filtros , M,D: média e desvio-padrão das notas atribuídas pelos ouvintes para a qualidade da conversão das vozes. 53
- 4.6 Distância Euclidiana entre a sentença *sa1.wav* do diretório */timit/ test/ dr1/ mwbt0* e a *sa1.wav* do diretório */timit/ test/ dr1/ fjem0*, onde o *source speaker* é do sexo masculino e o *target speaker* é do sexo feminino ($M \rightarrow F$) ; Parte inferior: idem para as sentenças *sa1.wav* do diretório */timit/ test/ dr1/ fjem0* e *sa1.wav* do diretório */timit/ test/ dr1/ mwbt0*, respectivamente, ou seja, o *source speaker* é do sexo feminino e o *target speaker* é do sexo masculino ($F \rightarrow M$); Legenda: DWT: base *wavelet* , SW: suporte correspondente dos filtros , M,D: média e desvio-padrão das notas atribuídas pelos ouvintes para a qualidade da conversão das vozes. . . . 54

Lista de Figuras

2.1	Exemplo dos principais parâmetros de um filtro digital, baseado na curva de resposta em frequências de um filtro passa-baixas. . .	17
2.2	Funcionamento da DWT, exemplificado para um sinal $s[n]$ de n amostras discretas e máxima frequência π , decomposto até o terceiro nível. Note o espectro de frequências e a quantidade de amostras presentes em cada sub-banda.	22
2.3	Relação entre os filtros de análise e síntese.	25
2.4	Formato das respostas aos impulsos dos filtros <i>wavelet</i> , para diversos suportes. Da esquerda para direita, de cima para baixo: Haar, Daubechies, Vaidyanathan, Beylkin, Coiflet, and Symmlet. .	28
2.5	Formatos das funções <i>scaling</i> dos filtros <i>wavelet</i> . Da esquerda para direita, de cima para baixo: Haar, Daubechies, Vaidyanathan, Beylkin, Coiflet, and Symmlet.	28
2.6	Formatos das funções <i>wavelet</i> dos filtros <i>wavelet</i> . Da esquerda para direita, de cima para baixo: Haar, Daubechies, Vaidyanathan, Beylkin, Coiflet, and Symmlet.	28
2.7	Exemplo de uma estrutura típica de uma RNA, de 3 camadas, com 4 neurônios na camada de entrada (XI), 2 na intermediária (XH), e 1 na saída (XO).	30
2.8	Funcionamento de um neurônio ativo de uma RNA.	30
2.9	Interpretação física simplificada do sistema bio-gerador de voz. . .	33
2.10	[esquerda]: visão básica do sistema de produção de voz humana; [direita]: detalhe do trato vocal humano e suas sub-partes.	34
2.11	Esquema básico para síntese de voz.	35
2.12	Ouvido humano: parte externa, média e interna [26]-p.168.	36
2.13	Membrana basilar e as frequências captadas em Hertz (Hz) [26]-p.173.	36
2.14	[acima]: espectro de magnitude (DFT) de um sinal <i>voiced speech</i> em escala logarítmica ; [abaixo]: <i>cepstrum</i> , sendo possível observar claramente as frequências formantes ($F_1, \dots, F_n$) à esquerda e o período de <i>pitch</i> ($\frac{1}{F_0}$) à direita.	38

3.1	Arquitetura básica do sistema proposto.	42
-----	---	----

Resumo

Dentre as inúmeras técnicas de conversão de voz utilizadas atualmente, aquelas baseadas em bancos de filtros *wavelet*, associadas com redes neurais artificiais, têm se destacado. Este trabalho se concentra em tais técnicas, realizando um estudo que relaciona qual a melhor *wavelet* para conversão de determinados padrões de voz, apresentando uma análise detalhada de quais são as características que levam a estes resultados. Os testes são realizados com vozes da base de dados TIMIT do *Linguistic Data Consortium* (LDC).

Abstract

Nowadays, among many usual techniques for voice morphing, the ones based on *wavelet* filter banks, associated with artificial neural networks, have been emphasized. This work concentrates in such techniques, presenting a study on the best *wavelet* for converting each particular voice pattern, with a detailed analysis of the characteristics that take to such outcomes. Tests are performed with speech data collected from TIMIT corpus provided by Linguistic Data Consortium (LDC).

Capítulo 1

Introdução e Motivação

Conversão de voz [1] [2], também conhecida como *voice morphing*, *voice transformation*, ou *voice conversion*, é o processo pelo qual uma sentença pronunciada por determinado locutor, o *source speaker*, se transforme de forma que pareça ser pronunciada por um outro locutor, o *target speaker*. A voz resultante de tal processo de transformação deve ser reconhecida como pertencente ao *target speaker* de forma inteligível e natural para possibilitar o uso comercial do sistema de conversão. Dentre as aplicações possíveis, destacam-se as seguintes:

- burlar um sistema de autenticação biométrica por comandos de voz, verificando a eficácia dos algoritmos autenticador e conversor;
- reproduzir a voz de uma pessoa famosa em aplicações multimídia e educacionais;
- editar um filme em que atua determinado ator, ou atriz, que já faleceu, sendo necessário que os *dublês* tenham suas vozes convertidas;
- reprodução de um texto escrito, para uso em leitores de *e-mails* por exemplo, através da síntese da voz de um locutor indisponível;
- edição de voz em geral, substituindo locuções indesejadas por outras desejadas;

entre outras aplicações.

A principal motivação para o estudo e a construção do algoritmo de *voice morphing* proposto é o uso em sistemas de autenticação biométrica por voz. O algoritmo desenvolvido suporta o projeto *SpeechAuth*, em andamento no laboratório *SpeechLab*¹ do Instituto de Física de São Carlos, da Universidade de São Paulo

¹<http://speechlab.ifsc.usp.br>

(IFSC/USP), e financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) ².

O estudo apresentado nesta dissertação enfoca um algoritmo para *voice morphing* que está baseado em bancos de filtros *wavelet* [3][4][5][6], apoiado por uma rede neural artificial [7] [8]. Esta classe de algoritmos de conversão de voz é a que tem apresentado o melhor desempenho em termos de qualidade da voz convertida, isto é, similaridade com a voz real do *target speaker*. Particularmente, fixado o algoritmo, diversas famílias de *wavelets* são testadas e os resultados são comparados, levando em consideração características como as respostas em frequência dos filtros, as respostas em fase, os números de momentos nulos e os formatos das funções *scaling* e *wavelet*, entre outros fatores. Os testes são realizados com vozes digitalizadas extraídas da base de dados TIMIT, do *Linguistic Data Consortium* (LDC) [9], que é uma das bases mais utilizadas para trabalhos com processamento digital de voz.

Este trabalho está organizado da seguinte forma: o capítulo 2 apresenta uma revisão da literatura envolvendo os conceitos utilizados para compreensão do trabalho, em especial o funcionamento dos sistemas biológicos e artificiais para produção de voz humana, reconhecimento natural e artificial de fala, as técnicas primitivas e recentes para conversão de voz, assim como a transformada *wavelet* e os princípios de redes neurais. Já o capítulo 3 apresenta com detalhes o algoritmo proposto de conversão de voz, sendo que a escolha da *wavelet*, baseada em inúmeros testes, está descrita no capítulo 4. Tendo em vista os resultados obtidos, e apoiado nos conceitos e características estudadas durante a revisão da literatura, o capítulo 5 apresenta, de um ponto de vista teórico-prático, as conclusões que explicam porque determinada *wavelet* é mais propícia para converter determinado padrão de voz. Finalmente, logo após a lista de referências bibliográficas, três apêndices apresentam, respectivamente, uma lista dos coeficientes dos filtros *wavelet* utilizados nas experiências, o código fonte da implementação em linguagem C++, assim como as publicações obtidas durante o curso de mestrado do autor.

²<http://www.fapesp.br> - processo nr. 05/0015-1

Capítulo 2

Revisão da Literatura

Neste capítulo, uma revisão da literatura é apresentada, relacionando fundamentalmente os seguintes tópicos: conceitos elementares de processamento de sinais, geração e interpretação de voz, transformada wavelet, redes neurais artificiais e os algoritmos para voice morphing. A revisão apresentada para cada um dos conceitos se restringe em um nível suficiente para o entendimento do trabalho proposto, de maneira que cada tópico apresentado suporta o entendimento dos sub-seqüentes. As referências apresentadas fornecem explicações muito mais detalhadas.

2.1 Conceitos Elementares de Processamento de Sinais

Antes de apresentar os tópicos relacionados diretamente com *voice morphing*, esta seção relembra alguns conceitos básicos de processamento de sinais que serão utilizados neste trabalho.

2.1.1 Processamento de sinais: analógico X digital

Durante décadas, todo o processamento de sinais era realizado unicamente de modo analógico, ou seja, baseado em tempo contínuo. Até hoje, ainda é viável realizar o processamento analógico de alguns sinais, entretanto, esse não é o caso de um sistema de *voice morphing*, como o deste trabalho. Como diz o próprio nome, o sistema de processamento analógico de sinais faz uso de circuitos eletrônicos analógicos. Em contrapartida, os sistemas de processamento digital de sinais recorrem basicamente ao uso de um processador capaz de realizar operações aritméticas [10].

A principal vantagem do processamento analógico sobre o digital é a resposta precisa, e em tempo real, aos estímulos recebidos, através dos circuitos eletrônicos analógicos, devido à ausência de quantização de amplitude e discretização no tempo. Já a abordagem digital, como requer computações aritméticas, não pode garantir operação em tempo real em velocidades infinitamente altas. Em contrapartida, a abordagem digital possui três vantagens principais:

- **Imunidade:** não existe a influência de agentes que distorcem os valores de componentes passivos nos circuitos eletrônicos, tais como resistores e capacitores que, com o tempo ou dependendo de fatores externos (umidade / temperatura / envelhecimento), podem ter seus valores alterados modificando, assim, o comportamento global dos sistemas de processamento. Tal fato era muito comum, por exemplo, nas centrais telefônicas antigas que utilizavam filtros para multiplexar a banda de comunicação que constantemente sofria interferências de outros canais adjacentes. Os únicos erros resultantes do processamento digital são devidos à precisão na representação numérica.
- **Flexibilidade:** obviamente, quando falamos em processadores e memórias, ou seja, computadores, temos em mente que todo o processamento pode ser controlado via *software* e que, portanto, qualquer modificação que se queira fazer no processamento não implica utilizar ferramentas para manipular valores de componentes nos circuitos eletrônicos.
- **Repetitividade:** uma operação de processamento digital de sinais pode ser repetida diversas vezes de maneira exata enquanto que nos sistemas analógicos, podem existir variações em razão da falta de imunidade comentada anteriormente.

Devido a todas as características mencionadas, a abordagem utilizada neste trabalho será o processamento *digital* de sinais.

2.1.2 Sinais e sistemas

Quer possamos nos dar conta do fato ou não, os sinais estão presentes em todos os momentos de nossas vidas. Os sons que ouvimos e interpretamos a todo instante, as imagens que nos chegam ao cérebro através de nossos olhos e nervos ópticos, um eletrocardiograma (ECG) ou eletroencefalograma (EEG) de um ser humano são alguns exemplos desses sinais. Na prática, enumerar tudo aquilo que constitui um sinal é tarefa impossível. De acordo com [12], um sinal pode ser definido como uma função de uma ou mais variáveis que vincula informações sobre um fenômeno físico. O presente trabalho envolve apenas os sinais digitais de

vozes humanas. Em todos os exemplos mencionados, existe sempre, de maneira implícita, um sistema associado. Por exemplo, o sistema associado à interpretação de uma imagem por um inseto é o conjunto formado pelo olho, neurônios associados, nervo óptico e cérebro. Da mesma forma, os sons das vozes humanas utilizadas no presente trabalho, fazem uso do sistema formado pelo pulmão e trato vocal e nasal, para sua correta produção, bem como do ouvido, cérebro e demais órgãos para sua interpretação pelo ser humano.

Um conceito básico relativo a um sinal discreto $x[n]$, é a sua energia, $E(x[n])$, que é dada por $E = \sum_{i=0}^{N-1} x_i^2$. Por sua vez, dizemos que o sinal sofre uma subamostragem (*downsampling*) por K , toda vez que uma determinada amostra é considerada e as $K - 1$ seguintes são descartadas, e assim por diante até o final do sinal ($K \in \mathbb{Z}$). Dizemos também que um sinal digital sofre um *upsampling* por K quando K zeros são inseridos entre cada amostra. Os símbolos para *downsampling* e *upsampling* são respectivamente $(\downarrow_K)$ e $(\uparrow_K)$. Finalmente, um sinal caracterizado como estacionário é aquele que mantém constante sua frequência ao longo de todo o intervalo de tempo considerado. No presente trabalho em que estamos lidando com sinais de vozes não estacionários, utilizaremos o cálculo de energia para examinar algumas propriedades destes sinais, e também serão utilizados *downsamplings* e *upsamplings* durante os processos de filtragem.

Um sistema, por sua vez, é definido como uma entidade que manipula um ou mais sinais, processando-os e produzindo, assim, outros sinais que representam, para nós, determinada informação. O trato vocal humano, por exemplo, é um sistema que recebe como entrada um sinal de excitação dos pulmões e produz um sinal de voz inteligível. O sistema computadorizado de *voice morphing*, proposto neste trabalho, recebe como entrada um sinal de voz digital de determinado locutor, e modifica-o de forma que ele pareça ser pronunciado por outro locutor. Quando um sistema obedece às condições: *i*) o deslocamento da entrada por uma determinada constante de tempo implica o mesmo deslocamento na saída; *ii*) a multiplicação da entrada do sistema por uma constante implica a saída multiplicada pela mesma constante; *iii*) a soma de dois ou mais sinais na entrada de um sistema implica a soma das saídas individuais; dizemos que o sistema é Linear e Invariante no Tempo (LTI).

2.1.3 Filtros digitais

Um filtro digital [13] é nada mais do que um sistema que realiza uma combinação linear de um sinal de entrada com certos coeficientes, para obter um sinal de saída com determinadas características de frequência selecionadas. Os parâmetros mais relevantes de um filtro digital, ilustrados com a ajuda da figura 2.1, são:

Figura 2.1: Exemplo dos principais parâmetros de um filtro digital, baseado na curva de resposta em frequências de um filtro passa-baixas.

- frequência de corte: é a frequência para a qual o filtro já passa a ter uma atenuação maior ou igual a -3dB (aproximadamente 70,7%), que é o ponto onde termina a banda de passagem e inicia a banda de transição.
- frequência de rejeição: definida neste trabalho como a frequência para a qual o filtro já passa a ter uma atenuação maior ou igual a 95 % da atenuação máxima, que é o ponto onde termina a banda de transição e se inicia a banda de rejeição.
- banda de passagem: é a faixa de frequências anterior a frequência de corte.
- banda de transição: é a faixa de frequências que se inicia no final da banda de passagem e termina no início da banda de rejeição.
- banda de rejeição: faixa de frequências posterior à frequência de rejeição.
- tipo:
 - resposta ao impulso finita (*finite impulse response* - FIR): quando a quantidade de coeficientes do filtro digital, no domínio do tempo, é finita. Os filtros digitais utilizados neste trabalho, que são do tipo *wavelet*, conforme descrito adiante, são todos FIR.
 - resposta ao impulso infinita (*infinite impulse response* - IIR): quando a quantidade de coeficientes do filtro digital, no domínio do tempo, é infinita. Nesses casos, que estão fora do escopo do presente trabalho,

a filtragem é realizada através de uma equação de diferenças recursiva [13].

- função: passa-baixas, passa-altas, passa-faixas ou rejeita-faixas, conforme as características específicas de seletividade de frequências. No presente trabalho, serão utilizados pares de filtros passa-baixas e passa-altas cujas respostas em frequências são espelhadas em relação ao eixo vertical (amplitude). Tais filtros são denominados de *quadrature mirror filters* (QMF) [13].
- ordem: número de pólos da função de transferência do filtro, conforme descrito adiante. Um filtro digital com $N + 1$ coeficientes possui ordem N . À medida que a ordem do filtro aumenta, sua resposta em frequências fica mais próxima da ideal, ou seja, a banda de transição é mais estreita. O presente trabalho faz uso de filtros de ordens variadas.
- fase: linear (atraso constante da saída para toda a faixa de frequências) ou não linear. Os filtros utilizados neste trabalho possuem fase aproximadamente linear, ou de fato não linear. A interferência deste fator nos resultados é analisado adiante.

2.1.4 Teorema da convolução

Este teorema enuncia que a multiplicação de dois sinais discretos no domínio da frequência, $H[z]$ e $X[z]$, corresponde à convolução dos mesmos no domínio do tempo, $h[n]$ e $x[n]$. A convolução, $y[n]$, dos dois sinais discretos $x[n]$ e $h[n]$, representada pelo símbolo $*$, é dada por:

$$y[n] = x[n] * h[n] = \sum_{k=0}^{M-1} h_k x_{n-k}, \quad (2.1)$$

onde M é o número de amostras de $h[n]$.

2.1.5 Resposta ao impulso de um filtro digital

A resposta ao impulso é a resposta do filtro para uma entrada impulsiva $\delta[n] = \{1, 0, 0, 0, \dots, 0\}$, que corresponde aos coeficientes do filtro digital no domínio do tempo, ou seja, a transformada de Fourier inversa da função de transferência do filtro.

2.1.6 Teorema da amostragem

Também conhecido como teorema de Nyquist [13], enuncia que um sinal analógico precisa ser amostrado pelo menos o dobro de vezes da máxima frequência presente nele, para que as amostras discretas possam representá-lo sem *aliasing*.

2.1.7 Aliasing

Aliasing corresponde às frequências 'fantasmas' que surgem em um sinal digitalizado, quando o correspondente analógico é amostrado a uma taxa insuficiente.

2.1.8 Transformada Discreta de Fourier

A DFT - *Discrete Fourier Transform* [14], como também é conhecida, é a ferramenta matemática utilizada para converter um sinal do domínio do tempo para o da frequência. A transformada de Fourier inversa (IDFT - *Inverse Discrete Fourier Transform*) realiza a operação contrária. A DFT e a IDFT estão expressas respectivamente nas equações 2.2 e 2.3, onde $x[n]$ é o sinal no domínio do tempo, $X[\omega]$ é o correspondente no domínio da frequência, e N é o comprimento dos sinais.

$$X[\omega] = \sum_{n=0}^{N-1} x_n e^{-j2\pi n\omega/N} \quad . \quad (2.2)$$

$$x[n] = \sum_{\omega=0}^{N-1} X_\omega e^{j2\pi n\omega/N} \quad . \quad (2.3)$$

Existem algoritmos mais eficientes para o cômputo da DFT e IDFT, conhecidos como algoritmos de transformada rápida de Fourier (FFT - *Fast Fourier Transform*) [15], que reduzem a ordem de complexidade computacional da DFT, de quadrática para logarítmica. O presente trabalho utiliza a DFT / FFT para obtenção das curvas de resposta em frequência dos filtros.

2.1.9 Transformada Z

A Transformada Z [13] (TZ) é uma generalização da DFT, que converte um sinal do domínio do tempo para o domínio z , z representando a frequência, onde o argumento $e^{-j2\pi n\omega/N}$ é trocado pela variável complexa z :

$$X[z] = \sum_{t=0}^{N-1} x_t z^{-t} \quad . \quad (2.4)$$

A TZ é utilizada nesta dissertação para expressar a função de transferência dos filtros.

2.1.10 Função de transferência

Esta função consiste na TZ da resposta ao impulso de um sistema (filtro digital, no caso). Particularmente, se $h[n]$ é a resposta ao impulso do filtro, $x[n]$ é a entrada, e $y[n]$ é a saída, então, $y[n] = x[n] * h[n]$. No domínio z isto equivale a $Y[z] = X[z]H[z]$, ou seja, $H[z] = \frac{Y[z]}{X[z]}$ é a função de transferência. As raízes de $Y[z]$ são chamadas zeros da função de transferência (valores onde ela se torna zero) e as raízes de $X[z]$ são chamadas pólos da função de transferência (valores onde ela não existe ou tende para o infinito). Quando os pólos da função têm módulo no máximo 1, o sistema é estável e causal, e quando os zeros da função têm módulo no máximo 1, o sistema inverso é estável e causal. No caso de filtros tipo FIR, que serão os utilizados neste trabalho, a função de transferência será sempre uma função polinomial em expoentes negativos de z que possuirá somente zeros, ou seja o denominador de $H[z]$ será sempre 1.

2.1.11 A Transformada de Fourier de Tempo Reduzido (STFT)

Uma modificação na DFT de um sinal corresponde a *Shortest Time Fourier Transform* (STFT) [16], que supõe que um dado sinal não estacionário, quando repartido em partes pequenas, pode ter cada uma dessas partes consideradas isoladamente como estacionárias. A equação 2.5 ilustra o fato, onde $w[\]$ é uma janela temporal e os demais parâmetros são idênticos a DFT.

$$STFT[\omega] = w[\] \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi n\omega}{N}} . \quad (2.5)$$

Existe, entretanto, um problema com o uso da STFT: a largura da função que "janela" o sinal. Janelas estreitas resultam em boa resolução no tempo, mas uma resolução mais pobre em frequência, enquanto que janelas largas resultam numa resolução melhor em frequência e pior no tempo, além de violarem a suposição de estacionariedade do trecho do sinal envolvido. Uma possibilidade para contornar este fato consiste no uso da transformada *wavelet*, que traz diversos níveis de resolução de tempo em diversas faixas de frequência com diferentes resoluções. A STFT não será utilizada no presente trabalho e foi mencionada apenas para fazer a ligação entre a DFT e a DWT.

2.2 A Transformada *wavelet* Discreta (DWT)

A transformada *wavelet* discreta [3] [4][5] [6] consiste numa alternativa mais eficiente do que a STFT para realizar a análise tempo-frequência de um sinal, bem

como a filtragem e separação em sub-bandas de frequências. A DWT, que é objeto fundamental do presente trabalho, age na verdade com um par de filtros, sendo um deles passa-baixas ($h[\]$) e o outro passa-altas QMF ($g[\]$), em geral, com frequência de corte (-3dB) em $\frac{\pi}{2}$, sendo π a máxima frequência angular. Dado um sinal discreto, ele é submetido a ambos os filtros via convolução. Cada vez que esse processo é aplicado, diz-se que se tem um nível de decomposição e obtém-se dois novos sinais, sendo que um deles contém as frequências abaixo da metade da máxima frequência original do sinal e, o outro, contém as frequências acima desse limiar. Em particular, os termos chamados coeficientes de detalhamento designam o sinal obtido quando da passagem do sinal original pelo filtro passa-altas e os termos chamados coeficientes de aproximação designam o sinal obtido quando da passagem do sinal original pelo filtro passa-baixas. Após aplicar um nível de decomposição no sinal, apenas o novo sinal obtido pela aplicação do filtro passa-baixas é usado para continuar o processo. Existem casos em que ambos os sinais são submetidos novamente ao processo de filtragem.

Para tal particularidade dá-se o nome de transformada *wavelet-packet* (DWTP) [17], que será utilizada neste trabalho. Um detalhe fundamental a ser notado é que cada vez que um nível da transformação é realizado, os dois novos sinais obtidos são sub-amostrados por 2, pois eles contém apenas metade da faixa de frequências do sinal original, de acordo com o Teorema da Amostragem. Um sinal de n amostras tem a sua transformada *wavelet* com a mesma quantidade de amostras, sendo composta por uma sequência de coeficientes, iniciando-se com os coeficientes provenientes da aplicação do filtro passa-baixas no último nível, seguidos pelos coeficientes resultantes da aplicação dos filtros passa-altas nos níveis intermediários e terminando com os coeficientes resultantes da aplicação do filtro passa-altas do primeiro nível de decomposição. Todo esse processo encontra-se explicado na figura 2.2. Para realizar a decomposição até o último nível possível, é necessário que o sinal discreto tenha comprimento equivalente a uma potência de 2, sendo possível realizar $\frac{\log(n)}{\log(2)}$ decomposições para um sinal de comprimento n . Um fator muito importante para que um filtro digital seja considerado um filtro *wavelet* é que a resposta em frequência do filtro passa-baixas seja 0 em $\omega = \pi$, e, obviamente, seja 0 em $\omega = 0$, no caso do passa-altas QMF.

O processo conjunto de filtragem e sub-amostragem por 2, realizado nos sinais da transformada *wavelet* em cada nível, pode ser representado por uma convolução modificada da seguinte forma:

$$y[n] = x[n] * t[n] = \sum_{k=0}^{n-1} t_k x_{2n-k} \quad , \quad (2.6)$$

Figura 2.2: Funcionamento da DWT, exemplificado para um sinal $s[n]$ de n amostras discretas e máxima frequência π , decomposto até o terceiro nível. Note o espectro de frequências e a quantidade de amostras presentes em cada sub-banda.

ou, mais especificamente:

$$y_{passa-baixas}[n] = x[] * h[] = \sum_{k=0}^{n-1} h_k x_{2n-k} \quad , \quad (2.7)$$

$$y_{passa-altas}[n] = x[] * g[] = \sum_{k=0}^{n-1} g_k x_{2n-k} \quad . \quad (2.8)$$

A DWT está diretamente relacionada com a análise de multi-resolução (MRA), proposta por Mallat, Meyer, Stromberg e outros [18][19], que consiste em decompor um vetor (sinal sob análise) $\vec{f}$ em uma soma de outros vetores pertencentes a uma sequência de sub-espacos vetoriais [20]. Em outras palavras, isso significa representar um sinal em vários níveis de resolução. Então, de acordo com a MRA, para um vetor $\vec{f}$ de n pontos tem-se:

$$\vec{f} = \vec{A} + \vec{D} \quad (2.9)$$

onde

$$\vec{A} = \sum_{k=0}^{\frac{n}{2}-1} \langle \vec{f}, \vec{v}_k \rangle \vec{v}_k \quad \vec{D} = \sum_{k=0}^{\frac{n}{2}-1} \langle \vec{f}, \vec{w}_k \rangle \vec{w}_k.$$

ou seja:

- $\vec{A}$ é a projeção de $\vec{f}$ num sub-espaço V , com uma base de $\frac{n}{2}$ vetores;
- $\vec{D}$ é a projeção de $\vec{f}$ num sub-espaço W , com uma base de $\frac{n}{2}$ vetores;
- $V \perp W \leftrightarrow \vec{A} \perp \vec{D}$;
- $\vec{v}_i \perp \vec{w}_i \leftrightarrow \langle \vec{v}_i, \vec{w}_i \rangle = 0$.

O processo acima consiste na decomposição em nível 1. Numa transformada *wavelet* de nível 2, o vetor A é novamente decomposto na soma de dois outros vetores ortogonais. Esse processo pode ser repetido, $\frac{\log(n)}{\log(2)}$ vezes, conforme já foi mencionado. Dessa forma, generalizando, para uma decomposição de nível j , temos:

$$\vec{f} = \vec{A}_j + \sum_{i=1}^j \vec{D}_i. \quad (2.10)$$

sendo que:

- $\vec{A}_j$ é a projeção de $\vec{f}$ num sub-espaço V_j , com uma base contendo $\frac{n}{2^j}$ vetores;

- $\vec{D}_i$ é a projeção de $\vec{f}$ num sub-espço V_j , com uma base contendo $\frac{n}{2^j}$ vetores;
- $V_j \perp W_j \leftrightarrow \vec{A}_j \perp \vec{D}_j$;
- $\vec{v}_{i,j} \perp \vec{w}_{i,j} \leftrightarrow \langle \vec{v}_{i,j}, \vec{w}_{i,j} \rangle = 0$.

O processo acima equivale a [3]

$$f[n] = \sum_{k=0}^{\frac{n}{2^j}-1} H_{j,k}[n] \phi_{j,k}[n] + \sum_{t=1}^j \sum_{k=0}^{\frac{n}{2^j}-1} G_{t,k}[n] \psi_{t,k}[n] \quad (2.11)$$

onde

- $\phi[n]$ e $\psi[n]$ formam uma base de Riezs [3] para escrever $\vec{f}$;
- $\phi[n] = \sum_k h_n \phi[2n - k]$, definida recursivamente por dilatações e translações de si mesma é chamada função *scaling* [3];
- $\psi[n] = \sum_k g_n \phi[2n - k]$, também definida recursivamente, é chamada função *wavelet* e é ortogonal a função *scaling*;
- $H_{j,k}[n] = \langle f, \phi_{j,k}[n] \rangle$;
- $G_{t,k}[n] = \langle f, \psi_{t,k}[n] \rangle$;
- $\{0\} \leftarrow \dots \subset V_{-1} \subset V_0 \subset V_1 \subset \dots \rightarrow L^2$;
- se $f[n] \in V_j \rightarrow f[2n] \in V_{j+1}$;
- $V_{j+1} = V_j \oplus W_j$;
- os coeficientes h_k correspondem ao filtro passa-baixas;
- os coeficientes g_k correspondem ao filtro passa-altas;
- $h[\]$ e $g[\]$ são chamados filtros de análise;
- um filtro com k coeficientes é dito filtro de suporte k .

Cada par de filtros de análise, $h[\]$ e $g[\]$, possui uma única função *scaling* (ϕ) e uma única função *wavelet* (ψ) associadas. A forma de obtenção dessas funções a partir dos filtros, e vice-versa, está documentada com detalhes em [5][6], não sendo apresentada aqui por estar fora do escopo do trabalho.

É muito importante também o fato de que $h[\]$ e $g[\]$ possuem outros filtros

Figura 2.3: Relação entre os filtros de análise e síntese.

associados, chamados filtros de síntese, representados por $\bar{h}[n]$ e $\bar{g}[n]$, que são utilizados para inverter a transformada, recuperando o sinal original a partir do transformado. Tais filtros obedecem às relações das equações 2.12, 2.13 e 2.14, para $k = 0, \dots, n-1$, que ficam mais claras através do exemplo na figura 2.3, para filtros de suporte 4.

$$g_k = (-1)^k h_{N-k-1} \quad , \quad (2.12)$$

$$\bar{h}_k = h_{N-k-1} \quad , \quad (2.13)$$

$$\bar{g}_k = (-1)^{k+1} h_k \quad . \quad (2.14)$$

Quando $h[n]$, $g[n]$, $\bar{h}[n]$, e $\bar{g}[n]$ mantêm as relações acima, eles constituem um banco de filtros de reconstrução perfeita (*perfect reconstruction filter bank* - PRFB) [18][19], ou seja, as condições de *anti-aliasing* e *no-distortion*, no domínio Z , representadas nas equações 2.15 e 2.16, respectivamente, são satisfeitas. No presente trabalho, apenas PRFBs são utilizados, pois a reconstrução perfeita dos sinais é desejada.

$$\bar{H}[z] = G[-z] \quad , \quad \bar{G}[z] = -H[-z] \quad . \quad (2.15)$$

$$\bar{H}[z]H[z] + \bar{G}[z]G[z] = 2z^{-N+1} \quad . \quad (2.16)$$

2.2.1 Cálculo da DWT

Para o cálculo da DWT de um sinal, aplica-se o algoritmo de Mallat, que está minuciosamente descrito em [5][6]. É importante observar que apenas os filtros $h[n]$ e $g[n]$ são utilizados, não sendo necessária a utilização de ϕ e ψ . O procedimento de cálculo envolve apenas a multiplicação de duas matrizes para cada nível de transformação. Se $A[n][n]$ é a matriz de coeficientes dos filtros e $B[n]$ é o sinal original, então $C[n] = A[n][n] B[n]$ corresponde ao sinal transformado,

sendo que a disposição dos coeficientes nas matrizes é a seguinte:

$$A[\][\] = \begin{pmatrix} h_0 & h_1 & h_2 & \dots & \dots & \dots & h_n & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 \\ g_0 & g_1 & g_2 & \dots & \dots & \dots & g_n & 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & \dots & \dots & h_{n-1} & h_n & 0 & 0 & \dots & \dots & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & \dots & \dots & g_{n-1} & g_n & 0 & 0 & \dots & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ h_{n-1} & h_n & 0 & 0 & \dots & \dots & \dots & 0 & 0 & h_0 & h_1 & \dots & \dots & h_{n-3} & h_{n-2} \\ g_{n-1} & g_n & 0 & 0 & \dots & \dots & \dots & 0 & 0 & g_0 & g_1 & \dots & \dots & g_{n-3} & g_{n-2} \end{pmatrix},$$

$$B[\] = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ \dots \\ \dots \\ \dots \\ b_{n-1} \\ b_n \end{pmatrix}, \quad C[\] = \begin{pmatrix} c_0 \\ c_{\frac{n}{2}} \\ c_1 \\ c_{\frac{n}{2}+1} \\ \dots \\ \dots \\ \dots \\ c_n \\ c_{\frac{n}{2}-1} \end{pmatrix}.$$

Pode-se notar na matriz $A[\][\]$ que dois procedimentos estão embutidos no algoritmo de Mallat: *downsampling* e *wrap-around* [5][6]. O primeiro já foi mencionado anteriormente, e o segundo consiste em fazer com que os últimos coeficientes dos filtros ocupem as posições iniciais de cada linha. Isso faz com que a DWT tenha sempre o mesmo número de elementos do sinal original.

2.2.2 Cálculo da DWT inversa (IDWT)

Da mesma forma como ocorre no cálculo da DWT, para calcular a IDWT através do algoritmo de Mallat, apenas $\bar{h}[\]$ e $\bar{g}[\]$ são necessários. O cálculo se procede de forma a obter novamente o vetor do sinal original $B[\]$ a partir da multiplicação de $A^{-1}[\][\]$ por $C[\]$, onde $A^{-1}[\][\]$, que é a inversa de $A[\][\]$, corresponde à matriz dos coeficientes dos filtros de síntese, isto é, $\bar{h}[\]$ e $\bar{g}[\]$.

2.2.3 Momentos Nulos

A quantidade de momentos nulos [5][6] é uma propriedade interessante da DWT. Ela implica que, para um sinal que pode ser (aproximadamente) descrito por um polinômio de grau menor que M e uma *wavelet* que possui M momentos nulos, os coeficientes de detalhamento serão (aproximadamente) zero. Embora esse fato seja primordialmente importante nos esquemas de compressão de dados, ele será analisado no presente trabalho. O M -ésimo momento pode ser calculado como segue:

$$\sum_{k=0}^{p-1} t_k^m \psi(t_k) \quad , \quad (2.17)$$

Família	Suporte(n)	Fase	Observação	Momentos
Haar	2	linear	é a mais simples das <i>wavelets</i> , criada por Alfred Haar [5][6]	1
Daubechies	par, maior que 4	não linear	resposta ao impulso <i>maximally flat</i> , criada por Ingrid Daubechies [5][6]	$\frac{n}{2}$
Symmlets	par, múltiplo de 8	não linear	resposta ao impulso mais simétrica[5][6]	$\frac{n}{2} - 2$
Coiflets	par, múltiplo de 6	quase linear	resposta ao impulso quase simétrica, criada por Ronald Coifman [5][6]	$\frac{n}{2} - 1$
Vaidyanathan	24	não linear	otimizada para voz, criada por P. P. Vaidyanathan [5][6]	—
Beylkin	18	não linear	otimizada para áudio em geral [5][6]	$\frac{n}{2} - 2$

Tabela 2.1: Características das famílias de *wavelets* utilizadas no presente trabalho, incluindo a quantidade de momentos da função *wavelet*.

sendo p a quantidade de pontos da função *wavelet*, m o momento desejado e t cada ponto onde a função pode possuir valor diferente de 0 ($t = 0.5s$, onde s é um escalar inteiro maior ou igual a 0).

2.2.4 Famílias de Transformadas *Wavelet*

As diversas famílias de filtros existentes [5][6] diferem no suporte, assim como nas características de resposta em frequência e fase, o que faz com que as funções ϕ e ψ também sofram reflexo de tais diferenças. Serão utilizadas neste trabalho as *wavelets* de Haar, Daubechies, Symmlets, Coiflets, Vaidyanathan e Beylkin, com diversos suportes, todas constituindo filtros do tipo FIR, sendo que as respostas em frequência se aproximam das ideais à medida que o suporte cresce. As características de cada uma dessas *wavelets* estão descritas resumidamente na tabela 2.1 e nas figuras 2.4, 2.5 e 2.6.

Figura 2.4: Formato das respostas aos impulsos dos filtros *wavelet*, para diversos suportes. Da esquerda para direita, de cima para baixo: Haar, Daubechies, Vaidyanathan, Beylkin, Coiflet, and Symmlet.

Figura 2.5: Formatos das funções *scaling* dos filtros *wavelet*. Da esquerda para direita, de cima para baixo: Haar, Daubechies, Vaidyanathan, Beylkin, Coiflet, and Symmlet.

Figura 2.6: Formatos das funções *wavelet* dos filtros *wavelet*. Da esquerda para direita, de cima para baixo: Haar, Daubechies, Vaidyanathan, Beylkin, Coiflet, and Symmlet.

2.3 Princípios Elementares de Redes Neurais Artificiais (RNAs)

As RNAs têm se mostrado altamente capazes de solucionar problemas de classificar padrões [7]. No caso do presente trabalho, uma estrutura relativamente simples de RNA será utilizada para transformar um padrão de entrada, relativo ao *source speaker*, em um padrão de saída, relativo ao *target speaker*. Para isso, um conjunto de coeficientes, chamados pesos, realiza uma operação similar a uma convolução discreta do sinal de entrada com os pesos. Estes são determinados durante a fase de treinamento da RNA, com base nas estatísticas do padrão de entrada.

A constituição de uma RNA é especificada pela sua topologia, pela função de ativação dos neurônios e pelos algoritmos de treinamento, os quais especificam um conjunto inicial de pesos e indicam como eles devem ser modificados para melhorar o seu desempenho na tarefa para a qual foi especificada. A topologia que se mostra mais adequada para a do tipo de problema que trata a presente dissertação é a *feed-forward*, em particular a chamada *perceptron* multicamadas (*multi-layer perceptron*).

A figura 2.7 mostra a topologia básica de uma *multi-layer perceptron*, onde se destacam três camadas de neurônios: *input layer*, *hidden layer*, e *output layer*, que imitam o comportamento dos neurônios humanos. A primeira camada, *input layer*, é dita passiva, pois apenas transmite os valores de entrada aos neurônios da camada intermediária, *hidden layer*. Os neurônios desta camada, assim como os da camada de saída, *output layer*, são ditos ativos.

O funcionamento de um neurônio ativo, ilustrado na figura 2.8, dá-se da seguinte forma: a saída do neurônio corresponde à saída de uma função sigmóide, que por sua vez recebe como entrada uma combinação linear dos sinais de entrada com os pesos. A função sigmóide, exibida na equação 2.18, tem a interessante propriedade de que para entradas ilimitadas ($-\infty \leq x \leq \infty$), a saída é limitada no intervalo $[0 - 1]$, ou seja, $0 \leq s(x) \leq 1$.

$$s(x) = \frac{1}{1 + e^{-x}} \quad . \quad (2.18)$$

A quantidade de neurônios de cada camada depende muito da aplicação específica. O exemplo da figura 2.7 possui 4 neurônios na camada de entrada, 2 na intermediária, e 1 na final. No caso do presente trabalho, as quantidades variam de acordo com a sub-banda de frequências sendo processada, conforme será visto no próximo capítulo. Em termos da quantidade de camadas, uma revisão mais

Figura 2.7: Exemplo de uma estrutura típica de uma RNA, de 3 camadas, com 4 neurônios na camada de entrada (XI), 2 na intermediária (XH), e 1 na saída (XO).

Figura 2.8: Funcionamento de um neurônio ativo de uma RNA.

ampla na literatura [8] mostra que não são necessárias mais que três camadas de neurônios para um perceptron, pois uma RNA desse tipo já é capaz de gerar regiões de decisão arbitrariamente complexas [7].

Por fim, o treinamento da rede, pode ser de dois tipos: supervisionado ou não supervisionado. O primeiro exige a disponibilidade de um conjunto de treinamento formado por pares de vetores de entrada e de saída, chamados pares de treinamento. Já no treinamento não supervisionado, o conjunto de treinamento consiste somente de vetores de entrada. As RNAs presentes neste trabalho terão treinamento supervisionado. Uma rede pode ser treinada com três objetivos:

- auto-associação: após o treinamento com um conjunto de vetores, quando submetida a um vetor similar a um dos exemplos, mas deturpado, reconstituir o vetor original;
- hetero-associação: após o treinamento com um conjunto de pares de vetores, quando submetida a um vetor similar ao primeiro elemento de um par, mas deturpado, reconstituir o segundo elemento do par;
- detecção de regularidades: descobrir as regularidades inerentes aos vetores de treinamento e criar padrões para classificá-los de acordo com tais regularidades.

A maioria dos algoritmos de treinamento de rede neurais é inspirada, direta ou indiretamente, na lei de Hebb [21], descoberta pelo biólogo que lhe empresta o nome. O algoritmo mais conhecido para treinamento de redes neuronais é a retropropagação (*backpropagation*). A retropropagação é um algoritmo de treinamento supervisionado. Seu funcionamento pode ser descrito da seguinte forma:

- i) apresenta-se um exemplo à rede e obtém-se a saída correspondente;
- ii) calcula-se o vetor de erro que consiste na diferença entre a saída obtida e a esperada;
- iii) calcula-se o gradiente do vetor de erro e atualizam-se os pesos da camada de saída;
- iv) propagam-se para trás (*backpropagation*) os valores desejados de modo a atualizar os pesos das demais camadas.

Como o algoritmo de retropropagação requer o cálculo do gradiente do vetor de erro, é interessante que a função de ativação seja derivável em todos os pontos. Isso explica o uso da função sigmóide. Outros algoritmos de treinamento de redes neurais não utilizados neste trabalho são: contrapropagação, aprendizado competitivo e algoritmos genéticos.

2.4 Os Sistemas Geradores de Voz

Para compreender qualquer algoritmo de *voice morphing*, deve-se iniciar pelo estudo do sistema de produção de voz pelo corpo humano [22] e também da produção artificial de voz por computadores e circuitos eletrônicos, conhecida como síntese de voz [23].

2.4.1 O sistema bio-físico de produção de voz

A produção de voz pelo corpo humano, embora seja um mecanismo repleto de detalhes, consiste basicamente da propulsão de ar pelos pulmões, seguida de um processo de filtragem, realizado pelo trato vocal e elementos associados, como ilustram as figuras 2.9 e 2.10. O primeiro detalhe a ser observado na figura 2.9 é que as cordas vocais controlam o fluxo de ar fornecido pelos pulmões, fazendo com que esse sinal de excitação seja periódico, vibrando em determinada frequência, ou aperiódico, similar a um sinal ruidoso. Se o sinal for periódico, esse período é chamado de período de *pitch* e a voz produzida será classificada como *voiced speech*, caso contrário a voz será classificada como *unvoiced speech*. No primeiro caso encontram-se basicamente as vogais, enquanto no segundo caso estão os demais sons. Dependendo de como agem as estruturas seguintes às cordas vocais, em particular o véu palatino, que controla a passagem do fluxo de ar pelo trato vocal ou nasal, e os próprios tratos vocal e nasal, pode-se ainda refinar essa classificação dos sinais de voz da seguinte forma [24]:

- *fricatives*: é um *unvoiced speech* que surge quando há fricção do ar em movimento contra a constrição, causando, em geral, uma turbulência de ar entre a língua e os dentes superiores. Exemplo: th na palavra *thin* da língua Inglesa.
- *plosives*: é um *unvoiced speech* impulsivo, como o t na palavra *top*.
- *whispers*: é um *unvoiced speech* onde uma barreira é criada nas cordas vocais de forma que elas permaneçam parcialmente fechadas e sem oscilação, como ocorre quando se pronuncia o h na palavra *he*.
- *voiced fricatives*: são fonemas *voiced*, ou seja de excitação periódica, porém misturado com o ruído criado na constrição do trato vocal, atrás dos dentes e contra o palato. Exemplo: z na palavra *zebra*.
- *unvoiced fricatives*: idem anterior, porém as cordas vocais não vibram simultaneamente com a fricção.

Figura 2.9: Interpretação física simplificada do sistema bio-gerador de voz.

- *voiced plosives*: são fonemas *voiced*, ou seja de excitação periódica, porém misturada com ruído impulsivo criado no trato vocal.
- *unvoiced plosives*: idem anterior, porém as cordas vocais não vibram simultaneamente com o impulso. Exemplo: b na palavra *boat*.

Qualquer palavra ou frase pronunciada por um locutor pode ser dividida em fonemas, cada qual podendo ser classificado como explicado anteriormente.

2.4.2 Síntese de voz

Existe uma grande quantidade de algoritmos para produção artificial de voz humana [22], sendo que os mais eficazes são aqueles baseados em Modelos Ocultos de Markov (*Hidden Markov Models* - HMM) [25]. Independente do algoritmo específico, pode-se dizer que, numa visão pouco detalhista, o esquema básico utilizado para síntese de fala consiste num gerador de sinal, periódico ou aperiódico, cuja saída é aplicada a um filtro que imita as características do trato vocal de determinado indivíduo hipotético. As frequências de ressonância que correspondem aos pólos da sua função de transferência, são conhecidas como formantes, abreviadas por $F_1, F_2, \dots, F_n$, e a frequência de vibração do gerador de sinal, quando este for periódico, é conhecida como frequência fundamental, que corresponde ao inverso do período de *pitch*, abreviada por F_0 . A figura 2.11 ilustra este esquema.

Figura 2.10: [esquerda]: visão básica do sistema de produção de voz humana; [direita]: detalhe do trato vocal humano e suas sub-partes.

Alterando os parâmetros $F_0, F_1, F_2, \dots, F_n$, associado à correta evolução temporal do sinal sintetizado, pode-se produzir uma ampla gama de sinais de fala, com características diversas, isto é, locutor masculino ou feminino, jovem ou idoso, entre outras.

Durante a síntese, ou mesmo a produção natural, de voz, o sinal de excitação pode ser modelado como um sinal discreto $s[n]$, no domínio do tempo. Da mesma forma, o trato vocal pode ser encarado como um filtro que tem função de transferência $F[f]$. Se a resposta ao impulso de $F[f]$ for representada por $f[n]$, então a saída do sistema, $y[n]$, que corresponde à voz produzida, é a convolução de $s[n]$ com $f[n]$, ou seja $y[n] = s[n] * f[n]$. Esse modelo simplificado de produção de voz é muito útil para extração de características do sinal de voz.

2.5 Interpretação de Voz

Tendo em vista que alguns métodos mais rudimentares para *voice morphing* fazem uso de algoritmos de interpretação de voz, como será visto adiante, esta seção revisa, de maneira superficial, o princípio básico de reconhecimento e interpretação de fala.

Figura 2.11: Esquema básico para síntese de voz.

2.5.1 O Sistema bio-físico de interpretação de fala

O ouvido humano percebe os sons através de um mecanismo bastante elaborado [24]. A figura 2.12 exibe um diagrama simplificado desse mecanismo, onde é possível destacar três partes: ouvido externo, ouvido médio e ouvido interno. O ouvido externo, que controla a captação e direcionabilidade, coleta os sons e os conduz até o ouvido médio através do canal auditivo. No ouvido médio, a pressão do ar é convertida em movimentação de um fluido que é levada a uma estrutura de fundamental importância: a *cochlea*, que faz parte do ouvido interno. Esta estrutura, que está associada com a membrana basilar, separa os sons de acordo com as frequências e converte a movimentação fluídica em impulsos elétricos no nervo auditivo. Finalmente, tais impulsos são interpretados pelo cérebro. A figura 2.13 dá uma idéia das partes da membrana que são sensíveis à determinadas frequências. Uma particularidade interessante do ouvido humano, é que as frequências percebidas pela membrana basilar podem ser divididas em faixas, chamadas de bandas críticas, que são em número de 25. Quando dois ou mais sons, de frequências diferentes mas pertencentes à mesma banda crítica, estão sendo captados simultaneamente, o de maior amplitude mascara o(s) outro(s), portanto, o de menor amplitude não é percebido [26]. Esse fato é aproveitado por alguns

Figura 2.12: Ouvido humano: parte externa, média e interna [26]-p.168.

Figura 2.13: Membrana basilar e as frequências captadas em Hertz (Hz) [26]-p.173.

sistemas de *voice morphing*, como é o caso do algoritmo proposto neste trabalho. A tabela 2.2 mostra as bandas críticas, na escala de frequências em Hertz (Hz) e na escala Bark (z), que é uma escala adotada para linearizar o intervalo entre as bandas. A conversão entre Hertz e Bark é dada pela equação 2.19, onde f representa a frequência em Hertz.

$$z = \frac{26.81f}{1960 + f} - 0.53, \quad (2.19)$$

sendo que se $z < 2$ então aplica-se a correção $z \leftarrow z + 0.15(2 - z)$ e se $z > 20$ então faz-se $z \leftarrow z + 0.22(z - 20.1)$.

2.5.2 Reconhecimento de fala por intermédio computacional

De acordo com [24], o reconhecimento computacional de fala, que geralmente se dá com base em HMMs, é uma tarefa bastante complexa, estando longe do objetivo deste trabalho a descrição de um sistema para tal fim. De qualquer forma,

<i>Bark (z)</i>	<i>Hertz (Hz)</i>	<i>Bark (z)</i>	<i>Hertz (Hz)</i>
0	0 - 100	12	1720 - 2000
1	100 - 200	13	2000 - 2320
2	200 - 300	14	2320 - 2700
3	300 - 400	15	2700 - 3150
4	400 - 510	16	3150 - 3700
5	510 - 630	17	3700 - 4400
6	630 - 770	18	4400 - 5300
7	770 - 920	19	5300 - 6400
8	920 - 1080	20	6400 - 7700
9	1080 - 1270	21	7700 - 9500
10	1270 - 1480	22	9500 - 12000
11	1480 - 1720	23	12000 - 15500
		24	15500 - 22050

Tabela 2.2: As 25 bandas críticas do sistema auditivo humano.

alguns sistemas de *voice morphing* não utilizam partes dos algoritmos de reconhecimento, mas sim conceitos pertinentes, em especial, o *cepstrum*. Esse conceito, portanto, é brevemente lembrado a seguir.

2.5.3 O *cepstrum*

Conforme foi visto anteriormente, o sinal de voz é composto por uma fonte de excitação, periódica ou não, seguida de um processo de filtragem, através dos tratos vocal e nasal. Este último processo de filtragem, e até o processo de geração do sinal excitador, depende muito do locutor e suas características bio-físicas peculiares. Em particular, é muito importante separar o sinal de fala, de tal forma que se isole o sinal de excitação das características dos tratos vocal e nasal, principalmente quando a excitação for periódica, isto é, *voiced speech*. Este é justamente o papel da técnica de *cepstrum* [24].

Se $x[n]$ for o sinal temporal de excitação periódica dos pulmões e $h[n]$ for o filtro correspondente aos tratos vocal e nasal, então o sinal de fala, isto é, a saída do sistema, será:

$$y[n] = x[n] * h[n].$$

No domínio da frequência, isso corresponde a:

$$Y[\omega] = X[\omega]H[\omega] \rightarrow$$

$$\log(Y[\omega]) = \log(X[\omega]H[\omega]) \rightarrow$$

Figura 2.14: [acima]: espectro de magnitude (DFT) de um sinal *voiced speech* em escala logarítmica ; [abaixo]: *cepstrum*, sendo possível observar claramente as frequências formantes ($F_1, \dots, F_n$) à esquerda e o período de *pitch* ($\frac{1}{F_0}$) à direita.

$$\log(Y[\omega]) = \log(X[\omega]) + \log(H[\omega]).$$

No domínio temporal, isso corresponde a $\bar{y} = \bar{x} + \bar{h}$, onde $\bar{x}$ e $\bar{h}$ são, respectivamente, versões de $x[\]$ e $h[\]$ em outra escala mais "dilatada", fazendo com que um esteja distante do outro. Essa escala é a *quefrequency*, medida em segundos, e a separação dos dois é chamada de *lifter*. A figura 2.14 elucida a presente explicação.

2.6 Algoritmos para *Voice Morphing*

Finalmente, esta seção apresenta uma revisão dos principais algoritmos para *voice morphing*, desde os mais simples até os mais elaborados, fazendo uso dos conceitos anteriormente apresentados.

Os sistemas mais elementares de *voice morphing* eram baseados em dois estágios, onde o primeiro utilizava alguma técnica de reconhecimento de voz (*speech-to-text*) para então sintetizar a *morphed speech* com base em alguns parâmetros que modelavam o *target speaker*. Esse tipo de sistema é bastante complexo e está sujeito a vários problemas, principalmente porque o estágio inicial de reconhecimento de fala nunca é perfeito. Além disso, a complexidade computacional de um sistema como esse é bastante grande, sem contar os artefatos introduzidos.

Mais à frente, outras tentativas foram experimentadas. Arslan e Talkin [27] propuseram um sistema no qual as vozes de ambos os locutores, *source* e *target*, eram segmentadas em fonemas e transcritos por intermédio de um dicionário pré-existente, sendo que a qualidade da voz produzida ficava limitada ao tamanho do dicionário.

Stylianou *et al* [28] sugeriu melhorias nesse sistema através do uso de modelos de mistura Gaussianos [25] para modelar os parâmetros espectrais dos locutores. Particularmente, as vozes do *source* e do *target* são alinhadas no tempo, seguido da aplicação do modelo Gaussiano, previamente treinado por intermédio do algoritmo de *expectation-maximization* (EM), para realizar o mapeamento. Essa técnica produz resultados um pouco mais próximos de uma voz natural.

Kain [29] apresentou um algoritmo de mapeamento do envelope espectral da mesma maneira como fizeram Arslan e Talkin, incluindo a predição do erro na tentativa de melhora dos resultados. Orphanidou *et al* [30] usou *Generative Topographic Mapping*, que é um modelo de mistura Gaussiana não-linear para transformar os parâmetros espectrais do *source* naqueles do *target*, de acordo com o modelo do trato vocal do locutor.

Um dos problemas mais significantes nas abordagens anteriores é a perda de informação durante o processo de síntese da voz do *target speaker*, especialmente nas frequências mais altas, o que leva a resultados não muito satisfatórios em termos de qualidade dos sons não vocálicos convertidos. Uma abordagem tempo-frequência baseada em *wavelets* foi proposta em [31] e apresentou resultados promissores, já que as características de reconstrução perfeita do sinal sob análise, assim como a baixa perda de informação, estavam asseguradas. Mais especificamente, a representação dos sinais de voz em multi-escala faz com que o problema de conversão seja sub-dividido em vários problemas de menores dimensões. Em seguida à aplicação da transformada *wavelet*, uma rede neural tipo *Radial Basis Function* (RBF) foi usada para fazer o mapeamento do *source speaker* para o *target speaker*, sub-banda por sub-banda. O algoritmo completo é o seguinte:

- INÍCIO
- PASSO 1: escolher os sinais do *source speaker* e do *target speaker*, sendo que ambos devem estar pronunciando a mesma sentença;
- PASSO 2: alinhar os sinais no tempo, isto é, reamostrá-los de forma que eles contenham o mesmo comprimento;

- PASSO 3: normalizar os sinais para que eles possuam média 0 e desvio-padrão 1;
- PASSO 4: aplicar a DWT nível 5, obtendo 6 conjuntos de coeficientes: a aproximação do nível 6, e os detalhes dos níveis 1 até 5;
- PASSO 5: setar os detalhes dos níveis 1 e 2 para zero;
- PASSO 6: normalizar os detalhes dos níveis 3 até 6;
- PASSO 7: para a aproximação do nível 6, e os detalhes dos níveis 3 até 6, treinar uma rede *RBF*, de forma que o sinal *source* se transforme, mediante passagem pela rede, no sinal *target*, amostra por amostra;
- PASSO 8: aplicar a DWT inversa de forma a inverter o passo 4;
- PASSO 9: desnormalizar o sinal *target* obtido de forma a inverter o passo 3;
- PASSO 10: testar o sinal de saída obtido.
- **FIM.**

Embora este algoritmo seja considerado um dentre os que fazem parte do estado-da-arte em termos de conversão de voz, os autores não apresentam um estudo sobre a escolha dos filtros *wavelet* utilizados, ao contrário, eles simplesmente adotam o uso de uma família de filtros com determinado comprimento.

O algoritmo proposto nesta dissertação, detalhadamente descrito no próximo capítulo, baseia-se diretamente na última técnica apresentada, com algumas modificações e melhorias. Além disso, diversas famílias de filtros, com diversos suportes, são testadas e os resultados são discutidos.

Capítulo 3

Descrição do Sistema Proposto

Este capítulo descreve com detalhes o sistema proposto para voice morphing, incluindo a arquitetura do sistema, assim como os detalhes sobre cada passo do algoritmo.

3.1 A Arquitetura do sistema proposto

Basicamente, o sistema proposto de *voice morphing* é constituído de um módulo de treinamento e outro de aplicação, conforme exhibe a figura 3.1. O módulo de treinamento recebe os sinais de voz do *source* e do *target speaker* na entrada, faz a separação de ambos em bandas críticas, e treina uma rede *multilayer perceptron* para converter o sinal do *source speaker*. Em seguida, o sinal do *test speaker* é usado como entrada do módulo de aplicação, logo após ter sido sub-dividido em bandas críticas, para ser convertido. A saída do módulo de aplicação, devidamente convertida para o domínio temporal, é gravada no disco rígido em formato *wav* [10]. O processo de separação dos sinais em bandas críticas, bem como a inversão deste, é totalmente baseada na transformada *wavelet*, conforme detalhado no algoritmo a seguir.

3.2 O Algoritmo de *voice morphing*

O algoritmo completo de *voice morphing* utilizado no presente trabalho é apresentado a seguir.

- **INÍCIO**
- **PASSO 1:** definir os arquivos de voz do *source speaker*, *target speaker*, e o arquivo a ser convertido (*test speaker*), lembrando que os dois primeiros correspondem a uma mesma sentença pronunciada pelos dois locutores;

Figura 3.1: Arquitetura básica do sistema proposto.

- **PASSO 2:** desmontar os três arquivos de voz, de formato wav, para extrair os dados brutos;
- **PASSO 3:** alinhar o início e o fim dos sinais do *source* e do *target speaker*, fazendo com que eles tenham a mesma quantidade de amostras discretas, aumentando a quantidade de pontos do sinal mais curto. Isso equivale a re-amostrar o sinal mais curto, o que é feito por meio de interpolação quadrática entre os pontos.
- **PASSO 4:** completar com amostras de valor zero (*zero-padding*) os sinais do *source* e do *target speaker* de forma que eles passem a ter comprimento múltiplo de 256;
- **PASSO 5:** normalizar os sinais do *source* e do *target speaker* de forma que eles passem a ter média 0 e desvio-padrão 1. Para isso, são armazenados os coeficientes que possibilitem inverter esse processo, que são a média dos sinais *source* e *target*, M_s e M_t , assim como os desvios originais de cada um, D_s e D_t ;
- **PASSO 6:** segmentar o sinal do *source speaker* em sub-sinais de 256 amostras, separando as partes *voiced* e *unvoiced*. A separação é realizada verificando se existem alterações no período de *pitch*, também fazendo uso da DWTP, de acordo com o algoritmo em [11]. Para cada sub-sinal s_i :
 - **PASSO 6.1:** obter a DWTP nível 8, $(DWTP8(s_i))$, que também possui 256 pontos;

- **PASSO 6.2:** transladar e normalizar $DWTP8(s_i)$ de forma que a menor e maior amplitude sejam 0 e 1 respectivamente. Os coeficientes de translação e normalização que possibilitam reverter o processo são, respectivamente, $CT_{DWTP8(s_i)}$ e $CN_{DWTP8(s_i)}$;
- **PASSO 6.3:** dividir $DWTP8(s_i)$ em 21 sub-sinais, de forma que eles correspondam as sub-bandas críticas do ouvido humano, cada qual com um comprimento. Para isso, foi criada uma aproximação de tais sub-bandas de acordo com as amostras das transformadas *wavelet*, como segue na tabela 3.1. Cada novo sub-sinal é referenciado como $DWTP8(s_i)_j$;
- **PASSO 7:** segmentar o sinal do *target speaker* em sub-sinais de 256 amostras. Para cada sub-sinal t_i :
 - **PASSO 7.1:** obter a DWTP nível 8, ($DWTP8(t_i)$), que também possui 256 pontos;
 - **PASSO 7.2:** transladar e normalizar $DWTP8(t_i)$ de forma que a menor e maior amplitudes sejam 0 e 1, respectivamente. Os coeficientes de translação e normalização que possibilitam reverter o processo são, respectivamente, $CT_{DWTP8(t_i)}$ e $CN_{DWTP8(t_i)}$;
 - **PASSO 7.3:** idem do passo 6.3, para $DWTP8(t_i)$ também de acordo com a tabela 3.1. Cada novo sub-sinal é referenciado como $DWTP8(t_i)_j$;
- **PASSO 8** para cada par $DWTP8(s_i)_j$ e $DWTP8(t_i)_j$:
 - **PASSO 8.1:** criar n redes neurais *multilayer perceptron* onde n corresponde ao comprimento de $DWTP8(s_i)_j$ e $DWTP8(t_i)_j$. As redes recebem como entrada cada amostra de $DWTP8(s_i)_j$, ou seja a quantidade de neurônios das camadas de entrada é n , que é a mesma quantidade de neurônios das camadas intermediárias. Cada rede possui um único neurônio de saída que deverá, após o treinamento, produzir um valor de saída correspondente ao n -ésimo componente de $DWTP8(t_i)_j$;
 - **PASSO 8.2:** treinar cada rede usando o algoritmo de retropropagação, até que o erro de cada rede seja inferior a 10^{-3} ;
- **PASSO 9:** apresentar o sinal do *test speaker* para o sistema treinado, obtendo o sinal de voz convertido.
- **FIM.**

O processo de treinamento comumente utilizado para atualizar os pesos da RNA, de acordo com os dados de entrada, encontra-se descrito com detalhes em [7] e [8], e também foi adotado neste trabalho, corresponde ao seguinte:

B.S.(z)	B.C. (Hz)	B.P. (Hz)	I.A.
0	0 - 100	0 - 93.75	0 - 2
1	100 - 200	93.75 - 187.5	3 - 5
2	200 - 300	187.5 - 312.5	6 - 9
3	300 - 400	312.5 - 406.25	10 - 12
4	400 - 510	406.25 - 500	13 - 15
5	510 - 630	500 - 625	16 - 19
6	630 - 770	625 - 781.25	20 - 24
7	770 - 920	781.25 - 906.25	25 - 28
8	920 - 1080	906.25 - 1093.75	29 - 34
9	1080 - 1270	1093.75 - 1281.25	35 - 40
10	1270 - 1480	1281.25 - 1468.75	41 - 46
11	1480 - 1720	1468.75 - 1718.75	47 - 54
12	1720 - 2000	1718.75 - 2000	55 - 63
13	2000 - 2320	2000 - 2312.5	64 - 73
14	2320 - 2700	2312.5 - 2687.5	74 - 85
15	2700 - 3150	2687.5 - 3156.25	86 - 100
16	3150 - 3700	3156.25 - 3687.5	101 - 117
17	3700 - 4400	3687.5 - 4406.5	118 - 140
18	4400 - 5300	4406.5 - 5312.75	141 - 169
19	5300 - 6400	5312.75 - 6406.5	170 - 204
20	6400 - 7700	6406.5 - 8000	205 - 255

Tabela 3.1: Escala Bark (B.S.), banda crítica correspondente (B.C.), banda crítica proposta (B.P.) e índice da amostra (I.A.) da DWT que leva a tais frequências, levando em conta que cada *frame* do sinal tem 256 amostras e foi amostrado a uma taxa de 16000 amostras por segundo.

- para encontrar os novos pesos, (wh), da camada intermediária, repita:
 - para $0 \leq j < \text{quantidade-de-neurônios-da-camada-intermediária}$:
 - * para $0 \leq i < \text{quantidade-de-neurônios-da-camada-de-entrada}$:
 - $slope_o \leftarrow XO * (1 - XO)$
 - $slope_h \leftarrow XH_{i,j} * (1 - XH_{i,j})$
 - $\frac{\Delta XO}{\Delta w} \leftarrow X_{1,i} slope_h w_j slope_o$
 - $wh_{i,j} \leftarrow wh_{i,j} + \frac{\Delta XO}{\Delta wh} tm$
- até que o erro seja menor que 10^{-3}
- para encontrar os novos pesos da camada de saída (wo), repita:
 - para $0 \leq j < \text{quantidade-de-neurônios-da-camada-intermediária}$:
 - * $slope_o \leftarrow XO * (1 - XO)$
 - * $\frac{\Delta XO}{\Delta w} \leftarrow XH_{i,i} slope_o$
 - * $wo_{,j} \leftarrow wo_{,j} + \frac{\Delta XO}{\Delta wo} tm$
- até que o erro seja menor que 10^{-3}

onde t é o erro para a entrada corrente, e m é uma constante fixada em 10^{-1} , XO e XH representam neurônios de saída e da camada intermediária da RNA, respectivamente. Como pode ser observado, o algoritmo de treinamento está baseado na curvatura da função de atualização dos pesos (variáveis $slope_o$ e $slope_h$), fazendo com que, implicitamente, sejam necessários cálculos de derivadas. Este é, conforme já foi dito, um dos motivos para escolha de funções do tipo sigmóide, que são facilmente diferenciáveis, já que $s'(x) = s(x)(1 - s(x))$.

Todo o algoritmo foi implementado em linguagem C++ sob ambiente Mandriva LINUX, sendo que o código-fonte correspondente se encontra no apêndice II.

Capítulo 4

Testes e Resultados

Neste capítulo encontram-se os testes realizados, variando-se os filtros wavelet, bem como os tamanhos dos suportes, e os resultados obtidos em cada uma das experimentações. Particularmente, os testes avaliam o número de iterações necessárias para convergência das RNAs, com determinado erro fixado, assim como a qualidade da voz convertida. Esta qualidade é medida em termos de distância Euclidiana e também de uma avaliação perceptual, que compara sentenças convertidas com aquelas ditadas pelo próprio locutor, ambas de mesmo conteúdo.

4.1 Testes de Convergência

Os testes de convergência das RNAs, variando-se as *wavelets*, encontram-se nas tabelas 4.1 e 4.2.

Em particular, na parte superior da tabela 4.1, a convergência foi analisada para a sentença *sa1.wav* do diretório */timit/test/dr1/mdab0* da base TIMIT [9], do *Linguistic Data Consortium*, utilizada como *source speech*, para ser convertida na sentença *sa1.wav* do diretório */timit/test/dr1/mjsw0*, da mesma base e de mesmo conteúdo textual, sendo que ambos os locutores são do sexo masculino. Seguindo o mesmo procedimento, a parte inferior desta tabela descreve os resultados obtidos quando a sentença *sa1.wav* do diretório */timit/test/dr1/faks0* é convertida para a *sa1.wav* do diretório */timit/test/dr1/fdac1*, sendo ambas de mesmo conteúdo textual, mas com ambos os locutores femininos.

A parte superior da tabela 4.2 descreve os resultados obtidos quando a sentença *sa1.wav* do diretório */timit/test/dr1/mwb0* é convertida para a *sa1.wav* do diretório */timit/test/dr1/fjem0*, sendo ambas de mesmo conteúdo textual, o *source speaker* masculino e o *target speaker* feminino. Finalmente, a parte inferior desta ta-

bela faz o teste contrário, ou seja, quando a sentença *sal.wav* do diretório */timit/test/dr1/ffjem0* é convertida para a *sal.wav* do diretório */timit/test/dr1/mwbt0*, caracterizando conversão de voz de um locutor de sexo feminino para masculino.

As sentenças utilizadas são de conteúdo fonético rico, contendo trechos *voiced* e *unvoiced*, assegurando a experimentação para variados casos.

4.2 Testes de comparação dos resultados com distância Euclidiana

Os testes comparativos das vozes convertidas com suas versões originais, levando-se em conta a distância Euclidiana entre os sinais, original e convertido, encontram-se nas tabelas 4.3 e 4.4. A distância Euclidiana, d , entre dois vetores $s[]$ e $v[]$, ambos de comprimento c , é dada pela equação 4.1.

$$d = \sqrt{\sum_{i=0}^{c-1} (s_i - v_i)^2} \quad (4.1)$$

Da mesma forma como anteriormente, na parte superior da tabela 4.3, a convergência foi analisada para a sentença *sal.wav* do diretório */timit/test/dr1/mdab0* da TIMIT, utilizada como *source speech*, para ser convertida na sentença *sal.wav* do diretório */timit/test/dr1/mjsw0*, da mesma base e de mesmo conteúdo textual, sendo que ambos os locutores são do sexo masculino. Seguindo o mesmo procedimento, a parte inferior desta tabela descreve os resultados obtidos quando a sentença *sal.wav* do diretório */timit/test/dr1/faks0* é convertida para a *sal.wav* do diretório */timit/test/dr1/fadac1*, sendo ambas de mesmo conteúdo textual, mas com ambos os locutores femininos. Já a parte superior da tabela 4.4 descreve os resultados obtidos quando a sentença *sal.wav* do diretório */timit/test/dr1/mwbt0* é convertida para a *sal.wav* do diretório */timit/test/dr1/ffjem0*, sendo ambas de mesmo conteúdo textual, o *source speaker* masculino e o *target speaker* feminino. Finalmente, a parte inferior desta tabela faz o teste contrário, ou seja, quando a sentença *sal.wav* do diretório */timit/test/dr1/ffjem0* é convertida para a *sal.wav* do diretório */timit/test/dr1/mwbt0*, caracterizando conversão de voz de um locutor de sexo feminino para o masculino.

4.3 Testes de comparação dos resultados via critério perceptual

Os testes comparativos das vozes convertidas com suas versões pronunciadas pelos locutores originais, levando-se em conta a avaliação perceptual entre os sinais, original e convertido, encontram-se nas tabelas 4.5 e 4.6. Para essa comparação, 10 ouvintes voluntários, 5 do sexo masculino e 5 do sexo feminino, de idades entre 10 e 70 anos, foram questionados. Eles avaliaram cada uma das conversões numa escala de 1 até 5, onde 1 significa que a voz original não se parece em nada com a voz convertida, e 5 significa que ambas são idênticas, do ponto de vista perceptual. Independente de qual a opinião dos ouvintes sobre a qualidade da conversão, para todas as sentenças, originais e convertidas, é possível entender perfeitamente o que está sendo dito, sendo que isto já é uma garantia de que, pelo menos, as conversões são inteligíveis.

[M $\rightarrow$ M]:

<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>
Haar	2	2051	Daubechies	4	501	Daubechies	6	505	Daubechies	8	500	Daubechies	10	500	Daubechies	12	500
Daubechies	14	490	Daubechies	16	480	Daubechies	18	480	Daubechies	20	479	Daubechies	22	480	Daubechies	24	479
Daubechies	26	493	Daubechies	28	479	Daubechies	30	477	Daubechies	32	479	Daubechies	34	480	Daubechies	36	479
Daubechies	38	482	Daubechies	40	469	Daubechies	42	470	Daubechies	44	476	Daubechies	46	471	Daubechies	48	470
Daubechies	50	480	Daubechies	52	465	Daubechies	54	461	Daubechies	56	470	Daubechies	58	442	Daubechies	60	440
Daubechies	62	471	Daubechies	64	450	Daubechies	66	440	Daubechies	68	440	Daubechies	70	441	Daubechies	72	439
Daubechies	74	470	Daubechies	76	440	Coiflet	6	1745	Coiflet	12	1790	Coiflet	18	1800	Coiflet	24	1845
Coiflet	30	1848	Symmlet	8	2100	Symmlet	16	2038	Beylkin	18	1946	Vaidyanathan	24	466			

[F $\rightarrow$ F]:

<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>
Haar	2	426	Daubechies	4	600	Daubechies	6	599	Daubechies	8	590	Daubechies	10	589	Daubechies	12	588
Daubechies	14	588	Daubechies	16	585	Daubechies	18	585	Daubechies	20	584	Daubechies	22	580	Daubechies	24	575
Daubechies	26	540	Daubechies	28	539	Daubechies	30	537	Daubechies	32	537	Daubechies	34	528	Daubechies	36	521
Daubechies	38	420	Daubechies	40	419	Daubechies	42	419	Daubechies	44	418	Daubechies	46	410	Daubechies	48	411
Daubechies	50	400	Daubechies	52	389	Daubechies	54	389	Daubechies	56	388	Daubechies	58	387	Daubechies	60	370
Daubechies	62	361	Daubechies	64	360	Daubechies	66	350	Daubechies	68	349	Daubechies	70	349	Daubechies	72	348
Daubechies	74	348	Daubechies	76	348	Coiflet	6	450	Coiflet	12	445	Coiflet	18	439	Coiflet	24	439
Coiflet	30	421	Symmlet	8	1900	Symmlet	16	1839	Beylkin	18	440	Vaidyanathan	24	182			

Tabela 4.1: Parte superior: Convergência do algoritmo para converter a sentença *sal.wav* do diretório */timit/test/dr1/mdab0* na *sal.wav* do diretório */timit/test/dr1/mjsw0*, onde ambos os locutores são do sexo masculino (M $\rightarrow$ M) ; Parte inferior: idem para converter a sentença *sal.wav* do diretório */timit/test/dr1/faks0* na *sal.wav* do diretório */timit/test/dr1/fadac1*, onde ambos os locutores são do sexo feminino (F $\rightarrow$ F); Legenda: DWT: base *wavelet* , SW: suporte correspondente dos filtros , QI: quantidade máxima de iterações para atingir um erro de $< 10^{-3}$ para cada amostra do sinal.

[M $\rightarrow$ F]:

<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>
Haar	2	2010	Daubechies	4	2050	Daubechies	6	2048	Daubechies	8	2048	Daubechies	10	2047	Daubechies	12	2038
Daubechies	14	2037	Daubechies	16	2035	Daubechies	18	2035	Daubechies	20	2033	Daubechies	22	2030	Daubechies	24	2029
Daubechies	26	2029	Daubechies	28	2028	Daubechies	30	2001	Daubechies	32	2000	Daubechies	34	2000	Daubechies	36	1990
Daubechies	38	1975	Daubechies	40	1975	Daubechies	42	1974	Daubechies	44	1975	Daubechies	46	1973	Daubechies	48	1970
Daubechies	50	1971	Daubechies	52	1960	Daubechies	54	1961	Daubechies	56	1960	Daubechies	58	1955	Daubechies	60	1950
Daubechies	62	1949	Daubechies	64	1948	Daubechies	66	1948	Daubechies	68	1945	Daubechies	70	1944	Daubechies	72	1940
Daubechies	74	1939	Daubechies	76	1939	Coiflet	6	600	Coiflet	12	590	Coiflet	18	589	Coiflet	24	580
Coiflet	30	526	Symmlet	8	550	Symmlet	16	522	Beylkin	18	425	Vaidyanathan	24	166			

[F $\rightarrow$ M]:

<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>	<i>DWT</i>	<i>SW</i>	<i>QI</i>
Haar	2		Daubechies	4	2051	Daubechies	6	2100	Daubechies	8	2099	Daubechies	10	2098	Daubechies	12	2099
Daubechies	14	2098	Daubechies	16	2097	Daubechies	18	2095	Daubechies	20	2095	Daubechies	22	2094	Daubechies	24	2080
Daubechies	26	2075	Daubechies	28	2070	Daubechies	30	2069	Daubechies	32	2060	Daubechies	34	2030	Daubechies	36	2030
Daubechies	38	2001	Daubechies	40	1976	Daubechies	42	1977	Daubechies	44	1975	Daubechies	46	1975	Daubechies	48	1970
Daubechies	50	1965	Daubechies	52	1960	Daubechies	54	1961	Daubechies	56	1960	Daubechies	58	1959	Daubechies	60	1961
Daubechies	62	1955	Daubechies	64	1950	Daubechies	66	1950	Daubechies	68	1940	Daubechies	70	1941	Daubechies	72	1939
Daubechies	74	1938	Daubechies	76	1938	Coiflet	6	700	Coiflet	12	690	Coiflet	18	691	Coiflet	24	600
Coiflet	30	600	Symmlet	8	610	Symmlet	16	601	Beylkin	18	415	Vaidyanathan	24	410			

Tabela 4.2: Parte superior: Convergência do algoritmo para converter a sentença *sal.wav* do diretório */timit/test/dr1/mwbt0* na *sal.wav* do diretório */timit/test/dr1/fjem0*, onde o *source speaker* é do sexo masculino e o *target speaker* é do sexo feminino (M $\rightarrow$ F) ; Parte inferior: idem para converter a sentença *sal.wav* do diretório */timit/test/dr1/fjem0* na *sal.wav* do diretório */timit/test/dr1/mwbt0*, onde *source speaker* é do sexo feminino e o *target speaker* é do sexo masculino (F $\rightarrow$ M); Legenda: DWT: base *wavelet* , SW: suporte correspondente dos filtros , QI: quantidade máxima de iterações para atingir um erro de $< 10^{-3}$ para cada amostra do sinal.

[M $\rightarrow$ M]:

<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>
Haar	2	812	Daubechies	4	810	Daubechies	6	810	Daubechies	8	806	Daubechies	10	806	Daubechies	12	800
Daubechies	14	800	Daubechies	16	801	Daubechies	18	800	Daubechies	20	802	Daubechies	22	800	Daubechies	24	800
Daubechies	26	798	Daubechies	28	799	Daubechies	30	800	Daubechies	32	798	Daubechies	34	797	Daubechies	36	797
Daubechies	38	795	Daubechies	40	794	Daubechies	42	796	Daubechies	44	790	Daubechies	46	792	Daubechies	48	790
Daubechies	50	792	Daubechies	52	794	Daubechies	54	790	Daubechies	56	793	Daubechies	58	800	Daubechies	60	789
Daubechies	62	793	Daubechies	64	794	Daubechies	66	790	Daubechies	68	794	Daubechies	70	794	Daubechies	72	789
Daubechies	74	794	Daubechies	76	790	Coiflet	6	808	Coiflet	12	803	Coiflet	18	800	Coiflet	24	800
Coiflet	30	796	Symmlet	8	800	Symmlet	16	798	Beylkin	18	795	Vaidyanathan	24	794			

[F $\rightarrow$ F]:

<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>
Haar	2	819	Daubechies	4	794	Daubechies	6	796	Daubechies	8	796	Daubechies	10	796	Daubechies	12	795
Daubechies	14	802	Daubechies	16	787	Daubechies	18	794	Daubechies	20	798	Daubechies	22	798	Daubechies	24	785
Daubechies	26	800	Daubechies	28	788	Daubechies	30	796	Daubechies	32	780	Daubechies	34	800	Daubechies	36	795
Daubechies	38	789	Daubechies	40	794	Daubechies	42	789	Daubechies	44	780	Daubechies	46	798	Daubechies	48	795
Daubechies	50	801	Daubechies	52	794	Daubechies	54	800	Daubechies	56	780	Daubechies	58	798	Daubechies	60	781
Daubechies	62	798	Daubechies	64	789	Daubechies	66	789	Daubechies	68	798	Daubechies	70	799	Daubechies	72	797
Daubechies	74	798	Daubechies	76	780	Coiflet	6	794	Coiflet	12	794	Coiflet	18	798	Coiflet	24	799
Coiflet	30	782	Symmlet	8	794	Symmlet	16	794	Beylkin	18	796	Vaidyanathan	24	792			

Tabela 4.3: Distância Euclidiana entre a sentença *sal.wav* do diretório */timit/ test/ dr1/ mdab0* e a *sal.wav* do diretório */timit/ test/ dr1/ mjsw0*, onde o *source speaker* e o *target speaker* são do sexo masculino (M $\rightarrow$ M) ; Parte inferior: idem para as sentenças *sal.wav* do diretório */timit/ test/ dr1/ faks0* e *sal.wav* do diretório */timit/ test/ dr1/ fadac1*, respectivamente, onde o *source speaker* e o *target speaker* são do sexo feminino (F $\rightarrow$ F); Legenda: DWT: base *wavelet* , SW: suporte correspondente dos filtros , DI: distância Euclidiana entre os sinais (original e convertido).

[M $\rightarrow$ F]:

<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>
Haar	2	1606	Daubechies	4	1602	Daubechies	6	1600	Daubechies	8	1593	Daubechies	10	1593	Daubechies	12	1593
Daubechies	14	1578	Daubechies	16	1584	Daubechies	18	1601	Daubechies	20	1595	Daubechies	22	1594	Daubechies	24	1591
Daubechies	26	1574	Daubechies	28	1586	Daubechies	30	1588	Daubechies	32	1587	Daubechies	34	1589	Daubechies	36	1594
Daubechies	38	1582	Daubechies	40	1584	Daubechies	42	1586	Daubechies	44	1588	Daubechies	46	1586	Daubechies	48	1588
Daubechies	50	1588	Daubechies	52	1588	Daubechies	54	1589	Daubechies	56	1586	Daubechies	58	1586	Daubechies	60	1586
Daubechies	62	1587	Daubechies	64	1588	Daubechies	66	1586	Daubechies	68	1584	Daubechies	70	1588	Daubechies	72	1586
Daubechies	74	1589	Daubechies	76	1586	Coiflet	6	1586	Coiflet	12	1584	Coiflet	18	1586	Coiflet	24	1584
Coiflet	30	1581	Symmlet	8	1586	Symmlet	16	1589	Beylkin	18	1582	Vaidyanathan	24	1580			

[F $\rightarrow$ M]:

<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>	<i>DWT</i>	<i>SW</i>	<i>DI</i>
Haar	2	1700	Daubechies	4	1701	Daubechies	6	1613	Daubechies	8	1612	Daubechies	10	1610	Daubechies	12	1610
Daubechies	14	1610	Daubechies	16	1613	Daubechies	18	1605	Daubechies	20	1603	Daubechies	22	1605	Daubechies	24	1606
Daubechies	26	1612	Daubechies	28	1612	Daubechies	30	1607	Daubechies	32	1609	Daubechies	34	1609	Daubechies	36	1609
Daubechies	38	1601	Daubechies	40	1613	Daubechies	42	1618	Daubechies	44	1601	Daubechies	46	1601	Daubechies	48	1605
Daubechies	50	1613	Daubechies	52	1612	Daubechies	54	1601	Daubechies	56	1601	Daubechies	58	1601	Daubechies	60	1609
Daubechies	62	1615	Daubechies	64	1589	Daubechies	66	1603	Daubechies	68	1601	Daubechies	70	1604	Daubechies	72	1608
Daubechies	74	1615	Daubechies	76	1590	Coiflet	6	1703	Coiflet	12	1702	Coiflet	18	1705	Coiflet	24	1690
Coiflet	30	1578	Symmlet	8	1590	Symmlet	16	1580	Beylkin	18	1579	Vaidyanathan	24	1575			

Tabela 4.4: Distância Euclidiana entre a sentença *sal.wav* do diretório */timit/ test/ dr1/ mwbt0* e a *sal.wav* do diretório */timit/ test/ dr1/ fjem0*, onde o *source speaker* é do sexo masculino e o *target speaker* é do sexo feminino (M $\rightarrow$ F) ; Parte inferior: idem para as sentenças *sal.wav* do diretório */timit/ test/ dr1/ fjem0* e *sal.wav* do diretório */timit/ test/ dr1/ mwbt0*, respectivamente, ou seja, o *source speaker* é do sexo feminino e o *target speaker* é do sexo masculino (F $\rightarrow$ M); Legenda: DWT: base *wavelet* , SW: suporte correspondente dos filtros , DI: distância Euclidiana entre os sinais (original e convertido).

[M $\rightarrow$ M]:

<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>
Haar	2	3,1	Daubechies	4	3,1	Daubechies	6	3,1	Daubechies	8	3,1	Daubechies	10	4,1	Daubechies	12	4,1
Daubechies	14	4,1	Daubechies	16	4,1	Daubechies	18	4,1	Daubechies	20	4,1	Daubechies	22	4,1	Daubechies	24	4,1
Daubechies	26	4,1	Daubechies	28	4,1	Daubechies	30	4,1	Daubechies	32	4,1	Daubechies	34	4,1	Daubechies	36	4,1
Daubechies	38	4,1	Daubechies	40	4,1	Daubechies	42	4,1	Daubechies	44	4,1	Daubechies	46	4,1	Daubechies	48	5,1
Daubechies	50	5,1	Daubechies	52	5,1	Daubechies	54	5,1	Daubechies	56	5,1	Daubechies	58	5,1	Daubechies	60	5,1
Daubechies	62	5,1	Daubechies	64	5,1	Daubechies	66	5,1	Daubechies	68	5,1	Daubechies	70	5,1	Daubechies	72	5,1
Daubechies	74	5,1	Daubechies	76	5,1	Coiflet	6	4,1	Coiflet	12	4,1	Coiflet	18	4,1	Coiflet	24	4,1
Coiflet	30	5,1	Symmet	8	3,1	Symmet	16	4,1	Beylkin	18	5,1	Vaidyanathan	24	5,1			

[F $\rightarrow$ F]:

<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>
Haar	2	3,1	Daubechies	4	3,1	Daubechies	6	3,1	Daubechies	8	3,1	Daubechies	10	4,1	Daubechies	12	4,1
Daubechies	14	4,1	Daubechies	16	4,1	Daubechies	18	4,1	Daubechies	20	4,1	Daubechies	22	4,1	Daubechies	24	4,1
Daubechies	26	4,1	Daubechies	28	4,1	Daubechies	30	4,1	Daubechies	32	4,1	Daubechies	34	4,1	Daubechies	36	4,1
Daubechies	38	4,1	Daubechies	40	4,1	Daubechies	42	4,1	Daubechies	44	4,1	Daubechies	46	4,1	Daubechies	48	5,1
Daubechies	50	5,1	Daubechies	52	5,1	Daubechies	54	5,1	Daubechies	56	5,1	Daubechies	58	5,1	Daubechies	60	5,1
Daubechies	62	5,1	Daubechies	64	5,1	Daubechies	66	5,1	Daubechies	68	5,1	Daubechies	70	5,1	Daubechies	72	5,1
Daubechies	74	5,1	Daubechies	76	5,1	Coiflet	6	3,1	Coiflet	12	4,1	Coiflet	18	4,1	Coiflet	24	4,1
Coiflet	30	5,1	Symmet	8	3,1	Symmet	16	4,1	Beylkin	18	5,1	Vaidyanathan	24	5,1			

Tabela 4.5: Distância Euclidiana entre a sentença *sal.wav* do diretório */timit/ test/ dr1/ mdab0* e a *sal.wav* do diretório */timit/ test/ dr1/ mjsw0*, onde o *source speaker* e o *target speaker* são do sexo masculino (M $\rightarrow$ M) ; Parte inferior: idem para as sentenças *sal.wav* do diretório */timit/ test/ dr1/ faks0* e *sal.wav* do diretório */timit/ test/ dr1/ fadac1*, respectivamente, onde o *source speaker* e o *target speaker* são do sexo feminino (F $\rightarrow$ F); Legenda: DWT: base *wavelet* , SW: suporte correspondente dos filtros , M,D: média e desvio-padrão das notas atribuídas pelos ouvintes para a qualidade da conversão das vozes.

[M $\rightarrow$ F]:

<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>
Haar	2	3,1	Daubechies	4	3,1	Daubechies	6	3,1	Daubechies	8	3,1	Daubechies	10	4,1	Daubechies	12	4,1
Daubechies	14	4,1	Daubechies	16	4,1	Daubechies	18	4,1	Daubechies	20	4,1	Daubechies	22	4,1	Daubechies	24	4,1
Daubechies	26	4,1	Daubechies	28	4,1	Daubechies	30	4,1	Daubechies	32	4,1	Daubechies	34	4,1	Daubechies	36	4,1
Daubechies	38	4,1	Daubechies	40	4,1	Daubechies	42	4,1	Daubechies	44	4,1	Daubechies	46	4,1	Daubechies	48	5,1
Daubechies	50	5,1	Daubechies	52	5,1	Daubechies	54	5,1	Daubechies	56	5,1	Daubechies	58	5,1	Daubechies	60	5,1
Daubechies	62	5,1	Daubechies	64	5,1	Daubechies	66	5,1	Daubechies	68	5,1	Daubechies	70	5,1	Daubechies	72	5,1
Daubechies	74	5,1	Daubechies	76	5,1	Coiflet	6	4,1	Coiflet	12	4,1	Coiflet	18	4,1	Coiflet	24	4,1
Coiflet	30	5,1	Symmlet	8	3,1	Symmlet	16	4,1	Beylkin	18	5,1	Vaidyanathan	24	5,1			

[F $\rightarrow$ M]:

<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>	<i>DWT</i>	<i>SW</i>	<i>M,D</i>
Haar	2	3,1	Daubechies	4	3,1	Daubechies	6	3,1	Daubechies	8	3,1	Daubechies	10	4,1	Daubechies	12	4,1
Daubechies	14	4,1	Daubechies	16	4,1	Daubechies	18	4,1	Daubechies	20	4,1	Daubechies	22	4,1	Daubechies	24	4,1
Daubechies	26	4,1	Daubechies	28	4,1	Daubechies	30	4,1	Daubechies	32	4,1	Daubechies	34	4,1	Daubechies	36	4,1
Daubechies	38	4,1	Daubechies	40	4,1	Daubechies	42	4,1	Daubechies	44	4,1	Daubechies	46	4,1	Daubechies	48	5,1
Daubechies	50	5,1	Daubechies	52	5,1	Daubechies	54	5,1	Daubechies	56	5,1	Daubechies	58	5,1	Daubechies	60	5,1
Daubechies	62	5,1	Daubechies	64	5,1	Daubechies	66	5,1	Daubechies	68	5,1	Daubechies	70	5,1	Daubechies	72	5,1
Daubechies	74	5,1	Daubechies	76	5,1	Coiflet	6	3,1	Coiflet	12	4,1	Coiflet	18	4,1	Coiflet	24	4,1
Coiflet	30	5,1	Symmlet	8	3,1	Symmlet	16	4,1	Beylkin	18	5,1	Vaidyanathan	24	5,1			

Tabela 4.6: Distância Euclidiana entre a sentença *sal.wav* do diretório */timit/ test/ dr1/ mwbt0* e a *sal.wav* do diretório */timit/ test/ dr1/ fjem0*, onde o *source speaker* é do sexo masculino e o *target speaker* é do sexo feminino (M $\rightarrow$ F) ; Parte inferior: idem para as sentenças *sal.wav* do diretório */timit/ test/ dr1/ fjem0* e *sal.wav* do diretório */timit/ test/ dr1/ mwbt0*, respectivamente, ou seja, o *source speaker* é do sexo feminino e o *target speaker* é do sexo masculino (F $\rightarrow$ M); Legenda: DWT: base *wavelet* , SW: suporte correspondente dos filtros , M,D: média e desvio-padrão das notas atribuídas pelos ouvintes para a qualidade da conversão das vozes.

Capítulo 5

Conclusões e Trabalhos Futuros

Este trabalho apresentou um estudo sobre conversão digital de voz. Após uma ampla revisão bibliográfica sobre os conceitos fundamentais de processamento de sinais, entre outros tópicos, definiu-se um algoritmo de conversão baseado numa combinação de redes neurais *multilayer perceptron* e transformada *wavelet*. Este algoritmo, que converte isoladamente as partes *voiced* e *unvoiced* dos sinais de voz, se mostrou bastante promissor, como já relatavam os testes similares documentados na literatura científica.

Durante os testes, foi possível identificar que a família Vaidyanathan, com filtros de suporte 24, foi a melhor, tanto em termos de velocidade de convergência no treinamento das redes, como na qualidade dos sinais de voz convertidos, em ambos os aspectos: perceptual, de acordo com os testes de preferência realizados com voluntários, assim como pela distância métrica adotada, isto é, a distância Euclidiana. Por outro lado, os filtros de Haar se mostraram os piores para ambos os aspectos. Todos os resultados se mantiveram para conversões de vozes de vários indivíduos da base TIMIT, independentemente do sexo dos locutores.

Observando as características da família Vaidyanathan, disponíveis na revisão bibliográfica, é possível verificar que tais filtros possuem as respostas ao impulso bem assimétricas em comparação com as outras *wavelets*, excluindo os filtros de Haar, que têm respostas ao impulso exatamente simétricas, mas respostas em frequência muito pobres. Portanto, a distorção de fase dos sinais filtrados com as *wavelets* de Vaidyanathan não foi fator prejudicial e a fase linear dos filtros de Haar não fez com que eles produzissem resultados ótimos.

Fazendo a comparação dos resultados obtidos para cada família de filtros isoladamente, percebe-se que os filtros de suportes não muito altos, isto é, por volta de 20 ou 30 elementos, apresentam melhores resultados, em todos os aspectos,

para todas as famílias de filtros. Isso significa que as respostas em frequências dos filtros devem ser suficientes para separar bem as frequências filtradas, mas não devem ser respostas ideais. Esse resultado abre uma interessante discussão, que é a de que os melhores filtros, em termos de respostas em frequência, não são os melhores para conversão de voz. Isso pode ser um indício de que o algoritmo de conversão se beneficia da contaminação de frequências que ocorre em todas as sub-bandas. Ainda que seja de pequenas proporções, esta contaminação é oriunda das sub-bandas adjacentes, devido às imperfeições nas respostas em frequências dos filtros.

Durante a reconstrução dos sinais, após a passagem pelas RNAs, os formatos suaves das funções *scaling* e *wavelet* das *wavelets* de Vaidyanathan, que não obedecem nenhuma condição de momento, se mostraram propícios, não introduzindo artefatos.

Em termos de trabalhos futuros, durante o doutorado do autor, primeiramente, pretende-se construir um algoritmo robusto o suficiente para realizar toda a conversão de voz sem a separação das partes *voiced* e *unvoiced* das vozes. Na sequência, serão tentadas abordagens baseadas em filtros *wavelets* "casados", isto é, *matched wavelets* [32] [33]. Para determinados padrões de voz, tais filtros devem otimizar os resultados, tanto em termos de velocidade de convergência no treinamento das redes, como em termos de qualidade das vozes convertidas.

Apêndice I - Coeficientes dos filtros *wavelet* utilizados nas experiências.

Coeficientes referentes aos filtros $h[\]$ (passa-baixas de análise).

- Haar :
{0.7071067, 0.7071067};
- Daub4 :
{ 4.82962913144e-01, 8.36516303737e-01, 2.24143868042e-01, -1.29409522551e-01};
- Daub6 :
{ 3.32670552950e-01, 8.06891509311e-01, 4.59877502118e-01, -1.35011020010e-01, -8.54412738820e-02, 3.52262918857e-02};
- Daub8 :
{ 2.30377813308e-01, 7.14846570552e-01, 6.30880767929e-01, -2.79837694168e-02, -1.87034811719e-01, 3.08413818355e-02, 3.28830116668e-02, -1.05974017850e-02};
- Daub10 :
{ 1.60102397974e-01, 6.03829269797e-01, 7.24308528437e-01, 1.38428145901e-01, -2.42294887066e-01, -3.22448695846e-02, 7.75714938400e-02, -6.24149021279e-03, -1.25807519990e-02, 3.33572528547e-03};
- Daub12 :
{ 1.11540743350e-01, 4.94623890398e-01, 7.51133908021e-01, 3.15250351709e-01, -2.26264693965e-01, -1.29766867567e-01, 9.75016055873e-02, 2.75228655303e-02, -3.15820393174e-02, 5.53842201161e-04, 4.77725751094e-03, -1.07730108530e-03};
- Daub14 :
{ 7.78520540850e-02, 3.96539319481e-01, 7.29132090846e-01, 4.69782287405e-01, -1.43906003928e-01, -2.24036184993e-01, 7.13092192668e-02, 8.06126091510e-02, -3.80299369350e-02, -1.65745416306e-02, 1.25509985560e-02, 4.29577972921e-04, -1.80164070404e-03, 3.53713799974e-04};
- Daub16 :
{ 5.44158422431e-02, 3.12871590914e-01, 6.75630736297e-01, 5.85354683654e-01, -1.58291052563e-02, -2.84015542961e-01, 4.72484573913e-04, 1.28747426620e-01, -1.73693010018e-02, -4.40882539307e-02, 1.39810279173e-02, 8.74609404740e-03, -4.87035299345e-03, -3.91740373376e-04, 6.75449406450e-04, -1.17476784124e-04};

- Daub18 :
{ 3.80779473638e-02, 2.43834674612e-01, 6.04823123690e-01, 6.57288078051e-01, 1.33197385825e-01, -2.93273783279e-01, -9.68407832229e-02, 1.48540749338e-01, 3.07256814793e-02, -6.76328290613e-02, 2.50947114831e-04, 2.23616621236e-02, -4.72320475775e-03, -4.28150368246e-03, 1.84764688305e-03, 2.30385763523e-04, -2.51963188942e-04, 3.93473203162e-05};
- Daub20 :
{ 2.66700579005e-02, 1.88176800077e-01, 5.27201188931e-01, 6.88459039453e-01, 2.81172343660e-01, -2.49846424327e-01, -1.95946274377e-01, 1.27369340335e-01, 9.30573646035e-02, -7.13941471663e-02, -2.94575368218e-02, 3.32126740593e-02, 3.60655356695e-03, -1.07331754833e-02, 1.39535174705e-03, 1.99240529518e-03, -6.85856694959e-04, -1.16466855129e-04, 9.35886703200e-05, -1.32642028945e-05};
- Daub22 :
{ 1.86942977614e-02, 1.44067021150e-01, 4.49899764356e-01, 6.85686774916e-01, 4.11964368947e-01, -1.62275245027e-01, -2.74230846817e-01, 6.60435881966e-02, 1.49812012466e-01, -4.64799551166e-02, -6.64387856950e-02, 3.13350902190e-02, 2.08409043601e-02, -1.53648209062e-02, -3.34085887301e-03, 4.92841765605e-03, -3.08592858815e-04, -8.93023250666e-04, 2.49152523552e-04, 5.44390746993e-05, -3.46349841869e-05, 4.49427427723e-06};
- Daub24 :
{ 1.31122579572e-02, 1.09566272821e-01, 3.77355135214e-01, 6.57198722579e-01, 5.15886478427e-01, -4.47638856537e-02, -3.16178453752e-01, -2.37792572560e-02, 1.82478605927e-01, 5.35956967435e-03, -9.64321200965e-02, 1.08491302558e-02, 4.15462774950e-02, -1.22186490697e-02, -1.28408251983e-02, 6.71149900879e-03, 2.24860724099e-03, -2.17950361862e-03, 6.54512821250e-06, 3.88653062820e-04, -8.85041092082e-05, -2.42415457570e-05, 1.27769522193e-05, -1.52907175806e-06};
- Daub26 :
{ 9.20213353896e-03, 8.28612438729e-02, 3.11996322160e-01, 6.11055851158e-01, 5.88889570431e-01, 8.69857261796e-02, -3.14972907711e-01, -1.24576730750e-01, 1.79476079429e-01, 7.29489336567e-02, -1.05807618187e-01, -2.64884064753e-02, 5.61394771002e-02, 2.37997225405e-03, -2.38314207103e-02, 3.92394144879e-03, 7.25558940161e-03, -2.76191123465e-03, -1.31567391189e-03, 9.32326130867e-04, 4.92515251262e-05, -1.65128988556e-04, 3.06785375793e-05, 1.04419305714e-05, -4.70041647936e-06, 5.22003509845e-07};
- Daub28 :
{ 6.46115346008e-03, 6.23647588493e-02, 2.54850267792e-01, 5.54305617940e-01, 6.31187849104e-01, 2.18670687758e-01, -2.71688552278e-01, -2.18033529993e-01, 1.38395213864e-01, 1.39989016584e-01, -8.67484115681e-02, -7.15489555040e-02,

02, 5.52371262592e-02, 2.69814083079e-02, -3.01853515403e-02, -5.61504953035e-03, 1.27894932663e-02, -7.46218989268e-04, -3.84963886802e-03, 1.06169108560e-03, 7.08021154235e-04, -3.86831947312e-04, -4.17772457703e-05, 6.87550425269e-05, -1.03372091845e-05, -4.38970490178e-06, 1.72499467536e-06, -1.78713996831e-07};

- Daub30 :

{ 4.53853736157e-03, 4.67433948927e-02, 2.06023863986e-01, 4.92631771708e-01, 6.45813140357e-01, 3.39002535454e-01, -1.93204139609e-01, -2.88882596566e-01, 6.52829528487e-02, 1.90146714007e-01, -3.96661765557e-02, -1.11120936037e-01, 3.38771439235e-02, 5.47805505845e-02, -2.57670073284e-02, -2.08100501696e-02, 1.50839180278e-02, 5.10100036040e-03, -6.48773456031e-03, -2.41756490761e-04, 1.94332398038e-03, -3.73482354137e-04, -3.59565244362e-04, 1.55896489920e-04, 2.57926991553e-05, -2.81332962660e-05, 3.36298718173e-06, 1.81127040794e-06, -6.31688232588e-07, 6.13335991330e-08};

- Daub32 :

{ 3.18922092534e-03, 3.49077143236e-02, 1.65064283488e-01, 4.30312722846e-01, 6.37356332083e-01, 4.40290256886e-01, -8.97510894024e-02, -3.27063310527e-01, -2.79182081330e-02, 2.11190693947e-01, 2.73402637527e-02, -1.32388305563e-01, -6.23972275247e-03, 7.59242360442e-02, -7.58897436885e-03, -3.68883976917e-02, 1.02976596409e-02, 1.39937688598e-02, -6.99001456341e-03, -3.64427962149e-03, 3.12802338120e-03, 4.07896980849e-04, -9.41021749359e-04, 1.14241520038e-04, 1.74787245225e-04, -6.10359662141e-05, -1.39456689882e-05, 1.13366086612e-05, -1.04357134231e-06, -7.36365678545e-07, 2.30878408685e-07, -2.10933963010e-08};

- Daub34 :

{ 2.24180700103e-03, 2.59853937036e-02, 1.31214903307e-01, 3.70350724152e-01, 6.10996615684e-01, 5.18315764056e-01, 2.73149704032e-02, -3.28320748363e-01, -1.26599752215e-01, 1.97310589565e-01, 1.01135489177e-01, -1.26815691778e-01, -5.70914196316e-02, 8.11059866541e-02, 2.23123361781e-02, -4.69224383892e-02, -3.27095553581e-03, 2.27336765839e-02, -3.04298998135e-03, -8.60292152032e-03, 2.96799669152e-03, 2.30120524215e-03, -1.43684530480e-03, -3.28132519409e-04, 4.39465427768e-04, -2.56101095665e-05, -8.20480320245e-05, 2.31868137987e-05, 6.99060098507e-06, -4.50594247722e-06, 3.01654960999e-07, 2.95770093331e-07, -8.42394844600e-08, 7.26749296856e-09};

- Daub36 :

{ 1.57631021844e-03, 1.92885317241e-02, 1.03588465822e-01, 3.14678941337e-01, 5.71826807766e-01, 5.71801654888e-01, 1.47223111969e-01, -2.93654040736e-01, -2.16480934005e-01, 1.49533975565e-01, 1.67081312763e-01, -9.23318841508e-02, -1.06752246659e-01, 6.48872162119e-02, 5.70512477385e-02, -4.45261419029e-02, -2.37332103958e-02, 2.66707059264e-02, 6.26216795430e-03, -1.30514809466e-03};

02, 1.18630033858e-04, 4.94334360546e-03, -1.11873266699e-03, -1.34059629833e-03, 6.28465682965e-04, 2.13581561910e-04, -1.98648552311e-04, -1.53591712353e-07, 3.74123788074e-05, -8.52060253744e-06, -3.33263447888e-06, 1.76871298362e-06, -7.69163268988e-08, -1.17609876702e-07, 3.06883586304e-08, -2.50793445494e-09};

- Daub38 :

{ 1.10866976318e-03, 1.42810984507e-02, 8.12781132654e-02, 2.64388431740e-01, 5.24436377464e-01, 6.01704549127e-01, 2.60894952651e-01, -2.28091394215e-01, -2.85838631755e-01, 7.46522697081e-02, 2.12349743306e-01, -3.35185419023e-02, -1.42785695038e-01, 2.75843506256e-02, 8.69067555558e-02, -2.65012362501e-02, -4.56742262772e-02, 2.16237674095e-02, 1.93755498891e-02, -1.39883886785e-02, -5.86692228101e-03, 7.04074736710e-03, 7.68954359257e-04, -2.68755180070e-03, 3.41808653458e-04, 7.35802520505e-04, -2.60676135678e-04, -1.24600791734e-04, 8.71127046721e-05, 5.10595048707e-06, -1.66401762971e-05, 3.01096431629e-06, 1.53193147669e-06, -6.86275565776e-07, 1.44708829879e-08, 4.63693777578e-08, -1.11640206703e-08, 8.66684883899e-10};

- Daub40 :

{ 7.79953613666e-04, 1.05493946249e-02, 6.34237804590e-02, 2.19942113551e-01, 4.72696185310e-01, 6.10493238938e-01, 3.61502298739e-01, -1.39212088011e-01, -3.26786800434e-01, -1.67270883090e-02, 2.28291050819e-01, 3.98502464577e-02, -1.55458750707e-01, -2.47168273386e-02, 1.02291719174e-01, 5.63224685730e-03, -6.17228996246e-02, 5.87468181181e-03, 3.22942995307e-02, -8.78932492390e-03, -1.38105261371e-02, 6.72162730225e-03, 4.42054238704e-03, -3.58149425960e-03, -8.31562172822e-04, 1.39255961932e-03, -5.34975984399e-05, -3.85104748699e-04, 1.01532889736e-04, 6.77428082837e-05, -3.71058618339e-05, -4.37614386218e-06, 7.24124828767e-06, -1.01199401001e-06, -6.84707959700e-07, 2.63392422627e-07, 2.01432202355e-10, -1.81484324829e-08, 4.05612705555e-09, -2.99883648961e-10};

- Daub42 :

{ 5.48822509852e-04, 7.77663905235e-03, 4.92477715381e-02, 1.81359625440e-01, 4.19687944939e-01, 6.01506094935e-01, 4.44590451927e-01, -3.57229196172e-02, -3.35664089530e-01, -1.12397071568e-01, 2.11564527680e-01, 1.15233298439e-01, -1.39940424932e-01, -8.17759429808e-02, 9.66003903237e-02, 4.57234057492e-02, -6.49775048937e-02, -1.86538592021e-02, 3.97268354278e-02, 3.35775639033e-03, -2.08920536779e-02, 2.40347092080e-03, 8.98882438197e-03, -2.89133434858e-03, -2.95837403893e-03, 1.71660704063e-03, 6.39418500512e-04, -6.90671117082e-04, -3.19640627768e-05, 1.93664650416e-04, -3.63552025008e-05, -3.49966598498e-05, 1.53548250927e-05, 2.79033053981e-06, -3.09001716454e-06, 3.16609544236e-07, 2.99213663046e-07, -1.00040087903e-07, -2.25401497467e-09, 7.05803354123e-09, -1.47195419765e-09, 1.03880557102e-10};

- Daub44 :

{ 3.86263231491e-04, 5.72185463133e-03, 3.80699372364e-02, 1.48367540890e-01, 3.67728683446e-01, 5.78432731009e-01, 5.07901090622e-01, 7.37245011836e-02, -3.12726580428e-01, -2.00568406104e-01, 1.64093188106e-01, 1.79973187992e-01, -9.71107984091e-02, -1.31768137686e-01, 6.80763143927e-02, 8.45573763668e-02, -5.13642542974e-02, -4.65308118275e-02, 3.69708466206e-02, 2.05867076275e-02, -2.34800013444e-02, -6.21378284936e-03, 1.25647252183e-02, 3.00137398507e-04, -5.45569198615e-03, 1.04426073918e-03, 1.82701049565e-03, -7.70690988123e-04, -4.23787399839e-04, 3.28609414213e-04, 4.34589990453e-05, -9.40522363481e-05, 1.13743496621e-05, 1.73737569575e-05, -6.16672931646e-06, -1.56517913199e-06, 1.29518205731e-06, -8.77987987336e-08, -1.28333622875e-07, 3.76122874933e-08, 1.68017140492e-09, -2.72962314663e-09, 5.33593882166e-10, -3.60211348433e-11};

- Daub46 :

{ 2.71904194128e-04, 4.20274889318e-03, 2.93100036578e-02, 1.20515531783e-01, 3.18450813852e-01, 5.44931147873e-01, 5.51018517241e-01, 1.81392625363e-01, -2.61392148030e-01, -2.71402098607e-01, 9.21254070824e-02, 2.23573658242e-01, -3.30374470942e-02, -1.64011321531e-01, 2.02830745756e-02, 1.12297043618e-01, -2.11262123562e-02, -7.02073915749e-02, 2.17658568344e-02, 3.84953325225e-02, -1.85235136501e-02, -1.75371010030e-02, 1.27519439315e-02, 6.03184065002e-03, -7.07531927370e-03, -1.13486547335e-03, 3.12287644981e-03, -2.46501400516e-04, -1.06123122888e-03, 3.19420492709e-04, 2.56762452007e-04, -1.50021850349e-04, -3.37889483412e-05, 4.42607120310e-05, -2.63520788924e-06, -8.34787556785e-06, 2.39756954684e-06, 8.14757483477e-07, -5.33900540520e-07, 1.85309178563e-08, 5.41754917953e-08, -1.39993549543e-08, -9.47288590181e-10, 1.05044645369e-09, -1.93240511131e-10, 1.25020330235e-11};

- Daub48 :

{ 1.91435800947e-04, 3.08208171490e-03, 2.24823399497e-02, 9.72622358336e-02, 2.72908916067e-01, 5.04371040839e-01, 5.74939221095e-01, 2.80985553233e-01, -1.87271406885e-01, -3.17943078999e-01, 4.77661368434e-03, 2.39237388780e-01, 4.25287296414e-02, -1.71175351370e-01, -3.87771735779e-02, 1.21016303469e-01, 2.09801137091e-02, -8.21616542080e-02, -4.57843624181e-03, 5.13016200399e-02, -4.94470942812e-03, -2.82131070949e-02, 7.66172188164e-03, 1.30499708710e-02, -6.29143537001e-03, -4.74656878632e-03, 3.73604617828e-03, 1.15376493683e-03, -1.69645681897e-03, -4.41618485614e-05, 5.86127059318e-04, -1.18123323796e-04, -1.46007981776e-04, 6.55938863930e-05, 2.18324146046e-05, -2.02288829261e-05, 1.34115775080e-08, 3.90110033859e-06, -8.98025314393e-07, -4.03250775687e-07, 2.16633965327e-07, -5.05764541979e-10, -2.25574038817e-08, 5.15777678967e-09, 4.74837582425e-10, -4.02465864458e-10, 6.99180115763e-11, -4.34278250380e-12};

- Daub50 :

{ 1.34802979347e-04, 2.25695959185e-03, 1.71867412540e-02, 7.80358628721e-

02, 2.31693507886e-01, 4.59683415146e-01, 5.81636896746e-01, 3.67885074802e-01, -9.71746409646e-02, -3.36473079641e-01, -8.75876145876e-02, 2.24537819745e-01, 1.18155286719e-01, -1.50560213750e-01, -9.85086152899e-02, 1.06633805018e-01, 6.67521644940e-02, -7.70841110565e-02, -3.71739628611e-02, 5.36179093987e-02, 1.55426059291e-02, -3.40423204606e-02, -3.07983679484e-03, 1.89228044766e-02, -1.98942578220e-03, -8.86070261804e-03, 2.72693625873e-03, 3.32270777397e-03, -1.84248429020e-03, -8.99977423746e-04, 8.77258193674e-04, 1.15321244046e-04, -3.09880099098e-04, 3.54371452327e-05, 7.90464000396e-05, -2.73304811996e-05, -1.27719529319e-05, 8.99066139306e-06, 5.23282770815e-07, -1.77920133265e-06, 3.21203751886e-07, 1.92280679014e-07, -8.65694173227e-08, -2.61159855611e-09, 9.27922448008e-09, -1.88041575506e-09, -2.22847491022e-10, 1.53590157016e-10, -2.52762516346e-11, 1.50969208282e-12};

- Daub52 :

{ 9.49379575071e-05, 1.65052023353e-03, 1.30975542925e-02, 6.22747440251e-02, 1.95039438716e-01, 4.13292962278e-01, 5.73669043034e-01, 4.39158311789e-01, 1.77407678098e-03, -3.26384593691e-01, -1.74839961289e-01, 1.81291832311e-01, 1.82755409589e-01, -1.04323900285e-01, -1.47977193275e-01, 6.98231861132e-02, 1.06482405249e-01, -5.34485616814e-02, -6.86547596040e-02, 4.22321857963e-02, 3.85357159711e-02, -3.13781103630e-02, -1.77609035683e-02, 2.07349201799e-02, 5.82958055531e-03, -1.17854979061e-02, -5.28738399262e-04, 5.60194723942e-03, -9.39058250473e-04, -2.14553028156e-03, 8.38348805654e-04, 6.16138220457e-04, -4.31955707426e-04, -1.06057474828e-04, 1.57479523860e-04, -5.27779549303e-06, -4.10967399639e-05, 1.07422154087e-05, 7.00007868296e-06, -3.88740016185e-06, -4.65046322064e-07, 7.93921063370e-07, -1.07900423757e-07, -8.90446637016e-08, 3.40779562129e-08, 2.16932825985e-09, -3.77601047853e-09, 6.78004724582e-10, 1.00230319104e-10, -5.84040818534e-11, 9.13051001637e-12, -5.25187122424e-13};

- Daub54 :

{ 6.68713138543e-05, 1.20553123167e-03, 9.95258878087e-03, 4.94525999829e-02, 1.62922027502e-01, 3.67110214125e-01, 5.53849860990e-01, 4.93406122677e-01, 1.02840855061e-01, -2.89716803314e-01, -2.48264581903e-01, 1.14823019517e-01, 2.27273288414e-01, -3.87864186318e-02, -1.78031740959e-01, 1.57993974602e-02, 1.31197971717e-01, -1.40627515558e-02, -9.10229065295e-02, 1.73110182654e-02, 5.79694057347e-02, -1.85124935619e-02, -3.27390666310e-02, 1.61469669223e-02, 1.56655956489e-02, -1.15771864589e-02, -5.86209634546e-03, 6.85663560968e-03, 1.34262687730e-03, -3.33285446952e-03, 1.45752962593e-04, 1.30117745024e-03, -3.41835122691e-04, -3.87901857410e-04, 2.01971987969e-04, 7.66005838706e-05, -7.71114551779e-05, -3.51748361490e-06, 2.06344264773e-05, -3.90116407063e-06, -3.65750090818e-06, 1.63436962472e-06, 3.05088068625e-07, -3.47246814739e-07, 3.28655896805e-08, 4.02625505286e-08, -1.32133227399e-08, -1.30946560685e-09, 1.52161498477e-09, -2.41552692801e-10, -4.37498622429e-11, 2.21366208806e-11, -3.29579012247e-12, 1.82818835288e-13};

- Daub56 :
{ 4.71080777501e-05, 8.79498515984e-04, 7.54265037764e-03, 3.90926081154e-02, 1.35137914253e-01, 3.22563361285e-01, 5.24998231630e-01, 5.30516293441e-01, 2.00176144045e-01, -2.30498954047e-01, -3.01327809532e-01, 3.28578791633e-02, 2.45808151373e-01, 3.69068853157e-02, -1.82877330732e-01, -4.68382337445e-02, 1.34627567910e-01, 3.44786312750e-02, -9.76853558056e-02, -1.73419228313e-02, 6.77478955019e-02, 3.44801895554e-03, -4.33333686160e-02, 4.43173291006e-03, 2.46880600101e-02, -6.81554976455e-03, -1.20635919682e-02, 5.83881662774e-03, 4.78486311245e-03, -3.72546124707e-03, -1.36037384563e-03, 1.87599866820e-03, 1.41567239314e-04, -7.48674955911e-04, 1.15465606365e-04, 2.29579098223e-04, -8.90390149004e-05, -4.90771341619e-05, 3.64140121105e-05, 4.63866498139e-06, -1.00432604133e-05, 1.24790031757e-06, 1.84036373451e-06, -6.67021547995e-07, -1.75746117320e-07, 1.49066001353e-07, -8.26238731562e-09, -1.78413869087e-08, 5.04404705638e-09, 6.94454032894e-10, -6.07704124722e-10, 8.49222001105e-11, 1.86736726378e-11, -8.36549047125e-12, 1.18885053340e-12, -6.36777235471e-14};
- Daub58 :
{ 3.31896627984e-05, 6.40951680304e-04, 5.70212651777e-03, 3.07735802214e-02, 1.11370116951e-01, 2.80653455970e-01, 4.89758804762e-01, 5.51374432758e-01, 2.89105238335e-01, -1.54028734459e-01, -3.30040948917e-01, -5.57068000729e-02, 2.36105236153e-01, 1.12419174873e-01, -1.60877988594e-01, -1.07845949938e-01, 1.14472295893e-01, 8.32207471624e-02, -8.51254926156e-02, -5.50274895253e-02, 6.34791645842e-02, 3.05315432727e-02, -4.51879812777e-02, -1.29171425542e-02, 2.94704318717e-02, 2.64832730767e-03, -1.70412245736e-02, 1.73788033272e-03, 8.46972549356e-03, -2.55080712778e-03, -3.47379898968e-03, 1.87712092572e-03, 1.08705394222e-03, -1.00077832708e-03, -2.00071136307e-04, 4.11128345474e-04, -2.29201804121e-05, -1.29304484008e-04, 3.64502606856e-05, 2.91334475016e-05, -1.65732839530e-05, -3.59364480402e-06, 4.75060924645e-06, -3.02905459205e-07, -8.97570175063e-07, 2.63389838699e-07, 9.38719741109e-08, -6.28615692201e-08, 1.07659190661e-09, 7.76897885477e-09, -1.89399538617e-09, -3.42680086326e-10, 2.40709945350e-10, -2.94058925076e-11, -7.83250973362e-12, 3.15276241337e-12, -4.28565487006e-13, 2.21919131158e-14};
- Daub60 :
{ 2.33861617273e-05, 4.66637950428e-04, 4.30079716504e-03, 2.41308326715e-02, 9.12383040670e-02, 2.42020670940e-01, 4.50487821853e-01, 5.57572232912e-01, 3.66242683371e-01, -6.61836707759e-02, -3.32966975020e-01, -1.41968513330e-01, 1.99462121580e-01, 1.77829873244e-01, -1.14558219432e-01, -1.57236817959e-01, 7.27786589703e-02, 1.22747746045e-01, -5.38064654582e-02, -8.76586900363e-02, 4.38016646714e-02, 5.67123657447e-02, -3.56733974967e-02, -3.22637589193e-02, 2.70786195952e-02, 1.52879607698e-02, -1.83997438681e-02, -5.29685966613e-03, 1.09156316583e-02, 6.19671756497e-04, -5.53073014819e-03, 8.43384586662e-04, 2.32452009406e-03, -8.60927696811e-04, -7.67878250438e-04, 5.05094823903e-05}

04, 1.72482584235e-04, -2.16171830116e-04, -8.54830546758e-06, 6.98200837080e-05, -1.33971686329e-05, -1.63615247872e-05, 7.25214553589e-06, 2.32754909849e-06, -2.18726767699e-06, 1.09947433852e-08, 4.26166232601e-07, -1.00041468235e-07, -4.76437996513e-08, 2.60544275497e-08, 5.55339786139e-10, -3.33110568046e-09, 6.98486269183e-10, 1.61362297827e-10, -9.46138799727e-11, 1.00010513139e-11, 3.23942863853e-12, -1.18523759210e-12, 1.54399757084e-13, -7.73794263095e-15};

- Daub62 :

{ 1.64801338645e-05, 3.39412203776e-04, 3.23688406862e-03, 1.88536916129e-02, 7.43360930116e-02, 2.07012874485e-01, 4.09192200037e-01, 5.51139840914e-01, 4.29468808206e-01, 2.71692124973e-02, -3.10955118319e-01, -2.17978485523e-01, 1.40178288765e-01, 2.24966711473e-01, -4.99263491604e-02, -1.86962360895e-01, 1.54369884294e-02, 1.45089500931e-01, -8.13983227346e-03, -1.07612773323e-01, 1.09412974523e-02, 7.53536117432e-02, -1.48800266181e-02, -4.86190754648e-02, 1.61541715659e-02, 2.80476193667e-02, -1.42762752777e-02, -1.39005529392e-02, 1.05176394873e-02, 5.51616357331e-03, -6.52085237587e-03, -1.42826422321e-03, 3.39306677671e-03, -6.39790110601e-05, -1.45904174198e-03, 3.43139829690e-04, 4.99881617563e-04, -2.39658346940e-04, -1.24341161725e-04, 1.08958435041e-04, 1.50133572744e-05, -3.63125515786e-05, 4.03452023518e-06, 8.79530134269e-06, -3.03514236589e-06, -1.36906023094e-06, 9.81001542204e-07, 5.32725065697e-08, -1.97592512917e-07, 3.61682651733e-08, 2.32830971382e-08, -1.06152960215e-08, -6.47431168795e-10, 1.40856815102e-09, -2.52404395415e-10, -7.34893003248e-11, 3.69210880887e-11, -3.32700896712e-12, -1.32433491724e-12, 4.44546709629e-13, -5.55944205057e-14, 2.69938287976e-15};

- Daub64 :

{ 1.16146330213e-05, 2.46656690638e-04, 2.43126191957e-03, 1.46810463814e-02, 6.02574991203e-02, 1.75750783639e-01, 3.67509628597e-01, 5.34317919340e-01, 4.77809163733e-01, 1.20630538265e-01, -2.66698181476e-01, -2.77421581558e-01, 6.47133548055e-02, 2.48310642356e-01, 2.46624448396e-02, -1.92102344708e-01, -4.89951171846e-02, 1.45232079475e-01, 4.44049081999e-02, -1.09456113116e-01, -2.96278725084e-02, 8.08741406384e-02, 1.41061515161e-02, -5.69263140624e-02, -2.38026446493e-03, 3.70514579235e-02, -4.14590766082e-03, -2.16628228363e-02, 6.16752731068e-03, 1.10174007154e-02, -5.41156825727e-03, -4.64921675118e-03, 3.62722464068e-03, 1.46895510046e-03, -1.96474055582e-03, -2.21167872957e-04, 8.67305851845e-04, -1.02453731060e-04, -3.05965442382e-04, 1.05391546173e-04, 8.10367832913e-05, -5.25980928268e-05, -1.29404577940e-05, 1.82426840198e-05, -6.36178153226e-07, -4.55830957626e-06, 1.20288903632e-06, 7.56004762559e-07, -4.28597069315e-07, -5.00336186874e-08, 8.96596631195e-08, -1.21992435948e-08, -1.10438302172e-08, 4.25042231198e-09, 4.38438779994e-10, -5.88109146263e-10, 8.90472379622e-11, 3.26327074133e-11, -1.43091876516e-11, 1.07561065350e-12, 5.36148222961e-13, -1.66380048943e-13, 2.00071530381e-14, -9.42101913953e-16};

- Daub66 :
{ 8.18635831417e-06, 1.79101615370e-04, 1.82270943516e-03, 1.13959433745e-02, 4.86146665317e-02, 1.48186313180e-01, 3.26718130117e-01, 5.09376172514e-01, 5.11254770583e-01, 2.09582350713e-01, -2.04202622398e-01, -3.15997410766e-01, -1.92783394369e-02, 2.45420612119e-01, 9.98515586803e-02, -1.71428099051e-01, -1.10844133116e-01, 1.21967856403e-01, 9.47880880506e-02, -9.11469683513e-02, -7.03024850540e-02, 7.01911439409e-02, 4.57345618938e-02, -5.34712513358e-02, -2.52485829774e-02, 3.86870607602e-02, 1.07032658200e-02, -2.57287617547e-02, -2.16775861735e-03, 1.53169541158e-02, -1.59428878241e-03, -7.95354038705e-03, 2.38906240816e-03, 3.48080095340e-03, -1.86071821445e-03, -1.20430925760e-03, 1.07438069635e-03, 2.72730584733e-04, -4.90832900759e-04, 4.39316625176e-06, 1.78043189825e-04, -4.16043851627e-05, -4.92956442341e-05, 2.42333539881e-05, 9.07080575782e-06, -8.86612136675e-06, -3.60751610287e-07, 2.28837127614e-06, -4.42692340795e-07, -3.98579129198e-07, 1.82244333257e-07, 3.37797270373e-08, -3.98783819851e-08, 3.67286357683e-09, 5.11121185734e-09, -1.67139267725e-09, -2.49640210524e-10, 2.42683310230e-10, -3.04957445394e-11, -1.42023685988e-11, 5.50941472076e-12, -3.34348121895e-13, -2.15248838683e-13, 6.21474024717e-14, -7.19651054536e-15, 3.28937367841e-16};
- Daub68 :
{ 5.77051063273e-06, 1.29947620067e-04, 1.36406139005e-03, 8.81988940388e-03, 3.90488413517e-02, 1.24152482111e-01, 2.87765059233e-01, 4.78478746279e-01, 5.30555099656e-01, 2.90366329507e-01, -1.28246842174e-01, -3.31525301508e-01, -1.03891915515e-01, 2.16907220187e-01, 1.66601750412e-01, -1.27337358223e-01, -1.60924927177e-01, 7.79918469379e-02, 1.34125960271e-01, -5.44829680641e-02, -1.02947596992e-01, 4.35760946496e-02, 7.31852354367e-02, -3.70128384178e-02, -4.74385596452e-02, 3.07397465739e-02, 2.72283507563e-02, -2.36717379228e-02, -1.31439800166e-02, 1.64093741998e-02, 4.71364926099e-03, -1.00455067083e-02, -6.19474884515e-04, 5.33495076875e-03, -7.69212797506e-04, -2.39945394353e-03, 8.58995987436e-04, 8.75199906407e-04, -5.52735576214e-04, -2.32673214023e-04, 2.65077239755e-04, 2.66005001845e-05, -9.91469777078e-05, 1.35311722724e-05, 2.84495141969e-05, -1.05765749425e-05, -5.71082651099e-06, 4.16987175854e-06, 4.97971810142e-07, -1.11630653481e-06, 1.44819570833e-07, 2.02599066666e-07, -7.52670174041e-08, -1.99034650153e-08, 1.74042333293e-08, -8.66574426136e-10, -2.31650194699e-09, 6.44637821032e-10, 1.30041031860e-10, -9.90477453763e-11, 1.00420873546e-11, 6.08012535400e-12, -2.10787910891e-12, 9.79945115821e-14, 8.57919405179e-14, -2.31708370390e-14, 2.58733838193e-15, -1.14894475448e-16};
- Daub70 :
{ 4.06793406114e-06, 9.42146947557e-05, 1.01912268037e-03, 6.80729288431e-03, 3.12362885114e-02, 1.03404455861e-01, 2.51307378994e-01, 4.43592739224e-01, 5.37008427509e-01, 3.60345640518e-01, -4.38838818739e-02, -3.23822864912e-01, -1.81786976766e-01, 1.66041357490e-01, 2.17299289321e-01, -6.52628713106e-01};

02, -3.35235840641e-02, -8.83349389041e-03, 2.26186515445e-02, 1.69047238348e-03, -1.37639819628e-02, 1.51930577883e-03, 7.38775745285e-03, -2.24805318700e-03, -3.39452327640e-03, 1.81687134380e-03, 1.26393425811e-03, -1.11148486531e-03, -3.28078847088e-04, 5.49053277337e-04, 1.53443902319e-05, -2.20894403245e-04, 4.33672612594e-05, 7.05513878206e-05, -3.09866292761e-05, -1.63916249616e-05, 1.35432771841e-05, 1.84994500311e-06, -4.30994155659e-06, 4.85473139699e-07, 1.00212139929e-06, -3.49494860344e-07, -1.50988538867e-07, 1.10903123221e-07, 5.35065751546e-09, -2.25219383672e-08, 4.22448570636e-09, 2.79397446595e-09, -1.29720500146e-09, -1.03141112909e-10, 1.94616489408e-10, -3.20339824412e-11, -1.39841571553e-11, 6.33495544097e-12, -2.09636319423e-13, -4.42161240987e-13, 1.13805283092e-13, -4.51888960746e-16, -5.24302569188e-15, 1.18901238750e-15, -1.19928033585e-16, 4.90661506493e-18};

- Daub76 :

{ 1.42577664167e-06, 3.57625199426e-05, 4.21170266472e-04, 3.08308811925e-03, 1.56372493475e-02, 5.78899436128e-02, 1.60071993564e-01, 3.30775781411e-01, 4.96591175311e-01, 4.93356078517e-01, 2.13050571355e-01, -1.82867667708e-01, -3.21675637808e-01, -6.22665060478e-02, 2.32125963835e-01, 1.49985119618e-01, -1.41795685973e-01, -1.59912565158e-01, 8.56381215561e-02, 1.41414734073e-01, -5.65864586307e-02, -1.14731170710e-01, 4.30958954330e-02, 8.72043982620e-02, -3.66051034028e-02, -6.17662087084e-02, 3.19898775315e-02, 4.00549811051e-02, -2.68914938808e-02, -2.31141340205e-02, 2.09046452556e-02, 1.12904972786e-02, -1.47018820653e-02, -4.13130665603e-03, 9.21478503219e-03, 5.62571574840e-04, -5.07131450921e-03, 7.16982182106e-04, 2.40069778189e-03, -8.44862666553e-04, -9.42461407722e-04, 5.81075975053e-04, 2.81763925038e-04, -3.03102046072e-04, -4.55568269666e-05, 1.26204335016e-04, -1.15540910383e-05, -4.17514164854e-05, 1.33417614992e-05, 1.03735918404e-05, -6.45673042846e-06, -1.55084435011e-06, 2.14996026993e-06, -8.48708758607e-08, -5.18773373887e-07, 1.39637754550e-07, 8.40035104689e-08, -4.88475793745e-08, -5.42427480028e-09, 1.03470453927e-08, -1.43632948779e-09, -1.34919775398e-09, 5.26113255735e-10, 6.73233649018e-11, -8.27825652253e-11, 1.10169293459e-11, 6.29153731703e-12, -2.48478923756e-12, 2.62649650406e-14, 1.80866123627e-13, -4.24981781957e-14, -4.56339716212e-16, 2.04509967678e-15, -4.40530704248e-16, 4.30459683955e-17, -1.71615245108e-18};

- sym8:

{0.032223100604, -0.012603967262, -0.099219543577, 0.297857795606, 0.803738751807, 0.497618667633, -0.029635527646, -0.075765714789};

- sym16:

{0.001889950333, -0.000302920515, -0.014952258337, 0.003808752014, 0.049137179674, -0.027219029917, -0.051945838108, 0.364441894835, 0.777185751701, 0.481359651258, -0.061273359068, -0.143294238351, 0.007607487325, 0.031695087811, -0.000542132332, -0.003382415951};

- coif6:
{-0.072732619513, 0.337897662458, 0.852572020212, 0.384864846864, -0.072732619513, -0.015655728135};
- coif12:
{0.016387336464, -0.041464936782, -0.067372554722, 0.386110066823, 0.812723635450, 0.417005184424, -0.076488599079, -0.059434418647, 0.023680171946, 0.005611434819, -0.001823208871, -0.000720549445};
- coif18:
{-0.003793512864, 0.007782596427, 0.023452696142, -0.065771911282, -0.061123390003, 0.405176902410, 0.793777222626, 0.428483476378, -0.071799821619, -0.082301927107, 0.034555027573, 0.015880544864, -0.009007976137, -0.002574517689, 0.001117518771, 0.000466216960, -0.000070983303, -0.000034599773};
- coif24:
{0.000892313669, -0.001629492013, -0.007346166328, 0.016068943965, 0.026682300156, -0.081266699681, -0.056077313317, 0.415308407030, 0.782238930921, 0.434386056491, -0.066627474263, -0.096220442034, 0.039334427123, 0.025082261845, -0.015211731528, -0.005658286687, 0.003751436157, 0.001266561929, -0.000589020756, -0.000259974552, 0.000062339034, 0.000031229876, -0.000003259680, -0.000001784985};
- coif30:
{-0.000212080840, 0.000358589688, 0.002178236358, -0.004159358782, -0.010131117521, 0.023408156788, 0.028168028974, -0.091920010569, -0.052043163181, 0.421566206733, 0.774289603730, 0.437991626216, -0.062035963969, -0.105574208714, 0.041289208754, 0.032683574270, -0.019761778945, -0.009164231163, 0.006764185449, 0.002433373213, -0.001662863702, -0.000638131343, 0.000302259582, 0.000140541150, -0.000041340432, -0.000021315027, 0.000003734655, 0.000002063762, -0.000000167443, -0.000000095177};
- Beylkin18 :
{0.099305765374353, 0.424215360812961, 0.699825214056600, 0.449718251149468, -0.110927598348234, -0.264497231446384, 0.026900308803690, 0.155538731877093, -0.017520746266529, -0.088543630622924, 0.019679866044322, 0.042916387274192, -0.017460408696028, -0.014365807968852, 0.010040411844631, 0.0014842347824723, -0.002736031626258, 0.0006404853285212};
- Vaidyanathan24:
{-0.000062906118, 0.000343631905, -0.000453956620, -0.000944897136, 0.002843834547, 0.000708137504, -0.008839103409, 0.003153847056, 0.019687215010, -0.014853448005, -0.035470398607, 0.038742619293, 0.055892523691, -0.077709750902, -0.083928884366, 0.131971661417, 0.135084227129, -0.194450471766, -0.263494802488, 0.201612161775, 0.635601059872, 0.572797793211, 0.250184129505, 0.045799334111}.

Apêndice II - Código fonte do algoritmo conversor em linguagem C++.

Apêndice III - Publicações durante o mestrado.

- GUIDO, R. C. ; VIEIRA, L. S. ; FONSECA, E. . (Status: to appear) A Neural network and wavelet-based approach for voice conversion. Neuro-computing, Elsevier, v. -, p. in-press, 2007.
- GUIDO, R.C., VIEIRA, L.S., BARBON, S.Jr, et al. A Fractal and Wavelet-based Approach for Audio Coding. Proc. of the 8th IEEE Int. Symposium on Multimedia (IEEE ISM), pp. 253-256, San Diego, CA, USA, 2006.
- GUIDO, R. C. ; PEREIRA, J. C. ; FONSECA, E. ; SANCHEZ, F. L. ; VIEIRA, L. S. ; GUILHERM, M. B. A. ; BARBON JR, S. . Support Vector Machines and Wavelets for Voice Disorder Sorting. In: 38th IEEE SSST - Southeastern Symposium on System Theory, 2006, Cookeville, TN, US. Proc. 38th IEEE SSST - Southeastern Symposium on System Theory. US : IEEE Press, 2006. v. 1. p. 434-438.
- GUIDO, R. C.; MACIEL, C. D.; MONTEIRO, M., FONSECA; E., PANCHAPAGESAN, S.; PEREIRA, J. C.; VIEIRA, L.; BARBON, S.Jr.; SANCHEZ, F.L.; GUILHERME, M.B.A.; SERGIO, K.I.C.; SCARPA, T.L.; FANTINATO, P.C.; MOURA, .J.R. A study on the best wavelet for audio compression. In: 40th IEEE ASILOMAR Int. Conference on Signals, Systems and Computers, Pacific-Grove, CA, USA, 2006.
- SANCHEZ, F. L. ; BARBON JR, S. ; GUIDO, R. C. ; VIEIRA, L. S. ; CARVALHO, N. F. . Wavelet-based Dynamic Time Warping for Speech Recognition. In: XXIX CNMAC - Congresso Nacional de Matemática Aplicada e Computacional., 2006, Campinas-SP. Proceedings of the XXIX CNMAC - Congresso Nacional de Matemática Aplicada e Computacional, 2006. p. CD-ROM.
- VIEIRA, L. S. ; GUIDO, R. C. ; PIMENTEL, C. C. ; FAVARO, G. A. C. ; BARBOSA, M. H. ; BARTOLOMEU, A. . Wavelet-based ECG abnormalities Detection. In: XXIX CNMAC - Congresso Nacional de Matemática Aplicada e Computacional., 2006, Campinas-SP. Proceedings of the XXIX CNMAC - Congresso Nacional de Matemática Aplicada e Computacional, 2006. p. CD-ROM.
- VIEIRA, L. S. ; GUILHERM, M. B. A. ; GUIDO, R. C. ; SEGIO, K. I. C. ; ESCARPA, T. L. . On the use of wavelets for image compression. In: XXIX CNMAC - Congresso Nacional de Matemática Aplicada e Computacional.,

2006, Campinas-SP. Proceedings of the XXIX CNMAC - Congresso Nacional de Matemática Aplicada e Computacional, 2006. p. CD-ROM.

- VIEIRA, L.S. ; DUCATTI, J.A. ; GUIDO, R.C. Construção de uma ferramenta de ensino baseada em logo. Proc. Jornada de Iniciação Científica da FATEC - Faculdade de Tecnologia de São José do Rio Preto-SP, 2006 (Resumo, CDROM).
- VIEIRA, L.S. ; GUIDO, R.C. ; BARBOSA, M. H. ; FAVARO, G. A. ; SANCHEZ, F. L. ; BARBON, S. Utilizando as transformadas wavelet e Fourier na detecção de Anormalidades em sinais de ECG. Proc. Jornada de Iniciação Científica da FATEC - Faculdade de Tecnologia de São José do Rio Preto-SP, 2006 (Resumo, CDROM).
- VIEIRA, L.S. ; SANCHEZ, F. L. ; GUIDO, R. C. Construção de um algoritmo para reconhecimento de patologias em sinais de fala humana. Proc. Jornada de Iniciação Científica da FATEC - Faculdade de Tecnologia de São José do Rio Preto-SP, 2006 (Resumo, CDROM).
- VIEIRA, L.S. ; FURLANI, J. ; BARBON, S. Jr. Compressão de imagens digitais utilizando a transformada discreta cosseno. Proc. Jornada de Iniciação Científica da FATEC - Faculdade de Tecnologia de São José do Rio Preto-SP, 2006 (Resumo, CDROM).
- GUIDO, R. C. ; PEREIRA, J. C. ; FONSECA, E. ; VIEIRA, L. S. ; SANCHEZ, F. L. Trying different wavelets on the search for voice disorders sorting. In: 37th IEEE International Southeastern Symposium on System Theory, 2005, Tuskegee-AL-EUA. Proc. 37th IEEE SSST - International Southeastern Symposium on System Theory. Tuskegee-AL-EUA : IEEE Press, 2005. v. 1. p. 495-499.
- GUIDO, R. C. ; VIEIRA, L. S. ; SANCHEZ, F. L. ; SLAETS, J. F. W. ; ALMEIDA, L. O. B. ; BIANCHI, M. F. . A matched FIR filter bank for audio coding. In: 7th IEEE International Symposium on Multimedia, Special Session on Audio Processing, 2005, Irvine, CA, US. Proc. IEEE ISM 2005. Irvine, CA, US : IEEE Press, 2005. v. 1. p. 796-801.

Referências Bibliográficas

- [1] WU, C.H. et al. Voice conversion using duration-embedded bi-HMMs for expressive speech synthesis. **IEEE Trans. on Audio, Speech and Language Processing**, v.14, p.1109 - 1116, n.4, July 2006.
- [2] MOUCHTARIS, A.; VAN DER SPIEGEL, J.; MUELLER, P. Non-parallel training for voice conversion by maximum likelihood constrained adaptation. **Proc. of the Int. Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings (ICASSP '04)**, v.1, p. I 1-4, May 2004, .
- [3] STRANG, G.; NGUYEN, T. **Wavelets and filter banks**. Wellesley-Cambridge Academic Press, Wellesley, 1997.
- [4] ADDISON, P. S. **The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance**. Edinburg-UK: Institute of Physics Publishing, 2002.
- [5] WALKER, J.S. **A primer on wavelets and their scientific applications**. Whashington-USA: Chapman and Hall/CRC, 1999.
- [6] JENSEN, A.; COUR-HARBO, A. **Ripples in mathematics: the discrete wavelet transform**. New York-USA: Springer-Verlag, 2000.
- [7] HAYKIN, S. **Redes neurais: princípios e prática**. 2.ed. Porto Alegre: Bookman, 2001.
- [8] KOVACS, Z.L. **Redes neurais artificiais: fundamentos e aplicações**. 3.ed. São Paulo: Livraria da Física, 2002.
- [9] Base de Dados TIMIT do *Linguistic Data Consortium*. <http://www.ldc.upenn.edu/>
- [10] GUIDO, R.C. **Spikelet: uma nova transformada wavelet aplicada ao reconhecimento digital de padrões, em tempo-real, de spikes e overlaps em sinais neurofisiológicos do campo visual da mosca**. IFSC-USP: São Carlos-SP, 11/2003, Tese de Doutorado.

- [11] GUIDO, R. C. ; VIEIRA, L.S.; FONSECA, E. S. (Status: to appear) A neural network and wavelet-based approach for voice conversion. **Neurocomputing, Elsevier**, v. -, p. in–press, 2007.
- [12] HAYKIN, S.; VEEN, B.V. **Sinais e sistemas**. Porto Alegre: Bookman, 2002.
- [13] OPPENHEIN, A.V.; SCHAFER, R.W. **Discrete time signal processing**. 2. ed. New York: Prentice Hall, 1999.
- [14] DEVASAHAYAM, S.R. **Signals and systems in biomedical engineering: signal processing and physiological system modeling**. New York: Kluwer Academic Publishers, 2000.
- [15] DELLER Jr. J.R.; PROAKIS, J. G. ;HANSEN, J. H. L. **Discrete-time processing of speech signals**, MacMillan Publishing Co., New York, 1993.
- [16] QUATIERE, T.F. **Discrete-time speech signal processing**. Upper Saddle River-USA: Prentice Hall, 2002.
- [17] MALLAT, S. G. **A wavelet tour of signal processing**. San Diego: Academic Press, 1999.
- [18] WILLIAMS, J. R.; AMARATUNGA, K. **Introduction to wavelets in engineering**, Int. Journal for Numerical Methods in Engineering, v. 37, pp. 2365-2388, 1994.
- [19] AKAY, M. **Time frequency and wavelets in biomedical signal processing**. New York: IEEE Press in Biomedical Engineering, 1998.
- [20] LIPSCHUTZ, S. **Álgebra Linear**. 3.ed. São Paulo: Makron Books, 1994.
- [21] VAPNIK, V. **The nature of statistical learning theory**. New York: Springer-Verlag, 1995.
- [22] BAKEN, R. J. **Clinical measument of speech and voice**, Singular Publishing Group Inc., San Diego, 1996.
- [23] NARAYANAN, S.; ALWAN, A. **Text to speech synthesis**. Upper Saddle River-USA: Prentice Hall, 2004.
- [24] DENG, L.; O'SHAUGHNESSY, O. **Speech processing: a dynamic and optimization-oriented approach**. Marcel Dekker Inc: New York-USA, 2003.
- [25] CHOU, W.; JUANG, B.H. **Pattern recognition in speech and language processing**. Boca Raton-USA: CRC Press, 2003.

- [26] BOSI, B.; GOLDBERG, R.E. **Introduction to digital audio coding and standards**. Boston-USA, 2003.
- [27] ARSLAN, L.; TALKIN, D. Voice conversion by codebook mapping of line spectral frequencies and excitation spectrum. **Proc. EUROSPEECH**, p. 1347-1350, 1997.
- [28] STYLIANOU, Y.; CAPPE, O.; MOULINES, E. Statistical methods for voice quality transformation. **In: European Conference on Speech Communication and Technology, EUROSPEECH, 1995, Madri-Spain. Proceedings Madri, 1995**. p. 447-450.
- [29] KAIN, A.; MACON, M.W. Spectral voice conversion for text-to-speech synthesis. **In: International conference on acoustics, speech, and Signal Processing, ICASSP 1998, Seattle-Washington, Proceedings Seattle-Washington, 1998** p. 285-288.
- [30] ORPHANIDOU, C.; MOROZ, I. M.; ROBERTZ, S. J. Voice morphing using the generative topographic mapping. **Proceedings of the International Conference on Computer, Communication and Control Technologies, 2003**. p. 222-225.
- [31] ORPHANIDOU, C. et al. Multiscale voice morphing. Pre-print submitted to **Pattern Recognition Letters, Elsevier, on September 2006, to appear in 2007**.
- [32] GUIDO, R. C. et al. A new technique to construct a *wavelet* transform matching a specified signal with applications to digital, real-time, spike and overlap pattern recognition. **Digital Signal Processing, Elsevier, EUA, 2006**, v. 16, n. 1, p. 24-44.
- [33] CHAPA, J.O.; RAO, R.M. Algorithms for designing *wavelets* to match a specified signal. **IEEE Transactions on Signal Processing. V 48. N 12. USA, Dez-2000**.