

**Supressão de ruído acústico no sinal de áudio da fala durante
exames de ressonância magnética**

Vinícius Pessoa Duarte
N^o USP: 8941043

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

MAC0499 - Projeto de conclusão de curso supervisionado

Curso: Ciência da Computação
Supervisor: Prof. Dr. Marcel Jackowski

São Paulo
Novembro, 2018

Resumo

Capturar a fala de um paciente durante uma sessão de ressonância magnética pode permitir uma mais fácil identificação dos momentos em que problemas na fala ocorrem, auxiliando no pareamento entre a manifestação física de problemas de fala e o formato do trato vocal. Com isso, pode se tornar mais fácil identificar a causa de desordens de fala apresentadas. Quando adquirindo esses áudios, diversos ruídos de interferência podem poluir a fala, como aqueles gerados pela movimentação do paciente, a interferência dos campos magnéticos gerados pelo *scanner* da máquina, ou o ruído acústico gerado pelas bobinas que produzem os campos magnéticos supracitados. Esses ruídos podem dificultar a compreensão da fala original, impondo a necessidade de supressão de ruído, o que permite melhor entendimento da fala do paciente. Este trabalho propõe a realização do estudo de diferentes técnicas para a supressão de ruído acústico, culminando na análise dos métodos estudados e no desenvolvimento de um software para a aplicação desses métodos em áudios semelhantes aos provenientes de sessões de ressonância magnética.

Sumário

1	Introdução e objetivos	4
2	Materiais e métodos	7
2.1	Áudios utilizados	7
2.2	Métodos de supressão	9
2.2.1	Subtração espectral	9
2.2.2	FLMS	11
2.2.3	Aprendizado de dicionários e análise <i>wavelet</i>	16
2.3	<i>Software</i>	19
2.3.1	Versão com IG	20
2.3.2	Versão de linha de comando	22
2.4	Testes comparativos	23
2.4.1	Teste de complexidade	23
2.4.2	Teste de qualidade	24
2.4.3	Comparação entre a amplitude da fala e do ruído	25
3	Resultados	27
3.1	Testes de complexidade	27
3.2	Espectros de energia	30
3.2.1	Ruídos	31
3.2.2	Subtração espectral	32
3.2.3	FLMS	32
3.2.4	Aprendizado de dicionários e análise <i>wavelet</i>	33
3.3	Diferença de energia	34
3.4	Comparação entre a amplitude da fala e do ruído	35
4	Conclusões	37
5	Apreciação crítico-subjetiva	39

5.1	Desafios	39
5.2	Disciplinas	40
5.3	Agradecimentos	40

1 Introdução e objetivos

Crianças com transtornos residuais ou persistentes da fala (TRPF) acabam por ter dificuldades, tanto para comunicação social quanto para sua alfabetização. Essas dificuldades acontecem pois os TRPF dificultam a produção de fonemas de acordo com as regras do idioma, podendo ser causados por alterações fonológicas motoras ou vocais, as quais podem ser observadas como alterações do trato vocal. Tais alterações, graças a recentes avanços nas técnicas de imageamento por ressonância magnética (RM), podem ser observadas em tempo real, o que auxilia na elaboração de tratamentos para tais transtornos.

Com o intuito de facilitar a identificação de TRPF na fala do paciente, é interessante, além do uso de imageamento por RM, a associação dessas imagens com o áudio da fala produzida durante a sessão de RM. Com a relação entre áudio e imagem, se torna mais simples determinar a TRPF associada a determinada deformação do trato vocal observado.

Esse áudio, adquirido dentro da sala de RM, é normalmente poluído por diversos ruídos de interferência, como aqueles gerados pela movimentação do paciente, a interferência dos campos magnéticos gerados pela máquina e o ruído acústico gerado pelas bobinas que produzem os campos magnéticos supracitados (Ravicz, Melcher e Kiang 2000). Esses ruídos dificultam a compreensão e análise da fala original pois a intensidade destes, quando comparadas ao sinal proveniente da fala do paciente, é alta, além de terem em sua composição frequências que intersectam o conjunto de frequências que compõem a fala. Tais dificuldades acabam por impor a necessidade de supressão de ruído acústico (SRA) no áudio obtido, de modo a melhorar a compreensão da fala do paciente.

Dado que tais ruídos possuem características periódicas, em conjuntos de frequências específicas, esta supressão pode ser concretizada por algoritmos que visam identificar essas características e, a partir dos padrões encontrados, extrair a fala "limpa" do paciente, i.e., a fala com diminuição do ruído acústico proveniente

da máquina de RM.

Este projeto consiste no estudo e implementação de três diferentes métodos para a realização de SRA em áudios semelhantes aos provenientes de exames de RM, sendo eles: Subtração espectral; método dos mínimos quadrados no domínio das frequências (FLMS) e supressão de ruído baseada no aprendizado de dicionários e análise *wavelet*. Ainda, serão descritas as implementações de duas versões de um *software* para realização de SRA utilizando os métodos estudados, sendo uma versão com interface gráfica (IG) e outra com interface de linha de comando (IC).

Por fim, serão realizadas análises de tempo de execução, da diferença de energia total dos espectros dos áudios obtidos e da capacidade de supressão de acordo com a amplitude da fala em relação ao ruído. Essas análises têm o intuito de apresentar subsídios para a escolha de método de SRA a partir dos resultados obtidos. Serão descritos também os *softwares* implementados, os quais buscam facilitar a utilização dos métodos de supressão pelo usuário.

Por fim, com a realização das análises citadas e implementação dos *softwares*, buscamos gerar recursos iniciais para facilitar a realização de procedimentos fonoaudiológicos que objetivam a identificação de TRPF com o auxílio de máquinas de RM e, indiretamente, reduzir eventuais custos associados à realização de tais procedimentos, com a produção de *software* livre.

2 Materiais e métodos

2.1 Áudios utilizados

Para captar de forma satisfatória o áudio do ambiente no qual ocorre um procedimento de imageamento por RM, com a finalidade da utilização deste áudio na análise dos métodos implementados neste projeto, seria necessária a utilização de dois microfones na sala na qual o procedimento é realizado. Enquanto o primeiro microfone captaria diretamente o áudio produzido pela fala do paciente, o segundo microfone, com a captação de áudio sincronizada com o primeiro, deveria captar o áudio do ambiente, mais precisamente, o ruído acústico gerado pela máquina de RM.

Em posse do ruído acústico e da fala do paciente poluída por tal ruído, ambos sincronizados, de modo a ser possível relacionar temporalmente os *frames* de cada áudio, seria possível analisar os métodos implementados utilizando dados reais de sessões de RM. Porém, dado que não é possível utilizar materiais metálicos dentro das máquinas de RM, para realizar a captação desejada, seria necessária a utilização de *hardware* especializado, composto por microfones de fibra ótica, os quais além de terem uma baixa qualidade de captação, têm um alto custo de aquisição.

Dado que não tínhamos o *hardware* indicado, tentamos recorrer ao uso do microfone utilizado pelos biomédicos para comunicação com o paciente durante a sessão de RM, mas os áudios captados não tiveram qualidade suficiente para serem úteis na análise dos métodos implementados.

Devido às dificuldades supracitadas, foi necessário criar sinais de áudio com características semelhantes àqueles provenientes da captação de áudio durante exames de RM. Sendo assim, se tornou necessário construir tanto áudios de fala sem ruído, quanto ruídos acústicos cujas características se aproximam àqueles gerados pelas máquinas de RM.

Para a construção do áudio de fala limpa, foi utilizado um trecho de diálogo de 30 segundos do filme *Die Hard*. Este trecho específico foi escolhido por não conter

música, dado que as frequências emitidas por instrumentos musicais poderiam causar algum viés no resultado obtido por terem alguma intersecção com os conjuntos de frequências que compõem os ruídos acústicos utilizados.

Os ruídos acústicos construídos constistem em amostras de 5 segundos de três diferentes tipos: *echo planar*; *diffusion* e *fast spin echo*. Estes ruídos específicos foram escolhidos por serem ruídos realmente emitidos por máquinas de RM, o que permite uma maior aproximação da realidade do que utilizar ruídos sintéticos tais como senóides. Esses ruídos foram obtidos a partir do site <http://mriquestions.com/whats-that-noise.html>.

Para construir o áudio de fala com ruído, os trechos de ruído criados foram repetidos de modo a terem o mesmo tamanho que o áudio de fala limpa, obtendo trechos de áudio com duração de 30 segundos. Ainda, para a criação de arquivos de áudio de fala com ruído acústico com diferentes durações, os trechos de 30 segundos, anteriormente contruídos, foram repetidos até obtermos os áudios com as durações desejadas.

Após a construção dos áudios base, os áudios de fala com ruído acústico foram contruídos da seguinte forma:

- Realização de transformada de Fourier em ambos os áudios base;
- Obtenção dos espectros de energia dos sinais;
- Soma dos espectros obtidos;
- Realização da transformada inversa da soma.

2.2 Métodos de supressão

Para realizar a supressão de ruído acústico, três métodos foram implementados, sendo eles: subtração espectral; métodos dos mínimos quadrados no domínio das frequências e supressão de ruído baseada no aprendizado de dicionários e análise *wavelet*.

2.2.1 Subtração espectral

Para realização satisfatória de subtração espectral, é necessária a suposição de que o ruído acústico presente no áudio a passar por SRA é aditivo, i.e., no domínio das frequências, a fala e o ruído podem ser separados em componentes distintas que, quando somadas, resultam no espectrograma do áudio com ruído original (Ravicz, Melcher e Kiang 2000). Formalmente, dizemos que um sinal $S(t)$ é aditivo caso existam espectros $X(t)$ e $Y(t)$ que satisfaçam a equação:

$$S(t) = X(t) + Y(t) \tag{1}$$

Partindo de tal suposição, o algoritmo de subtração espectral consiste em, a partir de uma estimativa do ruído presente no áudio, realizar, no domínio das frequências, a subtração entre o espectrograma do áudio original e do ruído acústico estimado. Assim, a versão melhorada do áudio é obtida após realizar a transformação do espectrograma resultante para o domínio de sinais (Pribil, Pribilova e Frollo 2016).

Esse método, ao supor aditividade do ruído, pode gerar artefatos no áudio resultante, quando da utilização de dados de fala obtidos em situações de sessões de RM, pois o ruído gerado pelas bobinas que criam os campos magnéticos possuem frequências que conflitam com o espectro de fala. O algoritmo a seguir demonstra a subtração espectral para áudio de fala e um ruído:

Algoritmo 1: Subtração espectral

```
1: SUBTRACAOESPECTRAL(fala, ruido)
2:   falaTransformada  $\leftarrow$  fourier(fala)
3:   fase  $\leftarrow$  fase(falaTransformada)
4:   amplitude  $\leftarrow$  amplitude(falaTransformada)
5:   potenciaDaFala  $\leftarrow$  falaTransformada ** 2
6:   potenciaDoRuido  $\leftarrow$  fourier(ruido) ** 2
7:   subtracao  $\leftarrow$  raiz(maximo(0, potenciaDaFala - potenciaDoRuido))
8:   return fourierInversa(subtracao)
```

Por fim, dado que para realizar a subtração espectral em um determinado *frame* dos espectros envolvidos não há necessidade de utilizar os demais *frames* destes espectros, os áudios foram divididos em blocos de tamanhos iguais, os quais podem passar pelo algoritmo definido acima ao mesmo tempo, de modo a utilizar entradas de menor tamanho. Com essa paralelização é possível diminuir o tempo de execução do método, de acordo com o número de blocos utilizados e com o número de processos atribuídos ao algoritmo para realizar a supressão em todos os blocos. Temos, então, o algoritmo final:

Algoritmo 2: Subtração espectral por blocos

```
1: SUBTRACAOESPECTRALPORBLOCOS(fala, ruido, blocos)
2:   blocosDeFala  $\leftarrow$  divide(fala, blocos)
3:   blocosDeRuido  $\leftarrow$  divide(ruido, blocos)
   for  $i \leftarrow 0$  to  $\text{blocos} - 1$  do
   |   blocoLimpo  $\leftarrow$  subtracaoespectral(blocosDeFala[ $i$ ], blocosDeRuido[ $i$ ])
   |   falaLimpa  $\leftarrow$  acrescenta(blocoLimpo, falaLimpa)
4:   return falaLimpa
```

2.2.2 FLMS

O método FLMS consiste na realização do método de mínimos quadrados (LMS), uma variante do algoritmo de gradiente descendente, no domínio de frequências do sinal, i.e., aplicação do filtro LMS no produto da transformada de Fourier de um dado sinal (Haykin 1996, p. 446-453). Tal filtro é treinado com base em pesos de ativação, os quais são atualizados iterativamente de acordo com o áudio desejado, recebido como entrada do algoritmo, e a estimativa do áudio suprimido calculado na etapa anterior, uma função de passo, a qual visa regular o tempo para convergência do filtro, uma amostra do áudio corrompido original e o próprio valor dos pesos durante a iteração anterior (Bresch et al. 2006).

A filtragem do sinal é dada por:

$$\delta(n) \approx w(n)u(n) \quad (2)$$

$$e(n) = d(n) - \delta(n) \quad (3)$$

$$w(n+1) = w(n) + \mu u(n)e(n) \quad (4)$$

Para justificar tais funções, partimos do algoritmo de gradiente descendente.

Seja um sinal de entrada $x(n)$. Podemos definir a estimativa da saída desejada como uma combinação entre o vetor de pesos $w(n)$ e o vetor $U(n) = [x(n), x(n-1), \dots, x(n-M+1)]$ (Haykin 1996, p. 340). Dessa forma, temos o seguinte produto interno:

$$\delta(n|U(n)) = w(n)U(n) \quad (5)$$

Assim, podemos estimar o erro $e(n)$ entre o sinal desejado $d(n)$ e o sinal

obtido como:

$$e(n) = d(n) - \delta(n)U(n) \quad (6)$$

A ideia é minimizar o erro $e(n)$, pois, com isso, o sinal estimado é o mais próximo possível de $d(n)$. Dado que tanto $d(n)$ quanto $U(n)$ são fixos para o instante n , podemos apenas ajustar os pesos do vetor $w(n)$ para obter a minimização desejada. O ajuste dos pesos se dá pela combinação entre o valor atual destes e o gradiente de uma função de custo, com influência controlada por uma taxa de passo (Haykin 1996, p. 342). Esse ajuste pode ser observado como:

$$w(n+1) = w(n) + \frac{1}{2}\mu(-\nabla J(n)) \quad (7)$$

Temos ainda que o gradiente pode ser dado pela combinação do vetor de correlação p entre $U(n)$ e $d(n)$ e a matriz de correlação R de $U(n)$ multiplicada pelo vetor de pesos (Haykin 1996, p. 340). Sendo assim, obtemos a seguinte equação:

$$w(n+1) = w(n) + \mu(p - Rw(n)) \quad (8)$$

Para aplicar o método LMS, ao método de gradiente descendente descrito acima, podemos fazer:

$$R(n) = U(n) * (U(n) * w(n))p(n) = U(n) * d(n) \quad (9)$$

Com isso, obtemos a seguinte equação para atualização dos pesos do vetor de coeficientes:

$$w(n+1) = w(n) + \mu U(n)e(n) \quad (10)$$

De modo a possibilitar a filtragem de sinal por blocos, podemos redefinir

$d(n)$, e $y(n)$, obtendo então:

$$\delta(kM + i|U(k)) = \sum_{l=0}^{M-1} w(k)U(kM - i - l) \quad (11)$$

$$d(n) = d(kM + i) \quad (12)$$

Sendo assim, podemos reescrever a equação de atualização dos pesos do vetor de coeficientes como:

$$w(k + 1) = w(k) + \mu \sum_{i=0}^{M-1} U(kM + i)e(kM + i) \quad (13)$$

Agora que temos o método LMS por blocos, podemos derivar o método FLMS.

O vetor de pesos $w(k)$ é reescrito como:

$$W(k) = FFT \begin{bmatrix} w(k) \\ 0 \end{bmatrix} \quad (14)$$

Em que FFT é a transformada de Fourier para N pontos, tal que $N = 2M$.

Ainda, devemos alterar $U(k)$ obtendo a diagonal da transformada de Fourier dos elementos do bloco $k - 1$ e do bloco k . Desse modo, temos:

$$U(k) = diagFFT[x((k - 1)M + i), x(kM + i)] \quad (15)$$

O elemento correspondente à estimativa do sinal é dado pelos últimos M elementos da transformada de Fourier de $U(k)W(k)$. Esse termo pode ser expresso por:

$$\delta(k) = [\delta(kM), \dots, \delta(kM + M - 1)] = IFFT[U(k)W(k)](M...2M) \quad (16)$$

O novo erro $E(k)$, de tamanho $2M$ é dado por:

$$E(k) = FFT \begin{bmatrix} 0 \\ e(k) \end{bmatrix} \quad (17)$$

Por fim, podemos redefinir a equação de atualização dos pesos como:

$$W(k+1) = W(k) + \mu FFT \begin{bmatrix} \phi(k) \\ 0 \end{bmatrix} \quad (18)$$

Em que $\phi(k)$ pode ser dado pelos primeiros M elementos da transformada inversa de Fourier de $U^H E(k)$, com U^H o conjugado de $U(k)$.

Seria possível utilizar uma versão com atraso do áudio com ruído para determinar o áudio desejado (Bresch et al. 2006). Não obstante, apenas conseguimos resultados satisfatórios quando utilizamos como áudio desejado o próprio áudio original.

O algoritmo a seguir expressa a aplicação do método FLMS para um bloco:

Algoritmo 3: FLMS para um bloco

- 1: **FLMSPORBLOCO**(bloco, passo, esquecimento)
 - 2: $U \leftarrow \text{fourier}(\text{bloco.entrada})$
 - 3: $U_{\text{conjugado}} \leftarrow \text{conjugado}(U)$
 - 4: $\text{saida} \leftarrow \text{atualizaSaida}(U, \text{bloco.coeficientes}, \text{bloco.M})$
 - 5: $\text{erro} \leftarrow \text{atualizaErro}(U, \text{bloco.saidaDesejada}, \text{bloco.M})$
 - 6: $\text{potencia} \leftarrow \text{esquecimento} * \text{bloco.pot} + (1 - \text{esquecimento}) * \text{abs}(U) ** 2$
 - 7: $\text{restricao} \leftarrow \text{restricaoGradiente}(\text{erro}, \text{conjugado}, 1/\text{potencia}, \text{bloco.M})$
 - 8: $\text{coeficientes} \leftarrow \text{atualizaCoeficientes}(\text{restricao}, \text{bloco.coeficientes}, \text{passo})$
 - 9: **return** *saida, coeficientes, potencia*
-

O algoritmo apresentado se utiliza de funções descritas nos algoritmos a seguir:

Algoritmo 4: Atualiza o valor da saída do sinal para um bloco

1: **ATUALIZASAI**DA(U, coeficientes, M)
2: convolucao \leftarrow U * coeficientes
3: **return** *fourierInversa(convolucao)*

Algoritmo 5: Calcula o valor do erro entre a saída e a saída desejada, com *padding*

1: **ATUALIZAERRO**(saida, saidaDesejada, M)
2: erro \leftarrow saida - saidaDesejada
3: zeros \leftarrow vetor(0, M)
4: **return** *acrescenta(zeros, erro)*

Algoritmo 6: Calcula o gradiente

1: **RESTRICAOGRADIENTE**(erro, conjugado, potenciaInversa, M)
2: convolucao \leftarrow potenciaInversa * conjugado * erro
3: zeros \leftarrow vetor(0, M)
4: restricaoTransformada \leftarrow *acrescenta(fourierInversa(convolucao), zeros)*
5: **return** *fourier(restricaoTransformada)*

Algoritmo 7: Calcula os novos pesos a serem utilizados para a supressão do próximo bloco

1: **ATUALIZACOEFI**CIENTES(restricao, coeficientes, passo)
2: convolucao \leftarrow passo * restricao
3: **return** *convolucao + coeficientes)*

2.2.3 Aprendizado de dicionários e análise *wavelet*

O método de SRA realizada a partir do aprendizado de dicionários de ruído e análise *wavelet* se baseia em um algoritmo de separação de componentes e limiarização (Vaz, Ramanarayanan e Narayanan 2013). Tal método consiste em duas fases: a aplicação do algoritmo *probabilistic latent component analysis* (PLCA) e realização de análise *wavelet*. Na primeira fase, um dicionário de ruído é descoberto, o qual é utilizado para encontrar uma aproximação do sinal de fala sem ruído original. Já na segunda fase, o sinal obtido é limiarizado de modo a aproximar ainda mais o sinal resultante do sinal de fala original.

Para aplicar o PLCA, supõe-se que um sinal pode ser observado como uma combinação linear de sinais, os quais podem ser linearmente separados. Observando um sinal de áudio que contém fonte de ruído e fonte de fala, podemos modelar o sinal com a seguinte equação:

$$S(t) = P(r) \sum_Z P(z|r)P(t|z, r) + P(s) \sum_Z P(z|s)P(t|z, s) \quad (19)$$

Em que $S(t)$ corresponde ao quadro t do sinal original, r representa a componente de ruído acústico, s representa a componente de fala, z representa uma subcomponente arbitrária que compõe r ou s , Z é o conjunto de todas as componentes e P é uma medida de probabilidade.

O dicionário de ruído é dado por $P(t|z, r)$, i.e., a probabilidade de uma subcomponente z do ruído r ter influência no sinal no momento t . Se esse dicionário é dado, a partir do cálculo das probabilidades que compõem a fórmula 2, conseguimos obter o dicionário da fala, dado por $P(t|z, s)$. Finalmente, para obter o sinal correspondente à componente de fala no quadro t , podemos aplicar a seguinte equação:

$$S'(t) = S(t) \sum_Z P(z, s|t) \quad (20)$$

O dicionário do ruído pode ser calculado a partir da aplicação do método PLCA para o sinal referente ao ruído acústico. Nesse caso, como não há componente

de fala no sinal a ser separado por componentes, podemos assumir que $P(t|z, s) = 0$.

O algoritmo para realização de PLCA é dado por:

Algoritmo 8: PLCA

```

1: PLCA(fala, dicionarioRuido, iteracoes)
2:    $P(t|z,r) \leftarrow$  dicionarioRuido
3:    $P(s), P(r) \leftarrow$  Probabilidade das componentes terem influência no sinal
4:    $P(z|s), P(z|r) \leftarrow$  Probabilidade de z ter influência na componente
5:    $P(z,s|t), P(z,r|t) \leftarrow$  Probabilidade de z e a componente terem influência no
   sinal no quadro t à posteriori
6:    $P(t|z,s) \leftarrow$  Probabilidade de z e s terem influência no sinal no quadro t à
   priori
   for  $i \leftarrow 1$  to iteracoes do
    $P(s), P(r) \leftarrow$  atualização de P(s) e P(r)
    $P(z|s), P(z|r) \leftarrow$  atualização de P(z|s) e P(z|r)
    $P(z, s|t), P(z, r|t) \leftarrow$  atualização de P(z,s|t) e P(z,r|t)
    $P(t|z, s) \leftarrow$  atualização de P(t|z,s)
7:   S  $\leftarrow$  Fala reconstruída
   for  $i \leftarrow 1$  to tamanhoda fala do
    $S(i) \leftarrow$  Fala(t) *  $\sum_z P(z, s|t)$ 
8:   return S

```

A atualizações definidas acima são dadas pelas seguintes equações:

$$P(s) = \frac{\sum_z \sum_t P(s, z|t)S(t)}{\sum_{z'} \sum_{s'} \sum_t P(s', z'|t)S(t)} \quad (21)$$

$$P(z|s) = \frac{\sum_t P(s, z|t)S(t)}{\sum_{z'} \sum_t P(s, z'|t)S(t)} \quad (22)$$

$$P(z, s|t) = \frac{P(s)P(z|s)P(t|z, s)}{\sum_{s'} P(s') \sum_{z'} P(z'|s')P(t|z', s')} \quad (23)$$

$$P(t|z, s) = \frac{\sum_t P(z, s|t)S(t)}{\sum_{t'} P(z, s|t')S(t')} \quad (24)$$

Essa primeira etapa, além de ser eficiente em remover ruído em momentos sem fala, consegue diminuir a potência do ruído sinal, viabilizando o uso de análise *wavelet*.

Na segunda etapa, a partir do uso de análise *wavelet*, decompomos em componentes os sinais de fala original, de ruído acústico e o sinal obtido na primeira fase. Para melhorar o sinal obtido na segunda fase, limiarizamos os coeficientes das componentes correspondentes ao sinal, utilizando uma métrica que busque diminuir o nosso erro (Vaz, Ramanarayanan e Narayanan 2013). Essa métrica de limiarização é dada por:

$$\lambda_k = \frac{\sigma_{Nk}^2}{\varepsilon_k} \sqrt{2(\varepsilon_k + \varepsilon_k^2) \ln\left(\sqrt{1 + \frac{1}{\varepsilon_k}}\right)} \quad (25)$$

Dado:

$$\varepsilon_k = \frac{\sigma_{Xk}^2}{\sigma_{Nk}^2} \quad (26)$$

Em que σ_{Xk}^2 é a variância da componente k do sinal de fala original e σ_{Nk}^2 é a variância da componente k do sinal do ruído.

Por fim, para obter o sinal de fala com ruído acústico suprimido final, as componentes limiarizadas do sinal obtido na primeira fase são utilizadas para realizar a transformada *wavelet* inversa.

2.3 *Software*

Para este projeto, dois softwares foram construídos com o intuito de realizar supressão de ruído acústico a partir da utilização dos métodos apresentados. Ambos os *softwares* consistem na mesma implementação dos métodos de supressão, porém, enquanto um é escrito com o intuito de sua utilização por linha de comando, o outro possui interface gráfica, o que facilita a utilização do mesmo por usuários menos versados na utilização de terminais.

Tais *softwares*, em formato livre, i.e., com permissão de reprodução, modificação e distribuição, foram utilizados para realizar a supressão de ruído acústico nos arquivos de áudio construídos para a realização das análises deste projeto.

A implementação dos *softwares* foi realizada utilizando a linguagem *python*, por ser uma linguagem facilmente adaptável para diferentes sistemas operacionais, além de possuir uma ampla gama de bibliotecas para a realização de computação científica, como realização de transformada de *Fourier* e análise *wavelet*, além da existência de diversas bibliotecas para a facilitação da manipulação de dados. As seguintes bibliotecas foram utilizadas para a produção dos *softwares*:

- NumPy - utilizado para realização de operações numéricas e manipulação de dados;
- Matplotlib - utilizado para plotar gráficos;
- PySoundFile - utilizado para ler e escrever arquivos de áudio no formato wav;
- PyQt5 - utilizado para implementar IG;
- Pywt - utilizado para realizar análise *wavelet*.

As ferramentas elaboradas recebem um arquivo de áudio em formato wav a ser processado e um segundo arquivo de áudio contendo o ruído acústico a ser suprimido ou o áudio original sem ruído.

A seguir, definimos a utilização dos *softwares* implementados, definindo os parâmetros que podem ser acessados pelo usuário.

2.3.1 Versão com IG

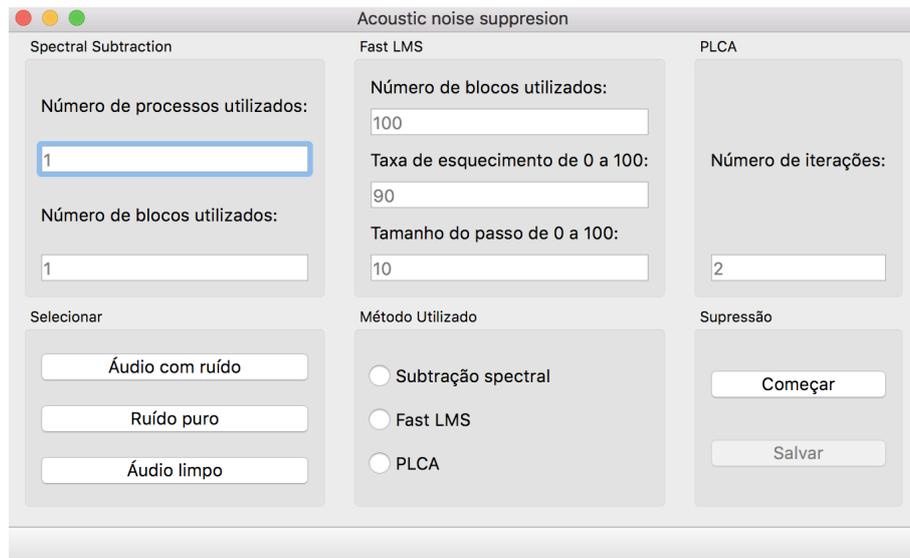


Figura 1: Interface gráfica do software implementado.

A interface desta versão se divide em seis módulos. Da esquerda para a direita, de cima para baixo os módulos são:

- **Subtração espectral**

Este módulo contém parâmetros utilizados para realização de supressão a partir do método de subtração espectral:

Número de processos - O número de processos que devem realizar a subtração dos blocos que compõem o áudio original.

Número de blocos - O número de trechos em que se deve dividir o áudio a ser suprimido.

- **Fast LMS**

Este módulo contém parâmetros utilizados para realização de supressão a partir do método FLMS:

Número de blocos - O número de trechos em que se deve dividir o áudio.

Taxa de esquecimento - Variando de 0 a 100, define o quão importante é o valor anterior da potência do sinal para a obtenção do novo valor da potência.

Passo - Variando de 0 a 100, esse parâmetro define o quanto a atualização do gradiente calculado deve contribuir para a atualização dos coeficientes.

- **PLCA**

Este módulo contém parâmetros utilizados para realização de supressão a partir do método de aprendizado de dicionários e análise wavelet:

Número de iterações - Define o número de iterações para o algoritmo PLCA.

- **Seleção dos arquivos de áudio**

Neste módulo, é possível selecionar os arquivos de áudio, em formato wav, a serem utilizados, sendo eles:

Áudio com ruído - O arquivo de áudio que a passar por supressão de ruído.

Ruído puro - O arquivo de áudio do ruído correspondente ao áudio com ruído.

Áudio limpo - o arquivo de áudio sem ruído. Esse áudio só é necessário para a realização de supressão utilizando o método FLMS.

- **Método utilizado**

Esse módulo consiste no seletor do método de supressão a ser utilizado. É possível escolher entre: subtração espectral; FLMS, dado pelo seletor "Fast LMS" ou aprendizado de dicionário com análise *wavelet*, dado pelo seletor "PLCA".

- **Supressão**

Este módulo contém os botões que permitem começar a supressão ou salvar o arquivo de áudio referente a esta supressão.

2.3.2 Versão de linha de comando

Para rodar essa versão do *software*, basta rodar o seguinte comando no terminal:

```
$ python3 -m app.cli.cli <argumentos>
```

Ao executar o programa, além de obter o arquivo de áudio suprimido, é impresso o tempo de execução para realizar a supressão desejada.

Para realizar a execução do código são necessários utilizar argumentos, sendo que alguns destes são obrigatórios apenas quando da utilização de determinado método de supressão. A seguir observamos a lista dos argumentos que podem ser utilizados na execução do programa:

- **mtd** - Indica o método de subtração de ruído. Temos os seguintes métodos: `plcaWavelet`; `specSub` e `flms`.
- **audioPath** - Indica o caminho do arquivo de áudio o qual deve passar por supressão de ruído.
- **noisePath** - Define o caminho do arquivo de áudio de ruído acústico referente ao áudio com ruído a ser suprimido. Esse arquivo é utilizado apenas para o método de supressão FLMS.
- **cleanPath** - Define o caminho do arquivo de áudio sem ruído. Esse arquivo é utilizado apenas para o método de supressão FLMS.
- **save** - Define o caminho no qual o arquivo de áudio suprimido deve ser salvo. Por padrão, o arquivo é salvo como `"suppressed.wav"`.
- **numProc** - Define o número de processos utilizados pelo algoritmo de subtração espectral. O número padrão de processos é 1.
- **blocks** - Define o número de blocos utilizados para realização de subtração espectral. O número padrão de blocos é 1.
- **m** - Define o número de blocos utilizados para realização de FLMS. O número padrão de blocos é 100.

- **fgt** - Esse parâmetro, entre 0.0 e 1.0, define a taxa de esquecimento para o algoritmo FLMS. A taxa padrão de esquecimento é 0.9.
- **step** - Esse parâmetro, entre 0.0 e 1.0, define o passo para o algoritmo FLMS. O tamanho padrão do passo é 0.1.
- **iterations** - Define o número de iterações do algoritmo PLCA para o método de supressão baseado em aprendizado de dicionários e análise *wavelet*. O número padrão de iterações é 2.

O código com a implementação dos dois *softwares* pode ser encontrado em <https://github.com/Vinyxr4/MAC0499>.

2.4 Testes comparativos

Com os métodos implementados, foram realizados testes de complexidade e qualidade de modo a comparar a complexidade de tempo de execução e qualidade final da supressão gerada por cada método.

2.4.1 Teste de complexidade

Com esses testes, era esperado poder determinar qual método obtém melhor performance de tempo diante dos sinais analisados. Para obter subsídios para tal análise, foram realizados 20 testes para cada combinação de método, ruído e tamanho de arquivo. Serão utilizados os métodos e ruídos definidos neste projeto, combinados com áudios de 1 minuto; 2 minutos e 3 minutos. As seguintes métricas de velocidade de execução foram extraídas:

- Média de execução;
- Desvio padrão;
- Variância.

Os testes foram realizados em um MacBook Pro 2017, com processador Intel Core I5 2,3GHz e 8 GB de memória RAM.

Para facilitar os testes para o método de subtração espectral, foi utilizado um número de blocos fixo igual a 4 e um número de processos também igual a 4.

Também, fixamos o número de iterações para o método de aprendizado de dicionários com *wavelet* em 10, pois percebemos que valores maiores eram muito custosos em termos de velocidade de execução mas melhoravam pouco o resultado final.

Por fim, os testes relacionados ao método FLMS tiveram o número de blocos fixado em 100, a taxa de esquecimento igual a 0.9 e o tamanho do passo igual a 0.1. Esses valores foram definidos de forma arbitrária após testar o método com diferentes valores.

2.4.2 Teste de qualidade

Os testes de qualidade se deram pela comparação entre os espectros de energia do áudio original, dos áudios com ruído e dos áudios suprimidos.

O formato destes espectros nos dão uma avaliação subjetiva para o desempenho dos métodos implementados, enquanto o valor das diferenças de energia total dos espectros com ruído em relação ao espectro de fala limpa e dos espectros suprimidos em relação ao espectro de fala limpa, nos dá subsídios para uma avaliação objetiva da capacidade dos métodos para realizar a supressão.

Para realizar essa avaliação objetiva, calculamos a diferença das áreas dos espectros. Dado que cada componente do espectro nos fornece a energia para determinada frequência, temos que a área, então, nos fornece a energia total do espectro. Sendo assim, calcular a diferença das áreas implica em calcular a diferença de energia total entre dois espectros.

O valor das diferenças das áreas dos espectros é dada por:

$$diff(original, transf) = \frac{\sum spec(transf - original)}{\sum spec(original) + \sum spec(transf)} \quad (27)$$

Em que *original* é o espectro do áudio original e *transformado* é o espectro do áudio transformado, seja ele o áudio com ruído ou o áudio suprimido.

2.4.3 Comparação entre a amplitude da fala e do ruído

Esse teste objetiva verificar se há diferença na qualidade da supressão de ruído acústico para os métodos implementados quando da diferença de amplitude do sinal de fala no áudio de fala com ruído.

Utilizamos novamente o cálculo de diferenças das áreas dos espectros obtidos para comparar o resultado da aplicação dos métodos de SRA em arquivos de áudio com 1 minuto de duração, poluídos pelo ruído *echo planar*, tendo essa comparação realizada para diversas amplitudes.

Definimos como amplitude máxima a amplitude do sinal de fala utilizado nos testes anteriores. A amplitude para cada teste foi modulada, então, pela multiplicação do sinal de fala de amplitude máxima por um valor escalar.

Mantendo fixa a amplitude do sinal de ruído acústico, esse teste foi realizado para sinais de fala de amplitude: 1%; 10%; 25%; 50%; e 100% do sinal de amplitude máxima.

3 Resultados

3.1 Testes de complexidade

A seguir estão listados os tempos médios de execução, em segundos, desvio padrão e variância para cada uma das combinações de métodos, tipos de ruído e duração de áudio.

Arquivos de áudio com duração de 1 minuto

Tabela 1: Ruído *echo planar*

	Subtração espectral	FLMS	Dicionário e wavelet
Média	0.543656	1.000183	6.842135
Desvio padrão	0.02294	0.022398	0.320086
Variância	0.000526	0.000502	0.102455

Tabela 2: Ruído *diffusion*

	Subtração espectral	FLMS	Dicionário e wavelet
Média	0.610286	1.095632	6.833603
Desvio padrão	0.053945	0.074799	0.281663
Variância	0.00291	0.005595	0.079334

Tabela 3: Ruído *fast spin echo*

	Subtração espectral	FLMS	Dicionário e wavelet
Média	0.596537	1.095728	6.772466
Desvio padrão	0.050234	0.04574	0.076956
Variância	0.002523	0.002092	0.005922

Arquivos de áudio com duração de 2 minutos

Tabela 4: Ruído *echo planar*

	Subtração espectral	FLMS	Dicionário e wavelet
Média	1.177144	1.951054	17.253504
Desvio padrão	0.057597	0.023092	0.838324
Variância	0.003317	0.000533	0.702788

Tabela 5: Ruído *diffusion*

	Subtração espectral	FLMS	Dicionário e wavelet
Média	1.183436	2.058522	17.34534
Desvio padrão	0.07098	0.062166	1.336983
Variância	0.005038	0.003865	1.787525

Tabela 6: Ruído *fast spin echo*

	Subtração espectral	FLMS	Dicionário e wavelet
Média	1.171933	2.115442	17.414454
Desvio padrão	0.051537	0.120175	1.441007
Variância	0.002656	0.014442	2.0765

Arquivos de áudio com duração de 3 minutos

Tabela 7: Ruído *echo planar*

	Subtração espectral	FLMS	Dicionário e wavelet
Média	1.899378	3.216285	40.645439
Desvio padrão	0.080345	0.34009	2.959143
Variância	0.006455	0.115661	8.756529

Tabela 8: Ruído *diffusion*

	Subtração espectral	FLMS	Dicionário e wavelet
Média	1.921839	3.389353	40.517081
Desvio padrão	0.067823	0.28146	3.598675
Variância	0.0046	0.07922	12.950464

Tabela 9: Ruído *fast spin echo*

	Subtração espectral	FLMS	Dicionário e wavelet
Média	1.937032	3.527013	44.969892
Desvio padrão	0.056291	0.524788	9.726319
Variância	0.003169	0.275403	94.601272

3.2 Espectros de energia

Dado que os arquivos de áudio utilizados são construídos a partir de repetições dos áudios base, os arquivos com o mesmo tipo de ruído e suprimidos com o mesmo método, mas que diferem em tamanho, possuem características espectrais muito semelhantes, como pode ser visto nas figuras 2, 3 e 4. Sendo assim, apresentaremos apenas os espectros dos áudios com duração de 1 minuto.

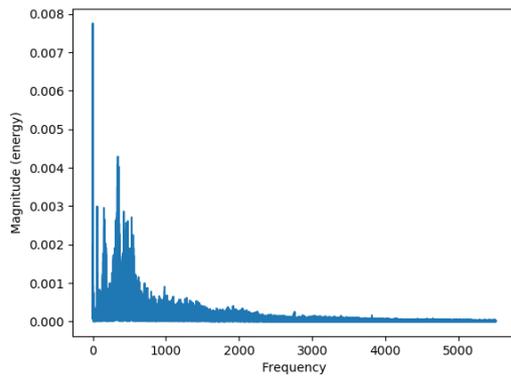


Figura 2: Espectro do áudio suprimido pelo método de aprendizado de dicionário com *wavelet*. O ruído utilizado foi o *echo planar* e o áudio utilizado tem duração de 1 minuto.

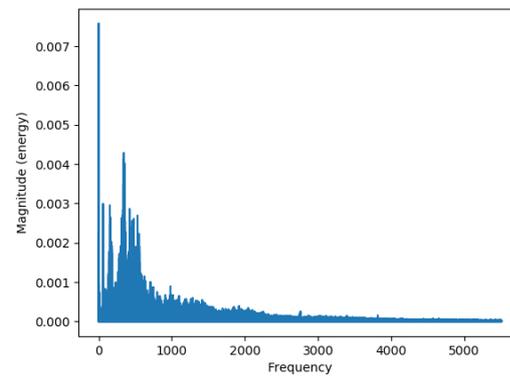


Figura 3: Espectro do áudio suprimido pelo método de aprendizado de dicionário com *wavelet*. O ruído utilizado foi o *echo planar* e o áudio utilizado tem duração de 2 minutos.

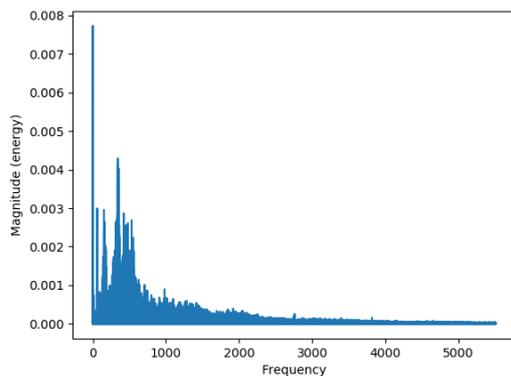


Figura 4: Espectro do áudio suprimido pelo método de aprendizado de dicionário com *wavelet*. O ruído utilizado foi o *echo planar* e o áudio utilizado tem duração de 3 minutos.

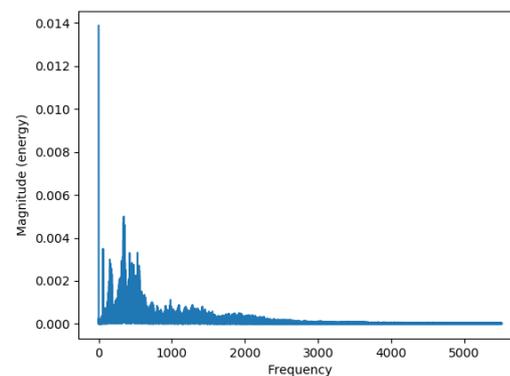


Figura 5: Espectro do áudio de fala limpo.

Na próxima seção, apresentamos os espectros dos áudios poluídos com ruído. A comparação entre esses espectros e o espectro do áudio original nos fornece uma noção do quão diferente o áudio limpo é em comparação com os áudios com ruído.

Já nas seções subsequentes, apresentaremos os espectros dos áudios suprimidos para cada método de supressão utilizado. A comparação das formas desses espectros com a forma do espectro da figura 5 nos dá noção do quão bem cada método de supressão utilizado se saiu para limpar o ruído adicionado.

3.2.1 Ruídos

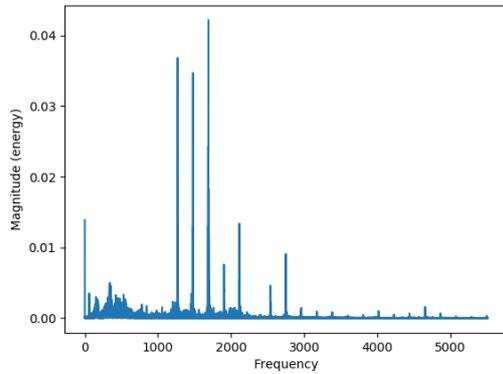


Figura 6: Espectro do áudio com ruído *echo planar*.

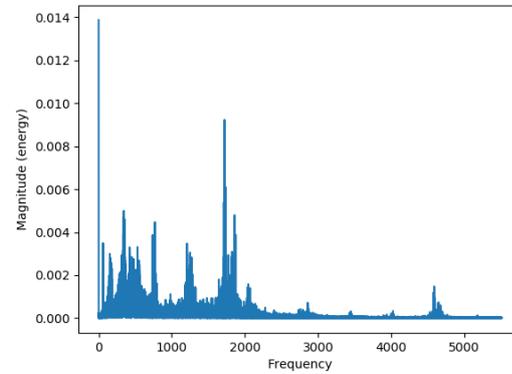


Figura 7: Espectro do áudio com ruído *diffusion*.

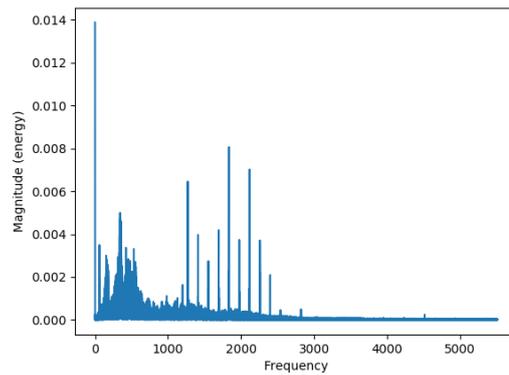


Figura 8: Espectro do áudio com ruído *fast spin echo*.

3.2.2 Subtração espectral

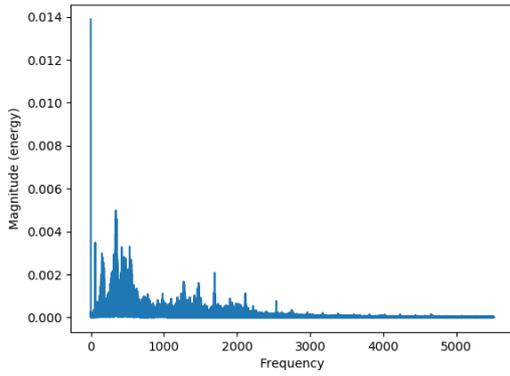


Figura 9: Espectro do áudio com ruído *echo planar* após a supressão utilizando subtração espectral.

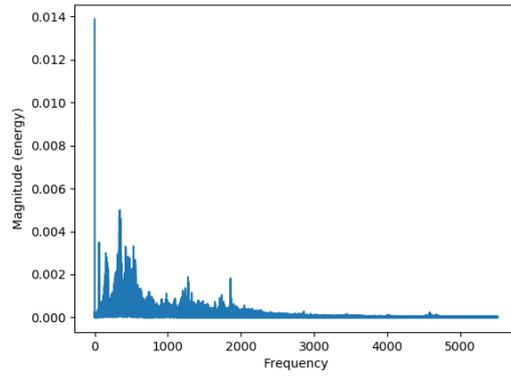


Figura 10: Espectro do áudio com ruído *diffusion* após a supressão utilizando subtração espectral.

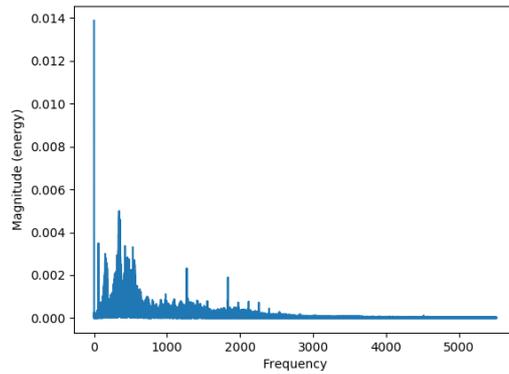


Figura 11: Espectro do áudio com ruído *fast spin echo* após a supressão utilizando subtração espectral.

3.2.3 FLMS

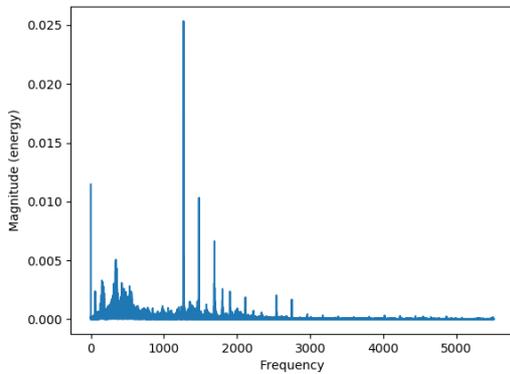


Figura 12: Espectro do áudio com ruído *echo planar* após a supressão utilizando o método FLMS.

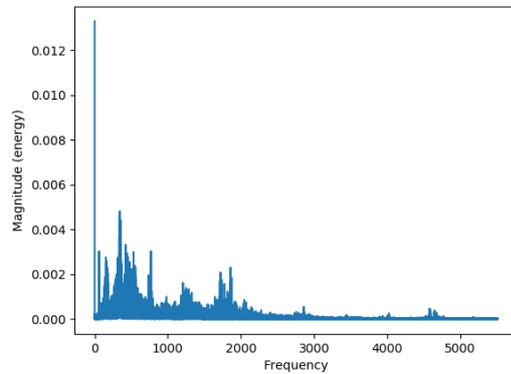


Figura 13: Espectro do áudio com ruído *diffusion* após a supressão utilizando o método FLMS.

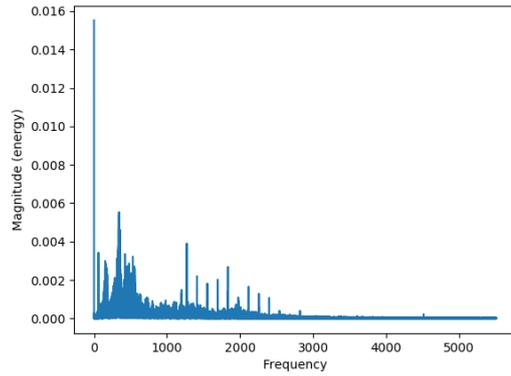


Figura 14: Espectro do áudio com ruído *fast spin echo* após a supressão utilizando o método FLMS.

3.2.4 Aprendizado de dicionários e análise wavelet

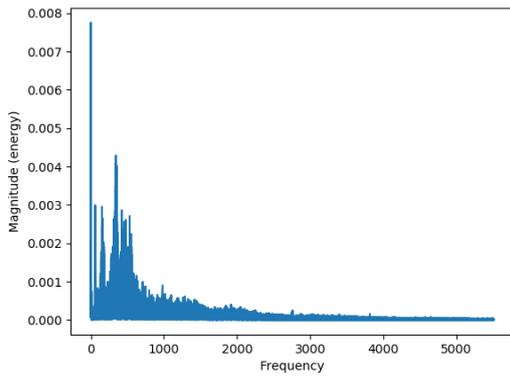


Figura 15: Espectro do áudio com ruído *echo planar* após a supressão utilizando o método baseado em aprendizagem de dicionários e análise *wavelet*.

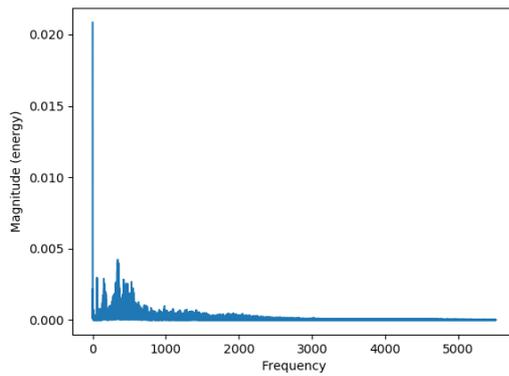


Figura 16: Espectro do áudio com ruído *diffusion* após a supressão utilizando o método baseado em aprendizagem de dicionários e análise *wavelet*.

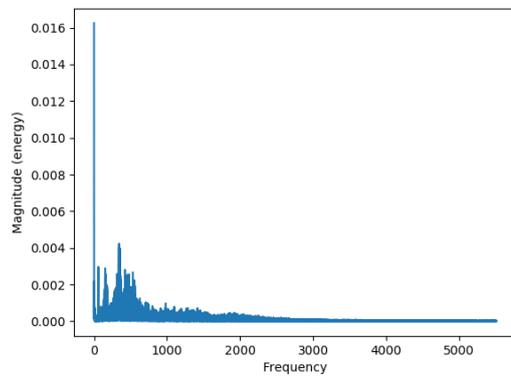


Figura 17: Espectro do áudio com ruído *fast spin echo* após a supressão utilizando o método baseado em aprendizagem de dicionários e análise *wavelet*.

3.3 Diferença de energia

Nessa seção apresentamos os valores obtidos para as diferenças de energia total entre o áudio original e os áudios com ruído e áudios suprimidos. Quanto mais próximo de zero, mais próximo um áudio é do áudio original. Valores negativos indicam que o áudio comparado possui menor energia total que o áudio original, enquanto valores positivos indicam que o áudio comparado tem mais energia que o áudio original. Novamente, devido à forma dos áudios utilizados nos experimentos, vamos apresentar os valores obtidos apenas para os áudios com duração de 1 minuto.

Tabela 10: Valores das diferenças de área dos espectros para o ruído *echo planar*

	Fala e ruído	Subtração espectral	FLMS	Dicionário e wavelet
Dif.	0.048967	0.002359	-0.042717	-0.067662

Tabela 11: Valores das diferenças de área dos espectros para o ruído *diffusion*

	Fala e ruído	Subtração espectral	FLMS	Dicionário e wavelet
Dif.	0.014961	0.000685	-0.012841	-0.056532

Tabela 12: Valores das diferenças de área dos espectros para o ruído *fast spin echo*

	Fala e ruído	Subtração espectral	FLMS	Dicionário e wavelet
Dif.	0.004637	-9.942676e-5	-0.014864	-0.055744

3.4 Comparação entre a amplitude da fala e do ruído

Nessa seção apresentamos os valores obtidos para as diferenças das áreas entre os áudios originais com amplitude do sinal de fala modulado e seus respectivos áudios suprimidos. Novamente, quanto mais próximo de zero, mais próximo um áudio é do áudio original.

Tabela 13: Diferenças de área dos espectros para o ruído *echo planar* com 1% da amplitude de fala

	Fala e ruído	Subtração espectral	FLMS	Dicionário e wavelet
Dif.	0.882254	0.162784	0.832077	-0.0121896

Tabela 14: Diferenças de área dos espectros para o ruído *echo planar* com 10% da amplitude de fala

	Fala e ruído	Subtração espectral	FLMS	Dicionário e wavelet
Dif.	0.413564	0.037801	0.322014	-0.209526

Tabela 15: Diferenças de área dos espectros para o ruído *echo planar* com 25% da amplitude de fala

	Fala e ruído	Subtração espectral	FLMS	Dicionário e wavelet
Dif.	0.207477	0.017063	0.026462	-0.168428

Tabela 16: Diferenças de área dos espectros para o ruído *echo planar* com 50% da amplitude de fala

	Fala e ruído	Subtração espectral	FLMS	Dicionário e wavelet
Dif.	0.106133	0.008061	-0.062564	-0.112022

Tabela 17: Diferenças de área dos espectros para o ruído *echo planar* com 100% da amplitude de fala

	Fala e ruído	Subtração espectral	FLMS	Dicionário e wavelet
Dif.	0.048967	0.002359	-0.042717	-0.067662

4 Conclusões

Neste projeto, estudamos e implementamos três métodos para realização de supressão de ruído acústico em sinais de áudio compostos por fala e ruído semelhante aos produzidos em sessões de RM, além de realizar testes comparativos e implementar um conjunto de *softwares* para a aplicação de tais métodos.

A maior parte do trabalho foi dedicada à implementação, em python, dos *softwares*, dada a necessidade do estudo e implementação dos métodos de supressão além, da implementação das interfaces e tratamento de erros. O produto dessas implementações encontra-se em <https://github.com/vinyxr4/MAC0499>.

Após a implementação, realizamos os testes pretendidos, sendo capazes de obter subsídios para definir o desempenho dos métodos implementados, tanto em termos de complexidade de tempo quanto em qualidade do resultado obtido.

Pudemos observar que o método de subtração espectral, o mais básico dentre os implementados, obteve bons resultados de supressão, sendo o método com menor complexidade de tempo. Ainda assim, como pode ser observado nas tabelas 10, 11 e 12, o sinal resultante desse método tende a ter energia maior que o sinal que contém apenas fala.

Já em relação ao método FLMS, pudemos perceber que este tende a produzir sinal de energia menor que o sinal de fala, tendo uma complexidade de tempo cerca de duas vezes maior que o primeiro método, o que ainda é pequeno, se comparado à duração dos áudios suprimidos. Não obstante, por não termos conseguido utilizar uma versão atrasada do ruído como sinal desejado de forma satisfatória, utilizamos o próprio sinal de fala como resultado desejado, o que dificulta a utilização do método de forma prática, pois esse sinal é justamente o que tentamos encontrar.

O método de aprendizado de dicionário de ruído e análise *wavelet* se mostrou como aquele com melhor resultado, em termos de aproximação do sinal com fala e ruído ao sinal de fala. Porém, esse método, como é perceptível pela observação das tabelas de 1 a 9, tem complexidade de tempo que chega a ser cerca de vinte vezes

maior que o método de subtração espectral, levando um tempo de até quase 25% da duração do áudio suprimido para ser executado.

Ainda em relação à qualidade dos resultados obtidos através da supressão de ruído, ao observar as figuras 9 a 17, referentes aos sinais após supressão de ruído, com as figuras 6 a 8, referentes aos espectros dos sinais poluídos por ruído acústico, notamos que os espectros dos sinais suprimidos pelo método de aprendizado de dicionários e análise *wavelet* tiveram maior redução de picos de energia ao longo das frequências.

Pudemos observar ainda, pelas tabelas 13 a 17, que quanto mais próxima a amplitude do sinal de fala em relação ao sinal referente ao ruído acústico, melhor é a aproximação ao sinal de fala original resultante da aplicação dos métodos de supressão implementados.

Por fim, dados os argumentos apresentados, o método de aprendizado de dicionários de ruído e análise *wavelet* se mostrou o mais eficiente qualitativamente para realização de supressão, podendo ser indicado para supressão *offline*, i.e., supressão de áudio já captado. Já para supressão *online*, indicamos o método de subtração espectral, dada sua baixa complexidade de tempo para execução associada a uma boa capacidade de supressão.

5 Apreciação crítico-subjetiva

5.1 Desafios

Realizar a construção de um projeto ao longo de um ano é um grande desafio. Encontrar disciplina para fazer o que tem que ser feito, estudar coisas que você já estudou um milhão de vezes e coisas que você nunca acreditou que estudaria, se preocupar com prazos e argumentações são desafios encontrados durante a confecção de um trabalho de conclusão de curso, mas também são desafios encarados durante toda a graduação.

Porém, alguns desafios me foram impostos especialmente durante a produção deste projeto:

Estudar áreas como o processamento de sinais digitais me foi um novo desafio dado que não havia cursado disciplina correlatas previamente ao início do trabalho.

Também, a dificuldade de encontrar conjuntos de dados para realização dos testes pretendidos me mostrou que a realização de trabalho científico não envolve apenas desenvolver e aplicar fórmulas, mas também ter a perspicácia de entender como apresentar argumentos para comprovação dos trabalhos.

Outro desafio foi o de conciliar a produção deste trabalho com a realização de estágio. Essa dificuldade apenas aumentou minha admiração por aqueles que não têm a possibilidade de apenas estudar, tendo que garantir o sustento e ainda conseguir alimentar a mente.

Por fim, a escrita da monografia trouxe à tona o desafio de escrever um texto conciso e coeso, tendo em mente que outras pessoas irão apreciar sua obra. Esse desafio me ajudou a lembrar o porquê de a educação básica ser tão importante.

5.2 Disciplinas

Nesta seção, gostaria de destacar algumas das disciplinas que, acredito eu, acrescentaram muito, tanto à minha formação acadêmica, quanto à minha formação como pessoa:

- **Introdução à computação**

Aqui eu entrei no mundo da computação, comecei a aprender o que viria ser o meu futuro.

- **Álgebra I**

Aqui eu comecei a entender a formalização e a beleza da matemática, o poder assombroso que algumas poucas afirmações podem ter sobre o mundo.

- **Introdução à probabilidade e estatística**

Ainda me lembro de quando o professor, durante um período de greve, trouxe projeções dos gastos da Universidade, mostrando que a ciência não está apenas trancada nos livros, mas que também é capaz de informar.

5.3 Agradecimentos

Agradeço à minha mãe por seus esforços, por ser capaz de me permitir frequentar a Universidade sem se preocupar excessivamente com questões financeiras. Agradeço também a todos aqueles que conheci durante a graduação e que me ajudaram de alguma forma a continuar e aqueles também que eu ajudei.

Agradeço especialmente ao meu amigo, Patrick Abrahão Menani e a sua família, que sempre me acolheu muito bem quando precisávamos estudar. Também agradeço a todos aqueles que me deram uma chance de continuar, quando eu poderia ter parado.

Por fim, agradeço ao Marcel Jackowski, por ter criado em mim o interesse pela área de computação gráfica, ter me orientado durante minha iniciação científica e ainda ter aceito me orientar durante este trabalho final de graduação.

Referências

- [1] Erik Bresch et al. “Synchronized and noise-robust audio recordings during realtime magnetic resonance imaging scans”. Em: *The Journal of the Acoustical Society of America* 120.4 (2006), pp. 1791–1794.
- [2] Radhika Chinaboina et al. “Adaptive algorithms for acoustic echo cancellation in speech processing”. Em: *International Journal of Research and Reviews in Applied Sciences* 7.1 (2011), pp. 38–42.
- [3] Simon Haykin. *Adaptive Filter Theory (3rd Ed.)* Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996. ISBN: 0-13-322760-X.
- [4] Moriel S NessAiver et al. “Recording high quality speech during tagged cine-MRI studies using a fiber optic microphone”. Em: *Journal of Magnetic Resonance Imaging* 23.1 (2006), pp. 92–97.
- [5] Jiri Pribil, Anna Pribilova e Ivan Frolo. “Analysis of acoustic noise and its suppression in speech recorded during scanning in the open-air MRI”. Em: *Advances in Noise Analysis, Mitigation and Control*. InTech, 2016.
- [6] Michael E Ravicz, Jennifer R Melcher e Nelson Y-S Kiang. “Acoustic noise during functional magnetic resonance imaging”. Em: *The Journal of the Acoustical Society of America* 108.4 (2000), pp. 1683–1696.
- [7] Colin Vaz, Vikram Ramanarayanan e Shrikanth Narayanan. “A two-step technique for MRI audio enhancement using dictionary learning and wavelet packet analysis.” Em: *INTERSPEECH*. 2013, pp. 1312–1315.