

Pietro Basile Neto RA: 11000511

Trabalho de Graduação III
Restauração Digital de Gravações de Áudio

Santo André

1º Quadrimestre 2019

Dedico essa nova realização aos meus familiares, sobretudo à minha avó Cida, que ofereceu força, apoio e motivação em minha trajetória acadêmica, mas não pôde estar ao meu lado neste momento tão importante.

Agradecimentos

Agradeço ao Professor Doutor Irineu Antunes Júnior por todo o apoio, paciência, orientação e dedicação durante todo o longo processo de desenvolvimento deste trabalho.

À toda a comunidade acadêmica da UFABC e todos que ajudaram, direta ou indiretamente, em toda a minha trajetória nesta instituição.

Aos meus familiares, principalmente meus pais, por tudo. À minha esposa que sempre me deu todo o apoio e motivação para que esse trabalho fosse possível.

Resumo

A digitalização de gravações de áudio é um esforço de manutenção da história e democratização da informação. Este trabalho estuda formas de reduzir os ruídos encontrados nesse tipo de gravação. São estudados e implementados dois métodos: Subtração Espectral e Transformada *Wavelet*. Foram feitas simulações (com resultados tanto qualitativos quanto quantitativos) e uma aplicação de caso real, onde foi possível observar qualitativamente o ganho da digitalização e da redução de ruído.

Lista de ilustrações

Figura 1 – Forma de onda de uma música degradada por <i>clicks</i> . Fonte: (GODSILL; RAYNER; CAPPÉ, 2002).	13
Figura 2 – Exemplo de Árvore de decomposição <i>Wavelet</i> . Fonte: (MISITI et al., 1996).	16
Figura 3 – Fitas de rolo que foram digitalizadas.	20
Figura 4 – Sistema de digitalização das fitas de rolo.	21
Figura 5 – Forma de onda do áudio de entrada no domínio do tempo.	26
Figura 6 – Forma de onda do áudio de entrada no domínio da frequência.	27
Figura 7 – Espectrograma do áudio de entrada.	27
Figura 8 – Forma de onda do áudio ruidoso no domínio do tempo.	28
Figura 9 – Forma de onda do áudio ruidoso no domínio da frequência.	28
Figura 10 – Espectrograma do áudio ruidoso.	29
Figura 11 – Forma de onda do áudio filtrado por Subtração Espectral no domínio do tempo.	29
Figura 12 – Forma de onda do áudio filtrado por Subtração Espectral no domínio da frequência.	30
Figura 13 – Espectrograma do áudio filtrado por Subtração Espectral.	30
Figura 14 – Forma de onda do áudio filtrado por Transformada Coiflet no domínio do tempo.	31
Figura 15 – Forma de onda do áudio filtrado por Transformada Coiflet no domínio da frequência.	32
Figura 16 – Espectrograma do áudio filtrado por Transformada Coiflet.	32
Figura 17 – Forma de onda do áudio filtrado por Transformada de Daubechies no domínio do tempo.	33
Figura 18 – Forma de onda do áudio filtrado por Transformada de Daubechies no domínio da frequência.	34
Figura 19 – Espectrograma do áudio filtrado por Transformada de Daubechies.	34
Figura 20 – Forma de onda do áudio "Lina.wav" no domínio do tempo.	36
Figura 21 – Forma de onda do áudio "Lina.wav" no domínio da frequência.	36
Figura 22 – Espectrograma do áudio "Lina.wav".	37
Figura 23 – Forma de onda do áudio "Lina.wav" filtrado por Transformada <i>Wavelet</i> tipo Daubechies no domínio do tempo.	37
Figura 24 – Forma de onda do áudio "Lina.wav" filtrado por Transformada <i>Wavelet</i> tipo Daubechies no domínio da frequência.	38
Figura 25 – Espectrograma do áudio "Lina.wav" filtrado por Transformada <i>Wavelet</i> tipo Daubechies.	38

Lista de tabelas

Tabela 1 – Resumo da história do áudio gravado e principal forma de ruído presente. Fonte: (FERREIRA; JÚNIOR, 2011).	10
Tabela 2 – Resultados quantitativos do sinal filtrado pelo método de Subtração Espectral.	31
Tabela 3 – Resultados quantitativos do sinal filtrado pelo método de Transformada <i>Wavelet</i> (coif5).	33
Tabela 4 – Resultados quantitativos do sinal filtrado pelo método de Transformada <i>Wavelet</i> (db10).	35
Tabela 5 – Resultados quantitativos de todos os métodos propostos.	35

Lista de abreviaturas e siglas

AWGN - *Additive White Gaussian Noise*

CD - *Compact Disk*

DAT - *Digital Audio Play*

DFT - *Discrete Fourier Transform*

LD - *Laser Disk*

LP - *Long Play*

MOS - *Mean Opinion Score*

PESQ - *Perceptual Evaluation of Speech Quality*

rpm - Rotações por minuto

SNR - *Signal to Noise Ratio*

SNR_{seg} - Relação Sinal Ruído Segmentada

Sumário

1	INTRODUÇÃO	9
1.1	Objetivos	9
2	TÉCNICAS DE REDUÇÃO DE RUÍDO	12
2.1	Revisão Bibliográfica	12
2.1.1	Remoção de <i>Clicks</i>	12
2.1.2	Correlated Noise Pulse Removal	13
2.1.3	Subtração Espectral de Potência	14
2.1.4	Filtro de Kalman	14
2.1.5	Transformada <i>Wavelet</i>	16
2.2	Métricas de Desempenho	17
2.2.1	Relação Sinal Ruído Segmentada	17
2.2.2	<i>Perceptual Evaluation of Speech Quality</i>	18
3	DESCRIÇÃO EXPERIMENTAL	19
3.1	Materiais	19
3.2	Metodologia	19
3.2.1	Revisão Bibliográfica:	19
3.2.1.1	Estudo das principais técnicas de redução de ruído:	19
3.2.1.2	Estudo das técnicas escolhidas:	19
3.2.2	Coleta e Digitalização de amostras de Áudio	20
3.2.3	Início da simulação em computador:	20
3.2.3.1	Subtração Espectral	20
3.2.3.2	Transformada <i>Wavelet</i>	23
3.2.4	Verificação dos programas de restauração de sinais de áudio:	25
3.2.5	Teste com Dados Reais	25
4	RESULTADOS E DISCUSSÃO	26
4.1	Subtração Espectral	26
4.2	Transformada <i>Wavelet</i>	31
4.2.1	Coiflet	31
4.2.2	Daubechies	33
4.3	Discussão dos Resultados Quantitativos	35
4.4	Áudio Lina.wav	35
	Conclusão	39

REFERÊNCIAS	40
APÊNDICE A – CÓDIGO FINAL	42

1 Introdução

Na era digital, temos acesso a um número cada vez maior de informações. Devido a esse grande volume, duas tarefas mostram-se fundamentais: curar e facilitar o acesso à informações de qualidade. No campo do áudio, isso significa a digitalização de áudios históricos, que antes ficavam confinados em salas escuras de museus, e sua disponibilização às massas. Isso sem contar, como citado em (ORIO et al., 2010), com o desaparecimento, cada vez mais rápido, das tecnologias de reprodução obsoletas, tornando os áudios armazenados, sempre mais difíceis de reproduzir.

O principal problema encontrado na digitalização de áudio é o ruído que os acompanha. Ruído, neste contexto, pode ser definido como todo som indesejável (PRATA-SHIMOMURA et al., 2017) que acompanha, mas não agrega à mensagem sendo transmitida. A Tabela 1 apresenta um histórico das tecnologias de gravação e reprodução de áudio, assim como os principais tipos de ruído encontrado em cada uma delas.

Infelizmente o ruído não pode ser completamente eliminado, mas existem diversas técnicas para a redução do mesmo. Algumas técnicas levantadas em (FERREIRA; JÚNIOR, 2011) são apresentadas a seguir.

- Técnicas de alteração da amplitude: compressores, limitadores e *noise gates* (ROADS, 1996).

- Técnicas baseadas em transformadas e que manipulam o espectro do sinal ruidoso para eliminar ruído de banda larga, entre elas: filtros e equalizadores digitais (ROADS, 1996); subtração espectral; modificação de coeficientes espectrais usando funções de limiar (*thresholding-functions*) (ANTUNES, 2006).

- Técnicas para detectar e eliminar pulsos espúrios (*clicks* e *clacks*) (GODSILL; RAYNER; CAPPÉ, 2002).

- Técnicas para reconstruir trechos interrompidos: interrupções longas (ESQUEF; BISCAINHO, 2006) ou breves.

Este trabalho foca nas técnicas baseadas em transformada e na manipulação do espectro do sinal ruidoso. Dentre estas, foram escolhidos dois métodos, Subtração Espectral e Transformada *Wavelet*, que serão discutidas mais a fundo na Seção 2.

1.1 Objetivos

Neste trabalho, investigam-se técnicas de processamento digital de sinais para restauração de gravações de áudio deterioradas, buscando reduzir o ruído de fundo e

Tabela 1 – Resumo da história do áudio gravado e principal forma de ruído presente. Fonte: (FERREIRA; JÚNIOR, 2011).

Data	Equipamento	Mídia	Principal forma de ruído presente
1877	Fonógrafo de Thomas Edison	Cilindro de latão	Estalos rápidos e agudos (<i>clicks</i>), pulsos longos (<i>cracks</i>), ruído “chiado” produzido pelo atrito da agulha (<i>hiss</i>), ressonâncias de partes mecânicas (<i>rumble</i>), distorções de velocidade de reprodução (<i>wow</i>).
1885	Gramofone da Bell & Tainter	Cilindro de cera	
1888	Gramofone de Berliner	Discos de vulcanite, 70 rpm.	
1906	Vitrola com corneta acoplada.	Discos de goma-laca, 78 rpm.	
1935	Magnetofone, BASF e AEG-Telefunken	Fita magnética	Ruído “chiado” de alta frequência (<i>tape hiss</i>), distorções de velocidade (<i>wow</i>) e “zumbido” proveniente das partes eletrônicas e da rede elétrica 50/60Hz (<i>buzz</i>).
1948	Vitrola	Discos LP (<i>long-play</i>) em vinilite, 33 1/3 rpm.	Os mesmos dos equipamentos mecânicos juntamente com ruídos da parte eletrônica.
1948	Gravador de fita de rolo, Magnecord	Fita magnética, gravação em estéreo.	Os mesmos do “Magnetofone” com possível melhoria de qualidade e uso de técnicas analógicas para reduzir o ruído (sistema Dolby, 1966), atualmente, obsoletas.
1963	Gravador de audiocassete compacto da Philips.	Fita magnética de 1/8”, 1 7/8 ips. (Fitas K7)	
1972	Reprodutor de disco laser da Philips	Disco laser (LD)	Largura de banda dada pela taxa de amostragem e relação sinal-ruído depende do número de bits de digitalização. Presença de ruído (erros) na forma de aliasing, faixa dinâmica e <i>jitter</i> .
1980	Reprodutor de CD's	Discos compactos (CD's) de 5”.	Quando os dados estão muito corrompidos, podem apresentar interrupções na reprodução.
1983	Gravador digital	<i>Digital audio tape</i> (DAT)	
Atualidade	Meios digitais	Arquivos digitalizados em diversos formatos (wav, mp3 etc.)	

combater outros sinais espúrios ou distorções. Duas técnicas foram selecionadas para estudo mais detalhado, implementação e testes computacionais: Subtração Espectral e Transformada *Wavelet*.

2 Técnicas de Redução de Ruído

Neste capítulo será apresentada uma revisão bibliográfica dos principais tópicos abordados neste trabalho.

2.1 Revisão Bibliográfica

Neste trabalho, são enfatizadas as técnicas que visam combater o ruído, não sendo objeto de estudo o uso de “efeitos especiais”, como a introdução de reverberação, atrasos, alteração de escala de tempo e altura (*flangers, chorus*), os quais, normalmente, são empregados em trabalhos de “remasterização” de áudio. Ou seja, considera-se aqui a restauração com a finalidade de obter uma versão do sinal semelhante ao áudio original e, em especial, busca-se mitigar imperfeições introduzidas por uma dada forma de armazenamento.

Nesta seção tem-se algumas das técnicas estudadas.

2.1.1 Remoção de *Clicks*

Degradações relativamente curtas de áudio, que ocorrem tanto em meios analógicos como digitais, que podem ser percebidas pela nossa audição como “ticks” ou arranhados (em discos LP), são conhecidos como *Clicks* (GODSILL; RAYNER; CAPPÉ, 2002).

A Figura 1 apresenta a forma de onda de um trecho de música degradado por este tipo de ruído.

Um dos métodos de remoção de tais degradações é composto das seguintes etapas (GODSILL; RAYNER; CAPPÉ, 2002):

- Modelagem de *Clicks*: Definição do modelo de *click* a ser substituído. Pode ser aditivo ou substitutivo. O caso aditivo é o mais usado por ser mais efetivo (trata casos com duração mais curta, portanto, mais fáceis de tratar).

O modelo aditivo pode ser expresso como:

$$x[n] = s[n] + i[n].v[n] \quad (2.1)$$

onde $s[n]$ é o sinal de áudio utilizado, $v[n]$ é um processo de corrupção de ruído e $i[n]$ é um booleano que marca se existe *click* ou não naquele ponto.

- Detecção: O processo de detecção pode ser muito complexo, ao envolver outros sinais de áudio, técnicas estáticas, entre outros. Porém a remoção deve ser orientada no

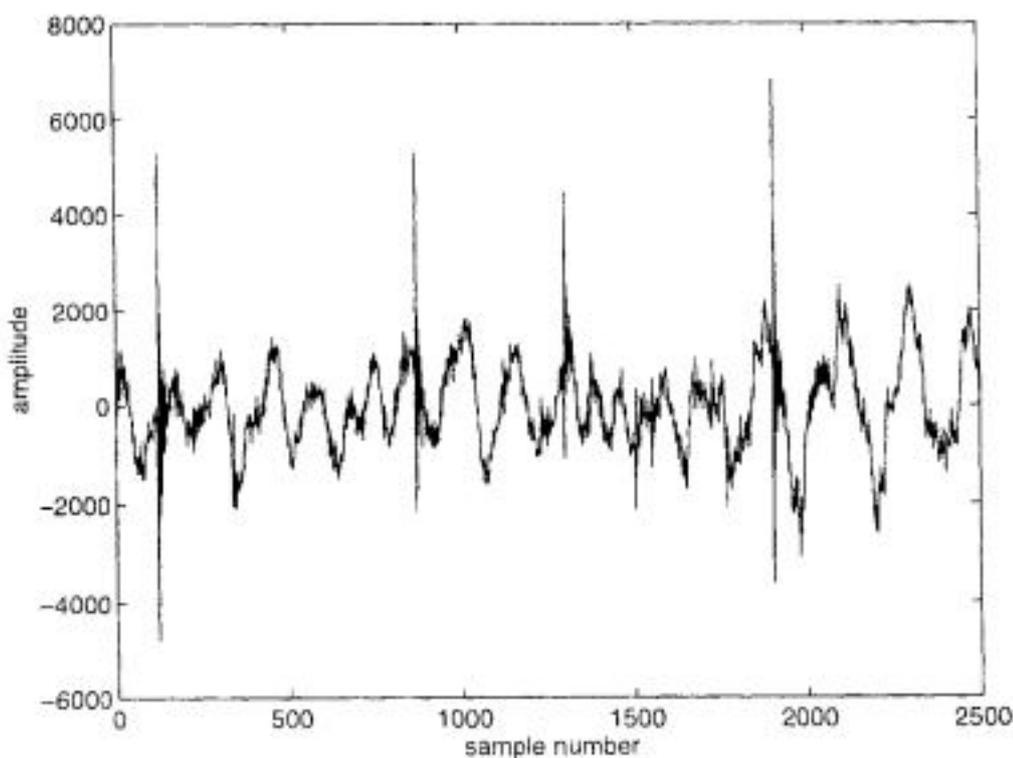


Figura 1 – Forma de onda de uma música degradada por *clicks*. Fonte: (GODSILL; RAYNER; CAPPÉ, 2002).

sentido de fazer o sinal ficar o mais audível possível ao ouvinte, portanto a melhor maneira de fazê-lo seria levar em consideração o fator humano e onde percebemos as degradações. Para isso, podemos usar um filtro passa alta, pois a maioria das gravações de voz apresentam pouca informação em alta frequência, ao contrário do ruído, que apresenta em todas as faixas.

- Substituição das amostras corrompidas: Nesta etapa, deve-se preencher as lacunas deixadas pela remoção dos *clicks*. Isto é feito, normalmente, usando algum método de interpolação para gerar um sinal do tamanho do sinal removido, levando em consideração os sinais anteriores e posteriores.

2.1.2 Correlated Noise Pulse Removal

Outra forma comum de degradação são pulsos de baixa frequência. Estes pulsos são normalmente causados por problemas mecânicos ou elétricos no sistema de captação (GODSILL; RAYNER; CAPPÉ, 2002).

Para a remoção deste tipo de ruído, normalmente são utilizados métodos baseados em *Template*. Tais técnicas surgiram da descoberta de que o formato de sucessivos pulsos de baixa frequência numa mesma gravação são muito parecidos (GODSILL; RAYNER; CAPPÉ, 2002). Bastando identificá-los (através de sua repetição) e substituí-los.

2.1.3 Subtração Espectral de Potência

Seguindo a suposição já descrita neste trabalho que o sinal de áudio é a soma do sinal original sem ruído com o sinal do ruído, temos:

$$y(n) = x(n) + v(n) \quad (2.2)$$

Onde $y(n)$ é o sinal de áudio com ruído, $x(n)$ é o sinal original e $v(n)$ o ruído aditivo gaussiano.

Como descrito em (BEROUTI; SCHATZ; MAKHOUL, 1979), o método de Subtração Espectral consiste em calcular a potência espectral de cada trecho do sinal de áudio e subtrair uma estimativa da potência espectral do ruído. Tal estimativa é feita durante os períodos considerados como silenciosos. Portanto podemos descrever o método a partir de:

$$P'_s(w) = \begin{cases} P_s(w) - P_n(w); & P_s(w) - P_n(w) > 0 \\ 0; & \text{caso contrário} \end{cases} \quad (2.3)$$

Sendo $P'_s(w)$ a potência espectral modificada, $P_s(w)$ a potência espectral do sinal com ruído, $P_n(w)$ a estimativa da potência espectral de ruído.

A estimativa da potência espectral do ruído ($P_n(w)$) é feita calculando a média da potência espectral de quadros silenciosos do sinal de áudio estudado.

O sinal restaurado é conseguido com a junção da potência espectral modificada ($P'_s(w)$) e a transformada inversa de Fourier da fase original:

$$s'(t) = F^{-1} \left\{ \sqrt{P'_s(w)} e^{j\Theta(w)} \right\} \quad (2.4)$$

Onde $\Theta(w)$ é a fase da DFT (*Discrete Fourier Transform*) do áudio estudado.

Como a Equação 2.2 é apenas uma aproximação e nem sempre é válida, pode ocorrer que a potência espectral modificada de alguns quadros sejam negativos. Caso ocorra podemos considerá-los como nulos.

Este processo, apesar de diminuir o ruído, não o elimina, além de adicionar um novo ruído, conhecido como "ruído musical" (BEROUTI; SCHATZ; MAKHOUL, 1979).

2.1.4 Filtro de Kalman

O filtro de Kalman foi desenvolvido pelo matemático Rudolf Emil Kalman em 1960. Inicialmente solucionava problemas lineares relacionados à filtragem de dados discretos. Com o passar do tempo outros estudiosos utilizaram o filtro de Kalman, adaptando-o

à diversas aplicações, além de problemas lineares e com dados contínuos. O filtro é um conjunto de equações matemáticas, que através de um processo recursivo gera estimações de parâmetros desconhecidos (PUC-RIO, S.I.).

Para a redução de ruídos pode-se utilizar o filtro de Kalman de tempo discreto. Para isto, modela-se o sinal de áudio como um processo autoregressivo como descrito em (SILVA et al., 2016):

$$x(n) = \sum_{k=1}^P a_p(k)x(n-k) + w(n) \quad (2.5)$$

Onde $a_p(k)$ são os coeficientes de predição linear, $w(n)$ é o erro de predição e $x(n)$ o sinal de áudio.

A partir das equações 2.5 e 2.2 podemos construir um modelo de espaço de estados, gerando assim as equações:

$$x(n) = A(n-1)x(n-1) + w(n) \quad (2.6)$$

$$y(n) = H(n)x(n) + v(n) \quad (2.7)$$

Onde $x(n)$ é um vetor de estado de dimensão $P \times 1$, $A(n-1)$ é uma matriz transição de estados com dimensão $P \times P$, $w(n)$ é um vetor modelado por um ruído branco, $y(n)$ é o vetor de observação de dimensão $M \times 1$, $H(n)$ é uma matriz de distorção de canal de ordem $M \times P$ e $v(n)$ é um vetor de ruído branco aditivo de dimensão $M \times 1$.

Como $w(n)$ e $v(n)$ são ruídos brancos e independentes, temos as seguintes equações de correlação:

$$E[v(n)v^t(k)] = \begin{cases} R(n); & \text{if } k = n \\ 0; & \text{if } k \neq n \end{cases} \quad (2.8)$$

$$E[w(n)w^t(k)] = \begin{cases} Q(n); & \text{if } k = n \\ 0; & \text{if } k \neq n \end{cases} \quad (2.9)$$

Onde $R(n)$ e $Q(n)$ são os termos da diagonal das matrizes de covariância, dizendo respeito, respectivamente ao ruído aditivo e ao erro de predição.

Finalmente, as equações para o filtro de Kalman ainda podem ser divididas em dois grupos. Um com as equações de predição, como a Equação 2.10.

$$\hat{x}\left(\frac{n}{n-1}\right) = A(n-1)\hat{x}\left(\frac{n-1}{n-1}\right) \quad (2.10)$$

O segundo grupo traz as equações de correção:

$$K(n) = P \left(\frac{n}{n-1} \right) H^T(n) [H(n)P \left(\frac{n}{n-1} \right) H^T(n) + R(n)]^{-1} \quad (2.11)$$

$$\hat{x} \left(\frac{n}{n} \right) = \hat{x} \left(\frac{n}{n-1} \right) + K(n)[y(n) - H(n)\hat{x} \left(\frac{n}{n-1} \right)] \quad (2.12)$$

$$P \left(\frac{n}{n} \right) = [1 - K(n)H(n)]P \left(\frac{n}{n-1} \right) \quad (2.13)$$

Onde $K(n)$ é o ganho de Kalman, $\hat{x} \left(\frac{n}{n} \right)$ é a estimativa do sinal em n e $P \left(\frac{n}{n} \right)$ é a matriz de erro de covariância em n .

2.1.5 Transformada *Wavelet*

A redução de ruído através da Transformada *Wavelet* consiste em decompor o sinal em coeficientes de aproximação, parte inteligível do áudio para o caso de gravações de voz (A_m , que carregam informações de baixa frequência da *Wavelet* associada), e detalhe, que carrega as características que diferem a voz de uma pessoa e outra (D_m , informações de alta frequência). A Figura 2 apresenta um exemplo de árvore de decomposição de um sinal.

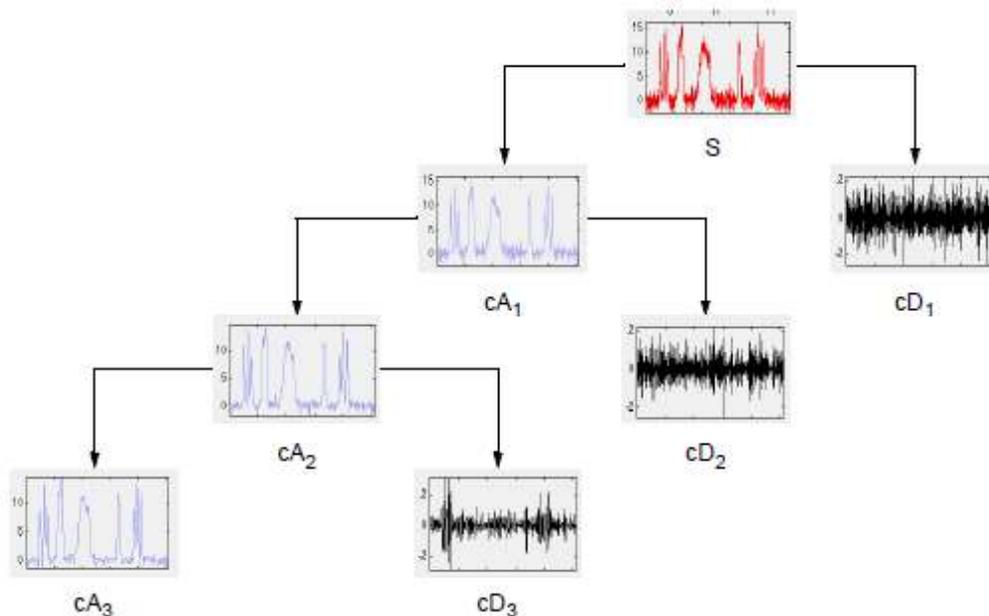


Figura 2 – Exemplo de Árvore de decomposição *Wavelet*. Fonte: (MISITI et al., 1996).

Então é aplicado um limiar, para decidir quais desses coeficientes carregam informações importantes e quais podem ser eliminados, reduzindo o ruído (SILVA et al., 2014).

Em (SILVA et al., 2014) o algoritmo de redução de ruído usando a Transformada *Wavelet* é proposto da seguinte forma:

$$Wf(a, b) = \int_{-\infty}^{\infty} f(t)\psi_{a,b}(t)dt \quad (2.14)$$

Em que $f(t)$ é um sinal de tempo contínuo e $\psi_{a,b}$ é a função *Wavelet*. Porém, podemos aproximar $f(t)$ para um sinal discreto de N pontos, assim obtendo:

$$Wf(a, b) = \sum_{t=0}^{N-1} f(t)\psi_{a,b}(t) \quad (2.15)$$

A função *Wavelet* ($\psi_{a,b}$) é derivada da transformação:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \quad (2.16)$$

Sendo "b" a translação da *Wavelet* e "a" o parâmetro de escala, associado à largura da janela.

Após testes, (SILVA et al., 2014) utilizaram a Equação 2.17 como tal limiar.

$$D_1(n) = \begin{cases} D_1(n); & \text{se } D_1(n) \geq 0, 3Máx(D_1(n)) \\ 0; & \text{caso contrário} \end{cases} \quad (2.17)$$

Finalmente, após a eliminação dos coeficientes, o sinal tratado é obtido calculando a Transformada *Wavelet* Inversa.

2.2 Métricas de Desempenho

Para a comparação e discussão dos resultados é necessário a definição de parâmetros não subjetivos aos quais os sinais tratados possam ser submetidos. Utilizaremos a *Perceptual Evaluation of Speech Quality* (PESQ) e, assim como nos trabalhos de (SILVA et al., 2016) e (SILVA et al., 2014), a Relação Sinal Ruído Segmentada (SNR_{seg}).

2.2.1 Relação Sinal Ruído Segmentada

A (SNR_{seg}) é definida como a média da SNR de pequenas janelas do sinal, reduzindo assim um pouco da distorção causada por trechos com intensidade muito alta ou baixa. Pode ser descrita pela equação (GRUNDLEHNER et al., 2005):

$$SNR_{seg} = \frac{1}{N} \sum_{K=1}^N 10\log_{10} \left(\frac{\sum_{n \in frame_k} |x(n)|^2}{\sum_{n \in frame_k} |\hat{x}(n) - x(n)|^2} \right) \quad (2.18)$$

Onde N é o número de janelas, $n \in frame_k$ são os instantes n que pertencem à k -ésima janela, $x(n)$ o sinal livre de ruídos e $\hat{x}(n)$ o sinal tratado.

A (SNR_{seg}) pode assumir qualquer valor, mas quanto maior, melhor será a qualidade do áudio.

2.2.2 *Perceptual Evaluation of Speech Quality*

A SNR_{seg} não é uma medida que leva em consideração a percepção humana, o que pode torná-la menos útil em alguns casos. Já a *Perceptual Evaluation of Speech Quality* (PESQ) tenta simular a *Mean Opinion Score* (MOS), medida de áudio desenvolvida calculando-se a média das notas dadas por pessoas, em uma escala de 0 a 5. Diferentes pessoas ouvem o mesmo áudio e a média das notas ao quadrado é a MOS. A PESQ foi desenvolvida com o processamento de uma biblioteca de áudios dos quais a MOS era conhecida até que o algoritmo pudesse se aproximar da nota auferida em áudios em que a MOS não era conhecida pelo mesmo (PENNOCK, S.I.). Assim como a MOS, a PESQ é medida de 0 a 5, sendo 5 a melhor nota possível.

3 Descrição Experimental

Neste capítulo será apresentado os materiais e metodologias utilizados neste trabalho.

3.1 Materiais

Para o estudo teórico e para a implementação das técnicas de restauração, necessita-se de um microcomputador e de aplicativos para simulação computacional, como o Audacity (gratuito) e o Matlab®, que possui versões de baixo custo para estudantes.

3.2 Metodologia

Para se atingir as metas descritas neste trabalho, propõe-se o seguinte conjunto de atividades:

3.2.1 Revisão Bibliográfica:

Nesta fase, inicialmente, foram estudados alguns fundamentos de áudio, de processamento digital e de implementação no Matlab. Alguns tópicos que foram considerados são: som (características e medição), noções de acústica e percepção, sinais de áudio (captação, armazenamento, amostragem e quantização, manipulação e processamento), processamento digital (amplificação, compressão, limitação, espectro, filtragem).

3.2.1.1 Estudo das principais técnicas de redução de ruído:

Nesta etapa, foi feito um levantamento bibliográfico com o objetivo de determinar as técnicas mais promissoras. Deve-se salientar que será dada prioridade a técnicas dotadas de fundamentação teórica, evitando-se soluções ad hoc ou métodos heurísticos frequentemente adotados no desenvolvimento de produtos.

3.2.1.2 Estudo das técnicas escolhidas:

Nesta fase, foram estudadas mais detalhadamente as técnicas escolhidas (Subtração Espectral e Transformada *Wavelet*).

3.2.2 Coleta e Digitalização de amostras de Áudio

Foram utilizadas duas fitas de rolo, datadas de 1968 e 1969. Elas foram gravadas na Itália e enviadas por meio de um Padre que estava a caminho de São Paulo para um trabalho com a comunidade local. Tinham como objetivo a atualização das notícias de familiares separados por mais de 10 anos, mas também passam um relato histórico de como era a vida de famílias como essa na Itália do final de 1960, com seus problemas e a recuperação da economia no país. A Figura 3 apresenta os dois rolos de fita utilizados.



Figura 3 – Fitas de rolo que foram digitalizadas.

Para digitalizá-las foi utilizado um gravador de fitas de rolo Akai 4000DS MKII e um microcomputador rodando o software gratuito Audacity. A Figura 4 mostra o sistema em funcionamento.

3.2.3 Início da simulação em computador:

Após a seleção das duas técnicas (Subtração Espectral e Transformada *Wavelet*), foi iniciada a implementação destas no Matlab.

3.2.3.1 Subtração Espectral

Seguindo os parâmetros e fundamentos descritos na Subseção 2.1.3, iniciou-se a programação do código no Matlab 2015.

Como boa prática é recomendado iniciar todo código em matlab com:



Figura 4 – Sistema de digitalização das fitas de rolo.

```
clc;  
clear all;  
close all;
```

Na sequência temos a adição de caminhos dos arquivos auxiliares que foram usados no projeto e a definição dos parâmetros necessários para rodar o código:

```
SNR=20 %in dB
```

```
infile='\SI1217_16k.wav'; % clean speech file
```

```
LB=512 % block length
```

```
denoise_method = 'subtEspec' %<-- 'none', 'wavelet', 'subtEspec'  
type='none' %<-- 'none', 'coif5', 'db10'
```

A variável *SNR* recebe a relação sinal ruído com a qual deseja-se criar o ruído gaussiano que irá ser adicionado ao arquivo de áudio limpo, que é carregado pelo caminho salvo na variável *infile*. O processamento dos sinais é feito em bloco, para otimização e *LB* recebe o tamanho que esses blocos devem ter. O método utilizado é definido pela variável *denoise_method*, onde devemos escolher o valor *subtEspec* para filtrarmos por

Subtração Espectral. `type` recebe o tipo de transformada *wavelet*, o que não vem ao caso no momento.

Então é feita a importação do arquivo de áudio limpo para a simulação. No caso um arquivo `.wav` de uma gravação de áudio com a frase de Platão: *"Only those storytellers will remain who can imitate the style of the virtuous"*.

```
[input,fs] = audioread(infile);
```

A próxima etapa é a normalização do sinal (limitando a amplitude do sinal entre -1 e 1), através do código:

```
[l,ch]=size(input);
input_norm=input;
for i=1:ch
    input_ch=input_norm(:,i);
    mx=max(abs(input_ch));
    input_ch=input_ch/mx;
    input_norm(:,i)=input_ch;
end
```

O sinal então recebe a adição de ruído branco gaussiano.

```
noisy_g=awgn(input_norm,SNR,'measured');
```

Então o sinal ruidoso é dividido em blocos de 512 amostras (definido pela variável `LB`). Como o último bloco provavelmente não terá o mesmo tamanho, é necessário completá-lo com zeros. Cada bloco passa por um filtro onde é feita a subtração espectral. Todos os blocos são empilhados e ao final os zeros que foram adicionados são removidos.

```
R=rem(n,LB);
input_norm0=noisy_u;
if R~=0
    input_norm0=[noisy_u;zeros(LB-R,1)];
end
output_se_u=[];
for k=1:LB:numel(input_norm0)
    bloco=input_norm0(k:k+L-1);
    bloco_out=basspec(bloco,0.5);
    output_se_u=[output_se_u;bloco_out];
end
```

```
end
output_se_g = output_se_g(1:n);
```

O filtro criado necessita de dois parâmetros. O primeiro é o sinal a ser filtrado, já o segundo é o parâmetro de ruído. O filtro foi criado, de acordo com o que foi tratado na subseção 2.1.3, como se segue:

```
function fn = basspec(f,D)
F=fft(f);
P=angle(F);
A=abs(F);
An=A-D;
Fn=An.*exp(i*P);
f1=ifft(Fn);
fn=real(f1);
end
```

Finalmente, o sinal filtrado é normalizado e são salvos o sinal sujo e o filtrado como arquivos .wav, além de imagens de cada um no tempo, frequência e seus respectivos espectrogramas.

3.2.3.2 Transformada *Wavelet*

Para a redução de ruído usando a transformada *Wavelet*, o código é o mesmo que o desenvolvido na subseção anterior, diferindo apenas nos parâmetros e na etapa de divisão em blocos e filtragem.

Os parâmetros devem ser definidos da seguinte forma:

```
SNR=20 %in dB

infile='\SI1217_16k.wav'; % clean speech file

LB=512 % block length

denoise_method = 'wavelet' %<-- 'none', 'wavelet', 'subtEspec'
type='none' %<-- 'none', 'coif5', 'db10'
```

O `denoise_method` deve receber `wavelet` e `type` pode receber tanto `coif5` quanto `db10`, que implementam, respectivamente, as transformadas *wavelet* Coiflet e Daubechies,

pois apresentam o melhor resultado para arquivos de áudio, devido à suas características quase simétrica, ortogonal e biortogonal (VILLANUEVA-LUNA et al., 2011).

Na etapa de filtragem, o sinal ruidoso também é dividido em blocos de LB amostras. Como o último bloco provavelmente não terá o mesmo tamanho, é necessário completá-lo com zeros.

```
R=rem(n, LB);
input_0=noisy_g;
if R~=0
    input_0=[noisy_g;zeros(LB-R,1)];
end

wname = type;
lev = 10;

output_w_g=[];
for k=1:L:numel(input_0)
    bloco=input_0(k:k+LB-1);
    tree = wpdec(bloco,lev,wname);
    det1 = wpccoef(tree,2);
    sigma = median(abs(det1))/0.6745;
    alpha = 2;
    thr = wpbmpen(tree,sigma,alpha);
    keepapp = 1;
    bloco_out = wpdencmp(tree,'s','nobest',thr,keepapp);
    output_w_g=[output_w_g;bloco_out];
end
output_w_g = output_w_g(1:n);
```

Para realizar a decomposição do sinal, como descrito em 2.1.5, é utilizada a função `wpdec` que deve receber o sinal a ser decomposto e dois outros parâmetros. A variável `wname` recebe o tipo de transformada *Wavelet* que será usada. Já `lev` recebe o nível de *Wavelet* que será implementado (quantas "quebras" a árvore de decomposição terá), foi escolhido o nível 10 pois apresenta uma boa relação entre resultado e tempo de processamento.

A variável `det1` recebe os coeficientes de decomposição da árvore criada, através do comando `wpccoef`. O parâmetro 2 se refere ao número do nó de melhor performance da árvore, no caso 2.

Para a determinação do limiar de eliminação dos coeficientes (`thr`) é utilizada a função `wpbmpen`. Tal função necessita da árvore gerada, `sigma` (desvio padrão do ruído

branco Gaussiano adicionado ao sinal, obtido pela análise da média dos coeficientes de decomposição) e α (parâmetro de afinação, normalmente recebe o valor 2).

Finalmente, com todos os parâmetros definidos, podemos fazer a filtragem do sinal com o comando `wpdencmp`. Além da árvore gerada, a função exige '`s`' (*soft thresholding*, '`nobest`' (para que não sejam feitas otimizações), o `thr` calculado e `keepapp=1` significa que os coeficientes de aproximação não devem ser descartados (caso contrário, serão).

Novamente, todos os blocos são empilhados e ao final os zeros que foram adicionados são removidos.

3.2.4 Verificação dos programas de restauração de sinais de áudio:

Neste ponto, foram adotadas medidas objetivas ([BEERENDS, 2002](#)) como a Relação Sinal Ruído Segmentada e a PESQ, através da função:

```
[SNR_in, SegSNR_in, LSD_in, PESQ_in]=mede_aes2012
(input_norm,noisy_g,128,fs);
[SNR_out, SegSNR_out, LSD_out, PESQ_out]=mede_aes2012
(input_norm,out_g_norm,128,fs);
```

A função que calcula a SNR_{seg} foi desenvolvida pelo Prof. Dr. Irineu Antunes Junior e a função PESQ pelos pesquisadores Yi Hu e Philipos C. Loizou. Os argumentos de entrada são, respectivamente, o sinal original (limpo), o sinal a ser testado, o tamanho dos blocos de cálculo da SNR_{seg} e a frequência do sinal.

3.2.5 Teste com Dados Reais

Para o teste com dados reais o algoritmo descrito foi utilizado para filtrar o áudio capturado na Subseção [3.2.2](#), demonstrando um caso de aplicação real.

4 Resultados e Discussão

Neste capítulo serão descritos e discutidos todos os resultados alcançados no desenvolvimento deste projeto.

As figuras 5, 6 e 7 apresentam a forma de onda do sinal de entrada (limpo) no domínio do tempo, da frequência e seu espectrograma, respectivamente.

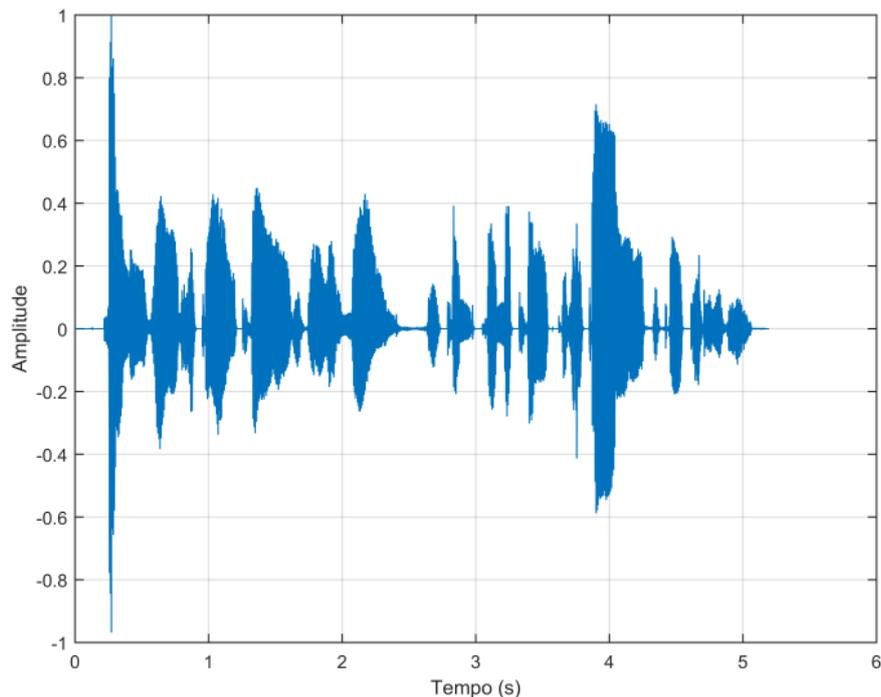


Figura 5 – Forma de onda do áudio de entrada no domínio do tempo.

Após a adição de ruído, o sinal pode ser visto nas figuras 8, 9 e 10.

Como pode ser visto, na Figura 5, do sinal limpo, a amplitude do sinal em momentos de silêncio é mais baixa, em comparação com a amplitude, nos mesmos momentos, do sinal ruidoso (Figura 8).

4.1 Subtração Espectral

Usando o método de Subtração Espectral, após a filtragem, foi obtido o resultado apresentado nas figuras 11, 12 e 13.

Na Figura 11 é possível perceber que não houve uma grande diminuição na amplitude do sinal em momentos de silêncio. Também pode-se observar pouca diferença nos

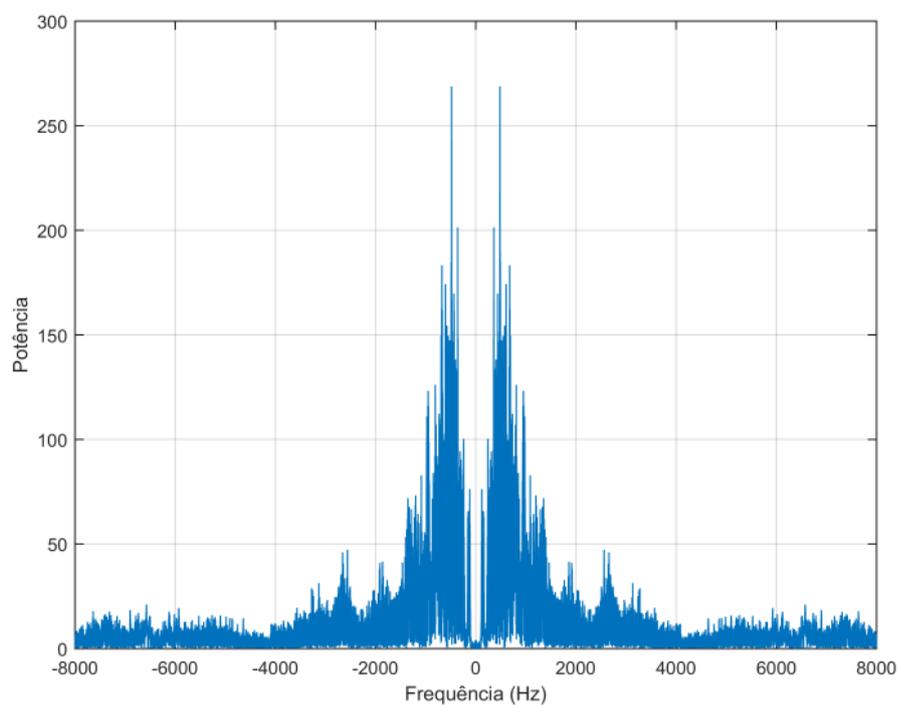


Figura 6 – Forma de onda do áudio de entrada no domínio da frequência.

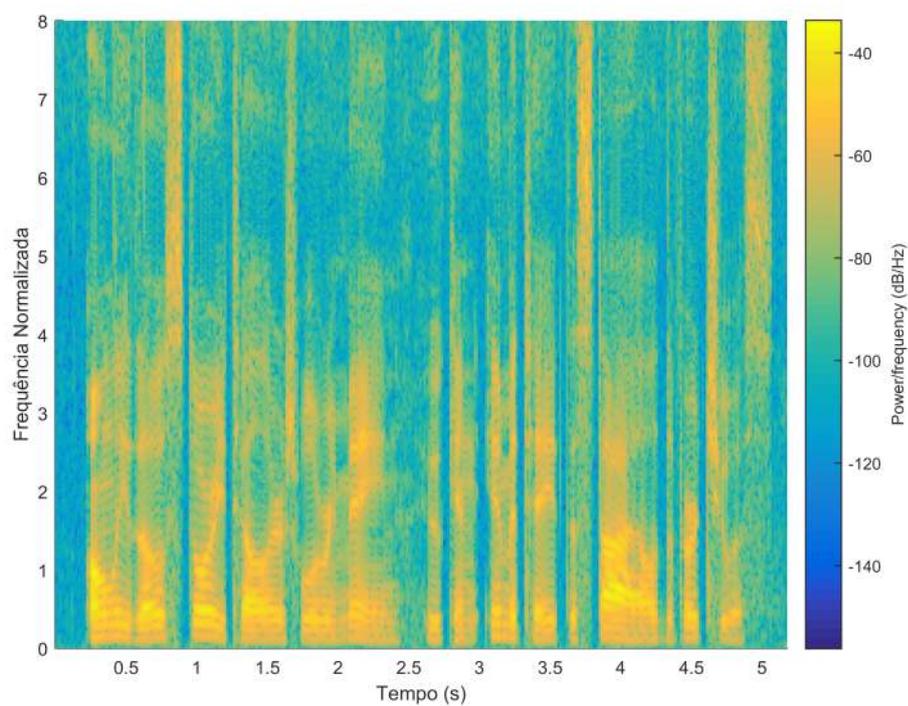


Figura 7 – Espectrograma do áudio de entrada.

espectrogramas do sinal ruidoso e filtrado (10 e 13). Ambos apresentam sinais do ruído branco adicionado, comprovado pela distribuição de potência mais alta em quase todas

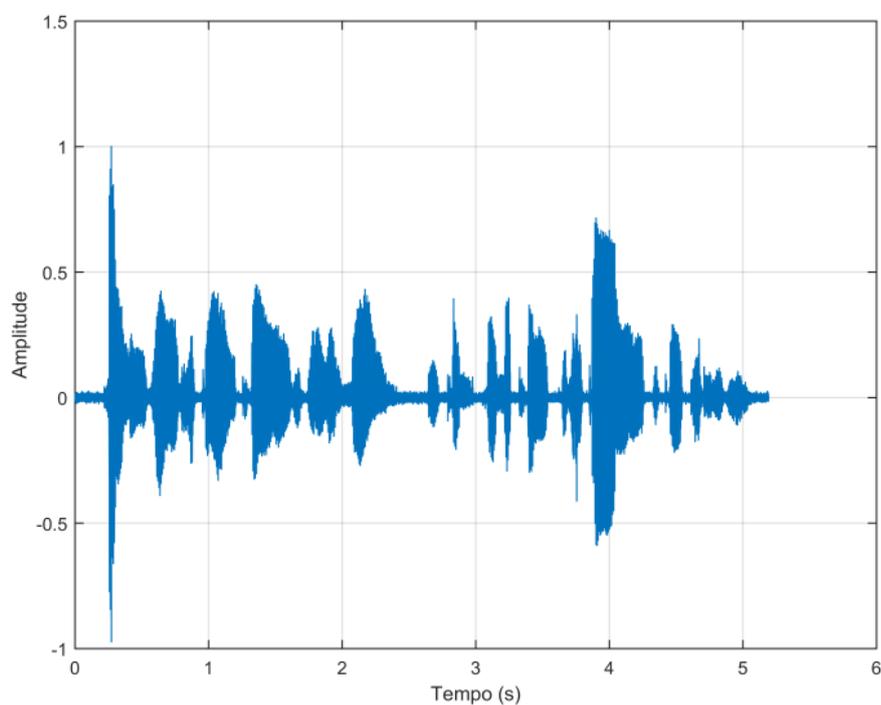


Figura 8 – Forma de onda do áudio ruidoso no domínio do tempo.

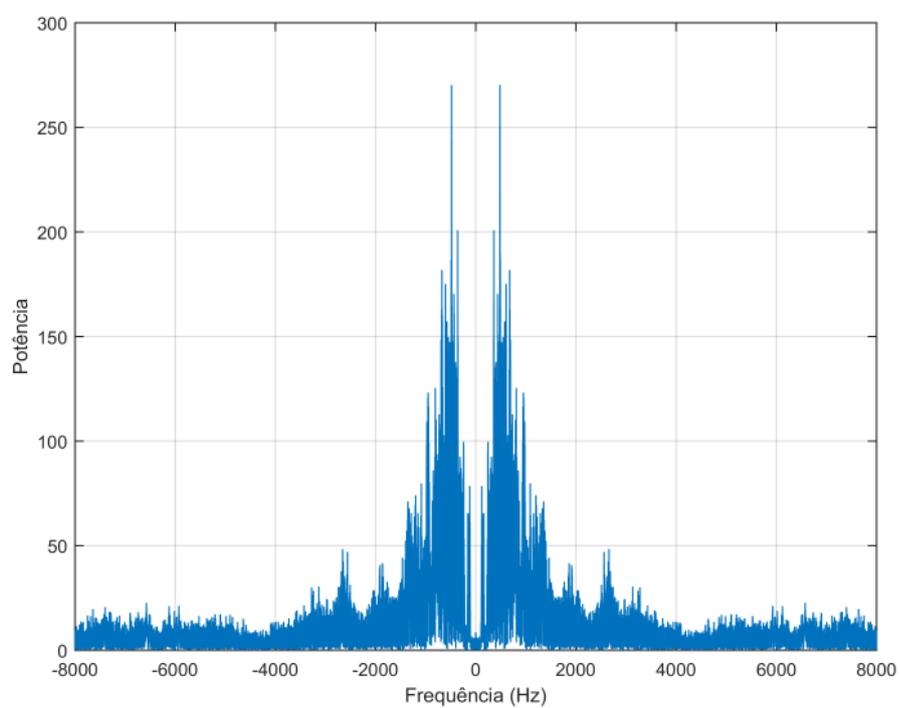


Figura 9 – Forma de onda do áudio ruidoso no domínio da frequência.

as frequências, diferente do sinal limpo (Figura 7 em que está mais claro quais são as frequências dominantes).

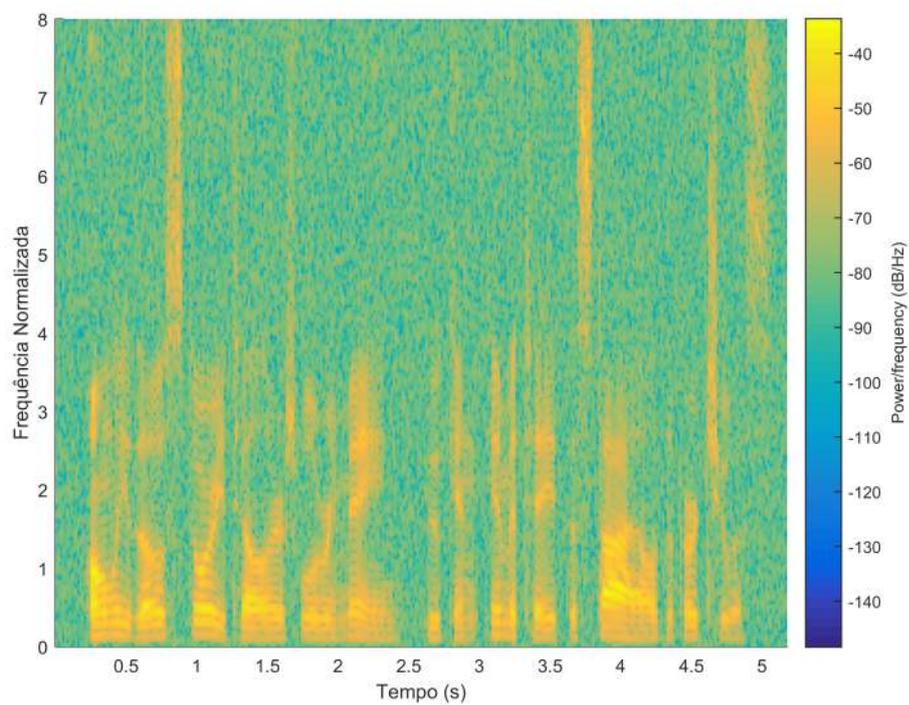


Figura 10 – Espectrograma do áudio ruidoso.

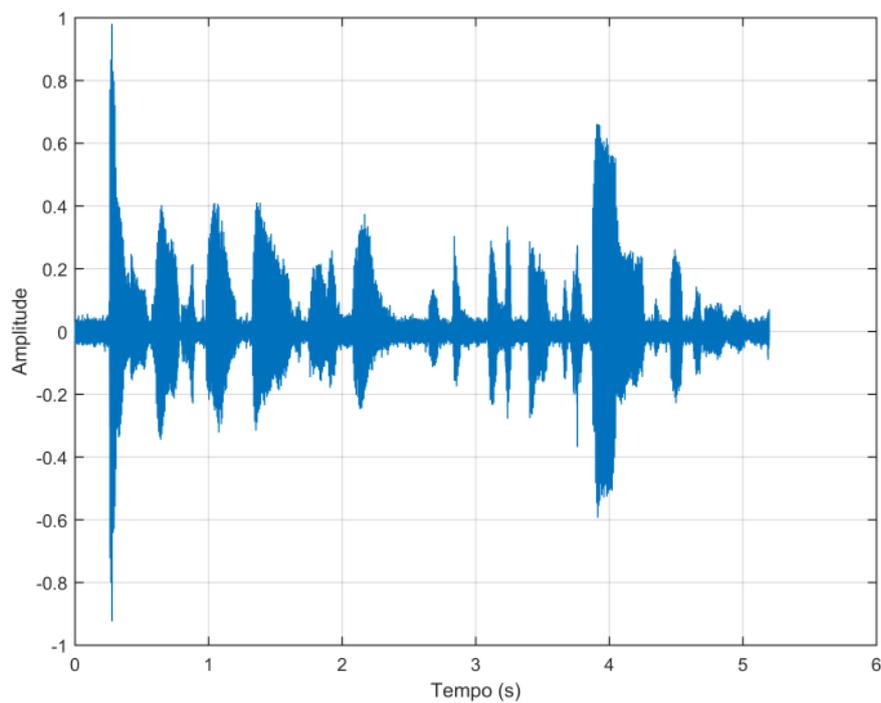


Figura 11 – Forma de onda do áudio filtrado por Subtração Espectral no domínio do tempo.

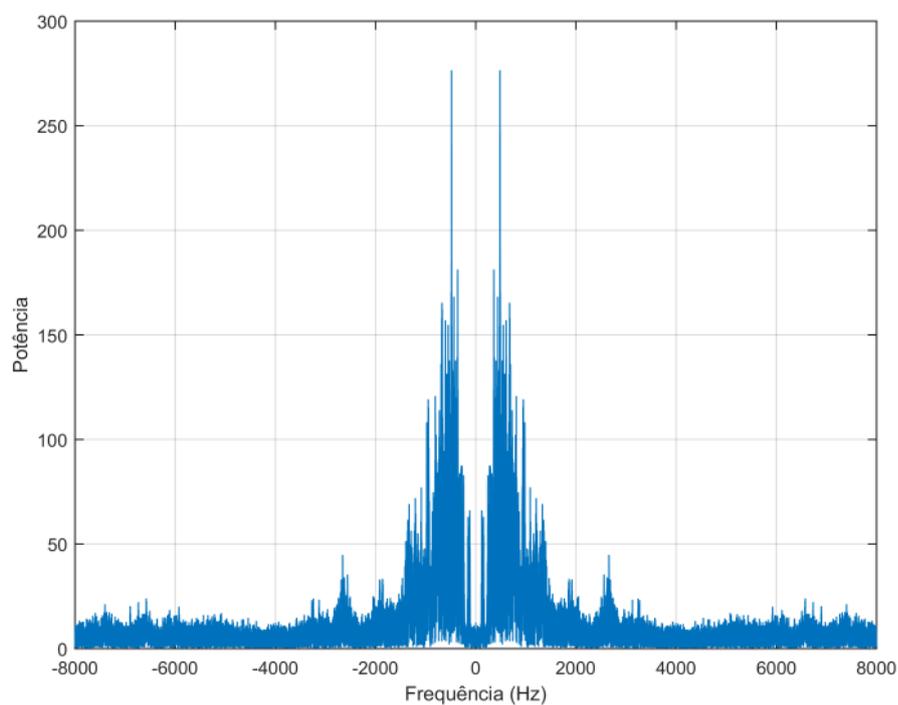


Figura 12 – Forma de onda do áudio filtrado por Subtração Espectral no domínio da frequência.

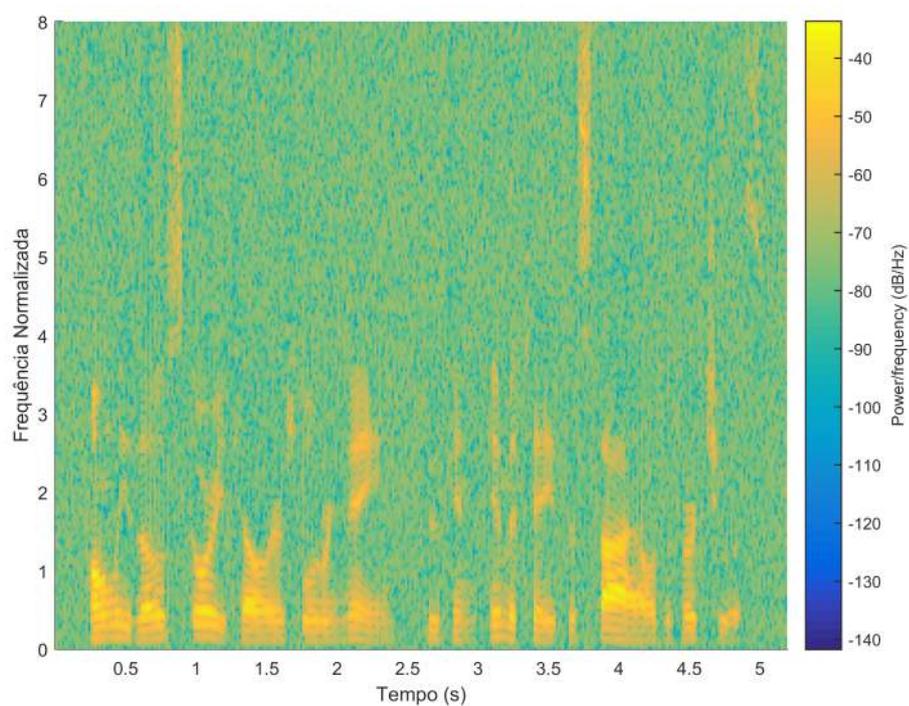


Figura 13 – Espectrograma do áudio filtrado por Subtração Espectral.

Usando as medidas definidas na subseção 2.2, chegou-se ao resultado da Tabela 2. Tabela 2 – Resultados quantitativos do sinal filtrado pelo método de Subtração Espectral.

	SNR_{seg}	PESQ
Sinal Ruidoso	8.84 dB	3,14
Sinal Filtrado	1.49 dB	2,49

Finalmente, ouvindo o áudio do sinal ruidoso e do sinal filtrado percebeu-se uma leve melhora, porém o "chiado" característico do ruído branco ainda é bastante perceptível.

4.2 Transformada *Wavelet*

O método de Transformada *Wavelet* foi implementado com dois tipos diferentes de transformada. Seus resultados são apresentados na sequência.

4.2.1 Coiflet

Após a filtragem, usando a transformada Coiflet, foi obtido o resultado apresentado nas figuras 14, 15 e 16.

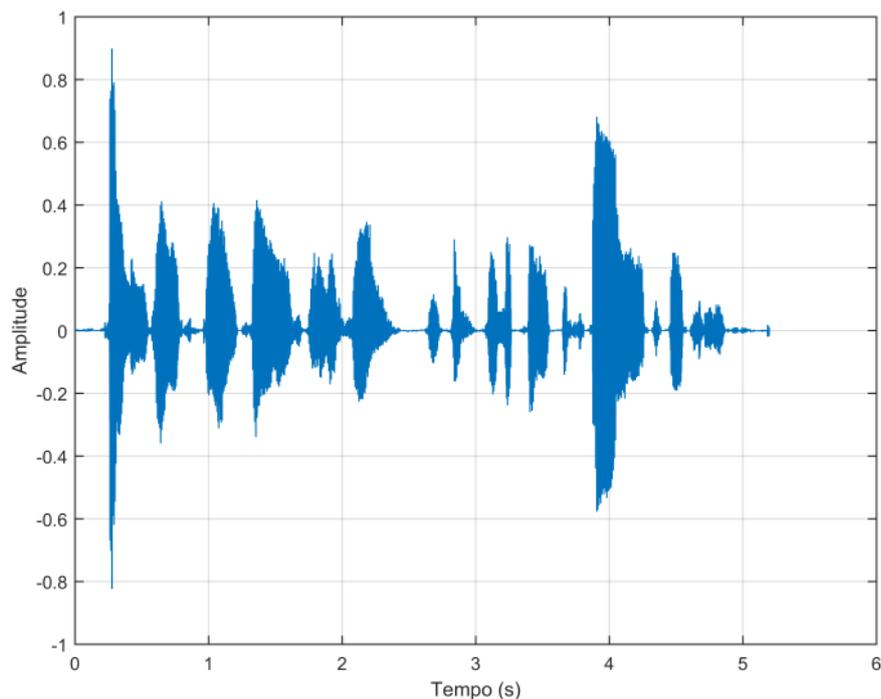


Figura 14 – Forma de onda do áudio filtrado por Transformada Coiflet no domínio do tempo.

Novamente, se compararmos a Figura 14 a Figura 8 é possível perceber que, desta vez houve uma significativa diminuição na amplitude do sinal em momentos de silêncio.

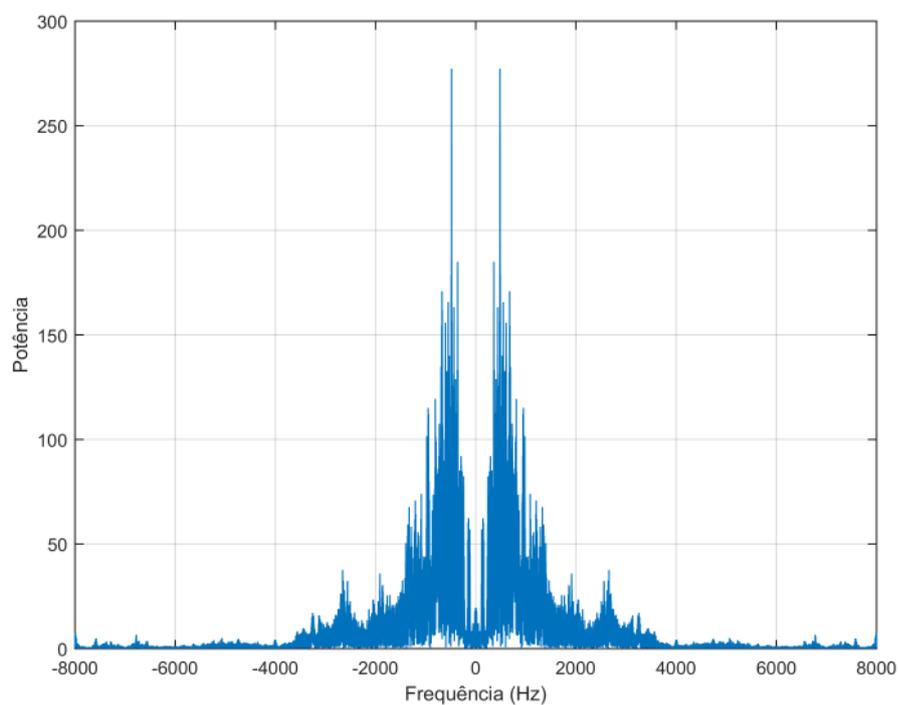


Figura 15 – Forma de onda do áudio filtrado por Transformada Coiflet no domínio da frequência.

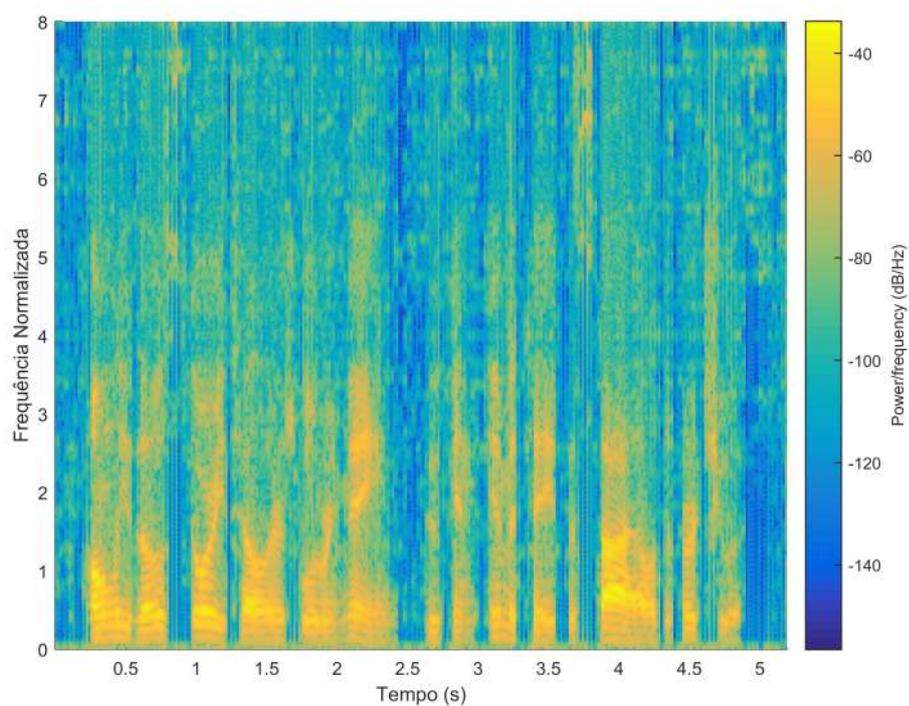


Figura 16 – Espectrograma do áudio filtrado por Transformada Coiflet.

Também pode-se observar uma maior diferença nos espectrogramas do sinal ruidoso e filtrado (10 e 16), aproximando o espectrograma do filtrado ao do sinal original.

Ouvindo o áudio do sinal filtrado percebeu-se uma melhora no "chiado" característico do ruído branco, porém foi possível perceber outro efeito que incomoda ao ouvir. Esse efeito é chamado de "ruído musical". É causado pela largura de banda estreita do espectro do sinal e caracterizado pelo audível "batido metálico" (ANTUNES, 2006).

Usando as medidas definidas na subseção 2.2, chegou-se ao resultado da Tabela 3.

Tabela 3 – Resultados quantitativos do sinal filtrado pelo método de Transformada *Wavelet* (coif5).

	SNR_{seg}	PESQ
Sinal Ruidoso	8.89 dB	3,16
Sinal Filtrado	7.29 dB	2,85

4.2.2 Daubechies

O filtro projetado com a transformada de Daubechies, resultou no sinal apresentado nas figuras 17, 18 e 19.

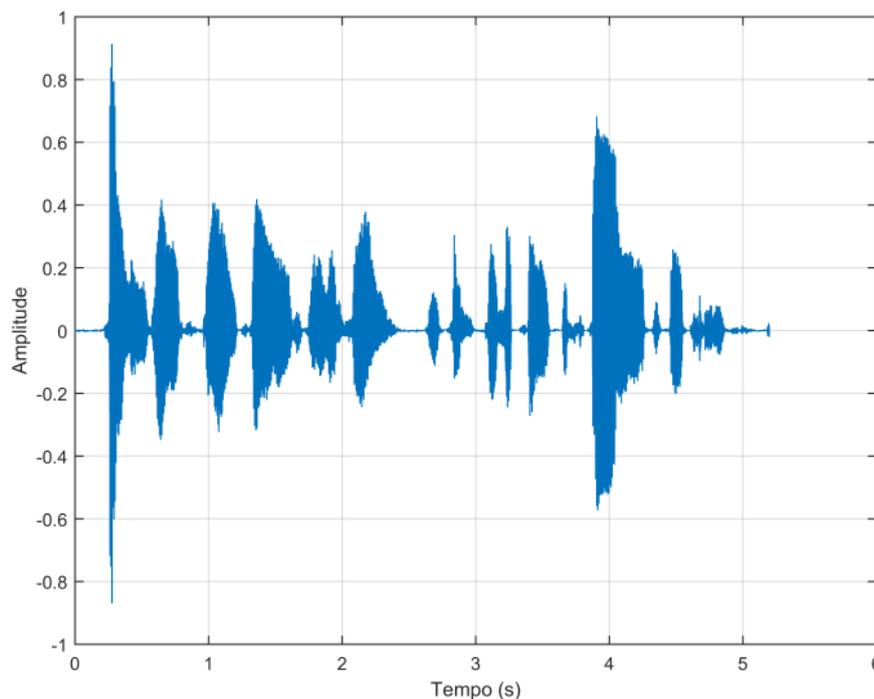


Figura 17 – Forma de onda do áudio filtrado por Transformada de Daubechies no domínio do tempo.

Os resultados observados de forma qualitativa foram bastante similares aos alcançados pelo filtro com a transformada tipo Coiflet. Apresentando as mesmas características

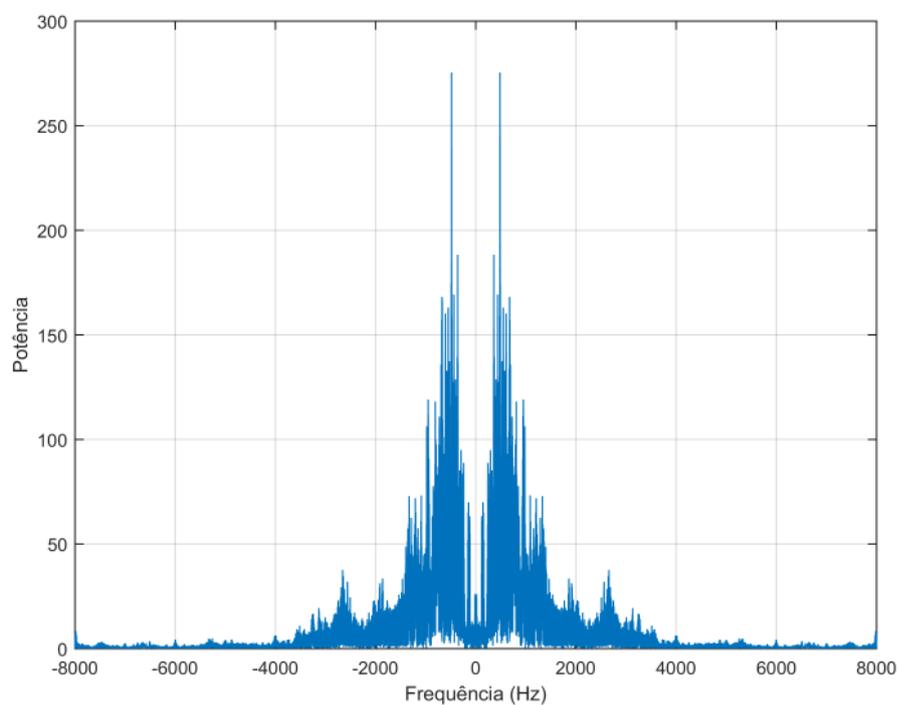


Figura 18 – Forma de onda do áudio filtrado por Transformada de Daubechies no domínio da frequência.

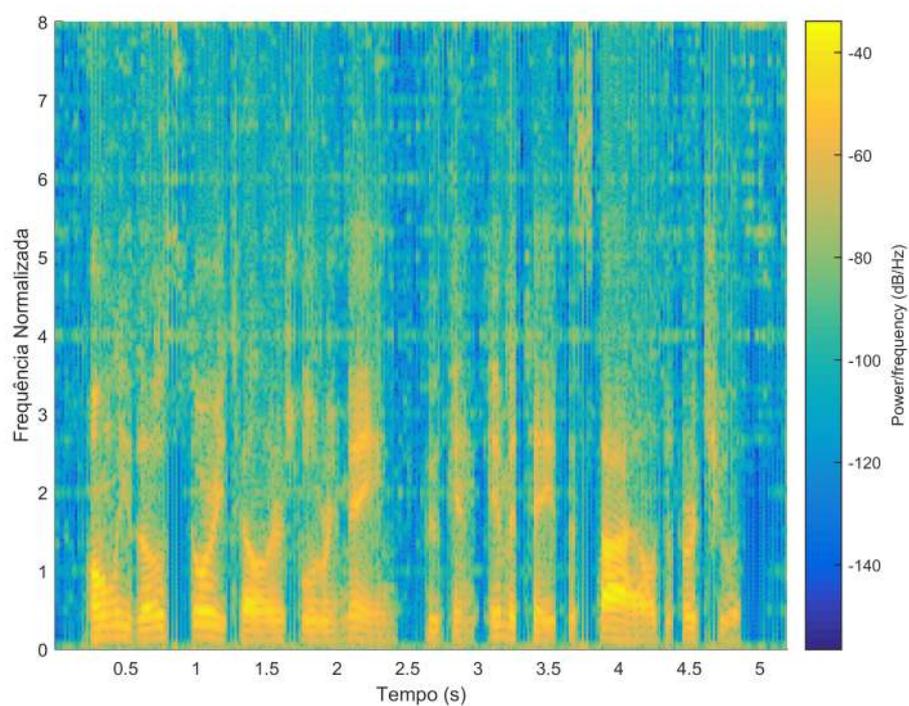


Figura 19 – Espectrograma do áudio filtrado por Transformada de Daubechies.

de redução de ruído e o efeito do ruído metálico. Também apresentou significativa redução de amplitude nos momentos de silêncio e espectrograma próximo ao original.

Usando as medidas definidas na subseção 2.2, chegou-se ao resultado da Tabela 4.

Tabela 4 – Resultados quantitativos do sinal filtrado pelo método de Transformada *Wavelet* (db10).

	SNR_{seg}	PESQ
Sinal Ruidoso	8.85 dB	3,12
Sinal Filtrado	7.50 dB	2,97

4.3 Discussão dos Resultados Quantitativos

A Tabela 5 apresenta os resultados quantitativos de todos os métodos proposto.

Tabela 5 – Resultados quantitativos de todos os métodos propostos.

	Subtração Espectral		Coiflet		Daubechies	
	SNR_{seg}	PESQ	SNR_{seg}	PESQ	SNR_{seg}	PESQ
Sinal Ruidoso	8.84 dB	3,14	8.89 dB	3,16	8.85 dB	3,12
Sinal Filtrado	1.49 dB	2,49	7.29 dB	2,85	7.50 dB	2,97

Através da Tabela 5 é possível verificar que nenhum dos métodos conseguiu gerar um PESQ melhor que o arquivo ruidoso, o que demonstra que essa medida não é a melhor indicada para sinais desse tipo. Mesmo assim, temos o método de Transformada *Wavelet* tipo Daubechies com o melhor resultado no quesito.

Também para a SNR_{seg} , nenhuma sinal filtrado superou o resultado do sinal ruidoso. Ainda assim, é possível destacar o método de subtração espectral que obteve resultado consideravelmente pior.

4.4 Áudio Lina.wav

Finalmente, o trecho do sinal de áudio coletado das fitas de rolo foi processado através do filtro com o melhor resultado agregado, o filtro por Transformada *Wavelet* tipo Daubechies. As figuras 20, 21 e 22 apresentam o sinal original, já as figuras 23, 24 e 25 mostram o sinal filtrado.

Assim como na Seção 4.2, o sinal apresentou visível melhora nos momentos de silêncio e um espectrograma mais concentrado em frequências menores (caso da voz humana).

Como não existe o sinal sem ruído, não foi possível a geração das medidas quantitativas, porém, ao ouvir o áudio é possível perceber uma melhora significativa. O chiado de fundo é quase imperceptível, assim como não é audível o ruído musical.

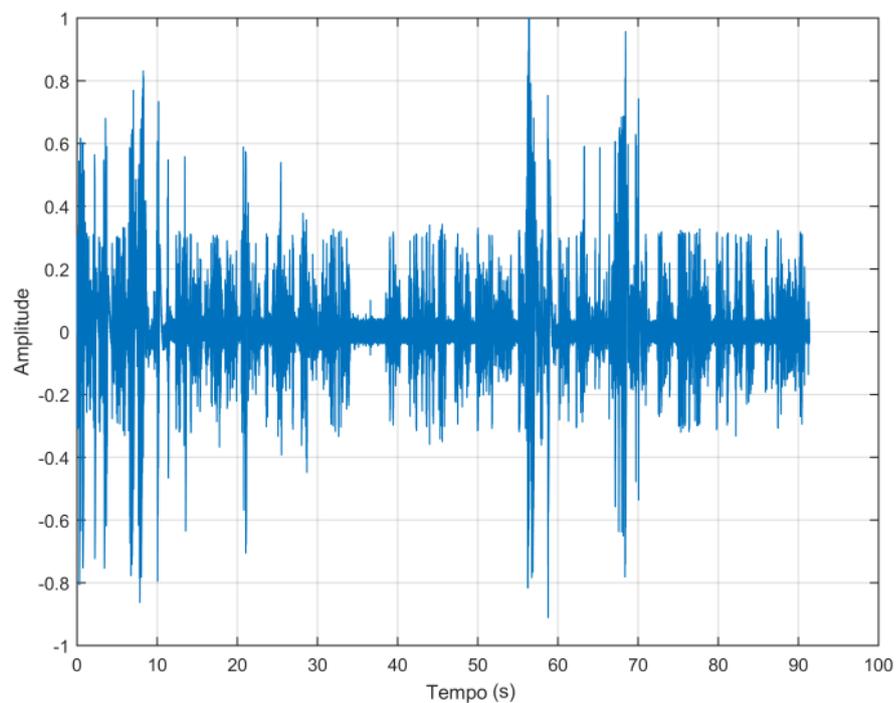


Figura 20 – Forma de onda do áudio "Lina.wav" no domínio do tempo.

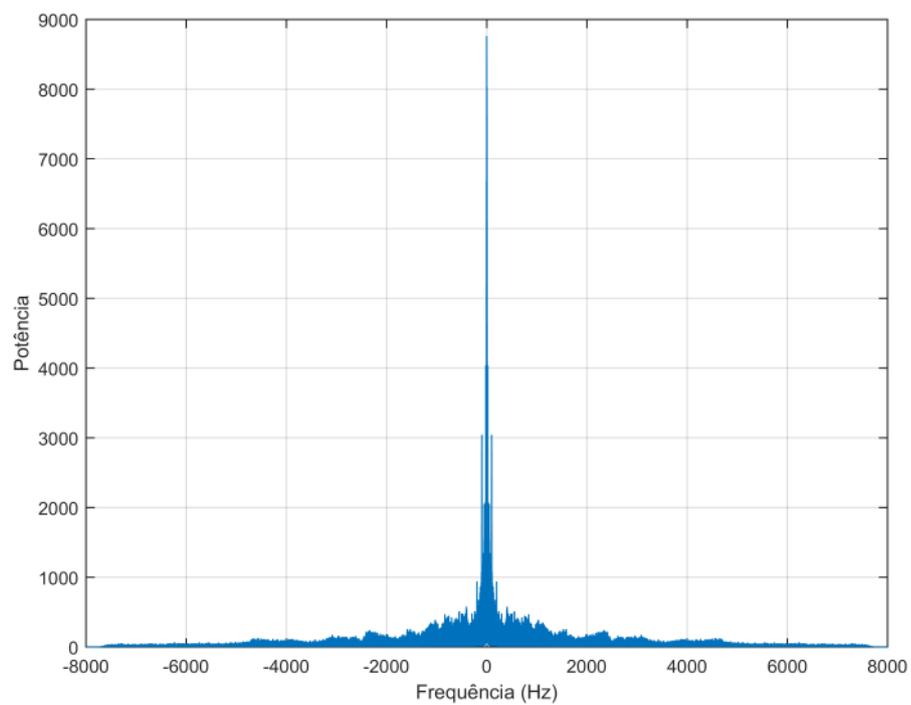


Figura 21 – Forma de onda do áudio "Lina.wav" no domínio da frequência.

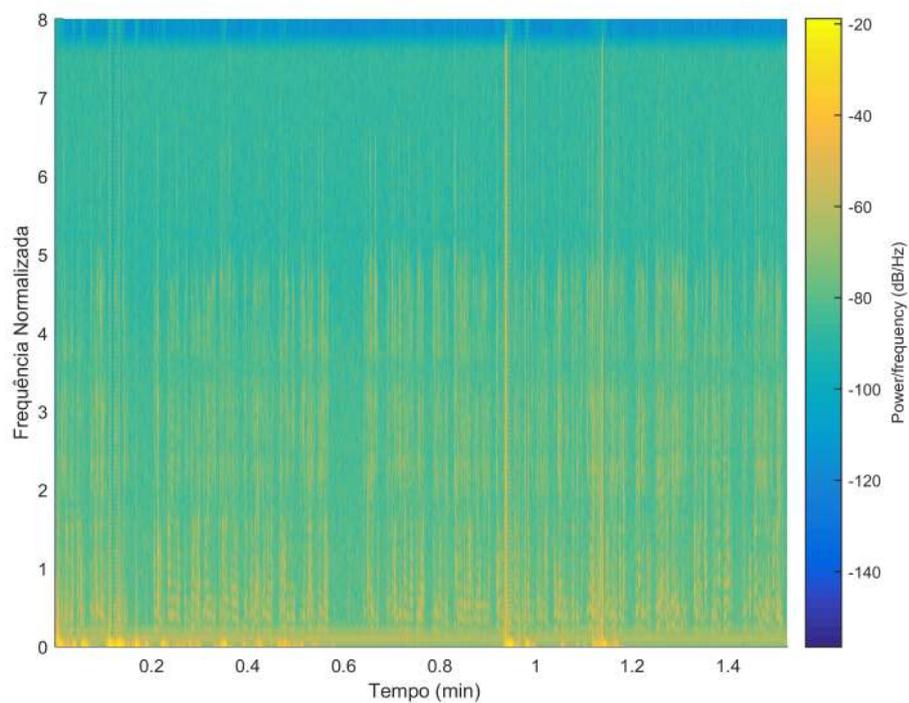


Figura 22 – Espectrograma do áudio "Lina.wav".

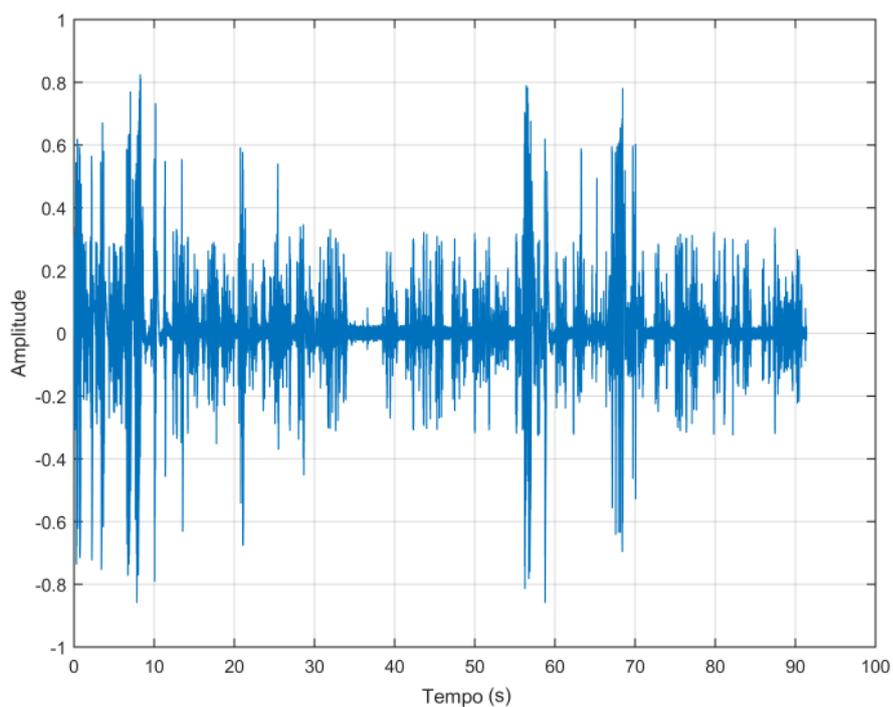


Figura 23 – Forma de onda do áudio "Lina.wav" filtrado por Transformada *Wavelet* tipo Daubechies no domínio do tempo.

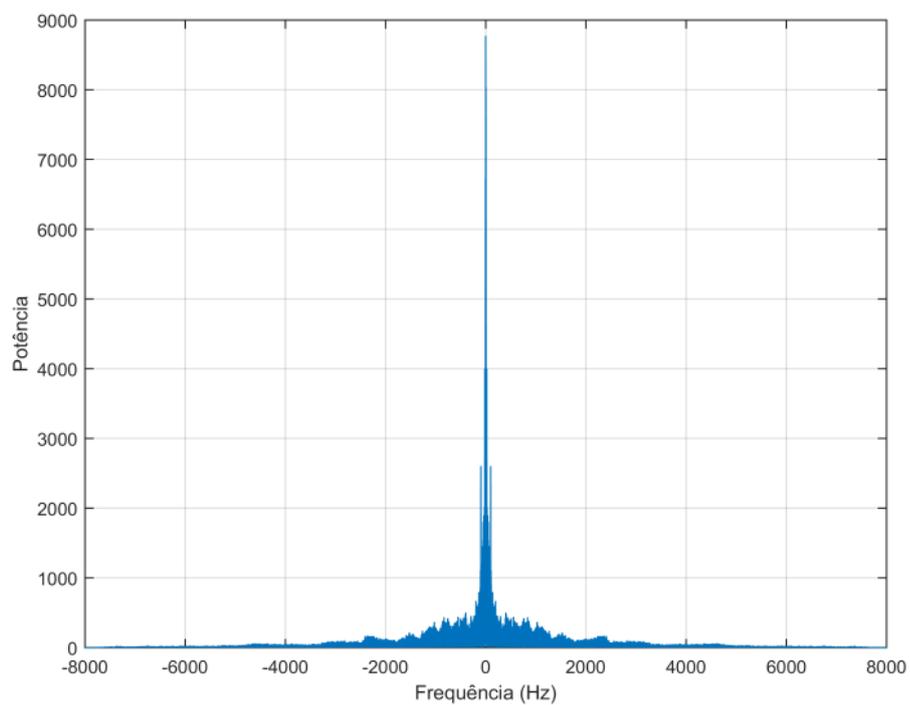


Figura 24 – Forma de onda do áudio "Lina.wav" filtrado por Transformada *Wavelet* tipo Daubechies no domínio da frequência.

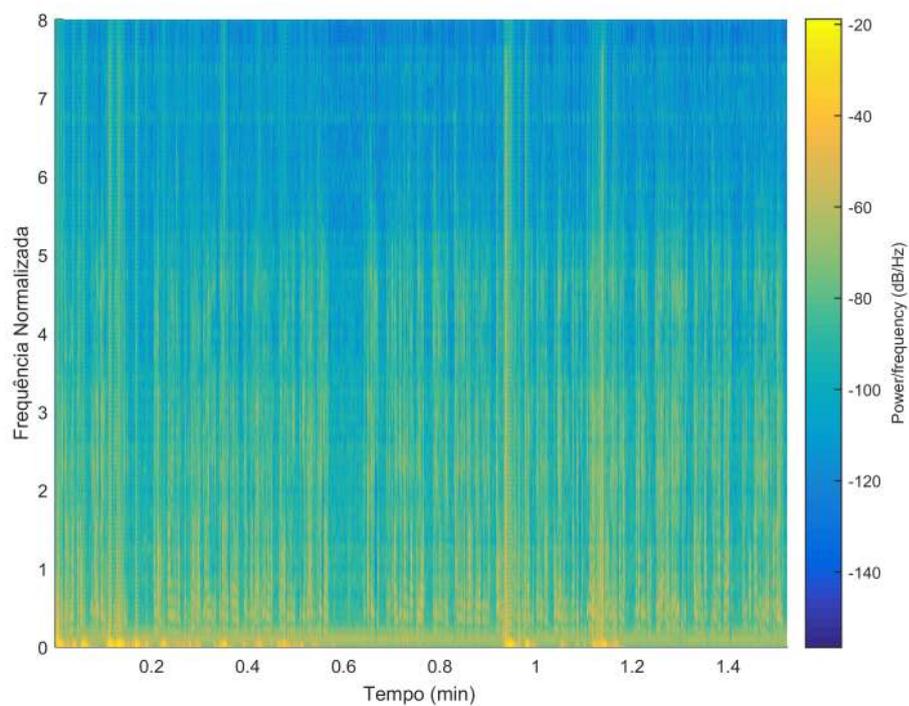


Figura 25 – Espectrograma do áudio "Lina.wav" filtrado por Transformada *Wavelet* tipo Daubechies.

Conclusão

Neste trabalho pode-se concluir que, dos métodos estudados, o método com melhor resultado foi o da Transformada *Wavelet* tipo Daubechies. Porém nos dois casos de Transformada *Wavelet* foi verificado o surgimento de ruído musical, comprometendo a qualidade do áudio. Também foi possível notar pequenas modificações nas características da voz, mesmo não atrapalhando a inteligibilidade.

Uma forma de atenuar os efeitos do ruído musical seria analisar o espectro de frequência de trechos onde o efeito é mais relevante, procurando por características (como a sua banda estreita) e após identificados, removê-los.

Também foi possível concluir que as medidas quantitativas usadas estão abaixo da expectativa, falhando em prover o resultados satisfatórios, se comparados ao nosso sistema psicoacústico. Este último também não pode ser utilizado sozinho devido à relatividade de seus resultados, que variam de indivíduo a indivíduo.

Além disso, com o teste prático, de um áudio gravado em fitas de rolo e digitalizado, concluiu-se que o método da Transformada *Wavelet* tipo Daubechies foi bastante satisfatório, eliminando o chiado de fundo. Além disso, alguns filtros poderiam ser aplicados em baixíssimas frequências (até 50Hz) e altas frequências (maiores que 3,5kHz) para atenuar também os *clicks* e batidas capturadas no áudio.

Referências

- ANTUNES, I. J. *Redução de ruído em sinais de voz usando curvas especializadas de modificação dos coeficientes da transformada em co-seno*. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, Departamento de Telecomunicações e Controle, 2006. Citado 2 vezes nas páginas 9 e 33.
- BEERENDS, J. G. "Audio Quality Determination Based on Perceptual Measurement Techniques", in *Applications of Digital Signal Processing to Audio and Acoustics*. 1. ed. [S.l.]: Kluwer Academic Publishers, Boston, USA, 2002. 1–38 p. Citado na página 25.
- BEROUTI, M.; SCHATZ, R.; MAKHOUL, J. Enhancement of speech corrupted by acoustic noise. *Proc. IEEE Int. conf. Acoust., Speech, Signal Process*, p. 208–211, 1979. Citado na página 14.
- ESQUEF, P. A. A.; BISCAINHO, L. W. P. An efficient model-based method for reconstruction of audio signals across long gaps. *IEEE Trans. Audio, Speech, and Language Processing*, v. 14, p. 1391–1400, 2006. Citado na página 9.
- FERREIRA, R. D.; JÚNIOR, I. A. Restauração digital de gravações de Áudio: Técnicas para combater o ruído. *Ficha de Inscrição para PDPD – Edital 01/2011*, 2011. Citado 3 vezes nas páginas 5, 9 e 10.
- GODSILL, S.; RAYNER, P.; CAPPÉ, O. "Digital Audio Restoration" in *Applications of Digital Signal Processing to Audio and Acoustics*. 1. ed. [S.l.]: Kluwer Academic Publishers, Boston, USA, 2002. 133–194 p. Citado 4 vezes nas páginas 4, 9, 12 e 13.
- GRUNDLEHNER, B. et al. Performance assessment method for speech enhancement systems. *Proc. 1st Annu. IEEE BENELUX/DSP Valley Signal Process. Symp.*, 2005. Citado na página 17.
- MISITI, M. et al. *Wavelet Toolbox For Use with MATLAB*. 1. ed. [S.l.]: The MathWorks, Inc, 1996. Citado 2 vezes nas páginas 4 e 16.
- ORIO, N. et al. Methodologies and tools for audio digital archives. In: . [s.n.], 2010. Disponível em: <http://users.dimi.uniud.it/~lauro.snidaro/Download/publications/Orio_Snidaro_Canazza_Foresti.pdf>. Citado na página 9.
- PENNOCK, S. Accuracy of the perceptual evaluation of speech quality algorithm. In: . [s.n.], S.I. Disponível em: <<http://wireless.feld.cvut.cz/mesaqin2002/full09.pdf>>. Citado na página 18.
- PRATA-SHIMOMURA, A. et al. *Transmissão Sonora*. 2017. Disponível em: <<http://www.fau.usp.br/arquivos/disciplinas/au/aut0278/Aulas/AUT%200278%20Aula%2007%20-%20Transmiss%C3%A3o%20Sonora.pdf>>. Citado na página 9.
- PUC-RIO. *Filtro de Kalman*. S.I. Disponível em: <https://www.maxwell.vrac.puc-rio.br/7604/7604_5.PDF>. Citado na página 15.
- ROADS, C. *The Computer Music Tutorial*. 1. ed. [S.l.]: MIT Press, 1996. Citado na página 9.

SILVA, L. A. da et al. Estudo comparativo entre o filtro de kalman de frequência discreta e o filtro de kalman de tempo discreto com aplicação na redução de ruído em sinais de voz. *XIV CEEL - Universidade Federal de Uberlândia*, 2016. Citado 2 vezes nas páginas 15 e 17.

SILVA, L. A. da et al. Aplicação da transformada wavelet na redução de ruído em sinais de voz. *XII CEEL - Universidade Federal de Uberlândia*, 2014. Citado 2 vezes nas páginas 16 e 17.

VILLANUEVA-LUNA, A. E. et al. De-noising audio signals using matlab wavelets toolbox. *InTech*, 2011. Citado na página 24.

APÊNDICE A – Código Final

Listing A.1 – Código Final.

```

clc;
clear all;
close all;

%----- Paths -----
Wdir = what;
addpath([Wdir.path '\subtEspec']);
addpath([Wdir.path '\AudioSamples']);
addpath([Wdir.path '\CommonTools']);
addpath([Wdir.path '\DenoiseMethods']);
addpath([Wdir.path '\pesq']);

%----- Adjustable parameters -----

SNR=20 %in dB

infile='\SI1217_16k.wav'; % clean speech file

LB=512 % block length

denoise_method = 'wavelet' %<— 'none', 'wavelet', 'subtEspec'
type='coif5' %<— 'none', 'coif5', 'db10'

%%%%%%%%%%
%SIGNAL

% Pegando o arquivo de audio limpo - "Only those story tellers
will remain who can imitate the style of the virtuous."
[input, fs]=audioread(infile);
[n, p]=size(input);

%%Normalizando
[l, ch]=size(input);
input_norm=input;

```

```

for ii=1:ch
    input_ch=input_norm(:,ii);
    mx=max(abs(input_ch));
    input_ch=input_ch/mx;
    input_norm(:,ii)=input_ch;
end

%%%%%%%%%%
%SIG+NOISE

noisy_g=awgn(input_norm,SNR,'measured');

%Normalizando o sinal com ru do para salvar
[l,ch]=size(noisy_g);
noisy_g_norm=noisy_g;
for ii=1:ch
    norm=noisy_g_norm(:,ii);
    mx=max(abs(norm));
    norm=norm/mx;
    noisy_g_norm(:,ii)=norm;
end

%Filtrando sinal com ru do

R=rem(n,LB);
input_0=noisy_g;
if R~=0
    input_0=[noisy_g;zeros(LB-R,1)];
end

if strcmp(denoise_method,'subtEspec')

    output_se_g=[];
    for k=1:LB:numel(input_0)
        bloco=input_0(k:k+LB-1);
        bloco_out=basspec(bloco,0.5);
        output_se_g=[output_se_g;bloco_out];
    end
    output_se_g = output_se_g(1:n);

```

```

        output_w_g = output_se_g;

elseif strcmp(denoise_method, 'wavelet')

    wname = type;
    lev = 10;

    output_w_g=[];
    for k=1:LB:numel(input_0)
        bloco=input_0(k:k+LB-1);
        tree = wpdec(bloco, lev, wname);
        det1 = wpccoef(tree, 2);
        sigma = median(abs(det1))/0.6745;
        alpha = 2;
        thr = wpbmpen(tree, sigma, alpha);
        keepapp = 1;
        bloco_out = wpdencmp(tree, 's', 'nobest', thr, keepapp);
        output_w_g=[output_w_g; bloco_out];
    end
    output_w_g = output_w_g(1:n);

end

%Normalizando o sinal filtrado para salvar
[l, ch]=size(output_w_g);
out_g_norm=output_w_g;
for ii=1:ch
    norm=out_g_norm(:, ii);
    mx=max(abs(norm));
    norm=norm/mx;
    out_g_norm(:, ii)=norm;
end

%%%%%%%%%
%RESULT

audiowrite('\Resultados\Ruidoso.wav', noisy_g_norm, fs);

```

```

audiowrite( '\Resultados\Filtrado.wav', out_g_norm, fs );

%Validando resultado com SNR segmentada

[SNR_in, SegSNR_in, LSD_in, PESQ_in]=mede_aes2012(input_norm,
noisy_g,128,fs);
[SNR_out, SegSNR_out, LSD_out, PESQ_out]=mede_aes2012(input_norm,
out_g_norm,128,fs);
fprintf( '\nSNR_in: %0.2f dB\nSegSNR_in: %0.2f dB\n
SNR_out: %0.2f dB\nSegSNR_out: %0.2f dB\n
LSD_in: %0.2f dB\nPESQ_in: %0.2f dB\n
LSD_out: %0.2f dB\nPESQ_out: %0.2f dB\n',
SNR_in, SegSNR_in, SNR_out, SegSNR_out,
LSD_in, PESQ_in, LSD_out, PESQ_out );

%----- Graphics -----

%%Exibe o sinal de audio normalizado
ExibeSinal(input_norm, fs, 1);
print -dpng -r300 '\Resultados\Orig_Temp.png'
ExibeSinal(input_norm, fs, 2);
print -dpng -r300 '\Resultados\Orig_Freq.png'
ExibeSinal(input_norm, fs, 3);
print -dpng -r300 '\Resultados\Orig_Espec.png'

%%Exibe os sinais de audio com ruído
ExibeSinal(noisy_g, fs, 1);
print -dpng -r300 '\Resultados\Ruidoso_Gau_Temp.png'
ExibeSinal(noisy_g, fs, 2);
print -dpng -r300 '\Resultados\Ruidoso_Gau_Freq.png'
ExibeSinal(noisy_g, fs, 3);
print -dpng -r300 '\Resultados\Ruidoso_Gau_Espec.png'

%%Exibe os sinais do audio filtrado
ExibeSinal(output_w_g, fs, 1);
print -dpng -r300 '\Resultados\Out_Temp.png'
ExibeSinal(output_w_g, fs, 2);
print -dpng -r300 '\Resultados\Out_Freq.png'

```

```
ExibeSinal(output_w_g, fs, 3);  
print -dpng -r300 '\Resultados\Out_Espec.png'
```