

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

JUAN VICENTE DE ALMEIDA MARTINEZ

**SEPARAÇÃO DE VOZES UTILIZANDO ANÁLISE DE COMPONENTES
INDEPENDENTES E FILTRAGEM WAVELET**

CAMPO MOURÃO

2023

JUAN VICENTE DE ALMEIDA MARTINEZ

**SEPARAÇÃO DE VOZES UTILIZANDO ANÁLISE DE COMPONENTES
INDEPENDENTES E FILTRAGEM WAVELET**

**Voices Separation Utilizing Independent Component Analysis and Wavelet
Filtering**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica do Curso de Bacharelado em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Osmar Tormena Júnior

CAMPO MOURÃO

2023



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

JUAN VICENTE DE ALMEIDA MARTINEZ

**SEPARAÇÃO DE VOZES UTILIZANDO ANÁLISE DE COMPONENTES
INDEPENDENTES E FILTRAGEM WAVELET**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Eletrônica do Curso de Bacharelado em Engenharia Eletrônica da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 23/novembro/2023

Andre Luiz Regis Monteiro
Doutor em Engenharia Elétrica e Informática Industrial
Universidade Tecnológica Federal do Paraná - UTFPR

Marcio Rodrigues da Cunha
Doutor em Engenharia Elétrica
Universidade Tecnológica Federal do Paraná - UTFPR

Osmar Tormena Júnior
Doutor em Engenharia Elétrica
Universidade Tecnológica Federal do Paraná - UTFPR

CAMPO MOURÃO
2023

RESUMO

Este trabalho é composto da aplicação de um método para separação de vozes utilizando a análise de componentes independentes e aprimoramento com filtragem *wavelet* por meio do *software* MATLAB®. Este objetivo foi alcançado com diversos processamentos dos sinais utilizados como entrada no algoritmo, como agrupamento dos sinais do banco de dados em pares, mistura das vozes, filtragens e cálculo de indicadores para análise. Os resultados são mostrados por meio de imagens, áudios, e tabelas contendo os parâmetros de análise.

Palavras-chave: separação cega de fontes; aprimoramento de voz; matlab.

ABSTRACT

This work is composed of the application of a method for separation of speeches utilizing independent component analysis and the enhancement with wavelet filtering made with the MATLAB® software. This objective was achieved by several processing steps of the signal utilized as input in the algorithm, such as grouping signals from the database in pairs, mixing of speeches, filtering, and computing indicators for analysis. The results are shown by images, audios, and tables with the analysis parameters.

Keywords: blind source separation; speech enhancement; matlab.

LISTA DE FIGURAS

Figura 1 – Translação e dilatação da <i>wavelet</i>	11
Figura 2 – Banco de filtros de dois canais	12
Figura 3 – Processo de filtragem básico	12
Figura 4 – Árvore de decomposição wavelet	13
Figura 5 – Limiarização rígida.	14
Figura 6 – Limiarização suave.	15
Figura 7 – ICA aplicada ao problema do coquetel	16
Figura 8 – Diagrama do processamento	20
Figura 9 – Diagrama detalhado do processamento	22
Figura 10 – db10, coif3 e sym8.	23
Figura 11 – Par de sinais originais	26
Figura 12 – Par de sinais com adição de ruído	26
Figura 13 – Sinais misturados	27
Figura 14 – Sinais misturados e filtrados	28
Figura 15 – Par de sinais separados	29
Figura 16 – Estado original e final do sinal 1	30
Figura 17 – Estado original e final do sinal 2	30
Figura 18 – Estado original e final do sinal 1 com filtragem seletiva	31
Figura 19 – Estado original e final do sinal 2 com filtragem seletiva	31
Figura 20 – Sinal com pior desempenho	32
Figura 21 – Estado original e final do sinal	33
Figura 22 – Sinal original e sinal com filtragem seletiva	34
Figura 23 – Sinal com pior desempenho	34
Figura 24 – Estado original e final do sinal	35
Figura 25 – Sinal original e sinal com filtragem seletiva	36
Figura 26 – Sinal com pior desempenho	36
Figura 27 – Estado original e final do sinal	37
Figura 28 – Sinal original e sinal com filtragem seletiva	38
Figura 29 – Sinal com pior desempenho	38
Figura 30 – Estado original e final do sinal	39

Figura 31 – Sinal original e sinal com filtragem seletiva	40
Figura 32 – Sinal com pior desempenho	40

LISTA DE TABELAS

Tabela 1 – Cálculo de correlação	32
Tabela 2 – Cálculo de norma do erro	33
Tabela 3 – Cálculo de correlação	35
Tabela 4 – Cálculo de norma do erro	35
Tabela 5 – Cálculo de correlação	37
Tabela 6 – Cálculo de norma do erro	37
Tabela 7 – Cálculo de correlação	39
Tabela 8 – Cálculo de norma do erro	39
Tabela 9 – Cálculo de correlação	41
Tabela 10 – Cálculo de norma do erro	41
Tabela 11 – Sumarização dos cálculos	41

SUMÁRIO

1	INTRODUÇÃO	8
1.1	Considerações iniciais	8
1.2	Objetivos	8
1.2.1	Objetivo geral	8
1.2.2	Objetivos específicos	9
1.3	Justificativa	9
2	REFERENCIAL TEÓRICO	10
2.1	Wavelet	10
2.1.1	Banco de Filtros	11
2.1.2	Limiarização	13
2.2	Análise de Componentes Independentes	14
2.2.1	Ambiguidades	18
2.2.2	Algoritmo	18
3	MATERIAIS E MÉTODOS	20
3.1	Escolha e tratamento dos dados	21
3.2	Código	21
3.2.1	Mistura dos sinais	21
3.2.2	Filtragem wavelet e análise do resultado	22
3.2.3	Separação das vozes	24
4	RESULTADOS	25
4.1	Simulações	25
4.1.1	Primeira simulação	25
4.1.2	Segunda simulação	33
4.1.3	Terceira simulação	35
4.1.4	Quarta simulação	37
4.1.5	Quinta simulação	39
4.2	Discussões	41
5	CONCLUSÃO	42
	REFERÊNCIAS	43

1 INTRODUÇÃO

Neste capítulo serão apresentados uma breve contextualização do tema, assim como os objetivos e a justificativa do trabalho.

1.1 Considerações iniciais

As pessoas estão cercadas por sons. Um ambiente tão ruidoso torna difícil a captação de uma fala e é difícil manter uma conversa confortável em tal ambiente. Isto faz com que seja importante a capacidade de separar e extrair um sinal de fala desejado de observações ruidosas de comunicações homem-máquina assim como humano-humano (Makino; Lee; Sawada, 2007).

A separação cega de fontes (BSS—*Blind Source Separation*) é uma abordagem para estimar sinais fonte utilizando apenas informação de misturas observadas. A estimativa é realizada sem o conhecimento de informações de cada fonte, como suas características de frequência e localização, ou como as fontes são misturadas. O uso da BSS no desenvolvimento de canais de comunicação entre humanos e máquinas é amplamente aceito (Makino; Lee; Sawada, 2007).

Pertencente a uma classe de métodos de BSS está a Análise de Componentes Independentes (ICA—*Independent Component Analysis*) que realiza a separação de dados em componentes informacionais subjacentes, onde tais dados podem existir na forma de imagens, sons, canais de telecomunicação ou preços no mercado de ações. O termo “cega” é intencional, pois implica que tais métodos podem separar dados em sinais de origem mesmo que pouco se conheça sobre a natureza destes sinais (Stone, 2004).

A técnica de ICA é uma invenção relativamente nova. Foi introduzida pela primeira vez no início da década de 1980 no contexto de modelagem de redes neurais. Na metade da década de 1990, alguns algoritmos de sucesso foram introduzidos por diversos grupos de pesquisa, junto com incríveis demonstrações em problemas como o efeito do coquetel, onde formas de onda de falas individuais são retiradas de suas misturas (Hyvarinen, 2001).

1.2 Objetivos

1.2.1 Objetivo geral

Implementar um código no *software* MATLAB[®] que separa e filtra, utilizando a análise de componentes independentes e filtragem *wavelet*, um conjunto de vozes misturadas em pares.

1.2.2 Objetivos específicos

Para alcançar o objetivo do trabalho, os seguintes objetivos específicos devem ser atingidos:

- Preparar o conjunto de dados para adequar sua utilização no código.
- Implementar o código que realiza a separação e a filtragem das vozes.
- Validar o código por meio da análise da correlação e da norma do erro entre os sinais separados e os sinais originais.

1.3 Justificativa

Sinais reais de fala são comumente contaminados por ruídos do ambiente, outras vozes, e/ou reverberação (Vincent; Virtanen; Gannot, 2018). O aprimoramento da qualidade destes sinais é desejável uma vez que isto pode reduzir a fadiga do ouvinte, causada por longos períodos de exposição ao ruído. A necessidade de aprimorar sinais de fala surge em diversas situações onde o sinal se origina de um local ruidoso ou é afetado pelo ruído por meio do canal de comunicação (Loizou, 2007).

A separação de fala é um tópico de pesquisa vital no processamento de sinais. Os métodos de separação se concentram em sua maioria a situações de ausência ou pouco ruído. No entanto, em comunicações reais, o sinal possui grande interferência pelo ruído do ambiente. Desta forma, é de grande valia resolver o problema da separação de fala em um ambiente ruidoso (Zhao *et al.*, 2005).

2 REFERENCIAL TEÓRICO

Este capítulo apresenta a fundamentação teórica utilizada para o desenvolvimento do trabalho. Primeiramente é apresentada a teoria da representação dos sinais no domínio *wavelet*, na sequência é apresentada a análise de componentes independentes.

2.1 Wavelet

A transformada *wavelet* é uma ferramenta que divide dados ou funções em diferentes componentes de frequência, e então estuda cada componente com uma resolução de acordo com sua escala (Daubechies, 1992).

...pode-se interpretá-la como uma técnica por janelas com regiões de dimensão variável, onde as wavelets, diferentemente de Fourier, tem como base uma função de duração limitada, isso é, de suporte compacto, que é uma propriedade na qual seu domínio é diferente de zero em uma extensão finita e igual a zero em todo o resto (Bianchi, 2006, p. 54).

A transformada *wavelet* contínua (CWT—*Continuous Wavelet Transform*) é descrita na Eq. (1), que toma o produto interno de uma função f com uma família de funções indicadas pela forma $\psi^{a,b}(s) = |a|^{-1/2} \psi\left(\frac{s-b}{a}\right)$. Sendo as funções $\psi^{a,b}$ chamadas “*wavelets*” e ψ chamadas de “*wavelet* mãe”. Por sua vez, a transformada *wavelet* discreta (DWT—*Discrete Wavelet Transform*) é descrita na Eq. (2), restringindo a e b a valores discretos: $a = a_0^m$, $b = nb_0a_0^m$, com m, n dentro do conjunto \mathbb{Z} , e $a_0 > 1$, $b_0 > 0$ fixos (Daubechies, 1992).

$$(T^{\text{WAV}} f)(a,b) = |a|^{-1/2} \int f(t) \psi\left(\frac{t-b}{a}\right) dt \quad (1)$$

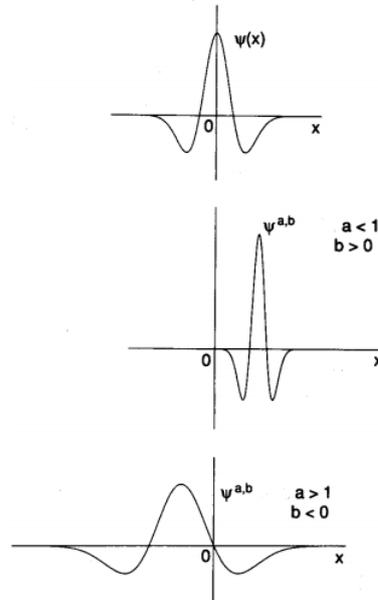
$$T_{m,n}^{\text{WAV}}(f) = a_0^{-m/2} \int f(t) \psi(a_0^{-m}t - nb_0) dt \quad (2)$$

Em ambos os casos assumindo que ψ satisfaz

$$\int \psi(t) dt = 0 \quad (3)$$

Os parâmetros b e a são parâmetros de translação e dilatação respectivamente. Na Eq. (1), reduzindo a , o suporte de ψ é reduzido no tempo e desta forma cobre um maior alcance na frequência e vice-versa. Portanto, $1/a$ é uma medida de frequência. Por outro lado, o parâmetro b indica a localização da janela da *wavelet* no eixo do tempo. Deste modo, alterando (a, b) , $T^{\text{WAV}} f$ pode ser computada em todo o plano tempo-frequência (Goswami; Chan, 2011). O comportamento destes parâmetros sobre a função *wavelet* pode ser visto na Figura 1.

Figura 1 – Translação e dilatação da *wavelet*.



Fonte: Adaptado de Daubechies (1992).

Para que o sinal original seja reconstruído após a transformação aplica-se a transformada *wavelet* inversa. A Eq. (4) descreve a transformada inversa, que envolve uma integração em duas dimensões em relação ao parâmetro de escala $a > 0$ e ao parâmetro de translação b . A Eq. (5) é conhecida como *condição de admissibilidade*, que restringe a classe de funções que podem ser *wavelets*. Isso implica que todas as *wavelets* devem obedecer a $\hat{\psi}(0) = 0$ (Goswami; Chan, 2011).

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{\infty} db \int_0^{\infty} \frac{1}{a^2} [(T^{\text{WAV}} f)(a,b)] \psi_{a,b}(t) da, \quad (4)$$

onde C_ψ é uma constante que depende da escolha da *wavelet* e é dada por

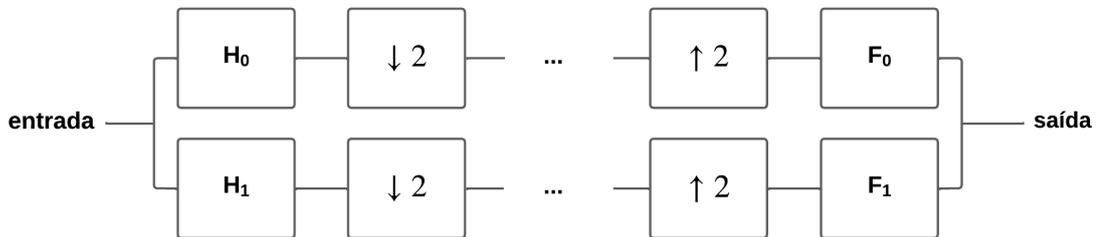
$$C_\psi = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty \quad (5)$$

2.1.1 Banco de Filtros

Algoritmos que implementam a DWT utilizam bancos de filtros para atingir este objetivo. No banco de *análise*, à esquerda na Figura 2, são aplicados um filtro passa-altas H_1 e um filtro passa-baixas H_0 no sinal discretizado, assim é obtido o sinal separado em duas bandas de frequência. No entanto, o número de amostras resultante deste processo é dobrado e a solução para este problema é um processo chamado decimação ($\downarrow 2$). O processo de decimação consiste em remover as componentes ímpares dos sinais resultantes da filtragem, que por sua vez resulta em dois sinais com metade do número de amostras do sinal original e com conteúdo

complementar em frequência. O banco de *síntese*, à direita na Figura 2, se inicia com a operação chamada *upsampling* ($\uparrow 2$), que insere zeros nas componentes ímpares, seguindo para os filtros F_1 e F_0 que devem ser adaptados aos filtros de análise a fim de eliminar os erros no banco de análise. Combinando o resultado de ambos os filtros em uma única matriz, aplica-se uma constante de normalização para que o conteúdo desta matriz sejam componentes unitárias. Esta matriz representa a análise na transformada *wavelet*. Uma vez que esta matriz está normalizada, ela se torna uma matriz ortogonal, cuja inversa é igual a sua transposta. Esta matriz inversa por sua vez representa a síntese da transformada *wavelet*. Executando a convolução da matriz de síntese com o sinal no domínio *wavelet* obtém-se o sinal reconstruído (Strang; Nguyen, 1997).

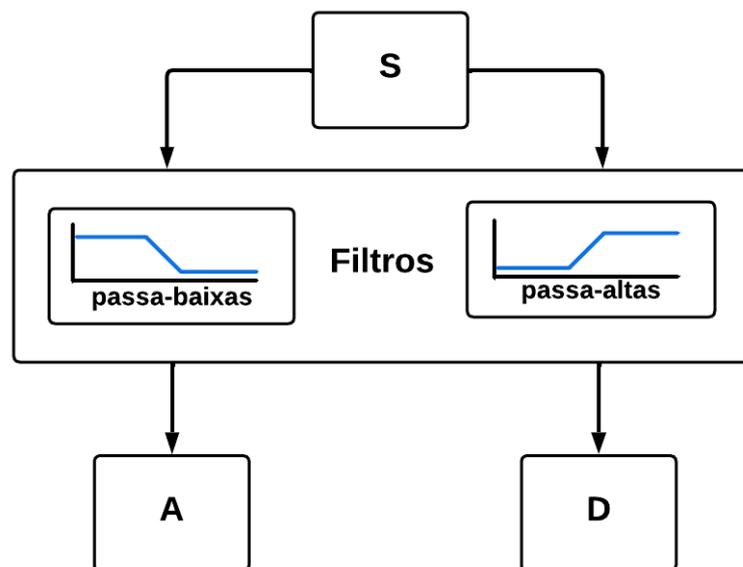
Figura 2 – Banco de filtros de dois canais



Fonte: Adaptado de Strang e Nguye Strang e Nguyen (1997).

Na análise *wavelet*, é comum falar de *aproximação* e *detalhe*. As aproximações são as componentes de alta escala e baixa frequência do sinal. Os detalhes são as componentes de baixa escala e alta frequência. O processo de filtragem é ilustrado na Figura 3:

Figura 3 – Processo de filtragem básico

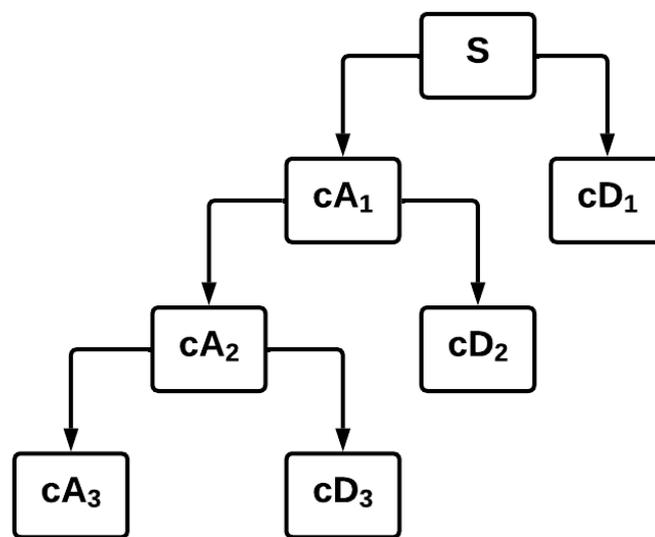


Fonte: Adaptado de Misiti et al. Misiti et al. (1996).

O sinal original S passa pelos filtros descritos anteriormente, o que resulta em dois sinais que são os coeficientes de detalhe (D) e aproximação (A) (Misiti *et al.*, 1996).

O processo de decomposição pode ser iterado em diversos *níveis*, com sucessivas aproximações sendo decompostas, de forma que o sinal é quebrado em diversos componentes com resoluções menores. Isto é chamado de *árvore de decomposição wavelet* como ilustra a Figura 4. Este processo pode ser realizado até que resulte em uma única amostra mas na prática o nível de decomposição deve ser selecionado de acordo com a natureza do sinal ou algum outro critério apropriado (Misiti *et al.*, 1996).

Figura 4 – Árvore de decomposição wavelet



Fonte: Adaptado de Misiti et al. Misiti *et al.* (1996).

2.1.2 Limiarização

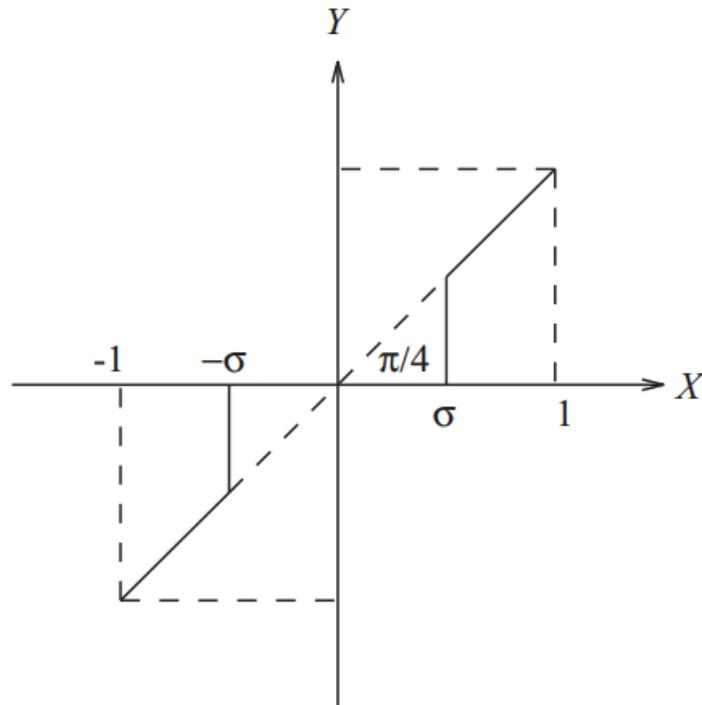
A limiarização é uma das ferramentas mais utilizadas no processamento de sinais *wavelet*. Amplamente utilizada em redução de ruídos, compressão de sinais e imagens, e até mesmo reconhecimento de sinais. Consideram-se 3 métodos simples de limiarização, limiarização rígida, limiarização suave e limiarização por porcentagem, sendo a primeira a mais utilizada. A escolha deve ser feita de acordo com a aplicação (Goswami; Chan, 2011).

Para exemplificar a limiarização rígida que é comumente chamada de *gating*, se o valor de um sinal (ou coeficiente) está abaixo de um valor pré-definido, este valor é definido como zero:

$$y = \begin{cases} x, & \text{para } |x| \geq \sigma; \\ 0, & \text{para } |x| < \sigma; \end{cases} \quad (6)$$

onde σ é o valor do limiar. Uma representação dessa limiarização é mostrada na Figura 5 (Goswami; Chan, 2011).

Figura 5 – Limiarização rígida.



Fonte: Goswami e Chan Goswami e Chan (2011).

A limiarização suave pode ser definida como:

$$y = \begin{cases} \operatorname{sgn}(x)f(|x| - \sigma), & \text{para } |x| \geq \sigma; \\ 0, & \text{para } |x| < \sigma; \end{cases} \quad (7)$$

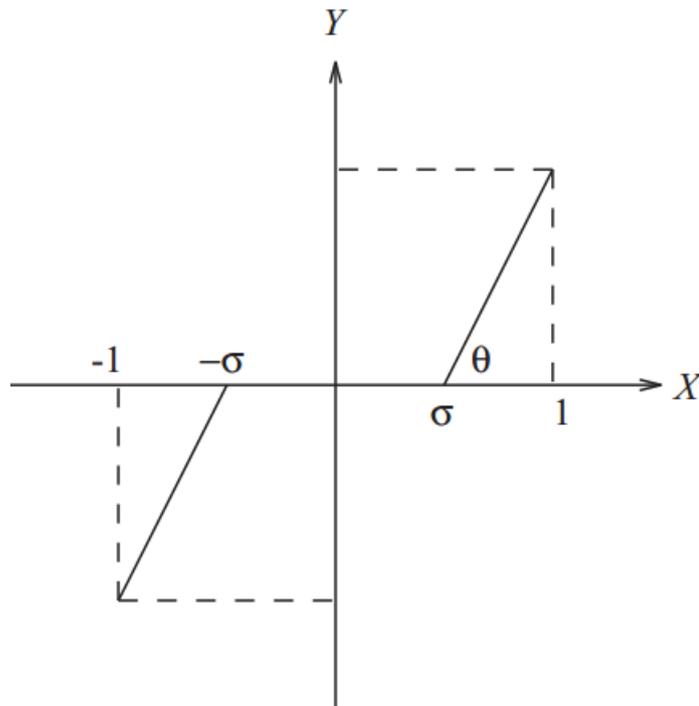
onde $f(x)$ geralmente é uma função linear (uma linha reta com inclinação a ser escolhida; Figura 6) (Goswami; Chan, 2011).

Em alguns casos como na compressão de imagens onde o arquivo comprimido deve estar numa faixa específica de tamanho, é mais vantajoso definir uma porcentagem dos coeficientes *wavelet* como zero para satisfazer esta necessidade. Neste caso o valor do limiar é baseado no histograma e no número total de coeficientes. A regra é a mesma da limiarização rígida uma vez que o valor de limiar é definido (Goswami; Chan, 2011).

2.2 Análise de Componentes Independentes

A ICA é uma técnica computacional e estatística que revela fatores que subjazem conjuntos de variáveis aleatórias, medições, ou sinais. A ICA define um modelo generativo para

Figura 6 – Limiarização suave.

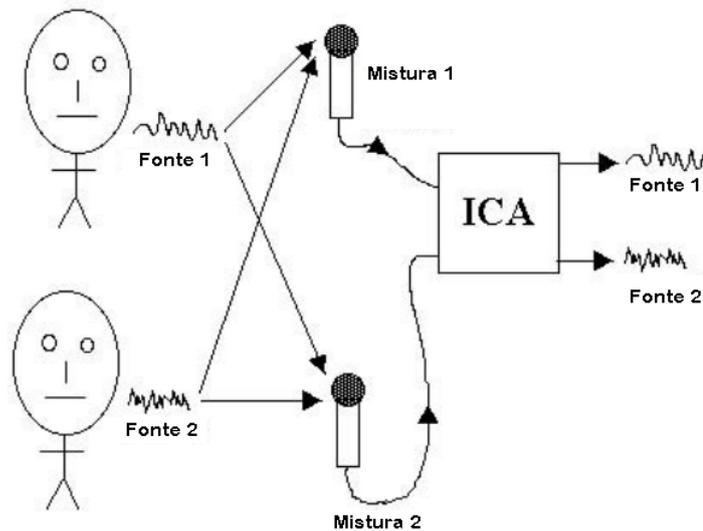


Fonte: Goswami e Chan Goswami e Chan (2011).

os dados observados, os quais geralmente são uma grande base de dados de amostras. No modelo, as variáveis de dados são consideradas misturas lineares ou não-lineares de alguma latente variável desconhecida, sendo também o sistema de mistura desconhecido. As variáveis latentes são consideradas não-gaussianas e mutuamente independentes, e são chamadas de componentes independentes dos dados observados. Estas componentes independentes, também chamadas fontes ou fatores, podem ser encontradas pela ICA (Hyvarinen, 2001).

Um exemplo de problema que a ICA pode resolver é quando duas pessoas conversam num ambiente contendo 2 microfones e a saída de cada microfone é uma mistura das duas vozes, exemplo este também conhecido como problema do coquetel demonstrado na Figura 7. Dada essa mistura a ICA pode recuperar as duas vozes separadamente (Stone, 2004).

Figura 7 – ICA aplicada ao problema do coquetel



Fonte: Adaptado de Stone 2004.

A premissa fundamental da ICA é que os sinais observados em um conjunto de dados são uma combinação linear de sinais originais independentes, também conhecidos como fontes. Fontes estas que são desconhecidas, cada uma das quais é gerada independentemente das outras e que as observações são obtidas através de misturas dessas fontes. Essa suposição é diferente da abordagem de Análise de Componentes Principais (PCA—*Principal Component Analysis*), que assume que os sinais observados são uma combinação linear de sinais originais não necessariamente independentes (Hyvärinen; Oja, 2000).

A ICA é baseada na hipótese de que as fontes são estatisticamente independentes, o que é uma propriedade forte e significa que as distribuições conjuntas das fontes são fatoráveis em distribuições marginais independentes. Essa suposição de independência estatística é fundamental na ICA, pois é a base para a recuperação dos sinais originais a partir das misturas observadas (Hyvärinen; Oja, 2000).

Basicamente, variáveis aleatórias y_1, y_2, \dots, y_n são ditas independentes se a informação do valor de y_i não provê nenhuma informação do valor de y_j para $i \neq j$ (Hyvarinen, 2001).

A ICA também assume que as misturas lineares dos sinais originais são observáveis, enquanto os sinais originais em si são desconhecidos. A ICA assume que apenas as misturas lineares das fontes são observáveis e que as fontes em si são desconhecidas, isso significa que a ICA busca identificar as componentes independentes que compõem as misturas observadas, sem ter acesso direto aos sinais originais (Hyvärinen; Oja, 2000).

Desta forma, o problema consiste em recuperar fontes não observadas, denotadas por $\mathbf{s}(t) = (s_1(t), \dots, s_N(t))^T \in \mathbb{R}^N$, assumindo média zero e estacionária das misturas observadas, $\mathbf{x}(t) = (x_1(t), \dots, x_P(t))^T \in \mathbb{R}^P$, que podem ser escritas como:

$$\mathbf{x}(t) = \mathbf{A}(\mathbf{s}(t)) \quad (8)$$

onde \mathbf{A} é o mapeamento desconhecido de \mathbb{R}^N em \mathbb{R}^P , e onde t denota o *index* da amostra, que pode representar o tempo por exemplo (Comon; Jutten, 2010).

Se o mapeamento \mathbf{A} é invertível (geralmente, a condição $N \leq P$ é então necessária, onde N e P correspondem ao número de fontes e ao número de observações, respectivamente), a identificação de \mathbf{A} ou sua inversa \mathbf{B} , leva diretamente a separação de fontes, isto é, fornece as fontes estimadas $y_i(t)$, da forma:

$$y_i(t) = k_i(s_{\sigma(i)}(t)), i = 1, 2, \dots, N \quad (9)$$

onde σ é a permutação em $\{1, 2, \dots, N\}$, e k_i é o mapeamento correspondente a uma distorção residual (Comon; Jutten, 2010).

Outra restrição a ICA além da independência dos componentes, é que a distribuição dos mesmos deve ser não-gaussiana, isso devido aos cumulantes de maior ordem serem zero em distribuições gaussianas e tal informação é essencial para a estimação do modelo (Hyvarinen, 2001).

Assume-se também que a matriz de mistura \mathbf{A} é quadrada, afim de simplificação da estimação, mas esta suposição não é estritamente necessária, sendo necessário uma generalização da inversa de uma matriz retangular não-singular como solução única de um certo conjunto de equações de acordo com o trabalho de Penrose (1955). Então, após estimar a matriz \mathbf{A} , pode ser calculada sua inversa \mathbf{B} e obter as componentes independentes da forma:

$$\mathbf{s} = \mathbf{B}\mathbf{x} \quad (10)$$

Sem perder a generalidade pode-se assumir que as variáveis de mistura e as componentes independentes possuem média zero. Essa suposição simplifica a teoria e os algoritmos. Caso isto não seja verdade, um pré-processamento pode ser realizado. Isto é possível com a *centralização* das variáveis observáveis, isto é, subtraindo a média de suas amostras. Considerando x' como as observações originais tem-se:

$$x = x' - E\{x'\} \quad (11)$$

A matriz de mistura permanece a mesma durante este processo, e após a estimação a média subtraída pode ser reconstruída fazendo a adição de $\mathbf{A}^{-1}E\{x'\}$ às componentes independentes (Hyvarinen, 2001).

A etapa de *branqueamento* é outro pré-processamento útil para a execução do algoritmo de ICA. A brancura de um vetor aleatório y com média zero, significa a não-correlação entre seus componentes e que suas respectivas variâncias são iguais a unidade. Em outras palavras, a matriz de covariância (assim como a matriz de correlação) de y equivale à matriz identidade. Consequentemente, o branqueamento significa que se transforma linearmente o vetor de dados observados x com a multiplicação por uma matriz \mathbf{V} :

$$z = \mathbf{V}x \quad (12)$$

Um método comumente usado para o branqueamento é a decomposição por autovalor da matriz de covariância:

$$\mathbf{V} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T \quad (13)$$

onde \mathbf{E} é a matriz ortogonal de autovetores $\mathbf{E}\{xx\}^T$ e \mathbf{D} é a matriz diagonal de autovalores (Hyvarinen, 2001).

Considerando que os dados do modelo ICA tenham passado pelo branqueamento pela matriz da Eq. (13) tem-se uma nova matriz de mistura $\tilde{\mathbf{A}}$ que é ortogonal, reduzindo assim o espaço de busca do algoritmo apenas a matrizes ortogonais (Hyvarinen, 2001):

$$z = \mathbf{V}\mathbf{A}s = \tilde{\mathbf{A}}s \quad (14)$$

2.2.1 Ambiguidades

No modelo da ICA existem duas ambiguidades que em alguns casos não serão prejudiciais mas valem ser ressaltados. São elas:

- Não é possível determinar a variância (energia) das componentes independentes. Isto devido a \mathbf{s} e \mathbf{A} serem desconhecidos, qualquer escalar multiplicado em uma das fontes s_i pode sempre ser cancelado dividindo a coluna correspondente a_i de \mathbf{A} . Isto também significa que as componentes podem ser multiplicadas por -1 sem afetar o modelo, o que se traduz na inversão de fase da componente separada (Hyvarinen, 2001).
- Não é possível determinar a ordem das componentes independentes. Devido \mathbf{s} e \mathbf{A} serem desconhecidos, qualquer uma das componentes pode ser dada como a primeira, sendo as fontes encontradas, uma permutação das fontes originais (Hyvarinen, 2001).

2.2.2 Algoritmo

Na implementação dos algoritmos da ICA normalmente se utilizam, além dos pré processamentos como centralização e branqueamento, a *negentropia* como função objetivo a ser maximizada. O conceito de negentropia se deriva da entropia diferencial, que por sua vez pode ser usada para definir a informação mútua entre n variáveis aleatórias. Dado que informação mútua é uma medida natural de dependência, utiliza-se a negentropia visando minimizar a dependência entre as variáveis (Hyvarinen, 1999).

Em sistemas de informação vinculada, que existem quando seus casos possíveis podem ser interpretados como aspectos de um sistema físico, existe uma relação entre a informação

e a entropia de forma que a entropia decresce quando informação é obtida. Essa informação deve ser fornecida por um agente externo, o qual terá sua entropia aumentada. Esta relação pode ser descrita como na Equação 15:

$$S_1 = S_0 - I_{b1} \quad (15)$$

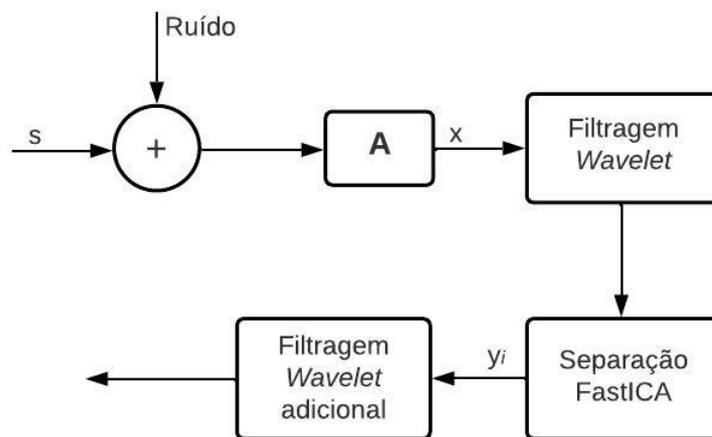
onde S_1 é a entropia final do sistema, S_0 é entropia inicial e I_{b1} é a informação final do sistema. A informação aparece como a parcela negativa da entropia total do sistema físico, assim concluí-se que *informação vinculada = decréscimo na entropia* ou então *informação vinculada = acréscimo na negentropia*, definindo negentropia como o inverso da entropia. Assim, a maximização da negentropia se traduz em maximizar a informação de um dado sistema (Brillouin, 2013).

3 MATERIAIS E MÉTODOS

De acordo com os conceitos definidos por Gerhardt e Silveira (2009), a natureza deste trabalho se dá de forma aplicada uma vez que será realizada a implementação de algoritmo computacional para a resolução do problema. Em relação aos objetivos, este trabalho pode ser classificado como exploratório pois visa conhecer o problema e descrever os passos necessários para sua solução. Quanto aos procedimentos, pode ser dito como experimental considerando a natureza de manipulação dos sinais de estudo e seu objetivo de validar uma ideia. Sua aplicabilidade se dá de maneira quantitativa pois os dados utilizados serão manipulados e posteriormente avaliados em relação às suas características estatísticas, no entanto, utiliza-se também uma abordagem qualitativa na apresentação dos resultados.

A metodologia utilizada se deu a partir do artigo de Zhao et al. (2005), onde é realizada a separação de duas vozes misturadas e corrompidas por ruído branco utilizando o algoritmo FastICA entre duas etapas de filtragem *wavelet* como mostra a Figura 8.

Figura 8 – Diagrama do processamento



Fonte: Autoria própria (2023).

A implementação do processamento proposto neste trabalho foi realizada por meio do software MATLAB®, que permite a implementação de algoritmos complexos de forma simplificada por conter um enorme arcabouço de ferramentas e algoritmos relevantes já implementados em pacotes com funções computacionais de áreas de estudo específicas.

As principais *toolbox* utilizadas no trabalho foram *PCA and ICA Package* que contém a função *fastICA* que realiza a separação por componentes independentes e outras funções auxiliares, *Signal Processing Toolbox* que contém diversas funções para processamento de sinais de forma geral, *Statistics and Machine Learning Toolbox* com diversas funções para análise estatística de sinais e *Wavelet Toolbox* com diversas opções para processamento no domínio *wavelet*.

3.1 Escolha e tratamento dos dados

Para a implementação proposta neste trabalho foi utilizado o banco de vozes *LibriSpeech*, que é derivado do projeto *LibriVox* que acumula leituras de audiolivros. O *LibriSpeech* agrupa os áudios em subconjuntos de forma a permitir uma escolha adequada aos fins desejados.

Nesta implementação foram utilizados 100 áudios de um conjunto de gravações sem ruídos para fins de teste. Os áudios foram escolhidos de forma arbitrária contendo 10 gravações de voz de 10 pessoas diferentes, sendo estas 5 mulheres e 5 homens.

O armazenamento dos arquivos é feito com os áudios em sequência, agrupados de 10 em 10 com cada grupo sendo com as vozes de uma única pessoa. Desta forma, para a formação dos pares é possível fazer a distinção entre os grupos.

Com intuito de padronização dos áudios, limitou-se o número de amostras de cada áudio a 300.000 concatenando repetições dos sinais menores.

3.2 Código

Nesta seção serão apresentadas as etapas do código que fazem as filtrações e separação dos arquivos de voz.

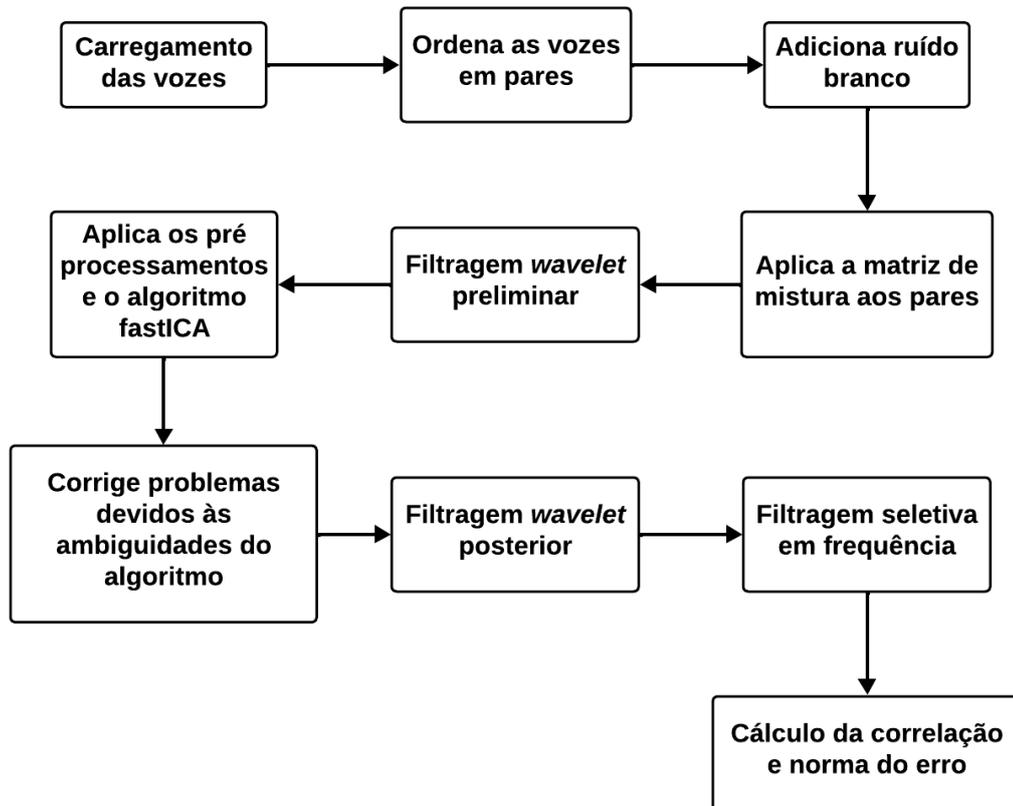
A Figura 9 mostra de maneira mais detalhada as etapas que são submetidos os sinais na execução do código.

O código principal se inicia com o carregamento dos arquivos de voz para a área de trabalho do MATLAB®, em seguida os dados são processados por uma função que ordena os áudios em pares de forma que um par não contenha áudios da mesma pessoa. Assim como na metodologia de Zhao et al. (2005), é adicionado então ruído branco gaussiano em todas as gravações com uma razão sinal ruído (SNR—*Signal to Noise Ratio*) de 15 para simular um ambiente ruidoso que comumente acontece neste tipo de contexto.

3.2.1 Mistura dos sinais

A separação dos sinais é realizada em pares, uma matriz de mistura 2x2 é gerada de forma aleatória para cada par de forma que após a mistura cada um dos vetores resultantes contenha uma parcela dos 2 sinais enquanto a parcela restante se encontra no outro vetor. Esta etapa simula o que seria a gravação de cada voz sob influência de outro interlocutor, uma vez que cada observação do ambiente possui componentes de ambas as vozes.

Figura 9 – Diagrama detalhado do processamento



Fonte: Autoria própria (2023).

3.2.2 Filtragem wavelet e análise do resultado

Sabe-se que ruídos em comunicações são detrimenais ao processo, logo são indesejáveis. A proposta de processamento aqui realizada tem intuito de se aproximar da realidade incluindo ruídos aos sinais, desta forma necessita-se de uma solução robusta que seja capaz de reduzir os níveis de ruído sem causar distorções aos sinais, afetando o algoritmo de separação e o resultado final.

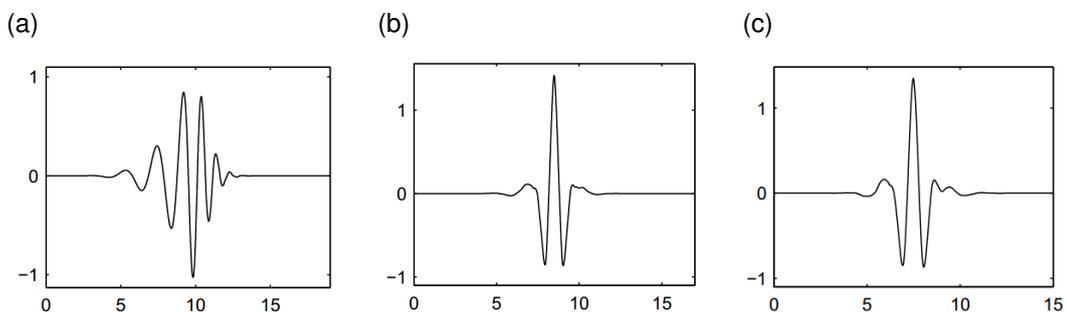
Neste método existem 2 etapas de filtragem, uma inicial para que as vozes estejam mais “limpas” possíveis no algoritmo de separação com a filtragem suave o suficiente para que não se comprometam as componentes úteis da fala. A segunda etapa se dá ao final para remover ruídos residuais das vozes separadas.

As etapas de filtragem são executadas por meio da transformada *wavelet* e aplicação da limiarização aos seus coeficientes de detalhe. A função utilizada para a filtragem pertence ao pacote de ferramentas *Wavelet Toolbox* do MATLAB®, e nela é possível definir diversos parâmetros como o nível de decomposição do sinal, a função *wavelet* e o método de definição dos limiares de filtragem.

Para obter os melhores resultados possíveis foram realizados testes alterando-se a função *wavelet* de ambos os filtros. O processo se deu da seguinte forma:

- fixou-se os parâmetros do filtro preliminar de acordo com o método de Zhao et al. (2005), sendo eles:
 - regra de limiarização SURE (*Stein's unbiased risk estimate*) que segundo Stein (1981) é uma estimativa de risco não enviesada obtida a partir de uma estimativa arbitrária das médias de uma variável aleatória independente cujos resultados são aplicados para suavização por meio do uso de médias móveis.
 - função *wavelet* "sym8" que possui suporte compacto sendo também quase simétrica.
 - nível de decomposição 3 para manter componentes úteis de fala dos sinais.
- foram alteradas as funções *wavelet* do filtro posterior, utilizando as que obtiveram melhor desempenho no aprimoramento de voz de acordo com o trabalho de Mourão e Abreu (2016), sendo essas *db10* da família de *Daubechies*, *coif3* da família de *Coiflets* e *sym8* da família de *Symlets*, mostradas na Figura 10.
- com o resultado do teste anterior, altera-se a função *wavelet* do filtro preliminar realizando testes com duas das funções, uma vez que uma das combinações de parâmetros seria repetida, e com o filtro posterior fixado com os seguintes parâmetros:
 - regra de limiarização BlockJS (*Block James-Stein*): Estima os limiares em bloco de forma adaptativa, considerando os efeitos do bloco localmente e globalmente para sua otimização segundo Cai (2002).
 - utiliza-se a função que obteve melhor resultado nas simulações anteriores.
 - nível de decomposição 5 para uma filtragem mais rígida.

Figura 10 – db10, coif3 e sym8.



Fonte: Adaptado de Misiti et al. 1996.

A partir desta metodologia fez-se a análise dos sinais por meio da correlação e da norma do erro entre os sinais separados e os sinais originais, utilizando os valores mínimos, máximos e médios. Testes de correlação são uns dos mais usados em procedimentos estatísticos, e são

usados como base em diversas aplicações como análise exploratória de dados, modelagem estrutural, engenharia de dados etc (Makowski *et al.*, 2020).

Para o cálculo da norma do erro se utilizou do método de *maior valor singular* (LSV—*Least Singular Value*), o que segundo El Ghaoui (2021) é uma maneira alternativa de medir o tamanho de uma matriz que se baseia em verificar a máxima razão entre a norma da saída e a norma da entrada. Onde o quadrado da norma LSV de uma matriz A é o maior autovalor da matriz simétrica $A^T A$, que é denotado por λ_{max} , assim:

$$\|A\| = \sqrt{\lambda_{max}(A^T A)} \quad (16)$$

3.2.3 Separação das vozes

O algoritmo de separação *FastICA* se inicia pelo pré-processamento em duas etapas, sendo elas a centralização e o branqueamento, que como se sabe são necessárias para preparar o sinal de entrada de forma adequada, garantindo que as fontes possam ser identificadas corretamente e que o algoritmo seja robusto às diferenças de escala e covariância entre os sinais.

Após o pré-processamento o algoritmo segue com o cálculo iterativo dos pesos iniciados de forma aleatória, de acordo com o método convergência escolhido, que para esta aplicação foi o da maximização da negentropia, pelo motivo de sua estrutura teórica ser mais intuitiva e acessível. Com os pesos aplicados às observações obtêm-se as estimativas dos sinais originais.

4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos das simulações no MATLAB® a partir da metodologia proposta no capítulo anterior. Os resultados são apresentados por meio de figuras com a representação dos sinais no tempo, áudios que acompanham as figuras podendo ser reproduzidos com duplo clique sobre a imagem, por tabelas contendo cálculo de correlação e ranqueamento das respectivas simulações utilizando a métrica da norma do sinal de erro entre o sinal original e o resultante. Após a apresentação dos resultados fez-se uma discussão acerca dos mesmos, a fim de oferecer uma avaliação dos aspectos positivos e negativos do que se obteve.

4.1 Simulações

Foram realizadas no total 5 simulações, cada uma utilizando as 100 vezes passando por todas as etapas de processamento descritas, 3 alterando as funções *wavelet* do segundo filtro e 2 similares aplicadas ao primeiro, como mostra o Quadro 1.

Quadro 1 – Parâmetros das simulações

Filtro	Parâmetros	SIM 1	SIM 2	SIM 3	SIM 4	SIM 5
Filtro Preliminar	Regra de limiarização	SURE	SURE	SURE	SURE	SURE
	Função <i>wavelet</i>	sym8	sym8	sym8	db10	coif3
	Nível de decomposição	3	3	3	3	3
Filtro Posterior	Regra de limiarização	BlockJS	BlockJS	BlockJS	BlockJS	BlockJS
	Função <i>wavelet</i>	db10	coif3	sym8	sym8	sym8
	Nível de decomposição	5	5	5	5	5

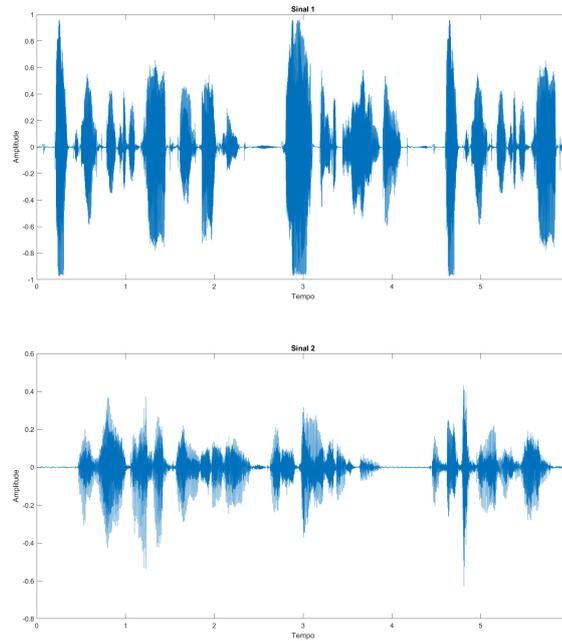
Fonte: Autoria própria (2023).

Serão apresentados a seguir os resultados das simulações sob a forma de figuras, áudios e tabelas de cada uma, sendo a primeira simulação mostrando os resultados das principais etapas do processamento e nas simulações subsequentes contendo apenas a comparação do sinal original em relação ao sinal separado e filtrado. Os sinais utilizados para apresentação são os que obtiveram o melhor resultado no ranqueamento e seu respectivo par, assim como o que obteve o pior desempenho.

Considerando que as misturas são feitas com duas observações, e que os sinais mostrados são o que obteve o melhor resultado no ranqueamento e seu respectivo par, os sinais podem ser diferentes para cada simulação.

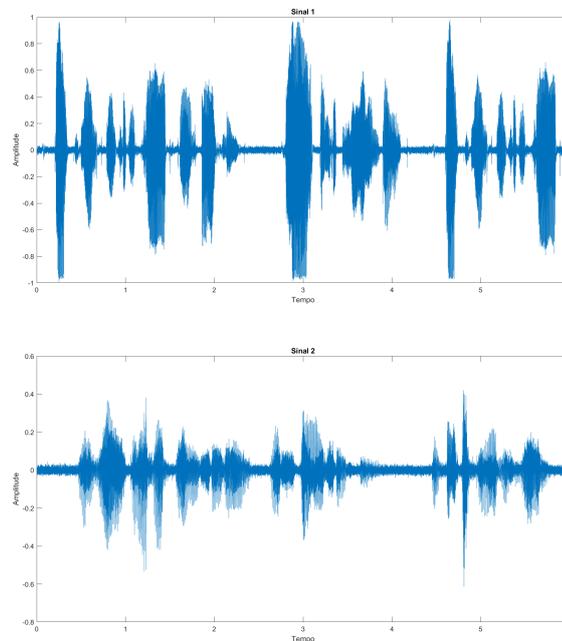
4.1.1 Primeira simulação

A Figura 11 mostra o par de sinais originais que obteve o melhor resultado na primeira simulação.

Figura 11 – Par de sinais originais

Fonte: Autoria própria (2023).

Após a adição de ruído gaussiano branco com uma SNR igual a 15 dB, o resultado pode ser visto na Figura 12.

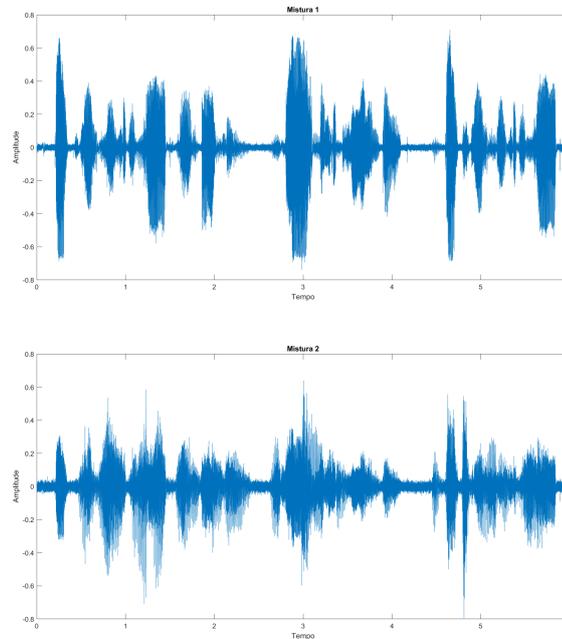
Figura 12 – Par de sinais com adição de ruído

Fonte: Autoria própria (2023).

Após a aplicação da matriz de mistura descrita na equação 17, gerada de forma aleatória, são obtidas duas misturas lineares que representam as gravações por microfones das vozes como pode ser observado na Figura 13.

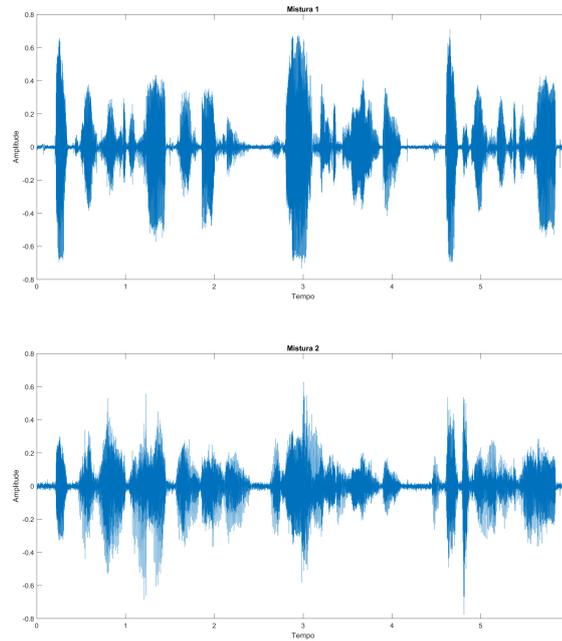
$$\mathbf{A} = \begin{pmatrix} 0,8047 & 0,3117 \\ 0,1953 & 0,6883 \end{pmatrix} \quad (17)$$

Figura 13 – Sinais misturados



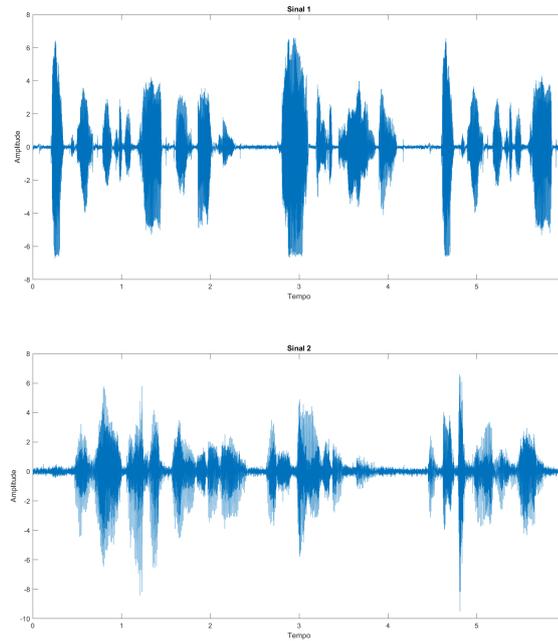
Fonte: Autoria própria (2023).

Após a filtragem preliminar são obtidas as misturas filtradas como mostra a Figura 14. É possível perceber que o nível de ruído é reduzido, no entanto ainda em quantidade considerável, isso devido à filtragem com nível três de decomposição e à regra de limiarização escolhidas de forma a preservar o conteúdo do sinal original.

Figura 14 – Sinais misturados e filtrados

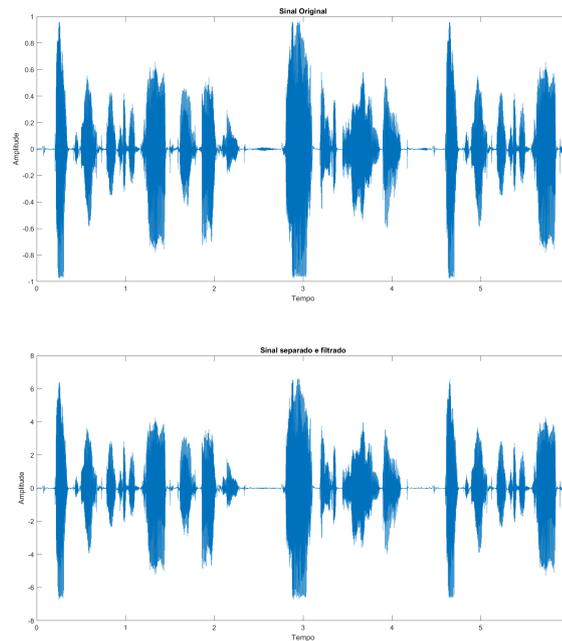
Fonte: Autoria própria (2023).

A aplicação do algoritmo de separação ICA resulta na Figura 15, que são as estimativas dos sinais originais. Nota-se que existe uma diferença de escala dos sinais após a separação, devido às ambiguidades do algoritmo que fazem com que seja incapaz de determinar a amplitude das componentes para a reconstrução exata. Desta forma, também é possível que ocorra uma inversão de fase que é corrigida no código, caso aconteça. Normalmente inversões de fase em aplicações de áudio não são um grande problema, no entanto para que possam ser feitos os cálculos de correlação e norma do erro é necessário que as inversões sejam resolvidas.

Figura 15 – Par de sinais separados

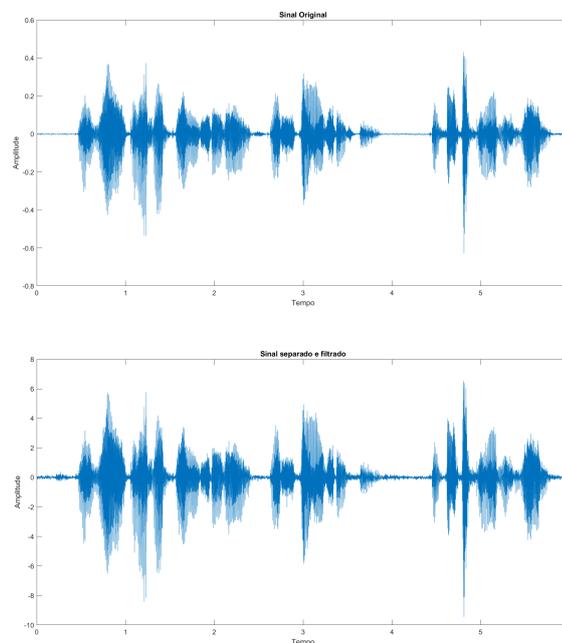
Fonte: Autoria própria (2023).

A fim de comparação tem-se na Figura 16 e na Figura 17 os sinais originais em conjunto com suas respectivas estimativas obtidas pela ICA e filtradas pela limiarização na transformada *wavelet*. Nota-se um resultado bem satisfatório no sinal da Figura 16 pois apresenta pouco ruído e se mantém bastante inteligível. É possível notar também a diferença na escala dos sinais apresentados com o sinal resultante do processamento com amplitude muito maior que o original, devido à ambiguidade no processo de separação.

Figura 16 – Estado original e final do sinal 1

Fonte: Autoria própria (2023).

O sinal resultante na Figura 17 apesar de ser o par do sinal com melhor desempenho, apresenta um ruído mais acentuado ainda mantendo inteligibilidade.

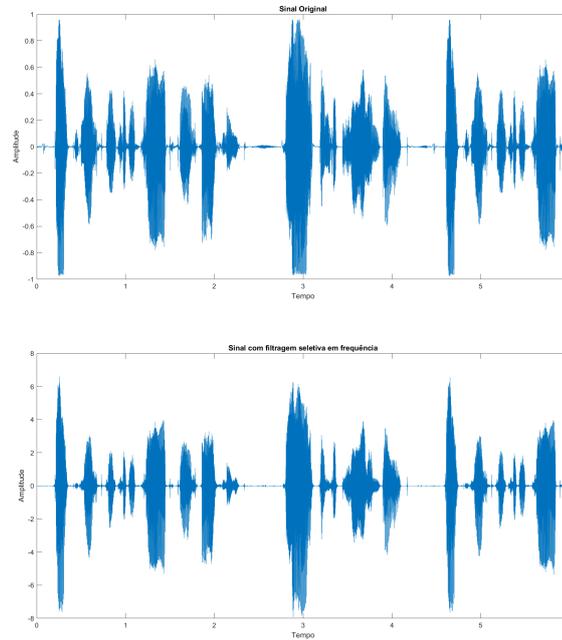
Figura 17 – Estado original e final do sinal 2

Fonte: Autoria própria (2023).

Após a filtragem seletiva em frequência com banda passante entre 300 Hz e 3000 Hz, faixa onde se situa a maior parte do conteúdo em frequência de um sinal de voz, pode-se realizar

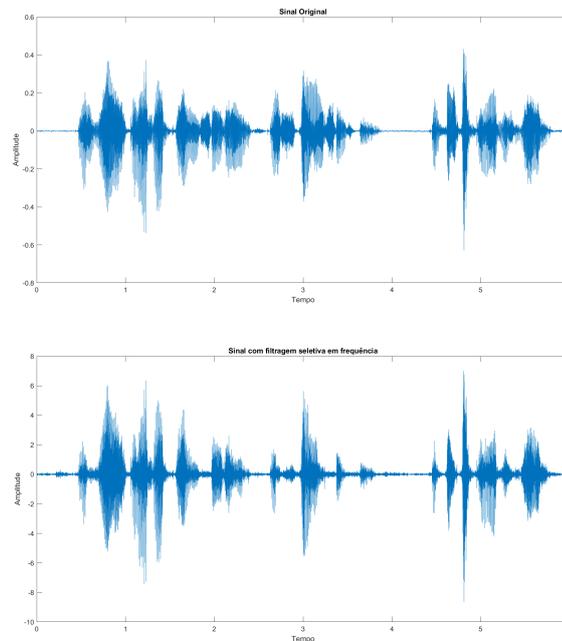
as comparações como na Figura 18 e Figura 19. Em ambos os casos o ruído é atenuado, no entanto isso acontece em detrimento da inteligibilidade dos sinais.

Figura 18 – Estado original e final do sinal 1 com filtragem seletiva



Fonte: Autoria própria (2023).

Figura 19 – Estado original e final do sinal 2 com filtragem seletiva

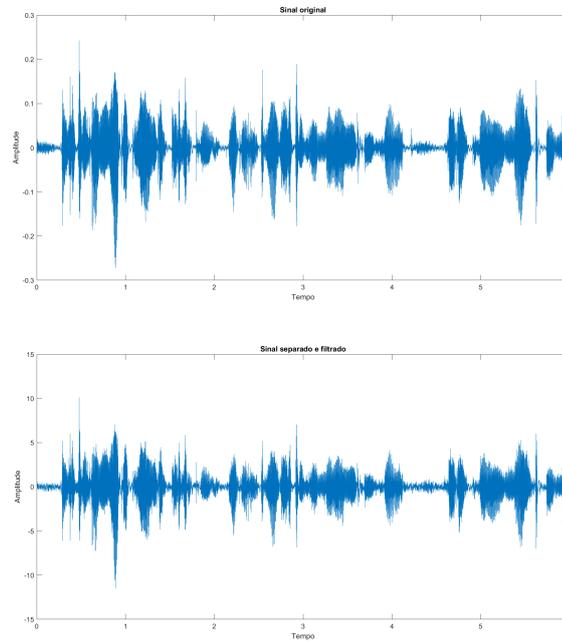


Fonte: Autoria própria (2023).

A Figura 20 mostra o sinal com o pior desempenho quanto ao ranqueamento feito pela norma do erro. Neste caso é possível perceber um ruído que possivelmente é residual da se-

paração incompleta dos sinais. No entanto, a voz ainda mantém um grau de inteligibilidade aceitável.

Figura 20 – Sinal com pior desempenho



Fonte: Autoria própria (2023).

Para uma análise quantitativa têm-se na Tabela 1 os valores de correlações mínima, máxima e média dos 100 sinais. Considerando que a correlação indica quanto um sinal se parece com outro e tem valores entre -1 e 1, pode-se concluir que na média os sinais resultantes são representações fiéis dos originais.

Tabela 1 – Cálculo de correlação

Mínima	Máxima	Média
0,9540	0,9986	0,9864

Fonte: Autoria própria (2023).

A Tabela 2 mostra o resultado do cálculo da norma do erro dos sinais resultantes filtrados em relação aos sinais originais. Ainda que no meio discreto a norma do erro de um sinal não possua uma interpretação atrelada a um fenômeno físico é possível utilizá-la como uma grandeza que mostra quanto dois sinais se diferenciam. Desta forma, pode-se perceber que existe uma variação considerável entre os valores mínimos e máximos, o que pode indicar que certas diferenças, como a amplitude, entre os sinais originais podem ocasionar em uma grande diferença na performance do algoritmo.

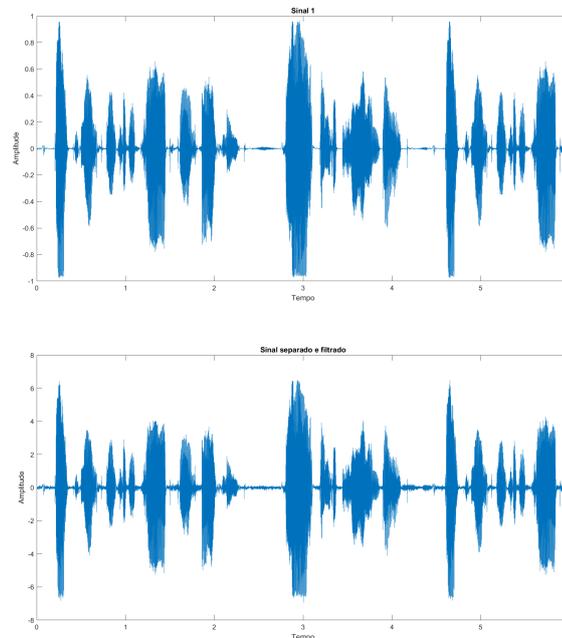
Tabela 2 – Cálculo de norma do erro

Mínima	Máxima	Média
28,8546	166,1641	85,0000

Fonte: Autoria Própria (2023).

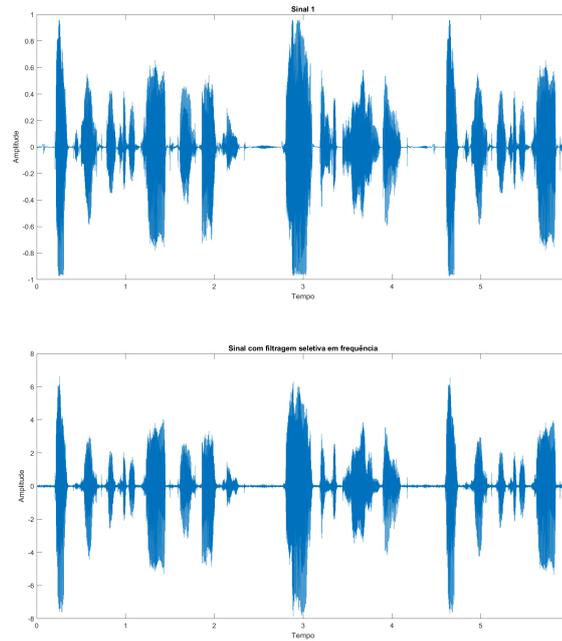
4.1.2 Segunda simulação

Na Figura 21 é apresentada a comparação entre o sinal original e sua respectiva estimativa obtida pela ICA e filtradas pela limiarização na transformada *wavelet*. No sinal resultante é possível perceber um ruído que possivelmente é causado pela separação incompleta, mas apresenta-se bastante inteligível.

Figura 21 – Estado original e final do sinal**Fonte: Autoria própria (2023).**

Na Figura 22 é apresentada a comparação entre o sinal original e sinal com filtragem seletiva em frequência. Mostra como o ruído residual foi bastante atenuado em contraponto com uma pequena perda de inteligibilidade.

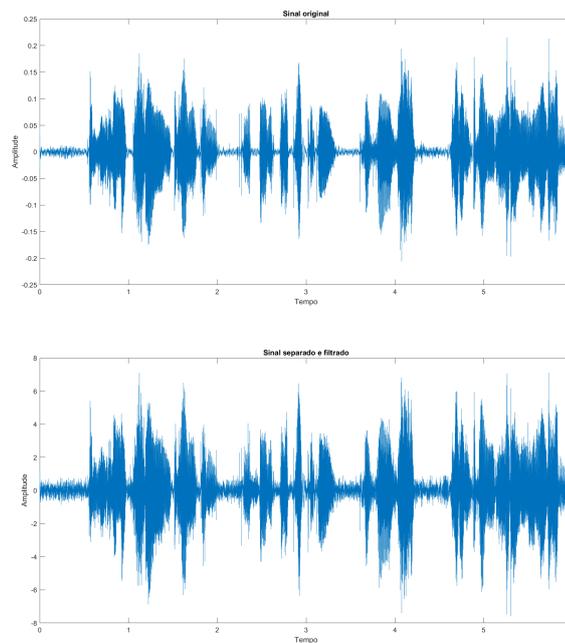
Figura 22 – Sinal original e sinal com filtragem seletiva



Fonte: Autoria própria (2023).

A Figura 23 mostra o sinal com pior desempenho no ranqueamento. Aqui são bem aparentes duas componentes de ruído, possivelmente provenientes do ruído branco adicionado e uma pequena componente residual de voz do respectivo par.

Figura 23 – Sinal com pior desempenho



Fonte: Autoria própria (2023).

O cálculo de correlação da segunda simulação é mostrado na Tabela 3.

Tabela 3 – Cálculo de correlação

Mínima	Máxima	Média
0,9502	0,9975	0,9847

Fonte: A autoria própria (2023).

A Tabela 4 mostra o cálculo da norma do erro da segunda simulação.

Tabela 4 – Cálculo de norma do erro

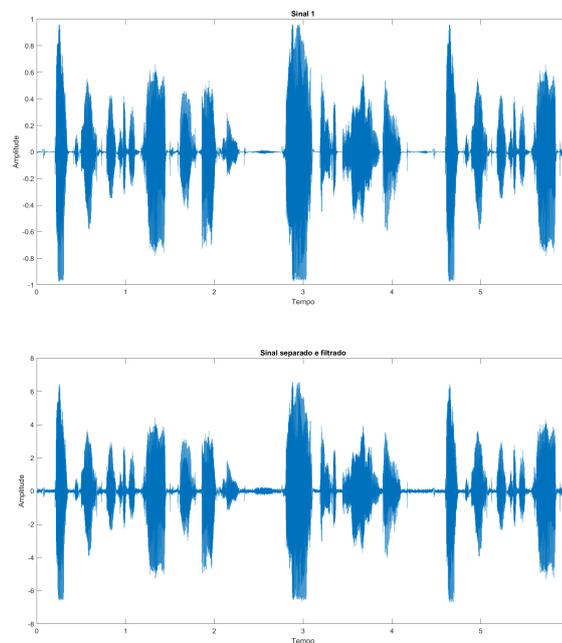
Mínima	Máxima	Média
38,9355	172,9413	90,6180

Fonte: A autoria própria (2023).

4.1.3 Terceira simulação

Os resultados da terceira simulação são mostrados abaixo como realizado anteriormente.

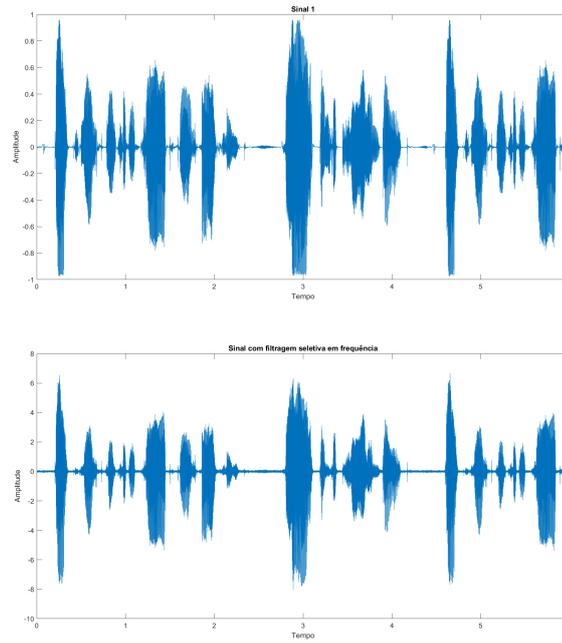
Na Figura 24 é possível notar que existem duas componentes de ruído no sinal resultante, possivelmente uma componente remanescente do respectivo par de voz e outra residual do ruído branco.

Figura 24 – Estado original e final do sinal

Fonte: A autoria própria (2023).

Com a filtragem seletiva em frequência o sinal da Figura 25 se mostra com ruído reduzido e com alguma perda de inteligibilidade.

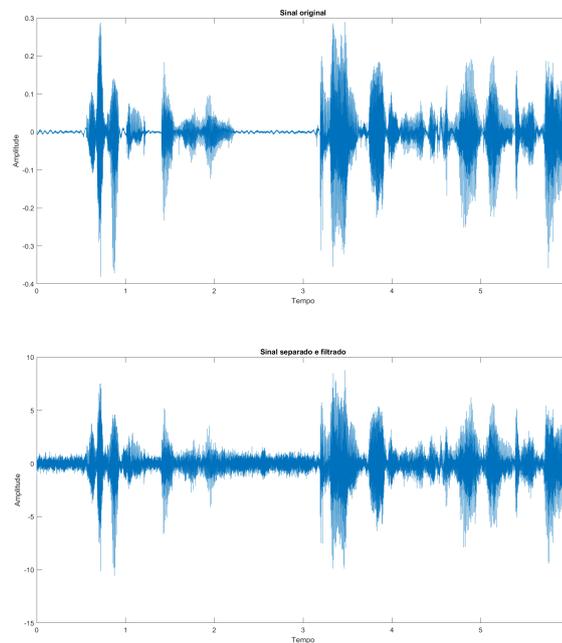
Figura 25 – Sinal original e sinal com filtragem seletiva



Fonte: Autoria própria (2023).

Na Figura 26 o sinal com pior desempenho no ranqueamento aparece com componentes bem perceptíveis do ruído branco e da outra voz misturadas ao sinal.

Figura 26 – Sinal com pior desempenho



Fonte: Autoria própria (2023).

A Tabela 5 mostra o cálculo de correlação dos sinais na terceira simulação.

Tabela 5 – Cálculo de correlação

Mínima	Máxima	Média
0,9335	0,9982	0,9834

Fonte: A autoria própria (2023).

A Tabela 6 mostra o cálculo de norma do erro dos da terceira simulação.

Tabela 6 – Cálculo de norma do erro

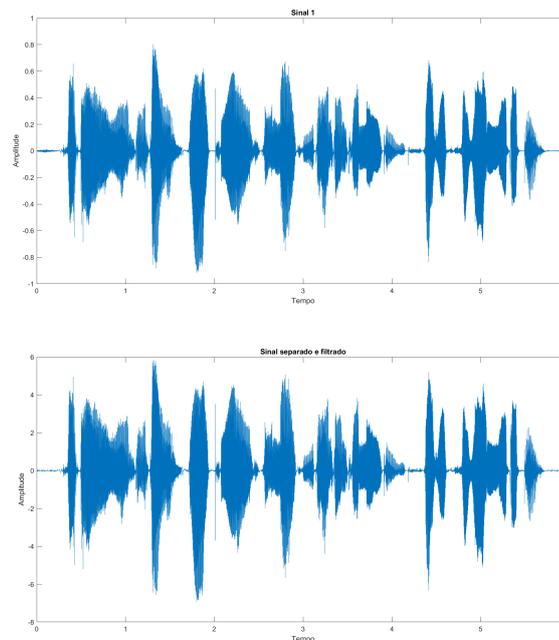
Mínima	Máxima	Média
32,4377	199,7370	93,9896

Fonte: A autoria própria (2023).

4.1.4 Quarta simulação

Os resultados da quarta simulação são mostrados abaixo como realizado anteriormente.

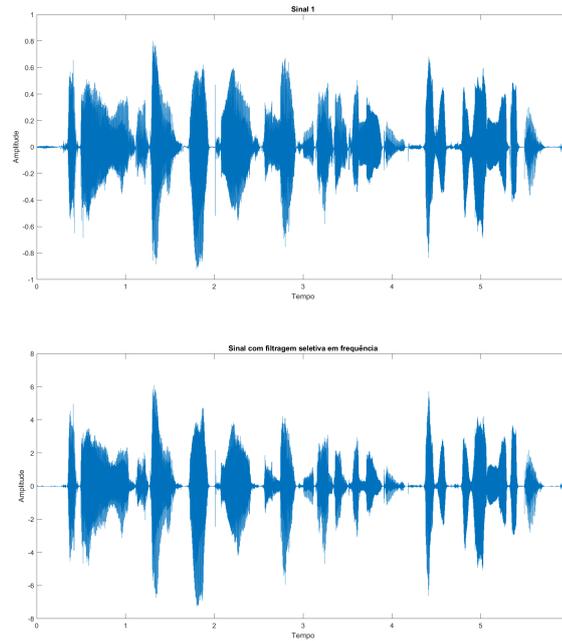
O sinal da Figura 27 possui pouco ruído e apresenta boa inteligibilidade. Vale ressaltar que esta foi a única simulação que o sinal com melhor resultado foi diferente, o que pode ter como causa a combinação dos pares por serem selecionados de forma aleatória em cada simulação ou alguma interação desconhecida entre os parâmetros da simulação e o áudio em questão.

Figura 27 – Estado original e final do sinal

Fonte: A autoria própria (2023).

O sinal da Figura 28 possui diminuição marginal da percepção do ruído de fundo e com alguma perda de inteligibilidade.

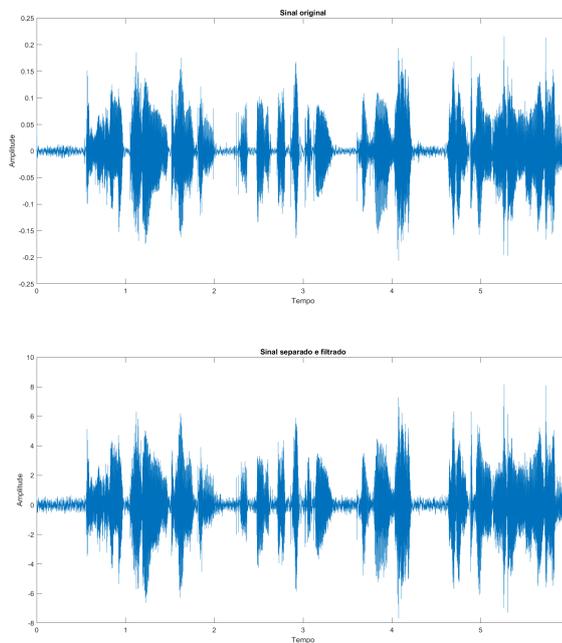
Figura 28 – Sinal original e sinal com filtragem seletiva



Fonte: Autoria própria (2023).

No sinal da Figura 29 é possível perceber claramente um ruído de fundo em conjunto com uma componente remanescente da separação incompleta dos sinais.

Figura 29 – Sinal com pior desempenho



Fonte: Autoria própria (2023).

A Tabela 7 mostra os valores calculados de correlação da quarta simulação.

Tabela 7 – Cálculo de correlação

Mínima	Máxima	Média
0,9561	0,9983	0,9862

Fonte: Autoria própria (2023).

A Tabela 8 apresenta o cálculo de norma do erro dos sinais da quarta simulação.

Tabela 8 – Cálculo de norma do erro

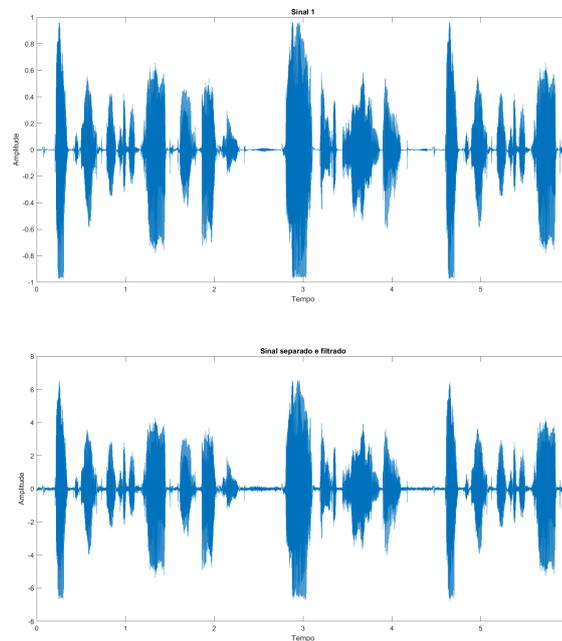
Mínima	Máxima	Média
32,3842	162,2529	85,5416

Fonte: Autoria própria (2023).

4.1.5 Quinta simulação

Os resultados da quinta simulação são mostrados abaixo como realizado anteriormente.

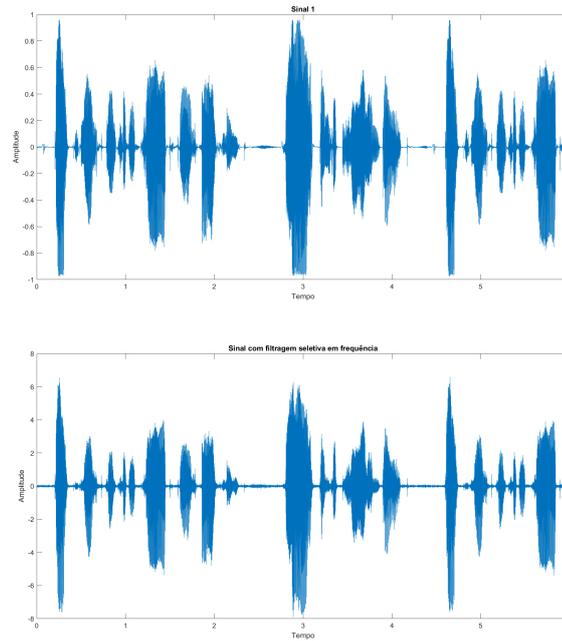
No sinal da Figura 30 é possível notar uma pequena componente remanescente do par de voz em conjunto com um leve ruído de fundo mas ainda mantém boa inteligibilidade.

Figura 30 – Estado original e final do sinal

Fonte: Autoria própria (2023).

Com a filtragem seletiva em frequência, o sinal da Figura 19 quase não se pode mais perceber a componente de voz do par ainda que se mantenha um certo nível de ruído.

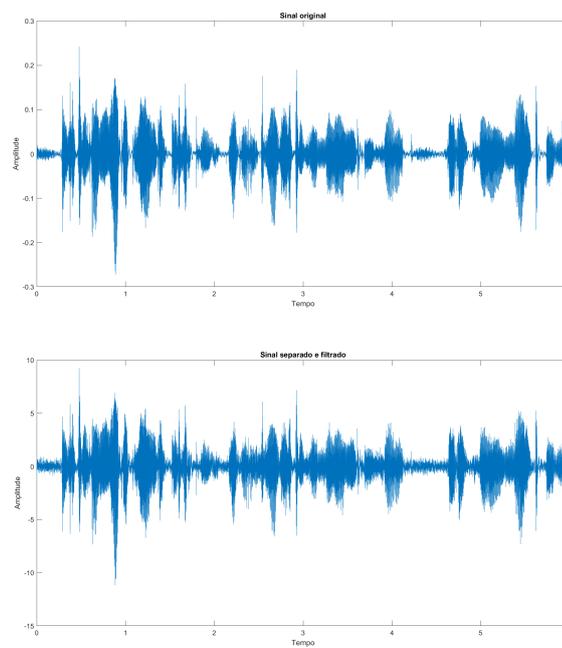
Figura 31 – Sinal original e sinal com filtragem seletiva



Fonte: Autoria própria (2023).

Na Figura32 é possível perceber uma componente bastante presente da segunda voz, ainda que bastante distorcida, em conjunto com um forte ruído de fundo mas ainda mantendo alguma inteligibilidade da voz.

Figura 32 – Sinal com pior desempenho



Fonte: Autoria própria (2023).

A Tabela 9 fornece o cálculo de correlação dos sinais da quinta simulação.

Tabela 9 – Cálculo de correlação

Mínima	Máxima	Média
0,9494	0,9984	0,9842

Fonte: Autoria própria (2023).

A Tabela 10 mostra os valores de norma do erro dos sinais da quinta simulação.

Tabela 10 – Cálculo de norma do erro

Mínima	Máxima	Média
31,3500	174,1749	91,8675

Fonte: Autoria própria (2023).

4.2 Discussões

Para uma visão mais geral do resultado pode-se analisar a Tabela 11, que apresenta os valores de média das medidas de correlação e norma do erro. Nota-se que um ranqueamento utilizando ambas as medidas traz o mesmo resultado. Desta forma, ainda que seja difícil a interpretação do resultado da norma do erro é possível ancorá-lo à correlação neste caso.

Tabela 11 – Sumarização dos cálculos

Medidas	Simulação 1	Simulação 2	Simulação 3	Simulação 4	Simulação 5
Correlação	0,9864	0,9847	0,9834	0,9862	0,9842
Norma do erro	85,0000	90,6180	93,9896	85,5416	91,8675

Fonte: Autoria própria (2023).

Nota-se que o algoritmo se mostra razoável visto que seu melhor resultado mantém boa inteligibilidade, enquanto no pior caso se mantém alguma inteligibilidade. No entanto, é possível notar que em quase todos os casos o sinal indesejado permanece em alguma medida. Seja como um leve ruído como na Figura 30 ou como na Figura 26 onde é possível perceber o segundo interlocutor, ainda que ininteligível.

Em contrapartida, o melhor resultado entre as simulações apresenta uma separação bastante razoável, não contendo parcela perceptível do segundo interlocutor como mostra a Figura 16.

Mediante a análise dos resultados, é possível notar que existe um certo padrão quanto aos melhores e os piores resultados, a maioria resulta em um mesmo sinal. Isto pode indicar que a diferença na escala dos sinais originais tenha grande impacto no resultado, ainda que os sinais tenham sido normalizados antes da mistura.

5 CONCLUSÃO

Este trabalho teve como propósito validar um método de separação de voz em um ambiente ruidoso com a aplicação da transformada *wavelet* e do algoritmo *fastICA* para análise de componentes independentes utilizando o software MATLAB®.

Os objetos de estudo utilizados no trabalho são de grande importância para a ciência, visto que se tornaram bastante relevantes no meio. No entanto, estes temas são pouco ou nada estudados no contexto da graduação, em parte por suas especificidades e complexidades. Assim, um dos interesses em construir este trabalho se materializa trazendo estas áreas da ciência para um trabalho de conclusão de curso.

Na métrica da norma do erro os resultados das simulações para seu valor mínimo oscilaram entre 28,8546 e 39,9355 e máximo entre 162,2529 e 199,7370 mostrando uma boa distinção entre os melhores e piores resultados e relativa consistência no método.

Considerando os resultados em ambos os extremos, conclui-se que os objetivos deste trabalho foram alcançados uma vez que suas etapas foram exitosas e os sinais com melhor e pior desempenho na métrica proposta são inteligíveis.

REFERÊNCIAS

- BIANCHI, M. F. d. **Extração de características de imagens de faces humanas através de wavelets, PCA e IMPCA**. 2006. Tese (Mestrado em Engenharia Elétrica) — Universidade de São Paulo, São Carlos - SP, 2006. Disponível em: <https://www.teses.usp.br/teses/disponiveis/18/18133/tde-10072006-002119/en.php>. Acesso em: 12 jul. 2021.
- BRILLOUIN, L. **Science and information theory**. [S.l.]: Courier Corporation, 2013.
- CAI, T. T. On block thresholding in wavelet regression: Adaptivity, block size, and threshold level. **Statistica Sinica**, Institute of Statistical Science, Academia Sinica, v. 12, n. 4, p. 1241–1273, 2002. ISSN 10170405, 19968507. Disponível em: <http://www.jstor.org/stable/24307026>.
- COMON, P.; JUTTEN, C. **Handbook of Blind Source Separation: Independent component analysis and applications**. [S.l.]: Academic press, 2010.
- DAUBECHIES, I. **Ten lectures on wavelets**. Philadelphia - PA: SIAM, 1992. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=cwdjT3CWY1kC&oi=fnd&pg=PP2&ots=RUEtbB6YRr&sig=OeE-jXUDrKoEiFW6Y0nUFE4CIY8#v=onepage&q&f=false>. Acesso em: 6 jul. 2021.
- GERHARDT, T. E.; SILVEIRA, D. T. **Métodos de pesquisa**. Porto Alegre - RS: Editora da UFRGS, 2009. Disponível em: <http://www.ufrgs.br/cursopgdr/downloadsSerie/derad005.pdf>. Acesso em: 11 ago. 2021.
- GHAOUI, L. E. **Optimization models and applications**. 2021. Disponível em: https://inst.eecs.berkeley.edu/~ee127/sp21/livebook/l_svd_mat_prop.html. Acesso em: 10 out. 2023.
- GOSWAMI, J. C.; CHAN, A. K. **Fundamentals of wavelets: theory, algorithms, and applications**. 2. ed. Hoboken - NJ: John Wiley & Sons, 2011. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=zN2JJPeUYfcC&oi=fnd&pg=PR15&dq=Fundamentals+of+wavelets:+theory,+algorithms,+and+applications&ots=Saz3hrzpPh&sig=PLE34oWiXYzs-fyF--f0SXVUJmw#v=onepage&q=Fundamentals%20of%20wavelets%3A%20theory%2C%20algorithms%2C%20and%20applications&f=false>. Acesso em: 29 jun. 2021.
- HYVARINEN, A. Fast and robust fixed-point algorithms for independent component analysis. **IEEE Transactions on Neural Networks**, v. 10, n. 3, p. 626–634, 1999.
- HYVARINEN, A. **Independent Component Analysis**. [S.l.]: Wiley-Interscience, 2001.
- HYVÄRINEN, A.; OJA, E. Independent component analysis: algorithms and applications. **Neural networks**, Elsevier, v. 13, n. 4-5, p. 411–430, 2000.
- LOIZOU, P. C. **Speech enhancement: theory and practice**. 2. ed. [S.l.]: CRC Press, 2007. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=HgN-DwAAQBAJ&oi=fnd&pg=PP1&ots=gHha-gA9RE&sig=NcsYZIbNOD9Vo7nAZnOFTinjSTk#v=onepage&q&f=false>. Acesso em: 27 jun. 2021.
- MAKINO, S.; LEE, T.-W.; SAWADA, H. **Blind speech separation**. [S.l.]: Springer, 2007. v. 615.
- MAKOWSKI, D. *et al.* Methods and algorithms for correlation analysis in r. **Journal of Open Source Software**, v. 5, n. 51, p. 2306, 2020.
- MISITI, M. *et al.* Wavelet toolbox. **The MathWorks Inc., Natick, MA**, v. 15, p. 21, 1996.

- MOURÃO, E. S.; ABREU, C. C. E. de. Sobre a escolha dos filtros wavelet e da função janela para sistemas de melhoramento de voz baseados na dwt. 2016.
- PENROSE, R. A generalized inverse for matrices. **Mathematical Proceedings of the Cambridge Philosophical Society**, Cambridge University Press, v. 51, n. 3, p. 406–413, 1955.
- STEIN, C. M. Estimation of the mean of a multivariate normal distribution. **The annals of Statistics**, JSTOR, p. 1135–1151, 1981.
- STONE, J. V. Independent component analysis: a tutorial introduction. MIT press, 2004. Disponível em: <http://pzs.dstu.dp.ua/DataMining/ica/bibl/Stone.pdf>. Acesso em: 29 set. 2022.
- STRANG, G.; NGUYEN, T. **Wavelets and filter banks**. USA: Wellesley-Cambridge Press, 1997.
- VINCENT, E.; VIRTANEN, T.; GANNOT, S. **Audio source separation and speech enhancement**. [S.l.]: John Wiley & Sons, 2018. Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=KEhmDwAAQBAJ&oi=fnd&pg=PP2&ots=VvoRS3eCCf&sig=qYKjZiYzYFHWpwag4VyxN1RTclY#v=onepage&q&f=false>. Acesso em: 27 jun. 2021.
- ZHAO, C. *et al.* An effective method on blind speech separation in strong noisy environment. *In: Proceedings of 2005 IEEE International Workshop on VLSI Design and Video Technology, 2005*. [S.l.: s.n.], 2005. p. 211–214.