UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

José de Sousa Ribeiro Filho

Extração de Parâmetros de um Sintetizador por Formantes Utilizando Rede Neural e Algoritmo Genético

Belém
2015

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

José de Sousa Ribeiro Filho

Extração de Parâmetros de um Sintetizador por Formantes Utilizando Rede Neural e Algoritmo Genético

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Pará como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Belém
2015

José de Sousa Ribeiro Filho

Extração de Parâmetros de um Sintetizador por Formantes Utilizando Rede Neural e Algoritmo Genético

Dissertação de Mestrado apresentada para obtenção do grau de Mestre em Ciência da Computação. Programa de Pós-Graduação em Ciência da Computação (área de concentração: Sistemas Inteligentes). Instituto de Ciências Exatas e Naturais. Universidade Federal do Pará.

Banca Examinadora:

Prof. Dr. Adrião Duarte Dória Neto
Universidade Federal do Rio Grande do Norte - UFRN - Membro

Prof. Dr. Aldebaro Barreto da Rocha Klautau Júnior
Instituto de Tecnologia - UFPA - Orientador

Prof. Dr. Jefferson Magalhães de Morais
Instituto de Ciências Exatas e Naturais - UFPA - Membro

Prof. Dra. Yomara Pinheiro Pires
Instituto de Tecnologia - UFPA - Membro

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Extração de Parâmetros de um Sintetizador por Formantes Utilizando Rede Neural e Algoritmo Genético

Autor: José de Sousa Ribeiro Filho

Prof. Dr. Aldebaro Barreto da Rocha Klautau Júnior
(ORIENTADOR)

José de Sousa Ribeiro Filho
(ALUNO)

UFPA-ICEN-PPGCC
CAMPOS UNIVERSITÁRIO DO GUAMÁ
BELÉM-PARÁ-BRASIL
2015

Agradecimentos

Primeiramente agradeço a tudo que Deus é na minha vida.

Aos meus familiares que me deram apoio em tudo que precisei durante estes dois anos de trabalho duro no mestrado. Principalmente à minha mãe Jandira de Sousa e minha irmã Jime Rodrigues, que me ajudaram nos momentos tristes e felizes da vida em família. E ao meu pai (falecido a cinco anos), que me serviu de companhia nos momentos que me senti mais sozinho nesta caminhada. Família, amo muito vocês.

À minha namorada, que nos momentos de lazer ouviu minhas explicações referentes a como uma Rede Neural Artificial e um Algoritmo Genético podem ser capazes de produzir sinais de voz similares a fala humana. Missão complicada explicar tais assuntos à uma Futura Fisioterapeuta. Beijos Amor.

Ao meu professor orientador, Dr. Aldebaro Klautau, que durante toda a pesquisa não meu deixou sem sua orientação, mesmo sendo uma pessoa com grandes responsabilidades na UFPA ou mesmo estando do outro lado do mundo. Agradeço muito por toda a contribuição que você me deu nesta conquista.

Agradeço também a uma pessoa que posso dizer ser minha irmã de pesquisa, futura Dra. Fabíola Araújo, muito obrigado pelos longos momentos de conversas e análises sobre o problema de imitação da fala humana, pois sem sua contribuição eu não teria chegado onde cheguei.

A todos os meus amigos que diretamente ou indiretamente me ajudaram durante estes dois últimos anos, segue meu muito obrigado e saibam que do fundo do meu coração divido esta conquista com todos vocês.

Este trabalho é dedicado à minha família e amigos

Resumo

Sintetizadores de fala são softwares largamente utilizados em estudos relacionados à síntese de fala, como em problemas do tipo *Text-To-Speech*, onde são produzidos sinais de fala a partir de texto. Um problema similar, porém de natureza diferenciada, é a *Utterance Copy* (em português, 'imitação da fala') que consiste na realização do processo de reconstrução de um sinal de voz, tomando-se como base a própria voz original. Este processo é de alta complexidade, uma vez que sua metodologia é baseada na combinação dos parâmetros numéricos de um sintetizador de fala (por exemplo, *HLSyn* ou mesmo *KLSYN88*) de maneira que tais configurações representem adequadamente a fala alvo. Esta pesquisa consiste na proposta de três técnicas implementadas a partir dos frameworks *Neural Speech Mimic (NSM)* e *High Level Speech processed by Genetic Algorithm (HLSpeechGA)*. A primeira proposta é o *NSM*, ferramenta baseada em redes neurais que tem como principal objetivo extrair aproximações do que vêm a ser os parâmetros do sintetizador de fala *HLSyn*. A segunda proposta é o *HLSpeechGA*, ferramenta fundamentada em algoritmo genético capaz de extrair parâmetros do sintetizador *HLSyn* a partir de otimização. Já a terceira proposta é a união dos dois frameworks, chamando-se assim *NSM+HLSpeechGA*, ou seja, uma abordagem mista. Nos estudos realizados nesta pesquisa são comparados resultados obtidos a partir das três propostas defendidas juntamente com resultados produzidos a partir do software *WinSnoori*, utilizado como *baseline*. De acordo com as figuras de mérito adotadas, o framework *HLSpeechGA* obteve desempenho superior ao *baseline* tanto em processos utilizando voz sintética como natural.

Palavras-chave: HLSyn, KLSYN88, Algoritmo Genético, Sintetizador, Rede Neural.

Abstract

Speech synthesizers are widely used in studies related to speech synthesis, in problems like Text-To-Speech, that is the production of voice signals from text. A similar problem although of a different nature is the Utterance Copy which consists in the reconstruction process of a speech signal, taking as basis the original voice. This process is highly complex, since its methodology is based on a combination of numerical parameters of a speech synthesizer (for example, HLSyn or KLSYN88), so that these settings adequately represent speech target. This research is the proposal of three techniques implemented from the frameworks Neural Speech Mimic (NSM) and High Level Speech processed by Genetic Algorithm (HLSpeechGA). The first proposal is the NSM, a tool based on neural networks which performs the extraction of HLSyn's parameters approximations. The second proposal is the HLSpeechGA, a tool based on genetic algorithm capable of extracting HLSyn's settings from optimization. The third proposal is the union of the two frameworks, the so called NSM+HLSpeechGA. Results obtained from these three proposals are compared with results produced with the software WinSnoori, used in this study as baseline. It should be noted that the developed framework HLSpeechGA outperformed the baseline in all processes using synthetic speech and natural speech.

Keywords: HLSyn, KLSYN88, Genetic Algorithm, Synthesizer, Neural Network.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

NSM *Neural Speech Mimic*
RNA *Rede Neural Artificial*
GRNN *Generalized Regressive Neural Network*
RBF *Radial Bases Function*
HLSyn *High Level Synthesizer*
KLSYN88 *Klatt Synthesizer 1988*
HLSpeechGA *High Level Speech processed by Genetic Algorithm*
AG *Algoritmo Genético*
NSGA-II *Non-Dominated Sorting Genetic Algorithm II*
PDS *Processamento Digital de Sinais*
HTK *Hidden Markov Toolkit*
MFCC *Mel-Frequency Ceptral Coefficients*
PLP *Perceptual Linear Prediction*
FBANK *Filter Bank*
DE *Distorção Espectral*
RMSE *Root-mean-square Error*
PESQ *Perceptual Evaluation of Speech Quality*
P.563 *Single-ended*
SNR *Signal to Noise Ratio*
TTS *Text-to-Speech*

Capítulo 1

Introdução

1.1 Motivação

A atual necessidade dentro da área de voz em explorar novos procedimentos, que simplifiquem e produzam melhores resultados, no processo de imitação da fala (*utterance copy*) é um fato que instiga grupos de pesquisa e empresas a realizarem investimentos nesta área. Desta forma, técnicas foram desenvolvidas e aperfeiçoadas através de anos a partir de pesquisas sobre o tema.

Pesquisas relacionadas ao problema de *utterance copy* podem não suscitar o mesmo interesse do mercado como as pesquisas relacionadas a *text-to-speech (TTS)*, porém deve-se destacar que para áreas como a Fonética, as pesquisas sobre o problema aqui abordado são de relevante importância.

A existência de poucos softwares que realizam o processo de imitação da fala, baseando-se em sintetizadores por formantes, é uma evidência de como esta área de estudos tem sido pouco explorada nos últimos anos. Observando a tendência de importantes trabalhos da área de voz relacionados ao problema como [Honda et al. 1999], [Kanda et al. 2007] e [Aylett e Yamagishi 2008], percebe-se que a utilização de RNAs ou mesmo AGs no processo são técnicas pouco exploradas neste meio. Sendo assim, estas técnicas representam uma fonte de resultados importantes para a comunidade de pesquisa de voz.

O desenvolvimento de ferramentas com o objetivo de realizar o processo de imitação da fala é um fato presente em pesquisas anteriores [Liu e Kewley-Port 2004] [Hansona e Stevens 2002] [Bangayan et al. 1997]. A criação de uma versão digital da voz de uma pessoa específica, ou mesmo suporte a pessoas que perderam a fala por motivo de doença são exemplos de aplicações de técnicas de imitação da fala na área da saúde, motivações estas importantes para as pesquisas relacionadas ao assunto.

Pesquisadores e profissionais da área de Fonética realizam a manipulação dos parâmetros de sintetizadores por formantes com o intuito de analisar, em ambiente controlado, os principais aspectos que envolvem a fisiologia da voz humana [Liu e Kewley-Port 2004]. Tal procedimento é demorado e trabalhoso ao ser realizado manualmente. Um dos importantes desafios desta área é desenvolver um processo de extração dos parâmetros dos sintetizadores de maneira automática. Apesar de atualmente se ter ferramentas que permitam tal feito, como o *baseline* deste trabalho, ainda espera-se uma maior exploração deste problema pela comunidade científica, a fim de se desenvolver uma técnica que proporcione melhores resultados. A presente dissertação encontra-se alinhada a este propósito.

1.2 Trabalhos relacionados

De maneira a mapear os principais trabalhos relacionados ao assunto abordado por esta pesquisa, foi realizado um estudo bibliográfico. Neste estudo verificou-se a existência de propostas que trabalham com o mesmo problema apresentado pela pesquisa aqui descrita (*utterance copy*) e também propostas similares.

Inseridas nas temáticas Reconhecimento de Voz, Síntese de Voz e Imitação da Fala (*utterance copy*), destacaram-se as iniciativas listadas abaixo, listando-se as mais importantes primeiro:

- *newGASpeech: A genetic algorithm with look-ahead mechanism to estimate formant synthesizer input parameters* [Trindade et al. 2013] (Trabalho realizado na UFPA);
- *The WinSnoori user's manual version 1.32* [Laprie 2002].
- *Procsy: A Hybrid Approach to High-Quality Formant Synthesis Using HLSYN* [Heid e Hawkins 1998];
- *STRAIGHT: A new speech synthsizer for vowel formant discrimination* [Liu e Kewley-Port 2004].
- *Analysis by synthesis of pathological voices using the Klatt synthesizer* [Bangayan et al. 1997];
- *Automatic speech synthesizer parameter estimation using HMMs* [Donovan e Woodland 1995];
- *Vocal imitation using physical vocal tract model* [Kanda et al. 2007].

O *newGASpeech* [Trindade et al. 2013] é um framework baseado em Algoritmo Genético que realiza o processo de estimação dos parâmetros do sintetizador de voz da família

Klatt chamado *KLSYN88* [Klatt e Klatt 1990*a*]. Tal framework, chamado inicialmente *GASpeech*, foi desenvolvido inicialmente na Universidade Federal do Pará (UFPA) por [Borges et al. 2008], dando início ao uso de algoritmo genético multi-objetivo como técnica capaz de realizar o processo de imitação da voz humana. As principais diferenças entre as versões *GASpeech* e *newGASpeech* são: implementações de novas funções objetivo, implementações de novas arquiteturas gerais e maior especialização do programa desenvolvido ao problema proposto.

O framework desenvolvido no trabalho aqui descrito é chamado *High Level Speech processed by GA (HLSpeechGA)* e pode ser visto como uma nova versão do *newGASpeech*. A principal diferença dentre estas duas versões é que na nova versão é utilizado o sintetizador *HLSyn* [Heid e Hawkins 1998] no lugar do *KLSYN88* [Klatt e Klatt 1990*a*] juntamente com o desenvolvimento de um framework que pode ser utilizado como auxiliar na extração de parâmetros do tipo *HLSYN* utilizando para isso rede neural (chamado de *Neural Speech Mimic - NSM*).

WinSnoori [Laprie 2002] é uma ferramenta utilizada para realização do processo de imitação da fala que conta com uma interface visual que permite acompanhar as trajetórias dos principais parâmetros do sintetizador *Klatt* existente em sua arquitetura. Deve-se destacar que este software utiliza uma versão modificada do sintetizador *Klatt80* [Klatt 1980], sendo que tais modificações são descritas em [Laprie 2002].

Já o *Procsy* [Heid e Hawkins 1998] é uma ferramenta pertencente ao projeto *ProSynth* e tem como principal objetivo realizar o processo de imitação da fala utilizando como base o mesmo núcleo interno encontrado no sintetizador *HLSyn* [Hanson et al. 1999]. Os processos realizados nesta ferramenta são baseados em regras chamadas *Procsy Rules* que foram desenvolvidas com o intuito de mapear os principais comportamentos dos valores dos parâmetros do sintetizador *HLSyn* em relação às características do sinal de voz alvo.

Experimentos com [Heid e Hawkins 1998] atualmente só podem ser realizados com a base de dados do *Procsy*, chamada de *ProSynth*, tal limitação é o que impediu que esta ferramenta fosse utilizada como *baseline* da pesquisa aqui defendida. Sendo que deve-se destacar que o projeto *Procsy* foi descontinuado.

STRAIGHT [Liu e Kewley-Port 2004], é uma ferramenta desenvolvida em *MATLAB* e conta com um conjunto de bibliotecas proprietárias que possibilitam a extração/execução de importantes características/processos relacionados a voz, como: visualização de espectrogramas, extração de *F0*, extração de coeficientes de frequências de aperiodicidades, análise de *pitch* e síntese de voz.

A metodologia de funcionamento do *STRAIGHT* não se baseia na extração de parâmetros de sintetizadores de fala, o que torna não tão simples o processo de interpretação

das regras referentes à produção de um sinal de voz específico. Por este motivo, este software não é utilizado como *baseline* desta pesquisa.

A pesquisa [Bangayan et al. 1997] faz o estudo quantitativo do desempenho do sintetizador *KLSYN88* aplicado na realização do processo de *utterance copy* em sinais de voz patológica. Neste estudo são sugeridas modificações que podem ser realizadas no referido sintetizador, com o objetivo de torna-lo apto a síntese de voz patológica.

O trabalho [Donovan e Woodland 1995] é uma iniciativa que defende a utilização de árvores de decisão juntamente com *Hidden Markov Models (HMMs)* aplicados ao problema de *utterance copy* (no trabalho, referenciado como *speech mimic*). O diferencial da metodologia desta iniciativa, além do uso de árvore de decisão e *HMMs*, é o fato desta se basear na extração dos parâmetros de um sintetizador por concatenação para realizar o processo de *utterance copy*. Ou seja, não é utilizado um sintetizador por formantes, os quais são adotados por este trabalho pelo seu alto grau de interpretabilidade de seus parâmetros.

Sintetize por concatenação [Donovan e Woodland 1995] trabalha sobre as minuciosas características de um sinal de voz, dividindo-o em múltiplos segmentos sendo capaz de realizar modificação em sua duração ou mesmo eu seu *pitch*. Devido às diferenças entre a interpretabilidade dos parâmetros dos sintetizadores de concatenação e os sintetizadores por formantes (adotados pela pesquisa aqui descrita) uma comparação entre estes não se faz justa.

A pesquisa em [Kanda et al. 2007], consiste na proposta de um sistema baseado em redes neurais recorrentes e um modelo físico de trato vocal capaz de produzir sons similares a voz humana. A metodologia desenvolvida por tal pesquisa direciona a mesma ao problema de imitação vocal, não necessariamente o processo de *utterance copy*.

O problema de imitação vocal trabalhado em [Kanda et al. 2007], pode ser entendido como um problema diferente da *utterance copy*, já que não é realizado imitação da voz com base na própria voz, mas sim é realizada a produção da voz sintética com base na produção de movimentos articulatórios gerados por um trato vocal sintético.

Portanto, é utilizado como *baseline* da pesquisa aqui descrita o software *WinSnoori*, já que este busca resolver o mesmo problema de *utterance copy* que as propostas aqui apresentadas, usando para isso sintetizadores por formantes. Valendo-se destacar que mesmo que o *WinSnoori* possua um sintetizador da família *Klatt* como base para síntese de voz, a comparação com a metodologia apresentada por esta pesquisa (baseada no sintetizador *HLSyn*) é justa, dado que interno ao sintetizador *HLSyn* existe um sintetizador da família *Klatt* (como descrito no Capítulo 2).

1.3 Metodologia e contribuições

Este trabalho busca avaliar o desempenho, do processo de imitação da fala, realizado a partir do framework *NSM* e também do *HLSpeechGA*, desenvolvidos durante a pesquisa. Tais frameworks são baseados respectivamente em Rede Neural e Algoritmo Genético. A partir da combinação das técnicas citadas esta pesquisa avalia três possíveis soluções para o problema de *utterance copy*, ou seja, a primeira baseada em algoritmo genético, uma segunda baseada em redes neurais e uma terceira baseada na combinação destas.

O estudo aqui apresentado nasceu de uma necessidade do framework *newGASpeech*, ou seja, dado o custo computacional elevado para a realização do processo de imitação da fala, optou-se por trocar o sintetizador de voz *KLSYN88* que possui 48 parâmetros de entrada pelo sintetizador *HLSyn* que apresenta 13 parâmetros somente. Desta forma, uma vez que se tem menos parâmetros tem-se menos combinações possíveis e daí um espaço de busca menor.

Além da mudança de sintetizador, o que já proporciona uma abordagem completamente diferente ao problema, optou-se ainda por criar um framework chamado *Neural Speech Mimic (NSM)* que realiza o pré-processamento do sinais de voz a ser copiado gerando pseudo-parâmetros do tipo *HLSyn*. Tais parâmetros podem ou não serem repassados para o framework *HLSpeechGA* de maneira a acelerar o processo de convergência do algoritmo. Desta forma, por se tratar de um problema difícil (*utterance copy*), tais opções (utilizar somente rede neural, utilizar somente algoritmo genético e utilizar rede neural juntamente com algoritmo genético) são três propostas à solução do problema de *utterance copy* avaliadas por esta pesquisa.

Os experimentos realizados por este trabalho comparam os resultados obtidos pelo *NSM*, *HLSpeechGA*, *HLSpeechGA+NSM* e *WinSnoori*, para voz natural e sintética, utilizando como métricas: *Root-mean-square Error (RMSE)*, Distorção Espectral (DE), *Signal to Noise Ratio (SNR)* e *Perceptual Evaluation of Speech Quality (PESQ)* [*ITU-T Recommendation P.862. Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-To-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs* 2001].

1.4 Publicação

Esta pesquisa teve como publicação o artigo completo apresentado em importante workshop internacional da área de voz, a SLT 2014. A seguir a referência de tal publicação:

- Sousa, J.; Araújo, F.; Klautau, A. *UTTERANCE COPY FOR KLATT'S SPEECH SYNTHESIZER USING GENETIC ALGORITHM*. IEEE Spoken Language Technology Workshop - SLT 2014, South Lake Tahoe - NV, United States, 2014.

1.5 Objetivos da pesquisa

1.5.1 Objetivo Geral

Avaliar objetivamente o desempenho de três técnicas aplicadas ao problema de *utterance copy* e compara-las ao resultados obtidos com o *baseline* da pesquisa. Sendo tais técnicas implementadas a partir dos frameworks *Neural Speech Mimic (NSM)* e *High Level processed by Genetic Algorithm (HLSpeechGA)*. A primeira proposta é utilizar o *NSM* Figura 1.1 (A), ferramenta baseada em rede neural, para extrair os parâmetros do sintetizador *HLSyn*. A segunda proposta é utilizar o *HLSpeechGA* Figura 1.1 (B), ferramenta baseada em algoritmo genético, para extrair os parâmetros do sintetizador anteriormente citado. Já a terceira proposta é o *NSM+HLSpeechGA* Figura 1.1 (C), uma abordagem mista, aplicada à mesma finalidade.

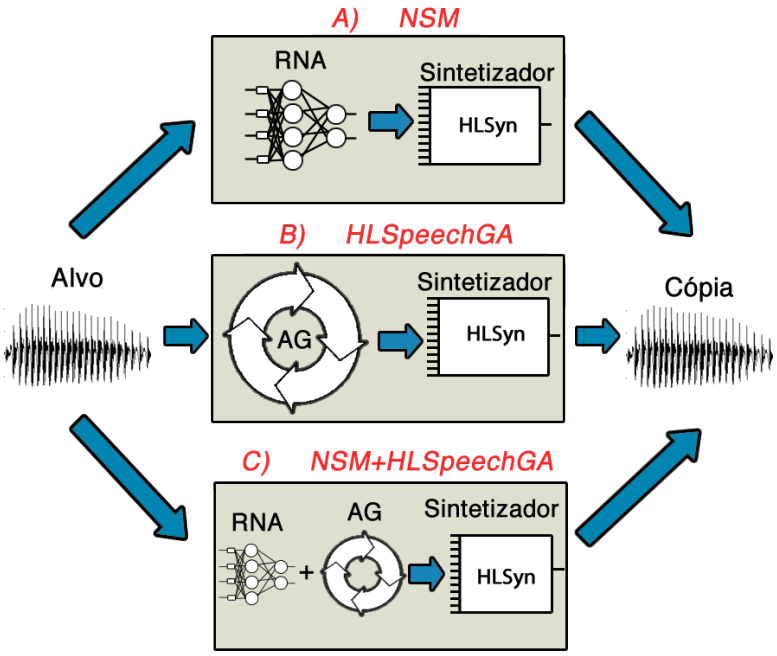

Figura 1.1: Proposta das técnicas: *NSM* (A), *HLSpeechGA* (B) e *NSM+HLSpeechGA* (C).

1.5.2 Objetivos Específicos

- Realizar estudo bibliográfico sobre iniciativas de pesquisas que buscam trabalhar com o problema de imitação da fala;
- Levantar as particularidades relacionadas à solução do problema de imitação da fala tomando-se como base sintetizadores por formantes;
- Avaliar o desempenho das técnicas de aprendizagem de máquina Algoritmo Genético e Rede Neural aplicadas ao problema de imitação da fala;
- Comparar o desempenho das técnicas utilizadas na pesquisa, ou seja, resultados gerados pelas três propostas desenvolvidas juntamente com os resultados gerados pelo *baseline* da pesquisa;
- Destacar quais das técnicas abordadas apresentam resultados relevantes à comunidade de voz;
- Justificar os desempenhos de cada uma das técnicas avaliadas.

1.6 Organização do trabalho

Este trabalho é organizado em 6 capítulos que descrevem de maneira contextualizada o conteúdo da pesquisa realizada. A seguir é feita uma breve apresentação de cada um destes capítulos.

- **Capítulo 1:** Apresenta uma introdução geral sobre os principais assuntos teóricos trabalhados durante a pesquisa juntamente com a descrição geral da metodologia e contribuições da mesma, destacando ao final a publicação e os objetivos da pesquisa aqui realizada.
- **Capítulo 2:** Apresenta os principais conceitos referentes à tecnologia de síntese de voz baseada em sintetizadores de voz por formantes e apresenta também o problema de *utterance copy* juntamente com seus principais aspectos.
- **Capítulo 3:** Este capítulo faz a contextualização do trabalho com relação aos conceitos dos algoritmos inteligentes utilizados na metodologia da pesquisa. Desta forma, em cada tópico deste capítulo são apresentadas introduções, conceitos e conclusões relacionadas aos algoritmos de Rede Neural, Algoritmo Genético e Algoritmo de Redução da Dimensionalidade.
- **Capítulo 4:** Apresenta de maneira detalhada os principais aspectos da construção dos frameworks *Neural Speech Mimic (NSM)* e o *High Level Speech processed by Genetic Algorithm (HLSpeechGA)*, juntamente com a metodologia por trás destes softwares, que de maneira simplificada são o foco do trabalho aqui apresentado.

- **Capítulo 5:** Demonstra os principais experimentos tomados como prova de conceito à proposta de metodologia desta pesquisa.
- **Capítulo 6:** Realiza o fechamento sobre o que foi a pesquisa, apresentando as principais conclusões e trabalhos futuros.

Capítulo 2

Sintetizadores de Voz

O problema de *utterance copy* é intimamente relacionado a sintetizadores da fala. Este fato é comumente observado em metodologias como [Laprie 2002] e [Heid e Hawkins 1998], onde são realizados processos de estimação dos parâmetros de sintetizadores por formantes como forma de resolver o problema citado. Neste capítulo serão discutidos os principais aspectos relacionados à sintetizadores de voz.

2.1 Introdução

De acordo com os estudos realizados em [Oliveira et al. 2011], o processo de síntese de voz consiste na produção da voz humana artificialmente, utilizando um gerador automático de sinal de voz. Sendo que, de maneira a mensurar a qualidade de tal voz, são consideradas a naturalidade e a inteligibilidade da voz artificial. Vários trabalhos relacionados à síntese de voz vêm sendo desenvolvidos há décadas. Diversos avanços foram alcançados, porém a qualidade da voz artificial, considerando a naturalidade, entonação e emoção, ainda são grandes desafios que ainda formam uma lacuna em processos computacionais relacionados à fala [Neto et al. 2005].

Dados históricos comprovam que por volta de 1779, houve o surgimento das primeiras pesquisas relacionadas ao reconhecimento e produção de voz. Um destes dados é a pesquisa que o professor russo *Christian Kratzenstein* realizou, criando um ressonador acústico similar ao trato vocal, no qual era possível produzir sons de vogais. Tal pesquisa serviu de base para outros pesquisadores como *Wolfgang von Kempelen* e *Charles Wheatstones*, que deram sua contribuição à área de voz construindo máquinas como a da Figura 2.1, por exemplo, que mostra uma gravura de um ressonador acústico construído na época.

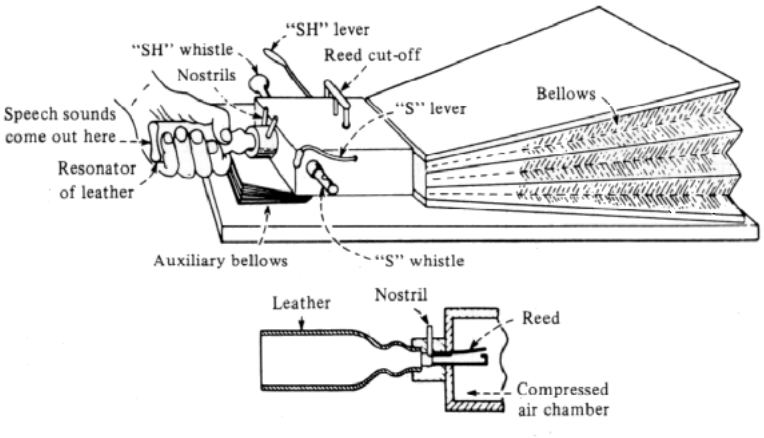

Figura 2.1: Máquina falante criada por von Kempelen.
Fonte: Wheatstone 1835.

O primeiro sintetizador elétrico foi construído por *Homer Dudley* em 1939, O *Voice Operating Demonstrator* (VODER). O equipamento era formado por uma barra que era utilizada para selecionar o tipo de voz (vozeada e não vozeada), um pedal para controlar a frequência fundamental e dez teclas que controlavam o trato vocal artificial. A estrutura do VODER se assemelha aos atuais sistemas baseado no modelo fonte-filtro.

Vale destacar a diferença entre sinais de voz vozeados e não vozeados, já que esta característica do sinal de voz é determinante no processo de síntese e funciona como chave básica de controle do sintetizador. Sinais de voz vozeados são aqueles gerados com ajuda das vibrações oriundas das cordas vocais, já os sinais de voz não vozeados são sinais gerados sem a ajuda direta das vibrações das cordas vocais e portanto são entendidos como sussurros [Klatt 1980].

Sintetizadores de fala por formantes, são modelos computacionais de sintetizadores baseados em estudos como a máquina falante de *von Kempelen*. Tais softwares são utilizados comumente em pesquisas e processos relacionados à voz humana nas áreas de: Reconhecimento da fala, Síntese de voz, Processamento de sinais, entre outros. Estes programas sãos baseados em princípios fisiológicos da formação natural da fala humana, desta forma este tipo de ferramenta simula o funcionamento de órgãos humanos como: língua, cavidade bocal, faringe, glote, etc, órgãos fundamentais para o processo de produção da voz [Klatt e Klatt 1990*b*].

2.2 Sintetizador *KLSYN88*

Sintetizadores da família *Klatt* usam um mecanismo baseado no modelo fonte-filtro [Klatt 1980] que permitem a modelagem do trato vocal através de um filtro linear, com resonadores em paralelo ou/e cascata que variam com o tempo. Existem variadas versões de sintetizadores da família *Klatt* como: *Klatt80* [Klatt 1980], *KLSYN88* [Klatt e Klatt 1990*a*] e uma versão modificada do *Klatt80* por *Jon Iles* [Laprie e Bonneau 2002]. A visão geral da disposição de tais estruturas do sintetizador em questão pode ser visualizada na Figura 2.2.

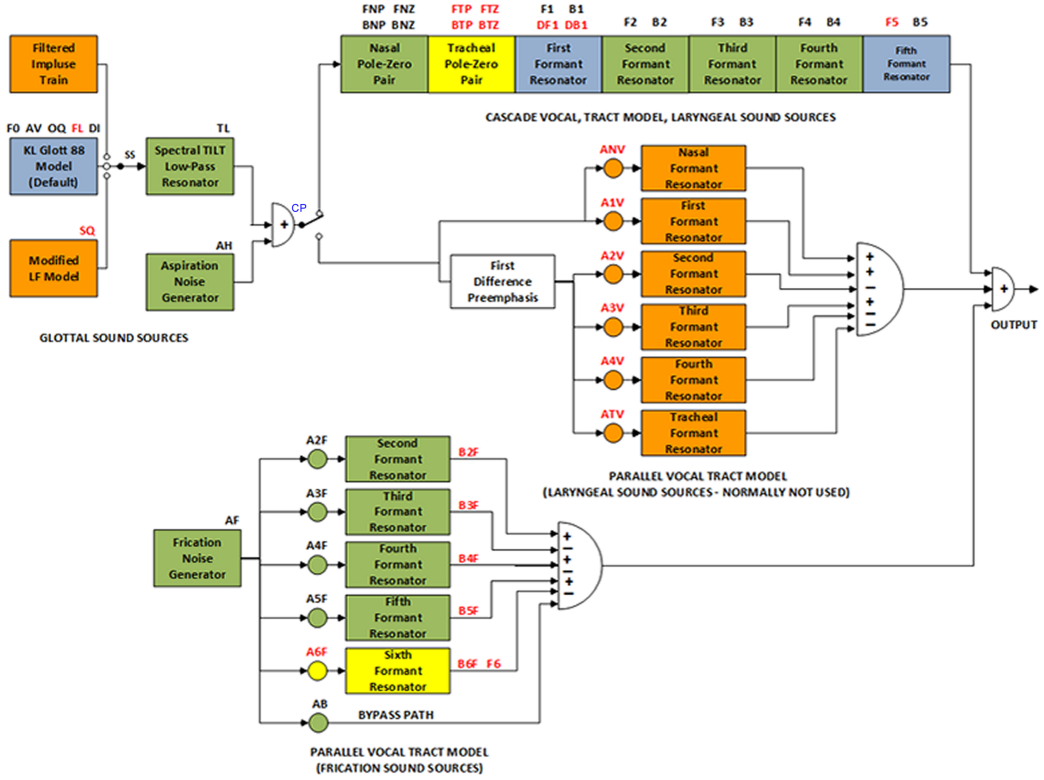

Figura 2.2: Representação do *KLSYN88*.
Fonte: Klatt D. e Klatt L. 1990.

Um resumo individualizado, quanto as faixas de valores de cada parâmetro do sintetizador são apresentados na Tabela 2.1.

KLSYN88 possui 48 parâmetros de entrada, sendo 6 considerados nulos *ANV*, *ATV*, *A1V A2V*, *A3V A4V*, logo tais parâmetros assumem o valor de zero. Os parâmetros que representam a fonte de voz são representados por $F0$ (*fundamental frequency*), *AV* (*amplitude of voicing*), *OQ* (*open quotient*), FL (*flutter*) e, *DI* (*diplophonia*).

Quando $F0$ é diferente de zero, o som é vozeado, *AV* é diferente de zero e *OQ* apre-

senta um valor intermediário entre 40 e 60, indicando que a glote está em posição neutra. TL (inclinação extra do expectro de vozeamento) e AH (amplitude de aspiração) também compõem a fonte de voz e são responsáveis por uma inclinação no espectro da voz e amplitude da aspiração, respectivamente. O ramo em cascata é responsável pela função de transferência que simula sons da laringe e dos cinco ressonadores. Cada formante n apresenta uma frequência F_n e banda B_n. O ramo paralelo contém os parâmetros $A2F$ até o $A6F$ que são amplitudes de controle de excitação da voz e AF que é a fonte fricativa desta.

2.3 Sintetizador *HLSyn*

O sintetizador *HLSyn* [Heid e Hawkins 1998], representado na Figura 2.3, é uma camada de alto nível para o sintetizador *KLSYN88*, sendo que neste ponto sua utilização é diferenciada, devido o *HLSyn* utilizar somente 13 parâmetros para efetuar a síntese de voz, diferente do sintetizador *KLSYN88* que utiliza 48 [Hansona e Stevens 2002].

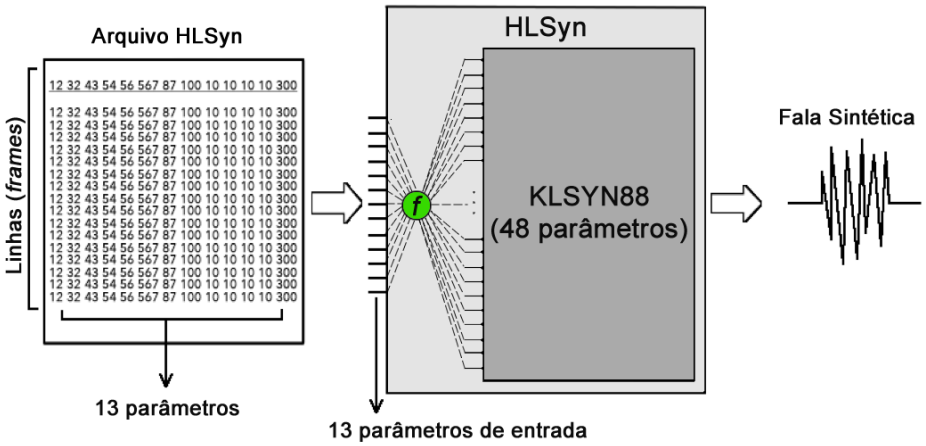

Figura 2.3: Disposição do *HLSyn* e seu núcleo interno sintetizador *KLSYN88*. Destaque para a função f em verde.

A Figura 2.3 (ao centro) ilustra a função f que torna os parâmetros de entrada *HLSyn* equivalentes aos parâmetros do sintetizador *KLSYN88*. Esta função representa a principal tarefa do *HLSyn*, já que é ela que retira as redundâncias existentes dentre os parâmetros do sintetizador *KLSYN88*. Tal função está ligada a diversos tratamentos de sinal e condições lógicas implementadas no interior do sintetizador em questão.

A Tabela 2.2 resume as faixas de valores dos parâmetros *HLSyn*.

Tabela 2.1: *KLSYN88*: Conjunto dos 48 parâmetros.

Parâmetro	Valor mínimo	Valor máximo	Descrição
F0	0	5000	Frequência fundamental.
AV	0	80	Amplitude do vozeamento.
OQ	10	99	Coeficiente de abertura glotal.
SQ	100	500	Velocidade de abertura.
TL	0	41	inclinação extra do espectro de vozeamento.
FL	0	100	Agitação.
DI	0	100	Diplofonia.
AH	0	80	Amplitude de aspiração.
AF	0	80	Amplitude de fricção
F1	180	1300	Frequência da primeira formante.
B1	30	1000	Largura de banda da primeira formante.
DF1	0	100	Mudança no $F1$ durante o abertura da partição de um período.
DBI	0	400	Mudança no $B1$ durante o abertura da partição de um período.
F2	550	3000	Frequência da segunda formante.
B2	40	1000	Largura de banda da segunda formante.
F3	1200	4800	Frequência da terceira formante.
B3	60	1000	Largura de banda da terceira formante.
F4	2400	4990	Frequência da quarta formante.
B4	100	1000	Largura de banda da quarta formante.
F5	3000	4990	Frequência da quinta formante.
B5	100	1500	Largura de banda da quinta formante.
F6	3000	4990	Frequência da sexta formante.
B6	100	4000	Largura de banda da sexta formante.
FNP	180	500	Frequência do pole nasal.
BNP	40	1000	Largura de banda do pole nasal.
FNZ	180	800	Frequência do zero nasal.
BNZ	40	1000	Largura de banda do zero nasal.
FTP	300	3000	Frequência do polo traqueal.
BTP	40	1000	Largura de banda do polo traqueal.
FTZ	300	3000	Frequência do zero traqueal.
BTZ	40	2000	Largura de banda do zero traqueal.
A2F	0	80	Amplitude da excitação paralela de fricção da segunda formante.
A3F	0	80	Amplitude da excitação paralela de fricção da terceira formante.
A4F	0	80	Amplitude da excitação paralela de fricção da quarta formante.
A5F	0	80	Amplitude da excitação paralela de fricção da quinta formante.
A6F	0	80	Amplitude da excitação paralela de fricção da sexta formante.
AB	0	80	Largura de banda da excitação paralela de fricção do caminho *bypass*.
B2F	40	1000	Largura de banda da excitação paralela de fricção da segunda formante.
B3F	60	1000	Largura de banda da excitação paralela de fricção da segunda formante.
B4F	100	1000	Largura de banda da excitação paralela de fricção da segunda formante.
B5F	100	1500	Largura de banda da excitação paralela de fricção da segunda formante.
B6F	0	4000	Largura de banda da excitação paralela de fricção da segunda formante.
ANV	0	80	Amplitude da excitação paralela de vozeamento da formante nasal.
A1V	0	80	Amplitude da excitação paralela de vozeamento da primeira formante.
A2V	0	80	Amplitude da excitação paralela de vozeamento da segunda formante.
A3V	0	80	Amplitude da excitação paralela de vozeamento da terceira formante.
A4V	0	80	Amplitude da excitação paralela de vozeamento da quarta formante.
ATV	0	80	Amplitude da excitação paralela de vozeamento da formante traqueal.

Fonte: [Figueiredo et al. 2006]

2.4 Imitação da fala (*utterance copy*)

O problema de imitação da fala consiste na realização da construção de um sinal de voz sintética a partir de um sinal original, de maneira que o sinal sintético seja o mais próximo possível do sinal de voz alvo, principalmente com relação as características naturais deste sinal. Em importantes pesquisas da área de voz verifica-se que este proce-

Tabela 2.2: *HLSyn*: Conjunto dos 13 parâmetros do sintetizador.

Parâmetro	Mínimo	Máximo	Descrição
AG	0	4000	Área média da abertura da glote em mm^2.
AB	0	1000	Área da seção transversal da construção da língua em mm^2.
AL	0	1000	Área da seção transversal de compressão nos lábios em mm^2.
AN	0	1000	Área da seção transversal da porta *velopharyngeal* em mm^2
UE	0	1000	Taxa de aumento do volume do trato vocal em cm^3/s.
F0	0	5000	Frequência fundamental em Hz.
F1	180	1300	Primeira frequência fundamental em Hz.
F2	550	3000	Segunda frequência fundamental em Hz.
F3	1200	4800	Terceira frequência fundamental em Hz.
F4	2400	4990	Quarta frequência fundamental em Hz.
PS	0	1000	Pressão sub-glotal em $cm\ H_2O$.
DC	-150	150	Mudança na prega vocal em %.
AP	0	1000	Área da abertura posterior da glote em mm^2.

Fonte: [Figueiredo et al. 2006]

dimento é realizado utilizando-se sintetizadores de voz por formantes, como em [Heid e Hawkins 1998].

A síntese paramétrica [Donovan e Woodland 1995] [Anumanchipalli et al. 2010] [Lalwani e Childers 1991] basicamente é o processo imitação da voz realizado através da busca por diferentes combinações dos parâmetros de um sintetizador de voz. Este processo é realizado com base na estimação dos parâmetros de entrada do sintetizador calculados através de análises das propriedades do sinal de voz avaliado.

Ao se realizar síntese paramétrica em um sintetizador por formante, entende-se este como um problema inverso, já que de um lado tem-se os aspectos físicos da voz humana e do outro os parâmetros do sintetizador de voz em questão, somando-se a isso a fato dos sintetizadores por formantes apresentarem memória, ou seja, a síntese do sinal de voz de um frame F_i pode depender da síntese do sinal de voz de um frame F_{i-1} anterior.

Segundo [Engl et al. 1996], problemas relacionados a inversão podem ser entendidos como modelos matemáticos do relacionamento de uma variável X com outra Y, não necessariamente obedecendo a proporcionalidade de um para um, ou seja, podem existir uma quantidade diferente entre o número de variáveis independentes e dependentes do problema. Sendo tais variáveis regidas pelo conceito de 'Causa e Efeito', portanto a partir da variável Y pode-se chegar a variável X e vice-versa.

A solução mais comum para problemas inversos são métodos capazes de realizar o processo de inversão das duas extremidades de um problema [Engl et al. 1996], sendo que tais métodos podem ser generalizados por redes neurais especializadas em problemas de regressão, como é o caso de redes neurais de função base radial, por exemplo.

2.5 Conclusão do capítulo

Este capítulo apresentou uma breve introdução sobre sintetizadores de voz, retratando rapidamente os primeiros estudos relacionados ao assunto e destacando os primeiros pesquisadores da área. De maneira mais conceitual este capítulo também descreveu os principais aspectos referentes ao processo de produção de voz sintética apresentados pelos sintetizadores de voz *KLSYN* e *HLSyn* buscando deixar claro suas principais relações.

Foi destacada a forma que esta pesquisa interpreta o problema de *utterance copy*, sendo este conceito um fator prioritário para o entendimento e validação da metodologia defendida por este trabalho.

Capítulo 3

Técnicas de Inteligência Computacional

Este capítulo contextualiza o trabalho quanto aos assuntos Rede Neural Artificial, Redução da Dimensionalidade e Algoritmo Genético. Desta forma, nos tópicos seguintes são apresentados introduções e as principais características sobre as referidas técnicas uma vez que estas são suportes básicos para a compreensão da metodologia defendida por esta pesquisa.

3.1 Redes Neurais

3.1.1 Introdução

As chamadas Redes Neurais Artificiais (RNAs) são algoritmos inspirados em formas básicas de funcionamento do cérebro humano, ou seja, o sistema nervoso e a representações dos neurônios em si. Tal inspiração permite que este tipo de algoritmo realize processos de aquisição de conhecimento em ambiente computacional controlado baseando-se no agrupamento em forma de rede de neurônios artificiais [Bishop 1995].

Os primeiros a introduzirem o conceito de RNA foram *McCulloch* e *Pitts* em 1943. Em suas pesquisas estes estudiosos desenvolveram um modelo matemático de neurônio artificial que executavam funções lógicas simples. Tais pesquisadores também propuseram a utilização do neurônio em rede a fim de se ter um ganho computacional considerável no processo de obtenção do conhecimento, ou seja, neste momento iniciaram os primeiros estudos com Redes Neurais Artificiais como conhecemos hoje [Faceli et al. 2011].

O neurônio é considerado a unidade básica de funcionamento da RNA, ao se analisar cada unidade básica separadamente pode-se perceber que esta desempenha uma atividade relativamente simples, se comparada com a capacidade de uma rede de neurônios em solucionar problemas específicos. De maneira geral um neurônio artificial é composto

por: entradas, pesos, função de ativação e saída [Faceli et al. 2011]. As mais comuns funções de ativação são: Linear, Limiar e Sigmoidal.

Deve-se destacar que uma RNA pode ser do tipo *Feedforward* e *Feedback* (também chamada de Recorrente). A principal diferença entre estes dois tipos de redes neurais está relacionada ao fluxo da informação, pois nas redes do tipo *Feedforward* o fluxo de informação ocorre a partir da camada de entrada da rede à camada de saída desta, mesmo se tratando de uma rede multicamada. Já nas rede neurais do tipo recorrente este fluxo pode ocorrer tanto no sentido 'entrada-saída' como no sentido 'saída-entrada', devido o algoritmo desta rede utilizar as saídas da própria como entradas da rede a fim de diminuir o erro.

Esta pesquisa utiliza em sua metodologia uma Rede Neural Generalizada de Função Base Radial, que de maneira geral é uma rede do tipo *feedforward* baseada no processo de generalização dos erros que utiliza *Radial Bases Function* como base para seu neurônio. Desta forma a sessão seguinte irá abordar sobre as principais características de tal rede.

3.1.2 Rede neural generalizada de função base radial

Segundo [Haykin 2004] o processo de aprendizado apresentado por uma rede perceptron de múltiplas camadas (sob supervisão) pode ser entendido como uma técnica recursiva conhecida como aproximação estocástica. Já o processo de aprendizado apresentado por uma rede neural com função base radial pode ser entendido como um problema de ajuste de curva (aproximação). Desta forma, para as redes neurais de função base radial, aprender é equivalente a encontrar uma superfície, em um espaço multidimensional, que forneça o melhor ajuste estatístico a um conjunto de treino que represente determinado problema de aproximação.

O projeto de uma rede neural de função base radial (*RBF*, *radial bases function*) se baseia na transformação dos dados utilizados para realizar as aproximações do problema em um espaço de solução com um maior número de dimensões, podendo assim ser mais facilmente linearmente separável, tanto para problema de classificação como para regressão [Bishop 1995].

Uma RBF em sua forma mais básica consiste em três camadas com papeis diferenciados. A primeira camada é a entrada, que basicamente é constituída por nós de fonte (entendidas também como unidades sensoriais) que conectam a rede aos dados do ambiente externo. A segunda camada é a única camada oculta e é responsável por aplicar transformações não-lineares do espaço de entrada para o espaço oculto (espaço multidimensional) criado a partir das funções *RBFs* (chamadas neste caso de funções ocultas)

presentes nesta camada. E por último a camada de saída, que neste caso é linear e fornece a resposta desejada da rede [Haykin 2004].

O processo de generalização de uma *RBF* pode ser entendido como o uso de uma superfície em um espaço multidimensional com o objetivo de interpolar os dados de testes da rede. A estrutura responsável por tal feito é chamada de matriz de interpolação. Com isso, pode-se afirmar que a técnica *RBF* consiste na escolha de uma função F dentre as funções ocultas presentes em sua composição e definida pela Equação 3.1.

$$F(x) = \sum_{i=1}^{N} w_i \phi(||x - x_i||) \tag{3.1}$$

Onde $\phi(||x - x_i||)|i = 1,2,...,N$ é um conjunto de N funções (geralmente não-lineares) arbitrárias, conhecidas como as funções base radial que neste caso são desconhecidas pela rede juntamente com w_i (pesos da rede), o $||.||$ representa uma *norma* que normalmente é euclidiana. Já os pontos $x_i \in \mathbb{R}^{m_0}, i = 1,2,...,N$ são dados conhecidos e representam os centros das funções base radial.

O tipo mais comum de *RBF* é a que implementa a função Gaussiana nos nós ocultos da rede (ϕ), sua fórmula pode ser visualizada na Equação 3.2.

$$\phi(r) = exp\left(\frac{-r^2}{\beta^2} \right) \text{para um } \beta > 0 \text{ e r} \in \mathbb{R} \tag{3.2}$$

Onde $-r^2$ mede a distância para o centro da função base radial e β representando a suavidade da interpolação desta função. Existem outros tipo de *RBF*s como: Multiquadrática, Quadrática Inversa, Multiquadrática inversa, *Thin-plate Spline*, dentre outras.

As redes neurais generalizadas de função base radial, unem as características das redes neurais generalizadas com as características das *RBF*s. O resultado desta união é a *Generalized Regressive Neural Network (GRNN)* que consistem em uma rede neural com alta capacidade à generalização de problemas de regressão, apresentando baixo custo computacional [Chen et al. 1991].

As chamadas redes neurais generalizadas são redes neurais voltadas para a realização do processo de aprendizado através da generalização dos erros, que basicamente consiste em mensurar o quão bom é o aprendizado da rede em relação ao conjunto de dados treino, desta forma é realizado o calculo da distância (erro) para tal conjunto de dados [Bishop 1995].

A arquitetura da *GRNN* é similar ao de uma rede neural de função base radial comum, sendo assim esta rede apresenta, além das camadas de entrada (com E_n receptores) e saída (com S_n funções lineares), uma camada intermediária (oculta) contendo um número de

neurônios igual a quantidade de instâncias do conjunto de treino utilizado no aprendizado. Ou seja, a camada oculta busca generalizar cada uma das instâncias presentes no conjunto de dados treino da rede de maneira particular (uma-a-uma), sendo assim, pode-se afirmar que a quantidade de espaços multidimensionais criados pela *GRNN* equivale ao número de instâncias presentes no conjunto de treino da rede [Chen et al. 1991]. Vale ressaltar também que esta rede faz a utilização do bias (viés indutivo) em sua camada de saída.

Uma visão geral da arquitetura da *GRNN* pode ser visualizada na Figura 3.1.

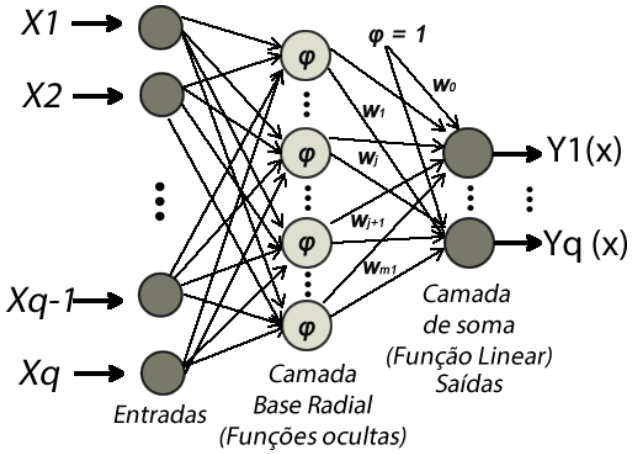

Figura 3.1: Representação da *GRNN*.

Uma vez que a *GRNN* sofre com a adição do bias, se comparado com a *RBF*, a fórmula que representa a função oculta deste novo tipo de rede é representada pela Equação 3.3, onde b representa a adição do valor do bias, já os demais termos recebem a mesma interpretação da Equação 3.1.

$$F(x) = \sum_{i=1}^{N} w_i \phi(||x - x_i||) + b \tag{3.3}$$

De acordo com [Haykin 2004], as principais diferenças de arquitetura entre redes *RBF* e *GRNN* estão concentradas em dois aspectos: quantidade de neurônios presentes na camada oculta e parâmetros aprendidos pela rede. Para a *RBF* o número de neurônios presente na camada oculta é exatamente o número de instâncias presentes no conjunto de dados treino da rede T_n, enquanto que para a *GRNN* o número de neurônios pode ser menor do que T_n.

Com relação a parâmetros aprendidos pela rede, para a *RBF* as funções de ativação da camada oculta são conhecidas, sendo os pesos lineares da camada de saída elementos não conhecidos. Já para a *GRNN* os pesos lineares associados com a camada de saída e as posições das funções de base radial juntamente com a matriz de ponderação de norma

associada com a camada oculta são todos parâmetros desconhecidos que devem ser aprendidos pela rede.

3.2 Redução da dimensionalidade dos dados

3.2.1 Introdução

Um conjunto de dados coletados na natureza pode ser entendido como sendo multivariado, ou seja, tais dados fazem referência a observações de um número p de características de um grupo n de indivíduos. Desta forma, como o conjunto de características são extraídas tomando-se por base cada indivíduo de maneira singular, é esperado que tais características apresentem correlações entre si. Partido desta premissa, pode-se aferir que assim como se tem atributos correlacionados, pode-se ter também atributos que não tenham nenhuma associação com os demais [Silvestre 2007].

Um problema relativamente comum na área de reconhecimento estatístico de padrões é a seleção de atributos. Tal processo de seleção é projetado de forma que um conjunto de dados pode ser representado por um conjunto reduzido de atributos. Em outras palavras, o conjunto de dados sofre uma redução em sua dimensionalidade [Haykin 2004].

O objetivo das técnicas de redução da dimensionalidade é realizar a diminuição do número de atributos de um conjunto de dados rotulados sem perder informações significativas sobre o conhecimento que os mesmos podem vir a representar [Silvestre 2007]. Dentre as técnicas mais populares referentes a este assunto estão: Análises de Componentes Principais, Algoritmos de Agrupamentos e Algoritmos de Seleção de Atributos.

Dentre os apresentados acima, este trabalho dá ênfase à técnica chamada Seleção de Atributos, ou em inglês *Feature Selection*, que consiste na análise dentre um grupo de atributos X e Y (dados rotulados), onde o objetivo principal é mensurar a importância de cada atributo de X no processo de predição dos atributos de Y, ou seja, mensurar quais os melhores *features* de X que possibilitem a interpolação destes dados a partir de uma rede neural (por exemplo) a fim de se produzir o conjunto de respostas Y.

Os algoritmos de seleção de atributos podem ser categorizados como sendo do tipo: *wrappers*, *filters* e *embedded methods* [Klautau 2002]. O foco deste trabalho é o algoritmo *RReliefF* [Robnik-Sikonja e Kononenko 2003], que é do tipo *filter*, por isso na sessão seguinte serão trabalhados os principais aspectos sobre o referido algoritmo.

3.2.2 Algoritmo *RReliefF*

O algoritmo de seleção de atributos chamado *RReliefF* [Robnik-Sikonja e Kononenko 2003] é uma extensão do algoritmo *ReliefF* adaptado por [Kononenko et al. 1997] a partir da pesquisa desenvolvida por [Kira e Rendell 1992]. Em sua primeira versão, chamada de *Relief*, este algoritmo é capaz de detectar dependências condicionais entre atributos e prover uma visão unificada destas *features* em problemas de classificação, sendo limitado apenas a duas classes. Em sua extensão, chamada *ReliefF*, a principal diferença é que este algoritmo se torna capaz de ser aplicado à problemas de classificação para n classes e em alguns casos até mesmo regressão, porém neste último com baixo desempenho.

Com a crescente necessidade de algoritmos capazes de realizar o processo de redução da dimensionalidade em problemas de regressão, surgiu a mais nova versão do algoritmo da família *Relief*, neste caso chamado de *RReliefF* ou mesmo *Regressional ReliefF* [Robnik-Sikonja e Kononenko 2003]. Este algoritmo se baseia em conceitos de probabilidade Bayesiana para calcular o peso de cada atributo em um conjunto de dados.

O conceito mais geral que define o processo apresentado pelo algoritmo *RReliefF* é a definição de que este algoritmo estima a qualidade dos atributos de acordo com o quão cada atributo se distingue entre instâncias que estão mais próximas uma das outras, levando-se em conta o cálculo de distância entre os atributos baseado-se na Equação 3.4.

Para um melhor entendimento sobre as particularidades e fluxo de execução do algoritmo *RReliefF* deste ponto em diante irá se adotar a seguinte legenda disponível na Tabela 3.1.

Em problemas de regressão geralmente tem-se a seguinte situação: um conjunto de dados A composto de um conjunto de dados X e outro τ (dados rotulados), onde cada ocorrência de X refere-se a uma ocorrência de τ e este último é do tipo contínuo. Como os valores de τ são contínuos, não pode-se entender este como sendo divididos em classes, uma vez que os valores em τ são muito próximos uns dois outros e neste caso multivalorados. Nesta situação o algoritmo *RReliefF* atua calculando os pesos (importâncias) $W[A]$ de cada atributo de A através da distância relativa entre outra instância mais próxima.

A seguir é apresentada uma descrição geral dos processos realizados pelo algoritmo *RReliefF*, para facilitar a compreensão são apresentadas também as principais funções (Equações 3.4 até 3.8) e o pseudo-código (Figura 3.2) juntamente com a Tabela 3.1 que descreve a legenda da codificação utilizada na descrição.

Como citado anteriormente, o *RReliefF* calcula a qualidade de um atributo de acordo com o quão cada atributo se distingue dos demais. Tal processo é realizado com base na distância relativa entre os atributos calculada a partir da Equação 3.4.

Tabela 3.1: Legenda de variáveis.

Variável	Descrição
$I_1, I_2, ...I_n$	Instâncias de um conjunto de dados do problema.
X	Conjunto de dados pertencentes a A sem seus referidos rótulos.
τ	Conjunto de dados pertencentes a A referente somente aos rótulos de X.
A	Conjunto de instâncias de dados que possuem os atributos do problema (dados rotulados X e τ).
a	Quantidade total de atributos.
m	Quantidade de vezes que o algoritmo é executado.
l	Em um *rank* de instâncias ordenadas a partir da distância relativa em relação a instância I_0, l (L) é a segunda distância mais próxima de I_0.
K	Quantidade de vizinhos a serem comparados.
$P(x\|y)$	Probabilidade de x ocorrer dado que y ocorre.
Dva	Uma instância em A com diferentes atributos.
Imp	Instância mais próxima.
$ImpDp$	Instância mais próxima com diferentes predição.

$$diff(A, I_1, I_2) = \frac{|valor(A, I_1) - valor(A, I_2)|}{max(A) - min(A)} \tag{3.4}$$

Onde, $diff(A, I_1, I_2)$ será a distância relativa entre uma instância I_1 e outra I_2, ambas pertencentes ao conjunto A. Deve-se levar em conta que nesta equação a função "valor"está recebendo o verto Ii que é correspondente ao vetor de atributos da instância. Esta função é a base do *RReliefF*, já que os demais passos realizados pelo algoritmo são baseados no *rank* da distância relativa calculada por instâncias selecionadas de maneira aleatória.

O fundamento básico do algoritmo em questão é a sua base fundamentada em regras probabilísticas capazes de mensurar a importância de cada atributo. Desta forma as Equações 3.5, 3.6 e 3.7 ilustram regras que calculam a probabilidade de duas instâncias do conjunto A apresentarem diferentes valores de predição (ou seja, τ), sendo estas duas instâncias o mais próximas possíveis uma da outra (no sentido de distância).

$$P_{diffA} = P(Dva|Imp) \tag{3.5}$$

$$P_{diffC} = P(ImpDp|Imp) \tag{3.6}$$

Ou seja, em P_{diffA} é calculada a partir da probabilidade de uma instância Dva ocorrer

no conjunto de dados, dado que a instância mais próxima a ela (*Imp*) ocorreu (Equação 3.5). Já na Equação 3.6 P_{diffC} é a probabilidade de uma instância mais próxima com diferente valor de predição (τ) ocorrer dado que *Imp* (a mesma *Imp* mais próxima de *Dva*) ocorreu. Logo a Equação 3.7 resume as duas equações em uma só.

$$P_{diffC|diffA} = P(ImpDp|Dva \cup Imp) \tag{3.7}$$

A importância (também entendida como peso) de cada atributo $W[A]$ é calculada com base na diferença das regras Bayesianas dispostas na Equação 3.8.

$$W[A] = \frac{P_{diffC|diffA} P_{diffA}}{P_{diffC}} - \frac{(1 - P_{diffC|diffA}) P_{diffA}}{1 - P_{diffC}} \tag{3.8}$$

Algorithm RReliefF
Input: for each training instance a vector of attribute values **x** and predicted value τ(**x**)
Output: vector W of estimations of the qualities of attributes

1. set all N_{dC}, $N_{dA}[A]$, $N_{dC\&dA}[A]$, $W[A]$ to 0;
2. **for** i := 1 **to** m **do begin**
3. randomly select instance R_i;
4. select **k** instances I_j nearest to R_i;
5. **for** j := 1 **to** k **do begin**
6. $N_{dC} := N_{dC} + \text{diff}(\tau(\cdot), R_i, I_j) \cdot d(i, j)$;
7. **for** A := 1 **to** a **do begin**
8. $N_{dA}[A] := N_{dA}[A] + \text{diff}(A, R_i, I_j) \cdot d(i, j)$;
9. $N_{dC\&dA}[A] := N_{dC\&dA}[A] + \text{diff}(\tau(\cdot), R_i, I_j) \cdot$
10. $\text{diff}(A, R_i, I_j) \cdot d(i, j)$;
11. **end**;
12. **end**;
13. **end**;
14. **for** A := 1 **to** a **do**
15. $W[A] := N_{dC\&dA}[A]/N_{dC} - (N_{dA}[A] - N_{dC\&dA}[A])/(m - N_{dC})$;

Figura 3.2: Pseudo-código do algoritmo *RReliefF*.
Fonte: Robnik-Sikonja e Kononenko 2003.

A Figura 3.2 apresenta o pseudo-código do algoritmo RReliefF, onde são mostrados os passos de execução de processos realizados por este algoritmo. Na linha 2 observa-se uma interação que vai de 1 à m sendo que esta variável é normalmente o valor de k^2 e desta forma, como k é normalmente 10 (valor utilizado para avaliar 10 vizinhos) o valor de m é 100 o que indica que o algoritmo irá criar 100 *ranks* de 10 indivíduos selecionados de maneira aleatória.

Na linha 3 faz-se a seleção de uma instância R_i de maneira aleatória e na sequência (linha 4) são selecionadas k instâncias I_j com menores distâncias para R_i (cálculo de distância com base na Equação 3.4).

Ainda com relação ao pseudo-código destaca-se ponderações para diferentes valores de predição $\tau(.)$ (linha 6), diferentes atributos (linha 8) e diferentes predições de τ e diferentes atributos (linhas 9 e 10), coletados em N_{dc}, $N_{dA}[A]$, e $N_{dC\&dA}[A]$ respectivamente. Ao final (linha 14 e 15) a Equação 3.8 é calculada com base em cada atributo.

Contudo, vale destacar que uma das principais características desta técnica é selecionar atributos diferenciados uns dos outros, já que atributos similares de X (com valores aproximados tendo ou não o mesmo valor de predição τ) são avaliados com a ideia de *rank* e portanto são penalizados no processo como um todo.

3.3 Algoritmo Genético

3.3.1 Introdução

Na área de sistemas inteligentes os chamados Algoritmos Genéticos são bastantes populares, já que este tipo de algoritmo são comumente utilizados em problemas relacionados a otimização de soluções, é fácil encontrar pesquisas que utilizam desta técnica como solução dos mais variados tipos de problemas encontrados no mundo real. Este algoritmo é inspirado na natureza, desta forma, dentro da ciência da computação, faz parte da área de Computação Bio-inspirada (CB).

Os principais fundamentos em que o algoritmo genético se baseia foram inspirados na teoria da evolução, defendida por *Charles Darwin*, ou seja, são realizados processos similares aos observados na vida animal, principalmente referentes a evolução dos seres vivos. Tais processos referem-se a forma como são realizadas as trocas de material genético entre indivíduos, ou seja, o cruzamento, seleção, mutação e avaliação.

Os algoritmos genéticos, também chamados de algoritmos evolucionários, surgiram por volta da década de 60 através do pesquisador *John Holland*, apesar de antes desta época outros pesquisadores já terem publicado pesquisas do que pode ser entendido como os primórdios do algoritmo genético. Foi *John Holland* quem propôs metodologias matemáticas para utilização de mecanismos de adaptação implementados em sistemas computacionais [Holland 1975].

Assim como na evolução natural das espécies (em ambiente natural), a população de um algoritmo genético tem um comportamento onde o objetivo geral é selecionar indivíduos que no caso da AG são soluções do problema proposto e geralmente são representados por funções matemáticas. Fatores como quantidade de membros, variabilidade genética e operadores genéticos são fatores que afetam o desempenho do algoritmo genético.

Os indivíduos pertencentes a um AG possuem genes que compõem um alfabeto ou *string*, a codificação do gene do indivíduo depende do problema a ser resolvido. Os mais comuns tipos de codificação de indivíduos é a binária e a real. Na binária os genes são representados por um vetor de bits, já na real os genes são normalmente representados por um vetor de números reais.

O AG realiza a busca por melhores indivíduos através do processo de evolução, tal busca se inicia através da competição dentre os indivíduos de uma população a fim de se encontrar os indivíduos mais aptos à solução do problema proposto. Através do procedimento de cruzamento dois indivíduos progenitores (pais) realizam a produção de dois genitores (filhos). Os genitores são formados pela combinação dos materiais genéticos dos progenitores, sendo que a maneira que esta combinação ocorre depende da técnica de cruzamento utilizada pelo algoritmo, como: único ponto de corte, múltiplos pontos de corte, aritmético e uniforme por exemplo [de Paula Bueno 2010].

Um outro elemento gerador de variabilidade genética no algoritmo genético é a mutação, que basicamente é o processo de modificação do gene de uma parcela mínima da população de maneira "especial" a fim de se modificar a estrutura genética do indivíduo, deixando assim a população com maior variedade. Vale ressaltar que a mutação é responsável por variabilidades mínimas na população que irão lhe permitir a superação dos chamados mínimos locais [Deb 2001].

De acordo com [Oliveira et al. 2011] um algoritmo genético pode ser aplicado a problemas que apresentem três principais características:

- Possuem parâmetros que se combinados possibilitam diferentes soluções ao problema;
- Apresentam restrições ou condições que não podem ser modeladas matematicamente;
- Possuem elevada dimensão (no sentido de tamanho) do espaço de busca.

A mais recente abordagem que representa as evoluções da técnica de Algoritmo Genético é chamada Algoritmo Genético Multi-objetivo (AGMO), onde são utilizadas mais de uma função para avaliar um indivíduo. Deve-se destacar que a arquitetura desta nova técnica introduz conceitos diferenciados e estruturas mais complexas se comparado com a arquitetura do algoritmo genético mono-objetivo, porém vale ressaltar que muitos dos conceitos inseridos em técnicas multi-objetivo podem ser utilizados em técnicas mono-objetivo como é o foco deste trabalho. O *Non-dominated sorting genetic algorithm II* é um algoritmo genético utilizado no trabalho aqui descrito e portanto será descrito no tópico a seguir.

3.3.2 *Non-dominated sorting genetic algorithm II (NSGA-II)*

O *Non-Dominated Sorting Genetic Algorithm (NSGA-II)* tem como estratégia encontrar soluções espalhadas sobre o espaço de busca a ser explorado utilizando baixo custo computacional para isso. Dentre os estudos realizados na área de Computação Genética, destaca-se o trabalho [Srinivas e Deb 1994] como sendo a primeira versão do *NSGA-II*, sendo que a referida pesquisa representa a versão mais básica do algoritmo, porém com o passar do tempo outros pesquisadores apresentaram melhorias consideráveis a arquitetura do NSGA-II.

De acordo com [Deb 2001], pode-se destacar como sendo as principais características do *NSGA-II* os ítens abaixo:

- **Elitismo:** mecanismo que preserva os melhores indivíduos encontrados em gerações posteriores;
- **Atribuição de *rank*:** de maneira geral, é a técnica de ordenação dos indivíduos da população de forma não-dominante;
- **Distância de Multidão:** técnica que utiliza operadores de seleção por torneio para preservar a diversidade entre as soluções não dominadas nos estágios de execução posteriores, almejando o melhor espalhamento dentre as soluções.

Um melhor entendimento sobre os três principais aspectos listados acima poderão ser melhor assimilados através da observação do processo de formação de novas populações realizado pelo *NSGA-II*, Figura 3.3.

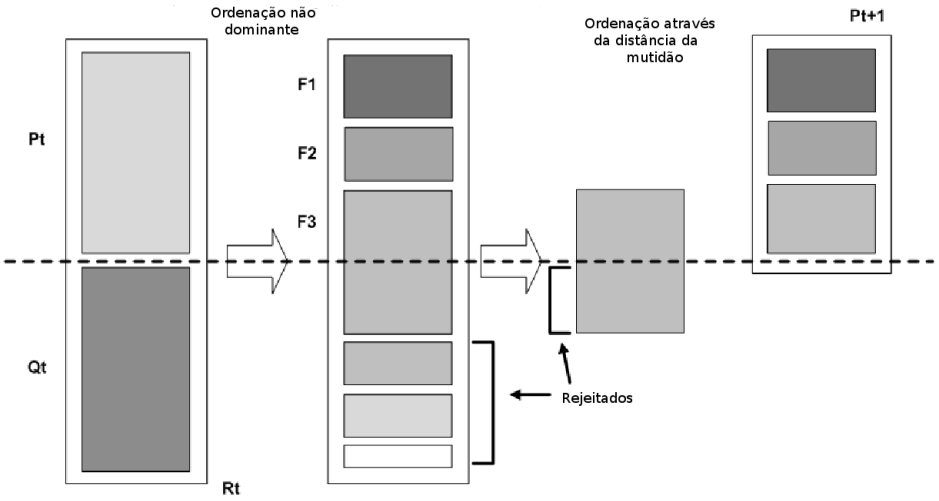

Figura 3.3: Formação de nova população no *NSGA-II*.
Fonte: Deb 2011.

De acordo com a Figura 3.3 e com [Trindade et al. 2013] deve-se entender que o processo de geração de uma nova população no *NSGA-II* ocorre da seguinte maneira: dada uma população Q_t, essa é criada a partir de uma população P_t, sendo ambas populações do tamanho N (número de indivíduos). A população R_t é criada a partir da combinação de ambas as populações, sendo assim esta população possui $2N$ (dobro de indivíduos).

A ordenação dos indivíduos é realizada de maneira a destacar os melhores indivíduos de R_t, portanto tal ordenação é realizada de maneira decrescente. É a criação deste *rank* que permite chegar-se a uma não-dominância global entre as populações P_t e Q_t.

No final do processo de ordenação das soluções não dominadas um novo conjunto P_t é formado por soluções de diferentes frente não dominadas (neste caso F_1, F_2, ... F_n). O preenchimento de P_t é realizado com base na seleção dos melhores indivíduos da primeira frente não dominada, seguido das frentes subsequentes formadas. Tendo-se o cuidado para não ultrapassar as N soluções, sendo assim, ao se obter as N soluções as demais são descartadas.

Cada F_i é inserida completamente na nova população quanto $|P_{t+1} + |F_i| > N$. Com isso, através da distância de multidão as soluções mais espalhadas do conjunto de F_i são preferidas, com o objetivo de garantir a diversidade genética da população [Deb 2001].

Ainda segundo [Trindade et al. 2013] os passos de cálculo da distância da multidão seguem o seguinte algoritmo:

1. O número de solução em F é chamado de $l = |F|$. Para cada solução i no conjunto F atribui-se $d_t = 0$.
2. Para cada objetivo $m = 1, 2, ..., M$, ordena de forma decrescente as soluções por f_m na lista l_m.
3. Para cada solução extrema faça: $d_{I_1^m} = d_{I_l^m} = \infty$ para as outras faça a Equação 3.9:

$$d_{I_i^m} = d_{I_i^m} + \frac{f_m^{(I_{i+1}^m)} - f_m^{(I_{i-1}^m)}}{f_m^{max} - f_m^{min}} \tag{3.9}$$

Na Equação 3.9, I_i^m representa a $i-esima$ solução na lista ordenada pelo objetivo m. I_1^m e I_l^m são as soluções extremas de máximo e mínimo objetivo m. $f_m^{(I_{i+1}^m)}$ e $f_m^{(I_{i-1}^m)}$ são os valores dos objetivos das soluções vizinhas à i. f_m^{max} e f_m^{min} são os parâmetros dos limites máximo e mínimo em cada objetivo. A Equação 3.9 assegura que as soluções mais dispersas tenham distâncias de multidão maiores.

A Figura 3.4 representa os passos de definição da distância da multidão definida por [Deb 2001].

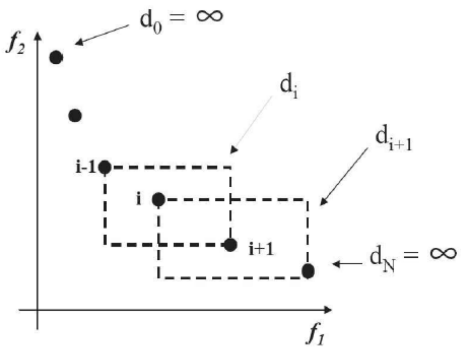

Figura 3.4: Procedimentos para o cálculo da distância de multidão.
Fonte: Deb 2001.

O NSGA-II utiliza torneio como método de seleção. Uma solução é considerada ganhadora do tornei quando:

- Uma solução possui melhor *rank* de não-dominância na população.
- Quando uma solução possui o mesmo *rank*, a distância da multidão é considerada como critério de desempate. A que tiver maior distância é a vencedora

3.4 Conclusão do capítulo

Este capítulo apresentou os principais aspectos dos três principais assuntos que envolvem a metodologia da pesquisa, que são: Rede Neural Generalizada de Função Base Radial, Algoritmo de Redução da Dimensionalidade *RReliefF* e Algoritmo Genético *NSGA-II*. Sendo tais conceitos de fundamental importância para o entendimento e validação dos processos metodológicos no qual esta pesquisa se baseia. Contudo, além das descrições feitas nos tópicos anteriores, vale ressaltar as referências citadas por este trabalho com relação a cada técnica utiliza.

Capítulo 4

Metodologia dos frameworks *NSM* e *HLSpeechGA*

Neste capítulo são descritos os aspectos que envolvem a técnica desenvolvida para a solução do problema de extração de parâmetros de sintetizadores por formantes. Durante as pesquisas, foram desenvolvidos dois frameworks, um chamado *Neural Speech Mimic (NSM)* e outro chamado *High Level Speech processed by Genetic Algorithm (HLSpeechGA)*.

4.1 Descrição do framework *NSM*

Atuando como *front-end*, o framework *Neural Speech Mimic* foi desenvolvido com o objetivo de melhorar os processos realizados pelo framework *HLSpeechGA*, já que os resultados obtidos por este primeiro podem ser reutilizados pelo framework *HLSpeechGA*.

O *NSM* tem como principal função realizar o processo de estimação dos parâmetros do sintetizador *HLSyn* baseando-se em procedimentos fundamentados em Processamento Digital de Sinal e Rede Neural. Este framework foi implementado em ambiente de desenvolvimento *MATLAB* [Matlab n.d.] e utiliza bibliotecas nativas de Redes Neurais Artificiais deste ambiente. Deve-se destacar que este software apresenta interface de comunicação com as ferramentas *Praat* [Boersma 2001], *HTK* [Young 2000] e *Straight* [Liu e Kewley-Port 2004].

O princípio de funcionamento do framework *NSM* consiste em processos de extração de coeficientes de um sinal de voz alvo, como *MFCC, PLP, MELSPEC* e *FBANK*, sendo capaz de realizar a seleção de tais coeficientes e os repassar como entradas para uma rede neural, que uma vez já treinada, realiza o processo de regressão para os parâmetros

do sintetizador de voz em questão, ou seja, as saídas da RNA presentes no software são representações dos parâmetros do sintetizador *HLSyn*. Logo o *NSM* pode ser entendido como um processador ou mesmo pré-processador de parâmetros *HLSyn.*

Nos tópicos seguintes são descritas características do framework relacionadas ao: tipo de rede neural utilizada, coeficientes de voz que são utilizadas como entradas da RNA, pré-processamentos realizados, treinamento da rede e visão geral da arquitetura do framework.

4.1.1 Rede neural de função base radial

Durante o processo de construção da ferramenta *NSM* foram estabelecidos critérios que permitiram selecionar o tipo de rede neural mais adequada ao problema proposto. Desta forma, pode-se resumir tais critérios em duas principais vertentes, que são:

- **Baixo custo computacional:** principalmente com relação aos processos de treinamento e teste, pois como já foi citado, a ideia deste framework é atuar como mecanismo de suporte ao framework *HLSpeechGA*;
- **Adaptabilidade a problemas de regressão:** uma vez que se almeja extrair *n* parâmetros de um sintetizador de voz por formantes a partir de *m* coeficientes extraídos de um sinal de voz alvo, a rede neural atua realizando a regressão (ou mesmo inversão) entre os *m* coeficientes (entradas da RNA) e os *n* parâmetros do *HLSyn* (saídas da RNA).

Contanto, no interior do *NSM* foi implementada uma Rede Neural Generalizada de Função Base Radial [Chen et al. 1991], já que esta rede apresentou as características desejadas à solução do problema aqui apresentado e também se mostrou eficiente dentro do esperado em testes realizados.

Dentre as definições necessárias para a utilização da rede neural de função base radial, pode-se destacar o *Spread*, que mensura a abrangência da soma das funções radiais que generalizam a rede neural do problema. Desta forma, o *NSM* utiliza um *Spread* igual a 1, que representa o ponto de equilíbrio entre uma rede muito ou pouco sensível à variações de entrada.

Na Figura 4.1, observa-se que a extração de 60 *frames* do parâmetro *AG* do sintetizador *HLSyn* foi alcançada com um erro de aproximadamente $0,04$, o que é relativamente baixo para este problema. Deve-se levar em conta que para a extração deste parâmetro foram necessários coeficientes da voz alvo (neste caso, *MFCC, PLP, FBANK e MELSPEC*). As definições destes coeficientes são descritas no tópico seguinte.

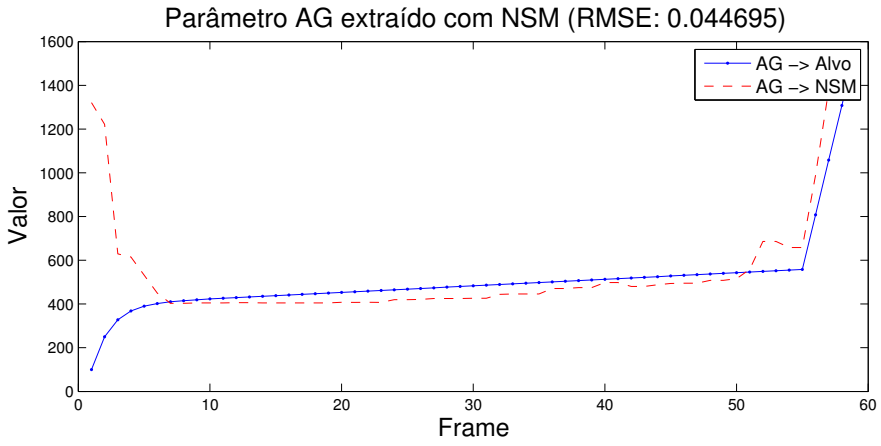

Figura 4.1: Comparativo entre valores do parâmetro alvo e valores do parâmetro extraído pelo *NSM*.

4.1.2 Coeficientes de voz e pré-processamento

Como citado anteriormente, o framework desenvolvido apresenta interfaces de comunicação com os softwares *Praat* [Boersma 2001], *HTK* [Young 2000] e *Straight* [Liu e Kewley-Port 2004].

Utilização do *Praat* e *Straight*

O *Praat* [Boersma 2001] é utilizado para extrair as formantes *F1-F4*. Já o *Straight* [Liu e Kewley-Port 2004] é utilizado para extrair o *F0*. Este último, é repassado como uma das entradas da RNA, juntamente com sua primeira e segunda derivada, o que proporciona maior compatibilidade entre os parâmetros gerados pela RNA, Figura 4.9 (c) (d).

Utilização do *HTK*

Em processos internos ao *NSM* o *HCopy* [Young 2000], ferramenta presente no pacote *HTK*, é responsável por realizar o processo de extração de coeficientes do sinal de voz alvo, Figura 4.9 (a) (b).

Durante esta pesquisa, realizou-se experimentos com coeficientes que podem ser extraídos de um sinal de voz utilizando-se o *HCopy* e *Straight*, como: *MFCC*, *PLP*, *FBANK*, frequências de aperiodicidades, *LPC*, *LPCEPSTRA* e *MELSPEC* a fim de encontrar os melhores coeficientes a serem repassados à RNA obtendo-se assim os menores erros das estimações das saídas da rede.

Tais experimentos consistem em utilizar os sinais de voz sintética das palavras *'air', 'end', 'no' e 'gill'*, produzidos pelo software DECTalk, buscando-se obter a estimações

dos parâmetros *AG, AL, AB, AN, UE, PS, DC* e *AP* do sintetizador *HLSyn* utilizando diferentes quantidades e tipos de coeficientes dos sinais e os processando por meio da RNA presente no *NSM*. Deve-se destacar que o erro foi calculado com base na diferença entre os parâmetros alvo e os parâmetros extraídos pela rede neural, através de *Root Mean Squared Error (RMSE)*. Deve-se deixar claro que para tal experimento foram escolhidos quatro sinais de voz ao acaso.

Os melhores resultados referentes aos experimentos com coeficientes são mostrados nas Figuras 4.2 e 4.3.

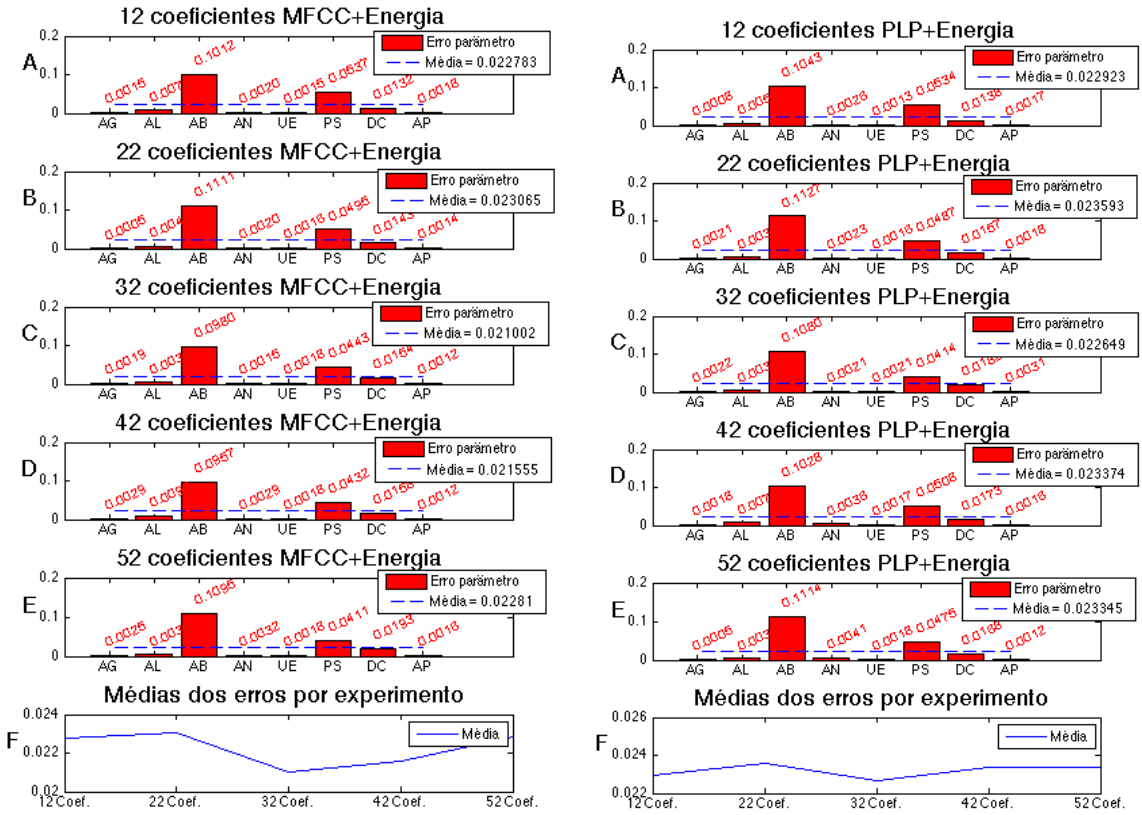

Figura 4.2: Coeficientes *MFCC* (esquerda) e *PLP* (direita), média dos erros dos parâmetros dos 4 sinais.

Na Figura 4.2 consegue-se perceber que os erros existentes nas saída da RNA são relativamente baixos e variam de acordo com o incremento da quantidade de coeficientes de *MFCC* a esquerda e *PLP* a direita, principalmente ao se utilizar 32 coeficientes já que analisando-se as médias dos erros dos parâmetros (linha tracejada em azul) em ambos experimentos, consegue-se os menores erros. Portanto considera-se estes dois tipos de coeficientes importantes para a entrada da RNA.

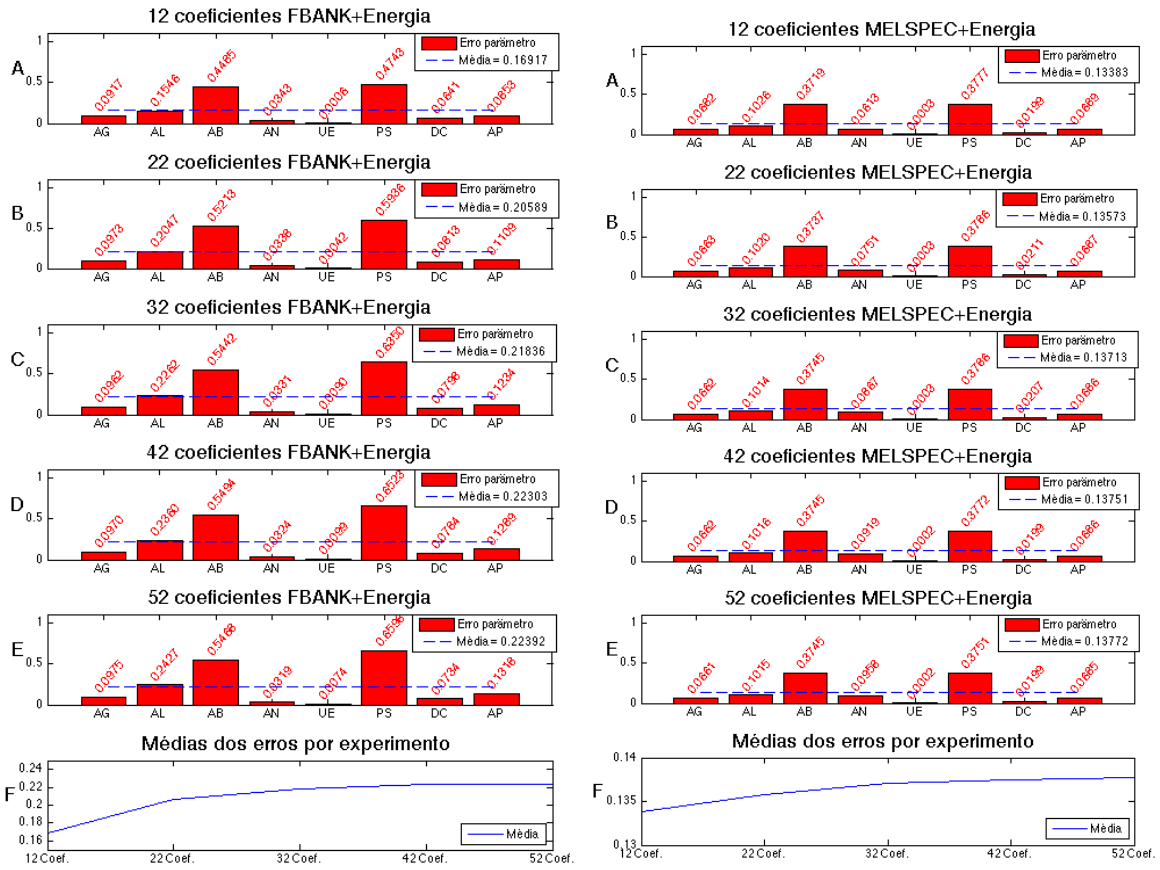

Figura 4.3: Coeficientes *FBANK* (esquerda) e *MELSPEC* (direita), média dos erros dos parâmetros dos 4 sinais.

Ao se utilizar coeficientes do tipo *FBANK* ou mesmo *MELSPEC* (Figura 4.3) nas entradas da RNA, pode-se perceber erros significativamente menores nos parâmetros *HLSyn* estimados (com exceção de *AB* e *PS*), se comparados com os erros encontrados utilizando-se *MFCC* e *PLP* (Figura 4.2). Também pode-se observar na Figura 4.3.F que os menores erros foram encontrados utilizando-se 12 coeficientes em ambos experimentos, portanto considera-se estes dois tipos de coeficientes importantes para a entrada da RNA na respectiva quantidade de coeficientes.

Já nos experimentos com frequências de aperiodicidades, *LPC* e *LPCEPSTRA* os erros das saídas da RNA se comportaram de maneira constante mesmo com a variação do número de coeficientes extraídos e também permaneceram relativamente altos, se comparados com os erros dos experimentos anteriores. Para tais experimentos teve-se respectivamente as seguintes médias de erros constantes: 0.09178, 0.3020 e 0.3020.

Portanto os coeficientes frequências de aperiodicidades, *LPC* e *LPCEPSTRA* são entendidos como não importantes para as entradas da RNA do sistema proposto e desta

forma não são utilizados na pesquisa.

Pré-processamento

Uma vez que o sistema *NSM* utiliza coeficientes *MFCC, PLP, FBANK* e *MELSPEC* juntamente com suas energias, dentro das proporções 32, 32, 12 e 12, pode-se entender que a redução da dimensionalidade dos dados utilizados é um fato necessário, já que dentre os conjuntos de coeficientes utilizados há a possibilidade de se ter unidades de coeficientes que não ofereçam um ganho de informação importante ao problema à RNA do framework *NSM*.

O algoritmo utilizado para a redução da dimensionalidade dos dados foi o *Feature Selection RReliefF* [Robnik-Sikonja e Kononenko 2003], já que este apresenta alta adaptabilidade à problemas de regressão como é o caso do trabalho por este trabalho aqui descrito.

Feature Selection RReliefF

O *NSM* aplica o algoritmo de *Feature Selection RReliefF* [Robnik-Sikonja e Kononenko 2003] para cada parâmetro do *HLSyn* a ser estimado pela rede neural e ao final realiza a união dos grupos de coeficientes selecionados. Desta forma o framework consegue equilibrar a quantidade de coeficiente igual dentre cada parâmetro a ser estimado.

Entendendo a totalidade dos coeficientes utilizados pelo *NSM* como sendo o conjunto total dos dados à sofrerem redução da dimensionalidade, o algoritmo de seleção de atributos *RReliefF* é aplicado como no esquema da Figura 4.4.

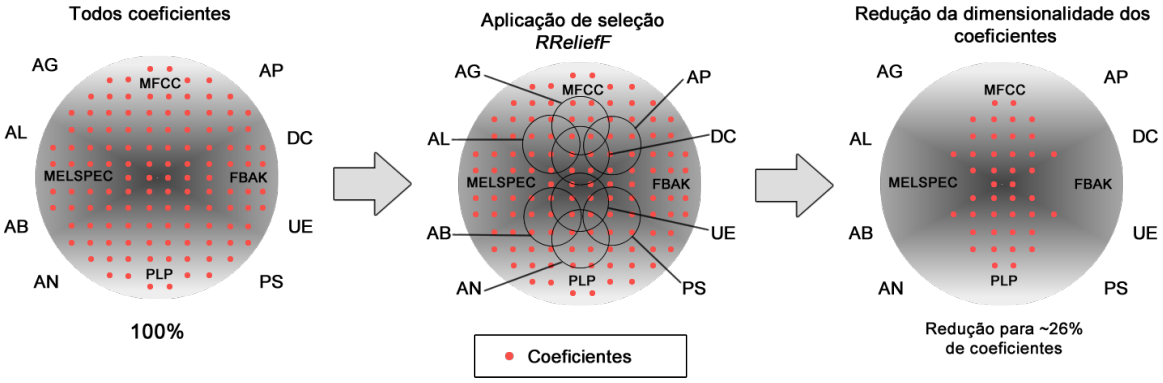

Figura 4.4: Esquema do processo de diminuição da dimensionalidade usando *RReliefF* vs parâmetros *HLSyn*.

Uma vez que se aplica o processo de diminuição da dimensionalidade dos dados, precisa-se ter a prova de que tal redução não vem a impactar a acurácia do sistema de

maneira negativa, ou mesmo deve-se provar que mesmo com perdas de precisão, o ganho computacional é o melhor caminho. No caso do problema aqui trabalhado, teve-se melhoria pelos dois lados, ou seja, obteve-se ganho de acurácia por parte de RNA (diminuição dos erros nas suas saídas e redução do custo computacional, já que os coeficientes utilizados foram reduzidos para aproximadamente 26% do total inicial (passando de 94 para 24 coeficientes). A comparação geral destes fatores podem ser visualizados na Figura 4.5 A e B.

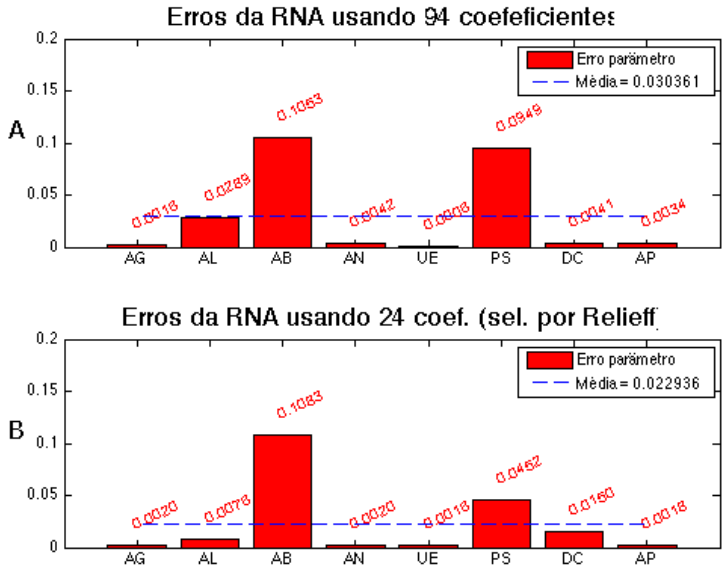

Figura 4.5: Utilização do *RReliefF* diminui a média dos erros da RNA e a dimensionalidade dos dados em experimentos com as palavras *'air', 'end', 'no' e 'gill'*.

Um ponto importante a ser destacado nesta etapa da pesquisa, é o reconhecimento da importância de se passar à rede neural o coeficiente *F0*, pois dado os experimentos com *RReliefF* verificou-se que este coeficiente proporciona ganho de informação a basicamente todos os parâmetros *HLSyn* a serem extraídos. Portanto deste ponto em diante verificou-se a necessidade de repassar além dos coeficientes já utilizados anteriormente (*MFCC+E, PLP+E, FBANK+E* e *MELSPEC+E*) o coeficiente *F0* juntamente com a sua primeira e segunda derivada extraído por meio do software *Straight* [Liu e Kewley-Port 2004] inserido no framework *NSM*.

4.1.3 Dados de treino da rede neural

Como a rede neural utilizada no interior do *NSM* necessita de aprendizado supervisionado são utilizados dados rotulados que representam uma visão geral do problema a

ser abordado. A escolha de tais dados influenciam diretamente os resultados gerados pela rede, já que é a partir dos mesmos que a rede poderá generalizar o erro do conjunto de treino e assim aprender como converter coeficientes físicos da fala humana para parâmetros do sintetizador *HLSyn*.

Como dados de treino foram utilizados 40 frases selecionadas do conjunto *Harvsents* desenvolvido em [Rothauser et al. 1969] e sintetizadas pelo software DECTalk [Hallahan 1995]. O diferencial de se utilizar as frases selecionadas do conjunto *Harvsents* é que as mesmas são foneticamente balanceadas, desta forma, busca-se equilibrar a quantidade de ocorrências das pronuncias dos mais variados tipos de fonemas no processo de treino da rede.

A razão de se utilizar sinais de voz sintética como dados de treino da rede neural em questão é por que esta foi a melhor forma de apresentar à rede neural uma visão geral e controlada (sem ruídos e fatores naturais) do problema de *utterance copy*.

4.1.4 Utilização do *HLSyn*

Como foi citado anteriormente, o *NSM* tem por principal função extrair estimativas dos parâmetros do sintetizador *HLSyn*, desta forma, como se tratam de estimativas e não do parâmetro final, é esperado que os valores encontrados pelo framework seja aproximações brutas do que vem a ser realmente o valor final de um parâmetro do sintetizador. Porém interno ao framework desenvolvido existe uma instância do sintetizador de voz *HLSyn* que tem por principal objetivo sintetizar arquivos *.hlsyn* gerados ao final dos processos executados pelo *NSM*. Com isso, são gerados resultados de voz sintética intermediários.

Relacionado aos resultados intermediários gerados pela execução de processos referentes ao framework *NSM*, pode-se analisar a Figura 4.6 que mostra a avaliação do sinal de voz gerado pelo framework, utilizando como voz alvo a palavra sintética ‘air’, gerado por *TTS* (*Text-To-Speech*) no *DECTalk* [Hallahan 1995] e avaliado pelo PESQ [*ITU-T Recommendation P.862. Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-To-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs* 2001].

4.1.5 Visão geral do framework

A iniciativa de utilizar um framework baseado em redes neurais artificiais no processo de estimação de parâmetros específicos de um sinal de voz consiste em propiciar que os

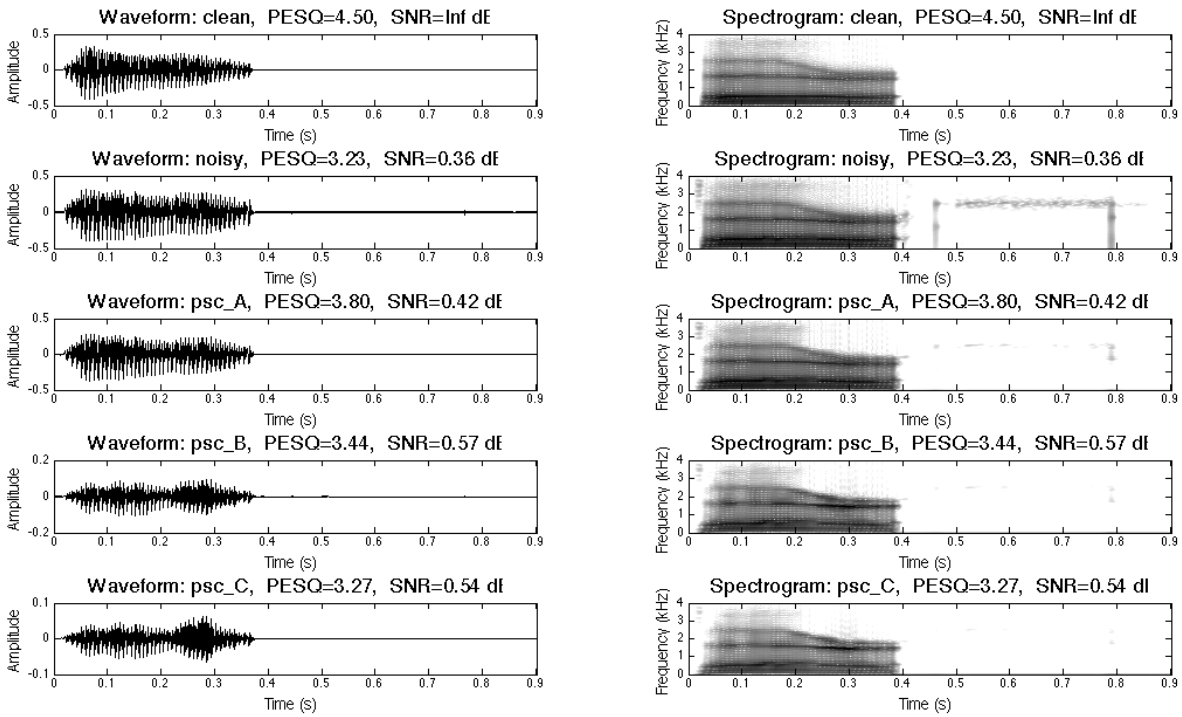

Figura 4.6: Sinal da palavra 'air' computado pelo *NSM* e avaliado por PESQ.

resultados obtidos com esta técnica possam ser refinados por um algoritmo de otimização. Desta forma, almeja-se a diminuição do espaço de busca por parte do algoritmo de otimização, diminuição do custo computacional da operação e melhoria significativa nos resultados finais.

Avaliações objetivas sobre custos computacionais não são trabalhadas por esta pesquisa, porém a afirmação de que o custo computacional é reduzido é uma hipótese fundamentada na capacidade da técnica *NSM* obter resultados melhores ou aproximados dos coletados a partir de framewoks como *WinSnoori* ou mesmo *HLSpeechGA*.

A utilização de softwares, como *Praat* e *Straight* vem propiciar que o framework se baseie nas *features* de voz por eles gerados e a partir daí poupe recursos computacionais por parte da RNA, já que a extração de tais *features* já são de domínio da comunidade de pesquisa de voz.

A necessidade de se utilizar um algoritmo como o *RReliefF* para realizar o processo de diminuição da dimensionalidade dos coeficientes extraídos pelo *HTK* se faz necessária, já que coeficientes como os do tipo *MFCC*, por exemplo, apresentam valores semelhantes dependendo de seu grau, como pode ser visualizado na Figura 4.7 Coeficientes 25 a

32, onde observa-se que os valores referentes a cada coeficiente são similares aos seus vizinhos, portanto pode-se ter coeficientes muito parecidos.

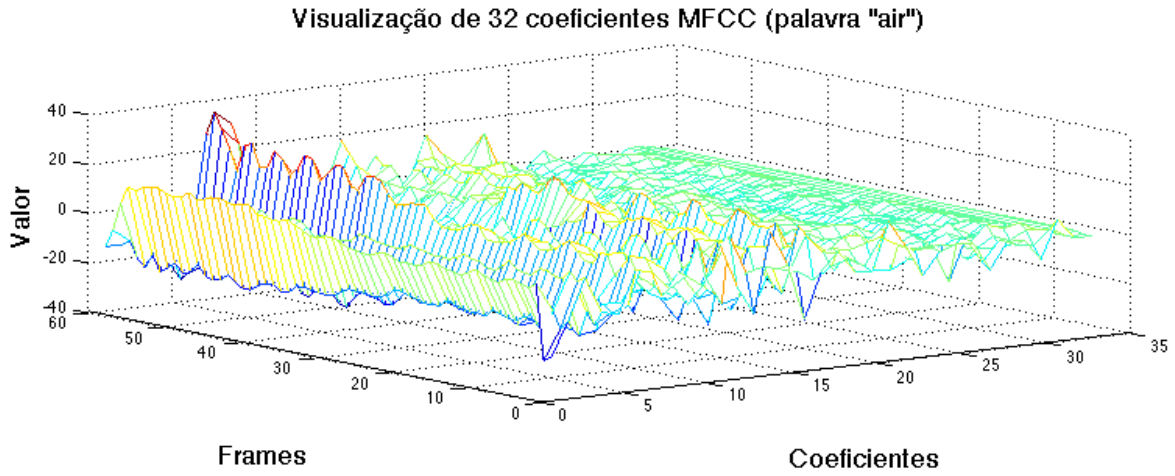

Figura 4.7: Visualização de 32 coeficientes *MFCC*.

Uma vez que a rede neural generalizada de função base radial, responsável pelo processo de *utterance copy* presente no *NSM* é treinada, pode-se entender esta como sendo uma generalização da função f que torna equivalente os coeficientes físicos extraídos da voz, aos parâmetros do sintetizador *HLSyn*. Caracterizando assim o processo de inversão, já que esta função f apresenta dois sentidos, pois uma vez que se tem os coeficientes de voz pode-se alcançar os parâmetros do sintetizador e por outro lado, uma vez que se tem os parâmetros do sintetizador pode-se alcançar os coeficientes da voz, como representado na Figura 4.8. Deve-se notar que a referida figura faz a representação de duas redes neurais diferentes para generalizar o problema em questão.

Deve-se ressaltar que o problema defendido por este trabalho, no que tange os fundamentos do *NSM*, estão somente relacionados ao processo de regressão representado pela rede neural da esquerda presente na Figura 4.8, ou seja, o processo de regressão entre os coeficientes de voz para os parâmetros do sintetizador *HLSyn*.

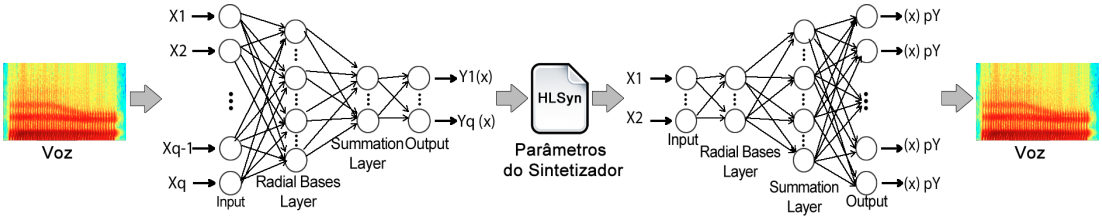

Figura 4.8: Representação do processo de inversão em *utterance copy*.

Uma visão geral do framework *NSM* é observada na Figura 4.9, uma vez que esta mostra o fluxo de processos realizados e relacionados com os ítens a seguir.

- (a) Voz alvo;
- (b) Extração dos coeficientes *MFCC, PLP, FBANK* e *MELSPEC*;
- (c) Extração da *feature F0*;
- (d) Extração das formantes *F1, F2, F3* e *F4* (*features*);
- (e) Execução de algoritmo *RReliefF* (somente no treino da rede) para diminuir a dimensionalidade dos dados extraídos em (b);
- (f) Treino ou teste da rede neural;
- (g) Construção do arquivo *HLSyn* e síntese;
- (h) Obtenção de voz cópia da original;

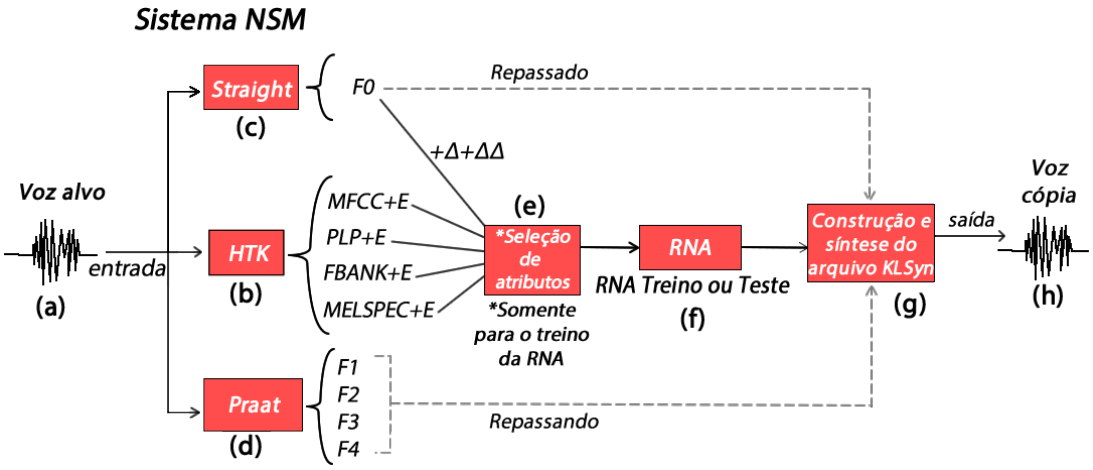

Figura 4.9: Estrutura do framework *NSM*.

4.2 Descrição do framework *HLSpeechGA*

A estrutura de um arquivo de entrada do sintetizador *HLSyn* é formanda por 13 colunas, representando cada parâmetro do sintetizador e linhas representando a quantidade *n* de amostras do sinal de voz a ser sintetizado, nos experimentos realizados por este trabalho adotou-se uma quantidade de amostras igual a 71, o que significa que cada linha (*frame*) do arquivo *.hlsyn* produz 71 amostras do sinal de voz.

De maneira geral o framework *HLSpeechGA* atua na estimação otimizada dos parâmetros do *HLSyn*, desta forma toda sua estrutura é baseada em uma codificação que permite abstrair as principais características do problema de imitação da fala.

O *HLSpeechGA* é baseado no algoritmo NSGA-II (*Non-Dominated Sorting Genetic Algorithm II*)) [Srinivas e Deb 1994], as principais informações referentes aos processos por este realizados são descritas nos tópicos a seguir.

4.2.1 Algoritmo *NSGA-II*

Assim como a versão *newGASpeech*, a versão *HLSpeechGA* utiliza *NSGA-II* [Srinivas e Deb 1994] como algoritmo inteligente capaz de realizar o processo de imitação da voz. Porém, deve-se destacar que tal implementação foi feita de maneira adaptada e não utiliza de todas as características do algoritmo *NSGA-II*.

Uma das modificações realizadas é com relação a utilização de somente um objetivo ao invés de múltiplos. Tal adaptação se fez necessária devido os melhores resultados coletados dentre as pesquisas realizadas terem sido processados utilizando somente o objetivo *RMSE*, como comprovado no subtópico 'Função Objetivo' mais a frente.

Outra diferença com relação as práticas adotadas pelo algoritmo *NSGA-II* é quanto a codificação, já que este algoritmo é indicado para trabalhar com codificação binária ao invés de codificação real [Srinivas e Deb 1994], como é o caso do framework *HLSpeechGA*.

Codificação dos indivíduos

A linha do arquivo *.hlsyn* a ser produzido pelo framework é utilizada para representar um indivíduo, sendo assim, o cromossomo deste é representado por um vetor de números reais, onde cada posição deste vetor equivale ao valor do parâmetro a ser estimado. Porém existe a possibilidade desta codificação mudar de acordo com a importância do *frame* a ser otimizado em relação ao sinal de voz alvo a ser produzido. À esta particularidade do problema utiliza-se a implementação da técnica de *Look-Ahead*, em português 'olhar para frente'.

Na Figura 4.10 A) pode-se visualizar o modelo de codificação de um cromossomo normal, ou seja, sem a utilização de *look-ahead* e na Figura 4.10 B) um cromossomo com o *look-ahead* de tamanho 1, ou seja, este cromossomo apresenta 1 frame a mais a ser computado. Em ambos exemplos as variáveis $V1$ à $V8$ representam os valores dos respectivos parâmetros *HLSyn*.

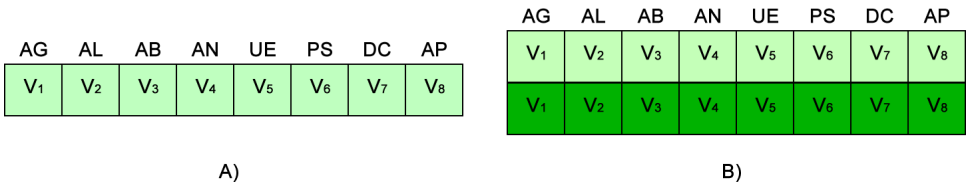

Figura 4.10: A) Estrutura do cromossomo normal e B) com *look-ahead*.

Implementação de *look-ahead*

A necessidade de se utilizar a implementação do *look-ahead* foi verificada em [Trindade et al. 2013] na versão *newGASpeech* do framework, que utiliza o sintetizador *KLSyn88* como base. Nesta versão do framework foi verificado, através de análises, que a estimação dos parâmetros de um determinado frame pelo framework pode gerar um sinal de voz equivalente ao sinal alvo. Porém, em algumas situações, há configurações que impedem que a próxima otimização (dos próximos *frames*) encontre o sinal desejado. Ou seja, o erro na configuração de um frame pode ser transmitido pra *n* subsequentes, resultando em um sinal distorcido. Devido a esse comportamento, percebeu-se a necessidade de aumentar o escopo do problema, aumentando a visão de análise do framework em relação a um frame, ou seja, implementação do *look-ahead*.

De acordo com os processos criados por [Trindade et al. 2013] a avaliação de um frame de *look-ahead* no framework *newGASpeech* é semelhante a realizada no frame principal, já que é efetuada a comparação entre as características comportamentais do sinal sintetizado e o sinal alvo. Como o cromossomo é formado por *n frames* de parâmetros, estes são sintetizados. O sinal gerado é comparado com o sinal alvo respectivos aos *frames*. Porém, como o objetivo continua sendo o frame principal, para cada frame de *look-ahead* é adicionado um peso. Esse peso é decrescente, ou seja, quanto mais afastado do frame principal, menor será seu peso na avaliação do cromossomo. Daí é possível definir a relevância que cada *frame* pertencente ao *look-ahead*, como na Figura 4.11, que mostra um frame com um *look-ahead* igual a 2.

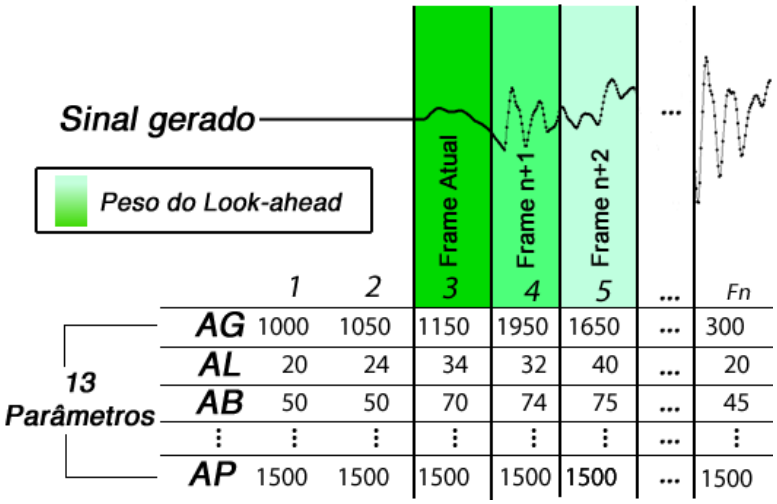

Figura 4.11: Representação dos *frames* que formam o cromossomo. Parâmetros referentes ao frame principal "Frame Atual"e aos frames de *look-ahead* "Frame n+1"e "Frame n+2".

A forma de implementação do *Look-ahead* apresentada na versão *newGASpeech* do framework defendido por [Trindade et al. 2013] foi adotada na versão *HLSpeechGA* (defendida por este trabalho) em sua integra, salve a modificação de redução de parâmetros utilizados entre tais frameworks, já que o newGASpeech usa 42 parâmetros devido se baseado no sintetizador *KLSyn88* e o *HLSpeechGA* utilizar somente 13 parâmetros devido se baseado no sintetizador *HLSyn*.

Função objetivo

O cálculo de *Root Mean Squared Error* (*RMSE*), é utilizado como função *fitness* para avaliar a solução mais apropriada do algoritmo genético. Outras funções foram testadas, como Distorção Espectral (*DE*) e Correlação Cruzada (*CC*), porém o cálculo *RMSE* foi a que apresentou melhor desempenho no algoritmo genético, segundo [Trindade et al. 2013] na versão *newGASpeech* e também em novos testes com a versão *HLSpeechGA*.

O *HLSpeechGA* executa o algoritmo genético para cada linha do arquivo *.hlsyn* a ser otimizado ou mesmo construído do zero. Desta forma, o processo de imitação da fala será mais demorado quanto maior for a duração da voz alvo. Os processos avaliativos do algoritmo genéticos são fundamentados nos sinais de voz produzidos pela síntese dos cromossomos dos indivíduos da população do algoritmo, isto, levando-se em consideração que cada linha (ou *frame*) do arquivo *.hlsyn* influencia na síntese dos *frames* a sua frente, já que o sintetizador em questão possui memória [Heid e Hawkins 1998].

A escolha da função de avaliação, dado o problema de imitação de voz apresentado, pode ser definida por diferentes metodologias uma vez que pode-se avaliar tanto os erros dos parâmetros *HLSyn* gerados, ou mesmo avaliar o sinal de voz resultante do processo de construção do sinal como um todo. Diante destas particularidades, realizou-se um estudo comprobatório, comparando as métricas *RMSE*, *DE* e *CC*, realizado de duas maneiras:

- Calculando o erro (*RMSE*) do sinal de voz cópia gerado pelo sistema;
- Calculando os erros (*RMSE*) dos parâmetros do sintetizador *HLSyn*, gerado na saída do sistema.

Para as duas formas acima, foram executadas simulações do *HLSpeechGA* utilizando como funções objetivo as seguintes combinações: $RMSE$, CC, DE, $RMSE + CC$ e $RMSE + DE$. O resultado é apresentado na Figura 4.12.

Como pode-se notar, de acordo com os erros encontrados na Figura 4.12 A-F coluna 1, a função *RMSE* quando utilizada sozinha, gera um sinal de voz com os menores erros (Figura 4.12 A1). Porém o mesmo não pode ser dito ao se avaliar tal experimento por

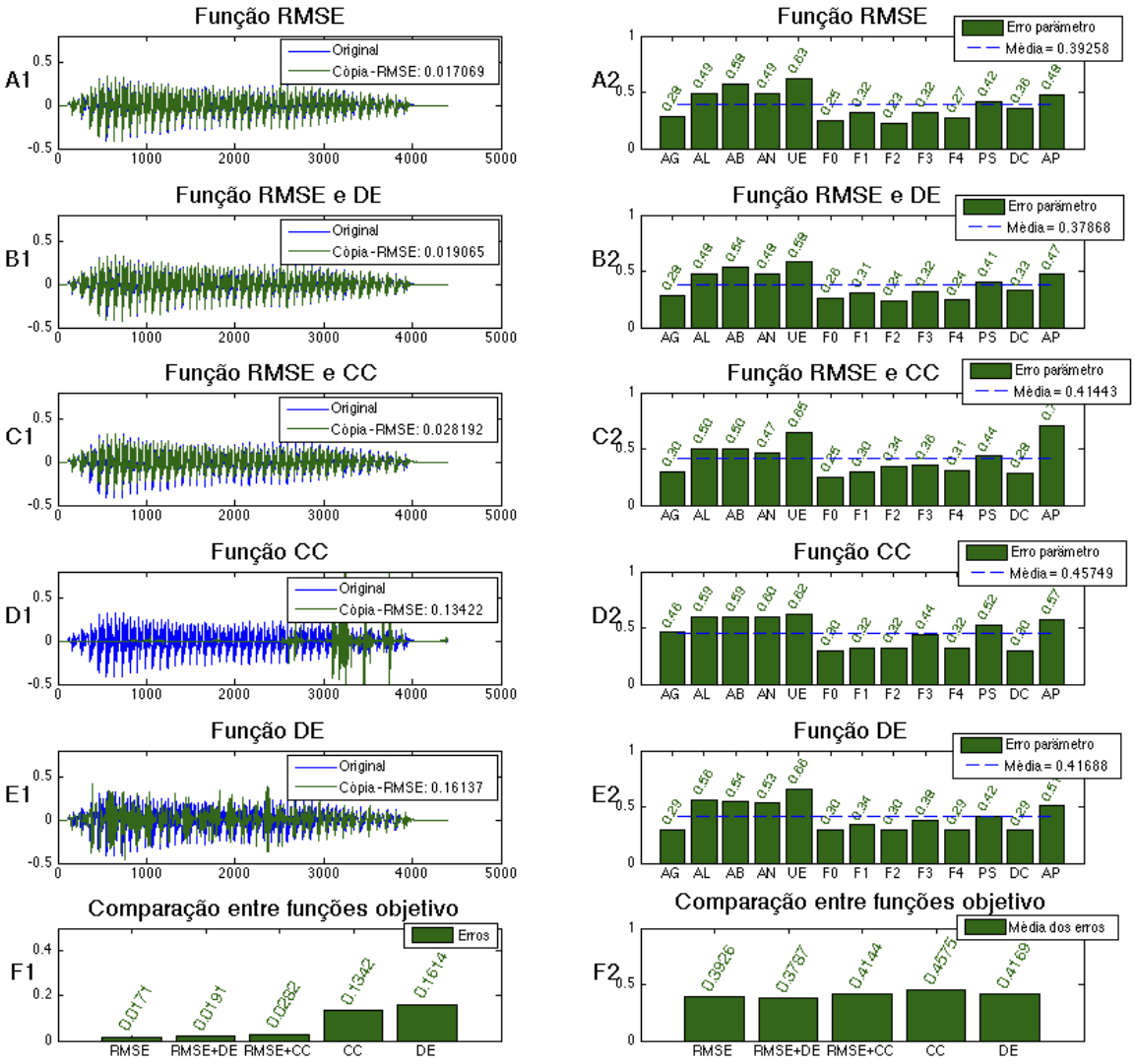

Figura 4.12: Comparativo entre funções objetivo utilizadas no *HLSpeechGA* para a imitação da voz sintética 'air'.

meio dos erros encontrados nos parâmetros *HLSyn*, já que as funções $RMSE + DE$, ao serem utilizadas como função objetivo do algoritmo genético, proporcionam uma menor média dos erros dos parâmetros *HLSyn*, segundo a Figura 4.12 B2.

Desta forma, para se chegar a um consenso de qual sinal de voz gerado é melhor, e assim entender qual função a ser utilizada pelo *HLSpeechGA*, baseou-se na premissa que dado o sintetizador *HLSyn* e suas entradas, é possível que diferentes combinações dos va-

lores dos parâmetros gerem sinais idênticos. Desta forma, altos erros dos parâmetros não refletem necessariamente em um sinal de voz diferente da voz alvo. Com isso, entende-se como melhor saída avaliar o erro do sinal de voz gerado no final de todo o processo. Portanto, de acordo com a Figura 4.12 pode-se entender que a função *RMSE* é a melhor a ser utilizada como função objetivo pelo GA do framework *HLSpeechGA*.

Cruzamento

Como é observado na natureza, a busca por melhores indivíduos é realizado através de processos de seleção, a competição entre indivíduos é algo natural para que ocorra a evolução da população como um todo. Em implementações de algoritmo genético isso não é diferente, já que este é inspirado na forma que a evolução natural ocorre. Com isso, um aspecto que não poderia ficar de fora deste algoritmo é o cruzamento entre indivíduos (nesta caso, soluções).

O cruzamento utilizado pelo framework *HLSpeechGA* é o cruzamento real, com base em ponto de corte aleatório. Este é um cruzamento relativamente simples e pode ser entendido a partir da visualização da Figura 4.13, que mostra dois pais gerando dois filhos.

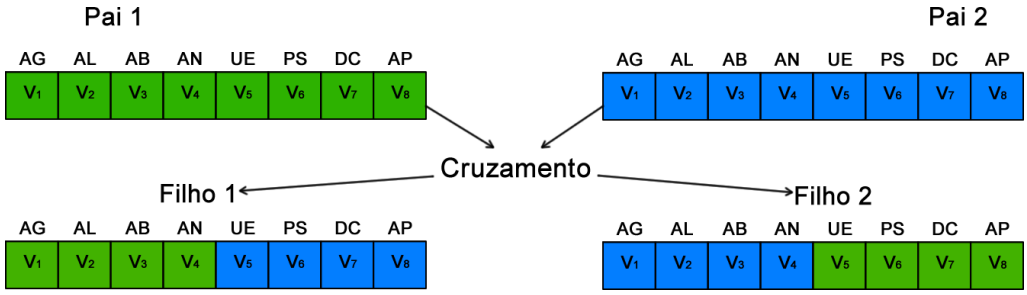

Figura 4.13: Cruzamento real por ponto de corte.

O processo de cruzamento presente na versão *newGASpeech* do framework foi mantido na versão *HLSpeechGA*.

Mutação

A mutação realizada pelo *HLSpeechGA* teve modificações consideráveis se comparado com a mutação da versão anterior do framework, já que neste último a mutação ocorre de maneira especializada à parâmetros do sintetizador *KLSYN88*, mas especificamente aos parâmetros *F0* e *AV*.

Deve-se destacar que o processo de mutação presente na versão *HLSpeechGA* do framework é mais simples do que o do *newGASpeech* devido a nova versão tratar os parâ-

metros de maneira igual, já que o sintetizador *HLSyn* reduz as redundâncias e particularidades existentes entre os parâmetros do sintetizador *KLSYN88* (neste caso, interno ao *HLSyn*).

O processo de mutação implementado pelo *HLSpeechGA* é fundamentado em mutação real. Desta forma, uma posição do gene é sorteada e em seguida esta sofre mudança em seu valor de maneira aleatória, tomando cuidado para que o novo valor não esteja fora dos limites esperados pelo sintetizador, vistos na Tabela 2.2.

4.2.2 Utilização do *HLSyn*

O sintetizador *HLSyn* é utilizado como base para todo o processo de síntese de voz realizado por este framework, desta forma cálculos de avaliação dos indivíduos, *look-ahead* e também a síntese do sinal final (cópia da voz alvo) são realizadas por meio deste sintetizador.

4.2.3 Visão geral do framework

Ao se utilizar um algoritmo de otimização, para realizar a estimação dos parâmetros de um sintetizador por formantes, a metodologia aplicada a este processo deve se basear nas principais características que regem o funcionamento de um sintetizador de voz por formantes como o *KLSyn88* ou mesmo o *HLSyn*. Tais características fazem referência aos valores de entrada que este sintetizador apresenta e suas referidas faixas.

Como citado anteriormente, para cada *frame HLSyn* a ser gerado pelo *HLSpeechGA*, o algoritmo genético realiza um percurso completo de sua execução. Sendo a ideia principal fazer com que o algoritmo realize a busca dentro de espaço de soluções que são as entradas do sintetizador *HLSyn*. Vale lembra que este problema de natureza combinatória, ou seja, ideal para algoritmos de otimização como o algoritmo genético aqui descrito.

Desta forma, a metodologia aplicada aos processos aqui apresentados são fundamentadas nos principais aspectos do sintetizador de voz *HLSyn* aplicadas ao algoritmo genético *NSGA-II*, implementado no framework *HLSpeechGA*.

A estrutura geral do *HLSpeechGA* pode ser observada na Figura 4.14.

4.3 União *NSM+HLSpeechGA*

Como descrito anteriormente, o motivo de se desenvolver o framework *NSM* foi objetivando que os resultados obtidos com este pudessem ser repassados para o HLSpeechGA

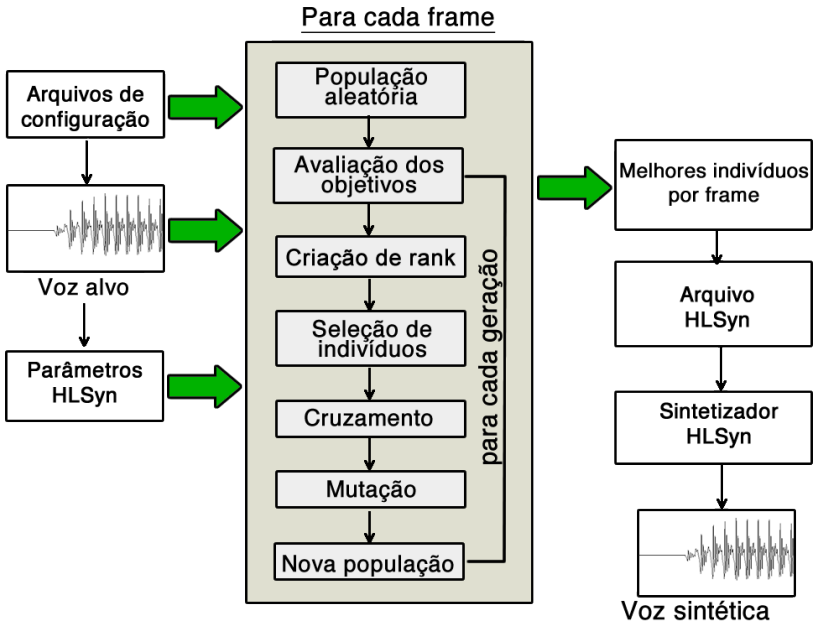

Figura 4.14: Estrutura do framework *HLSpeechGA*.

e assim otimizados.

Os benefícios de se utilizar esta técnica estão relacionados ao fato de se ter a diminuição do espaço de busca a ser explorado pelo algoritmo genético. À integração entre os dois frameworks, dá-se o nome de *NSM+HLSpeechGA*.

A integração *NSM+HLSpeechGA* é realizada com base nos arquivos *.hlsyn* construídos na saída do sistema *NSM*, ou seja, saídas da RNA juntamente com *features* extraídas pelo softwares *Praat* e *Straight*, Figura 4.9 (g).

Ao se utilizar somente o *HLSpeechGA* deve-se levar em conta que a inicialização da população do algoritmo genético deste framework ocorre de maneira aleatória, somente tendo-se cuidado para que os valores dos cromossomos dos indivíduos, ou seja, valores de parâmetros *HLSyn*, não sejam gerados fora dos intervalos esperados pelo sintetizador, Tabela 2.2.

Com a integração *NSM+HLSpeechGA* a forma de se realizar a inicialização da população é feita calculando-se intervalos de 50% para mais e para menos dos valores fornecidos pelos arquivos gerados pelo *NSM*, conseguindo-se assim a diminuição do espaço de busca a ser explorado pelo algoritmo genético.

A visualização do parâmetro *AG* gerado pelo *NSM* juntamente com o intervalo (50% para mais e para menos) utilizado para realizar a inicialização da população do *HLSpeechGA* pode ser observado na Figura 4.15.

Na figura citada acima, é criado um espaço de busca reduzido com base nas saídas

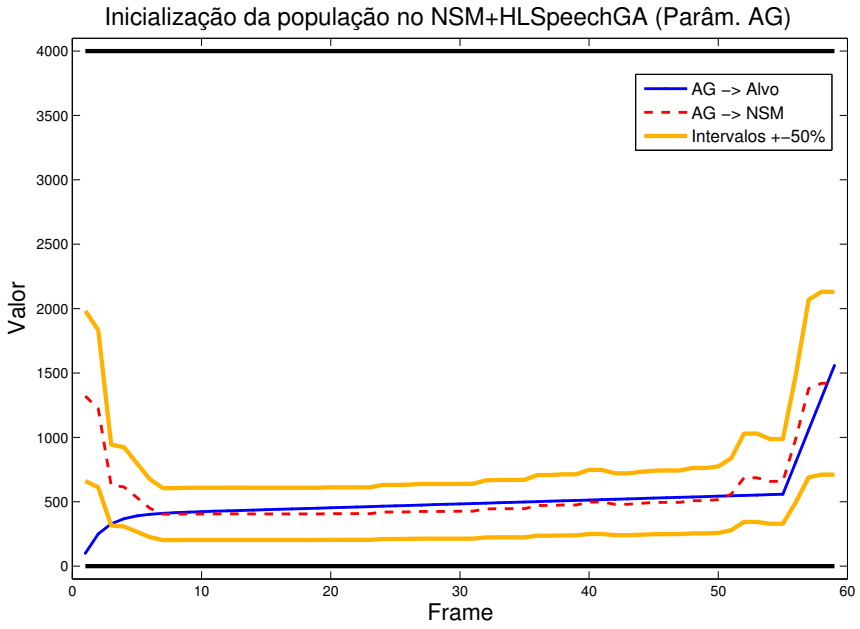

Figura 4.15: Os intervalos (linha em amarelo) calculados a partir da multiplicação dos valores do parâmetro *AG* por 0.5 e 1.5 são utilizados como faixas para a inicialização da população na abordagem *NSM+HLSpeechGA*, reduzindo o espaço de busca original (linhas pretas).

do *NSM*, desta forma o *HLSpeechGA* já inicia sua população estando potencialmente menos distante da configuração do parâmetro desejado. Observe que a maioria dos valores objetivados (linha em azul) estão dentro dos intervalos. Outra observação é que os novos intervalos são calculados para todos os parâmetros *HLSyn* a serem extraídos.

Visão geral da abordagem

O processo como um todo executado pelo *NSM+HLSpeechGA* pode ser entendido facilmente como a aplicação de técnicas de aprendizagem de máquina para realizar a extração otimizada dos parâmetros do sintetizador *HLSyn*, baseando-se em Redes Neurais e Algoritmo Genético. Daí têm-se a *RNA* atuando como pré-processamento dos parâmetros a serem gerados.

Com a ajuda do *NSM* o framework *HLSpeechGA* atua em um espaço de busca menor, já que as inicializações das populações do AG existente neste framework são realizadas com base nas saídas da RNA. Contudo, é de interesse que os resultados com a abordagem *NSM+HLSpeechGA* sejam melhores do que com ambas técnicas executadas separadamente. Sendo necessário calcular de maneira estimada o tamanho da redução do espaço de busca a ser explorado nas duas situações.

O tempo de processamento e qualidade dos resultados de algoritmos de otimização estão diretamente ligados com o tamanho do espaço de busca da solução a ser explorada [Langdon 1998].

Duas variáveis do algoritmo genético, que estão diretamente ligadas ao processo de exploração do espaço de solução do problema, é a quantidade de indivíduos e o número de gerações, pois quanto maior a quantidade de indivíduos maior será a capacidade do algoritmo em explorar grandes áreas do espaço de solução e quanto mais gerações melhor este espaço será explorado [Langdon 1998].

Uma estimação do tamanho do espaço de busca do problema aqui apresentado pode ser calculado com base no princípio de combinação. Uma vez que se tem os 13 parâmetros do sintetizador *HLSyn* representados pelo cromossomo de um indivíduo do algoritmo genético presente no *HLSpeechGA*, pode-se calcular o espaço de busca pela multiplicação do número de possibilidades de valor que cada parâmetro pode assumir, com isso, de acordo com a Tabela 2.2 entende-se o espaço de busca total do problema como $1.5351e^{+41}$, que são referentes ao número de combinações possíveis.

Uma forma simples de mensurar a redução do espaço de busca realizado pela abordagem *NSM+HLSpeechGA* é calculando o número de possibilidades que pode-se ter para cada parâmetro *HLSyn* a ser gerado, tomando como verdade (para este cálculo) que os parâmetros gerados pelo *NSM* apresentem valores igual à metade exata do valor apresento na Tabela 2.2. Com isso ao se calcular os novos intervalos para cada parâmetro pode-se calcular o novo espaço de busca. Neste caso, este novo espaço é de aproximadamente $1.8739e^{+37}$ (aproximadamente $0,01\%$ do espaço anteriormente calculado).

Apesar do custo computacional não ter sido avaliado neste trabalho, estima-se um ganho computacional e melhoria dos resultados.

4.4 Conclusão do capítulo

Neste capítulo foram apresentadas as principais características da técnica defendida por esta pesquisa, desta forma foram trabalhados de maneira detalhada os aspectos específicos que envolvem o framework *Neural Speech Mimic* e o *HLSpeechGA*, buscando apresentar uma visão minuciosa de cada um, juntamente com alguns experimentos que buscam servir de prova de conceito para questões referentes a escolha das tecnologias e metodologias empregadas a estes.

Foi definida a forma de integração dos frameworks *NSM* e *HLSpeechGA*, um dos focos deste trabalho, uma vez que pretende-se ter melhores resultados na solução do problema de *utterance copy*.

Capítulo 5

Experimentos

O objetivo principal deste capítulo é demonstrar e descrever através de experimentos práticos em ambiente computacional controlado, os resultados da metodologia anteriormente descrita é válida para a comunidade de pesquisas relacionadas à voz.

Primeiramente é realizada uma descrição metodológica dos experimentos a fim de se ter a definição do que vem a ser o teste ideal que mensure o quão bom pode ser o desempenho do framework proposto. Em seguida é realizada a descrição dos experimentos práticos juntamente com exposição de gráficos e tabelas com dados sobre os experimentos.

Vale ressaltar logo inicialmente que os testes são divididos em dois grupos, que são: experimentos com voz sintética e experimentos com voz natural.

5.1 Metodologia

Uma vez que esta pesquisa se baseia em princípios quantitativos para realizar a avaliação da utilização do *NSM*, do *HLSpeechGA*, do *NSM+HLSpeechGA* e do *WinSnoori* ao problema proposto, se faz necessária a definição de uma prova de conceito que permita demonstrar e medir o potencial da metodologia aqui defendida.

Os primeiros aspectos a serem levados em consideração ao se definir uma prova de conceito para os softwares em questão passam por três pontos principais: definição do conjunto de dados, definição de métricas avaliativas e aplicação de tratamentos nos sinais gerados;

- **Definição do conjunto de dados:** uma vez que o objetivo do teste é avaliar a capacidade do sistema em copiar a voz humana, espera-se que os testes sejam baseados em um número considerável de sinais de voz (relacionados aos mais diferentes tipos

de palavras pronunciadas por um ou mais locutores) a fim de se ter uma avaliação o mais geral possível do sistema como um todo;

- **Definição de métricas avaliativas:** uma vez que se tem um conjunto de vozes produzidas pelos experimentos, deve-se realizar a avaliação das mesmas com base em métricas objetivas que permitam mensurar as discrepâncias entre a voz alvo e a voz gerada e também avaliar a inteligibilidade (clareza) do sinal de voz produzido;
- **Aplicação de tratamentos nos sinais gerados:** para que se tenha o mínimo de interferência no processo de avaliação das métricas faz-se necessário a realização do alinhamento e corte de amostras não vozeadas dos sinais de voz cópias e originais.

Deve-se destacar que todos os experimentos realizados por esta pesquisa foram computados por um cluster Linux formado por 8 *host*, sendo que cada um possui um processador Intel Xenon 3.0 GHz com 8 núcleos cada e memória de 12 GB.

5.1.1 Base de dados

Buscando atender aos três pré-requisitos citados anteriormente esta pesquisa definiu que os experimentos realizados como prova de conceito tivessem como dados sinais de voz sintética e natural. O motivo de se utilizar voz sintética para tais experimentos está relacionado a necessidade de mensurar o quão bom o sistema é em condições ideais, ou seja sem a interferência direta de ruído ou mesmo particularidades da forma de falar do falante. Já com relação a voz natural, é a partir destes testes que o sistema será testado de maneira mais intensa, já que a voz natural apresenta fatores que tornam o trabalho dos softwares mais difícil.

Como base de dados de voz sintética foram selecionadas 30 palavras sintetizadas pelo sistema *Text-To-Speech (TTS)* DECTalk [Hallahan 1995], utilizando 3 falantes masculinos diferentes (10 palavras para cada). Como forma de referenciar tais falantes este trabalho os chama da mesma forma como o software DECTalk os chama, ou seja, *Paul, Harry* e *Frank*. A lista total destas palavras pode ser observada na Tabela 5.1.

A opção de utilizar palavras sintetizadas pelo *DECTalk* foi definida como essencial para o processo aqui apresento, já que este software oferece como resultado da síntese os parâmetros do tipo *HLSyn* referente a voz produzida. Deve-se notar também que, uma vez utilizadas sinais provenientes do *DECTalk*, ficou-se limitado à pronúncias de palavra da língua inglesa.

Como base de dados de voz natural foram selecionadas 20 sinais de voz da base de dados TIDIGITS [TIDIGITS n.d.], tendo-se o cuidado de selecionar sinais de 2 falantes

diferentes (10 sinais de cada). Este trabalho utiliza a mesma referência que a base TI-DIGITS utiliza para se referir aos falantes, sendo assim: *Ara* e *Fta*. Os 10 sinais de voz selecionados a partir de cada falantes são as pronúncias em inglês dos números 0 (zero) até 9 (nove).

Optou-se por utilizar a base de dados TIDIGITS, baseada no idioma Inglês, já que para voz sintética foram utilizadas palavras do mesmo idioma.

Tabela 5.1: Base de voz sintética.

Índice	Frank	Harry	Paul
1	air	awe	are
2	dill	earl	end
3	farl	bean	is
4	gnaw	else	no
5	hurl	gill	there
6	jam	our	bar
7	law	shoe	death
8	sit	tang	then
9	who	them	there
10	why	wish	toe

5.1.2 Métricas avaliativas

De maneira a não depender somente de uma métrica avaliativa, que produziriam conclusões pouco gerais, esta pesquisa realizou a seleção de quatro métricas que podem ser divididas em duas categorias: avaliação do erro e avaliação da qualidade, ambas referentes comparação dos sinais de voz originais e cópias.

Dentro da categoria de avaliação dos erros foram utilizados *Root Mean Squared Error (RMSE)* e a Distorção Espectral (DE). Já com relação a avaliação da qualidade foram utilizados o *Signal-to-noise Ratio (SNR)* e *Perceptual Evaluation of Speech Quality (PESQ)*, nos subtópicos a seguir uma breve descrição sobre cada um deles.

RMSE

O *Root Mean Squared Error (RMSE)* é uma métrica bastante popular utilizada freqüentemente em pesquisas voltadas aos mais diferentes assuntos. O interessante desta métrica é que esta sofre influência da média dos erros e pode-se dizer que o resultado da avaliação vem representar uma visão geral (no sentido de abrangência) dos erros encontrados entre os sinais avaliados. Deve-se destacar também que esta métrica coloca mais pesos em altos

erros do que em pequenos erros, sendo isso influenciado pelo resultado dos quadrados dos altos erros [Wackerly et al. 2008].

A formula que define esta métrica pode ser visualizada na Equação 5.1.

$$RMSE = \frac{1}{n}\sum_{j=1}^{n}(\theta_a(j) - \theta_s(j))^2 \tag{5.1}$$

Para esta métrica, quanto mais aproximado de zero o valor do erro, melhor é o sinal avaliado.

Distorção Espectral

Quando o objetivo é mensurar a força espectral entre dois sinais, neste caso chamado também de *Power Spectral Density (PSD)*, a Distorção Espectral [Ramachandran e Mammone 1995] pode ser usada a partir de

$$DE = \sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi}\left[10\log_{10}\left(\frac{S_x(e^{j\Omega})}{S_y(e^{j\Omega})}\right)\right]^2 d\Omega} \tag{5.2}$$

Onde $x[n]$ e $y[n]$ são sinais discretos no tempo e $S_x(e^{j\Omega})$ juntamente com $S_y(e^{j\Omega})$ são suas respectivas *PSDs*. Distorções espectrais são calculadas em *dB* e correspondem aos valores da raiz entre o $10log_{10}S_x(e^{j\Omega})$ e $10log_{10}S_y(e^{j\Omega})$ sobre a frequência Ω. Deve-se notar que a DE também pode ser usada para calcular o *mean-squared spectrum*. Para esta métrica, quanto menor o valor, melhor é o sinal avaliado.

SNR

Signal-to-noise Ratio (SNR) [Tan e Lindberg 2008], é uma métrica que realiza a medição da qualidade de sinais de voz em meio ruidoso. Sua definição mais geral é dada pela razão da potência de um sinal pela potência do erro do sinal. A relação sinal-ruído (neste caso sinal-erro) é calculada com base em *dB* (decibel) a partir de

$$SNR_{dB} = 10log_{10}\left[\left(\frac{P_{sinal}}{P_{erro}}\right)^2\right] \tag{5.3}$$

Onde P_{sinal} representa a potência do sinal original e P_{erro} representa a potência do erro (ou seja, $P_{sinal-sinalRuidoso} = P_{erro}$). Para esta métrica assume-se que quanto maior o valor do *SNR*, melhor é o sinal avaliado.

PESQ

O *Perceptual Evaluation of Speech Quality* [*ITU-T Recommendation P.862. Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-To-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs* 2001], também chamado de *PESQ*, é uma métrica muito utilizada por sistemas de telefonia, esta é capaz de mensurar de maneira aproximada um coeficiente na escala *Mean Opinion Score (MOS)*, equivalente a uma nota entre 0 e 5, que sugere a escala da qualidade de voz oferecida por um serviço. Nesta métrica é atribuído um *MOS* próximo de 0 (zero) para sinais muito diferentes e próximos de 5 para sinais similares (alta qualidade).

5.1.3 Alinhamento e tratamento dos sinais

Os sinais de voz gerados pelos softwares citados normalmente podem variar em número de amostras, algo em todo de 10 a 100 amostras para mais ou para menos, com isso os sinais ficam deslocados no tempo se comparados com os sinais originais. Desta forma antes de se aplicar as métricas do último passo da prova de conceito, os sinais comparados serão primeiramente submetidos a um procedimento que realiza a minimização do deslocamento que um sinal apresenta em relação ao outro, por meio do algoritmo de *Cross-correlation*, ou Correlação Cruzada como também é chamado [Stoica e Moses 2004].

O motivo de se utilizar *Cross-Correlation* para corrigir o deslocamento entre voz original e voz sintetizada é melhor visualizado ao se observar a Figura 5.1. A figura mostra que mesmo se tratando de sinais que apresentem diferenças como ruídos, ao executar o algoritmo de *Cross-correlation* os dois são alinhados de acordo com suas maiores semelhanças. Com isso, ao calcular o *MSE* (por exemplo), a diferença dentre os sinais é a mínima possível.

A eliminação de amostras não vozeadas é uma prática necessária ao se utilizar a métrica *RMSE* devido esta sofrer influência dos valores 0 (zero) presentes nestes tipo de amostras, já que a métrica baseia-se na função média como base para o cálculo do erro.

5.1.4 Visão geral da prova de conceito

A partir do que foi trabalhado nos subtópicos anteriores pode-se definir a prova de conceito desta pesquisa como sendo a realização dos seguintes passos:

- Seleção de um conjunto contendo 30 palavras sintetizadas pelo DECTalk [Hallahan 1995];

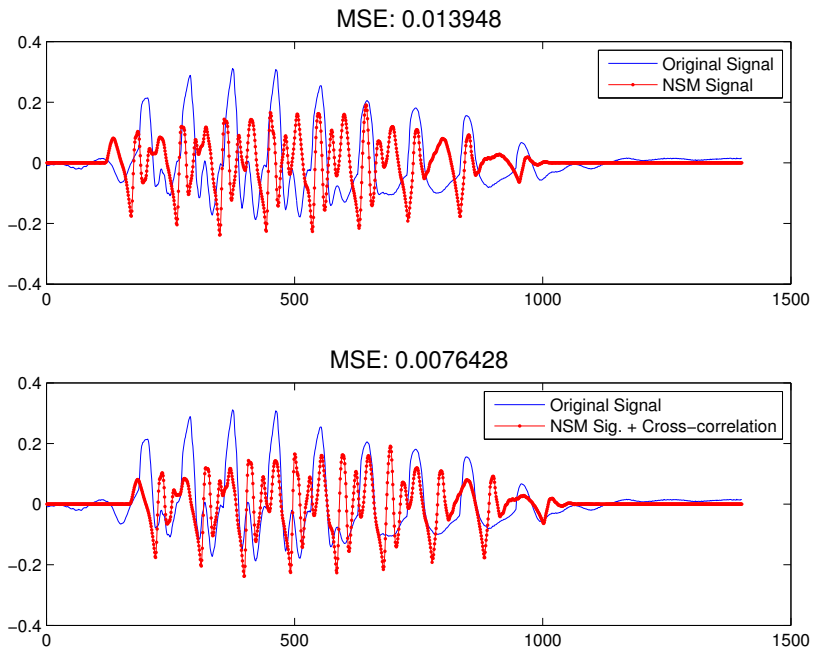

Figura 5.1: Sinal de voz da palavra *'with'* original e o proveniente do *NSM*.

- Seleção de um conjunto contendo 20 sinais de voz natural da base de dados TIDIGITS [TIDIGITS n.d.];
- Execução do *NSM*, *HLSpeechGA*, *NSM+HLSpeechGA* e *WinSnoori* para cada uma das 50 palavras selecionas. Nesta etapa são produzidos 200 sinais de voz sintética ao todo;
- Aplicação do algoritmo de *Cross-correlation* entre os sinais de voz produzidos e seus respectivos sinais originais;
- Eliminação de amostras não vozeadas dos sinais de voz gerados e seus sinais originais;
- Alinhamento dos sinais de voz produzidos nos experimentos;
- Avaliação dos sinais de voz com base nas métricas: *RMSE*, *DE*, *SNR* e *PESQ*.
- Análise da qualidade das formantes produzidas pelos sinais de voz cópias, gerados pelos sistemas;
- Resumo geral do desempenho dos sistemas juntando dados dos experimentos com voz natural e sintética.

5.1.5 Experimentos com voz sintética

Nas Figuras 5.2 a 5.5 são mostrados resultados da execução do A) *HLSpeechGA*, B) *NSM*, C) *NSM+HLSpeechGA* e D) *WinSnoori* mensurados a partir das métricas *RMSE*, *DE*, *SNR* e *PESQ*, para a base de voz sintética. Em seguida, Figura 5.6 é realizada a avaliação da qualidade das formantes produzidas nos experimentos realizados.

Avaliação do sinal segundo *RMSE*

Analisando a dispersão das avaliações apresentadas na Figura 5.2 pode-se afirmar que os menores erros dos sinais, calculados com base em *RMSE*, foram gerados pelo *HLSpeechGA+NSM* e o *HLSpeechGA* (empatados tecnicamente), seguidos do *NSM* e *WinSnoori*. Este último apresentou os maiores erros e portanto os piores resultados.

Ter o *HLSpeechGA* ou mesmo o *NSM+HLSpeechGA* produzindo cópias de sinais de voz com os menores erros do tipo *RMSE* podem ser explicados, já que a função objetivo do algoritmo genético é o cálculo do *RMSE* entre o sinal alvo e o sinal produzido pelo framework, porém realizado frame-a-frame do sinal. Com isso, as abordagens que utilizam algoritmo genético aqui defendidas se destacam nesta métrica.

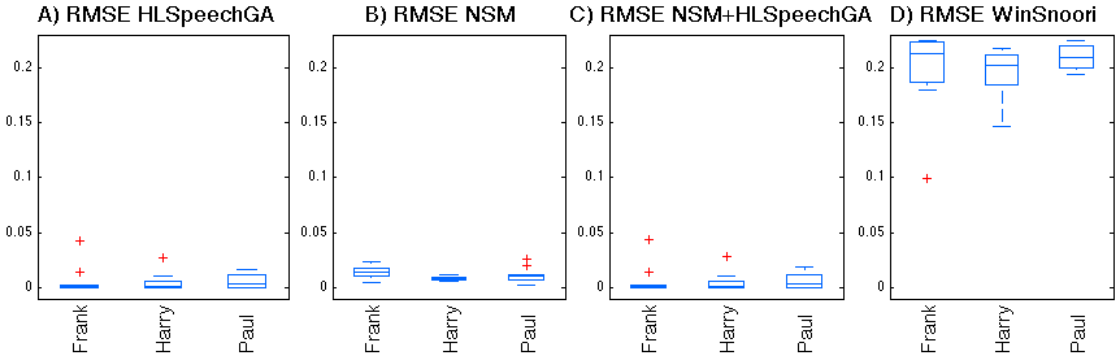

Figura 5.2: Avaliação do desempenho dos frameworks, para voz sintética, a partir da métrica *RMSE*.

Um ponto importante a ser destacado ainda com relação a Figura 5.2 é que este experimento mostra como os softwares utilizados são estáveis quanto a diferentes falantes (segundo a métrica *RMSE*), ou seja, os resultados dentre falantes se mostraram similares (neste caso, *Frank*, *Harry* e *Paul*) para cada ferramenta. Tal resultado é justificável, por se tratar de diferentes falantes do mesmo sexo, neste caso masculino, resultado este que pode não ser o mesmo caso sejam utilizados falantes masculinos e femininos, pois a voz feminina carrega características diferenciadas da voz masculina e que são diferentemente

interpretadas por sintetizadores por formantes como o *HLSyn* por exemplo.

O software *WinSnoori* foi quem apresentou os maiores erros *RMSE*, ficando na casa de aproximadamente 0.2, porém deve-se levar em conta que este software é voltado principalmente para o problema de *utterance copy* utilizando voz natural como entrada ao mesmo, como em [Laprie 2002]. Portanto tal baixo desempenho é justificável.

Outro fato observado na Figura 5.2 são os resultados encontrados pelo sistema *NSM*, pois pode-se notar que os erros dos sinais de voz gerados a partir deste estão próximos dos erros gerados pelo *HLSpeechGA* ou mesmo *NSM+HLSpeechGA*. Tais baixos erros são justificáveis uma vez que estes mensuram a capacidade da RNA utilizada em generalizar o problema de *utterance copy* em um ambiente ideal, ou seja, voz sintética.

Avaliação do sinal segundo a Distorção Espectral

A análise da Figura 5.3 mostra uma situação muito parecida com a encontrada na Figura 5.2, porém devido as características particulares da métrica distorção espectral (que se baseia no log da função *RMSE*) é possível visualizar a diferença entre os erros encontrados de maneira mais clara para cada abordagem, já que a diferença dentre as dispersões mostradas nos *boxplots* são mais evidentes.

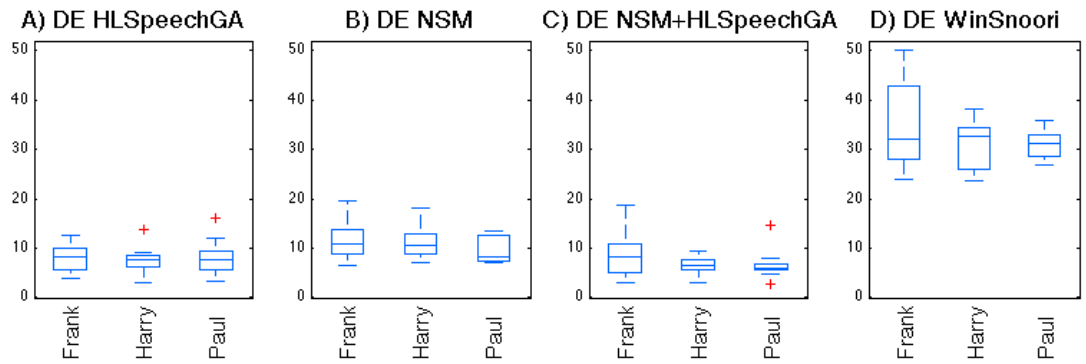

Figura 5.3: Avaliação do desempenho dos frameworks, para voz sintética, a partir da métrica Distorção Espectral.

Um fato importante é que esta métrica foi capaz de diferenciar resultados do *NSM+HLSpeechGA* e *HLSpeechGA*, anteriormente empatados de acordo com a métrica *RMSE* Figura 5.2 A) e C). Com isso, pode-se afirmar que os resultados encontrados pelo sistema *NSM+HLSpeechGA* são superiores, principalmente com relação ao falante *Paul*, que de maneira geral, pode-se dizer que este é o falante (entre três utilizados) que apresenta maior clareza na pronúncia dos sinais aqui utilizados, já que a maioria dos sistemas avaliados tiveram seus menores erros calculados a partir dos sinais deste falante.

Segundo a dispersão dos dados mostrados na Figura 5.3 A) e B) visualiza-se de maneira mais detalhada a diferença do desempenho obtido pelo *HLSpeechGA* e o *NSM*. Mesmo com o *NSM* obtendo erros mais altos do que o *HLSpeechGA*, evidencia-se que os resultados da rede neural (usada sozinha, ou seja no *NSM*) são relevantes, já que em termos de custo de tempo o *NSM* leva aproximadamente $0,21$ segundos para processar 1 frame de voz sintética, já o *HLSpeechGA* precisa de aproximadamente 927 segundos para fazer o mesmo.

Avaliação do sinal segundo *SNR*

Levando-se em conta que a métrica *SNR* é capaz de mensurar a existência de ruído entre dois sinais de voz, a visualização da Figura 5.4, mostra que as técnicas baseadas em algoritmo genético A) e C) apresentaram altos índices de *SNR* o que evidencia que as abordagens *HLSpeechGA* e o *NSM+HLSpeechGA* são capazes de gerar sinais com o mínimo de ruído. Este fato, é justificável já que o *HLSpeechGA* trabalha na otimização do processo de *utterance copy* construindo o sinal de voz cópia frame-a-frame, com isso ruídos ou mesmo desalinhamentos que poderiam ser produzidos na cópia são problemas abstraídos pela minimização dos erros através das iterações do algoritmo.

Os resultados coletados com o *WinSnoori*, mostram que esta ferramenta apresenta pouca capacidade de gerar sinais sem ruído ou mesmo sinais alinhados com os alvos. Fato este observado mesmo utilizando-se a técnica de alinhamento de sinais, baseada em algoritmo *Cross-Correlation* descrita anteriormente.

De acordo com esta métrica o sistema *NSM* não é capaz de gerar sinais de voz com altos índices de alinhamento e baixo ruído, o que torna o trabalho do algoritmo genético, na abordagem *NSM+HLSpeechGA* mais difícil. Tal problema é justificável uma vez que a rede neural presente no *NSM* não possui memória, ou seja a produção de um frame $F_{(i)}$ não é interferida pela produção de um frame $F_{(i-1)}$, daí é normal que o sinal de voz gerado por este sistema tenha baixos índices de *SNR*.

Avaliação do sinal segundo *PESQ*

Pode-se notar que em critérios de inteligibilidade o sistema *NSM+HLSpeechGA* é a abordagem que apresenta melhores resultados para voz sintética. Fato este comprovado a partir da visualização da Figura 5.5 C).

De acordo com o exposto pela métrica *PESQ*, pode-se afirmar que o sistema *NSM*, foi capaz de contribuir com o sucesso do algoritmo genético a partir da técnica *NSM+HLSpeechGA*, desta forma, os valores encontrados pela RNA foram otimizados pelo algoritmo gené-

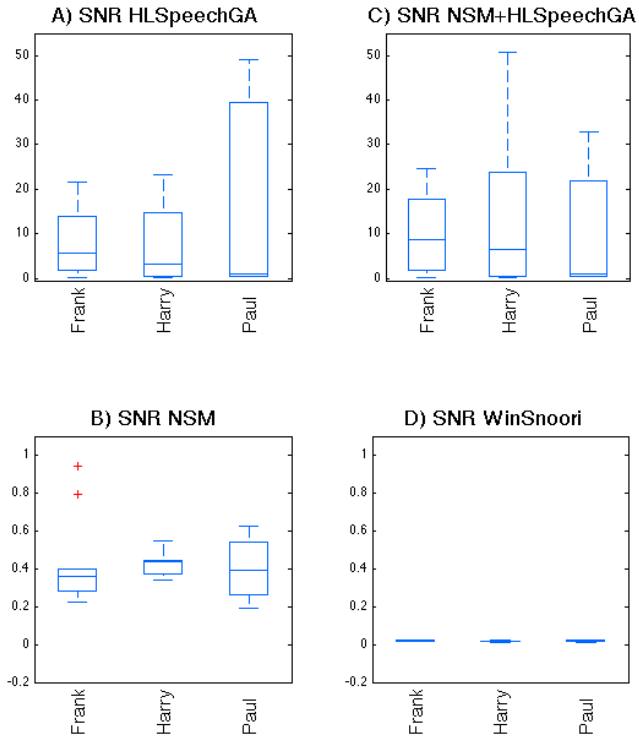

Figura 5.4: Avaliação do desempenho dos frameworks, para voz sintética, a partir da métrica *SNR*.

tico. Este comportamento pode ser explicado uma vez que ao se utilizar a abordagem *NSM+HLSpeechGA*, as populações do algoritmo genético são inicializadas a partir de intervalos de valores mais próximos da solução do problema (saída do *NSM*), se comparados com a prática de inicialização aleatória.

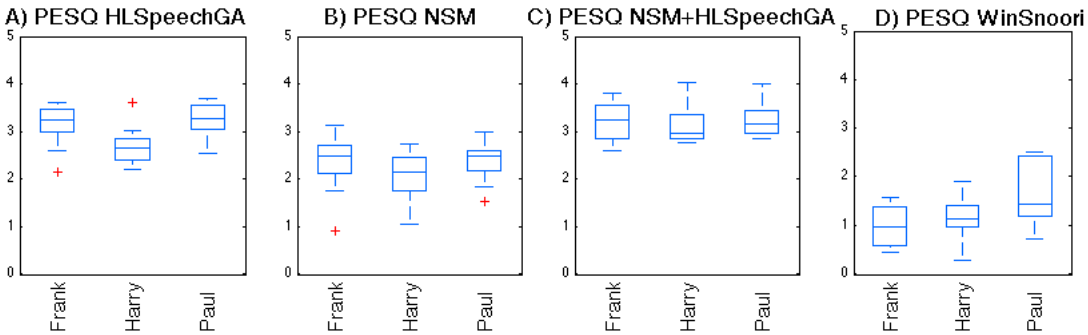

Figura 5.5: Avaliação do desempenho dos frameworks, para voz sintética, a partir da métrica *PESQ*.

Este experimento também mensura o quão voltado para voz natural o *WinSnoori* é, já que seus resultados, de acordo com as estimativas de *MOS* calculadas pelo *PESQ*, são relativamente baixos para voz sintética.

Avaliação do sinal segundo a qualidade das formantes

Uma forma minuciosa de avaliar o processo de produção de voz realizado pelos softwares aqui testados, pode ser feita a partir da avaliação objetiva da qualidade das formantes que compõem o sinal de fala produzido por cada uma ferramenta.

Esta avaliação pode ser feita, através do cálculo do *RMSE* entre as formantes originais do sinal e as formantes obtidas como cópia. Tal avaliação objetiva pode ser realizada somente para voz sintética uma vez que se possui as formantes originais que compõem o sinal de voz alvo (neste caso o sinal de voz produzido pelo *DECTalk*). A Figura 5.6 mostra tal comparação.

A comparação da formantes realizada na Figura 5.6 mostra que as formantes produzidas pela abordagem *HLSpeechGA* apresentam os mais altos erros, sendo que na abordagem *NSM+HLSpeechGA* tais erros são reduzidos, porém ainda são maiores que os erros encontrados pelos softwares *NSM* ou *WinSnoori*.

Os erros das formantes, descritos acima, podem ser justificados a partir de um conceito referente a arquitetura do sintetizador *HLSyn*, já que de acordo com [Heid e Hawkins 1998], existe a possibilidade de se encontrar diferentes combinações de parâmetros para sinais de voz semelhantes ou mesmo idênticos.

Uma vez que se tem formantes com baixos erros sendo produzidas pelo sistema *NSM*, pode-se reduzir o trabalho do algoritmo genético não otimizando estes parâmetros, e sim somente utilizando-os como parâmetros fixos no processo de síntese presente no algoritmo genético em questão.

Outro fato a se observar na Figura 5.6 A) e B) é que na abordagem onde são utilizados tanto RNA quanto Algoritmo Genético, ou seja *NSM+HLSpeechGA*, são produzidas formantes com menores erros, se comparados com os resultados da abordagem *HLSpeechGA*. Tal fato comprova o quão o sistema *NSM* melhora os resultados do algoritmo genético.

5.1.6 Experimentos com voz natural

Nas Figuras 5.7 a 5.10 são mostrados resultados da execução do A) *HLSpeechGA*, B) *NSM*, C) *NSM+HLSpeechGA* e D) *WinSnoori* mensurados a partir das métricas *RMSE*,

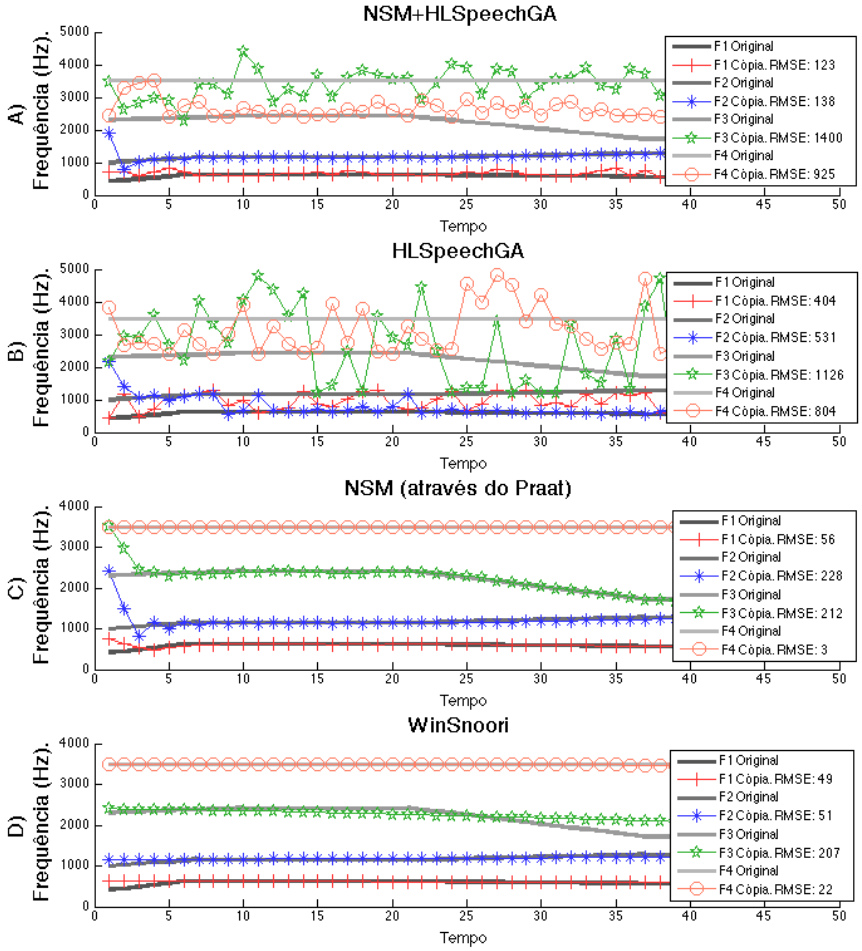

Figura 5.6: Comparação dos erros (*RMSE*) das formantes *F1* a *F4* geradas pelos frameworks. Sinal de voz da palavra *'is'*.

DE, *SNR* e *PESQ*, para a base de voz natural. Na seqüência, Figura 5.11 é avaliada a qualidade das formantes produzidas por meio de visualização de espectrograma.

Utilizar voz natural em experimentos que buscam avaliar o desempenho de sistemas como os apresentados por esta prova de conceito expõem os sistemas testados à fatores naturais da fala humana que vão desde ruídos do ambiente em que a voz foi gravada até possíveis patologias ou mesmo formas de falar únicas de um falante humano.

Quando optou-se por utilizar a base da dados do TIDIGITS como base de voz natural para os experimentos, foi objetivando minimizar tais adversidades, uma vez que esta base foi gravada em estúdio profissional. Contudo, mesmo utilizando tal base de voz, é esperado que os resultados dos experimentos com voz natural sejam mais instáveis dos que os

resultados encontrados com voz sintética.

Avaliação do sinal segundo *RMSE*

Mesmo para voz natural, o *HLSpeechGA* Figura 5.7 A) demonstrou ser capaz de produzir sinais de voz com o mínimo de erro *RMSE*. Como justificado anteriormente, tal fato ocorreu já que o algoritmo genético presente neste framework apresenta como função objetivo a função *RMSE*, calculada frame-a-frame no processo de *utterance copy* realizado pelo sistema em questão.

Um fato importante que deve ser levado em consideração na avaliação deste experimento, é que o sistema *NSM* é baseado em uma RNA que apresenta o modelo de aprendizado supervisionado, ou seja, exige dados rotulados para que seja possível realizar a generalização do problema. Desta forma, não é possível ter dados rotulados para o problema de *utterance copy* utilizando voz natural, o que limita a RNA presente no NSM a ser treinada somente com dados de voz sintética, justificando assim os altos erros *RMSE* obtidos como resultado na Figura 5.7 B).

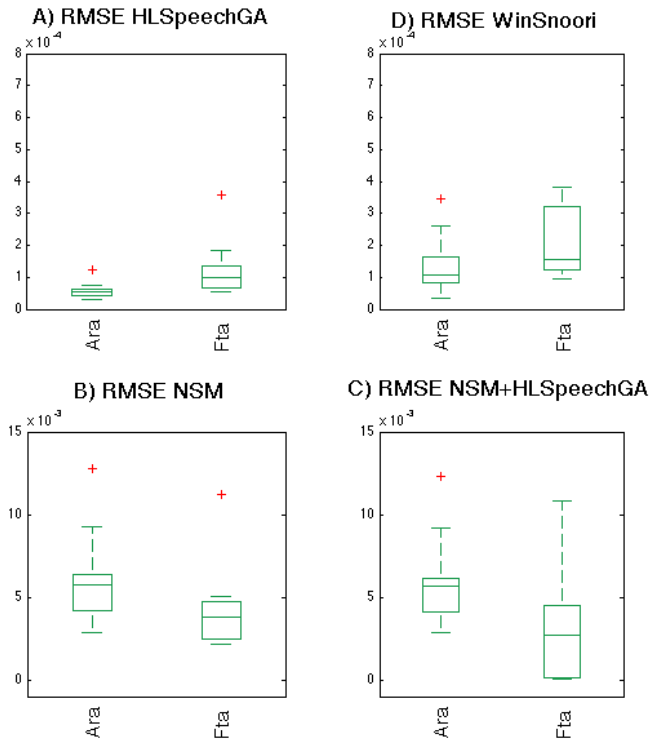

Figura 5.7: Avaliação do desempenho dos frameworks, para voz natural, a partir da métrica *RMSE*.

A hipótese científica que esta pesquisa levantou, que justificaria a possível capacidade do sistema *NSM* em realizar o processo de *utterance copy* mesmo para voz natural, está relacionada a possibilidade da RNA aprender a função f que tornaria os coeficientes extraídos da voz sintética equivalentes aos parâmetros *HLSyn*, pressupondo-se desta forma que tal função pudesse ser capaz de generalizar também voz natural. Porém, como visto na Figura 5.7, tal hipótese não se mostrou real.

Deve-se notar que os resultados obtidos pela avaliação *RMSE* às vozes geradas pelo sistema *NSM+HLSpeechGA* apresentam altos erros. Este resultado é justificável, pois uma vez que se tem o *NSM* com altos erros, tais erros são propagados para o algoritmo genético na abordagem mista. Com isso, o algoritmo genético presente na abordagem *NSM+HLSpeechGA* têm mais trabalho para convergir suas populações à solução do problema de *utterance copy.*

Avaliação do sinal segundo a Distorção Espectral

Baseando-se na métrica distorção espectral foram encontrados resultados que mostram o *HLSpeechGA* novamente com os menores erros. Tal resultado pode ser entendido como similar aos da métrica *RMSE*, uma vez que a distorção espectral utiliza o log da função *RMSE* deixando assim as diferenças dos erros mais evidentes.

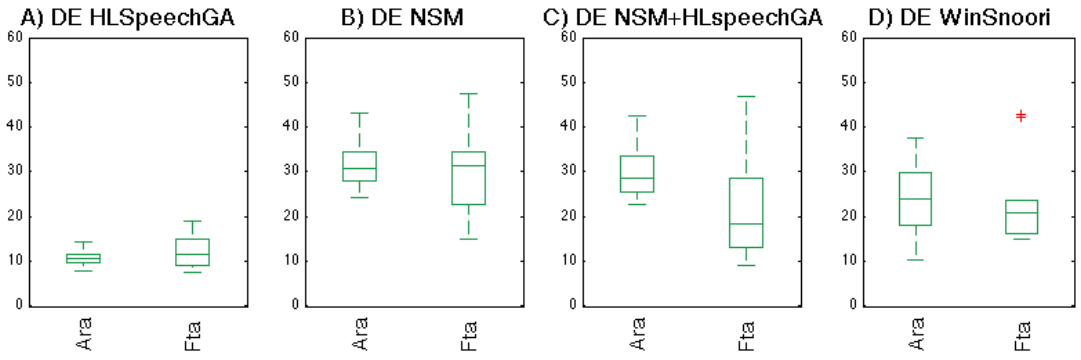

Figura 5.8: Avaliação do desempenho dos frameworks, para voz natural, a partir da métrica Distorção Espectral.

Um destaque importante com relação aos resultados deste experimento são as altas concentrações (baixa dispersão) dos *boxplots* presentes na Figura 5.8 A) B) e C) falante 'Ara', que evidencia que o falante apresenta alto índice de clareza nas pronúncias dos sinais de voz produzidos. Tal afirmação pode também ser comprovada a partir da visualização dois demais gráficos do tipo *boxplot* gerados a partir de voz natural.

Avaliação do sinal segundo *SNR*

Como visto anteriormente, altos índices de *SNR* estão ligados a ruídos e ao alinhamento entre os sinais analisados. Com isso, mesmo que todos os sistemas avaliados tenham apresentado baixos índices de *SNR*, visualizados a partir da Figura 5.9, pode-se destacar o sistema *HLSpeechGA*, que este apresenta os mais altos *SNRs*. Já que mesmo valores negativos de *SNR* são considerados em avaliações desta natureza.

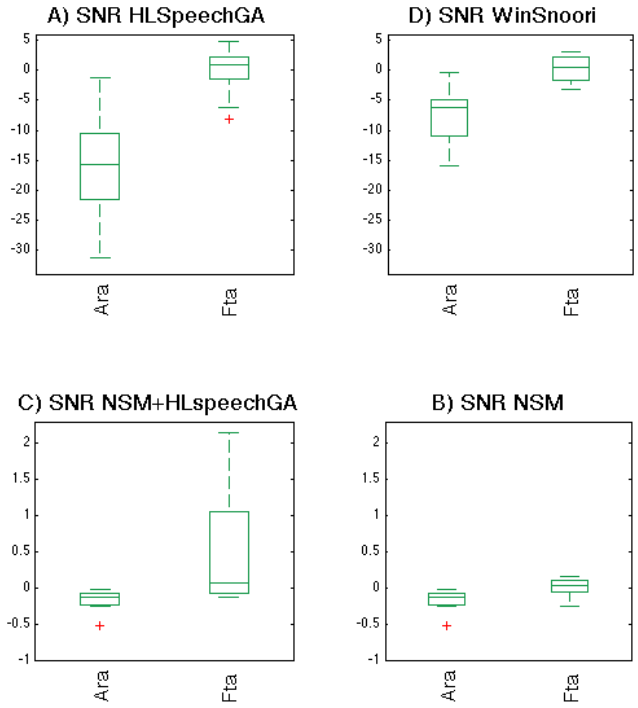

Figura 5.9: Avaliação do desempenho dos frameworks, para voz natural, a partir da métrica *SNR*.

Diferente dos resultados obtidos com voz sintética através do *WinSnoori*, este experimento evidencia que tal ferramenta apresenta melhor desempenho com voz natural, já que obteve-se índices de *SNR* mais distantes de zero.

Avaliação do sinal segundo *PESQ*

Com relação a avaliação da qualidade do sinal a partir da estimação do *MOS* calculado com base no algoritmo *PESQ* teve-se o equilíbrio considerável dentre as 4 técnicas comparadas, porém vale ressaltar que as notas na escala *MOS* estimadas são consideradas baixas. Com isso, pode-se concluir que os sistemas propostos por esta pesquisa ainda

precisam de ajustes internos para lidar de maneira melhor com as adversidades presentes em sinais de voz natural.

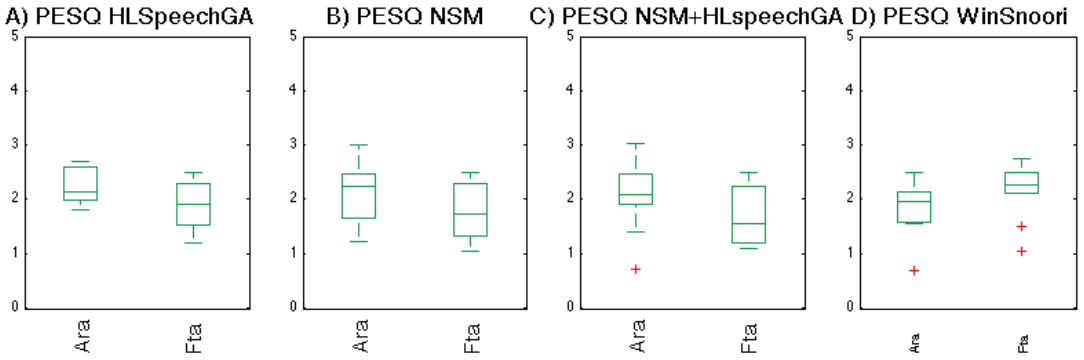

Figura 5.10: Avaliação do desempenho dos frameworks, para voz natural, a partir da métrica *PESQ*.

Apesar de ser por muito pouca diferença, neste experimento pode-se destacar o *HLSpeechGA* e o *WinSnoori* como tendo os melhores resultados, Figura 5.12 D).

Avaliação do sinal segundo a qualidade das formantes

Uma avaliação objetiva, como a realizada com voz sintética na Figura 5.6 (sessão anterior), não pode ser realizada para voz natural, uma vez que para se calcular os erros *RMSE* das formantes produzidas precisa-se das formantes do sinal original. Uma avaliação similar pode ser realizada através de inspeção (avaliação subjetiva) das formantes produzidas pelos frameworks comparadas às formantes naturais do sinal de voz alvo, Figura 5.11.

A forma subjetiva de se avaliar uma formante é qualificando esta como boa ou não, tomando como base o espectro de um sinal de voz, é feito inspecionando-se os valores da formante e os pontos com maior energia (sombreamento) do espectrograma da voz alvo. A partir desta avaliação, Figura 5.11 A) e B) pode-se observar que mesmo para voz natural o sistema *NSM+HLSpeechGA* apresenta formantes melhores do que as formantes produzidas pelo *HLSpeechGA*. Desta forma, a partir da mesma justificativa dada no experimento com voz sintética (sessão anterior), tal fato é a comprovação de que o *NSM* geram resultados que são otimizados pelo algoritmo genético presente na abordagem *NSM+HLSpeechGA*, só que agora para voz natural.

Segundo as formantes produzidas pelo sistema *WinSnoori*, que de acordo com o experimento mostrado na Figura 5.11 D) são consideradas boas, é válido justificar o baixo desempenho desta ferramenta atribuído a estimação dos demais atributos utilizados pelo

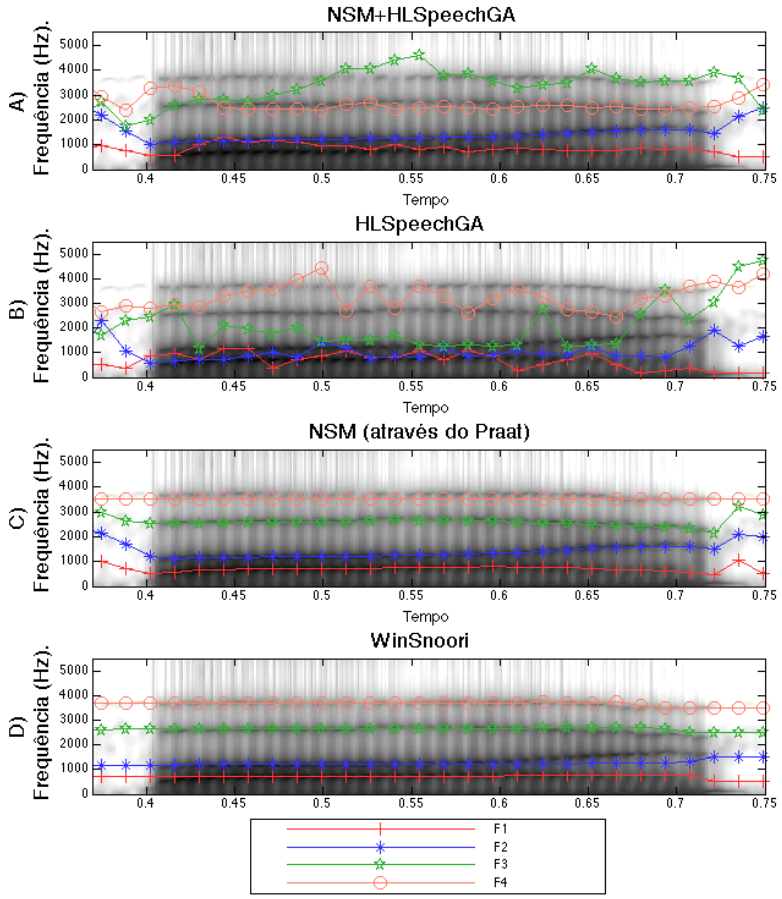

Figura 5.11: Comparação visual das formantes *F1* a *F4* geradas pelas ferramentas (linhas) com o sinal de voz natural (espectrograma da voz alvo ao fundo).

sintetizador da família *Klatt* presente em sua arquitetura, uma vez que este software possui um sintetizador com 46 parâmetros de entrada.

5.1.7 Resumo geral das avaliações com voz sintética e natural

Uma forma de confrontar os resultados obtidos a partir do processamento de voz sintética e natural pode ser observada na Figura 5.12, que mostra as médias dos resultados das avaliações realizadas a partir das quatro métricas anteriormente descritas, categorizando-as somente entre voz natural e voz sintética.

O primeiro fato a se observar nesta visualização de dados é que os valores de *RMSE*, Figura 5.12 A), são menores quando calculados em experimentos com voz natural. Este se mostra um fato curioso e que merece ser melhor explorado (a partir de novos experimentos), pois dadas as particularidades naturais destes sinais de voz, eram esperados

que seus erros fossem maiores do que os encontrados com voz sintética, ou seja eram esperados altos erros como os visualizados na 5.12 B) (com exceção do *WinSnoori*).

Os melhores resultados encontrados pelo *HLSpeechGA* de acordo com a métrica *SNR*, Figura 5.12 C), é justificada visto que os processos internos a este framework proporcionam que o sinal de voz produzido tenha alto grau de sincronia e baixo ruído com a voz alvo, fato comprovado tanto para voz natural como para voz sintética.

A avaliação por meio do *PESQ* mensura de maneira considerável a perda de precisão dos sistemas defendidos por este trabalho quando a voz em questão é natural. Porém deve-se destacar a acurácia do sistema *NSM+HLSpeechGA* que obteve a melhor média de notas para voz sintética.

5.2 Conclusão do capítulo

Este capítulo apresentou os experimentos realizados na pesquisa aqui descrita separando-os em dois grupos, voz sintética e voz natural. Pode-se notar mais padrões de comportamento relacionado ao desempenho dos frameworks, em experimentos com voz sintética e natural. De maneira geral, pode-se dizer que em experimentos utilizando-se voz sintética a abordagem *NSM+HLSpeechGA* possui melhor desempenho que os demais. Já em experimentos com voz natural a abordagem *HLSpeechGA* apresenta melhor desempenho, sendo superior em todas as métricas.

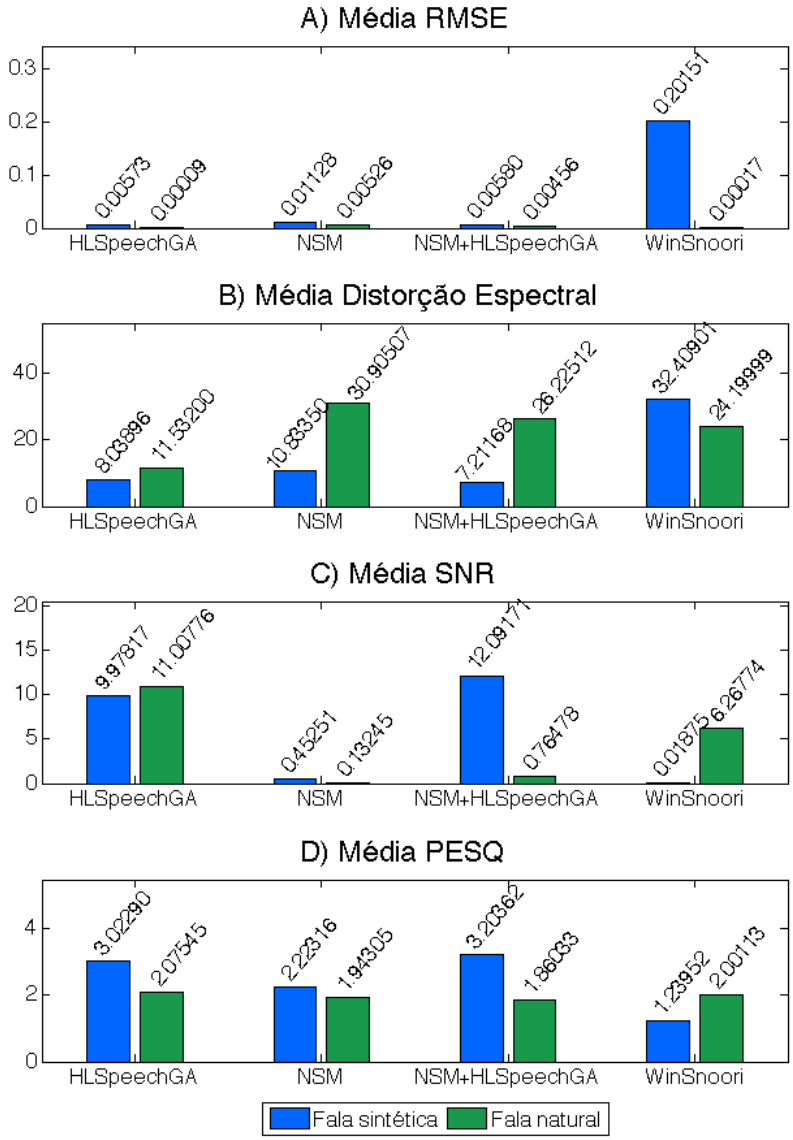

Figura 5.12: Resumo dos experimentos com voz sintética e natural.

Capítulo 6

Considerações Finais

Este capítulo apresenta as conclusões gerais entendidas a partir da pesquisa desenvolvida, juntamente com sugestões de trabalhos futuros.

6.1 Conclusões

Comprovou-se a partir de experimentos práticos que o problema de *utterance copy* é de complexidade elevada, sendo suas particularidades relacionadas à presença de memória no sintetizador, espaço de busca de tamanho considerável e alta sensibilidade dos parâmetros do sintetizador em questão.

Realizar o processo de imitação da fala com base em um sistema fundamentado em rede neural (como o *NSM*) é uma prática que se mostrou eficiente, porém limitada à fala sintética. A limitação que esta técnica apresenta, de acordo com a metodologia apresentada, está relacionada ao fato de que para se ter uma rede neural capacitada a realizar o processo de regressão entre coeficientes físicos extraídos da voz humana e os parâmetros do sintetizador *HLSyn* precisa-se de dados rotulados, isso porque o treinamento da rede é supervisionado. Com isso, para se ter configurações confiáveis dos parâmetros *HLSyn* precisou-se trabalhar com voz sintética. Em outras palavras, a rede neural só pôde ser treinada com voz sintética e não com voz natural (sendo este último algo desejável).

Utilizar um framework baseado em algoritmo de otimização, como é o caso do *HLSpeechGA*, se mostrou uma técnica capaz de gerar resultados relevantes, tanto para voz natural como para voz sintética. Visto que, a partir desta técnica foram gerados sinais de voz com os mais baixos índices de *RMSE* e Distorção Espectral, juntamente com elevados índices de *SNR* e as melhores estimações de *MOS* utilizando o *PESQ*.

A mudança do sintetizador base, da versão *newGASpeech* que utiliza sintetizador

KLSYN88 para a versão *HLSpeechGA* que utiliza *HLSyn*, se mostrou uma prática viável já que esta mudança permitiu a redução do número de parâmetros a serem extraídos pelo algoritmo genético, passando de 25 (*KLSYN88*) para 13 (*HLSyn*), inevitavelmente reduzindo o trabalho do AG.

Através da abordagem mista, utilizando tanto rede neural quanto algoritmo genético (*NSM+HLSpeechGA*) no processo de imitação da fala, pode-se afirmar que os resultados obtidos pela RNA foram otimizados com sucesso pelo algoritmo genético, porém limitados a voz sintética. Tal fato é justificado devido aos altos erros gerados pelo *NSM* ao se utilizar voz natural, que acabam sendo propagados para o algoritmo genético e com isso tem-se uma pior acurácia do sistema misto.

Verificou-se a partir de avaliações das formantes produzidas pelos sistemas propostos, que na abordagem *NSM+HLSpeechGA* não se faz necessária a extração dos parâmetros *F1* a *F4* por parte do algoritmo genético, isso porque as formantes produzidas pelo sistema *NSM* são estáveis e podem ser utilizadas (não sofrendo modificação) pelo *HLSpeechGA*. Desta forma, isso permitiria alcançar-se melhores resultados com o sistema misto e também diminuir o custo computacional já que o algoritmo genético ficaria encarregado de otimizar somente os parâmetros *AG, AB, AL, AN, UE, PS, DC, AP e F0* do sintetizador *HLSyn*.

Esta pesquisa entende como sendo a melhor proposta apresentada o *HLSpeechGA*, uma vez que a mesma produziu relevantes resultados tanto para voz natural quanto para voz sintética, sendo somente superada pelo *NSM+HLSpeechGA* nesta última.

Com relação ao *NSM+HLSpeechGA* esta pesquisa entende que ainda não foi encontrada uma maneira de utilizar Rede Neural com eficácia no problema, uma vez que este sistema se encontra limitado à imitação de somente voz sintética. Mas outras alternativas (descritas em trabalhos futuros) podem ser tentadas com o objetivo de superar tal limitação.

Esta pesquisa não alcança a solução completa do problema de *utterance copy*, mais é entendida como uma contribuição à área. Os principais pontos que retratam tal contribuição são descritos a seguir:

- Criação de uma versão alternativa à ferramenta *newGASpeech*, chamada de *HLSpeechGA*, capaz de realizar o processo de *utterance copy* com redução dos custos computacionais se comparada com o *newGASpeech*, já que esta última baseia-se em 25 parâmetros do sintetizador *KLSYN88* e a nova versão baseia-se em 13 parâmetros do sintetizador *HLSyn*;
- Desenvolvimento da ferramenta *HLSpeechGA*, capaz de realizar o processo de ex-

tração automática de parâmetros do sintetizador *HLSyn*. Algo inédito para a comunidade científica da área de voz;
- Resultados de experimentos que mensuram o desempenho da utilização de algoritmo genético aplicado ao problema de *utterance copy* para voz natural e sintética;
- Desenvolvimento do framework *NSM*, sistema baseado em rede neural capaz de extrair de maneira automática parâmetros do sintetizador *HLSyn* para voz sintética;
- Resultados de experimentos que demonstram que a utilização de rede neural no problema de *utterance copy* é uma prática viável para voz sintética. Fato comprovado a partir do comparativo entre experimentos com voz sintética e voz natural;

6.2 Trabalhos Futuros

Apesar das contribuições geradas por esta pesquisa, deve-se destacar que a metodologia aqui apresentada pode e deve ser melhorada, principalmente com relação aos fatores individuais de cada framework desenvolvido e pontos específicos da metodologia utilizada, com isso sugere-se que o(a):

- ***NSM*:** utilize de uma maior gama de tipos de coeficientes de voz, de maneira que a rede neural possa ter mais informações a respeito da voz repassada ao sistema, podendo assim ser capaz de apresentar maior capacidade de processamento da fala humana e até mesmo melhores resultados;
- ***NSM+HLSpeechGA:*** seja submetido à experimentos possibilitando que o algoritmo genético não otimize os parâmetros *F1* a *F4* extraídos pela rede neural do *NSM*, a fim de se avaliar o quão melhor pode ser a voz produzida por esta abordagem;
- ***HLSpeechGA*:** possibilite o monitoramento visual do processo de evolução do sinal de fala sintetizado geração-a-geração processada pelo algoritmo genético. Tal análise pode representar de maneira visual o processo de construção do sinal de voz, possibilitando conclusões mais específicas relacionadas ao desempenho do framework;
- **Base de dados:** submetida a testes com os frameworks *NSM*, *HLSpeechGA* e *NSM+HLSpeechGA* seja de sinais de voz longos, como frases por exemplo. A fim de se obter a avaliação dos referidos sistemas para sinais mais estáveis, ou seja, mais longos que 3 segundos;
- **Avaliação objetiva:** não apresente interferência quanto ao alinhamento entre o sinal de voz alvo e a cópia. Sendo assim, uma sugestão seria adotar a métrica *Single-*

ended [ITU-T 2004], tal algoritmo funciona similar ao *PESQ*, porém ele utiliza somente o sinal de voz cópia para estimar o *MOS*, com isso não sofre influência de falta de alinhamento entre diferentes sinais. Este algoritmo não foi utilizado como métrica nesta pesquisa devido o mesmo necessitar que os sinais de voz avaliados tenham duração acima de 3 segundos;

- **Avaliação subjetiva:** seja adotada a fim de se entender o quão é a qualidade dos sinais de voz produzidos, pelos sistemas defendidos nesta pesquisa, de acordo com a capacidade de audição humana. Tal análise é realizada por ouvintes humanos, onde os mesmos ouvem os sinais de fala e atribuem notas a estes, de acordo com as similaridades percebidas entre os sinais alvo e cópia.

Apêndice A

Ferramentas utilizadas

Este apêndice contém a listagem de todas as ferramentas utilizadas nos processos apresentados por esta dissertação, juntamente com o endereço eletrônico onde pode-se obter cada uma destas ferramentas. O objetivo desta listagem é servir de base para pesquisadores que vierem a dar continuidade a pesquisas relacionadas ao assunto abordado por esta dissertação.

Foram utilizadas as seguintes ferramentas computacionais:

- **MATLAB:** ambiente de desenvolvimento de software fundamentado em matriz disponível em: `http://www.mathworks.com`, necessário para executar o *NSM*.
- **Eclipse Java Developer:** ambiente de desenvolvimento de software java disponível em: `https://www.eclipse.org/downloads/`, utilizado para compilar o *HLSpeechGA*;
- **WinSnoori:** ferramenta capaz de realizar o processo de imitação da fala, utilizada como *baseline* desta pesquisa, disponível em: `http://www.loria.fr/~laprie/WinSnoori/`.
- **NSM:** ferramenta baseada em rede neural artificial, defendida por este trabalho. Atualmente distribuída através de solicitações aos endereços de e-mail: `laps.contato@gmail.com` ou mesmo `jose.sousa.filho@gmail.com`.
  - **SOX:** ferramenta utilizada pelo *NSM* para extração de coeficientes de voz. Atualmente disponível em: `http://sox.sourceforge.net/`, necessária à execução do *NSM*.
- **HLSpeechGA:** ferramenta baseada em algoritmo genético, defendida por este trabalho. Atualmente distribuída através de solicitações aos endereços de e-mail: `laps.contato@gmail.com` ou mesmo `jose.sousa.filho@gmail.com`.

- **Sintetizador HLSyn em Java:** este sintetizador foi reescrito de sua linguagem nativa C para a linguagem Java, sendo atualmente distribuído por meio de solicitações enviadas aos endereços de e-mail: `laps.contato@gmail.com` ou mesmo `jose.sousa.filho@gmail.com`.

Referências Bibliográficas

Anumanchipalli, Gopala K., Ying-Chang Cheng, Joseph Fernandez, Xiaohan Huang, Qi Mao e Alan W. Black (2010), KlaTTStat: Knowledge-based Statistical Parametric Speech Synthesis, *em* '7th ISCA Workshop on Speech Synthesis'.

Aylett, M. P. e J. Yamagishi (2008), Combining statistical parameteric speech synthesis and unit-selection for automatic voice cloning, *em* 'Proc. LangTech', Brisbane, Australia.

Bangayan, P., C. Long, A. A. Alwan, J. Kreiman e B. R. Gerratt (1997), 'Analysis by synthesis of pathological voices using the Klatt synthesizer', *Speech Communication* **22**(4), 343–368.
URL: *http://www.sciencedirect.com/science/article/pii/S0167639397000320*

Bishop, C. (1995), *Neural networks for pattern recognition*, Oxford Univerisy Press.

Boersma, P. (2001), 'Praat, a system for doing phonetics by computer.', *Glot International* **5**(9/10), 341–345.

Borges, J., I. Couto, F. Oliveira, T. Imbiriba e A. Klautau (2008), 'GASpeech: A framework for automatically estimating input parameters of Klatt's speech synthesizer', *Neural Networks* pp. 81–86.

Chen, S., C. F. N. Cowan e P. M. Grant (1991), 'Orthogonal least squares learning algorithm for radial function networks', *IEEE Trans. on Neural Networks* **2**(2), 8.

de Paula Bueno, Marcos Luiz (2010), Heurísticas e Algoritmos Evolucionários para Formulações Mono e Multiobjetivo do Problema do Roteamento Multicast, Tese de doutorado, Universidade Estadual de Urbelandia.

Deb, K. (2001), *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley.

Donovan, R. E. e P. C. Woodland (1995), Automatic speech synthesiser parameter estimation using HMMs, *em* 'IEEE Acoustics, Speech, and Signal Processing', Vol. 1, pp. 640–643.

Engl, Heinz Werner, Martin Hanke e Guinther Neubauer (1996), 'Regularization of inverse problems', *Mathematics and Its Applications* **375**(VIII), 322.

Faceli, Katti, Ana Carolina Lorena, João Gama e André de Carvalho (2011), *Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina*, LTC.

Figueiredo, Aline, Tales Imbiriba, Edward Bruckert e Aldebaro Klautau (2006), 'Automatically estimating the input parameters of formant-based speech synthesizers', *Proceedings of the International Joint Conference IBERAMIA/SBIA/SBRN* p. 10.

Hallahan, W. I. (1995), 'DECtalk software: Text-to-speech technology and implementation', *Digital Technical Journal* **7**(4), 5–19.

Hanson, H. M., R. S. McGowan, K. N. Stevens e R. E. Beaudoin (1999), Development of rules for controlling the hlsyn speech synthesizer, *em* 'Proceedings of the Acoustics, Speech, and Signal Processing, 1999. On 1999 IEEE International Conference - Volume 01', ICASSP '99, IEEE Computer Society, Washington, DC, USA, pp. 85–88.
URL: *http://dx.doi.org/10.1109/ICASSP.1999.758068*

Hansona, Helen M. e Kenneth N. Stevens (2002), 'A quasiarticulatory approach to controlling acoustic source parameters in a Klatt-type formant synthesizer using HLsyn', *J. Acoust. Soc. Am.* **112**, 1158–82.

Haykin, Simon (2004), *Redes Neurais: princípios e práticas*, Vol. 1 de *Inteligência Artificial*, 2ª edição, Bookman.

Heid, S. e S. Hawkins (1998), Procsy: A hybrid approach to high-quality formant synthesis using HLsyn, *em* 'Proceedings of the 3rd ESCA/COCOSDA'.

Holland, John H. (1975), *Adaptation in Natural and Artificial Systems*, Ann Arbor, University of Michigan Press.

Honda, M., T. Kaburagi e T. Okadome (1999), Speech synthesis by mimicking articulatory movements, *em* 'IEEE Systems, Man, and Cybernetics.', Vol. 2, pp. 463–468.

ITU-T (2004), 'ITU-T recommendation P.563. Single-ended method for objective speech quality assessment in narrow-band telephony applications'.

ITU-T Recommendation P.862. Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-To-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs (2001).

Kanda, H., T. Ogata, K. Komatani e H. G. Okuno (2007), Vocal imitation using physical vocal tract model, *em* 'Intelligent Robots and Systems.', pp. 1846–1851.

Kira, Kenji e Larry A. Rendell (1992), 'A practical approach to feature selection', *ML92 Proceedings of the Ninth International Workshop on Machine Learning* pp. 249–256.

Klatt, D. (1980), 'Software for a cascade / parallel formant synthesizer', *Journal of the Acoustical Society of America* **67**, 971–95.

Klatt, D. e L. Klatt (1990*a*), 'Analysis, synthesis, and perception of voice quality variations among female and male speakers', *Journal of the Acoustical Society of America* **87**, 820–57.

Klatt, D. e L. Klatt (1990*b*), 'Analysis, synthesis, and perception of voice quality variations among female and male speakers', *Journal of the Acoustical Society of America* **87**, 820–57.

Klautau, A. (2002), Classification of Peterson and Barney's vowels using Weka, Relatório técnico, UFPA, *http://www.laps.ufpa.br/aldebaro/papers*.

Kononenko, Igor, Edvard Simec e Marko Robnik-Sikonja (1997), *Overcoming the myopia of inductive learning algorithms with RELIEFF*, Vol. 7 de *Applied Intelligence*, Applied Intelligence, pp. 39–55.

Lalwani, A. L. e D. G. Childers (1991), 'A flexible formant synthesizer', *International Conference on Acoustics, Speech and Signal Processing* **2**, 777–780.

Langdon, William B. (1998), *Genetic Programming and Data Structures: Genetic Programming + Data Structures = Automatic Programming!*, Vol. 1 de *Genetic Programming*, Kluwer, Boston.

Laprie, Y. (2002), 'The WinSnoori user's manual version 1.32'. http://hal.inria.fr/inria-00107630, Visited 19-June-2014.
URL: *http://hal.inria.fr/inria-00107630*

Laprie, Y. e A. Bonneau (2002), A copy synthesis method to pilot the Klatt synthesizer., *em* J. H. L.Hansen e B. L.Pellom, eds., 'INTERSPEECH', ISCA.
URL: *http://dblp.uni-trier.de/db/conf/interspeech/interspeech2002.htmlLaprieB02*

Liu, Chang e Diane Kewley-Port (2004), 'Straight: A new speech synthsizer for vowel formant discrimination', *Acoustic Research Letters Online* pp. 31–36.

Matlab (n.d.), 'www.mathworks.com'.

Neto, N., E. Sousa, V. Macedo, A. Adami e A. Klautau (2005), 'Desenvolvimento de software livre usando reconhecimento e sintese de voz: O estado da arte para o portugues brasileiro', *6th Forum Internacional Software Livre* .

Oliveira, F., J. Trindade, J. Borges, I. Couto e A. Klautau (2011), *Multi-objective genetic algorithm to automatically estimating the input parameters of formant-based speech synthesizers*, Vol. 2, 1ª edição, Intech, capítulo 14, pp. 283–302.

Ramachandran, Ravi e Richard Mammone (1995), *Modern Methods of Speech Processing*, número pag. 26, Kluwer Academic Publishers.

Robnik-Sikonja, Marki e Igor Kononenko (2003), 'Theoretical and empirical analysis of ReliefF and RReliefF', *Machine Learning* (53), 23–69.

Rothauser, E. H., W. D. Chapman, N. Guttman, M. H. L. Hecker, K. S. Nordby, H. R. Silbiger, G. E. Urbanek e M. Weinstock (1969), *IEEE Recommended Practice for Speech Quality Measurements*, Vol. 17 de *IEEE Std*, IEEE Transactions on Audio and Electroacoustics.
URL: *http://books.google.com.mx/books?id=blu1NwAACAAJ*

Silvestre, Antônio Luís (2007), *Análise de Dados e Estatística Descritiva*, Estatística, 1ª edição, Editora Escolar.

Srinivas, N. e K. Deb (1994), 'Multiobjective optimization using nondominated sorting in genetic algorithms', *Evolutionary Computation* **2**(3), 221–248.
URL: *citeseer.ist.psu.edu/srinivas94multiobjective.html*

Stoica, Petre e Randolph Moses (2004), *Spectral Analysis of Signals*, Prentice Hall.

Tan, Zheng-Hua e Borge Lindberg (2008), *Automatic Speech Recognition on Mobile Devices and over Communication Networks*, Springer.

TIDIGITS (n.d.), 'https://catalog.ldc.upenn.edu/ldc93s10', Visitado 20-Junho-2014.

Trindade, J., F. Araujo, A. Klautau e P. Batista (2013), A genetic algorithm with look-ahead mechanism to estimate formant synthesizer input parameters., *em* 'IEEE Congress on Evolutionary Computation', Institute of Electrical and Electronics Engineers, pp. 3035–3042.

Wackerly, Dennis, William Mendenhall e Richard Scheaffer (2008), *Mathematical Statistics with Applications*, número 9780495110811, Cengage Learning.

Young, S. et. al (2000), *The HTK Book*, Microsoft Corporation, Version 3.0.