UNIVERSIDADE FEDERAL DE PERNAMBUCO
Graduação em Engenharia da Computação
Centro de Informática
2011.2

Análise e Remoção de Ruídos em Sinal de Voz através de Técnica de Atenuação de Espectro

Lauro Gonçalves da Rocha

Recife, Dezembro de 2011

Universidade Federal de Pernambuco
Centro de Informática

Lauro Gonçalves da Rocha

Análise e Remoção de Ruídos em Sinal de Voz através de Técnica de Atenuação de Espectro

Agradecimentos

Gostaria de agradecer a toda minha família, especialmente minha mãe Diva, meu pai Flávio, minha irmã Marina, minha avó Divanice e minha Tia Adélia.

Também agradeço a minha namorada Paula, que esteve comigo em toda minha graduação, e a sua família.

Aos amigos do “teia-nela”, “éramos 25” e do colégio, que ajudaram, aguentaram e comemoraram sempre, todos juntos. Agradecer aos amigos Erasmus, que me ajudaram e compartilharam muitas coisas na França, fazendo minha a estadia fora do país ainda melhor.

Aos professores, ao meu orientador Tsang Ing Ren e ao Centro de Informática, pela possibilidade do aprendizado e oportunidade de crescimento.

A pessimist sees the difficulty in every opportunity; an optimist sees the opportunity in every difficulty.

Winston Churchill

Resumo

Sistemas que utilizam sinal de áudio estão cada vez mais em evidência, seja para aplicações de *computer telephony*, como Skype, Google Voice, iChat, gtalk, ou seja para *speech-assited human-computer interface*, em que ordens são dadas através de comandos pré-estabelecidos, seja para programas de gravação e edição, como Garage Band e o Audacity.

Essas aplicações têm em comum a utilização do sinal de áudio como ferramenta principal. O sinal de áudio sofre um processo de conversão A/D, e a qualidade da entrada dependerá da quantidade de bits do conversor, ou seja, quantos níveis ele irá conseguir representar discretamente.

Depois dessa conversão, com o sinal já digitalizado, existe ainda a possibilidade do sinal possuir ruídos, devido ao ambiente em que o som é captado. Isso é muito comum em toda a abordagem de processamento de sinal, visto que um ambiente real possui diversas variáveis.

Atualmente, existem diversas técnicas para o tratamento do sinal de áudio para remoção de ruídos, principalmente na área de sinal de voz. Neste trabalho, será abordada a técnica de *short-time spectrum attenuation techiniques*, um método de filtragem baseado em dois estágios de filtro de Wiener. Além disso, o espectro é analisado através de uma transformada chamada MCLT e o ruído é classificado em duas etapas. Resultados serão analisados e sugestões de possíveis modificações na pesquisa serão mostradas.

Palavras-chave: Filtro de Wiener, Processamento de Voz, Ruído, Remoção de Ruído

Abstract

Systems that use audio signals are in evidence, especially computer telephony applications, such as Skype, Google Voice, iChat, gtalk etc., speech-assisted human-computer interface, in which orders are given by pre-established command, and recording and editing programs like Garage Band or Audacity.

These applications have in common the use of the audio signal as the main tool. The audio signal undergoes a process of A / D conversion, and the quality of entry will depend on the number of bits of the converter, i.e., how many levels it will be able to represent discreetly.

After this conversion, the signal is digitized, and it is possible to have some noise due to the environment in which sound is recorded. This is very common in all signal processing approaches, since a real environment has many variables.

Currently, there are several techniques for audio signal processing for noise removal, especially in the area of voice signal. This paper will look at the technique of short-time spectrum attenuation, a filtering method based on two stages of Wiener filter. Moreover, the spectrum is analyzed using the MCLT, and the noise is classified into two steps. Results will be analyzed and suggestions for possible changes are shown.

Keywords: Wiener Filter, Voice Processing, Noise, Noise Reduction.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Equações

Capítulo 1

Introdução

Sabemos que ao longo dos anos, as evoluções no campo computacional foram extremamente rápidas e tendem a continuar crescendo, já que a demanda de novos *softwares* e *hardwares* está em alta. Cada vez mais aplicações ligadas a processamento de sinal estão sendo utilizadas, seja processamento de áudio, vídeo ou imagem.

1.1 Motivação

Com o grande crescimento de aplicações que utilizam processamento de voz, principalmente aplicações de VoIP (*computer telephony*), existe uma constante necessidade de se melhorar este tipo de serviço, já que ele está cada vez mais está substituindo telefonia comum. Uma operadora geralmente controla o serviço de VoIP e sua rede apresenta uma certa qualidade, baseada no tráfico de dados. Além disso, como sabemos, e foi descrito pela anatel:

"VoIP é um conjunto de tecnologias, largamente utilizadas em redes IP, Internet ou Intranet, com o objetivo de realizar comunicação de voz."

Se formos analisar desde o princípio, o procedimento consiste em digitalizar a voz em pacotes de dados para que estes trafeguem pela rede IP e sejam convertidos em voz novamente em seu destino. Como o sinal de voz será capturado em um ambiente real, está sujeito a qualquer interferência externa e esse ruído indesejado prejudica a qualidade da comunicação entre os dois pontos envolvidos.

1.2 Objetivos

O objetivo geral deste projeto é a remoção de ruídos em sinais de voz, usados em distintas aplicações, entre elas e como citado, VoIP.

Com isto, teremos uma melhora no serviço prestado pelas operadoras de *computer telephony*, já que é possível integrar o sistema de remoção de ruídos à aplicação que conecta dois usuários.

Para efetuar o proposto, os seguintes passos estarão presentes:

- Estudo e teoria do sinal de voz;
- Estudo e teoria de análise espectral;
- Estudo dos filtros de Wiener;
- Implementação do Sistema, como um todo;
- Experimentos e Resultados analisados.

1.3 Estrutura do Documento

Este trabalho de graduação está dividido em 7 capítulos. No segundo capítulo, será mostrada uma introdução ao sinal de áudio, seu histórico, sua obtenção no meio e algumas tecnologias que utilizam este tipo de sinal para diversas aplicações. No terceiro capítulo, serão analisados alguns sistemas de redução de ruído em sinais de voz, aplicados atualmente. Um dos sistemas presentes neste capítulo será o objeto principal do estudo deste documento. No quarto capítulo, será abordado mais detalhadamente o filtro de Wiener, módulo imprescindível no processo de remoção de ruído aditivo e que está presente no sistema desenvolvido e detalhado no capítulo subsequente. No quinto capítulo será mostrado detalhadamente todo o sistema implementado, com descrições e explicações de todos os módulos propostos. O sexto capítulo aborda experimentos variados executados no sistema e os seus resultados. Parâmetros e seus papéis no programa e nos resultados também terão destaque. Por fim, conclusões serão levantadas acerca do resultado final e futuras abordagens e mudanças serão citadas e sugeridas.

Capítulo 2

Sinal de Áudio

Este capítulo tem como intuito explorar os sinais de áudio, em especial os sinais de voz. Um breve histórico será mostrado, além de como o sinal é obtido e de algumas áreas atuais de pesquisa.

2.1 Histórico

A história do sinal de áudio remete ao século XIX, pois só a partir daí foram desenvolvidos os primeiros equipamentos de gravação na prática. Isso ocorreu por volta de 1877, com o fonógrafo cilíndrico criado por Thomas Edison, e patenteado em 1878. Em um curto espaço de tempo, um novo negócio foi gerado e expandido: a gravação e reprodução de áudio. Já no início do século XX, a gravação e venda de discos chegava à casa dos milhões e um negócio praticamente inexistente se transformou em uma grande fonte monetária.

O próximo passo na indústria do áudio foi a invenção do gramofone de disco. A produção dos discos era barata e de boa qualidade, além de fácil transporte. Os vinis vieram em seguida e gravações foram feitas na década de 40, melhorando qualidade de reprodução e gravação.

Gravações eram baseadas em processos mecânicos até o desenvolvimento da eletrônica, que proporcionou um grande impulso às gravações de áudio e estudo da área de sinais de áudio. Microfones, amplificadores e caixas de som, além de diversos dispositivos para manipulação do sinal foram desenvolvidas ao longo do tempo. Juntamente com esses desenvolvimentos, a guitarra elétrica foi criada, que nada mais é que a utilização desses novos dispositivos eletrônicos para gerar sinais de áudio.

Outro ponto crucial na parte da aquisição do sinal de áudio foi a criação da gravação digital de som. Com ela, foram possíveis diferentes análises e novos tipos de manipulação do sinal, além do desenvolvimento de novas mídias.

Os passos de gravação e reprodução de uma maneira simplificada são demonstrados:

Gravação

- O sinal de áudio analógico é transmitido e um conversor A/D passa o sinal para níveis discretos, com valores variando de acordo com uma quantidade de bits.
- Os dados são coletados e estimados em uma determinada frequência, chamada de *sample rate*.

Reprodução

- Os dados digitalizados passam para um conversor D/A, que converte o sinal digital em analógico.
- O sinal é transmitido e amplificado.

2.2 Obtenção do Sinal de Áudio

Como citado acima, o sinal de áudio, para ser obtido (gravado), passa por todo um processo de conversão. Para isso, é utilizado em geral um dispositivo eletrônico que converte uma entrada de voltagem analógica em um número proporcional a esta entrada.

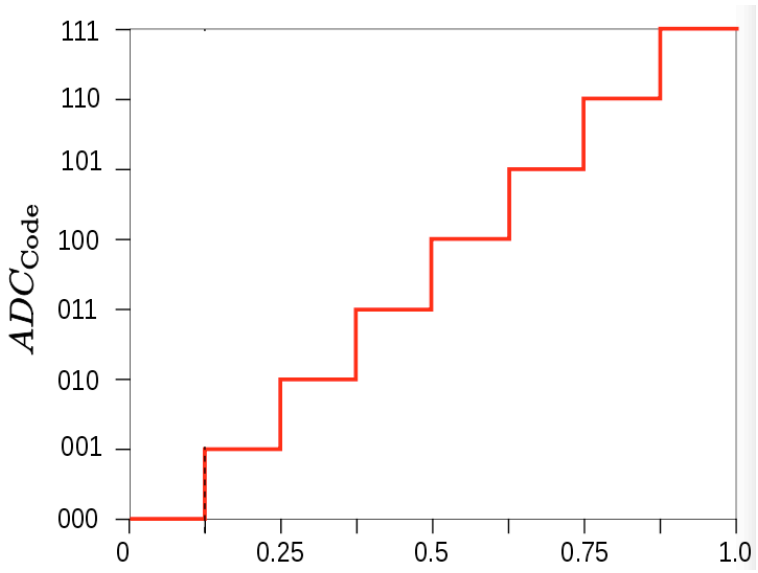


Figura 1 Codificação de um conversor AD de 8 bits (Analog to Digital Converter,2011)

Conversores analógicos digitais armazenam eletronicamente as informações em binário. A precisão do conversor geralmente é expresso em bits. Esse número representa os valores discretos os quais os valores analógicos podem receber, também chamados de *levels*, na potência de dois. Se temos por exemplo um conversor com 8 bits de precisão, teremos 256 possíveis valores (2^8 = 256). Os valores podem ser representados de 0 a 255 ou de -127 a 128.

Podemos então definir N como o número de intervalos de voltagem presentes no conversor:

N = 2 ^ M, onde M representa a precisão do dispositivo.

Conversores deste tipo estão sempre presentes na obtenção de sinal de áudio e sua precisão deve ser verificada de acordo com a aplicação desejada (Walden, 1999).

2.3 Pesquisas com sinais de áudio - tecnologias de Voz

Depois de analisar a história e sua obtenção, iremos abordar alguns campos de pesquisa de sinais de áudio. Ele é bastante extenso e possui várias áreas. Podemos citar por exemplo, Computação Musical, área que trata problemas musicas, lida e investiga métodos, técnicas e algoritmos para processamento do sinal. Além disso, existem sistemas de software que lidam com sinais de áudio apenas para manipular suas características, como em casos de softwares musicais para shows ou gravações. Neste último caso, existe opções para se tratar, por exemplo, só o sinal de voz, com o tão conhecido *autotune* (Antares Auto-Tune, 2011), ferramenta que utiliza uma matriz sonora para corrigir performances vocais. Além desse tipo de processamento de voz, existem outras aplicações com diversas funcionalidades que serão detalhadas abaixo.

2.3.1 Sistema para Reconhecimento de Emoção em Sinal de Voz

Sabemos que sinais de voz possuem palavras e significados, mas também emoções em seu contexto. Além de expressões, quando se fala é possível identificar diversos sinais de emoções, na maneira de falar, no espaçamento entre palavras, etc.. Este tipo de informação que não é diretamente passada para o ouvinte é chamada de paralinguagem (Tawari, Trivedi, 2010). Para qualquer humano é clara a diferença entre um diálogo com emoção e sem emoção, mas para uma máquina este processo de diferenciação não é trivial.

Como mostrado acima, podemos verificar que a emoção em um sinal de voz é algo bastante significativo e que contém informações valiosas para diversas aplicações que necessitam receber o *feedback* das pessoas que estão falando. Apesar desta importância, a grande maioria das interações usuário-máquina simplesmente ignoram este conteúdo tão relevante. Sistemas do futuro tendem a cada vez mais a adquirir essas informações, processá-las, e adicioná-las à solução desejada.

Numa abordagem mais prática, a classificação (reconhecimento) seria baseada no número de emoções permitidas, porém, atualmente, as técnicas visam reconhecimento das emoções conhecidas como básicas, por exemplo: emoção positiva em relação a um fato, uma pessoa neutra e outra pessoa com emoção negativa em relação a determinada coisa. A maioria das abordagens são treinadas e testadas a

partir de dados de atores, ou seja, eles simulam uma determinada emoção para treinar o modelo, já que um banco de dados com emoções reais gravadas é muito complicado. Abaixo, é mostrado um diagrama de blocos presente em (Tawari, Trivedi, 2010):

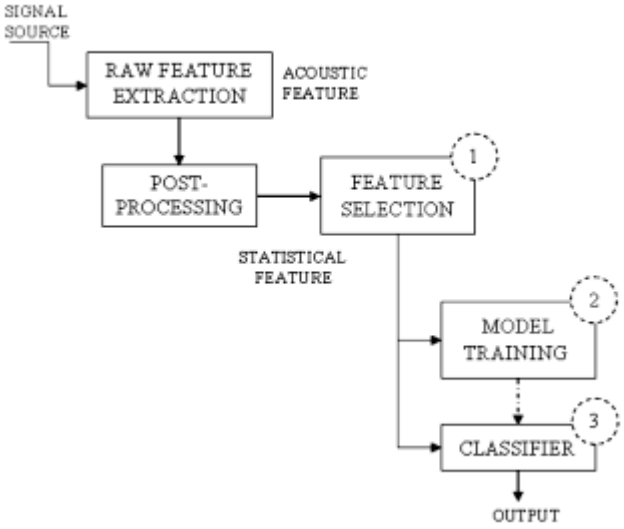


Figura 2 Diagrama de Blocos - Sistema de Reconhecimento de Emoção (Tawari, Trevedi, 2010)

Como é mostrado, o diagrama de blocos relativo a parte de identificação de emoção possui 3 fases. A primeira, que faz extração e seleção, a segunda, que é o *model training phase*, e a terceira, que serve para avaliar o desempenho do sistema.

É importante notar que cada língua possui suas particularidades, ou seja, a expressão de alguma emoção em uma fala em português, é diferente de uma fala em alemão, que já possui um bom banco de dados. Então o modelo em português necessitaria de outra base de dados para ser treinada, já que o modelo alemão é diferente.

2.3.2 Sistema para Reconhecimento de Voz (Locutor)

Aplicações de reconhecimento de voz, sejam aplicadas à verificação de locutor, como medida de segurança adicional, sejam como, por exemplo, STT, que podem ser aplicadas desde *smartphones*, passando aplicações de comunicação pela internet ou mesmo para simplificar transcrição de textos. Essas aplicações começaram a ser

estudas e técnicas desenvolvidas a partir dos anos 50, quando análise espectral através de transformadas foram feitas e o reconhecimento baseado na potência do sinal. Este foi apenas o início de uma série de algoritmos aplicados ao sinal de voz para classificação. O algoritmo K-NN, largamente utilizado atualmente, foi proposto nesta época, mas somente expandido mais recentemente. Outra técnica muito conhecida na área é o LPC, que é simples em relação a parâmetros de voz e técnicas para se analisar. O LPC foi o início e este extrator de características se desenvolveu e deu origem a muitas outras técnicas, tais como MFCC e LPCC. Depois deste paradigma, foi introduzido um modelo chamado HMM, ou *Hide Markov Model* para reconhecimento de voz. Atualmente, o GMM e o FGMM são muito usados, além claro, de RNA's.

2.3.3 Sistema para Detecção de Atividade de Voz

Em muitas aplicações é de extrema importância saber se uma pessoa está, falando ou não em um determinado momento, seja para economizar processamento, desativando um processo enquanto não houver discurso, seja para filtrar/codificar apenas uma parte da mensagem salvando banda em alguns tipos de aplicação, como VoIP por exemplo (Ramirez, Gorriz, Segura, 2007). Uma das técnicas de pré-processamento chamada VAD - *Voice activity detection* ou *speech active detection* faz exatamente o citado: detecta a presença ou ausência de um discurso. Como se nota, é necessário que se tenha uma boa solução para VAD, já que é extremamente prático em sistemas de tempo real.

Algoritmos de VAD possuem passos em comum que são implementados de diversas formas, porém, de uma maneira geral, o sinal inicialmente passa por um estágio de redução de ruído, no qual fica mais fácil a detecção ou não do sinal de voz. Em seguida, são calculados a partir do sinal de entrada valores que servirão de base para uma terceira etapa, que é uma classificação através de um *threshold,* que irá dizer se aquilo caracteriza ou não um sinal de voz (Ramirez, Gorriz, Segura, 2007).

Além das aplicações já citadas, podemos notar que o VAD se aplica em campos muito mais amplos, tais como: conferências de áudio e vídeo, supressão de ruídos fora do intervalo de fala, reconhecimento de voz, além de, claro, telefonia convencional.

Capítulo 3
Sistemas de Redução de Ruído em Sinais de Voz

Ruídos sempre estarão presentes em sinais, sejam eles de áudio, vídeo ou imagem. Neste capítulo, descrição e objetivos sistemas de redução de ruídos serão mostrados e soluções atuais serão abordadas.

3.1 Descrição e Objetivos

Como citado acima, ruídos são inerentes a sinais, podendo estar presentes em maior ou menor escala. Na maioria dos casos, é desejado ter sinais sem interferências para um determinado processamento. Esta etapa, de pré-processamento, que possui o objetivo de eliminação de ruído, é feita pois dispositivos de gravação de todos os tipos possuem susceptibilidades. Existem diversos tipos de ruído, seja ele aleatório, como por exemplo, *white noise*, periódicos ou de outros tipos.

O objetivo desta etapa é reduzir interferências externas. Para que isso seja feito, algumas técnicas utilizadas atualmente são mostradas.

3.2 Sistema Baseado na remoção de ruído do sistema auditivo humano

Uma solução atual, baseada no sistema auditivo humano é abordada em (Virag, 1999). Este sistema é efetivo para alguns tipos de *white noise*. A proposta é implementar computacionalmente o modelo de audição humana baseado em um fenômeno denominado *masking phenomenum*. Este conceito é relacionado com *critical band analysis*, mecanismo central do ouvido. As *masking properties* são modeladas calculando o *threshold* do ruído. Um humano tolera ruído aditivo até o ruído estar abaixo do limite estipulado. No processo de otimização, como o sistema é baseado em um sistema subtrativo, os parâmetros são adaptados a partir do *noise masking threshold*. No final, o objetivo deste algoritmo é explorar as *masking properties* do sistema auditivo humano para superar as limitações do melhoramento de um canal do tipo subtrativo, com ruído aditivo com SNR's baixo (10dB).

É possível notar que este tipo de sistema é bem específico, ou seja, remove ruídos com caracterísitcas bem detalhadas, não sendo útil para qualquer aplicação.

3.3 Sistema baseado na supressão de ruído audível

Este método visto em (Virag, 1999) é baseado na supressão de ruído audível. Nele é apresentada uma técnica de supressão de ruído baseado em definição do que é chamado *psychoacoustically derived quantity* do ruído do espectro audível e sua eliminação usando um filtro não linear ótimo do *short-time spectral amplitude (STSA) envelope***.** O filtro utiliza a *sparse spectral estimates* obtida do STSA e quando se definem os parâmetros de uma forma satisfatória, ganhos de até 40% são obtidos comparados com o sinal de entrada. Esses parâmetros também podem ser estimados baseados no dado ruidoso, resultado em ganhos pequenos, porém significantes.

Para que esse modelo fique pronto, são necessários alguns passos. Um modelo *psychoacoustically* para discurso precisa ser definido. Inicialmente, *Definitions of the Perceptually Significant Spectra* são feitos, para encontrar componentes espectrais que contribuem para ruído. Em seguida, existe um passo que é um critério psico-acústico para remoção de ruído. Parâmetros são definidos para fazer modificações ótimas no discurso. Uma análise no *Parameter Error Analysis and Sensitivity* é feita, pois alguns parâmetros são determinantes para uma boa redução de ruído e caso eles estejam com grande erro, o sistema pode não funcionar corretamente. *Psychoacoustic Speech Enhancement and Reconstruction Based on Sparse Speech Data* otimiza o discurso com algo chamado *subband regions*. Esses são todos os passos deste método e de uma maneira geral, podem obter excelentes resultados como citado no início (até 40%).

3.4 Sistema baseado na atenuação de espectro

Inicialmente, temos que definir este tipo de sistema, do tipo de atenuação de espectro. Este tipo de técnica é comprovadamente uma maneira efetiva e de relativo

baixo custo para se remover ruídos em sinais. Numa abordagem convencional e sabendo que o ruído que iremos tratar é aditivo, ou seja, em um determinado ambiente, um outro sinal não desejado se mistura ao original, e há dois passos considerados básicos: o primeiro, é estimar o espectro do ruído e o segundo é filtrar o sinal, para obtê-lo sem distorções. No sistema de subtração espectral, a magnitude do espectro do ruído é estimado e então é subtraído da magnitude do sinal original, com isso, reduz-se significativamente o ruído.

Existem diversos sistemas de atenuação de espectro, porém, o objetivo deste documento é abordar o modelo (Jiang, Malvar, 2000).

O modelo proposto para o sistema é mostrado abaixo:

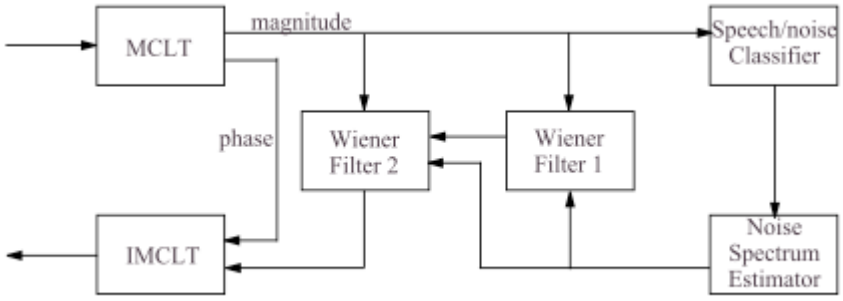


Figura 3 Sistema de Remoção de Ruído Utilizando Atenuação de Espectro (Jiang, Malvar, 2000)

Inicialmente, a análise espectral é feita através da transformada MCLT, que pode ser vista em (Malvar, 1999) e (Malvar, 1991). Este tipo de transformada é baseada em outra transformada, chamada de MLT, e sua extensão possui algumas propriedades que a original não possui. Além disso, é mostrado também em (Malvar, 1999) que este tipo de transformada pode substituir *DFT Filter Bank*, além de ser boa para aplicações de redução de ruídos, como esta, e supressão de eco, além de possuir uma implementação rápida que pode ser usada em aplicações de tempo real.

Após a MCLT, e como citado acima, iremos apenas utilizar a magnitude do sinal de saída, já que está sendo utilizado um sistema de atenuação de espectro. A magnitude então irá passar por uma classificação chamada de *context adaptative*, na qual inicialmente é calculada a energia do sinal. Em seguida, é necessário achar um

threshold inicial para se utilizar na classificação, que possui duas fases, chamadas de *hard decision* e *soft decision*.

O ruído é então classificado e os filtros de Wiener são utilizados para minimizar o erro quadrático médio, e consequentemente diminuindo o ruído no sinal.

Essa abordagem apresenta bons resultados além de ser uma maneira relativamente simples de se obter sinais com menos ruídos.

Capítulo 4

Filtros de Wiener

4.1 Histórico e Descrição

Durante a segunda guerra, muitos esforços e pesquisas em diversas áreas estavam sendo realizados para que técnicas fossem obtidas para que pudessem ser usadas na guerra e para trazer obviamente, vantagens estratégicas. Uma dessas pesquisas estava sendo realizada pelo matemático Norbert Wiener. Ele estudava a possibilidade de um sistema de mira e tiro automático para abater aviões inimigos. Isso fez com que Wiener começasse a estudar um campo nunca antes explorado por ele: teoria da informação. Como resultado desse esforço, foi desenvolvido o filtro de Wiener, também chamado de Wiener-Kolgomorov, já que na mesma época o matemático russo desenvolveu em paralelo a mesma teoria.

De uma maneira geral, o filtro de Wiener reduz o ruído presente em um sinal comparando com uma estimativa de um sinal sem ruído. É importante ressaltar que o filtro de Wiener não é um filtro adaptativo, isso porque ele assume que suas entradas são estacionárias. Ele é um filtro estatísco, primeiro de muitos, baseado em MSE.

Abaixo segue um pequeno esquema dos filtros de Wiener:

- Hipótese: o sinal e o ruído aditivo são processos lineares estacionários e estocásticos com características espectrais conhecidas ou mesmo com correlação ou correlação cruzada;

- Requisitos: o filtro precisa ser realizável/causal;

- Critério de Desempenho: baseado em MMSE *minimum mean-square error.*

Em estatísitca, o MSE (*mean squarred error*) de um estimador é uma das maneiras de quantificar a diferença entre valores. MSE é também considerada uma função de risco, correspondendo ao valor esperado do *squared error loss* ou *quadratic loss***,** ou seja, o MSE faz a medida da média do quadrado dos erros. O erro é o valor implícito pelo estimador menos a quantidade a ser estimada.

4.2 Analisando o filtro de Wiener

Analisando o Filtro de Wiener, estarão presentes algumas varáveis, tais como:

- *s(t)* sinal original, o objetivo da filtragem;
- *n(t)* é o ruído;
- ^*s(t)* sinal estimado (intenção de ser igual a s(t));
- *g(t)* é a resposta do filtro ao impulso.

Em seguida, o resultado do filtro de Wiener será:

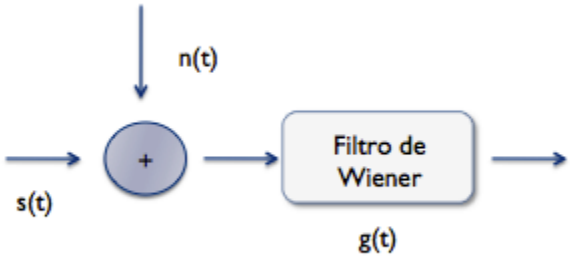


Figura 4 Esquema do Filtro de Wiener

$$\hat{s}(t) = g(t) * [s(t) + n(t)]$$

Equação 1 Saída do Filtro de Wiener

O erro associado ao processo é definido como:

$$e(t) = s(t + \alpha) - \hat{s}(t)$$

Equação 2 Erro associado ao filtro de Wiener

onde,
s(t + a) é a saída desejada do filtro
e(t) é o erro

O filtro de Wiener é provado e descrito de maneira continua, através de integrais, entretanto, para implementações de trabalhos como esse, que manipulam sinais discretos, é interessante abordar a forma discreta do mesmo.

4.3 Aplicações

Além de ser extensamente conhecido por remoção de ruídos, seja em sinais de áudio, como discursos (Jiang, Malvar 2000), músicas, gravações, seja em imagens, como fotos convencionais (Image Deblurring, 2011), imagens biomédicas (Biomedical Images Analysis, 2011), ou vídeos (Mina, Raghunandan, 2008), os filtros de Wiener apresentam uma grande versatilidade de possíveis aplicações. Claro que os outros problemas são reduzidos para que se possa aplicar esta técnica já que ela é de fácil implementação e baixo custo computacional.

Uma abordagem interessante é vista em (Del Negro, 2006), onde se é feito algo chamado *magnetic profiling* em um ambiente vulcânico, na Itália. Ele consegue distinguir o ruído e o sinal geológico desejado, sendo assim possível uma análise mais precisa.

Uma aplicação extremamente interessante, mas distinta da abordagem convencional é a mostrada em (Bhansali, 1982), onde se é proposta uma análise com o filtro de Wiener para uma questão econômica. A questão é baseada na relação dos preços de produtos argentinos com o suprimento de dinheiro fornecido pelo país. O trabalho ilustra uma metodologia que aplica o filtro de Wiener para estimar *distributed lag relationship to practical time series* ou *finite time series*.

Capítulo 5

Sistema Desenvolvido

Este capítulo tem como intuito explicar em nível detalhado toda a implementação do sistema desenvolvido para o trabalho de graduação. Desde pequenos detalhes, passando por todas as ideias, além de explicações mais técnicas sobre a linguagem utilizada, bibliotecas externas, técnicas e todos os passos envolvidos para a construção da aplicação.

5.1 Descrição

O sistema proposto é a implementação mostrada em (Jiang, Malvar, 2000), com algumas modificações. Inicialmente, foi necessário especificar algumas definições para o começo do projeto. A primeira questão a ser pensada foi a linguagem de programação a ser utilizada. Foi necessário escolher entre três linguagens de programação levantadas:

- C
- C++
- Matlab

A dúvida sobre esta questão foi baseada em performance, portabilidade, e facilidade de implementação. A primeira opção foi utilizar C, linguagem bastante flexível, que possui biblioteca para interface externa com áudio, chamada de *portaudio* (Bencina, Burk, 2001) para tempo real, e com ela é possível manipular arquivos de áudio, além de ser extremamente rápida e uma implementação para sistema de tempo real ou mesmo para desktop é possível. C++ teria a mesma facilidade de C, porém uma variável até então não tinha sido considerada: tempo de implementação. A ideia era a possibilidade de implantar isso em uma aplicação real, porém, com o tempo curto, verificou-se a isso não era viável, tanto para C quanto para C++. MATLAB então foi a escolha mais acertada. Alguns pontos provam isso, tais como:

- Fácil manipulação de arquivos de áudio
- Bibliotecas de processamento de áudio
- Manipulação trivial de matrizes e vetores
- Funções necessárias já implementadas em MATLAB
- Opção de gráficos e comparações

Diferentemente de C e C++, MATLAB possui nativamente opção de abertura de arquivos de áudio. Isso pode parecer trivial, mas em C e C++ esse tipo de implementação não é fácil, nem mesmo achar isso em bibliotecas externas. Como MATLAB já é uma ferramenta largamente utilizada em aplicações de processamento de sinal, foi possível encontrar boas bibliotecas (além das próprias funções nativas) de processamento de áudio. Com elas, foi possível efetuar operações que teriam que ser feitas em C/C++ e que estão fora do contexto da aplicação, por exemplo. A análise de resultado também ficou muito mais simples através das opções gráficas e de análise.

Passada a fase inicial de escolha de linguagem, foi necessário o entendimento do sistema para decidir o que seria feito. A imagem abaixo mostra o diagrama de blocos do sistema proposto:

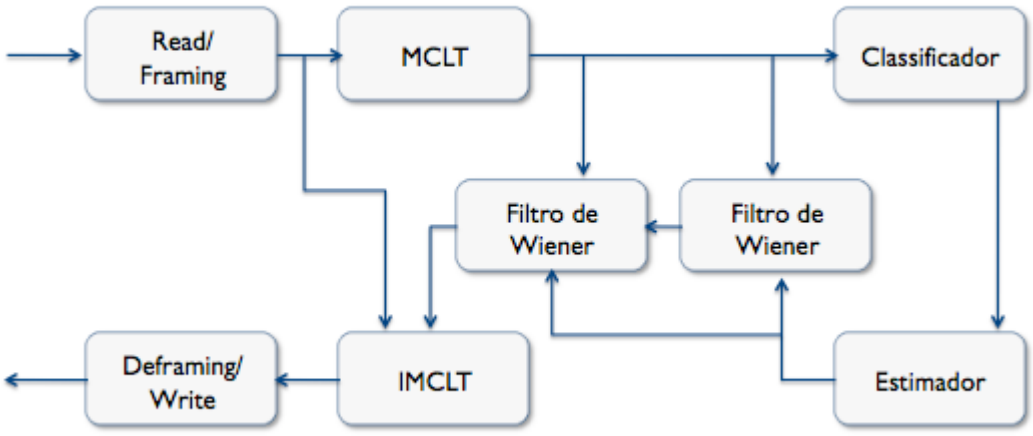


Figura 5 Sistema Proposto

Cada módulo mostrado no diagrama de blocos será detalhado individualmente para um melhor entendimento.

5.2 Dividindo o Sinal em Frames

Figura 6 Dividindo o Sinal em Frames

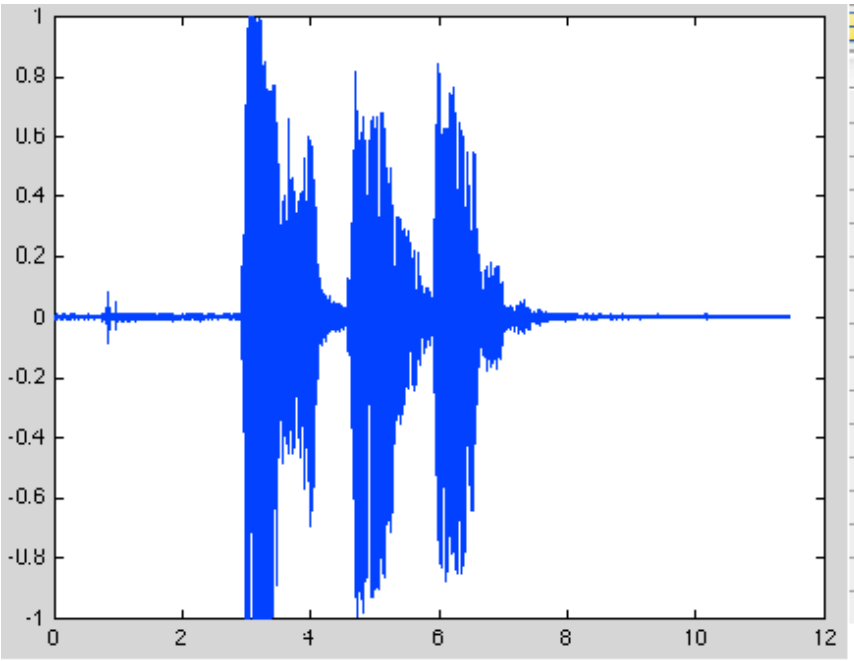
Esta etapa está basicamente dividida em duas etapas:

1. Ler o arquivo de áudio
2. Dividir o sinal em *Frames*

O primeiro passo faz parte do I/O do sistema. Como a implementação está em MATLAB, não foi considerado que o programa leia em tempo real os dados que irão ser processados, isto é, um possível *loop* infinito com uma função *callback* esperando algum evento para ser processado. Então, é usada uma função do MATLAB para ler arquivos de áudio. A função escolhida é chamada *wavread*, porém existem diversas implementações na internet que possibilita a leitura de arquivos mp3, por exemplo, sendo a escolha apenas uma opção pela facilidade de ser uma parte *default* do MATLAB e não ser necessário uma adição ao projeto. Ela por definição como visto em (VOICEBOX, 2011), pode retornar diversos valores, porém os mais relevantes são os seguintes:

1. Vetor que representa o sinal de áudio
2. *Sample Rate* do sinal

Foi mostrado no capítulo 2 deste documento as tecnologias de voz e conversão A/D. Neste processo, o *Sample Rate* faz papel importante no processo de obtenção dos valores digitais de um sinal. Ele é a medida do número de *samples* (amostras) por unidade de tempo que irão ser coletadas em um sinal. A unidade de tempo adotada em geral é o Hz (1/s). *Sample rate* clássico, por exemplo, tem valor de 44100 Hz. Isso se deve ao fato da lei proposta por Shannon-Nyquist para reconstrução de um sinal sem perda de dados. Isso porque foi provado por eles que caso se recolha uma quantidade de amostras maior que duas vezes a frequência da amostra, é possível reconstruir perfeitamente a mensagem original. Como a máxima frequência audível por um humano é de 20KHz, podemos reconstruir a mensagem perfeitamente com 44,1KHz. Podemos notar também que o tamanho do arquivo vai variar de acordo com essa taxa, já que quanto mais amostras, mais dados para registrar e mais bytes são ocupados. O arquivo todo será representado por um vetor, contendo cada uma das amostras. A imagem abaixo mostra o sinal e sua representação de uma parte de um arquivo lido:



speech <114688x1 double>

	1	2
258	-0.0053	
259	-0.0022	
260	-0.0023	
261	-0.0063	
262	-0.0027	
263	-0.0021	
264	-0.0050	
265	-8.8501e...	
266	-0.0016	
267	-0.0034	
268	-4.8828e...	
269	-0.0031	
270	-0.0024	
271	0.0011	
272	-0.0031	
273	-0.0013	

Figura 7 Sinal e sua Representação Vetorial

Após termos a frequência do *sample rate* e os seus dados coletados, é possível dividir o sinal em frames. Na implementação, foi utilizado o tempo de frame na faixa de 0.024s, baseado no que foi usado pelo autor do *paper* original. A partir dessas informações, foi possível montar frames para o sinal de entrada. Para esta etapa foi utilizada uma toolbox para MATLAB de processamento de sinal chamada *VoiceBox*. Nela, é possível se dividir o sinal em frames e obter outras informações como o total de frames, tamanho do frame, janela etc. Só a partir deste momento que a aplicação do algoritmo pode ser feita, já que em (Jiang, Malvar, 2000), toda a implementação é baseada em frames.

A principal diferença a ser notada é que o sinal original é um vetor, enquanto o sinal em frames é uma matriz, com o número de linhas determinada pela quantidade de frames.

5.3 MCLT

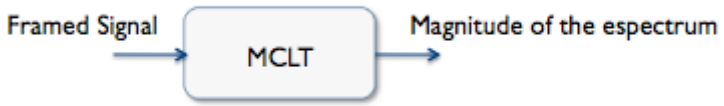


Figura 8 MCLT

A MCLT ou *modulated complex lapped transform* é uma transformação construída a partir de uma simples modificação da MLT. O intuito do desenvolvimento da MCLT é o fato da MLT não possuir importantes propriedades desejáveis para algumas aplicações como já citado anteriormente. Ela é do tipo *cosine-modulated filter bank* e mapeia blocos com valores inteiros em blocos com valores complexos de coeficientes transformados (Malvar, 1991). A decomposição de frequência da MCLT é similar a DFT e sua principal vantagem em relação à transformada de Fourier convencional é que possui formulas simples de reconstrução, sendo assim um ponto positivo para encoders, por exemplo.

Além disso, é proposto em (Malvar, 1991) algumas possibilidades de computação rápida, ou seja, uma versão da transformada que terá um desempenho computacional superior, possibilitando seu uso em aplicações que requerem maior velocidade, seja para hardware ou software (Malvar, 1999). Com essa implementação, e com características de sub-bandas complexas (informação de fase), pois carregam informações de áudio relevante à remoção de ruído, a transformada foi utilizada no documento base deste projeto.

Então, utilizando essa base teórica, e após dividir o sinal de áudio com ruído aditivo em frames, o sinal passará para o domínio da frequência através da MCLT proposta. Como resultado, teremos a magnitude e a fase do sinal.

Como mostrado no esquema de seção 5.1, a componente da fase só será utilizada no final de todo o processo, para encontrar o valor do sinal no tempo, então o passo final desta etapa é a divisão do sinal nessas duas partes: magnitude e fase, feita através do MATLAB.

5.4 Classificador

Figura 9 Classificador

Um classificador possui diversas aplicações, indo desde aprendizagem de máquina, tendo como objetivo identificar uma subpopulação até análises estatísticas mais complexas. Como explicado, a aplicação é um caso da técnica conhecida como atenuação de espectro. Ela segue dois passos básicos:

1. Estimar o espectro do ruído.
2. Filtrar o sinal, para obtê-lo sem distorções.

No caso da aplicação proposta, a classificação é feita para se identificar o que é ruído e o que é discurso em um sinal de voz no domínio MCLT, sendo um dos passos principais nesse tipo de técnica.

Para isso, o classificador tem que ser implementado baseado em alguma métrica. Uma forma extremamente trivial seria aplicar um *threshold* diretamente aos valores do sinal para se fazer uma filtragem. Essa abordagem é a mais simples possível e não estaria utilizando toda a informação disponível no sinal. Para utilizar a maior quantidade de informações possível, o classificador do sistema é baseado em energia.

A Energia do i-ésimo frame do espectro do sinal é definida como:

$$E^2(i) = \frac{1}{k_1 - k_0} \sum_{k=k_0}^{k_1} [|X(i,k)| - \bar{X}(i)]^2$$

Equação 3 Energia do i-ésimo frame (Jiang, Malvar, 2000)

Onde o valor do frame médio vale:

$$\bar{X}(i) = \frac{1}{k_1 - k_0 + 1} \sum_{k=k_0}^{k_1} |X(i,k)|$$

Equação 4 Valor do frame médio (Jiang, Malvar, 2000)

Com isso, é gerado um vetor de energias, onde cada elemento representa a energia de um determinado frame.

Pelas formulas mostradas acima, é possível identificar duas variáveis: k0 e k1, que valem respectivamente:

$$k_0 = 300N/f_s \;\; k_1 = 3000N/f_s$$

Equação 5 Definição de k0 e k1 (Jiang, Malvar, 2000)

Em (Jiang, Malvar, 2000), o autor define os limitantes como 3000 e 300 Hz por uma razão: o sinal de voz do ser humano está altamente concentrado entre esses valores. Esse valor também é conhecido como frequência de voz e é bastante utilizado em telefonia, tendo seu limitante superior às vezes alterado para 3400 Hz.

É importante ressaltar que a frequência fundamental da voz humana não se encontra nesse intervalo, como mostrado abaixo:

	Limitante Inferior (Hz)	Limitante Superior (Hz)
Homem	85	180
Mulher	165	255

Tabela 1 Frequência Fundamental da Voz

Entretanto, através da série harmônica, será possível obter todas as informações desejáveis na frequência aplicada.

Com o vetor de energia calculado, é hora de se aplicar o primeiro *threshold*. O primeiro processo de avaliação dessa energia com o limiar é chamado de *hard thresholding* ou *hard decision*. Esse termo refere a um tipo de processo mais superficial que avalia a seguinte regra:

$$E(i) > T$$

Equação 6 Regra do *threshold*

Um ponto que precisa ser avaliado é o cálculo do *threshold* inicial. Existem algumas abordagens conhecidas como o cálculo da média dos frames iniciais, que foi a utilizada nesta implementação.

Além disso, a regra de *threshold* é adaptada ao longo do tempo, ou seja, a cada iteração o valor é atualizado, como mostrado abaixo:

$$T = E_{min} + \delta(E_{max} - E_{min})$$

Equação 7 Adaptação do *threshold* (Jiang, Malvar, 2000)

onde,

$$E_{min} = min\{E(j)\},\ E_{max} = max\{E(j)\} \text{ and } j = i - W_e, i + 1 - W_e, \cdots, i - 1$$

Equação 8 Definições das energias (Jiang, Malvar, 2000)

É possível diminuir a velocidade de adaptação do *threshold* através do aumento da janela We, além de ser possível alterar a constante e o deixar mais robusto em relação a variações de energia.

Passada esta etapa, é necessário que se avalie um problema gerado por essa técnica: um discurso com baixa energia. Eles por terem valor inferior ao *threshold* determinado, irão ser classificados incorretamente e o processo de filtragem será consequentemente comprometido. Para evitar que esse tipo de erro ocorra, é proposta uma segunda etapa na classificação, chamada de *soft-thresholding* ou *soft-decision*. Para reavaliar o sinal de saída do *hard-decision*, a seguinte regra de classificação foi estabelecida, baseada em uma janela: se a energia do frame atual e de We frames anteriores for abaixo do *threshold*, então o *frame* atual será classificado como ruído. Caso contrário, será classificado como sinal de voz válido. A variável utilizada para esta função tem valor típico igual a 5. Uma maneira de entender esta regra é que um frame válido na grande maioria das vezes está junto de outros frames válidos, assim como ruídos.

5.5 Estimador de Ruído

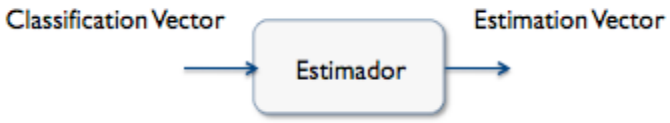


Figura 10 Estimador de Ruído

A saída dos classificadores é um vetor composto por 1's e 0's. Eles indicam se aquele frame é um ruído ou não. Sua saída será processada por outro módulo antes de ir para a filtragem de dois estágios de filtro de Wiener, que é o estimador de ruído. Cada um dos frames de ruído é usado para adaptar o *noise spectrum estimate*. A regra de adaptação é mostrada abaixo:

$$|\hat{N}(i,k)| = \beta|\hat{N}(i-1,k)| + (1-\beta)|X(i,k)|$$

Equação 9 *Noise Spectrum Estimate* (Jiang, Malvar, 2000)

A variável beta é controla a velocidade de adaptação. O valor típico utilizado nesta função é de 0.9.

O valor de saída da função é uma matriz com o número de linhas igual ao tamanho do vetor de classificação que é igual ao número de frames. O número de colunas será o tamanho do frame. A saída do estimador, como já falado, é uma estimativa do espectro do ruído e irá diretamente para o primeiro dos dois filtros de Wiener.

5.6 Filtro de Wiener - Dois Estágios

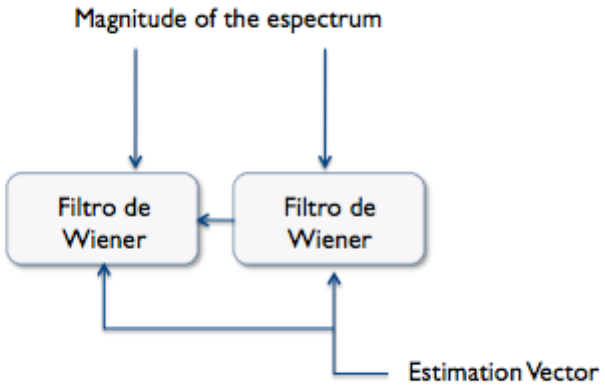


Figura 11 Dois estágios de Filtro de Wiener

Como discutido no capítulo 4, os filtros de Wiener são estruturas que lidam com ruído aditivo, e através de comparação com o sinal de saída desejado dão uma saída filtrada (Wiener Filter, 2011). Os Filtros de Wiener tradicionais e suas equações foram explicitadas e aplicações discutidas. Esse esquema porém é diferente do tradicional pois usa dois filtros de Wiener em sequência. Abaixo é mostrado a formula do ganho do filtro de Wiener:

$$G(k) = \frac{|S(k)|^2}{|S(k)|^2 + |N(k)|^2} = \frac{P(k)}{1 + P(k)}$$

Equação 10 Ganho do filtro de Wiener (Jiang, Malvar, 2000)

Com ela, *low level speeches frames* vai gerar uma oscilação do filtro gerando um ruído chamado *musical noise*. (Jiang, Malvar, 2000) e que é bastante comum e abordado por exemplo em (Audio Denoising by Time-Frequency Block Thresholding, 2011). Ele é gerado como dito acima e vai possuir notas musicais aleatórias e é de diferente natureza que o som, fazendo com que ele seja realmente perceptível. Abaixo é mostrado imagens relativas ao *musical noise*:

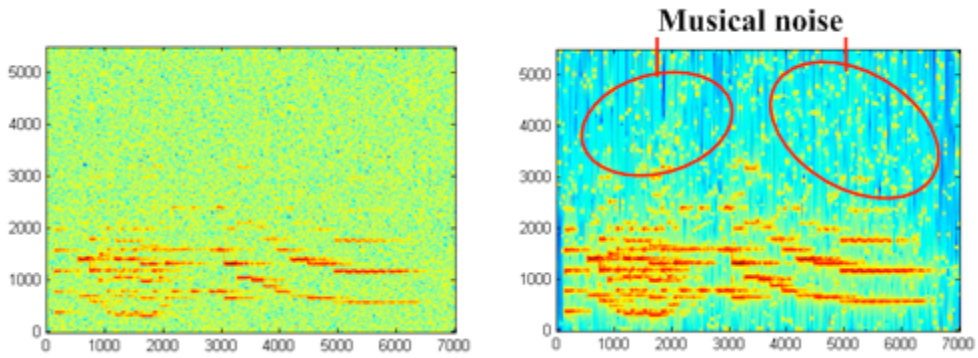


Figura 12 (a) Imagem Ruidosa (b) Imagem Filtrada com *Musical Noise (*Audio Denoising by Time-Frequency Block Thresholding, 2011)

Algumas soluções propõem um filtro após toda a remoção de ruído, como pode ser visto em (Esch, Vary, 2009), diferentemente da abordagem utilizada.

O primeiro filtro de Wiener usado é caracterizado pelas seguintes equações:

$$P^0(i,k) = \alpha\hat{P}(i-1,k) + (1-\alpha)P(i,k)$$

Equação 11 Filtragem do sinal utilizando estimativa ajustada de SNR

$$P(i,k) = (|X(i,k)|^2 - |\hat{N}(i,k)|^2)/|\hat{N}(i,k)|^2$$

Equação 12 Subtração Espectral do sinal e ruído

Onde,

$$\hat{P}(i-1,k) = |\hat{S}(i-1,k)|^2/|\hat{N}(i-1,k)|^2$$

Com o *smoothed estimate* P0, reduzimos a variação de ganho do filtro de Wiener e ajuda a suprimir o *musical noise*. Quanto maior o parâmetro alfa, menor o nível do *musical noise*.

Com isso, teremos a saída do primeiro filtro de Wiener. Foi notado entretanto que se utilizado um alfa de valor mais alto para evitar o ruído musical, um outro fenômeno será gerado, chamado de reverberação, que é a reflexão múltipla de uma freqüência (So, Why does reverberation affect speech intelligibility, 2011). Para que isso seja evitado, uma nova abordagem para o segundo filtro é proposta:

$$P^1(i,k) = \alpha\hat{P}(i-1,k) + (1-\alpha)P^u(i,k)$$

Equação 13 Segunda filtragem utilizando SNR

Onde,

$$P^u(i,k) = |\hat{S}(i,k)|^2/|\hat{N}(i,k)|^2$$

O objetivo da combinação dessas duas técnicas, além da remoção da interferência inicial, foi a eliminação do *musical noise*, bastante comum em abordagens de remoção de ruído e da reverberação, conseqüência da primeira parte, mas que conseguiu ser suprimida com sucesso.

5.7 IMCLT

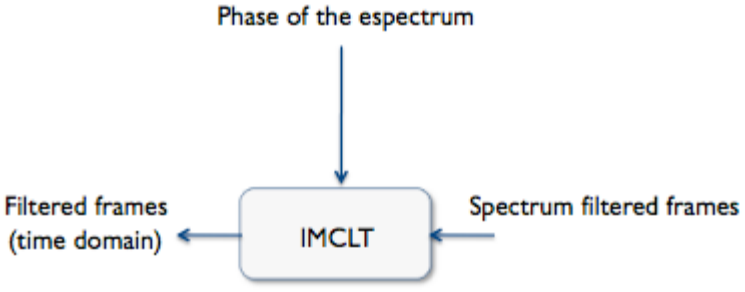


Figura 13 IMCLT

Após todos os cálculos, o sinal irá para o módulo que calcula a inversa da MCLT para que o sinal sem ruído possa ser salvo.

Para o cálculo da IMCLT, assim como MCLT, foi usado o algoritmo rápido proposto em [], e já pronto em MATLAB.

É possível notar que para calcular a inversa, diferentemente dos outros módulos, é necessário ter as duas partes do sinal na frequência, ou seja, a magnitude e a fase.

Com isso, teremos como resultado uma matriz com o número de linhas igual ao número de frames, e com a quantidade de colunas igual ao tamanho dos frames, ou seja, o sinal estará ainda dividido em frames, porém já no domínio do tempo.

5.8 Obtendo resultado final

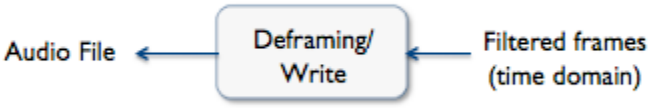


Figura 14 Processamento final do Sistema - Deframing e Escrevendo

Como última parte do processo, serão necessários dois passos para se ter o arquivo sem ruído:

- Transformar de Matriz de Frames para Vetor de Dados
- Escrever o arquivo final usando o vetor de dados

Para passar de matriz de frames para o vetor de dados foi usada a *toolbox* de MATLAB *VoiceBox* (VOICEBOX, 2011) que já possui uma função para tal, que recebe como parâmetros algumas variáveis geradas pela função de transformar o sinal em frames, da seção 5.1.

Com o vetor de dados já pronto, é possível escrever o arquivo em formato de áudio. Podemos então analisar a comparação entre o arquivo original (sem ruído), arquivo de entrada acometido por um ruído branco e o arquivo de saída.

Capítulo 6

Experimentos e Resultados

Esse capítulo tem como alvo explicar os experimentos realizados e avaliar os resultados obtidos. Os parâmetros utilizados e variados serão abordados além do módulo responsável pela inserção do ruído aditivo.

6.1 Inserção de ruídos

Inicialmente, foi proposto um módulo para teste que insere ruído aditivo em um sinal de áudio. Além disso, foi proposta a utilização de uma função padrão do MATLAB chamada agwn (*add white gaussian noise*) adicionando ruído ao sinal de entrada. Como a função de adição implementada não apresenta controle do nível ruído inserido, i.e., adicionar um determinado SNR pré-estabelecido, os testes foram realizados apenas com a solução do MATLAB que possui essa característica.

6.2 Parâmetros da Aplicação

A aplicação possui uma vasta quantidade de parâmetros, que afetam desde o cálculo de energia, passando pelo classificador, até o filtro de Wiener. Abaixo, são mostrados os parâmetros e sua função específicos, além dos valores padrão utilizados pelo autor.

Nome	Função	Valor
k0	Valor que limita inferiormente a frequência do discurso.	300.N/fs
k1	Valor que limita superiormente a frequência do discurso.	3000.N/fs
We	Janela no *hard-decision.*	20

δ	Constante do *Thresholding* no hard-decision.	0.2
Ws	Constante para regra no *soft-decision*.	5
β	Constante que controla a velocidade de adaptação no estimador de ruído.	0.9
α	Constante que controla a variação do filtro de Wiener, fazendo com que o *musical noise* seja suprimido.	0.97

Tabela 2 Parâmetros da Aplicação

6.3 Variação dos Parâmetros

Os parâmetros acima foram propostos com valores mostrados em (Jiang, Malvar, 2000), porém, durante os testes, outros valores foram testados. Os outros valores utilizados são mostrados abaixo:

Nome	Valor 1	Valor 2
We	10	25
δ	0.1	0.3
Ws	2	8
β	0.7	-
α	0.85	-

Tabela 3 Outros valores utilizados para as variaveis

Os casos de teste, com os parâmetros distintos serão:

Numero	We	δ	Ws	β	α
1	10	0.3	2	0.7	0.85
2	10	0.3	8	0.7	0.85
3	25	0.3	2	0.7	0.85
4	25	0.3	8	0.7	0.85
5	10	0.1	2	0.7	0.85

6	10	0.1	8	0.7	0.85
7	25	0.1	2	0.7	0.85
8	25	0.1	8	0.7	0.85

Tabela 4 Todos os testes realizados

Além disso, o teste será feito utilizando *white noise* ou ruído branco, com SNR de valor 8, com agwn mostrada acima, adicionando ruído a um arquivo de voz masculino gravado com um *sample rate* de 44100 Hz. O *frame* tem tamanho de 20 milissegundos e o ruído é do tamanho do arquivo de entrada. A tabela abaixo mostra de uma maneira concisa esses valores:

Descrição	Valor
Tipo de Voz	Masculina
Freqüência de Gravação do Arquivo	44.1 kHz
Tempo de cada *Frame*	20 milissegundos
Tipo do Ruído	*White Gaussian Noise*
Nível de ruído (equivalente SNR)	8 db
Tempo do Ruído	Duração do arquivo de entrada

Tabela 5 Descrição dos Arquivos de Teste

6.4 Resultados

Os resultados obtidos são mostrados abaixo, em termos do SNR, o s*ignal to noise ratio*, que segue a equação abaixo:

$$SNR = 10\log 10\frac{\sum_{n=0}^{N-1} s^2(n)}{\sum_{n=0}^{N-1}[y(n) - s(n)]^2}$$

Equação 14 Cálculo do SNR

Casos de Testes	SNR antes	SNR depois
1	8.0014	8.6030
2	8.0195	9.8935
3	7.9993	10.3476
4	7.9827	12.8404
5	8.0049	10.4986
6	7.9832	13.0093
7	8.0087	11.2990
8	**7.9910**	**14.6404**
Valores Propostos	7.9890	12.0228

Tabela 6 Valores de SNR das configurações

6.5 Análise dos Resultados

É possível verificar nesse caso específico (gravação com voz masculina acometida por *white noise*, como descrito na tabela 5) que ocorrem ganhos significativos em relação ao SNR. Um problema que ocorreu foi a falta de outros experimentos baseados em bases de dados distintas, além de um grande volume de testes, com outros tipos de ruídos, como marrom, rosa e ruídos reais como chiado de televisão, barulho de ventilador, chuva etc..

Foi possível verificar também que janelas maiores tem bastante impacto no resultado (na aplicação específica), e que a constante delta não apresenta grande influência, pois a melhora do SNR nesses casos não é tão significante.

Vale ressaltar que as melhores configurações nesse caso foram obtidas com valores diferentes dos propostos em (Jiang, Malvar, 2000), notando que os parâmetros podem variar de acordo com a aplicação, isto é, vão depender da entrada e do ruído presente.

Uma coisa a ser notada e que deve ser ressaltada é a presença de um ruído metálico ou um *metalic noise*. Com a variação dos parâmetros, a presença do ruído também variou, mas foi notada sua presença principalmente antes do discurso

começar, no Período de silêncio, sem atividade de voz. Uma possível solução para o problema seria a aplicação do algoritmo de VAD, já que detecta a atividade de voz no sinal de áudio e a partir dai poderia aplicar todo o processo de remoção de ruído proposto.

Capítulo 7

Conclusões e Trabalhos Futuros

Este capítulo aborda as conclusões obtidas com o trabalho além de abordagens e trabalhos futuros a partir desta monografia.

Foi mostrado um sistema de remoção de ruídos baseado na técnica de atenuação de espectro, que utiliza dois estágios de filtro de Wiener com objetivo de reduzir o chamado *musical noise*. Além disso, um sistema de classificação é feito em duas fases, para que ocorra menos erros. Essas são as duas características que mais se diferenciam do restante das aplicações que visam redução de ruído (Jiang, Malvar, 2000).

Algumas possíveis implementações e modificações são propostas para que ocorra tanto uma melhora no sistema, quanto a possibilidade de sua utilização em uma quantidade grande de contextos. Segue abaixo uma pequena lista de alterações e em seguida, o motivo para cada um dos tópicos.

- Sistema de Avaliação de Voz
- VAD
- Implementação do sistema em C/C++
- Embarcar a solução

- Sistema de Avaliação de Voz

Se analisarmos o sistema, iremos identificar no classificador dois parâmetros denominados k0 e k1. Eles são os limiares inferiores e superiores que caracterizam a voz humana, objeto de estudo e de filtragem. Como mostrado na mesma seção, é sabido que a frequência da voz de mulheres e homens na sua origem já é distinta. Além disso, cada pessoa possui suas próprias características de fala, já que é um sistema que envolve várias partes do corpo humano. A idéia de fazer um sistema de

avaliação de voz é identificar o range da frequência de voz do locutor, para que com isso seja possível fazer uma avaliação de energia mais precisa.

- VAD

O VAD, ou *voice activity detection*, é uma técnica utilizada em processamento de voz para detectar a presença ou ausência de um sinal de voz válido, como mostrado na seção 2.3.3. A intenção com o VAD seria detectar quando, no sinal de áudio, a voz estaria presente, para que então todo o processo de remoção de ruído atuasse. Isso seria uma economia de processamento, além de não alterar outras partes que não possuíssem o sinal de voz.

- Implementação do Sistema em C/C++

Apesar da grande facilidade da utilização de MATLAB, o ambiente de desenvolvimento é muito mais utilizado para validação e comprovação de resultados do que para um sistema que possa ser realmente utilizado. Para isso, se escolhe alguma linguagem e a implementação para que a aplicação possa ser aplicada em algum contexto. A proposta para futuros trabalhos seria a implementação do sistema em C ou C++, devido a sua flexibilidade e velocidade além de uma grande quantidade de bibliotecas *open source* que auxiliariam no processo de desenvolvimento.

- Embarcar a Solução

O última sugestão como um possível trabalho, seria a utilização da aplicação em um sistema embarcado ou mesmo em um software já desenvolvido, para se ter a real noção do desempenho da solução em tempo real. Além disso, seria a oportunidade de utilizar o software na prática.

Referências Bibliográficas

Analog to Digital Converter – Online, Acesso em 15/11/2012 na URL: http://en.wikipedia.org/wiki/Analog-to-digital_converter

Antares Auto-Tune – Online, Acesso em 16/11/2012 na URL: http://www.antarestech.com/products/auto-tune-evo.shtml

Audio Denoising by Time-Frequency Block Thresholding - Online, Acesso em 01/12/2012 na URL: http://www.cmap.polytechnique.fr/~yu/research/ABT/samples.html

Bencina, R.; Burk, P. PortAudio – an Open Source Cross Platform Audio API, ICMC, Estados Unidos, 2001.

Bhansali, B.J. Applications of the Wiener Filter Technique to some economic time series, Estudios de Economia, vol. 9, issue 1, pages 145-153, Chile, 1982.

Biomedical Images Analysis – Wiener Filter, Online, Acesso em 23/11/2012 na URL: http://informatik.unibas.ch/lehre/fs10/cs252/slides/wiener.pdf

Del Negro, C. Application of the Wiener filter to magnetic profiling in the volcanic environment of Mt. Etna (Italy), Annals of Geophysics, Italia, 2006.

Esch, T.; Vary P. Efficient Musical Noise Suppression For Speech Enhancement Systems, IEEE International Conference on Acoustics, Speech and Signal Processing , Alemanha, 2009.

Image Deblurring – Wiener Filter, Online, Acesso em 23/11/2012 na URL: http://blogs.mathworks.com/steve/2007/11/02/image-deblurring-wiener-filter/

Jiang, W; Malvar, H. Adaptive Noise Reduction of Speech Signals, Microsoft Technical Report, Estados Unidos, Julho 2000.

Malvar, H. A Modulated Complex Lapped Transform And Its Applications to Audio Processing, Proc. International Conference on Acoustics, Speech and Signal Processing, Estados Unidos, 1999.

Malvar, H. Fast Algorithm for the Modulated Complex Lapped Transform, IEEE Signal Processing Letters, vol. 10, No. 1, Estados Unidos, 2003.

Marks II**;** R.J**.** Introduction to Shannon Sampling and Interpolation Theory, Spinger-Verlag, Estados Unidos**,**1991.

Mina A. M; Raghunandan H. K. Wiener and Kalman Filters for Denoising Video Signals, EE378 Class Project - Stanford University, Estados Unidos, 2008

Ramirez, J.; Gorriz, J. M; Segura, J. C. Voice Activity Detection. Fundamentals and Speech Recognition System Robustness, InTech, Estados Unidos, Junho 2007.

So, Why does reverberation affect speech intelligibility, Online, Acesso em 06/12/2012 na URL: http://www.mcsquared.com/y-reverb.htm

Tawari, A.; Trivedi, M.M. Speech Emotion Analysis: Exploring the Role of Context, IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 12, NO. 6, Estados Unidos, Outubro 2010.

Tsoukalas, D. E.; Mourjopoulos; Kokkinakis, G. Speech enhancement based on audible noise suppression. IEEE Trans. On speech and audio processing, pp497 – 514, Grecia,1997.

Virag, N. Single Channel speech enhancement based on masking properties of the human auditory system, IEEE Trans, On speech and audio processing, pp.126-137,Suiça,1999.

VOICEBOX: Speech Processing Toolbox for MATLAB Online, Acesso em 5/12/2012 na URL: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

Walden, R.H. Analog-to-Digital Converter Survey and Analysis, IEEE Journal on Selected Areas In Communication, VOL 17, Estados Unidos, Abril 1999.

Wiener Filter – Online, Acesso em 23/11/2012 na URL: http://en.wikipedia.org/wiki/Wiener_filter

Glossário

A/D - Analógico/Digital

D/A - Digital/Analógico

HMM – *Hidden Markov Model*

IMCLT - *Inverse Modulated Complex Lapped Transform*

KNN – *K Nearest Neighbor*

LPC - *Linear predictive coding*

LPCC - *Linear predictive coding coefficient*

MCLT - *Modulated Complex Lapped Transform*

MFCC - *Mel-frequency cepstral coefficients*

VoIP - *Voice Over Internet Protocol*

RNA – Rede Neural Artificial

SNR – *Signal to Noise Ratio*

STT – *Speech to Text*

VAD – *Voice Activity Detection*