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

Dynamic Time Warping baseado na Transformada *Wavelet*

Sylvio Barbon Júnior

São Carlos - SP - Brasil
Agosto de 2007

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

Dynamic Time Warping baseado na Transformada *Wavelet*

Sylvio Barbon Júnior

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
Agosto de 2007

”Se o malandro soubesse como é bom ser honesto, seria honesto só de malandragem.”

Jorge Ben Jor

Ao meu orientador, companheiro de pesquisa e amigo, Rodrigo Capobianco Guido, pelo apoio, confiança, conhecimento e apreço.

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus pais e minha irmã que, mesmo sem saber, me apoiavam até em pequenos atos e gestos de confiança, sem contar as palavras de incentivo e motivação.

Em especial, a minha namorada, que muitas vezes até mesmo à distância, me fortalece e alegra com o seu carinho e amor.

Aos meus companheiros de jornada: Lucimar, *Turana*, *Marcião*, Kim e Fabrício. Sem a ”galera”, acho que seria muito difícil aguentar os momentos monótonos das viagens entre São José do Rio Preto e São Carlos.

A professora Rosely Sanches, pela oportunidade de cursar a disciplina de Qualidade de *Software*, que modificou muito a minha visão no desenvolvimento de produtos de *software*.

Ao Instituto de Física de São Carlos, por sediar e prover a infra-estrutura necessária para o desenvolvimento deste projeto de trabalho.

À Capes pelo incentivo através da bolsa que ajudou a financiar meus estudos.

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Abreviações

ASR	*Automatic Speech Recognition*.
DWT	*Discrete Wavelet Transform*.
DWTP	*Discrete Wavelet-Packet Transform*.
DTW	*Dynamic Time Warping*.
ECG	Eletrocardiograma.
EEG	Eletroencefalograma.
HMM	*Hidden Markov Model*.
LDC	*Linguistic Data Consortium*.
LTI	Linear e Invariante no Tempo.
MRA	Análise de Multi-Resolução.
QMF	*Quadrature Mirror Filters*.
WAV	*Waveform Audio Format*.

Resumo

Dynamic Time Warping (DTW) é uma técnica do tipo *pattern matching* para reconhecimento de padrões de voz, sendo baseada no alinhamento temporal de um sinal com os diversos modelos de referência. Uma desvantagem da DTW é o seu alto custo computacional. Este trabalho apresenta uma versão da DTW que, utilizando a Transformada *Wavelet* Discreta (DWT), reduz a sua complexidade. O desempenho obtido com a proposta foi muito promissor, ganhando em termos de velocidade de reconhecimento e recursos de memória consumidos, enquanto a precisão da DTW não é afetada. Os testes foram realizados com alguns fonemas extraídos da base de dados TIMIT do *Linguistic Data Consortium* (LDC).

Abstract

Dynamic Time Warping (DTW) is a pattern matching technique for speech recognition, that is based on a temporal alignment of the input signal with the template models. One drawback of this technique is its high computational cost. This work presents a modified version of the DTW, based on the Discrete Wavelet Transform (DWT), that reduces the complexity of the original algorithm. The performance obtained with the proposed algorithm is very promising, improving the recognition in terms of time and memory allocation, while the precision is not affected. Tests were performed with speech data collected from TIMIT corpus provided by Linguistic Data Consortium (LDC).

Capítulo 1

Introdução e Motivação

O reconhecimento automático de fala (*automatic speech recognition* - ASR) já tem sua importância e espaço garantidos no mundo de hoje, assim, diversas técnicas tem sido desenvolvidas e aprimoradas para a obtenção de melhores resultados com esta classe de algoritmos. Com a avanço no poder de processamento dos computadores e sistemas eletrônicos embarcados, o papel do ASR cresce a cada dia. A interação com computadores, eletrodomésticos e outros aparelhos, além das buscas automatizadas em listas de assinantes de sistemas telefônicos, são clássicos exemplos.

Basicamente, existem duas abordagens utilizadas para ASR [7]: técnicas do tipo *pattern matching* e técnicas do tipo *knowledge-based*. Dois algoritmos que implementam tais técnicas são, respectivamente, baseados em *Dynamic Time Warping* (DTW) e em *Hidden Markov Moddels* (HMMs). Neste trabalho, o interesse está restrito à técnica DTW, ou seja, na abordagem de *pattern matching*. Essa técnica compreende a identificação de uma palavra ou fonema baseada em uma biblioteca de modelos [9]. Para realização desta identificação, o algoritmo da DTW analisa completamente o sinal de entrada, comparando-o com todos os modelos disponíveis, conhecidos como *templates*. Assim sendo, caso os sinais tenham comprimentos consideravelmente grandes, o tempo de resposta da identificação fica comprometido devido ao acréscimo do custo computacional.

Tendo em vista as considerações mencionadas, a proposta do presente trabalho é justamente melhorar o desempenho da técnica de DTW para aplicações em ASR. Para isto, fez-se uso da Transformada Wavelet Discreta (*Discrete Wavelet Transform* - DWT) [24], sendo que os testes utilizaram fonemas extraídos da base de arquivos de vozes TIMIT [3]. Em particular, o trabalho mostra o ganho de desempenho na aplicação do algoritmo da DTW baseado na DWT, onde foram comparadas as diferentes famílias de filtros *wavelets* e como elas reagem a cada

tipo de fonema. O algoritmo desenvolvido da suporte ao projeto *SpeechAuth*, em andamento no laboratório *SpeechLab* [1] do Instituto de Física de São Carlos da Universidade de São Paulo (IFSC/USP), e financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) [2]. Além disso, as contribuições do presente trabalho são intercambiadas com o grupo de pesquisa em processamento de voz da *Microsoft Research* em *Redmond*, WA, USA e também do INESC, em Lisboa, Portugal.

Este trabalho está organizado da seguinte forma: o capítulo 2 apresenta uma revisão da literatura, envolvendo desde os princípios básicos de processamento de sinais de voz, até os conceitos-chave utilizados no presente trabalho, isto é, DTW e DWT. Já o capítulo 3 apresenta com detalhes o algoritmo proposto, sendo que os resultados obtidos com as diversas *wavelets*, baseados em inúmeros testes, estão descritos no capítulo 4. Tendo em vista os resultados, e apoiado nos conceitos e características estudadas durante a revisão de literatura, o capítulo 5 apresenta, de um ponto de vista teórico-prático, as conclusões. Por fim, 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 de programação de alto nível, assim como as publicações obtidas durante o curso de mestrado do autor.

[1] http://speechlab.ifsc.usp.br

[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 tópicos elementares de processamento de sinais de voz, além da DTW e da DWT. 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 suporte o entendimento dos sub-sequentes. As referências apresentadas fornecem explicações muito mais detalhadas.

2.1 Conceitos Elementares de Processamento de Sinais de Voz

Antes de apresentar os tópicos relacionados diretamente com o algoritmo proposto, que são a DTW e a DWT, esta seção descreve alguns conceitos básicos de processamento de sinais de voz que serão direta ou indiretamente utilizados neste trabalho.

2.1.1 Processamento analógico *X* digital de sinais de voz

No passado, todo o processamento de sinais era realizado unicamente de modo analógico, ou seja, baseada em tempo contínuo. Até hoje, ainda é viável realizar o processamento analógico de alguns sinais, entretanto, este não é o caso de um sistema de ASR, como o deste trabalho. Como diz o próprio nome, o sistema de processamento analógico de sinais faz uso de circuitos eletrônicos analógicos. Já os sistemas de processamento digital de sinais recorrem basicamente ao uso de um processador capaz de realizar operações aritméticas [11].

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 relativamente altas. Em contrapartida, a abordagem digital possui três vantagens principais:

- Flexibilidade: obviamente, quando se fala em processadores e memórias, ou seja, computadores, tem se em mente que todo o processamento pode ser controlado via *software* e que, portanto, qualquer modificação que se queira fazer no processamento não implica em utilizar ferramentas para manipular valores de componentes nos circuitos eletrônicos.

- 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 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 de voz. Os únicos erros resultantes do processamento digital são devidos à precisão na representação numérica.

- Repetitividade: uma operação de processamento digital de sinais pode ser repetida diversas vezes de maneira exata enquanto que nos sistemas analógicos, podem existir variações devido à falta de imunidade.

O algoritmo proposto no presente trabalho se beneficia de todas as características acima, tendo em vista que a abordagem utilizada será o processamento *digital* de sinais de voz.

2.1.2 Sinais e sistemas para filtragem digital

Os sinais estão presentes em todos os momentos de nossas vidas. Os sons ouvidos e interpretados todos os instantes são meramente alguns exemplos. Na prática, enumerar tudo aquilo que constitui um sinal é tarefa impossível. De acordo com [13] [22], 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 casos onde um sinal interpretado, existe sempre, de maneira implícita, um sistema associado. Por exemplo, o sistema associado à interpretação das vozes humanas faz uso de um mecanismo formado pelo ouvido, cérebro e demais órgãos para sua interpretação.

Para sinais discretos de amplitude e duração finitos, um conceito básico é a sua

energia, $E(x[\quad])$, que é dada por $E = \sum_{i=0}^{N-1} {x_i}^2$. Por sua vez, um sinal sofre uma sub-amostragem (*downsampling*) por K toda vez que uma determinada amostra é considerada e as K, ($K \in \mathbb{Z}$), seguintes são descartadas, e assim por diante até o final do sinal. Diz-se 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 ($\uparrow_K$) e ($\downarrow_K$). Finalmente, um sinal caracterizado como estacionário é aquele que mantém constante sua frequência ao longo de todo o inervalo de tempo considerado. No presente trabalho estamos lidando com sinais de vozes não estacionários, sendo utilizado o cálculo de energia para examinar algumas propriedades destes sinais, e também serão utilizados *downsamplings* 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 ASR proposto neste trabalho recebe como entrada um sinal de voz digital de determinado locutor e compara-o com diversos *templates*, produzindo uma saída que corresponde a identificação do fonema de entrada. Quando um sistema obedece as condições: *i)* o deslocamento da entrada por uma determinada constante de tempo implica no mesmo deslocamento na saída; *ii)* a multiplicação da entrada do sistema por uma constante implica na saída multiplicada pela mesma constante; *iii)* a soma de dois ou mais sinais na entrada de um sistema implica na soma das saídas individuais ; dizemos que o sistema é Linear e Invariante no Tempo (LTI).

O sistema de interesse no presente trabalho efetua, além de outras tarefas, filtragem digital de sinais de voz. Um filtro digital [18] 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:

- frequência de corte: é a frequência para a qual o filtro já tem 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.

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

- banda de passagem: é a faixa de frequências anterior a frequência de corte.
- banda de transição: é a faixa de frequências que 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 [18].
- 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) [18].

- ordem: número de pólos da função de transferência do filtro, conforme descrito adiante. Um filtro digital com $N + 1$ coeficientes possui ordem N. A 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.

2.1.3 Teorema da convolução

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

$$y[\circ] = x[\circ] * h[\circ] = \sum_{k=0}^{M-1} h_k x_{n-k} \qquad , \tag{2.1}$$

onde M é o número de amostras de $h[\ \ ]$. No presente trabalho, os processos de filtragem são realizados por intermédio da convolução.

2.1.4 Resposta ao impulso de um filtro digital

A resposta ao impulso é a resposta do filtro para uma entrada impulsiva $\delta[\circ] = \{1, 0, 0, 0, ..., 0\}$, que corresponde aos coeficientes do filtro digital no domínio do tempo.

2.1.5 Teorema da amostragem e *aliasing*

Também conhecido como teorema de Nyquist [18], 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*.

2.1.6 Análise em frequência: Transformada Discreta de Fourier e Transformada Z

A DFT - *Discrete Fourier Transform* [10] 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 $x[\ \ ]$ é o sinal no domínio do tempo, $X[\ \ ]$ é o correspondente no domínio da frequência, e N é o comprimento dos sinais.

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

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

Existem algoritmos mais eficientes para o cômputo da DFT e IDFT, conhecidos como algoritmos de transformada rápida de Fourier (FFT - *Fast Fourier Transform*) [5], que reduzem a odem 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.

Uma outra ferramenta utilizada no presente trabalho é a Transformada Z [18] (TZ), que converte um sinal do domínio do tempo para o domínio z, z representando a frequência:

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

A TZ foi utilizada, em *background*, nesta dissertação para expressar a função de transferência dos filtros.

2.1.7 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 a função 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.

2.1.8 A Transformada de Fourier de Tempo Reduzido (STFT)

Uma modificação na DFT de um sinal corresponde a *Shortest Time Fourier Transform* (STFT) [19], que supõe que um dado sinal não estacionário, quando dividido em pequenas partes, 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 aos utilizados na DFT.

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

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 na frequência, enquanto que janelas largas resultam numa resolução melhor na 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 [1] [14] consiste numa alternativa mais eficiente do que a STFT para realizar a análise tempo-frequência de um sinal [21] [23], bem como a filtragem e separação em sub-bandas de frequências. A DWT, que é objeto fundamental do presente trabalho, age na verdade de 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 este 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 deste 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 recursivo de decomposição.

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 este processo se encontra 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 $\omega = \pi$.

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:

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

ou, mais especificamente:

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

$$y_{passa-altas}[\circ] = x[\circ] * g[\circ] = \sum_{k=0}^{n-1} g_k x_{2n-k} \qquad , \tag{2.8}$$

onde $h[\circ]$ e $g[\circ]$ são os filtros passa-baixas e passa-altas, respectivamente.

A DWT está diretamente relacionada com a análise de multi-resolução (MRA), proposta por Mallat, Meyer, Stromberg e outros [2] [24], que consiste em decompor um vetor (sinal sob análise) $\vec{f}$ em uma soma de outros vetores pertencentes a uma sequência de sub-espaços vetoriais [16]. 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:

$$\vec{f} = \vec{A} + \vec{D} \tag{2.9}$$

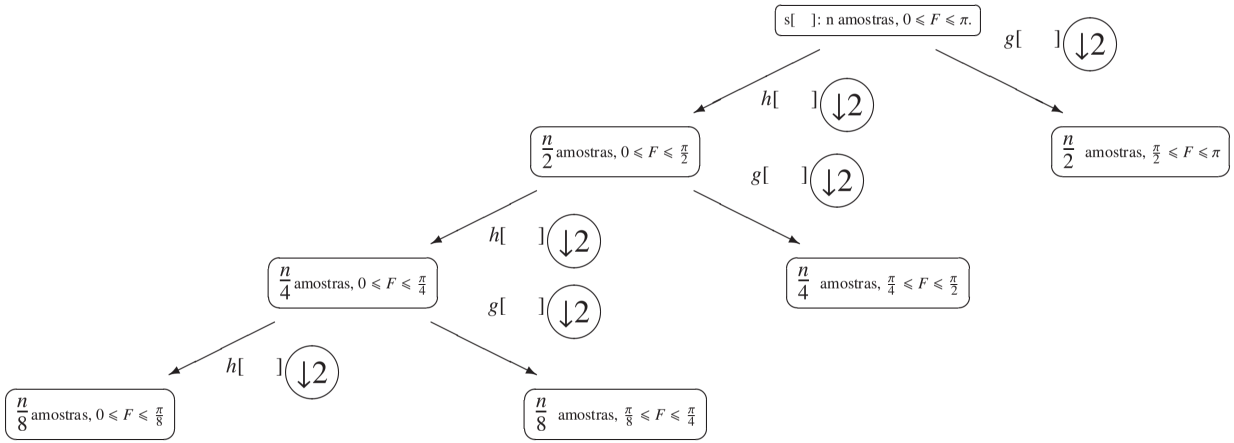

Figura 2.2: Funcionamento da DWT, exemplificado para um sinal $s[\ ]$ de n amostras discretas e máxima frequência π, decomposto até o terceiro nível. Note o espectro de frequências e a quantidade de amostras presentes em cada sub-banda.

onde

$$\vec{A} = \sum_{k=0}^{\frac{n}{2}-1} < \vec{f}, \vec{v_k} > \vec{v_k} \qquad\qquad \vec{D} = \sum_{k=0}^{\frac{n}{2}-1} < \vec{f}, \vec{w_k} > \vec{w_k}.$$

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;
- $V \perp W \leftrightarrow \vec{A} \perp \vec{D}$;
- $\vec{v_i} \perp \vec{w_i} \leftrightarrow < \vec{v_i}, \vec{w_i} >= 0$.

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. Este processo pode ser repetido, $\frac{log(n)}{log(2)}$ vezes, conforme já foi mencionado. Dessa forma, generalizando, para uma decomposição de nível j, temos:

$$\vec{f} = \vec{A_j} + \sum_{i=1}^{j} \vec{D_i}. \tag{2.10}$$

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-espaço W_i, com uma base contendo $\frac{n}{2^i}$ vetores;
- $V_j \perp W_j \leftrightarrow \vec{A_j} \perp \vec{D_j}$;
- $\vec{v_{i,j}} \perp \vec{w_{i,j}} \leftrightarrow < \vec{v_{i,j}}, \vec{w_{i,j}} >= 0$.

Este processo acima equivale a [21]

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

onde

- $\phi[n]$ e $\psi[n]$ formam uma base de Riesz [21] 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* [21];

- $\psi[n] = \sum_k g_n \phi[2n - k]$, também definida recursivamente, é chamada função *wavelet* e é ortogonal a função *scaling*;
- $H_{j,k}[n] =< f, \phi_{j,k}[n] >$;
- $G_{t,k}[n] =< f, \psi_{t,k}[n] >$;
- $\{0\} \leftarrow ... \subset V_{-1} \subset V_0 \subset V_1 \subset ... \rightarrow L^2$;
- 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 $g[\ \ ]$, possuem uma única função *scaling* (ϕ) e uma única função *wavelet* (ψ) associadas. A forma de obtenção destas funções a partir dos filtros, e vice-versa, está documentada com detalhes em [14] [23], não sendo apresentada aqui por estar fora dos escopo do trabalho.

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

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

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

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

Quando $h[\ \ ]$, $g[\ \ ]$, $\bar{h}[\ \ ]$, e $\bar{g}[\ \ ]$ mantém as relações acima, eles constituem um banco de filtros de reconstrução perfeita (*perfect reconstruction filter bank* - PRFB) [2] [24], 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 a inversão da DWT não se faz necessária, entretanto, é desejável que o algoritmo proposto utilize apenas PRFBs. Isso se deve ao fato de que o

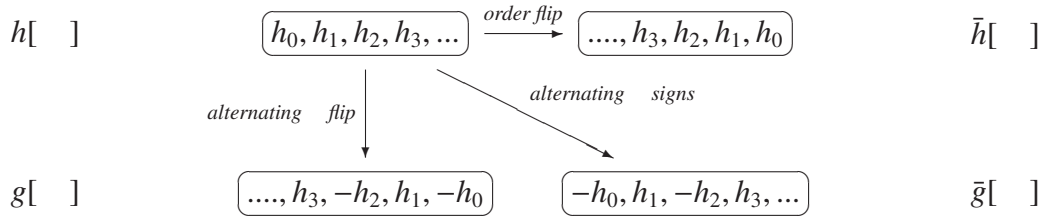

Figura 2.3: Relação entre os filtros de análise e síntese.

algoritmo pode ser embutido em um sistema de ASR maior, com funções mais diversificadas, que necessite em algum momento da inversão da DWT.

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

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

2.2.1 Cálculo da DWT

Para o cálculo da DWT de um sinal, aplica-se o algoritmo de Mallat, que está minunciosamente descrito em [14] [23]. É importante observar que apenas os filtros $h[\ ]$ e $g[\ ]$ 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 $A[\ ][\ ]$ é a matriz de coeficientes dos filtros e $B[\ ]$ é o sinal original, então $C[\circ] = A[\ ][\ ]B[\ ]$ corresponde ao sinal transformado, sendo que a disposição dos coeficientes nas matrizes é a seguinte:

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

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

Pode-se notar na matriz $A[\ ][\ ]$ que dois procedimentos estão embutidos no algoritmo de Mallat: *downsampling* e *wrap-arroud* [14] [23]. O primeiro já foi

mencionado anteriormente, e o segundo consiste em fazer com que os últimos coeficientes dos filtros ocupem as posições inciais 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 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 coefficientes dos filtros de síntese, isto é, $\bar{h}[\ ]$ e $\bar{g}[\ ]$. Tendo em vista que $A[\ ][\ ]$ é ortogonal, $A^{-1}[\ ][\ ] = A^T[\ ][\ ]$, o que facilita muito a inversão da transformada

2.2.3 Momentos Nulos

A quantidade de momentos nulos [14] [23] 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 este fato seja primordialmente importante nos esquemas de compressão de dados, ele pode ser levado em conta no presente trabalho por ter ligação com as características dos filtros. O m-ésimo momento pode ser calculado como $m = \sum_{k=0}^{p-1} t_k{}^m \psi(t_k)$, 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 ($t = \frac{1}{2}s$, onde s é um escalar inteiro maior ou igual a 0).

2.2.4 Famílias de Transformadas *Wavelet*

As diversas famílias de filtros *existentes* [14] [23] diferem no suporte dos filtros, assim como nas características de resposta em frequência e fase dos mesmos, 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 na figura 2.4.

Reunindo os conceitos apresentados até agora, o próximo capítulo descreve com-

Figura 2.4: PRIMEIRA LINHA: Formato das respostas ao impulso dos filtros *wavelet*, para diversos suportes. Da esquerda para direita: Haar, Daubechies, Vaidyanathan, Beylkin, Coiflet, and Symmlet ; SEGUNDA LINHA: Formatos das funções *scaling* dos filtros *wavelet*. Da esquerda para direita: Haar, Daubechies, Vaidyanathan, Beylkin, Coiflet, and Symmlet ; TERCEIRA LINHA: Formatos das funções *wavelet* dos filtros *wavelet*. Da esquerda para direita: Haar, Daubechies, Vaidyanathan, Beylkin, Coiflet, and Symmlet.

Família	Suporte(n)	Fase	Observação	Momentos
Haar	2	linear	é a mais simples das *wavelets*, criada por Alfred Haar [23][14]	1
Daubechies	par, maior que 4	não linear	resposta ao impulso *maximally flat*, criada por Ingrid Daubechies [23][14]	$\frac{n}{2}$
Symmlets	par, múltiplo de 8	não linear	resposta ao impulso mais simétrica[23][14]	$\frac{n}{2} - 2$
Coiflets	par, múltiplo de 6	quase linear	resposta ao impulso quase simétrica, criada por Ronald Coifman [23][14]	$\frac{n}{2} - 1$
Vaidyanathan	24	não linear	otimizada para voz, criada por P. P. Vaidyanathan [23][14]	–
Beylkin	18	não linear	otimizada para áudio em geral [23][14]	$\frac{n}{2} - 2$

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

pletamente o sistema proposto para melhoria da eficiência da DTW através do uso da DWT.

2.3 *Reconhecimento Biológico e Computacional de Voz*

2.3.1 Estudo da fala humana

A produção de voz pelo corpo humano, embora seja um mecanismo repleto de detalhes, consite basicamente da propulsão de ar pelos pulmões, seguida de um processo de filtragem, realizado pelo trato vocal e elementos associados, como ilustram as figuras 2.5 e 2.6. O primeiro detalhe que deve ser observado na figura 2.5 é que as pregas 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, este 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

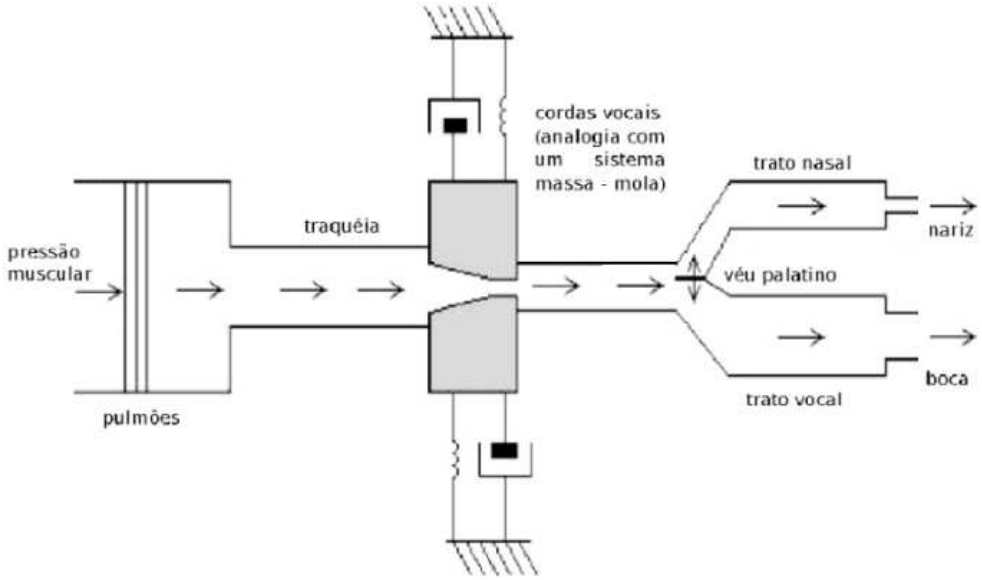

Figura 2.5: Interpretação física simplificada do sistema bio-gerador de voz [19].

trato vocal ou nasal, e os próprios tratos vocal e nasal, pode-se ainda refinar essa classificação dos sinais de voz da seguinte forma [9]:

- *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 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 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 fricação.
- *voiced plosives*: são fonemas *voiced*, ou seja de excitação periódica, porém misturado com ruído impulsivo criado no trato vocal.

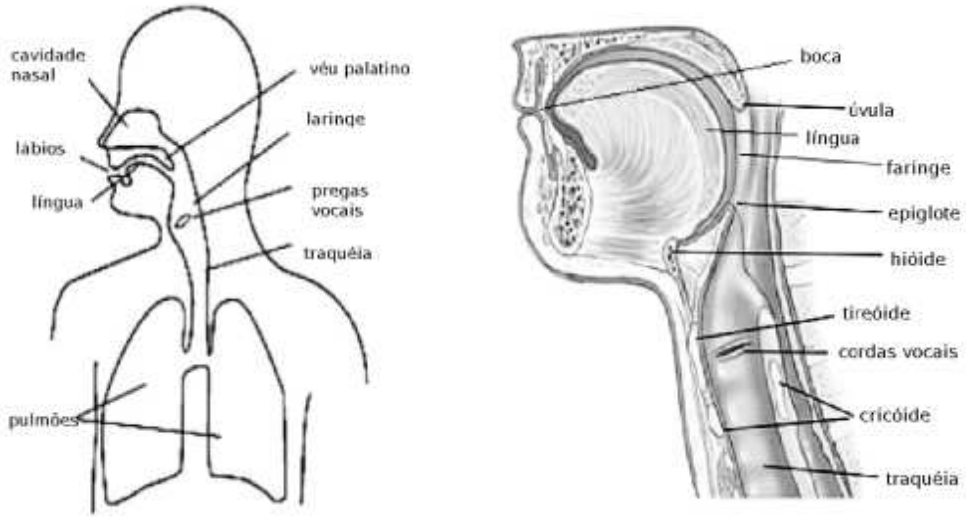

Figura 2.6: [esquerda]: visão básica do sistema de produção de voz humana; [direita]: detalhe do trato vocal humano e suas sub-partes [19].

- *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. No presente trabalho, a análise está mais focada em trechos de *voiced speech*.

2.3.2 O Sistema bio-físico de interpretação de fala

O ouvido humano percebe os sons através de um mecanismo bastante elaborado [9]. A figura 2.7 exibe um diagrama simplificado deste 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 fluído que é levada à 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 impulso são interpretados pelo cérebro. A figura 2.8 dá uma idéia das partes da membrana que são sensíveis a determinadas frequências.

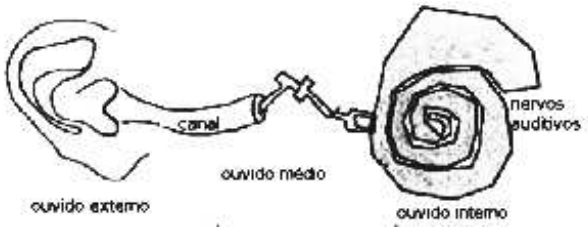

Figura 2.7: Ouvido humano: parte externa, média e interna [4]-p.168.

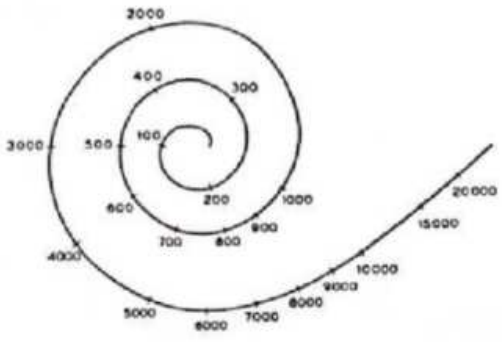

Figura 2.8: Membrana basilar e as frequências captadas em Hertz (Hz) [4]-p.173.

2.3.3 Reconhecimento de fala por intermédio computacional

De acordo com [9] [22], o reconhecimento automático de fala por intermédio computacional, ASR, é uma tarefa bastante complexa quando o vocabulário se torna grande. Como já foi observado anteriormente, o ASR pode ser baseado em duas abordagens: *knowledge-based* e *pattern-matching*. O primeiro modelo, em geral, utiliza os Modelos Ocultos de Markov (*Hidden Markov Models* - HMMs) ou redes neurais artificiais, que são métodos que se apoiam em hipóteses probabilísticas controladas por máquinas de estado finito, incorporando critérios de evolução temporal da fala, assim como estatísticas dos dados de treinamento, quando for o caso. Esta classe de métodos é a mais utilizada atualmente para grandes vocabulários, tendo em vista a sua capacidade de tratar bem a grande diversidade de variações existentes nos modelos de fala humana, com grande precisão.

Já o segundo modelo, de caráter determinístico, é baseado geralmente em uma técnica conhecida como *Dynamic Time Warping* - DTW, e é empregado quando o vocabulário a ser reconhecido é mais restrito, principalmente baseado em *voiced speech*. Esta abordagem, que pode trabalhar bem com sinais e *templates* de tamanhos variados, tem a desvantagem de que o custo computacional pode crescer bastante a medida que o comprimento dos sinais aumenta, aliás o presente tra-

balho propõe uma alternativa para redução deste custo computacional sem afetar consideravelmente a precisão original do método. A próxima seção descreve com detalhes a técnica de DTW.

2.4 *Dynamic Time Warping* (DTW)

A técnica de DTW, que é do tipo *pattern matching*, pode ser utilizada em aplicações de reconhecimento de voz com vocabulário restrito [19]. Um exemplo de sua utilização pode ser visto na tarefa de reconhecer qual dos números, de 0 até 9, foi pronunciado por determinado locutor. O sinal de entrada, que corresponde ao número ditado, pode ser comparado com cada *template* de uma biblioteca, sendo que a que mais se assemelha ao sinal de entrada será considerada o número pronunciado. Umas das características da DTW é que o sinal de entrada pode ter um comprimento diferente dos *templates*. Isso faz com que um mesmo sinal de entrada possa ser reconhecido mesmo quando ocupar um período de tempo diferente, ou seja, se for pronunciado mais rápido ou mais devagar. O algoritmo DTW se propõe a encontrar o melhor caminho, w, através de uma tabela de associação de índices, conhecida como *best path*, construída pela delimitação do fonema [17].

2.4.1 O algoritmo DTW e um exemplo prático

Assumindo que as amostras discretas do sinal de entrada, $x[\ \ ]$, e de um *template*, $y[\ \ ]$, são conhecidas, deve-se proceder como segue:

- **INÍCIO**
- **PASSO 1**: formar a matriz solução, com n linhas e m colunas, onde cada elemento da linha i e coluna j corresponde ao módulo da diferença entre cada ponto do sinal de entrada (*input*) com o *template*, sendo que n representa o comprimento do *input* e m representa o comprimento do *template*.
- **PASSO 2**: formar a matriz de distância acumulada (DA). Esta matriz é formada pela iteração na soma dos valores de cada elemento dela mesma com o elemento superior da matriz solução, conforme a equação 2.17.

$$DA_{i,j} = DA_{i-1,j} + S_{i,j} \quad , \quad i > 1 \quad , \quad j > 1 \qquad . \tag{2.17}$$

- **PASSO 3**: formar a matriz de movimento. A matriz de movimento deve ter o último ítem da primeira coluna igual a 0. Deve ser realizada uma iteração no sentido de baixo para cima, analisando qual valor é menor na matriz DA. Caso o menor valor seja o elemento abaixo, a matriz movimento deve

ser preenchida com o valor 1; caso na matriz DA o menor elemento seja o elemento imediatamente à esquerda, deve-se preencher a matriz movimento com o valor 3. Caso o menor valor seja a diagonal inferior à esquerda ou os valores forem iguais, deve-se colocar o valor 2.

- **PASSO 4**: formar a matriz *best path*, w, ou seja, melhor caminho. Para isso, observando a matriz movimento, partindo do último elemento da primeira linha, escolhe-se o próximo elemento com a menor distância, *md*, dos valores dos elementos [19], como demonstrado na equação 2.18.

$$md = min\{|w_{i,j} - w_{i-1,j}|, |w_{i,j} - w_{i,j-1}|, |w_{i,j} - w_{i-1,j-1}|\} \quad , \quad i > 1 \quad , \quad j > 1 \quad . \tag{2.18}$$

  Desta forma, a cada elemento escolhido cria-se uma marca na matriz *best path*, até que se alcance o último elemento da primeira coluna.

- **FIM.**

Baseado na DTW, a aplicação da DTW em um sinal de voz armazenado em um arquivo digital de voz consistiria, como na figura 2.14, em extrair as amostras relativas ao(s) fonema(s) *template*(s) e o fonema de entrada, analisando-as através da matriz solução e, recursivamente, através da matriz *best path*.

Para exemplificar, assumindo $x[\circ] = \{1, 1, 2, 3, 2, 0\}$ e $y[\circ] = \{0, 1, 1, 2, 3, 2, 1\}$, tem-se a matriz solução da figura 2.9, referente ao *passo 1* do algoritmo anterior. Executando o *passo 2*, obtém-se a matriz DA. Após preencher a primeira coluna de DA, assume-se que $DA_{i,j} = DA_{i,j} + S_{i,j+1}$, sendo $j <$ quantidade de colunas. A matriz DA deve então ser inicializada com o valor 1 no último elemento da primeira coluna, como na figura 2.10. Primeiramente, devem ser preenchidos os valores das colunas, do último elemento para o primeiro. Em seguida, devem ser calculados os elementos da esquerda para a direita, como nas setas indicativas da figura 2.10. Quando a matriz DA estiver finalizada, o próximo passo é criar a matriz de movimento, de acordo com a figura 2.11 e o *passo 3*. Finalmente, após completada a matriz movimento 2.12, deve-se executar o *passo 4*, obtendo o resultado mostrado na figura 2.13. A matriz *best path* para o sinal $x[\quad]$ comparado ao modelo $y[\quad]$, tem distância igual a 7.

1	0	0	1	2	1	1
2	1	1	0	1	0	2
3	2	2	1	0	1	3
2	1	1	0	1	0	2
1	0	0	1	2	1	1
1	0	0	1	2	1	1
0	1	1	2	3	2	0
	1	1	2	3	2	0

Figura 2.9: Matriz solução para o exemplo dado.

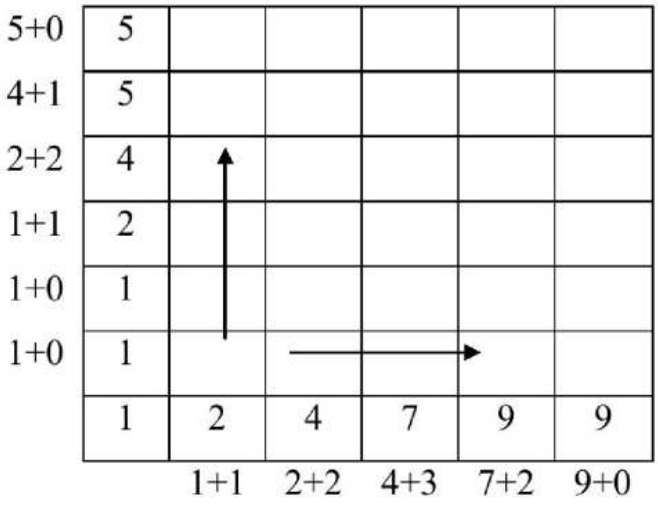

Figura 2.10: Primeira iteração na matriz de distância acumulada.

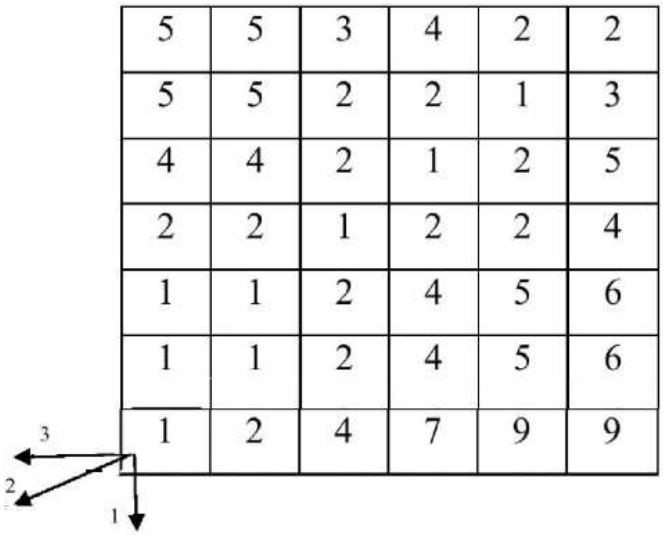

Figura 2.11: Matriz de distância acumulada completa.

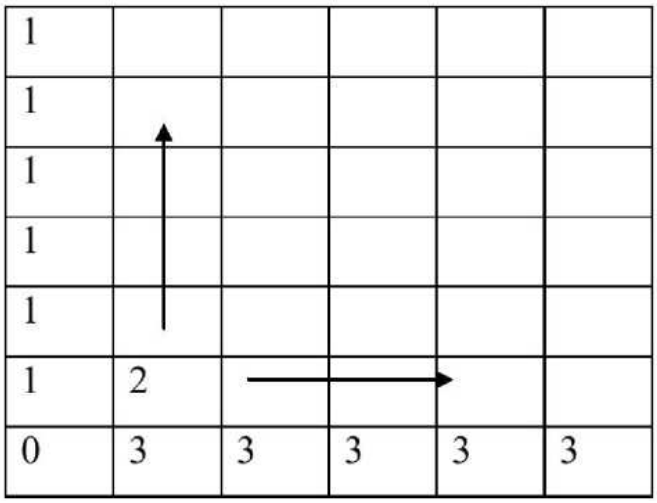

Figura 2.12: Matriz movimento.

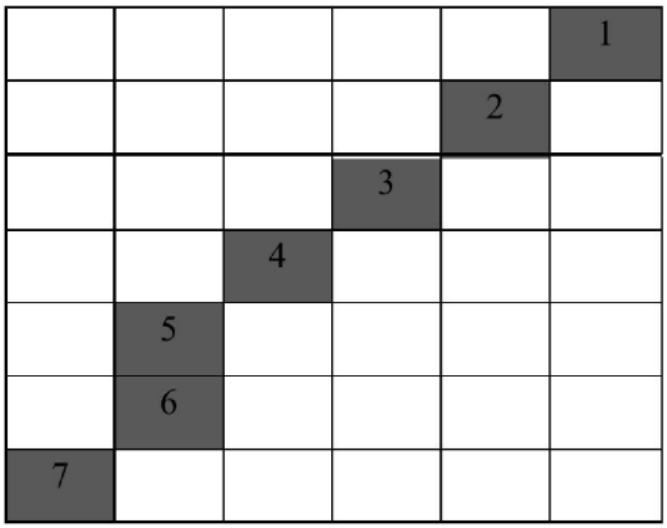

Figura 2.13: Matriz *best path*.

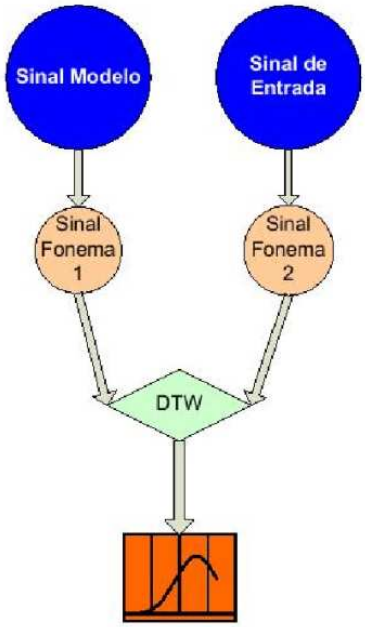

Figura 2.14: Diagrama de análise do algoritmo original da DTW.

Capítulo 3

Descrição do Sistema Proposto

Este capítulo descreve o sistema proposto, incluindo cada passo do algoritmo, além de detalhes sobre a implementação.

3.1 A Arquitetura e o algoritmo do sistema

A arquitetura do sistema proposto encontra-se na figura 3.1, sendo que o algoritmo detalhado e os comentários adicionais seguem.

- **INÍCIO**
- **PASSO 1**: definir o arquivo de voz de entrada (*input*), assim como cada um dos *templates*;
- **PASSO 2**: janelar o *input* e cada *template* de forma que eles possuam um tamanho igual a uma potência de 2, lembrando que os tamanhos podem ser diferentes entre si;
- **PASSO 3**: aplicar a DWT nível j no *input* e nos *templates*, utilizando a família de filtros *wavelet* f, onde j e f serão discutidos adiante;
- **PASSO 4**: considerar apenas a sub-banda s de cada um dos sinais transformados (*input* e *templates*), onde s será discutida adiante. A sub-banda s do *input* fica doravante denominada *s-input* e a sub-banda s de cada um dos n templates fica da mesma forma denominada *s-template-1*, *s-template-2*, ..., *s-template-n*;
- **PASSO 5**: os sinais *s-input* e *s-template-1*, *s-template-2*, ..., *s-template-n* passam a ser, respectivamente, o novo *input* e os novos *templates* a serem utilizados;

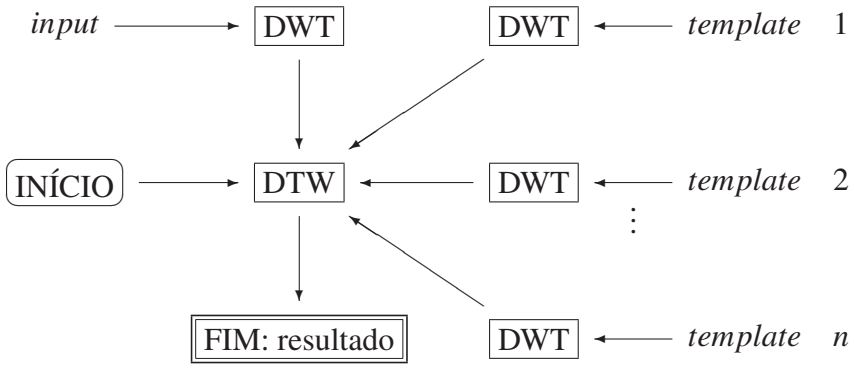

Figura 3.1: Arquitetura básica do sistema proposto (DTW modificado).

- **PASSO 6**: aplicar o algoritmo DTW nos sinais anteriores, obtendo a resposta que consiste no *template* mais semelhante com o sinal *input*.
- **FIM.**

A idéia básica do algoritmo proposto consiste em reduzir a complexidade computacional da técnica de DTW por intermédio da decomposição do *input* e dos *templates* em sub-bandas *wavelet*, utilizando apenas um número reduzido de sub-bandas específicas de cada um dos sinais como entrada para o algoritmo DTW. Particularmente, para a classe de sinais de interesse no presente trabalho, uma única sub-banda de cada sinal mostrou-se suficiente, em particular a aproxima£o do último nível. Este fato sem dúvida implica na redução do custo computacional da DTW original, conforme descrito a seguir.

3.2 Custo computacional

O algoritmo original da DTW, descrito no capítulo anterior, utiliza quatro matrizes para comparar o *input* com cada um dos *templates*. Se o *input* possui comprimento c e existem n *templates* de comprimentos t_1, t_2, ..., t_n, então são necessárias:

- quatro matrizes de dimensões c x t_1,
- quatro matrizes de dimensões c x t_2,
- . . . ,
- quatro matrizes de dimensões c x t_n,

o que totaliza $4 \sum_{i=1}^{n} ct_i$ elementos. Já no algoritmo proposto, se a decomposição *wavelet* for realizada em nível 1, este montante fica reduzido ao meio, se for em nível 2 fica reduzido 4 vezes, se for em nível 3 fica reduzido 8 vezes, e assim por diante, considerando o uso de uma única sub-banda, ou seja, $\frac{4}{2^{2j}} \sum_{i=1}^{n} ct_i$ elementos. O nível *j* ideal varia e foi determinado de maneira teórico-experimental, de acordo com o próximo capítulo.

3.3 Implementação do algoritmo

O algoritmo proposto foi implementado em linguagem C/C++ [20] em ambiente Linux e a análise posterior dos resultados foi realizada através de visualização gráfica, utilizando um *software* desenvolvido em linguagem Java [8] para ambientes Windows / Linux, ambos constantes do apêndice II do presente trabalho.

Particularmente, um modelo padrão de arquivo em formato de texto puro é enviado ao *software* para análise. Neste modelo estão as informações do *input*, dos *templates*, qual é o fonema sendo analisado, além de outras especificações. A informação de qual sinal (*input* ou *template*) seria utilizado foi passada como localização física do arquivo em disco, arquivo este em formato WAV (*Waveform Audio Format*) [4]. O *software* importa o arquivo do sinal modelo, extraindo apenas os dados brutos do fonema em questão, ou seja, a parte que contém o valor das amostras efetivamente e, em seguida, aplica a DWT sobre os dados do fonema extraído. O mesmo ocorre para o arquivo do sinal de entrada (*input*). Após extraídas e filtradas, as novas amostras são passadas para o algoritmo DTW, que realiza os cálculos desejados. A saída do algoritmo da DTW é então gravada em um arquivo, sendo o conteúdo relativo às informações da filtragem aplicada, seguido da matriz *best path* da análise.

O próximo capítulo descreve os testes e resultados obtidos com o algoritmo proposto e sua implementação correspondente.

Capítulo 4

Testes e Resultados

Neste capítulo encontram-se os testes realizados e os resultados obtidos em cada uma das experimentações. Diversas experimentações foram realizadas, incluindo variações da base wavelet, variações no suporte dos filtros, nos tipos de fonemas, entre outros fatores.

4.0.1 Materiais e Métodos

Todos os sinais analisados foram fonemas *voiced* (vogais ou semi-vogais), extraídos da base TIMIT do LDC, e foram amostrados em uma taxa de 16000 amostras por segundo, 16 bits, WAV. As comparações realizadas pelo algoritmo mantinham ambos os sinais, *input* e *template*, sob as mesmas condições, por exemplo, se o *template* sofria uma filtragem com a *wavelet* da família Daubechies de suporte quatorze até o nível oito, o mesmo procedimento foi aplicado no sinal *input*. O *software* de análise descrito anteriormente recebeu como entrada a listagem de fonemas a serem analisados e as análises eram realizadas em *batch*, sendo que a matriz *best path* de cada comparação corresponde a saída desejada. Posteriormente, para uma análise mais clara, as matrizes *best path* obtidas foram importadas pelo *software* desenvolvido para exibição dos resultados em um gráfico cartesiano.

A bateria completa dos testes realizados segue, incluindo uma descrição sobre os fonemas, presente nas tabelas 4.1 até 4.4, assim como os resultados obtidos, presentes nas demais tabelas e gráficos, sendo que a discussão e interpretação dos dados, que levam as conclusões finais, encontram-se no próximo capítulo.

Fonema	Posição Inicial	Posição Final	Tipo	Exemplo
iy	11240	12783	vogal	b**ee**t
ae	14078	16157	vogal	b**a**t
y	17103	17587	semi-vogal	**y**acht
aa	19692	21514	vogal	b**o**t
ih	29179	30337	vogal	b**i**t
ao	52378	54500	vogal	b**ou**ght

Tabela 4.1: Fonemas do arquivo **sa1.wav**, referente a sentença *She had your dark suit in greasy wash water all year* da base TIMIT.

Fonema	Posição Inicial	Posição Final	Tipo	Exemplo
ow	13419	15093	vogal	b**oa**t
oy	33149	36133	vogal	b**oy**

Tabela 4.2: Fonemas do arquivo **si573.wav**, referente a sentença *His captain was thin and haggard and his beautiful boots were worn and shabby* da base TIMIT.

Fonema	Posição Inicial	Posição Final	Tipo	Exemplo
ah	13727	15160	vogal	b**u**t
ax	35028	35718	vogal	**a**bout
ix	44930	45605	vogal	**d**ebit
ey	47980	49866	vogal	b**ai**t

Tabela 4.3: Fonemas do arquivo **si943.wav**, referente a sentença *Production may fall far below expectations* da base TIMIT.

Fonema	Posição Inicial	Posição Final	Tipo	Exemplo
er	35410	37240	vogal	b**ir**d
ux	45736	46520	vogal	t**oo**t
uw	52700	54440	vogal	b**oo**t

Tabela 4.4: Fonemas do arquivo **sa1.wav**, referente a sentença *She had your dark suit in greasy wash water all year* da base TIMIT.

4.0.2 Bateria de Testes 1

A primeira bateria de testes realiza uma comparação de um fonema original (*input*) com diversos outros, aplicando o algoritmo original da DTW e, posteriormente, aplicando o algoritmo proposto, com as decomposições em níveis 1 até 8. As figuras que ilustram estes testes são 4.1 até 4.10.

4.0.3 Bateria de Testes 2

A segunda bateria de testes compara dois fonemas diferentes, porém nos níveis de um até oito, com a mesma base *wavelet*. Estes testes correspondem às figuras de 4.11 até 4.17.

4.0.4 Bateria de Testes 3

O terceiro conjunto de testes visa comparar dois fonemas diferentes, porém em um mesmo nível, com diferentes filtros *wavelets*. As figuras 4.18 até 4.25 documentam estes testes.

4.0.5 Bateria de Testes 4

Finalmente, o último grupo de testes compara dois fonemas diferentes, com grau de diferença considerável, porém em um mesmo nível, com diferentes filtros *wavelets*. A figura 4.26 documenta este teste.

Testes adicionais foram realizados com alguns outros fonemas e suportes de filtros, entretanto os resultados seguem os mesmos padrões visualizados com nas baterias de testes descritas, além disso o enorme espaço exigido para a inclusão de tais testes com resultados repetitivos levou a não inclusão dos mesmos aqui.

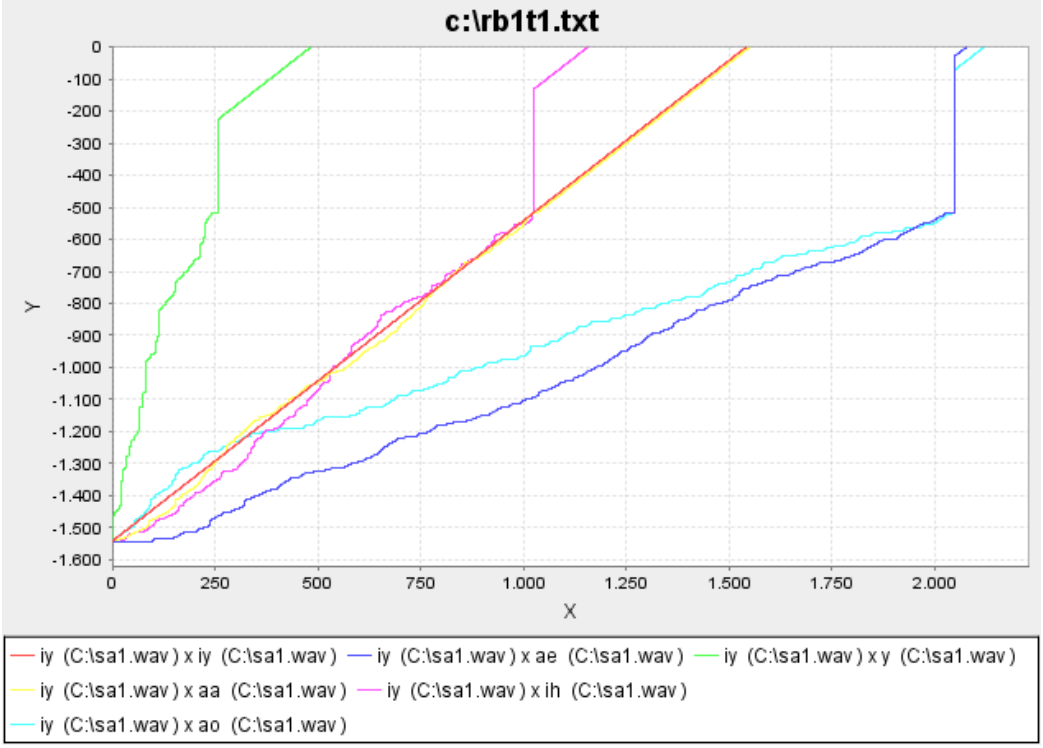

Figura 4.1: Resultados com **algoritmo DTW original**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMAS DE TESTE: **ae** (quantidade de pontos: 2079, tipo: vogal) ; **y** (quantidade de pontos: 484, tipo: semi-vogal) ; **aa** (quantidade de pontos: 1552, tipo: vogal) ; **ih** (quantidade de pontos: 1158, tipo: vogal) ; **ao** (quantidade de pontos: 2122, tipo: vogal) ; **iy** (quantidade de pontos: 1543, tipo: vogal).

ARQUIVO *TEMPLATE*:
#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#iy#11240#12783#
#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#y#17103#17587#
#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ih#29179#30337#
#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ao#52378#54500#

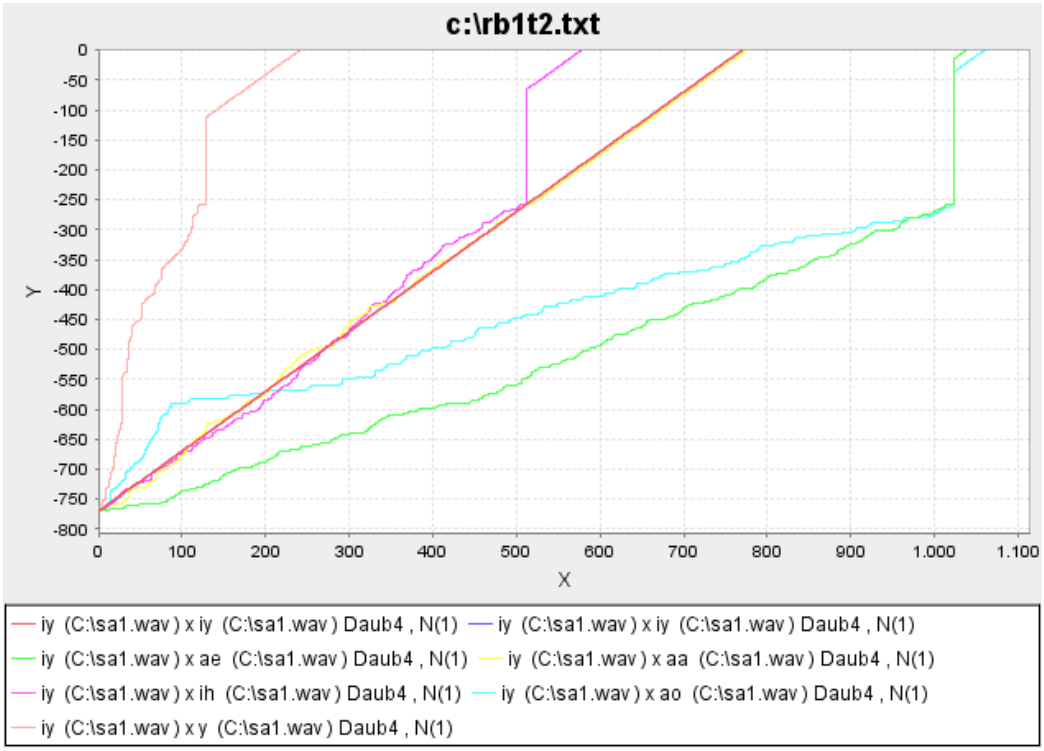

Figura 4.2: Resultados com **algoritmo DTW proposto usado Daubechies 4 nível 1**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMAS DE TESTE: **ae** (quantidade de pontos: 2079, tipo: vogal) ; **y** (quantidade de pontos: 484, tipo: semi-vogal) ; **aa** (quantidade de pontos: 1552, tipo: vogal) ; **ih** (quantidade de pontos: 1158, tipo: vogal) ; **ao** (quantidade de pontos: 2122, tipo: vogal) ; **iy** (quantidade de pontos: 1543, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#iy#11240#12783#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#y#17103#17587#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ih#29179#30337#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ao#52378#54500#

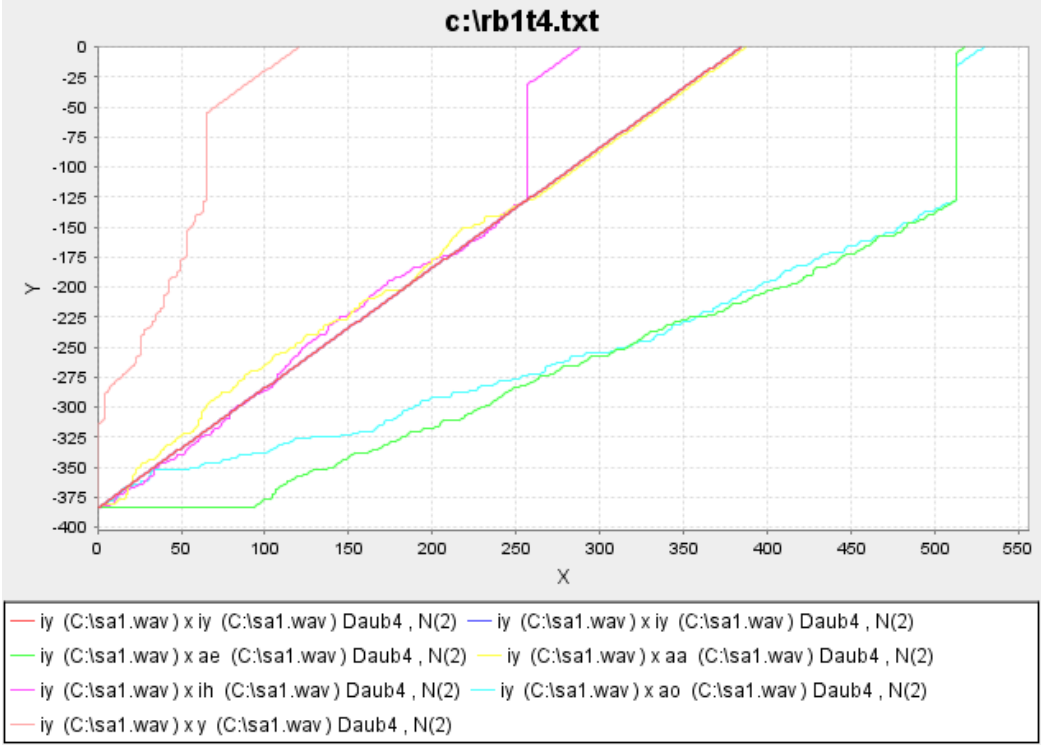

Figura 4.3: Resultados com **algoritmo DTW proposto usado Daubechies 4 nível 2**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMAS DE TESTE: **ae** (quantidade de pontos: 2079, tipo: vogal) ; **y** (quantidade de pontos: 484, tipo: semi-vogal) ; **aa** (quantidade de pontos: 1552, tipo: vogal) ; **ih** (quantidade de pontos: 1158, tipo: vogal) ; **ao** (quantidade de pontos: 2122, tipo: vogal) ; **iy** (quantidade de pontos: 1543, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub4#4#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#iy#11240#12783#
#Daub4#4#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Daub4#4#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#y#17103#17587#
#Daub4#4#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ih#29179#30337#
#Daub4#4#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ao#52378#54500#

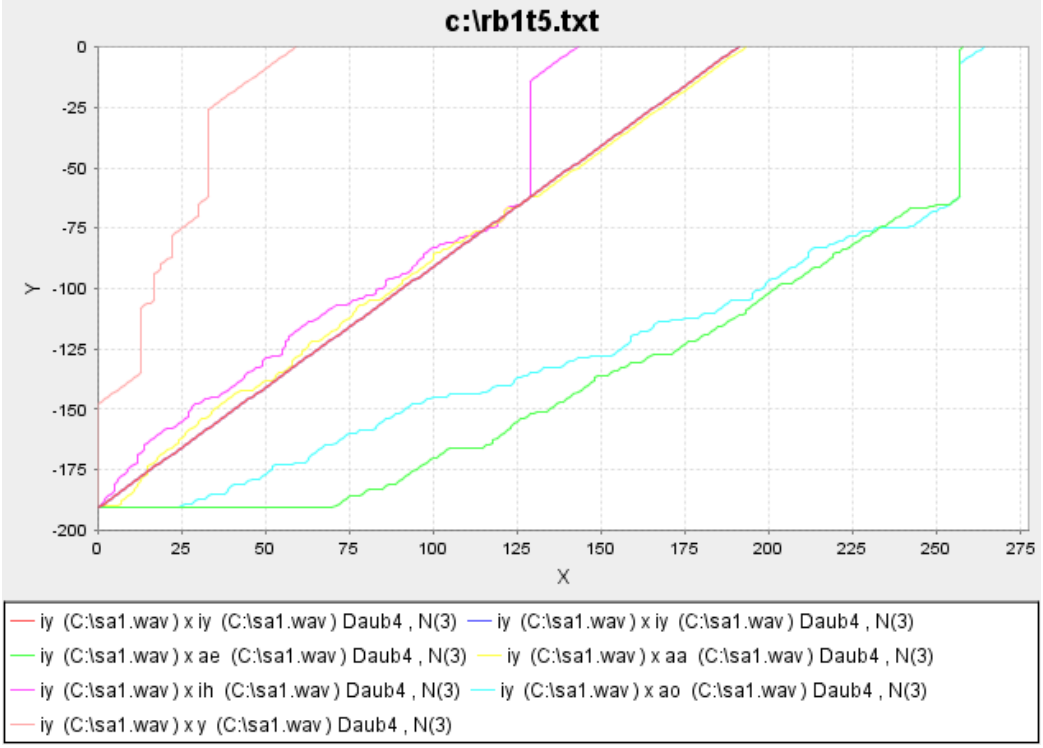

Figura 4.4: Resultados com **algoritmo DTW proposto usado Daubechies 4 nível 3**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMAS DE TESTE: **ae** (quantidade de pontos: 2079, tipo: vogal) ; **y** (quantidade de pontos: 484, tipo: semi-vogal) ; **aa** (quantidade de pontos: 1552, tipo: vogal) ; **ih** (quantidade de pontos: 1158, tipo: vogal) ; **ao** (quantidade de pontos: 2122, tipo: vogal) ; **iy** (quantidade de pontos: 1543, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub4#4#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#iy#11240#12783#
#Daub4#4#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Daub4#4#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#y#17103#17587#
#Daub4#4#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ih#29179#30337#
#Daub4#4#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ao#52378#54500#

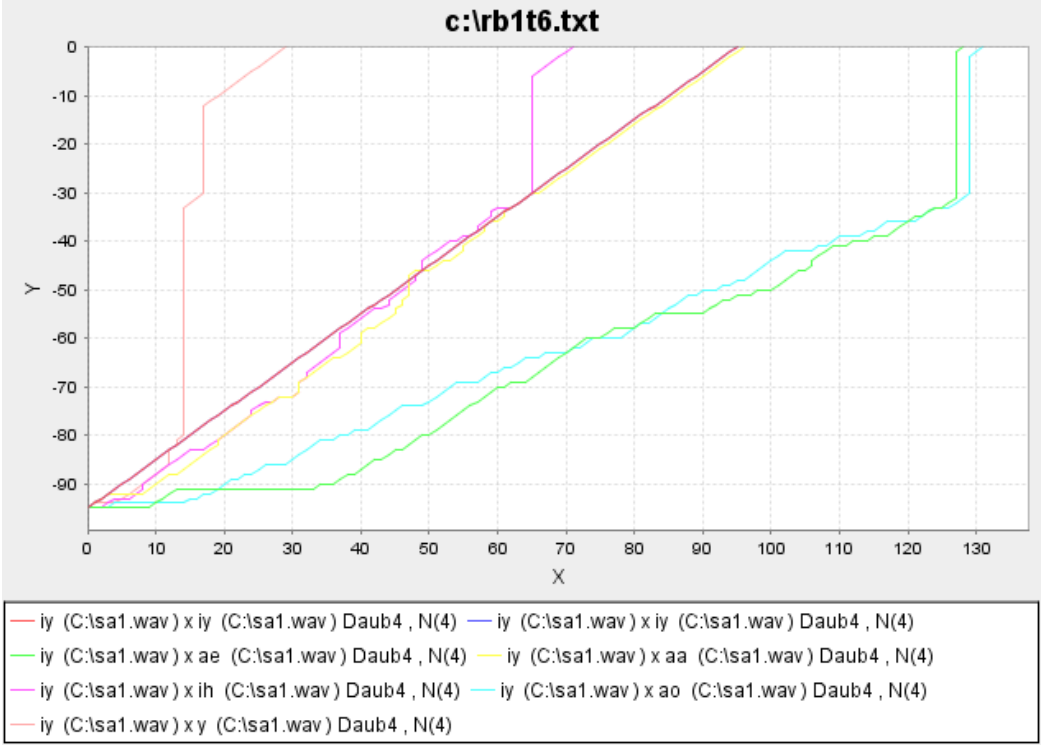

Figura 4.5: Resultados com **algoritmo DTW proposto usado Daubechies 4 nível 4**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMAS DE TESTE: **ae** (quantidade de pontos: 2079, tipo: vogal) ; **y** (quantidade de pontos: 484, tipo: semi-vogal) ; **aa** (quantidade de pontos: 1552, tipo: vogal) ; **ih** (quantidade de pontos: 1158, tipo: vogal) ; **ao** (quantidade de pontos: 2122, tipo: vogal) ; **iy** (quantidade de pontos: 1543, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub4#4#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#iy#11240#12783#
#Daub4#4#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Daub4#4#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#y#17103#17587#
#Daub4#4#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ih#29179#30337#
#Daub4#4#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ao#52378#54500#

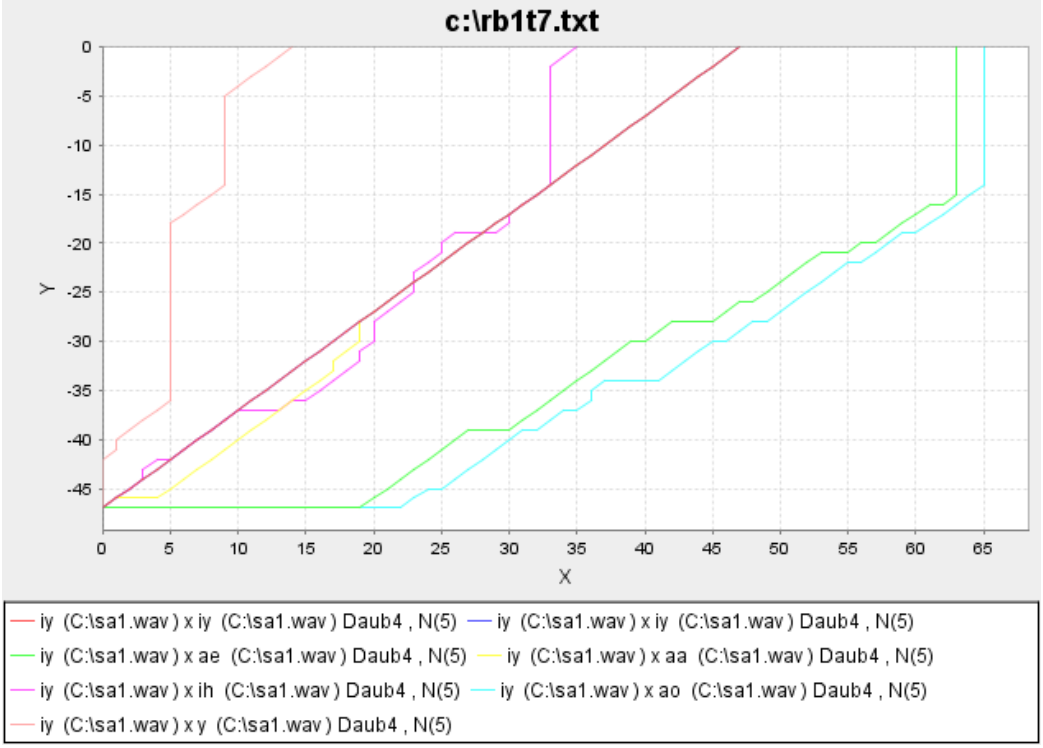

Figura 4.6: Resultados com **algoritmo DTW proposto usado Daubechies 4 nível 5**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMAS DE TESTE: **ae** (quantidade de pontos: 2079, tipo: vogal) ; **y** (quantidade de pontos: 484, tipo: semi-vogal) ; **aa** (quantidade de pontos: 1552, tipo: vogal) ; **ih** (quantidade de pontos: 1158, tipo: vogal) ; **ao** (quantidade de pontos: 2122, tipo: vogal) ; **iy** (quantidade de pontos: 1543, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub4#4#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#iy#11240#12783#
#Daub4#4#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Daub4#4#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#y#17103#17587#
#Daub4#4#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ih#29179#30337#
#Daub4#4#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ao#52378#54500#

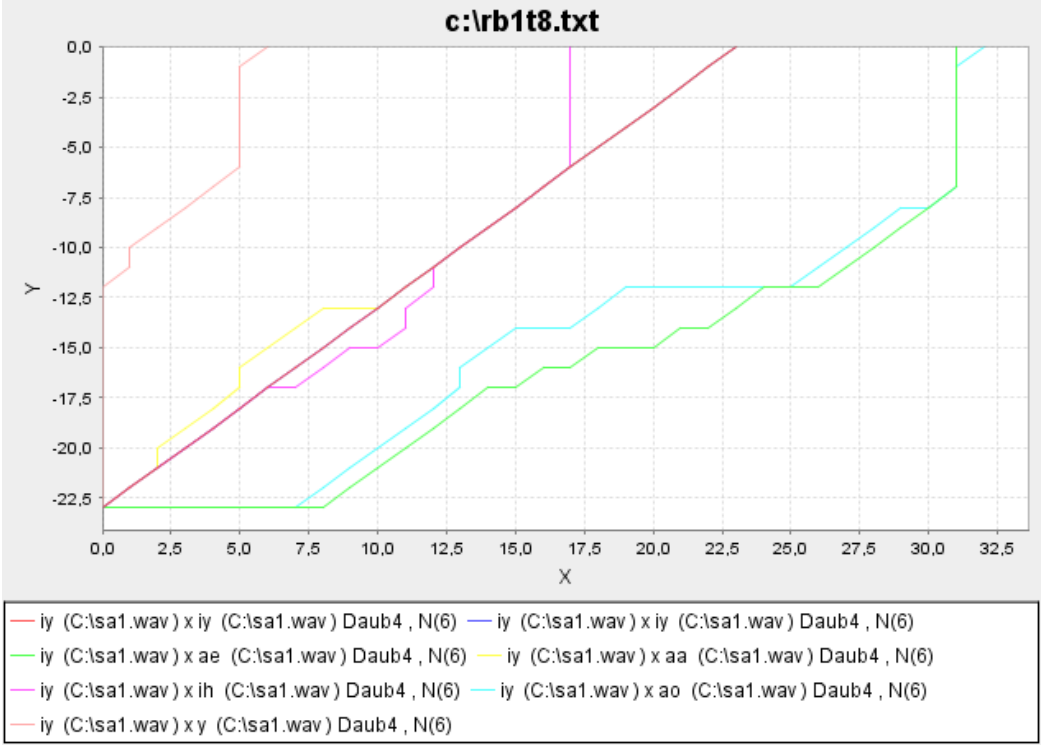

Figura 4.7: Resultados com **algoritmo DTW proposto usado Daubechies 4 nível 6**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMAS DE TESTE: **ae** (quantidade de pontos: 2079, tipo: vogal) ; **y** (quantidade de pontos: 484, tipo: semi-vogal) ; **aa** (quantidade de pontos: 1552, tipo: vogal) ; **ih** (quantidade de pontos: 1158, tipo: vogal) ; **ao** (quantidade de pontos: 2122, tipo: vogal) ; **iy** (quantidade de pontos: 1543, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub4#4#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#iy#11240#12783#
#Daub4#4#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Daub4#4#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#y#17103#17587#
#Daub4#4#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ih#29179#30337#
#Daub4#4#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ao#52378#54500#

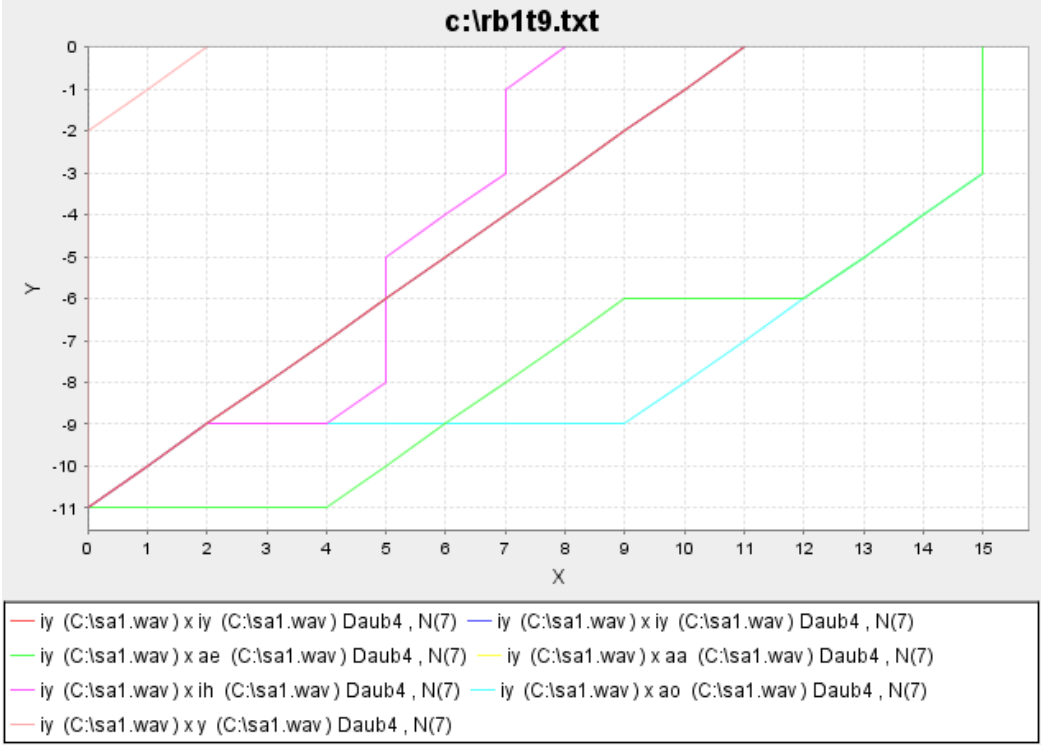

Figura 4.8: Resultados com **algoritmo DTW proposto usado Daubechies 4 nível 7**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMAS DE TESTE: **ae** (quantidade de pontos: 2079, tipo: vogal) ; **y** (quantidade de pontos: 484, tipo: semi-vogal) ; **aa** (quantidade de pontos: 1552, tipo: vogal) ; **ih** (quantidade de pontos: 1158, tipo: vogal) ; **ao** (quantidade de pontos: 2122, tipo: vogal) ; **iy** (quantidade de pontos: 1543, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub4#4#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#iy#11240#12783#
#Daub4#4#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Daub4#4#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#y#17103#17587#
#Daub4#4#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ih#29179#30337#
#Daub4#4#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ao#52378#54500#

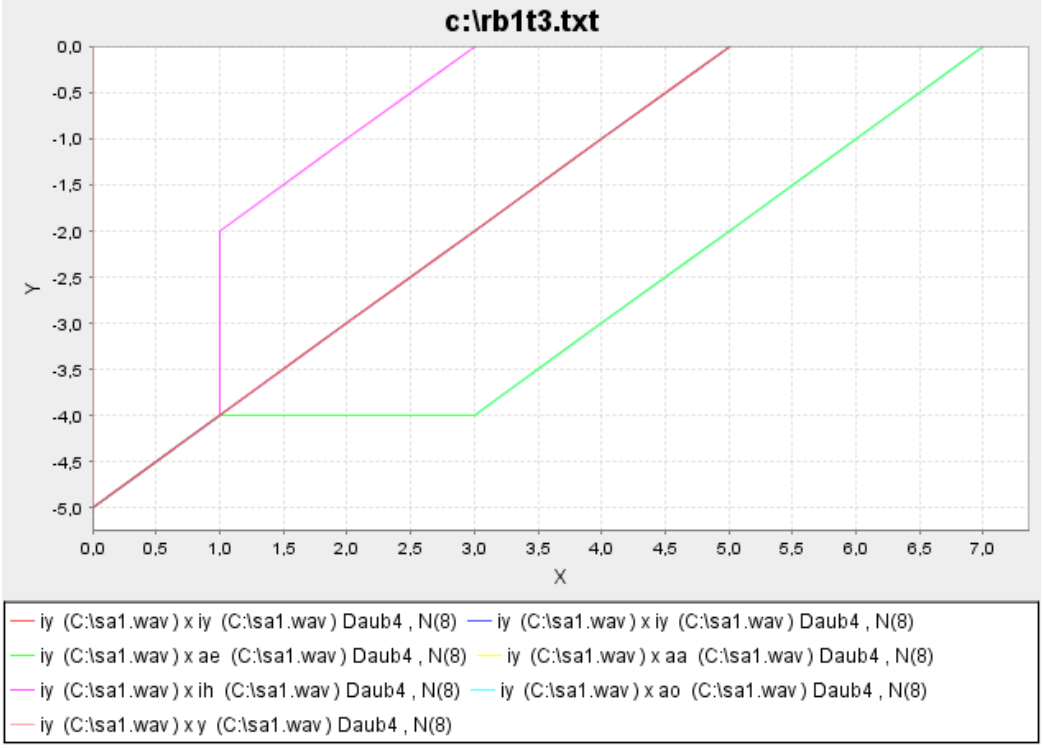

Figura 4.9: Resultados com **algoritmo DTW proposto usado Daubechies 4 nível 8**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMAS DE TESTE: **ae** (quantidade de pontos: 2079, tipo: vogal) ; **y** (quantidade de pontos: 484, tipo: semi-vogal) ; **aa** (quantidade de pontos: 1552, tipo: vogal) ; **ih** (quantidade de pontos: 1158, tipo: vogal) ; **ao** (quantidade de pontos: 2122, tipo: vogal) ; **iy** (quantidade de pontos: 1543, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub4#4#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#iy#11240#12783#
#Daub4#4#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Daub4#4#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#y#17103#17587#
#Daub4#4#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ih#29179#30337#
#Daub4#4#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ao#52378#54500#

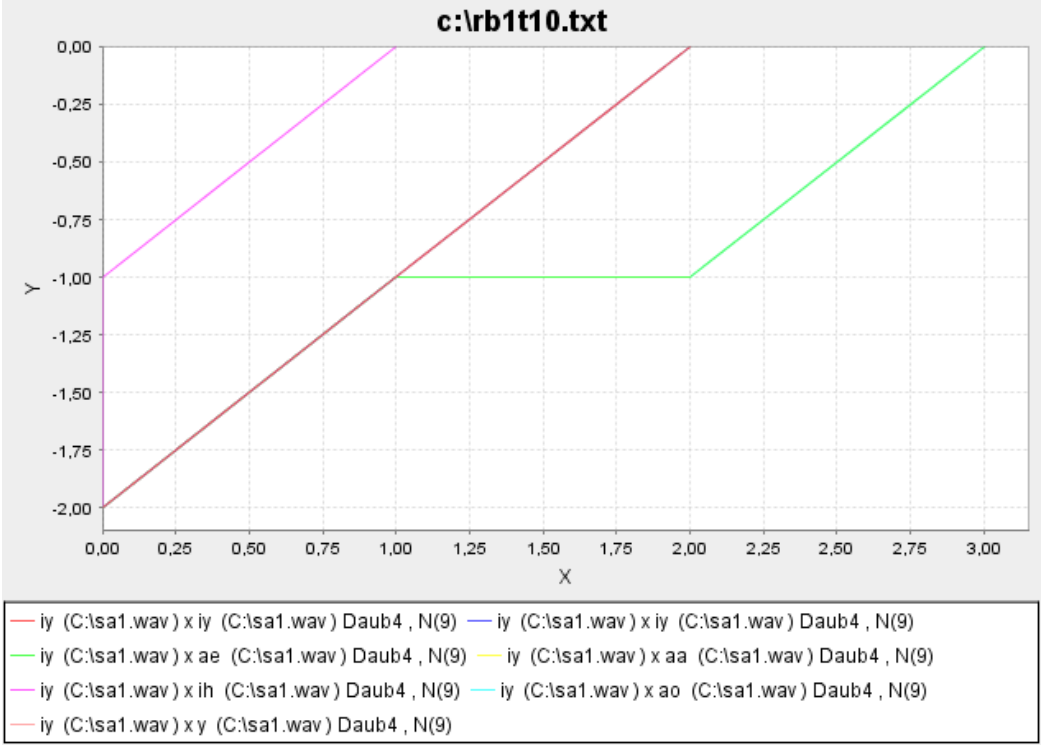

Figura 4.10: Resultados com **algoritmo DTW proposto usado Daubechies 4 nível 9**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMAS DE TESTE: **ae** (quantidade de pontos: 2079, tipo: vogal) ; **y** (quantidade de pontos: 484, tipo: semi-vogal) ; **aa** (quantidade de pontos: 1552, tipo: vogal) ; **ih** (quantidade de pontos: 1158, tipo: vogal) ; **ao** (quantidade de pontos: 2122, tipo: vogal) ; **iy** (quantidade de pontos: 1543, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub4#4#9#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#iy#11240#12783#
#Daub4#4#9#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Daub4#4#9#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#y#17103#17587#
#Daub4#4#9#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#9#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ih#29179#30337#
#Daub4#4#9#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ao#52378#54500#

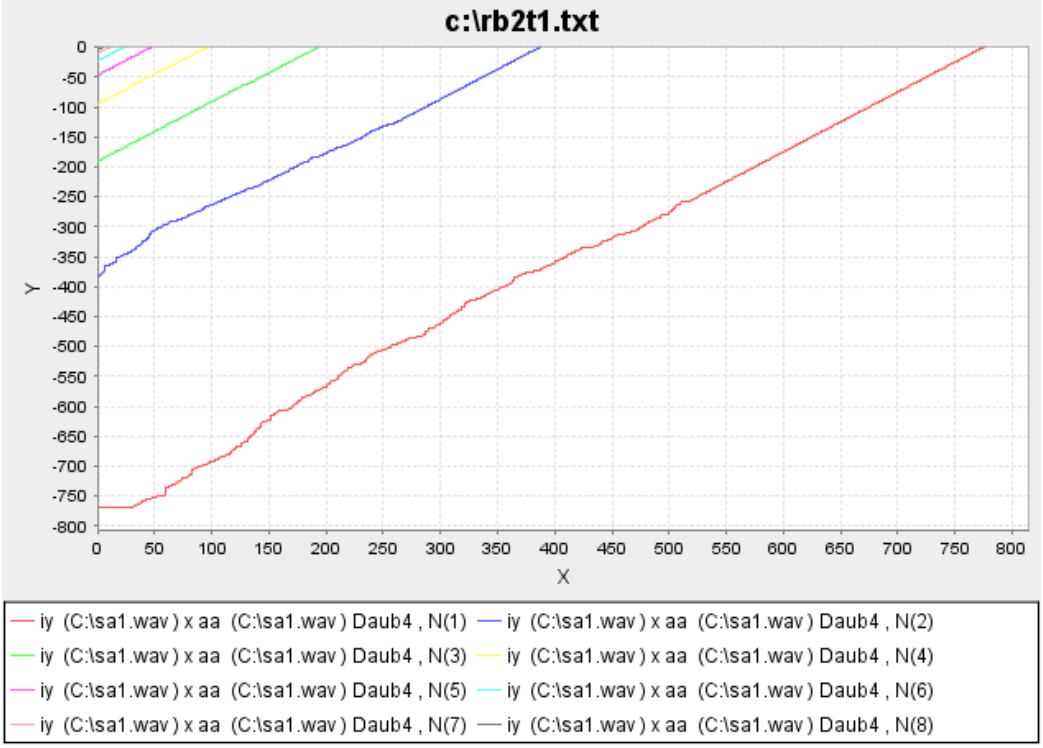

Figura 4.11: Resultados com **algoritmo DTW proposto usado Daubechies 4 níveis de 1 até 8**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub4#4#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#

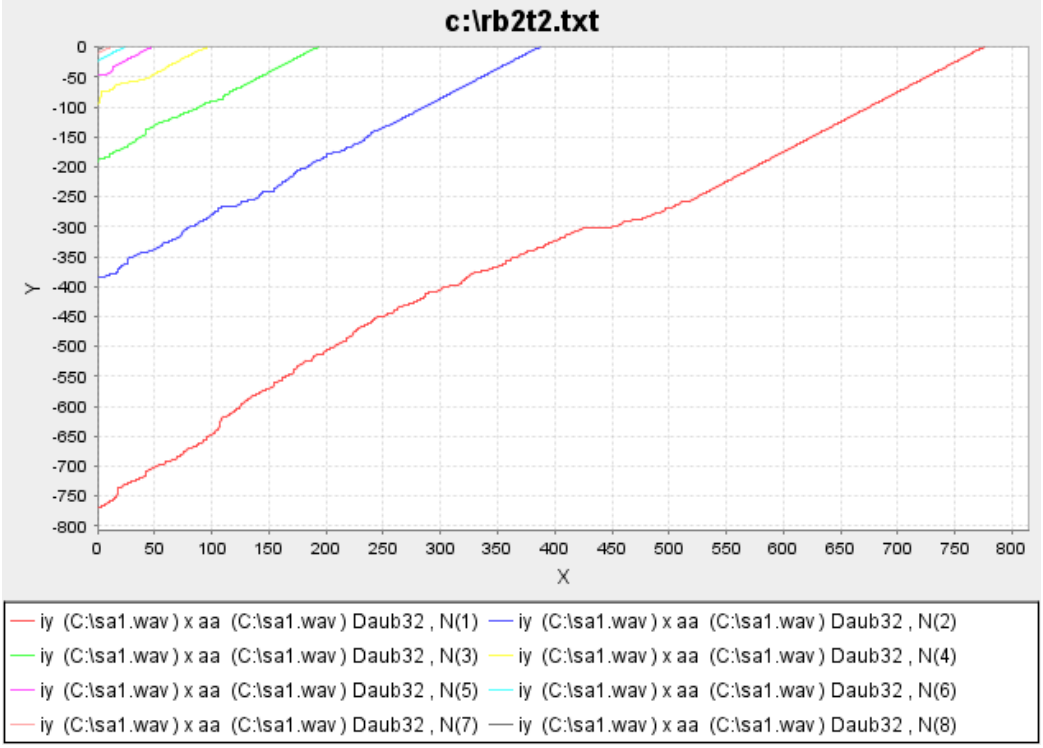

Figura 4.12: Resultados com **algoritmo DTW proposto usado Daubechies 32 níveis de 1 até 8**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub32#32#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub32#32#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub32#32#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub32#32#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub32#32#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub32#32#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub32#32#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub32#32#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#

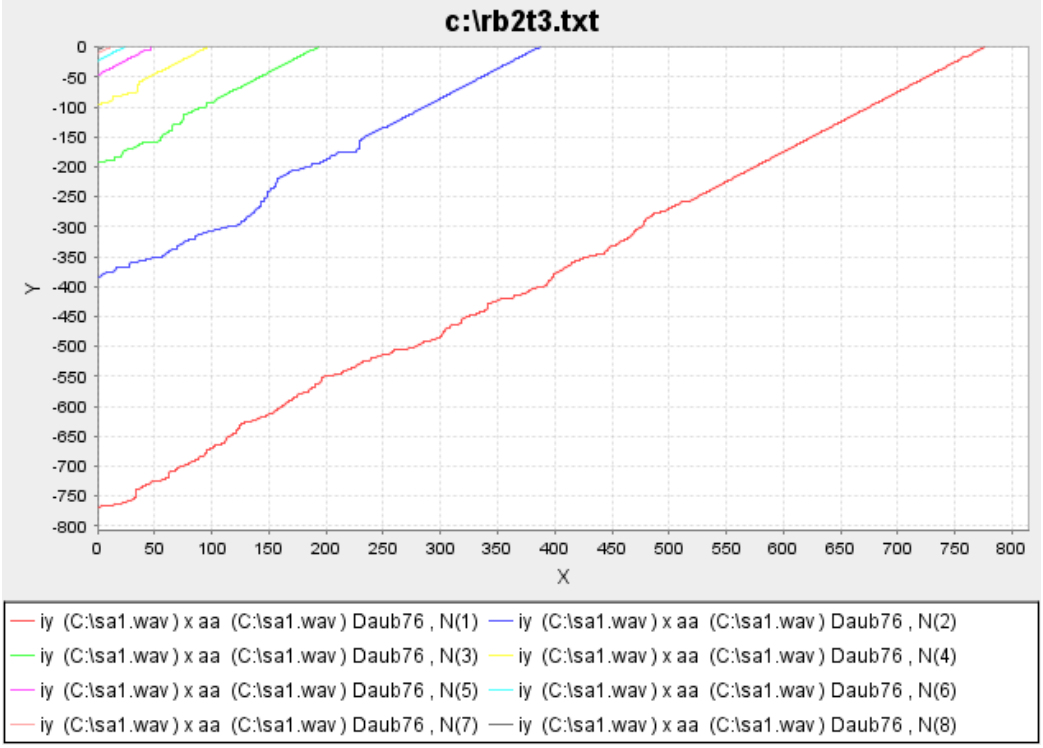

Figura 4.13: Resultados com **algoritmo DTW proposto usado Daubechies 76 níveis de 1 até 8**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Daub76#76#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub76#76#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub76#76#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub76#76#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub76#76#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub76#76#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub76#76#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Daub76#76#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#

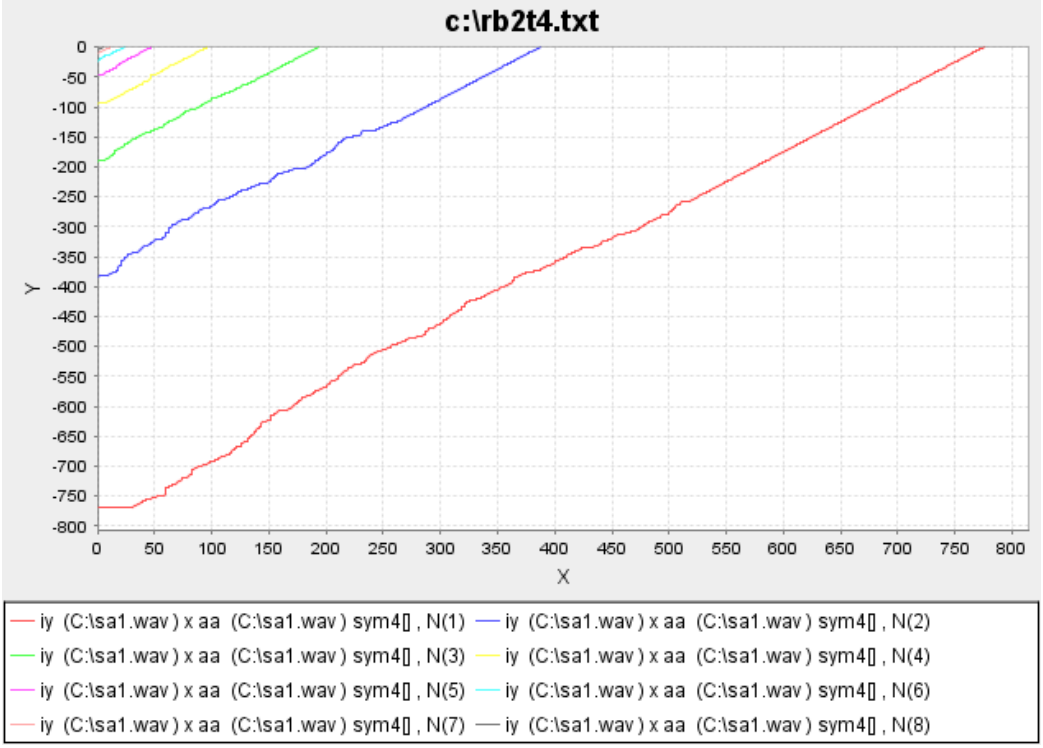

Figura 4.14: Resultados com **algoritmo DTW proposto usado Symmlet 4 níveis de 1 até 8**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Symm4#4#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm4#4#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm4#4#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm4#4#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm4#4#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm4#4#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm4#4#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm4#4#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#

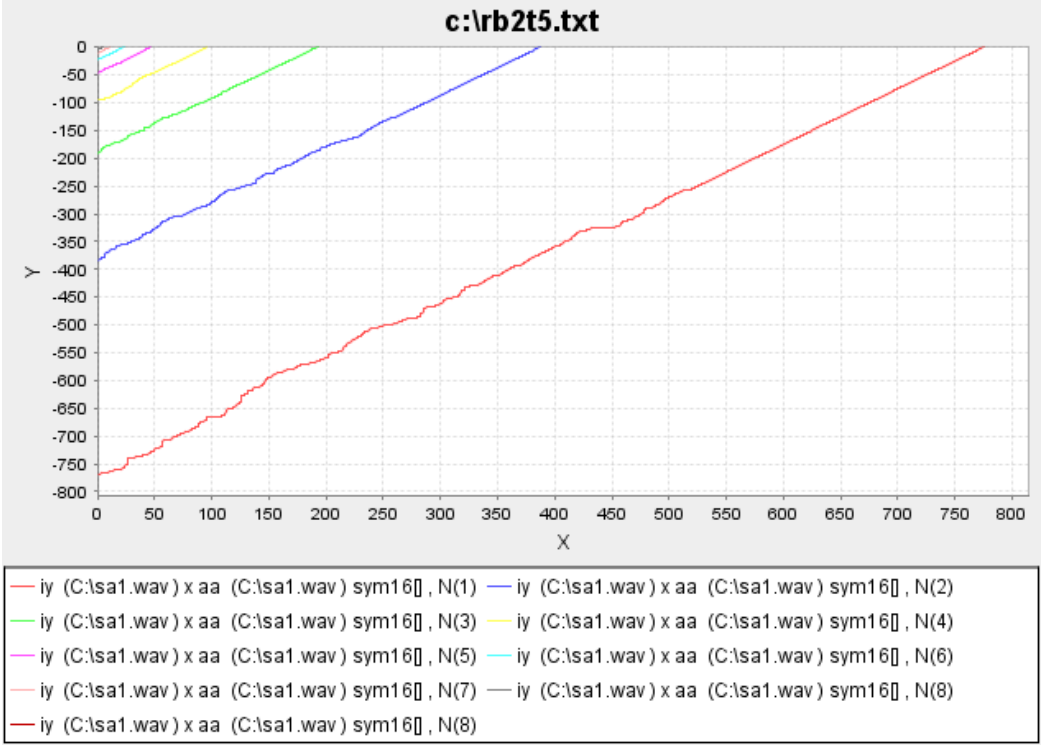

Figura 4.15: Resultados com **algoritmo DTW proposto usado Symmlet 16 níveis de 1 até 8**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Symm16#16#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm16#16#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm16#16#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm16#16#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm16#16#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm16#16#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm16#16#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Symm16#16#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#

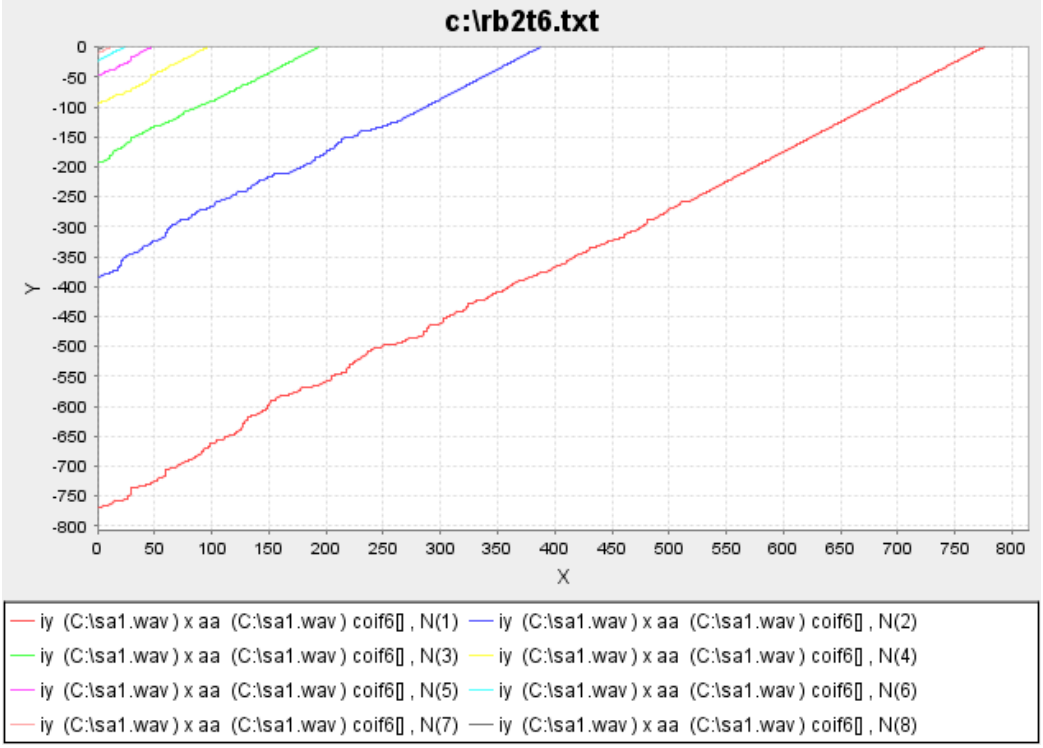

Figura 4.16: Resultados com **algoritmo DTW proposto usado Coiflet 6 níveis de 1 até 8**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Coif6#6#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Coif6#6#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Coif6#6#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Coif6#6#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Coif6#6#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Coif6#6#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Coif6#6#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Coif6#6#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#

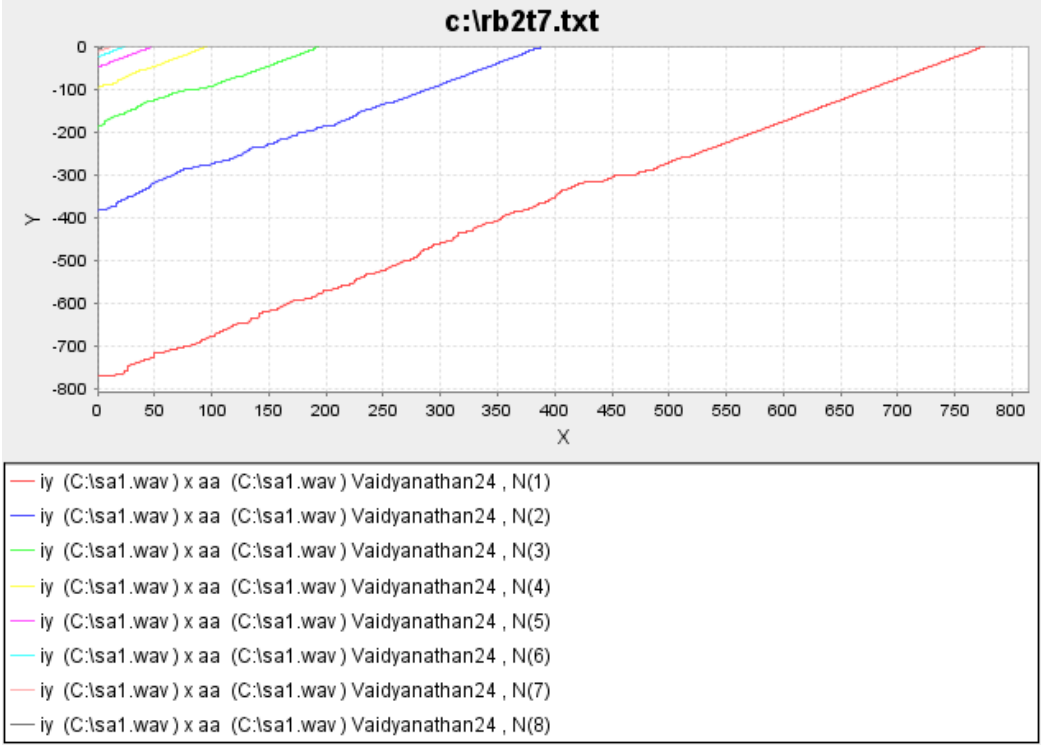

Figura 4.17: Resultados com **algoritmo DTW proposto usado Vaidyanathan 24 níveis de 1 até 8**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Vaidyanathan24#24#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#2#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#3#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#4#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#5#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#6#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#7#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#8#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#aa#19962#21514#

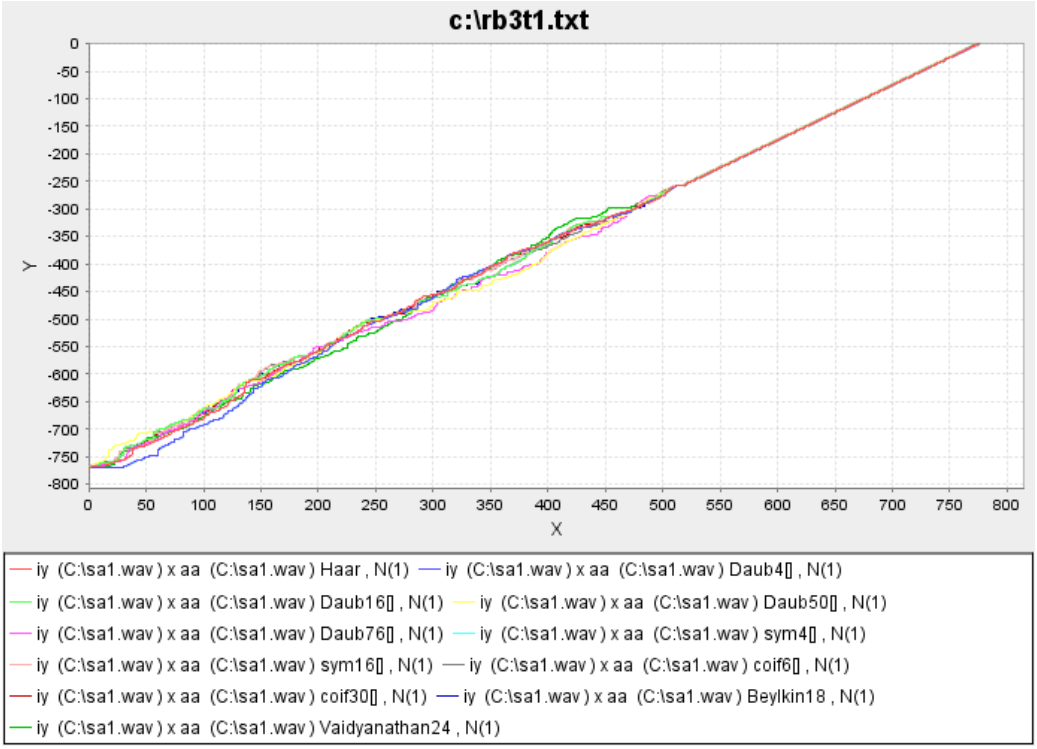

Figura 4.18: Resultados com **algoritmo DTW proposto usando diferentes filtros e nível 1**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Haar#2#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub50#50#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub76#76#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif6#6#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif30#30#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Beylkin18#18#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#

	iy x iy	iy x ae	iy x y	iy x aa	iy x ih	iy x ao
Rb1t1	**1,00000**	**0,96976**	**0,97195**	**0,99940**	**0,97567**	**0,96569**
Rb1t2	**1,00000**	**0,96523**	**0,96336**	**0,99946**	**0,97709**	**0,96136**
Rb1t3	**1,00000**	**0,93719**	**1,00000**	**1,00000**	**0,90362**	**0,91829**
Rb1t4	**1,00000**	**0,96320**	**0,97525**	**0,99779**	**0,97561**	**0,95539**
Rb1t5	**1,00000**	**0,95508**	**0,97477**	**0,99889**	**0,97272**	**0,96666**
Rb1t6	**1,00000**	**0,96312**	**0,83564**	**0,99788**	**0,97451**	**0,97298**
Rb1t7	**1,00000**	**0,95762**	**0,94898**	**0,99773**	**0,96521**	**0,94508**
Rb1t8	**1,00000**	**0,95563**	**0,90878**	**0,99521**	**0,96821**	**0,96555**
Rb1t9	**1,00000**	**0,95697**	**0,62874**	**1,00000**	**0,92740**	**0,88959**
Rb1t10	**1,00000**	**0,94868**	**1,00000**	**1,00000**	**0,86603**	**0,94868**

Tabela 4.5: Resultado da primeira bateria de destes comparando o Coeficiente de Correlação de cada alinhamento.

	Nível 1	Nível 2	Nível 3	Nível 4	Nível 5	Nível 6	Nível 7	Nível 8
Daub4	**0,99840**	**0,99779**	**0,99889**	**0,99788**	**0,99773**	**0,99521**	**1,00000**	**1,00000**
Daub4	**0,99358**	**0,99968**	**0,99956**	**0,99710**	**0,99885**	**0,99743**	**1,00000**	**1,00000**
Daub76	**0,99935**	**0,99798**	**0,99804**	**0,99946**	**1,00000**	**1,00000**	**1,00000**	**1,00000**
Sym4	**0,99840**	**0,99779**	**0,99889**	**0,99788**	**0,99773**	**0,99521**	**1,00000**	**1,00000**
Sym16	**0,99926**	**0,99960**	**0,99972**	**0,99788**	**0,99951**	**0,99937**	**1,00000**	**1,00000**
Coif6	**0,99932**	**0,99768**	**0,99825**	**0,99727**	**0,99232**	**0,99931**	**0,99438**	**1,00000**

Tabela 4.6: Resultado da segunda bateria de destes comparando o Coeficiente de Correlação de cada alinhamento.

Através da observação da correlação entre o alinhamento das respostas da matriz, utilizando o Coeficiente de Correlação entre eles, foi possvel verificar as variações e otimizações para determinadas Famílias Wavelets.

Esta correlação deve ser analisada considerando os valores mais distantes de 1 (menores que 1) como sendo resultado de um alinhamento distorcido, onde o reconhecimento foi inferior aos dos valores próximos a 1.

Nas tabelas 4.5, 4.6 e 4.7 podem ser observadas os valores dos respectivos coeficientes de correlação.

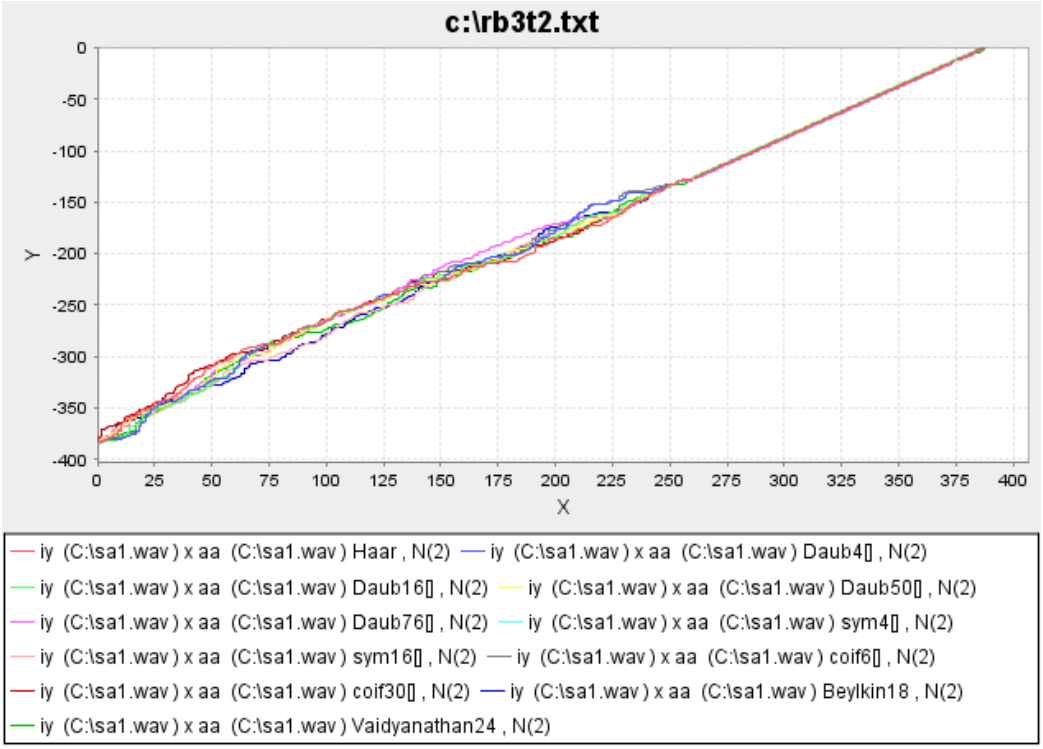

Figura 4.19: Resultados com **algoritmo DTW proposto usando diferentes filtros e nível 2**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Haar#2#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub50#50#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub76#76#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif6#6#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif30#30#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Beylkin18#18#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#

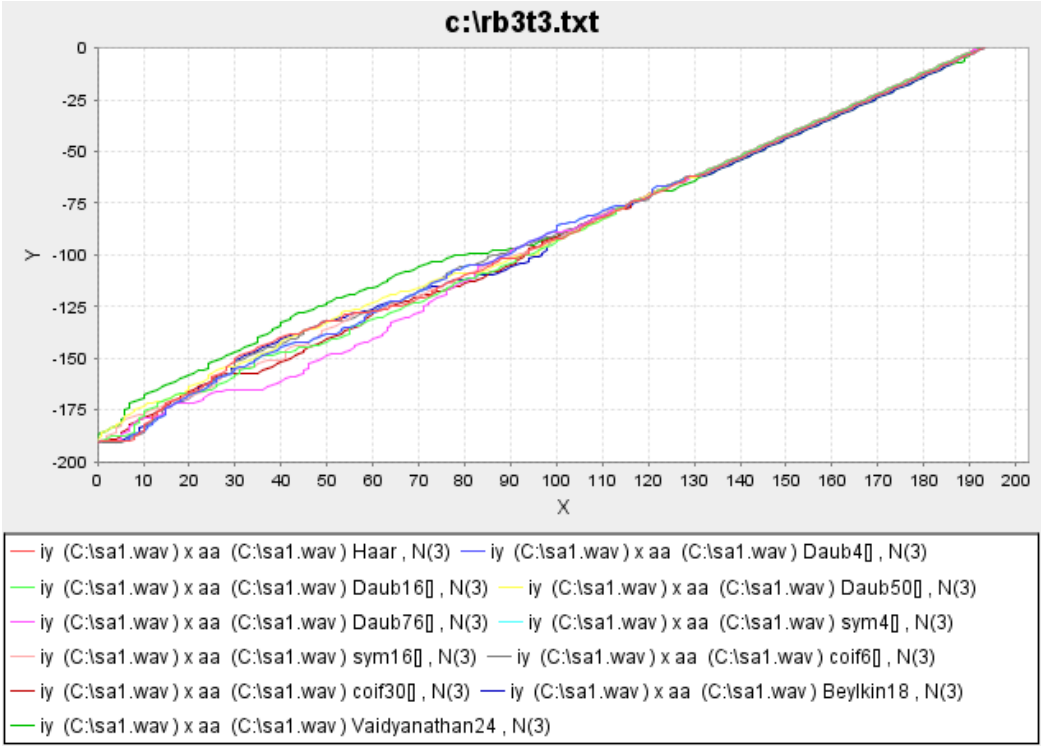

Figura 4.20: Resultados com **algoritmo DTW proposto usando diferentes filtros e nível 3**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Haar#2#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub50#50#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub76#76#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif6#6#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif30#30#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Beylkin18#18#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#

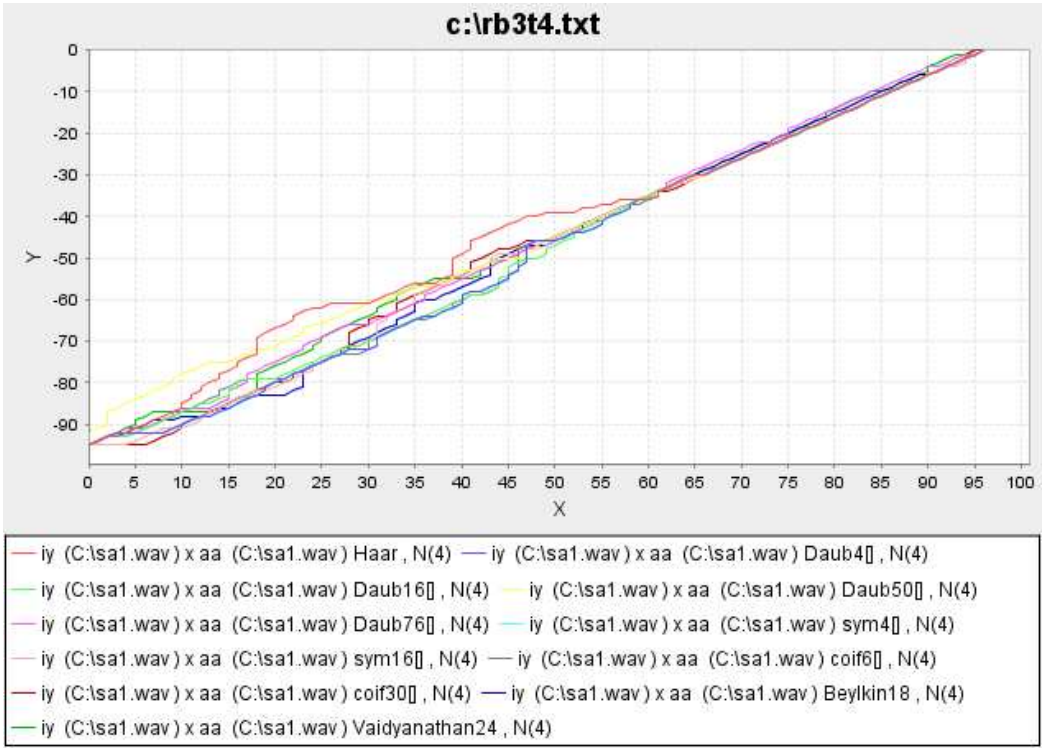

Figura 4.21: Resultados com **algoritmo DTW proposto usando diferentes filtros e nível 4**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Haar#2#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub50#50#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub76#76#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif6#6#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif30#30#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Beylkin18#18#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#

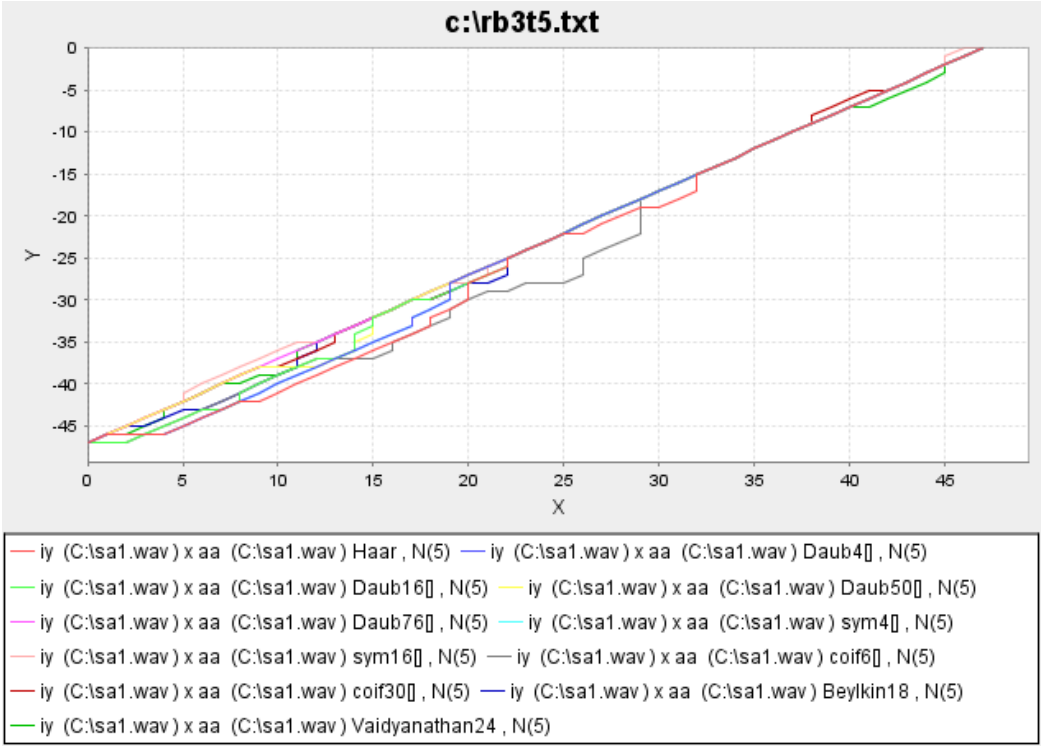

Figura 4.22: Resultados com **algoritmo DTW proposto usando diferentes filtros e nível 5**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Haar#2#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub50#50#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub76#76#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif6#6#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif30#30#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Beylkin18#18#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#

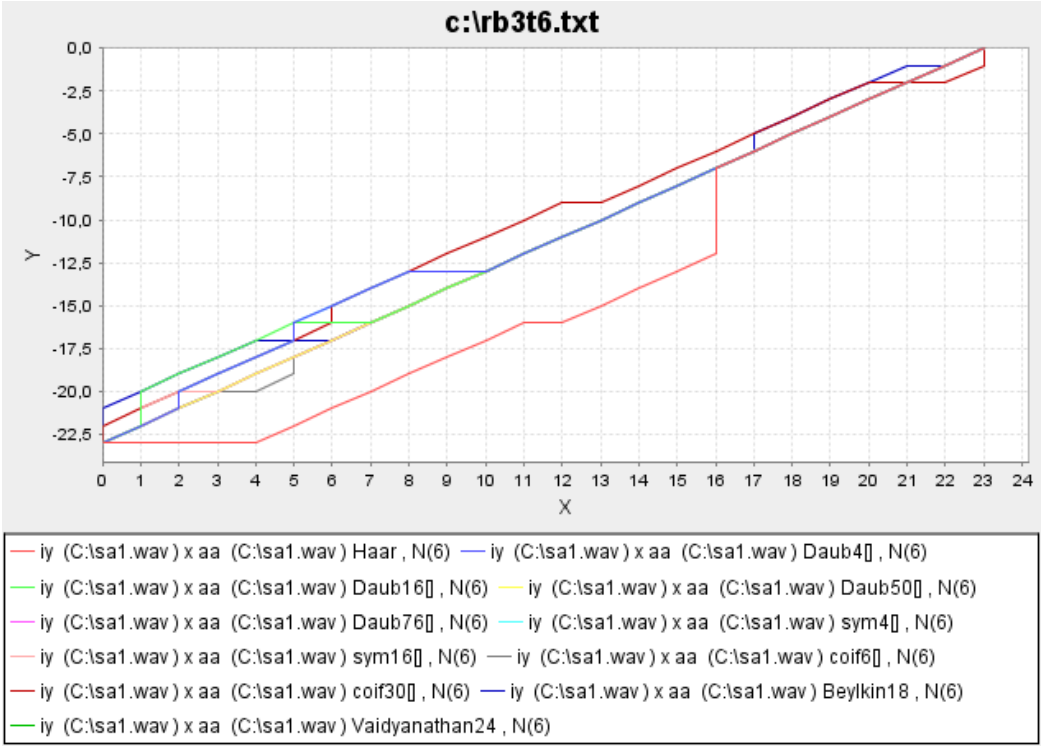

Figura 4.23: Resultados com **algoritmo DTW proposto usando diferentes filtros e nível 6**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Haar#2#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub50#50#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub76#76#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif6#6#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif30#30#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Beylkin18#18#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#

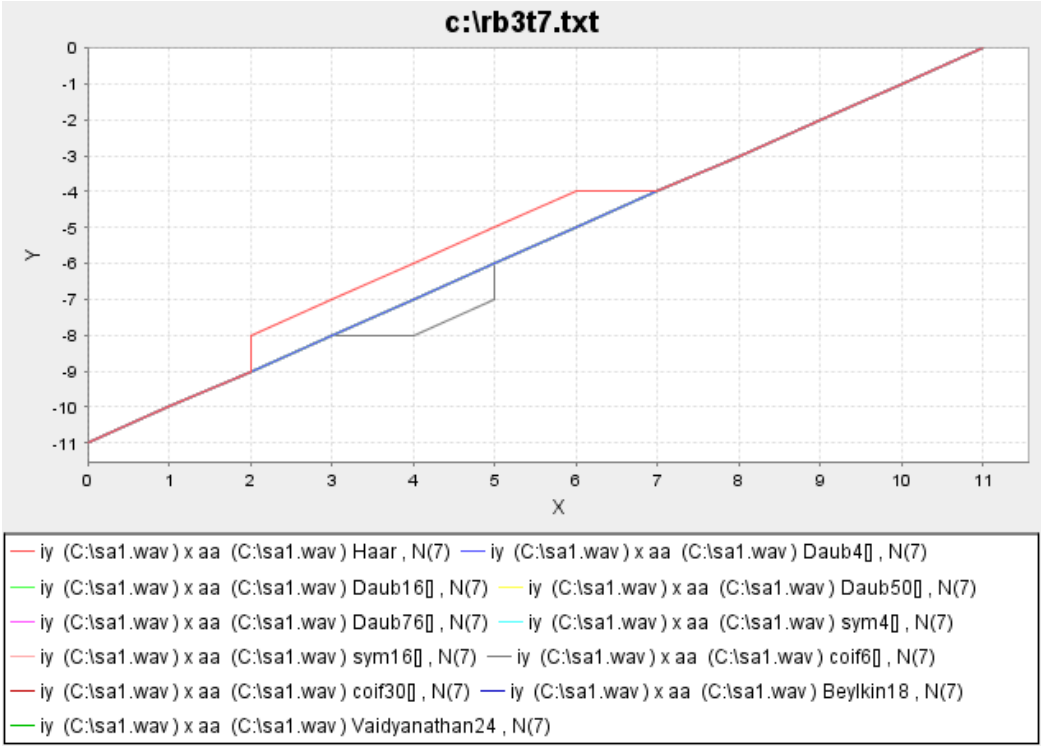

Figura 4.24: Resultados com **algoritmo DTW proposto usando diferentes filtros e nível 7**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Haar#2#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub50#50#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub76#76#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif6#6#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif30#30#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Beylkin18#18#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#

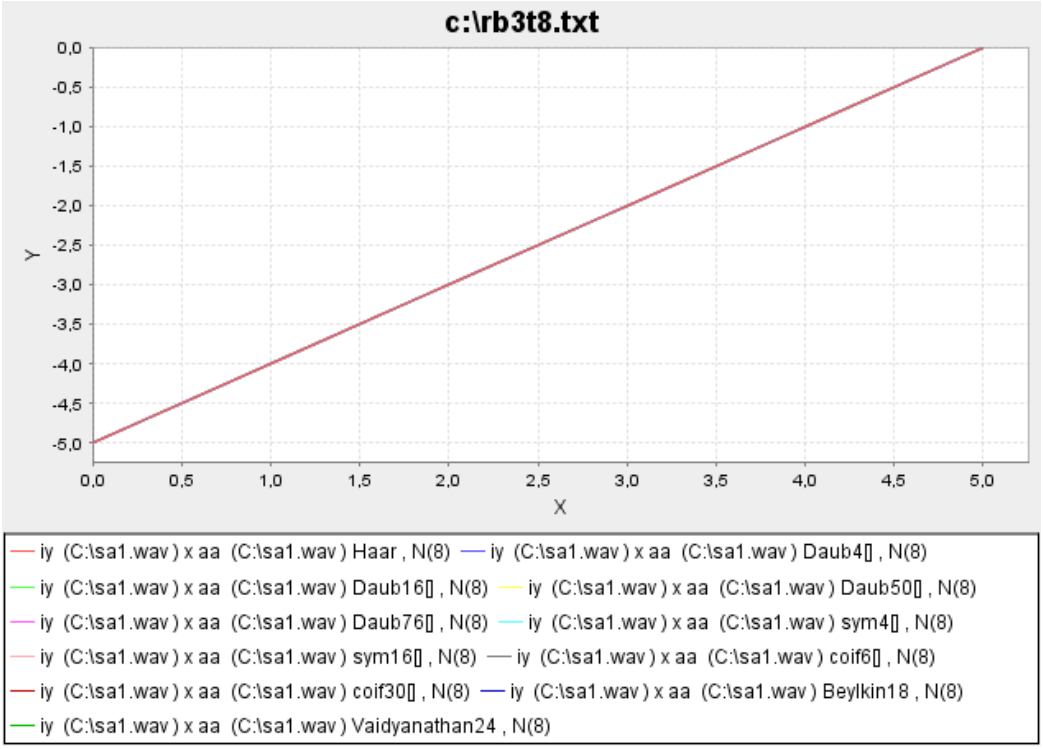

Figura 4.25: Resultados com **algoritmo DTW proposto usando diferentes filtros e nível 8**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **aa** (quantidade de pontos: 1552, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Haar#2#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub50#50#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Daub76#76#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym4#4#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#sym16#16#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif6#6#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#coif30#30#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Beylkin18#18#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#
#Vaidyanathan24#24#1#C:/ sa1.wav#iy#11240#12783&C:sa1.wav#aa#19962#21514#

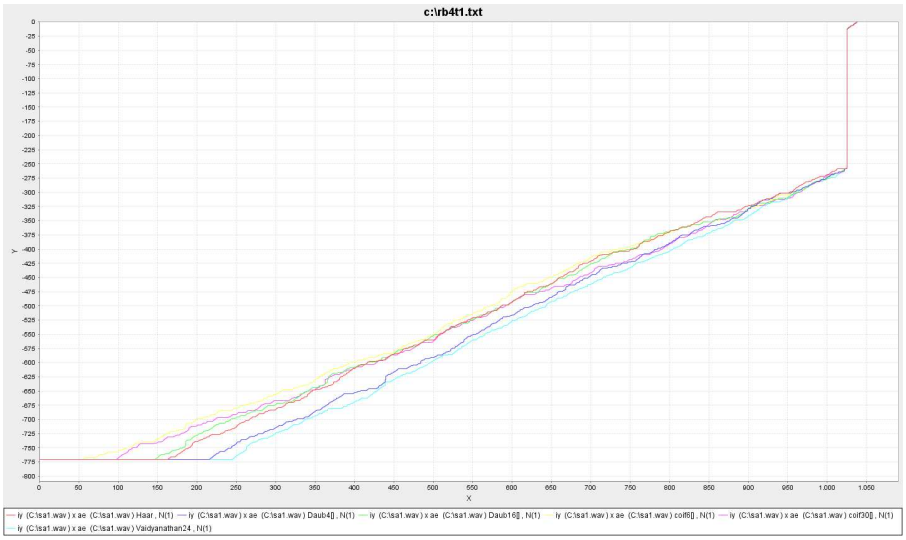

Figura 4.26: Resultados com **algoritmo DTW proposto usando diferentes filtros em um mesmo nível**.

FONEMA DE REFERÊNCIA: **iy** (quantidade de pontos: 1543, tipo: vogal).

FONEMA DE TESTE: **ae** (quantidade de pontos: 2079, tipo: vogal).

ARQUIVO *TEMPLATE*:
#Haar#2#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Daub4#4#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Daub16#16#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#coif6#6#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#coif30#30#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#
#Vaidyanathan24#24#1#C:/ sa1.wav#iy#11240#12783&C:/ sa1.wav#ae#14078#16157#

	Nível 1	Nível 2	Nível 3	Nível 4	Nível 5	Nível 6	Nível 7	Nível 8
Haar	**0,99840**	**0,99789**	**0,99786**	**0,99291**	**0,99639**	**0,96876**	**0,99019**	**1,00000**
Daub4	**0,99840**	**0,99779**	**0,99889**	**0,99788**	**0,99773**	**0,99521**	**1,00000**	**1,00000**
Daub16	**0,99941**	**0,99812**	**0,99939**	**0,99845**	**0,99880**	**0,99565**	**1,00000**	**1,00000**
Daub50	**0,99928**	**0,99891**	**0,99886**	**0,99870**	**0,99835**	**1,00000**	**1,00000**	**1,00000**
Daub76	**0,99935**	**0,99798**	**0,99804**	**0,99946**	**1,00000**	**1,00000**	**1,00000**	**1,00000**
Sym4	**0,99840**	**0,99779**	**0,99889**	**0,99788**	**0,99773**	**0,99521**	**1,00000**	**1,00000**
Sym16	**0,99926**	**0,99960**	**0,99972**	**0,99788**	**0,99951**	**0,99937**	**1,00000**	**1,00000**
Coif6	**0,99932**	**0,99768**	**0,99825**	**0,99727**	**0,99232**	**0,99931**	**0,99438**	**1,00000**
Coif30	**0,99935**	**0,96683**	**0,99946**	**0,99736**	**0,99957**	**0,99449**	**1,00000**	**1,00000**
Beylkin18	**0,99935**	**0,99944**	**0,99769**	**0,99741**	**0,99895**	**0,99492**	**1,00000**	**1,00000**
Beylkin18	**0,99910**	**0,99877**	**0,99562**	**0,99847**	**0,99931**	**1,00000**	**1,00000**	**1,00000**

Tabela 4.7: Resultado da Terceira bateria de destes comparando o Coeficiente de Correlação de cada alinhamento.

Capítulo 5

Conclusões e Trabalhos Futuros

Este trabalho apresentou uma versão modificada da DTW, que é baseada na DWT. Após uma ampla revisão bibliográfica sobre os conceitos fundamentais de processamento de sinais de voz, DTW e DWT, entre outros tópicos, definiu-se o algoritmo proposto. Este algoritmo se mostrou bastante promissor, atingindo o objetivo de reduzir o custo computacional da DTW para identificação de sinais de vozes com uma abordagem *pattern matching*. De modo geral, foi possível reduzir consideravelmente a quantidade de pontos do sinal *input* e dos *templates*, enquanto a precisão na identificação não foi prejudicada.

Durante os testes, foi possível verificar que a família *wavelet* não foi um fator de destaque para o sucesso do algoritmo, embora o uso de famílias com resposta ao impulso quase simétricas, como é o caso da Symmlet ou Coiflet, parecem se comportar ligeiramente melhor, devido provavelmente à resposta em fase quase linear das mesmas. O fator mais significante para a mudança no desempenho foi o nível máximo alcançado pelo sinal. Particularmente, a família escolhida para a filtragem influencia no reconhecimento e não na complexidade do método. Isso pode ser constatado pelo exame dos testes que, em um mesmo nível, comparavam diferentes sinais após a aplicação de diferentes *wavelets*. Estes testes demonstraram que cada *wavelet* se torna "ótima" em um determinado trecho do sinal, não possuindo um resultado homogêneo ao longo dele.

Outra importante constatação foi que as famílias intercalavam-se em desempenho ao longo dos níveis, isto é, uma família obteve um bom destaque em um trecho mas, quando um próximo nível era alcançado, o desempenho não se mantinha como nos níveis anteriores.

Uma característica interessante observada é que existe um limiar para a redução das amostras, isto é, existe um limite para o alcance de níveis. Este limite é encon-

trado no momento em que a quantidade de amostras não fornece suporte suficiente para haver a distinção entre os sinais. Este limite varia de acordo com a quantidade inicial de amostras, pois esta influencia diretamente a quantidade de amostras resultantes em cada nível. O que se pode afirmar é que uma quantidade inferior a trinta e duas amostras começa a prejudicar diretamente a análise dos resultados, interferindo na comparação entre os *templates* e o *input*.

Com relação às sub-bandas utilizadas nos experimentos, *s-input*, *s-template-1*, ..., *s-template-n*, mencionadas nos capítulos anteriores, para todos os testes verificou-se que, tanto para o *input* como para os *templates*, a sub-banda que preserva as características necessárias dos sinais é a aproximação do último nível. Este resultado era de se esperar, devido as características de baixas frequência dos sinais de voz.

Em termos de trabalhos futuros, pretende-se construir filtros *wavelets* "casados", isto é *matched wavelets* [6] [12]. Para determinados padrões de voz, tais filtros devem otimizar os resultados.

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, 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-

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-

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-

02, -1.91919589298e-01, 1.93095446660e-02, 1.55292480396e-01, -4.75268083411e-03, -1.20585522643e-01, 4.73422917264e-03, 8.99135475707e-02, -9.31855894990e-03, -6.33560374404e-02, 1.32285495850e-02, 4.12546930647e-02, -1.43668397842e-02, -2.41694978016e-02, 1.27664567156e-02, 1.22894360081e-02, -9.57779789923e-03, -5.08599164923e-03, 6.13775458674e-03, 1.42808879407e-03, -3.35764438092e-03, 7.61596943517e-06, 1.54963746970e-03, -3.34669216425e-04, -5.86481031899e-04, 2.64832881996e-04, 1.70001228366e-04, -1.36588307226e-04, -2.97699596284e-05, 5.30414312291e-05, -2.43700152682e-06, -1.57244207727e-05, 4.30804786171e-06, 3.35334586287e-06, -1.89592961769e-06, -3.90393173328e-07, 5.30236861690e-07, -3.70030837820e-08, -9.99039694453e-08, 3.00818865071e-08, 1.08490273378e-08, -7.45811655289e-09, 5.89795131038e-11, 1.03082334548e-09, -2.43354557375e-10, -6.40793825650e-11, 4.00053662725e-11, -3.12563935710e-12, -2.56706547615e-12, 8.01508853368e-13, -2.59795432889e-14, -3.39772085679e-14, 8.62403743472e-15, -9.29801252932e-16, 4.01462871233e-17};

- Daub72 :
  { 2.86792518275e-06, 6.82602867854e-05, 7.60215109966e-04, 5.24029737740e-03, 2.48905656448e-02, 8.56520925952e-02, 2.17756953097e-01, 4.06433697708e-01, 5.32266895260e-01, 4.17875335600e-01, 4.39751975293e-02, -2.94421039589e-01, -2.46807036978e-01, 9.81142041631e-02, 2.46537277608e-01, 7.27851509579e-03, -1.99337205608e-01, -4.58614007463e-02, 1.54106236627e-01, 5.02761800735e-02, -1.18803754310e-01, -3.98808535755e-02, 9.11567822580e-02, 2.50387214495e-02, -6.82090166368e-02, -1.13191003168e-02, 4.85130835478e-02, 1.42497266176e-03, -3.19807206776e-02, 3.98404019871e-03, 1.90635947806e-02, -5.65781324505e-03, -9.99026347328e-03, 5.02298910666e-03, 4.41348483535e-03, -3.48454144540e-03, -1.50307406629e-03, 1.99079377185e-03, 2.77681279571e-04, -9.46340382326e-04, 8.61456575899e-05, 3.69350728496e-04, -1.15511889584e-04, -1.13189946808e-04, 6.69474119693e-05, 2.37510668366e-05, -2.73139082465e-05, -1.18347105998e-06, 8.37221819816e-06, -1.58614578243e-06, -1.87081160285e-06, 8.31142127970e-07, 2.54842352255e-07, -2.45537765843e-07, 2.75324907333e-09, 4.79904346545e-08, -1.15609368881e-08, -5.61278434332e-09, 3.13884169578e-09, 1.09081555371e-10, -4.51254577856e-10, 8.96241820385e-11, 3.03742909811e-11, -1.59971668926e-11, 8.87684628721e-13, 1.07096935711e-12, -3.02928502697e-13, 5.54226318263e-15, 1.33807138629e-14, -3.20462854340e-15, 3.33997198481e-16, -1.40327417537e-17};

- Daub74 :
  { 2.02206086249e-06, 4.94234375062e-05, 5.66241837706e-04, 4.02414036825e-03, 1.97622861538e-02, 7.05848259771e-02, 1.87326331862e-01, 3.68440972400e-01, 5.18167040855e-01, 4.62207553661e-01, 1.30878963233e-01, -2.46180429761e-01, -2.94375915262e-01, 1.96715004523e-02, 2.51523254360e-01, 8.18060283872e-02, -1.81962291778e-01, -1.08451713823e-01, 1.29929646959e-01, 1.01780296838e-01, -9.66075406166e-02, -8.23302119065e-02, 7.50476199483e-02, 5.95674108715e-02, -5.92568156326e-02, -3.82538294793e-02, 4.58079441512e-02, 2.09728005925e-

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.

Apêndice III - Publicações durante o mestrado.

- GUIDO, Rodrigo Capobianco, VIEIRA, L. S., BARBON Jr, Sylvio, SANCHEZ, F. L., MACIEL, C. D., FONSECA, E., PEREIRA, J. C. *A neural-wavelet architecture for Voice Conversion*. Neurocomputing (Amsterdam). , v.-, p.in press - , 2007.

- SCALASSARA, P. R., MACIEL, C. D., GUIDO, Rodrigo Capobianco, PEREIRA, J. C., FONSECA, E., MONTAGNOLI, A., BARBON Jr, Sylvio, VIEIRA, L. S., SANCHEZ, F. L. *Autoregressive Decomposition and Pole Tracking Applied to Vocal Folds Nodule Identification*. Pattern Recognition Letters. , v.-, p.in press - , 2007.

- GUIDO, Rodrigo Capobianco, BARBON Jr, Sylvio, FONSECA, E. *Wavelet-based dynamic time warping*. Journal of Computational and Applied Mathematics. , v.1, p. in-press , 2007.

- VIEIRA, L. S., SANCHEZ, F. L., GUIDO, Rodrigo Capobianco, BARBON Jr, Sylvio. *On the Use of Wavelets for Voice Conversion*. In: Congresso de Engenharia de Audio, 2007, São Paulo. Anais do 5o Congresso de Engenharia de Audio, 2007.

- SANCHEZ, F. L., VIEIRA, L. S., BARBON Jr, Sylvio, GUIDO, Rodrigo Capobianco. *Wavelet-based Audio Watermarking*. In: Congresso de Engenharia de udio, 2007, São Paulo. Anais do 5o Congresso de Engenharia de Audio. , 2007.

- BARBON Jr, Sylvio, VIEIRA, L. S., GUIDO, Rodrigo Capobianco, SANCHEZ, F. L. *Wavelets* aplicadas à *Dynamic Time Warping* In: Congresso de Engenharia de Audio, 2007, São Paulo.

- GUIDO, Rodrigo Capobianco, VIEIRA, L. S., BARBON Jr, Sylvio, SANCHEZ, F. L. *A Fractal and wavelet-based approach for audio coding*. In: 8th IEEE International Symposium on Multimedia (ISM2006), 2006, San Diego, CA. Proceedings of the 8th IEEE International Symposium on Multimedia (ISM2006), 2006.

- GUIDO, Rodrigo Capobianco, MACIEL, C. D., MONTEIRO, M., FONSECA, E., PANCHAPAGESAN, S., PEREIRA, J. C., BARBON Jr, Sylvio. *A study on the best wavelet for audio compression*. In: 40th IEEE ASILOMAR Int. Conf. on Signals, Systems and Computers, 2006, Pacific Grouve,

CA. 40th IEEE ASILOMAR Int. Conf. on Signals, Systems and Computers. , 2006.

- BARBON Jr, Sylvio, GUIDO, Rodrigo Capobianco, PEREIRA, J. C., FONSECA, E., SANCHEZ, F. L., VIEIRA, L. S., GUILHERM, M. B. A. *Support Vector Machines and Wavelets for Voice Disorder Sorting*. In: 38th IEEE SSST - Southeastern Symposium on System Theory, 2006, Cookeville. 38th IEEE SSST - Southeastern Symposium on System Theory. , 2006. v.1. p.434 - 438

- MONTAGNOLI, A., RUBERT, J. B., GUIDO, Rodrigo Capobianco, PEREIRA, J. C., BARBON Jr, Sylvio. *Vocal Folds Vibrations with a three-dimensional deformable model*. In: 8th IEEE ISM2006 - Internation al Symposium on Multimedia, 2006, San Diego. Proceedings of the 8th IEEE ISM2006 - Internation al Symposium on Multimedia, 2006.

- FURLANY, J. ; VIEIRA, L. S. ; BARBON JUNIOR, S. . Compress£o de Imagens Digitais Utilizando a Transformada Discreta Cosseno. In: I Jornada de Iniciação Científica, 2006, São José do Rio Preto. I Jornada de Iniciação Científica, 2006.

- VIEIRA, L. S. ; FAVARO, G. C. ; BARBOSA, M. H. ; BARBON JUNIOR, S. ; GUIDO, R. C. .Utilizando as Transformadas Wavelet e Fourier na Detecção de Anormalidades em Sinais de ECG. In: I Jornada de Iniciação Científica, 2006, São José do Rio Preto. I Jornada de Iniciação Científica, 2006.

- VIEIRA, L. S. ; GUIDO, R. C. ; SANCHES, F. L. ; BARBON JUNIOR, S. ; PIMENTEL, C. C. ; BARTOLOMEU, L. A.;FAVARO, G. C. ; BARBOSA, M. H. *Wavelet-based ECG abnormalities detection*. In: Congresso Nacional de Matemática Aplicada e Computacional - CNMAC, 2006, Campinas. Congresso Nacional de Matemática Aplicada e Computacional - CNMAC, 2006.

- VIEIRA, L. S. ; BARBOSA, M. H. ; FAVARO, G. C. ; GUIDO, R. C. ; BARBON JUNIOR, S. ; SANCHES, F. L. Transformadas *Wavelet* e Fourier na Detecção de Anormalidades em Sinais de ECG. In: I Workshop de Computação, 2006, São José do Rio Preto. I Workshop de Computação, 2006.

- BARBON Jr, Sylvio, GUIDO, Rodrigo Capobianco, SANCHEZ, F. L., 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. XXIX CNMAC - Congresso Nacional de Matemática Aplicada e Computacional, 2006.

Referências Bibliográficas

[1] 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.

[2] AKAY, M. **Time Frequency and Wavelets in Biomedical Signal Processing.** New York: IEEE Press in Biomedical Engineering, 1998.

[3] Base de Dados TIMIT do *Linguistic Data Consortium*. http://www.ldc.upenn.edu/

[4] BOSI, M.; GOLDBERG, R.E. **Introduction to Digital Audio Coding and Standards.** Boston-USA, 2003.

[5] DELLER, J. R.; PROAKIS, J. G. and HANSEN J. H. L. **Discrete-Time Processing of Speech Signals, MacMillan Publishing Co.**, New York, 1993.

[6] CHAPA J.O.; RAO R.M. Algoritms for Designing Wavelets to Match a Specified Signal. **IEEE Transactions on Signal Processing**. V 48. N 12. USA, Dez-2000.

[7] COLEMAN J. **Introducing Speech and Language Processing**, Cambridge University Press, 2005, Cambridge, United Kingdom.

[8] DEITEL, H.; DEITEL P. **Java How to Program**, 7.ed., Prentice Hall, 2006.

[9] DENG L.;O'SHAUGHNESSY O. **Speech Processing: A Dynamic and Optimization-Oriented Approach**. Marcel Dekker Inc: New York-USA, 2003.

[10] DEVASAHAYAM, S.R. **Signals and Systems in Biomedical Engineering: Signal Processing and Physiological System Modeling**, New York: Kluwer Academic Publishers, 2000.

[11] 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.

[12] GUIDO, R.C.; SLAETS J.F.W.; KOBERLE R.; ALMEIDA L.O.B.; PEREIRA J.C. **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, v. 16, n. 1, p. 24-44, 2006.

[13] HAYKIN S.; VEEN B.V. **Sinais e Sistemas**, Porto Alegre: Bookman, 2002.

[14] JENSEN A.; COUR-HARBO A. **Ripples in Mathematics: The Discrete Wavelet Transform**, New York-USA: Springer-Verlag, 2000.

[15] MALLAT S. G. **A *wavelet* tour of signal processing**, San Diego: Academic Press, 1999.

[16] LIPSCHUTZ S. **Álgebra Linear**, 3.ed. São Paulo: Makron Books, 1994.

[17] MYERS C.; RABINER L. Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition, **IEEE Transactions On Acoustics, Speech, And Signal Processing**, Vol. Assp-28, No. 6, Dezembro 1980.

[18] OPPENHEIN A.V.; SCHAFER R.W. **Discrete Time Signal Processing**, 2.ed. New York: Prentice Hall,1999.

[19] QUATIERI T.F. **Discrete-Time Speech Signal Processing**. Upper Saddle River-USA: Prentice Hall, 2002.

[20] SCHILDT H. C++: **The Complete Reference**, 4.ed., McGraw-Hill Osborne Media, 2002.

[21] STRANG G.; NGUYEN T. **Wavelets and filter banks**, Wellesley-Cambridge Academic Press, Wellesley, 1997.

[22] VIEIRA L.S. **Conversão de Voz Baseada na Transformada *Wavelet***. IFSC-USP: São Carlos-SP, 03/2007, Dissertação de Mestrado.

[23] WALKER J.S. **A Primer on Wavelets and Their Scientific Applications**, Whashington-USA: Chapman and Hall/CRC, 1999.

[24] WILLIAMS J. R.; AMARATUNGA K. **Introduction to wavelets in engineering, Int. Journal for Numerical Methods in Engineering**, v. 37, 1994, pp. 2365-2388.