



NOVAS TÉCNICAS DE SEPARAÇÃO CEGA DE FONTES E DE FILTRAGEM
ADAPTATIVA EM SUBBANDAS

Jéssica Bartholdy Sanson

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Mariane Rembold Petraglia

Rio de Janeiro
Outubro de 2016

NOVAS TÉCNICAS DE SEPARAÇÃO CEGA DE FONTES E DE FILTRAGEM
ADAPTATIVA EM SUBBANDAS

Jéssica Bartholdy Sanson

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE
ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA
ELÉTRICA.

Examinada por:

Prof. Mariane Rembold Petraglia, Ph.D.

Prof. Amaro Azevedo de Lima, Ph.D.

Prof. Fernando Gil Vianna Resende Junior, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
OUTUBRO DE 2016

Sanson, Jéssica Bartholdy

Novas Técnicas de Separação Cega de Fontes e de Filtragem Adaptativa em Subbandas/Jéssica Bartholdy Sanson. – Rio de Janeiro: UFRJ/COPPE, 2016.

XIX, 80 p.: il.; 29, 7cm.

Orientador: Mariane Rembold Petraglia

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2016.

Referências Bibliográficas: p. 73 – 80.

1. Separação Cega de Fontes. 2. Misturas Convolutivas. 3. Filtragem Adaptativa. 4. Estruturas em Subbandas. 5. Sinais de Áudio. I. Petraglia, Mariane Rembold. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

*Dedico este trabalho a minha
família e meu namorado que me
deram todo apoio perante as
dificuldades durante este
percurso.*

Agradecimentos

Gostaria de agradecer primeiramente aos meus pais que me deram força e apoio para chegar até aqui. Ao meu namorado Luiz Henrique pelo apoio e compreensão em todos os momentos, mesmo durante os períodos que estive dedicada à esse trabalho.

Eu gostaria de expressar minha sincera gratidão à pessoa ao qual eu tive a sorte de ter como minha orientadora, Mariane Rembold Petraglia, que em todo esse tempo que estive sob sua orientação, sempre demonstrou paciência e dedicação comigo, sempre preferindo me incentivar ao invés de cobrar, e me ajudando todas as vezes que eu precisei.

Gostaria de agradecer também aos professores Amaro Azevedo de Lima e Fernando Gil Vianna Resende Junior que aceitaram participar da banca para a defesa desta dissertação. A Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro durante o mestrado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

NOVAS TÉCNICAS DE SEPARAÇÃO CEGA DE FONTES E DE FILTRAGEM ADAPTATIVA EM SUBBANDAS

Jéssica Bartholdy Sanson

Outubro/2016

Orientador: Mariane Rembold Petraglia

Programa: Engenharia Elétrica

Nesta dissertação são investigadas novas estruturas e algoritmos para a filtragem adaptativa e a separação de fontes. É proposto o emprego de filtragem em subbandas em algoritmos de separação cega de fontes (BSS) para misturas convolutivas no domínio do tempo e da frequência. Primeiramente revisitamos estudos realizados em filtragem adaptativa em subbandas para a separação supervisionada de fontes, o que nos permite avaliar as limitações do modelo que utiliza filtros FIR na separação cega de fontes para os diferentes cenários considerados neste trabalho. Em seguida, abordamos o tema de filtragem adaptativa para sistemas esparsos, e propomos uma forma de ajuste automático dos parâmetros do algoritmo l_0 -LMS. Dando continuidade aos estudos de separação cega de fontes, apresentamos o algoritmo TRINICON, e propomos modificações nas equações de atualização dos coeficientes do sistema de separação para realizá-las utilizando os sinais das misturas decompostos em subbandas, proporcionando uma melhora na taxa de convergência, maior razão sinal-interferência em regime estacionário e menor custo computacional em relação ao algoritmo em banda cheia. Por fim são apresentadas as modificações necessárias para utilizar o algoritmo TRINICON no domínio da frequência em subbandas, obtendo-se um novo método com desempenho ainda melhor do que o algoritmo TRINICON em subbandas no domínio do tempo, principalmente em ambientes com tempos de reverberação elevados.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

NEW TECHNIQUES OF BLIND SOURCE SEPARATION AND ADAPTIVE FILTERING IN SUBBANDS

Jéssica Bartholdy Sanson

October/2016

Advisor: Mariane Rembold Petraglia

Department: Electrical Engineering

In this dissertation new structures and algorithms in subbands for adaptive filtering and source separation are investigated. The use of subband filtering in blind source separation (BSS) for convolutive mixtures is proposed, in both time and frequency domains. First, we review previous studies on adaptive filtering in subbands for supervised source separation, which can be employed to evaluate the limitations of the model that employs FIR filters on the blind source separation for the different scenarios evaluated in this work. After that, we discuss the topic of adaptive filtering for sparse systems and propose a method for the automatic adjustment of the parameters of the l_0 -LMS algorithm. Proceeding with the studies on blind source separation, we present the TRINICON algorithm and propose changes to its update equations to carry them out in subbands, providing improved convergence rate, higher signal-interference ratio in steady-state and lower computational cost. Finally, the modifications necessary for using the frequency-domain TRINICON in subbands are presented, resulting in a new BSS method with better performance than the time-domain TRINICON in subbands, mainly for high reverberation time environments.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Símbolos	xiv
Lista de Abreviaturas	xviii
1 Introdução	1
1.1 Motivação	2
1.2 Objetivos	3
1.3 Estrutura da Dissertação	3
2 Filtragem Adaptativa	5
2.1 Solução Ótima de Wiener	7
2.2 Algoritmos Adaptativos	8
2.2.1 Algoritmo LMS	9
2.2.2 Algoritmo NLMS	10
2.2.3 Algoritmo NSAF	12
3 Conceitos de Separação Cega de Fontes	15
3.1 Definição do Problema	15
3.2 Modelo de Mistura Linear e Instantânea	17
3.3 Modelo de Misturas Convolutivas	19
3.4 Características do Sinal de Voz e sua Utilização na Separação Cega de Fontes	21
3.4.1 Propriedades básicas de sinais acústicos	22
3.4.2 Não-estacionário	22
3.4.3 Não-gaussianidade	23
3.4.4 Não-brancura	23
3.5 Ambiguidades na separação cega de fontes	24
3.6 Análise de Componentes Independentes	26

3.7	Medidas de Desempenho	27
3.7.1	Razão Sinal-Interferência	27
3.7.2	Razão Sinal-Ruído	28
3.7.3	Distorção do Sinal	29
4	Filtragem Adaptativa em Subbandas para Separação Supervisionada de Fontes	30
4.1	Algoritmo NLMS para o problema de separação de fontes	31
4.2	Algoritmo NSAF para o problema de separação de fontes	32
4.2.1	Modificação no algoritmo NSAF para o problema de separação	32
4.3	Sinal desejado para a avaliação da separação cega de fontes	33
4.4	Resultados das simulações	35
5	Estudo da Parametrização Automática de Algoritmo Adaptativo para Sistemas Esparsos	38
5.1	Introdução	38
5.2	Sistemas com resposta ao impulso esparsa	39
5.3	Algoritmo l_0 -LMS	40
5.4	Desempenho do algoritmo l_0 -LMS com relação aos parâmetros κ e β_{l_0}	41
5.5	Estimativa da potência de ruído	42
5.6	Simulações e Resultados	42
5.7	Comentários Finais	47
6	Separação Cega de Fontes no Domínio do Tempo	48
6.1	TRINICON	48
6.1.1	Notação Matricial para Misturas Convolutivas	49
6.1.2	Função Custo	50
6.1.3	Regra de atualização	52
6.2	BSS em Subbandas no Domínio do Tempo	54
6.3	Fatores de Aprendizagem Variantes	55
6.4	Simulações	56
7	Separação Cega de Fontes no Domínio da Frequência	59
7.1	Formulação Equivalente do algoritmo TRINICON no domínio DFT	59
7.1.1	Representação DFT da Matriz de Filtro de Separação $\mathbf{W}(b)$	61
7.1.2	Cálculo da matriz inversa $\mathbf{R}_{y_p y_q}^{-1}(b)$	62
7.1.3	Regra de atualização	63
7.2	BSS em Subbandas no Domínio da Frequência	64
7.3	Fatores de Aprendizagem	65
7.4	Simulações	65

7.5	Comparações Finais	67
8	Conclusões	71
8.1	Principais contribuições	71
8.2	Trabalhos Futuros	72
	Referências Bibliográficas	73

Lista de Figuras

2.1	Filtro Adaptativo.	6
2.2	Estrutura do algoritmo NSAF	13
3.1	Problema de separação cega de sinais.	16
3.2	Modelo MIMO para mistura convolutiva em BSS	21
3.3	Exemplos de distribuições supergaussiana (em verde), gaussiana (em vermelho) e subgaussiana (em azul).	24
3.4	Contribuição da fonte 1 na saída 2 para um cenário 2×2	28
4.1	Evolução da SIR(dB) para $T60 = 0.1$ s para os algoritmos NLMS, NSAF e NSAF modificado.	35
4.2	Evolução da SIR(dB) para $T60 = 0.25$ s para os algoritmos NLMS, NSAF e NSAF modificado.	36
4.3	Evolução da SIR(dB) para $T60 = 0.5$ s para os algoritmos NLMS, NSAF e NSAF modificado.	37
5.1	Resposta ao impulso da função de transferência de comprimento $L = 640$ utilizada nas simulações.	43
5.2	Desempenho do l_0 -LMS com $\kappa = 10^{-6}$, para SNR de 15 dB, comparado ao LMS.	44
5.3	Desempenho do l_0 -LMS com $\kappa = 10^{-6}$, para SNR de 28 dB, comparado ao LMS.	44
5.4	Curva de amostras e curva aproximada de valores ótimos de k para valores de SNR entre 5dB e 35dB.	45
5.5	Desempenho do l_0 -LMS com $k = 2.3686 \times 10^{-7}$ obtido pelo padrão proposto para SNR de 28 dB, comparado ao LMS.	45
5.6	Desempenho do l_0 -LMS com κ obtido pelo padrão proposto <i>versus</i> espasidade (ξ_w) para uma SNR de 10 dB e 15 dB, comparado ao LMS. 46	
6.1	Evolução da SIR (em dB) para $T60 = 0.1$ s para BSS no domínio do tempo em subbandas e em banda cheia..	57

6.2	Evolução da SIR (em dB) para $T60 = 0.3$ s para BSS no domínio do tempo em subbandas e em banda cheia.	58
6.3	Evolução da SIR(dB) para $T60 = 0.5$ s para BSS no domínio do tempo em subbandas e em banda cheia.	58
7.1	relação entre matriz circulante $\mathbf{C}_{\tilde{\mathbf{Y}}_q}(b)$ e a matriz Toeplitz $\tilde{\mathbf{Y}}_q(b)$. . .	61
7.2	Evolução da SIR (em dB) para $T60 = 0.1$ s para BSS no domínio da frequência.	67
7.3	Evolução da SIR (em dB) para $T60 = 0.5$ s para BSS no domínio da frequência.	68
7.4	Evolução da SIR (em dB) para $T60 = 0.3$ s para BSS no domínio do tempo e da frequência.	69
7.5	Evolução da SIR (em dB) para $T60 = 0.5$ s para BSS no domínio do tempo e da frequência.	70

Lista de Tabelas

6.1	Valores utilizados nas simulações	56
7.1	Valores da SIR (em dB) obtidos após a convergência dos algoritmos para $T_{60} = 0.3s$	68
7.2	Valores da SIR (em dB) obtidos após a convergência dos algoritmos para $T_{60} = 0.5 s$	68

Lista de Símbolos

$\text{bdiag}\{\cdot\}$	matriz de bloco-diagonal, p. 63
$\mathbf{G}_{PR \times PR}^{1N^0}$	matriz de restrição, p. 63
B	número total de blocos, p. 51
D	número de atrasos, p. 49
$F(z)$	filtro de análise, p. 14
L	tamanho dos blocos de saída, p. 49
M	número de subbandas, p. 14
N	comprimento do filtro adaptativo, p. 6
P	número de misturas/sensores, p. 15
Q	número de fontes, p. 15
SD	distorção do sinal, p. 29
SIR_{gain}	ganho de SIR, p. 28
SIR_x	SIR da mistura, p. 28
SIR_y	SIR da saída, p. 28
SNR_{gain}	ganho de SNR, p. 29
SNR_x	SNR da mistura, p. 29
SNR_y	SNR da saída, p. 29
$T60$	tempo de reverberação, p. 35
U	comprimento dos filtros FIR de mistura, p. 52
Z	domínio da transformada Z , p. 34

$\Delta \mathbf{W}$	função custo, p. 53
$\underline{\Delta \mathbf{W}}$	função custo no domínio DFT, p. 63
β_{l_0}	parâmetro do algoritmo l_0 -LMS, p. 40
δ	fator de regulagem, p. 12
\downarrow	decimado, p. 14
γ	fator de controle de magnitude, p. 40
$\hat{\mathbf{R}}$	matriz de correlação dos sinais de mistura, p. 32
$\hat{\mathbf{S}}$	matriz correlação inversa dos sinais de misturas, p. 33
$\hat{p}_D(\cdot)$	PDF de cada sensor de dimensão D , p. 51
$\hat{p}_{PD}(\cdot)$	PDF conjunta de dimensão PD , p. 51
κ	parâmetro referente ao termo penalizador algoritmo l_0 -LMS, p. 40
\mathbf{C}	matriz circulante, p. 60
\mathbf{F}_R	matriz DFT, p. 60
\mathbf{H}	matriz do canal de comunicação, p. 16
\mathbf{L}_I	matriz de conversão, p. 63
\mathbf{R}_x	matriz de correlação dos dados de entrada, p. 8
\mathbf{S}	matriz das fontes, p. 18
\mathbf{W}	matriz dos filtros de separação, p. 16
$\mathbf{W}_{R \times R}^{1N0}$	matrizes janelas cujo índice inferior denota as dimensões e o índice superior descreve as posições dos uns e zeros, p. 60
\mathbf{X}	matriz dos sinais recebidos nos sensores, p. 18
\mathbf{Y}	matriz das fontes estimadas, p. 19
$\underline{\mathbf{W}}$	matriz do filtro de separação no domínio DFT, p. 61
$\underline{\mathbf{Y}}$	matriz do sinal de saída no domínio DFT, p. 60
\mathbf{p}	vetor de correlação cruzada do sinal desejado, p. 8

$\mathbf{s}_i(n)$	sinal original gerado pela i -ésimo fonte, p. 16
$\mathbf{w}(n)$	vetor de coeficientes do filtro adaptativo, p. 6
$\mathbf{x}(n)$	vetor com amostras do sinal de entrada, p. 6
$\mathbf{x}_i(n)$	sinal obtido pelo i -ésimo sensor , p. 16
$\mathbf{y}_i(n)$	i -ésima estimativa de um método de separação cega de fontes, p. 16
\mathcal{J}	função de custo TRINICON, p. 51
μ	fator de adaptação do algoritmo, p. 10
μ	fator de convergência, p. 11
$\xi_{\mathbf{w}}$	medida de esparsidade, p. 39
“ $*$ ”	operador de convolução, p. 17
b	índice de bloco, p. 50
$d(n)$	sinal desejado, p. 6
$d_{m\downarrow}$	sinal desejado decomposto em subbandas e decimado, p. 14
$e(n)$	erro, p. 6
$e_{m\downarrow}$	erro do sinal decomposto em subbandas e decimado, p. 14
h_{ij}	coeficientes da matriz do canal que relaciona a j -ésima fonte com a i -ésima mistura, p. 17
w_{ji}	coeficientes da matriz dos filtros de separação que relaciona a j -ésima fonte com a i -ésima mistura, p. 17
$x_{c,p}(n)$	componente da p -ésima mistura decorrente das outras fontes interferentes, p. 27
$x_{s_r,p}(n)$	componente da mistura decorrente da fonte desejada, p. 27
$y(n)$	sinal de saída, p. 6
$y_{c,q}(n)$	componente de interferência no canal de saída, p. 27
$y_m(n)$	sinal de saída estimado decomposto em subbandas e decimado, p. 14

$y_{s_r,q}(n)$	componente da saída que contém a fonte desejada, p. 27
$y_{v,q}$	contribuições do ruído de fundo na saída, p. 27
$\text{Diag}\{\cdot\}$	matriz quadrada com os elementos do vetor “ $\{\cdot\}$ ” em sua diagonal principal, p. 60

Lista de Abreviaturas

l_0 -LMS	<i>Norm Constraint Least-Mean Squares</i> , p. 38
BASS	<i>Blind Audio Source Separation</i> , p. 30
BSS	<i>Blind Source Separation</i> , p. 15
CASA	<i>Computational Auditory Scene Analysis</i> , p. 30
DFT	<i>Discrete Fourier Transform</i> , p. 29
DSP	Dispositivo para Processamento de Sinais, p. 54
EQM	Erro Quadrático Médio, p. 6
FIR	<i>Finite Impulse Response</i> , p. 5
ICA	<i>Independent Component Analysis</i> , p. 26
IIR	<i>Infinite Impulse Response</i> , p. 5
KLD	Distância de <i>Kullback-Leibler</i> , p. 51
LMS	<i>Least-Mean Squares</i> , p. 8
LMS	<i>Normalized Subband Adaptive Filter</i> , p. 12
MIMO	<i>Multiple-Input Multiple-Output</i> , p. 21
MLI	Mistura Linear e Instantânea, p. 18
NLMS	<i>Normalized Least-Mean Squares</i> , p. 8
RLS	<i>Recursive Least Square</i> , p. 8
SAF	<i>Subband Adaptive Filter</i> , p. 13
SD	<i>Sparse Decomposition</i> , p. 30
SIR	<i>Signal-to-Interference Ratio</i> , p. 28

SNR *Signal-to-Noise Ratio*, p. 29

pdf *Probability Density Function*, p. 22

Capítulo 1

Introdução

A maioria dos sinais de áudio capturados por microfones correspondem a misturas de sinais de várias fontes, como, por exemplo, sinais de fala, música e ruídos ambientes e de equipamentos. Separação de fontes consiste na recuperação dos sinais originais das fontes de interesse a partir de um ou mais sinais de mistura. Aplicações diretas incluem teleconferências, palestras em tempo real com tradução simultânea e amostragem de sons para composição de música eletrônica.

Já muitas aplicações derivadas visam modificar o sinal de mistura, como, por exemplo, no aprimoramento do sinal da fala dentro de aparelhos auditivos, cancelamento de voz para karaokês e renderização de áudio para dispositivos multicanal a partir de gravações em CD estéreo. Em algumas aplicações, são conhecidos trechos de sinais originais presentes nas misturas antes de se fazer a separação. Nesses casos, pode-se usar algoritmos adaptativos supervisionados.

Nos últimos anos, grandes progressos foram feitos no campo das interfaces acústicas, tanto em pesquisa básica, quanto no desenvolvimento de produtos. Nessa área muito esforço é dedicado a equipamentos de multimídia ou de telecomunicações, que têm de ser projetados para operar em vários cenários diferentes, devido à ampla gama de aplicações.

Em tais aplicações, os algoritmos de processamento de sinal digital visam à estimativa de um sinal de fonte desejado, que pode conter várias fontes pontuais interferentes, como sinais de alto-falantes ou ruído de fundo, tais como ruído de carro ou da fala de pessoas em ambientes lotados. Também em ambientes com paredes ou obstáculos podem existir reflexões dos sinais de interesse e dos sinais de interferência, que captados em conjunto com o sinal desejado complicam significativamente o problema da recuperação do sinal fonte desejado. Tais sinais misturados são chamados de misturas convolutivas. Nessas situações, as fontes são os sinais desejados, mas apenas as gravações das fontes misturadas estão disponíveis e o processo de mistura é desconhecido.

1.1 Motivação

Até poucos anos atrás, a maioria das interfaces acústicas oferecia apenas um microfone para aquisição do sinal de áudio, o que restringiu as abordagens para recuperar o sinal de fonte desejado para algoritmos de canal único, como em [1]. Mesmo agora, esse tópico continua a ser um campo importante da investigação, como pode ser visto, por exemplo, em [2].

No entanto, hoje em dia, devido aos custos de hardware mais baratos, os fabricantes também começaram a acomodar microfones adicionais nos seus produtos e, assim, permitir a aplicabilidade dos algoritmos de processamento multicanal, os quais serão abordados neste trabalho.

As versões mais recentes de algoritmos de separação cega de fontes têm utilizado sinais capturados por arranjos de microfones para a separação dos sons chegando de diferentes direções por meio da adaptação de filtros de separação. As razões sinal-interferência resultantes consideravelmente altas mostram a eficácia deste tipo de processamento. Porém, esses algoritmos possuem alguns desafios.

Muitas aplicações atuais exigem que os sensores para captar os sinais de voz estejam a uma distância considerável da fonte em ambientes com obstáculos e paredes que refletem as ondas sonoras, como, por exemplo, em telefones com vídeo, em que os microfones são geralmente colocados perto da câmera. Outro exemplo é no controle de voz em casas automatizadas ou em dispositivos móveis, cujo uso vem crescendo a cada dia e os usuários em geral não desejam estar com o aparelho a uma distância extremamente pequena do emissor e/ou em ambientes silenciosos para o mesmo funcionar corretamente. Enfim, essas interfaces de voz distantes da fonte sonora introduzem uma série de desafios adicionais.

Um dos principais fatores para o mau desempenho em reconhecimento de voz em tais aplicações é a reverberação. Altos tempos de reverberação provocam alterações temporais nas características da fala, que afetam tanto a inteligibilidade quanto o desempenho do reconhecimento automático da fala [3].

Apesar de muitos algoritmos terem sido propostos para ambientes com alta reverberação, até o momento nenhum obteve resultados satisfatórios. Estudos para melhoria da separação cega de fontes nesses casos mostram-se ainda necessários. Com o objetivo de tentar resolver esse problema, aplicaremos neste trabalho a filtragem em subbandas no algoritmo proposto em [4], tanto no domínio do tempo como no da frequência. Uma melhora considerável na qualidade da separação é obtida, sendo essa a principal contribuição desta dissertação.

1.2 Objetivos

O principal objetivo deste trabalho é realizar um estudo e análise de técnicas de processamento de sinais em filtragem adaptativa e em separação cega de fontes, que servirão como ferramentas para o desenvolvimento de novos métodos.

Nesta dissertação novas técnicas de filtragem adaptativa e de separação cega de fontes são propostas com a finalidade de se conseguir melhores desempenhos do que os algoritmos atualmente encontrados na literatura, principalmente para situações desafiadoras, como na recuperação de sinais de voz e no cancelamento de eco acústico em ambientes com alta reverberação.

Utilizando-se técnicas de processamento em subbandas, obtém-se algoritmos que apresentam maiores taxas de convergência e razões sinal-interferência (SIR) em regime estacionário, e menor complexidade computacional devido ao tamanho de filtro reduzido de cada subbanda do sistema de separação.

Dois novos métodos de baixa complexidade serão derivados, aplicando-se algoritmos de separação cega de fontes já consagrados na literatura no processamento em subbandas, sendo que um deles opera no domínio do tempo e o outro no domínio da frequência.

1.3 Estrutura da Dissertação

O trabalho apresentado nesta dissertação está estruturado da seguinte forma: no Capítulo 2 apresentaremos conceitos básicos de filtragem adaptativa, e alguns algoritmos já bastante conhecidos nas literaturas. No Capítulo 3, introduziremos o modelo BSS. Depois de descrever brevemente o caso mais simples, dado pelo modelo BSS instantâneo, nos concentraremos no modelo BSS convolutivo, que considera o fato de que em ambientes acústicos também há reflexos dos sinais das fontes originais capturados pelos sensores.

Em seguida, as propriedades dos sinais de fontes sonoras utilizadas em abordagens BSS são examinadas e as ambiguidades que surgem devido à cegueira dos métodos BSS são abordadas. Com base nesses dois temas, apresentaremos no Capítulo 4 estudos já realizados em filtragem adaptativa em subbandas para separação supervisionada de fontes, podendo através deles avaliar as limitações na separação cega de fontes para os diferentes cenários que serão avaliados neste trabalho. No Capítulo 5, abordaremos o tema de filtragem adaptativa para sistemas esparsos, onde um estudo da parametrização automática de um algoritmo adaptativo desenvolvido para sistemas esparsos é realizado.

Com base no modelo BSS convolutivo, uma estrutura genérica para o BSS em ambientes reverberantes é introduzida no Capítulo 6. A otimização da estrutura

BSS é realizada através da formulação de um critério de otimização no domínio do tempo de banda larga genérico. Um algoritmo baseado no gradiente natural é então apresentado e são introduzidas certas aproximações. Em seguida, é proposto um novo método BSS em subbandas, que apresenta melhor desempenho do que o método em banda única. Resultados experimentais obtidos com os algoritmos BSS estudados e propostos em ambientes com variados tempos de reverberação são apresentados.

No Capítulo 7, mostra-se como algoritmos de banda larga podem ser implementados de forma eficiente usando a Transformada de Fourier Discreta (DFT). Esses algoritmos se comportam de forma equivalente aos algoritmos correspondentes no domínio do tempo e, portanto, não exibem as ambiguidades dos algoritmos BSS de banda estreita, que efetuam a separação independentemente em cada saída da DFT. Além disso, mostra-se que através da introdução da decomposição em subbandas proposta nesta dissertação consegue-se alcançar melhores resultados em ambientes com tempos reverberação altos.

Finalmente, são apresentadas no Capítulo 8 as conclusões desta dissertação e algumas sugestões para trabalhos futuros.

Capítulo 2

Filtragem Adaptativa

Devido aos poderosos processadores digitais de sinais e o desenvolvimento de algoritmos adaptativos avançadas há um grande número de diferentes aplicações nas quais são utilizados filtros adaptativos. O número de diferentes aplicações nas quais técnicas adaptativas estão sendo usadas com sucesso aumentou enormemente durante as últimas décadas, e com os avanços dessa teoria junto das melhorias na área de tecnologia de circuitos integrados, filtragem adaptativa vem se tornando cada vez mais sofisticada [5].

Um filtro adaptativo pode ser entendido como um filtro digital cujos coeficientes são ajustados de modo a minimizar uma função de erro. Esta função de erro, também referida como função de custo, é uma medida da distância entre o sinal desejado e a saída do filtro adaptativo. Devido à complexidade dos algoritmos de otimização utilizados para a atualização dos seus coeficientes, a grande maioria dos filtros adaptativos são filtros digitais.

Algoritmos adaptativos podem ser necessários tanto na filtragem convencional do tipo resposta impulsiva finita (FIR) como do tipo resposta impulsiva infinita (IIR) para aplicações em que os parâmetros da operação de processamento desejado não são conhecidos antecipadamente ou estão mudando. Devido à função de custo ser quadrática e não ser necessário o monitoramento de polos (que estão todos na origem do plano Z) para filtros FIR, estes são preferidos na quase totalidade das aplicações.

De acordo com [6] um sistema adaptativo como mostrado na Figura 2.1 é um filtro digital que realiza o processamento de sinais e busca adaptar sua resposta às condições encontradas no meio em que o mesmo opera.

Os vetores de entrada e de coeficientes de um filtro adaptativo FIR podem ser definidos por:

$$\mathbf{x}(n) = \begin{bmatrix} x(n) & x(n-1) & \dots & x(n-N+1) \end{bmatrix}^T \quad (2.1)$$

$$\mathbf{w}(n) = \begin{bmatrix} w_0(n) & w_1(n) & \dots & w_{N-1}(n) \end{bmatrix}^T \quad (2.2)$$

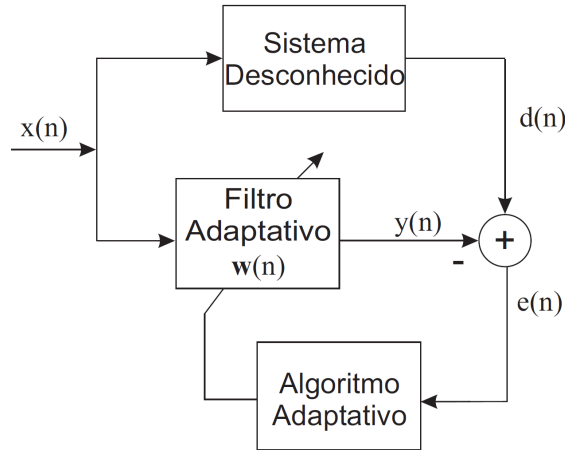


Figura 2.1: Filtro Adaptativo.

onde N é o comprimento do filtro adaptativo. Portanto, temos a seguinte saída e erro:

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n) \quad (2.3)$$

$$e(n) = d(n) - y(n) \quad (2.4)$$

Em muitas aplicações que requerem filtragem, a resposta do sistema em frequência pode variar com o tempo ou então não ser conhecida. Para situações como essas um filtro fixo não conseguiria ser eficaz, pois nele os coeficientes permanecem constantes e não seria possível solucionar o problema quando houvesse variação na resposta do sistema; já os filtros adaptativos, capazes de efetuar auto ajuste e rastrear as variações que o processo analisado sofre, mostram-se de extremamente eficientes. Devido a isso os filtros adaptativos são amplamente utilizados em uma grande variedade de aplicações.

A eficiência dos filtros adaptativos depende, principalmente, da técnica de criação utilizada e do algoritmo adaptativo, sendo esse último responsável pelo mecanismo de aprendizagem do filtro, que ajusta de maneira recursiva seus coeficientes em busca a minimizar uma função custo determinada, geralmente o erro quadrático médio (EQM).

A função custo é um critério para um melhorar o desempenho do filtro e para alinhar um algoritmo que determina a forma de modificar a função de transferência do filtro minimizando o custo da iteração seguinte. Com base em tal função custo, o algoritmo faz com que os coeficientes do filtro convirjam, iterativamente, para a solução ótima no sentido médio quadrático do erro.

Os filtros adaptativos podem ser analógicos, digitais ou mistos que possuem suas vantagens e desvantagens, por exemplo, os filtros analógicos são de baixo consumo de energia e tem resposta rápida, porém apresentam problemas de deslocamento, que afetam o funcionamento do algoritmo de adaptação. Os filtros digitais podem

trabalhar com algoritmos mais complexos e possuem maior precisão.

Também os filtros adaptativos podem ser uma combinação de diferentes tipos de filtros, como filtros de entrada única ou multi-entrada, filtros IIR ou FIR, linear ou não, etc. Além disso, um filtro adaptativo, devido à sua característica de auto-ajuste em tempo real, é esperado que seja operado para controlar o comportamento ótimo de um ambiente de variação lenta ou estacionário. A estrutura do filtro a ser utilizada e o algoritmo empregado em sua adaptação constituem os problemas fundamentais na filtragem adaptativa [5].

Os filtros adaptativos podem ser aplicados em distintas áreas: identificação de sistemas, engenharia biomédica, comunicações, previsão, etc.

2.1 Solução Ótima de Wiener

Os coeficientes de um filtro de Wiener são calculados para minimizar o quadrado da distância média entre a saída do filtro e um sinal desejado, o erro médio quadrático. Na sua forma básica, a teoria de Wiener assume que os sinais são processos estacionários. Um filtro de Wiener pode ser um filtro do tipo IIR ou do tipo FIR. Em geral, a formulação de um tipo IIR filtrar os resultados de um conjunto de equações não lineares, enquanto que a formulação de um filtro Wiener FIR resulta em um conjunto de equações lineares e tem uma solução de forma fechada.

O objetivo da solução do Wiener é fornecer os coeficientes ótimos \mathbf{w}_0 , também chamado ponto ótimo. Porém esse ponto ótimo para uma solução em tempo real é algo inalcançável, uma vez que para a utilização da solução de Wiener é preciso conhecermos antecipadamente a matriz de correlação dos dados de entrada \mathbf{R}_x e vetor de correlação cruzada do sinal desejado \mathbf{p} , conforme observado nas equações abaixo [7]:

$$\mathbf{w}_0 = \mathbf{R}_x^{-1} \mathbf{p}, \quad (2.5)$$

A matriz de auto-correlação \mathbf{R}_x dos dados de entrada $\mathbf{x}(n)$ para a sistema de Wiener é dada por:

$$\mathbf{R}_x = E\{\mathbf{x}(n)\mathbf{x}(n)^T\}, \quad (2.6)$$

e o vetor de correlação cruzada do sinal desejado $d(n)$ e do vetor de entrada $\mathbf{x}(n)$ é dado por :

$$\mathbf{p} = E\{d(n)\mathbf{x}(n)\}, \quad (2.7)$$

O filtro de Wiener é insuficiente para lidar com situações em que não há estacionariedade do sinal. Em tais situações, o filtro ideal tem de assumir uma forma

variável no tempo. A finalidade do algoritmo adaptativo é ajustar os coeficientes do filtro para minimizar o valor quadrático médio do sinal de erro $e(n)$ que é definido como a diferença entre uma resposta desejada $d(n)$ e a saída do filtro real $\mathbf{y}(n)$. O comportamento do filtro, sua complexidade computacional e tempo de resposta dependem diretamente do algoritmo adaptativo implementado no filtro [8].

2.2 Algoritmos Adaptativos

Um algoritmo adaptativo é um algoritmo que muda seu comportamento com base nas informações disponíveis no momento em que é executado. Isso pode ser informações sobre recursos computacionais disponíveis ou dados recebidos. Muitos algoritmos são adaptáveis ou ter variantes de adaptação, o que geralmente significa que os parâmetros dos algoritmos são automaticamente ajustadas de acordo com estatísticas sobre a otimização, como por exemplo a taxa de convergência.

Os algoritmos adaptativos mais comuns para filtragem adaptativa que podemos citar, entre os diversos algoritmos existentes na literatura são, o *Least-Mean Square* (LMS), o *Normalized-LMS* (NLMS) e o *Recursive Least Square* (RLS), entre outros. Devido à simplicidade computacional, o algoritmo de LMS é mais citado e usado nas derivações de novos algoritmos em filtragem adaptativa. A aplicação de cada algoritmo depende da configuração do filtro adaptativo utilizado. Configurações clássicas de filtragem adaptativa que podemos citar são, identificação do sistema, previsão, cancelamento de ruído, entre outras. As diferenças entre as configurações são dadas pela forma como a entrada, o sinal desejado e os sinais de saída são usados.

Na filtragem adaptativa mesmo quando o conhecimento das estatísticas do sinal e do ruído não estão disponíveis, ainda é possível desenvolver um filtro útil usando um algoritmo adaptativo para ajustar os parâmetros do filtro com base no fluxo de dados de entrada. Se as estatísticas do sinal e ruído são estacionários, é esperado que o filtro adaptável eventualmente convirja para a solução ótima de Wiener apresentada na seção anterior. A taxa de adaptação deve ser mais rápida do que a taxa de mudança de estatísticas para manter o bom funcionamento do algoritmo.

Os parâmetros de um filtro adaptável são atualizados continuamente como os dados que fluem através dele. Por conseguinte, o filtro adaptativo é estritamente um sistema não linear. No entanto, é comum a distinção entre filtros adaptativos lineares e não-lineares. Um filtro adaptativo linear é um cuja saída é uma combinação linear da entrada real em qualquer momento no tempo entre as operações de adaptação.

As principais características de um algoritmo adaptativo são [8]:

- Taxa de convergência: definida como o número de iterações necessárias para o algoritmo atingir o regime permanente, ou seja, sua função de custo convergir para o mínimo;

- Desajuste: diferença entre o erro médio quadrático obtido pelo algoritmo em regime permanente e o erro médio quadrático mínimo (produzido pelo filtro de Wiener). Em geral é apresentado como um percentual em relação ao erro médio quadrático mínimo;
- *Tracking*: capacidade do algoritmo de acompanhar variações das estatísticas dos sinais em meios não estacionários;
- Robustez: sensibilidade da solução do algoritmo a perturbações de fontes internas ou externas;
- Complexidade Computacional: número de operações computacionais necessárias por iteração, e requisitos de armazenamento de dados e de programação. Pode determinar a viabilidade da implementação do algoritmo em tempo real e o consumo de potência;
- Estabilidade numérica: comportamento do algoritmo quando sujeito a erros de quantização.

2.2.1 Algoritmo LMS

O algoritmo LMS é caracterizado por sua simplicidade e robustez, sendo o algoritmo mais popular utilizado em filtragem adaptativa. A popularidade do algoritmo LMS é devida, principalmente, à sua robustez numérica e baixa complexidade computacional, características extremamente atrativas em aplicações práticas. Por outro lado, ele tem como principal desvantagem a forte dependência de sua velocidade de convergência com o grau de correlação do sinal de entrada. Sinais de entrada que apresentem uma elevada dispersão de autovalores de sua matriz de autocorrelação, como, por exemplo, sinais de voz, podem acarretar séria degradação na velocidade de convergência do algoritmo [9].

Pertencente à classe dos algoritmos baseados no gradiente estocástico, o algoritmo LMS pode ser obtido pela substituição do vetor gradiente real por uma estimativa ruidosa, baseada em informações instantâneas do sistema, visando sua convergência (na média) para a solução de Wiener. Dessa forma, a utilização do algoritmo LMS não requer conhecimento prévio das estatísticas dos sinais envolvidos.

O algoritmo LMS é utilizado amplamente em identificação de sistemas. A derivação deste algoritmo fundamenta-se na regra de gradiente descendente aplicada na minimização do erro quadrático instantâneo, sendo o erro obtido pela diferença entre a resposta desejada (ou de referência) e a resposta atual do sistema adaptativo [10].

O sinal de referência neste trabalho é denotado por $d(k)$, cumpre ao algoritmo adaptativo reduzir a discrepância entre a saída do sistema desconhecido e a do

filtro adaptativo. O algoritmo LMS atualiza os coeficientes adaptativos por meio do gradiente do erro quadrático instantâneo [11]. Para a solução do algoritmo LMS temos que o erro e a atualização dos coeficientes são dados, respectivamente, por [8] e [12]:

$$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n) \quad (2.8)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu e(n)\mathbf{x}(n) \quad (2.9)$$

onde μ é o passo (ou fator) de adaptação do algoritmo, cuja escolha deve contemplar o inevitável compromisso entre taxa de convergência e desempenho em regime permanente, bem como questões de estabilidade [8].

O algoritmo LMS convencional opera com passo de adaptação fixo. O valor do passo de adaptação é determinante para o desempenho de convergência do filtro adaptativo. Observa-se que quanto maior for o valor do passo de adaptação, desde que dentro do limite de estabilidade, maior será a velocidade de convergência do algoritmo, o que é desejável; porém, maior será também o erro em excesso em regime permanente. Já um valor de passo de adaptação reduzido proporciona desajustes menores em regime permanente, ao preço de uma convergência mais lenta. Dessa forma, a escolha do valor do passo de adaptação deve levar em conta o compromisso existente entre velocidade de convergência e desajuste em regime permanente. Em aplicações envolvendo o algoritmo LMS, é usualmente desejável que o filtro adaptativo apresente simultaneamente alta velocidade de convergência e reduzido desajuste em regime permanente [9].

Como no algoritmo LMS cada coeficiente em (2.9) é atualizado usando o mesmo passo de aprendizagem, não é levado em consideração o fato de que em sistemas esparsos a atração de coeficientes para zero pode ser interessante para atenuar flutuações desnecessárias em torno dos coeficientes nulos, o que permite incrementar o desempenho do sistema adaptativo de identificação.

2.2.2 Algoritmo NLMS

O algoritmo NLMS (*Normalized Least-Mean Squares*) surgiu como um avanço do LMS (*Least-Mean Squares*) com um passo de adaptação normalizado para minimizar o problema que o algoritmo LMS sofre de amplificação de ruído do gradiente quando o tamanho do vetor de entrada é muito grande, tornando-o imune à variação de potência na entrada [8].

Como características do algoritmo NLMS pode-se citar que ele:

- Possui maior velocidade de convergência que o LMS;

- É um pouco mais complexo que o LMS, mas ainda assim possui baixa complexidade, se comparado com outros algoritmos apresentados na literatura;
- É imune a alterações da potência do sinal de entrada, pois normaliza a entrada do sinal;
- E assim como o LMS apresenta baixa taxa de convergência para sinais de entrada correlacionados.

A única distinção entre o algoritmo LMS e NLMS reside na normalização do passo de adaptação pela norma euclidiana ao quadrado do vetor de sinal de entrada. Tal procedimento reduz a sensibilidade do algoritmo a variações na potência do sinal de entrada, tornando assim o NLMS atrativo para aplicações como o cancelamento adaptativo de ruído e o cancelamento de eco de rede. A sua robustez também pode ser verificada para sinais altamente correlacionados, nos quais o efeito da normalização se reflete através da redução da dispersão de autovalores da matriz de autocorrelação do sinal de entrada [13]. A equação de adaptação do algoritmo NLMS é dada por:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\mu_n e(n)\mathbf{x}(n) = \mathbf{w}(n) + \Delta\tilde{\mathbf{w}}(n) \quad (2.10)$$

onde μ_n é o fator de convergência variável do algoritmo.

Partindo da estratégia de minimizar o erro quadrático instantâneo [12], temos que este erro é dado por:

$$e^2(n) = d^2(n) + \mathbf{w}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n) - 2d(n)\mathbf{w}^T(n)\mathbf{x}(n) \quad (2.11)$$

Aplicando uma mudança no vetor de coeficientes do filtro tal que $\tilde{\mathbf{w}}(n) = \mathbf{w}(n) + \Delta\tilde{\mathbf{w}}(n)$, o erro quadrático com os coeficientes $\tilde{\mathbf{w}}(n)$ passa a ser dado por:

$$\begin{aligned} \tilde{e}^2(n) = e^2(n) + 2\Delta\tilde{\mathbf{w}}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n) + \Delta\tilde{\mathbf{w}}^T(n)\mathbf{x}(n)\mathbf{x}^T(n)\Delta\tilde{\mathbf{w}}(n) \\ - 2d(n)\Delta\tilde{\mathbf{w}}^T(n)\mathbf{x}(n) \end{aligned} \quad (2.12)$$

Substituindo $d(n)$ por $d(n) = e(n) + \mathbf{w}^T(n)\mathbf{x}(n)$ e $\Delta\tilde{\mathbf{w}}(n)$ por $\Delta\tilde{\mathbf{w}}(n) = 2\mu_n e(n)\mathbf{x}(n)$, temos que:

$$\begin{aligned} \Delta e^2(n) &\triangleq \tilde{e}^2(n) - e^2(n) \\ &= -4\mu_n e^2(n)\mathbf{x}^T(n)\mathbf{x}(n) + 4\mu_n^2 e^2(n)[\mathbf{x}^T(n)\mathbf{x}(n)]^2 \end{aligned} \quad (2.13)$$

Derivando a equação acima em relação a μ_n e igualando a zero, para minimizar a função $\Delta e^2(n)$, encontramos o valor de μ_n :

$$\mu_n = \frac{1}{2\mathbf{x}^T(n)\mathbf{x}(n)} \quad (2.14)$$

Utilizando μ_n na equação (2.10), a atualização dos coeficientes passa a ser dada por:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{e(n)\mathbf{x}(n)}{\mathbf{x}^T(n)\mathbf{x}(n)} \quad (2.15)$$

Normalmente, um fator de convergência fixo μ também é introduzido para controlar o desalinhamento e um parâmetro δ para evitar passos muito grandes quando $\mathbf{x}^T(n)\mathbf{x}(n)$ se torna muito pequeno. Tais situações ocorrem quando a potência do sinal de entrada é muito pequena, como, por exemplo, em períodos de silêncio no processamento de um sinal de fala. Apesar de o parâmetro de regularização ser escolhido independentemente do passo de adaptação, é sabido que ele afeta o erro em regime permanente, devendo assim ser cuidadosamente escolhido. Assim, a equação final para atualização dos coeficientes é dada por:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu}{\delta + \mathbf{x}^T(n)\mathbf{x}(n)} e(n)\mathbf{x}(n) \quad (2.16)$$

Os valores adequados para escolha de μ são [12]:

$$0 \leq \mu \leq 2 \quad (2.17)$$

2.2.3 Algoritmo NSAF

O algoritmo NLMS converge lentamente quando os sinais de entrada são coloridos. Para resolver este problema, uma abordagem atraente é utilizar filtragem em subbandas, na qual o sinal de entrada colorido é decomposto em faixas de frequência quase mutuamente exclusivas e o sinal decimado de cada subbanda é aproximadamente branco [14].

O algoritmo NSAF (*Normalized Subband Adaptive Filter*) proposto em [15], decompõe o sinal de entrada e o sinal desejado em subbandas, permitindo explorar as propriedades de cada sinal resultante. Este algoritmo utiliza o princípio da perturbação mínima, em que, de uma iteração para a seguinte, os coeficientes do filtro adaptativo devem ser alterados de forma mínima, sujeito a restrições impostas aos erros em subbandas após a atualização (erros *a posteriori* em subbandas nulos). Como a atualização dos coeficientes é feita em uma taxa mais baixa, decimando-se os sinais de erro, o aumento no custo computacional é muito pequeno [16] quando comparado ao do algoritmo NLMS, mas a convergência dos coeficientes se dá de forma mais rápida para sinais de entrada coloridos.

Uma característica única do algoritmo NSAF é que o sinal de erro é calculado em subbandas, porém a atualização dos coeficientes do filtro adaptativo é feita em banda completa, conforme ilustrado na Figura 2.2. Esse procedimento faz com que o NSAF difira de seu algoritmo antecessor SAF (*Subband Adaptive Filter*) [17], que emprega adaptação e subfiltros distintos nas diferentes subbandas.

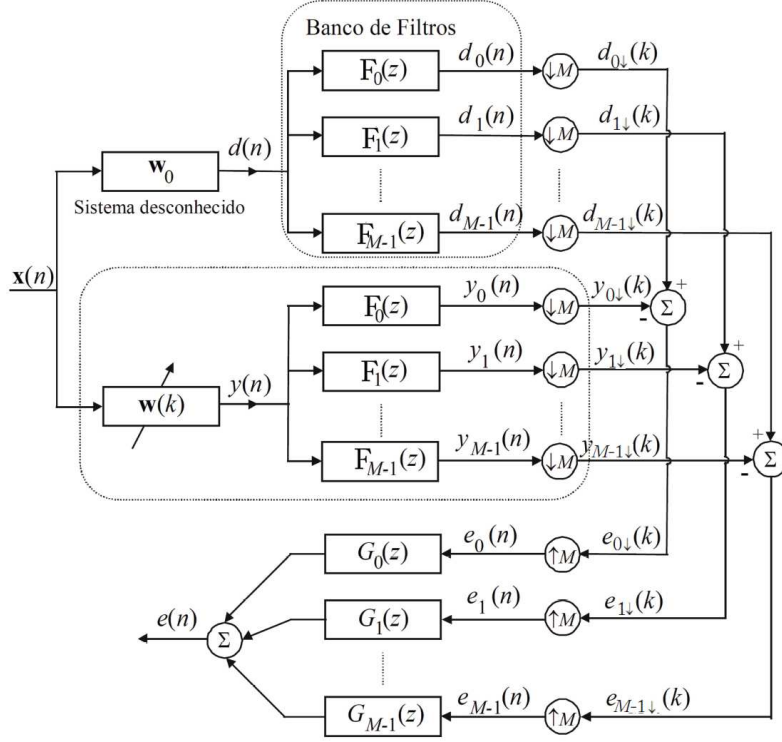


Figura 2.2: Estrutura do algoritmo NSAF

No algoritmo NSAF, o sinal desejado $d(n)$ e a saída do filtro $y(n)$ são decompostos em M subbandas pelos filtros de análise $F_0(z), \dots, F_{M-1}(z)$ e criticamente decimados (fator de decimação igual ao número de subbandas). Assim, o sinal de saída decimado da m -ésima subbanda é dado por

$$y_{m\downarrow}(n) = \sum_{l=0}^{N-1} w_l(n) x_m(nM - l) = \mathbf{w}^T(n) \mathbf{x}_m(n) \quad (2.18)$$

onde

$$\mathbf{x}_m(n) = \begin{bmatrix} x_m(nM) & x_m(nM - 1) & \dots & x_m(nM - N + 1) \end{bmatrix}^T \quad (2.19)$$

$$\mathbf{w}(n) = \begin{bmatrix} w_0(n) & w_1(n) & \dots & w_{N-1}(n) \end{bmatrix}^T \quad (2.20)$$

O sinal de erro correspondente é dado por

$$e_{m\downarrow}(n) = d_{m\downarrow}(n) - \mathbf{w}^T(n) \mathbf{x}_m(n) \quad (2.21)$$

onde $d_{m\downarrow}(n)$ é o sinal desejado decimado da m -ésima subbanda. A equação de atualização dos coeficientes do algoritmo NSAF é dada por

$$\hat{\mathbf{w}}(n+1) = \mathbf{w}(n) + \mu \sum_{m=0}^{M-1} \frac{\mathbf{x}_m(n)}{\delta + \|\mathbf{x}_m(n)\|^2} e_{\downarrow m}(n) \quad (2.22)$$

onde os parâmetros μ e δ foram introduzidos com os mesmos objetivos descritos para o algoritmo NLMS.

Capítulo 3

Conceitos de Separação Cega de Fontes

A seguir descreveremos o problema de separação cega de fontes e em seguida apresentaremos os diferentes modelos de mistura que são encontrados em várias aplicações. Subsequentemente, as características do sinal de voz e suas utilizações em separação cega de fontes serão discutidas. Além disso, as ambiguidades na separação cega de fontes e seus possíveis problemas para algoritmos BSS serão explicados. Apresentaremos ainda uma breve introdução da metodologia de Análise de Componentes Independentes para o problema de BSS. No final deste capítulo uma visão geral sobre as medidas de desempenho que são úteis na avaliação de algoritmos de BSS em razão ao desempenho da separação e qualidade do sinal é dada.

3.1 Definição do Problema

Na separação cega de fontes (BSS – *Blind Source Separation*) uma situação geralmente considerada é que há uma série de sinais emitidos por algumas fontes físicas. Essas fontes podem ser, por exemplo, vários oradores na mesma sala. Além disso, é assumido que existem vários sensores que estão localizados em posições diferentes. Portanto, cada sensor adquire uma mistura ligeiramente diferente dos sinais das fontes originais.

O objetivo da separação cega de fontes é recuperar os sinais de fonte original a partir dos sinais deste conjunto de sensores. O termo “cego” salienta o fato de que os sinais de fonte e o sistema de mistura são assumidos como desconhecidos. O problema da separação cega de fontes consiste em recuperar Q sinais originais, apenas a partir do conhecimento de P misturas destes. Trechos isolados de cada fonte são, nessa configuração, inacessíveis, sendo portanto uma estratégia não-supervisionada.

O pressuposto fundamental necessário à aplicação de métodos BSS é que os sinais de fonte originais são estatisticamente independentes. Dentro dessa realidade, este pressuposto é válido para uma grande quantidade de sinais, tais como sinais de vários alto-falantes. Portanto, o problema da BSS refere-se a encontrar um sistema de misturas cujas saídas são estatisticamente independente.

O mecanismo para identificar os filtros de separação é frequentemente derivado com base no conceito de minimizar a informação mútua entre os sinais de saída. O problema clássico que introduz a BSS é o problema “cocktail party” [18], no qual existem múltiplos oradores e ouvintes simultâneos e o objetivo é extrair um sinal de voz desejado a partir da mistura.

A estrutura do problema de BSS é apresentada na Figura 3.1 onde se considera que Q sinais, $\mathbf{s}(n) = [s_1(n), s_2(n), \dots, s_Q(n)]^T$, estão sendo emitidos por algumas fontes físicas e transmitidos através de um canal \mathbf{H} , considerando que existem vários sensores, receptores ou antenas, em diferentes posições que medem e gravam misturas dos sinais das fontes. No canal de transmissão, além de misturados, os sinais originais sofrem o efeito do ruído e de interferências $v(n)$. Os P sinais misturados que são captados pelos sensores são representados por $\mathbf{x}(n) = [x_1(n), x_2(n), \dots, x_P(n)]^T$.

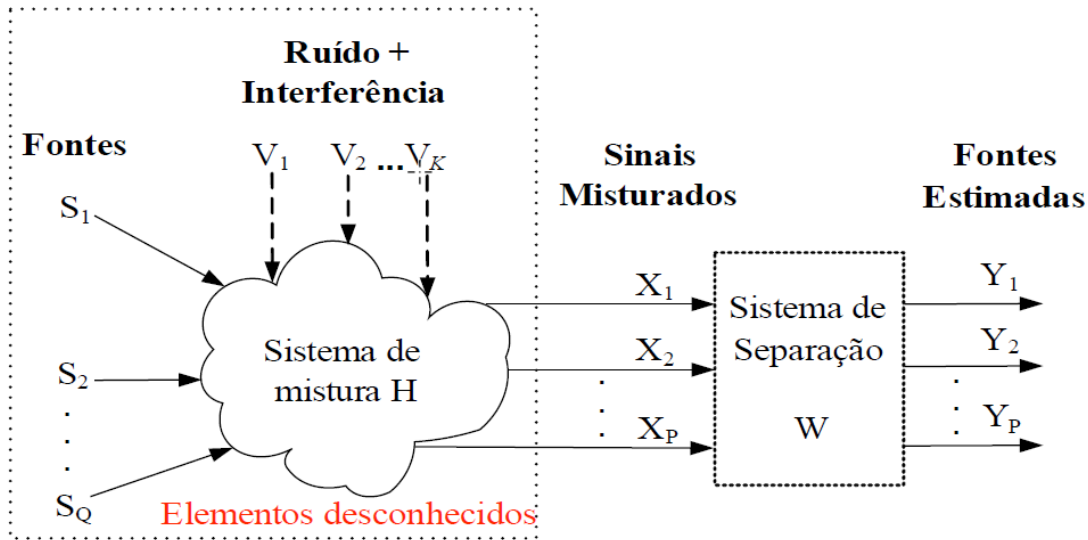


Figura 3.1: Problema de separação cega de sinais.

O problema de separação cega de fontes é encontrar os sinais originais ($\mathbf{s}(n)$) a partir das medidas dos sensores ($\mathbf{x}(n)$). A solução é extrair os sinais originais da mistura através de um sistema de separação \mathbf{W} que forneça estimativas, $\mathbf{y}(n) = [y_1(n), y_2(n), \dots, y_P(n)]^T$, destes sinais.

De maneira geral os problemas tratados pela separação cega de sinais são geralmente do tipo determinado $Q = P$ (mesmo número de sensores e fontes) ou sobredeterminado $Q > P$ (mais sensores do que fontes) [19], e em menor número,

subdeterminado $Q < P$ (menos sensores do que fontes) [20], [21], [22]. Os sinais de mistura são combinados a partir de Q sinais fontes $\mathbf{s}(n)$ e do canal \mathbf{H} .

Em geral, podemos classificar os tipos de misturas em duas grandes classes: lineares e não lineares, sendo aquelas costumeiramente modeladas de duas formas: misturas lineares instantâneas e misturas lineares convolutivas, sendo que nas misturas instantâneas os filtros que correspondem aos canais entre as fontes e os sensores podem ser representados por constantes (uma vez que as misturas num determinado instante de tempo são combinações lineares das amostras das fontes no mesmo instante de tempo) [23], enquanto nas misturas convolutivas os canais são representados por filtros com diversos coeficientes (as misturas ocorrem através da convolução dos sinais das fontes com as respostas ao impulso dos filtros) [24], [25]. Considera-se ainda a presença de ruído.

As fontes $s(n)$ são consideradas, em geral, variantes no tempo, com média zero e totalmente desconhecidas. O número Q de fontes também é desconhecido e pode variar com o tempo, salvo alguma consideração inicial. O usuário tem disponíveis apenas medidas dos sensores $\mathbf{x}(n)$. Assumindo-se que as misturas são lineares, os sinais adquiridos pelos sensores podem ser representados por:

$$x_j(n) = \sum_{i=1}^Q \sum_{l=0}^{N-1} h_{ij}(l) s_i(n-l), j = 1, \dots, P \quad (3.1)$$

ou, na forma matricial,

$$\mathbf{x}(n) = \mathbf{H} * \mathbf{s}(n), \quad (3.2)$$

onde a matriz característica do sistema de mistura \mathbf{H} tem dimensões $P \times Q$, e o operador “*” representa a convolução. Assim, para solucionar este problema é preciso adaptar, ou estimar, os pesos w_{ji} da matriz \mathbf{W} para que as medidas dos sensores $\mathbf{x}(n)$ possam gerar boas estimativas $y_j(n)$ dos sinais, ou em termos matemáticos:

$$y_j(n) = \sum_{i=1}^Q w_{ji}(l) x_i(n), j = 1, \dots, P \quad (3.3)$$

Os pesos ótimos são encontrados quando os sinais de saída $y_j(n)$ atendem restrições estabelecidas, como por exemplo, apresentam mínima informação mútua, e temos assim a descrição do problema de Separação cega de sinais [26], [19], [27], [28].

3.2 Modelo de Mistura Linear e Instantânea

Este é o caso mais simples de um processo de mistura. Nesse caso, temos um sistema de mistura linear invariante no tempo em que todos os sinais chegam aos microfones

ao mesmo tempo, ou seja, misturas instantâneas, onde versões atrasadas dos sinais de fonte não aparecem. Sejam então Q fontes, conforme [29]:

$$\mathbf{s}(n) = \begin{bmatrix} s_1(n) \\ s_2(n) \\ \vdots \\ s_Q(n) \end{bmatrix} \quad (3.4)$$

Seja a matriz de mistura \mathbf{H} , $P \times Q$. Considerando a mistura linear, instantânea e não-ruidosa, podemos expressar o vetor $\mathbf{x}(n)$ que contém as n -ésimas amostras das P misturas pela equação :

$$\mathbf{x}(n) = \mathbf{H}\mathbf{s}(n) \quad (3.5)$$

onde a matriz \mathbf{H} é definida como:

$$\mathbf{H} = \begin{bmatrix} h_{11}(n) & h_{12}(n) & \dots & h_{1Q}(n) \\ h_{21}(n) & h_{22}(n) & \dots & h_{2Q}(n) \\ \vdots & \vdots & \ddots & \vdots \\ h_{P1}(n) & h_{P2}(n) & \dots & h_{PQ}(n) \end{bmatrix} \quad (3.6)$$

Cada escalar $h_{ij}(n)$ determina o quanto da fonte i está presente na mistura j , denotando assim os pesos escalares de cada fonte para cada sensor. Podemos concatenar os vetores $\mathbf{x}(n)$ e obter a matriz dos sinais recebidos nos sensores \mathbf{X} , onde K é o número total de observações de cada fonte.

$$\mathbf{X} = \begin{bmatrix} x_1(1) & x_1(2) & \dots & x_1(K) \\ x_2(1) & x_2(2) & \dots & x_2(K) \\ \vdots & \vdots & \ddots & \vdots \\ x_P(1) & x_P(2) & \dots & x_P(K) \end{bmatrix} \quad (3.7)$$

Fazendo agora o mesmo com o vetor $\mathbf{s}(n)$ e obtemos a matriz das fontes \mathbf{S} .

$$\mathbf{S} = \begin{bmatrix} s_1(1) & s_1(2) & \dots & s_1(K) \\ s_2(1) & s_2(2) & \dots & s_2(K) \\ \vdots & \vdots & \ddots & \vdots \\ s_Q(1) & s_Q(2) & \dots & s_Q(K) \end{bmatrix} \quad (3.8)$$

Essas representações só são válidas para processamento em bloco, i.e., quando todas as observações de todas as fontes estão disponíveis. No caso MLI (Mistura Linear e Instantânea), cada mistura é uma combinação linear das fontes, ou seja:

$$x_j(n) = \sum_{i=1}^Q h_{ij}(l) s_i(n), j = 1, \dots, P \quad (3.9)$$

Para recuperar as fontes $s_i(n)$ a partir das misturas $x_j(n)$, definimos a matriz separadora \mathbf{W} , e o sinal $y_i(n)$, que é a estimativa do sinal $s_i(n)$. Dessa forma, temos:

$$\begin{bmatrix} y_1(n) \\ y_2(n) \\ \vdots \\ y_Q(n) \end{bmatrix} = \mathbf{W} \begin{bmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_P(n) \end{bmatrix} \quad (3.10)$$

Sendo a matriz de separação \mathbf{W} , de dimensões $Q \times P$, é definida por:

$$\mathbf{W} = \begin{bmatrix} w_{11}(n) & w_{12}(n) & \dots & w_{1P}(n) \\ w_{21}(n) & w_{22}(n) & \dots & w_{2P}(n) \\ \vdots & \vdots & \ddots & \vdots \\ w_{Q1}(n) & w_{Q2}(n) & \dots & w_{QP}(n) \end{bmatrix} \quad (3.11)$$

Concatenando-se todas as observações das fontes estimadas $y_i(n)$, chegamos à matriz \mathbf{Y} de fontes estimadas:

$$\mathbf{Y} = \begin{bmatrix} y_1(1) & y_1(2) & \dots & y_1(K) \\ y_2(1) & y_2(2) & \dots & y_2(K) \\ \vdots & \vdots & \ddots & \vdots \\ y_Q(1) & y_Q(2) & \dots & y_Q(K) \end{bmatrix} \quad (3.12)$$

Um tratamento mais abrangente do caso BSS instantâneo e dos respectivos algoritmos pode ser encontrado em [19]. Neste trabalho lidaremos com BSS para ambientes acústicos e, assim, o modelo de mistura instantânea não é apropriado uma vez que versões atrasadas dos sinais de fonte não são considerados. Devido à reflexão e às diferenças no tempo de propagação das ondas sonoras, modelos instantâneos de mistura não podem ser usados para ambientes acústicos reais [30]. Assim é preciso usar um modelo que também leve os atrasos em conta. Para este efeito, podemos usar o modelo de mistura convolutiva, que será visto na próxima seção.

3.3 Modelo de Misturas Convolutivas

O modelo de misturas lineares e instantâneas é demasiado simples. Não modela o atraso entre sensores, tampouco o fenômeno de múltiplos percursos (no caso de sinais de comunicações digitais) ou a reverberação (no caso de sinais de áudio). Nesse tipo de cenário, as misturas são convolutivas, devido aos atrasos que resultam da

propagação do som através do espaço gerando reflexões do som em diferentes objetos (reverberação). Um método de separação de fontes, para poder tratar de casos mais realistas, deve estender sua aplicabilidade para o caso de misturas convolutivas. Estendendo o modelo de mistura instantânea dado na seção anterior, consideraremos também a combinação de versões atrasadas dos sinais das fontes $\mathbf{s}(n)$, correspondendo a um sistema de mistura que é formado por filtros com resposta ao impulso finita (FIR), ao invés de escalares. Como resultado, as P misturas são filtradas por um sistema multicanal com filtros $h_{ij}(n)$ de comprimento N , ou seja:

$$x_j(n) = \sum_{i=1}^Q \sum_{l=0}^{N-1} h_{ij}(l) s_i(n-l), j = 1, \dots, P \quad (3.13)$$

podendo ser representadas também da seguinte forma:

$$x_j(n) = \sum_{i=1}^Q h_{ij}(l) * s_i(n), j = 1, \dots, P \quad (3.14)$$

onde “ $*$ ” é o operador de convolução, ou na forma matricial:

$$\mathbf{x}(n) = \mathbf{H} * \mathbf{s}(n), \quad (3.15)$$

com matriz de mistura \mathbf{H} dada por

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_{11}(n) & \mathbf{h}_{12}(n) & \dots & \mathbf{h}_{1Q}(n) \\ \mathbf{h}_{21}(n) & \mathbf{h}_{22}(n) & \dots & \mathbf{h}_{2Q}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_{P1}(n) & \mathbf{h}_{P2}(n) & \dots & \mathbf{h}_{PQ}(n) \end{bmatrix}, \quad (3.16)$$

sendo cada elemento de \mathbf{H} um vetor linha contendo os N coeficientes $h_{ij}(n)$ do filtro FIR que relaciona a i -ésima fonte com a j -ésima mistura. O produto da matriz \mathbf{H} e do $\mathbf{s}(n)$ agora representa o produto entre cada vetor $h_{ij}(n)$ e o vetor coluna $s_i(n)$ que contém N amostras da j -ésima fonte $s_i(n)$. Da mesma forma, pode-se reescrever a estimativa do canal no caso convolutivo como:

$$\mathbf{y}(n) = \mathbf{W} * \mathbf{x}(n) \quad (3.17)$$

onde

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_{11}(n) & \mathbf{w}_{12}(n) & \dots & \mathbf{w}_{1P}(n) \\ \mathbf{w}_{21}(n) & \mathbf{w}_{22}(n) & \dots & \mathbf{w}_{2P}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{w}_{1P}(n) & \mathbf{w}_{2P}(n) & \dots & \mathbf{w}_{QP}(n) \end{bmatrix} \quad (3.18)$$

em que cada elemento de \mathbf{W} contém os coeficientes $\mathbf{w}_{ji}(n)$ dos filtros de separação usados para estimar a fonte $s_i(n)$ a partir dos sinais dos P sensores para da seguinte forma:

$$x_i(n) = \sum_{j=1}^P \sum_{l=0}^{N-1} w_{ji}(l)x_j(n-l), i = 1, \dots, Q. \quad (3.19)$$

Um modelo de mistura convolutiva junto com o sistema de separação é mostrado na Figura 3.2 . A partir dessa figura, pode-se verificar que a BSS pode ser interpretada como um sistema cego de múltiplas entradas e saídas (MIMO - *Multiple - Input Multiple-Output*).

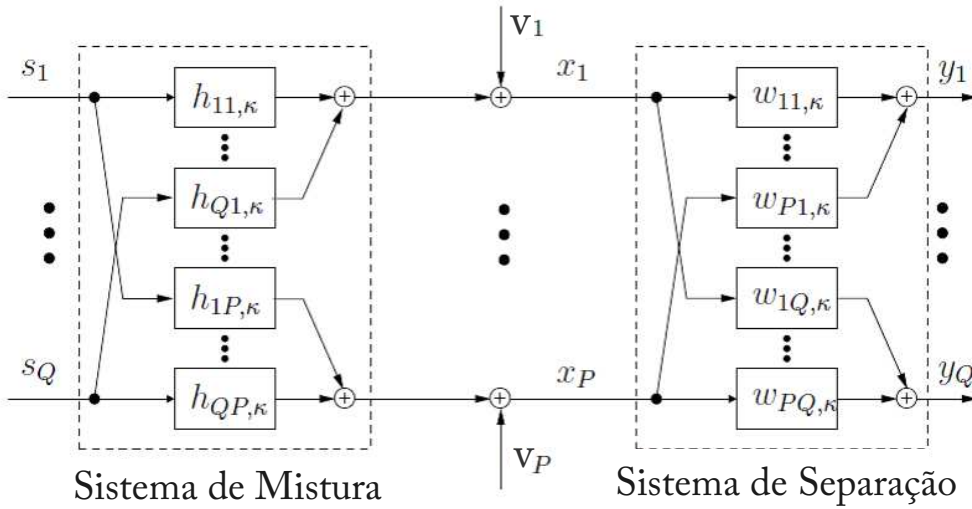


Figura 3.2: Modelo MIMO para mistura convolutiva em BSS

Ao longo deste trabalho, vamos considerar o modelo BSS padrão onde o número Q de fontes é igual ou maior ao número de sensores, ou seja, $Q \leq P$. Deve se notar que o foco principal deste trabalho é a separação cega de fontes para aplicações acústicas para o qual o modelo convolutivo é utilizado. Deve-se comentar também que as misturas convolutivas também podem ser usadas, por exemplo, para modelar vias de transmissão em cenários de comunicação sem fio (por exemplo em [19], [31]) e mais recentemente têm sido aplicadas em ciência do cérebro para a análise de sinais [32], [33].

3.4 Características do Sinal de Voz e sua Utilização na Separação Cega de Fontes

Nesta seção, vamos discutir as propriedades de sinais de fontes acústicas, tais como sinais de voz, e suas utilizações relevantes para algoritmos BSS. Como sabemos,

os sinais de fala são ricos em recursos e possuem certas características as quais os algoritmo BSS podem tirar proveito para um melhor desempenho. Um algoritmo que utiliza todas essas propriedades de forma simultânea será mostrado nos capítulos seguintes.

3.4.1 Propriedades básicas de sinais acústicos

As propriedades que descrevem as estatísticas de sinal no contexto do processamento do sinal acústico que serão discutidas são:

- Distribuição do sinal: A distribuição do sinal é descrito pela função densidade de probabilidade (*pdf* - *Probability Density Function*). Sinais de áudio, como a fala ou música exibem *pdfs* não-gaussianas, que em geral podem ser descritas por densidades super-gaussianas, isto é, apresentam um pico nítido e caudas mais longas que a *pdf* gaussiana, como, por exemplo, a *pdf* laplaciana [34].
- Propriedades temporais: uma das propriedades temporais, amplamente utilizada, de um sinal de ruído é a estacionariedade. Se as amostras temporais de um sinal estacionário não são correlacionadas, o sinal é referido como branco. Os sinais de voz têm estruturas temporais que correspondem a correlações entre amostras em diferentes instantes não-nulas, e, portanto, não são brancos. As correlações, ou momentos de segunda ordem, têm sido investigadas em detalhe na literatura de predição linear (por exemplo, [35]).
- Estacionariedade: O sinal de fala, além de não ser branco, é também um sinal altamente não-estacionário. Na literatura, distinguem-se processos estacionários no sentido restrito e no sentido amplo. Este último supõe que a média do sinal é constante e que a correlação entre amostras (momentos de segunda ordem) depende apenas das diferenças de tempo entre elas e não dos seus instantes absolutos de tempo. Sinais estacionários no sentido restrito apresentam momentos de ordens superiores também dependentes apenas da diferença dos tempos. A maioria dos sinais de áudio são considerados na literatura como não-estacionários; estacionariedade no sentido restrito e no sentido amplo apenas é assumida, por exemplo, para sinais de voz em intervalos de tempo curtos, de 5 ms até 20 ms [36].

3.4.2 Não-estacionário

Algoritmos de separação cega de fontes podem ser derivados explorando a independência estatística dos diferentes locutores num ambiente acústico.

Sabe-se que as estatísticas de processos aleatórios conjuntamente-gaussianos podem ser completamente especificadas pelas estatísticas de primeira e segunda ordens;

assim as características das estatísticas de ordens maiores não carregam qualquer informação adicional sobre os sinais gaussianos. Na maioria das aplicações, a separação cega de fontes acústicas de sinais não-estacionários pode ser efetuada pela diagonalização simultânea de matrizes de correlação de saída de curto intervalo em diferentes instantes de tempo (por exemplo, em BSS instantânea [37], [38] e em BSS convolutiva [39], [40]).

A não-estacionariedade dos sinais de voz é menos intensa a curto prazo, o que motiva o uso de segmentação em quadros destes sinais em análises feitas no espaço tempo \times frequência. O comprimento destes quadros deve refletir uma duração média na qual a informação frequencial pode ser considerada inalterada, a qual normalmente está estimada entre 10 e 30 ms [41].

3.4.3 Não-gaussianidade

Os sinais de voz possuem uma distribuição que difere de uma gaussiana; em tal caso, a independência estatística dos sinais da fonte não precisa ser assumida (a natureza não-gaussiana dos sinais de voz não é usada nos algoritmos baseados em independência). Uma medida estatística popular para a avaliação da gaussianidade de um sinal é a curtose, a qual pode ser compreendida como uma versão normalizada do momento de quarta ordem [19].

Em geral, variáveis aleatórias não-gaussianas apresentam curtose não-nula. As variáveis aleatórias que apresentam curtose negativa, nula e positiva são regidas por distribuições chamadas, respectivamente, subgaussiana, mesocurtótica e supergaussiana.

Distribuições supergaussianas costumam apresentar um pico mais concentrado do que suas congêneres gaussianas, assim como uma cauda mais pronunciada. A Figura 3.3 permite uma visualização destas propriedades (as caudas das distribuições foram ampliadas para que suas magnitudes relativas possam ser observadas). Além disso, a não-gaussianidade pode ser explorada por meio de estatísticas de ordem superior. Algoritmos BSS utilizando estatísticas de ordem superior também são denominados algoritmos de análise de componentes independentes (ICA), por exemplo, em BSS instantânea [42],[43],[44],[45] e em BSS convolutiva [46].

3.4.4 Não-brancura

O espectro de um sinal de voz apresenta consistentemente uma característica colorida, o que pode ser inferido a partir da modelagem física do fenômeno de geração da fala. Como os sinais de áudio apresentam dependências temporais, essa característica pode ser explorada pelo critério BSS. Por conseguinte, pode se presumir que as amostras de cada fonte de sinal não são independentes ao longo do eixo do

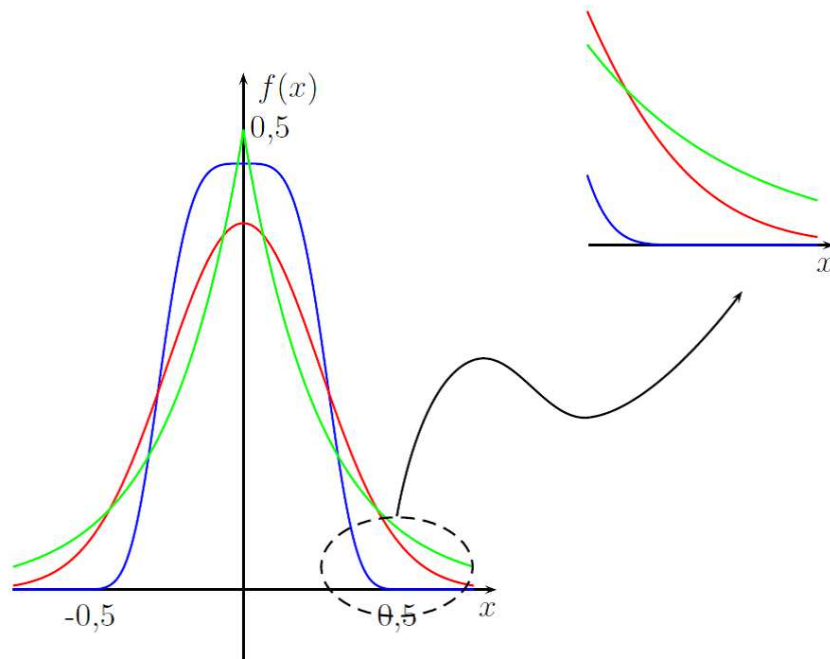


Figura 3.3: Exemplos de distribuições supergaussiana (em verde), gaussiana (em vermelho) e subgaussiana (em azul).

tempo; no entanto, as amostras de sinais de diferentes fontes são independentes entre si. Com base no pressuposto de independência mútua estatística para fontes não-brancas, vários algoritmos podem ser encontrados na literatura.

Principalmente a não-brancura é explorada usando estatísticas de segunda ordem por diagonalização simultânea de matrizes de correlação de saída ao longo de vários intervalos de tempo (por exemplo, em BSS instantânea [47], [48], [49] e em BSS convolutiva [50], [51]). Deve-se notar que os algoritmos de BSS convolutivos baseiam-se na independência estatística mútua para sinais temporais brancos, e, portanto, distorcem os sinais de saída separados [34].

3.5 Ambiguidades na separação cega de fontes

Como o conceito de BSS é exclusivamente baseado no pressuposto de independência mútua dos sinais de fonte, surgem algumas ambiguidades. Na BSS instantânea as seguintes indeterminações aparecem [19]:

- **Ambiguidade do Escalamento:** Os componentes independentes estimados só podem ser determinados até um fator escalar. Isto acontece porque, como não conhecemos nem a fonte nem os componentes da matriz de mistura, qualquer fator k que multiplique a fonte poderia ser cancelado multiplicando-se o componente da matriz de mistura por $\frac{1}{k}$, e a mistura (que é a única informação disponível) seria a mesma, tornando-se impossível recuperar este valor de k .

- **Ambiguidade da Permutação:** A ordem dos componentes independentes não pode ser determinada. Isto também acontece porque não conhecemos nem o vetor das fontes nem a matriz de mistura \mathbf{H} . A propriedade de comutação da soma de diversos termos torna irrelevante para esta operação a ordem destes. Isso implica a arbitrariedade do ordenamento, o qual, portanto, não é passível de recuperação na ausência de conhecimentos a priori acerca da matriz de mistura ou das fontes.

Como ambos os sinais de fonte originais e o sistema de mistura são desconhecidos, um possível escalonamento ou permutação dos sinais de fonte sempre poderia ser desfeito(a) por um sistema de mistura diferente. Devido à impossibilidade de distinguir se a escala ou a permutação ocorreu nos sinais de origem ou no sistema de mistura, tais ambiguidades não podem ser resolvidas sem o uso de informações a priori adicionais, se forem observados apenas os sinais dos sensores. Assim, as fontes originais só podem ser recuperadas até uma mudança de escala desconhecida e de permutação.

No caso BSS convolutivo as indeterminações se tornam:

- **Ambiguidade da Filtragem:** Os componentes independentes estimados só podem ser determinados até uma operação de filtragem arbitrária;
- **Ambiguidade da Permutação:** A ordem dos componentes independentes não pode ser determinada.

Mais uma vez a ambiguidade da permutação não pode ser resolvida sem a informação adicional a priori. No entanto, se, por exemplo, as posições dos sensores são conhecidas, em seguida, a posição de cada fonte separada pode ser determinada a partir do sistema de separação [52]. Para algumas aplicações isso pode ser suficiente para resolver o problema de permutação.

Em geral, a ambiguidade de escala para BSS convolutiva se traduz em uma filtragem arbitrária dos sinais de saída. Portanto, é necessário se distinguir entre separação cega de fonte, em que o objetivo é meramente separar os sinais de fonte original, e desconvolução cega de sinais, que tem a tarefa mais difícil de recuperar os sinais de fonte originais a menos de um fator de escala arbitrário e um atraso constante.

Quando trabalhamos em cenários com transmissão de ondas acústicas no domínio do tempo, essas ambiguidades não interferem na qualidade da separação, pois tanto a ordem dos sinais quanto suas amplitudes não são importantes no resultado final. Em compensação, nos algoritmos no domínio da frequência, esses problemas são críticos e devem ser solucionados para obtermos boas estimativas das fontes [29].

3.6 Análise de Componentes Independentes

A análise de componentes independentes (ICA, do inglês *Independent Component Analysis*) é uma metodologia utilizada para encontrar, a partir de observações de misturas, as fontes de informação originais. A abordagem dessa técnica para BSS foi escolhida devido à popularidade do método na literatura e seu relativamente baixo esforço computacional. Os sinais das misturas podem ser provenientes de diferentes direções. Aqui, devemos salientar que as características dos sinais de fonte são estatisticamente independentes, assim como independentes dos componentes de ruído. Portanto, o objetivo da ICA é separar uma mistura linear e instantânea de fontes independentes não-gaussianas. ICA é o método de separação cega de fontes clássico para lidar com os problemas que estão intimamente relacionados com o problema “*cocktail-party*”.

Análise de Componentes Independentes usa propriedades teóricas para separar os sinais de fonte misturados por um processo linear [53]. Nesse método, não se tem nenhum conhecimento prévio de como as fontes se comportam. Para que esse processamento seja possível, três fatores são importantes:

- As componentes (fontes de informação) são estatisticamente independentes entre si;
- A matriz misturadora é quadrada, ou seja, o número de observações é o mesmo que o número de fonte;.
- As componentes devem ser não-gaussianas.

Ele opera sobre a suposição de que as fontes nas quais estamos interessados são de natureza não-gaussiana. O sinal de voz pode ser considerado não-gaussiano em ambos os domínios de tempo e frequência. Há um conjunto de algoritmos baseados na maximização da gaussianidade dos sinais desenvolvidos para se obter a solução da ICA. Entre os principais encontram-se o algoritmo INFOMAX, proposto em [45], e sua versão modificada, apresentada em [54], que utiliza um método de atualização dos coeficientes da matriz de separação baseado no gradiente natural.

O que diferencia a ICA das outras técnicas existentes é justamente o fato dela trabalhar com componentes que são ao mesmo tempo não gaussianos e estatisticamente independentes. Essa representação pode ser usada para capturar a estrutura dos componentes (ou dados) em várias aplicações, como extração de características ou separação de sinais.

3.7 Medidas de Desempenho

Para se comparar os resultados e avaliar o desempenho dos estimadores, faz-se necessário o uso de algumas medidas de desempenho. A medida de desempenho útil deve apresentar uma elevada correlação com os resultados obtidos a partir de avaliações subjetivas. Dentre as disponíveis, a mais utilizada pela literatura é a Razão Sinal Interferência (SIR).

Para avaliar o desempenho BSS de forma adequada, tem de ser salientado que a qualidade percebida dos sinais de saída BSS é determinada por três fatores que devem ser abordadas individualmente [34]:

- Supressão de interferência de fontes pontuais;
- Atenuação de ruído de fundo;
- A distorção do sinal desejado.

Para realizar devidamente as medidas de desempenho, decompõem-se os sinais de saída $y_q(n)$, $q = 1, \dots, Q$ como:

$$y_q(n) = y_{s_r,q}(n) + y_{c,q}(n) + y_{v,q}(n) \quad (3.20)$$

onde $y_{s_r,q}(n)$, é o componente que contém a fonte desejada. Em geral, a origem desejada no canal de saída pode ser qualquer um dos sinais de fonte, devido à ambiguidade de permutação. Na avaliação das experiências no presente trabalho, os canais de saída foram reordenados manualmente para evitar qualquer permutação. O sinal $y_{c,q}(n)$ é o componente de interferência no canal de saída decorrente das fontes pontuais restantes, e por fim $y_{v,q}$ denota as contribuições do ruído de fundo na saída. Os sinais de microfone $x_p(n)$, $p = 1, \dots, P$ podem ser decompostos de forma análoga como:

$$x_p(n) = x_{s_r,p}(n) + x_{c,p}(n) + v_p(n) \quad (3.21)$$

onde $x_{s_r,p}(n)$ é o componente decorrente da fonte desejada e $x_{c,p}(n)$ contém as contribuições das outras fontes pontuais interferentes. O ruído de fundo é denotado por v_p .

3.7.1 Razão Sinal-Interferência

A decomposição do sinal de saída em componentes desejados e interferentes permite avaliar a supressão de interferências através do cálculo da razão sinal-interferência (SIR - *Signal-to-Interference Ratio*). A SIR relaciona a potência do sinal desejado

com a dos interferentes. Desta forma, temos que a SIR para cada canal q , medida em decibéis (dB), é dada por:

$$SIR_y = 10 \log_{10} \frac{E\{y_{s,r,q}^2\}}{E\{y_{c,q}^2\}} \quad (3.22)$$

Na equação acima, o operador esperança $E\{\cdot\}$ tem de ser substituído na prática por uma média no tempo. Esta estimativa da SIR tem apenas uma fraca correlação com a qualidade percebida por medições auditivas [55]. A contribuição da fonte interferente pode ser mostrada através da Figura 3.4 (caso 2×2).

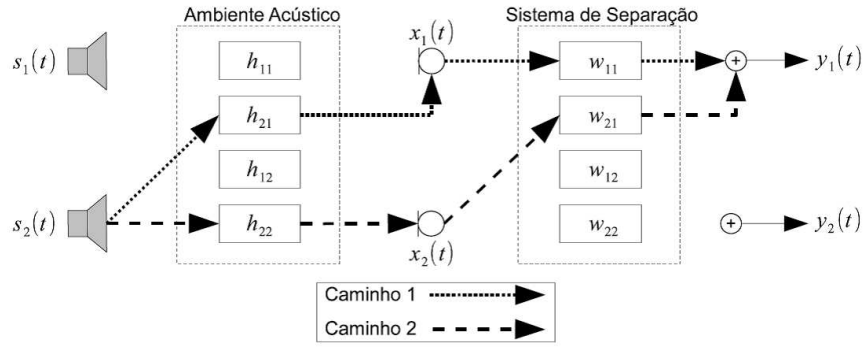


Figura 3.4: Contribuição da fonte 1 na saída 2 para um cenário 2×2 .

Por exemplo, se considerarmos que o sinal $s_1(n)$ é o sinal desejado e $s_2(n)$ é o sinal interferente, o cálculo da contribuição de $s_2(n)$ na saída 1 pode ser elaborado seguindo os caminhos a partir da fonte 2 até os microfones, e, em seguida, até a saída 1.

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_{gain} [34]. 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:

$$SIR_{gain} = SIR_y - SIR_x \quad (3.23)$$

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

3.7.2 Razão Sinal-Ruído

Em aplicações práticas, além da presença do sinal interferente, pode ser observada a presença de ruído nas misturas. BSS em geral não tem o objetivo de suprimir

o ruído de fundo. No entanto, a influência dos filtros BSS sobre o ruído de fundo também pode ser investigada. Portanto, além da razão sinal-interferência introduzimos também a razão sinal-ruído (*SNR-Signal-to-Noise Ratio*), que é definida para um sistema de separação cega de fontes como:

$$SNR_y = 10 \log_{10} \frac{E\{y_{s_r,q}^2\}}{E\{y_{n,q}^2\}} \quad (3.24)$$

E assim como no cálculo da SIR, podemos calcular a SNR relativa do sistema:

$$SNR_{gain} = SNR_y - SNR_x \quad (3.25)$$

onde SNR_x é a SNR em um microfone e SNR_y a SNR na saída correspondente.

3.7.3 Distorção do Sinal

Para avaliar a distorção do sinal introduzido pelo algoritmo BSS o sinal desejado na saída do BSS é comparado com o componente do sinal de mistura desejado. Este baseia-se no pressuposto de que o objetivo do BSS é meramente separar os sinais de fonte original e preservar o conteúdo espectral da mistura [34].

A ambiguidade de filtragem pode levar a uma filtragem arbitrária, e, sendo assim, levar a uma distorção do sinal desejado. A grandeza que é razoavelmente bem correlacionada com a percepção subjetiva de distorção de voz é a distância espectral logarítmica [56]. A distância log-espectral não ponderada (SD) em decibéis (dB) é definida para o sinal de fonte desejada na saída como:

$$SD = \sqrt{\left(20 \log_{10} \frac{|Y_{s_r,q}|}{|X_{s_r,q}|}\right)^2} \quad (3.26)$$

em que $Y_{s_r,q}$ e $X_{s_r,q}$ são as representações no domínio da frequência (DFT) de $y_{s_r,q}$, e $x_{s_r,q}$, respectivamente. Alternativamente, também a representação no domínio DFT pode ser estimada através da utilização de técnicas de predição lineares. A partir dessa equação pode-se observar que um valor de $SD = 0$ dB corresponde a nenhuma distorção do sinal.

Capítulo 4

Filtragem Adaptativa em Subbandas para Separação Supervisionada de Fontes

A evolução das telecomunicações tem impulsionado o desenvolvimento de técnicas eficientes para processamento digital de sinais. Em particular, técnicas de filtragem adaptativa têm atraído um grande interesse. Devido ao bom desempenho, baixa complexidade computacional e alta robustez, essas técnicas têm sido amplamente utilizadas em diversas aplicações, tais como identificação de sistemas, equalização de canais, cancelamento de eco e separação de fontes [8],[57],[58],[59], sendo esta última aplicação a que será abordada neste trabalho.

A separação cega de fontes de áudio (*Blind Audio Source Separation* - BASS) tem sido um tema de intensa investigação durante os últimos anos. Vários métodos de sucesso surgiram, tais como ICA (descrito no Capítulo 3) [27], CASA (*Computational Auditory Scene Analysis*) [60] e SD (*Sparse Decomposition*) [20]. No entanto, ainda é difícil de se avaliarem as características e limitações de algoritmos de separação de fontes usando as medidas de desempenho descritas no Capítulo 3, principalmente no caso desafiador de misturas convolutivas. A separação de fontes supervisionada pode ser uma boa ferramenta para auxiliar no aperfeiçoamento das técnicas de separação cega de fontes.

Neste capítulo, investigaremos a separação de fontes a partir de misturas convolutivas de forma supervisionada, através de filtragem adaptativa, utilizando dois algoritmos: o NLMS (*Normalized Least-Mean-Square*) [8], o NSAF (*Normalized Subband Adaptive Filters*) [61] e o NSAF modificado proposto em [62] e [63]. Neste último, a partir da observação de que existe correlação entre os sinais das misturas, é proposta uma modificação no algoritmo NSAF, introduzindo-se as matrizes de correlação dos sinais das misturas em subbandas na equação de atualização dos

coeficientes, a fim de acelerar a convergência do algoritmo.

Utilizaremos os algoritmos adaptativos supervisionados para avaliar as limitações do modelo linear convolutivo, composto de filtros FIR, na separação cega de fontes. Os algoritmos apresentados também podem ser utilizados em aplicações em que trechos dos sinais originais presentes nas misturas são conhecidos antes de se fazer a separação [64].

4.1 Algoritmo NLMS para o problema de separação de fontes

Para adequar o algoritmo NLMS ao problema de separação supervisionada de fontes (considerando-se por simplicidade o caso de duas fontes e dois sensores), obtêm-se os sinais de saída, $y_1(n)$ e $y_2(n)$, e de erro, $e_1(n)$ e $e_2(n)$, através das equações

$$y_i(n) = \mathbf{w}_i^T(n)\mathbf{x}(n) \quad (4.1)$$

$$e_i(n) = d_i(n) - y_i(n) \quad (4.2)$$

onde o vetor de entrada $\mathbf{x}(n)$ é composto pelos dois vetores dos sinais de mistura

$$\mathbf{x}_i(n) = \left[x_i(n) \quad x_i(n-1) \quad \dots \quad x_i(n-N+1) \right]^T \quad (4.3)$$

para $i = 1, 2$, ou seja,

$$\mathbf{x}(n) = \left[\mathbf{x}_1^T(n) \quad \mathbf{x}_2^T(n) \right]^T \quad (4.4)$$

e os vetores de coeficientes, $\mathbf{w}_1(n)$ e $\mathbf{w}_2(n)$, contêm os coeficientes dos dois filtros que geram cada saída, dados por

$$\mathbf{w}_1(n) = \left[\mathbf{w}_{11}^T(n) \quad \mathbf{w}_{12}^T(n) \right]^T \quad (4.5)$$

$$\mathbf{w}_2(n) = \left[\mathbf{w}_{21}^T(n) \quad \mathbf{w}_{22}^T(n) \right]^T \quad (4.6)$$

onde

$$\mathbf{w}_{ij}(n) = \left[w_{ij,0}(n) \quad w_{ij,1}(n) \quad \dots \quad w_{ij,N-1}(n) \right]^T \quad (4.7)$$

sendo i e j os índices relativos às fontes e misturas, respectivamente, e N o número de coeficientes de cada filtro de separação. Portanto, a cada iteração, haverá a atualização dos vetores de coeficientes, $\mathbf{w}_1(n)$ e $\mathbf{w}_2(n)$, de acordo com a equação

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \frac{\mu}{\delta + \mathbf{x}^T(n)\mathbf{x}(n)} e_i(n)\mathbf{x}(n) \quad (4.8)$$

4.2 Algoritmo NSAF para o problema de separação de fontes

Para adequar o algoritmo NSAF ao problema de separação de fontes supervisionada, foi empregado um procedimento semelhante ao do algoritmo NLMS. Assim, as equações (2.21) e (2.22) foram modificadas para gerar os sinais de erro em subbandas e atualizar os coeficientes dos filtros de separação. Considerando o caso de duas fontes e dois sensores, os vetores com os coeficientes dos filtros que geram as estimativas das fontes são definidos como

$$\mathbf{w}_1(n) = \left[\mathbf{w}_{11}^T(n) \quad \mathbf{w}_{12}^T(n) \right]^T \quad (4.9)$$

$$\mathbf{w}_2(n) = \left[\mathbf{w}_{21}^T(n) \quad \mathbf{w}_{22}^T(n) \right]^T \quad (4.10)$$

sendo

$$\mathbf{w}_{ij}(n) = \left[w_{ij,0}(n) \quad w_{ij,1}(n) \quad \dots \quad w_{ij,N-1}(n) \right]^T \quad (4.11)$$

o vetor com os N coeficientes do filtro $\mathbf{w}_{ij}(n)$ que gera a parcela da estimativa da i -ésima fonte a partir da j -ésima mistura. Definindo o vetor com as amostras dos sinais de mistura na m -ésima subbanda como:

$$\mathbf{x}_m(n) = \left[\mathbf{x}_{1,m}^T(n) \quad \mathbf{x}_{2,m}^T(n) \right]^T \quad (4.12)$$

onde

$$\mathbf{x}_{i,m}(n) = \left[x_{i,m}(nM) \quad x_{i,m}(nM-1) \quad \dots \quad x_{i,m}(nM-N+1) \right]^T \quad (4.13)$$

o sinal de erro em subbanda decimado e a equação de atualização dos filtros do sistema de separação relativos à estimativa da i -ésima fonte são dados, respectivamente, por

$$e_{i,m\downarrow}(n) = d_{i,m\downarrow}(n) - \mathbf{w}_i^T(n) \mathbf{x}_m(n) \quad (4.14)$$

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \mu \sum_{m=0}^{M-1} \frac{\mathbf{x}_m(n)}{\delta + \|\mathbf{x}_m(n)\|^2} e_{i,m\downarrow}(n) \quad (4.15)$$

4.2.1 Modificação no algoritmo NSAF para o problema de separação

Partindo da observação de que existe correlação entre os sinais das misturas, para acelerar a convergência do algoritmo NSAF foram introduzidas estimativas das matrizes de correlação dos sinais de mistura em subbandas $\hat{\mathbf{R}}_m(n) = E[\mathbf{x}_m(n) \mathbf{x}_m^T(n)]$

na equação de atualização dos coeficientes, obtendo-se

$$\mathbf{w}_i(n+1) = \hat{\mathbf{w}}_i(n) + \mu \sum_{m=0}^{M-1} \hat{\mathbf{R}}_m^{-1}(n) \frac{\mathbf{x}_m(n)}{\delta + \|\mathbf{x}_m(n)\|^2} e_{i,m\downarrow}(n) \quad (4.16)$$

onde

$$\hat{\mathbf{R}}_m(n) = \mathbf{x}_m(n) \mathbf{x}_m^T(n) = \begin{bmatrix} \hat{\mathbf{R}}_{m,11}(n) & \hat{\mathbf{R}}_{m,12}(n) \\ \hat{\mathbf{R}}_{m,21}(n) & \hat{\mathbf{R}}_{m,22}(n) \end{bmatrix} \quad (4.17)$$

Considerando que os sinais de entrada decimados nas subbandas são aproximadamente brancos, as matrizes $\hat{\mathbf{R}}_{m,ij}(n)$ podem ser aproximadas pelas matrizes diagonais

$$\hat{\mathbf{R}}_{m,ij}(n) \approx \sigma_{m,ij}^2(n) \mathbf{I} \quad (4.18)$$

onde

$$\sigma_{m,ij}^2(n) = \mathbf{x}_{i,m}^T(n) \mathbf{x}_{j,m}(n) \quad (4.19)$$

Definindo a matriz correlação inversa dos sinais de misturas na m -ésima subbanda como

$$\hat{\mathbf{R}}_m(n)^{-1} = \begin{bmatrix} \hat{\mathbf{S}}_{m,11}(n) & \hat{\mathbf{S}}_{m,12}(n) \\ \hat{\mathbf{S}}_{m,21}(n) & \hat{\mathbf{S}}_{m,22}(n) \end{bmatrix} \quad (4.20)$$

usando a aproximação (4.18) e observando que $\sigma_{m,12}^2(n) = \sigma_{m,21}^2(n)$, obtemos

$$\hat{\mathbf{S}}_{m,11}(n) = (\sigma_{m,11}^2(n) - \sigma_{m,12}^4(n) \sigma_{m,22}^{-2}(n))^{-1} \mathbf{I} \quad (4.21)$$

$$\hat{\mathbf{S}}_{m,22}(n) = (\sigma_{m,22}^2(n) - \sigma_{m,12}^4(n) \sigma_{m,11}^{-2}(n))^{-1} \mathbf{I} \quad (4.22)$$

$$\hat{\mathbf{S}}_{m,12}(n) = -\sigma_{m,11}^{-2}(n) \sigma_{m,12}^2(n) (\sigma_{m,22}^2(n) - \sigma_{m,12}^4(n) \sigma_{m,11}^{-2}(n))^{-1} \mathbf{I} \quad (4.23)$$

$$\hat{\mathbf{S}}_{m,21}(n) = -\sigma_{m,22}^{-2}(n) \sigma_{m,12}^2(n) (\sigma_{m,11}^2(n) - \sigma_{m,12}^4(n) \sigma_{m,22}^{-2}(n))^{-1} \mathbf{I} \quad (4.24)$$

O aumento na complexidade computacional resultante da introdução das inversas das matrizes de correção na equação de atualização dos coeficientes (4.15) é de $3M^2 + 8M$ multiplicações. Como, em geral, o número de subbandas M é bem menor do que o comprimento dos filtros adaptativos N , esse aumento não é significativo.

4.3 Sinal desejado para a avaliação da separação cega de fontes

No problema de separação cega de sinais de áudio em misturas convolutivas, em geral não é possível recuperar os sinais originais das fontes, sendo permitido obter como solução válida os sinais originais filtrados. Denotando por $H_{ij}(z)$ a função de transferência correspondente da i -ésima fonte ao j -ésimo sensor, pode-se escrever,

no domínio Z , os sinais das misturas $X_i(z)$ e de saída do sistema de separação $Y_i(z)$ para o problema de duas fontes e dois sensores como:

$$X_1(z) = H_{11}(z)S_1(z) + H_{12}(z)S_2(z) \quad (4.25)$$

$$X_2(z) = H_{21}(z)S_1(z) + H_{22}(z)S_2(z) \quad (4.26)$$

$$Y_1(z) = W_{11}(z)X_1(z) + W_{12}(z)X_2(z) \quad (4.27)$$

$$Y_2(z) = W_{21}(z)X_1(z) + W_{22}(z)X_2(z) \quad (4.28)$$

Substituindo as equações (4.25) e (4.26) em (4.27) e (4.28), tem-se:

$$\begin{aligned} Y_1(z) &= (W_{11}(z)H_{11}(z) + W_{12}(z)H_{21}(z))S_1(z) \\ &\quad + (W_{11}(z)H_{12}(z) + W_{12}(z)H_{22}(z))S_2(z) \end{aligned} \quad (4.29)$$

$$\begin{aligned} Y_2(z) &= (W_{21}(z)H_{11}(z) + W_{22}(z)H_{21}(z))S_1(z) \\ &\quad + (W_{21}(z)H_{12}(z) + W_{22}(z)H_{22}(z))S_2(z) \end{aligned} \quad (4.30)$$

Para encontrarmos uma solução possível para o problema foi imposto ao sistema que o sinal desejado de saída $D_1(z)$ deveria representar o sinal original $S_1(z)$ filtrado e o sinal desejado de saída $D_2(z)$, o sinal $S_2(z)$ filtrado. Para isso, basta igualarmos a zero a parcela do sinal correspondente a $S_2(z)$ na equação de $Y_1(z)$ e a parcela do sinal correspondente a $S_1(z)$ na equação de $Y_2(z)$, ou seja,

$$W_{11}(z)H_{12}(z) + W_{12}(z)H_{22}(z) = 0 \quad (4.31)$$

$$W_{21}(z)H_{11}(z) + W_{22}(z)H_{21}(z) = 0 \quad (4.32)$$

As equações acima possuem infinitas soluções e a solução adotada foi $W_{11}(z) = H_{22}(z)$, $W_{12}(z) = -H_{12}(z)$, $W_{21}(z) = H_{21}(z)$ e $W_{22}(z) = -H_{11}(z)$. Assim, os sinais $D_1(z)$ e $D_2(z)$ passam a ser:

$$\begin{aligned} D_1(z) &= (W_{11}(z)H_{11}(z) + W_{12}(z)H_{21}(z))S_1(z) \\ &= (H_{22}(z)H_{11}(z) - H_{12}(z)H_{21}(z))S_1(z) \end{aligned} \quad (4.33)$$

$$\begin{aligned} D_2(z) &= (W_{21}(z)H_{12}(z) + W_{22}(z)H_{22}(z))S_2(z) \\ &= (H_{12}(z)H_{21}(z) - H_{11}(z)H_{22}(z))S_2(z) \end{aligned} \quad (4.34)$$

Os sinais desejados das equações acima foram usados nas simulações apresentadas a seguir.

4.4 Resultados das simulações

Os algoritmos adaptativos para separação supervisionada de fontes descritos nesse trabalho foram avaliados empregando-se duas misturas convolutivas de dois sinais de voz, correspondendo aos sinais captados por dois microfones afastados de 5 cm em ambiente reverberante e amostrados em 8 kHz.

As simulações da propagação acústica de salas foram desenvolvidas usando o modelo “Image-Source” descrito em [16]. Foram feitas simulações para três diferentes tempos de reverberação ($T60$): 0.1 s, 0.25 s e 0.5 s. A medida de avaliação utilizada foi a SIR (*Signal to Interference Ratio*). Para a decomposição em subbandas, empregou-se um banco de filtros modulado por cosseno [65] com $M = 4$ subbandas e filtro protótipo de comprimento 32. O passo de adaptação e parâmetro de regularização utilizados em todos os algoritmos foram $\mu = 0.5$ e $\delta = 0.1$.

Nas Figuras 4.1, 4.2 e 4.3 são mostradas as evoluções da SIR ao longo das iterações obtidas com os algoritmos NLMS, NSAF e NSAF modificado para $T60 = 0.1$ s, 0.25 s e 0.5 s, respectivamente, com filtros de separação de comprimentos $N = 623$, 1599 e 3227, iguais aos comprimentos dos filtros de mistura (denominados L).

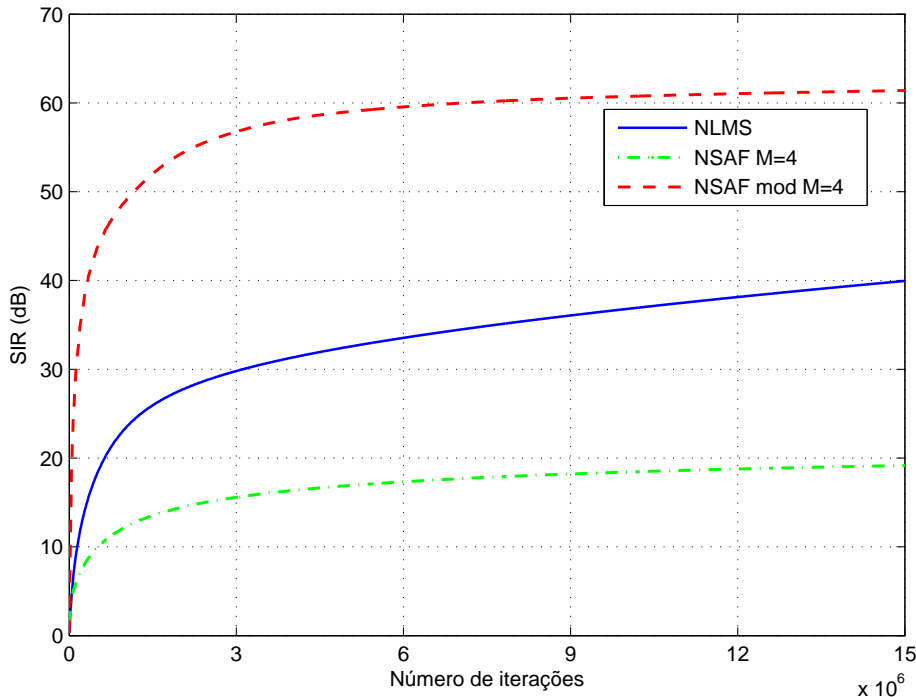


Figura 4.1: Evolução da SIR(dB) para $T60 = 0.1$ s para os algoritmos NLMS, NSAF e NSAF modificado.

Pode-se verificar nessas figuras que o algoritmo NSAF modificado apresenta taxa de convergência consideravelmente maior que as do NLMS e do NSAF convencional

para o problema de separação supervisionada de fontes.

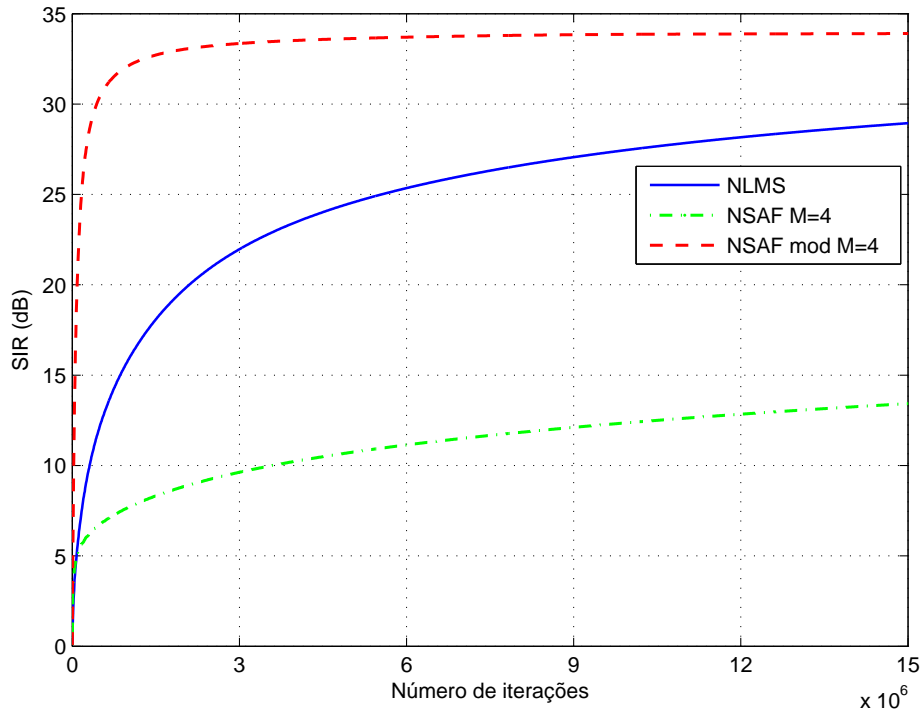


Figura 4.2: Evolução da SIR(dB) para $T60 = 0.25$ s para os algoritmos NLMS, NSAF e NSAF modificado.

Como esperado, a introdução da matriz de correlação dos sinais de mistura na equação de atualização dos coeficientes acelera a convergência do algoritmo NSAF, a qual é lenta devido à forte correlação entre os sinais das misturas. O processamento em subbandas explora as características dos sinais de mistura nas subbandas, através da normalização do passo de adaptação, resultando em valores de SIR mais altos do que os obtidos com o algoritmo NLMS.

Das Figs. 4.1, 4.2 e 4.3, pode-se também concluir que há uma limitação no modelo de separação de fontes que emprega filtros de resposta ao impulso finita (FIR), a qual tem efeito mais acentuado na SIR conforme o tempo de reverberação aumenta.

Portanto, por melhor que seja o algoritmo de separação cega implementado, para os casos de alta reverberação, a SIR máxima a ser alcançada com filtros de separação FIR de comprimento $N = L$ está limitada pelo modelo do sistema de separação empregado (em torno de 23 dB, para $T60 = 0.5$ s), enquanto que para tempos de reverberação menores é possível obter altos valores de SIR (acima de 60 dB para $T60 = 0.1$ s).

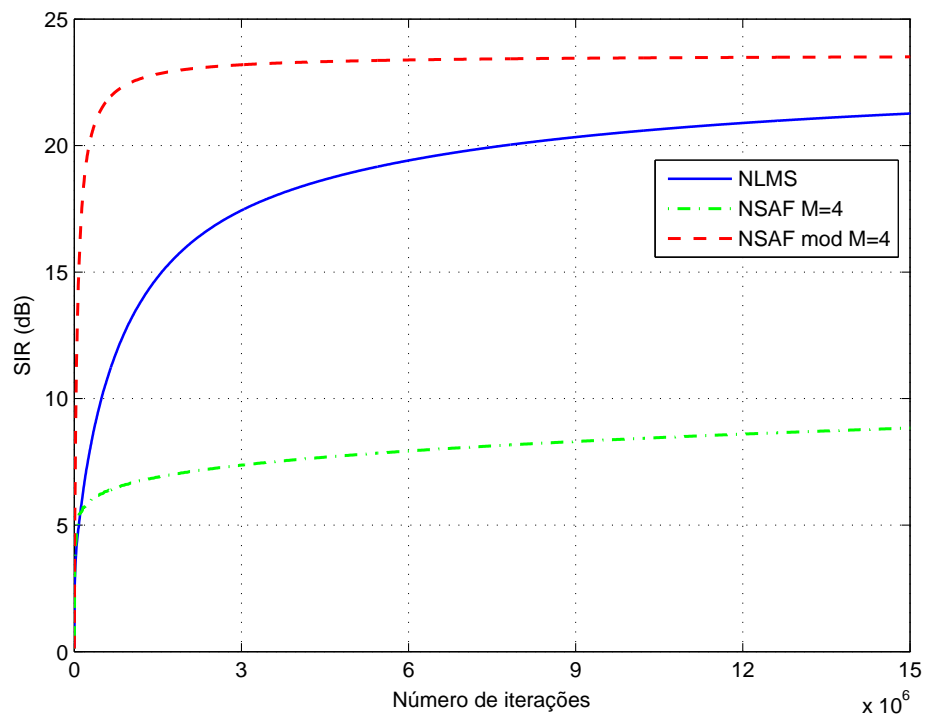


Figura 4.3: Evolução da SIR(dB) para $T_{60} = 0.5$ s para os algoritmos NLMS, NSAF e NSAF modificado.

Capítulo 5

Estudo da Parametrização Automática de Algoritmo Adaptativo para Sistemas Esparsos

O l_0 -LMS é um algoritmo adaptativo recentemente proposto para promover a identificação de sistemas esparsos de modo mais eficiente. Sua estratégia consiste em alterar a função-custo do algoritmo padrão LMS, por meio da adição de um termo que penaliza soluções pouco esparsas. Comparado com seus precursores, esse algoritmo se revelou bastante competitivo, podendo sua função custo ser futuramente usada em conjunto com algoritmos de separação cega de fontes apresentados neste trabalho. Entretanto, a vantagem do algoritmo l_0 -LMS depende não raro de uma judiciosa escolha de seus parâmetros, de sorte a evitar um desempenho até mesmo inferior ao LMS tradicional. Neste capítulo apresentaremos um estudo feito em [66], no contexto de identificação de sistemas, uma forma de ajuste automático (dependente da razão sinal-ruído) do parâmetro referente ao termo penalizador, de modo a garantir um desempenho vantajoso do algoritmo l_0 -LMS.

5.1 Introdução

O algoritmo LMS é um dos mais populares algoritmos de filtragem adaptativa, devido ao seu bom desempenho, baixa complexidade computacional e alta robustez.

Os sistemas desconhecidos a serem identificados costumam ser esparsos em diversos cenários físicos, como cancelamento de eco e canais de transmissão de TV digital. Em outras palavras, a maior parte dos coeficientes das respostas ao impulso a identificar destes sistemas são nulos ou próximos de zero [67].

O algoritmo LMS, entretanto, não aproveita uma eventual concentração de energia do sistema a identificar em poucos coeficientes. Para contornar esta deficiência, algoritmos que recorrem à esparsidade para otimizar o desempenho do processo de identificação têm sido buscados. Um desses algoritmos é o l_0 -LMS (*Norm Constraint Least-Mean Squares*) [68], o qual incrementa a taxa de convergência com relação ao LMS. Como a maioria dos coeficientes de um sistema esparsos são iguais a zero, o desempenho global de identificação do l_0 -LMS pode superar o do LMS.

O algoritmo l_0 -LMS apresenta maior robustez contra o ruído, exige menos medidas para a reconstrução perfeita [69] e pode apresentar uma aceleração da taxa de convergência em sistemas esparsos. Tais vantagens, no entanto, somente ocorrem com determinadas escolhas de seus parâmetros [10].

A parametrização incorreta não apenas pode tornar o algoritmo menos eficiente, quanto também pode engendrar um desempenho inferior ao do algoritmo LMS tradicional. Daí a importância de adotarmos um procedimento de ajuste automático dos parâmetros pré-definidos (particularmente o parâmetro κ , correspondente ao peso dado ao atrator dos coeficientes para zero na função custo), de sorte a sempre garantir soluções interessantes para o processo de identificação.

Neste capítulo, é realizada uma análise de escolha do parâmetro κ em função da SNR (razão sinal-ruído) e da esparsidade de um sistema, de modo a garantir a competitividade do algoritmo l_0 -LMS perante o algoritmo padrão LMS.

5.2 Sistemas com resposta ao impulso esparsa

É sabido que a taxa de convergência dos algoritmos de filtragem adaptativa tende a se reduzir com o aumento do número de coeficientes a identificar. No entanto, em muitas aplicações, o sistema que está sendo modelado apresenta uma resposta ao impulso esparsa, de sorte que apenas um pequeno número de coeficientes da sua resposta impulsiva é considerado relevante. O apelo prático de tais aplicações tem impulsionado o desenvolvimento de muitos algoritmos de filtragem adaptativa destinados à exploração da natureza esparsa das funções de transferência envolvidas.

Conforme foi mencionado, se um algoritmo adaptativo pode identificar e explorar os coeficientes diferentes de zero do sistema a ser identificado, então é possível obter melhorias de desempenho e uma redução da complexidade computacional. Algoritmos clássicos como o LMS não aproveitam a esparsidade nos modelos do sistema, ignorando assim uma estrutura inerente ao problema capaz de ser explorada para aumentar a velocidade de convergência.

A fim de acelerar a taxa de convergência do processo de identificação, vários algoritmos foram propostos recentemente que utilizam a característica dessa dispersão, resultando em uma convergência mais rápida. Uma popular medida de esparsidade

de uma resposta ao impulso \mathbf{w} de comprimento L é dada por [70]:

$$\xi_{\mathbf{w}} = \frac{L}{L - \sqrt{L}} \left(1 - \frac{\|\mathbf{w}\|_1}{\sqrt{L}\|\mathbf{w}\|_2} \right), \quad (5.1)$$

onde $\|\mathbf{w}\|_p$ é a norma l_p do vetor \mathbf{w} . Cumpre observar que $0 \leq \xi_{\mathbf{w}} \leq 1$, e que $\xi_{\mathbf{w}} = 0$ quando todos os elementos de \mathbf{w} são iguais em amplitude (indicando uma resposta ao impulso maximamente dispersa) e $\xi_{\mathbf{w}} = 1$ quando apenas um dos elementos de \mathbf{w} é diferente de zero (referente a uma resposta ao impulso maximamente esparsa).

5.3 Algoritmo l_0 -LMS

Um novo algoritmo da família LMS com penalização de respostas não esparsas, abordado neste trabalho, acelera a identificação de sistemas esparsos. Especificamente, inserindo o termo penalizador na função-custo padrão do LMS, a recursão do gradiente descendente vai acelerar a convergência dos coeficientes cujos valores são nulos ou quase nulos [71]. É importante incorporar tal penalização quando a função de transferência a identificar é esparsa. A função-custo do l_0 -LMS é definida como [68]:

$$\xi_n = e_n^2 + \gamma \|\mathbf{w}_n\|_0, \quad (5.2)$$

onde a norma¹ l_0 reflete o número de coeficientes diferentes de zero em \mathbf{w}_n e $\gamma > 0$ é um fator que controla a magnitude do termo de penalização a aplicar. Tomando alguma aproximação $F_{\beta_{l_0}}(\mathbf{w}_n)$ da norma l_0 (capaz de torná-la diferenciável), temos que

$$\|\mathbf{w}_n\|_0 \approx F_{\beta_{l_0}}(\mathbf{w}_n), \quad (5.3)$$

e a equação de atualização do algoritmo l_0 -LMS é dada por:

$$w_{i,n+1} = w_{i,n} + \mu e_n x_{n-i} + \kappa f_{\beta_{l_0}}(w_{i,n}) \quad (5.4)$$

onde $k = \mu\gamma$ e uma escolha popular [68] para $f_{\beta_{l_0}}(w_n)$ que aproxima com baixo custo computacional a derivada $-\frac{\partial F_{\beta_{l_0}}(\mathbf{w}_n)}{\partial w_n}$, é dada por:

$$f_{\beta_{l_0}}(x) = \begin{cases} 2\beta_{l_0}^2 x - 2\beta_{l_0} \operatorname{sgn}(x), & |x| \leq \frac{1}{\beta_{l_0}}; \\ 0, & x = 0 \end{cases} \quad (5.5)$$

¹Em termos rigorosos, a norma l_0 é uma pseudo-norma.

onde $\text{sgn}(\cdot)$ é a função sinal definida como:

$$\text{sgn}(x) = \begin{cases} \frac{x}{|x|}, & x \neq 0 \\ 0, & \text{nos demais casos} \end{cases} \quad (5.6)$$

ou, equivalentemente:

$$f_{\beta_{l_0}}(x) = \begin{cases} \beta_{l_0}^2 x + \beta_{l_0}, & \frac{-1}{\beta_{l_0}} \leq x < 0 \\ \beta_{l_0}^2 x - \beta_{l_0}, & 0 < x \leq \frac{1}{\beta_{l_0}} \\ 0, & \text{nos demais casos} \end{cases} \quad (5.7)$$

5.4 Desempenho do algoritmo l_0 -LMS com relação aos parâmetros κ e β_{l_0}

À medida que os parâmetros κ e β_{l_0} determinam o desempenho do algoritmo l_0 -LMS, a adoção de valores ótimos desses parâmetros reveste-se de grande importância. É sabido que valores elevados de β_{l_0} concorrem para implementar uma forte intensidade de atração para zero num intervalo estreito de valores (os quais, portanto, devem estar bem próximos de zero). Assim, a escolha de um valor alto para β_{l_0} pode implicar uma exploração deficiente da característica de esparsidade [67]. De todo modo, o impacto de β_{l_0} sobre a taxa de convergência é de difícil avaliação. Para fins práticos, como em [68], adotaremos $\beta_{l_0} = 5$.

A magnitude do parâmetro κ reflete a importância da norma l_0 , ou a intensidade de atração para zero. A escolha ideal do parâmetro κ no algoritmo l_0 -LMS pode modificar profundamente o desempenho do algoritmo. Assim, um κ elevado pode resultar em uma convergência mais rápida, implicando como desvantagem o aumento do desajuste em estado estacionário. Por conseguinte, o parâmetro κ deve ser determinado tendo em vista o compromisso entre a velocidade de adaptação e a qualidade de adaptação em aplicações específicas [68].

Durante as simulações do algoritmo pode-se notar que os valores ótimos de κ em um determinado sistema são dependentes da razão sinal-ruído. Portanto, ainda que o valor ótimo de κ seja dependente da esparsidade da função de transferência a identificar, para respostas ao impulso suficientemente esparsas o impacto da SNR se revelou muito mais relevante.

Portanto, optamos por ajustar o valor de κ em termos da SNR do sistema, de modo a prescindir de uma determinação arbitrária, por parte do usuário.

5.5 Estimativa da potência de ruído

O ruído tem, tipicamente, um efeito não estacionário no espectro do sinal. Então para possibilitar uma parametrização automática do algoritmo l_0 -LMS baseada na SNR, cumpre atualizar constantemente uma estimativa da razão sinal-ruído. Para isso podemos utilizar, para estimação da potência do ruído, algoritmos como mínimos de rastreamento [72], de amostragem de média tempo-recursiva [73] ou mesmo a DFT no domínio do subespaço [74].

Pode-se também fazer essa estimação com o algoritmo NPVSS-NLMS (algoritmo não-paramétrico proposto em [75]), o qual também foi desenvolvido num contexto de identificação de sistemas.

Outra solução interessante é a proposta em [76], onde um algoritmo estima a potência do ruído do sistema de filtragem adaptativa por meio da relação entre o erro quadrado médio (MSE) e o MSE em excesso (EMSE), cujas definições são fornecidas, respectivamente, pelas Eqs. (5.8) e (5.9).

$$\text{MSE} = \lim_{n \rightarrow \infty} E|e_n|^2, \quad (5.8)$$

$$\text{EMSE} = \lim_{n \rightarrow \infty} E|\hat{e}_n|^2, \quad (5.9)$$

onde \hat{e}_n é igual a

$$\hat{e}_n \triangleq \mathbf{x}_n^T (\mathbf{s} - \mathbf{w}_n). \quad (5.10)$$

Sob algumas condições bastante relaxadas, podemos assim escrever:

$$\text{MSE} = \text{EMSE} + \sigma_v^2, \quad (5.11)$$

onde σ_v^2 é a potência do ruído do sistema.

Como pode ser visto, na literatura atual há muitas formas de estimação da SNR do sistema que poderiam ser utilizadas de modo a se poder adotar a parametrização automática proposta do algoritmo l_0 -LMS.

5.6 Simulações e Resultados

Foi utilizado nas simulações a seguir o modelo 4 de canal, segundo as recomendações da *International Telecommunication Union* (ITU) [77], com ruído aditivo gaussiano branco, com resposta ao impulso mostrada na Fig. 5.1 e $\beta_{l_0} = 5$, para demonstrar a variação do desempenho do algoritmo l_0 -LMS.

As Figs. 5.2 e 5.3 apresentam a evolução do desvio médio quadrático dos coeficientes adaptativos em relação aos reais (denominado MSD - *Mean Squared Deviation*,

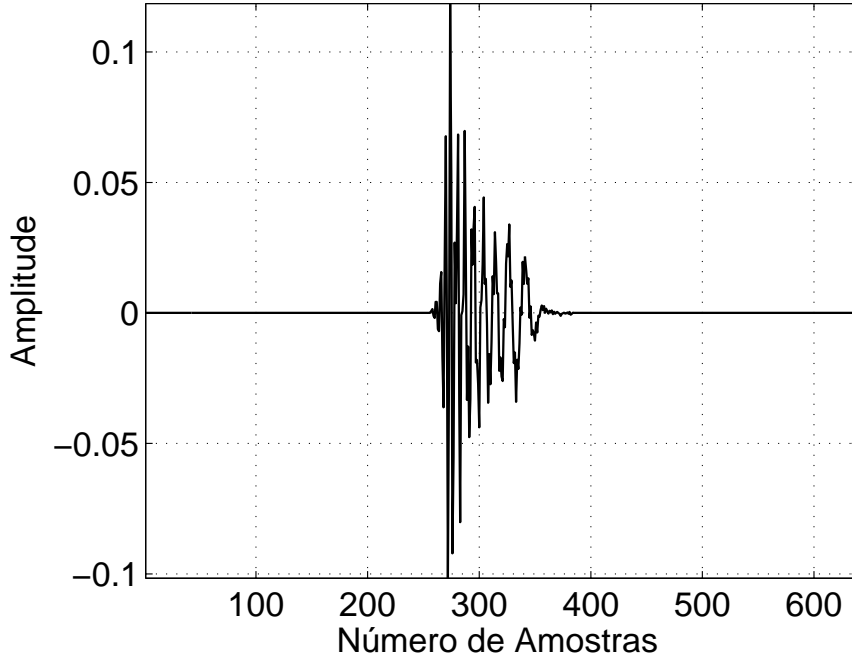


Figura 5.1: Resposta ao impulso da função de transferência de comprimento $L = 640$ utilizada nas simulações.

definido na Eq. 5.12) dos algoritmos LMS e l_0 -LMS para valores de SNR iguais a 15 dB e 28 dB, respectivamente, mantendo o parâmetro κ fixo e igual a 10^{-6} . Comparando os resultados dessas figuras, pode-se verificar que o desempenho do algoritmo l_0 -LMS com $\kappa = 10^{-6}$ foi superior ao do LMS apenas para SNR igual a 15 dB.

$$\text{MSD} = \frac{1}{L} \sum_{n=1}^L (d_n - \mathbf{x}_n^T \mathbf{w}_n)^2, \quad (5.12)$$

Para encontrar uma regra de ajuste automático do parâmetro κ , efetuou-se uma análise da variação do seu valor ótimo em função da SNR da configuração. A Fig. 5.4 mostra os valores do parâmetro κ para os quais o algoritmo l_0 -LMS resultou no menor MSD em regime permanente para SNR variando de 5 dB a 35 dB, com passo de 5 dB. Em seguida, obteve-se uma função que permite selecionar o parâmetro κ para uma determinada SNR, através da interpolação da sequência de valores ótimos da Fig. 5.4. A interpolação foi realizada por meio de duas funções: um polinômio de segundo grau para $\text{SNR} \leq 20$ dB e a função do tipo *Power* (do tipo $y = a + b|x - c|^d$, onde a, b, c e d , são as variáveis a se determinar) para $\text{SNR} > 20$ dB .

Com a função resultante, conseguiu-se chegar a um κ adaptável em função da SNR do sistema, de modo a promover um bom resultado em regime permanente do algoritmo l_0 -LMS, sem exigir uma escolha às cegas por parte do usuário do algoritmo. A Fig. 5.4 mostra em vermelho a curva de valores de κ em relação ao SNR obtida com o procedimento.

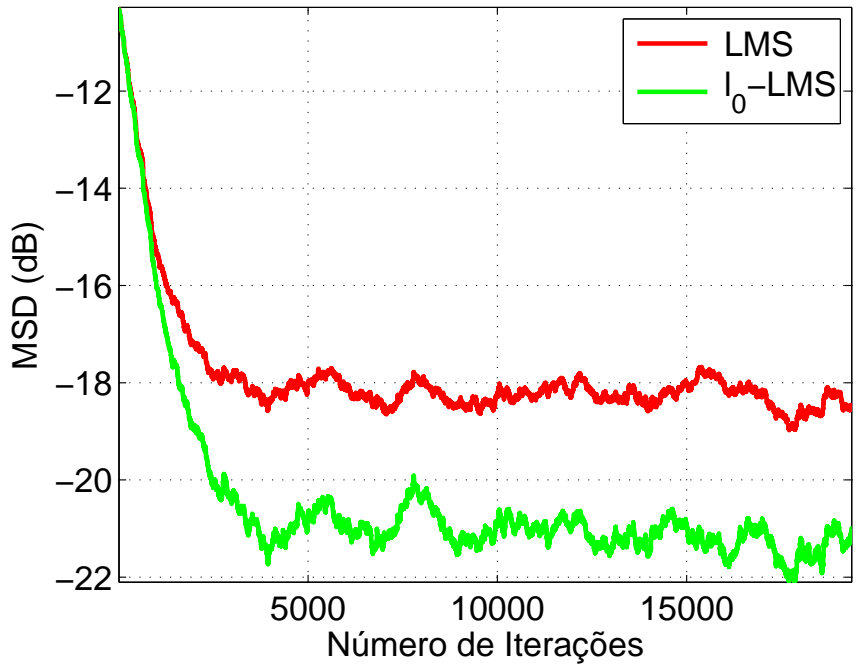


Figura 5.2: Desempenho do l_0 -LMS com $\kappa = 10^{-6}$, para SNR de 15 dB, comparado ao LMS.

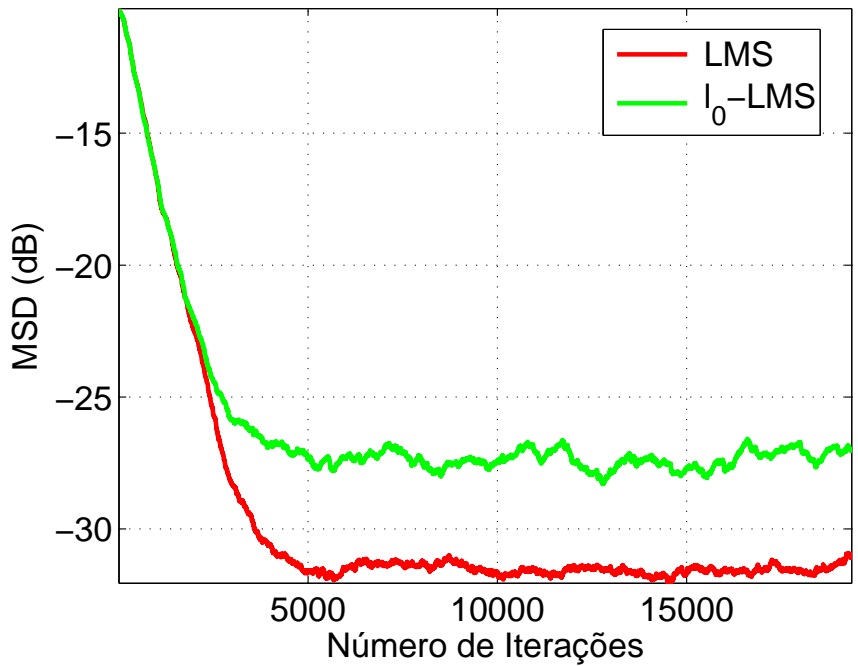


Figura 5.3: Desempenho do l_0 -LMS com $\kappa = 10^{-6}$, para SNR de 28 dB, comparado ao LMS.

Utilizado-se esse procedimento para selecionar o valor de κ , verifica-se na Fig. 5.5, que o algoritmo l_0 -LMS obteve um bom desempenho (superior ao LMS) para SNR de 28 dB. Efetuando-se diversas simulações (empregando-se todos os 8 mode-

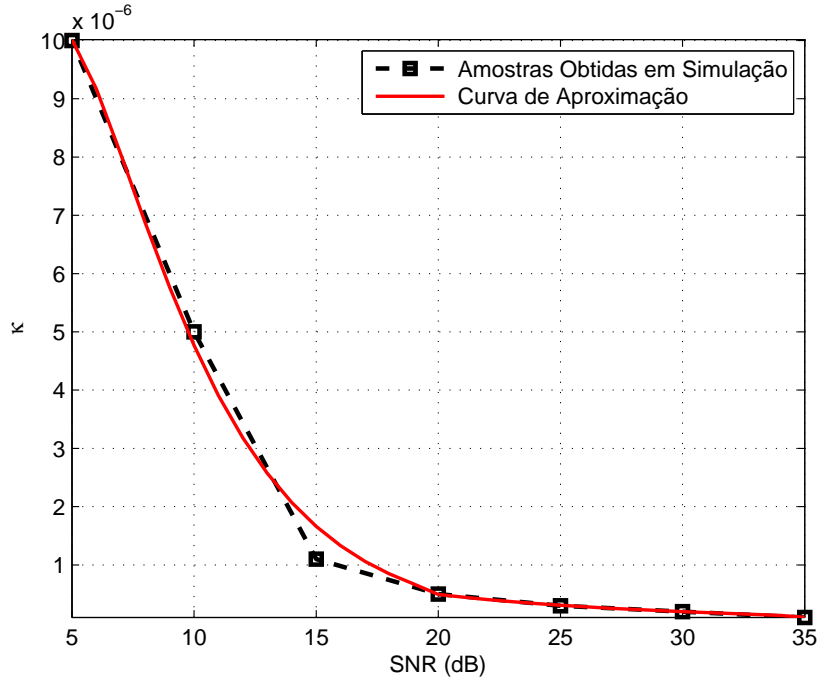


Figura 5.4: Curva de amostras e curva aproximada de valores ótimos de k para valores de SNR entre 5dB e 35dB.

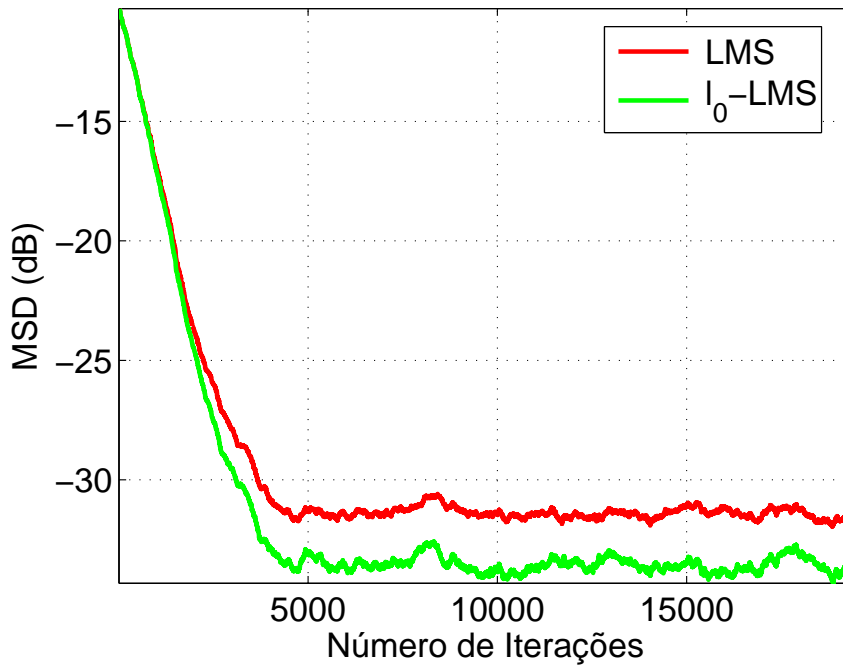


Figura 5.5: Desempenho do l_0 -LMS com $k = 2.3686 \times 10^{-7}$ obtido pelo padrão proposto para SNR de 28 dB, comparado ao LMS.

los de canais presentes em [18] e com canais com diversos graus de esparsidade), obtivemos sempre bons resultados para valores de ξ_w (dado em (4)) superiores a 0,75. Desse modo, nossa estratégia de ajuste é adequada para respostas ao im-

pulso esparsas, justamente aquelas para as quais o algoritmo l_0 -LMS foi projetado. Abaixo desse valor, observa-se uma alteração do desempenho para os valores de k encontrados, indicando a necessidade de uma variação dos mesmos.

Para fatores de esparsidade ξ_w menores do que 0.75, obtivemos, empiricamente, a seguinte função, que permite a seleção do parâmetro κ a partir dos valores de ξ_w e da SNR estimados:

$$\kappa_\xi = \frac{0.4\kappa\text{SNR}}{\max(0.05, \hat{\xi}_w)}, \quad (5.13)$$

onde $\hat{\xi}_w = 0.75 - \xi_w$ e $\max(\cdot, \cdot)$ fornece o maior de dois valores.

Para demonstrar o bom desempenho obtido com a função proposta a Fig. 5.6 apresenta o MSD em regime permanente do algoritmo l_0 -LMS, com valores de SNR iguais a = 10 dB e = 15 dB, com o valor do parâmetro κ obtido pela Eq. (5.13) e pela função da Fig. 5.4, variando-se a esparsidade do sistema desconhecido (através da inclusão de amostras nulas na sua resposta ao impulso). Cumpre observar que os resultados deste trabalho foram obtidos utilizando ruído branco gaussiano como sinal de entrada, sendo que um estudo do desempenho do ajuste proposto para sinais coloridos ainda merece maior atenção.

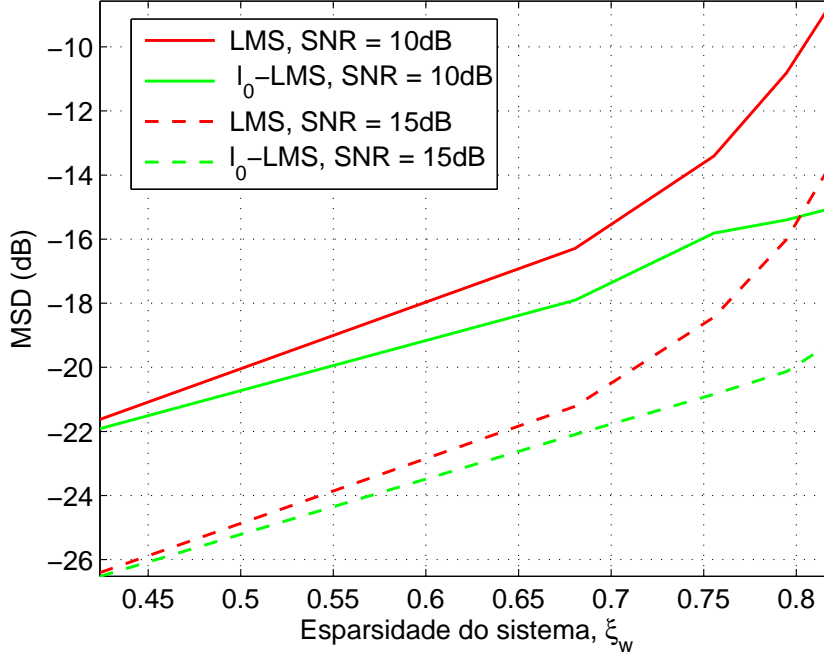


Figura 5.6: Desempenho do l_0 -LMS com κ obtido pelo padrão proposto *versus* esparsidade (ξ_w) para uma SNR de 10 dB e 15 dB, comparado ao LMS.

5.7 Comentários Finais

O algoritmo l_0 -LMS, proposto para melhorar o desempenho em identificação de sistemas esparsos, pode apresentar um funcionamento aquém do desejável, a depender do valor escolhido do parâmetro κ (responsável por controlar a atração dos coeficientes para zero). Neste capítulo, uma forma de ajuste automático do valor de κ para respostas esparsas foi proposta, de modo que o desempenho do algoritmo l_0 -LMS se torne vantajoso, quando o sinal de entrada é um ruído branco e gaussiano.

Os resultados numéricos de simulação indicam que o κ encontrado pela função empírica proposta obteve um excelente desempenho para sistemas com fatores de esparsidade $\xi_{\mathbf{w}}$ maiores do que 0.75. Para sistemas com $\xi_{\mathbf{w}} < 0.75$, verificou-se uma necessidade da alteração do valor de κ encontrado conforme o valor de $\xi_{\mathbf{w}}$, obtendo-se uma nova função que relaciona os mesmos. Ao final, propôs-se um novo modo de ajuste, capaz de contemplar variações da SNR e do fator de esparsidade $\xi_{\mathbf{w}}$.

Essa melhoria pode evidentemente aumentar o uso do algoritmo l_0 -LMS, uma vez que não seria mais necessário arbitrar um valor de κ para o sistema e presumir que o mesmo apresente um bom resultado. Os trabalhos futuros poderão incluir um estudo de como tornar esse procedimento viável, através de uma função dos parâmetros de entrada do sistema, para qualquer sistema dado, e não mais utilizando um ajuste de curvas, conseguindo-se chegar automaticamente ao κ ótimo.

Capítulo 6

Separação Cega de Fontes no Domínio do Tempo

Neste capítulo será apresentada a separação cega de fontes no domínio do tempo através do algoritmo TRINICON e posteriormente será proposta sua aplicação no caso da separação cega de fontes em subbandas.

A separação cega de fontes no domínio da frequência pode apresentar problemas de permutação no sistema de separação de cada *bin* de frequência. A separação no domínio do tempo pode ser uma saída para evitar problemas desse tipo. O TRINICON (*Triple-N ICA for Convolutional Mixtures*) [4] é um algoritmo de separação cega de fontes no domínio do tempo baseado nas três propriedades do sinal descritas no Capítulo 3: não-gaussianidade, não-brancura e não-estacionariedade. O objetivo é encontrar um conjunto de filtros de separação que minimiza a distância “*Kullback-Leiber*” (equivalente a minimizar a informação mútua) entre os sinais originais e a saída do sistema de separação. Essa medida requer o conhecimento das funções de densidade de probabilidade de origem, que, em geral, não são conhecidas antecipadamente. Uma simplificação que pode ser utilizada é assumir sinais Gaussianos, simplificando assim as equações de atualização de filtro com o custo de diminuir o desempenho do sistema de separação.

6.1 TRINICON

O algoritmo TRINICON explora a não-gaussianidade através de estatísticas de ordens elevadas, utilizadas para formular sua função-custo. O espectro colorido é explorado através da diagonalização simultânea de matrizes de correlação das saídas para múltiplos atrasos, e por fim, a não-estacionariedade é explorada através da diagonalização simultânea das matrizes de correlação de saída para diferentes instantes de tempo.

6.1.1 Notação Matricial para Misturas Convolutivas

Para derivar um algoritmo para processamento de bloco de misturas convolutivas, primeiro precisamos formular o sistema de separação como um sistema MIMO (*Multiple-Input Multiple-Output*) composto de filtros FIR de comprimento N . Além do comprimento do filtro N e do número de canais P , é necessário introduzir mais dois parâmetros para a formulação geral:

- Número de defasamentos temporais D considerados para as correlações na função custo ($1 \leq D \leq N$);
- Comprimento de blocos de sinal de saída L , como base para as estimativas de correlações.

Para expressar a convolução linear, a matriz \mathbf{W}_{pq} de dimensão $2N \times D$ que contém os coeficientes do filtro de separação em cada coluna, é definida como:

$$\mathbf{W}_{pq}(b) = \begin{bmatrix} w_{pq,0} & 0 & \cdots & 0 \\ w_{pq,1} & w_{pq,0} & \ddots & \vdots \\ \vdots & w_{pq,1} & \ddots & 0 \\ w_{pq,L-1} & \vdots & \ddots & w_{pq,0} \\ 0 & w_{pq,L-1} & \ddots & w_{pq,1} \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & w_{pq,L-1} \\ 0 & \cdots & 0 & 0 \\ \vdots & \cdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix} \quad (6.1)$$

Os vetores contendo os dados dos sinais de entrada $\mathbf{x}_p(b, j)$ e do sinal de saída $\mathbf{y}_q(b, j)$ do b -ésimo bloco são dados por:

$$\mathbf{x}_p(b, j) = \begin{bmatrix} x_p(bN + j) & \cdots & x_p(bN - 2N + 1 + j) \end{bmatrix} \quad (6.2)$$

$$\mathbf{y}_q(b, j) = \begin{bmatrix} y_q(bN + j) & \cdots & y_q(bN - D + 1 + j) \end{bmatrix} \quad (6.3)$$

Define-se um bloco de vetor do sinal de saída com o comprimento $L \geq PD$ (para obter uma matriz de correlação de posto completo), de acordo com [78]:

$$\mathbf{y}_q(b, j) = \sum_{p=1}^P \mathbf{x}_p(b, j) \mathbf{W}_{pq}(b) \quad (6.4)$$

onde b é o índice do bloco e j é o deslocamento no tempo dentro do bloco de tamanho L e varia de 0 a $L - 1$. Para permitir uma notação mais compacta e já considerando o número de fontes igual ao número de sensores, ou seja, $Q = P$, as equações (6.4) podem ser reescritas na forma abaixo:

$$\mathbf{y}(b, j) = \mathbf{x}(b, j)\mathbf{W}(b) \quad (6.5)$$

onde

$$\mathbf{x}(b, j) = \begin{bmatrix} x_1(b, j) & \dots & x_P(b, j) \end{bmatrix} \quad (6.6)$$

$$\mathbf{y}(b, j) = \begin{bmatrix} y_1(b, j) & \dots & y_P(b, j) \end{bmatrix} \quad (6.7)$$

e

$$\mathbf{W}(b) = \begin{bmatrix} \mathbf{W}_{11}(b) & \dots & \mathbf{W}_{1P}(b) \\ \vdots & \ddots & \vdots \\ \mathbf{W}_{P1}(b) & \dots & \mathbf{W}_{PP}(b) \end{bmatrix} \quad (6.8)$$

Pode ser observado que, para o caso geral $1 \leq D \leq N$, as últimas $N - D + 1$ linhas da matriz $\mathbf{W}_{pq}(b)$ são preenchidos com zeros para assegurar a compatibilidade com a matriz $\mathbf{X}_p(b)$. Finalmente, para permitir que uma notação conveniente do algoritmo combinando todos os canais, podemos escrever:

$$\mathbf{Y}(b) = \mathbf{X}(b)\mathbf{W}(b) \quad (6.9)$$

onde

$$\mathbf{X}(b) = \begin{bmatrix} \mathbf{X}_1(b) & \dots & \mathbf{X}_P(b) \end{bmatrix} \quad (6.10)$$

e

$$\mathbf{Y}(b) = \begin{bmatrix} \mathbf{Y}_1(b) & \dots & \mathbf{Y}_P(b) \end{bmatrix} \quad (6.11)$$

6.1.2 Função Custo

Existem diferentes abordagens para estimar cegamente a matriz \mathbf{W} . Utilizando as seguintes propriedades dos sinais de voz das fontes de forma combinada [79], tem-se um algoritmo eficiente e versátil:

- Não-gaussianidade é explorada por meio de estatísticas de ordem superior para a análise de componentes independentes (ICA). Essas abordagens podem ser divididas em várias classes. Embora todas elas conduzam a regras de atualização semelhantes, a minimização da informação mútua (MMI) entre os canais de saída pode ser considerada como a abordagem mais geral para a BSS [79]. Para obter um estimador que não só permite a separação espacial, mas também a separação temporal, utilizamos a distância de Kullback-Leibler

(KLD) [80] entre uma certa PDF conjunta desejada e a PDF conjunta dos sinais estimados de saída;

- Não-brancura é explorada por minimização simultânea de correlações cruzadas dos sinais de saída ao longo de vários intervalos de tempo. Consideramos, portanto, PDFs multivariáveis, incluindo D desfasamentos temporais;
- Não-estacionariedade é explorada por minimização simultânea de correlações cruzadas dos sinais de saída em diferentes instantes. Assumimos ergodicidade dentro de blocos de comprimento L , de modo que as médias estatísticas do conjunto passa a ser igual às médias temporais ao longo desses blocos.

Com base na KLD, vamos agora definir a seguinte função de custo geral tendo em conta todas as três propriedades fundamentais do sinal de voz:

$$\mathcal{J} = - \sum_{i=0}^b \beta(i, b) \frac{1}{L} \sum_{j=0}^{L-1} \{ \log(\hat{p}_D(\mathbf{y}(i, j))) - \log(\hat{p}_{PD}(\mathbf{y}(i, j))) \} \quad (6.12)$$

onde $\hat{p}_D(\cdot)$ representa a PDF de cada sensor de dimensão D e $\hat{p}_{PD}(\cdot)$ representa a PDF conjunta de dimensão PD para todos os sensores.

Ao assumir PDF gaussiana, dada por

$$\hat{p}_D(\mathbf{y}_p(i, j)) = \frac{1}{\sqrt{2\pi^D \det(\mathbf{R}_{pp}(i))}} e^{\frac{1}{2} \mathbf{y}_p(i, j) \mathbf{R}_{pp}^{-1}(i) \mathbf{y}_p^H(i, j)} \quad (6.13)$$

o algoritmo passa a utilizar apenas as características de não-estacionariedade e espectro colorido, tornando o algoritmo mais simples computacionalmente.

Como mencionado anteriormente, o parâmetro D expressa o tamanho da memória para modelar o espectro colorido do sinal e deve ser escolhido entre $1 \leq D \leq N$. O parâmetro L representa o tamanho do bloco em que o sinal seria considerado estacionário. Para uma estimação adequada da PDF, devem ser considerados $L \geq PD$ instantes de tempo.

A função $\beta(i, b)$ é uma função peso para permitir implementações *offline*, *online* e *block-online* do algoritmo e deve ser normalizada de acordo com:

$$\sum_{i=0}^{\infty} \beta(i, b) = 1 \quad (6.14)$$

- Implementação *offline*: $\beta(i, b) = 1/B$, em que B é o número total de blocos;

- Implementação *online*:

$$\beta(i, m) = \begin{cases} (1 - \lambda)\lambda^{(m-i)} & \text{para } 0 \leq i \leq m \\ 0 & \text{caso contrário} \end{cases} \quad (6.15)$$

onde λ é o fator de esquecimento, geralmente um pouco menor do que 1;

- Implementação *block-online*: $\beta(i, b) = 1/K$, em que apenas os últimos K blocos são levados em consideração pelo algoritmo.

6.1.3 Regra de atualização

Analogamente ao sistema de separação \mathbf{W} , este sistema pode ser expresso por uma matriz \mathbf{C} do tipo *Sylvester* de dimensões adequadas. O processo de mistura é de modo análogo descrito por $\mathbf{X} = \mathbf{S}\mathbf{H}$, onde \mathbf{S} é a matriz $L \times P(U + N - 1)$ dos sinais das fontes com deslocamentos de tempo, e \mathbf{H} é a matriz $P(U + N - 1) \times 2PN$ de mistura na estrutura de *Sylvester*. As dimensões dessas matrizes resultam da condição de linearidade da convolução.

Assim, é possível obter uma matriz diagonal por blocos arbitrária, $\mathbf{C} = \mathbf{H}\mathbf{W}$, onde $\mathbf{C} - \text{bdiag}\{\mathbf{C}\} = \text{boff}\{\mathbf{C}\} = \mathbf{0}$, como apresentado em [24]. A operação bdiag em uma matriz formada por várias submatrizes define todas as submatrizes que não pertencem à diagonal principal para zero. Aqui, as submatrizes referem-se aos diferentes canais de sinal. Analogamente, a operação boff atribui zeros aos elementos de todas as submatrizes presentes na diagonal principal.

Para alcançar este objetivo, serão utilizadas soluções de banda larga com base em estatísticas de segunda ordem. Assim, os sinais estocásticos são descritos aqui por uma sequência de matrizes reduzidas de correlação temporal com atrasos. Definem-se então as matrizes de correlação:

$$\mathbf{R}_{xx}(b) = \mathbf{X}^H(b)\mathbf{X}(b) \quad (6.16)$$

e

$$\mathbf{R}_{yy}(b) = \mathbf{Y}^H(b)\mathbf{Y}(b) \quad (6.17)$$

de tamanho $2PN \times 2PN$ e $PD \times PD$, respectivamente. Note-se que a matriz (6.17) é posto completo desde que $L \geq PD$. Com a formulação de matriz compacta $\mathbf{Y}(b) = \mathbf{X}(b)\mathbf{W}$, pode-se agora definir a seguinte função de custo, com base na sequência de matrizes de correlação reduzidas e atrasadas como:

$$\mathcal{J} = - \sum_{i=0}^b \beta(i, b) \{ \log[\det(\text{bdig}(\mathbf{Y}^H(i)\mathbf{Y}(i)))] - \log[\det(\mathbf{Y}^H(i)\mathbf{Y}(i))] \} \quad (6.18)$$

em que o ponto de equilíbrio corresponde exatamente à solução da BSS desejada, ou seja, $\text{boff}\{\mathbf{C}\} = \mathbf{0}$. Considerando um algoritmo baseado no método do gradiente, e para minimização da função custo, usando o cálculo do gradiente natural [81] de \mathcal{J} em relação à matriz de separação \mathbf{W} temos:

$$\Delta\mathbf{W} \propto \mathbf{W}\mathbf{W}^H \frac{\partial\mathcal{J}}{\partial\mathbf{W}^*} \quad (6.19)$$

de onde é obtida a seguinte regra de atualização:

$$\mathbf{W}(b) = \mathbf{W}(b-1) - \mu\Delta\mathbf{W}(b) \quad (6.20)$$

onde

$$\Delta\mathbf{W}(b) = 2 \sum_{i=0}^{\infty} \beta(i, b) \mathbf{W}(b) \{ \mathbf{R}_{yy}(i) - \text{bdiag}\mathbf{R}_{yy}(i) \} \text{bdiag}^{-1}\mathbf{R}_{yy}(i) \quad (6.21)$$

Consideramos agora o caso $P = Q = 2$ por simplicidade. Neste caso, temos:

$$\Delta\mathbf{W}(b) = 2 \sum_{i=0}^{\infty} \beta(i, b) \mathbf{W}(b) \begin{bmatrix} \mathbf{0} & \mathbf{R}_{y_1y_2}\mathbf{R}_{y_2y_2}^{-1} \\ \mathbf{R}_{y_2y_1}\mathbf{R}_{y_1y_1}^{-1} & \mathbf{0} \end{bmatrix} \quad (6.22)$$

equivalente à:

$$\Delta\mathbf{W}(b) = 2 \sum_{i=0}^{\infty} \beta(i, b) \begin{bmatrix} \mathbf{W}_{12}\mathbf{R}_{y_2y_1}\mathbf{R}_{y_1y_1}^{-1} & \mathbf{W}_{11}\mathbf{R}_{y_1y_2}\mathbf{R}_{y_2y_2}^{-1} \\ \mathbf{W}_{22}\mathbf{R}_{y_2y_1}\mathbf{R}_{y_1y_1}^{-1} & \mathbf{W}_{21}\mathbf{R}_{y_1y_2}\mathbf{R}_{y_2y_2}^{-1} \end{bmatrix} \quad (6.23)$$

Por fim, a equação de atualização do algoritmo do modo *off-line*, em que todo o sinal é processado para gerar uma iteração, é dada por:

$$\mathbf{W}(b) = \mathbf{W}(b-1) - \frac{2\mu}{B} \sum_{b=0}^B \begin{bmatrix} \mathbf{W}_{12}\mathbf{R}_{y_2y_1}\mathbf{R}_{y_1y_1}^{-1} & \mathbf{W}_{11}\mathbf{R}_{y_1y_2}\mathbf{R}_{y_2y_2}^{-1} \\ \mathbf{W}_{22}\mathbf{R}_{y_2y_1}\mathbf{R}_{y_1y_1}^{-1} & \mathbf{W}_{21}\mathbf{R}_{y_1y_2}\mathbf{R}_{y_2y_2}^{-1} \end{bmatrix} \quad (6.24)$$

onde $\mathbf{R}_{y_p y_q}$, $p, q \in \{1, 2\}$, são as submatrizes de dimensão $D \times D$ correspondentes de \mathbf{R}_{yy} , i é o número de iterações (*off-line*) e μ é o passo de adaptação do algoritmo.

A atualização dessa função resulta numa complexidade alta para o cálculo de $\mathbf{R}_{qq}^{-1}(b)$ em uma aplicação simples. No entanto, para reduzir a complexidade computacional do algoritmo, os fatores de normalização $\mathbf{R}_{qq}^{-1}(b)$ podem ser simplificados considerando $\mathbf{R}_{qq}(b)$ uma matriz diagonal; desta maneira, $\mathbf{R}_{qq}^{-1}(b)$ pode ser obtida invertendo os elementos da sua diagonal, que correspondem aos inversos das potências dos blocos atrasados no tempo do sinal da q -ésima saída, ou seja:

$$\mathbf{R}_{qq}(b) \approx \text{diag}\{\mathbf{R}_{qq}(b)\} \quad (6.25)$$

Para reduzir ainda mais o custo computacional no cálculo de $\mathbf{R}_{qq}^{-1}(b)$, foi proposto em [82] se reduzir o fator de normalização a um escalar, ou seja:

$$\mathbf{R}_{qq}(b) \approx (\mathbf{y}_q(b))^T \mathbf{y}_q(b) \mathbf{I} \quad (6.26)$$

6.2 BSS em Subbandas no Domínio do Tempo

Nesta seção, algumas modificações foram feitas no algoritmo TRINICON na tentativa de melhorar a medida de avaliação SIR. Apresentamos um novo método para separação cega de misturas convolutivas em subbandas, semelhante aos métodos apresentados em [63] e [83], porém com algumas diferenças, como não necessitar dos filtros de síntese, apenas dos de análise, e realizar a atualização de \mathbf{W} em banda completa.

Devido aos filtros de separação em subbandas poderem ter ordens menores com melhores resultados, o método proposto apresenta menor complexidade computacional, maior taxa de convergência e maior razão sinal-interferência em regime estacionário, quando comparado ao algoritmo correspondente em banda cheia. Outra vantagem do método proposto consiste no uso de filtros com coeficientes reais, que são atraentes para implementação em dispositivos para processamento de sinais (DSPs).

Nesta seção apresentamos as modificações necessárias no algoritmo BSS em blocos no domínio do tempo TRINICON descrito anteriormente. A idéia é explorar as características de melhoria da taxa de convergência e de redução da complexidade computacional inerentes às estruturas em subbandas. Neste trabalho não se utilizará decimação dos sinais nas subbandas, ficando essa proposta para futuros estudos.

Para as simulações em subbandas, cada sinal observado da mistura $x_p(n)$ é decomposto pelos filtros de análise $\mathbf{F}_m(z)$, sendo os sinais resultantes então filtrados pelos subfiltros de separação de banda completa dados por $\mathbf{W}_{pq}(n)$, gerando sinais de saída para a m -ésima subbanda utilizados no cálculo de atualização da nova matriz de separação \mathbf{W} . A atualização da matriz \mathbf{W} se dá em banda completa utilizando um somatório da função $\Delta \mathbf{W}_m$ de cada subbanda, similar ao apresentado no algoritmo NSAF mostrado no Capítulo 2. Para ajustar os coeficientes de cada subfiltro de separação, empregamos o algoritmo *off-line* em banda cheia TRINICON adaptado para subbandas, sendo a equação de atualização dos coeficientes dos filtros de separação dada por:

$$\mathbf{W}(i) = \mathbf{W}(i-1) - \frac{2}{b} \sum_{m=1}^M \sum_{b=1}^{B_m} \begin{pmatrix} \mathbf{W}_{12} \mathbf{R}_{21}^m \mathbf{R}_{11}^{m-1} & \mathbf{W}_{11} \mathbf{R}_{12}^m \mathbf{R}_{22}^{m-1} \\ \mathbf{W}_{22} \mathbf{R}_{21}^m \mathbf{R}_{11}^{m-1} & \mathbf{W}_{21} \mathbf{R}_{12}^m \mathbf{R}_{22}^{m-1} \end{pmatrix} \begin{pmatrix} \mu_1 \mathbf{I} & 0 \\ 0 & \mu_2 \mathbf{I} \end{pmatrix} \quad (6.27)$$

sendo M o número de subbandas, B_m o número de blocos para cada subbanda e

$$\mathbf{R}_{pq}^m(b) = [\mathbf{Y}_p^m(b)]^T \mathbf{Y}_q^m(b) \mathbf{I} \quad (6.28)$$

onde

$$\mathbf{Y}_p^m(b) = \begin{bmatrix} y_q^m(bN) & \dots & y_q^m(bN - D_m + 1) \\ y_q^m(bN + 1) & \dots & y_q^m(bN - D_m + 2) \\ \vdots & \ddots & \vdots \\ y_q^m(bN + L_m - 1) & \dots & y_q^m(bN - D_m + L_m) \end{bmatrix} \quad (6.29)$$

sendo a matriz $\mathbf{R}_{pq}^m(b)$ de dimensão $D_m \times D_m$ (com $1 \leq D_m \leq N$) e $\mathbf{Y}_p^m(b)$ de $L_m \times D_m$ (com $L_m \geq D_m$), L_m é o tamanho de cada bloco, μ_q é o q -ésimo passo de adaptação, i é o número de iterações (*offline*) e y_q^m é o sinal da q -ésima saída na m -ésima subbanda. Para reduzir a complexidade computacional do algoritmo, o fator de normalização $\mathbf{R}_{pq}^{m-1}(b)$ pode ser simplificado para um escalar, como feito no modo banda cheia, ou seja:

$$\mathbf{R}_{qq}^m(b) \approx (\mathbf{y}_q^m(b))^T \mathbf{y}_q^m(b) \mathbf{I} \quad (6.30)$$

com $\mathbf{y}_q^m(b)$ correspondendo à primeira coluna da matriz de blocos atrasados no tempo.

6.3 Fatores de Aprendizagem Variantes

Em geral, a escolha adequada do tamanho de passo é muito importante para o desempenho de algoritmos adaptativos. O algoritmo TRINICON, mesmo na sua versão *off-line*, é suscetível à instabilidade, particularmente em regime permanente. Tal instabilidade pode ser significativa mesmo quando o fator de aprendizagem é suficientemente pequeno para promover uma aprendizagem bastante lenta e suave, de sorte que, neste caso, a convergência excessivamente lenta não é garantia de convergência para uma boa solução. Essa característica indesejável é um empecilho para um emprego efetivo do algoritmo TRINICON.

Por essa razão, utilizaremos um mecanismo robusto de adaptação do fator de aprendizagem apresentado em [84], de modo a evitar que um aumento na taxa de convergência se traduza em instabilidade ou perda de desempenho em regime permanente, mostrado abaixo:

$$\mu(i+1) = \begin{cases} \alpha * \mu(i), & \text{se } f_{wpq} > f_{pq} \\ \mu(i) = \mu(0), & \text{se } f_{wpq} < f_{pq} \text{ e } \mu(i) > \mu(0) \\ \mu(i)/\alpha, & \text{nos demais casos} \end{cases} \quad (6.31)$$

sendo f_{wpq} a função de proporção dos coeficientes adaptativos dos filtros $\mathbf{W}_{pq}(b)$ que não alteram sua direção de atualização [84], sendo $f_{pq} = 0.7$ e $\alpha = 1.02$.

6.4 Simulações

Os algoritmos adaptativos para separação supervisionada de fontes descritos neste trabalho foram avaliados empregando-se duas misturas convolutivas de dois sinais de voz, supondo $Q = P$, ou seja, número de fontes igual ao número de sensores e $Q = 2$, apenas duas fontes, sendo a mistura composta de um sinal de voz masculino e outro feminino, disponibilizados em [85], com duração de 10 s e frequência de amostragem 16 kHz.

As simulações da propagação acústica de salas foram desenvolvidas usando o modelo "Image-Source" descrito em [16], reamostrando os sinais de voz em 8 kHz. Além disso, para gerar as misturas, foram utilizados os parâmetros da Tabela 6.1.

Tabela 6.1: Valores utilizados nas simulações

Parâmetro	Valor
Ângulo da fonte 1	-45°
Ângulo da fonte 2	45°
Distância entre os microfones	5 cm
Coefficiente de absorção das paredes	0.8
Distância das fontes até o centro do arranjo de microfones	1 m

Foram feitas simulações para três diferentes tempos de reverberação (T_{60}): 0.1 s, 0.3 s e 0.5 s. A medida de avaliação utilizada foi a SIR (*Signal to Interference Ratio*). Para a decomposição em subbandas, empregou-se um banco de filtros modulado por cosseno [65] com $M = 4$ subbandas e filtro protótipo de comprimento 32. O tamanho dos blocos foi fixado em $2D$, sendo o valor do atraso D utilizado igual ao comprimento do filtro de separação N . O fator de aprendizagem utilizado foi o mostrado na seção anterior, com valor inicial variável de $\mu = 0.001$ a $\mu = 0.003$, dependendo do comportamento do algoritmo no caso aplicado.

A Figura 6.1 apresenta as evoluções da SIR ao longo das iterações obtidas com os algoritmos em banda cheia e em subbandas para $T_{60} = 0.1$ s, com filtros de separação de comprimentos iguais aos comprimentos dos filtros de mistura $N = 1024$ e $N = 512$ para o algoritmo em banda cheia e $N = 512$ para o algoritmo em subbandas. Pode-se verificar nessa figura que o algoritmo TRINICON em subbandas apresenta desempenho superior, chegando a uma SIR de 24.6 dB, enquanto que o algoritmo em banda cheia alcançou o valor de 17.54 dB com o mesmo comprimento de filtro ($N = 512$) e de 18.09 dB com $N = 1024$, bem inferiores ao conseguido com o método em subbandas.

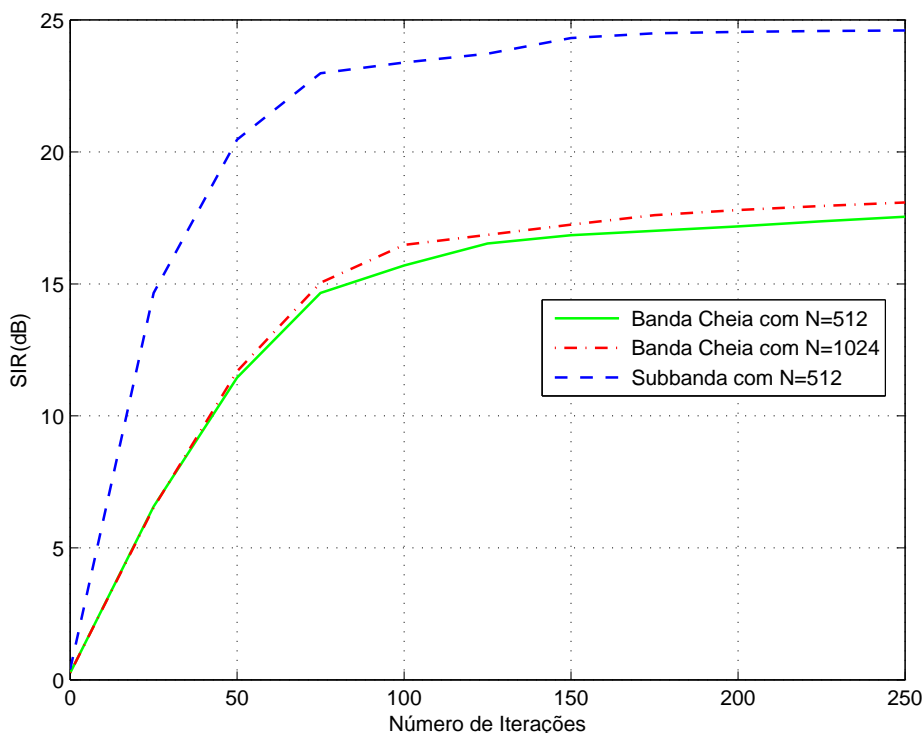


Figura 6.1: Evolução da SIR (em dB) para $T60 = 0.1$ s para BSS no domínio do tempo em subbandas e em banda cheia..

Na Figura 6.2 são mostrados os resultados das simulações para $T60 = 0.3$ s do algoritmo em subbandas com filtros de separação de comprimentos $N = 512$ e do algoritmo em banda cheia com $N = 512$ e com $N = 1024$. Na Figura 6.3 são mostrados os resultados das simulações para $T60 = 0.5$ s do algoritmo em subbandas com $N = 1024$ e do algoritmo em banda cheia com $N = 2048$.

Aqui novamente pode-se verificar que o algoritmo TRINICON aplicado em subbandas apresenta melhor convergência comparado ao em banda cheia, mesmo com um comprimento de filtro menor. Para o tempo de reverberação de $T60 = 0.3$ s, o algoritmo em banda cheia alcançou o valor de 10.22 dB e o em subbandas de 10.72 dB; já para $T60 = 0.5$ s obteve-se 5.69 dB e 6.27dB para os algoritmos em banda cheia e em subbandas, respectivamente, o que demonstra um ganho no processo de separação tanto na SIR como no custo computacional, além da velocidade da convergência, que ocorreu sempre antes no algoritmo em subbanda.

Porém, é possível observar que os valores encontrados para um tempo de reverberação alto com os algoritmos em banda cheia e em subbandas ainda são insuficientes para uma separação de fontes aceitável aos ouvidos humanos. No próximo capítulo serão apresentados esses mesmos métodos no domínio da frequência, visando obter, através de um pequeno aumento no custo computacional, melhores desempenhos para misturas com reverberações altas.

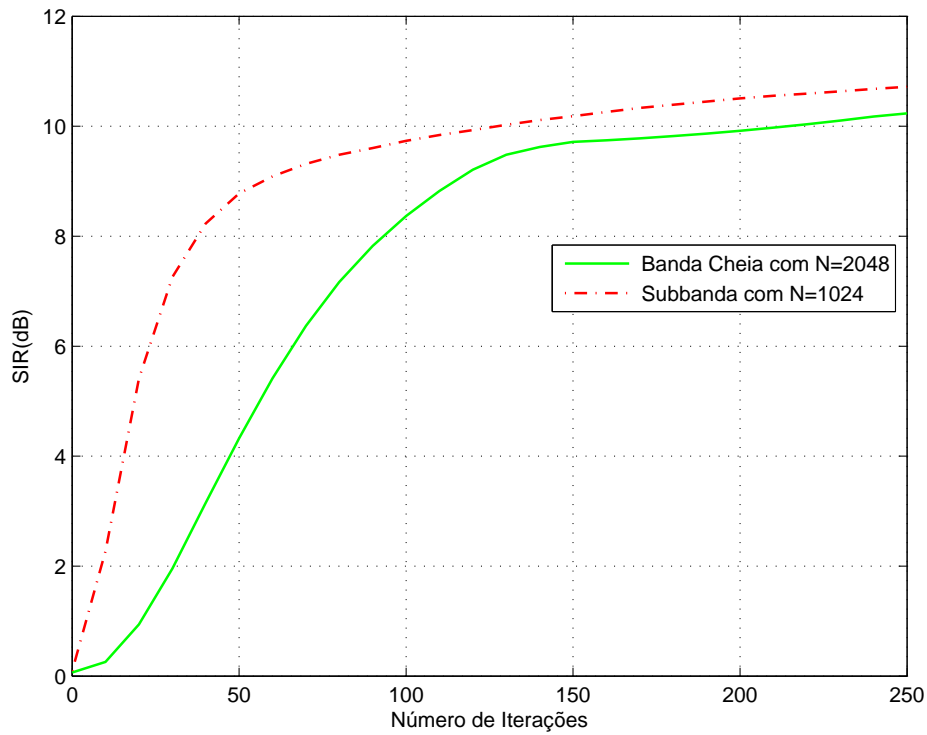


Figura 6.2: Evolução da SIR (em dB) para $T_{60} = 0.3$ s para BSS no domínio do tempo em subbandas e em banda cheia.

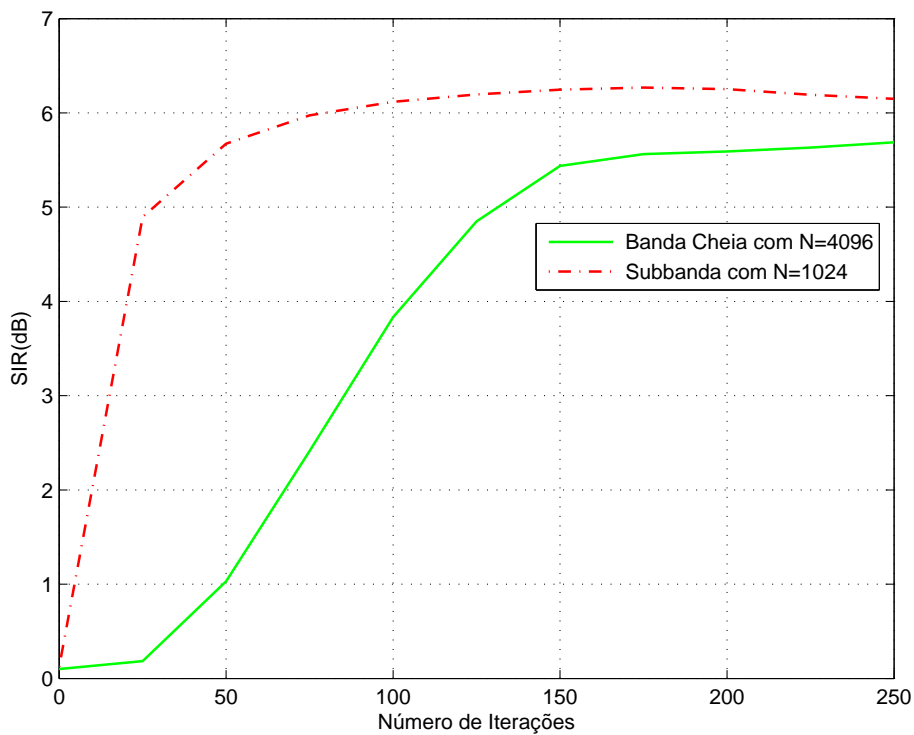


Figura 6.3: Evolução da SIR(dB) para $T_{60} = 0.5$ s para BSS no domínio do tempo em subbandas e em banda cheia.

Capítulo 7

Separação Cega de Fontes no Domínio da Frequência

Neste capítulo apresentamos um método de separação cega de fontes (BSS) para misturas convolutivas que combina as vantagens de algoritmos de banda larga com a eficiência computacional de técnicas de banda estreita. O método proposto baseia-se no algoritmo de banda larga TRINICON apresentado no capítulo anterior. Por aplicação do teorema “*Szegő*”, que se refere às propriedades Toeplitz e circulantes de matrizes, uma nova normalização é apresentada. O novo esquema, portanto, resulta em um algoritmo de convergência rápida e eficiente computacionalmente, enquanto ainda evita problemas típicos de banda estreita, tais como os efeitos de permutações e circularidade.

Além disso, a aplicação desse algoritmo no domínio da frequência em subbandas, seguindo o que foi feito no capítulo anterior, é proposto, resultando em um novo algoritmo com desempenho computacional e convergência melhores que os do algoritmo em banda cheia. Os resultados das simulações em ambientes acústicos confirmam o melhor desempenho do novo algoritmo em comparação com aproximações anteriores, principalmente em ambientes com altas reverberações. Esses cenários desafiadores são o foco deste capítulo, uma vez que os algoritmos apresentados no capítulo anterior não forneceram bons resultados para esses ambientes.

Ao final deste capítulo, as simulações realizadas com o algoritmo proposto serão apresentadas, comparando seus resultados aos encontrados no capítulo anterior.

7.1 Formulação Equivalente do algoritmo TRINICON no domínio DFT

Em [86], foi demonstrado que qualquer matriz Toeplitz pode ser expressa de forma equivalente no domínio da frequência pela geração de uma matriz circulante por

extensão apropriada da matriz Toeplitz. Em seguida, a matriz circulante é diagonalizada aplicando-se a matriz DFT \mathbf{F}_R , de tamanho $R \times R$, onde $R \geq L + N$ representa o comprimento da transformada. Esses dois passos são aplicados em [87] à matriz Toeplitz do sinal de saída $\mathbf{Y}_q(b)$, conforme as equações:

$$\mathbf{Y}_q(b) = \mathbf{W}_{L+N \times R}^{01_{L+N}} \mathbf{C}_{\mathbf{Y}_q}(b) \mathbf{W}_{R \times N}^{1_{N0}} \quad (7.1)$$

ou

$$\mathbf{Y}_q(b) = \mathbf{W}_{L+N \times R}^{01_{L+N}} \mathbf{F}_R^{-1} \underline{\mathbf{Y}}_q(b) \mathbf{F}_R \mathbf{W}_{R \times N}^{1_{N0}} \quad (7.2)$$

onde $\mathbf{C}_{\mathbf{Y}_q}(b)$ é uma matriz circulante de tamanho $R \times R$, e as matrizes correspondentes ao “janelamento” dos sinais são dadas por:

$$\mathbf{W}_{L+N \times R}^{01_{L+N}} = [\mathbf{0}_{L+N \times R-L-N}, \mathbf{I}_{L+N \times L+N}] \quad (7.3)$$

e

$$\mathbf{W}_{R \times N}^{1_{N0}} = [\mathbf{I}_{N \times N}, \mathbf{0}_{R-N \times N}] \quad (7.4)$$

Neste trabalho, a convenção utilizada é que o índice inferior da matriz de “janelamento” denota as suas dimensões e o índice superior descreve as posições dos uns e zeros. O tamanho das submatrizes é indicado em subscripto (por exemplo, “ 01_L ”). A matriz $\underline{\mathbf{Y}}_q(b)$ exibe uma estrutura diagonal contendo os valores da matriz circulante $\mathbf{C}_{\mathbf{Y}_q}(b)$ na diagonal principal. Seus valores são calculados pela DFT da primeira coluna de $\mathbf{C}_{\mathbf{Y}_q}(b)$ e assim $\underline{\mathbf{Y}}_q(b)$, ou seja:

$$\underline{\mathbf{Y}}_q(b) = \text{Diag}\{\mathbf{F}_R[0, \dots, 0, y_q(bN), \dots, y_q(bN + L - 1), 0, \dots, 0]^T\} \quad (7.5)$$

O operador $\text{Diag}\{\mathbf{x}\}$ denota uma matriz quadrada com os elementos do vetor “ \mathbf{x} ” na sua diagonal principal. A Figura 7.1 contém uma ilustração de $\mathbf{Y}_q(b)$, da matriz circulante $\mathbf{C}_{\mathbf{Y}_q}(b)$ e das matrizes de janelamento, que restringem a matriz circulante para a matriz original $\underline{\mathbf{Y}}_q(b)$.

Podemos então escrever $\mathbf{R}_{y_p y_q}$ como:

$$\mathbf{R}_{y_p y_q}(b) = \mathbf{W}_{N \times R}^{1_{N0}} \mathbf{F}_R^{-1} \underline{\mathbf{Y}}_q^H(b) \mathbf{F}_R \mathbf{W}_{R \times L+N}^{01_{L+N}} \mathbf{W}_{L+N \times R}^{01_{L+N}} \mathbf{F}_R^{-1} \underline{\mathbf{Y}}_q(b) \mathbf{F}_R \mathbf{W}_{R \times N}^{1_{N0}} \quad (7.6)$$

Pode-se observar no canto superior esquerdo da ilustração na Figura 7.1 que, a matriz de janelamento $\mathbf{W}_{L+N \times R}^{01_{L+N}}$ é uma extensão de $\mathbf{W}_{R \times R}^{01_R}$, apenas com linhas de zeros introduzidas no início da matriz $\tilde{\mathbf{Y}}_q$. Estes zeros anexados não têm nenhum efeito sobre o cálculo da matriz de correlação $\mathbf{R}_{y_p y_q}(b)$ e, assim, a multiplicação das

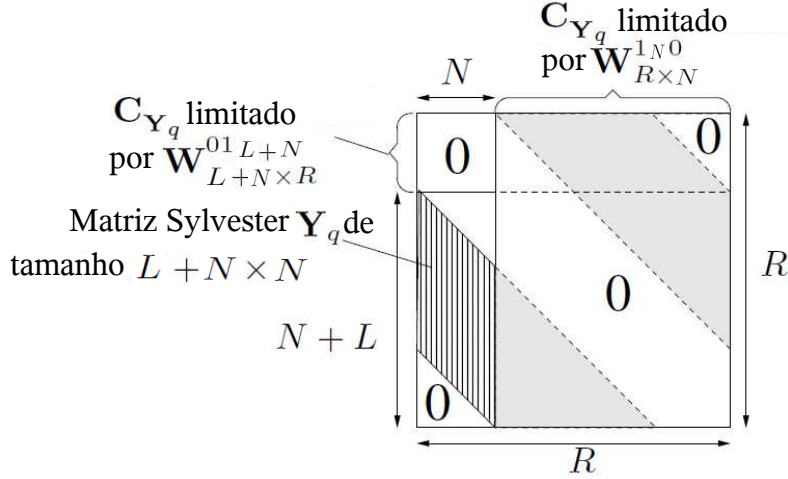


Figura 7.1: relação entre matriz circulante $\mathbf{C}_{\tilde{\mathbf{Y}}_q}(b)$ e a matriz Toeplitz $\tilde{\mathbf{Y}}_q(b)$

matrizes de janelamento é dada por:

$$\mathbf{W}_{R \times R}^{01 R} \mathbf{W}_{R \times R}^{01 R} = \mathbf{I}_{R \times R} \quad (7.7)$$

o que leva a:

$$\mathbf{R}_{y_p y_q}(b) = \mathbf{W}_{N \times R}^{1N 0} \mathbf{F}_R^{-1} \mathbf{Y}_q^H(b) \mathbf{Y}_q(b) \mathbf{F}_R \mathbf{W}_{R \times N}^{1N 0} \quad (7.8)$$

ou

$$\mathbf{R}_{y_p y_q}(b) = \mathbf{W}_{N \times R}^{1N 0} \mathbf{C}_{\mathbf{Y}_p \mathbf{Y}_q}(b) \mathbf{W}_{R \times N}^{1N 0} \quad (7.9)$$

A matriz de correlação acima é uma expressão equivalente no domínio da frequência para a apresentada no capítulo anterior. Assim, a normalização com base na inversão continua a corresponder à normalização exata com base na matriz inversa da matriz Toeplitz. Nas seções seguintes, mostra-se como esse inverso pode ser aproximado para obter uma implementação eficiente.

7.1.1 Representação DFT da Matriz de Filtro de Separação $\mathbf{W}(b)$

Primeiro, a matriz $\mathbf{W}_{pq}(b)$ será transformada no domínio DFT, conforme [34]. Assim, a matriz $2N \times D$ Toeplitz $\mathbf{W}_{pq}(b)$ tem de ser estendida para uma matriz circulante $\mathbf{C}_{\mathbf{W}_{pq}}(b)$ de dimensão $R \times R$, com $R \geq 2N$ (note que $1 \leq D \leq N$, sendo adotado neste trabalho $D = N$). A relação entre a matriz original $\mathbf{W}_{pq}(b)$ e a circulante $\mathbf{C}_{\mathbf{W}_{pq}}(b)$ é dada por:

$$\mathbf{W}_{pq}(b) = \mathbf{W}_{2N \times R}^{12N 0} \mathbf{C}_{\mathbf{W}_{pq}}(b) \mathbf{W}_{R \times D}^{1D 0} \quad (7.10)$$

A propriedade fundamental de uma matriz circulante, isto é, que ela pode ser diagonalizada pela matriz DFT, é agora explorada, originando:

$$\mathbf{C}_{\mathbf{W}_{pq}}(b) = \mathbf{F}_R^{-1} \underline{\mathbf{W}}_{pq}(b) \mathbf{F}_R \quad (7.11)$$

com

$$\underline{\mathbf{W}}_{pq}(b) = \text{Diag}\{\mathbf{F}_R[\mathbf{W}_{pq}^T(b), \mathbf{0}_{R-N \times 1}^T]^T\} \quad (7.12)$$

Assim, $\underline{\mathbf{W}}_{pq}(b)$ tem os valores de DFT da primeira coluna de $\mathbf{C}_{\mathbf{W}_{pq}}(b)$ na sua diagonal principal. A relação entre $\mathbf{W}_{pq}(b)$ no domínio de tempo e $\underline{\mathbf{W}}_{pq}(b)$ no domínio DFT é dada por:

$$\mathbf{W}_{pq}(b) = \mathbf{W}_{2N \times R}^{1_{2N} 0} \mathbf{F}_R^{-1} \underline{\mathbf{W}}_{pq}(b) \mathbf{F}_R \mathbf{W}_{R \times D}^{1_D 0} \quad (7.13)$$

7.1.2 Cálculo da matriz inversa $\mathbf{R}_{y_p y_q}^{-1}(b)$

Em [86] o teorema *Szegő* é formulado e comprovado para matrizes Toeplitz de ordem finita, a qual é definida como uma matriz Toeplitz $R \times R$ para a qual um N finito existe de modo a que todos os elementos da matriz com a linha ou coluna do índice maior do que N é igual a zero.

Foi mostrado em [86] que a matriz Toeplitz $R \times R$ de ordem N é assintoticamente equivalente à matriz circulante $R \times R$ gerada a partir de uma matriz Toeplitz $N \times N$ adequadamente complementada. Além disso, se as duas matrizes são de estrutura Hermitiana, temos:

- Os autovalores de ambas as matrizes situam-se entre um limite inferior e um limite superior;
- As médias aritméticas dos autovalores de ambas as matrizes são iguais se o tamanho R de ambas as matrizes se aproxima do infinito.

Pode ser visto que a matriz autocorrelação necessária para a normalização pode ser expressa como uma matriz $N \times N$ Toeplitz $\mathbf{R}_{y_p y_q}(b)$ ou uma matriz $R \times R$ circulante $\mathbf{C}_{\mathbf{Y}_p \mathbf{Y}_q}(b)$, gerada a partir da matriz Toeplitz estendendo-a apropriadamente e multiplicando a matriz resultante por algumas matrizes de janelamento.

De acordo com [86], ambas as matrizes são assintoticamente equivalentes. Como ambas as matrizes Toeplitz e circulante são hermitianas, é possível aplicar o teorema *Szegő*. Os autovalores da matriz $\mathbf{C}_{\mathbf{Y}_p \mathbf{Y}_q}(b)$ são dados em (7.7) pelos elementos na diagonal principal de $\underline{\mathbf{Y}}_q^H \underline{\mathbf{Y}}_q(b)$. O teorema *Szegő* afirma que os autovalores da matriz Toeplitz $R \times R$, que foi gerada anexando-se zeros à $\mathbf{R}_{y_p y_q}(b)$, podem ser assintoticamente aproximados por $\underline{\mathbf{Y}}_q^H \underline{\mathbf{Y}}_q(b)$ para $R \rightarrow \infty$.

O benefício desta aproximação torna-se evidente se observarmos o inverso de uma matriz circulante. O inverso de uma matriz circulante pode ser facilmente calculado

pela inversão de seus autovalores, de acordo com:

$$\mathbf{C}_{\mathbf{Y}_p \mathbf{Y}_q}^{-1}(b) = \mathbf{F}_R^{-1} \underline{\mathbf{Y}}_p^H(b) \underline{\mathbf{Y}}_q(b) \mathbf{F}_R \quad (7.14)$$

Utilizando o teorema *Szegő*, podemos aproximar o inverso da matriz $\mathbf{R}_{y_p y_q}(b)$ pelo inverso da matriz circulante para $R \rightarrow \infty$ como feito em [87], ou seja,

$$\mathbf{R}_{y_p y_q}^{-1}(b) = \mathbf{W}_{N \times R}^{1N0} \mathbf{F}_R^{-1} (\underline{\mathbf{Y}}_q^H(b) \underline{\mathbf{Y}}_q(b))^{-1} \mathbf{F}_R \mathbf{W}_{R \times N}^{1N0} \quad (7.15)$$

Isto também pode ser denotado como aproximação de banda estreita porque os autovalores $\underline{\mathbf{Y}}_q^H(b) \underline{\mathbf{Y}}_q(b)$ podem ser facilmente determinados pela DFT da primeira coluna da matriz circulante $\mathbf{C}_{\mathbf{Y}_p \mathbf{Y}_q}(b)$.

O inverso pode agora ser obtido de forma eficiente através de uma inversão escalar, uma vez que $\underline{\mathbf{Y}}_q^H(b) \underline{\mathbf{Y}}_q(b)$ representa uma matriz diagonal. Além disso, é importante notar que o inverso de uma matriz circulante é também circulante. Assim, após o janelamento os resultantes da matriz $\mathbf{R}_{y_p y_q}^{-1}(b)$ apresentam novamente uma estrutura Toeplitz.

Em resumo, essa função pode ser eficazmente implementada pela DFT da primeira coluna de $\mathbf{C}_{\mathbf{Y}_p \mathbf{Y}_q}(b)$ seguida por uma inversão escalar dos valores no domínio da frequência e pela DFT inversa. Após a operação de janelamento, estes valores são então replicados para gerar a estrutura Toeplitz de $\mathbf{R}_{y_p y_q}^{-1}(b)$.

7.1.3 Regra de atualização

Como apresentado em [34] utilizando-se o gradiente natural, temos que a equação de atualização dos coeficientes da matriz de separação no domínio da frequência é dada por:

$$\underline{\mathbf{W}}(b) = \underline{\mathbf{W}}(b-1) - \mu \Delta \underline{\mathbf{W}}(b) \quad (7.16)$$

onde

$$\Delta \underline{\mathbf{W}}(b) = \sum_{i=0}^{\infty} \beta(i, b) \mathbf{G}_{PR \times PR}^{1N0} \Delta \tilde{\underline{\mathbf{W}}}(i) * \mathbf{L}_I \quad (7.17)$$

sendo

$$\Delta \tilde{\underline{\mathbf{W}}}(i) \propto \underline{\mathbf{W}}(i) \text{boff}\{\underline{\mathbf{Y}}(i) \underline{\mathbf{Y}}^H(i)\} \text{bdiag}^{-1}\{\underline{\mathbf{Y}}^H(i) \underline{\mathbf{Y}}(i)\} \quad (7.18)$$

com a matriz de restrição dada por

$$\mathbf{G}_{PR \times PR}^{1N0} = \text{bdiag}\{\mathbf{F}_R \mathbf{W}_{R \times R}^{1N0} \mathbf{F}_R^{-1}, \dots, \mathbf{F}_R \mathbf{W}_{R \times R}^{1N0} \mathbf{F}_R^{-1}\} \quad (7.19)$$

e $\mathbf{L}_I = \text{bdiag}\{\mathbf{1}_{R \times 1}, \dots, \mathbf{1}_{R \times 1}\}$ uma matriz bloco-diagonal que consiste em colunas de vetores $\mathbf{1}_{R \times 1}$ contendo apenas “uns”. A multiplicação no lado direito pela matriz \mathbf{L}_I converte a matriz $R \times R$ $\Delta \tilde{\underline{\mathbf{W}}}$ no domínio DFT em vetores colunas $R \times 1$

contendo os R valores da DFT. A multiplicação no lado esquerdo pela matriz de restrições $\mathbf{G}_{PR \times PR}^{1N^0}$ transforma a matriz do domínio DFT para o domínio do tempo, restringindo todos os valores maiores do que o comprimento do filtro N a zero, e depois volta para o domínio DFT.

7.2 BSS em Subbandas no Domínio da Frequência

Nesta seção, algumas modificações foram feitas no algoritmo TRINICON no domínio da frequência para a sua execução em subbandas, procedendo-se de maneira semelhante à apresentada no capítulo anterior.

Apresentaremos assim um novo método para separação cega de misturas convolutivas em subbandas no domínio da frequência, que apresenta melhor desempenho que os algoritmos anteriormente abordados nesta dissertação para BSS, tendo um ganho maior de desempenho em ambientes com altos tempos de reverberação. Esses cenários serão os que focaremos neste capítulo, devido ao bom desempenho do algoritmo proposto (em relação ao no domínio do tempo) compensar seu gasto computacional superior.

Comparado ao algoritmo TRINICON no domínio DFT apresentado anteriormente, esse algoritmo a primeira vista apresenta uma complexidade computacional maior, devido a realizar M vezes o cálculo de $\Delta \underline{\mathbf{W}}_m$, sendo M o número de subbandas. Porém, ele apresenta um resultado bastante superior, principalmente em altos tempos de reverberação, com filtros de separação de tamanho consideravelmente menor do que o usado em banda completa, tendo assim uma complexidade computacional comparável, além de apresentar uma maior taxa de convergência e maior relação sinal-interferência em regime estacionário, quando comparado ao algoritmo correspondente em banda cheia.

As modificações necessárias no algoritmo BSS em blocos no domínio da frequência apresentado na seção anterior consistem basicamente das mesmas modificações realizadas no capítulo anterior no domínio do tempo, não sendo necessários os filtros de síntese, apenas os de análise, sendo a atualização de \mathbf{W} realizada em banda completa.

Cada sinal observado da mistura, $x_p(n)$ é decomposto pelos filtros de análise $\mathbf{F}_m(z)$, sendo os sinais resultantes processados pelos subfiltros de separação de banda completa, dados por $\mathbf{W}_{pq}(n)$, gerando sinais de saída $y_p^m(n)$ para a m -ésima subbanda, os quais serão passados para o domínio DFT, como mostrado nas seções anteriores, para o cálculo de atualização da nova matriz \mathbf{W} .

A atualização da matriz \mathbf{W} se dá em banda completa, utilizando um somatório das funções $\Delta \tilde{\underline{\mathbf{W}}}_m$ das M subbandas, como realizado no algoritmo apresentado no domínio do tempo.

Por fim para obter a equação de atualização da matriz de filtros de separação \mathbf{W} , usamos a mesma recursão apresentada na seção anterior, com a adição dos cálculos de $\Delta\tilde{\mathbf{W}}_m$ em cada subbanda, como mostrado a seguir:

$$\Delta\mathbf{W}(b) = \sum_{i=0}^{\infty} \beta(i, b) \mathbf{G}_{PR \times PR}^{1N0} \Delta\tilde{\mathbf{W}}(i) * \mathbf{L}_I \quad (7.20)$$

sendo

$$\Delta\tilde{\mathbf{W}}(i) = \sum_{m=0}^M \mathbf{W}(i) \text{boff}\{\mathbf{Y}^m(i)\mathbf{Y}^{mH}(i)\} \text{bdiag}^{-1}\{\mathbf{Y}^{mH}(i)\mathbf{Y}^m(i)\} \quad (7.21)$$

7.3 Fatores de Aprendizagem

Assim como no domínio do tempo, a escolha do tamanho de passo é muito importante para o desempenho do algoritmo. Para tornar a adaptação mais robusta, um controle no tamanho do fator de aprendizagem do algoritmo é desejável. Usaremos uma estratégia simples, mas eficaz, para atualizar o tamanho do passo, baseado no método apresentado em [34].

O procedimento é o de aumentar o tamanho do passo, se o valor de $\Delta\mathbf{W}(i)$ é diminuído em comparação com $\Delta\mathbf{W}(i-1)$ (indicando convergência) e de diminuir rapidamente, se o valor de $\Delta\mathbf{W}(i)$ excede o anterior por mais do que um fator pré-definido (indicando divergência). Após começar com um fator de aprendizagem $\mu(0)$ pequeno, suas variações são descritas por

$$\mu(i+1) = \begin{cases} a\mu(i), & \text{se } \Delta\mathbf{W}(i) < \Delta\mathbf{W}(i-1) \\ b\mu(i), & \text{se } \Delta\mathbf{W}(i) \geq \Delta\mathbf{W}(i-1) \\ \mu(i-1), & \text{nos demais casos} \end{cases} \quad (7.22)$$

sendo, para a nossa aplicação, utilizados os valores $a = 1.1$, $b = 0.5$, $c = 1.3$. Além disso, para evitar a instabilidade, na prática, o tamanho do passo adaptativo deve ser restrito a um intervalo finito $[\mu_{min}, \mu_{max}]$, o qual durante o decorrer do algoritmo pode ser necessário modificar.

7.4 Simulações

Os algoritmos no domínio da frequência descritos neste capítulo foram avaliados tal como no capítulo anterior (algoritmos no domínio do tempo), empregaram-se duas misturas convolutivas de dois sinais de voz, supondo $Q = P = 2$, sendo a mistura composta de um sinal de voz masculina e outro de voz feminina, disponibilizados em [85], com duração de 10 s e frequência de amostragem 16 kHz.

As simulações da propagação acústica de salas foram desenvolvidas usando o modelo "Image-Source" descrito em [16], sendo utilizados novamente os parâmetros da Tabela 6.1.

As simulações aqui apresentadas foram realizadas para os tempos de reverberação maiores, onde o ganho de desempenho do algoritmo justificou o custo computacional superior ao do algoritmo no domínio do tempo. Para tempos de reverberação pequenos, os ganhos de taxa de convergência e maior razão sinal-interferência em regime estacionário não foram tão significativos, considerando-se o aumento computacional do mesmo.

Os tempos de reverberação avaliados (T_{60}) foram: 0.3 s e 0.5 s. A medida de avaliação utilizada foi a SIR. Para a decomposição em subbandas, empregou-se o mesmo banco de filtros modulado por cosseno [65] com $M = 4$ subbandas e filtro protótipo de comprimento 32 utilizado anteriormente. O tamanho dos blocos foi fixado em $2D$, sendo o valor do atraso D utilizado igual ao comprimento do filtro de separação N .

Os fatores de aprendizagem utilizados nos algoritmos no domínio da frequência, diferentemente dos empregados no domínio do tempo, foram os mostrados na seção anterior, com valores iniciais variáveis de $\mu = 0.001$ à $\mu = 0.004$ dependendo do desempenho do algoritmo no caso considerado. Os limites $[\mu_{min}, \mu_{max}]$ variaram para cada simulação, com menor limite inferior $\mu_{min} = 0.00001$ e maior limite superior $\mu_{max} = 0.005$.

A Figura 7.2 apresenta as evoluções da SIR ao longo das iterações obtidas com os algoritmos em banda cheia e em subbandas para $T_{60} = 0.3$ s, com filtros de separação de comprimentos $N = 1024$ e $N = 512$ para o algoritmo em banda cheia e $N = 512$ para o algoritmo em subbandas. Pode-se verificar nessa figura que o algoritmo BSS no domínio da frequência em subbandas proposto neste trabalho apresenta desempenho superior, obtendo SIR de 15.6 dB, enquanto que o algoritmo em banda cheia alcançou SIR de 9.89 dB com o mesmo comprimento de filtro $N = 512$ e 12.32 dB de com $N = 1024$, novamente bem inferiores à SIR obtida com o algoritmo em subbandas.

Na Figura 7.3 são mostrados os resultados das simulações para $T_{60} = 0.5$ s com filtros de separação de comprimentos $N = 1024$ e $N = 512$ para o algoritmo de banda cheia e $N = 512$ para o algoritmo em subbandas. Novamente pode-se verificar a superioridade do algoritmo em subbandas, apresentando convergência mais rápida e razão sinal-interferência maior quando comparado ao de banda cheia, mesmo com comprimento de filtro menor. Para o tempo de reverberação de $T_{60} = 0.5$ s o algoritmo em banda cheia alcançou SIR de 11.03 dB para $N = 1024$ e de 8.38 dB para $N = 512$ enquanto que o em subbandas alcançou SIR de 12.34 dB, demonstrando o ganho no processo de separação em subbandas tanto em relação à SIR quanto em

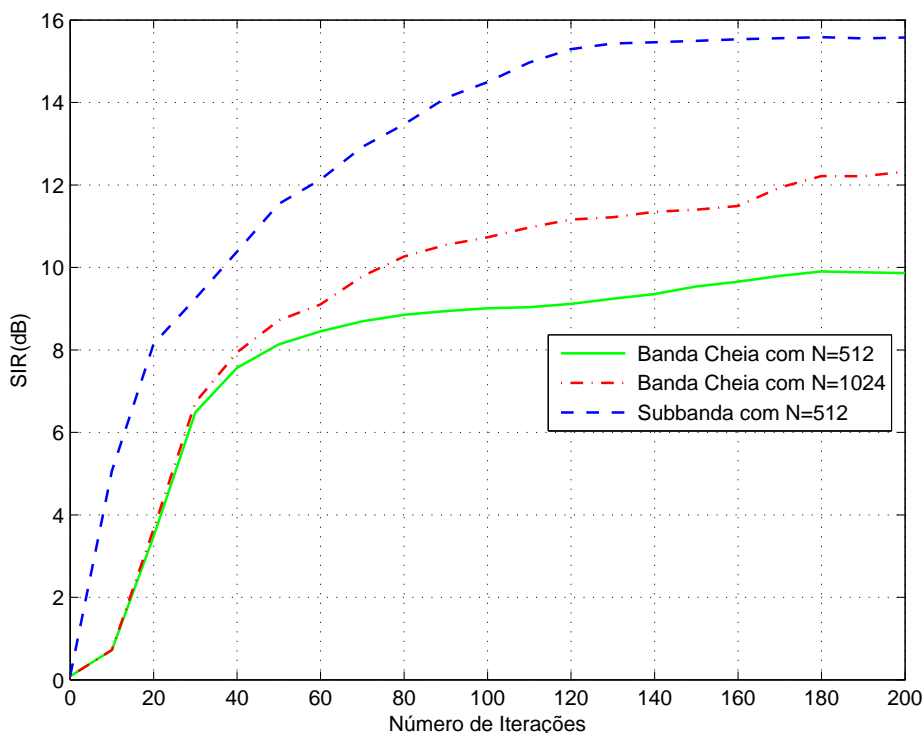


Figura 7.2: Evolução da SIR (em dB) para $T_{60} = 0.1$ s para BSS no domínio da frequência.

relação ao custo computacional, uma vez que o tamanho dos filtros de separação é de apenas $N = 512$.

É possível verificar que os valores encontrados para esses algoritmos, ambos em banda cheia e em subbandas, são superiores aos encontrados no capítulo anterior no domínio do tempo. Na próxima seção serão apresentadas comparações das simulações com os algoritmos nos domínios do tempo e da frequência.

7.5 Comparações Finais

Apresentamos nesta seção comparações dos desempenhos dos métodos de separação cega de fontes apresentados neste trabalho: TRINICON em banda cheia e em subbandas no domínio do tempo e TRINICON em banda cheia e em subbandas no domínio da frequência.

Na Figura 7.4 são mostrados os resultados das simulações para $T_{60} = 0.3$ s dos quatro algoritmos e os comprimentos dos seus respectivos filtros de separação. Na Tabela 7.1 são mostrados os valores da SIR obtidos após a convergência dos algoritmos para $T_{60} = 0.3$ s. Como esperado, o algoritmo no domínio da frequência em subbandas apresentou o melhor resultado, seguido do algoritmo em banda completa

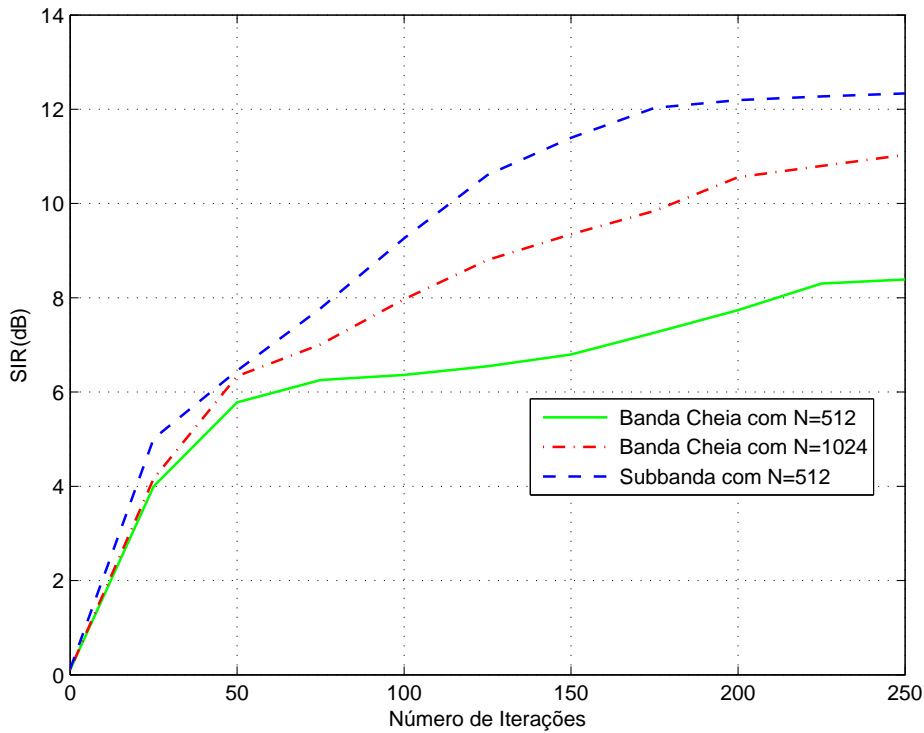


Figura 7.3: Evolução da SIR (em dB) para $T_{60} = 0.5$ s para BSS no domínio da frequência.

no domínio da frequência. Na Figura 7.5 são mostrados os resultados das simulações

Tabela 7.1: Valores da SIR (em dB) obtidos após a convergência dos algoritmos para $T_{60} = 0.3$ s

Domínio	Banda Cheia	Subbanda
Tempo	9.92	10.51
Frequência	12.32	15.58

para $T_{60} = 0.5$ s dos quatro algoritmos e os comprimentos dos seus respectivos filtros de separação.

Na Tabela 7.2 são mostrados os valores da SIR obtidos após a convergência dos algoritmos para $T_{60} = 0.5$ s. Novamente, o algoritmo no domínio da frequência em subbandas apresentou o melhor resultado, seguido do algoritmo em banda completa no domínio da frequência.

Tabela 7.2: Valores da SIR (em dB) obtidos após a convergência dos algoritmos para $T_{60} = 0.5$ s

Domínio	Banda Cheia	Subbanda
Tempo	5.68	6.15
Frequência	11.03	12.34

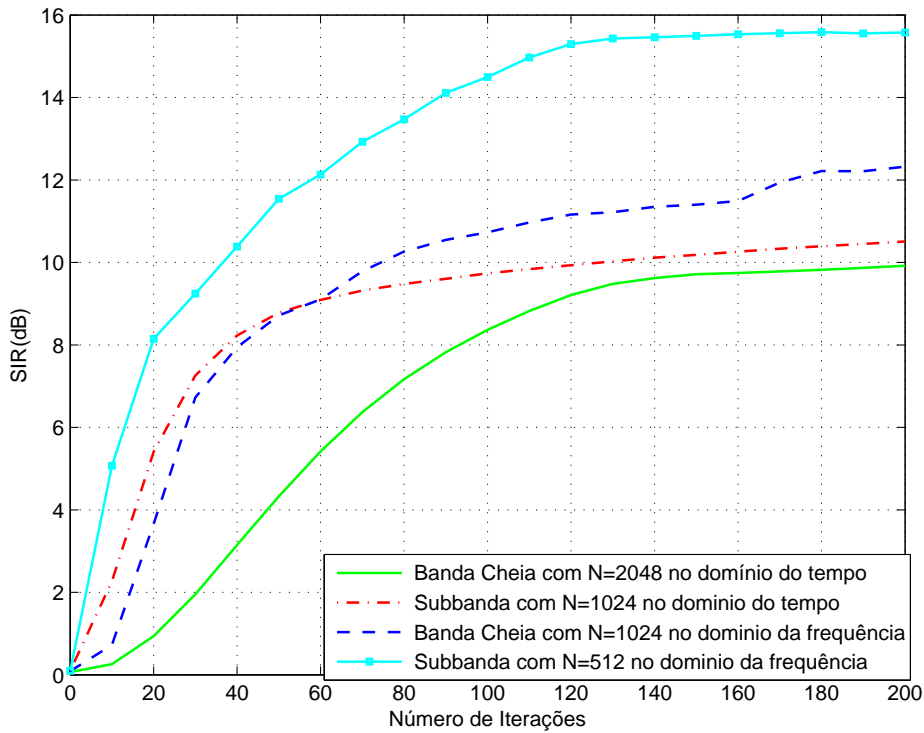


Figura 7.4: Evolução da SIR (em dB) para $T_{60} = 0.3$ s para BSS no domínio do tempo e da frequência.

Analisando os resultados obtidos nos domínios do tempo e da frequência, verificou-se a superioridade dos métodos em subbandas propostos, tanto no domínio do tempo como no da frequência, tendo eles uma complexidade computacional menor devido ao reduzido tamanho de filtro utilizado no sistema de separação, uma maior taxa de convergência e maior razão sinal-interferência em regime estacionário quando comparados aos algoritmos correspondentes em banda cheia. Podemos notar também que os algoritmos em subbandas apresentaram uma convergência mais rápida comparado aos algoritmos em banda cheia, independente do domínio.

Podemos observar ainda que o algoritmo em subbandas no domínio da frequência apresentou resultado aceitável para tempos de reverberação altos, tendo esse método uma complexidade computacional menor devido ao reduzido tamanho de filtro utilizado no sistema de separação. Além disso, esse algoritmo apresentou maiores taxa de convergência e razão sinal-interferência em regime estacionário quando comparado a todos os outros algoritmos, sendo, portanto, a melhor opção em ambientes com alta reverberação.

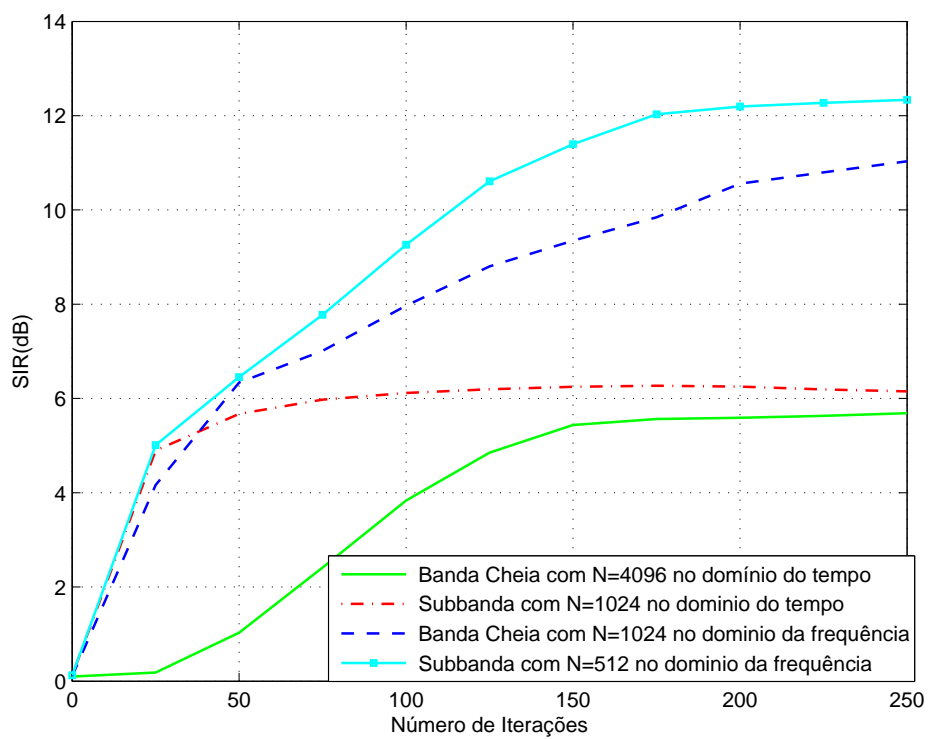


Figura 7.5: Evolução da SIR (em dB) para $T_{60} = 0.5$ s para BSS no domínio do tempo e da frequência.

Capítulo 8

Conclusões

Neste capítulo serão apresentadas as conclusões finais deste trabalho e algumas sugestões de trabalhos futuros.

As técnicas de BSS para sinais acústicos têm recebido uma grande atenção da comunidade de pesquisadores da área de processamento de sinais nos últimos anos. Apresentamos nesta dissertação um novo método de separação cega de fontes para misturas convolutivas, derivado a partir de modificações incorporadas na estrutura do algoritmo TRINICON para permitir sua utilização em subbandas de frequência. Posteriormente aplicamos essas mesmas modificações no algoritmo TRINICON no domínio da frequência, resultando em um novo algoritmo altamente eficiente para ambientes com alta reverberação, tanto do ponto de vista da razão sinal-interferência em regime estacionário quanto do custo computacional.

8.1 Principais contribuições

Nos Capítulos 2 e 3 foram apresentados conceitos fundamentais de filtragem adaptativa e separação cega de fontes. No Capítulo 4 foram identificadas as limitações dos algoritmos adaptativos supervisionados NLMS e NSAF para o problema de separação supervisionada de fontes, onde pode-se verificar que o modelo do sistema de separação com filtros FIR é um grande limitante para ambientes com tempos de reverberação altos.

No Capítulo 5 os resultados numéricos de simulações indicaram que a função empírica proposta para ajuste automático dos parâmetros do algoritmo l_0 -LMS obteve excelente desempenho para sistemas com fatores de esparsidade $\xi_{\mathbf{w}}$ maiores do que 0.75. Para sistemas com $\xi_{\mathbf{w}} < 0.75$, verificou-se a necessidade da alteração do valor de κ encontrado conforme o valor de $\xi_{\mathbf{w}}$, obtendo-se uma nova função que relaciona os mesmos. Ao final, propôs-se um novo modo de ajuste, capaz de contemplar variações da SNR e do fator de esparsidade ξ_w .

No Capítulo 6 foram apresentadas as simulações do algoritmo TRINICON com

as modificações propostas para gerar o método em subbandas proposto. Houve uma boa melhora na razão sinal-interferência e um aumento na velocidade de convergência, porém os resultados encontrados para ambientes com tempos de reverberação altos não foram suficientes para obter-se boa separação.

Por fim, no Capítulo 7 foi apresentada uma formulação equivalente do algoritmo TRINICON no domínio DFT, onde se realizou modificações semelhantes às do Capítulo 6 para a utilização desse algoritmo em subbandas. Com esse novo algoritmo, obtiveram-se bons resultados em ambientes com altos tempos de reverberação. O algoritmo proposto apresenta complexidade computacional menor do que o algoritmo em banda cheia, devido ao reduzido tamanho dos filtros utilizados no sistema de separação, além de uma maior taxa de convergência e maior razão sinal-interferência em regime estacionário. Quando comparado a todos os outros algoritmos simulados, mostrou-se a melhor opção em ambientes com alta reverberação.

8.2 Trabalhos Futuros

As seguintes sugestões de trabalhos futuros foram identificadas, podendo trazer melhorias nos desempenhos das técnicas desenvolvidas durante esta dissertação:

- Encontrar um ajuste automático melhor para o parâmetro de aprendizagem μ utilizado nas técnicas em subbandas, uma vez que esses métodos se mostraram bastante sensíveis à escolha dos limites inferiores e superiores utilizados para esse parâmetro em diferentes situações;
- Investigar o uso de decimação nas técnicas em subbandas, uma vez que isso pode gerar um grande ganho no desempenho computacional;
- Investigar a possibilidade de trabalhar com parâmetros diferentes em cada subbanda, explorando assim as características individuais de cada uma delas;
- Explorar a esparsidade das respostas impulsivas, usada na função custo apresentada no Capítulo 5, nos algoritmos de separação cega de fontes dos Capítulos 6 e 7, buscando assim um ganho no desempenho dos algoritmos BSS;
- Avaliar os desempenhos das técnicas propostas em outros cenários.

Referências Bibliográficas

- [1] BOLL, S. “Suppression of acoustic noise in speech using spectral subtraction”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 27, n. 2, pp. 113–120, Apr. 1979.
- [2] BENESTY, J., MAKINO, S., CHEN, J. *Speech Enhancement*. Signals and Communication Technology. Springer, 2005.
- [3] SCHWARZ, A., REINDL, K., KELLERMANN, W. “A two-channel reverberation suppression scheme based on blind signal separation and wiener filtering”. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 113–116, Mar. 2012.
- [4] BUCHNER, H., AICHNER, R., KELLERMANN, W. “Blind Source Separation for Convolutional Mixtures Exploiting Nongaussianity, Nonwhiteness, and Nonstationarity”, *International Workshop on Acoustic Echo and Noise Control (IWAENC2003)*, pp. 275–278, 2003.
- [5] CORADINE, L. C. *Filtagem Adaptativa em Cascata: Proposta de Estrutura e Algoritmo, Análise e Aplicações*. Tese de D.Sc., Programa de Pós-Graduação em Engenharia Elétrica- Unicamp, 1993.
- [6] PUTTINI, S. *Emprego de Antenas Adaptativas para Estimção de Dados em Ambiente CDMA*. Tese de D.Sc., Departamento de engenharia Elétrica, Universidade de Brasília, Brasilia, Brasil, 2006.
- [7] JR., C. R. *Implementação de Filtros Adaptativos em FPGA*. Tese de D.Sc., Programa de Pós-Graduação em Engenharia Elétrica- PUC, Pelotas, 2006.
- [8] HAYKIN, S. O. *Adaptive Filter Theory*. Prentice-Hall, 2013.
- [9] ZIPF, J. G. F. *Classificação, Análise Estatística e Novas Estratégias de Algoritmos LMS de Passo Variável*. Tese de D.Sc., Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Elétrica, 2011.

- [10] ZEIDLER, J. “Performance Analysis of LMS Adaptive Prediction Filters”, *IEEE J. Sel. Topics Signal Process.*, v. 78, pp. 1781–1806, 1990.
- [11] JURNIOR, F. *Arranjos de Antenas Adaptativas em Sistemas Ofdma: Um Comparativo entre as Técnicas de Filtragem Pré e Pós-FFT*. Tese de D.Sc., Faculdade de Engenharia Elétrica, Universidade Federal do Pará, Belém, Brasil, 2009.
- [12] DINIZ, P. S. R. *Adaptive Filtering: Algorithms and Practical Implementation*. New York, NY, USA, Springer US, 2013.
- [13] KUHN, E. V. *Modelagem Estocástica do Algoritmo NLMS : Revisão e Aprimoramentos*. Tese de D.Sc., Programa de Pós-Graduação em Engenharia Elétrica- UFSC, 2012.
- [14] LEE, K.-A., GAN, W.-S., KUO, S. M. *Subband Adaptive Filtering: Theory and Implementation*. Wiley Publishing, 2009.
- [15] LEE, K. A., GAN, W. S. “Improving Convergence of the NLMS Algorithm Using Constrained Subband Updates”, *IEEE Signal Processing Letters*, v. 11, n. 9, pp. 736–739, Sep. 2004.
- [16] LEHMANN, E. A., JOHANSSON, A. M., NORDHOLM, S. “Reverberation-Time Prediction Method for Room Impulse Responses Simulated with the Image-Source Model”. In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 159–162, Oct. 2007.
- [17] SHYNK, J. J. “Frequency-domain and Multirate Adaptive Filtering”, *IEEE Signal Processing Magazine*, v. 9, n. 1, pp. 14–37, Jan. 1992.
- [18] HAYKIN, S., CHEN, Z. “The Cocktail Party Problem”, *Neural Comput.*, v. 17, n. 9, pp. 1875–1902, Sep. 2005.
- [19] HYVARINEN, A., KARHUNEN, J., OJA, E. *Independent Component Analysis*. New York, Chichester, Weinheim, John Wiley, 2001.
- [20] ZIBULEVSKY, M., PEARLMUTTER, B. A. “Blind Source Separation by Sparse Decomposition in a Signal Dictionary”, *Neural Comput.*, v. 13, n. 4, pp. 863–882, Apr. 2001.
- [21] ARAKI, S., SAWADA, H., MUKAI, R., et al. “Underdetermined Blind Sparse Source Separation for Arbitrarily Arranged Multiple Sensors”, *Signal Process.*, v. 87, n. 8, pp. 1833–1847, Aug. 2007.

- [22] YILMAZ, O., RICKARD, S. “Blind Separation of Speech Mixtures via Time-frequency Masking”, *Trans. Sig. Proc.*, v. 52, n. 7, pp. 1830–1847, jul 2004.
- [23] CAO, X.-R., LIU, R.-W. “General Approach to Blind Source Separation”, *IEEE Transactions on Signal Processing*, v. 44, n. 3, pp. 562–571, Mar. 1996.
- [24] BUCHNER, H., AICHNER, R., KELLERMANN, W. “A Generalization of Blind Source Separation Algorithms for Convolutional Mixtures Based on Second-order Statistics”, *IEEE Transactions on Speech and Audio Processing*, v. 13, n. 1, pp. 120–134, Jan. 2005.
- [25] KIM, T., ATTIAS, H. T., LEE, S. Y., et al. “Blind Source Separation Exploiting Higher-Order Frequency Dependencies”, *IEEE Transactions on Audio, Speech, and Language Processing*, v. 15, n. 1, pp. 70–79, Jan. 2007.
- [26] KOFIDIS, E. “Blind Source Separation: Fundamentals and Recent Advances (A Tutorial Overview Presented at SBrT).” *CoRR*, v. abs/1603.03089, 2016.
- [27] CARDOSO, J. F. “Blind Signal Separation: Statistical Principles”, *Proceedings of the IEEE*, v. 86, n. 10, pp. 2009–2025, Oct. 1998.
- [28] MANSOUR, A., BARROS, A. K., OHNISHI, N. “Blind Separation of Sources: Methods, Assumptions and Applications”, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, v. 83, n. 8, pp. 1498–1512, 2000.
- [29] LAPORTE, L. V. M. *Algoritmos de Separação Cega de Sinais de Áudio no Domínio da Frequência em Ambientes Reverberantes: Estudo e Comparações*. Ph.D. dissertation, UFRJ/COPPE/Programa de Engenharia Elétrica, Rio de Janeiro, RJ, BR, 2010.
- [30] SAWADA, S. M. T. L. H. *Blind Speech Separation*. Signals and Communication Technology. Dordrecht, The Netherlands, Springer Netherlands, 2007.
- [31] CICHOCKI, A., AMARI, S.-I. *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. New York, NY, USA, John Wiley & Sons, Inc., 2002.
- [32] DYRHOLM, M., MAKEIG, S., HANSEN, L. K. “Model Structure Selection in Convolutional Mixtures”. In: *Independent Component Analysis and Blind*

Signal Separation: 6th International Conference, ICA, pp. 74–81, Berlin, Heidelberg, Springer Berlin Heidelberg, 2006.

- [33] ANEMÜLLER, J., SEJNOWSKI, T. J., MAKEIG, S. “Complex Independent Component Analysis of Frequency-domain Electroencephalographic Data”, *Neural Netw.*, v. 16, n. 9, pp. 1311–1323, Nov. 2003.
- [34] AICHNER, R. *Acoustic Blind Source Separation in Reverberant and Noisy Environments*. Tese de D.Sc., Technischen Fakultät der Friedrich-Alexander-Universität, Erlangen, Nurnberg, 2007.
- [35] DELLER, JR., J. R., PROAKIS, J. G., HANSEN, J. H. *Discrete Time Processing of Speech Signals*. 1st ed. Upper Saddle River, NJ, USA, Prentice Hall PTR, 1993.
- [36] RABINER, L. R., SCHAFER, R. W. *Digital Processing of Speech Signals*. Prentice-Hall signal processing series. Englewood Cliffs, N.J. Prentice-Hall, 1978.
- [37] MATSUOKA, K., KAWAMOTO, M. “A neural Net for Blind Separation of Nonstationary Signal Sources”. In: *Neural Networks, IEEE World Congress on Computational Intelligence*, v. 1, pp. 221–232 vol.1, Jun. 1994.
- [38] WEINSTEIN, E., FEDER, M., OPPENHEIM, A. V. “Multi-Channel Signal Separation by Decorrelation”, *IEEE Transactions on Speech and Audio Processing*, v. 1, n. 4, pp. 405–413, Oct. 1993.
- [39] PARRA, L., SPENCE, C., VRIES, B. D. “Convolutional Blind Source Separation Based on Multiple Decorrelation”. In: *Neural Networks for Signal Processing VIII. Proceedings of the IEEE Signal Processing Society Workshop*, pp. 23–32, Aug. 1998.
- [40] PARRA, L., SPENCE, C. “Convolutional Blind Separation of Non-Stationary Sources”, *IEEE Transactions on Speech and Audio Processing*, v. 8, pp. 320–327, 2000.
- [41] A., E. P. A., P., B. L. W. “Spectral-Based Analysis and Synthesis of Audio Signals”. In: Pérez-Meana, H. (Ed.), *Advances in Audio and Speech Signal Processing: Technologies and Applications*, 1 ed., pp. 56–92, Hershey, Idea Group, 2007.
- [42] CARDOSO, J. F. “Source Separation Using Higher Order Moments”. In: *Acoustics, Speech, and Signal Processing, International Conference*, pp. 2109–2112 vol.4, May 1989.

- [43] COMON, P., JUTTEN, C., HERAULT, J. “Blind Separation of Sources, Part II: Problems Statement”, *Signal Processing*, v. 24, n. 1, pp. 11 – 20, 1991.
- [44] JUTTEN, C., HERAULT, J. “Blind Separation of Sources, Part I: An Adaptive Algorithm Based on Neuromimetic Architecture”, *Signal Processing*, v. 24, n. 1, pp. 1 – 10, 1991.
- [45] BELL, A. J., SEJNOWSKI, T. J. “An Information-Maximization Approach to Blind Separation and Blind Deconvolution”, *Neural Computation*, v. 7, n. 6, pp. 1129–1159, Nov. 1995.
- [46] SMARAGDIS, P. “Blind separation of convolved mixtures in the frequency domain”, *Neurocomputing*, v. 22, n. 1–3, pp. 21 – 34, 1998.
- [47] TONG, L., LIU, R., SOON, V. C., et al. “Indeterminacy and Identifiability of Blind Identification”, *IEEE Transactions on Circuits and Systems*, v. 38, n. 5, pp. 499–509, May 1991.
- [48] MOLGEDEY, L., SCHUSTER, H. G. “Separation of a Mixture of Independent Signals Using Time Delayed Correlations”, *Physical Review Letters*, v. 72, pp. 3634–3637, 1994.
- [49] BELOUHRANI, A., ABED-MERAIM, K., CARDOSO, J. F., et al. “A Blind Source Separation Technique Using Second-order Statistics”, *IEEE Transactions on Signal Processing*, v. 45, n. 2, pp. 434–444, Feb. 1997.
- [50] GERVEN, S. V., COMPERNOLLE, D. V. “Signal Separation by Symmetric Adaptive Decorrelation: Stability, Convergence, and Uniqueness”, *IEEE Transactions on Signal Processing*, v. 43, n. 7, pp. 1602–1612, Jul 1995.
- [51] KRONGOLD, B. S., JONES, D. L. “Blind Source Separation of Nonstationary Convolutively Mixed Signals”. In: *Statistical Signal and Array Processing. Proceedings of the Tenth IEEE Workshop*, pp. 53–57, 2000.
- [52] AICHNER, R., BUCHNER, H., WEHR, S., et al. “Robustness of Acoustic Multiple-Source Localization in Adverse Environments”. In: *Proc. ITG Fachtagung Sprachkommunikation*, Kiel, Germany, Apr. 2006.
- [53] HYVARINEN, A. “Gaussian Moments for Noisy Independent Component Analysis”, *IEEE Signal Processing Letters*, v. 6, n. 6, pp. 145–147, June 1999.
- [54] AMARI, S., CICHOCKI, A., YANG, H. H. “A New Learning Algorithm for Blind Signal Separation”. In: *Advances in Neural Information Processing Systems*, pp. 757–763. MIT Press, 1996.

- [55] VARY, P., HEUTE, U., HESS, W. *Digitale Sprachsignalverarbeitung: mit 30 Tabellen*. Teubner, 1998.
- [56] QUACKENBUSH, S. R., BARNWELL, T. P., CLEMENTS, M. A. *Objective Measures of Speech Quality*. Englewood Cliffs, NJ, Prentice-Hall, 1988.
- [57] SAYED, A. H. *Fundamentals of Adaptive Filtering*. Hoboken, NJ, J. Wiley & Sons, New York, 2003.
- [58] SAYED, A. H. *Adaptive Filters*. Wiley-IEEE Press, 2008.
- [59] FARHANG-BOROJENY, B. *Adaptive Filters: Theory and Applications*. 1 ed. New York, NY, USA, John Wiley & Sons, Inc., 1998.
- [60] ELLIS, D. *Prediction-Driven Computational Auditory Scene Analysis*. Tese de D.Sc., MIT, USA, 1998.
- [61] LEE, K. A., GAN, W. S. “Improving Convergence of the NLMS Algorithm Using Constrained Subband Updates”, *IEEE Signal Processing Letters*, v. 11, n. 9, pp. 736–739, 2004.
- [62] SANSON, J. B., VASQUES, B. L. R. P., PETRAGLIA, M. R. “Filtragem Adaptativa em Subbandas para Separação Supervisionada de Fontes”. In: *XXXIV Simpósio Brasileiro de Telecomunicações*, Set. 2016.
- [63] VASQUES, B. L. R. P. *Limitações de Algoritmos de Separação Cega de Sinais de Voz em Ambientes Reverberantes*. Ph.D. dissertation, UFRJ/COPPE/Programa de Engenharia Elétrica, Rio de Janeiro, RJ, BR, 2015.
- [64] SMARAGDIS, P., RAJ, B., SHASHANKA, M. “Supervised and Semi-Supervised Separation of Sounds from Single-Channel Mixtures”. In: *International Conference on Independent Component Analysis and Signal Separation*, pp. 414–421. Springer Berlin Heidelberg, 2007.
- [65] MITRA, S. K. K. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill Higher Education, 2000.
- [66] SANSON, J. B., PETRAGLIA, M. R., HADDAD, D. B. “Parametrização Automática do Algoritmo l_0 -LMS com Relação à Razão Sinal-Ruído”. In: *XXXIII Simpósio Brasileiro de Telecomunicações*, Set. 2015.
- [67] GUI, G., MEHBODNIYA, A., ADACHI, F. “Sparse LMS/F Algorithms with Application to Adaptive System Identification”, *Wireless Communications and Mobile Computing*, v. 15, n. 12, pp. 1649–1658, 2015.

- [68] GU, Y., JIN, J., MEI, S. “ l_0 Norm Constraint LMS Algorithm for Sparse System Identification”, *IEEE Signal Processing Letters*, pp. 774–777, 2009.
- [69] JIN, J., GU, Y., MEI, S. “A Stochastic Gradient Approach on Compressive Sensing Signal Reconstruction Based on Adaptive Filtering Framework”, *CoRR*, v. abs/1303.2257, 2013.
- [70] HADDAD, D. B., PETRAGLIA, M. R., HADDAD, D., et al. “Transient and Steady-State MSE Analysis of The IMPNLMS Algorithm”, *Digital Signal Processing*, v. 33, pp. 50–59, 2014.
- [71] SU, G., JIN, J., GU, Y. “Performance analysis of L_0 -LMS with Gaussian Input Signal”. In: *IEEE 10th International Conference on Signal Processing Proceedings*, pp. 235–238, Oct. 2010.
- [72] MARTIN, R. “Noise Power Spectral Density Estimation Based on Optimal Smoothing and Minimum Statistics”, *IEEE Transactions on Speech and Audio Processing*, v. 9, n. 5, pp. 504–512, Jul. 2001.
- [73] COHEN, I. “Noise Spectrum Estimation in Adverse Environments: Improved Minima Controlled Recursive Averaging”, *IEEE Transactions on Speech and Audio Processing*, v. 11, n. 5, pp. 466–475, Sep. 2003.
- [74] HENDRIKS, R. C., JENSEN, J., HEUSDENS, R. “Noise Tracking Using DFT Domain Subspace Decompositions”, *IEEE Transactions on Audio, Speech, and Language Processing*, v. 16, n. 3, pp. 541–553, Mar. 2008.
- [75] BENESTY, J., REY, H., VEGA, L. R., et al. “A Nonparametric VSS NLMS Algorithm”, *IEEE Signal Processing Letters*, v. 13, n. 10, pp. 581–584, Oct. 2006.
- [76] CHO, H., BAN, S. J., KIM, S. W. “Estimation of the Power of the System Noise for Adaptive Filter Algorithms”. In: *ICCAS-SICE, 2009*, pp. 2514–2518, Aug. 2009.
- [77] INTERNATIONAL TELECOMMUNICATION UNION. “ITU-T Recommendation G.168”. 2004.
- [78] BUCHNER, H., AICHNER, R., KELLERMANN, W. “TRINICON: a Versatile Framework for Multichannel Blind Signal Processing”. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, v. 3, pp. iii–889. IEEE, 2004.
- [79] HYVÄRINEN, A., KARHUNEN, J., E. OJA. *Independent Component Analysis*. John-Wiley & Sons, 2001.

- [80] COVER, T., THOMAS, J. *Elements of Information Theory*. New York, Wiley & Sons, 1991.
- [81] S. AMARI, A. C., YANG, H. H. “A New Learning Algorithm for Blind Signal Separation”, *Advances in Neural Information Processing Systems*, v. 8, pp. 757–763, 1996.
- [82] AICHNER, R., B. H. A. S. “On-Line Time-Domain Blind Source Separation of Nonstationary Convolved Signals”. In: *European Signal Processing Conference*, pp. 987–992, 2003.
- [83] BATALHEIRO, P. B. *Novos Algoritmos em Subbandas para Filtragem Adaptativa e Separação Cega de Fontes*. Tese de D.Sc., UFRJ/COPPE/Programa de Engenharia Elétrica, Rio de Janeiro, RJ, BR, 2009.
- [84] HADDAD, D. B. *Estruturas em Subbandas para Filtragem Adaptativa e Separação Cega e Semi-Cega de sinais de voz*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2013.
- [85] VINCENT, E., SAWADA, H. “Código-fonte MATLAB para simulação de sala reverberante”. <http://www.irisa.fr/metiss/SASSECO7>, 2015.
- [86] GRAY, R. “On the asymptotic eigenvalue distribution of Toeplitz matrices”, *IEEE Transactions on Information Theory*, v. 18, n. 6, pp. 725–730, Nov. 1972.
- [87] AICHNER, R., BUCHNER, H., KELLERMANN, W. “A Novel Normalization and Regularization Scheme for Broadband Convolutional Blind Source Separation”. In: *Independent Component Analysis and Blind Signal Separation: 6th International Conference, ICA*, pp. 527–535, Berlin, Heidelberg, Springer Berlin Heidelberg, 2006.