TÉCNICAS PARA CONVERSÃO DE ORADOR EM SINAIS DE VOZ

Victor Pereira da Costa

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Luiz Wagner Pereira Biscainho

Rio de Janeiro
Março de 2017

TÉCNICAS PARA CONVERSÃO DE ORADOR EM SINAIS DE VOZ

Victor Pereira da Costa

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Luiz Wagner Pereira Biscainho, D.Sc.

Prof. Sergio Lima Netto, Ph.D.

Prof. Amaro Azevedo de Lima, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
MARÇO DE 2017

Costa, Victor Pereira da

Técnicas para Conversão de Orador em Sinais de Voz/Victor Pereira da Costa. – Rio de Janeiro: UFRJ/COPPE, 2017.

X, 71 p.: il.; 29, 7cm.

Orientador: Luiz Wagner Pereira Biscainho

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2017.

Referências Bibliográficas: p. 66 – 71.

1. Conversão de Falante. 2. Processamento da Fala. 3. Aprendizado de Máquina. I. Biscainho, Luiz Wagner Pereira. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Agradecimentos

Agradeço, primeiramente aos meus pais, pelo amor que me deram, e por me proporcionarem as oportunidades que tantos não têm. Agradeço aos meus amigos, colegas e familiares, pelo suporte e pelas memórias que compartilhamos, de momentos bons e ruins. Agradeço aos meus professores, pelas várias lições que me deram, fossem elas de assuntos técnicos quanto de outras naturezas. E finalmente agradeço ao meu orientador, por toda a ajuda que me deu e pela paciência que teve comigo.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

TÉCNICAS PARA CONVERSÃO DE ORADOR EM SINAIS DE VOZ

Victor Pereira da Costa

Março/2017

Orientador: Luiz Wagner Pereira Biscainho

Programa: Engenharia Elétrica

Esse trabalho apresenta um sistema de conversão de falante, um sistema que possa transformar um sinal de fala dito por um falante em um sinal que pareça ter sido dito por outro falante, sem alterar o que é dito nem características como emoção ou ênfase. O objetivo principal é a comparação do desempenho de diferentes técnicas para a realização da conversão. Para isso foi implementado um sistema unificado que realiza as etapas de análise, conversão e síntese necessárias para a transformação do falante. Foram avaliadas quatro técnicas de conversão: três da literatura, baseadas em modelos de misturas gaussianas, modelos ocultos de Markov e redes neurais *feed-foward*; e uma nova, baseada em redes neurais recorrentes. Além disso, também foram implementadas duas técnicas para gerar a excitação na síntese, uma utilizando um pulso paramétrico treinado a partir os sinais de fala e uma utilizando o algoritmo PSOLA. Sobre esse sistema foram realizados dois experimentos para medir a qualidade da conversão, um utilizando como métrica a distância entre os *cepstra* dos sinais e um utilizando um sistema de identificação de falante. Os testes mostraram que o método baseado em modelo de misturas gaussianas obteve melhores resultados, mas todos os métodos possuem desempenho próximo.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

TECHNIQUES FOR SPEAKER CONVERSION IN SPEECH SIGNALS

Victor Pereira da Costa

March/2017

Advisor: Luiz Wagner Pereira Biscainho

Department: Electrical Engineering

This work presents a voice conversion system, a system that transforms a voice signal spoken by some speaker into a signal that sounds like it was spoken by another speaker, without changing the textual content of the speech or changing information like emotion or emphasis. The main objective of this work is to compare the conversion as done by different methods. To accomplish this, a unified voice conversion system containing the analysis, conversion and synthesis steps necessary to transform the speaker was implemented. Four voice conversion techniques, three from the literature, based on Gaussian mixture models, hidden Markov models and feed forward neural networks, and one novel based on recurrent neural networks, were evaluated. Two methods to generate the excitation used in the synthesis step were also implemented, one utilizing a parametric pulse trained on the speech signals, and one utilizing the PSOLA algorithm. On this system a couple of experiments were conducted to assess the conversion quality of each method: one measuring the distance between the cepstra of the signals, and the other employing a speaker recognition system. In these experiments the conversion based on Gaussian mixture models yielded the best results, but all techniques were relatively close in terms of performance.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A linguagem falada é, talvez, a principal forma de comunicação entre seres humanos. Ela é extremamente rica em informações relevantes para os interlocutores: ela obviamente carrega a informação textual, ou seja, o que é dito, mas também contém informações sobre emoção, identidade, ênfases etc. A capacidade de alterar uma dessas informações transmitidas sem alterar as outras pode ser útil em uma gama de aplicações.

Esse trabalho trata da conversão de falante, ou seja, a alteração da identidade de um sinal de fala. Um sistema que realize a conversão altera um sinal de fala dito por um falante fonte de forma que este sinal seja perceptivelmente similar a um sinal dito por um falante alvo, sem alterar outras características originais, como o que é dito, a emoção transmitida etc. As aplicações de sistemas desse tipo são várias. Uma das mais citadas é na síntese de fala, onde treinar uma nova voz é muitas vezes bastante custoso. Como treinar uma conversão é geralmente menos trabalhoso, a conversão de falantes pode ser uma maneira eficiente de aumentar o número de vozes sintéticas possíveis. Em geral, qualquer sistema que seja sensível à identidade de um falante pode utilizar a conversão para aumentar o número de falantes suportados. Além disso, também existem aplicações artísticas, como dublagens ou produções musicais, entre outras. Nesse sentido, um sistema de conversão de falante pode ser considerado um "imitador automático".

Para que seja possível alterar a identidade do falante, é necessário entender quais características da fala são utilizadas para realizar essa identificação, e como separá-las das outras características da fala. Esse conhecimento pode ser usado, então, para desenvolver um modelo matemático da fala que explicite essas características, de forma que a alteração da identidade se resuma a alterar alguns dos parâmetros desse modelo de um jeito específico.

A estrutura simplificada de um sistema de conversão é apresentada na Figura 1.1. Inicialmente o sinal da fala passa por uma mudança de domínio, para um conjunto de parâmetros que facilite a conversão. Esses parâmetros são transformados de acordo com uma função que depende do método, de forma a que se assemelhem a parâmetros do falante alvo. Esse parâmetros são usados, então, para sintetizar o sinal convertido.

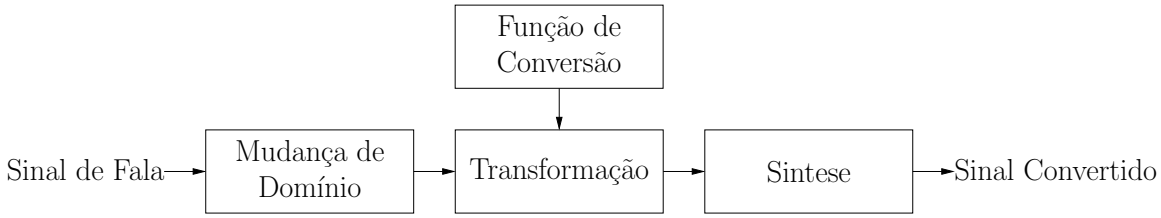

Figura 1.1: Diagrama simplificado de um sistema de conversão.

Possuindo um modelo que permita alterar a identidade, é necessário um método que diga como alterar os parâmetros desse modelo de forma que o sinal resultante pareça ter sido produzido pelo falante alvo. Funções de conversão simples, como por exemplo uma deformação na frequência por um fator constante, podem alterar a identidade, mas elas não são próprias para esse problema, pois não levam em consideração a relação entre fonte e alvo. Para converter a identidade entre dois falantes específicos, é necessária uma função de conversão mais complexa, cujas características dependem dos dois falantes. Alguns programas comerciais de edição de voz possuem módulos para a alteração da identidade, mas em geral eles utilizam essas conversões mais simples, geralmente dando ao usuário algum controle sobre os parâmetros de conversão.

A forma mais popular na literatura para desenvolver a função de conversão é tratar a conversão como um problema de regressão, com esta função sendo obtida a partir de um conjunto de dados dos dois falantes a partir de um processo de treinamento. Uma das primeiras técnicas a utilizar esta abordagem foi o trabalho de Abe *et al.* [1], que utiliza quantização vetorial para associar os parâmetros da fonte com os do alvo para cada quadro do falante. Uma técnica posterior, proposta por Stylianou *et al.* [2], introduziu o uso de transformações lineares combinadas com modelagem estatística e se tornou uma das propostas mais influentes da área. É sobre essa proposta que a maioria dos trabalhos tentam introduzir melhorias, utilizando, por exemplo, um modelo estatístico diferente [3][4] ou transformações não-lineares [5][6], ou introduzindo dependências temporais [7][8].

Saber qual dos métodos disponíveis na literatura possui o melhor desempenho não é fácil. A maioria dos trabalhos compara seus resultados com no máximo um trabalho anterior, e diferenças nas bases de dados e nos experimentos realizados tornam difícil a comparação direta entre os resultados reportados. Além disso, a maioria dos trabalhos não disponibiliza sinais convertidos e, mesmo quando estes são disponibilizados, a técnica utilizada na síntese do sinal convertido afeta a qualidade e dificulta a comparação entre métodos.

Esse trabalho procura, então, realizar uma comparação entre diferentes métodos. Para isso, foi desenvolvido um sistema unificado de extração de parâmetros e síntese, em que diferentes métodos de conversão possam ser utilizados. Foram implementados então quatro métodos para converter os parâmetros, três da literatura, escolhidos por exemplificarem diferentes abordagens para melhorar a conversão, e um novo que tenta combinar duas dessas abordagens. Foram realizados então experimentos para comparar o desempenho de cada técnica.

1.1 Organização

No Capítulo 2 são apresentados alguns dos métodos para representar o sinal em um domínio próprio para a conversão.

No Capítulo 3 são apresentados os métodos para realizar a conversão utilizados nesse trabalho.

No Capítulo 4 a estrutura detalhada do sistema é apresentada, e são apresentados detalhes da extração de parâmetros extras e de como é feita a síntese.

No Capítulo 5 são apresentados os experimentos realizados para medir o desempenho dos métodos e seus resultados.

Finalmente, no Capítulo 6, são apresentadas conclusões finais do trabalho e são propostos alguns trabalhos futuros.

Capítulo 2

Extração de Parâmetros da Fala

O primeiro passo para a conversão da fala é representar a fala em um domínio em que essa conversão seja possível. O sinal no domínio do tempo possui todas as informações relevantes para a identificação de falante, mas ele também possui muitas informações não relevantes. Por isso, realizar a conversão diretamente no domínio do tempo não é viável, e é necessário achar uma representação alternativa que concentre as características relevantes para identificação. Na Seção 2.1 apresenta-se de forma simplificada como a fala é produzida, quais elementos do trato vocal são mais relevantes para a identificação de falante e como esses elementos podem ser modelados. Em seguida a Seção 2.2 expande os detalhes desse modelo, com as Subseções 2.2.1 e 2.2.2 apresentando duas representações desse modelo. Finalmente, a Subseção 2.2.3 mostra uma modificação desse modelo que incorpora algumas características psicoacústicas da audição.

2.1 Modelo de Produção da Fala

A produção da fala humana é um processo complexo que envolve a interação de diversos sistemas, como o sistema nervoso, pulmões, um grande número de músculos, pregas vocais, mandíbula, língua, lábios, cavidade nasal etc. O processo é descrito a seguir de uma maneira simplificada e focando na ação do sistema respiratório.

O diafragma e a cavidade torácica agem sobre os pulmões forçando o ar contido neles pela traqueia e pela laringe, onde estão localizadas as pregas vocais. Estas são um tecido muscular cujo papel é dar à fala sua característica periódica em certos fonemas, chamados de vozeados. Para isso as pregas se fecham, interrompendo a passagem de ar e causando o aumento da pressão do ar abaixo da laringe. Quando a pressão se torna grande o bastante, as pregas se abrem, o que iguala a pressão e permite que as pregas vocais se fechem novamente, completando um ciclo. Esse processo se repete periodicamente, e o resultado é que o ar saindo da laringe possui

picos periódicos de pressão. Para fonemas não-vozeados, as pregas vocais não agem e o ar que sai da laringe possui uma característica turbulenta e aproximadamente constante [9].

O ar que sai da laringe reverbera na boca e na cavidade nasal, transformando-se em algo que entendemos como voz. A articulação dos vários elementos da boca, como dentes, lábios, língua etc., permite ao ser humano produzir a grande quantidade de sons que usamos para nos comunicar. Características menos mutáveis, como tamanho da cavidade nasal ou estrutura craniana, causam diferenças no sinal da fala que são particulares de cada falante e dão identidade à voz.

Um modelo matemático para a produção da voz que segue naturalmente dessa descrição é o modelo fonte-filtro [9], cujo diagrama é apresentado na Figura 2.1. Ele é composto de uma fonte que gera um sinal de excitação e um filtro que transforma esse sinal de excitação no sinal de voz. A fonte representa o fluxo de ar que sai da laringe, e o sinal de excitação pode possuir característica impulsiva, o que indica a ativação das pregas vocais, ou ruidosa, se as pregas vocais não estiverem ativas. O filtro representa a reverberação que o ar sofre na boca e na cavidade nasal. Existem diversos tipos de filtros que podem ser empregados no modelo, mas o mais comum é um filtro só-polos.[1] Tanto o filtro quanto a fonte mudam com o tempo de acordo com o que é dito, mas em escalas de tempo relativamente lentas em comparação com a escala do sinal da fonte.

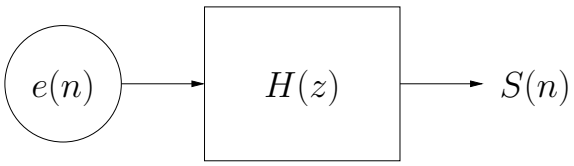

Figura 2.1: Diagrama do modelo fonte-filtro.

Por esse modelo, as informações de intensidade e de frequência fundamental estão contidas principalmente na fonte, enquanto a envoltória espectral (a distribuição de energia no espectro) está contida no filtro. O timbre ou cor do som é a característica psicoacústica que concentra as informações de identidade do falante, e ele está fortemente correlacionado com a envoltória espectral, e portanto o filtro. O *pitch* ou altura percebida é a característica psicoacústica relacionada à frequência fundamental e portanto à fonte. Ele também possui importância para a identidade, mas menor, estando concentrada principalmente no *pitch* médio.

[1] Na realidade esse tipo de filtro possui tanto polos quanto zeros, mas todos os zeros estão na origem, e portanto não afetam a resposta em frequência, apenas definindo o atraso global.

Assim, fica claro como a identidade de um sinal de voz pode ser alterada. Se o filtro controla a característica psicoacústica responsável pela identidade, então alterando-se o filtro de maneiras específicas essa identidade também é alterada. Para isso, uma parametrização apropriada do filtro é necessária, para o que algumas alternativas serão apresentadas na seção seguinte.

2.2 Representação do Filtro e da Envoltória

2.2.1 Predição Linear

A predição linear de ordem p é o processo que estima as amostras futuras de um sinal $s(n)$ a partir de p amostras anteriores. Ela é definida como:

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k), \tag{2.1}$$

em que a_k são os chamados coeficientes de predição e $\hat{s}(n)$ é o sinal estimado. O erro de predição é a diferença entre o sinal real e o estimado:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k). \tag{2.2}$$

Aplicando a transformada z nessa expressão, temos:

$$E(z) = S(z) - \sum_{k=1}^{p} a_k S(z) z^{-k} = S(z)\left(1 - \sum_{k=1}^{p} a_k z^{-k}\right) = S(z)A(z). \tag{2.3}$$

A função de transferência de $E(z)$ a $S(z)$ é, então:

$$H(z) = \frac{S(z)}{E(z)} = \frac{1}{A(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.4}$$

A predição linear pode ser entendida então como a filtragem de um sinal $e(n)$ por um filtro só-polos $H(z)$ para a obtenção de $s(n)$. Se $s(n)$ é um sinal de fala, esse filtro $H(z)$ é uma boa aproximação para o filtro do modelo fonte-filtro [9]. Assim, $H(z)$ pode ser usado para representar o filtro acústico do trato vocal e o sinal de erro $e(n)$ pode representar o fluxo de ar produzido pelos pulmões e pregas vocais.

Então, para alterar o filtro e com isso a identidade da voz, basta alterar os coeficientes a_k. Infelizmente, a alteração desses coeficientes afeta a resposta em frequência

do filtro de maneiras complexas e difíceis de prever. Existem representações alternativas para a_k que solucionam esse problema, uma das quais será vista na Subseção 2.2.2.

Para representar a variação no tempo do filtro do modelo da fala, a predição linear é feita em janelas do sinal de alguns milissegundos, onde o filtro pode ser considerado fixo e o sinal, estacionário. O n-ésimo bloco do sinal janelado é representado pela função $s_n(m) = s(n+m)w(m)$, onde $w(n)$ é uma função janela.

Os coeficientes a_k podem ser estimados pela minimização do erro quadrático médio:

$$\mathrm{E}[e^2(n)] = \mathrm{E}\left[\left(s(n) - \sum_{k=1}^{p} a_k s(n-k)\right)^2\right], \tag{2.5}$$

o que pode ser feito tornando-se o gradiente em relação a a_k igual a zero, ou seja:

$$\frac{\partial \mathrm{E}[e^2(n)]}{\partial a_{k'}} = 0\Big|_{k'=1,\ldots,p}. \tag{2.6}$$

Assim, temos:

$$\frac{\partial \mathrm{E}[e^2(n)]}{\partial a_{k'}} = \mathrm{E}\left[2s(n-k')\left(s(n) - \sum_{k=1}^{p} a_k s(n-k)\right)\right] = 0, \tag{2.7}$$

que pode ser reescrita como:

$$\mathrm{E}\left[s(n-k')\sum_{k=1}^{p} a_k s(n-k)\right] = \mathrm{E}\left[s(n)s(n-k')\right], \tag{2.8}$$

ou na forma matricial:

$$\begin{bmatrix} \mathrm{E}\left[s^2(n-1)\right] & \cdots & \mathrm{E}\left[s(n-1)s(n-p)\right] \\ \vdots & \ddots & \vdots \\ \mathrm{E}\left[s(n-p)s(n-1)\right] & \cdots & \mathrm{E}\left[s^2(n-p)\right] \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \mathrm{E}\left[s(n)s(n-1)\right] \\ \vdots \\ \mathrm{E}\left[s(n)s(n-p)\right] \end{bmatrix}. \tag{2.9}$$

Considerando o sinal estacionário dentro da janela de análise, podemos escrever (2.9) como:

$$\begin{bmatrix} R(0) & \cdots & R(p-1) \\ \vdots & \ddots & \vdots \\ R(p-1) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ \vdots \\ R(p) \end{bmatrix}, \tag{2.10}$$

$$\mathbf{Ra} = \mathbf{r}, \tag{2.11}$$

onde

$$R(i) = \mathrm{E}\left[s(n)s(n-i)\right] \tag{2.12}$$

é a função de autocorrelação, que independe de n por ser o sinal estacionário. Para calcular uma aproximação de $R(k)$, utiliza-se a autocorrelação temporal:

$$R_n(k) = \sum_{m=-\infty}^{\infty} s_n(m)s_n(m-k) = \sum_{m=-\infty}^{\infty} s(n+m)w(m)s(n+m-k)w(m-k). \tag{2.13}$$

Essa aproximação é tão melhor quanto mais próximo de ergódico for o sinal. Para sinais de voz, essa é uma boa aproximação [9].

A matriz $\mathbf{\Phi}$ em (2.11) é uma matriz de diagonais constantes, ou de Toeplitz, simétrica e positiva definida. Por isso, (2.11) pode ser resolvida eficientemente pela recursão de Levinson-Durbin.

2.2.2 Análise Homomórfica e *Cepstrum*

Análise ou filtragem homomórfica é um conjunto de técnicas de filtragem não linear. O procedimento básico consiste em utilizar um mapeamento não linear para um domínio em que técnicas de filtragem linear possam se utilizadas. A primeira ferramenta desse tipo a ser desenvolvida foi a transformada em *cepstrum*. O *cepstrum* foi inicialmente definido por Bogert *et al.* [10] como a transformada de Fourier inversa do logaritmo do espectro de potência do sinal, motivado pelo fato de o logaritmo do espectro de um sinal com eco possuir uma componente aditiva que depende apenas das características desse eco. Oppenhein *et al* expandiram o conceito e mostraram que ele se relaciona ao conceito mais geral de homomorfismo [11][12][13]. Eles definem o *cepstrum* como:

$$\hat{x}(n) = \frac{1}{2\pi}\int_{\pi}^{\pi} \log|X(\mathrm{e}^{j\omega})|\mathrm{e}^{j\omega n}d\omega, \tag{2.14}$$

onde $|X(\mathrm{e}^{j\omega})|$ é a magnitude da transformada de Fourier do sinal $x(n)$. Eles também definiram o *cepstrum* complexo como:

$$\hat{x}(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} \log\{X(\mathrm{e}^{j\omega})\}\mathrm{e}^{j\omega n}d\omega, \tag{2.15}$$

onde $\log\{X(\mathrm{e}^{j\omega})\}$ é o logaritmo complexo de $X(\mathrm{e}^{j\omega})$ e pode ser escrito como:

$$\log\{X(\mathrm{e}^{j\omega})\} = \log|X(\mathrm{e}^{j\omega})| + j\angle\left[X(\mathrm{e}^{j\omega})\right], \tag{2.16}$$

onde $\angle[X(\mathrm{e}^{j\omega})]$ é a fase da transformada de Fourier do sinal $x(n)$. O *cepstrum* de um sinal é nominalmente um sinal no domínio do tempo; entretanto, para enfatizar a transformação realizada, esse domínio é comumente chamado de quefrência.

Uma propriedade do *cepstrum* útil para o processamento de sinais da fala é a propriedade de separar sinais misturados por convolução. Considerando o sinal

$$s(n) = (r * h)(n), \tag{2.17}$$

no domínio da frequência a convolução se torna uma multiplicação:

$$S(\mathrm{e}^{j\omega}) = R(\mathrm{e}^{j\omega})H(\mathrm{e}^{j\omega}). \tag{2.18}$$

Se aplicarmos o logaritmo, obtemos:

$$\log\{S(\mathrm{e}^{j\omega})\} = \log\{R(\mathrm{e}^{j\omega})\} + \log\{H(\mathrm{e}^{j\omega})\}. \tag{2.19}$$

O logaritmo transforma a operação de multiplicação em uma de adição. Ao aplicar a transformada inversa a adição é preservada, e a convolução também se torna uma adição. Assim, temos:

$$\hat{s}(n) = \mathcal{D}_*\{(r * h)(n)\} = \hat{r}(n) + \hat{h}(n), \tag{2.20}$$

onde $\mathcal{D}_*\{\}$ é a operação de computar o *cepstrum*. Na teoria de sistemas homomórficos $\mathcal{D}_*\{\}$ é chamado de sistema característico para convolução. É essa propriedade de tornar uma convolução em uma soma que caracteriza o cálculo do *cepstrum* como uma análise homomórfica.

Se $h(n)$ for a resposta ao impulso de um filtro, $\hat{h}(n)$ é o *cepstrum* dessa resposta ao impulso. A resposta desse filtro no domínio z pode ser obtida diretamente do *cepstrum*:

$$H(z) = \exp\left(\sum_{n=0}^{\infty} \hat{h}(n) z^{-n}\right). \tag{2.21}$$

Se esse filtro possui resposta ao impulso infinita, seu *cepstrum* também é uma sequência infinita. Na prática essa sequência pode ser truncada em um comprimento L, já que para a maioria dos filtros envolvidos no processamento da fala o *cepstrum* tende a zero. O *cepstrum* truncado pode ser interpretado como os coeficientes de um filtro de resposta impulso finita:

$$D(z) = \log H_{\text{trunc}}(z) = \sum_{n=0}^{L} \hat{h}(n) z^{-n}. \tag{2.22}$$

A representação de um filtro em termos do *cepstrum* possui, em geral, algumas propriedades úteis quando comparada às representações tradicionais. Por exemplo, ela possui menos sensibilidade a ruído, melhor capacidade de interpolação etc. [14]. Essas propriedades tornam o *cepstrum* uma representação adequada para a conversão de fala. Pela importância dessa representação e para simplificar a notação, no resto desse trabalho o *cepstrum* do filtro será representado por $c_n = \hat{h}(n)$.

Da propriedade de transformar a convolução em soma vem uma das maiores utilidades do *cepstrum* para o processamento da fala. Ao se aplicar essa transformação a um sinal de fala, o *cepstrum* resultante é o resultado da soma do *cepstrum* correspondente ao filtro do modelo com o *cepstrum* correspondente à excitação. Esses dois *cepstra* possuem características bem diferentes: o *cepstrum* da excitação possui sua energia concentrada em torno da quefrência correspondente à frequência fundamental, enquanto o *cepstrum* do filtro está concentrado nas quefrências baixas. Utilizando essa informação, os *cepstra* da fonte e do filtro podem ser separados.

Essa propriedade é a base para o cálculo direto do *cepstrum* da envoltória. Nesse processo o sinal é primeiramente janelado, e o *cepstrum* é calculado utilizando-se a transformada de Fourier discreta (DFT) no lugar da transformada de Fourier de tempo discreto (DTFT). Assim, $X(e^{j\omega})$ se torna $X(k)$, e a equação se torna:

$$\hat{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \log\{X(k)\} \mathrm{e}^{j2\pi \frac{kn}{N}}, \tag{2.23}$$

As L primeiras amostras de $\hat{x}(n)$ são então consideradas o *cepstrum* do filtro. O *cepstrum* obtido dessa maneira gera uma envoltória imprecisa, por causa da interferência do *cepstrum* da excitação, que ainda possui alguma energia nas quefrências baixas. Uma maneira melhor de se calcular o *cepstrum* é minimizar [15][16]

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} (\exp R(\omega) - R(\omega) - 1)\, d\omega, \tag{2.24}$$

onde

$$R(\omega) = \log I_N(\omega) - \log \mid H(e^{j\omega}) \mid^2, \tag{2.25}$$

sendo $I_N(\omega)$ o periodograma modificado

$$I_N(\omega) = \frac{\left|\sum_{m=0}^{N-1} w(m)s(n+m)e^{-j\omega n}\right|^2}{\sum_{m=0}^{N-1} w^2(m)}, \tag{2.26}$$

e $H(e^{j\omega})$ a resposta em frequência definida pelo *cepstrum*. Esse método minimiza a interferência da excitação e aproxima melhor o filtro $H(e^{j\omega})$ da envoltória do espectro.

Um outro método de calcular o *cepstrum* é a partir da transformada z do filtro. Para um filtro

$$X(z) = G\frac{\prod_{k=1}^{M}(1-\beta_k z^{-1})}{\prod_{k=1}^{N}(1-\alpha_k z^{-1})}, \tag{2.27}$$

o *cepstrum* pode ser obtido pela soma dos logaritmos de cada termo:

$$\hat{X}(z) = \log|G| + \sum_{k=1}^{M}\log(1-\beta_k z^{-1}) - \sum_{k=1}^{N}\log(1-\alpha_k z^{-1}); \tag{2.28}$$

usando a expansão

$$\log(1-a) = -\sum_{n=1}^{\infty}\frac{a^n}{n}, \quad |a|<1, \tag{2.29}$$

temos:

$$\begin{aligned}\log(1-\alpha_k z^{-1}) &= -\sum_{n=1}^{\infty}\frac{\alpha_k^n}{n}z^{-n} \\ &= -\mathcal{Z}\left\{\frac{\alpha_k^n}{n}u(n-1)\right\},\end{aligned} \tag{2.30}$$

onde $\mathcal{Z}\{\}$ é a transformada z e $u(n)$ é a função degrau. Assim, se todos os pólos e zeros estiverem no interior do circulo unitário, o *cepstrum* fica:

$$\hat{x}(n) = \begin{cases} 0 & \text{se } n<0 \\ \log|G| & \text{se } n=0 \\ -\sum_{k=1}^{M}\frac{\beta_k^n}{n} + \sum_{k=1}^{N}\frac{\alpha_k^n}{n} & \text{se } n>0. \end{cases} \tag{2.31}$$

Essa relação pode ser utilizada para calcular o *cepstrum* a partir dos coeficientes da predição linear. O filtro obtido pela predição linear é dado por:

$$H(z) = \frac{G}{1-\sum_{k=1}^{p}a_k z^{-k}} = \frac{G}{\prod_{k=1}^{p}(1-\alpha_k z^{-1})}; \tag{2.32}$$

a equação (2.31) pode ser, então, utilizada para calcular o *cepstrum* do filtro:

$$c_n = \begin{cases} 0 & \text{se } n < 0 \\ \log|G| & \text{se } n = 0 \\ \sum_{k=1}^{p} \frac{\alpha_k^n}{n} & \text{se } n > 0. \end{cases} \tag{2.33}$$

Pode-se mostrar também que existe uma relação de recorrência entre o *cepstrum* e os coeficientes de predição linear [12], o que evita o cálculo dos pólos. Essa relação é:

$$c_n = \begin{cases} 0 & \text{se } n < 0 \\ \log|G| & \text{se } n = 0 \\ a_n + \sum_{k=1}^{n-1} \frac{k}{n} c_k a_{n-k} & \text{se } n > 0, \end{cases} \tag{2.34}$$

e a relação inversa é:

$$a_n = c_n - \sum_{k=1}^{n-1} \frac{k}{n} c_k a_{n-k} \quad 1 \leq n \leq p. \tag{2.35}$$

Uma das vantagens desse método é permitir a conversão entre as duas representações. O *cepstrum* é melhor para a realização da conversão em si, mas o filtro definido pela predição linear é mais eficiente para a obtenção da excitação a partir do sinal e para a síntese após a conversão. O filtro definido pelo *cepstrum* não é diretamente implementável por um filtro, pois envolve uma exponenciação. Essa exponenciação pode ser aproximada, como será visto na Subseção 4.2.1, mas este método possui um custo computacional maior.

2.2.3 Escalas Empenadas

As representações vistas anteriormente possuem uma resolução constante no domínio da frequência. A audição humana, entretanto, possui uma resolução maior em frequências baixas. Utilizar uma representação que possua uma característica similar permite concentrar o poder de representação na parte perceptivamente mais relevante do espectro.

Escalas que procuram imitar tal resolução são chamadas de escalas perceptivas. Existem várias escalas desse tipo, como a mel [17], bark [18] ou ERB (*equivalent rectangular bandwidth*) [19]. A escala usada nesse trabalho será a escala mel, definida por:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right), \tag{2.36}$$

sendo f a frequência em hertz.

As escalas perceptivas podem ser incluídas no conceito de escala de frequência empenada (ou simplesmente escala empenada), qualquer escala gerada por um mapeamento da escala linear. Técnicas de processamento de sinais em escalas empenadas se baseiam na substituição do atraso unitário z^{-1} por um filtro passa-tudo [20][21]:

$$\tilde{z}^{-1} = \Phi(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \tag{2.37}$$

em que α é o fator de empenamento. A Figura 2.2 mostra o diagrama desse filtro.

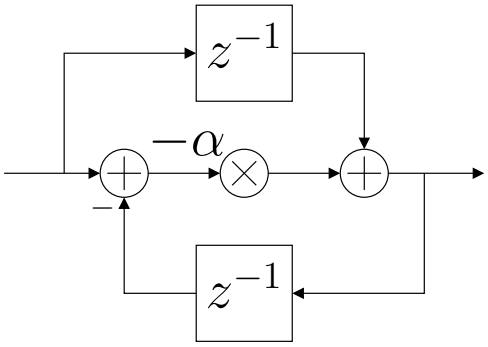

Figura 2.2: Diagrama do filtro passa-tudo que substitui o atraso.

O mapeamento na frequência é dado pela fase desse filtro:

$$\tilde{\omega} = \beta(\omega) = \arg(\Phi(z)) = \tan^{-1}\left(\frac{(1-\alpha^2)\sin(\omega)}{(1+\alpha^2)\cos(\omega) - 2\alpha}\right). \tag{2.38}$$

Escolhendo-se valores apropriados de α é possível aproximar diversas escalas perceptivas. A Tabela 2.1 mostra valores de α para diversas escalas [22], e a Figura 2.3 mostra como uma dessas escolhas aproxima a escala mel.

	Frequência de amostragem					
	8 kHz	10 kHz	12 kHz	16 kHz	20 kHz	22,05 kHz
Escala mel	0,31	0,35	0,37	0,42	0,44	0,45
Escala bark	0,42	0,47	0,50	0,55	—	—

Tabela 2.1: Valores do fator de empenamento α para diferentes escalas e frequências de amostragem.

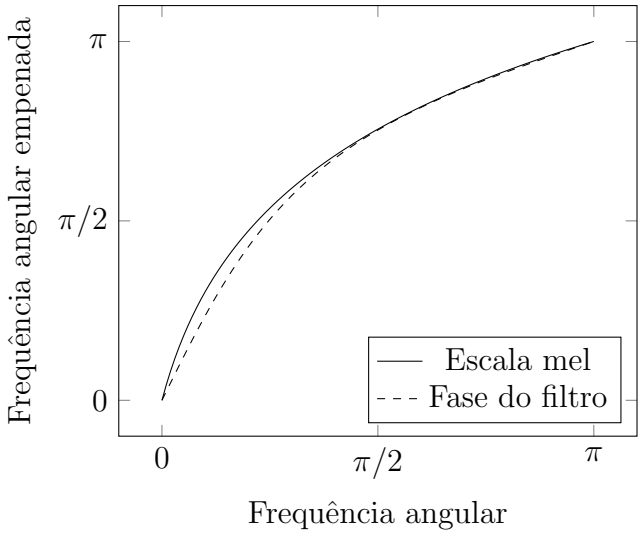

Figura 2.3: Aproximação da escala mel pela fase de um filtro.

A definição de $\tilde{z}^{-1}$ permite definir versões das análises por predição linear e por *cepstrum* em uma escala empenada. Assim, o filtro definido pela predição linear fica:

$$H(z) = \frac{\tilde{G}}{1 - \sum_{k=1}^{p} \tilde{a}_k \tilde{z}^{-k}}, \tag{2.39}$$

enquanto o filtro definido pelo *cepstrum* é:

$$H(z) = \exp\left(\sum_{k=0}^{L} \tilde{c}_k \tilde{z}^{-k}\right). \tag{2.40}$$

Nessas equações, os coeficientes $\tilde{a}_k$ e $\tilde{c}_k$ são calculados na escala empenada. Existem duas maneiras de se obter as versões empenadas dos coeficientes: calculando-os diretamente ou obtendo-os a partir de suas versões na escala linear. Para o *cepstrum*, o cálculo direto pode ser obtido a partir da sua definição:

$$\tilde{c}_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log\{H(\mathrm{e}^{j\omega})\} \mathrm{e}^{j\tilde{\omega}n} d\tilde{\omega}, \tag{2.41}$$

em que é utilizada uma versão empenada da transformada de Fourier inversa. Para a predição linear a relação é menos direta, mas também existe [20]. O cálculo de $\tilde{c}_k$ diretamente do sinal possui os mesmos problemas do cálculo de c_k diretamente do sinal, já discutidos anteriormente, e admite medidas similares para mitigá-los. Em particular, a minimização de (2.24) também pode ser usada para obtê-los.

O *cepstrum* na escala empenada também pode ser obtido a partir dos coeficientes de predição linear na escala linear [22]. Para isso, primeiramente são obtidos os

coeficientes de predição linear na escala empenada pela relação recursiva:

$$\tilde{a}_k^{(i)} = \begin{cases} a_{-i} + \alpha\tilde{a}_0^{(i-1)} & \text{se } k = 0 \\ (1-\alpha^2)\tilde{a}_0^{(i-1)} + \alpha\tilde{a}_1^{(i-1)} & \text{se } k = 1 \\ \tilde{a}_{k-1}^{(i-1)} + \alpha(\tilde{a}_k^{(i-1)} + \tilde{a}_{k-1}^{(i)}), & \text{se } k > 1 \end{cases}$$
$$i = -p, \cdots, -1, 0 \tag{2.42}$$

$$\begin{aligned} \tilde{G} &= G/\tilde{a}_0^{(0)}, \\ \tilde{a}_k &= \tilde{a}_k^{(i)}/\tilde{a}_0^{(0)}. \end{aligned} \tag{2.43}$$

A partir dos $\tilde{a}_k$, é possível calcular os $\tilde{c}_k$ a partir da relação:

$$\tilde{c}_k = \begin{cases} \log(\tilde{G}) & \text{se } k = 0 \\ -\tilde{a}_k - \sum_{i=1}^{k-1} \frac{i}{k}\tilde{c}_i\tilde{a}_{k-i} & \text{se } k > 0. \end{cases} \tag{2.44}$$

A equação (2.42) pode ser usada para calcular uma sequência infinita de $\tilde{a}_k$, mas como para o cálculo de L coeficientes $\tilde{c}_k$ são necessários L coeficientes $\tilde{a}_k$, a sequência pode ser truncada sem introduzir erros adicionais.

As relações acima transformam a_k em $\tilde{a}_k$ e, em seguida, $\tilde{a}_k$ em $\tilde{c}_k$. É possível também seguir um caminho diferente, que transforma a_k em c_k e c_k em $\tilde{c}_k$. O problema dessa versão é que a transformação de a_k em c_k envolve um truncamento, o que prejudica a qualidade da transformação seguinte. Na versão apresentada também há um truncamento ao se limitar o cálculo apenas aos L primeiros índices, mas como ele só ocorre no final o erro não é propagado.

Neste trabalho, a escala empenada utilizada aproxima a escala mel, e o *cepstrum* nessa escala é chamado de mel-*cepstrum*. Ele é a representação escolhida para realizar a conversão, pois preserva as características de interpolação do *cepstrum* e concentra a representação na parte mais relevante do espectro para a audição humana. O método utilizado para se obter o mel-*cepstrum* é a partir dos coeficientes de predição linear, pois isso permite utilizar o filtro de predição nas etapas em que ele é mais eficiente.

Capítulo 3

Técnicas para Conversão da Envoltória Espectral

O capítulo anterior apresentou representações para a envoltória espectral do sinal da fala, que contém boa parte da informação usada para a identificação de falante. Neste capítulo serão exploradas técnicas para mapear essas representações, transformando as representações características de um falante fonte nas de um falante alvo, realizando a conversão desejada. A Seção 3.1 apresenta a conversão por modelo de misturas gaussianas, uma das técnicas pioneiras na área. As Seções 3.2 e 3.3 apresentam respectivamente as conversões por modelos ocultos de Markov e por redes neurais *feed-foward*, técnicas da literatura que tentam melhorar a técnica baseada em modelo gaussiano de maneiras diferentes. Finalmente, a Seção 3.4 introduz uma conversão por redes neurais recorrentes, uma técnica original que busca combinar as melhorias introduzidas pelas duas técnicas anteriores.

3.1 Modelos de Misturas Gaussianas

Um dos primeiros métodos para conversão de fala utilizando a envoltória espectral foi o trabalho de Abe *et al.* [1]. Nele é usada quantização vetorial para criar um mapeamento entre os falantes fonte e alvo. O problema principal dessa técnica é que a discretização do espaço de parâmetros dos falantes causa descontinuidades no sinal convertido. Algumas variações dessa técnica, como quantização vetorial *fuzzy* [23], procuraram resolver esse problema. A solução que obteve mais sucesso foi o trabalho de Stylianou *et al.* [2], que introduz uma técnica baseada em modelo de misturas gaussianas, e possui tanto representação quanto transformação contínuas. É uma variação dessa técnica, introduzida em [24], que será analisada nesta seção.

Um modelo de misturas gaussianas (*Gaussian mixture model*, GMM) é um modelo probabilístico que representa uma distribuição arbitrária como uma soma de

distribuições gaussianas ponderadas. Ele é definido como:

$$p(\mathbf{x} \mid \lambda) = \sum_{i=1}^{K} P(\mathcal{C}_k)\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{3.1}$$

Cada gaussiana que compõe a mistura é chamada de componente e a ordem do modelo K é o número dessas componentes. $P(\mathcal{C}_k)$ é o peso da componente $\mathcal{C}_k$ na mistura e obedece a restrição $\sum_{i=1}^{K} P(\mathcal{C}_k) = 1$, e $\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ é a k-ésima distribuição normal D-dimensional, dada por:

$$\mathcal{N}(\mathbf{x}, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi)^{\frac{D}{2}} \mid \boldsymbol{\Sigma}_k \mid^{\frac{1}{2}}} \mathrm{e}^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)}, \tag{3.2}$$

sendo $\boldsymbol{\mu}_k$ seu vetor de média e $\boldsymbol{\Sigma}_k$ sua matriz de autocovariância. A mistura como um todo é definida pelo conjunto $\lambda = \{P(\mathcal{C}_k), \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}, k = 1, \ldots, K$.

A Figura 3.1 ilustra como um GMM usa gaussianas para compor uma distribuição mais complexa. Ela mostra o histograma de um conjunto de dados, no caso o primeiro componente *mel-cepstral* de um falante, cada uma das componentes e a distribuição resultante.

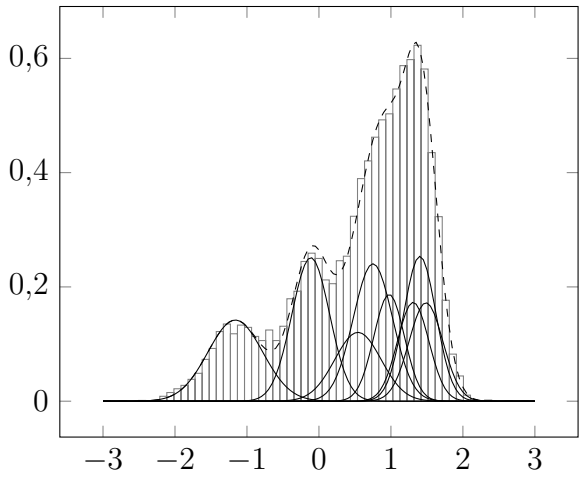

Figura 3.1: Exemplo de um GMM modelando um conjunto de dados.

A motivação para o uso do modelo de misturas gaussianas vem da suposição que a distribuição estatística dos parâmetros da voz de um falante carrega informação sobre sua identidade. Por essa suposição, se os parâmetros obtidos de um falante fonte forem modificados de forma que a sua distribuição fique parecida com a de um falante alvo, a identidade do sinal sintetizado a partir desses parâmetros também será alterada.

O uso de um modelo de misturas implica a classificação das amostras em classes. Para alguns tipos de dados essas classes são apenas construções matemáticas que não possuem nenhuma interpretação mais profunda, mas para muitos tipos de dados essas classes representam algum aspecto do processo que deu origem aos dados. No caso de parâmetros obtidos da fala, essas classes estão correlacionadas a fenômenos acústicos [25].

A função de transformação que é aplicada ao vetor de parâmetros do falante fonte é calculada a partir de uma base contendo N pares $(\mathbf{x}_n, \mathbf{y}_n)$. Seu cálculo se baseia em uma analogia com o caso monogaussiano [2]. Se os vetores $\mathbf{x}_n$ e $\mathbf{y}_n$ são ambos descritos por uma única gaussiana, então a transformação que minimiza o erro quadrático

$$\epsilon = \sum_{n=1}^{N} \|\mathbf{y}_n - \mathcal{F}(\mathbf{x}_n)\|^2 \tag{3.3}$$

é dada por:

$$\hat{\mathbf{y}}_n = \mathcal{F}(\mathbf{x}_n) = \mathrm{E}(\mathbf{y}_n \mid \mathbf{x}_n) = \boldsymbol{\mu}^{(y)} + \boldsymbol{\Sigma}^{(xy)} \left(\boldsymbol{\Sigma}^{(xx)}\right)^{-1} (\mathbf{x}_n - \boldsymbol{\mu}^{(x)}), \tag{3.4}$$

onde $\boldsymbol{\mu}^{(x)}$ e $\boldsymbol{\mu}^{(y)}$ são as médias de x e y, $\boldsymbol{\Sigma}^{(xx)}$ é a matriz de autocovariância de x e $\boldsymbol{\Sigma}^{(xy)}$ é a matriz de covariância cruzada entre x e y. Para uma mistura de gaussianas, a transformação fica então:

$$\hat{\mathbf{y}}_n = \sum_{i=1}^{K} \mathrm{P}(\mathcal{C}_k \mid \mathbf{x}_n) \left[\boldsymbol{\mu}_k^{(y)} + \boldsymbol{\Sigma}_k^{(xy)} \left(\boldsymbol{\Sigma}_k^{(xx)}\right)^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k^{(x)})\right]. \tag{3.5}$$

Essa função pode ser entendida como a transformação de cada uma das componentes como descrito na equação (3.4) ponderada pela probabilidade de $\mathbf{x}_n$ pertencer àquela componente.

Todos os parâmetros da conversão podem ser obtidos a partir do cálculo de um único modelo de mistura, o modelo de $\mathbf{z} = \left[\mathbf{x}^\top \; \mathbf{y}^\top\right]^\top$ [24]. Nesse modelo, as médias e covariâncias ficam

$$\boldsymbol{\mu}_k^{(z)} = \begin{bmatrix} \boldsymbol{\mu}_k^{(x)} \\ \boldsymbol{\mu}_k^{(y)} \end{bmatrix}, \tag{3.6}$$

$$\boldsymbol{\Sigma}_k^{(zz)} = \begin{bmatrix} \boldsymbol{\Sigma}_k^{(xx)} & \boldsymbol{\Sigma}_k^{(xy)} \\ \boldsymbol{\Sigma}_k^{(yx)} & \boldsymbol{\Sigma}_k^{(yy)} \end{bmatrix}. \tag{3.7}$$

O cálculo do modelo em si é feito pelo critério da máxima verossimilhança (*maximum likehood*, ML). A estimativa ML busca achar o conjunto $\lambda = \{P(\mathcal{C}_k), \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}, k = 1, \ldots, K$, que maximiza

$$p(\mathbf{Z} \mid \lambda) = \prod_{n=1}^{N} p(\mathbf{z}_n \mid \lambda), \tag{3.8}$$

em que $\mathbf{Z}$ é o conjunto de todos os $\mathbf{z}_n$. Essa é uma função não linear em λ, que pode ser resolvida iterativamente pelo algoritmo *expectation maximization* (EM). A cada iteração os parâmetros são recalculados pelas seguintes equações [26] [25]:

$$P(\mathcal{C}_k) = \frac{1}{N} \sum_{n=1}^{N} p_k(n), \tag{3.9}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} p_k(n)\mathbf{z}_n}{\sum_{n=1}^{N} p_k(n)}, \tag{3.10}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} p_k(n) \left[\mathbf{z}_n - \boldsymbol{\mu}_k\right] \left[\mathbf{z}_n - \boldsymbol{\mu}_k\right]^\top}{\sum_{n=1}^{N} p_k(n)}, \tag{3.11}$$

em que $p_k(n)$ é a probabilidade *a posteriori* da classe, dada por:

$$p_k(n) = P(\mathcal{C}_k \mid \mathbf{z}_n, \lambda) = \frac{P(\mathcal{C}_k)\mathcal{N}(\mathbf{z}_n, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k=1}^{M} P(\mathcal{C}_k)\mathcal{N}(\mathbf{z}_n, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}. \tag{3.12}$$

Essas fórmulas garantem que a cada iteração a verossimilhança aumenta. O algoritmo EM consiste, então, em iterar os parâmetros até que esse aumento da verossimilhança fique menor que um limiar. É importante notar que esse algoritmo só garante a convergência para um ponto estacionário da verossimilhança [27], e portanto é sensível às condições iniciais. Felizmente, para o caso de modelagem de falante, a inicialização causa pouca diferença no desempenho [25]. Este trabalho inicia as covariâncias com a identidade, as médias com amostras de $\mathbf{Z}$ escolhidas aleatoriamente e as probabilidades de cada componente iguais.

Outra questão importante é a eficiência. O cálculo da variância possui um custo computacional relativamente grande. Esse cálculo pode ser acelerado utilizando-se matrizes diagonais. Para a modelagem de um *cepstrum*, essa simplificação é justificada pelo fato de as covariâncias entre os *cepstra* em diferentes quefrências serem naturalmente pequenas [28].

3.2 Modelos Ocultos de Markov

Uma das características da conversão utilizando GMM é que cada par $(\mathbf{x}_n, \mathbf{y}_n)$ é independente de todos os pares em diferentes instantes de tempo. Como consequência disso, o modelo não leva em conta nenhuma característica temporal das sequências de parâmetros da fala. É de se esperar que isso seja uma deficiência do modelo, já que a fala é inerentemente sequencial.

Diversas propostas da literatura tentam remediar essa deficiência. A utilização de parâmetros dinâmicos, usualmente a primeira e segunda derivada dos parâmetros (comumente chamados de parâmetros delta e delta-delta, respectivamente), é uma adição comum a várias propostas, mas eles só modelam dinâmicas locais. Toda *et al.* [8] utiliza um modelo de trajetória dos parâmetros, enquanto várias propostas utilizam modelos ocultos de Markov [29][30][31]. Nesse trabalho vamos analisar a técnica de Quiao *et al.* [7], também baseada em modelos ocultos de Markov e que tenta resolver alguns problemas das anteriores.

Uma cadeia de Markov é um processo estocástico discreto no tempo e no espaço de estados que obedece a propriedade de Markov. Essa propriedade se refere ao fato de que previsões sobre o futuro do processo podem ser feitas usando apenas a informação do presente, sendo o futuro independente do passado quando se conhece o presente; dizendo-se de outra maneira, a estatística da variável em um instante qualquer depende apenas da variável no instante imediatamente anterior. A Figura 3.2 mostra uma representação esquemática de uma cadeia de Markov por meio de um grafo. Nesse grafo, os vértices representam os estados que a variável aleatória pode assumir, e as arestas representam a probabilidade de transição de um estado para outro.

Uma cadeia de Markov é definida por sua matriz de transição de estados:

$$A = \begin{bmatrix} P(s_t = 1 \mid s_{t-1} = 1) & P(s_t = 1 \mid s_{t-1} = 2) & \cdots & P(s_t = 1 \mid s_{t-1} = K) \\ P(s_t = 2 \mid s_{t-1} = 1) & P(s_t = 2 \mid s_{t-1} = 2) & \cdots & P(s_t = 2 \mid s_{t-1} = K) \\ \vdots & \vdots & \ddots & \vdots \\ P(s_t = K \mid s_{t-1} = 1) & P(s_t = K \mid s_{t-1} = 2) & \cdots & P(s_t = K \mid s_{t-1} = K) \end{bmatrix}, \tag{3.13}$$

onde cada elemento a_{ij} é da probabilidade de s_t, o estado que a cadeia assume no instante t, possuir valor j sabendo-se que no instante anterior o estado possuiu valor i. Para cadeias homogêneas no tempo essa probabilidade não muda com o tempo, e portanto é valida para todo t.

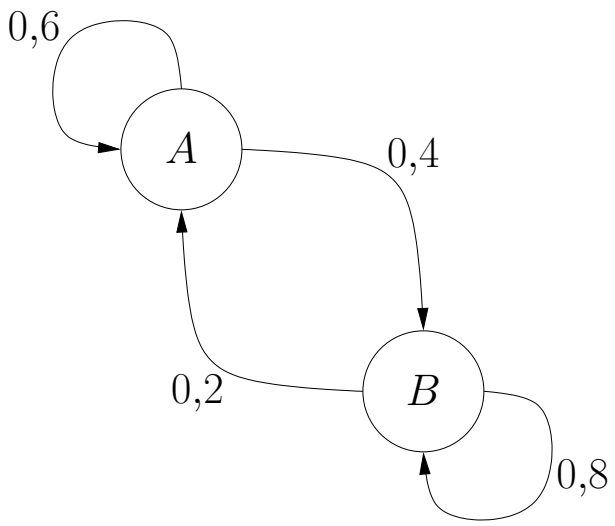

Figura 3.2: Uma cadeia de Markov de dois estados.

A partir dessa matriz, pode-se usar as estatísticas de um instante qualquer para descobrir as estatísticas de qualquer instante posterior. Um instante pode ser escolhido como instante inicial, e assim são definidas as condições iniciais da cadeia:

$$\boldsymbol{\pi} = \begin{bmatrix} P(s_1 = 1) \\ P(s_1 = 2) \\ \vdots \\ P(s_1 = K) \end{bmatrix}. \tag{3.14}$$

A partir da condição inicial e da matriz de transição, é possível calcular a estatística de todos os instantes futuros. Assim a probabilidade de uma sequência $\mathbf{S} = [s_1, s_2, \cdots, s_T]$ é:

$$P(\mathbf{S}) = P(s_1) \prod_{t=2}^{T} P(s_t \mid s_{t-1}). \tag{3.15}$$

Um modelo oculto de Markov (*hidden Markov model*, HMM) é um modelo de Markov em que os estados não podem ser observados diretamente, e o que pode ser observado é uma variável aleatória cuja distribuição depende do estado. Essa distribuição, chamada probabilidade de emissão, pode ser discreta (por exemplo, descrevendo um conjunto de símbolos possíveis), ou contínua (por exemplo, uma distribuição gaussiana). A Figura 3.3 mostra um exemplo de HMM.

Um HMM é definido pela matriz de transição de estados, assim como o Modelo de Markov convencional, e também as funções de emissão de estado $p_i(\mathbf{x}_t) = P(\mathbf{x}_t \mid s_t = i)$. Cada uma representa a probabilidade da observação $\mathbf{x}$ conhecendo-se o estado

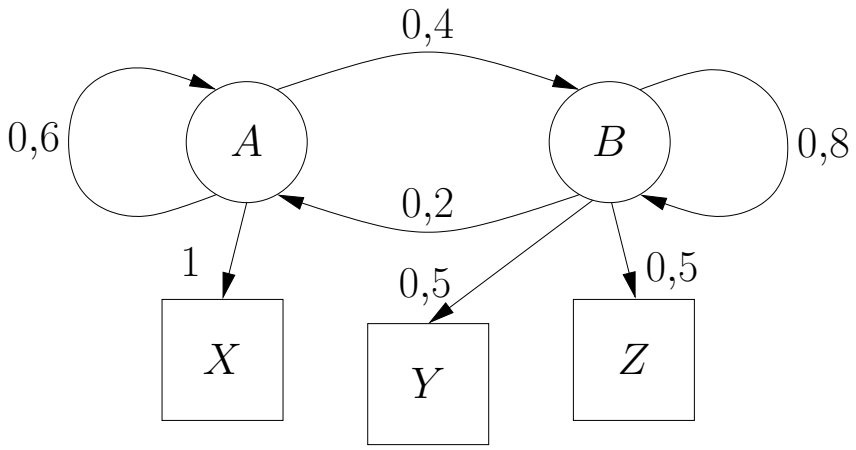

Figura 3.3: Um modelo oculto de Markov com dois estados e três emissões possíveis.

oculto s. A probabilidade de uma sequência de observações $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]$ e uma sequência associada de estados ocultos $\mathbf{S} = [s_1, s_2, \cdots, s_T]$ é:

$$P(\mathbf{X}, \mathbf{S}) = P(s_1) \prod_{t=2}^{T} P(s_t \mid s_{t-1}) \prod_{t=1}^{T} P(\mathbf{x}_t \mid s_t). \tag{3.16}$$

Um HMM pode ser visto como uma extensão de um modelo de mistura em que os estados ocultos fazem o papel das componentes. Em particular, um HMM com emissões gaussianas se reduz a um GMM se $P(s_t = j \mid s_{t-1} = i) = P(s_t = j) = P(\mathcal{C}_j)$, ou seja, todos os estados possuírem probabilidade de transição para um determinado estado iguais. Isso elimina a característica temporal, pois torna cada estado independente do estado anterior da sequência.

Para a conversão, será utilizado um HMM de K estados e emissões gaussianas. Assim, a probabilidade do vetor fonte é dada por:

$$P(\mathbf{x}_t \mid s_t = k) = \mathcal{N}(\mathbf{x}_t, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{3.17}$$

e o vetor alvo possui distribuição:

$$P(\mathbf{y}_t \mid s_t = k) = \mathcal{N}(\mathbf{y}_t, \boldsymbol{\nu}_k, \boldsymbol{\Gamma}_k). \tag{3.18}$$

O estado oculto controla então duas emissões, que correspondem aos falantes fonte e alvo. A transformação que minimiza o erro quadrático dado um estado é então novamente dada pela equação (3.4):

$$\begin{aligned} \mathrm{E}[P(\mathbf{y}_t \mid \mathbf{x}_t, s_t = k)] &= \boldsymbol{\nu}_k + \boldsymbol{\Gamma}_k \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_k) \\ &= \mathbf{B}_k \mathbf{x}_t + \mathbf{b}_k. \end{aligned} \tag{3.19}$$

Tomando-se a sequência fonte $\mathbf{X}$, a probabilidade de um vetor alvo é:

$$P(\mathbf{y}_t \mid \mathbf{X}) = \sum_{\mathbf{S} \in \mathbb{S}} P(\mathbf{y}_t, \mathbf{S} \mid \mathbf{X}) = \sum_{\mathbf{S} \in \mathbb{S}} P(\mathbf{y}_t \mid \mathbf{S}, \mathbf{X}) P(\mathbf{S} \mid \mathbf{X}), \tag{3.20}$$

onde $\mathbb{S}$ é o conjunto de todas as sequências de estado $\mathbf{S}$ possíveis.

Pela equação (3.19), dado um s_t, $\mathbf{y}_t$ só depende do vetor fonte $\mathbf{x}_t$:

$$P(\mathbf{y}_t \mid \mathbf{S}, \mathbf{X}) = P(\mathbf{y}_t \mid \mathbf{x}_t, s_t). \tag{3.21}$$

Assim, podemos simplificar a equação (3.20):

$$\begin{aligned} \sum_{\mathbf{S} \in \mathbb{S}} P(\mathbf{y}_t \mid \mathbf{S}, \mathbf{X}) P(\mathbf{S} \mid \mathbf{X}) &= \sum_{s_t} P(\mathbf{y}_t \mid \mathbf{x}_t, s_t) \sum_{\mathbf{S}^{/s_t}} P(\mathbf{S}^{/s_t} \mid \mathbf{X}) \\ &= \sum_{k=1}^{K} P(\mathbf{y}_t \mid \mathbf{x}_t, s_t = k) P(s_t = k \mid \mathbf{X}), \end{aligned} \tag{3.22}$$

onde $\mathbf{S}^{/s_t} = [s_1, \cdots, s_{t-1}, s_{t+1}, \cdots, s_T]$

O mapeamento da sequência $\mathbf{X}$ no vetor $\mathbf{y}_t$ é dado pelo valor esperado de $P(\mathbf{y}_t, \mid \mathbf{X})$:

$$\hat{\mathbf{y}}_t = E[P(\mathbf{y}_t \mid \mathbf{X})] = \sum_{k=1}^{K} P(s_t = k \mid \mathbf{X})(\mathbf{B}_k \mathbf{x}_t + \mathbf{b}_k). \tag{3.23}$$

A probabilidade $P(s_t = k \mid \mathbf{X})$ pode ser calculada pelo algoritmo *forward–backward* [32][33][34]:

$$\begin{aligned}\alpha_t(i) &= P(s_t = i, \mathbf{X}_{1:t}) \\ &= P(\mathbf{x}_t \mid s_t = i) \sum_{j=1}^{M} P(s_t = i \mid s_{t-1} = j)\alpha_{t-1}(i), \qquad (3.24)\\ \beta_t(i) &= P(\mathbf{X}_{t+1:T} \mid s_t = i) \\ &= \sum_{j=1}^{M} P(\mathbf{x}_{t+1} \mid s_{t+1} = j)P(s_{t+1} = j \mid s_t = i)\beta_{t+1}(j), \qquad (3.25)\end{aligned}$$

$$P(s_t = i \mid \mathbf{X}) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{M} \alpha_t(j)\beta_t(j)}, \qquad (3.26)$$

onde $\mathbf{X}_{1:t} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_t]$.

Os parâmetros do modelo e os parâmetros $\mathbf{B}_k$ e $\mathbf{b}_k$ da transformação são estimados sobre uma base contendo N pares de sequências alinhadas $(\mathbf{X}^{(n)}, \mathbf{Y}^{(n)})$, onde $\mathbf{X}^{(n)} = \left[\mathbf{x}_1^{(n)}, \cdots, \mathbf{x}_{T_n}^{(n)}\right]$ e $\mathbf{Y}^{(n)} = \left[\mathbf{y}_1^{(n)}, \cdots, \mathbf{y}_{T_n}^{(n)}\right]$.

Os parâmetros do HMM, $P(s_t = j \mid s_{t-1} = i)$, $P(s_1 = i)$ e $P(\mathbf{x}_t \mid s_t = i)$ podem ser estimados utilizando-se *expectation maximization*, por um método chamado algoritmo Baum-Welch [32][33][34].

O algoritmo primeiramente utiliza o modelo atual λ para calcular os valores:

$$E[N_{1k}] = \sum_{n=1}^{N} P(s_1^{(n)} = k \mid \mathbf{X}^{(n)}, \lambda), \qquad (3.27)$$

$$E[N_{jk}] = \sum_{n=1}^{N} \sum_{t=1}^{T_n} P(s_{t-1}^{(n)} = j, s_t^{(n)} = k \mid \mathbf{X}^{(n)}, \lambda), \qquad (3.28)$$

$$E[N_k] = \sum_{n=1}^{N} \sum_{t=1}^{T_n} P(s_t^{(n)} = k \mid \mathbf{X}^{(n)}, \lambda). \qquad (3.29)$$

Esses valores são os valores esperados da quantidade de vezes que seus respectivos eventos ocorrem na base de dados. $E[N_{1k}]$ é o valor esperado da quantidade de vezes que o estado inicial é k, $E[N_{jk}]$ é o valor esperado do número de transições do estado j para o estado k e $E[N_k]$ é o valor esperado do número de vezes que o estado k ocorre, independentemente do estado anterior. $E[N_{1k}]$ e $E[N_k]$ podem ser calculados pela equação (3.26), e $E[N_{jk}]$ pode ser calculado utilizando-se as equações (3.24) e (3.25):

$$P(s_{t-1}^{(n)} = j, s_t^{(n)} = k \mid \mathbf{X}^{(n)}, \lambda) = \alpha_{t-1}^{(n)} P(s_t = k \mid s_{t-1} = j) P(\mathbf{x}_t^{(n)} \mid s_t = i) \beta_t^{(n)}. \tag{3.30}$$

Esses valores podem ser usados para calcular os parâmetros do novo modelo $\hat{\lambda}$:

$$P(s_t = k \mid s_{t-1} = j) = \frac{E[N_{jk}]}{E[N_k]}, \tag{3.31}$$

$$P(s_1 = i) = \frac{E[N_{1k}]}{N}, \tag{3.32}$$

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^{N} \sum_{t=1}^{T_n} P(s_t^{(n)} = k \mid \mathbf{X}^{(n)}, \lambda) \mathbf{x}_t^{(n)}}{E[N_k]}, \tag{3.33}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^{N} \sum_{t=1}^{T_n} P(s_t^{(n)} = k \mid \mathbf{X}^{(n)}, \lambda) (\mathbf{x}_t^{(n)} - \boldsymbol{\mu}_k)(\mathbf{x}_t^{(n)} - \boldsymbol{\mu}_k)^\top}{E[N_k]}. \tag{3.34}$$

Assim como no cálculo dos parâmetros da GMM, essas operações garantem que a verossimilhança do modelo cresce a cada iteração. O algoritmo repete então esses cálculos até que a verossimilhança pare de crescer.

Os parâmetros $\mathbf{B}_k$ e $\mathbf{b}_k$ podem ser estimados por otimização de mínimos quadrados:

$$\min_{\{B_k, b_k\}} \sum_{n=1}^{N} \sum_{t=1}^{T_n} \left| \mathbf{y}_t^{(n)} - \sum_{m=1}^{K} P(s_t^{(n)} = k \mid \mathbf{X}^{(n)}) (\mathbf{B}_k \mathbf{x}_t^{(n)} + \mathbf{b}_k) \right|^2, \tag{3.35}$$

que pode ser resolvido diretamente. Por simplicidade, são introduzidas as variáveis auxiliares:

$$\hat{\mathbf{x}}_{k,t}^{(n)} = P(s_t^{(n)} = k \mid \mathbf{X}^{(n)}) \begin{bmatrix} \mathbf{x}_t^{(n)} \\ 1 \end{bmatrix}, \tag{3.36}$$

$$A_k = \begin{bmatrix} \mathbf{B}_k & \mathbf{b}_k \end{bmatrix}, \tag{3.37}$$

de forma que $A_k \hat{\mathbf{x}}_{t,k}^{(n)} = P(s_t = k \mid \mathbf{X}^{(n)})(\mathbf{B}_k \mathbf{x}_t^{(n)} + \mathbf{b}_k)$. Além disso, também são utilizadas as notações:

$$\hat{\mathbf{X}}_k^{(n)} = \begin{bmatrix} \mathbf{x}_{k,1}^{(n)} & \cdots & \mathbf{x}_{k,T_n}^{(n)} \end{bmatrix}, \tag{3.38}$$

$$\hat{\mathbf{X}}_k = \begin{bmatrix} \mathbf{X}_k^{(1)} & \cdots & \mathbf{X}_k^{N} \end{bmatrix}, \tag{3.39}$$

$$\hat{\mathbb{X}} = \begin{bmatrix} \mathbf{X}_1^{\top} & \cdots & \mathbf{X}_K^{\top} \end{bmatrix}^{\top}, \tag{3.40}$$

$$\mathbf{Y}^{(n)} = \begin{bmatrix} \mathbf{y}_1^{(n)} & \cdots & \mathbf{y}_{T_n}^{(n)} \end{bmatrix}, \tag{3.41}$$

$$\mathbb{Y} = \begin{bmatrix} \mathbf{Y}^{(1)} & \cdots & \mathbf{Y}^{N} \end{bmatrix}. \tag{3.42}$$

A solução de mínimos quadrados é então:

$$\begin{bmatrix} \mathbf{A}_1 & \cdots & \mathbf{A}_K \end{bmatrix} = \mathbb{Y}\hat{\mathbb{X}}^{\top} \left(\hat{\mathbb{X}}\hat{\mathbb{X}}^{\top} \right)^{-1}. \tag{3.43}$$

Essa solução é computacionalmente intensiva, pois a matriz $\hat{\mathbb{X}}$ tem tamanho $K(D+1) \times \sum_{n=1}^{N} T_n$. Para simplificar esse cálculo, podemos utilizar o de fato que $\sum_{k=1}^{K} P(s_t = k \mid \mathbf{X}) = 1$ e $P(s_t = k \mid \mathbf{X}) \geq 0$ e utilizar a desigualdade de Jensen:

$$\left| \sum_{m=1}^{M} P(s_t = k \mid \mathbf{X})(\mathbf{B}_k \mathbf{x}_t + \mathbf{b}_k) - \mathbf{y}_t \right|^2 \leq \sum_{m=1}^{M} P(s_t = k \mid \mathbf{X}) \left| (\mathbf{B}_k \mathbf{x}_t + \mathbf{b}_k) - \mathbf{y}_t \right|^2. \tag{3.44}$$

Assim, a equação (3.35) pode ser aproximada pelo seu limite superior:

$$\min_{\{B_k, b_k\}} \sum_{m=1}^{K} \sum_{n=1}^{N} \sum_{t=1}^{T_n} P(s_t^{(n)} = k \mid \mathbf{X}^{(n)}) \left| (\mathbf{B}_k \mathbf{x}_t + \mathbf{b}_k) - \mathbf{y}_t \right|^2, \tag{3.45}$$

que pode ser decomposto em K problemas de otimização independentes:

$$\min_{B_k, b_k} \sum_{n=1}^{N} \sum_{t=1}^{T_n} P(s_t^{(n)} = k \mid \mathbf{X}^{(n)}) \left| (\mathbf{B}_k \mathbf{x}_t + \mathbf{b}_k) - \mathbf{y}_t \right|^2, \tag{3.46}$$

cujas soluções são:

$$\mathbf{A}_k = \mathbb{Y}\hat{\mathbf{X}}_k^{\top} \left(\hat{\mathbf{X}}_k \hat{\mathbf{X}}_k^{\top} \right)^{-1}. \tag{3.47}$$

Essa aproximação gera um ganho de eficiência da ordem de K.

3.3 Redes Neurais *Feed-Foward*

Tanto a conversão por GMM quanto a por HMM tentam solucionar o problema da conversão utilizando transformações lineares ponderadas por probabilidades. Transformações lineares são ferramentas poderosas, mas para alguns problemas, transformações não-lineares podem ser mais adequadas, e é possível que a conversão da voz seja um desses problemas. Redes neurais são uma família de técnicas capazes de realizar mapeamentos não lineares arbitrários. Algumas propostas utilizam redes neurais para realizar a conversão [35][36][37], mas elas possuem algumas limitações, como utilizar bases de dados especialmente preparadas. Nessa seção utilizaremos a proposta de Desai *et al.* [5], que possui muitos paralelos com as técnicas anteriores, o que permite que a comparação entre elas seja mais direta.

Uma rede neural (*neural network*, NN) é uma ferramenta computacional inspirada pelo modo como neurônios se conectam no sistema nervoso de animais. Existem várias topologias possíveis para redes neurais, e uma das mais populares é a rede neural *feed-foward*, também conhecida como *perceptron* multicamada (*multilayer perceptron*, MLP). Nela, uma rede neural é composta por elementos chamados neurônios organizados em camadas. Cada neurônio de uma camada realiza uma computação com os resultados dos neurônios da camada anterior e transfere o resultado para a próxima camada. Especificamente, o resultado de um neurônio é:

$$z_j^{(l)} = \sigma_l \left(\sum_{k=1}^{K_l} w_{kj}^{(l)} z_k^{(l-1)} + w_{0j}^{(l)} \right), \tag{3.48}$$

onde $z_j^{(l)}$ é a ativação do neurônio j da camada l, $w_{kj}^{(l)}$ é peso do neurônio k da camada anterior no cálculo correspondente ao neurônio j da camada atual, $w_{0j}^{(l)}$ é um viés constante e σ_l é a chamada função de ativação da camada l.

A função de ativação possui diferentes formas, dependendo de onde na rede a camada se localiza e qual é o seu papel. Na última camada, chamada de camada de saída, ela tem o papel de formatar o resultado para o tipo de dado de saída. Ela pode ser, por exemplo, uma função linear se a saída for um vetor de números reais, ou binária se a saída for um valor lógico. Para as outras camadas, conhecidas como camadas ocultas, o papel da função de ativação é introduzir não-linearidades, tornando-as algo além de uma série de combinações lineares. Uma classe de funções que cumpre bem esse papel são as funções sigmoides, um exemplo das quais é a tangente hiperbólica $\tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}}$. As Figuras 3.4 e 3.5 mostram, respectivamente, um exemplo de neurônio e como esses neurônios podem ser organizados em uma rede.

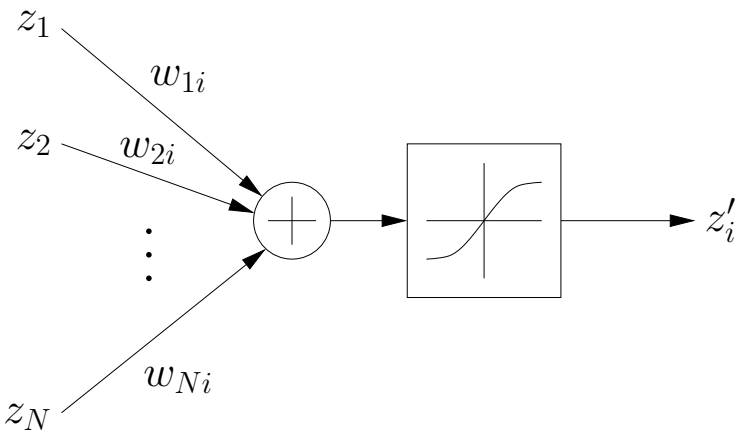

Figura 3.4: Neurônio com função de ativação sigmoide.

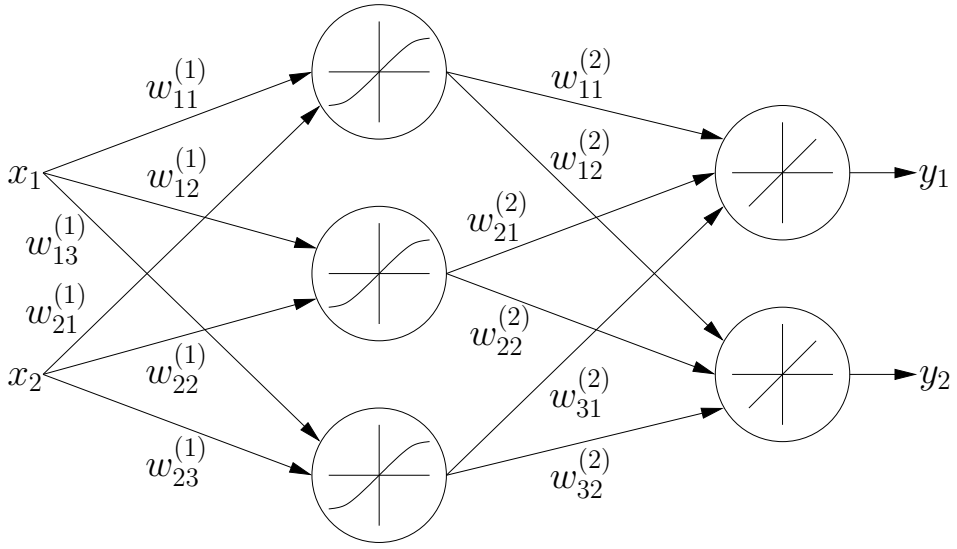

Figura 3.5: Rede neural *feed-foward* com uma camada oculta com três neurônios e ativação sigmoide e camada de saída com dois neurônios e ativação linear.

Considerando uma rede composta de duas camadas, uma oculta cuja função de ativação é uma tangente hiperbólica e uma de saída cuja função de ativação é a função identidade $f(x) = x$, a função de transferência da entrada para a saída será:

$$\mathcal{F}(\mathbf{x}) = \mathbf{W}^{(2)} \left[\tanh \left(\mathbf{W}^{(1)} \left[\mathbf{x}^\top \quad 1 \right]^\top \right)^\top \quad 1 \right]^\top , \tag{3.49}$$

em que $\mathbf{W}^{(l)}$ é uma matriz contendo os pesos e vieses correspondentes à camada l. Os vieses são incluídos na matriz $\mathbf{W}^{(l)}$ ao se adicionar uma ativação extra de valor unitário em cada camada. A função tanh é aplicada elemento a elemento do vetor. Uma rede desse tipo é um aproximador universal [38], ou seja, ela pode aproximar qualquer função contínua em um subconjunto compacto de $\mathcal{R}^n$ com precisão arbitrária dado um número grande o suficiente de neurônios na camada oculta.

Essa propriedade é o princípio por trás do uso de redes neurais para a conversão de falante: se existe uma função entre os vetores dos falantes fonte e alvo, uma rede neural pode ser usada para aproximar essa função.

O treinamento da rede utiliza uma base contendo N pares $(\mathbf{x}_n, \mathbf{y}_n)$ e se dá pelo bem conhecido algoritmo de *backpropagation* [39][40]. Nesse algoritmo, o erro de predição do resultado da rede é propagado na direção contrária à do funcionamento da rede. Isso permite associar a cada neurônio uma parcela proporcional de contribuição ao erro final. Essa parcela pode ser usada então para estimar o gradiente do erro em relação aos parâmetros da rede, que por sua vez pode ser usado para atualizar o valor desses parâmetros.

Existem diversas definições para o erro de predição, mas uma das mais comuns para redes desse tipo, que mapeiam $\mathcal{R}^n$ em $\mathcal{R}^n$, é o erro quadrático:

$$\epsilon = \frac{1}{2} \sum_{n=1}^{N} \left\| \mathbf{y}_n - \mathcal{F}(\mathbf{x}_n) \right\|^2, \tag{3.50}$$

onde $\mathcal{F}(\mathbf{x}_n)$ representa a função de transferência da rede. Esse erro também pode ser escrito na forma:

$$\epsilon = \sum_{n=1}^{N} \epsilon_n, \tag{3.51}$$

onde ϵ_n é o erro correspondente a uma amostra do conjunto de treinamento. O algoritmo *backpropagation* procura então calcular as derivadas parciais do erro em relação ao pesos:

$$\frac{\partial \epsilon}{\partial w_{ij}^{(l)}} = \sum_{n=1}^{N} \frac{\partial \epsilon_n}{\partial w_{ij}^{(l)}}. \tag{3.52}$$

A equação de ativação de um neurônio (3.48) pode ser separada em:

$$z_j^{(l)} = \sigma_l(a_j^{(l)}), \tag{3.53}$$

$$a_j^{(l)} = \sum_i w_{ij}^{(l)} z_i^{(l-1)}. \tag{3.54}$$

onde $z_i^{(l)}$ pode representar também uma entrada x_i ou uma saída $\mathcal{F}(\mathbf{x})_i$. A derivada parcial pode ser então reescrita utilizando-se a regra da cadeia como:

$$\frac{\partial \epsilon_n}{\partial w_{ij}^{(l)}} = \frac{\partial \epsilon_n}{\partial a_j^{(l)}} \frac{\partial a_j^{(l)}}{\partial w_{ij}^{(l)}}, \tag{3.55}$$

onde $\frac{\partial a_j^{(l)}}{\partial w_{ij}^{(l)}}$ é obtido derivando-se a equação (3.54):

$$\frac{\partial a_j^{(l)}}{\partial w_{ij}^{(l)}} = z_i^{(l-1)}. \tag{3.56}$$

O cálculo de $\frac{\partial \epsilon_n}{\partial a_j^{(l)}}$ depende de onde na rede o neurônio está. Para neurônios na camada de saída o cálculo é direto, e se a saída possuir ativação linear $z_j = a_j$, ela é

$$\frac{\partial \epsilon_n}{\partial a_j^{(l)}} = z_j^{(l)} - y_j; \tag{3.57}$$

a derivada em relação a w fica, então:

$$\frac{\partial \epsilon_n}{\partial w_{ij}^{(l)}} = z_i^{(l-1)}(z_j^{(l)} - y_j), \tag{3.58}$$

Nas camadas ocultas, a derivada pode ser calculada a partir das camada posteriores:

$$\frac{\partial \epsilon_n}{\partial a_j^{(l)}} = \sum_k \frac{\partial \epsilon_n}{\partial a_k^{(l+1)}} \frac{\partial \partial a_k^{(l+1)}}{\partial a_j^{(l)}}. \tag{3.59}$$

Usando as equações (3.53) e (3.54):

$$a_k^{(l+1)} = \sum_j w_{jk}^{(l+1)} \sigma_l(a_j^{(l)}), \tag{3.60}$$

$$\frac{\partial a_k^{(l+1)}}{\partial a_j^{(l)}} = \sigma_l'(a_j^{(l)}) w_{jk}^{(l)}, \tag{3.61}$$

$$\frac{\partial \epsilon_n}{\partial a_j^{(l)}} = \sigma_l'(a_j^{(l)}) \sum_k w_{jk}^{(l+1)} \frac{\partial \epsilon_n}{\partial a_k^{(l+1)}}, \tag{3.62}$$

e a derivada final fica:

$$\frac{\partial \epsilon_n}{\partial w_{ij}^{(l)}} = z_i^{(l-1)} \sigma_l'(a_j^{(l)}) \sum_k w_{jk}^{(l+1)} \frac{\partial \epsilon_n}{\partial a_k^{(l+1)}}. \tag{3.63}$$

As derivadas são calculadas sequencialmente da camada final para as camadas anteriores, no sentido inverso ao do funcionamento normal da rede, o que dá origem ao nome do algoritmo. Calculado o gradiente, os pesos podem ser atualizados utilizando-se um dos muitos método de otimização por descida por gradiente. Uma iteração do algoritmo *backpropagation* é então:

1. Calcular as ativações, da entrada para a saída;

2. Calcular as derivadas, da saída para a entrada;

3. Atualizar os pesos.

Essa processo é repetido então até que um critério de parada seja atingido, geralmente o erro de predição sobre uma parte do conjunto de dados que não foi utilizada na atualização dos pesos pare de diminuir. A razão de se utilizar um conjunto separado é impedir que a rede se adapte demais aos dados utilizados no treinamento e perca generalidade.

3.4 Redes Neurais Recorrentes

Nas seções anteriores foram apresentadas técnicas da literatura que buscam melhorar o resultado obtido utilizando GMMs introduzindo ou dependências temporais ou não linearidades na transformação. Um passo lógico é, então, a introdução de uma técnica que possua os dois tipos de melhorias. Para isso, este trabalho introduz a transformação por redes neurais recorrentes como uma contribuição original.

Uma rede neural recorrente (*recurrent neural network*, RNN) é uma rede neural em que pelo menos uma camada é realimentada com sua saída do instante de tempo anterior. Isso introduz uma dependência temporal, pois a informação circula pela camada realimentada, e pode então ser usada no cálculo dos resultados em instantes futuros. Isso a diferencia de uma rede *feed-foward*, em que o fluxo de informação é sempre em uma direção. A Figura 3.6 mostra um exemplo de rede neural recorrente.

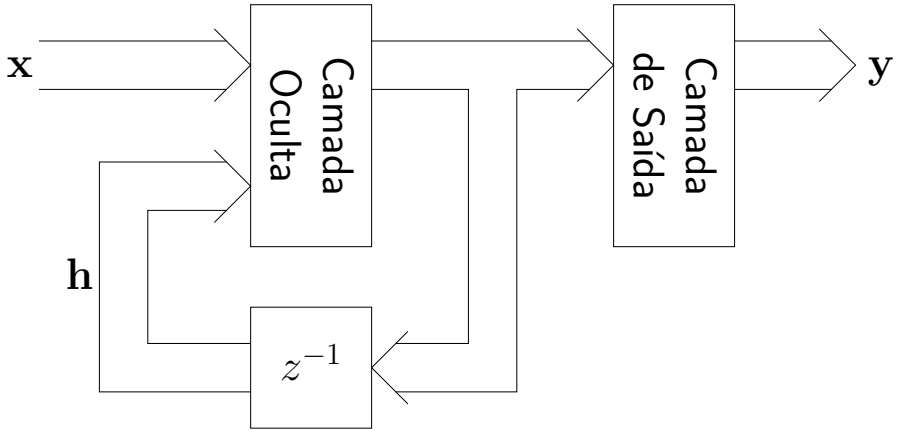

Figura 3.6: Rede neural recorrente com uma camada oculta recorrente.

A ativação de uma camada realimentada é dada por:

$$\mathbf{s}(t) = \sigma\left(\mathbf{W}_I\mathbf{x}(t) + \mathbf{W}_H\mathbf{s}(t-1) + \mathbf{w}_0\right), \tag{3.64}$$

onde $\mathbf{s}(t)$ é o chamado estado da rede. Assim como uma rede neural *feed-foward* pode aproximar qualquer função não-linear, uma rede neural recorrente pode aproximar qualquer sistema dinâmico não-linear com precisão arbitrária dado um número suficiente de neurônios na camada recorrente [41]. Assim, a conversão por RNN se baseia na suposição de que a trajetória da envoltória no tempo pode ser descrita por um sistema desse tipo.

Uma rede recorrente pode ser convertida em uma rede *feed-foward* por um processo chamado desdobramento. Nele a camada recorrente é expandida em uma sequência de camadas tradicionais idênticas, como ilustrado na Figura 3.7.

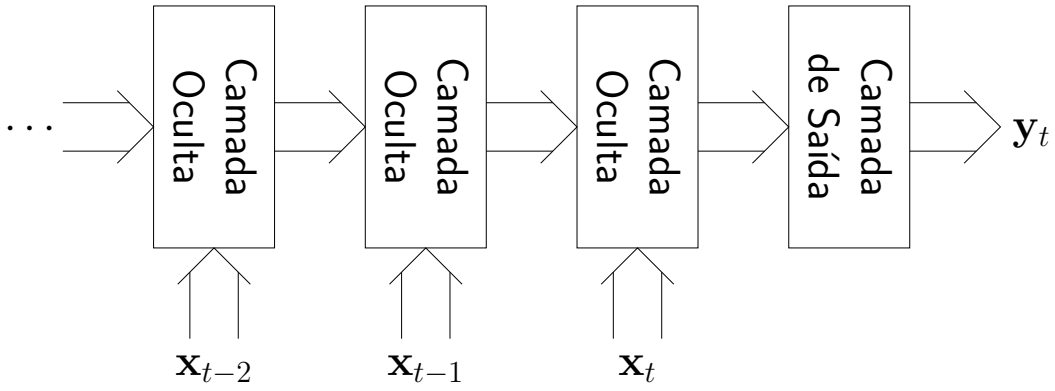

Figura 3.7: Desdobramento de uma rede recorrente.

O desdobramento exemplifica o poder computacional de uma rede recorrente quando comparada a uma rede *feed-foward*: a rede equivalente possui muito mais camadas do que é prático utilizar em uma rede *feed-foward*, o que aumenta sua capacidade de representação. Ao mesmo tempo, o fato de que o mesmo conjunto de pesos é aplicado em todos os instantes de tempo impede que o treinamento da rede se torne impraticável. O desdobramento também é a base para o algoritmo de treinamento de RNNs, chamado de *backpropagation through time* [42].

Sendo um conjunto de treinamento contendo N pares de sequências alinhadas $(\mathbf{X}^{(n)}, \mathbf{Y}^{(n)})$, onde $\mathbf{X}^{(n)} = \left[\mathbf{x}_0^{(n)}, \cdots, \mathbf{x}_{T_n}^{(n)}\right]$ e $\mathbf{Y}^{(n)} = \left[\mathbf{y}_0^{(n)}, \cdots, \mathbf{y}_{T_n}^{(n)}\right]$, o erro de predição é dado por:

$$\epsilon = \frac{1}{2}\sum_{n=1}^{N}\epsilon^{(n)} = \frac{1}{2}\sum_{n=1}^{N}\sum_{t=1}^{T_n}\epsilon_t^{(n)} = \frac{1}{2}\sum_{n=1}^{N}\sum_{t=1}^{T_n}(\mathbf{y}_t^{(n)} - \mathcal{F}(\mathbf{x}_t^{(n)}))^2. \tag{3.65}$$

Considerando uma rede como a da Figura 3.6, com uma camada recorrente e uma de saída, o treinamento envolve calcular as derivadas:

$$\frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(r)}}, \quad \frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(i)}}, \quad \frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(o)}}, \tag{3.66}$$

onde $w_{ij}^{(r)}$, $w_{ij}^{(i)}$ e $w_{ij}^{(o)}$ são, respectivamente, pesos recorrentes, de entrada e de saída.

Para a camada de saída, o cálculo é similar ao caso *feed-foward* e se dá por uma versão da equação (3.58)

$$\frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(o)}} = s_i(t)(z_j(t) - y_j(t)), \tag{3.67}$$

onde:

$$z_j(t) = \sum_k w_{kj}^{(o)} s_k(t) \tag{3.68}$$

é a saída da rede.

Para as derivadas da camada recorrente, é utilizado o desdobramento. Ao se transformar a rede recorrente em uma rede *feed-foward*, pode-se utilizar versões da equação (3.59):

$$\frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(i)}}(\tau - 1) = x_i(\tau - 1)\sigma'(a_j(\tau - 1)) \sum_k w_{jk}^{(o)} \frac{\partial \epsilon_t^{(n)}}{\partial z_k}, \quad \text{se } \tau = t \tag{3.69}$$

$$\frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(i)}}(\tau - 1) = x_i(\tau - 1)\sigma'(a_j(\tau - 1)) \sum_k w_{jk}^{(i)} \frac{\partial \epsilon_t^{(n)}}{\partial a_k(\tau)}, \quad \text{se } \tau < t \tag{3.70}$$

$$\frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(r)}}(\tau - 1) = s_i(\tau - 1)\sigma'(a_j(\tau - 1)) \sum_k w_{jk}^{(o)} \frac{\partial \epsilon_t^{(n)}}{\partial z_k}, \quad \text{se } \tau = t \tag{3.71}$$

$$\frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(r)}}(\tau - 1) = s_i(\tau - 1)\sigma'(a_j(\tau - 1)) \sum_k w_{jk}^{(r)} \frac{\partial \epsilon_t^{(n)}}{\partial a_k(\tau)}, \quad \text{se } \tau < t, \tag{3.72}$$

onde o índice τ denota qual instante de tempo essa camada representa e

$$a_j(\tau) = \sum_i w_{ij} x_i(\tau) + \sum_k w_{kj} s_k(\tau - 1); \tag{3.73}$$

as derivadas auxiliares são calculadas também de maneira similar ao caso *feed-foward*:

$$\frac{\partial \epsilon_t^{(n)}}{\partial a_k(\tau)} = \sigma'(a_j(\tau)) \sum_k w_{jk}^{(r)} \frac{\partial \epsilon_t^{(n)}}{\partial a_k(\tau+1)}, \tag{3.74}$$

$$\frac{\partial \epsilon_t^{(n)}}{\partial z_k} = z_k(t) - y_k(t). \tag{3.75}$$

Essas relações são usadas para calcular as derivadas de um número arbitrário L de instantes no passado. Para manter a restrição de que os pesos são constantes para todos os instantes de tempo, as derivadas de cada instante devem ser acumuladas:

$$\frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(r)}} = \sum_{\tau=t}^{t-L} \frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(r)}}(\tau), \tag{3.76}$$

$$\frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(i)}} = \sum_{\tau=t}^{t-L} \frac{\partial \epsilon_t^{(n)}}{\partial w_{ij}^{(i)}}(\tau). \tag{3.77}$$

Essas derivadas são calculadas para cada instante t de cada sequência n. A atualização dos pesos pode ser feita para cada instante ou as derivadas podem ser acumuladas por um número de instantes, com a atualização ocorrendo apenas no fim. Em geral, a atualização a cada instante produz resultados melhores, pois resulta em um caminho mais fino na otimização. Entretanto, ele também é mais custoso computacionalmente, pois os estados calculados não podem ser aproveitados. O processo é repetido então até que uma condição de parada seja atingida.

Capítulo 4

Considerações sobre Implementação

Os capítulos anteriores mostraram como representar o sinal da fala em um domínio em que a conversão de falante possa ser realizada e como realizar a conversão em si. Esse capítulo agora mostra como um sistema de conversão pode ser implementado na prática e apresenta diversos detalhes necessários para que as vozes convertidas possam ser de fato ouvidas. O sistema é dividido em duas partes: treinamento, apresentado na Seção 4.1, e conversão, apresentada na Seção 4.2. As respectivas subseções detalham alguns elementos que compõem o sistema. Finalmente, a Seção 4.3 apresenta detalhes sobre linguagens de programação e bibliotecas utilizadas.

4.1 Arquitetura do Treinamento

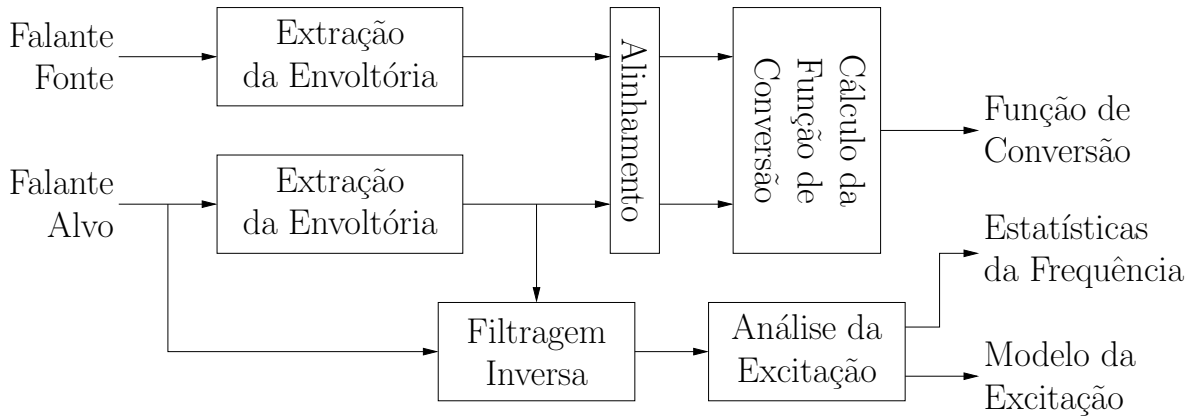

Figura 4.1: Arquitetura do sistema de treinamento.

A Figura 4.1 mostra o diagrama de blocos do processo de treinamento. O treinamento usa uma base de dados contendo um conjunto de frases faladas por ambos os falantes. Esses sinais possuem taxa de amostragem de 16 kHz, pois a maior parte da informação da fala está na faixa abaixo de 8 kHz. Os sinais são segmentados em janelas de 25 ms com 50% de sobreposição por uma janela de Hamming. Desses sinais são extraídos os coeficientes de predição linear de cada quadro, e desses coeficientes são obtidos os mel-*cepstra*, pelas equações (2.42), (2.43) e (2.44), como descrito no Capítulo 2.

A escolha da ordem da predição linear depende da frequência de amostragem. Uma regra prática é que a ordem da predição deve estar em torno de $4 + f_s/1000$, onde f_s e a frequência de amostragem em Hz [9]. Para uma frequência de 16 kHz, valores habituais estão entre 18 e 22, e nesse trabalho foi utilizada uma ordem de 20. A ordem do mel-*cepstrum* está relacionada à ordem da predição linear. Pelas equações (2.42), (2.43) e (2.44), a ordem do mel-*cepstrum* deve ser pelo menos a ordem da predição linear, e na prática deve ser maior. Valores maiores produzem menos distorções no espectro, mas valores muito altos podem introduzir informações desnecessárias para a identificação e conversão. A ordem utilizada neste trabalho foi 30.

Os sinais são alinhados temporalmente para garantir que os quadros correspondam aos mesmos trechos de fala, e os trechos não-vozeados são removidos. A razão desse descarte é que trechos vozeados são mais importantes para a identificação, e trechos não-vozeados possuem envoltórias menos bem definidas [2]. Possuindo-se as envoltórias alinhadas, a função de conversão correspondente ao método utilizado é treinada de acordo com os algoritmos apresentados no Capítulo 3.

Com as envoltórias, também é realizada a filtragem inversa do sinal, obtendo quadros do sinal de excitação. Esses quadros são analisados, obtendo-se um modelo de excitação e as estatísticas da frequência fundamental, especificamente a média e variância do logaritmo da frequência.

As subseções seguintes detalham o funcionamento de algumas dessas etapas. A Subseção 4.1.1 apresenta a base de dados utilizada, a Subseção 4.1.2 mostra o algoritmo de alinhamento, a Subseção 4.1.3 mostra como são extraídos alguns dos parâmetros do sinal de excitação e a Subseção 4.1.4 mostra como obter o modelo de excitação do falante.

4.1.1 Base de Dados

A base de dados consiste em duzentas frases divididas em vinte conjuntos de dez frases cada. Cada uma das frases é pronunciada por quatro falantes, duas mulheres e dois homens, e são pronunciadas com prosódia neutra, ou seja, sem adicionar características adicionais, como emoção ou ênfase. Cada um dos conjuntos teve suas frases escolhidas de forma que eles fossem foneticamente balanceados, ou seja, a distribuição dos fonemas é similar à distribuição no sotaque escolhido, no caso Português do Rio de Janeiro. Essa escolha é detalhada em [43], e garante que todos os fonemas sejam representados no treinamento, mas ao mesmo tempo dando um peso maior aos fonemas mais comuns.

Cada frase possui em torno de um segundo de duração. Elas foram originalmente gravadas em estúdio com taxa de amostragem de 48 kHz, reduzida para 16 kHz para uso neste projeto.

4.1.2 Alinhamento

Para que a conversão não altere o conteúdo, deve haver uma correspondência entre as envoltórias do falante fonte e do falante alvo, de forma que cada par usado no treinamento represente o mesmo trecho da fala. Caso isso não ocorra, a conversão pode perder qualidade ou, em casos extremos, transformar um fonema em outro, alterando o que é falado.

Existem diversas formas de garantir essa correspondência. Um jeito tradicional é possuir uma base anotada, ou seja, em que cada sinal de fala possui dados extras que informam o que está sendo dito em cada instante de tempo. Uma base desse tipo não é trivial de ser produzida, e introduz restrições adicionais ao uso do sistema, já que para adicionar um novo falante é necessária a anotação dos sinais de fala.

Nesse trabalho, o alinhamento se dá utilizando o algoritmo *dynamic time warping* (DTW). Esse algoritmo calcula distorções temporais de forma a minimizar o erro quadrático

$$\epsilon = \sum_{t'=0}^{T} \left\| \mathbf{y}_{t'} - \mathbf{x}_{t'} \right\|^2, \tag{4.1}$$

onde t' representa o tempo modificado. Essas distorções se dão pela repetição ou pelo descarte dos vetores quando o algoritmo julgar que as sequências ocorrem em velocidades diferentes. O algoritmo 4.1 detalha como o DTW calcula o erro entre as sequências alinhadas. Para que seja feito o alinhamento em si, basta que se guardem

as escolhas entre $D(i-1, j-1)$, $D(i-1, j)$ e $D(i, j-1)$. Essas escolhas descrevem as distorções ótimas que levam ao alinhamento.

Algoritmo 4.1 *Dynamic Time Warping*

Entrada: X e Y, as sequências de *cepstra* dos sinais
Saída: *dist*, a distância entre as duas sequências

```
N ← comprimento de X
M ← comprimento de Y
para i ← 0 até N faça
    D(i, 0) ← ∞
para j ← 0 até M faça
    D(0, j) ← ∞
D(0, 0) ← 0
para i ← 1 até N faça
    para j ← 1 até M faça
        d(i, j) ← ||X(i) − Y(j)||
para i ← 1 até N faça
    para j ← 1 até M faça
        D(i, j) ← Minimo(D(i − 1, j − 1), D(i − 1, j), D(i, j − 1)) + d(i, j)
dist ← D(N, M)
```

O alinhamento por DTW requer que a base de dados possua um *corpus* paralelo, ou seja, possua as mesmas frases sendo ditas pelos dois falantes. Essa é uma restrição menor que ter a base anotada, mas ainda limita as aplicações possíveis. Existem métodos na literatura que possuem requisitos menores. Por exemplo, Erro *et al.* [44] introduziu um algoritmo que iterativamente associa vetores dos dois falantes. Para isso ele treina uma conversão com a associação atual e utiliza a distância entre o sinal alvo e o convertido para atualizar a conversão. A desvantagem dessa técnica é o alto custo computacional, e por isso ela não foi utilizada nesse trabalho.

4.1.3 Obtenção de Parâmetros da Excitação

Instantes de Fechamento Glotal

Durante a produção de sons vozeados, as pregas vocais se abrem e fecham periodicamente, adicionando uma característica impulsiva ao fluxo de ar que sai dos pulmões. O instante em que as pregas vocais se fecham é conhecido com instante de fechamento glotal [1], e sua determinação é importante para qualquer tipo de processamento que requeira sincronismo com o *pitch*, como a obtenção do modelo de pulso da Subseção 4.1.4. Existem várias formas de se calcular esse instante [45], e neste trabalho foi utilizada a introduzida por Drugman *et al.* [46].

[1] A glote é a parte da laringe em que se localizam as pregas vocais.

Essa técnica se baseia no fato de que o fechamento glotal gera uma descontinuidade no fluxo de ar. Essa descontinuidade afeta todo o espectro, inclusive a frequência zero. Assim, o método consiste em filtrar o sinal por um filtro passa-baixas bem seletivo, e utilizar esse sinal filtrado, chamado de sinal de média, para limitar a área de busca. O instantes de fechamento glotal é dado, então, pelo máximo do resíduo de predição linear entre um mínimo e o próximo zero do sinal de média.

Frequência Fundamental

A obtenção da frequência fundamental é um dos problemas mais importantes e bem estudados do processamento da fala. Por isso, existem inúmeras propostas de algoritmos na literatura [47]. Por não ser o foco do trabalho, foi escolhida uma técnica relativamente tradicional, baseada na função de diferença média de magnitudes (*averange magnitude diference function*, AMDF). Ela é definida por:

$$\mathrm{AMDF}(\tau) = \sum_{t=0}^{T} \left| s_w(t) - s_w(t-\tau) \right|, \tag{4.2}$$

onde $s_w(t)$ é um trecho janelado do sinal. Essa função é uma medida da similaridade entre o sinal e uma versão atrasada dele mesmo, sendo menor quanto maior for a similaridade. Para sinais aproximadamente periódicos, essa similaridade é máxima no período correspondente à frequência fundamental. Assim, o mínimo da AMDF corresponde ao período fundamental do sinal.

Um erro comum no cálculo da frequência fundamental é o erro de oitava, quando o algoritmo acha uma frequência que é a metade ou o dobro da frequência real. Para diminuir esse problema, foi utilizada a distância média entre os instantes de fechamento glotal como estimativa inicial do período.

Separação entre Trechos Vozeados e Não-Vozeados

Assim como a obtenção da frequência fundamental, a decisão sobre se um trecho de voz é vozeado ou não é um problema importante e bem estudado do processamento da voz, com diversas técnicas desenvolvidas para realizar essa decisão [48]. Esse trabalho utiliza a taxa de cruzamento por zero e a energia como medidas para determinar se o trecho é vozeado, duas medidas tradicionais, e um trecho é considerado vozeado se ele for vozeado por ambos os critérios. A taxa de cruzamento por zero é definida como:

$$\mathrm{ZC} = \frac{1}{2} \sum_{t=0}^{T} \left| \mathrm{sign}(s_w(t)) - \mathrm{sign}(s_w(t+1)) \right|, \tag{4.3}$$

onde:

$$\text{sign}(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ -1 & \text{se } x < 0. \end{cases} \tag{4.4}$$

Por sua característica ruidosa, trechos não vozeados tendem a possuir uma taxa de cruzamento por zero alta. Sinais periódicos com características espectrais típicas de sinais de fala, por outro lado, geralmente só cruzam o zero algumas vezes por período. Trechos vozeados possuem, portanto, taxas de cruzamento por zero baixas. Assim, fica claro como essa medida pode ser usada para classificar um trecho como vozeado ou não: se o trecho possuir uma taxa de cruzamento por zero maior que um certo limiar, ele é não-vozeado; caso contrário, ele é vozeado.

A energia de um trecho é dada por:

$$E = \sum_{t=0}^{T} (s_w(t))^2. \tag{4.5}$$

Trechos de fala vozeados tendem a possuir energia maior do que trechos não vozeados, e portanto a energia pode ser usada para discriminar trechos vozeados de forma similar à taxa de cruzamento por zeros. É importante notar que trechos com energia baixa podem também ser trechos de silêncio. Como trechos não-vozeados não são utilizados na conversão, a discriminação entre os dois não é importante para essa aplicação, e os trechos de silêncio podem ser tratados como não-vozeados.

4.1.4 Modelagem da Excitação

Ao se utilizar o modelo fonte-filtro para a produção da voz, cada trecho de sinal é separado em um filtro de envoltória e um sinal de excitação. Os capítulos anteriores trabalharam principalmente sobre o filtro de envoltória, mas a excitação também é importante, já que sua escolha é diretamente proporcional à qualidade no sinal sintetizado.

Uma escolha natural é usar a própria excitação obtida por filtragem do sinal pelo inverso do seu filtro de envoltória. De fato, o uso dessa excitação produz sinais com boa naturalidade, mas ela não é boa para a conversão da fala. O sinal de excitação obtido dessa forma carrega informações sobre o *pitch*, e características relacionadas ao *pitch*, como sua média e sua variância, também são importantes para a identificação de falante. Por isso, um sinal convertido utilizando essa excitação é mais próximo perceptivamente do falante fonte do que seria desejado.

Em aplicações onde a excitação original não está disponível, o modelo fonte-filtro utiliza, tradicionalmente, um sinal de excitação paramétrico dado por:

$$e(x) = \begin{cases} \sum_k \delta(t - k\tau) & \text{se vozeado} \\ n(t) & \text{se não-vozeado,} \end{cases} \tag{4.6}$$

onde $\delta(t)$ é o impulso unitário, τ é o período e $n(t)$ é ruído branco gaussiano. Essa definição parte diretamente da concepção do modelo, explicado na Seção 2.1. Esse modelo permite controlar livremente informações como *pitch* e energia, o que é útil para a conversão, mas usar essa excitação na prática, entretanto, gera sinais de voz com uma série de distorções e artefatos, e que são comumente descritos como "robóticos".

Isso se dá porque as técnicas de extração de envoltória não conseguem extrair completamente toda a informação contida no sinal. A extração da envoltória de um trecho de um sinal gera um resíduo com um espectro aproximadamente plano, o que bastaria se o sinal fosse perfeitamente periódico ou perfeitamente ruidoso. Sinais reais, entretanto possuem ambas as componentes, e suas contribuições relativas variam com a frequência: em frequências baixas, domina a componente periódica, enquanto nas frequências altas domina a componente ruidosa.

Pitch Synchronous Overlap and Add

O algoritmo *pitch synchronous overlap and add* (PSOLA) é uma técnica de modificação de sinais semi-periódicos que permite alterar a frequência e a duração de um sinal de forma independente. Esse algoritmo é muito usado para alteração de *pitch* e duração de sinais de fala [49][48].

No contexto de conversão de falante, o PSOLA pode ser utilizado para alterar as estatísticas de *pitch* da excitação obtida do falante fonte, para que elas sejam similares às do falante alvo; essa excitação modificada pode ser então utilizada, então, para sintetizar a voz convertida. Como essas estatísticas, principalmente o *pitch* médio, são importantes para a identificação de falante, um sinal sintetizado por essa excitação é perceptivelmente mais próximo do falante alvo do que um sinal sintetizado utilizando a excitação natural. Ao mesmo tempo, como essa excitação é obtida diretamente do sinal, o sinal sintetizado possui maior naturalidade do que se fosse utilizado o sinal paramétrico simples da equação (4.6)

O PSOLA divide o sinal em segmentos contendo um período do sinal original, como ilustrado na Figura 4.2. Esses segmentos são então movidos temporalmente,

duplicados e removidos para se obter o número de ciclos por intervalo de tempo correspondentes ao *pitch* desejado.

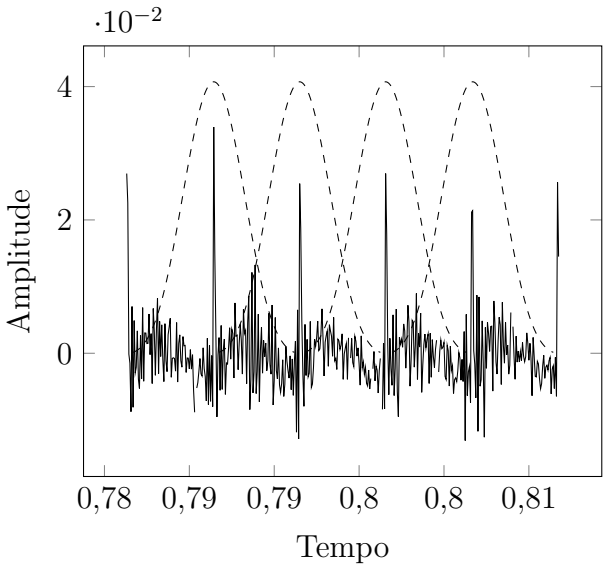

Figura 4.2: Divisão do residuo em segmentos no PSOLA.

A decomposição do sinal em segmentos pode ser descrita por:

$$s_m(n) = h(n)s(n - p_m), \tag{4.7}$$

onde $h(n)$ é uma função de janelamento e p_m é a marca de *pitch* correspondente ao m-ésimo segmento. Para que o PSOLA obtenha bons resultados, essas marcas do *pitch* devem ser determinadas de forma que suas posições sejam consistentes, ou seja, possuam a mesma posição dentro de um período. Para garantir esse fato, uma escolha comum de marca de *pitch* são os instantes de fechamento glotal, cuja obtenção foi explicada na Subseção 4.1.3.

O sinal modificado é obtido então somando-se os segmentos transladados para novas posições no tempo:

$$s'(n) = \sum_{m'} s_{m'}(n - p_{m'}), \tag{4.8}$$

onde o índice m' indica qual não há necessariamente correspondência um para um entre os segmentos obtidos e os utilizados na síntese. Se o resultado esperado for um aumento de *pitch*, o sinal resultante deve possuir mais períodos que o sinal original, e, necessariamente, alguns segmentos devem ser repetidos. De forma similar, se o sinal resultante possuir um *pitch* menor, alguns segmentos devem ser descartados.

Modelo Paramétrico

A excitação do falante fonte modificada pelo TD-PSOLA ainda carrega características desse falante. Se essas características forem distintas o bastante do falante alvo, utilizar essa excitação na síntese pode resultar em sinais que ainda são perceptivamente similares ao falante fonte.

Uma solução possível é a utilização de um modelo paramétrico treinado sobre excitações do falante alvo. Existem diversos modelos desse tipo. Um deles, com origem no estudo da síntese de fala, é utilizar um banco de filtros em que a potência relativa entre sinal e ruído é determinada para cada faixa [50][51]. Ela gera bons resultados, mas utilizá-la na conversão envolve a modelagem adicional dessas potências relativas. Outra técnica de síntese, introduzida por Drugman *et al.* [52], já resulta em um modelo que pode ser utilizado diretamente na conversão. Ela é que será utilizada neste trabalho.

Nesta técnica, cada quadro da excitação é dado por:

$$e_w(t) = \begin{cases} \sum_k p(t - k\tau - t_0) + (h_n * n)(t) & \text{se vozeado} \\ n(t) & \text{se não-vozeado,} \end{cases} \tag{4.9}$$

onde $p(t)$ e $h_n(t)$ são, respectivamente, um pulso e um filtro, ambos obtidos previamente, e t_0 é um valor de tempo que garante a coerência entre quadros que se sobrepõem.

O pulso é treinado sobre um conjunto de quadros, sincronizados pelo *pitch*, da excitação obtida por filtragem inversa com o filtro de envoltória. Para cada instante de fechamento glotal t_g, é obtida uma amostra de pulso do sinal de excitação através de um janelamento:

$$p_{t_g}(t) = e(t + t_g)w_\tau(t), \tag{4.10}$$

onde $w_\tau(t)$ é uma função janela de comprimento $2\tau + 1$, de forma que as bordas da janela se localizem aproximadamente nos instantes de fechamento glotal anterior e próximo. Essas janelas são então normalizadas em comprimento (através de uma reamostragem) e em energia, e é realizada uma análise de componentes principais (*principal component analysis*, PCA) desse conjunto de pulsos.

A PCA é uma transformação ortogonal aplicada sobre um conjunto de vetores de forma que suas dimensões sejam descorrelacionadas. Os eixos desse novo sistema são chamados componentes principais. Essas componentes podem ser ordenadas pelos

seus autovalores correspondentes, que estão relacionados à variância dos dados na direção dessa componente. Uma consequência disso é que as primeiras componentes principais por essa ordenação concentram a maior parte da informação contida no conjunto de vetores, e portanto descartar as componentes com autovalores menores é uma maneira eficiente de reduzir a dimensionalidade de um conjunto.

Nesse trabalho, um conjunto de pulsos e um conjunto de componentes principais é obtido para cada falante. Para simplificar o modelo, o pulso $p(t)$ é representado apenas pela primeira componente principal. A Figura 4.3 mostra um exemplo de um pulso natural e o compara com a primeira componente principal.

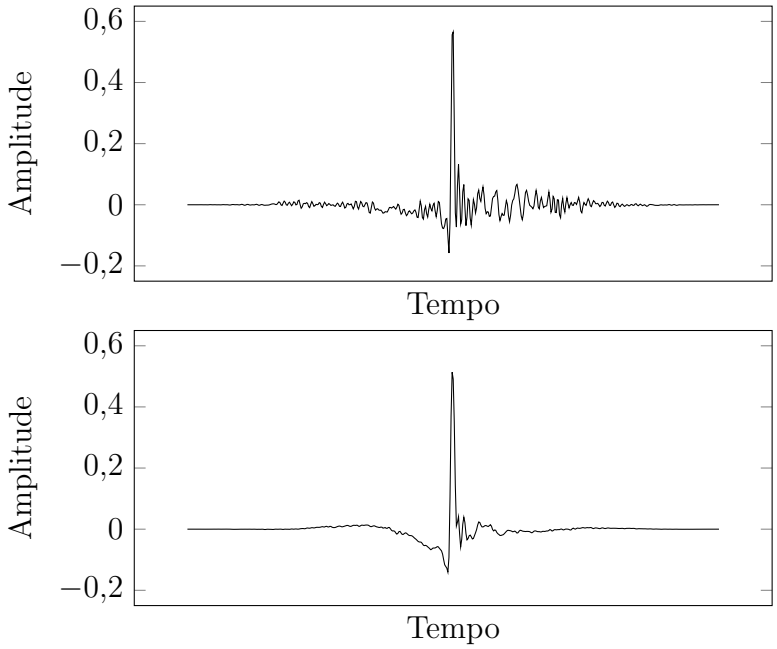

Figura 4.3: Exemplo de pulso natural (cima) e pulso treinado (baixo).

O conjunto de treinamento do filtro $h_n(t)$ que modula o ruído é obtido de maneira similar. O sinal também é janelado em torno dos instantes de fechamento glotal, e em seguida os quadros são normalizados em energia e passam por uma filtragem passa-altas. Essa filtragem busca eliminar a componente periódica, e deve possuir frequência de corte igual à frequência em que a componente ruidosa passa a predominar. Em geral essa frequência varia, mas 4 kHz pode ser usada como uma aproximação razoável [52].

Para a obtenção do filtro, é obtida a média da magnitude dos espectros dos quadros do conjunto. A transformada inversa dessa média é janelada, e o resultado é utilizado como coeficientes do filtro de resposta ao impulso finita.

Nesse trabalho, tanto o pulso quanto o filtro são considerados fixos para um falante, pois considerá-los variáveis envolve a modelagem adicional de como as variações do falante fonte se relacionam com variações do falante alvo.

4.2 Arquitetura da Conversão

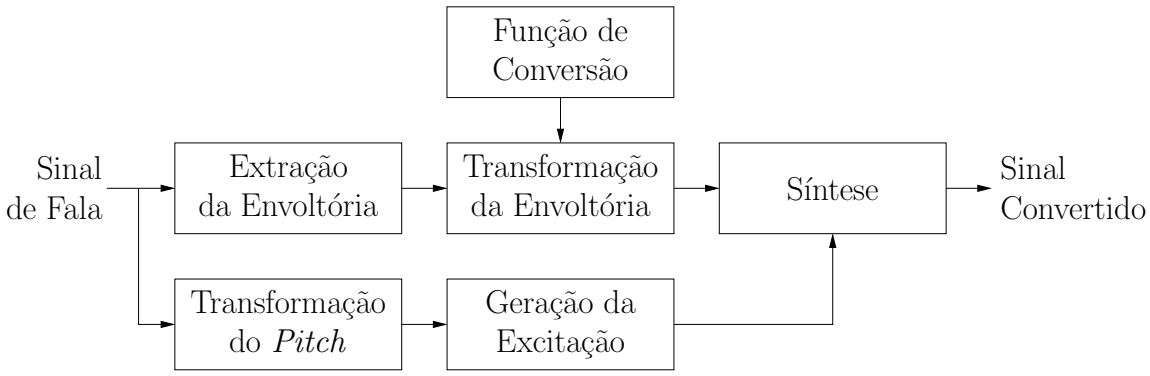

Figura 4.4: Arquitetura do sistema de conversão.

A Figura 4.4 mostra o diagrama de blocos da conversão. Assim como no treinamento, o sinal é janelado e são extraídas a envoltória e a frequência fundamental de cada quadro. As envoltórias que correspondem a quadros vozeados são modificadas utilizando-se as transformações aprendidas no treinamento, e as frequências fundamentais do falante fonte são transformadas pela relação [5]:

$$\mathcal{F}(f_0) = \exp\left(\mu_t + \frac{\sigma_t}{\sigma_s}(\log(f_0) - \mu_s)\right), \tag{4.11}$$

onde μ_s e μ_t são as médias do logaritmo das frequências dos falantes fonte e alvo, respectivamente, e σ_s e σ_t são suas variâncias. Essa transformação se dá no logaritmo da frequência, pois a audição humana percebe frequências em uma escala aproximadamente logarítmica.

Obtidas as frequências convertidas, os novos tempos de fechamento glotal são gerados. Para isso, considerando-se a frequência angular ω_i da i-ésima janela, é realizada uma interpolação linear para se obter a frequência instantânea:

$$\omega(t) = at + \omega_i, \qquad t_i \leq t \leq t_{i+1}, \tag{4.12}$$

onde:

$$a = \frac{\omega_{i+1} - \omega_i}{t_{i+1} - t_i}, \tag{4.13}$$

e t_i o tempo central da i-ésima janela. Essa interpolação é feita para aumentar a naturalidade da voz. A fase instantânea pode ser obtida integrando-se a frequência instantânea:

$$\theta(t) = \int_{t_i}^{t} \omega(t')dt' = \frac{1}{2}at^2 + \omega_i t + \theta_0, \qquad t_i \leq t \leq t_{i+1}, \tag{4.14}$$

onde θ_0 é obtido no trecho anterior de forma que a função seja continua. Os instantes de fechamento glotal são, então, os instantes em que um ciclo é completo:

$$\{t_g\} = \{t|\theta(t) \mod 2\pi = 0\}. \tag{4.15}$$

Esses instantes de fechamento glotal são aproximados para que se adequem ao tempo discreto utilizado, e são associados às suas janelas correspondentes. Para a geração de excitação por PSOLA, esses novos instantes de fechamento glotal são associados aos antigos, e os segmentos são somados de acordo com esses novos instantes. Para a geração de excitação utilizando o modelo paramétrico é utilizada uma versão da equação (4.9):

$$e(t) = \begin{cases} \sum_k p_\tau(t - t_g(k)) + (h_n * n)(t) & \text{se vozeado} \\ n(t) & \text{se não-vozeado,} \end{cases} \tag{4.16}$$

onde $p_\tau(t)$ é o pulso reamostrado para que tenha comprimento $2\tau + 1$. A excitação tem, então, sua energia igualada à energia da excitação original do falante alvo.

Com as envoltórias transformadas e tendo-se gerado a excitação, é possível realizar a síntese, que será explicada na Subseção 4.2.1. O sinal gerado é então "dejanelado", e a conversão está completa.

4.2.1 Síntese a partir do mel-*cepstrum*

Realizada a conversão da envoltória, o sinal resultante pode ser obtido pela filtragem de um sinal de excitação pelo filtro que descreve a envoltória. Quando a envoltória é representada por coeficientes de predição linear, essa filtragem é simples, pois estes são os coeficientes de um filtro. Neste trabalho foi utilizado o mel-*cepstrum* como domínio de conversão, e como a conversão de volta para os coeficientes de predição linear introduz erros, essa síntese é realizada diretamente a partir do mel-*cepstrum*.

Como foi visto no Capítulo 2, o mel-*cepstrum* define um filtro pela relação:

$$H(z) = \exp\left(\sum_{k=0}^{L} \tilde{c}_k \tilde{z}^{-k}\right). \tag{4.17}$$

Infelizmente, esse filtro não é realizável por filtros lineares, pois estes não podem realizar a exponenciação. Essa limitação pode ser contornada utilizando-se um filtro que aproxime a exponencial.

Para que uma aproximação seja realizável por um filtro linear, ela deve possuir um formato ou polinomial ou racional (razão de polinômios). Por exemplo, considerando-se a expansão de Taylor da exponencial em torno de zero:

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots, \tag{4.18}$$

pode-se definir um filtro

$$R(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \frac{z^4}{4!} \tag{4.19}$$

que aproxima o filtro $H(z) = \exp(z)$. Assim, uma solução possível para a realização do filtro de síntese é

$$H(z) = \exp D(z) \simeq 1 + \sum_{n=1}^{N} \frac{D^n(z)}{n!}, \tag{4.20}$$

onde N é a ordem da aproximação e

$$D(z) = \sum_{k=0}^{L} \tilde{c}_k \tilde{z}^{-k}. \tag{4.21}$$

Uma outra aproximação para $\exp(x)$ é o aproximante de Padé de ordem N por N em torno de zero:

$$\exp(x) \simeq \frac{1 + \sum_{n=1}^{N} A_{N,n} x^n}{1 + \sum_{n=1}^{N} A_{N,n} (-x)^n}, \tag{4.22}$$

onde:

$$A_{N,n} = \frac{1}{n!} \binom{N}{n} \Big/ \binom{2N}{n}. \tag{4.23}$$

A aproximação do filtro de síntese fica, então:

$$H(z) = \exp D(z) \simeq \frac{1 + \sum_{n=1}^{N} A_{N,n} D^n(z)}{1 + \sum_{n=1}^{N} A_{N,n} (-D)^n(z)}. \tag{4.24}$$

Essa aproximação é equivalente a uma série de Taylor de ordem $2N$. Como cada

potência na aproximação representa uma nova instância de $D(z)$ em cascata, essa realização é mais eficiente.

O filtro da equação (4.24) é chamado filtro de aproximação mel-log-espectral (*mel log spectral aproximation*, MLSA) [53][54]. A Figura 4.5 mostra o diagrama desse filtro.

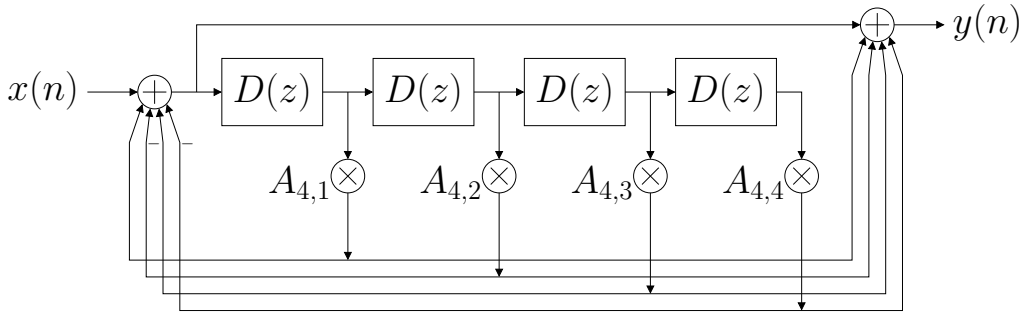

Figura 4.5: Diagrama do filtro MLSA de ordem 4.

Se o filtro $D(z)$ for realizado da maneira direta, utilizando o filtro passa-tudo

$$\tilde{z}^{-1} = \Phi(z) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}, \tag{4.25}$$

então o filtro MLSA possui um laço sem atraso, o que não é realizável. Para solucionar esse problema, o filtro $D(z)$ deve ser modificado de forma a remover o caminho livre de atraso. Para isso primeiramente o filtro é separado:

$$D(z) = K + \bar{D}(z), \tag{4.26}$$

onde:

$$\begin{aligned} K &= \sum_{k=0}^{L} \tilde{c}_k (-\alpha)^{-k}, \\ \bar{D}(z) &= \sum_{k=0}^{L} \bar{c}_k \tilde{z}^{-k}. \end{aligned} \tag{4.27}$$

A relação entre $\tilde{c}_k$ e $\bar{c}_k$ é dada por:

$$\bar{c}_k = \begin{cases} \tilde{c}_0 - \sum_{k=0}^{L} \tilde{c}_k (-\alpha)^{-k} & \text{se } k = 0 \\ \tilde{c}_k & \text{se } k > 0. \end{cases} \tag{4.28}$$

O fator K representa o ganho do caminho sem atraso do filtro $D(z)$, e $\bar{D}(z)$ é o filtro com esse ganho cancelado. Ao tomar a exponencial de $D(z)$ temos:

$$H(z) = \exp D(z) = \exp K \exp \bar{D}(z). \tag{4.29}$$

O fator $\exp K$ é um ganho constante, e portanto não precisa ser aproximado. Consequentemente, o filtro MLSA só precisa aproximar $\exp \bar{D}(z)$.

Se o filtro $\bar{D}(z)$ for representado na forma vetorial:

$$\bar{D}(z) = \sum_{k=0}^{L} \bar{c}_k \tilde{z}^{-k} = \tilde{\mathbf{z}}^\top \bar{\mathbf{c}}, \tag{4.30}$$

então uma transformação possível é:

$$\begin{aligned} \bar{D}(z) &= \tilde{\mathbf{z}}^\top \bar{\mathbf{c}} \\ &= \tilde{\mathbf{z}}^\top \mathbf{A}\mathbf{A}^{-1} \bar{\mathbf{c}} \\ &= \mathbf{\Phi}^\top \mathbf{b} \\ &= \sum_{k=1}^{L} \tilde{b}_k \Phi_k(z), \end{aligned} \tag{4.31}$$

onde:

$$\mathbf{A} = \begin{bmatrix} 1 & \alpha & 0 & \cdots & 0 & 0 \\ 0 & 1 & \alpha & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \alpha \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \tag{4.32}$$

$$\mathbf{A}^{-1} = \begin{bmatrix} 1 & (-\alpha) & (-\alpha)^2 & \cdots & (-\alpha)^L \\ 0 & 1 & (-\alpha) & \cdots & (-\alpha)^{L-1} \\ 0 & 0 & 1 & \cdots & (-\alpha)^{L-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{4.33}$$

O vetor $\mathbf{\Phi}$ é dado por:

$$\begin{aligned} \mathbf{\Phi} &= \mathbf{A}^\top \tilde{\mathbf{z}} \\ &= \begin{bmatrix} 1 & \Phi_1(z) & \Phi_2(z) & \cdots & \Phi_L(z) \end{bmatrix}, \end{aligned} \tag{4.34}$$

onde:

$$\Phi_k(z) = \frac{(1-\alpha^2)z^{-1}}{1-\alpha z^{-1}}\tilde{z}^{-(m-1)}. \tag{4.35}$$

Note que para $k \geq 1$, o ganho sem atraso de $\Phi_k(z)$ é zero.

Os coeficientes $\mathbf{b}$ são obtidos por

$$\begin{aligned}\mathbf{b} &= \mathbf{A}^{-1}\bar{\mathbf{c}} \\ &= \begin{bmatrix} 0 & b_1 & b_2 & \cdots & b_L \end{bmatrix}. \end{aligned} \tag{4.36}$$

Essa operação matricial pode ser substituída pela relação recursiva:

$$b_k = \begin{cases} \bar{c}_L & \text{se } k = L \\ \bar{c}_k - \alpha b_{k+1} & \text{se } 0 \leq k < L. \end{cases} \tag{4.37}$$

O primeiro elemento de $\mathbf{b}$ é zero, pois $\sum_{k=0}^{L} \bar{c}_k(-\alpha)^{-k} = 0$. Como $b_0 = 0$, e os $\Phi_k(z)$ não possuem ganho sem atraso, $\bar{D}(z)$ não possui nenhum caminho sem atraso, e portanto pode ser usado no filtro MLSA. A Figura 4.6 mostra o diagrama de blocos dessa representação do filtro.

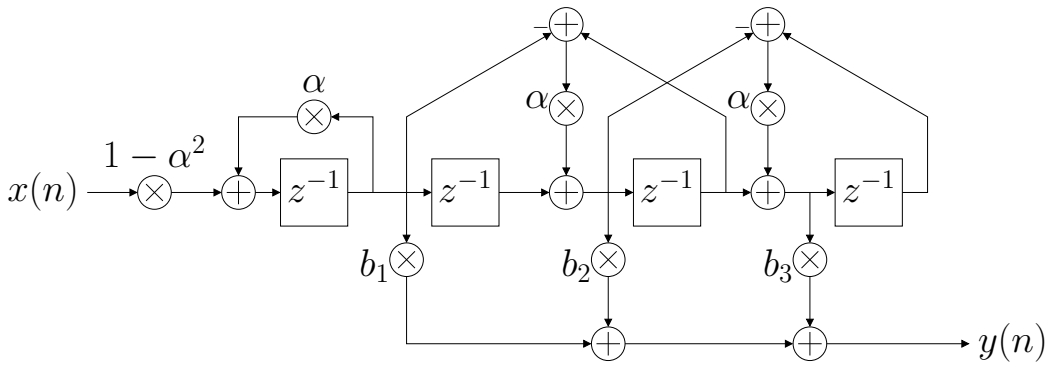

Figura 4.6: Filtro base $\bar{D}(z)$.

O filtro MLSA é uma aproximação, e portanto ele introduz alguma distorção. Essa distorção pode ser minimizada aumentando-se a ordem da aproximação, mas pode-se também otimizar diretamente os ganhos $A_{L,l}$. Os coeficientes obtidos por [54] são apresentados na Tabela 4.1.

l	$A_{4,l}$	$A_{5,l}$
1	$4{,}999273 \times 10^{-1}$	$4{,}999391 \times 10^{-1}$
2	$1{,}067005 \times 10^{-1}$	$1{,}107098 \times 10^{-1}$
3	$1{,}170221 \times 10^{-2}$	$1{,}369984 \times 10^{-2}$
4	$5{,}656279 \times 10^{-4}$	$9{,}564853 \times 10^{-4}$
5	—	$3{,}041721 \times 10^{-4}$

Tabela 4.1: Valores otimizados para os ganhos $A_{L,l}$ para ordens 4 e 5.

4.3 Linguagens e Bibliotecas

A maior parte do sistema foi implementado utilizando a linguagem *MATLAB*, com algumas partes sendo implementadas na linguagem *Python*. A implementação segue um paradigma funcional, ou seja, o programa é dividido em funções que realizam tarefas específicas. Funções padrões do *MATLAB*, como cálculo dos coeficientes de predição linear, foram utilizadas quando possível. A maior parte das funções, entretanto, é de implementação própria.

Além disso algumas bibliotecas foram utilizadas. Foi utilizada a implementação de modelo oculto de Markov do *Probabilistic Modeling Toolkit* (PMTK) [55], e as de redes neurais *feed-forward* e recorrentes da biblioteca *Keras* [56]. Os códigos podem ser obtidos contactando-se o autor em `vicpc@poli.ufrj.br`.

Capítulo 5

Experimentos e Resultados

Neste capítulo são realizados experimentos para avaliar comparativamente o desempenho dos métodos de conversão. A Seção 5.1 apresenta testes utilizando o erro quadrático como métrica de avaliação, e a Seção 5.2 apresenta um teste que utiliza um sistema de reconhecimento automático de falante para avaliar os métodos. Finalmente, a Seção 5.3 apresenta comentários baseados em escutas informais, discutindo alguns dos problemas perceptivos e algumas soluções.

5.1 Erro Quadrático

O primeiro experimento busca avaliar de forma objetiva a diferença entre a fala convertida e a fala original do falante alvo. Para isso, a grande maioria dos artigos da literatura utilizam alguma variação do erro quadrático entre os mel-*cepstra*, chamado de distorção mel-*cepstral* (*mel-cepstral distortion*, MCD). A forma mais simples de defini-la é:

$$d = \sqrt{\sum_{i=1}^{L} (c_c(i) - c_a(i))^2}, \tag{5.1}$$

onde c_c e c_a são os mel-*cepstra* de um quadro do sinal convertido e do sinal original do alvo, respectivamente, e L é o comprimento do mel-*cepstum*. O *cepstrum* de índice zero não é utilizado, pois representa apenas um ganho constante na frequência. A MCD também é uma medida natural para o desempenho das técnicas utilizadas nesse trabalho, já que todas elas envolvem, pelo menos parcialmente, uma otimização do erro quadrático em seu treinamento.

Como a MCD é uma medida do erro por quadro, uma medida da distância entre os siais convertidos e os originais pode ser obtida tomando-se a média do erro de cada quadro. Para um conjunto de teste com N frases, cada uma com comprimento

T_n, a forma mais simples dessa medida de distância é:

$$D = \frac{\sum_{n=1}^{N} \sum_{t=1}^{T_n} d_{(n,t)}}{\sum_{n=1}^{N} T_n}. \tag{5.2}$$

Para o cálculo desses erros, o sinais são alinhados entre si, e apenas os quadros vozeados são considerados.

Embora quase todos os artigos da literatura utilizem a MCD de alguma forma, eles frequentemente usam variações que, apesar de similares, dificultam a comparação direta entre os resultados apresentados. Algumas vezes a definição da MCD é ligeiramente diferente, como por por exemplo a utilizada por Desai *et al.* [5] e Quiao *et al.* [7]:

$$d = \frac{10}{\log 10} \sqrt{2 \sum_{i=1}^{L} (c_c(i) - c_a(i))^2}, \tag{5.3}$$

que difere da anterior por um valor constante. Algum artigos normalizam as distorções médias, como em [2], onde a MCD é normalizada pela distorção média original entre os falantes fonte e alvo, ou em [24], em que a MCD é normalizada pela diferença média entre os *cepstra* do falante alvo e o *cepstrum* médio do mesmo. A forma como a média é realizada também varia: a maioria utiliza a média das MCDs, mas Kain *et al.* [24] utiliza a média dos quadrados das MCDs. Algumas dessas variações são brevemente avaliadas nesse trabalho.

No teste realizado, cada um dos métodos foi treinado, para cada par de falantes fonte e alvo (importando a ordem), utilizando um conjunto de cem pares de frases entre um e três segundos cada, totalizando aproximadamente quatro minutos por falante. Para cada tipo de conversão, foram realizados diversos treinamentos, variando a complexidade do modelo, e o desempenho da conversão foi avaliado para cada complexidade. A complexidade do modelo é representada pelo número de componentes para o GMM, pelo número de estados para o HMM, pelo número de neurônios na camada oculta para a FFNN e pelo número de neurônios na camada recursiva para a RNN. O conjunto de teste consiste em quarenta frases que não pertencem ao conjunto de treinamento, e o erro é obtido utilizando diretamente as envoltórias convertidas, antes que seja feita a síntese. Exemplos de sinais convertidos podem ser obtidos em `www02.smt.ufrj.br/~victor.costa/examples/voice_conversion/`.

O primeiro experimento busca verificar como a escolha da métricas afeta os resultados. Para isso foram calculados os resultados da conversão por GMM para alguns pares de falantes utilizando a definição mais básica da MCD e da média (equações

(5.1) e (5.2)), a raiz quadrada da média dos quadrados das MCDs (versão modificada de Kain *et al.* [24]), e a média das MCDs normalizada pela distorção original entre fonte e alvo (Stylianou *et al.* [2]). A Figura 5.1 apresenta esses resultados. Como pode ser visto, os valores específicos variam, mas a forma geral de cada gráfico tende a permanecer similar. Pode-se notar também que no caso normalizado a mudança no valor médio de cada gráfico foi mais extrema, provocando uma mudança na ordem relativa dos gráficos. Entretanto, essa mudança não foi suficiente para colocar os resultados de diferentes pares de falantes em uma mesma escala, o que seria o resultado ideal de uma normalização. Por esses motivos, no resto deste trabalho, apenas a média tradicional foi utilizada para apresentação dos resultados.

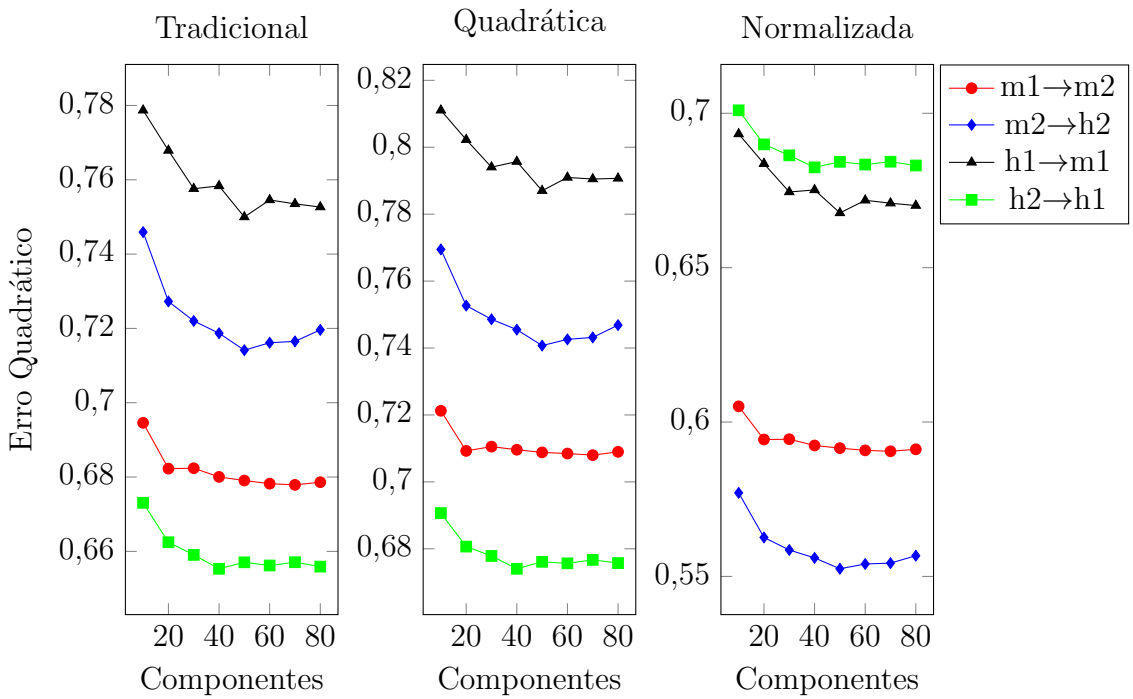

Figura 5.1: Comparação entre diferentes formas de definir o erro quadrático da conversão para alguns pares de falantes. Média das MCDs de cada quadro (esquerda), raiz quadrada da média dos quadrados (meio) e média normalizada pela distorção original entre fonte e alvo (direita).

O erro D se refere a um par de falantes fonte e alvo. Um problema de se usar essa medida é que ela é bastante dependente de quão distantes os falantes eram antes da conversão. A Figura 5.2 ilustra esse fato. Pode-se reparar que o erro varia mais com a identidade dos falantes do que com a complexidade do modelo. É possível que exista uma normalização que coloque os erros de cada par em uma mesma escala, mas, como visto anteriormente, as normalizações apresentadas na literatura não possuem essa propriedade.

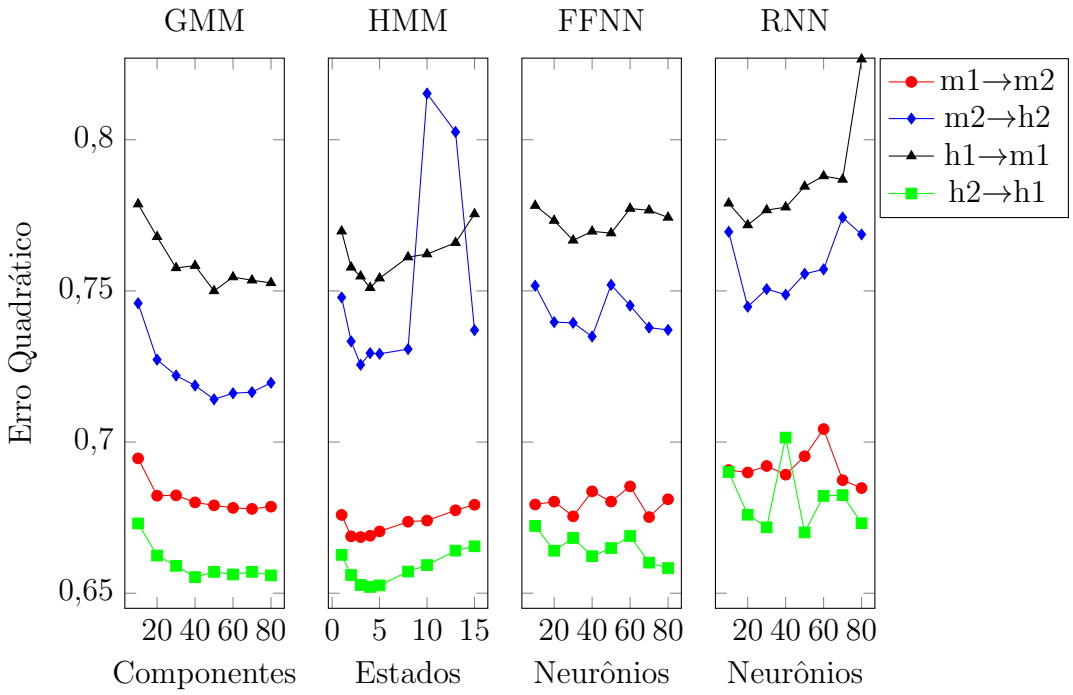

Figura 5.2: Comparação entre o erro quadrático da conversão para alguns pares de falante.

A variação do erro em relação ao falante também domina sobre a variação em relação ao método. Assim, a comparação entre os métodos fixando-se um par fonte/alvo faz mais sentido que a comparação entre pares diferentes. Pode-se notar também que a relação entre complexidade e erro para cada par de falantes tende a ser similar quando se utiliza o GMM ou o HMM, exceto por alguns pontos no segundo; essa relação tende a ser menos regular quando se utiliza a FFNN ou a RNN. Como cada ponto de cada gráfico envolve apenas um treinamento, não é possível saber quanto dessa variação se deve a falhas no treinamento, como mínimos locais, e quanto dessa variação se deve a uma tendência na relação complexidade-erro desse método.

Por essas considerações, a métrica utilizada para comparar os métodos foi a média de D para todos os pares. Os resultados dessa comparação são apresentados na Figura 5.3.

Em geral, o método que obteve melhor resultado foi o baseado em modelos de misturas gaussianas. O método de conversão por rede neural recorrente obteve o pior resultado na média, e a conversão por modelo oculto de Markov apresentou grande variabilidade no desempenho, mas seu mínimo é comparável ao da técnica baseada em GMM. A conversão baseada em redes neurais *feed-foward* obteve resultados médios.

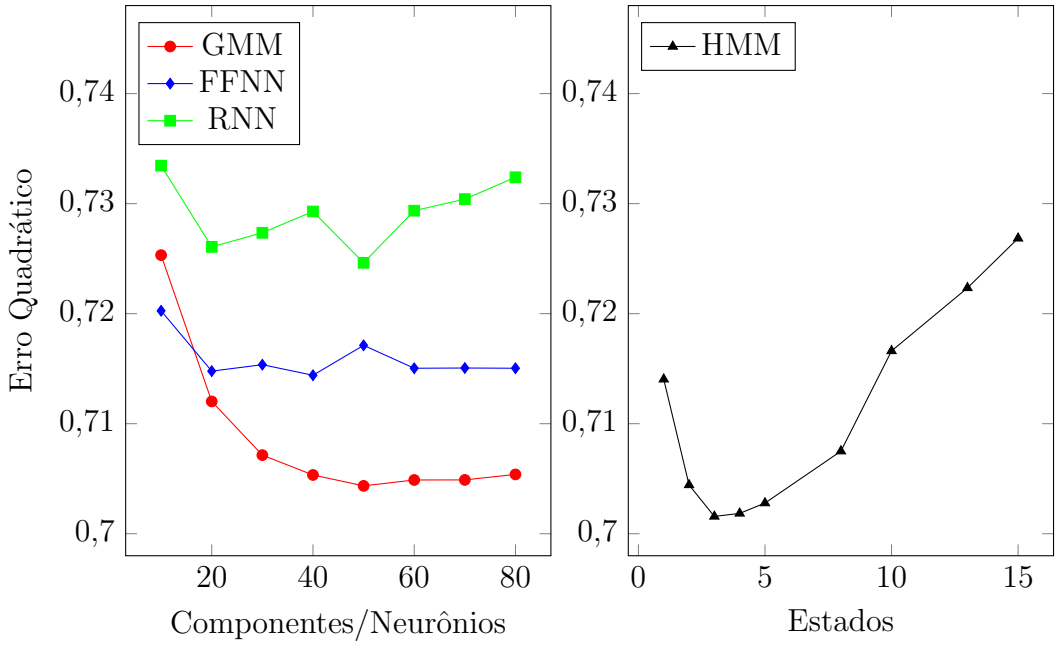

Figura 5.3: Média do erro quadrático entre todos os falantes, para os quatro métodos.

A tendência para a relação entre complexidade e erro é similar entre os métodos. O erro tende a ser alto para baixas complexidades, diminuir com o aumento da complexidade até um mínimo, e ou se estabilizar ou voltar a crescer para complexidades altas. O caso mais extremo foi a conversão por modelo oculto de Markov, que atingiu o mínimo com apenas três estados. Esse é um número de estados menor do que esperado. É possível que a quantidade de frases utilizadas no treinamento não seja suficiente para representar modelos maiores, o que resulta em estados pouco visitados e erros numéricos no cálculo da função de conversão. Testes adicionais devem ser realizados para comprovar essa hipótese, mas se ela for verdadeira, ela representa uma limitação adicional em um sistema que utiliza essa técnica.

Uma observação similar pode ser feita em relação à conversão por rede neural recorrente. É possível que ela obtenha resultados melhores com uma base maior, mas essa é uma restrição adicional à sua aplicação. Outro fato importante sobre redes neurais é que elas são extremamente customizáveis. Nesse trabalho foi utilizada a forma mais básica de rede recorrente, mas existe uma grande quantidade de variações, como tipos alternativos de recorrência [57] ou técnicas de regularização [58]. Foram experimentadas algumas dessas técnicas durante o desenvolvimento, mas elas não produziram melhoras significativas em testes preliminares.

5.2 Testes com Sistema de Reconhecimento de Falantes

O próximo experimento envolve o uso de um sistema de reconhecimento de falantes. Sistemas desse tipo utilizam informação da fala para identificar um falante. Assim, a taxa de acerto em um sistema desse tipo pode ser usada como métrica para avaliar um sistema de conversão.

Nesse teste foram utilizados sinais convertidos utilizando os mesmos modelos treinados no experimento anterior, e os sinais foram sintetizados utilizando tanto excitações obtidas utilizando pulsos paramétricos treinados a partir do falante alvo quanto excitações obtidas a partir da modificação da excitação fonte por PSOLA. Foi utilizada a API de reconhecimento de falante do *Microsoft Cognitive Services* [59], que proporciona um sistema de reconhecimento através de mensagens HTTP. Os métodos utilizados por esse sistema não são divulgados, mas espera-se que exista uma grande sobreposição entre as características da fala que ele usa no reconhecimento e as características que são alteradas na conversão. A saída do reconhecimento desse sistema é a identidade do falante e um grau de confiança, que pode ser "normal"ou "alto". Por esse sistema ter sido projetado para sinais mais longos do que os sinais da base, os testes foram realizados com pares de frases concatenadas.

O sistema de reconhecimento foi treinado para cada falante com um conjunto de cinquenta frases, totalizando aproximadamente dois minutos por falante. O primeiro teste realizado nesse sistema foi um teste de sanidade, para checar o funcionamento do sistema em sinais não convertidos. Foram realizados três testes: reconhecimento de sinais não modificados e de sinais que passaram pelo processo de extração de parâmetros e síntese utilizando os dois tipos de excitação, mas não conversão. Os resultados desse teste são apresentados na Tabela 5.1.

	Taxa de Acerto (%)							
	Qualquer Confiança				Confiança Alta			
	m1	m2	h1	h2	m1	m2	h1	h2
Sinal original	100	100	95	100	100	100	95	100
Pulso paramétrico	100	90	85	100	100	40	70	100
PSOLA	100	100	95	100	100	100	95	100

Tabela 5.1: Taxa de acerto dos sinais originais e de sinais que passaram pela extração de parâmetros e síntese.

Como esperado, os sinais originais possuíram taxas de acerto altas e, em termos de taxa de acerto, o sinal sintetizado manteve o desempenho do sinal original. En-

tretanto, houve uma perda quando só se consideram acertos com alta confiança na síntese utilizando o pulso paramétrico. Isso indica que, pelo menos para os falantes $m2$ e $h1$, houve perda significativa de qualidade na síntese por esse método.

Em seguida o teste de reconhecimento foi aplicado aos sinais convertidos. Para cada técnica, foram testados três níveis de complexidade. Para cada nível, cada combinação de falante fonte e alvo e cada tipo de excitação, foram testadas quarenta frases convertidas. Os resultados se encontram nas Tabelas 5.2, 5.3, 5.4 e 5.5.

Comparando a taxa de acerto total, os dois métodos de síntese obtiveram resultados similares, com o PSOLA tendo uma taxa de acerto em geral um pouco menor. Entretanto, a variação é muito maior quando se compara as taxas de acerto para apenas um par de falantes, com alguns pares tendo resultados muito maiores em um método quando comparado ao outro. Esses ganhos também são aproximadamente consistentes entre os métodos de conversão, indicando que o método de síntese é a principal causa dessa diferença.

A síntese por pulso paramétrico consistentemente obteve melhores resultados que a por PSOLA quando o falante fonte é uma mulher e o falante alvo é um homem. Isso pode ser explicado por uma maior mudança de *pitch*, e o fato de a excitação conter informações que são dependentes de gênero.

Os casos opostos, em que a conversão é de um homem para uma mulher, não se comportaram da mesma maneira. A conversão de $h1$ para $m1$ teve um desempenho similar, as conversões cujo alvo é a falante $m2$ obtiveram resultados muito melhores com o PSOLA e a conversão de $h2$ para $m1$ obteve melhores resultados com o pulso. Isso sugere que a informação extra não convertida está na excitação de falantes do sexo feminino.

O maior ganho do PSOLA foi nas conversões de $h1$ para $m2$ e de $h2$ para $m2$, enquanto que a maior perda que não envolve falantes de diferentes gêneros foi na conversão de $h2$ para $h1$. Essas diferenças provavelmente se dão por características dos falantes, por exemplo, características da falante $m2$ que o modelo não captura ou características do falante $h2$ que prejudiquem a qualidade do PSOLA.

Uma outra diferença notável entre a síntese por pulso paramétrico e por PSOLA é que o segundo possui, em quase todos os casos, uma taxa de acertos com confiança alta muito mais próxima da de acerto com confiança normal. Os sinais sintetizados por PSOLA possuem uma maior naturalidade, e isso pode afetar a identificação,

fazendo com que o algoritmo tenha maior certeza quando o falante identificado é o correto, mas sem evitar uma identificação errada.

Comparando os desempenhos de cada método, o modelo de misturas gaussianas novamente obteve o melhor resultado, tanto no quesito de acertos quanto no de acertos com alta confiança. A conversão por redes neurais, que possui um erro quadrático de nível médio, obteve resultados piores nesse teste, com algumas das taxas de acerto mais baixas. A relação entre a qualidade da conversão por modelo oculto de Markov e o número de estados também foi diferente da prevista pelo erro quadrático. Essas relações tendem a ser as mesmas independentemente da excitação utilizada.

Em geral, os resultados dos testes de reconhecimento de falante não possuem correlação forte com o erro quadrático. Existem diversas razões pelas quais isso pode ocorrer. Como, se os falantes fonte e alvo forem fixados, a única mudança é o método pela qual a conversão de envoltória é realizada, é de se esperar que o teste realmente avalie esses métodos. O erro quadrático é uma medida simples de qualidade, e pode ser que o método utilizado pelo sistema de reconhecimento utilize medidas mais complexas, que levem em conta características da audição humana. Por outro lado, é possível que o método utilizado pelo reconhecimento possua algum viés que não corresponda a algo perceptivo, o que seria uma artificialidade do método. Para obter mais informação sobre o desempenho do sistema de conversão, testes auditivos formais estão sendo planejados.

5.3 Comentários sobre os sinais

Essa seção apresenta comentários sobre os sinais convertidos, feitos com base em escutas informais. Os sinais podem ser ouvidos em `www02.smt.ufrj.br/~victor.costa/examples/voice_conversion/`.

A primeira observação a ser feita é que os sinais convertidos por diferentes métodos são bastante parecidos. As diferenças, quando perceptíveis, são bastante sutis. Uma razão provável para isso é que a qualidade da síntese não é suficiente para que os métodos sejam devidamente comparados.

Os métodos de síntese se comportam de maneira coerente com o que foi visto nos experimentos. O PSOLA possui maior naturalidade, e suas boas conversões são as que chegam mais próximas do ideal, como pode ser visto no Exemplo 1 da página. Por outro lado, alguns sinais gerados com PSOLA estão bastante distantes

		Taxa de Acerto (%) — GMM					
		Qualquer Confiança			Confiança Alta		
		Número de Componentes Gaussianas					
Falantes		20	50	80	20	50	80
$m1 \rightarrow m2$	Paramétrico	40	35	50	10	5	0
	PSOLA	40	35	35	40	35	35
$m1 \rightarrow h1$	Paramétrico	85	85	85	30	60	70
	PSOLA	45	65	55	45	60	50
$m1 \rightarrow h2$	Paramétrico	70	80	65	55	55	55
	PSOLA	30	25	35	30	25	35
$m2 \rightarrow m1$	Paramétrico	20	40	25	15	20	15
	PSOLA	10	15	15	10	15	15
$m2 \rightarrow h1$	Paramétrico	20	10	10	0	0	5
	PSOLA	10	35	20	5	15	5
$m2 \rightarrow h2$	Paramétrico	30	45	50	30	40	50
	PSOLA	5	5	5	5	5	5
$h1 \rightarrow m1$	Paramétrico	100	95	95	90	85	90
	PSOLA	95	95	100	90	95	95
$h1 \rightarrow m2$	Paramétrico	5	5	0	0	0	0
	PSOLA	80	80	80	65	65	70
$h1 \rightarrow h2$	Paramétrico	100	100	90	85	85	90
	PSOLA	100	95	95	100	95	95
$h2 \rightarrow m1$	Paramétrico	70	90	100	65	85	95
	PSOLA	55	55	55	55	55	55
$h2 \rightarrow m2$	Paramétrico	25	25	30	0	0	0
	PSOLA	70	65	60	70	65	60
$h2 \rightarrow h1$	Paramétrico	70	70	70	60	70	70
	PSOLA	15	20	25	15	20	25
Total	Paramétrico	52,92	56,67	55,83	36,67	42,08	45,00
	PSOLA	46,25	49,17	48,33	44,17	45,83	45,42

Tabela 5.2: Taxa de acerto do reconhecimento para Modelo de Misturas Gaussianas.

do falante alvo, principalmente quando é uma conversão de uma mulher em um homem, como no Exemplo 2. A síntese por pulso paramétrico produz resultados regulares e de qualidade mais homogênea. Os sinais não possuem qualidade tão boa, mas suas identidades percebidas tendem a ser mais próximas da do falante alvo, mesmo quando a conversão não é tão boa. Uma solução ideal é uma excitação que combine os melhores aspectos dos dois métodos, que possivelmente pode ser obtida a partir de uma modelagem paramétrica mais complexa, por exemplo.

Existem também outros problemas de menor porte. Por exemplo, em sinais como o Exemplo 3 ou Exemplo 4 pode-se perceber que alguns quadros que deveriam ser marcados como vozeados foram marcados como não-vozeados. Esse tipo de erro é muito mais aparente quando se utiliza o PSOLA, pois significa que nesse quadro

		Taxa de Acerto (%) — HMM					
		Qualquer Confiança			Confiança Alta		
		Número de Estados					
Falantes		3	5	13	3	5	13
$m1 \rightarrow m2$	Paramétrico	55	50	55	5	10	0
	PSOLA	45	35	44	45	35	44
$m1 \rightarrow h1$	Paramétrico	85	90	100	50	55	45
	PSOLA	40	45	45	35	40	40
$m1 \rightarrow h2$	Paramétrico	70	85	75	45	70	65
	PSOLA	20	30	45	20	30	40
$m2 \rightarrow m1$	Paramétrico	10	5	5	0	0	0
	PSOLA	10	5	0	10	5	0
$m2 \rightarrow h1$	Paramétrico	15	5	15	0	0	5
	PSOLA	5	5	0	5	5	0
$m2 \rightarrow h2$	Paramétrico	30	40	10	30	40	10
	PSOLA	5	5	5	5	5	5
$h1 \rightarrow m1$	Paramétrico	80	85	95	60	60	65
	PSOLA	85	95	90	85	95	90
$h1 \rightarrow m2$	Paramétrico	0	5	10	0	0	0
	PSOLA	75	70	80	60	35	50
$h1 \rightarrow h2$	Paramétrico	100	100	100	70	90	85
	PSOLA	95	95	95	95	95	95
$h2 \rightarrow m1$	Paramétrico	60	95	85	35	90	85
	PSOLA	40	55	55	40	55	55
$h2 \rightarrow m2$	Paramétrico	25	30	30	0	0	0
	PSOLA	55	60	55	55	60	55
$h2 \rightarrow h1$	Paramétrico	70	70	65	55	70	65
	PSOLA	15	20	20	15	20	20
Total	Paramétrico	50,00	55,00	53,75	29,17	40,42	35,42
	PSOLA	40,83	43,33	44,73	39,17	40,00	41,35

Tabela 5.3: Taxa de acerto do reconhecimento para Modelo Oculto de Markov.

tanto o filtro quanto a excitação são idênticos aos do falante fonte. A consequência é que a identidade do falante momentaneamente se torna a do falante fonte.

Os sinais sintetizados por ambos os métodos também apresentam defeitos que possivelmente podem ser resolvidos sem que se mude completamente o método. Sinais sintetizados com o PSOLA apresentam uma "rouquidão", que provavelmente é causada por erros nas marcas de *pitch*, ou nos instantes de fechamento glotal obtidos do sinal ou nas geradas para a síntese. Alguns sinais convertidos sintetizados por pulso paramétrico possuem potência mais alta ou baixa do que seria esperado. Esse erro ocorre principalmente com conversões que envolvem a falante $m2$, como pode ser visto no Exemplo 5 da página. Uma solução possível é incluir o ganho do filtro de síntese nos parâmetros da conversão.

		Taxa de Acerto (%) — FFNN					
		Qualquer Confiança			Confiança Alta		
		Número de Neurônios na Camada Oculta					
Falantes		20	50	80	20	50	80
$m1 \rightarrow m2$	Paramétrico	30	45	40	5	0	0
	PSOLA	40	45	40	40	45	40
$m1 \rightarrow h1$	Paramétrico	80	75	75	35	35	50
	PSOLA	45	45	45	35	30	40
$m1 \rightarrow h2$	Paramétrico	55	60	60	40	40	30
	PSOLA	15	15	15	15	15	15
$m2 \rightarrow m1$	Paramétrico	5	0	0	5	0	0
	PSOLA	0	0	0	0	0	0
$m2 \rightarrow h1$	Paramétrico	5	0	5	0	0	0
	PSOLA	10	11	10	5	5	5
$m2 \rightarrow h2$	Paramétrico	10	10	20	10	10	20
	PSOLA	0	0	0	0	0	0
$h1 \rightarrow m1$	Paramétrico	60	55	55	35	35	30
	PSOLA	70	70	70	70	65	60
$h1 \rightarrow m2$	Paramétrico	10	0	10	0	0	0
	PSOLA	85	80	80	70	55	55
$h1 \rightarrow h2$	Paramétrico	90	95	100	70	75	80
	PSOLA	90	90	95	90	90	95
$h2 \rightarrow m1$	Paramétrico	60	40	45	25	20	25
	PSOLA	25	25	30	25	25	30
$h2 \rightarrow m2$	Paramétrico	20	20	10	0	0	0
	PSOLA	55	55	55	55	55	55
$h2 \rightarrow h1$	Paramétrico	65	60	65	40	45	55
	PSOLA	20	5	10	20	5	10
Total	Paramétrico	40,83	38,33	40,42	22,08	21,67	24,17
	PSOLA	37,92	36,82	37,50	35,42	32,64	33,75

Tabela 5.4: Taxa de acerto do reconhecimento para Rede Neural *Feed-Foward*.

		Taxa de Acerto (%) — RNN					
		Qualquer Confiança			Confiança Alta		
		Número de Neurônios na Camada Recorrente					
Falantes		20	50	80	20	50	80
$m1 \rightarrow m2$	Paramétrico	35	45	47	0	5	5
	PSOLA	40	50	50	40	50	50
$m1 \rightarrow h1$	Paramétrico	85	90	65	35	40	30
	PSOLA	45	50	40	35	45	30
$m1 \rightarrow h2$	Paramétrico	55	70	55	30	45	35
	PSOLA	15	30	35	15	25	25
$m2 \rightarrow m1$	Paramétrico	0	0	0	0	0	0
	PSOLA	0	0	0	0	0	0
$m2 \rightarrow h1$	Paramétrico	5	10	5	0	0	0
	PSOLA	5	10	10	5	5	5
$m2 \rightarrow h2$	Paramétrico	15	25	20	15	25	15
	PSOLA	0	0	0	0	0	0
$h1 \rightarrow m1$	Paramétrico	70	70	60	25	20	10
	PSOLA	80	60	60	80	60	55
$h1 \rightarrow m2$	Paramétrico	0	10	5	0	0	0
	PSOLA	80	75	85	45	45	55
$h1 \rightarrow h2$	Paramétrico	100	100	95	55	80	65
	PSOLA	95	95	95	95	95	95
$h2 \rightarrow m1$	Paramétrico	55	50	40	15	15	15
	PSOLA	40	35	40	40	35	40
$h2 \rightarrow m2$	Paramétrico	25	15	20	0	0	0
	PSOLA	70	65	60	70	65	55
$h2 \rightarrow h1$	Paramétrico	60	70	60	55	45	35
	PSOLA	35	20	10	35	20	10
Total	Paramétrico	42,08	46,25	39,33	19,17	22,92	17,57
	PSOLA	42,08	40,83	40,42	38,33	37,08	35,00

Tabela 5.5: Taxa de acerto do reconhecimento para Rede Neural Recorrente.

Capítulo 6

Conclusões e Trabalhos Futuros

Nesse trabalho foram apresentadas quatro técnicas para a conversão de falantes, além de técnicas de extração de parâmetros da fala e de síntese. Um sistema base foi desenvolvido para que estas técnicas pudessem ser comparadas diretamente, e foram realizados testes para avaliar o desempenho relativo das técnicas, alem de testes para avaliar duas técnicas de síntese. Exemplos de resultados podem ser obtidos em `www02.smt.ufrj.br/~victor.costa/examples/voice_conversion/`.

Pelos testes realizados, nenhum dos métodos avaliados obteve um desempenho consistentemente melhor do que a conversão por GMM. A conversão por HMM obteve erro quadrático menor para alguns números de estados, mas para esses casos, seu resultado em testes com um sistema de reconhecimento de falantes foi pior. Os outros métodos também apresentaram discrepâncias entre os dois testes, e apenas com estes é impossível ordenar a qualidade relativa dos métodos. Por outro lado, em escutas informais todos os métodos obtiveram resultados bastante similares. Como teste adicional estão sendo planejados testes subjetivos sistemáticos, proporcionando uma comparação extra entre os métodos.

Além disso, também foram avaliadas duas técnicas para gerar a excitação utilizada na síntese. Nenhuma das duas técnicas se mostrou superior à outra, com cada uma possuindo situações em que seu desempenho é melhor. Escutas informais também concordaram com as conclusões dos testes objetivos.

Existem algumas maneiras em que esse trabalho pode ser estendido em trabalhos futuros. A primeira é a realização de mais experimentos. Além dos testes subjetivos já citados, algo que esse trabalho não mediu foi como o tamanho da base de treinamento afeta a qualidade da conversão, já que todas as conversões foram treinadas sobre uma base com tamanho fixo. Experimentos variando a base de dados podem investigar se algum método possui qualidade consideravelmente maior quando trei-

nado em uma base maior, ou qual o menor tamanho de uma base no qual a conversão ainda é convincente.

As diferenças entre o erro quadrático e a taxa de acerto do sistema de reconhecimento automático também podem motivar novos testes. Os valores de complexidade utilizados nos testes foram parcialmente motivados pelos seus erros quadráticos em testes preliminares. O sistema de reconhecimento pode ser utilizado, então, para avaliar complexidades maiores, e é possível que ele leve a resultados diferentes.

Uma outra possibilidade de trabalho futuro é a investigação de melhoras na conversão por redes neurais recorrentes. Esse método, introduzido nesse trabalho, ainda não obteve um desempenho satisfatório. Como mencionado anteriormente, redes neurais são extremamente customizáveis, e existem diversas variações e adições possíveis, e diferentes combinações podem melhorar o desempenho. A diferença entre o resultado dos dois testes realizados também indica que é possível que uma combinação apresente um melhor desempenho, mesmo que ela não necessariamente diminua o erro quadrático.

Outro trabalho futuro importante é a melhora da síntese, principalmente no que diz respeito à geração da excitação. Algumas melhorias e direções de investigação foram apresentadas na Seção 5.3. Essa etapa do sistema está relacionada à área de síntese paramétrica de fala, que é bem estudada, e pode inspirar outras soluções.

Referências Bibliográficas

[1] ABE, M., NAKAMURA, S., SHIKANO, K., et al. "Voice conversion through vector quantization". In: *Proceedings of the 1988 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, v. 1, pp. 655–658, New York, USA, April 1988. IEEE.

[2] STYLIANOU, Y., CAPPE, O., MOULINES, E. "Continuous probabilistic transform for voice conversion", *IEEE Transactions on Speech and Audio Processing*, v. 6, n. 2, pp. 131–142, March 1998.

[3] NAKASHIKA, T., TAKIGUCHI, T., ARIKI, Y. "Voice conversion in time-invariant speaker-independent space". In: *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7889–7893, Florence, Italy, May 2014. IEEE.

[4] CHEN, L.-H., LING, Z.-H., LIU, L.-J., et al. "Voice conversion using deep neural networks with layer-wise generative training", *IEEE/ACM Transactions on Audio, Speech and Language Processing*, v. 22, n. 12, pp. 1859–1872, December 2014.

[5] DESAI, S., BLACK, A. W., YEGNANARAYANA, B., et al. "Spectral Mapping Using Artificial Neural Networks for Voice Conversion", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 18, n. 5, pp. 954–964, July 2010.

[6] PILKINGTON, N. C., ZEN, H., GALES, M. J. "Gaussian process experts for voice conversion". In: *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2761–2764, Florence, Italy, August 2011.

[7] QIAO, Y., SAITO, D., MINEMATSU, N. "HMM-based sequence-to-frame mapping for voice conversion". In: *Proceedings of the 2010 International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 4830–4833, Dallas, USA, March 2010. IEEE.

[8] TODA, T., BLACK, A., TOKUDA, K. "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 15, n. 8, pp. 2222–2235, November 2007.

[9] RABINER, L. R., SCHAFER, R. W. *Introduction to Digital Speech Processing*. Hanover, USA, Now Publishers Inc., 2007.

[10] BOGERT, B. P., HEALY, M. J. R., TUKEY, J. W. "The quefrency alanysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking". In: Rosenblatt, M. (Ed.), *Proceedings of the Symposium on Time Series Analysis*, Wiley, cap. 15, pp. 209–243, New York, USA, 1963.

[11] OPPENHEIM, A. V. *Superposition in a Class of Nonlinear Systems*. Relatório Técnico 432, MIT, Research Lab. of Electronics, Massachusetts, USA, 1965.

[12] OPPENHEIM, A. V., SCHAFER, R. W., STOCKHAM JR., T. G. "Nonlinear filtering of multiplied and convolved signals", *IEEE Transactions on Audio and Electroacoustics*, v. 16, n. 3, pp. 437–466, August 1968.

[13] SCHAFER, R. W. *Echo Removal by Discrete Generalized Linear Filtering*. Relatório Técnico 466, MIT, Research Lab. of Electronics, Massachusetts, USA, 1969.

[14] KITAMURA, T., IMAI, S. "Spectral distortion and quality of synthesized speech in cepstral speech analysis-synthesis system", *Electronics and Communications in Japan (Part I: Communications)*, v. 65, n. 5, pp. 30–38, 1982.

[15] IMAI, S., FURUICHI, C. "Unbiased estimator of log spectrum and its application to speech signal processing", *Signal Processing IV: Theories and Applications – Proceedings of the 4th European Signal Processing Conference (EUSIPCO)*, v. 1, pp. 203–206, September 1988.

[16] TOKUDA, K., KOBAYASHI, T., MASUKO, T., et al. "Mel-generalized cepstral analysis-a unified approach to speech spectral estimation." In: *Proceedings of the 1994 International Conference on Spoken Language Processing (INTERSPEECH–ICSLP)*, v. 94, pp. 1043–1046, Yokohama, Japan, September 1994.

[17] STEVENS, S. S., VOLKMANN, J., NEWMAN, E. B. "A scale for the measurement of the psychological magnitude pitch", *The Journal of the Acoustical Society of America*, v. 8, n. 3, pp. 185–190, January 1937.

[18] ZWICKER, E. "Subdivision of the audible frequency range into critical bands (FrequenzGruppen)", *The Journal of the Acoustical Society of America*, v. 33, n. 2, pp. 248–248, February 1961.

[19] MOORE, B. C., GLASBERG, B. R. "Suggested formulae for calculating auditory-filter bandwidths and excitation patterns", *The Journal of the Acoustical Society of America*, v. 74, n. 3, pp. 750–753, 1983.

[20] STRUBE, H. W. "Linear prediction on a warped frequency scale", *The Journal of the Acoustical Society of America*, v. 68, n. 4, pp. 1071–1076, October 1980.

[21] HÄRMÄ, A., KARJALAINEN, M., SAVIOJA, L., et al. "Frequency-warped signal processing for audio applications", *Journal of the Audio Engineering Society*, v. 48, n. 11, pp. 1011–1031, November 2000.

[22] TOKUDA, K., KOBAYASHI, T., IMAI, S. *Recursive Calculation of Mel-Cepstrum from LP Coefficients*. Relatório técnico, Nagoya Institute of Technology, Nagoya, Japan, 1994.

[23] KUWABARA, H., SAGISAKA, Y. "Acoustic characteristics of speaker individuality: control and conversion", *Speech Communications*, v. 16, n. 2, pp. 165–173, February 1995.

[24] KAIN, A., MACON, M. W. "Spectral voice conversion for text-to-speech synthesis". In: *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, v. 1, pp. 285–288, Seattle, USA, May 1998. IEEE.

[25] REYNOLDS, D. A., ROSE, R. C. "Robust text-independent speaker identification using Gaussian mixture speaker models", *IEEE Transactions on Speech and Audio Processing*, v. 3, n. 1, pp. 72–83, January 1995.

[26] EVERITT, B. S., HAND, D. J. *Finite Mixture Distributions*. Monographs on Applied Probability and Statistics. London, UK, Chapman and Hall, 1981.

[27] DEMPSTER, A. P., LAIRD, N. M., RUBIN, D. B. "Maximum likelihood from incomplete data via the EM algorithm", *Journal of the Royal Statistical Society. Series B (Methodological)*, v. 39, n. 1, pp. 1–38, 1977.

[28] MERHAV, N., LEE, C.-H. "On the asymptotic statistical behavior of empirical cepstral coefficients", *IEEE Transactions on Signal Processing*, v. 41, n. 5, pp. 1990–1993, May 1993.

[29] KIM, E.-K., LEE, S., OH, Y.-H. "Hidden Markov model based voice conversion using dynamic characteristics of speaker." In: *Proceedings of the 5th European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 2519–2522, Rhodes, Greece, September 1997. ISCA.

[30] DUXANS, H., BONAFONTE, A., KAIN, A., et al. "Including dynamic and phonetic information in voice conversion systems". In: *Proceedings of the 8th International Conference on Spoken Language Processing (INTERSPEECH–ICSLP)*, pp. 1193–1196, Jeju Island, Korea, October 2004. ISCA.

[31] WU, C.-H., HSIA, C.-C., LIU, T.-H., et al. "Voice conversion using duration-embedded bi-HMMs for expressive speech synthesis", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 14, n. 4, pp. 1109–1116, July 2006.

[32] RABINER, L., JUANG, B. "An introduction to hidden Markov models", *IEEE ASSP Magazine*, v. 3, n. 1, pp. 4–16, January 1986.

[33] BAUM, L. E., PETRIE, T., SOULES, G., et al. "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains", *The Annals of Mathematical Statistics*, v. 41, n. 1, pp. 164–171, 1970.

[34] MURPHY, K. P. *Machine Learning: a Probabilistic Perspective*. Cambridge, MA, MIT press, 2012.

[35] RAO, K. S. "Voice conversion by mapping the speaker-specific features using pitch synchronous approach", *Computer Speech & Language*, v. 24, n. 3, pp. 474–494, July 2010.

[36] WATANABE, T., MURAKAMI, T., NAMBA, M., et al. "Transformation of spectral envelope for voice conversion based on radial basis function networks." In: *Proceedings of the 7th International Conference on Spoken Language Processing (INTERSPEECH–ICSLP)*, pp. 285–288, Denver, USA, September 2002. ISCA.

[37] NARENDRANATH, M., MURTHY, H. A., RAJENDRAN, S., et al. "Transformation of formants for voice conversion using artificial neural networks", *Speech Communication*, v. 16, n. 2, pp. 207–216, February 1995.

[38] GYBENKO, G. "Approximation by superposition of sigmoidal functions", *Mathematics of Control, Signals and Systems*, v. 2, n. 4, pp. 303–314, December 1989.

[39] RUMELHART, D. E., HINTON, G. E., WILLIAMS, R. J. "Learning representations by back-propagating errors", *Nature*, v. 323, pp. 533–536, October 1986.

[40] BISHOP, C. *Pattern Recognition and Machine Learning*. Secaucus, USA, Springer, 2006.

[41] SCHÄFER, A. M., ZIMMERMANN, H. G. "Recurrent neural networks are universal approximators", *International Journal of Neural Systems*, v. 17, n. 4, pp. 253–263, August 2006.

[42] WILLIAMS, R. J., ZIPSER, D. "A learning algorithm for continually running fully recurrent neural networks", *Neural Computation*, v. 1, n. 2, pp. 270–280, 1989.

[43] ALCAIM, A., SOLEWICZ, J. A., MORAES, J. A. "Freqüência de ocorrência dos fones e listas de frases foneticamente balanceadas no português falado no Rio de Janeiro", *Revista da Sociedade Brasileira de Telecomunicações*, v. 7, n. 1, pp. 23–41, dezembro 1992.

[44] ERRO, D., MORENO, A., BONAFONTE, A. "INCA algorithm for training voice conversion systems from nonparallel corpora", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 18, n. 5, pp. 944–953, July 2010.

[45] DRUGMAN, T., THOMAS, M., GUDNASON, J., et al. "Detection of glottal closure instants from speech signals: A quantitative review", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 20, n. 3, pp. 994–1006, March 2012.

[46] DRUGMAN, T., DUTOIT, T. "Glottal closure and opening instant detection from speech signals". In: *Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2891–2894, Brighton, UK, September 2009. ISCA.

[47] RABINER, L., CHENG, M., ROSENBERG, A., et al. "A comparative performance study of several pitch detection algorithms", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 24, n. 5, pp. 399–418, October 1976.

[48] DE PAIVA, R. C. D. *Transformações em Sinais de Voz: Morphing e Modificação de Pitch*. Tese de doutorado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2008.

[49] MOULINES, E., LAROCHE, J. "Non-parametric techniques for pitch-scale and time-scale modification of speech", *Speech Communication*, v. 16, n. 2, pp. 175–205, February 1995.

[50] KAWAHARA, H., ESTILL, J., FUJIMURA, O. "Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system STRAIGHT". In: *Proceedings of the 2nd International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications*, pp. 59–64, Firenze, Italy, September 2001. ISCA.

[51] YOSHIMURA, T., TOKUDA, K., MASUKO, T., et al. "Mixed excitation for HMM-based speech synthesis." In: *Proceedings of the 7th European Conference on Speech Communication and Technology (INTERSPEECH)*, pp. 2263–2266, Aalborg, Denmark, September 2001. ISCA.

[52] DRUGMAN, T., DUTOIT, T. "The deterministic plus stochastic model of the residual signal and its applications", *IEEE Transactions on Audio, Speech, and Language Processing*, v. 20, n. 3, pp. 968–981, March 2012.

[53] IMAI, S., SUMITA, K., FURUICHI, C. "Mel log spectrum approximation (MLSA) filter for speech synthesis", *Electronics and Communications in Japan (Part I: Communications)*, v. 66, n. 2, pp. 10–18, 1983.

[54] MASUKO, T. *HMM-Based Speech Synthesis and its Applications*. Tese de doutorado, Tokyo Institute of Technology, Tokyo, Japan, 2002.

[55] "PMTK - Probabilistic Modeling Toolkit". `github.com/probml/pmtk3`, 2010. Acessado: 2017-03-12.

[56] "Keras". `keras.io`, 2015. Acessado: 2017-03-12.

[57] HOCHREITER, S., SCHMIDHUBER, J. "Long Short-Term Memory", *Neural Computation*, v. 9, n. 8, pp. 1735–1780, November 1997.

[58] ZAREMBA, W., SUTSKEVER, I., VINYALS, O. "Recurrent neural network regularization", *ArXiv e-prints*, September 2014.

[59] "Microsoft Cognitive Services - Speaker Recognition API". `www.microsoft.com/cognitive-services/en-us/speaker-recognition-api`, 2016. Acessado: 2017-03-12.