

UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MÉTODOS PARA MELHORIA DA QUALIDADE DE  
SEPARAÇÃO CEGA DE FONTES SONORAS EM  
ÂNGULOS OBLÍQUOS DE RADIAÇÃO

RICARDO KEHRLE MIRANDA

ORIENTADOR: RICARDO ZELENOVSKY  
COORIENTADOR: JOÃO PAULO CARVALHO LUSTOSA DA COSTA

DISSERTAÇÃO DE MESTRADO EM  
ENGENHARIA DE SISTEMAS ELETRÔNICOS E AUTOMAÇÃO

PUBLICAÇÃO: PPGEA.DM-535/2013

BRASÍLIA/DF: JULHO - 2013.



UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

MÉTODOS PARA MELHORIA DA QUALIDADE DE  
SEPARAÇÃO CEGA DE FONTES SONORAS EM  
ÂNGULOS OBLÍQUOS DE RADIAÇÃO

RICARDO KEHRLE MIRANDA

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO  
DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA  
DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM SIS-  
TEMAS ELETRÔNICOS E AUTOMAÇÃO.

APROVADA POR:

---

Prof. Ricardo Zelenovsky,  
Doutor - PUC-RJ, UnB/ENE  
(Orientador)

---

Prof. Alexandre Ricardo Soares Romariz,  
Ph.D. - University of Colorado at Boulder, ENE/UnB  
(Examinador Interno)

---

Prof. Bruno Luigi Macchiavello Espinoza,  
Doutor - Universidade de Brasília (UnB), CIC/UnB  
(Examinador Externo)

BRASÍLIA/DF, 18 DE JULHO DE 2013.

## **FICHA CATALOGRÁFICA**

KEHRLE MIRANDA, RICARDO

Métodos Para Melhoria da Qualidade de Separação Cega de  
Fontes Sonoras em Ângulos Oblíquos de Radiação.  
[Distrito Federal] 2013.

xiv, 71p., 297 mm (ENE/FT/UnB, Mestre, Sistemas Eletrônicos e Automação,  
, 2013). Dissertação de Mestrado - Universidade de Brasília.  
Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

- |                                 |                                |
|---------------------------------|--------------------------------|
| 1. Separação Cega de Fontes     | 2. Processamento de Sinais     |
| 3. Processamento de Áudio e Voz | 4. Separação de Cantos de Aves |
| I. ENE/FT/UnB                   | II. Título (série)             |

## **REFERÊNCIA BIBLIOGRÁFICA**

KEHRLE MIRANDA, R. (2013). Métodos Para Melhoria da Qualidade de Separação Cega de Fontes Sonoras em Ângulos Oblíquos de Radiação. Dissertação de Mestrado em Sistemas Eletrônicos e Automação, Publicação PPGEA.DM-535/2013, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 71p.

## **CESSÃO DE DIREITOS**

NOME DO AUTOR: Ricardo Kehrle Miranda.

TÍTULO DA DISSERTAÇÃO DE MESTRADO: Métodos Para Melhoria da Qualidade de Separação Cega de Fontes Sonoras em Ângulos Oblíquos de Radiação.

GRAU / ANO: Mestre / 2013

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação de mestrado pode ser reproduzida sem a autorização por escrito do autor.

---

Ricardo Kehrle Miranda  
SHIS QI29 conjunto 12 casa 17  
71675-320 Brasília - DF - Brasil.

# DEDICATÓRIA

res(*in memoriam*).

À minha família e amigos.  
Ao meu tio e padrinho Ivo Tava-

## AGRADECIMENTOS

Agradeço ao meu professor orientador Ricardo Zelenovsky e ao meu coorientador professor João Paulo Lustosa da Universidade de Brasília pelo incentivo e suporte ao longo de todo período acadêmico. Agradeço ao professor Walter Kellermann pela gentileza de me receber na Universidade de Erlangen-Nuremberg e me permitir trabalhar com sua equipe em especial Soyuj Sahoo pela sua Tutoria. Agradeço também ao Centro de Pesquisa em Arquitetura da Informação (CPAI) e ao seu diretor, o professor Mamede Lima-Marques, por seu incentivo e apoio à minha pesquisa. Por fim, agradeço a todos os professores e profissionais da UnB e da Universidade de Erlangen-Nuremberg que contribuíram com a minha formação.

## RESUMO

### MÉTODOS PARA MELHORIA DA QUALIDADE DE SEPARAÇÃO CEGA DE FONTES SONORAS EM ÂNGULOS OBLÍQUOS DE RADIAÇÃO

**Autor:** Ricardo Kehrle Miranda

**Orientador:** Ricardo Zelenovsky

**Programa de Pós-graduação em Sistemas Eletrônicos e Automação.**

**Brasília, julho de 2013**

Os recentes algoritmos de Separação Cega de Fontes, do inglês *Blind Source Separation* (BSS), têm mostrado ser efetivos ao gerar em suas saídas Relações Sinal Interferência, do inglês *Signal to Interference Ratio* (SIR), consideravelmente altas. Porém, este tipo de algoritmo possui alguns desafios. A SIR decai claramente para ângulos de chegada oblíquos a um arranjo de microfones ou quando as fontes estão próximas umas das outras. Este trabalho tenta descobrir o que acontece nestes casos críticos, bem como busca por soluções. As tentativas para a melhoria da qualidade da separação é feita por meio da utilização de informações sobre o posicionamento das fontes do ambiente acústico, mais especificamente, a direção de chegada dos sinais sonoros, e por uma etapa inicial de decorrelação dos sinais de entrada. Além dos problema do decaimento da qualidade da separação em posições não favoráveis para sinais de voz, também a separação de sons silvestres é uma área ainda pouco estudada. Este trabalho também se utiliza de uma ferramenta avançada de separação de sinais sonoros, o TRINICON, para efetuar a separação de cantos de aves.

## **ABSTRACT**

### **METHODS FOR IMPROVEMNT OF BLIND SOUND SOURCE SEPARATION IN OBLIQUE IMPINGING ANGLES**

**Author: Ricardo Kehrle Miranda**

**Supervisor: Ricardo Zelenovsky**

**Programa de Pós-graduação em Sistemas e Eletrônicos e Automação**

**Brasília, July of 2013**

The earliest Blind Source Separation (BSS) algorithms have shown to be effective by generating at their outputs considerably high Signal to Interference Ratio (SIR). Though, this type of algorithm has some challenges. The SIR clearly decreases for too low or too high angles of arrival of the impinging waves on a sensor array or when the sources are too close together. This work tries to figure out what occurs in these critical cases and to find solutions as well. The attempts to increase the quality of the separation is made by the insertion of information about the position of the sources, more specifically, the direction of arrival of the source signals, and by a initial step of decorrelation of the input signals. Moreover, besides the problem of the decrease of separation quality at non favorable positions of the sources, separation of wild sounds was also not yet given much attention. This work also uses an advanced technique of sound separation, the TRINICON, in order to perform the separation of bird songs.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	MOTIVAÇÃO . . . . .	1
1.2	TÍPICO CENÁRIO E SEUS CASOS CRÍTICOS . . . . .	2
1.3	ORGANIZAÇÃO DO TRABALHO . . . . .	4
<b>2</b>	<b>MODELOS, MEDIDAS E MÉTODOS</b>	<b>5</b>
2.1	ANÁLISE DE COMPONENTES INDEPENDENTES . . . . .	7
2.2	MODELO DE DADOS PARA MISTURAS CONVOLUTIVAS . . . . .	9
2.3	MEDIDAS DE DESEMPENHO . . . . .	11
2.3.1	Norma do Erro do Sistema . . . . .	12
2.3.2	Relação Sinal Interferência . . . . .	12
2.3.3	Relação Sinal Interferência com Ruído . . . . .	13
2.4	MÉTODOS MATEMÁTICOS . . . . .	14
2.4.1	Método das Imagens . . . . .	14
2.4.2	Método do Gradiente . . . . .	15
2.4.3	Método do Gradiente Natural . . . . .	17
2.4.4	Funções de Densidade de Probabilidade de Transformações . . . . .	17
<b>3</b>	<b>SEPARAÇÃO CEGA DE FONTES</b>	<b>20</b>
3.1	SEPARAÇÃO CEGA NO DOMÍNIO DA FREQUÊNCIA . . . . .	20
3.1.1	Diferença Entrópica . . . . .	21
3.1.2	Gradiente da Diferença Entrópica . . . . .	23
3.1.3	O Princípio da Distorção Mínima (MDP) . . . . .	26
3.1.4	O Método da Direção de Chegada . . . . .	28
3.1.5	O Método das Correlações Vizinhas . . . . .	29
3.1.6	Utilizando Informações Prévias de Direção de Chegada . . . . .	31
3.1.7	Utilizando Filtros de Decorrelação . . . . .	34
3.2	SEPARAÇÃO UTILIZANDO O TRINICON . . . . .	35
<b>4</b>	<b>EXPERIMENTOS E RESULTADOS</b>	<b>39</b>
4.1	SEPARAÇÃO DE SINAIS DE VOZ EM ÂNGULOS VARIÁVEIS . . . . .	39

4.2 SEPARAÇÃO DE CANTOS DE AVES . . . . .	43
<b>5 CONCLUSÃO</b>	<b>49</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>51</b>
<b>APÊNDICES</b>	<b>54</b>
APÊNDICE A . . . . .	55
APÊNDICE B . . . . .	66

## LISTA DE TABELAS

4.1	Parâmetros Obtidos . . . . .	40
-----	------------------------------	----

## LISTA DE FIGURAS

1.1	Típico cenário e seus casos críticos. . . . .	2
1.2	Diagrama de blocos do método proposto. As variáveis $x_1, \dots, x_n$ representam as misturas na entrada do sistema e $y_1, \dots, y_N$ representam os sinais separados. . . . .	3
2.1	Arranjo linear de microfones. . . . .	5
2.2	Frente de onda planar incidindo sobre ao arranjo. . . . .	6
2.3	Diagrama de mistura e separação instantânea. . . . .	8
2.4	Sistema de mistura e separação de sinais. . . . .	10
2.5	Contribuição da fonte 1 na saída 2 para um cenário $2 \times 2$ . . . . .	13
2.6	(a) Ilustração de uma sala com um arranjo e uma fonte sonora. (b) Ilustração de um arranjo e de imagens das fontes sonoras. . . . .	15
2.7	Máximos e mínimos de uma função mostrando projeções de curvas de contorno e seu gradiente. . . . .	16
3.1	Representação das PDFs de distribuição de Laplace, distribuição de Gauss e de Gram-Charlier. As PDFs foram desenhadas considerando a média igual a zero e variância igual a um. . . . .	23
3.2	Representação da transformada discreta de Fourier (DFT) de tamanho $K$ para 2 fontes e 2 microfones. (a) Antes da separação as duas fontes estão presentes em ambos os microfones. (b) Frequências desalinhadas após separação. (c) Frequências alinhadas após correção da permutação. . . . .	26
3.3	Ilustração de um ambiente acústico com um locutor e um ouvinte. O sinal é filtrado entre o locutor e o ouvinte. . . . .	27
3.4	Resposta do sistema para duas fontes, uma em $-60^\circ$ e outra em $0^\circ$ . O sistema de separação tenta a formar nulos na direção dos sinais interferentes. . . . .	29
3.5	Evolução de $ Y_k(f, m) $ ao longo do tempo. As duas primeiras linhas representam a evolução de $ Y_k(f, m) $ ao longo dos blocos no tempo $m$ de frequências $f$ próximas. A última linha representa a evolução de $ Y_k(f, m) $ ao longo de $m$ em uma frequência $f$ distante das duas anteriores. . . . .	30

3.6	Diagrama de blocos do método proposto, iniciando com a fase de estimação de DOA para inicializar os filtros. . . . .	32
3.7	Arranjo de microfones e ângulos de chegada. . . . .	32
3.8	Filtros $h'_{ij}(t)$ e $w_{ji}^0(t)$ para duas fontes: uma em $20^\circ$ e outra em $70^\circ$ . . . . .	34
3.9	Filtro passa alta de primeira ordem. . . . .	35
3.10	Branqueamento do sinal. . . . .	36
3.11	Matrizes de correlação para misturas com duas fontes. (a) Correlação antes da separação. (b) Correlação depois da separação. Algoritmo de separação anula as matrizes de correlação cruzada. . . . .	37
4.1	Configuração da sala. . . . .	40
4.2	Resposta impulsional simulada. A resposta possui 7 picos. O primeiro da esquerda para a direita corresponde ao caminho de visada e os outros 6 a uma reflexão em cada uma das paredes. Frequência de amostragem utilizada é de 16 kHz. . . . .	40
4.3	Resultados para os parâmetros sugeridos. . . . .	41
4.4	Cenário $\theta = 30^\circ$ (a) e $\theta = 40^\circ$ (b). Em ambos os casos $\alpha = 60^\circ$ . . . . .	42
4.5	Resultados para os parâmetros sugeridos. . . . .	43
4.6	Comparação utilizando o método de conhecimento prévio de DOA. . . . .	44
4.7	DOA no domínio da frequência para fonte em $40^\circ$ . . . . .	44
4.8	Resultados para filtro de decorrelação. . . . .	45
4.9	Comparação entre distribuições laplacianas e histogramas de cantos de aves de diferentes espécies. O coeficiente $b$ é o fator de escala da curva laplaciana. . . . .	46
4.10	Comparação entre sinais da fontes, sinais misturados e separados. . . . .	47
4.11	Comparação entre sinais da fontes, sinais misturados e separados. . . . .	48

# LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIACÕES

DOA: direção de chegada, do inglês *direction of arrival*.

FDP: função densidade de probabilidade.

FIR: resposta impulsional finita, do inglês *finite impulse response*.

ICA: Análise de Componentes Independentes, do inglês *Independent Component Analysis*.

RIR: resposta impulsional de sala, do inglês *room impulse response*.

SEN: norma do erro do sistema, do inglês *system error norm*.

SINR: relação sinal interferência com ruído, do inglês *signal to interference plus noise ratio*.

SIR: relação sinal interferência, do inglês *signal to interference ratio*.

$h_{ij}$ : resposta impulsional de sala entre a fonte  $i$  e microfone  $j$ .

$w_{ji}$ : filtro de separação entre microfone  $j$  e estimativa  $i$ .

$s_i(t)$ : sinal da fonte  $i$ .

$x_j(t)$ : sinal capturado por um sensor  $j$ .

$y_i(t)$ : sinal estimado da fonte  $i$ .

$\mathbf{h}$ : vetor contendo coeficientes do filtro  $h_{ij}$ .

$\mathbf{w}$ : vetor contendo coeficientes do filtro  $w_{ji}$ .

$\mathbf{W}$ : matriz contendo vetores  $\mathbf{w}$ .

$\mathbf{x}$ : vetor contendo amostras do sinal  $x_j(t)$ .

$\mathbf{y}$ : vetor contendo amostras do sinal  $y_i(t)$ .

$\mathbf{P}$ : matriz de permutação.

$\mathbf{D}$ : matriz diagonal.

$\mathbf{I}$ : matriz identidade.

$diag\cdot$ : operador que elimina os elementos que estão fora da diagonal de uma matriz.

$u_{ii'}(t)$ : resposta do sistema entre uma fonte  $i$  e uma saída estimada  $i'$ .

$\mathbf{A}$ : matriz de mistura linear.

$P(\cdot)$ : pressão do ar.

$\mathcal{J}(\cdot)$ : função de custo.

$\nabla\cdot$ : operação do gradiente.

$\tilde{\nabla}\cdot$ : operação do gradiente natural.

$p_Y(y)$ : FDP da variável aleatória  $Y$ .

$\mathbf{J}$ : matriz jacobiana.

$\det\cdot$ : operador para cálculo do determinante.

$|\cdot|$ : operador para cálculo do valor absoluto.

$P(\omega, \mathbf{r}, \mathbf{r}')$ : pressão para uma frequência  $\omega$  gerada por fontes em uma posição representada pelo vetor  $\mathbf{r}$  em microfones em uma posição indicada pelo vetor  $\mathbf{r}'$ .

$cor(x, y)$ : operador correlação entre duas variáveis aleatórias.

## SÍMBOLOS GREGOS

$\theta$ : ângulo de chegada de sinais de uma fonte sonora.

$\sigma$ : desvio padrão de uma variável aleatória.

$\delta(t)$ : impulso de dirac.

$\mu$ : tamanho do passo.

# 1 INTRODUÇÃO

## 1.1 MOTIVAÇÃO

Misturas sonoras são frequentemente observadas nos mais diversos ambientes do dia a dia como salas de aula, escritórios, lojas, meios de transporte entre outros. Essas misturas ocorrem quando mais de uma fonte sonora estão ativas simultaneamente, sejam duas ou mais pessoas falando ao mesmo tempo ou aparelhos eletrônicos e mecânicos ligados e emitindo ondas sonoras. As versões mais recentes de algoritmos de Separação Cega de Fontes, do inglês *Blind Source Separation* (BSS), têm utilizado sinais capturados por arranjos de microfones (vide Capítulo 2) para a separação dos sons chegando de diferentes direções por meio da adaptação de filtros de separação. As relações sinal interferência, do inglês *Signal to Interference Ratio* (SIR), consideravelmente altas mostram a eficácia deste tipo de processamento. Porém, esses algoritmos possuem alguns desafios. A SIR decai claramente para ângulos de chegada oblíquos a um arranjo de microfones ou quando as fontes estão próximas umas das outras. Para tentar resolver este problema podemos inicializar o algoritmo de BSS de forma a esperar uma melhora na qualidade da separação. Uma forma de inicialização seria criando filtros de separação iniciais utilizando informações já obtidas anteriormente. Técnicas de processamento de sinais em arranjo permitem a estimação das direções de chegada, do inglês *Direction of Arrival* (DOA), de sinais sonoros a um arranjo de microfones que poderiam ser utilizadas para a inicialização dos filtros. Este trabalho utiliza este tipo de inicialização de filtros e também realiza o branqueamento dos sinais para tentar melhorar a SIR, ou seja, a qualidade da separação de algoritmos de BSS.

Poucos tipos de fontes de áudio foram estudados a fim de se realizar a separação cega. Um ramo do estudo da fauna silvestre utiliza o som dos animais a fim de monitorá-los e estudar o seu comportamento. Um gravador em tal ambiente também se depara por vezes com misturas sonoras. Veremos que é possível aplicar com sucesso algoritmos de separação para o estudo de sons da fauna silvestre, mais especificamente, veremos que é possível separar os sons de cantos de pássaros.

## 1.2 TÍPICO CENÁRIO E SEUS CASOS CRÍTICOS

Um cenário típico para a utilização de um algoritmo de BSS ocorre quando duas ou mais fontes sonoras são ativadas em um mesmo ambiente. Para o receptor do sinal, seja uma pessoa ou um equipamento, as fontes interferem entre si. Essa interferência degrada a qualidade da informação recebida e reduz o desempenho e a efetividade de dispositivos que processem áudio como próteses auditivas, televisores inteligentes e gravadores forenses[13], [14], [15] dentre outros.

Neste cenário, o algoritmo de BSS pode ser empregado, porém existem algumas situações onde este tipo de algoritmo possui funcionamento não aceitável. Na Figura 1.1, podemos ver setores que caracterizam situações de desempenho favoráveis e não favoráveis à separação. Em casos favoráveis, as fontes estão bem espaçadas e com ângulos de chegada não muito oblíquos ao arranjo de microfones. Nos casos críticos, as fontes estão próximas entre si ou posicionadas distantes do eixo de separação dos semiplanos esquerdo e direito, ou seja, com ângulos de chegada oblíquos.

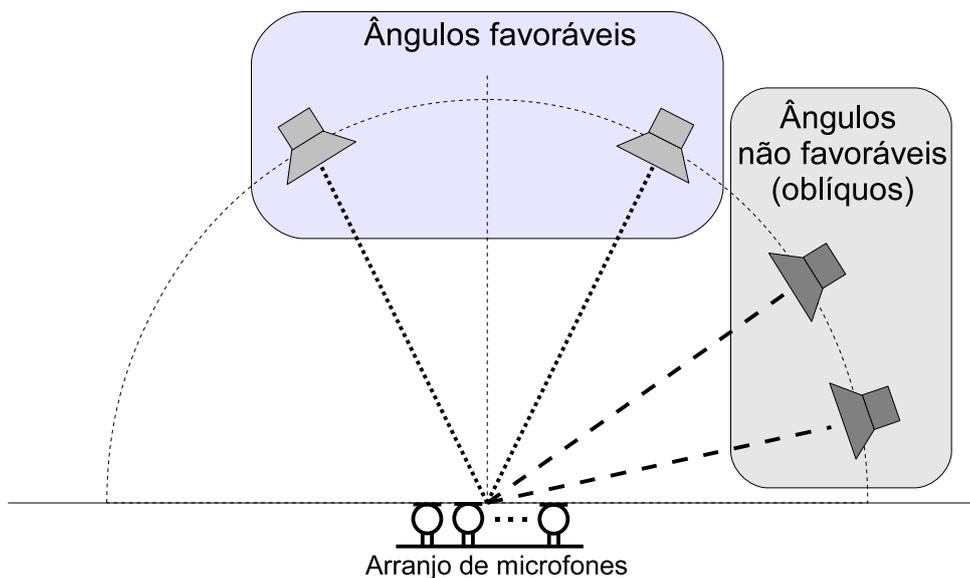


Figura 1.1: Típico cenário e seus casos críticos.

Em muitos casos, os sinais gravados em cada sensor são modelados como somas ponderadas dos sinais das fontes. Porém, em cenários reais, quando fontes de áudio ou voz estão ativas ao mesmo tempo, é esperado que uma mistura convolutiva seja observada nos sensores [18], neste caso, microfones. Este tipo de mistura ocorre pois os sinais percorrem múltiplos caminhos e assim várias versões do mesmo sinal com diferentes

atrasos chegam a cada microfone realizando uma mistura convolutiva dos sinais das fontes. Esses múltiplos caminhos são geralmente causados por reflexões em paredes e outros objetos presentes no ambiente.

Revisaremos um algoritmo clássico de análise de componentes independentes (ICA) que mostrou um bom funcionamento para BSS no domínio da frequência. Porém, estudos feitos por este trabalho mostrarão que o desempenho deste tipo de algoritmo tende a se degradar consideravelmente em fontes posicionadas em ângulos oblíquos ao arranjo de microfones. A fim de suavizar este problema, este trabalho busca alternativas para melhoria de desempenho de algoritmos BSS quando expostos à casos não favoráveis à separação.

A partir de informações prévias da DOA, é possível incluir informações sobre o posicionamento das fontes. Algoritmos de BSS efetuam uma separação espacial das fontes e pretendemos auxiliar o algoritmo por meio da inclusão da informação da DOA antes do processamento da separação. Assim, espera-se melhorar o desempenho do sistema, especialmente nos casos de cenários não favoráveis. Sinais sonoros, como a voz humana possuem, uma alta correlação temporal. Essa correlação pode ser vista como uma concentração da energia do sinal em uma parte do espectro. Para efetuar a separação, algoritmos de BSS precisam efetuar uma decorrelação espacial dos sinais das fontes. Essa decorrelação espacial pode ser seguida de uma decorrelação temporal causando alterações no espectro original do sinal. Homogeneizar o espectro do sinal significa branqueá-lo que por sua vez significa que o sinal está decorrelatado. O branqueamento do sinais de entrada do sistema, antes da separação, deve passar informações apenas sobre a correlação espacial dos sinais e portando um melhor desempenho da separação deve ser observado. Este trabalho se utiliza desses métodos de inicialização para contribuir com o aumento da qualidade da separação para fontes em posições não favoráveis à separação.



Figura 1.2: Diagrama de blocos do método proposto. As variáveis  $x_1, \dots, x_n$  representam as misturas na entrada do sistema e  $y_1, \dots, y_N$  representam os sinais separados.

Os métodos de ICA possuem alguns problemas clássicos de escalonamento e permutação que serão revisados bem como métodos para resolução deste problema. Um diagrama de blocos básico de todo o processo pode ser visto na Figura 1.2. Para avaliar as soluções, foram utilizados cenários artificiais como as que serão vistas no Capítulo 2. Já num ambiente real, um algoritmo já consagrado foi testado para a separação de cantos pássaros com o intuito fornecer um sinal de melhor qualidade para algoritmos capazes de identificar espécies de maneira automática.

### **1.3 ORGANIZAÇÃO DO TRABALHO**

Este trabalho se constitui de quatro capítulos. No Capítulo 1 se encontra esta introdução. O Capítulo 2 expõe o modelo de dados, medidas de desempenho do sistema, um modelo para geração de um ambiente artificial e os outros métodos matemáticos essenciais para o entendimento pleno deste trabalho. Com esse capítulo, espera-se que o leitor tenha uma noção clara das ferramentas necessárias para compreensão e avaliação dos algoritmos BSS. No capítulo 3, é encontrada uma descrição detalhada de uma técnica de separação cega de misturas, dentre outras técnicas complementares para seu sucesso e uma segunda técnica de separação mais genérica e complexa mostrada de maneira mais sucinta. No mesmo Capítulo 3, encontram-se técnicas propostas para a melhoria do desempenho de algoritmos de separação cega em ângulos não favoráveis. O capítulo 4 mostra resultados de experimentos para diversas posições e espaçamentos de fontes, tanto para ângulos favoráveis quanto para ângulos não favoráveis utilizando BSS no domínio da frequência. Simulações utilizando os métodos propostos e os resultados também são expostas. Os resultados serão exibidos para as duas técnicas utilizadas para a melhoria do funcionamento desse mesmo tipo de BSS. No mesmo Capítulo 4, encontram-se experimentos utilizando um cenário real para separação de cantos de pássaros. Por fim, uma conclusão é feita no Capítulo 5.

## 2 MODELOS, MEDIDAS E MÉTODOS

É comum observarmos a presença de vários sons vindos de diferentes fontes sonoras em um mesmo ambiente. Arranjos de microfones vêm sendo empregados para tentar separar os sinais das diferentes fontes sonoras em tais ambientes. Ao se colocar microfones em diferentes posições do ambiente em questão, permite-se a aquisição da informação espacial dos sinais das fontes que pode ser utilizada por diversos algoritmos para efetuar a separação. O tipo de arranjo mais simples e mais comum é o arranjo linear. Este tipo de arranjo consiste de dois ou mais microfones posicionados sobre um segmento de reta e igualmente espaçados por uma distância  $d$  como mostrado na Figura 2.1.

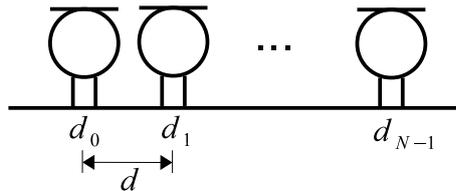


Figura 2.1: Arranjo linear de microfones.

Este tipo de arranjo pode obter informações espaciais das fontes como, por exemplo, estimação de DOA por meio de técnicas adequadas. Uma forma simples de entender como essa informação está presente nos sinais capturados pelo arranjo é pensarmos em um sinal banda estreita viajando com velocidade  $c$ . Neste caso a banda estreita é definida como uma faixa de frequência suficientemente pequena para se considerar que a resposta em frequência seja plana nesta faixa. A fonte também está distante o suficiente para considerarmos que a frente de onda incidindo sobre o arranjo é plana como ilustrado na Figura 2.2.

Com isso em mente, podemos supor que os sinais que chegam nos diferentes microfones são iguais, exceto por uma defasagem no tempo. Essa defasagem depende da DOA que, nesse caso, é determinada pelo ângulo de incidência  $\theta_i$  que é o ângulo entre um eixo perpendicular a base do arranjo e a direção da viagem do sinal  $s_i$ . O subscrito  $i$  indica a fonte emissora do sinal, uma vez que em misturas mais de um sinal estão presentes

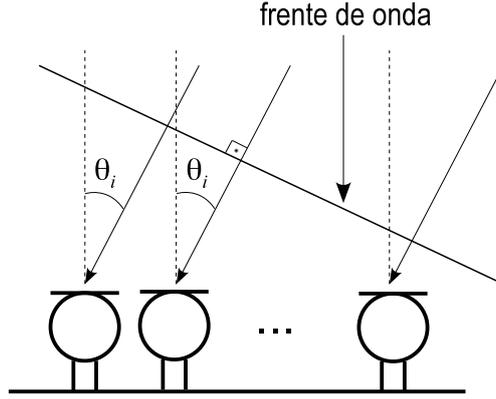


Figura 2.2: Frente de onda planar incidindo sobre ao arranjo.

como veremos mais a frente. A direção da viagem do sinal é perpendicular à frente de onda.

Considerando o exato momento em que a frente de onda atinge um microfone, podemos calcular a distância que a frente de onda ainda precisa percorrer para atingir o próximo microfone. Como a velocidade da frente de onda é conhecida, podemos calcular a defasagem de tempo entre a chegada de um mesmo sinal aos microfones vizinhos a partir do ângulo de chegada  $\theta_i$ . Isso significa que o tempo de chegada tem uma relação direta com a DOA. De maneira mais clara

$$\tau = c^{-1} \cdot d \cdot \sin \theta_i, \quad (2.1)$$

com  $\tau$  sendo a defasagem no tempo de chegada do sinal entre 2 microfones espaçados por uma distância  $d$ . Podemos, ainda, avaliar a defasagem angular  $\varphi$  em função do comprimento de onda  $\lambda$ :

$$\varphi = \omega \cdot \tau = 2\pi \cdot \lambda^{-1} \cdot d \cdot \sin \theta_i, \quad (2.2)$$

em que  $\omega$  é a frequência angular do sinal incidente. Porém, a Equação (2.2) pode possuir mais de uma solução. Várias soluções causam ambiguidades e devemos limitar as soluções para que ambiguidades com relação à  $\theta_i$  não aconteçam. Limitar as soluções para que o módulo de (2.2) esteja no intervalo de  $-\pi/2$  a  $\pi/2$  leva ao equivalente espacial do teorema de Nyquist:

$$d \leq \frac{\lambda}{2}. \quad (2.3)$$

Isto quer dizer que devemos preservar o espaçamento  $d$  entre os microfones menor ou igual à metade do comprimento de onda neste tipo de modelagem em banda estreita para que não ocorram ambiguidades. Com essas restrições, podemos modelar a forma

como um sinal  $x_r(t)$  capturado por um sensor  $r$  de um arranjo linear uniforme se comporta:

$$x_r(t) = e^{-(r-1)j\varphi(\theta_i)} s_i(t), \quad (2.4)$$

onde  $s_i(t)$  é um sinal banda estreita. O termo  $t$  indica o índice do tempo discreto. Para  $N$  sensores em um arranjo linear, reescrevos a Equação (2.4) na forma matricial:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_N(t) \end{bmatrix} = \begin{bmatrix} 1 \\ e^{-j\varphi(\theta_i)} \\ e^{-2j\varphi(\theta_i)} \\ \vdots \\ e^{-Nj\varphi(\theta_i)} \end{bmatrix} s_i(t). \quad (2.5)$$

Caso tenhamos  $K$  fontes, a equação se torna:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ \vdots \\ x_N(t) \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ e^{-j\varphi(\theta_1)} & e^{-j\varphi(\theta_2)} & & e^{-j\varphi(\theta_K)} \\ e^{-2j\varphi(\theta_1)} & e^{-2j\varphi(\theta_2)} & & e^{-2j\varphi(\theta_K)} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-Nj\varphi(\theta_1)} & e^{-Nj\varphi(\theta_2)} & \dots & e^{-Nj\varphi(\theta_K)} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \\ \vdots \\ s_K(t) \end{bmatrix}. \quad (2.6)$$

Se a fonte não estiver no mesmo plano do arranjo linear, a única informação de DOA presente é relativa ao seu azimute, ou seja, ângulo entre o eixo perpendicular à base do arranjo e a projeção da DOA no plano formado entre o arranjo e o eixo perpendicular. Para obtermos além do azimute, a elevação, seria necessária uma outra formação de arranjo capaz de obter informações do eixo vertical.

Na Seção 2.2, veremos a modelagem para misturas de sinais banda larga utilizando um arranjo linear. Veremos mais adiante que as informações contidas no sinais do arranjo sobre a DOA poderão ser utilizadas em benefício à separação de misturas.

## 2.1 ANÁLISE DE COMPONENTES INDEPENDENTES

A fim de estudar a separação de sinais devemos primeiramente tentar entender como misturas são formadas. O modelo básico de Análise de Componentes Independentes, do inglês *Independent Component Analysis* (ICA) [18], tenta mostrar como uma mistura instantânea é capturada por um sensor. Em uma mistura instantânea não há atrasos no tempo. Isso significa dizer que um componente independente, do inglês *independent component* (IC), será capturado de maneira idêntica em todos os sensores salvo uma multiplicação por um peso  $a_{ij}$  conforme a equação:

$$x_j = a_{1j}s_1 + a_{2j}s_2 + \dots + a_{Nj}s_N, \quad (2.7)$$

onde o termo  $x_j$  representa a mistura capturada por um sensor e  $N$  é o número de sinais presentes na mistura. Os índices  $i$  e  $j$  se referem aos ICs e sensores, respectivamente. Note que na Equação (2.7) os termos  $i$  assumem os valores  $\{1, 2, \dots, N\}$ . Nesta etapa ainda consideramos tanto os termos  $x_j$  quanto os ICs  $s_i$  como variáveis aleatórias e não como sinais no tempo propriamente ditos eliminando a necessidade de um termo indicativo do tempo  $t$  como na notação de Hyvärinen [18]. Na Figura 2.3 vemos um diagrama da formação das misturas instantâneas na parte correspondente ao canal, à esquerda da figura. Na parte da direita está o sistema de separação como veremos mais adiante.

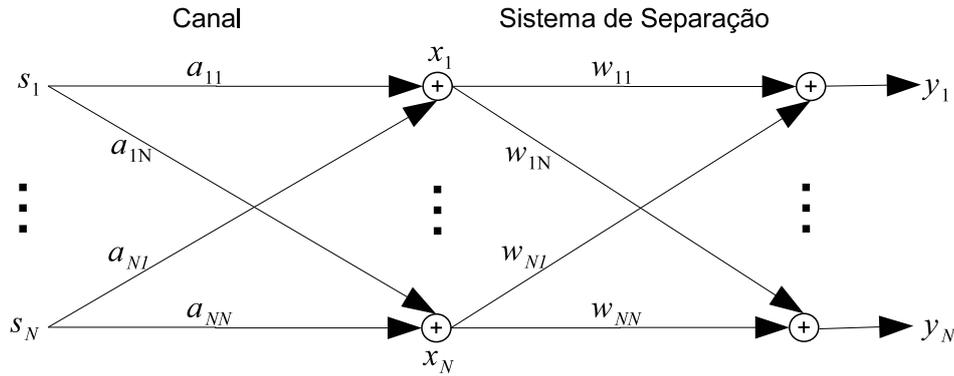


Figura 2.3: Diagrama de mistura e separação instantânea.

Para uma escrita mais compacta, podemos reescrever a Equação (2.7) na forma matricial para mais de uma variável  $x_j$ :

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \quad (2.8)$$

em que  $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ ,  $\mathbf{s} = [s_1, s_2, \dots, s_N]^T$  e  $\mathbf{A}$  uma matriz contendo os coeficientes  $a_{ij}$  da mistura. Este modelo desconsidera a presença de ruídos nos sinais capturados pelos sensores.

Porém, existem ambiguidades de ordenação e escalonamento que devem ser levadas em conta. Para visualizá-las reescreveremos a equação anterior:

$$\mathbf{x} = \sum_{i=1}^N \mathbf{a}_i s_i, \quad (2.9)$$

onde  $\mathbf{a}_i$  é uma coluna de  $\mathbf{A}$ . Agora, é fácil ver que se multiplicarmos  $s_i$  por uma constante qualquer, poderemos sempre dividir  $\mathbf{a}_i$  por essa mesma constante gerando a ambiguidade do escalonamento. Uma vez que não podemos observar nenhum dos

termos à direita da equação, não poderemos determinar a energia de  $s_i$  resultando em um problema de indeterminação do escalonamento dos ICs. Além disso, na Equação (2.9), a soma pode ser feita em qualquer ordem sem alterar a natureza do modelo, o que gera abiguidade quanto à ordenação das fontes. Se não podemos observar os ICs nem  $\mathbf{A}$  então qualquer IC pode ser o primeiro. Este problema é conhecido como problema da permutação [18], já que a ordem dos ICs podem ser permutadas sem que se fuja do modelo.

Para, de fato efetuarmos uma separação precisamos fazer algumas considerações sobre os sinais. A primeira condição importante que devemos considerar é a independência estatística das fontes. De maneira sucinta isto quer dizer que a função densidade de probabilidade (FDP) conjunta  $p(s_1, s_2)$  é igual ao produto das FDPs  $p(s_1)p(s_2)$  para duas variáveis aleatórias  $s_1$  e  $s_2$ . Então, para podermos verificar essa independência, precisamos ter uma certa noção de como as FDPs dos ICs se comportam. Hyvärinen[18] faz algumas observações sobre a análise das PDFs. Seus comportamentos podem ser analisado por meio dos cumulantes de uma variável aleatória. Caso essa variável seja gaussiana, todos os cumulantes a partir do terceiro serão zero e não será possível avaliar a independência. O segundo cumulante nos dá a correlação e Hyvärinen também mostra que a correlação avalia apenas fracamente a independência entre duas ou mais variáveis aleatórias. Portanto, variáveis aleatórias de distribuição gaussiana devem ser evitadas. Caso exista a presença de apenas um sinal de distribuição gaussiana, o modelo ICA ainda é capaz de separá-las. Caso exista mais de um sinal gaussiano, então ainda é possível separar os sinais não gaussianos da mistura e os demais sinais da saída apresentarão misturas dos sinais gaussianos.

A partir da formulação acima, começa uma busca por soluções para reversão da mistura, ou seja, obtenção dos ICs. O problema seria simples se os coeficientes da matriz  $\mathbf{A}$  fossem conhecidos, reduzindo o problema para a inversão desta matriz que será expressa por  $\mathbf{W}$ . Os coeficientes desta matriz são coeficientes vistos no sistema de separação da Figura 2.3. Na prática, como apenas  $\mathbf{x}$  pode ser medido, precisamos utilizar características dos sinais para fazer uma estimativa de  $\mathbf{W}$  como será visto do Capítulo 3.

## 2.2 MODELO DE DADOS PARA MISTURAS CONVOLUTIVAS

Em um ambiente acústico, não são esperadas misturas instantâneas como visto anteriormente. Em vez disso, esperam-se misturas convolutivas como veremos a partir de

agora. Para este trabalho, considera-se um ambiente acústico com  $N$  fontes sonoras emitindo sinais designados por  $s_i(t)$  e todas ativas ao mesmo tempo. Em um ambiente real são esperados atrasos nos sinais que chegam a cada microfone. Em um cenário muito comum, como em uma sala, ainda são esperadas múltiplas reflexões dos sinais nas paredes. Esses atrasos e reflexões causam uma filtragem no sinal tornando necessária a modelagem por uma mistura convolutiva. Uma vez que apenas pesos não são suficientes, precisamos convoluir os sinais com filtros para simular os sinais capturados por microfones. Logo, posicionando um arranjo linear neste ambiente espera-se que a mistura ocorra nos microfones seguindo os fluxos mostrados na figura 2.4. Os sinais serão convoluídos com filtros representados pelas caixas e somados no microfones. A separação acontece de forma semelhante, sendo a soma realizada por um processador digital.

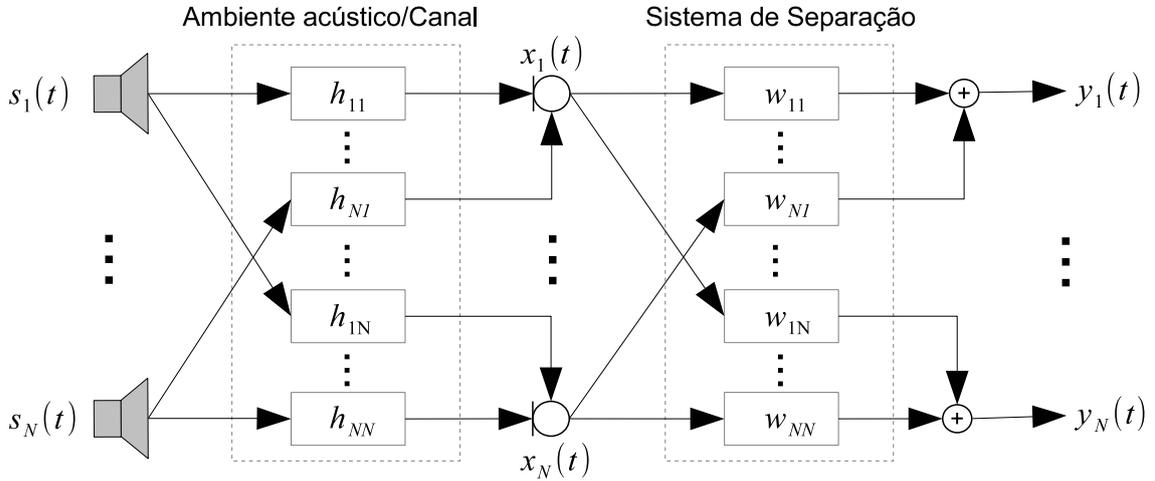


Figura 2.4: Sistema de mistura e separação de sinais.

Modela-se a geometria do ambiente acústico pela sua resposta impulsional  $h(t)$ . A resposta impulsional entre a fonte  $i$  e o microfone  $j$  será denominada por  $h_{ij}(t)$ . Um método de geração de respostas artificiais para salas será visto na Seção 2.4.1. O sinal recebido pelo sensor  $j$  é dado pela convolução e soma dos sinais de entrada conforme a equação:

$$x_j(t) = \sum_{i=1}^N \sum_l h_{ij}(l) s_i(t-l), \quad (2.10)$$

com  $l$  representando o deslocamento no tempo discreto para que se opere a convolução e lembrando que o  $t$  é o índice do tempo discretizado. Os sinais recebidos pelos diversos microfones devem ser separados por meio do emprego de filtros  $w_{ji}(t)$  de comprimento finito  $L$ . Após a criação e adaptação destes filtros, pode-se estimar os sinais de entrada

$s_i$  conforme a equação:

$$y_i(t) = \sum_{j=1}^N \sum_l w_{ji}(l)x_j(t-l). \quad (2.11)$$

De forma simples, os filtros devem idealmente executar a operação inversa dos canais  $h_{ij}(t)$  para gerar estimativas de  $s_i(t)$ , chamadas de  $y_i(t)$ . No caso ideal, cada sinal  $y_i(t)$  possui apenas uma das fontes, ou seja, existem  $N$  fontes ativas e se deseja apenas uma delas em cada uma das saídas. Assim, em cada saída  $y_i$ , tem-se  $N - 1$  sinais indesejados que serão denominados sinais interferentes.

Também podemos analisar o sistema pelo ponto de vista de sua resposta  $u_{ii'}(t)$  entre a entrada  $i$  e a saída  $i'$ . Esta resposta serve para avaliar o sistema sem se utilizar dos sinais, ou seja, utilizando apenas as respostas impulsivas do ambiente acústico e os filtros de separação. Isso é feito pela convolução e soma dos filtros entre uma entrada  $i$  e a saída  $i'$ . Logo, a resposta do sistema é

$$u_{ii'}(t) = \sum_{j=1}^N \sum_l h_{ij}(l)w_{ji'}(t-l) \quad (2.12)$$

e deve ser idealmente zero para as  $N - 1$  fontes interferentes. A equação (2.12) realiza a convolução e soma dos filtros que representam o ambiente acústico e o sistema de separação. Por exemplo, se quisermos analisar o quanto uma fonte  $s_2(t)$  ainda está presente numa saída  $y_1(t)$  podemos realizar a convolução  $s_2*(t)u_{21}(t)$  e verificar o quanto a fonte  $s_2(t)$  foi suprimida.

Neste trabalho, para simplificação, considera-se o mesmo número de fontes e sensores. A maioria dos exemplos será feito por simplicidade com o caso  $2 \times 2$ , isto é, com duas fontes e dois microfones podendo ser diretamente estendidos para um número maior de fontes e microfones.

## 2.3 MEDIDAS DE DESEMPENHO

Para se compararem os resultados e avaliar o desempenho dos estimadores, faz-se necessário o uso de algumas medidas de desempenho. Dentre as disponíveis, a mais utilizada pela literatura é a Relação Sinal Interferência (SIR). Uma outra medida por vezes utilizada é a Norma do Erro do Sistema (SEN). A primeira se utiliza dos sinais de entrada e saída. A segunda se utiliza apenas dos filtros vistos no modelo da seção anterior. Um terceira medida, a Relação Sinal Interferência com Ruído (SINR) também

é idicada, sendo esta para modelos que apresentam ruído além dos sinai interferentes. A seguir será feita uma descrição destas medidas, consideradas mais importantes.

### 2.3.1 Norma do Erro do Sistema

Em aplicações práticas, a estimação dos elementos  $w_{ji}$  do filtro terá resposta impulsional finita (FIR). Desta forma, aproximamos cada elemento por um vetor  $\mathbf{w}_{ji} = [w_{ji}(1), w_{ji}(2), \dots, w_{ji}(M)]^T$ . Para efeito de simulação, também são usados filtros FIR para realizar as misturas convolutivas dos sinais, segundo o modelo do ambiente acústico. Da mesma forma, temos que  $\mathbf{h}_{ij} = [h_{ij}(1), h_{ij}(2), \dots, h_{ij}(N)]^T$ . Podemos, então, medir o quanto o sistema anula as fontes interferentes por meio da

$$\text{SEN} = 10 \log_{10} \left( \frac{\sum_{i \neq i'} \|\mathbf{u}_{ii'}\|^2}{\|\mathbf{u}_{ii}\|^2} \right), \quad (2.13)$$

onde que  $\mathbf{u}_{ii'}$  é um vetor contendo as amostras  $u_{ii'}(t)$ . O seu valor é dado em dB e o operador  $\|\cdot\|$  é a norma euclidiana do vetor. Podemos dizer que a contribuição da fonte de interesse  $\mathbf{u}_{ii'} = \mathbf{u}_{ii}$  no caso em que  $i = i'$ . Por isso, utilizamos esta forma no denominador para simplificação da escrita. O numerador do argumento do logaritmo diz respeito a contribuição das fontes interferentes.

### 2.3.2 Relação Sinal Interferência

A qualidade da estimação do sinal desejado pode ser avaliada por meio da SIR. A SIR relaciona a potência o sinal desejado com os interferentes. Desta forma, temos que a

$$\text{SIR} = 10 \log_{10} \left( \frac{P_s}{P_i} \right), \quad (2.14)$$

onde  $P_s$  é a potência do sinal desejado e  $P_i$  a soma potência dos sinais interferentes e o valor é dado em dB. De maneira mais prática, por exemplo em uma implementação de software, a SIR é calculada em blocos de tamanho fixo. Uma estimativa pode ser calculada da seguinte forma:

$$\text{SIR} = 10 \log \left( \frac{\|\mathbf{v}_k\|^2}{\sum_{k'} \|\mathbf{v}_{k'}\|^2} \right), \quad (2.15)$$

em que  $\mathbf{v}_k$  é um vetor de tamanho  $N$  contendo as amostras do sinal e  $\mathbf{v}_{k'}$  contém apenas a contribuição de um sinal interferente  $k'$ . Esta contribuição pode ser descrita com ajuda da Figura 2.4 (caso  $2 \times 2$ ), mas “ligando” uma das fontes. Por exemplo, se considerarmos que o sinal  $s_1(t)$  é o sinal desejado e  $s_2(t)$  é o sinal interferente, o cálculo da contribuição de  $s_2(t)$  na saída 1 pode ser elaborado seguindo os caminhos a partir

da fonte 2 até os microfones e em seguida até a saída 1. Uma ilustração dos “caminhos” pode ser vista na Figura 2.5. É importante analisarmos que o posicionamento e a intensidade de uma fonte podem interferir na SIR da saída. Se uma fonte estiver com uma intensidade maior que a outra, mesmo que os filtros não separem o sinal, observaremos uma SIR favorável em uma das saídas. Para contornar esse problema, podemos calcular uma SIR relativa ou ganho de SIR denotado por  $SIR_{\text{gain}}$ . Isso significa calcular a SIR nas entradas do sistema de separação e subtraí-las nas saídas correspondentes calculando, assim, apenas a contribuição do sistema de separação:

$$SIR_{\text{gain}} = SIR_y - SIR_x, \quad (2.16)$$

onde  $SIR_x$  é a SIR em um microfones e  $SIR_y$  a SIR na saída correspondente.

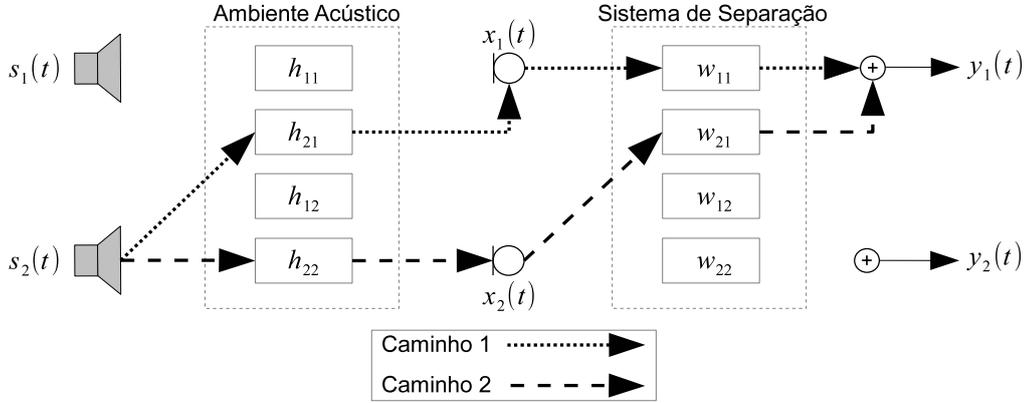


Figura 2.5: Contribuição da fonte 1 na saída 2 para um cenário  $2 \times 2$ .

### 2.3.3 Relação Sinal Interferência com Ruído

Em aplicações práticas, além da presença do sinal interferente, será observada a presença de ruído nas misturas. Para avaliar os algoritmos de separação levando em conta também o ruído, uma medida mais adequada se faz necessária. A Relação Sinal Interferência com Ruído (SINR) mede a razão entre a potência do sinal desejado e a soma da potência dos sinais interferentes e da potência do ruído:

$$SINR = 10 \log_{10} \left( \frac{P_s}{P_i + P_n} \right), \quad (2.17)$$

onde  $P_s$  é a potência do sinal desejado,  $P_i$  a soma potência dos sinais interferentes e  $P_n$  é a potência do ruído. Uma implementação prática de (2.17) pode ser feita de maneira semelhante à Equação (2.15) com a adição de mais um termo ao denominador contendo relativo ao quadrado da norma das amostras do ruído. Em caso de um ruído estacionário, sua norma ao quadrado poderá ser substituída por constante com o valor estimado da potência.

## 2.4 MÉTODOS MATEMÁTICOS

Para evitar os trabalhos de construção de ambientes acústicos adequados e ensaios de gravação, fez-se o uso de simuladores. Este tipo de cenário simulado evita degradações que possam ocorrer devido a imperfeições dos microfones, dos circuitos e dos auto falantes que emitem os sinais das fontes. Para para estudos de viabilidade as simulações são ideias já que é possível iniciar o estudo com ambientes simples e livres de ruídos e imperfeições e adicioná-las gradualmente ao problema ao passo em que problemas são resolvidos. Enfim, o sistema em questão pode ser testado em ambientes reais, o que costuma trazer novos desafios além dos experimentos em ambientes simulados. A seguir, veremos a técnica utilizada para criar ambientes simulados que serão utilizados neste trabalho. Nas Subseções 2.4.2 e 2.4.3 serão mostrados métodos utilizados para a adaptação automática úteis para BSS. Por fim, na Subseção 2.4.4 veremos transformações utilizadas no desenvolvimentos de algoritmos de separação.

### 2.4.1 Método das Imagens

A simulação das respostas impulsivas para pequenas salas retangulares foi descrito por Allen [1], que parte do princípio que fontes sonoras pontuais em espaço livre emitem uma onda para uma frequência individual com pressão

$$P(\omega, \mathbf{r}, \mathbf{r}') = \frac{\exp[i\omega(\mathbf{R}/c - t)]}{4\pi\mathbf{R}}, \quad (2.18)$$

com  $\omega$  sendo a frequência angular,  $c$  a velocidade do som e  $\mathbf{R} = |\mathbf{r} - \mathbf{r}'|$ . Os vetores  $\mathbf{r}$  e  $\mathbf{r}'$  contém as coordenadas cartesianas das fontes e dos microfones, respectivamente. Caso tenhamos no espaço em questão parede rígida e sem absorção (parede refletindo totalmente o sinal), a pressão final será a soma da pressão da fonte e da pressão de uma imagem desta fonte. Esta imagem é formada da mesma forma que um espelho formaria a imagem de um objeto posicionado a sua frente. Neste caso a parede funcionaria como um espelho sonoro, ou seja, criando uma imagem da fonte sonora. A distância da fonte ao microfone é dada por  $\mathbf{R}_{src}$  e da imagem por  $\mathbf{R}_{img}$ . Desta forma,

$$P(\omega, \mathbf{r}, \mathbf{r}') = \frac{\exp[i\omega(\mathbf{R}_{src}/c - t)]}{4\pi\mathbf{R}_{src}} + \frac{\exp[i\omega(\mathbf{R}_{img}/c - t)]}{4\pi\mathbf{R}_{img}}. \quad (2.19)$$

Para o caso de uma sala retangular com 6 paredes (ou 4 paredes, um teto e um chão) e considerando um sistema de primeira ordem, ou seja, com apenas uma reflexão em cada parede, é feita a soma sobre o conjunto  $\mathcal{R}$  de vetores que representa a posição da fonte e suas imagens:

$$P(\omega, \mathbf{r}, \mathbf{r}') = \sum_{\mathbf{r} \in \mathcal{R}} \frac{\exp[i\omega(\mathbf{R}/c - t)]}{4\pi\mathbf{R}}. \quad (2.20)$$

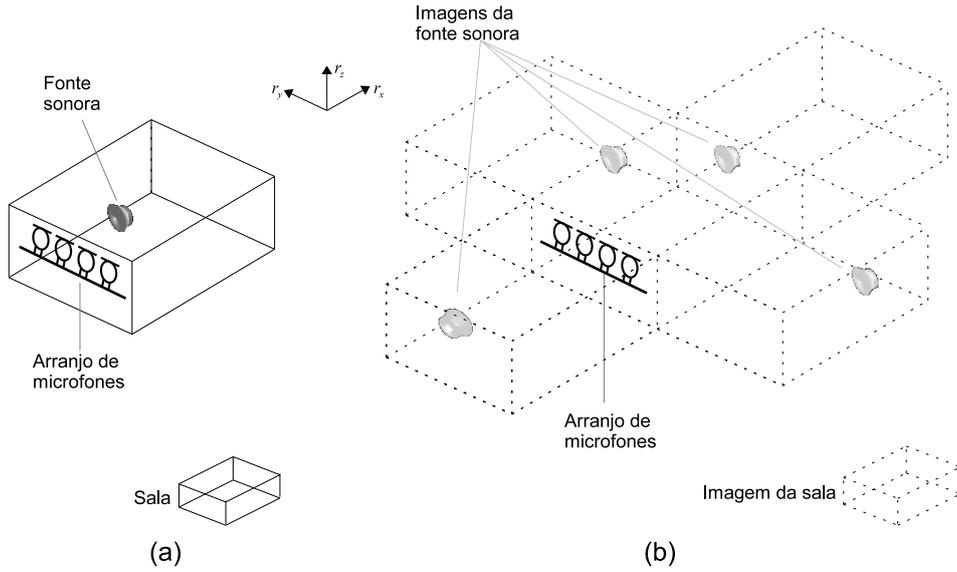


Figura 2.6: (a) Ilustração de uma sala com um arranjo e uma fonte sonora. (b) Ilustração de um arranjo e de imagens das fontes sonoras.

Transformando para o domínio do tempo, teremos a resposta impulsional da sala:

$$p(t, \mathbf{r}, \mathbf{r}') = \sum_{\mathbf{r} \in \mathcal{R}} \frac{\delta[t - (\mathbf{R}/c)]}{4\pi \mathbf{R}}. \quad (2.21)$$

Uma ilustração pode ser vista na Figura 2.6(b). Nela podem ser vistas apenas as imagens para as quatro paredes, pois as imagens no chão e no teto foram omitidas. É importante ressaltar que no caso de primeira ordem é necessário adicionar as imagens do chão e do teto.

### 2.4.2 Método do Gradiente

Muitos problemas práticos são modelados por uma função de custo  $\mathcal{J}(\mathbf{w})$  que deve ser minimizada ou maximizada. Nestes casos,  $\mathbf{w}$  é um vetor que assume valores  $[w_1, w_2, \dots, w_N]^T$ . Em alguns casos, a busca por este vetor  $\mathbf{w}$  que minimiza a função é feita de maneira adaptativa para evitar cálculos longos e excessivo gasto computacional. Uma maneira conveniente de se encontrar um mínimo é o método do gradiente, pelo qual o vetor  $\mathbf{w}$  percorre o gradiente de uma função, adaptativamente, a cada novo conjunto de dados. O gradiente da função  $\mathcal{J}(\mathbf{w})$  com respeito a  $\mathbf{w}$  é definido como

$$\nabla \mathcal{J}(\mathbf{w}) = \frac{\partial \mathcal{J}(\mathbf{w})}{\partial \mathbf{w}} = \begin{pmatrix} \frac{\partial \mathcal{J}(\mathbf{w})}{\partial w_1} \\ \frac{\partial \mathcal{J}(\mathbf{w})}{\partial w_2} \\ \vdots \\ \frac{\partial \mathcal{J}(\mathbf{w})}{\partial w_N} \end{pmatrix}. \quad (2.22)$$

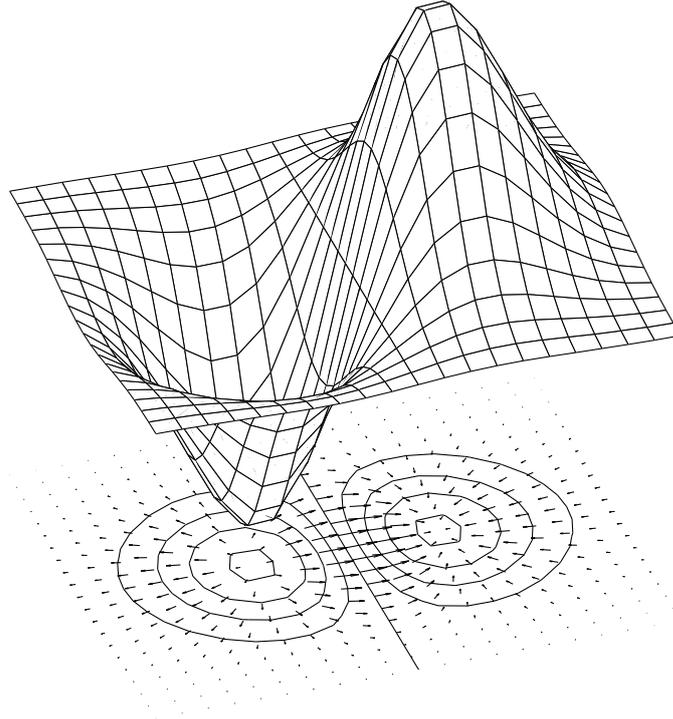


Figura 2.7: Máximos e mínimos de uma função mostrando projeções de curvas de contorno e seu gradiente.

Podemos, então, denotar este vetor por  $\mathbf{w}(t)$ , com  $t$  indicando o tempo. A cada nova etapa de processamento, ou seja, a cada novo instante  $t$ , calcula-se um novo  $\mathbf{w}$ . Esse método, primeiramente proposto em 1847 [10], caracteriza o conhecido método *steepest descent* ou método do gradiente utilizado para minimizar a função de custo:

$$\mathbf{w}(t) = \mathbf{w}(t - 1) - \mu \nabla \mathcal{J}(\mathbf{w})|_{\mathbf{w}=\mathbf{w}(t-1)}. \quad (2.23)$$

Para dar partida ao algoritmo, é necessário escolher um valor inicial para  $\mathbf{w}$ . A escolha deste valor, algumas vezes, é aleatória e outras vezes é predeterminada, de acordo com o problema em questão. Uma vez decidido o valor inicial  $\mathbf{w}(t = 0)$ , o vetor  $\mathbf{w}$  assumirá um novo valor a cada novo instante  $t$ . Isso significa que parte-se do ponto anterior somando mais um passo na direção oposta ao gradiente neste mesmo ponto. O escalar  $\mu$  define o tamanho desse passo, ou seja, ele escalona o vetor do gradiente. Sua escolha é importante. Caso ele seja muito pequeno, podem ser necessários muitos passos até que se chegue ao ponto desejado, desperdiçando poder computacional. Caso seja muito grande, o vetor pode “saltar” indefinidamente por cima e ao redor do ponto mínimo sem nunca alcançá-lo. Seu valor é negativo uma vez que queremos seguir no sentido do mínimo, oposto ao gradiente, e não do máximo da função. Com um valor adequado para  $\mu$ , passos são dados até que o vetor não evolua mais com o tempo. Essa evolução é geralmente medida pela distância euclidiana  $\|\mathbf{w}(t) - \mathbf{w}(t - 1)\|$ . Um valor suficientemente pequeno para esta distância pode ser ajustado de forma que, uma vez

atingido, o algoritmo considera o valor de  $\mathbf{w}$  como suficientemente perto do ponto de mínimo.

Note que este tipo de algoritmo funciona bem quando existe apenas um mínimo na função. Caso existam além do mínimo global, mínimos locais, o algoritmo poderá ficar “preso” em um destes mínimos e a distância euclidiana indicar o alcance de um bom valor. Em alguns casos, a escolha de um valor inicial  $\mathbf{w}(0)$  também pode ser crucial, especialmente para se evitar os mínimos locais. Na Figura 2.7 vemos um máximo e um mínimo da função bem distintos, bem como uma linha de contorno que divide a parte esquerda, onde fica o mínimo, e a direita, onde fica o máximo. Caso este valor inicial esteja do lado esquerdo, todos os vetores do gradiente levarão ao ponto mínimo. Já, caso esteja na face à direita do pico, o algoritmo poderá não convergir, pois não “subirá” em direção ao máximo. Assim, se for escolhido um ponto sobre a face à direita do pico, o gradiente levará para a direita, ou seja, no sentido contrário ao ponto onde se encontra o valor mínimo.

### 2.4.3 Método do Gradiente Natural

O gradiente de uma função de custo aponta para a direção da subida mais acentuada no caso do espaço euclidiano. Em alguns casos, o espaço pode ter uma estrutura de Riemann [4], como é o caso das funções de custo de vários algoritmos de separação cega. Neste caso o gradiente não apontará para o sentido de maior descida do gradiente. Nestes casos o gradiente natural proposto por Amari [3] e descrito na equação abaixo é aconselhado:

$$\tilde{\nabla} \mathcal{J}(\mathbf{w}) = \mathbf{G}^{-1}(\mathbf{w}) \nabla \mathcal{J}(\mathbf{w}) \quad (2.24)$$

Para modelos estatísticos o espaço dos parâmetros, no caso  $\mathbf{w}$ , é um espaço de Riemann em que a matriz  $\mathbf{G}$  é também conhecida como a matriz de Fischer [4]. Na separação cega de misturas,  $\mathbf{G}^{-1}$  assume a forma  $\mathbf{W}^T \mathbf{W}$  [3], com  $\mathbf{W}$  sendo a matriz de separação como visto no modelo de dados. Desta forma, o gradiente natural para ambientes de separação cega geralmente toma a forma:

$$\tilde{\nabla} \mathcal{J}(\mathbf{W}) = \nabla \mathcal{J}(\mathbf{W}) \mathbf{W}^T \mathbf{W}. \quad (2.25)$$

A partir da equação anterior obtém-se um algoritmo mais eficiente de descida do gradiente para separação cega de misturas sonoras.

### 2.4.4 Funções de Densidade de Probabilidade de Transformações

A separação de sinais é feita, na maioria das vezes, por meio de informações estatísticas que dão pistas sobre o formato de suas funções densidade de probabilidade (FDPs). Ao

efetuar uma mistura ou separação dos sinais, trabalhamos com transformações lineares sobre os sinais misturados. Portanto, o conhecimento de como essas transformações agem sobre as FDPs é fundamental para a construção de algoritmos de separação. O nosso objetivo é conseguir encontrar a FDP  $p_{\mathbf{Y}}(\mathbf{y})$  de um vetor  $\mathbf{y} = \mathbf{W}\mathbf{x}$  de variáveis aleatórias que é uma transformada linear de outro vetor de variáveis aleatórias  $\mathbf{x}$ . Esse procedimento foi descrito por Hyvärinen [18], Caselha [9] e outros.

Considerando uma variável aleatória  $X$  com densidade  $p_X(x)$  e uma função monotônica  $y = g(x)$ , queremos encontrar  $p_Y(y)$  a partir de  $p_X(x)$ . No caso favorável, podemos encontrar uma única solução  $g^{-1}(x)$  onde  $p_X(x) \neq 0$ . Assim, a probabilidade de  $Y$  estar em uma faixa de valores entre  $a$  e  $b$  é a probabilidade de  $X$  estar entre  $g^{-1}(a)$  e  $g^{-1}(b)$ . Desta forma,

$$P(a < Y < b) = P(g^{-1}(a) < X < g^{-1}(b)), \quad (2.26)$$

que é o mesmo que calcular a integral da PDF  $\int_{g^{-1}(a)}^{g^{-1}(b)} p_X(x) dx$ . Podemos tentar calcular a integral fazendo uma mudança de variável. Logo,

$$\int_{g^{-1}(a)}^{g^{-1}(b)} p_X(x) dx = \int_a^b p_X(g^{-1}(y)) \frac{dg^{-1}(y)}{dy} dy \quad (2.27)$$

Calcular a probabilidade de  $Y$  pertencer a um intervalo é o mesmo que calcular a diferença dos valores das funções de distribuição acumulada  $F(\cdot)$  nos limites do intervalo. Assim, podemos calcular a probabilidade (2.26):

$$P(a < Y < b) = \int_a^b p_Y(y) dy = F(b) - F(a). \quad (2.28)$$

Então,

$$p_Y(y) = p_X(g^{-1}(y)) \frac{dg^{-1}(y)}{dy}. \quad (2.29)$$

Porém, a igualdade acima só é válida quando  $g(\cdot)$  for uma função crescente. Quando  $g(\cdot)$  for decrescente,

$$P(a < Y < b) = P(g^{-1}(b) < X < g^{-1}(a)) = - \int_a^b p_X(g^{-1}(y)) \frac{dg^{-1}(y)}{dy} dy. \quad (2.30)$$

Isso porque  $dg^{-1}(y)/dy$  é negativo para  $g(\cdot)$  decrescente. Desta forma, para que tenhamos uma equação geral para todos os casos, podemos escrever

$$p_Y(y) = p_X(g^{-1}(y)) \cdot \left| \frac{dg^{-1}(y)}{dy} \right|. \quad (2.31)$$

Quando temos misturas de sinais, possuímos obviamente mais de um sinal. Por isso, é aconselhável a utilização de uma notação matricial adequada. Definimos, então,

os vetores  $\mathbf{y} = [y_1, y_2, \dots]^T$  e  $\mathbf{x} = [x_1, x_2, \dots]^T$ . De forma semelhante, fazendo uma transformação para o caso multidimensional, temos que

$$p_{\mathbf{Y}}(\mathbf{y}) = p_{\mathbf{X}}(g^{-1}(\mathbf{y})) \cdot |\det \mathbf{J}(\mathbf{g}(g^{-1}(\mathbf{y})))|^{-1}, \quad (2.32)$$

em que  $\mathbf{J}$  é a matriz jacobiana. Esta é definida como

$$\mathbf{J}(\mathbf{g}(\mathbf{x})) = \begin{pmatrix} \frac{\partial g_1(\mathbf{x})}{\partial x_1} & \frac{\partial g_2(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial g_M(\mathbf{x})}{\partial x_1} \\ \frac{\partial g_1(\mathbf{x})}{\partial x_2} & \frac{\partial g_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial g_M(\mathbf{x})}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_1(\mathbf{x})}{\partial x_N} & \frac{\partial g_2(\mathbf{x})}{\partial x_N} & \dots & \frac{\partial g_M(\mathbf{x})}{\partial x_N} \end{pmatrix}. \quad (2.33)$$

Para uma transformação linear do tipo  $\mathbf{y} = \mathbf{W}\mathbf{x}$ , é fácil ver que  $\mathbf{J}(\mathbf{W}\mathbf{x}) = \mathbf{W}$  e, assim, obter uma forma simplificada da Equação (2.32):

$$p_{\mathbf{Y}}(\mathbf{y}) = p_{\mathbf{X}}(\mathbf{W}^{-1}\mathbf{y}) \cdot |\det \mathbf{W}|^{-1}. \quad (2.34)$$

### 3 SEPARAÇÃO CEGA DE FONTES

Como visto anteriormente, é comum a presença de mais de uma fonte sonora ativa simultaneamente em um mesmo cenário gerando misturas convolutivas. Chamou-se Separação Cega de Fontes (BSS) o processo de separar estas fontes sem que se tenha um conhecimento prévio do cenário em questão. Vários trabalhos apresentam soluções para se obterem parâmetros dos filtros  $w_{ji}(t)$  apresentados na Equação (2.11) que efetuarão a separação [28], [8], dentre outros. Essas soluções fazem a abordagem do problema no domínio do tempo ou no domínio da frequência. Existem vantagens e desvantagens para cada caso. Os algoritmos no domínio do tempo, como acontece com o TRINICON (*Triple-N ICA for Covolutive Mixtures*), resultam numa melhor SIR que os algoritmos no domínio da frequência [8]. Já os algoritmos no domínio da frequência tendem a ter um menor esforço computacional. Infelizmente, para ambos os domínios, o desempenho diminui quando o sistema é exposto à ângulos oblíquos de radiação (vide Capítulo 1).

A seguir, mostraremos uma solução para ICA no domínio da frequência e, posteriormente e de maneira mais sucinta, outro algoritmo mais complexo no domínio do tempo.

#### 3.1 SEPARAÇÃO CEGA NO DOMÍNIO DA FREQUÊNCIA

A idéia básica de um algoritmo de separação cega no domínio da frequência, é utilizar algoritmos de separação de misturas instantâneas para separar misturas convolutivas. Isso é feito considerando cada frequência do sinal como uma mistura instantânea [28]. O sinal no domínio da frequência é obtido usando a transformada discreta de Fourier. Com uma mistura linear em cada frequência podemos fazer formulação semelhante à formulação básica do ICA:

$$\mathbf{X}(f, m) = \mathbf{A}(f)\mathbf{S}(f, m). \quad (3.1)$$

Onde  $\mathbf{A}(f)$  é uma matriz de mistura linear na frequência  $f$  e  $m$  o índice de um quadro no tempo. São usadas letras maiúsculas para indicar que as variáveis estão no domínio da frequência. A busca por uma matriz de separação caracteriza um problema de ICA. O seu objetivo é encontrar a matriz  $\mathbf{W}$  para cada frequência de forma que

$$\mathbf{Y}(f, m) = \mathbf{W}(f)\mathbf{X}(f, m) \quad (3.2)$$

é a estimativa do sinal separado. Dentre os métodos de ICA disponíveis para encontrar  $\mathbf{W}(f)$ , utilizaremos o método descrito por Amari [2] aliado ao método gradiente natural para encontrar a matriz de separação de maneira iterativa como será visto a partir da próxima seção. Uma função de ativação eficiente também foi sugerida [24] para este tipo de solução. Este algoritmo foi utilizado para desenvolver e testar técnicas de correção do problema da permutação [25, 26] que serão revisadas mais adiante.

### 3.1.1 Diferença Entrópica

Com o estabelecimento de algumas restrições, o problema da separação cega de sinais pode ser facilitado (Seção 2.1). No presente estudo se considera que as diversas fontes sejam estatisticamente independentes e não gaussianas e que o número de fontes é igual ao número de sensores.

No caso da simplificação acima, os parâmetros do filtro  $\mathbf{W}$  podem ser estimados pela maximização da independência das saídas do sistema de separação, também conhecido como maximização da informação [18]. Existem outras métricas para avaliar a separação sendo mais comum a utilização de medidas de não gaussianidade como kurtosis e a negentropia [17]. Essas medidas avaliam momentos de ordem maior que 2 da FDP, necessitando que etapas de decorrelação sejam efetuadas anteriormente ao processo de separação. A maximização da independência avalia a FDP como um todo eliminando a necessidade destas etapas de decorrelação. Para medir a independência, Amari [2] avalia a distância entre a FDP conjunta da saída  $p(\mathbf{Y})$  e o produto de suas distribuições marginais  $\prod_i p(Y_i)$ . Uma vez que as saídas se tornem independentes, esse produto tende a se igualar à FDP conjunta e, portanto, a distância tende a zero. A medida de distância utilizada é a distância de Kullback-Leibler [11] que é definida como:

$$D(p(x)||q(x)) = \int p(x) \log \frac{p(x)}{q(x)} dx, \quad (3.3)$$

onde  $p(x)$  e  $q(x)$  são as duas FDPs entre as quais se deseja medir a distância. No nosso caso mediremos  $D(p(\mathbf{Y})||\prod_i^N p(Y_i))$ . Verificando que  $\mathbf{Y}(f, m) = \mathbf{W}(f)\mathbf{X}(f, m)$ , a equação acima é definida como a função de custo  $\mathcal{J}(\mathbf{W}(f))$  que depende de  $\mathbf{W}(f)$ . Essa função é correspondente à função de custo como vista na seção 2.4.2. Por simplicidade de escrita, omitiremos os argumentos  $f$  e  $m$  a partir deste ponto, tornando nossa equação de custo é igual a

$$\mathcal{J}(\mathbf{W}) = D(p(\mathbf{Y})||\prod_i^N p(Y_i)) = \int p(\mathbf{Y}) \log \frac{p(\mathbf{Y})}{\prod_i^N p(Y_i)} d\mathbf{Y}. \quad (3.4)$$

Calcular esta distância é o mesmo que calcular a diferença das entropias ou informação mútua [11]:

$$\mathcal{J}(\mathbf{W}) = -H(\mathbf{Y}) + \sum_i^N H(Y_i), \quad (3.5)$$

em que  $H(\cdot)$  é a entropia que é definida como  $H(Y_i) = -\int p(Y_i) \log p(Y_i) dY_i$ .

O problema que surge em seguida é o da avaliação das FDPs, já que num ambiente de separação cega, não temos informações sobre as fontes e geralmente suas distribuições também são desconhecidas. Se elas fossem conhecidas, o problema se reduziria na busca por parâmetros suficientes para descrever esta distribuição. Porém, Amari [2] utiliza as expansões de Gram-Charlier truncadas esperando uma aproximação mais genérica, ou seja, para diferentes tipos de distribuição. Esta expansão é aproximada por:

$$p(Y_i) \approx \alpha(Y_i) \left\{ 1 + \frac{\kappa_3^i}{3!} C_3(Y_i) + \frac{\kappa_4^i}{4!} C_4(Y_i) \right\}, \quad (3.6)$$

$$(-1)^k \frac{d^k \alpha(Y)}{dy^k} = C_k(Y) \alpha(Y), \quad (3.7)$$

$$C_3(Y) = Y^3 - 3Y, \quad (3.8)$$

$$C_4(Y) = Y^4 - 6Y^2 + 3, \quad (3.9)$$

em que  $\kappa_3^i = m_3^i$ ,  $\kappa_4^i = m_4^i - 3$  (excesso de curtose) e  $m_n^i$  é o  $n$ -ésimo momento de  $Y_i$ ,  $\alpha(Y_i) = \frac{1}{\sqrt{2\pi}} e^{-\frac{Y_i^2}{2}}$  e  $C_k(Y_i)$  são polinômios de Chebyshev-Hermite definidos pela igualdade (3.7). A expansão (3.6) é resultado do truncamento da expansão de Edgeworth até o termo  $C_4(Y_i)$ , o que caracteriza a aproximação [19], que é, de forma simples, uma distribuição gaussiana com duas correções: uma com relação a obliquidade (terceiro momento - *skewness*) e outra com relação ao excesso de kurtosis (*excess kurtosis*). O cálculo de  $C_k(Y)$  para  $k = 3$  e  $k = 4$  levam as Equações (3.8) e (3.9), respectivamente. Um gráfico com as distribuições de Laplace, Gauss e aproximação de Gram-Charlier podem ser vistos na Figura .

Desenvolvendo a expressão da entropia  $H(Y_i)$  é possível chegar à seguinte aproximação:

$$H(Y_i) \approx \frac{1}{2} \log 2\pi e - \frac{(\kappa_3^i)^2}{2 \cdot 3!} - \frac{(\kappa_4^i)^2}{2 \cdot 4!} - \frac{5}{8} (\kappa_3^i)^2 \kappa_4^i - \frac{1}{16} (\kappa_4^i)^3. \quad (3.10)$$

Uma vez que  $\mathbf{y}$  é uma transformação linear de  $\mathbf{x}$ , podemos obter por meio da relação (2.34) que

$$H(\mathbf{Y}) = H(\mathbf{X}) + \log |\det(\mathbf{W})|. \quad (3.11)$$

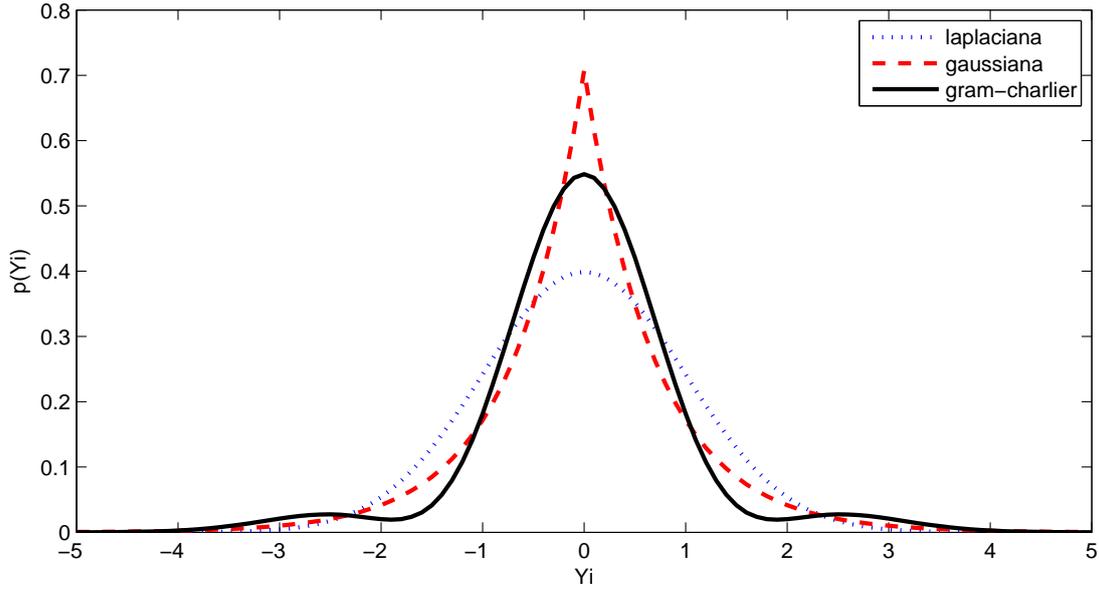


Figura 3.1: Representação das PDFs de distribuição de Laplace, distribuição de Gauss e de Gram-Charlier. As PDFs foram desenhadas considerando a média igual a zero e variância igual a um.

Por fim, substituímos as equações (3.10) e (3.11) em (3.5) e obtemos a aproximação

$$\mathcal{J}(\mathbf{W}) \approx -H(\mathbf{X}) - \log |\det(\mathbf{W})| + \frac{N}{2} \log 2\pi e - \sum_{i=1}^N \frac{(\kappa_3^i)^2}{2 \cdot 3!} - \frac{(\kappa_4^i)^2}{2 \cdot 4!} - \frac{5}{8} (\kappa_3^i)^2 \kappa_4^i - \frac{1}{16} (\kappa_4^i)^3. \quad (3.12)$$

### 3.1.2 Gradiente da Diferença Entrópica

Uma vez que a função de custo foi definida, devemos procurar minimizá-la a fim de encontrar o ponto onde as saídas estão mais independentes. Essa minimização deverá ser feita com relação a  $\mathbf{W}$ , pois é  $\mathbf{W}$  que irá efetuar a separação. Como vimos anteriormente, a função terá seu mínimo igual a zero para o ponto onde os coeficientes de  $\mathbf{W}$  tornam as saídas  $Y_i$  independentes. Sendo assim, o negativo do gradiente da função de custo apontará para o sentido de maior separação. Logo, utilizando um algoritmo de descida do gradiente poderemos gerar os coeficientes de  $\mathbf{W}$  adaptativamente.

Para obter um algoritmo de descida ao longo do gradiente, como visto no capítulo anterior, precisamos calcular as derivadas parciais da função de custo com relação aos elementos  $W_{ji}$  da matriz  $\mathbf{W}$ . O gradiente será calculado no domínio dos reais e ao fim adaptado para o domínio complexo. Assim, as derivadas parciais da aproximação terão

a forma:

$$\begin{aligned} \frac{\partial \mathcal{J}(\mathbf{W})}{\partial W_{ji}} \approx & -(\mathbf{W}^{-T})_{ji} + \left\{ \frac{15}{3} \kappa_3^i \kappa_4^i - \frac{1}{2} \kappa_3^i \right\} E[(Y_i)^2 X_j] \\ & + \left\{ \frac{5}{2} (\kappa_3^i)^2 - \frac{1}{6} \kappa_4^i + \frac{3}{4} (\kappa_4^i)^2 \right\} E[(Y_i)^3 X_j]. \end{aligned} \quad (3.13)$$

onde  $(\mathbf{W}^{-T})_{ji}$  é o elemento da linha  $j$  e coluna  $i$  de  $(\mathbf{W}^T)^{-1}$ . Feito isso, ainda temos que estimar os momentos de  $Y_i$ . De maneira grosseira, porém rápida, Amari [2] usa os valores instantâneos de  $Y_i$  como a estimativa destes momentos. Os valores instantâneos também são utilizando como estimativa das esperanças  $E[(Y_i)^2 X_j]$  e  $E[(Y_i)^3 X_j]$ . Com isso, uma forma muito mais simples de (3.13) é obtida:

$$\frac{\partial \mathcal{J}(\mathbf{W})}{\partial W_{ji}} \approx -(\mathbf{W}^{-T})_{ji} + \left( \frac{3}{4} Y_i^{11} + \frac{25}{4} Y_i^9 - \frac{14}{3} Y_i^7 - \frac{47}{4} Y_i^5 + \frac{29}{4} Y_i^3 \right) X_j. \quad (3.14)$$

Para obtermos uma fórmula reduzida, substituímos a parte multiplicadora de  $X_j$  pela função  $f(Y_i)$ . Isso quer dizer que  $f(Y_i) = \frac{3}{4} Y_i^{11} + \frac{25}{4} Y_i^9 - \frac{14}{3} Y_i^7 - \frac{47}{4} Y_i^5 + \frac{29}{4} Y_i^3$ . Assim, reescrevemos (3.14) para obter a forma compacta (3.15):

$$\frac{\partial \mathcal{J}(\mathbf{W})}{\partial W_{ji}} \approx -(\mathbf{W}^{-T})_{ji} + f(Y_i) X_j, \quad (3.15)$$

De posse da derivada parcial com relação a cada elemento de  $\mathbf{W}$ , podemos escrever um algoritmo adaptativo recursivo de descida do gradiente, individualmente para cada componente, assim como na Equação (2.23):

$$W_{ji}(t) = W_{ji}(t-1) + \mu \{ (\mathbf{W}^{-T})_{ji} - f(Y_i) X_j \}. \quad (3.16)$$

Podemos, também, adaptar simultaneamente todos os elementos de  $\mathbf{W}$  ao escrever a equação anterior na forma matricial a seguir:

$$\mathbf{W}(t) = \mathbf{W}(t-1) + \mu \{ \mathbf{W}^{-T} - \mathbf{f}(\mathbf{Y}) \mathbf{X}^T \}, \quad (3.17)$$

em que  $\mathbf{f}(\mathbf{Y}) = [f(Y_1), f(Y_2), \dots, f(Y_N)]^T$ . Ao analisarmos esta forma, vemos que é possível simplificar ainda mais a equação substituindo  $\mathbf{X}^T$  por  $\mathbf{Y}^T \mathbf{W}^{-T}$  e obtendo uma equação com apenas duas variáveis  $\mathbf{Y}$  e  $\mathbf{W}$  ao passo que (3.17) possui ainda  $\mathbf{X}$ . Logo,

$$\mathbf{W}(t) = \mathbf{W}(t-1) + \mu \{ \mathbf{I} - \mathbf{f}(\mathbf{Y}) \mathbf{Y}^T \} \mathbf{W}^{-T}. \quad (3.18)$$

Como visto na seção 2.4.3, este tipo de problema apresenta uma estrutura Riemanniana e podemos gerar o algoritmo de descida do gradiente natural ajustando a direção de descida do gradiente. Este ajuste é feito multiplicando o gradiente por  $\mathbf{W}^T \mathbf{W}$ . Assim, obtemos uma nova regra para atualização recursiva do sistema:

$$\mathbf{W}(t) = \mathbf{W}(t-1) + \mu [\mathbf{I} - \mathbf{f}(\mathbf{Y}) \mathbf{Y}^T] \mathbf{W}, \quad (3.19)$$

onde  $\mu$  é o tamanho do passo. Podemos, por fim, adaptar o algoritmo para o domínio dos complexos substituindo o transposto pelo conjugado complexo transposto, ou seja, pelo hermitiano que é definido pela operação  $(\cdot)^H$ :

$$\mathbf{W}(t) = \mathbf{W}(t-1) + \mu[\mathbf{I} - \mathbf{f}(\mathbf{Y})\mathbf{Y}^H]\mathbf{W}. \quad (3.20)$$

Em algoritmos práticos é comum observar a presença de um operador média no tempo representado por  $\langle \cdot \rangle_t$  dos termos  $\mathbf{f}(\mathbf{Y})\mathbf{Y}$ . Esta média é importante para evitar fortes oscilações durante a atualização como indicado por Sawada [24]. Assim a regra para adaptação de  $\mathbf{W}$  passa a ser:

$$\mathbf{W}(t) = \mathbf{W}(t-1) + \mu[\mathbf{I} - \langle \mathbf{f}(\mathbf{Y})\mathbf{Y}^H \rangle_t]\mathbf{W}. \quad (3.21)$$

Apesar de  $\mathbf{f}(\cdot)$ , em teoria, funcionar bem com as fontes em que suas distribuições podem ser aproximadas pela expansão de Gram-Charlier truncada, outras funções de ativação devem funcionar. Para uma adaptação adequada para números complexos, que é o caso no domínio da frequência, Sawada [24] sugere uma função de ativação em coordenadas polares:

$$f(Y_i) = \tanh(|Y_i|)e^{j\arg(Y_i)}. \quad (3.22)$$

Inicialmente, Smaragdis [28] sugeriu  $f(Y_i) = \tanh(\text{Re}\{Y_i\}) + j \cdot \tanh(\text{Im}\{Y_i\})$  que está notadamente no formato de coordenadas cartesianas, mas foi verificado posteriormente por Sawada [24] que (3.22) suaviza a atualização de  $W_{ji}$  e inclusive facilita a convergência mesmo em alguns casos extremos. Estes casos geralmente ocorrem quando  $W_{ji}$  se encontra próximo aos eixos (real e imaginário) do plano complexo.

Infelizmente, após execução do algoritmo acima, as ambiguidades de escalonamento e permutação, tais como vistos na Seção 2.1, típicos de algoritmos ICA, ainda ocorrem. Isso significa que

$$\mathbf{W} \leftarrow \mathbf{\Lambda}\mathbf{P}\mathbf{W}, \quad (3.23)$$

em que o símbolo  $\leftarrow$  indica que a matriz à esquerda assume o valor à sua direita. A matriz  $\mathbf{\Lambda}$  é uma matriz diagonal e  $\mathbf{P}$  uma matriz de permutação, ou seja,  $\mathbf{W}$  ainda é tido como um resultado válido de ICA mesmo após as multiplicações por  $\mathbf{\Lambda}$  e  $\mathbf{P}$ . O fato do algoritmo ICA não preservar a energia do sinal original e, também, atuar de maneira independente em cada frequência, significa que a matriz  $\mathbf{W}$  pode escalar os sinais de forma diferente em cada frequência. Isso acarretaria em alterações no espectro do sinal. A matriz  $\mathbf{P}$  causa permutações independentes em cada frequência, o que leva ao desalinhamento das frequências no que diz respeito às fontes sonoras

em cada saída. Isso significa que poderemos observar uma fonte diferente em cada frequência do sinal de saída, ou seja, embora os sinais de cada frequência estejam separados, eles ainda estarão embaralhados ao longo do espectro. Uma ilustração do problema da permutação é vista na Figura 3.2.

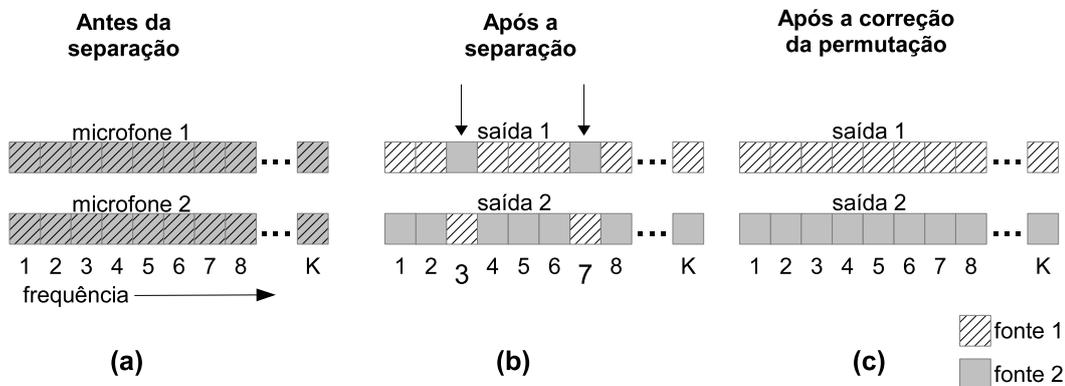


Figura 3.2: Representação da transformada discreta de Fourier (DFT) de tamanho  $K$  para 2 fontes e 2 microfones. (a) Antes da separação as duas fontes estão presentes em ambos os microfones. (b) Frequências desalinhadas após separação. (c) Frequências alinhadas após correção da permutação.

Para este trabalho o escalonamento é solucionado utilizando o Princípio da Distorção Mínima (MDP) [22]. Este princípio será abordado na subseção seguinte. Dois métodos para correção do problema da permutação são exibidos nas Seções 3.1.4 e 3.1.5.

### 3.1.3 O Princípio da Distorção Mínima (MDP)

Como visto ao fim da subseção anterior, a matriz  $\mathbf{W}(f)$  pode causar distorções em  $Y_i(f)$ . A matriz  $\mathbf{\Lambda}(f)$  na Equação (3.23) escala as linhas da matriz  $\mathbf{W}(f)$ , mas a mantém como uma solução válida de ICA. Portanto, podemos ajustar a matriz  $\mathbf{\Lambda}(f)$  de forma a manter a alteração de  $Y_i(f)$  em níveis satisfatórios.

O sinal emitido por uma fonte sonora, seja um alto falante ou um locutor é, na prática, alterado pelo ambiente acústico no qual ele está inserido, antes de chegar ao ouvinte ou aos microfones (Figura 3.3). Esta alteração ocorre de acordo com a geometria da sala como visto na Seção 2.4.1. No dia a dia, somos capazes de receber um sinal acústico e interpretá-los, na maioria dos casos, sem grandes problemas mesmo com a alteração imposta pela ambiente. Por isso, consideraremos que  $y_i(t) = s_i(t) * h_{ii}(t)$  é uma estimativa satisfatória de  $s_i(t)$ , ou seja, o sinal fonte filtrado pelo ambiente é considerado aceitável. O símbolo  $*$  representa o operador da convolução. Note que

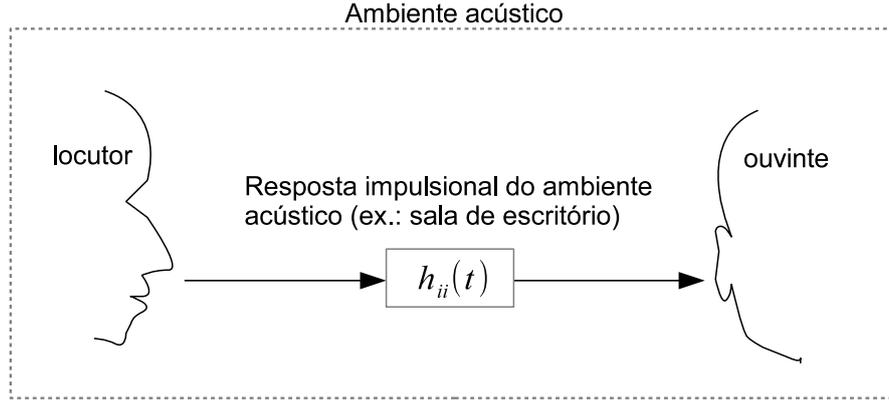


Figura 3.3: Ilustração de um ambiente acústico com um locutor e um ouvinte. O sinal é filtrado entre o locutor e o ouvinte.

mesmo aceitando a alteração provocada pelo ambiente, a meta principal de separar as fontes é preservada.

Dizer que  $y_i(t) = s_i(t)*h_{ii}(t)$  é uma resposta aceitável, significa dizer que é suficiente que os filtros  $w_{ji}(t)$  anulem as respostas  $h_{ij}(t)$  quando  $i \neq j$  e as mantenham intactas para  $i = j$ . Podemos escrever o equivalente da expressão  $y_i(t) = s_i(t)*h_{ii}(t)$  no domínio da frequência como:

$$Y_i(f) = S_i(f)H_{ii}(f). \quad (3.24)$$

Assim,  $H_{ii}(f)$  são os elementos da diagonal da matriz  $\mathbf{H}(f)$ . Se conseguirmos anular os sinais interferentes,teremos a seguinte igualdade:

$$\mathbf{WH} = \mathbf{D}, \quad (3.25)$$

em que  $\mathbf{D}(f)$  é uma matriz diagonal e voltou-se a omitir  $f$  para a compactação da escrita. Como  $\mathbf{\Lambda}$  não altera a validade da solução para ICA, podemos utilizá-lo e ainda assim manter o resultado da equação (3.25):

$$\mathbf{\Lambda WH} = \mathbf{D}, \quad (3.26)$$

Podemos dizer também que:

$$\mathbf{\Lambda WH} = \text{diag}\{\mathbf{H}\}, \quad (3.27)$$

onde o operador  $\text{diag}\{\cdot\}$  torna zero o valor de todos os elementos fora da diagonal de uma matriz. Isolando  $\mathbf{H}$  na Equação (3.25), observamos que  $\mathbf{H} = \mathbf{W}^{-1}\mathbf{D}$ . Então, podemos substituir  $\mathbf{H}$  em (3.27) e verificar que

$$\mathbf{\Lambda} = \text{diag}\{\mathbf{W}^{-1}\}. \quad (3.28)$$

O que implica que podemos ajustar  $\mathbf{W}$  para que não haja distorção do sinal, além daquela produzida pelo ambiente, da seguinte forma:

$$\mathbf{W} \leftarrow \text{diag}\{\mathbf{W}^{-1}\}\mathbf{W}. \quad (3.29)$$

O problema de redução dos efeitos de  $h_{ii}(t)$  é conhecido como deconvolução cega ou deverberação cega e é abordado em outros trabalhos, por exemplo [16].

### 3.1.4 O Método da Direção de Chegada

A matriz  $\mathbf{W}$ , solução ao problema ICA, não resolve o problema de forma completa, pois sofre do problema causado pela permutação. Em outras palavras, não se consegue saber à qual fonte uma determinada estimativa está relacionada. Em muitas aplicações diretas do ICA isto não é um grande problema, porém, para a separação de sinais banda larga (como é o caso da voz) no domínio da frequência é um problema crucial. O algoritmo ICA mostrado anteriormente calcula uma matriz  $\mathbf{W}$  individualmente para cada frequência. Logo, os sinais podem ser permutados de maneira diferente em cada uma delas como ilustrado pela Figura 3.2. Sawada [25] mostra dois diferentes métodos para solucionar este problema e também uma forma de tomar vantagem da melhor característica de cada um deles.

O método da direção de chegada (DOA) tenta estimar as direções de chegada das fontes em cada uma das individuais frequências a fim de identificar as fontes e efetuar o reposicionamento das frequências permutadas. Considerando que as fontes estão estáticas em cada bloco de processamento, é possível verificar se as frequências de um conjunto de filtros correspondentes a uma mesma fonte chegam todas segundo uma mesma DOA. Se elas chegam de outras direções, então devem ser permutadas. De acordo com van Veen [29] e respeitando o espaçamento dos microfones para que não haja *aliasing*, pode-se escrever uma aproximação da função de transferência aproximada (vide Equação (2.4))

$$H_{jk}(f) = e^{j2\pi f c^{-1} d_j \sin \theta_k}, \quad (3.30)$$

onde  $c$  é a velocidade do som,  $d_j$  é a posição de cada microfone  $j$  de um arranjo linear de microfones e cada fonte emite uma onda que chega ao arranjo vindo de um ângulo  $\theta$ . Essa função descreve um feixe, ajustando a fase, em que cada sinal deve chegar em cada microfone. Removendo o índice  $\theta$  e considerando-o como uma variável, é possível calcular a resposta do sistema com relação à direção  $\theta$ . A esta resposta do sistema se dá o nome de padrão de direcionamento:

$$U_k(f, \theta) = \sum_{j=1}^N H_{jk}(f, \theta) W_{kj}(f). \quad (3.31)$$

No caso de duas fontes e dois sensores e se considerarmos que o primeiro sensor está na origem do arranjo ( $d_0$ ), pode-se expandir a equação 3.31 e reescrevê-la como

$$U_k(f, \theta) = W_{k1}(f) + W_{k2}(f)e^{j2\pi f c^{-1} d_1 \sin \theta}. \quad (3.32)$$

Analisando o padrão de direcionamento (Figura 3.4), Sawada [25] e outros mostram que os padrões tendem a apresentar valores nulos na direção de chegada das fontes interferentes. Essa direção  $\theta_{\text{null}}$  pode ser identificada na figura 3.4 feita para o caso com duas fontes. Porém, buscar os valores nulos em todo o padrão não é uma tarefa computacionalmente eficiente. A partir do próximo parágrafo, será descrita uma forma bem menos custosa de se encontrar  $\theta_{\text{null}}$  sugerida por [25]. Ao analisar geometricamente

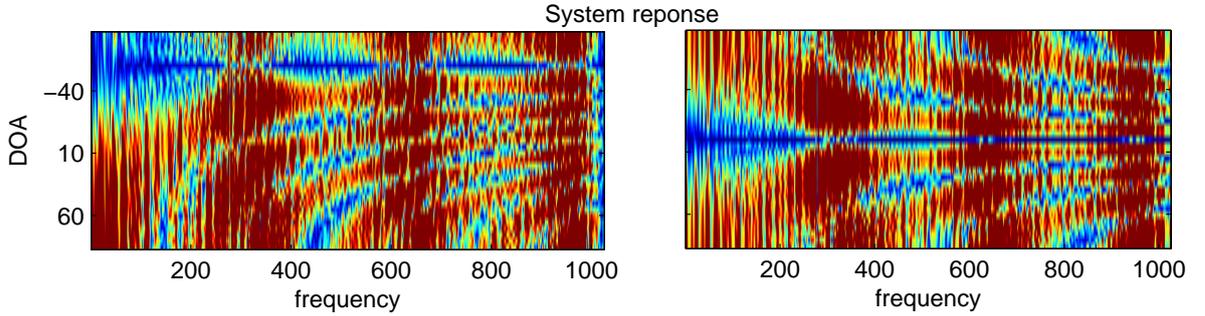


Figura 3.4: Resposta do sistema para duas fontes, uma em  $-60^\circ$  e outra em  $0^\circ$ . O sistema de separação tenta a formar nulos na direção dos sinais interferentes.

a Equação (3.32), vê-se que o segundo termo do lado direito nada mais é do que um vetor  $W_{k2}(f)$  no plano complexo rotacionado em  $\theta$  graus pela expressão  $e^{j2\pi f c^{-1} d_2 \sin \theta}$ . Conhecendo as fases de  $W_{k1}$  e  $W_{k2}$ , determina-se o ângulo  $\theta_{\text{null}}$  encontrando valor de  $\theta$  correspondente ao ponto em que a expressão tem o seu valor mínimo. É fácil perceber que a função terá seu mínimo quando  $W_{k1}$  e  $W_{k2}$  estiverem em sentidos opostos. Logo,

$$\theta_{\text{null}} = \sin^{-1} \frac{\arg W_{k1} + \arg W_{k2} \pm \pi}{2\pi f c^{-1} d_1}. \quad (3.33)$$

Uma vez que as DOAs são estimadas, basta verificar se cada frequência do padrão de direções chega ao arranjo vindo da mesma direção. Caso contrário, eles provavelmente foram permutados pelo algoritmo ICA e devem ser permutados novamente para a posição correta. Entretanto, este método possui uma baixa resolução espacial para baixas frequências e a DOA não pode ser estimado adequadamente para tais frequências. Na próxima subseção, veremos como superar este problema.

### 3.1.5 O Método das Correlações Vizinhas

Para baixas frequências, a resolução espacial do método da seção anterior é reduzida. Para complementar o método da DOA, Sawada [25] também sugere um método baseado

na correlação de frequências vizinhas. Este método analisa como  $|Y_k(f, m)|$  se comporta no tempo para uma determinada frequência e verifica que  $|Y_k(f, m)|$  possui uma alta correlação para as frequências próximas e pertencentes a uma mesma fonte. Como  $m$  é o índice de um quadro(ou bloco) no tempo, é ele quem vai ser variado a fim de se analisar a evolução temporal de  $|Y_k(f, m)|$ . Como somente a cada novo quadro teremos um novo valor na frequência  $f$ , o que observaremos em um gráfico de  $|Y_k(f, m)| \times m$  é apenas um envelope de  $|Y_k(f, t)|$ . Tal gráfico é visto na Figura 3.5.

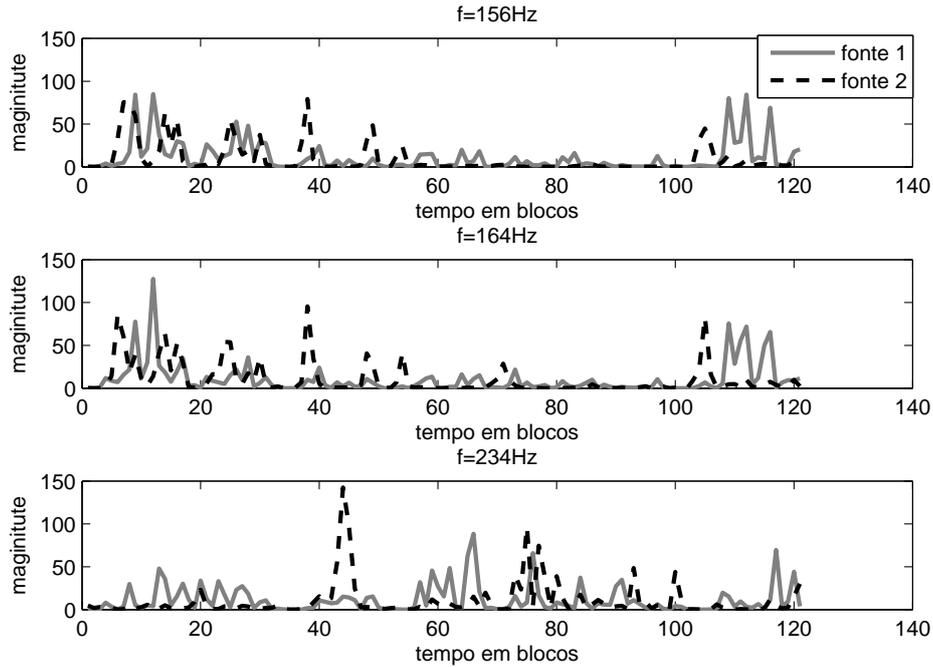


Figura 3.5: Evolução de  $|Y_k(f, m)|$  ao longo do tempo. As duas primeiras linhas representam a evolução de  $|Y_k(f, m)|$  ao longo dos blocos no tempo  $m$  de frequências  $f$  próximas. A última linha representa a evolução de  $|Y_k(f, m)|$  ao longo de  $m$  em uma frequência  $f$  distante das duas anteriores.

Analisando os padrões de cada frequência ao longo do tempo, observa-se esta alta correlação. Na Figura 3.5 vemos o progresso do envelope de duas fontes ao longo do tempo em 3 frequências, duas vizinhas e uma distante. Entre duas fontes diferentes, supondo que as fontes são independentes, a correlação deve ser sempre muito baixa. Podemos analisar isto na Figura 3.5 verificando que os picos das linhas tracejadas não coincidem com os picos das linhas sólidas mesmo que em uma mesma frequência. Para frequências vizinhas (duas primeiros gráficos da Figura 3.5), os picos relativos a uma mesma fonte coincidem e isso significa que sua correlação entre  $|Y_1(f, m)|$  e  $|Y_2(f, m)|$  será alta. Já se compararmos com uma frequência distante, como a da última linha da figura, os picos pertencentes a uma mesma fonte não coincidem mais, ou seja, sua

correlação deverá ser baixa para frequências distantes. Por isso, utilizaremos apenas as correlações das frequências vizinhas para detectar a permutação.

Quando o sinal de frequências vizinhas está na posição correta, a correlação deve ser alta, significando que as frequências estão alinhadas. Já, quando o sinal de frequências vizinhas está na posição errada, esta correlação deve tornar-se muito baixa por se tratarem de fontes diferentes, possibilitando a detecção da permutação. A correlação é definida como

$$cor(x, y) = \frac{\langle x \cdot y \rangle - \langle x \rangle \cdot \langle y \rangle}{\sigma_x \sigma_y}, \quad (3.34)$$

com  $\langle \cdot \rangle$  sendo o operador média e  $\sigma$  o desvio padrão. Todavia, este método pode criar um erro em cascata se ocorrer uma falha na permutação. Como o método depende de frequências vizinhas, caso ocorra um erro na permutação de uma frequência, a frequência vizinha será prejudicada no cálculo da correlação. Quando for calculada a correlação da próxima frequência, a anterior estará na posição errada. Isso causará que a correlação entre as duas seja baixa e ela será permutada de forma errada. Esse processo se repetirá criando o efeito cascata. Por este motivo usaremos este método apenas para os casos que o método do DOA não conseguiu determinar a ordem da permutação.

### 3.1.6 Utilizando Informações Prévias de Direção de Chegada

Se informações sobre o ambiente da mistura sonora estiverem disponíveis, é possível utilizá-las para inicializar o algoritmo descrito na Seção 3.1.2 anteriormente com o objetivo de melhorar seu desempenho. Parra [23] mostra que a utilização de DOA foi eficiente para inicialização de um algoritmo de BSS no domínio do tempo. Neste trabalho, a inicialização a partir de informações de DOA será utilizada para o algoritmo de BSS no domínio da frequência visto anteriormente neste Capítulo e, diferentemente de Parra [23], será aplicada a inicialização com foco nas configurações do ambiente não favoráveis à separação. Ainda, alteraremos o suavemente o posicionamento da DOA durante a criação dos filtros para os ângulos de maior energia espacial como será visto no Capítulo 4. Para efetuar a inicialização dos filtros, adicionaremos dois blocos ao processo de separação como mostrado na Figura 3.6. Primeiramente, é feita uma estimação de DOA. Em seguida, é calculada a inicialização dos filtros de separação utilizando os ângulos estimados. Então, o algoritmo adaptativo do gradiente natural é executado. Por fim, cada frequência é escalonada e verificada quanto à permutação.

A intenção é, de forma simples, calcular um valor inicial para os elementos dos filtros

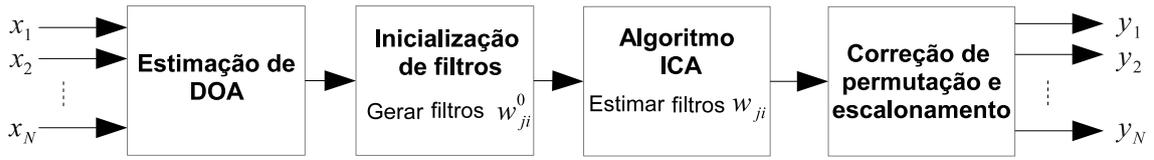


Figura 3.6: Diagrama de blocos do método proposto, iniciando com a fase de estimação de DOA para inicializar os filtros.

$w_{ji}(t)$  de forma que o maior esforço computacional continue com o algoritmo de BSS. Chamaremos este filtro inicial de  $w_{ji}^0$ . Neste trabalho consideramos a DOA como sendo já conhecido, porém, ela poderia ter sido estimada por algoritmos de estimação de DOA como é o caso do SRP-PHAT [12].

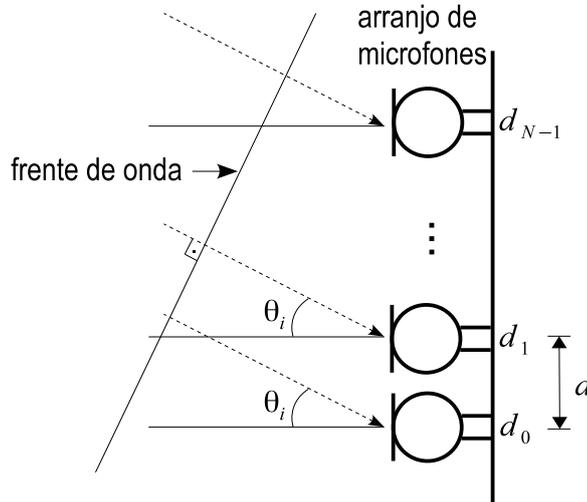


Figura 3.7: Arranjo de microfones e ângulos de chegada.

Um ponto de partida para gerar filtros  $w_{ji}^0$  simples é imaginar que as respostas  $h_{ij}(t)$  do ambiente também são simples. A resposta simplificada em questão corresponderia a apenas um atraso na chegada do sinal em um microfone com relação à outro microfone, ou seja, não são consideradas reflexões nem espalhamento do sinal no tempo. Em outras palavras, apenas o caminho direto entre a fonte do sinal e o microfone está sendo considerado. Então, podemos imaginar uma resposta impulsional do ambiente no formato de um impulso de Dirac deslocado no tempo  $\delta(t - \tau)$ , onde  $\tau$  é o atraso em que um sinal da fonte chega a um microfone com relação a chegada deste mesmo sinal ao microfone na origem  $d_0$ . Este atraso pode ser calculado por inspeção geométrica da Figura 3.7. Uma frente de onda planar emitida por uma fonte  $s_i$  vinda de uma direção

$\theta_i$  chega ao microfone  $j$  com um atraso

$$\tau_{ij} = c^{-1}d_{j-1} \sin \theta_i, \quad (3.35)$$

onde  $c$  é a velocidade do som. Note a semelhança com a Equação (2.1). Tomando o primeiro microfone como referencia e usando os atrasos calculados por meio da equação anterior, podemos gerar  $h'_{ij}(t)$  como uma forma simplificada de  $h_{ij}(t)$  da seguinte maneira:

$$h'_{ij}(t) = \begin{cases} \delta(t) & \text{caso } j = 1, \\ \delta(t + \tau_{ij}) & \text{caso } j \neq 1. \end{cases} \quad (3.36)$$

Se quisermos remover um dos sinais de uma das saídas do sistema, podemos tentar formar uma resposta nula do sistema para essa fonte indesejada, ou seja,  $h'_{ij}(t) * w_{ji}^0(t) = 0$ . Com isso em mente, os filtros  $w_{ji}^0(t)$  devem ser similares aos filtros  $h'_{ij}(t)$  para que criemos impulsos de Dirac com apenas um atraso de maneira semelhante a Equação (3.36). Diferentemente do passo anterior onde o microfone na origem ( $d_0$ ) era a referência, agora tomamos para cada saída um microfone para referência do atraso. Mais especificamente, cada saída utilizará seu microfone correspondente como referência, por exemplo, o microfone 1 é a referencia da saída 1 e o microfone 2, a referência da saída 2 e assim por diante. Desta maneira, podemos calcular os atrasos

$$\tau_{ij}^0 = c^{-1}(d_{(j-1)} - d_{(i-1)}) \sin \theta_i. \quad (3.37)$$

A diferença  $d_{(j-1)} - d_{(i-1)}$  marca a referência explicada anteriormente. Com os atrasos  $\tau_{ij}^0$  podemos gerar os filtros  $w_{ji}^0(t)$  da seguinte forma:

$$w_{ji}^0(t) = \begin{cases} \delta(t) & \text{caso } i = j, \\ -\delta(t + \tau_{ij}^0) & \text{caso } i \neq j. \end{cases} \quad (3.38)$$

A Figura 3.8 ilustra como que os filtros  $h'_{ij}(t)$  e  $w_{ji}^0(t)$  devem se parecer para duas fontes em 20 e 70°. A primeira coluna apresenta os filtros  $h'_{ij}(t)$  e a segunda os filtros  $w_{ji}^0$ . O objetivo é tornar a resposta impulsional nula gerando os filtros  $w_{21}^0(t)$  e  $w_{12}^0(t)$ . Verificando a Figura 3.8 podemos traçar os caminhos diretos desde a entrada 1 até a saída 1 e da entrada 2 à saída 2 para as usarmos como nossas referências. Os atrasos são calculados de forma que na saída 1 o filtro  $h'_{22}(t)$  seja cancelado e na saída 2 o filtro  $h'_{11}(t)$  seja cancelado. As equações 3.35 e 3.37 dão os trasos para que este cancelamento aconteça:  $\tau_{12} = 160\mu s$ ,  $\tau_{22} = 442\mu s$ ,  $\tau_{21}^0 = -442\mu s$  e  $\tau_{12}^0 = 160\mu s$ . Calculando os filtros de acordo com a Equação (3.38) e, então, convoluindo e somando com os filtros  $w_{ji}^0(t)$ , deve-se observar uma resposta nula para a fonte interferente em cada saída. Note que o filtro  $w_{12}^0(t)$  é não causal. Um filtro não causal se faz necessário quando as duas fontes estão no mesmo semiplano. De forma mais clara, no caso  $2 \times 2$ , quando a frente de

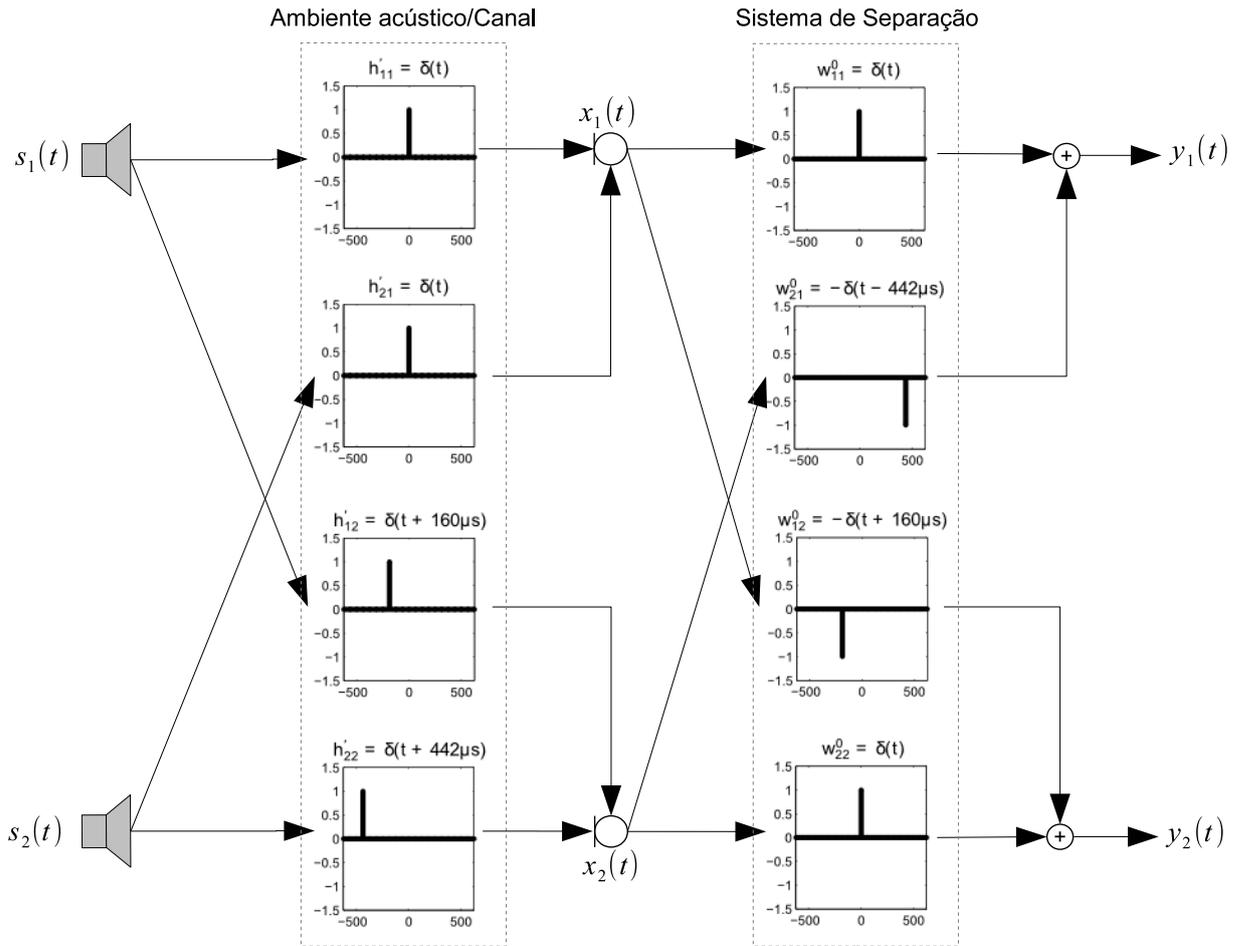


Figura 3.8: Filtros  $h'_{ij}(t)$  e  $w^0_{ji}(t)$  para duas fontes: uma em  $20^\circ$  e outra em  $70^\circ$

onda da fonte 1 atinge o microfone 2 antes anteriormente ao microfone 1, ou a frente de onda da fonte 2 atinge primeiro o microfone 1 e posteriormente o 2.

### 3.1.7 Utilizando Filtros de Decorrelação

Uma outra forma de procurar melhorar a qualidade de separação nos casos críticos seria o branqueamento do sinal. Pelo branqueamento, eliminar-se-ia a correlação temporal dos sinais, deixando apenas a correlação espacial nos sinais da mistura, que é o foco do algoritmo. A correlação temporal pode interferir na procura da separação espacial e, por isso, Kokkinakis [21] efetua a decorrelação temporal total utilizando algoritmos de predição linear. Neste trabalho é utilizada uma filtragem simples por meio de um filtro passa alta de primeira ordem (apenas um atraso) cuja resposta em frequência é vista na Figura 3.9.

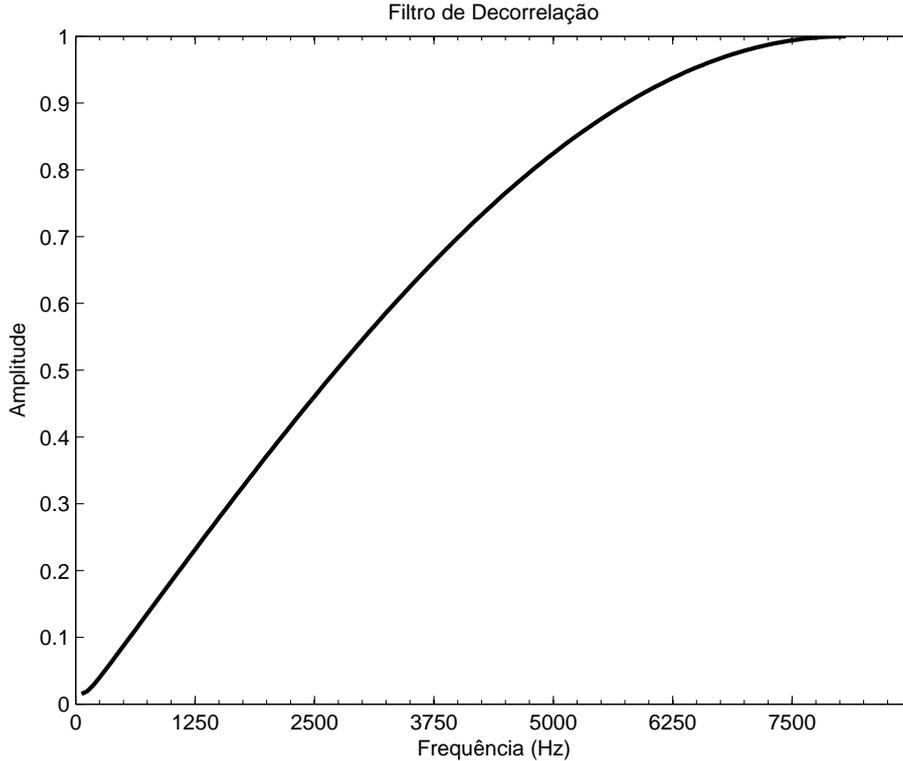


Figura 3.9: Filtro passa alta de primeira ordem.

A resposta deste filtro no domínio complexo é do tipo  $H(z) = \kappa_0 + \kappa_1 z^{-1}$ , em que  $z^{-1}$  pode ser entendido como um atraso de uma amostra no domínio do tempo. Os coeficientes do filtro da Figura 3.9 foram escolhidos de forma a atenuar as faixas de frequência onde a voz está mais concentrada. Mais especificamente, foram escolhidos os valores  $\kappa_0 = 1/1,95$  e  $\kappa_1 = -0,97/1,95$ . Como os sinais de voz são concentrados nas baixas frequências, esperamos equalizar o sinal homogeneizando seu espectro de frequências. Aplicando o filtro aos sinais de entrada  $x_j(t)$ , inserimos o sinal decorrelatado na entrada do sistema e esperamos que o filtro **W** resultante contenha informações para se efetuar a decorrelação espacial, ou seja, a separação das misturas. O efeito deste filtro em um sinal de voz é visto na figura 3.10.

### 3.2 SEPARAÇÃO UTILIZANDO O TRINICON

Dentre os algoritmos de separação, o que apresenta melhores resultados é o TRINICON [8]. O TRINICON é uma ferramenta de implementação e compreensão complexa, porém robusta ao ponto de podermos explorá-la em outros setores da separação cega além da voz humana. Pesquisadores da fauna do cerrado da Universidade de Brasília têm mostrado interesse na utilização de cantos de aves, a fim de identificar a diversidade

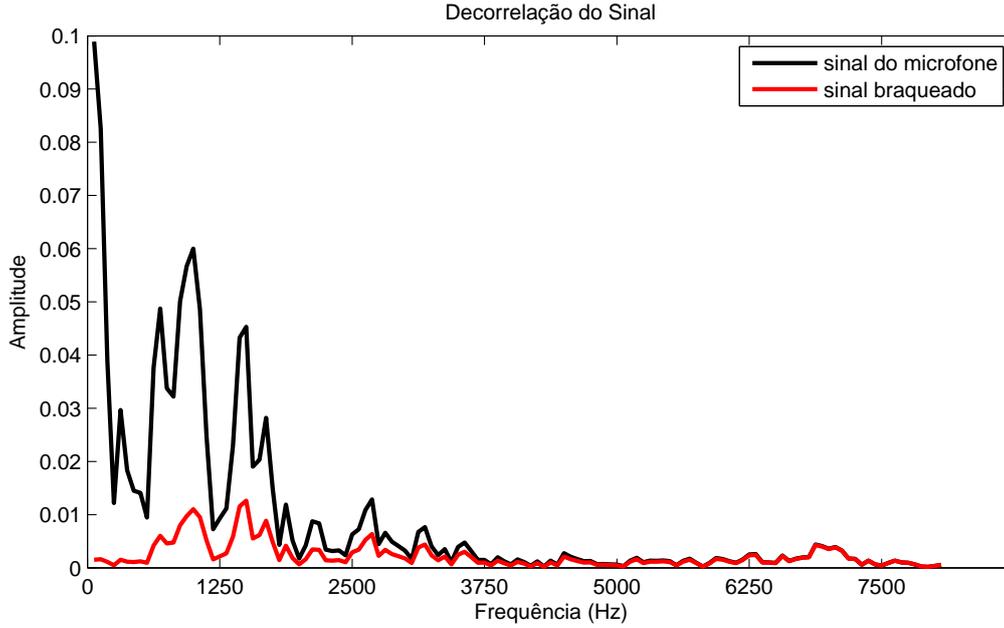


Figura 3.10: Branqueamento do sinal.

silvestre. Cantos de pássaros gravados por monitores acústicos mostram por muitas vezes misturas de cantos de pássaros de várias espécies. O TRINICON foi utilizado neste trabalho com o objetivo de separar cantos de pássaros para que, uma vez separados, eles possam ser utilizados como de entrada para ferramentas de identificação de cantos de pássaros em trabalhos futuros. A seguir revisaremos de maneira sucinta o funcionamento deste algoritmo.

O TRINICON parte da criação de uma função de custo também baseada na diferença de Kullback-Leibler como vimos na equação (3.3). Essa função de custo

$$\mathcal{J}(m, \mathbf{W}) = - \sum_{i=0}^{\infty} \beta(i, m) \frac{1}{N} \sum_{j=iL}^{iL+N-1} \{ \log(\hat{p}_{s,PD}(\mathbf{y}(j))) - \log(\hat{p}_{y,PD}(\mathbf{y}(j))) \}, \quad (3.39)$$

é calculada em cada bloco  $m$  e se aproveita da não estacionaridade, não branqueamento e não gaussianidade do sinal. A função  $\beta(i, m)$  é uma função janela que prioriza novos blocos, mas ainda se utiliza de blocos antigos. As funções  $\hat{p}_{s,PD}(\cdot)$  e  $\hat{p}_{y,PD}(\cdot)$  são as estimativas das FDPs da fonte e da saída, respectivamente. Diferentemente do método no domínio da frequência, estas estimativas são feitas por meio de *Spherically Invariant Random Variables* (SIRPs) [7]. O subscrito  $PD$  diz respeito ao número de fontes e atrasos com que a FDP será estimada. Este número de atrasos define o tamanho da matriz de correlação ou covariância como veremos adiante. A minimização da função de custo é feita pelo método do gradiente natural. Neste caso ele é levemente adaptado

resultando em

$$\tilde{\nabla} \mathcal{J}(m, \mathbf{W}) = \mathbf{W} \mathbf{W}^T \frac{\partial \mathcal{J}(m, \mathbf{W})}{\partial \mathbf{W}}. \quad (3.40)$$

Calculando o gradiente da Equação 3.39 e realizando a adaptação da direção de descida por meio do gradiente natural como em 3.40 é possível chegar à seguinte regra de adaptação:

$$\tilde{\nabla} \mathcal{J}(m, \mathbf{W}) = \frac{1}{N} \sum_{i=0}^{\infty} \beta(i, m) \sum_{j=iL}^{iL+N-1} \mathbf{W}(m) \mathbf{y}(j) \left[ \Phi_{s,PD}^T(\mathbf{y}(j)) - \Phi_{y,PD}^T(\mathbf{y}(j)) \right], \quad (3.41)$$

As funções  $\Phi_{s,PD}(\mathbf{y}(j))$  e  $\Phi_{y,PD}(\mathbf{y}(j))$  são chamadas funções de pontuação e tem a forma:

$$\Phi_{s,PD}(\mathbf{y}(j)) = -\frac{\partial \log \hat{p}_{s,PD}(\mathbf{y}(j))}{\partial \mathbf{y}(j)}, \quad (3.42)$$

$$\Phi_{y,PD}(\mathbf{y}(j)) = -\frac{\partial \log \hat{p}_{y,PD}(\mathbf{y}(j))}{\partial \mathbf{y}(j)}. \quad (3.43)$$

Quando se utilizam apenas matrizes de correlação, ou seja, apenas estatística de segunda ordem para estimar a FDP, obtém-se a simplificação da Equação 3.41:

$$\tilde{\nabla} \mathcal{J}(m, \mathbf{W}) = 2 \sum_{i=0}^{\infty} \beta(i, m) \mathbf{W}(i) \{ \hat{\mathbf{R}}_{\mathbf{y}\mathbf{y}} - \hat{\mathbf{R}}_{\mathbf{s}\mathbf{s}} \} \hat{\mathbf{R}}_{\mathbf{s}\mathbf{s}}^{-1}, \quad (3.44)$$

com  $\hat{\mathbf{R}}_{\cdot}$  representando uma estimativa da matriz de correlação. Utilizar este tipo de simplificação corresponde a utilizar funções gaussianas multivariáveis uma vez que necessitamos das distribuições em mais de uma dimensão. Isso resulta em matrizes de correlação de tamanho  $PD \times PD$ . A Figura 3.11 ilustra as matrizes de correlação. Cada

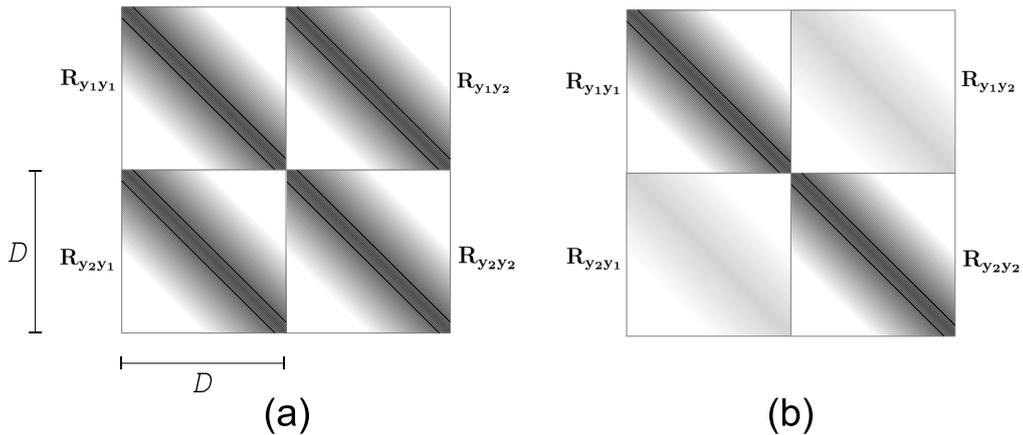


Figura 3.11: Matrizes de correlação para misturas com duas fontes. (a) Correlação antes da separação. (b) Correlação depois da separação. Algoritmo de separação anula as matrizes de correlação cruzada.

diagonal da matriz de correlação corresponde à correlação temporal em um atraso, em

que cada atraso corresponde a um período de amostragem  $T_s$ . Ao passo em que se move em cada diagonal no sentido contrario à diagonal principal se aumenta um atraso. Ao executar o algoritmo de separação, as matrizes de correlação cruzada tendem a ser anuladas, pois se as fontes agora estão separadas elas são independentes e, portanto, sua correlação é zero.

## 4 EXPERIMENTOS E RESULTADOS

Foram realizadas dois conjuntos de experimentos para este trabalho. Um com o algoritmo de BSS no domínio da frequência, revisado na Seção 3.1, e outro para o TRINICON, revisado na Seção 3.2. Para o BSS no domínio da frequência, foram realizados experimentos com foco na variação do posicionamento a fim de verificar o seu funcionamento em ângulos não favoráveis. Em seguida foram testados dois métodos de inicialização com o objetivo de melhorar a qualidade da separação para fontes nessas posições. Para o TRINICON, foram feitos testes com cantos de pássaros para testar a efetividade de algoritmos de BSS com este tipo de sinal.

### 4.1 SEPARAÇÃO DE SINAIS DE VOZ EM ÂNGULOS VARIÁVEIS

Este experimento tem por objetivo inicial analisar o funcionamento do algoritmo de BSS no domínio da frequência para diversas configurações das fontes no espaço. Antes disso, é preciso estimar alguns parâmetros para o bom funcionamento do algoritmo. Essa estimação é feita por meio de uma busca empírica por parâmetros que melhorem o resultado como menor esforço computacional possível, ou seja, é feita uma otimização. Para isso adicionamos um parâmetro de controle  $\eta$  à função não linear (3.22) conforme indicado por Sawada [24]:

$$f(Y_i) = \tanh(\eta|Y_i|)e^{j \arg(Y_i)}. \quad (4.1)$$

Este parâmetro também é conhecido como ganho da função não linear. A otimização levou em conta o tamanho do bloco, o número de iterações do algoritmo ICA, o tamanho do passo  $\mu$  e um parâmetro de controle  $\eta$  para a função não linear. Com os parâmetros estimados segundo a Tabela 4.1, simulações para vários ângulos foram executadas com o objetivo de verificar os ângulos críticos.

Foram criadas respostas impulsivas artificiais de primeira ordem de acordo com o Método das Imagens, isto é, só foram consideradas seis reflexões, uma para cada parede da sala com dimensões  $5,8 \times 5,8 \times 3,6$  m, coeficiente de reflexão  $\rho = 0.8$  e espaçamento entre microfones  $d = 16$  cm. A resposta impulsional da sala é uma resposta de primeira ordem, ou seja, (RIR) possui 7 picos: um relativo ao caminho direto (de visada) e os outros 6 relativos a uma reflexão em cada das paredes.

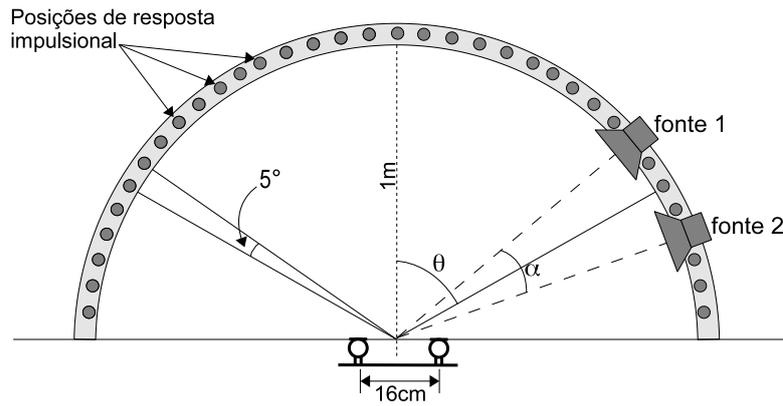


Figura 4.1: Configuração da sala.

Tabela 4.1: Parâmetros Obtidos

Parâmetro	Valor
Taxa de amostragem	$f_s = 16000 Hz$
Tamanho do quadro e da FFT	$N = 2048$
Tamanho do passo (m)	$\mu = 0.0002$
Número de iterações do ICA	100
Ganho da função não linear	$\eta = 100$

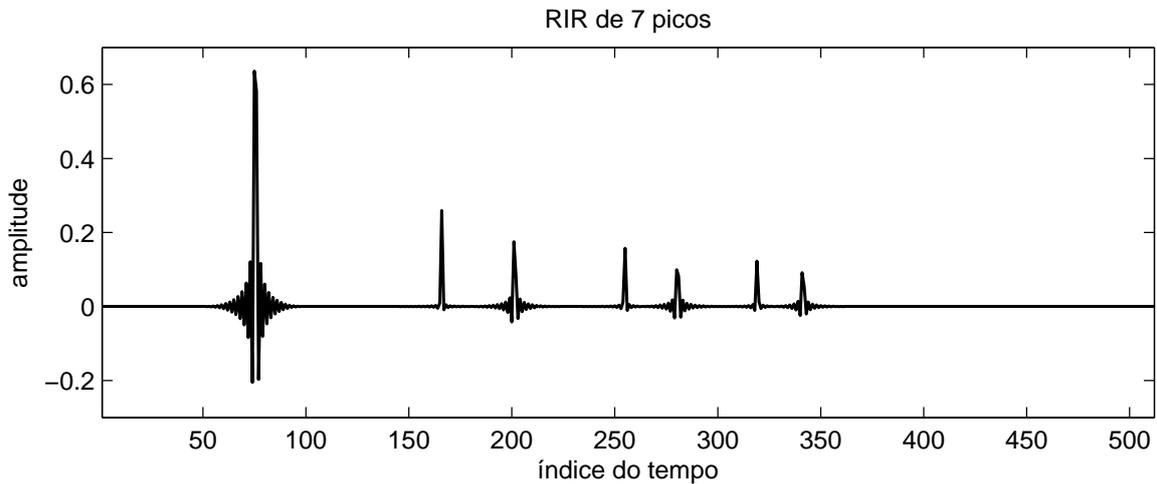


Figura 4.2: Resposta impulsional simulada. A resposta possui 7 picos. O primeiro da esquerda para a direita corresponde ao caminho de visada e os outros 6 a uma reflexão em cada uma das paredes. Frequência de amostragem utilizada é de 16 kHz.

Para avaliar o funcionamento do sistema é calculada a SIR em gravações de 40 segundos de duração com duas fontes ativas simultaneamente. Um esquemático da configuração do ambiente pode ser visto na Figura 4.1. Dois parâmetros são variados dentro do

cenário descrito anteriormente, o espaçamento entre as fontes  $\alpha$  e a média dos ângulos de incidência das duas fontes  $\theta$ . Inicialmente o espaçamento  $\alpha$  entre as fontes foi fixado em  $60^\circ$ . O ângulo  $\theta$  foi rotacionado em passos de 5 graus variando de  $-55^\circ$  à  $+55^\circ$ . Isso quer dizer que  $\theta \in \{-55^\circ, -50^\circ, -45^\circ, \dots, 45^\circ, 50^\circ, 55^\circ\}$ . O resultado é visto na Figura 4.3.

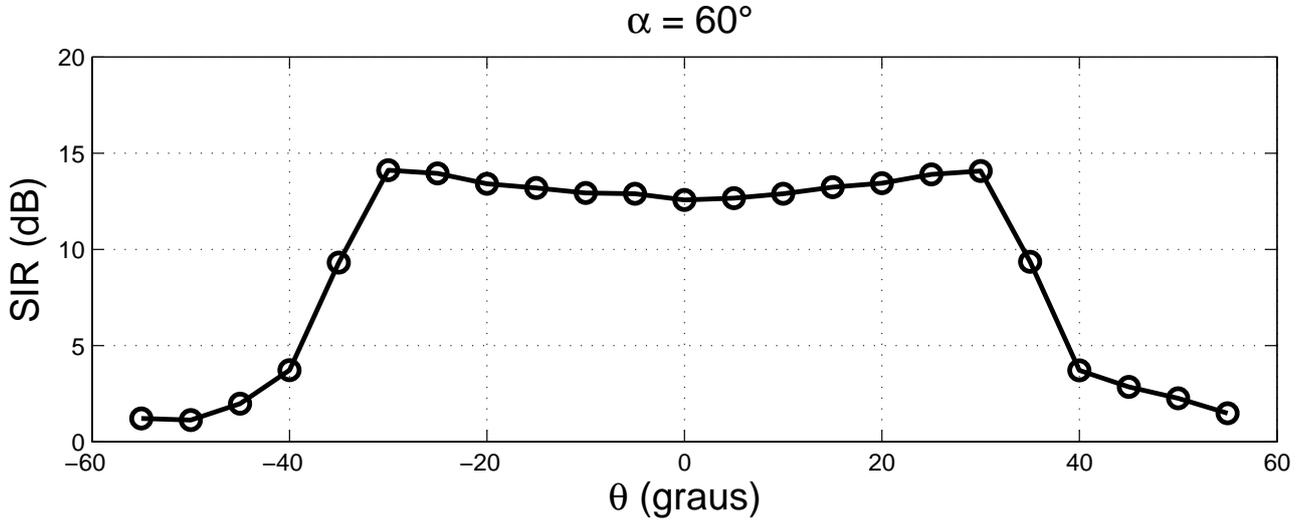


Figura 4.3: Resultados para os parâmetros sugeridos.

Para que possamos entender com clareza o gráfico da Figura 4.3, duas posições do gráfico são analisadas na Figura 4.4. Vemos o posicionamento das fontes na Figura 4.4(a) para o ponto onde  $\theta = 30^\circ$ , último antes do decaimento seguindo no sentido da esquerda para a direita, e na Figura 4.4(b) o ponto onde  $\theta = 40^\circ$ , ou seja, logo após o decaimento. Este ponto,  $\theta = 40^\circ$ , caracteriza a situação onde as fontes estão em um mesmo semiplano.

O primeiro evento a ser notado é a queda imediata de desempenho quando ambas as fontes estão no mesmo semiplano: esquerdo ou direito. Neste tipo de configuração, o algoritmo deve atuar com filtros não causais. Na Figura 4.3 isso acontece quando  $|\theta| > 30^\circ$ , ou seja para  $|\theta| > \alpha/2$ . O desempenho cai claramente em tais condições. O mesmo acontece para  $|\theta| > \alpha/2$  se alterarmos os valores de  $\theta$ . É possível observar a presença de um mínimo local para  $\theta = 0^\circ$

Em seguida, o mesmo experimento foi repetido alterando apenas o espaçamento  $\alpha$  entre as fontes. Este espaçamento foi variado de  $15^\circ$  a  $60^\circ$  em passos de  $5^\circ$ , ou seja,  $\alpha \in \{15^\circ, 20^\circ, 25^\circ, \dots, 50^\circ, 55^\circ, 60^\circ\}$ . O resultado geral pode ser visto na Figura 4.5.

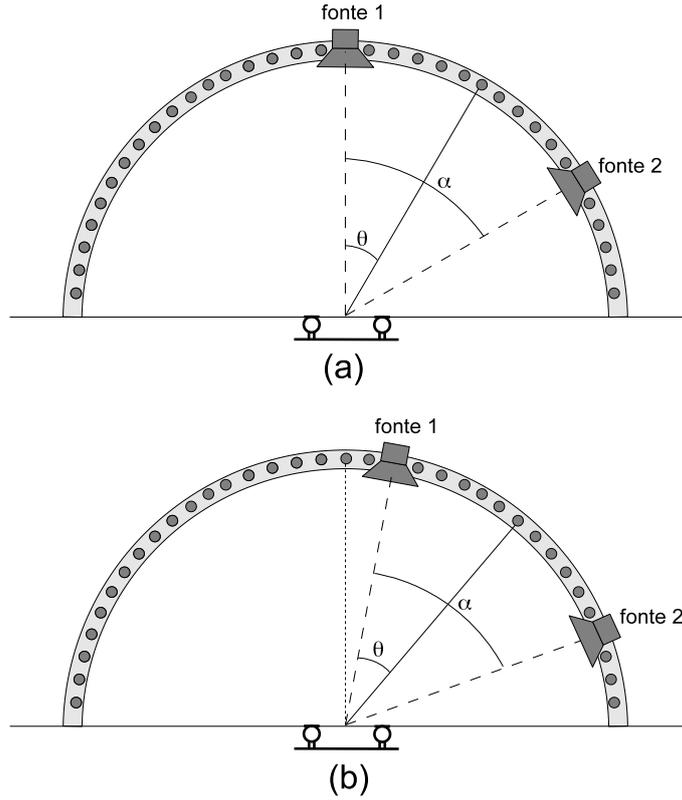


Figura 4.4: Cenário  $\theta = 30^\circ$  (a) e  $\theta = 40^\circ$  (b). Em ambos os casos  $\alpha = 60^\circ$ .

O mesmo problema encontrado na Figura 4.3 aparece para os diversos valores de  $\alpha$  na Figura 4.5. De forma a tentar suavizar o problema, o passo de inicialização com conhecimento prévio da DOA foi utilizado e o mesmo experimento realizado novamente. O resultado com a inicialização de DOA é mostrado na Figura 4.6. Um aumento da SIR pode ser observado em pontos não favoráveis à separação. Foi verificado que por vezes o algoritmo com a inicialização de DOA não oferece o resultado esperado apenas ajustando para a DOA e sim para  $|\theta| > DOA$ . A justificativa vem quando analisamos a DOA no domínio da frequência para um sinal de voz banda larga como visto na Figura 4.7. Na figura, apesar da fonte estar posicionada em  $-40^\circ$ , vemos vários picos entre  $-40$  e  $-90^\circ$ . O mesmo experimento foi realizado para o esquema utilizando filtros de decorrelação. Os resultados são vistos na Figura 4.8.

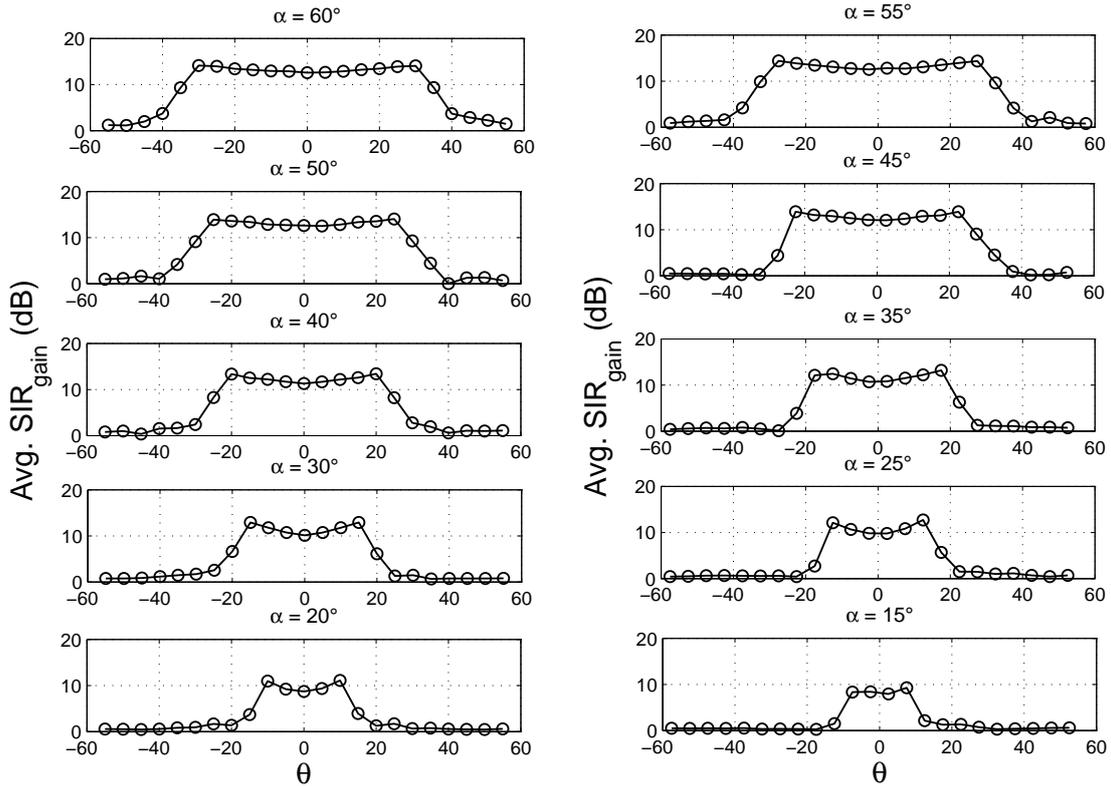


Figura 4.5: Resultados para os parâmetros sugeridos.

## 4.2 SEPARAÇÃO DE CANTOS DE AVES

Outras aplicações além da voz podem ser de grande utilidade. A análise dos sons da natureza tem um papel importante na detecção de variações, comportamento animal e conservação do meio ambiente [27]. Nesse contexto, podemos gravar sons de aves em arranjos de microfones e verificar que por vezes existem a presença de várias aves cantando simultaneamente. Tal cenário caracteriza o ambiente de BSS como visto neste trabalho. Os sons dos pássaros têm distribuições que podem ser bem aproximadas por curvas laplacianas (Figura 4.9). Isso significa que podemos tirar proveito do fato de serem fontes não gaussianas. Algoritmos como o visto na Seção 3.2 tiram proveito dessa característica do sinal para efetuar a separação cega. Aplicando a algoritmo simplificado de segunda ordem descrito pela Equação (3.44) obtivemos uma separação como vista na Figura 4.10 para duas aves a menos de 1 metro do arranjo. A Figura 4.11 apresenta os cantos de outras duas aves antes da mistura, misturados e depois do processamento do TRINICON.

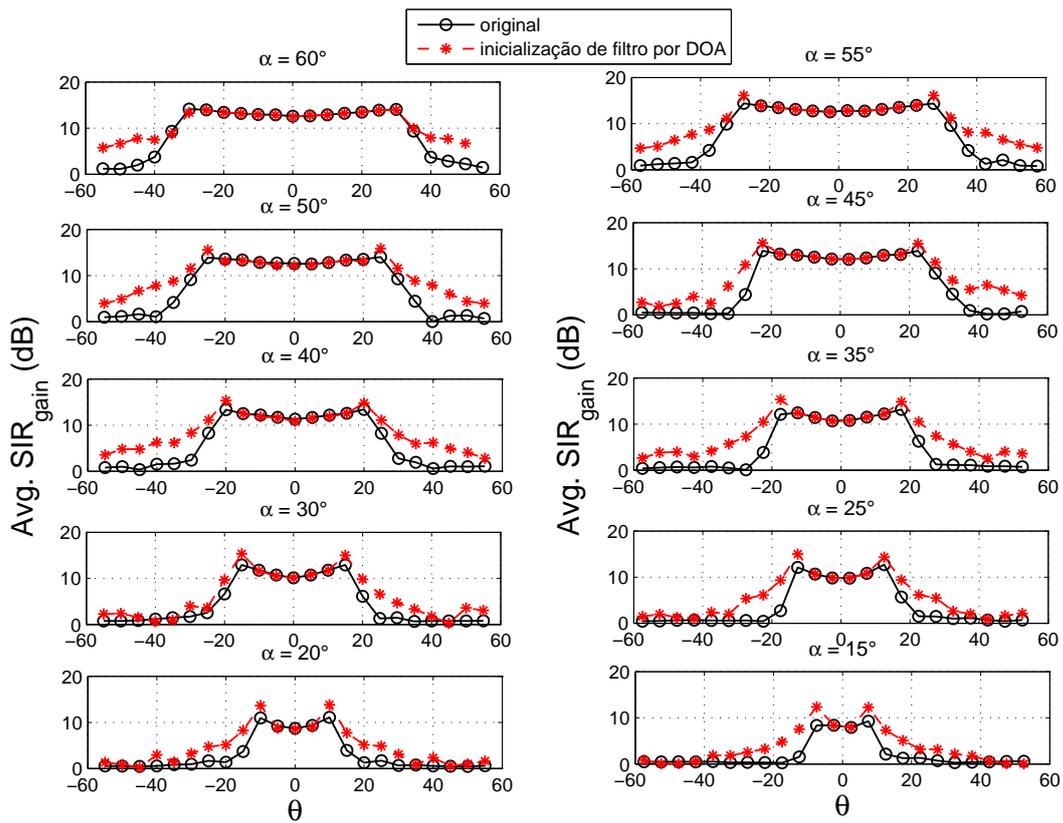


Figura 4.6: Comparação utilizando o método de conhecimento prévio de DOA.

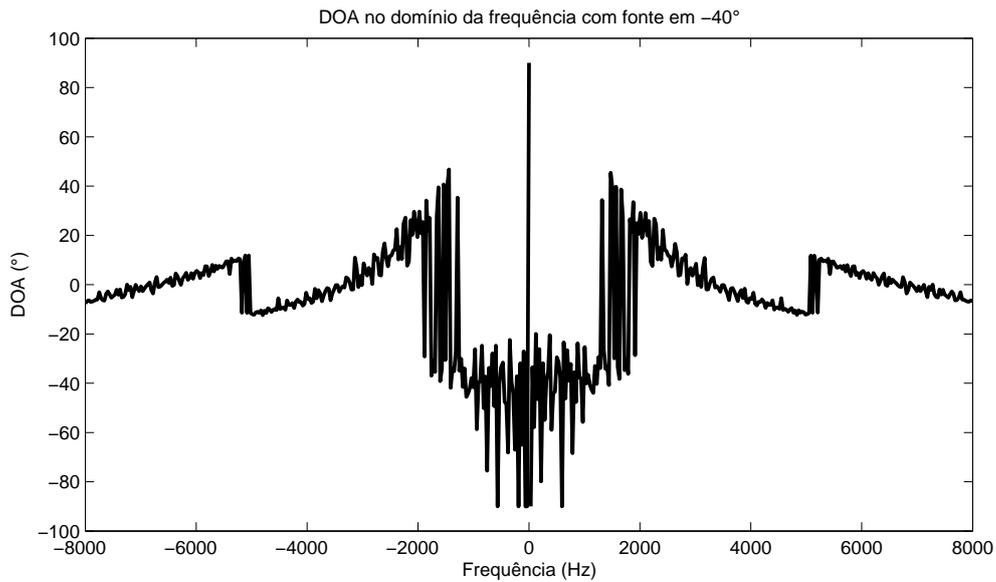


Figura 4.7: DOA no domínio da frequência para fonte em  $40^\circ$ .

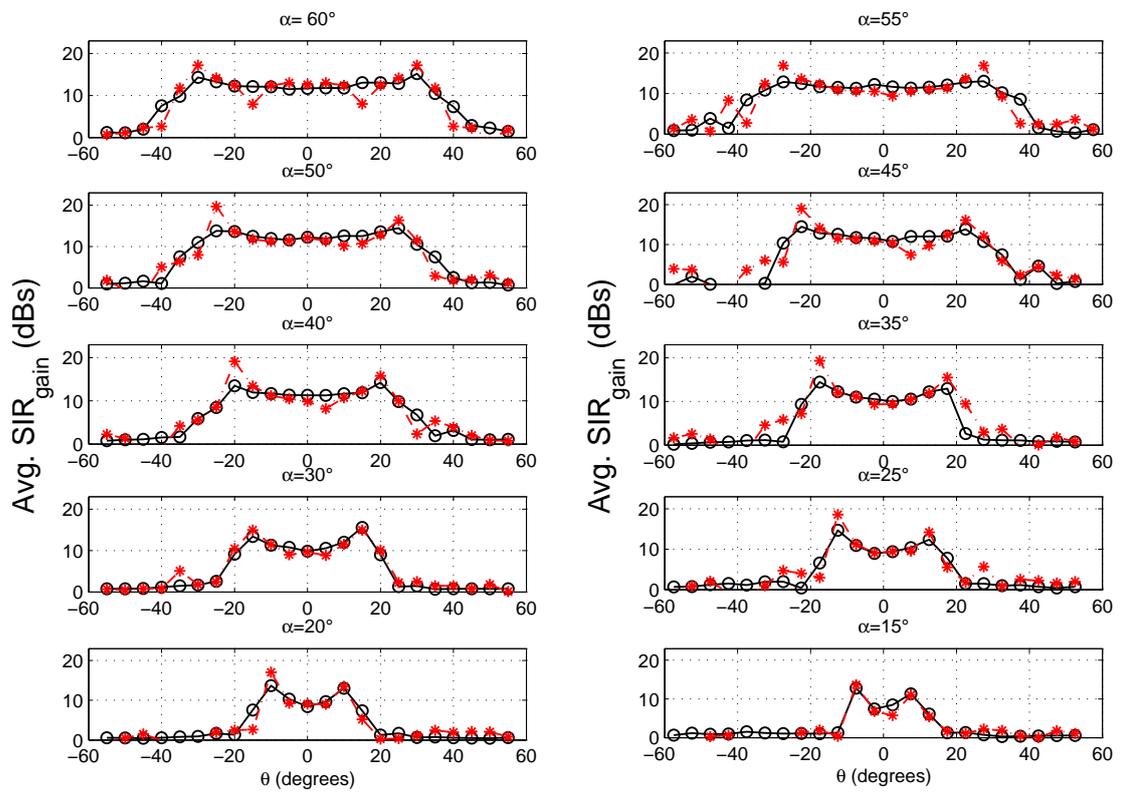


Figura 4.8: Resultados para filtro de decorrelação.

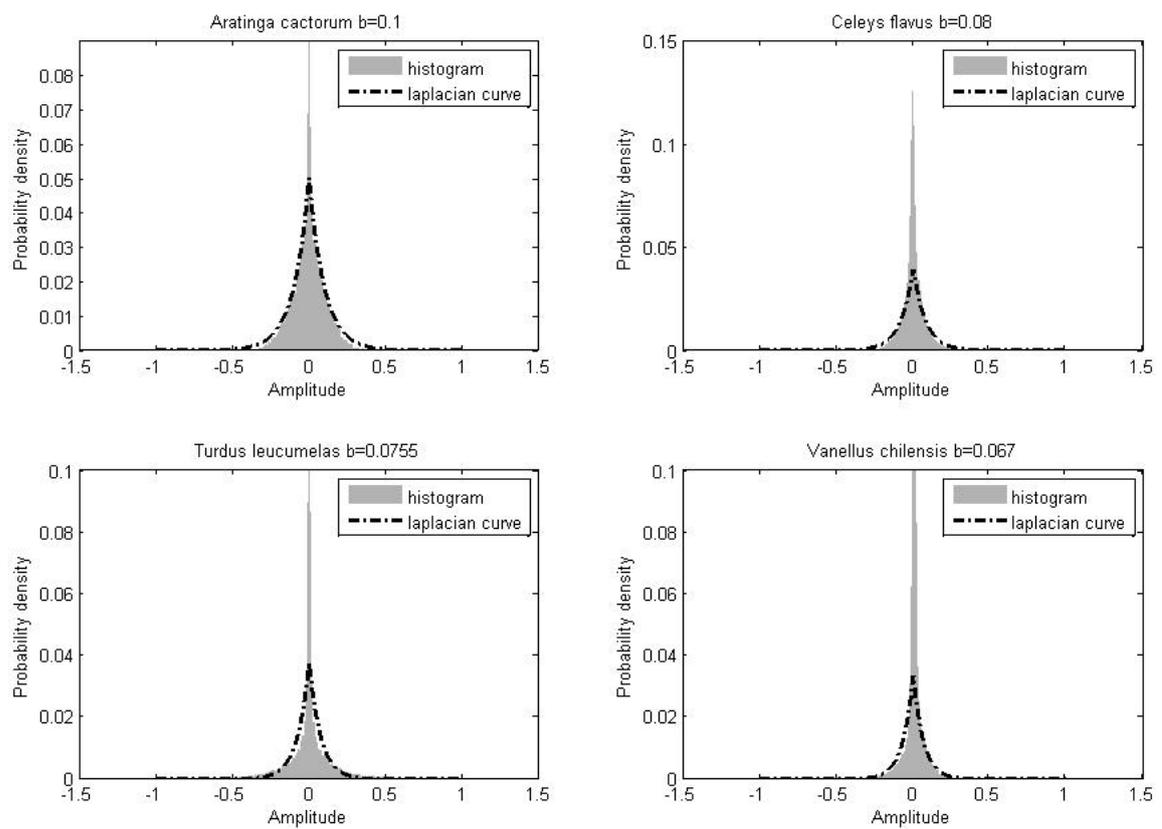


Figura 4.9: Comparação entre distribuições laplacianas e histogramas de cantos de aves de diferentes espécies. O coeficiente  $b$  é o fator de escala da curva laplaciana.

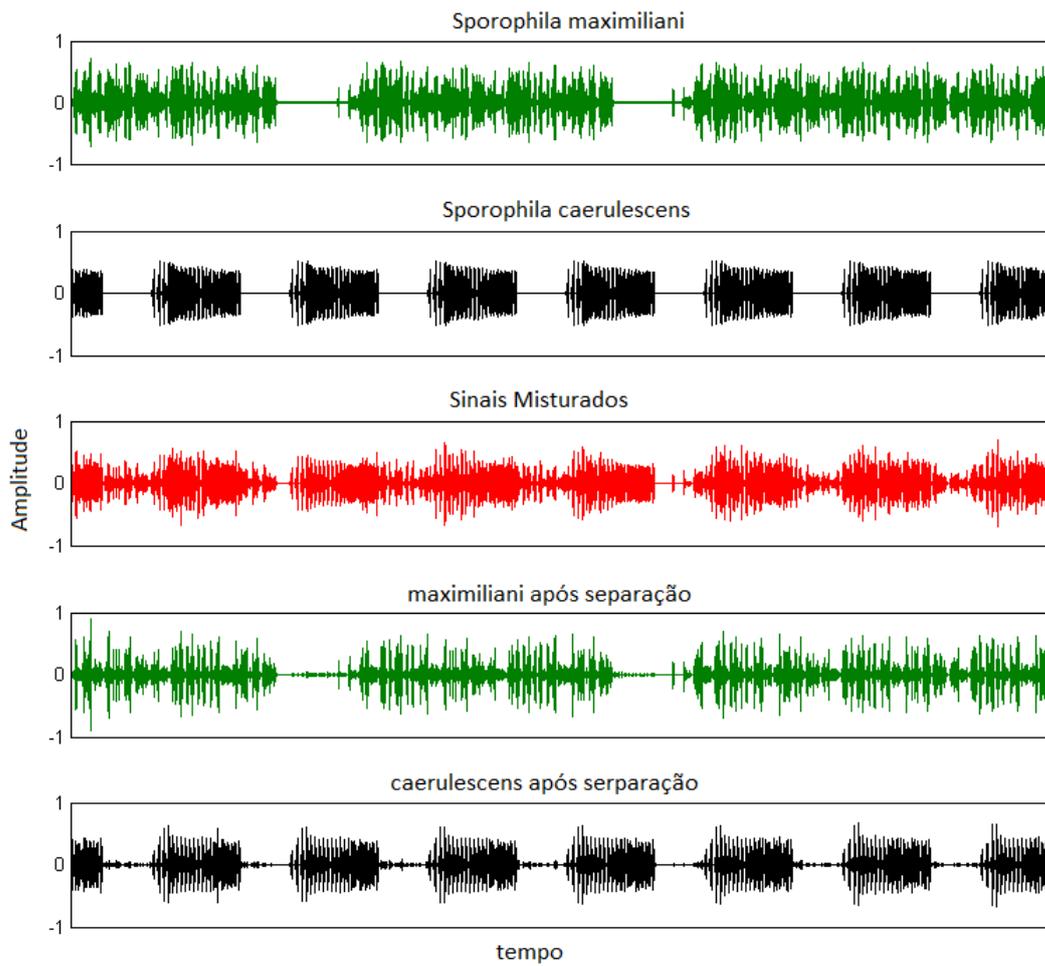


Figura 4.10: Comparação entre sinais da fontes, sinais misturados e separados.

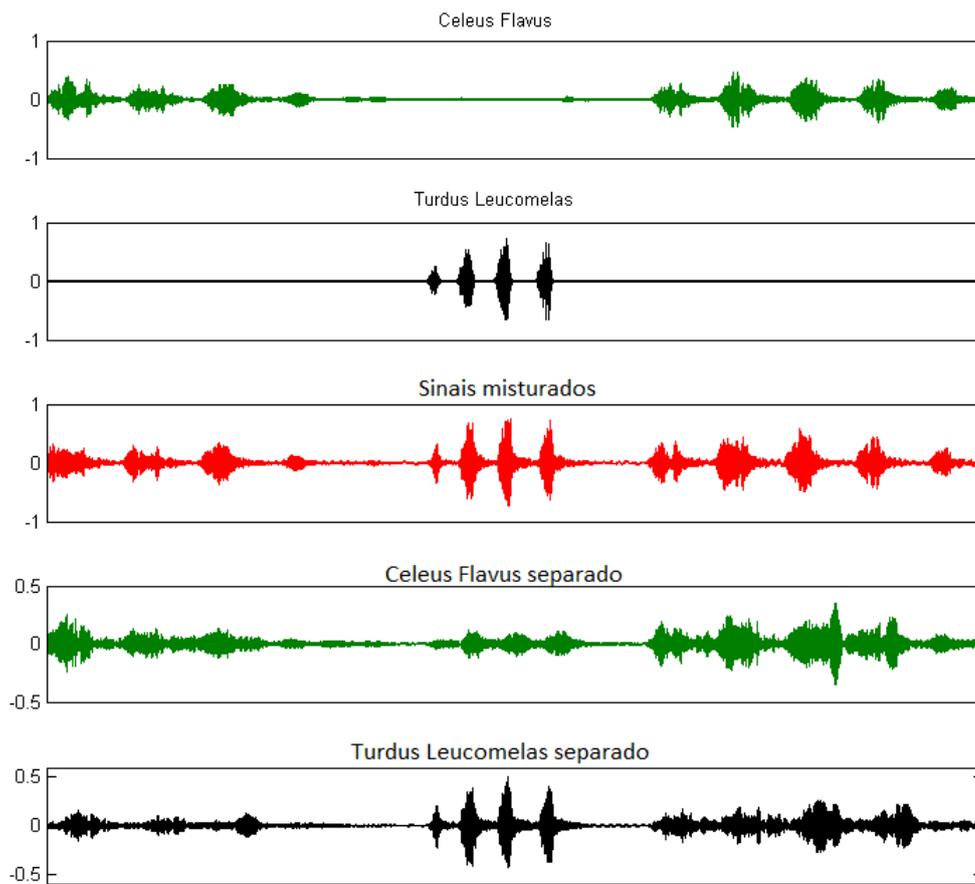


Figura 4.11: Comparação entre sinais da fontes, sinais misturados e separados.

## 5 CONCLUSÃO

Misturas sonoras são frequentemente observadas nos mais diversos ambientes do dia a dia como salas de aula, escritórios, lojas, meios de transporte, natureza entre outros. Essas misturas ocorrem quando mais de uma fonte sonora estão ativas simultaneamente. Porém, somos capazes de nos concentrarmos em apenas uma por vez. Além disso, misturas sonoras não são interessantes para equipamentos que processem áudio como próteses auditivas, televisores inteligentes, gravadores forenses etc. Algoritmos de BSS exploram características dos sinais sonoros a fim de separar estas misturas. Porém, apesar de apresentarem bons níveis de separação, os algoritmos de BSS apresentam alguns problemas quando expostos à certas configurações do ambiente, em especial fontes que irradiam sinais sonoros a partir de ângulos oblíquos. Este trabalho identificou que quando fontes estão em um mesmo semi plano o desempenho é reduzido de forma abrupta. Assim, utilizou-se técnicas de inicialização para tentar suavizar o problema. Por fim, uma técnica avançada de separação também foi utilizada para explorar uma área ainda pouco estudada no ramo da separação cega de fontes como o canto dos pássaros.

No Capítulo 1, uma breve introdução sobre o tema foi realizada. No Capítulo 2 apresentamos arranjos de microfones, modelos para misturas instantâneas e para misturas convolutivas além de abordar algumas medidas de desempenho e outros métodos matemáticos úteis para a pesquisa na área. O Capítulo 3 revisou técnicas importantes de BSS bem como técnicas complementares permitir e garantir o seu bom desempenho. Técnicas simples visando melhorar o funcionamento de algoritmos BSS foram introduzidas. No Capítulo 4 experimentos a cerca dos tópicos mostrados no Capítulo 3 foram realizados.

Os resultados dos experimentos mostraram que a inicialização com conhecimentos da DOA podem aumentar de maneira significativa o desempenho de algoritmos de BSS quando em configurações não favoráveis do ambiente. Este ambiente foi gerado por meio de respostas impulsivas artificiais para 35 diferentes posições das fontes. Foi verificado que inicializar o algoritmo de separação com a DOA exata dessas posições pode não ter o resultado esperado. Por isso, utilizaram-se ângulos onde foi verificada uma concentração maior de energia. Além do DOA, foi utilizada uma técnica de

decorrelação temporal do sinal por meio de um filtro passa alta. A inspiração veio de um trabalho que melhorou a SIR utilizando técnicas de decorrelação total que se utilizaram de predição linear. Infelizmente, a técnica usada neste trabalho não obteve o resultado esperado.

Por incentivo de pesquisadores do Departamento de Biologia da Universidade de Brasília, também foi utilizado um algoritmo mais robusto de BSS (TRINICON), cuja complexa implementação foi cedida pela Universidade de Erlangen-Nuremberg com o intuito de separar cantos de aves de maneira cega. Uma dificuldade deste tipo de separação é obter sinais livres de ruído e interferência para realização de um estudo adequado. Na maioria das gravações existentes é possível verificar a presença de uma grande quantidade de outros sons provenientes da natureza, o que aumenta a complexidade do cenário. Além disso, cenários naturais podem ter uma grande variedade de formas e configurações, havendo uma carência de modelos de respostas impulsivas artificiais para estes cenários que facilitariam o estudo. Por isso, utilizou-se um arranjo de microfones desenvolvido na Universidade de Brasília para fazer gravações reais, ou seja, com presença de ruído e respostas impulsivas reais de maior complexidade que as geradas artificialmente para este trabalho. Mesmo assim, o TRINICON apresentou resultados satisfatórios para fontes até 1 m de distância do arranjo.

As causas do decaimento do desempenho do algoritmo BSS no domínio da frequência para fontes em um mesmo semiplano ainda é desconhecida. Sabe-se apenas que existe a necessidade de filtros não causais nestas situações e, portanto, deve existir alguma relação com esta condição. Por isso, sugere-se uma investigação do funcionamento do algoritmo para estes casos visando uma futura proposta de correção do algoritmo. O uso de técnicas de decorrelação mais complexas que a utilizada neste trabalho para cenários não favoráveis pode trazer melhorias de desempenho. Trabalhos anteriores já mostraram que utilizar a decorrelação por meio da predição linear pode trazer benefícios, resta saber se este tipo de decorrelação também melhoraria o desempenho do BSS no domínio da frequência para fontes em posições não favoráveis à separação.

Aves na natureza não estão expostas a ambientes com paredes como salas e escritórios. Modelos para respostas impulsivas de florestas podem ser úteis para pesquisas futuras de BSS envolvendo cantos de aves e outros sons silvestres.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] J. Allen e D. Berkley, Image method for efficiently simulating small room acoustics, *J. Acoust. Soc. Am.*, vol. 65, Issue 4, pp. 943-950, 1979.
- [2] S. Amari, A. Cichoki e H. H. Yang, A New Learning Algorithm for Blind Signal Separation, *Advances in Neural Information Processing Systems*, pp. 757 - 763, MIT Press, 1996.
- [3] S. Amari, Natural Gradient Works Efficiently in Learning, *Neural Computation*, vol. 10, No. 2, pp. 251-276, 1998.
- [4] S. Amari, Why Natural Gradient?, in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 1213-1216, 1998.
- [5] A. J. Bell e T. J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, *Neural Comput.* vol. 7, pp. 1129-59, 1995 Nov,.
- [6] J. Benesty e T. Gänstler, A Robust Least Squares Adaptive Algorithm, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 6, pp. 3785 - 3788, ICASSP, 2001.
- [7] H. Brehm and W. Stammers, Description and generation of spherically invariant speech-model signals, *Signal Processing* vol. 12, pp. 119-141, 1987.
- [8] H. Buchner, R. Aichner e W. Kellermann, TRINICON: a versatile framework for multichannel blind signal processing, in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP, 2004.
- [9] G. Casella, R. L. Berger, *Statistical Inference*, Second Edition, Thomson Learning, 2002.
- [10] A. Cauchy, Methodes generales pour la resolution des syst'emes d'equations simultanées. *Compte Rendu des S'eances de L'Acad'emie des Sciences XXV*, Vol. S'erie A, No. 25, pp. 536-538, 1847.

- [11] T. M. Cover e J. A. Thomas. Elements of Information Theory New York, Wiley, 1991.
- [12] H. Do, H. F. Silverman e Y. Yu. *A real-time SRP-PHAT source localization implementation using stochastic region contraction(SRC) on a large-aperture microphone array*. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2007.
- [13] I. L. Freire e J. A. Apolinário Jr. GCC-based DoA Estimation of Overlapping Muzzleblast and Shockwave Components of Gunshot Signals. In Proc. of the 2nd IEEE Latin American Symposium on Circuits and Systems, Bogotá, 2011.
- [14] I. L. Freire e J. A. Apolinário Jr. DoA of gunshot signals in a spatial microphone array: performance of the interpolated Generalized Cross-Correlation method. In Proc. of the 5th Argentine Conference on Micro-Nanoelectronics, Technology, and Applications, Buenos Aires, 2011.
- [15] M. Fujimoto e Y. Ariki. Noise robust hands-free speech recognition using microphone array and kalman filter as frontend system of conversational TV. , In Proc. of the Workshop on Multimedia Signal Processing, pp. 268 - 271, 2002.
- [16] B. W. Gillespie, H. S. Malvar e D. A. F. Florêncio, Speech Dereverberation via Maximum Kurtosis Subband Adaptive Filtering, In Proc. of the ICASSP, IEEE, vol. 6, pp. 3701-3704, 2001.
- [17] A. Hyvärinen e E. Oja, Independent Component Analysis: Algorithms and Applications, Neural Networks, vol. 13, pp. 411-430, 2000.
- [18] A. Hyvärinen, J. Karhunen e E. Oja, Independent Component Analysis. Wiley, 2001
- [19] E. Jondeau, M. Rockinger Gram-Charlier densities, Journal of Economic Dynamics & Control, vol. 25, pp. 1457-1483, 2001
- [20] R. K. Miranda, S. K. Sahoo, J. P. C. L. da Costa e R. Zelenovsky, Improved Frequency Domain Blind Source Separation for Audio Signals via Direction of Arrival Knowledge. Submetido à SDPS 2013.
- [21] K. Kokkinakis, V. Zarzoso e A. K. Nandi, Blind Separation of Acoustic Mixtures Based on Linear Prediction Analysis, 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), Nara, Japan , Abr. 2003.

- [22] K. Matsuoka, Minimal distortion principle for blind source separation, SICE 2002. Proceedings of the 41st SICE Annual Conference, vol.4, pp. 2138-2143, Aug. 2002.
- [23] L. C. Parra e Christopher V. Alvino, Geometric source separation: merging convolutive source separation with geometric beamforming, IEEE Signal Processing Society Workshop, pp. 273-282, 2001.
- [24] H. Sawada, R. Mukai, S. Araki e S. Makino, Polar Coordinate Based Nonlinear Function for Frequency-Domain Blind Source Separation. In Proc. of ICASSP, vol. 1, pp. 1001-1004, Orlando, 2002.
- [25] H. Sawada, R. Mukai, S. Araki e S. Makino, A Robust Approach to the Permutation Problem of Frequency-Domain Blind Source Separation, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003), vol. V, pp. 381-384, Apr. 2003.
- [26] H. Sawada, R. Mukai, S. Araki e S. Makino, A Robust and Precise Method for Solving the Permutation Problem of Frequency-Domain Blind Source Separation, IEEE Trans. Speech and Audio Processing, vol.12, no. 5, pp. 530-538, Sep. 2004.
- [27] P. J. B. Slater, Fifty years of bird song research: a case study in animal behaviour., Animal behaviour, vol. 65, pp. 633-639, 2003.
- [28] P. Smaragdis, Blind separation of convolved mixtures in the frequency domain. Neurocomputing, vol. 22, pp. 21-34, 1998.
- [29] B. D. van Veen e K. M. Buckley, Beamforming: a versatile approach to spatial filtering, ASSP Magazine IEEE, vol. 5 , Issue 2, pp. 4-24, Abr. 1988.

## APÊNDICES

## APÊNDICE A

Código de BSS com código para geração de filtros de inicialização por DOA

```
1 function [y, ytar, yint, Wt, sir11, sir22, sir12, sir21] =  
    fdbss_HOS_constraint(x, x_ref, s, cfg, param)  
2  
3 %-----  
4 % Algoritmo online bloco a bloco utilizando estatísticas  
5 % de ordem alta por análise de banda estreita  
6 % x – sinais dos sensores  
7 % x_ref – sinais de referencia para calculo da SIR  
8 % cfg – estrutura com configuracoes  
9 % param – estrutura com parametros  
10 %-----  
11 %-----  
12  
13 %cfg.nmic = 2;  
14 %cfg.nsrc = 2;  
15 cfg.hfftlen = cfg.N/2+1; % frequencias nao redundantes  
16 param.freqRange = [0 cfg.fs/2];  
17  
18 param.freqStep = cfg.fs / cfg.N;  
19 param.initF = 1;  
20 param.lastF = cfg.N/2+1;  
21  
22 cfg.M = cfg.alpha .* floor((length(x)-cfg.N)/cfg.N); % numero  
    de blocos  
23 hwindow = repmat(hann(cfg.N, 'periodic'), [1 cfg.nmic]);  
24 hwindow3dim = repmat(hann(cfg.N, 'periodic'), [1 cfg.nmic cfg.  
    nsrc]);  
25  
26 % x = [zeros(cfg.N/2,2); x];  
27 % s = [zeros(cfg.N/2,2); s];  
28  
29 %
```

---

```

30 %-----InicializaÃ§Ã£o
    _____

31 %
    _____

32 %y = zeros(length(x)-cfg.N/2, cfg.nsrc);
33 %ytar = zeros(length(x)-cfg.N/2, cfg.nsrc);
34 %yint = zeros(length(x)-cfg.N/2, cfg.nsrc);
35 %yn = zeros(length(x)-cfg.N/2, cfg.nsrc);
36 y = zeros(cfg.M*cfg.N/cfg.alpha+cfg.N, cfg.nsrc);
37 ytar = zeros(cfg.M*cfg.N/cfg.alpha+cfg.N, cfg.nsrc);
38 yint = zeros(cfg.M*cfg.N/cfg.alpha+cfg.N, cfg.nsrc);
39 %Yref1_sir = zeros(cfg.M*cfg.N/cfg.alpha+cfg.N, cfg.nsrc);
40 %Yref2_sir = zeros(cfg.M*cfg.N/cfg.alpha+cfg.N, cfg.nsrc);
41 xref1(:,1) = x_ref(:,1,1);
42 xref1(:,2) = x_ref(:,1,2);
43 xref2(:,1) = x_ref(:,2,1);
44 xref2(:,2) = x_ref(:,2,2);
45
46
47 R_old = shiftdim(repmat(eye(cfg.nsrc), [1,1, cfg.N/2+1]), 2);
48 %R_old = zeros(cfg.N/2+1, cfg.nsrc, cfg.nsrc);
49 Rtilde_old = zeros(cfg.N/2+1, cfg.nsrc, cfg.nsrc);
50 %Rtilde_old = shiftdim(repmat(eye(cfg.nsrc), [1,1, cfg.N/2+1])
    ,2);
51
52
53 % Initialization of demixing filters
54 Wt = zeros(cfg.N, cfg.nsrc, cfg.nmic);
55 src_half_plane = zeros(1, cfg.nsrc); % 1 corresponde ao
    semiplano esquerdo 2 ao direito
56 DOA_deg = zeros(1, cfg.nsrc);
57 DOA_rad = zeros(1, cfg.nsrc);
58 delay = zeros(1, cfg.nsrc);
59 delay_taps = zeros(1, cfg.nsrc);
60

```

```

61 if (param.known_DOA)
62
63
64 %Impulso Sinc
65 x_sinc = -3:.25:3;
66 delay_transf = sinc(x_sinc).*hamming(length(x_sinc))'.*0.9;
67 delay_transf = [ zeros(1, cfg.inittap) delay_transf zeros(1,
        cfg.N-cfg.inittap-length(delay_transf)) ];
68
69     for i = 1:cfg.nsrc
70         %Wt(cfg.inittap, i, i) = 1;
71         % equacao da DOA
72         DOA_deg(i) = (cfg.alfa - cfg.theta/2) + (i-1) * (cfg.
            theta/(cfg.nsrc-1));
73         % converter para radianos
74         DOA_rad(i) = DOA_deg(i) * pi / 180;
75         % calculando atraso
76         delay(i) = sin(DOA_rad(i)) * cfg.dHead / cfg.c0;
77         % quantizando atrasos
78         delay_taps(i) = int8(delay(i) * cfg.fs);
79
80         for j = 1:cfg.nmic
81             if i == j
82                 Wt(cfg.inittap, i, j) = 1;
83                 %Wt(:, i, j) = delay_transf;
84             elseif i < j
85                 Wt(cfg.inittap-delay_taps(i), i, j) = -1;
86                 %Wt(:, j, i) = circshift(-delay_transf, [1 -
                    delay_taps(i)]);
87             elseif i > j
88                 Wt(cfg.inittap+delay_taps(i), i, j) = -1;
89                 %Wt(:, j, i) = circshift(-delay_transf, [1 +
                    delay_taps(i)]);
90             end
91         end
92     end
93 else

```

```

94         % inicializa o sem utilizar DOA
95     for i = 1:cfg.nmic
96         Wt(cfg.inittap, i, i) = 1;
97     end
98 end
99 W = fft(Wt, [], 1);
100 W = W(1:cfg.N/2+1, :, :); % devido a simetria utilizar apenas N
    /2+1 valores de W
101
102 % Filtros de decorrelação de primeira ordem
103 if (param.decorr_filter)
104     % criando no dominio do tempo
105     decorr_filter=zeros(1, cfg.N);
106     decorr_filter(1)=1/1.95;
107     decorr_filter(2)=-0.97/1.95;
108
109     % tranferindo para o dominio da frequencia
110     decorr_filter=fft(decorr_filter);
111     decorr_filter=decorr_filter(1:cfg.N/2+1);
112     %inv_decorr_filter=1./decorr_filter;
113 end
114
115 %


---


116 %-----Loop Principal


---


117 %


---


118
119 plot_index = floor(cfg.M/10);
120 for m = 0:cfg.M
121     index_start = m*(cfg.N/cfg.alpha)+1; % Índice de
        inicio do bloco
122     index_end = m*(cfg.N/cfg.alpha)+cfg.N;
123     X = fft(x(index_start:index_end, :).*hwindow, [], 1);

```

```

124 X = X(1:cfg.N/2+1,:);
125 Xref1 = fft(xref1(index_start:index_end, :).*hwindow, [], 1);
126 Xref2 = fft(xref2(index_start:index_end, :).*hwindow, [], 1);
127 Xref1 = Xref1(1:cfg.N/2+1,:);
128 Xref2 = Xref2(1:cfg.N/2+1,:);
129 Xs = fft(s(index_start:index_end, :).*hwindow, [], 1);
130 Xs = Xs(1:cfg.N/2+1,:);
131
132 %

```

---

```

133 % Regra de atualizaÃ§Ã£o usando nao linearidade em
    coordenadas polares
134     % (Sawada, ICASSP 2002)
135 %

```

---

```

136
137 for i=1:param.loops,
138     Y = fast_ip_3dim2dim(W,X)+eps;
    % convoluÃ§Ã£o circular Y=W
    *X
139     % filtros de decorrelaÃ§Ã£o
140     if (param.decor_filter)
141         for mic=1:size(Y,2)
142             Y(:,mic)=Y(:,mic) .* decor_filter.';
143         end
144     end
145
146     % funcao nao linear em coordenadas polares
147     PhiY = Y ./ (abs(Y) + eps) .* tanh( param.phi .* abs(
        Y)); % assumir ln(cos(y)) como a pdf (Benesty
        ICASSP 2001)
148     %PhiY = Y ./ abs(Y); %
        assumir pdf laplaciana
149
150     for p = 1:cfg.nsrc

```

```

151         for q = 1:cfg.nsrc
152             Rtilde(:,p,q) = cfg.lambda * Rtilde_old(:,p,q)
                + (1-cfg.lambda) * (PhiY(:,p) .* conj(Y(:,
                q))); % mÃ©dia movel (lambda) de Rtilde
153         end
154     end
155
156     tmp = -Rtilde;
157     for q = 1:cfg.nsrc
158         tmp(:,q,q) = 0; % non-holonomic
                - diag[] - []
159         %tmp(:,q,q) = 1-Rtilde(:,q,q); % holonomic
160     end
161
162     W = W + param.mu * fast_ip(tmp,W); % AtualizaÃ§Ã£o
                de W
163 end
164 Rtilde_old = Rtilde;
165
166 %


---


167 % RestriÃ§Ã£o do filtro
168 %


---


169     % explorando simetria
170     Wt = real(iff([W(1:cfg.N/2+1,:,:) ; conj(W(cfg.N
                /2:-1:2,:,:)]) , cfg.N));
171     Wt(cfg.N/4+1:end, :, :) = 0;
172     W = fft(Wt, [], 1);
173     W = W(1:cfg.N/2+1, :, :);
174
175 %


---



```

```

176 %—————MDP (Matsuoka ICA 2001)
    _____

177 % Minimal Distortion Principle – Principio da
    Distorcao Minima —————
178 %
    _____

179
180 for f = 1:cfg.hfftlen
181     Wf = squeeze(W(f, :, :));
182
183         %calcular DOA para correcao da permutacao
184     Theta(f, :) = doa( Wf, f, cfg.nsrc, param.freqStep, cfg
        .c0, cfg.dHead );
185
186         % verificar erro – algoritmo instavel quando
        mu  $\tilde{\alpha}$  grande
187     try
188         Winv(f, :, :) = pinv(Wf+eye(cfg.nmic)*cfg.delta);
189     catch err
190         display('ERRO');
191     end
192     W(f, :, :) = diag(diag(squeeze(Winv(f, :, :)))) * Wf;
193 end
194
195 %
    _____

196 %—————Corrigir permutacao por DOA
    _____

197 %
    _____

198 %nao calcular quando theta  $\tilde{\alpha}$  complexo
199 Theta(imag(Theta)  $\tilde{=}$  0) = 0;
200
201 %ordenar Theta => Theta_s

```

```

202     [Theta_s, PI_temp] = sort(Theta,2);
203
204     %obter a mÃ©dia da DOA nas frequencias onde nao hÃ¡
        aliasing
205     Theta_mean = mean(Theta_s(40:110,:));
206
207     %a lista de frequencias corrigidas
208     PI = zeros(cfg.hfftlen, cfg.nsrc);
209
210     %permutar
211     if(cfg.use_DOA_fix)
212         correctedDoa=0;
213         for f = 1:cfg.hfftlen
214             if( abs(Theta_s(f,:)-Theta_mean) < param.th_Theta
                ); % sum(abs(Theta_s(i,:)-Theta_mean) < param.
                th_Theta) > 0
215                 PI(f,:)=PI_temp(i,:);
216                 Wf = squeeze(W(f,:,:));
217                 if( PI(f,:) ~= ([ 1 2 ]));
218                     circshift(Wf,1);
219                     W(f,:,:) = circshift(Wf,2);
220                                     %incrementar numero de
                                                frequencias
                                                permutadas
221                 correctedDoa=correctedDoa+1;
222             end
223         end
224     end
225     display([ num2str(correctedDoa) ' bins corrected using
        DOA approach at block ' num2str(m) ]);
226 end
227
228 %

```

---

```

229 %—————Filtragem com superposicao e soma usando
        hann-window—————

```

```

230 %


---


231 % efetuar separação
232 Y = fast_ip_3dim2dim(W,X); %
    convolucao circular Y=W*X
233 Ys = fast_ip_3dim2dim(W,Xs);
234 Yref1 = fast_ip_3dim2dim(W,Xref1);
235 Yref2 = fast_ip_3dim2dim(W,Xref2);
236
237 Yreft1 = real( ifft( [Yref1(1:cfg.N/2+1,:); conj(Yref1(cfg.N/2:-1:2,:))] , cfg.N))/(cfg.alpha/2);
238 Yreft2 = real( ifft( [Yref2(1:cfg.N/2+1,:); conj(Yref2(cfg.N/2:-1:2,:))] , cfg.N))/(cfg.alpha/2);
239
240 % transformar para o dominio do tempo
241 Yt = real( ifft( [Y(1:cfg.N/2+1,:); conj(Y(cfg.N/2:-1:2,:))] , cfg.N))/(cfg.alpha/2); % escalonar
    por cfg.alpha/2 j $\tilde{\omega}$  que a janela adiciona uma
    constante
242 Yst = real( ifft( [Ys(1:cfg.N/2+1,:); conj(Ys(cfg.N/2:-1:2,:))] , cfg.N))/(cfg.alpha/2);
243
244 y(index_start:index_end,:) = y(index_start:index_end,:) +
    Yt;
245 ytar(index_start:index_end,1) = ytar(index_start:index_end,1) + Yst(:,1);
246 yint(index_start:index_end,2) = yint(index_start:index_end,2) + Yst(:,2);
247
248 ytar(index_start:index_end,2) = y(index_start:index_end,2)
    - yint(index_start:index_end,2);
249 yint(index_start:index_end,1) = y(index_start:index_end,1)
    - ytar(index_start:index_end,1);
250
251 % calcular a sir para cada bloco
252 % SIRs nos microfones

```

```

253 SIRin11(m+1) = (10*log10((sum(x_ref(index_start:index_end
    ,1,1).^2,1)+eps)./(sum(x_ref(index_start:index_end,2,1)
    .^2,1)+eps)));
254 SIRin21(m+1) = -SIRin11(m+1);
255 SIRin22(m+1) = (10*log10((sum(x_ref(index_start:index_end
    ,2,2).^2,1)+eps)./(sum(x_ref(index_start:index_end,1,2)
    .^2,1)+eps)));
256 SIRin12(m+1) = -SIRin22(m+1);
257
258 % SIRs nas saidas
259 SIRout11(m+1) = (10*log10((sum(Yreft1(:,1).^2,1)+eps)./(
    sum(Yreft2(:,1).^2,1)+eps)));
260 SIRout21(m+1) = -SIRout11(m+1);
261 SIRout22(m+1) = (10*log10((sum(Yreft2(:,2).^2,1)+eps)./(
    sum(Yreft1(:,2).^2,1)+eps)));
262 SIRout12(m+1) = -SIRout22(m+1);
263
264 % plotar a SIR a cada m blocos
265 if (mod(m, plot_index) == 0 || (m == cfg.M) || m == -1)
266     %sir=SIRout11-SIRin11;
267     subplot(2,1,1)
268     plot(SIRout22-SIRin22); grid on;
269     % parar para que o programa tenha tempo de
    efetuar o plot
270     pause(0.01)
271 end
272 end
273
274 % calcular o ganho de SIR
275 sir11(1,:) = SIRout11 - SIRin11;
276 sir22(1,:) = SIRout22 - SIRin22;
277 sir12(1,:) = SIRout12 - SIRin12;
278 sir21(1,:) = SIRout21 - SIRin21;
279
280
281 Wt = Wt(1:cfg.N/4, :, :);
282 Wt = permute(Wt, [1 3 2]);

```

## Código para cálculo da DOA utilizando os zeros dos filtros de separação

```
1 function [ thetas ] = doa( Wf, f, nsrc, freqStep, c0, dHead )
2 % Calcular direção de chegada baseado na direção nula dos
   filtros de separação
3 % conforme Sawada 2002
4 %
5 % Wf – matrix de separação de uma frequência (bin)
6 % f – frequência de Wf
7 % nsrc – numero de fontes
8 % freqStep – distancia em Hz entre duas frequencias
   vizinhas
9 % c0 – velocidade da onda
10 % dHead – distancia entre sensores
11 %
12 % based on Sawada 2002 | written by K. Miranda, Ricardo –
13 % rickehrle@gmail.com – 2012
14
15 thetas = zeros(1,nsrc);
16 for s = 1:nsrc
17     % evitar inversao, caso 2x2
18     if(nsrc == 4)
19         %formula da DOA caso 2x2
20         thetas(s)=acos((angle(-Wf(1,s)) – angle(Wf(2,s))) / (2
           * pi * freqStep * f * (c0^-1) * dHead));
21     else
22         if(s == 1)
23             Wfinv = pinv(Wf);
24         end
25         %formula da DOA, caso generico com inversao de
           matriz
26         thetas(s)=acos((angle(Wfinv(s,2)) – angle(Wfinv(s,1)))
           / (2 * pi * freqStep * f * (c0^-1) * dHead));
27     end
28 end
29 end
```

## APÊNDICE B

Código com exemplo de configuração

```
1 %% Testar o codigo de separacao
2
3 cfg.fs = 16000;
4 cfg.dHead = 0.16; % d=0.16m for Ch2-Ch5 && d=0.2m for Ch1-Ch6
5 cfg.c0 = 340;
6 cfg.DOASET = 2380;
7 cfg.roomIdx = 1;
8 cfg.Arrange = 0;
9 cfg.tlen = 40; % largura do tempo em segundos
10
11 %
12 %-----Ambiente
13 %
14 cfg.imp_resp_file = 'RIRs/RIR-Syn-[T60-30ms]-[R-1m]-[O-1]-[
    MicSpc-0.2m].mat'; % arquivo contendo as respostas
    impulsional
15 cfg.path_source = { './wav/E202A.WAV' , './wav/E205A.WAV' };
16
17 cfg.nsrc = 2; % numero de fontes
18 cfg.nmic = 2; % numero de microfones
19 cfg.micidx = [2 5]; % [1 2 3 4 5 6] selecionar
    microfones a utilizar
20 cfg.sig_len = 40; % tamanho em segundos a ser
    utilizado do arquivo de som - 0 para todo o sinal
21 %cfg.lsidx = [28 29]; % uais respostas impulsionalis
    utilizar - mudarao quando em loop
22
23
```

24

25 %

---

26 %-----Parametros

---

27 %

---

```
28 cfg.alpha = 2; % fator de superposicao
    (descolcamento do quadro = N/alpha)
29 cfg.N = 2048; % tamanho do bloco =
    tamanho da FFT
30 cfg.lambda = 0.5; % media movel
31 cfg.inittap = 30; %13 % posicao do tap o filtro
    de inicializacao
32 param.loops = 100; % numero de iteracoes ICA
33 param.mu = 0.0002; % tamanho do passo
34 param.phi = 100; % ganho da nao
    linearidade (Sawada ICASSP 2002, Table 1)
35 param.th_Theta = 10*pi/180; % gatilho da permutacao
    de DOA (rad)
36 param.th_U = 1; % gatilho da permutacao
    de correlacao
37 cfg.delta = 0.0000001;
38 cfg.param_phi = param.phi;
39 cfg.use_DOA_fix = false; % usar correção por DOA
40 cfg.use_perm_fix = false; % usar correcao por
    permutacao - master switch - ativar/desativar correcoes de
    permutacao
41 param.known_DOA = true; % inicializar utilizando
    DOA
42 param.decorr_filter = false % utilizar filtros de
    decorrelacao
43
44 %%
```

```

45 number_hops=12;      % numero de saltos(hops) entre as fontes.
    12hops=60graus, 10hops=50graus ...
46 scenario_config = [ 1 24 18 5]; % [<1a fonte> <2a fonte> <
    posicao do centro> <distancia angular entre pontos da
    resposta impulsional>]
47
48
49 scn_number_rings=length(scenario_config(:,1));
50
51 %


---


52 %-----Executar BSS


---


53 %


---


54
55 % andar entre aneis (se configurado)
56 for ii=1:scn_number_rings
57     disp(['-----Starting ring--' num2str(ii) '-calculations
    ...-----']);
58
59     % andar ao longo de um anel
60     for jj=scenario_config(ii,1):(scenario_config(ii,2)-
        number_hops)
61
62         % calcular DOAS
63         cfg.theta = scenario_config(ii,4)*number_hops;
64         angle_1st_source=0-(scenario_config(ii,3)-jj)*
            scenario_config(ii,4);
65         angle_bet_sources=angle_1st_source+(scenario_config(ii
            ,4)*number_hops)/2;
66         cfg.alfa=angle_bet_sources;
67
68         % posicoes
69         cfg.lsidx = [ jj , jj+number_hops];

```

```

70
71     % pastas a serem guardados os resultados
72     path = ['results/exp_DOA_sin16/' num2str(ii) '_m/
73           theta_' num2str(cfg.theta) '/' num2str(cfg.lsidx(1))
74           '_' num2str(cfg.lsidx(2)) '/'];
75     mkdir(path);
76
77     % gerar sinais
78     [x,x_ref,s,cfg] = GenSignals(cfg);
79     wavwrite(x, cfg.fs, 16, [path 'input.wav']);
80
81     % executar BSS
82     disp(['Calculate pos ' num2str(jj) '_' num2str(jj+
83           number_hops) ' ' num2str(cfg.tlen) 's ' num2str(
84           param.loops) 'iter ring' num2str(ii) ' ' num2str(
85           scenario_config(ii,4)*number_hops) ' spacing at '
86           num2str(angle_bet_sources) ' degrees']);
87     [y, ytar, yint, Wt, sir11, sir22, sir12, sir21] =
88     fdbss_HOS_constraint(x,x_ref,s,cfg,param);
89
90     % mostrar o que foi executado
91     disp(['Results for Syntetic pos' num2str(jj) '_'
92           num2str(jj+number_hops) '_' num2str(cfg.tlen) 's_'
93           num2str(param.loops) 'iter_ring' num2str(ii) '_'
94           num2str(scenario_config(ii,4)*number_hops) '
95           spacing_' num2str(angle_bet_sources) 'deg Saved.'])
96     ;
97
98     % guardar resultados
99     wavwrite(y(:,1), cfg.fs, 16, [path 'out1.wav']);
100    wavwrite(y(:,2), cfg.fs, 16, [path 'out2.wav']);
101    save([path 'sirs.mat'], 'sir11', 'sir22', 'sir12', '
102          sir21');
103    save([path 'cfg.mat'], 'cfg');
104    end
105    disp(['Ring ' num2str(ii) ' done!']);
106 end

```

```

94
95 %%
96 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Segunda parte
97 disp( 'SECOND PART' )
98 % trocar fontes para de fato testar todas as posicoes sob as
    mesmas condicoes
99 cfg.path_source = { './wav/E205A.WAV' , './wav/E202A.WAV' };
100
101 scenario_config = [ 13 35 18 5];
102
103 scn_number_rings=length( scenario_config (:,1) );
104
105 %
    _____

106 %-----Executar BSS
    _____

107 %
    _____

108
109
110 for ii=1:scn_number_rings
111     disp( [ '-----Starting ring-' num2str(ii) '-calculations
    ...-----' ] );
112     for jj=scenario_config(ii,1):(scenario_config(ii,2)-
    number_hops)
113         cfg.theta = scenario_config(ii,4)*number_hops;
114         angle_1st_source=0-(scenario_config(ii,3)-jj)*
            scenario_config(ii,4);
115         angle_bet_sources=angle_1st_source+(scenario_config(ii
            ,4)*number_hops)/2;
116         cfg.alfa=angle_bet_sources;
117         cfg.lsidx = [ jj , jj+number_hops];
118         path = [ 'results/exp_DOA_sin16/' num2str(ii) '_m/
            theta_' num2str(cfg.theta) '/' num2str(cfg.lsidx(1)
            ) '_' num2str(cfg.lsidx(2)) '/' ];

```

```

119     mkdir(path);
120     [x,x_ref,s,cfg] = GenSignals(cfg);
121     wavwrite(x, cfg.fs, 16, [path 'input.wav']);
122     disp(['Calculate pos ' num2str(jj) '- ' num2str(jj+
        number_hops) ' ' num2str(cfg.tlen) 's ' num2str(
        param.loops) 'iter ring ' num2str(ii) ' ' num2str(
        scenario_config(ii,4)*number_hops) ' spacing at '
        num2str(angle_bet_sources) ' degrees']);
123     [y, ytar, yint, Wt, sir11, sir22, sir12, sir21] =
        fdbss_HOS_constraint(x,x_ref,s,cfg,param);
124     disp(['Results for Syntetic pos ' num2str(jj) '- '
        num2str(jj+number_hops) '- ' num2str(cfg.tlen) 's '
        num2str(param.loops) 'iter ring ' num2str(ii) '- '
        num2str(scenario_config(ii,4)*number_hops) '
        spacing-' num2str(angle_bet_sources) 'deg Saved.'])
        ;
125     wavwrite(y(:,1), cfg.fs, 16, [path 'out1.wav']);
126     wavwrite(y(:,2), cfg.fs, 16, [path 'out2.wav']);
127     save([path 'sirs.mat'], 'sir11', 'sir22', 'sir12', '
        sir21');
128     save([path 'cfg.mat'], 'cfg');
129     end
130     disp(['Ring ' num2str(ii) ' done!']);
131 end

```