

UNIVERSIDADE FEDERAL DO ABC
BACHARELADO EM ENGENHARIA DE INFORMAÇÃO

CAROLINE CUSTÓDIO

**ESTUDO DA SEPARAÇÃO CEGA DE FONTES EXPLORANDO A ESTRUTURA
TEMPORAL DOS SINAIS**

TRABALHO DE GRADUAÇÃO

Santo André

2020

CAROLINE CUSTÓDIO

**ESTUDO DA SEPARAÇÃO CEGA DE FONTES EXPLORANDO A ESTRUTURA
TEMPORAL DOS SINAIS**

Trabalho de Graduação apresentado como requisito parcial para conclusão do curso de Engenharia
da Informação da Universidade Federal do ABC

Orientadora: Prof.^a Dr.^a Aline de Oliveira Neves Panazio

Santo André

2020

SUMÁRIO

1	INTRODUÇÃO.....	2
2	SEPARAÇÃO CEGA DE FONTES	3
2.1	MODELAGEM DO PROBLEMA DE BSS.....	3
2.2	O EFEITO “COCKTAIL-PARTY”	4
3	ANÁLISE POR COMPONENTES INDEPENDENTES	5
3.1	RESTRICÇÕES DA ICA.....	5
3.2	AMBIGUIDADES DA ICA.....	7
3.3	BRANQUEAMENTO APLICADO A ICA	9
4	MÉTODOS QUE EXPLORAM A ESTRUTURA TEMPORAL DOS DADOS	12
4.1	SEPARAÇÃO DE FONTES UTILIZANDO CORRELAÇÃO TEMPORAL	12
4.2	AMUSE.....	14
4.3	SOBI.....	15
4.4	MÉTODO DO GRADIENTE DESCENDENTE	16
4.5	ALGORITMO DE JACOBI	16
4.6	TDSEP.....	17
5	RESULTADOS EXPERIMENTAIS.....	18
5.1	SEPARAÇÃO DE FONTES ALEATÓRIAS	18
5.2	SEPARAÇÃO DE SINAIS DE ÁUDIO.....	26
6	CONCLUSÃO.....	28
7	REFERÊNCIAS BIBLIOGRÁFICAS	29

1 INTRODUÇÃO

Análise por Componentes Independentes (do inglês *Independent Component Analysis – ICA*) consiste em um conjunto de técnicas utilizadas para encontrar fatores subjacentes ou componentes de dados estatísticos multivariados (multidimensionais), partindo do pressuposto de que essas componentes são estatisticamente independentes entre si [1].

A ICA foi proposta por J. Héroult, C. Jutten, e B. Ans [2] no início da década de 80 no contexto de Separação Cega de Fontes (do inglês *Blind Source Separation – BSS*) que, por sua vez, refere-se a um tipo de problema cujo objetivo é recuperar um conjunto de fontes, utilizando como insumo um conjunto de misturas obtidas a partir dessas fontes, mas sem conhecimento prévio das fontes de origem ou sobre a forma em que elas foram misturadas. Inicialmente, essas aplicações se restringiam ao campo de modelagem de redes neurais. Entretanto, com o passar dos anos, as técnicas de ICA começaram a ser utilizadas para diversas finalidades, por exemplo, na separação de sinais de áudio, extração de *features*, análise de séries temporais e mineração de dados [1].

As técnicas de ICA se baseiam em duas premissas fundamentais para suas aplicações. A primeira delas é que as fontes buscadas devem ser estatisticamente independentes entre si, conforme mencionado anteriormente. Já a segunda consiste no critério da não-gaussianidade das fontes, ou seja, as fontes buscadas precisam ter distribuições de probabilidade não-gaussianas. Nos casos em que apenas uma das fontes buscadas é gaussiana, ainda é possível utilizar uma das técnicas de ICA para a separação das fontes, caso contrário, outro método precisa ser utilizado. Com essa finalidade, existem os métodos que exploram a estrutura temporal dos dados para fazer a separação cega de fontes. Esses métodos também pressupõem que as fontes são estatisticamente independentes entre si. Contudo, ao invés de utilizar estatísticas de ordem superior como em ICA, esses métodos se baseiam no cálculo da correlação temporal das fontes para recuperar a independência entre elas [1].

Neste trabalho, além do estudo da separação cega de fontes explorando a estrutura temporal dos dados, também tivemos como objetivo entender os conceitos básicos utilizados em ICA. Por fim, após o entendimento teórico dos métodos mencionados, implementamos e comparamos o desempenho dos algoritmos AMUSE, SOBI e TDSEP dentro dos cenários de separação de sinais de áudio e de fontes artificiais geradas de forma aleatória.

2 SEPARAÇÃO CEGA DE FONTES

O problema de separação cega de fontes (BSS) busca estimar um conjunto de fontes, por exemplo, um conjunto de sinais neurais captados de diferentes áreas do cérebro, a partir de misturas lineares dessas fontes. O termo “cega” significa que não há informações suficientes sobre as fontes origem ou mesmo como a mistura foi obtida [1].

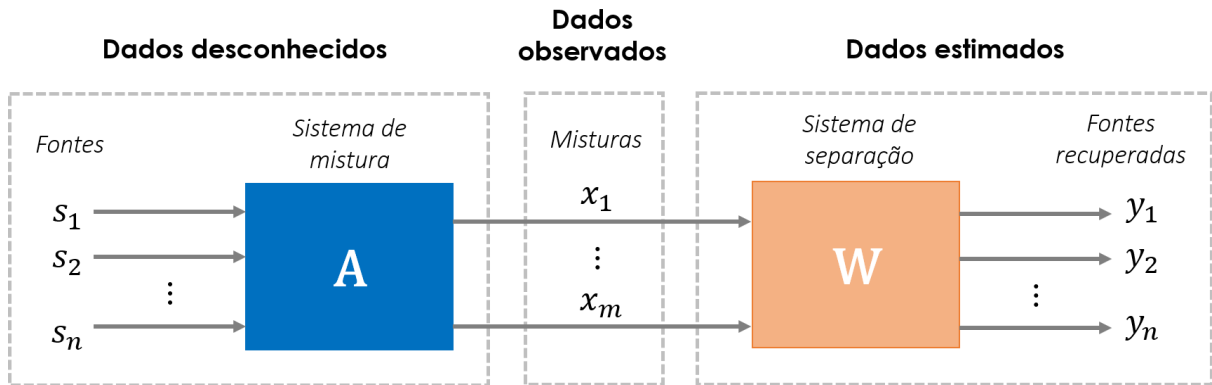


Figura 1 - O problema de separação cega de fontes

A Figura 1 ilustra o problema de separação cega de fontes. Conforme é mostrado, temos um conjunto de n fontes denotadas por $s_1(t), s_2(t), \dots, s_n(t)$ que passam por um sistema linear de mistura A , resultando assim nas m misturas $x_1(t), x_2(t), \dots, x_m(t)$. Tanto as fontes de origem quanto o sistema de mistura utilizado são desconhecidos. Sendo assim, com o objetivo de estimar as fontes $s_i(t)$, o problema busca encontrar um sistema de separação W que, a partir das misturas observadas, resulte no conjunto de n fontes recuperadas $y_1(t), y_2(t), \dots, y_n(t)$.

2.1 MODELAGEM DO PROBLEMA DE BSS

Denotando as misturas observadas por um vetor $\mathbf{x} = [x_1(t), x_2(t), \dots, x_m(t)]^T$, as fontes originais por um vetor $\mathbf{s} = [s_1(t), s_2(t), \dots, s_n(t)]^T$ e o sistema linear de mistura por uma matriz $\mathbf{A}_{n \times m}$, chegamos na seguinte equação para o problema de separação cega de fontes:

$$\mathbf{x}(t) = \mathbf{A} \cdot \mathbf{s}(t), \quad (2.1)$$

onde a matriz \mathbf{A} é conhecida como a matriz de mistura do problema [1].

Dado que tanto as fontes origem \mathbf{s} quanto a matriz de mistura \mathbf{A} são desconhecidas, então a equação do problema pode ser reescrita denotando-se o sistema de separação buscado por uma matriz $\mathbf{W}_{n \times m}$ e as fontes recuperadas por um vetor $\mathbf{y} = [y_1(t), y_2(t), \dots, y_n(t)]^T$ da seguinte forma [1]:

$$\mathbf{y}(t) = \mathbf{W} \cdot \mathbf{x}(t). \quad (2.2)$$

Portanto, o problema de separação cega de fontes tem como objetivo encontrar uma matriz de separação $\mathbf{W}_{n \times m}$ que, ao ser multiplicada pelo vetor \mathbf{x} com as m misturas observadas resulte no vetor \mathbf{y} contendo as n fontes buscadas pelo problema.

2.2 O EFEITO “COCKTAIL-PARTY”

O efeito “*cocktail-party*” descreve um cenário onde existem n fontes independentes de áudio em uma sala, assim como m microfones dispostos em diferentes lugares dessa sala gravando os sinais de áudio recebidos [1]. Essas fontes poderiam ser, por exemplo, pessoas conversando em uma festa com música de fundo tocando e, como resultado, teremos um conjunto de m misturas gravadas pelos microfones, sendo que cada microfone deverá capturar com maior ou menor intensidade cada fonte dependendo da posição que se encontra na sala.

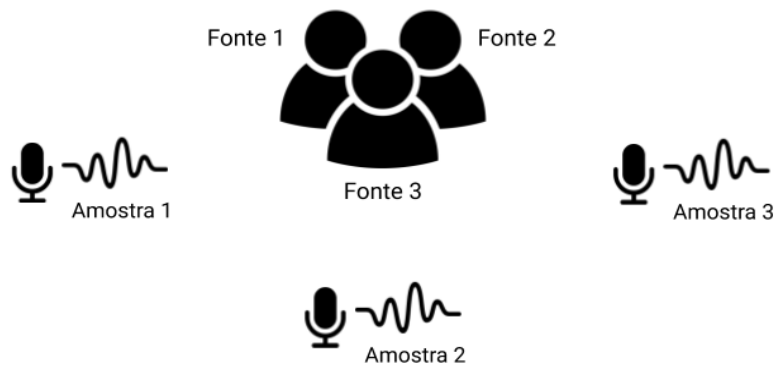


Figura 2 - O efeito “*cocktail-party*”

Para exemplificar o problema, podemos considerar $n = m = 3$, assumindo que não há reverberação na sala, conforme ilustrado na Fig. 2. Denotando por $x_1(t), x_2(t)$ e $x_3(t)$ as misturas obtidas pelos microfones 1, 2 e 3, respectivamente, e por $s_1(t), s_2(t)$ e $s_3(t)$ as fontes buscadas, então podemos representar esse problema através de um sistema de três equações lineares, onde cada mistura encontrada é uma soma ponderada das três fontes a serem recuperadas:

$$\begin{cases} a_{11}s_1(t) + a_{12}s_2(t) + a_{13}s_3(t) = x_1(t) \\ a_{21}s_1(t) + a_{22}s_2(t) + a_{23}s_3(t) = x_2(t) \\ a_{31}s_1(t) + a_{32}s_2(t) + a_{33}s_3(t) = x_3(t) \end{cases} \quad (2.3)$$

onde os parâmetros a_{ij} com $i, j = 1, 2$ e 3 , são os coeficientes de mistura e seus valores, nesse caso, dependem das distâncias dos microfones em relação a cada pessoa na sala [1].

Se os coeficientes de mistura fossem conhecidos, então bastaria inverter o sistema de equações lineares em (2.3), obtendo assim as fontes $s_i(t)$ em função das misturas disponíveis.

Contudo, dado que não temos qualquer informação a respeito dessas fontes ou dos coeficientes de mistura, trata-se de um problema de separação cega de fontes. Uma solução proposta para esse problema consiste em partir da hipótese de que os sinais buscados são estatisticamente independentes entre si. Essa solução dá origem a um conjunto de técnicas conhecidas como Análise por Componentes Independentes (ICA).

3 ANÁLISE POR COMPONENTES INDEPENDENTES

Análise por Componentes Independentes (ICA) corresponde a um conjunto de técnicas utilizadas para resolver o problema de separação cega de fontes partindo de algumas premissas para sua implementação. Conforme descrito por A. Hyvärinen, J. Karhunen e E. Oja [1], seguem as restrições nas quais baseiam-se as técnicas de ICA.

3.1 RESTRIÇÕES DA ICA

i. Assume-se que as componentes são estatisticamente independentes entre si

Essa é a premissa mais importante na qual se baseia a análise por componentes independentes. Ao assumirmos que as componentes (ou fontes) são estatisticamente independentes entre si, significa que não é possível obtermos informação sobre uma componente a partir de outra. Considerando a equação do problema de BSS definida em (2.2), então para cada par de fontes recuperadas $y_i(t)$ e $y_j(t)$, sendo $i \neq j$, as fontes devem ser independentes entre si.

A definição de independência se baseia na distribuição de probabilidade das fontes. Denotando por $p(y_1(t), y_2(t), \dots, y_n(t))$ a distribuição de densidade de probabilidade conjunta das fontes recuperadas $y_1(t), y_2(t), \dots, y_n(t)$, e por $p_i(y_i)$ a distribuição de probabilidade marginal da i -ésima fonte $y_i(t)$, então dizemos que o conjunto de fontes recuperadas são estatisticamente independentes entre si se, e apenas se, a sua distribuição de probabilidade conjunta for igual a multiplicação das distribuições marginais de cada fonte, conforme é descrito na equação abaixo:

$$p(y_1(t), y_2(t), \dots, y_n(t)) = p_1(y_1) p_2(y_2) \dots p_n(y_n). \quad (3.1)$$

Uma forma mais genérica de definir independência estatística é utilizando as respectivas esperanças das fontes analisadas. Sabendo que a esperança $E\{g(y_i(t))\}$ de uma função $g(y_i(t))$, sendo $y_i(t)$ a i -ésima fonte com distribuição de probabilidade $p_i(y_i)$, é dada por:

$$E\{g(y_i(t))\} = \int_{-\infty}^{\infty} g(y_i(t)) p_i(y_i) dy_i, \quad (3.2)$$

então, dizemos que o conjunto de fontes recuperadas $y_1(t), y_2(t), \dots, y_n(t)$ são estatisticamente independentes entre si, quando a respectiva função de distribuição de probabilidade cumulativa pode ser fatorada da seguinte forma:

$$\begin{aligned}
& E\{g_{y_1}(y_1) g_{y_2}(y_2) \dots g_{y_n}(y_n)\} \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [g_{y_1}(y_1) g_{y_2}(y_2) \dots g_{y_n}(y_n)] [p(y_1, y_2, \dots, y_n)] dy_1 dy_2 \dots dy_n \\
&= \int_{-\infty}^{\infty} g_{y_1}(y_1) p_1(y_1) dy_1 \int_{-\infty}^{\infty} g_{y_2}(y_2) p_2(y_2) dy_2 \dots \int_{-\infty}^{\infty} g_{y_n}(y_n) p_n(y_n) dy_n \\
&= E\{g_{y_1}(y_1)\} E\{g_{y_2}(y_2)\} \dots E\{g_{y_n}(y_n)\},
\end{aligned} \tag{3.3}$$

onde $g_{y_1}(y_1), g_{y_2}(y_2), \dots, g_{y_n}(y_n)$ são quaisquer funções de y_1, y_2, \dots, y_n , respectivamente, para as quais as esperanças $E\{g_{y_1}(y_1)\} E\{g_{y_2}(y_2)\} \dots E\{g_{y_n}(y_n)\}$ existem.

A partir da equação (3.3) podemos observar que a independência estatística é uma propriedade muito mais forte do que a decorrelação entre duas ou mais variáveis. Isso ocorre porque a decorrelação é um caso especial da independência estatística, ou seja, ela também pode ser definida através da equação (3.3), considerando apenas os casos em que as funções $g_{y_1}(y_1), g_{y_2}(y_2), \dots, g_{y_n}(y_n)$ são lineares.

Sendo assim, a decorrelação entre variáveis leva em conta apenas estatísticas de segunda ordem (correlações e covariâncias) das mesmas. A independência estatística, por outro lado, leva em conta também as estatísticas de ordem superior dos sinais analisados.

ii. No máximo uma componente independente pode ter distribuição de probabilidade gaussiana

Variáveis com distribuição de probabilidade gaussiana são completamente definidas pelos seus momentos de primeira e segunda ordem. Seja uma fonte gaussiana $y_i(t)$ e calculando a média m_{y_i} e a variância de $y_i(t)$ fazendo a substituição, respectivamente, de $g(y_i(t)) = y_i(t)$ e $g(y_i(t)) = y_i^2(t)$ em (3.2), então a fonte $y_i(t)$ é completamente caracterizada, já que a mesma é gaussiana e todos os seus momentos de ordem superior dependem da média m_{y_i} e da covariância c_{y_i} de $y_i(t)$.

Na restrição (i) foi explicado que as técnicas de ICA partem do princípio de independência estatística para poder recuperar as fontes iniciais, o qual por sua vez baseia-se no cálculo dos momentos de ordem superior do sinal. Sendo assim, no caso em que mais de uma componente possui distribuição de probabilidade gaussiana, não é possível obter

informações novas sobre as fontes utilizando os seus momentos de ordem superior, impossibilitando assim que a independência dessas fontes seja recuperada utilizando ICA.

Para resolver esse problema, podemos utilizar uma classe de métodos que não partem da premissa da não-gaussianidade dos sinais. Ao invés de utilizar as estatísticas de ordem superior, essa classe de métodos utiliza a estrutura temporal das componentes analisadas. Esses métodos serão estudados nos tópicos subsequentes.

iii. Para simplificação, assume-se que a matriz de mistura desconhecida é uma matriz quadrada (Caso determinado de ICA)

Nos casos determinados de ICA partimos do princípio de que o número n de fontes buscadas é igual ao número m de misturas disponíveis, resultando assim em uma matriz de mistura $\mathbf{A}_{m \times n}$, onde o número de linhas m é igual ao número de colunas n . Essa simplificação também foi utilizada para todas as simulações realizadas neste trabalho.

iv. Para simplificação, assume-se que tanto as variáveis de mistura quanto as componentes independentes possuem média nula

Sem perda de generalidade, também podemos utilizar essa premissa para simplificar a aplicação da ICA. Nos casos em que as variáveis não possuem a média nula, costuma-se fazer um pré-processamento para subtrair suas médias, resultando assim na centralização das variáveis observadas. Por exemplo, seja $\mathbf{x}(t)$ o vetor com as amostras das misturas originais observadas, então para centralizar as variáveis de mistura fazemos:

$$\mathbf{x}'(t) = \mathbf{x}(t) - E\{\mathbf{x}(t)\}, \quad (3.4)$$

onde $E\{\mathbf{x}(t)\}$ é a esperança do vetor \mathbf{x} e $\mathbf{x}'(t)$ é o vetor $\mathbf{x}(t)$ centralizado.

Consequentemente, as fontes independentes $s_i(t)$ também são centralizadas, dado que:

$$E\{\mathbf{s}(t)\} = \mathbf{A}^{-1}E\{\mathbf{x}(t)\}, \quad (3.5)$$

onde \mathbf{A}^{-1} é a inversa da matriz de mistura \mathbf{A} que por sua vez permanece a mesma após a centralização das variáveis.

3.2 AMBIGUIDADES DA ICA

A partir da definição do problema de separação cega de fontes em (2.2), podemos destacar algumas ambiguidades ou indeterminações na recuperação das fontes $y_1(t), y_2(t), \dots, y_n(t)$, conforme descrito nos itens (i) e (ii) abaixo.

i. Não é possível determinar as amplitudes das componentes independentes

Isso ocorre pois qualquer escalar multiplicado por uma das fontes $s_i(t)$ da equação (2.2) pode ser anulado dividindo-se os elementos da coluna a_j correspondente na matriz \mathbf{A} pelo mesmo escalar, resultando assim na mesma mistura $x_i(t)$ obtida. Sendo assim, para corrigir essa ambiguidade na magnitude da componente, costuma-se assumir que cada fonte possui variância unitária, ou seja, $E\{s_i^2\} = 1$. Resta ainda a ambiguidade do sinal da componente independente, o que acaba sendo na maioria das vezes insignificante para as aplicações da ICA [1].

ii. Não é possível determinar a ordem das componentes independentes

Após a aplicação de uma das técnicas de ICA para obtenção do vetor $\mathbf{y} = [y_1(t), y_2(t), \dots, y_n(t)]^T$, as fontes recuperadas $y_1(t), y_2(t), \dots, y_n(t)$ não correspondem necessariamente às fontes origem $s_1(t), s_2(t), \dots, s_n(t)$ nessa ordem [1]. Por exemplo, pode ser que a fonte origem $s_1(t)$ seja obtida através de $y_2(t)$ e $s_2(t)$ seja recuperada através da fonte $y_2(t)$ ou, ainda, $-y_2(t)$, devido à ambiguidade do sinal das fontes recuperadas.

As ambiguidades de ordem e de sinal também ocorrem quando utilizamos os métodos que exploram a estrutura temporal das fontes. Nas Figuras 3 e 4 abaixo são mostrados a recuperação de duas fontes $s_1(t)$ e $s_2(t)$ geradas de forma aleatória com distribuições uniformes e as suas respectivas fontes $y_2(t)$ e $-y_1(t)$ recuperadas utilizando-se um dos algoritmos estudados que exploram a estrutura temporal dos sinais, o AMUSE, utilizando um atraso no tempo $\tau = 2$. Os algoritmos implementados neste trabalho serão detalhados no tópico seguinte.

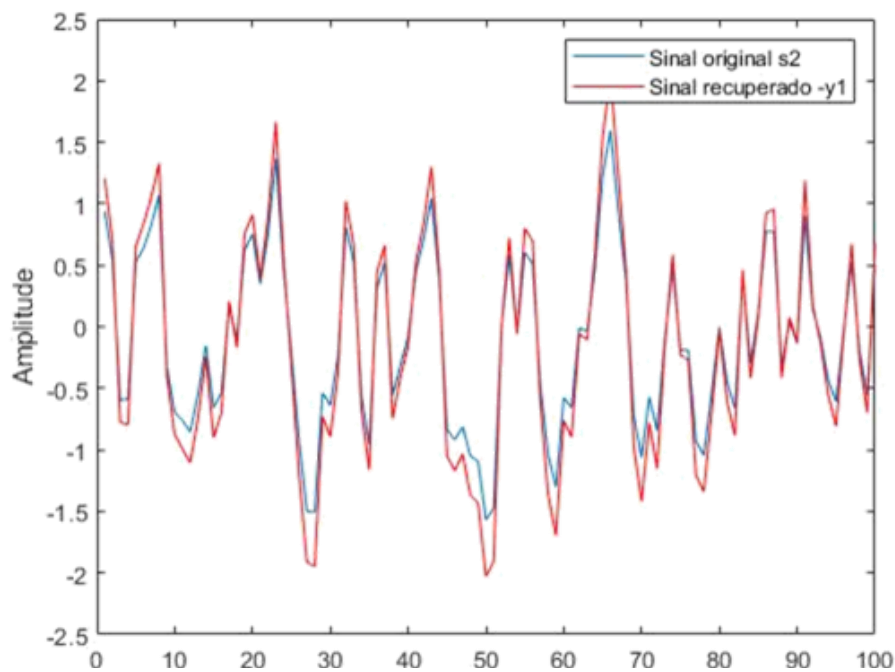


Figura 3 - Fonte uniforme $s_2(t)$ e as respectiva fonte recuperada $-y_1(t)$

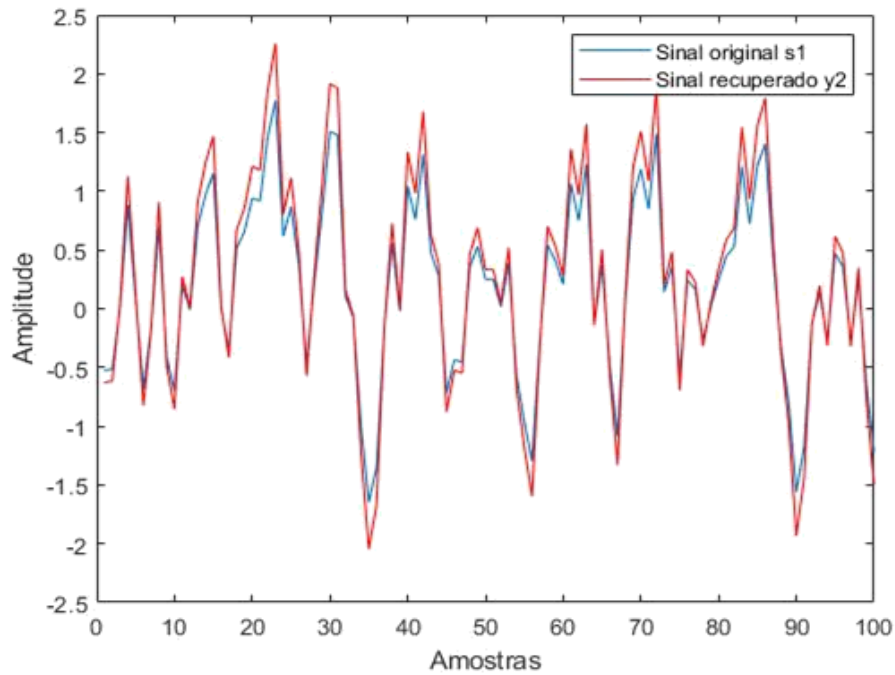


Figura 4 - Fonte uniforme $s_1(t)$ e as respectiva fonte recuperada $y_2(t)$

3.3 BRANQUEAMENTO APLICADO A ICA

A técnica conhecida como “branqueamento” é amplamente utilizada como uma etapa de pré-processamento visando simplificar o processo de recuperação das fontes independentes através das técnicas de ICA [1].

Dizemos que um vetor aleatório com média nula é “branco” se os seus elementos são decorrelacionados entre si e possuem variâncias unitárias. Essa característica pode ser observada através da matriz de autocovariância do vetor (ou a matriz de autocorrelação, dado que a média é nula) que, nesse caso, será igual a matriz identidade [1].

Por exemplo, seja $\mathbf{b} = [b_1, \dots, b_n]^T$ um vetor aleatório com média $m_{\mathbf{b}}$ nula, então a matriz de autocovariância de \mathbf{b} é dada por:

$$\mathbf{C}_{\mathbf{b}} = E\{(\mathbf{b} - m_{\mathbf{b}}) \cdot (\mathbf{b} - m_{\mathbf{b}})^T\} = E\{\mathbf{b} \cdot \mathbf{b}^T\} = \begin{bmatrix} E\{b_{11} \cdot b_{11}\} & \cdots & E\{b_{1n} \cdot b_{1n}\} \\ \vdots & \ddots & \vdots \\ E\{b_{n1} \cdot b_{n1}\} & \cdots & E\{b_{nn} \cdot b_{nn}\} \end{bmatrix}. \quad (3.6)$$

Se \mathbf{b} é um vetor branco, então a covariância e a correlação entre os seus elementos $E\{b_{ij} \cdot b_{ij}\}$ será unitária para todo $i = j$ e nula caso contrário, resultando assim em uma matriz de autocovariância $\mathbf{C}_{\mathbf{b}}$ diagonal:

$$\mathbf{C}_{\mathbf{b}} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} = \mathbf{I}. \quad (3.7)$$

Utilizando a mesma definição dada por A. Hyvärinen, J. Karhunen e E. Oja [1], dado um vetor $\mathbf{x}(t)$ com n elementos, uma das formas de se fazer o “branqueamento” desse vetor é encontrando uma transformação linear \mathbf{V} tal que:

$$\mathbf{V} = \mathbf{D}^{-1/2} \mathbf{E}^T, \quad (3.8)$$

onde \mathbf{E} é a matriz cujas colunas são formadas pelos autovetores (com norma unitária) da matriz \mathbf{C}_x , e \mathbf{D} é a matriz diagonal formada pelos autovalores de \mathbf{C}_x .

Por fim, basta multiplicar a matriz \mathbf{V} encontrada pelo vetor $\mathbf{x}(t)$ que buscamos branquear, chegamos em um novo vetor \mathbf{z} :

$$\mathbf{z}(t) = \mathbf{V} \mathbf{x}(t). \quad (3.9)$$

Para verificar se o vetor resultante de fato é um vetor branco, calculamos:

$$E\{\mathbf{z}(t) \cdot \mathbf{z}(t)^T\} = \mathbf{V} E\{\mathbf{x}(t) \cdot \mathbf{x}(t)^T\} \mathbf{V}^T = \mathbf{D}^{-1/2} \mathbf{E}^T \mathbf{E} \mathbf{D} \mathbf{E}^T \mathbf{E} \mathbf{D}^{-1/2} = \mathbf{I}. \quad (3.10)$$

Portanto, chegamos em um vetor branco \mathbf{z} obtido a partir do vetor \mathbf{x} dado inicialmente.

O efeito do branqueamento na recuperação das fontes pode ser observado graficamente. A Fig. 2 mostra a distribuição conjunta das fontes $s_1(t)$ e $s_2(t)$, ambas com distribuição uniforme. O eixo horizontal corresponde a $s_1(t)$ e o vertical a $s_2(t)$. A Fig. 3 mostra a distribuição conjunta de $x_1(t)$ e $x_2(t)$, obtidos a partir da mistura das fontes independentes $s_1(t)$ e $s_2(t)$ utilizando a matriz de mistura $\mathbf{A} = \begin{bmatrix} 0,9 & 0,7 \\ 0,3 & 0,5 \end{bmatrix}$.

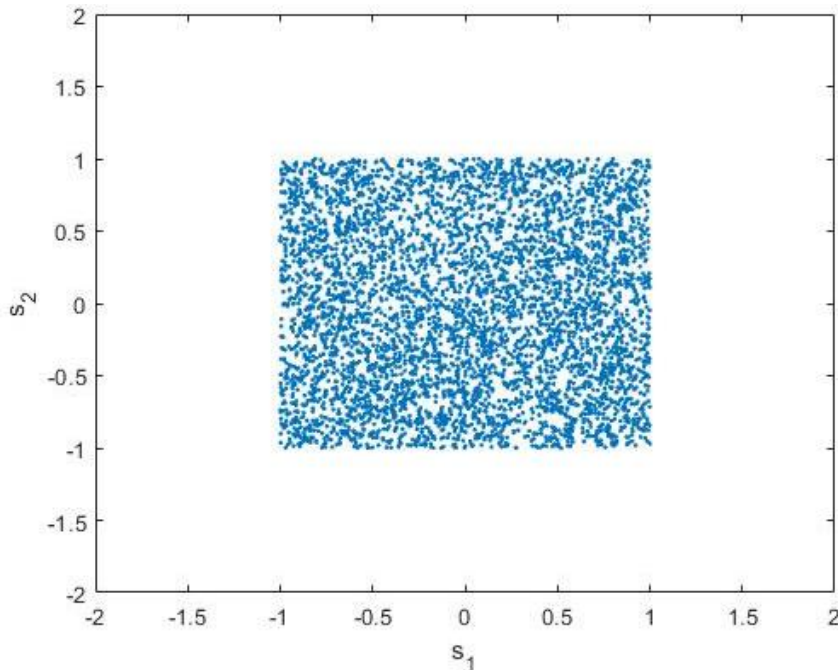


Figura 5 - Distribuição conjunta das fontes $s_1(t)$ e $s_2(t)$ com distribuição uniforme.

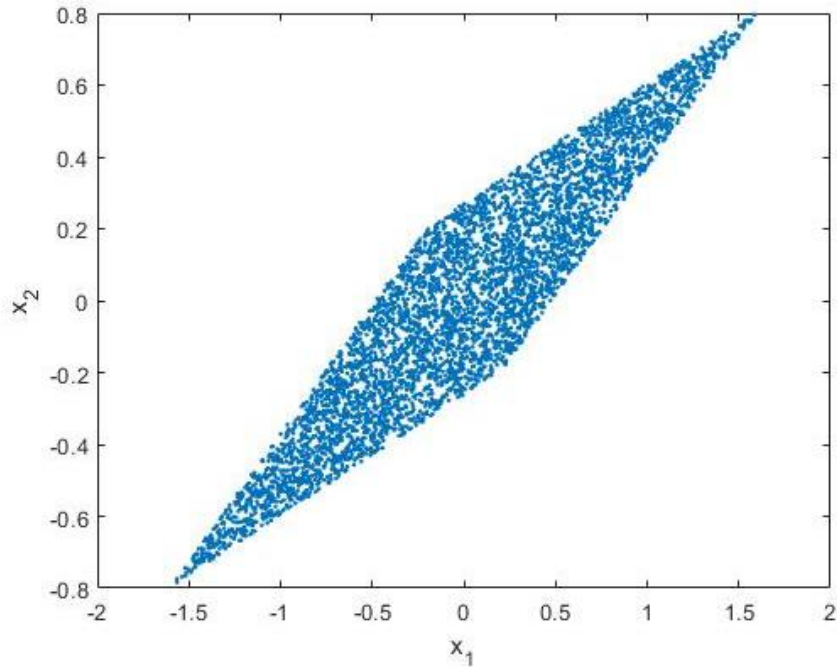


Figura 6 - Distribuição conjunta das misturas $x_1(t)$ e $x_2(t)$

Observando a distribuição conjunta dada na Fig. 3, fica claro que as misturas $x_1(t)$ e $x_2(t)$ obtidas não são mais independentes entre si. É possível observar ainda que as pontas do paralelograma encontrado estão na direção dos vetores dados pelas colunas da matriz de mistura **A**.

Fazendo o branqueamento das misturas $x_1(t)$ e $x_2(t)$, obtendo assim as componentes $z_1(t)$ e $z_2(t)$, encontramos a Fig. 4 abaixo.

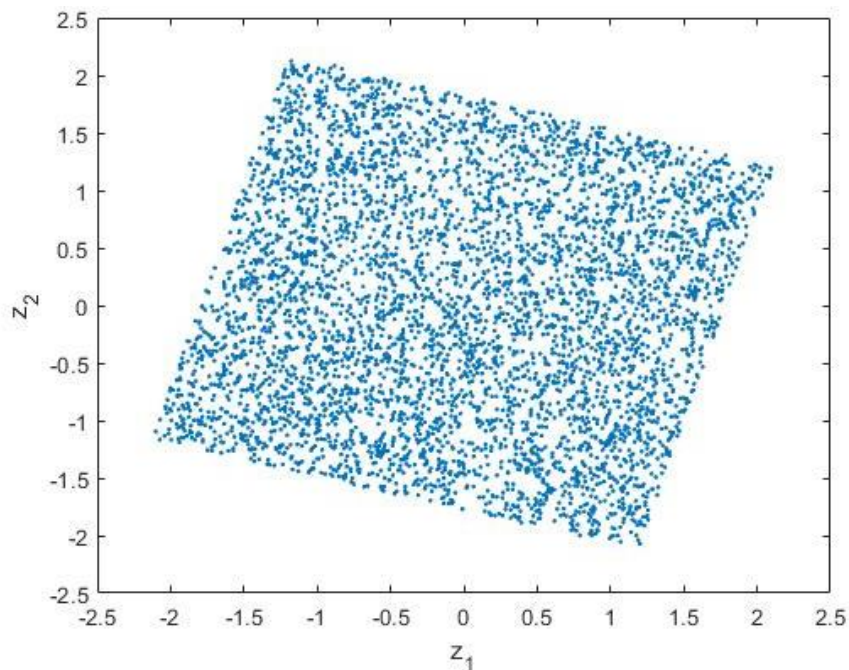


Figura 7 - Distribuição conjunta das misturas já branqueadas $z_1(t)$ e $z_2(t)$.

Após o branqueamento das misturas, a distribuição conjunta das variáveis retorna ao mesmo formato quadrado da distribuição conjunta de $s_1(t)$ e $s_2(t)$, diferindo apenas por uma rotação no plano. Portanto, o branqueamento faz a descorrelação das misturas, restando apenas o problema de se encontrar o ângulo para corrigir essa rotação, o que acaba simplificando o bastante o processo de separação das fontes aplicando a ICA.

4 MÉTODOS QUE EXPLORAM A ESTRUTURA TEMPORAL DOS DADOS

Uma outra classe de métodos utilizada na resolução do problema de separação cega de fontes, são os métodos que exploram a estrutura temporal dos dados. Nesse caso, diferentemente da ICA, ao invés de se basear no cálculo dos momentos de ordem superior das fontes, esses métodos buscam a descorrelação entre as amostras de cada fonte a ser recuperada. Para isso, tais métodos utilizam as correlações entre as amostras de cada fonte buscada para diferentes atrasos no tempo, com o objetivo de zerar essa dependência temporal e, conseqüentemente, recuperar as fontes independentes de origem [1].

A implementação dessa classe de métodos também possui restrições e indeterminações resultantes como visto anteriormente para ICA. Com exceção do critério de não-gaussianidade mencionado no tópico 3.1, as demais premissas utilizadas para a aplicação das técnicas de ICA também são válidas para os algoritmos que exploram a estrutura temporal das fontes. Isso também se aplica para as ambigüidades da ICA descritas no tópico 3.2, ou seja, tanto a ICA quanto os algoritmos mencionados possuem indeterminações na ordem e na amplitude das fontes recuperadas [1].

Além das restrições de ICA citadas, os métodos que exploram a estrutura temporal dos dados também se baseiam na premissa de que as fontes buscadas precisam ter perfis de correlação diferentes entre si. Isso pode ser verificado calculando-se a densidade espectral de potência das fontes $y_i(t)$ buscadas em (2.2), conforme a equação abaixo:

$$S_y = |Y_i(\omega)|^2, \quad (4.1)$$

onde $Y_i(\omega)$ é transformada de Fourier da fonte $y_i(t)$.

4.1 SEPARAÇÃO DE FONTES UTILIZANDO CORRELAÇÃO TEMPORAL

Seguindo o modelo de BSS definido na equação (2.2) e, considerando que as fontes origem possuem média nula, a forma mais simples de explorar a estrutura temporal dessas

fontes é através das correlações entre os pares de misturas $x_i(t)$ e $x_j(t - \tau)$, onde τ é um atraso qualquer no tempo. Sabendo que a correlação $r_{ij}(\tau)$ entre as misturas $x_i(t)$ e $x_j(t - \tau)$ para um atraso τ é dada por:

$$r_{ij}(\tau) = E\{x_i(t)x_j(t - \tau)\}, \quad (4.2)$$

então, calculando as correlações entre todos os pares de mistura $x_i(t)$ e $x_j(t - \tau)$, sendo i e j índices dos elementos do vetor $\mathbf{x}(t)$ em (2.2), temos:

$$\mathbf{R}_\tau^{\mathbf{x}} = E\{\mathbf{x}(t)\mathbf{x}(t - \tau)^T\} = \begin{bmatrix} E\{x_1(t)x_1(t - \tau)\} & \cdots & E\{x_1(t)x_n(t - \tau)\} \\ \vdots & \ddots & \vdots \\ E\{x_n(t)x_1(t - \tau)\} & \cdots & E\{x_n(t)x_n(t - \tau)\} \end{bmatrix}, \quad (4.3)$$

sendo $\mathbf{R}_\tau^{\mathbf{x}}$ a matriz de correlação do vetor $\mathbf{x}(t)$ com atraso τ escolhido.

Em geral, as densidades de probabilidade das fontes não são conhecidas, por isso, ao invés de utilizarmos a equação (3.3) para o cálculo dos momentos $E\{x_i(t)x_j(t - \tau)\}$ para cada par de mistura disponível, utilizamos uma média temporal das fontes contidas no vetor $\mathbf{x}(t)$:

$$E\{\mathbf{x}(t)\} \approx \frac{1}{K} \sum_{i=1}^K x_i(t), \quad (4.4)$$

onde K é o total de amostras contidas no vetor \mathbf{x} .

Dessa forma, aplicando a média temporal para o cálculo da correlação dada em (4.2), temos:

$$r_{ij}(\tau) = E\{x_i(t)x_j(t - \tau)\} \approx \frac{1}{K} \sum_{t=1}^K x_i(t)x_j(t - \tau). \quad (4.5)$$

Supondo que as amostras contidas em $\mathbf{x}(t)$ são correlacionadas entre si, então nem todos os elementos fora da diagonal da matriz $\mathbf{R}_\tau^{\mathbf{x}}$ serão nulos. Como o objetivo das técnicas de ICA é recuperar a independência dos sinais para chegar nas fontes originais, precisamos que a matriz $\mathbf{R}_\tau^{\mathbf{x}}$ seja diagonal, ou seja, a correlação entre qualquer par de misturas $x_i(t)$ e $x_j(t - \tau)$ para $i \neq j$ precisa ser nula.

Considerar apenas um atraso τ pode não ser suficiente para descorrelacionar as amostras entre si e, conseqüentemente, recuperar as fontes independentes desejadas. Dessa forma, podemos generalizar o caso anterior para um conjunto de P atrasos no tempo $\{\tau_i\}_{i=1}^P$, logo teremos um conjunto de P matrizes de correlação de $\mathbf{x}(t)$, o qual pode ser denotado por:

$$\mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_P\}, \quad (4.6)$$

sendo \mathbf{R}_i a i -ésima matriz de correlação de $\mathbf{x}(t)$ para o i -ésimo atraso no tempo τ_i dada por: $\mathbf{R}_i = E\{\mathbf{x}(t)\mathbf{x}(t - \tau_i)^T\}$.

Dessa forma, o objetivo dos métodos de separação de fontes que utilizam correlações temporal é encontrar uma matriz \mathbf{W} , conforme dado na equação (2.2) que, ao ser multiplicada

por cada matriz de correlação \mathbf{R}_i minimize os termos fora das suas diagonais, de forma que as matrizes pertencentes ao conjunto \mathbf{R} sejam diagonalizadas simultaneamente.

Na prática, não é possível fazer essa diagonalização de forma exata, pois os autovetores das matrizes de correlação são diferentes para cada atraso no tempo. Sendo assim, o objetivo será encontrar o maior grau possível de diagonalização conjunta dessas matrizes. Uma das formas que pode ser utilizada para medir o grau de diagonalização de uma matriz \mathbf{M} é dada pelo seguinte operador

$$off(\mathbf{M}) \equiv \sum_{i \neq j} m_{ij}^2, \quad (4.7)$$

que por sua vez calcula a soma dos quadrados dos elementos fora da diagonal da matriz.

Dois métodos para diagonalização conjunta de matrizes foram abordados nesse trabalho: o método do gradiente descendente e o algoritmo de Jacobi. Ambos serão detalhados nos tópicos 4.4 e 4.5, respectivamente.

4.2 AMUSE

O AMUSE (do inglês *Algorithm for Multiple Unknown Signals Extraction*) faz parte da classe dos métodos que exploram a estrutura temporal dos dados para a separação cega de fontes. Com essa finalidade, o algoritmo baseia-se no cálculo da matriz de correlação entre as amostras de misturas considerando um único atraso no tempo τ , como dado em (4.3), resultando assim em uma matriz de separação \mathbf{W} e, conseqüentemente no vetor $\mathbf{y}(t)$ com fontes recuperadas conforme mostrado em (2.2).

O AMUSE é um algoritmo relativamente simples de ser implementado e rápido para ser calculado. Conforme descrito por TONG *et al.* [3], o AMUSE pode ser implementado através dos seguintes passos:

1. Fazer o branqueamento do vetor de mistura $\mathbf{x}(t)$ já centralizado, ou seja, com média nula, obtendo assim o vetor $\mathbf{z}(t)$ conforme dado em (3.9).
2. Calcular os autovetores da matriz $\mathbf{R}_\tau^Z = \frac{1}{2}[\mathbf{R}_\tau + \mathbf{R}_\tau^T]$, onde $\mathbf{R}_\tau = E\{\mathbf{z}(t)\mathbf{z}(t - \tau)^T\}$ é a matriz de correlação com um atraso de tempo τ para o vetor $\mathbf{z}(t)$.
3. As linhas da matriz de separação \mathbf{W} consistem nos autovetores encontrados no passo anterior.

O AMUSE funciona muito bem para os casos em que os autovetores da matriz \mathbf{R}_τ^z são unicamente definidos, ou seja, quando os autovalores são todos distintos entre si. Entretanto, nos casos em que temos algum autovalor repetido, o AMUSE não consegue recuperar todas as fontes independentes buscadas. Uma forma de tentar contornar essa restrição é encontrando um atraso no tempo τ de forma que todos os autovalores sejam distintos entre si. O problema é que isso nem sempre é possível. Por exemplo, nos casos em que os sinais $s_i(t)$ possuem as mesmas autocorrelações, ou seja, a mesma densidade espectral de potência, então não existe um atraso τ que viabilize a estimativa das fontes através desse algoritmo.

4.3 SOBI

O SOBI (do inglês *Second-Order Blind Identification*) também é um algoritmo que explora a estrutura temporal das fontes buscando a separação entre elas, mas diferentemente do AMUSE, o SOBI considera múltiplos atrasos no tempo buscando descorrelacionar as amostras das misturas observadas.

Sabendo que o SOBI explora mais de um atraso no tempo, conseqüentemente teremos um conjunto de P matrizes de correlação para diagonalizar, sendo P o número total de atrasos considerados.

Denotando por $\{\mathbf{R}_i\}_{i=1}^P$ as matrizes de correlação para o conjunto $\{\tau_i\}_{i=1}^P$ de P atrasos no tempo considerados, onde $\mathbf{R}_i = E\{\mathbf{z}(t)\mathbf{z}(t - \tau_i)^T\}$ é a i -ésima matriz de correlação do vetor $\mathbf{z}(t)$ com as misturas após o branqueamento realizado em (3.9), baseando-se na definição dada por M. Joho e H. Mathis [4], dizemos que o SOBI tem como objetivo a minimização da seguinte função custo:

$$\mathcal{J}_1(\mathbf{W}) \equiv \sum_{i=1}^P \beta_i \text{off}(\mathbf{W}\mathbf{R}_i\mathbf{W}^T), \quad (4.8)$$

onde $\{\beta_i\}_{i=1}^P$ é o conjunto dos fatores de ponderação positivos normalizados, de forma que a função custo \mathcal{J}_1 se torne independente das normas absolutas das matrizes de correlação \mathbf{R}_i , conforme descrito na equação abaixo.

$$\sum_{i=1}^P \beta_i \|\mathbf{R}_i\|_F^2 = 1. \quad (4.9)$$

Para evitar que se chegue na resposta trivial onde $\mathbf{W} = 0$, podemos acrescentar um termo de penalização da função custo, tendo como objetivo forçar que a matriz \mathbf{W} se aproxime ao máximo de uma matriz unitária. A equação abaixo mostra uma função custo possível para o termo de penalização:

$$\mathcal{J}_2(\mathbf{W}) \equiv \|\mathbf{W}\mathbf{W}^T - \mathbf{I}\|_F^2. \quad (4.10)$$

Sendo assim, quanto mais distante a matriz \mathbf{W} estiver de ser uma matriz unitária, maior será a penalização dada pela função custo \mathcal{J}_2 . Combinando as funções \mathcal{J}_1 e \mathcal{J}_2 , chegamos em uma função custo que busca encontrar uma matriz unitária \mathbf{W} que diagonaliza simultaneamente o conjunto de matrizes considerado, conforme mostrado na equação abaixo:

$$\mathcal{J}_{\{1,2\}} \equiv \mathcal{J}_1 + \alpha_2 \mathcal{J}_2, \quad (4.11)$$

onde o parâmetro α_2 é a forma de ponderar o termo de penalização \mathcal{J}_2 . Sendo assim, α_2 acaba influenciando na velocidade de convergência na busca pelo mínimo da função objetivo $\mathcal{J}_{\{1,2\}}$, de forma que quanto maior o valor de α_2 , menor será o desvio entre a matriz \mathbf{W} e uma matriz unitária ao longo do processo de adaptação da trajetória da função.

4.4 MÉTODO DO GRADIENTE DESCENDENTE

Seja uma função custo geral denotada por $\mathcal{J}(\mathbf{W})$ e seu respectivo gradiente dado por $\nabla \mathbf{w} \mathcal{J}(\mathbf{W})$, conforme definido por M. Joho e H. Mathis [4], o método de busca do gradiente descendente dessa função pode ser aplicado da seguinte forma:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \Delta \mathbf{W}_k, \quad (4.12)$$

$$\Delta \mathbf{W}_k = -\mu \nabla \mathbf{w} \mathcal{J}(\mathbf{W}_k), \quad (4.13)$$

onde μ é o tamanho do passo utilizado para aumentar ou diminuir a velocidade de convergência até o mínimo da função e $\Delta \mathbf{W}_k$ é o incremento utilizado para atualizar o valor de \mathbf{W}_k .

Aplicando esse método para encontrar o mínimo da função objetivo $\mathcal{J}_{\{1,2\}}$ do SOBI dada em (4.11), temos a seguinte equação:

$$\nabla \mathbf{w} \mathcal{J}_{\{1,2\}} = \nabla \mathbf{w} \mathcal{J}_1 + \alpha_2 \nabla \mathbf{w} \mathcal{J}_2, \quad (4.14)$$

onde os gradientes individuais são dados por:

$$\nabla \mathbf{w} \mathcal{J}_1 = 2 \sum_{i=1}^P \beta_i \text{off}(\mathbf{W}_k \mathbf{R}_i \mathbf{W}_k^T) \mathbf{W}_k \mathbf{R}_i^T + 2 \sum_{i=1}^P \beta_i \text{off}(\mathbf{W}_k \mathbf{R}_i^T \mathbf{W}_k^T) \mathbf{W}_k \mathbf{R}_i, \quad (4.15)$$

$$\nabla \mathbf{w} \mathcal{J}_2 = 4(\mathbf{W}_k \mathbf{W}_k^T - \mathbf{I}) \mathbf{W}_k, \quad (4.16)$$

onde \mathbf{I} é a matriz identidade.

4.5 ALGORITMO DE JACOBI

Denotando por $\{\mathbf{R}_k\}_{k=1}^K$ um conjunto de K matrizes com $\mathbf{R}_k = E\{\mathbf{z}(t)\mathbf{z}(t - \tau_k)^T\}$, onde τ_k para $k = 1, \dots, K$ são os atrasos no tempo escolhidos e o vetor $\mathbf{z}(t)$ é dado na equação (3.9). Então, seguindo a definição dada por J-F Cardoso e A. Souloumiac [5], o algoritmo de Jacobi

busca encontrar os ângulos complexos c e s , para cada par de índices $i \neq j$ dos elementos das matrizes do conjunto $\{\mathbf{R}_k\}_{k=1}^K$ que minimizem a seguinte função custo:

$$O(c, s) \equiv \sum_{k=1}^K \text{off}(\mathcal{P}(i, j, c, s) \mathbf{R}_k \mathcal{P}^H(i, j, c, s)), \quad (4.17)$$

sendo $\mathcal{P}(i, j, c, s)$ a matriz de rotação para os índices (i, j) e os ângulos complexos c e s , e é dada por:

$$\begin{bmatrix} \mathcal{P}_{ii} & \mathcal{P}_{ij} \\ \mathcal{P}_{ji} & \mathcal{P}_{jj} \end{bmatrix} = \begin{bmatrix} c & \bar{s} \\ -s & \bar{c} \end{bmatrix}, \quad (4.18)$$

onde $|c|^2 + |s|^2 = 1$.

Uma das formas possíveis para obtermos os ângulos c e s é através da matriz (ou rotação) de Givens. Para isso, primeiro definimos um par de índices (i, j) e, em seguida calculamos:

$$\mathbf{G} \equiv \text{Real}(\sum_{k=1}^K h^H(\mathbf{R}_k) h(\mathbf{R}_k)), \quad (4.19)$$

onde o termo $h(\mathbf{R})$ para uma matriz \mathbf{R} é definido como:

$$h(\mathbf{A}) \equiv [r_{ii} - r_{jj}, r_{ij} + r_{ji}, i(r_{ji} - r_{ij})]. \quad (4.20)$$

Por fim, aplicamos o teorema que afirma que, dado a restrição $|c|^2 + |s|^2 = 1$, então a função custo em (4.17) será minimizada para:

$$c = \sqrt{(x+r)/2r}, \quad s = (y - iz)/\sqrt{2r(x+r)}, \quad r = \sqrt{x^2 + y^2 + z^2}, \quad (4.21)$$

sendo $[x, y, z]^T$ qualquer autovetor associado ao maior autovalor da matriz \mathbf{G} anteriormente definida. Portanto, ao minimizar a função custo em (4.17) encontramos uma forma de fazer a diagonalização conjunta das matrizes do conjunto $\{\mathbf{R}_k\}_{k=1}^K$ predefinido.

4.6 TDSEP

O TDSEP (do inglês *Temporal Decorrelation source SEPARation*) também tem como objetivo obter o vetor $\mathbf{y}(t)$ das fontes recuperadas, conforme dado em (2.2), considerando múltiplos atrasos no tempo para decorrelacionar as amostras de cada mistura entre si. Com essa finalidade, a função objetivo do TDSEP é dada por:

$$J = \frac{1}{m} \sum_{k=1}^m l(\mathbf{W}, \tau_k), \quad (4.22)$$

onde m é o número total de amostras em cada fonte, \mathbf{W} é a matriz de separação utilizada em (2.2) e $l(\mathbf{W}, \tau_k)$ é a função custo conforme definida por A. Ziehe, e KR. Muller [6]:

$$l(\mathbf{W}, \tau) \equiv \frac{1}{K} \sum_{i \neq j} \sum_{t=1}^K y_i(t) y_j(t - \tau), \quad (4.23)$$

sendo $y_i(t)$ dado pela equação (2.2) e τ um atraso no tempo igual ao atraso τ_k escolhido.

Portanto, da mesma forma que o SOBI, a função custo do TDSEP também precisa ser minimizada em relação a matriz \mathbf{W} . Para isso, podemos utilizar tanto o método do gradiente descendente quanto o de Jacobi, conforme citados no tópico anterior. Neste trabalho, para a realização das simulações do algoritmo TDSEP optamos pela diagonalização simultânea das matrizes através do método de Jacobi com as rotações de Givens. Todos os resultados das simulações realizadas serão apresentados no tópico a seguir.

5 RESULTADOS EXPERIMENTAIS

Utilizamos o MATLAB para fazer as simulações dos três algoritmos estudados: AMUSE, SOBI e TDSEP. As simulações foram realizadas com o objetivo de verificar o desempenho de cada algoritmo tanto na separação de fontes aleatórias com diferentes tipos de distribuições (uniforme, gaussiana e laplaciana) quanto na separação de sinais de áudio. Os resultados obtidos para cada caso são apresentados a seguir.

5.1 SEPARAÇÃO DE FONTES ALEATÓRIAS

No caso em que buscamos a separação de fontes aleatórias, simulamos os três algoritmos considerando duas fontes aleatórias com as seguintes distribuições cada:

- i. Duas fontes uniformes
- ii. Duas fontes gaussianas
- iii. Duas fontes laplacianas
- iv. Uma fonte uniforme e a outra gaussiana

Para cada cenário, o primeiro passo consiste na geração das fontes aleatórias. Para isso, precisamos definir o número n de fontes que queremos utilizar, assim como o número m de amostras que serão geradas para cada fonte. Nas simulações realizadas utilizamos 2 fontes aleatórias contendo 5000 amostras cada. Em seguida, aplicamos os filtros discretos F_1 e F_2 nas fontes $s_1(t)$ e $s_2(t)$, respectivamente, cujas transformadas Z são dadas por:

$$F_1(z) = 1 + 0,7z^{-1} + 0,5z^{-2} + 0,3z^{-3} + 0,1z^{-4} \quad (5.1)$$

$$F_2(z) = 1 + 0,8z^{-1} + 0,4z^{-2} + 0,1z^{-3} \quad (5.2)$$

Cada fonte resultante forma uma linha da matriz $\mathbf{S}_{n \times m}$ mostrada abaixo, sendo $n=2$ (o número total de fontes e $m=5000$ (o número total de amostras em cada fonte):

$$\mathbf{S} = \begin{bmatrix} s_1(1) & \dots & s_1(5000) \\ s_2(1) & \dots & s_2(5000) \end{bmatrix}. \quad (5.3)$$

O passo seguinte consiste na mistura das fontes criadas. Sendo assim, precisamos definir uma matriz de mistura quadrada $\mathbf{A}_{n \times n}$ que, ao ser multiplicada por \mathbf{S} , resultará na matriz $\mathbf{X}_{n \times m}$, onde cada linha corresponde a uma mistura das fontes aleatórias criadas:

$$\begin{bmatrix} x_1(1) & \dots & x_1(5000) \\ x_2(1) & \dots & x_2(5000) \end{bmatrix} = \begin{bmatrix} 0,75 & 1 \\ 0,5 & 0,25 \end{bmatrix} \cdot \begin{bmatrix} s_1(1) & \dots & s_1(5000) \\ s_2(1) & \dots & s_2(5000) \end{bmatrix}. \quad (5.4)$$

Em seguida, centralizamos as misturas obtidas, ou seja, subtraímos a média da matriz \mathbf{X} , conforme descrito na equação (3.4). Logo após seguimos com o processo de branqueamento, onde calculamos os autovalores e autovetores da matriz de covariância de \mathbf{X}' para então calcularmos a matriz \mathbf{V} , conforme é mostrado na equação (3.8). E, por fim, chegamos na matriz $\mathbf{Z}_{n \times m}$ que contém as misturas já branqueadas e é dada pela equação (3.9).

Após esse pré-processamento chegamos na etapa de aplicação do algoritmo escolhido, passando como entrada a matriz \mathbf{Z} contendo as misturas já centralizadas e branqueadas, e recebendo como saída a matriz $\mathbf{Y}_{n \times m}$, conforme é dado na equação (2.2), contendo as fontes recuperadas. Nas simulações realizadas para cada algoritmo, utilizamos todas as amostras disponíveis ($m=5000$) para o cálculo das matrizes de correlação das componentes de \mathbf{Z} .

Como foi visto no tópico 3.2 que descreve as ambiguidades da ICA, as fontes obtidas possuem ambiguidade na ordem na qual elas são recuperadas assim como na magnitude dos sinais encontrados. Para contornar esse problema, após obtermos a matriz \mathbf{Y} , fazemos a correção do sinal e da ordem das fontes recuperadas. Para isso, primeiro calculamos a matriz de correlação entre \mathbf{S} e \mathbf{Y} :

$$\mathbf{R}_{\mathbf{S}\mathbf{Y}} = E\{\mathbf{S} \cdot \mathbf{Y}\} = \begin{bmatrix} E\{s_1(t)y_1(t)\} & E\{s_1(t)y_2(t)\} \\ E\{s_2(t)y_1(t)\} & E\{s_2(t)y_2(t)\} \end{bmatrix}, \quad (5.5)$$

onde $y_1(t)$ e $y_2(t)$ são as fontes recuperadas e $E\{s_i(t)y_j(t)\}$ é calculada através da média temporal das i -ésimas fontes $s_i(t)$ e $y_j(t)$, conforme abaixo:

$$E\{s_i(t)y_j(t)\} = \frac{1}{K} \sum_{t=1}^K s_i(t)y_j(t). \quad (5.6)$$

Encontrando as correlações entre as fontes \mathbf{S} e \mathbf{Y} , corrigimos tanto o sinal das fontes quanto a ordem na qual elas são recuperadas. Por exemplo, supondo que $E\{s_1(t)y_2(t)\}$ é a correlação com o maior valor absoluto encontrado, então a componente $y_2(t)$ recuperada corresponde à fonte $s_1(t)$ e, conseqüentemente, a componente $y_1(t)$ corresponde a fonte $s_2(t)$. Após a correção da ordem, fazemos correção do sinal multiplicando as fontes recuperadas pelo sinal da correlação com a fonte inicial correspondente, ou seja, caso a correlação $E\{s_1(t)y_2(t)\}$ seja negativa, então multiplicamos cada elemento de $y_2(t)$ por (-1) , o mesmo vale para $y_1(t)$ utilizando o sinal de $E\{s_2(t)y_1(t)\}$. Com isso, obtemos as fontes com sinais e ordem corrigidos.

Em cada caso considerado, utilizamos a SIR (do inglês *Signal-to-Interference Ratio*) como forma de mensurar o desempenho obtido pelo algoritmo na recuperação das fontes iniciais. A SIR calculada para uma fonte $y_i(t)$ recuperada é dada por:

$$sir(i) = 10 \cdot \log \left(\frac{\sum_{t=1}^m y_i(t)^2}{\sum_{t=1}^m [s_i(t) - y_i(t)]^2} \right), \quad (5.7)$$

onde m é o número total de amostras e $s_i(t)$ é a fonte original correspondente.

Nas simulações realizadas com o AMUSE, consideramos dois casos, o primeiro utilizando um atraso $\tau = 1$ e o segundo com $\tau = 2$. No caso das simulações com o SOBI e o TDSEP, foram considerados dois atrasos: $\tau_1 = 1$ e $\tau_2 = 2$. No caso do TDSEP utilizamos os ângulos de Jacobi como método de diagonalização conjunta de matrizes e, para o SOBI foi utilizado o método do gradiente descendente com os seguintes parâmetros:

$$\mu = 1, \alpha_3 = 0,5, \beta_1 = \frac{1}{\|\mathbf{R}_{\tau_1}^z\| + \|\mathbf{R}_{\tau_2}^z\|}, \beta_2 = 1 - \beta_1, \quad (5.8)$$

onde $\mathbf{R}_{\tau_1}^z$ e $\mathbf{R}_{\tau_2}^z$ são as matrizes de correlação de \mathbf{Z} com atrasos $\tau_1 = 1$ e $\tau_2 = 2$, respectivamente.

Os desempenhos obtidos por cada algoritmo nos quatro cenários propostos são mostrados nas tabelas (1-4) a seguir. Os valores indicados em cada tabela mostram a SIR média obtida (em dB) para cada fonte em um total de 50 simulações realizadas para cada algoritmo.

	Fonte 1 (Uniforme)	Fonte 2 (Uniforme)
AMUSE ($\tau = 1$)	39.01	37.75
AMUSE ($\tau = 2$)	62.36	58.14
SOBI	59.19	56.70
TDSEP	60.77	57.30

Tabela 1 - SIR média (dB) para duas fontes com distribuições uniformes

	Fonte 1 (Gaussiana)	Fonte 2 (Gaussiana)
AMUSE ($\tau = 1$)	37.32	37.72
AMUSE ($\tau = 2$)	63.74	60.88
SOBI	61.48	58.70
TDSEP	62.89	59.88

Tabela 2 - SIR média (dB) para duas fontes com distribuições gaussianas

	Fonte 1 (Laplaciana)	Fonte 2 (Laplaciana)
AMUSE ($\tau = 1$)	38.58	36.44
AMUSE ($\tau = 2$)	62.89	59.33
SOBI	59.65	57.53
TDSEP	61.55	58.19

Tabela 3 - SIR média (dB) para duas fontes com distribuições laplacianas

	Fonte 1 (Uniforme)	Fonte 2 (Gaussiana)
AMUSE ($\tau = 1$)	36.05	34.58
AMUSE ($\tau = 2$)	64.90	61.88
SOBI	61.71	59.34
TDSEP	63.69	60.84

Tabela 4 - SIR média (dB) para uma fonte com distribuição uniforme e a outra gaussiana

Conforme mostrado nas Tabelas (1-4) acima, a SIR média encontrada para cada fonte pelos três algoritmos foram muito próximas dentro dos cenários propostos. O pior desempenho foi para o AMUSE considerando o atraso $\tau = 1$. Contudo, alterando o atraso para $\tau = 2$, chegamos em quase o dobro do valor da SIR média para o AMUSE, se aproximando assim dos valores obtidos para a SIR média do SOBI e do TDSEP.

As Figuras (5-12) a seguir mostram os gráficos obtidos para o cenário descrito na Tabela 2, onde temos duas fontes com distribuições gaussianas. Cada figura mostra as fontes $y_i(t)$ recuperadas (em azul) e as fontes originais $s_i(t)$ (em vermelho). Para simplificar a visualização dos gráficos, foram plotadas apenas as 100 primeiras amostras de cada fonte. Conforme pode ser observado nas figuras abaixo, com exceção do AMUSE com atraso $\tau = 1$, todas as fontes recuperadas se sobrepõem quase completamente em relação às fontes originais, mostrando o ótimo desempenho dos algoritmos.

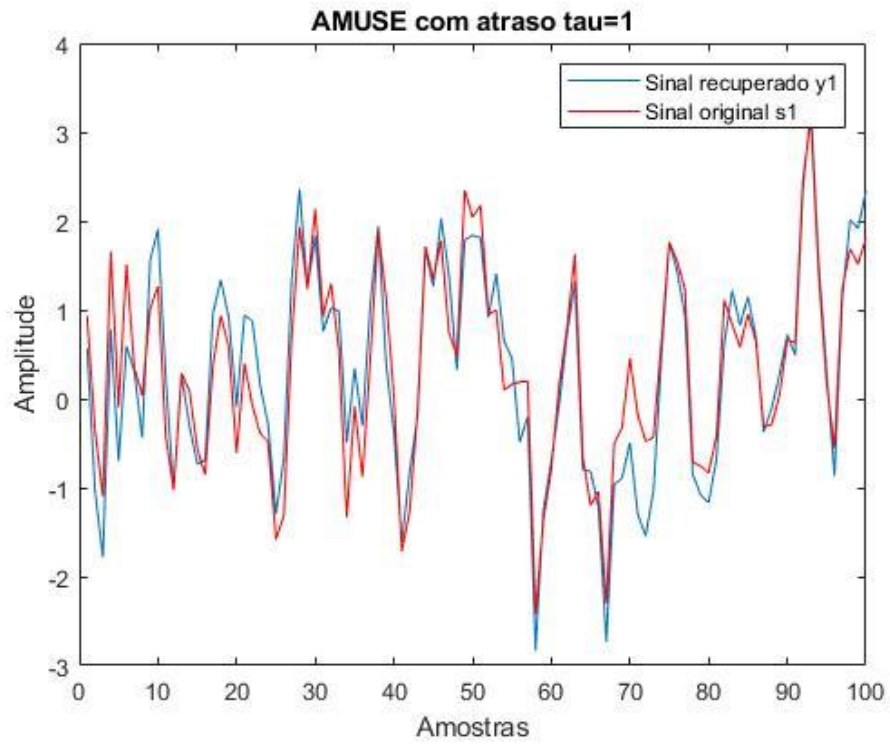


Figura 1 - Fonte gaussiana $s_1(t)$ e a respectiva fonte $y_1(t)$ recuperada com AMUSE ($\tau = 1$)

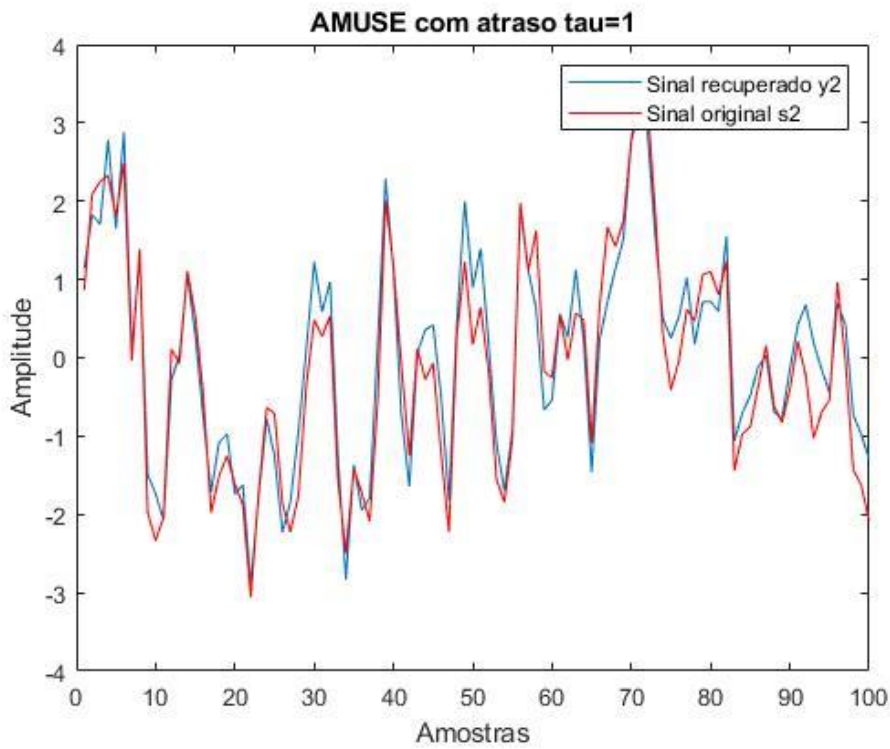


Figura 2 - Fonte gaussiana $s_2(t)$ e a respectiva fonte $y(t)$ recuperada com AMUSE ($\tau = 1$)

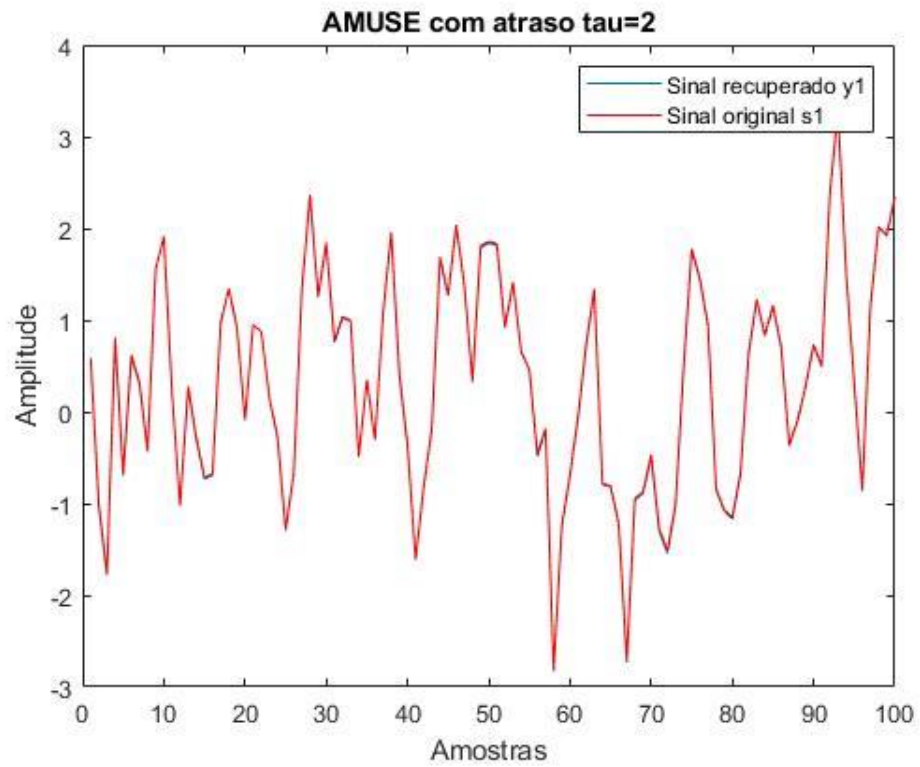


Figura 3 - Fonte gaussiana $s_1(t)$ e a respectiva fonte $y_1(t)$ recuperada com AMUSE ($\tau = 2$)

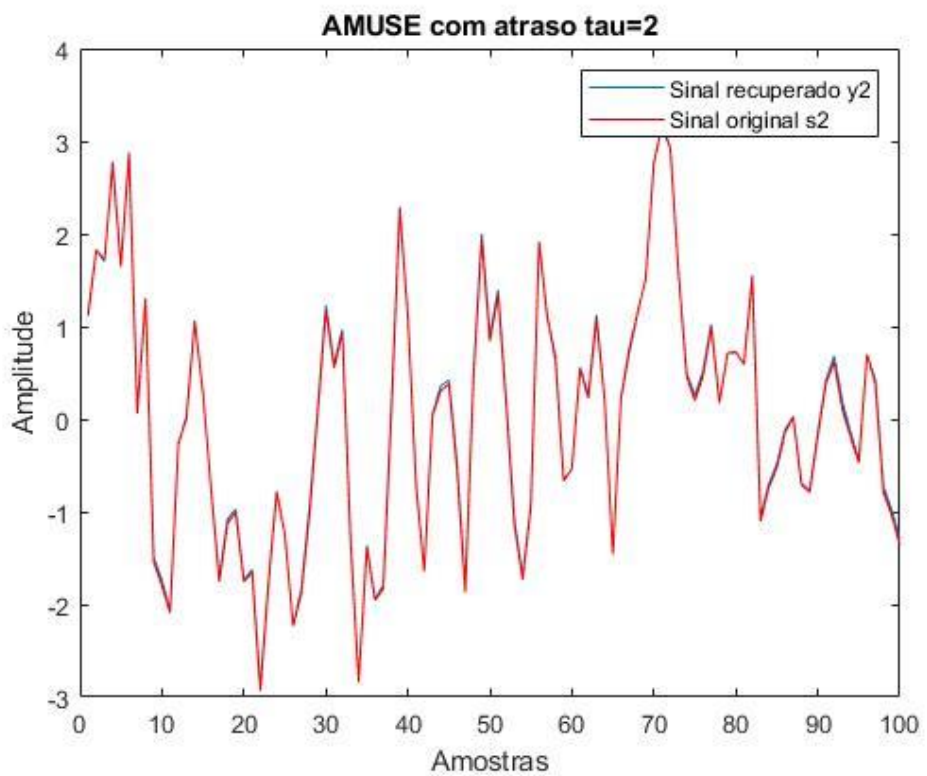


Figura 4 - Fonte gaussiana $s_2(t)$ e a respectiva fonte $y(t)$ recuperada com AMUSE ($\tau = 2$)

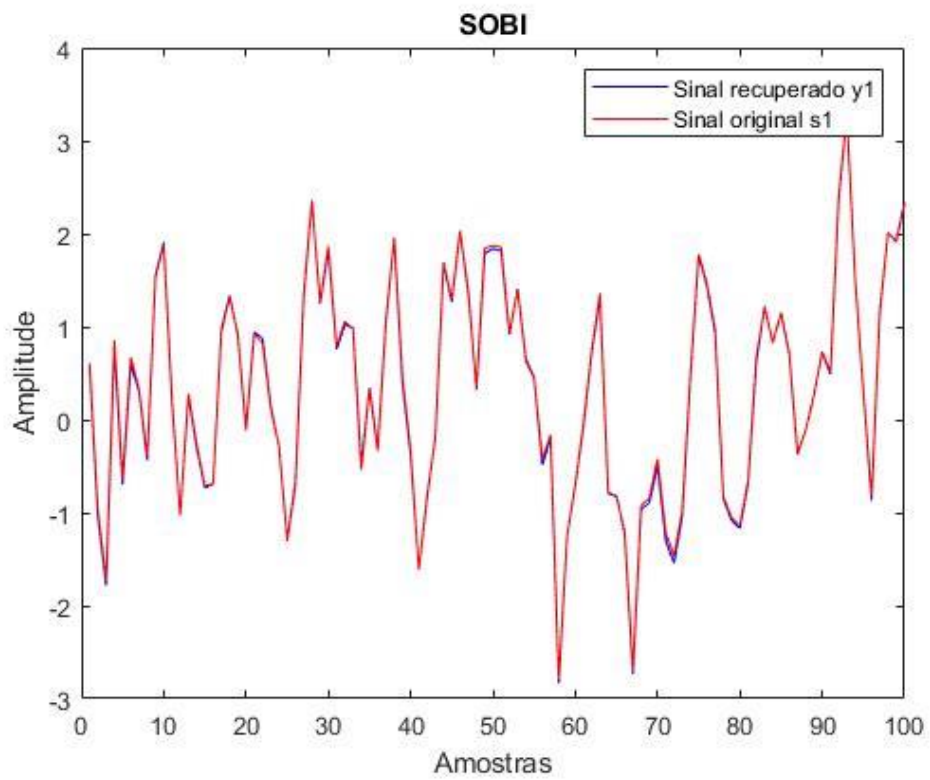


Figura 5 - Fonte gaussiana $s_1(t)$ e a respectiva fonte $y_1(t)$ recuperada com SOBI

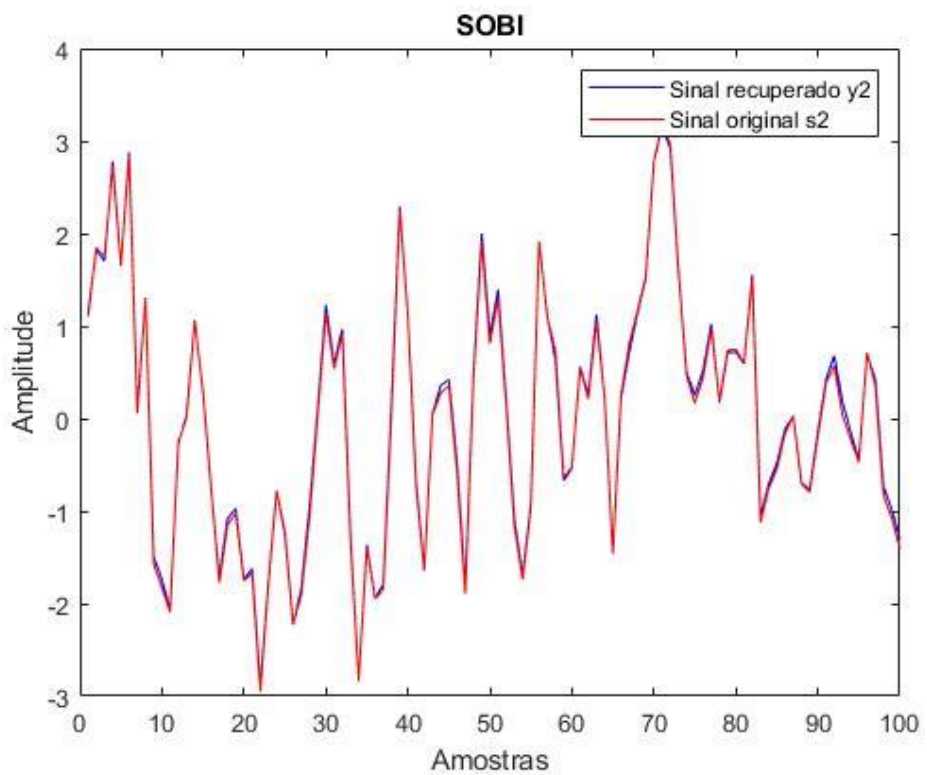


Figura 6 - Fonte gaussiana $s_2(t)$ e a respectiva fonte $y_2(t)$ recuperada com SOBI

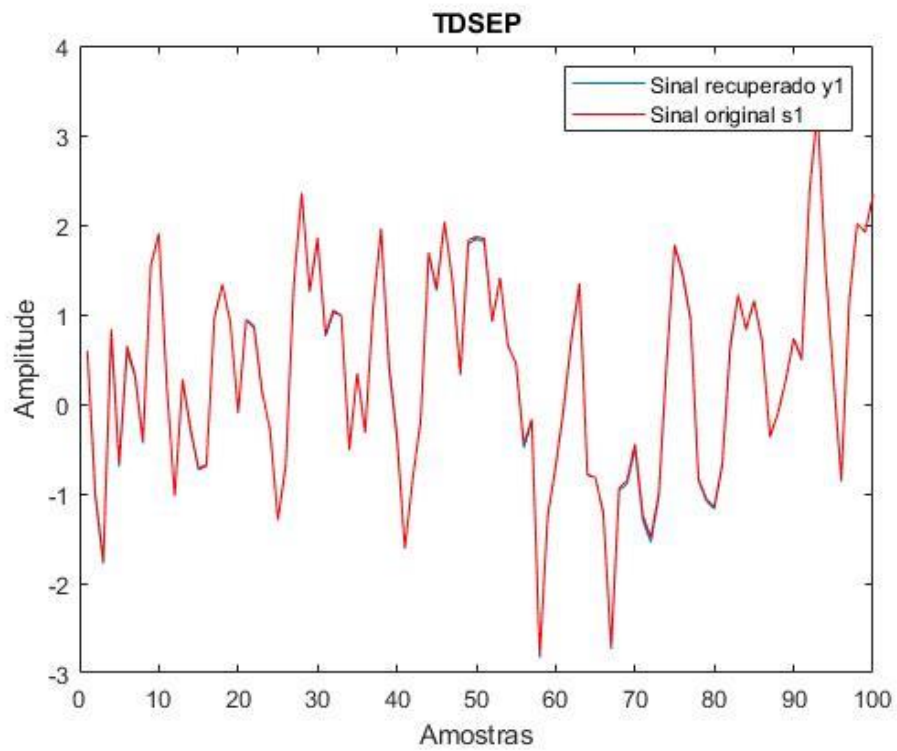


Figura 7 - Fonte gaussiana $s_1(t)$ e a respectiva fonte $y_1(t)$ recuperada com TDSEP

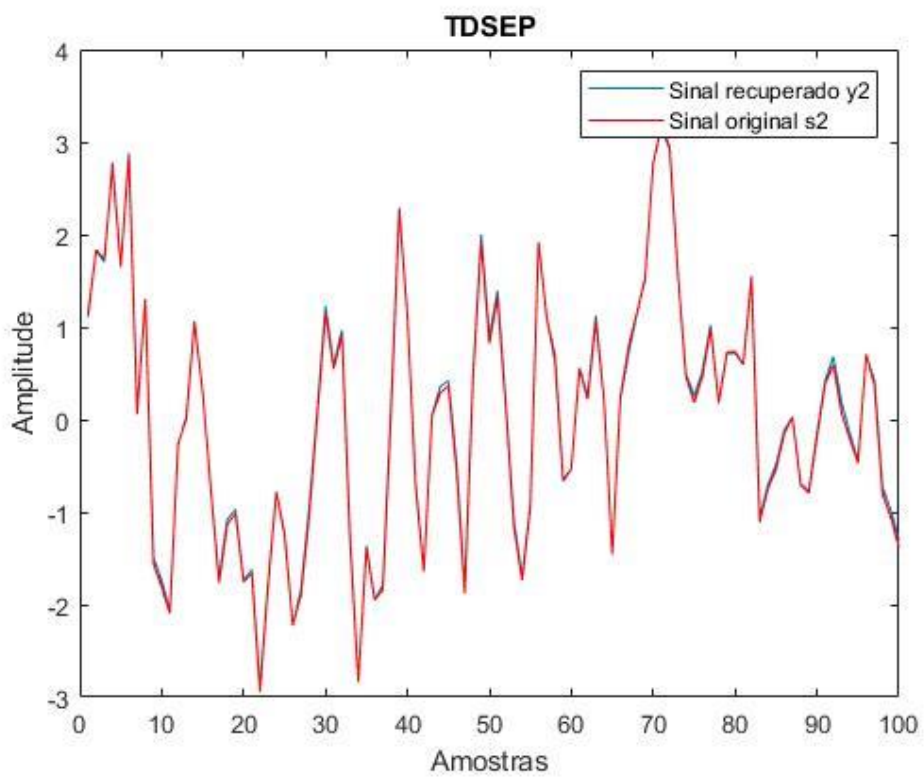


Figura 8 - Fonte gaussiana $s_2(t)$ e a respectiva fonte $y_2(t)$ recuperada com TDSEP

5.2 SEPARAÇÃO DE SINAIS DE ÁUDIO

Para a simulação do caso em que as fontes são sinais de áudios, utilizamos dois arquivos (.wav) contendo o mesmo número de amostras cada ($m = 1.213.281$). Após a leitura e ajuste do tamanho dos áudios utilizados, precisamos fazer a centralização das fontes obtidas, o que não era necessário no caso das fontes aleatórias dado que elas já eram geradas com média nula.

A próxima etapa consiste na mistura dos sinais de áudio obtidos. Na simulação realizada utilizamos a mesma matriz de mistura considerada no caso anterior. Após, seguimos com a centralização e branqueamento das misturas da mesma forma que foi realizada para as fontes aleatórias.

Os resultados obtidos com aplicação do AMUSE, SOBI e TDSEP para a separação das fontes de áudio consideradas são mostrados na Tabela 5 abaixo. Todos os parâmetros utilizados nas simulações anteriores foram mantidos para o cenário atual. Contudo, no caso em que temos dois sinais de áudio não é necessário realizar várias simulações buscando encontrar a SIR média de um algoritmo. Sendo assim, cada simulação foi realizada apenas uma vez para cada algoritmo resultando nos valores da SIR (em dB) mostrados na tabela abaixo.

	Música 1	Música 2
AMUSE ($\tau = 1$)	164.52	177.18
AMUSE ($\tau = 2$)	164.94	176.41
SOBI	164.81	176.65
TDSEP	164.92	176.46

Tabela 5 – SIR (dB) para sinais de áudio recuperados

Conforme pode ser observado na tabela acima, todos os algoritmos tiveram um desempenho excelente, podendo-se dizer que as fontes foram recuperadas completamente, praticamente sem distorção. Todos os algoritmos considerados tiveram um desempenho muito superior para a separação dos sinais de áudio quando comparamos com os cenários mostrados anteriormente para as fontes aleatórias. Isso pode ser justificado considerando que, no caso das músicas utilizadas, o número de amostras contidas em cada fonte é muito superior ao das fontes geradas artificialmente. Além disso, os sinais de áudio utilizados como fonte possuem uma maior correlação temporal entre as amostras quando comparados com as fontes geradas artificialmente, cujas correlações foram acrescentadas através dos filtros dados em (5.1) e (5.2).

As figuras 13 e 14 abaixo mostram os sinais de áudio utilizados como fontes através dos gráficos de $s_1(t)$ e $s_2(t)$ no domínio do tempo com frequências de amostragem $f_s = 44,1 \text{ MHz}$.

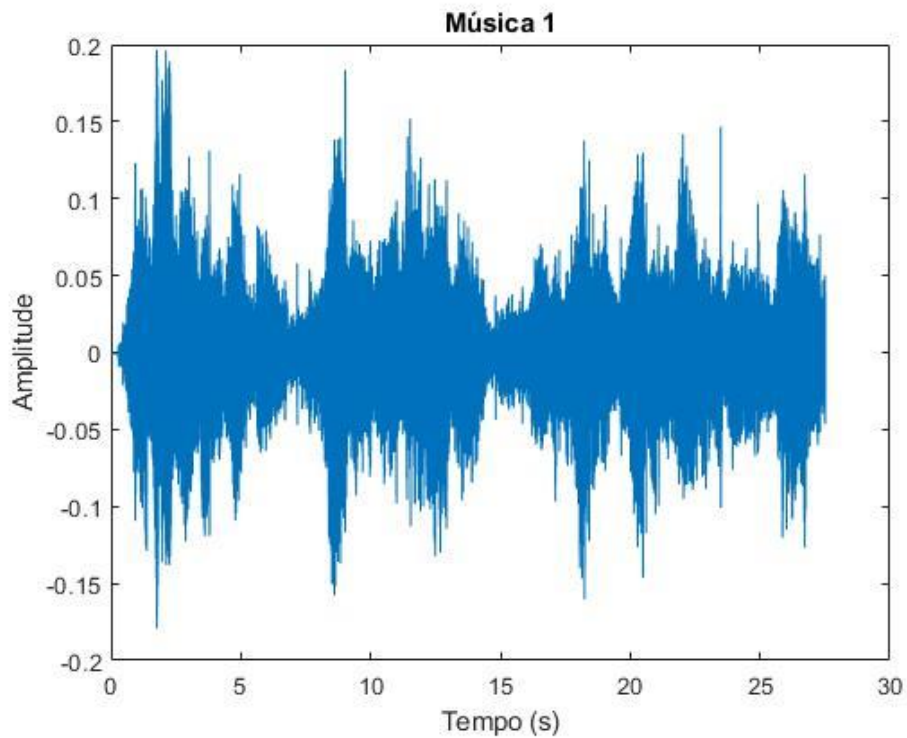


Figura 9 – Sinal de origem $s_1(t)$ no domínio do tempo com $f_s = 44,1 \text{ MHz}$.

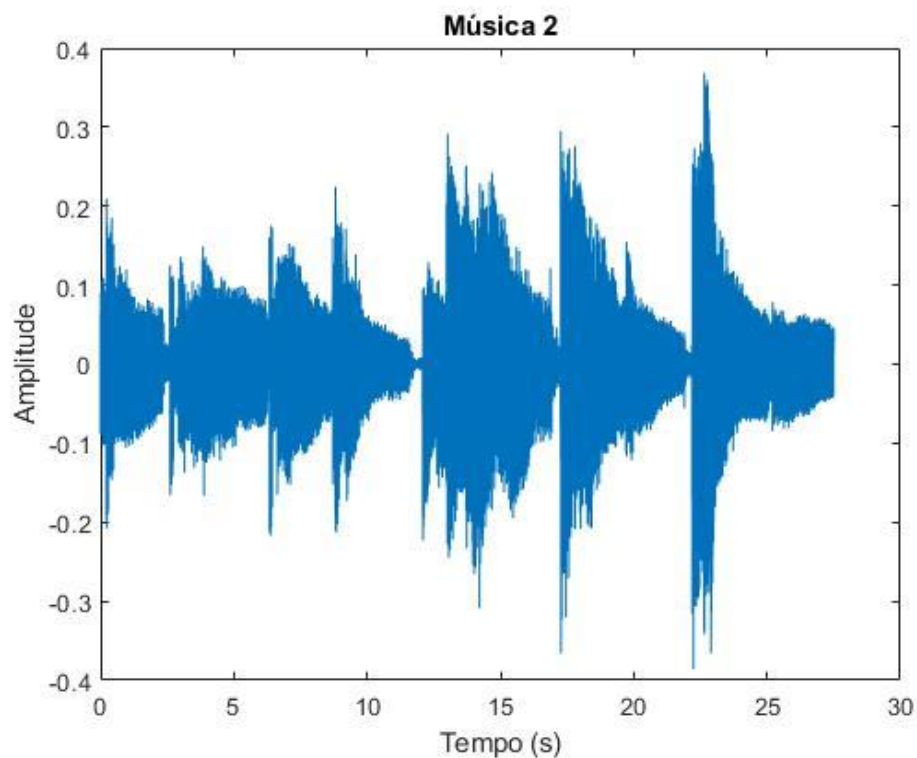


Figura 10 - Sinal de origem $s_2(t)$ no domínio do tempo com $f_s = 44,1 \text{ MHz}$.

6 CONCLUSÃO

Este trabalho teve como objetivo estudar os métodos de ICA que exploram a estrutura temporal dos dados no contexto de separação cega de fontes.

Dentro da classe de métodos estudados, os algoritmos implementados foram o AMUSE, o SOBI e o TDSEP. Cada algoritmo foi simulado considerando a separação tanto de fontes artificiais, quanto de sinais de áudio. Em ambos os casos, suas características e desempenhos foram comparados, conforme foi mostrado através dos resultados experimentais obtidos.

No caso do AMUSE, foi possível observar que a escolha adequada do valor do atraso τ é muito importante para o seu bom desempenho, ao contrário do que foi visto para os algoritmos que utilizam mais de um atraso no tempo, como o SOBI e o TDSEP.

Em relação ao SOBI e o TDSEP, mesmo sendo utilizados métodos diferentes para a diagonalização conjunta das matrizes de correlação, os seus desempenhos foram igualmente satisfatórios em todos os cenários considerados.

Nos casos em que foram utilizadas fontes artificiais, foi possível observar que as diferentes distribuições de probabilidade consideradas não refletiram em diferenças significativas nos desempenhos dos algoritmos. De forma geral, podemos dizer que os três algoritmos implementados tiveram desempenhos excelentes, principalmente quando consideramos as suas aplicações no caso de separação de sinais de áudio, onde foi possível recuperar as fontes praticamente sem nenhuma distorção em relação aos sinais de origem.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- [1] HYVÄRINEN, A.; KARHUNEN, J.; OJA, E. Independent Component Analysis. Nova York: John Wiley & Sons INC, 2001.
- [2] HÉRAULT, J., JUTTEN, C., ANS, B. Détection de grandeurs primitives dans un message composite par une architecture de calcul neuromimétique en apprentissage non supervisé, Actes du Xième colloque. GRETSI, maio de 1985, vol. 2, p. 1017-1022.
- [3] TONG, L.; LIU, R. -; SOON, V. C.; HUANG, Y. -. Indeterminacy and identifiability of blind identification. IEEE Transactions on Circuits and Systems, maio de 1991, vol. 38, no. 5, p. 499-509.
- [4] JOHO, M.; MATHIS, H. Joint diagonalization of correlation matrices by using gradient methods with application to blind signal separation. In: Sensor Array and Multichannel Signal Processing Workshop Proceedings, 2002, Rosslyn (USA), IEEE, p. 273-277.
- [5] CARDOSO, J. -; SOULOUMIAC, A. Jacobi angles for simultaneous diagonalization. In: SIAM Journal on Matrix Analysis and Applications, janeiro de 1996, vol. 17, no.1, p. 161-164.
- [6] ZIEHE, A.; MULLER, KR. TDSEP — an efficient algorithm for blind separation using time structure. In: Niklasson L., Bodén M., Ziemke T. (eds) ICANN 98, Perspectives in Neural Computing, 1998, Springer, Londres, p. 675-680.