



SEPARAÇÃO CEGA DE FONTES EM MISTURAS INSTANTÂNEAS E ANECOICAS EMPREGANDO ESPARSIDADE E CLUSTERIZAÇÃO

Jorge Costa Pires Filho

Tese de Doutorado 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 Doutor em Engenharia Elétrica.

Orientadora: Mariane Rembold Petraglia

Rio de Janeiro

Março de 2015

SEPARAÇÃO CEGA DE FONTES EM MISTURAS INSTANTÂNEAS E
ANECOICAS EMPREGANDO ESPARSIDADE E CLUSTERIZAÇÃO

Jorge Costa Pires Filho

TESE 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 DOUTOR
EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof^a. Mariane Rembold Petraglia, Ph.D.

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

Prof. José Manoel de Seixas, D.Sc.

Prof. Lisandro Lovisoló, D.Sc.

Prof. Paulo Bulkool Batalheiro, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2015

Pires Filho, Jorge Costa

Separação Cega de Fontes em Misturas Instantâneas e Anecoicas Empregando Esparsidade e Clusterização/Jorge Costa Pires Filho. – Rio de Janeiro: UFRJ/COPPE, 2015.

XIX, 193 p.: il.; 29, 7cm.

Orientadora: Mariane Rembold Petraglia

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2015.

Referências Bibliográficas: p. 172 – 183.

1. Separação de Fontes.
2. Misturas Anecoicas.
3. Misturas Instantâneas.
4. Caso subdeterminado.
5. Caso Determinado. I. Petraglia, Mariane Rembold. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

*Aos meus pais e aos meus filhos
dos quais furtei horas de atenção
e dedicação*

Agradecimentos

Gostaria de agradecer a todos que de alguma forma contribuíram para a realização desse trabalho, em especial à minha orientadora, Mariane Rembold Petraglia, pelas sugestões e correções efetuadas ao longo do processo de elaboração do trabalho, ao CMG(EN) Jorge Amaral Alves e à CC(EN) Carla de Sousa Martins, por terem me propiciado tempo e tranquilidade para que fosse possível a minha dedicação no trabalho e ao meu amigo Diego Barreto Haddad, pelas longas discussões as quais ajudaram na elucidação de diversas dúvidas.

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

SEPARAÇÃO CEGA DE FONTES EM MISTURAS INSTANTÂNEAS E ANECOICAS EMPREGANDO ESPARSIDADE E CLUSTERIZAÇÃO

Jorge Costa Pires Filho

Março/2015

Orientadora: Mariane Rembold Petraglia

Programa: Engenharia Elétrica

Neste trabalho, são investigadas técnicas de separação cega de fontes sonoras para misturas instantâneas e anecoicas, nos casos determinados e subdeterminados, baseadas na propriedade de esparsidade das fontes. Novas abordagens são propostas para o problema de estimação dos coeficientes das matrizes de misturas a partir de técnicas de clusterização em espaços unidimensionais, através da detecção de picos ou busca de regiões de baixa variância. A redução no espaço das soluções é possível através da combinação do princípio da esparsidade das fontes com a minimização da correlação entre as fontes estimadas ou com a maximização das suas medidas de curtose. São apresentados resultados de testes comparativos de desempenho dos algoritmos propostos com os clássicos SOBI, TIFROM, JADE e FastICA, para o caso instantâneo, e com versões dos algoritmos DUET e AD-TIFROM-CF, para o caso anecoico. Além disso, são avaliados os desempenhos dos algoritmos na presença de ruídos nos sensores.

Por fim, são desenvolvidas novas abordagens de reconstrução das fontes no caso subdeterminado, tanto para misturas instantâneas quanto para anecoicas, não sendo necessária, em algumas dessas abordagens, a disjunção das fontes no domínio da transformada para se obter a reconstrução. Para alcançar tal objetivo, usa-se uma técnica de subtração temporal combinada com a minimização de uma função objetivo. Resultados de testes comparativos de desempenho com alguns algoritmos clássicos de reconstrução, tais como os métodos de mascaramento binário e de minimização da norma l_1 , são apresentados.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

BLIND SOURCE SEPARATION FROM INSTANTANEOUS AND ANECHOIC MIXTURES USING SPARSITY AND CLUSTERIZATION

Jorge Costa Pires Filho

March/2015

Advisor: Mariane Rembold Petraglia

Department: Electrical Engineering

In this work, blind separation techniques for sound sources are investigated for instantaneous and anechoic mixtures, in the determined and undetermined cases, based on sparsity property of the sources. New approaches are proposed, where the estimation of the coefficients of the mixture matrices is solved by clustering techniques in one-dimensional spaces, through peak detection or by searching for low variance regions. The reduction in the space of solutions is achieved by combining the principle of the sparsity of the sources with the minimization of the correlation among the estimated sources or the maximization of their kurtosis. The results of comparative performance tests of the proposed algorithms with classical algorithms SOBI, TIFROM, JADE and FastICA, for the instantaneous case, and with versions of the DUET and AD-TIFROM-CF algorithms for the anechoic case. In addition, the performances of the algorithms in the presence of noise in the sensors are evaluated.

Finally, new techniques of reconstruction of the sources are developed for both instantaneous and anechoic mixtures in the underdetermined case, not being required in some of these approaches that the sources be disjoint in the transform domain to obtain the reconstruction. To achieve this goal, a temporal subtraction technique combined with the minimization of an objective function is employed. The results of comparative performance tests with some classical reconstruction algorithms, such as the binary masking and the norm l_1 minimization methods, are presented.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xvii
1 Introdução	1
1.1 Objetivos	2
1.2 Metodologia Empregada	6
1.3 Organização da Tese	7
2 Conceitos Básicos de Separação Cega de Fontes	8
2.1 Cenários de Misturas de Fontes	9
2.2 Análise de Componentes Esparsas	13
2.2.1 Pré-Processamento dos Sinais das Misturas	13
2.2.2 Obtenção do Vetor de Características	13
2.2.3 Estimação dos Coeficientes do Sistema de Mistura	17
2.3 Medidas de Avaliação das Técnicas de Separação	18
2.4 Revisão Bibliográfica	19
2.5 Algoritmos Clássicos com Esparsidade	28
2.5.1 Algoritmo SCA	28
2.5.2 Algoritmo DUET	31
2.5.3 Algoritmo TIFROM	34
3 Métodos de Estimação da Matriz de Mistura para o Caso Instantâneo	36
3.1 Algoritmos Propostos	37
3.1.1 Algoritmo SCAM	38
3.1.2 Algoritmo SCAC	42
3.1.3 Algoritmo SCAz	44
3.1.4 Algoritmo STUE	45
3.2 Desempenho dos Algoritmos Propostos em Misturas sem Ruído	46
3.2.1 Algoritmos Clássicos	47
3.2.2 Algoritmos SCAM e SCAC no Domínio do Tempo	48

3.2.3	Algoritmo SCAM no Domínio Tempo-Frequência	51
3.2.4	Algoritmo SCAC no Domínio Tempo-Frequência	53
3.2.5	Algoritmos SCAz e STUE	55
3.3	Desempenho dos Algoritmos Propostos em Misturas com Ruído . . .	59
3.3.1	Resultados com Ruído Branco	59
3.3.2	Resultados com Ruído <i>Babble</i>	64
3.4	Estudo dos Algoritmos Propostos Empregando Medidas Obtidas após a Reconstrução das Fontes	69
3.4.1	Resultados de Experimentos Individuais	69
3.5	Resultados Gerais na Presença de Ruído	76
4	Métodos de Estimação da Matriz de Mistura para o Caso Anecoico	84
4.1	Algoritmos Propostos para Misturas Anecoicas	85
4.1.1	Algoritmo ATIFROM	85
4.1.2	Algoritmo STUE	91
4.1.3	Algoritmo IMUE	92
4.2	Desempenho dos Algoritmos Sem a Presença de Ruído	95
4.3	Desempenho dos Algoritmos na Presença de Ruído	101
4.3.1	Misturas com Ruído Branco nos Sensores	101
4.3.2	Misturas com Ruído <i>Babble</i> nos Sensores	108
4.4	Desempenho dos Algoritmos após a Reconstrução das Fontes	115
5	Algoritmos de Reconstrução das Fontes	122
5.1	Principais Métodos de Reconstrução	123
5.1.1	Mascaramento	123
5.1.2	Módulo Mínimo	125
5.2	Métodos Propostos	127
5.2.1	Deflação na Frequência	127
5.2.2	Correlação Mínima	129
5.2.3	Solução Exata	130
5.2.4	Solução Aproximada	132
5.2.5	Mascaramento Corrigido	133
5.2.6	Subtração Temporal	134
5.3	Desempenhos dos Algoritmos de Reconstrução	136
5.3.1	Avaliações da Convergência e da Recursividade	137
5.3.2	Resultados com Misturas Instantâneas - Caso Subdeterminado	139
5.3.3	Resultados com Misturas Anecoicas - Caso Subdeterminado .	143
5.3.4	Resultados com Misturas Anecoicas - Caso Determinado . . .	146

6	Avaliação de Algoritmos de Separação de Fontes Aplicados no Reconhecimento Automático de Instrumentos Musicais em Misturas Polifônicas	149
6.1	Classificador de Instrumentos Musicais	150
6.2	Sistema de Reconhecimento Automático de Instrumentos Musicais . .	152
6.3	Resultados do Experimento	154
6.3.1	Misturas Instantâneas	155
6.3.2	Misturas Anecoicas	159
7	Conclusões e Trabalhos Futuros	165
7.1	Métodos de Estimação dos Coeficientes do Sistema de Mistura	165
7.2	Métodos de Reconstrução das Fontes	168
7.3	Emprego no SRAIM	169
7.4	Trabalhos Futuros	170
	Referências Bibliográficas	172
A	Fontes Sonoras e Ruídos	184
B	Sistemas de Mistura	188
B.1	Atenuações	188
B.2	Atrasos	189
B.3	Avaliação do Erro da Estimativa da Matriz de Mistura	190
C	Lista dos Métodos Elaborados nesta Tese	191

Lista de Figuras

1.1	Características dos Algoritmos de BSS que Exploram a Esparsidade das Fontes: Algoritmos Propostos (em vermelho), Clássicos (em azul) e Abordagens Não Exploradas (\times).	4
1.2	Esquema Geral Ilustrando o Emprego de Estimativas das Fontes como Passo Inicial do Algoritmo de Estimativa do Sistema de Mistura.	5
1.3	Etapa de Estimação dos Coeficientes de Mistura.	6
1.4	Etapa de Reconstrução das Fontes.	6
2.1	Sequência de Notas de Palhetas	29
2.2	Misturas Instantâneas da Sequência de Notas de Palhetas.	29
2.3	Histograma de Razão das Partes Reais das Misturas no Domínio da Transformada.	30
2.4	Histogramas Angulares Convencional (em vermelho) e Obtido pela Função de Zibulevsky (em azul).	31
2.5	Histograma Bidimensional de Atenuação e Atraso para 3 Fontes e 2 Misturas.	32
2.6	Histograma Combinado de Atenuação e Atraso para 3 Fontes e 2 Misturas.	33
2.7	Variância da Razão das Partes Reais das Misturas nas Raias 3, 10 e 12.	34
3.1	Esquema Geral do Algoritmo SCAM.	41
3.2	Esquema Geral do Algoritmo SCAC.	44
3.3	Esquema Geral do Algoritmo SCAZ.	45
3.4	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Voz.	47
3.5	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Áudio.	48
3.6	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Propostos SCAM-t e SCAC-t, e dos Clássicos SOBI e TIFROM - Sinal de Voz.	49

3.7	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Propostos SCAm-t e SCAc-t, e dos Clássicos TIFROM e JADE - Sinal de Áudio.	51
3.8	MSE dos Elementos das Matrizes de Mistura Estimadas pelo Algoritmo SCAmTF com 3 Tipos de Filtragem e pelos Algoritmos SOBI e TIFROM - Sinal de Voz.	52
3.9	MSE dos Elementos das Matrizes de Mistura Estimadas pelo Algoritmo SCAmTF com 3 Tipos de Filtragem - Sinal de Áudio.	53
3.10	MSE dos Elementos das Matrizes de Mistura Estimadas pelo Algoritmo SCAcTF - Sinal de Voz.	54
3.11	MSE dos Elementos das Matrizes de Mistura Estimadas pelo Algoritmo SCAcTF - Sinal de Áudio.	55
3.12	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos SCAzTF e STUE - Sinal de Voz.	56
3.13	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos SCAzTF e STUE - Sinal de Áudio.	57
3.14	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Voz - Ruído Branco, SNR = 10 dB.	60
3.15	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Áudio - Ruído Branco, SNR = 10 dB.	60
3.16	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Propostos - Sinal de Voz - Ruído Branco, SNR = 10 dB.	61
3.17	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Áudio - Ruído Branco, SNR = 10 dB.	62
3.18	Valores Médios do MSE em Função da SNR para Sinais de Voz com Ruído Branco.	63
3.19	Valores Médios do MSE em Função da SNR para Sinais de Áudio com Ruído Branco.	64
3.20	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Voz - Ruído <i>Babble</i> , SNR = 10 dB.	65
3.21	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Áudio - Ruído <i>Babble</i> , SNR = 10 dB.	65
3.22	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Propostos - Sinal de Voz - Ruído <i>Babble</i> , SNR = 10 dB.	66
3.23	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Propostos - Sinal de Áudio - Ruído <i>Babble</i> , SNR = 10 dB.	66
3.24	Valores Médios do MSE em Função da SNR para Sinais de Voz com Ruído <i>Babble</i>	68

3.25	Valores Médios do MSE em Função da SNR para Sinais de Áudio com Ruído <i>Babble</i>	68
3.26	Histogramas de DOA para Sinais de Voz.	72
3.27	SDR (em dB) do Algoritmo SCAMTF para Diferentes Funções Objetivo e do Algoritmo SCAcTF com Sinal de Voz em Inglês para Cada Cenário.	72
3.28	SDR (em dB) do Algoritmo SCAMTF para Diferentes Funções Objetivo e do Algoritmo SCAcTF com Sinal de Áudio (Flautas) para Cada Cenário.	73
3.29	SDR (em dB) dos Algoritmos SCAMTF e SCAcTF com Diferentes Filtragens do Vetor de Características para Sinal de Voz em Inglês.	74
3.30	SDR (em dB) dos Algoritmos SCAMTF e SCAcTF com Diferentes Filtragens do Vetor de Características para Sinal de Áudio (Flautas).	74
3.31	SDR (em dB) dos Algoritmos SCAM-t e SCAc-t com Emprego da Detecção de Pico e da Moda, e dos Algoritmos JADE e SOBI, para Sinais de Voz.	75
4.1	Esquema Geral do Algoritmo ATIFROM.	86
4.2	Diagrama de um Banco de Filtros DFT Polifásico.	87
4.3	Resposta em Frequência dos Filtros de um Banco DFT Uniforme.	87
4.4	Diagrama de um Banco de Filtros DFTNU Polifásico.	88
4.5	Resposta em Frequência do Banco de Filtros DFTNU - $\lambda = 0,5$	89
4.6	Estrutura de um Banco de Filtros WDFLT com Componentes Polifásicas.	89
4.7	Resposta em Frequência do Banco de Filtros WDFLT com $\lambda = -0,5$	90
4.8	Exemplo de Determinação do Atraso por Meio do Método de Correlação Cruzada. Máximo em $Lag = -30$, Resultando em $Atraso=Lag/Upsampling = -1,5$	90
4.9	Esquema Geral do Algoritmo STUE.	91
4.10	Fluxo de Decisão do Algoritmo IMUE.	92
4.11	Evolução dos Coeficientes de Atenuação do Algoritmo IMUE.	94
4.12	Evolução dos Coeficientes de Atraso do Algoritmo IMUE.	95
4.13	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM - Voz.	96
4.14	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM - Áudio.	98
4.15	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE - Voz.	98
4.16	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE - Áudio.	99

4.17	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE - Voz.	99
4.18	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE - Áudio.	100
4.19	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM com Ruído Branco - Voz (SNR = 10 dB).	102
4.20	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM com Ruído Branco - Áudio (SNR = 10 dB).	102
4.21	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE com Ruído Branco - Voz (SNR = 10 dB).	103
4.22	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE com Ruído Branco - Áudio (SNR = 10 dB).	103
4.23	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE com Ruído Branco - Voz (SNR = 10 dB).	104
4.24	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE com Ruído Branco - Áudio (SNR = 10 dB).	104
4.25	MSE dos Coeficientes de Atenuação \times SNR - Voz com Ruído Branco.	106
4.26	MSE dos Coeficientes de Atraso \times SNR - Voz com Ruído Branco. . .	106
4.27	MSE dos Coeficientes de Atenuação \times SNR - Áudio com Ruído Branco.	107
4.28	MSE dos Coeficientes de Atraso \times SNR - Áudio com Ruído Branco. .	107
4.29	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM com Ruído <i>Babble</i> - Voz (SNR = 10 dB).	109
4.30	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM com Ruído <i>Babble</i> - Áudio (SNR = 10 dB).	109
4.31	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE com Ruído <i>Babble</i> - Voz (SNR = 10 dB).	110
4.32	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE com Ruído <i>Babble</i> - Áudio (SNR = 10 dB).	110

4.33	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE com Ruído <i>Babble</i> - Voz (SNR = 10 dB).	111
4.34	MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE com Ruído <i>Babble</i> - Áudio (SNR = 10 dB).	111
4.35	MSE dos Coeficientes de Atenuação \times SNR - Voz com Ruído <i>Babble</i> .	113
4.36	MSE dos Coeficientes de Atraso \times SNR - Voz com Ruído <i>Babble</i> .	113
4.37	MSE dos Coeficientes de Atenuação \times SNR - Áudio com Ruído <i>Babble</i> .	114
4.38	MSE dos Coeficientes de Atraso \times SNR - Áudio com Ruído <i>Babble</i> .	114
5.1	Mascaramento <i>HARD</i> para uma Mistura com 3 Fontes e 2 Sensores.	124
5.2	Módulo Mínimo do Tipo <i>HARD</i> para uma Mistura com 3 Fontes e 2 Sensores.	126
5.3	Módulo Mínimo do Tipo <i>SOFT</i> para uma Mistura com 3 Fontes e 2 Sensores.	126
5.4	Deflação em Frequência para uma Mistura com 3 Fontes e 2 Sensores.	129
5.5	Correlação Mínima do Tipo <i>HARD</i> para uma Mistura com 3 Fontes e 2 Sensores.	131
5.6	Correlação Mínima do Tipo <i>SOFT</i> para uma Mistura com 3 Fontes e 2 Sensores.	131
5.7	Curva de Evolução da SIR Obtida pelo Algoritmo <i>MaskCorr</i> .	137
5.8	Curva de Evolução da SDR Obtida pelo Algoritmo <i>MaskCorr</i> .	138
5.9	Curva de Evolução da SAR Obtida pelo Algoritmo <i>MaskCorr</i> .	138
6.1	Classificador Empregado no SRAIM.	150
6.2	SRAIM Proposto.	152
6.3	Histograma de Ocorrências do Saxofone Soprano.	154
6.4	Resultados da SDR com Metais.	155
6.5	Resultados da SIR com Metais.	156
6.6	Taxa de Acerto para Misturas Monofônicas - Metais.	157
6.7	Taxa de Acerto para Misturas Monofônicas - Percussão.	157
6.8	Taxa de Acerto para Misturas Monofônicas - Palhetas.	158
6.9	Taxa de Acerto para Misturas Monofônicas - Cordas.	158
6.10	Resultados da SDR dos Algoritmos TIFROM_A nec e STUE para Metais.	160
6.11	Resultados da SIR dos Algoritmos TIFROM_A nec e STUE para Metais.	160
6.12	Resultados da SAR dos Algoritmos TIFROM_A nec e STUE para Metais.	161
6.13	Taxa de Acerto para Misturas Monofônicas - Metais.	161

6.14	Taxa de Acerto para Misturas Monofônicas - Percussão.	162
6.15	Taxa de Acerto para Misturas Monofônicas - Palhetas.	162
6.16	Taxa de Acerto para Misturas Monofônicas - Cordas.	163
A.1	Espectrogramas de 3 Sinais de Voz em Inglês (a)-(c) e da sua Soma (d).	185
A.2	Espectrogramas de 3 Sinais de Voz em Japonês (a)-(c) e da sua Soma (d).	185
A.3	Espectrogramas de 3 Sinais de Flautas (a)-(c) e da sua Soma (d). . .	186
A.4	Espectrogramas de 3 Sinais Formados por Sequências de Notas de Instrumentos Musicais (Metais) (a)-(c) e da sua Soma (d).	186
A.5	Espectrogramas de um Sinal de Cada Tipo de Ruído: Branco (a) e <i>Babble</i> (b).	187

Lista de Tabelas

3.1	Estatísticas dos MSEs Obtidos com os Algoritmos Propostos e Clássicos para Sinais de Voz.	57
3.2	Estatísticas dos MSEs Obtidos com os Algoritmos Propostos e Clássicos para Sinais de Áudio.	58
3.3	Estatísticas dos MSEs Obtidos pelos Algoritmos Propostos e Clássicos para Sinais de Voz com Ruído Branco (SNR=10 dB).	62
3.4	Estatísticas dos MSEs Obtidos pelos Algoritmos Propostos e Clássicos para Sinais de Áudio com Ruído Branco (SNR=10 dB).	63
3.5	Estatísticas dos MSEs Obtidos pelos Algoritmos Propostos e Clássicos para Sinais de Voz com Ruído <i>Babble</i> (SNR=10 dB).	67
3.6	Estatísticas dos MSEs Obtidos pelos Algoritmos Propostos e Clássicos para Sinais de Áudio com Ruído <i>Babble</i> (SNR=10 dB).	67
3.7	Características dos Algoritmos Avaliados.	70
3.8	Avaliação dos Algoritmos SCAM-t e SCAc-t com e sem Pré-Branqueamento para Sinais de Voz em Inglês.	70
3.9	Avaliação dos Algoritmos SCAM-t e SCAc-t com e sem Pré-Branqueamento para Sinais de Voz em Japonês.	70
3.10	Avaliação dos Algoritmos SCAM-t e SCAc-t com e sem Pré-Branqueamento para Sequências de Notas Musicais de Instrumentos Metálicos.	71
3.11	Avaliação dos Algoritmos SCAM-t e SCAc-t com e sem Pré-Branqueamento para Sinais de Áudio.	71
3.12	Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Voz na Presença de Ruído Branco com Diferentes SNRs.	77
3.13	Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Áudio na Presença de Ruído Branco com Diferentes SNRs.	78

3.14	Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Voz na Presença de Ruído <i>Babble</i> com Diferentes SNRs.	79
3.15	Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Áudio na Presença de Ruído <i>Babble</i> com Diferentes SNRs.	80
4.1	Estatísticas dos Algoritmos para Sinais de Voz.	97
4.2	Estatísticas dos Algoritmos para Sinais de Áudio.	100
4.3	Estatísticas dos Algoritmos para Sinais de Voz com Ruído Branco (SNR = 10 dB).	105
4.4	Estatísticas dos Algoritmos para Sinais de Áudio com Ruído Branco (SNR = 10 dB).	105
4.5	Estatística dos Algoritmos para Sinais de Voz com Ruído <i>Babble</i> (SNR = 10 dB).	112
4.6	Estatística dos Algoritmos para Sinais de Voz e Áudio com Ruído <i>Babble</i> (SNR = 10 dB).	112
4.7	Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Voz na Presença de Ruído Branco com Diferentes SNRs.	116
4.8	Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Áudio na Presença de Ruído Branco com Diferentes SNRs.	117
4.9	Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Voz na Presença de Ruído <i>Babble</i> com Diferentes SNRs.	118
4.10	Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Áudio na Presença de Ruído <i>Babble</i> com Diferentes SNRs.	119
5.1	Algoritmos com Recursão - Áudio - 3 Fontes.	139
5.2	Algoritmos com Recursão - Voz em Inglês - 3 fontes.	139
5.3	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Voz em Inglês - 3 Fontes.	140
5.4	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Voz em Inglês - 4 Fontes.	140
5.5	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Áudio - 3 Fontes.	140
5.6	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Áudio - 4 Fontes.	141

5.7	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Voz em Japonês - 3 Fontes.	141
5.8	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Voz em Japonês - 4 Fontes.	141
5.9	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Seq. de Notas - 3 Fontes.	142
5.10	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Seq. de Notas - 4 Fontes.	142
5.11	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Voz em Inglês - 3 Fontes.	143
5.12	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Voz em Inglês - 4 Fontes.	143
5.13	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Voz em Japonês - 3 Fontes.	144
5.14	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Voz em Japonês - 4 Fontes.	144
5.15	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Áudio - 3 Fontes.	144
5.16	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Áudio - 4 Fontes.	145
5.17	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Seq. de Notas - 3 Fontes.	145
5.18	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Seq. de Notas - 4 Fontes.	145
5.19	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Voz em Inglês.	146
5.20	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Voz em Japonês.	147
5.21	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Áudio.	147
5.22	SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Sequência de Notas.	147
6.1	Classes de Instrumentos Musicais do Classificador.	151
6.2	Taxa de Acerto do SRAIM.	154
6.3	Porcentagem da Identificação Correta dos Instrumentos nas Misturas.	163
A.1	Curtose dos Sinais Empregados.	186
A.2	Correlação entre as Fontes dos Sinais Empregados.	187

Capítulo 1

Introdução

A separação cega de fontes a partir de observações de misturas de seus sinais, adquiridas por um conjunto (*array*) de sensores, encontra aplicação em diversas áreas, tais como: eletroencefalografia (EEG) [1]; magnetoencefalografia (MEG) [2]; eletrocardiografia (ECG) [3]; ressonância magnética funcional (fMRI) [4]; acústica [5]; e rádio [6]. Este problema pode ser abordado com o emprego de diferentes técnicas, tais como: filtragem adaptativa [7], subtração espectral [8], conformação de feixe (*beamforming*) [9] e separação cega de fontes [10].

Em particular, os algoritmos de separação cega de fontes (BSS, *Blind Source Separation*) exigem pouco¹ (ou mesmo nenhum) conhecimento acerca das fontes, permitindo-nos aduzir o adjetivo “cega” a estas técnicas. Cabe ao algoritmo estimar os parâmetros do sistema de mistura para posteriormente reconstruir os sinais $s_i(t)$ emitidos por cada fonte, a partir das misturas $x_i(t)$ adquiridas pelos sensores, sem o emprego de informações específicas sobre as fontes.

Neste contexto, as misturas adquiridas pelos sensores podem apresentar características distintas, dependendo do cenário, resultando em alterações no método BSS a ser empregado. Assim, podemos classificar as misturas em lineares e não lineares, onde as lineares podem ser subdivididas em três categorias distintas: instantâneas, quando não há atrasos nem reflexões entre as fontes e os sensores, apenas atenuações; anecoicas, quando não há reflexões entre as fontes e os sensores, apenas atrasos e atenuações; e convolutivas ou ecóicas, quando há atenuações, atrasos e reflexões entre fontes e sensores.

Os algoritmos elaborados nesta tese empregam o princípio da esparsidade das fontes, ou seja, consideram que a maioria das amostras dos seus sinais são nulas ou próximas de zero num determinado domínio. Tal característica, na prática, ocorre em diversos sinais, tais como sinais de radares (ELINT, *Electronic Intelligence*) [11]), cardiorespiratórios [12], infra-vermelhos em meio interestelar [13], etc.

¹Nesta tese, se assumirá conhecido o número de fontes presentes no cenário.

Embora os sinais empregados nesta tese sejam sonoros, as diversas técnicas propostas podem ser facilmente aplicadas, com algumas modificações, para outros tipos de sinais, como por exemplo, sísmicos [14] ou eletromagnéticos [15] (radares). Para sinais de áudio e voz, tais técnicas também são passíveis de utilização na retirada de ruídos presentes numa gravação ou extração de fontes, podendo ser combinadas com algoritmos de classificação, tanto para a identificação de instrumentos musicais (catalogação de discotecas) quanto na identificação de notas para transcrição automática [16].

Esta tese possui emprego em um amplo espectro de aplicações, sendo a utilidade dos algoritmos propostos estabelecida pela adequação aos princípios contidos nas suas respectivas implementações, diante dos cenários confrontados. Assim, o uso destes algoritmos não está limitado a um problema específico, mas, basicamente, a qualquer problema em que se busca separar as fontes em cenários instantâneos e anecoicos, e que a esparsidade das fontes esteja presente.

Esta introdução está dividida em três seções. A primeira trata dos objetivos da tese, a segunda descreve a metodologia empregada, enquanto a terceira dispõe sobre sua organização.

1.1 Objetivos

O presente trabalho tem como um de seus objetivos avaliar a introdução de novos princípios, quando combinados com a esparsidade, para a estimação dos parâmetros do sistema de mistura, considerando ambientes ruidosos.

Dentre os princípios empregados neste trabalho, os três principais são: o desacoplamento dos sensores, o emprego recursivo de estimativas das fontes no lugar das misturas, e a baixa correlação entre as estimativas das fontes. Não se encontram na literatura realizações que empreguem princípio da esparsidade combinado com esses princípios.

Dessa forma, pretende-se elaborar algoritmos que empreguem esses novos princípios combinados com a esparsidade, e que separem sinais presentes em misturas instantâneas e anecoicas, particularmente para o caso desafiador, denominado subdeterminado, quando o número de sensores é menor que o número de fontes. Os algoritmos de separação de fontes em geral apresentam duas etapas: a de estimação do sistema de mistura e a de reconstrução das fontes. Normalmente, o problema de reconstrução das fontes no caso subdeterminado se apresenta como um problema de maior grau de complexidade do que o problema de estimação dos parâmetros do sistema de mistura, visto que, mesmo que se consiga estimar corretamente esses parâmetros, a reconstrução das fontes pode vir a apresentar interferências das demais fontes e/ou outras distorções.

Avaliaremos também os desempenhos dos algoritmos propostos e de alguns algoritmos clássicos (p. ex. TIFROM, *Time-Frequency Ratio of Mixtures*, [17]) que também utilizam o princípio da esparsidade, em situações onde nem sempre os sinais são fortemente esparsos, ou seja, quando as fontes não se apresentam completamente disjuntas no domínio da transformada empregada, e também em situações em que as fontes apresentam características distintas de gaussianidade.

Por fim, avaliaremos a taxa de acerto obtida pelo Sistema de Reconhecimento Automático de Instrumentos Musicais (SRAIM) [18] quando utilizados alguns dos algoritmos propostos nesta tese para separação dos sinais de diferentes instrumentos, ilustrando um dos possíveis empregos das técnicas elaboradas.

O desacoplamento dos sensores resolve o problema de estimação dos parâmetros do sistema de mistura num espaço unidimensional, enquanto que o emprego do princípio da simultaneidade das amostras (presente em misturas instantâneas) resolve o problema de estimação dos parâmetros do sistema de mistura num espaço de dimensão igual ao número de sensores.

O desacoplamento dos sensores introduz uma alteração na abordagem da técnica de análise de componentes esparsos quando há mais de duas misturas, obrigando o uso de uma função de reposicionamento dos parâmetros estimados. Diferentemente, o método clássico normalmente resolve o problema da separação cega das fontes utilizando as amostras de todos os sensores simultaneamente, valendo-se também de histogramas ou de regiões discriminantes (considerando que informações de algumas regiões do plano T-F são mais significativas para a separação das fontes do que as de outras regiões), como são por exemplo, as amostras SSP (*Single Source Points*) [19] ou as amostras presentes nas SSZ (*Single Source Zones*) [17].

O desacoplamento dos sensores, em geral, promove uma menor degradação do método na presença de ruído, uma vez que permite o tratamento diferenciado do sinal de cada sensor, o que não é possível com a abordagem clássica com esparsidade.

Além disso, o desacoplamento dos sensores requer o uso de menos memória durante a fase de estimação dos coeficientes do que o método clássico. Isso ocorre porque os coeficientes de cada mistura podem ser estimados em momentos distintos, o que possibilita o reuso da memória empregada nessas estimativas. Essa redução de memória pode ser vantajosa principalmente em cenários com alta taxa de amostragem, onde memórias rápidas possuem custo elevado. Contudo, se o desempenho for o fator imperativo, pode-se empregar vários DSPs numa arquitetura paralela, resolvendo o problema de forma mais rápida do que a abordagem clássica.

Na Fig. 1.1 apresentamos um quadro comparativo entre as técnicas de separação cega que exploram a esparsidade das fontes em misturas instantâneas, mostrando as diferenças entre as realizações presentes nesta tese e as principais encontradas na literatura.

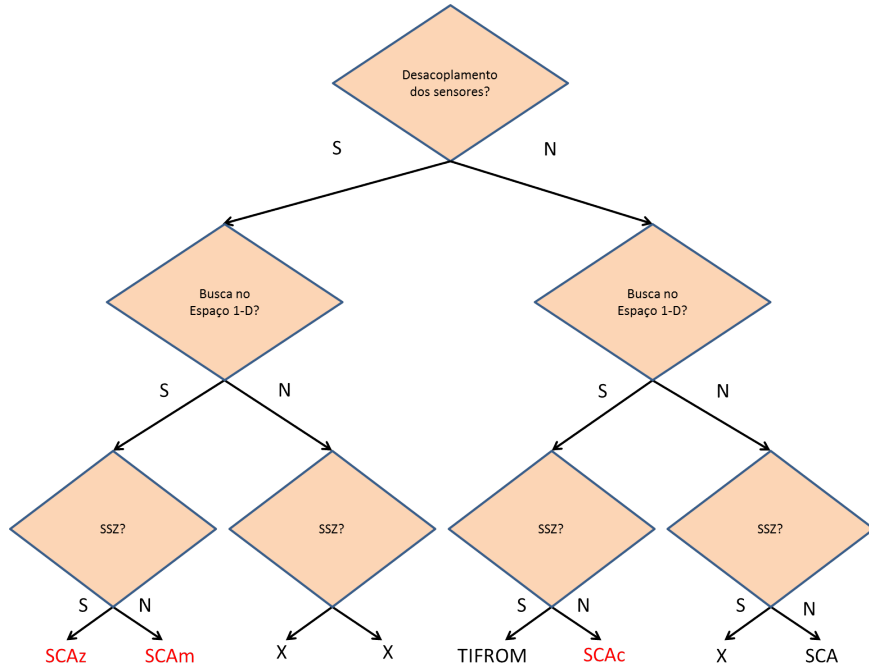


Figura 1.1: Características dos Algoritmos de BSS que Exploram a Esparsidade das Fontes: Algoritmos Propostos (em vermelho), Clássicos (em azul) e Abordagens Não Exploradas (\times).

Os algoritmos SCAm (SCA modificado), SCAz (SCA modificado com SSZ) e SCAC (SCA com classificador), destacados em vermelho, são propostos nesta tese para misturas instantâneas, os quais empregam o princípio da esparsidade combinado com outros princípios (Desacoplamento dos sensores, Busca no espaço 1-D, SSZ), produzindo realizações que não encontram correlatos na literatura.

Assim, a partir da Fig. 1.1, observa-se que a realização SCAm emprega o desacoplamento dos sensores, a busca dos coeficientes num espaço unidimensional e histogramas, sem explorar as regiões SSZ; já a realização SCAz, emprega o desacoplamento dos sensores, a busca dos coeficientes no espaço unidimensional e as regiões SSZ; enquanto a realização SCAC emprega a simultaneidade das amostras, a busca dos coeficientes no espaço unidimensional e histogramas.

As realizações destacadas em preto, ou seja, o SCA [20] e o TIFROM [17] foram propostas na literatura. As folhas da árvore representada na Fig. 1.1, destacadas por um “ \times ”, representam possíveis realizações que não estão presentes na literatura e tampouco foram realizadas nesta tese.

O principal problema do uso das misturas para estabelecer as amostras discriminantes (nas amostras SSP ou nas regiões SSZ) ou a formação de picos de histograma bem posicionados é o fato de que normalmente não se sabe, *a priori*, a transformação adequada que propiciará uma disjunção das fontes no seu domínio. Portanto, em determinadas situações, apesar do emprego de transformadas STFT ou outras vi-

sando esparsificar as misturas, podem ainda ocorrer problemas nas estimativas dos coeficientes, decorrentes das fontes nesse novo domínio não serem adequadamente disjuntas. Contudo, se usarmos no lugar das m misturas as m estimativas (associadas a cada sensor) para cada uma das n fontes, teremos então um mapeamento esparsificador com menor interferência das demais fontes, tornando mais fácil a obtenção das amostras discriminantes ou a formação adequada dos picos dos histogramas. Na Fig. 1.2 apresentamos um esquema geral que ilustra o emprego das estimativas das fontes como sinais de entrada do algoritmo de estimação do sistema de mistura.

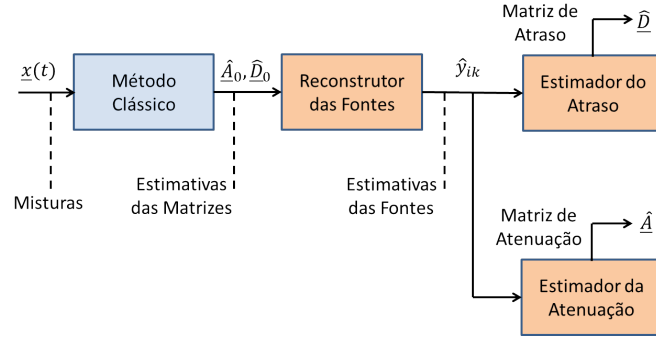


Figura 1.2: Esquema Geral Ilustrando o Emprego de Estimativas das Fontes como Passo Inicial do Algoritmo de Estimativa do Sistema de Mistura.

Nesta tese, o uso desse princípio estará presente em três algoritmos: o ATIFROM (*Attenuated TIFROM with modified delay*), que usa esse passo inicial para refinar as estimativas de atraso; o STUE (*Sparsifying Transform Unmixing Estimation*), que aplica o refinamento das estimativas num número pré-estabelecido de vezes para obter tanto os coeficientes de atenuação quanto os de atraso; e o IMUE (*Iterative Masking Unmixing Estimation*), que emprega o mesmo procedimento para estimar os coeficientes de atraso e de atenuação em um número variável de vezes, de acordo com o cenário confrontado.

Por fim, a medida da correlação entre as estimativas foi usado em três momentos distintos: no primeiro como uma função objetivo, visando viabilizar o desacoplamento dos sensores para estimar os coeficientes de mistura no caso instantâneo nos algoritmos SCAM, SCAz e STUE; no segundo como apoio à decisão na escolha da melhor solução entre um número limitado de possibilidades por meio de um algoritmo guloso, visando obter as estimativas das fontes com correlação mínima (*CorrMin*), em uma realização de reconstrução das fontes para o caso instantâneo; e por fim, visando reconstruir as fontes por meio de técnicas de subtração temporal, a fim de determinar o fator de escala correto para reduzir a indeterminação do sistema subdeterminado a partir de uma ou mais estimativas das fontes disponíveis nos algoritmos $T\text{SOE}_{TF}$, $T\text{SAE}_{T1}$, $eT\text{SAE}_{T2}$.

1.2 Metodologia Empregada

Podemos dividir o conteúdo desta tese de acordo com duas etapas distintas dos algoritmos BSS. Na primeira, investigam-se métodos de estimação dos coeficientes das matrizes de misturas, onde a medida de erro empregada é o MSE (*Mean Square Error*); e na segunda, investigam-se métodos de reconstrução das fontes a partir da matriz de mistura previamente estimada, sendo as medidas de erro empregadas a SDR (*Source to Distortion Ratio*) e a SIR (*Source to Interference Ratio*). As Figs. 1.3 e 1.4 ilustram as duas etapas, respectivamente.

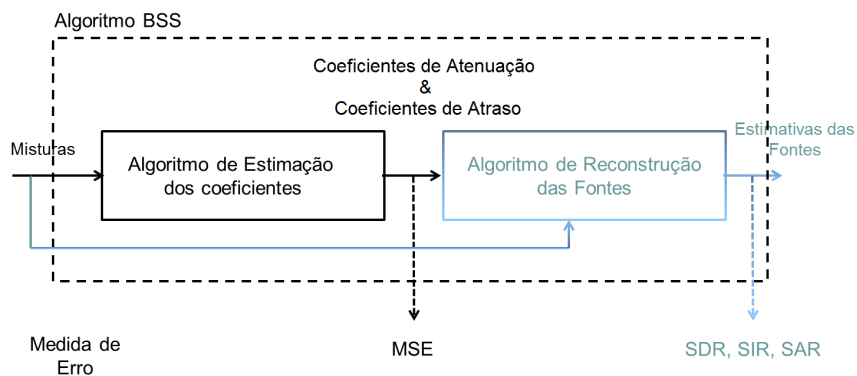


Figura 1.3: Etapa de Estimação dos Coeficientes de Mistura.

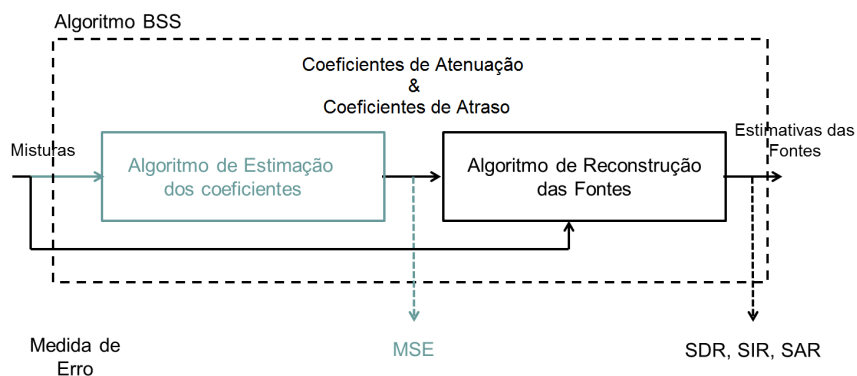


Figura 1.4: Etapa de Reconstrução das Fontes.

A análise desta tese procura verificar se os desempenhos dessas novas realizações são satisfatórios, quando comparados aos desempenhos de algoritmos consagrados descritos no Cap. 3. Para isso, tomou-se a precaução de não alterar as implementações dos algoritmos clássicos disponibilizadas pelos autores, empregando-se os parâmetros de entrada recomendados.

Os códigos referentes a esses algoritmos clássicos se encontram em pacotes específicos de separação de fontes, tal como o ICALAB [21], ou na página dos seus respectivos autores [22].

No caso de misturas anecoicas, os algoritmos clássicos estão descritos no Cap. 2. Houve dificuldade na obtenção de implementações disponibilizadas livremente e codificadas pelos próprios autores dos métodos. Assim, optamos pela implementação de versões dos algoritmos DUET (*Degenerate Unmixing Estimation Technique*) [23] e AD-TIFROM-CF (*Attenuated and Time-Delayed TIFROM Constant-Frequency zones*) [24], os quais usualmente são empregados em separação de misturas anecoicas. Assim, essas versões implementadas dos clássicos DUET e AD-TIFROM-CF são utilizadas como referências nas simulações.

As avaliações em cenários menos esparsos foram feitas a fim de verificar os desempenhos das novas realizações nessas condições. Isso porque os novos princípios empregados podem alterar os desempenhos obtidos pelos algoritmos que usam somente a esparsidade.

1.3 Organização da Tese

O Capítulo 2 faz uma breve introdução teórica aos problemas de separação cega de fontes para os cenários instantâneos, anecoicos e convolutivos; apresenta os principais algoritmos propostos na literatura que empregam esparsidade, as definições das medidas objetivas de erro empregadas, e um resumo de alguns dos principais trabalhos desenvolvidos na área de separação cega de fontes. O Capítulo 3 propõe métodos para estimar os coeficientes das matrizes de mistura em cenários instantâneos, sendo também apresentadas as simulações realizadas para estimar os seus coeficientes, com e sem a presença de ruído nos sensores. O Capítulo 4 descreve os métodos propostos para estimar os coeficientes das matrizes de mistura em cenários anecoicos, e também apresenta resultados de simulações com e sem a presença de ruído nos sensores. O Capítulo 5 traz novas propostas para a reconstrução das fontes, bem como simulações efetuadas com esses algoritmos. O Capítulo 6 utiliza um sistema de reconhecimento automático de instrumentos musicais (SRAIM), que em um de seus blocos emprega as técnicas de separação cega de fontes propostas nesta tese. O SRAIM é exercitado com misturas instantâneas e anecoicas, onde o resultado da separação é representado pela taxa de acerto obtida pelo sistema. No Capítulo 7 são apontadas as conclusões da tese e sugeridos trabalhos futuros. Por fim, os apêndices apresentam os sinais empregados nas simulações desta tese, a forma como foram construídas as matrizes de misturas, e uma lista dos métodos elaborados com uma breve descrição das suas principais características.

Capítulo 2

Conceitos Básicos de Separação Cega de Fontes

O presente capítulo apresenta algumas definições e propriedades relacionadas às técnicas de separação cega de fontes, e descreve os principais tipos de misturas e os problemas encontrados em cada caso, onde são apresentados os possíveis cenários em que os métodos de separação cega são utilizados e a formulação matemática dos sistemas de mistura e de separação para cada cenário. Em seguida é descrito o método SCA, que explora a esparsidade das fontes, e, são apresentadas as medidas utilizadas para a avaliação objetiva dos algoritmos de reconstrução das fontes, além de uma revisão bibliográfica dos principais métodos propostos na área de separação cega de fontes acústicas. Finalmente, são descritos os algoritmos clássicos que empregam a esparsidade das fontes.

Devido à grande abrangência da área de separação cega de fontes, foi necessário limitar o escopo deste trabalho, considerando alguns aspectos do problema. Primeiramente, o sistema mistura-separação pode ser classificado em quatro topologias básicas: SISO (*single input and single output*), SIMO (*single input and multiple output*), MISO (*multiple input and single output*) e MIMO (*multiple input and multiple output*). Para o problema de separação de fontes, considerando que todas são de interesse, somente as topologias MISO, SIMO e MIMO podem ser empregadas. No entanto, o caso SIMO, quando há apenas um sensor, requer o emprego de técnicas distintas. Não abordaremos também o caso MISO, pois somente faz parte do escopo desta tese ter múltiplas fontes. Em segundo lugar, matematicamente, o sistema de mistura dos sinais das fontes pode ser interpretado como uma transformação, a qual pode ser tanto linear quanto não linear. Assim, neste trabalho investigaremos apenas técnicas de separação de misturas lineares para a configuração MIMO.

2.1 Cenários de Misturas de Fontes

As abordagens empregadas na separação cega de fontes costumam ser subdivididas segundo alguns critérios baseados nas seguintes características do sistema:

1. Número de Fontes/Sensores;
2. Tipos de Fontes;
3. Tipos de Mistura.

Assim, dependendo da combinação das características da tríade acima, o problema de separação apresenta particularidades, que resultam em métodos distintos de separação. A rigor, ter conhecimento *a priori* desses elementos do problema sugere que os algoritmos de separação não são totalmente cegos. No entanto, na prática, esses elementos, na maioria das vezes, podem ser estimados independentemente numa fase anterior. Por exemplo, o ambiente no qual o sistema de captura do sinal está inserido “estabelece” o tipo de mistura. A definição do sistema de captura determina os tipos de fontes de interesse, e também o número de sensores presentes. No entanto, em geral, o número de fontes presentes pode não ser conhecido. Para contornar tal indefinição, é sempre possível empregar, anteriormente à aplicação do método de separação, um algoritmo de estimação do número de fontes. Logo, esses elementos, normalmente, são considerados conhecidos na maioria dos algoritmos encontrados, embora alguns deles incluam como uma de suas etapas a estimação do número de fontes presentes.

Normalmente, um dos primeiros critérios a ser avaliado é a configuração em que as observações foram feitas a partir dos números de fontes n e de sensores m (primeiro critério da tríade) presentes no cenário de gravação. Dessa forma, é possível estabelecer de antemão o grau de indeterminação que o sistema apresentará para estimarmos as fontes originais. Assim, em relação ao número de fontes/sensores, podemos ter três configurações básicas:

1. Caso determinado: $m = n$ (mesmo número de sensores e fontes) ;
2. Caso sobredeterminado: $m > n$ (mais sensores do que fontes);
3. Caso subdeterminado: $m < n$ (menos sensores do que fontes).

Nos casos determinado e sobredeterminado lineares, é possível obter o sistema de separação através da “inversão” de um estimador adequado para o sistema de mistura, a partir de formulação matemática, utilizando representação matricial, dos processos de mistura e de separação. No caso sobredeterminado, é possível reduzir a redundância (maior número de observações do que de fontes), aplicando-se a técnica

de análise de componentes principais (PCA) [10], mantendo somente as n principais componentes dos sinais dos m sensores para estimar a matriz de mistura. Contudo, para o caso subdeterminado, não é possível usar a inversa da matriz de mistura para a obtenção de uma estimativa da matriz de separação. Isso devido a termos menos observações m do que fontes n , resultando em um problema mal-condicionado, ou seja, com uma infinidade de soluções. Felizmente, para várias aplicações de interesse, essa infinidade de soluções pode ser reduzida para um número pequeno através da utilização de características como a esparsidade das fontes, permitindo que os sinais originais sejam satisfatoriamente estimados [20, 23].

O segundo critério da tríade acima considera as características das fontes originais. As fontes podem ser classificadas segundo suas propriedades estatísticas. Uma das propriedades mais utilizadas pelos algoritmos de separação é a não-gaussianidade das fontes. Em geral, emprega-se uma medida estatística de quarta ordem, a curtose,

$$kurt(x) \triangleq \frac{E[(x - E[x])^4]}{(E[(x - E[x])^2])^2} - 3 \quad (2.1)$$

onde x é uma variável aleatória (que representa uma fonte) e $E[\cdot]$ representa o valor esperado. Utilizando essa medida, um sinal x pode ser classificado em um dos seguintes tipos: subgaussianos, se $kurt(x) < 0$; gaussiano, se $kurt(x) = 0$; e supergaussiano (ou Laplaciano), se $kurt(x) > 0$. Conforme será visto mais adiante, ter conhecimento prévio dos tipos de fontes que compõem a mistura auxilia na escolha do método de separação mais apropriado. Normalmente, os sinais de voz têm distribuição supergaussiana e, portanto, um dos critérios frequentemente utilizados na sua separação é a maximização da curtose [25].

Por fim, o último critério da tríade avalia o processo de geração das misturas. Na prática, as misturas podem ser distinguidas a partir da forma como o ambiente interfere nas fontes. Pode também haver a introdução de não linearidades na combinação dos sinais das fontes devido a características não-ideais dos elementos que compõem o cenário (como alto-falantes e microfones, no caso de sinais de voz). No entanto, essas não linearidades em geral podem ser desprezadas ou controladas, e, portanto, não são consideradas neste trabalho.

Conforme já fora dito no Cap. 1, existem três categorias de misturas lineares: instantâneas, anecoicas, e convolutivas (ecoicas). No caso de misturas instantâneas, uma amostra do sinal de cada fonte é simultaneamente observada pelos diversos sensores, ou seja, cada uma das amostras dos sinais nos sensores $x_i(t)$ é uma combinação linear das amostras das fontes $s_j(t)$. Assim, o modelo matemático que representa

essa situação é dado por

$$x_i(t) = \sum_{j=1}^n a_{ij} s_j(t) \quad i = 1, \dots, m \quad (2.2)$$

onde m representa o número de sensores e n representa o número de fontes. Na sua forma matricial, para $m = 2$ e $n = 2$, temos

$$\underline{x}(t) = \underline{A} \underline{s}(t) \quad (2.3)$$

onde

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \underline{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad \underline{s}(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix}$$

A busca da solução para esse problema consiste em determinar a matriz \underline{W} que recupera os sinais das fontes a partir das misturas, ou seja,

$$\underline{s}(t) = \underline{W} \underline{x}(t) \quad (2.4)$$

No caso determinado ($m = n$), \underline{W} representa a inversa da matriz de mistura, ou seja $\underline{W} = \underline{A}^{-1}$. A obtenção de \underline{W} usualmente é feita por dois caminhos: estimando-se diretamente os coeficientes da matriz de separação ou estimando-se os coeficientes da matriz de mistura para depois achar a inversa.

No caso de misturas anecoicas, o sinal de cada fonte chega em momentos distintos nos diferentes sensores. Os atrasos sofridos pelo sinal são funções da distância da fonte até cada sensor. Portanto, nesse modelo, as misturas são combinações lineares das fontes com atrasos d_{ij} associados ao par fonte-sensor $[j, i]$, ou seja

$$x_i(t) = \sum_{j=1}^n a_{ij} s_j(t - d_{ij}), \quad i = 1, \dots, m \quad (2.5)$$

onde a_{ij} representa a atenuação sofrida pelo sinal da j -ésima fonte até chegar ao i -ésimo sensor. Na sua forma matricial para $m = 2$ e $n = 2$, temos

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} a_{11} \delta(t - d_{11}) & a_{12} \delta(t - d_{12}) \\ a_{21} \delta(t - d_{21}) & a_{22} \delta(t - d_{22}) \end{bmatrix} * \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} \quad (2.6)$$

onde “*” representa a operação de convolução.

As misturas convolutivas, que ocorrem frequentemente em ambientes acústicos, provocam alterações nos espectros de frequência das fontes, que podem ser modeladas por meio de operações de filtragem. As distorções sofridas pelos sinais das

fontes até um dado sensor são diferentes, uma vez que as fontes estão localizadas em posições distintas. Usualmente, a distorção de cada fonte observada em cada um dos sensores é modelada pela aplicação de um filtro FIR (*Finite Impulse Response*) ao sinal, sendo o número de coeficientes do filtro da ordem de centenas ou milhares para o caso de misturas de sinais de voz em ambientes fechados. Nessa abordagem, os sinais nos sensores $x_i(t)$ podem ser expressos em função dos sinais das fontes $s_j(t)$ como

$$x_i(t) = \sum_{j=1}^n \sum_{k=1}^{k=+\infty} a_{ij}(k) s_j(t-k) \quad i = 1, \dots, m \quad (2.7)$$

ou

$$x_i(t) = \sum_{j=1}^n a_{ij}(t) * s_j(t) \quad i = 1, \dots, m \quad (2.8)$$

onde $a_{ij}(t)$ corresponde à resposta ao impulso do filtro FIR associado à j -ésima fonte à i -ésima mistura. Podemos reescrever as equações acima na forma matricial, como, $m = n = 2$

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} a_{11}(t) & a_{12}(t) \\ a_{21}(t) & a_{22}(t) \end{bmatrix} * \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} \quad (2.9)$$

A solução para os casos anecoico e convolutivo pode ser idealmente obtida estimando-se o sistema de separação a partir das equações

$$\hat{s}_i(t) = \sum_{j=1}^m \hat{w}_{ij}(t) * x_j(t) \quad i = 1, \dots, n \quad (2.10)$$

onde $\hat{w}_{ij}(t)$ são as respostas ao impulso dos filtros FIR de separação, e $\hat{s}_1(t)$ e $\hat{s}_2(t)$ representam as estimativas das fontes, que podem ser escritas na forma matricial, para $m = n = 2$, como

$$\begin{bmatrix} \hat{s}_1(t) \\ \hat{s}_2(t) \end{bmatrix} = \begin{bmatrix} \hat{w}_{11}(t) & \hat{w}_{12}(t) \\ \hat{w}_{21}(t) & \hat{w}_{22}(t) \end{bmatrix} * \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \quad (2.11)$$

Conforme BUCHNER e KELLERMAN [26] e MAKINO *et al.* [27], uma escolha apropriada para os filtros de separação, que resulta na separação exata das fontes a menos de eventuais efeitos de filtragem, é dada por

$$\begin{bmatrix} \hat{w}_{11}(t) & \hat{w}_{12}(t) \\ \hat{w}_{21}(t) & \hat{w}_{22}(t) \end{bmatrix} = \begin{bmatrix} h_{22}(t) & -h_{12}(t) \\ -h_{21}(t) & h_{11}(t) \end{bmatrix} \quad (2.12)$$

onde $h_{ij}(t)$ são iguais às respostas ao impulso $a_{ij}(t)$ dos filtros que modelam as misturas para o caso convolutivo, ou aos impulsos discretos $a_{ij}\delta(t-d_{ij})$ que compõem a matriz de mistura para o caso anecoico.

2.2 Análise de Componentes Esparsas

A técnica de análise de componentes esparsas (SCA) [28, 29] é uma poderosa ferramenta na separação de fontes no caso subdeterminado, quando se tem menos sensores do que fontes. O principal requisito para que essa técnica possa ser bem empregada é que as fontes sejam esparsas em algum domínio, ou seja, possuam uma representação esparsa em alguma base.

Os métodos que utilizam o princípio da esparsidade contêm três etapas, na fase de análise (ou de estimação da matriz de mistura):

1. Pré-processamento dos sinais das misturas;
2. Construção do vetor de características;
3. Estimação dos coeficientes do sistema de mistura.

A fase de reconstrução das fontes, que completa o método de separação [30], será tratada no Cap. 5.

A seguir, descreveremos cada etapa dos métodos SCA, e, em seguida, apresentaremos os principais algoritmos propostos, ilustrando os seus resultados através de exemplos.

2.2.1 Pré-Processamento dos Sinais das Misturas

Geralmente, as técnicas baseadas em SCA se valem das misturas obtidas pelos sensores como entrada de seus algoritmos. No entanto, em alguns casos (como em alguns métodos propostos neste trabalho), a técnica SCA é aplicada a estimativas das fontes, obtidas previamente por algum processo de separação, com o objetivo de melhorá-las.

Na etapa de pré-processamento, além de se retirarem os valores médios dos sinais, é normalmente aplicada uma transformação esparsificadora. Existem várias transformadas que podem ser aplicadas visando obter uma representação mais esparsa das fontes, sendo as mais utilizadas a STFT [24], a wavelet [31], e a CQT [32]. Em alguns casos, a aplicação de uma matriz de branqueamento aos vetores das misturas é suficiente para esparsificar os sinais e obter boas estimativas [33].

2.2.2 Obtenção do Vetor de Características

A obtenção do vetor de características é feita em duas etapas: formação do vetor de características, a partir da razão dos sinais das misturas no domínio da transformada, e filtragem do vetor resultante, para eliminação de dados espúrios.

O vetor de características é formado a partir da razão entre cada mistura e uma mistura de referência, no domínio da transformada. Na maior parte dos métodos propostos [20, 24, 34, 35], a STFT é utilizada como transformada esparsificadora e as razões entre as misturas são obtidas para cada ponto no espaço T-F. Em seguida, são utilizadas as partes reais, imaginárias ou combinações das duas para formar o vetor de características, que pode ser relacionado ao DOA (*Direction of Arrival*) no caso instantâneo. Para o caso anecoico, são utilizados módulos e fases da razão das misturas para formar o vetor de características.

Durante ou após a formação do vetor de características, podem ser aplicados diferentes filtros que visam selecionar as representações que contêm informações adequadas ao processo de estimação dos coeficientes, a ser realizado na próxima fase.

A filtragem do vetor de características é feita visando minimizar o erro de estimação dos coeficientes do sistema de mistura, que pode ocorrer devido à presença de ruído ou a uma esparsificação insuficiente do vetor de misturas, levando à formação de falsos picos ou baricentros nos histogramas utilizados no processo de clusterização. Assim, apesar da transformação de esparsificação ajudar no processo de separação, sua não idealidade pode gerar falsos candidatos a coeficientes da matriz de mistura. A ideia da filtragem reside na eliminação de elementos que sabidamente contribuem de forma equivocada para o processo de estimação desses coeficientes. Existem quatro tipos básicos de filtros:

1. Filtro de Norma [23];
2. Filtro de *Outliers* [32];
3. Filtro SSP (*Single Source Points*) [19];
4. Filtro de Variância Mínima [24, 35, 36].

No Filtro de Norma e SSP, a filtragem é realizada diretamente nas amostras das misturas sobre o conjunto inteiro de dados; no Filtro de *Outlier* as amostras são selecionadas diferentemente em cada subconjunto de dados, sendo os subconjuntos selecionados por um processo prévio de clusterização; no Filtro de Variância Mínima, a busca é realizada em regiões de tamanho fixo, formadas por um processo de segmentação.

Filtro de Norma

O Filtro de Norma elimina amostras das misturas que correspondam a trechos em que nenhuma fonte está ativa ou em que todas as fontes chegam aos microfones com amplitudes muito baixas, podendo, portanto, conter ruído de baixa SNR (*Source to*

Noise Ratio), a ponto de distorcer a informação usada na estimação dos coeficientes da mistura. Assim, o objetivo é filtrar as amostras no domínio da transformada quando as energias das misturas são demasiadamente pequenas, ou seja,

$$\tilde{\underline{R}} = |\underline{X}(\tau, \kappa)| > L \quad \forall(\tau, \kappa) \quad (2.13)$$

$$\underline{X}(\tau, \kappa) = [X_1(\tau, \kappa) \quad X_2(\tau, \kappa)]^T \quad (2.14)$$

onde \underline{X}_1 e \underline{X}_2 representam as misturas x_1 e x_2 no domínio da transformada, τ representa uma discretização temporal, κ uma discretização em frequência, L o valor mínimo a partir do qual se considera que pelo menos uma fonte está ativa e $\tilde{\underline{R}}$ o vetor com as misturas no domínio da transformada após a filtragem. Este tipo de filtro é normalmente empregado pelos algoritmos SCA, e em geral é acompanhado de um dos outros filtros.

Filtro de *Outliers*

O Filtro de *Outliers* foi utilizado no domínio temporal em [32] para o caso anecoico, e apresentado de forma mais generalizada no domínio T-F neste trabalho, estando embutido no algoritmo SCAC proposto no Cap. 3. A ideia desse filtro é, após se obter o vetor de características e estimar os coeficientes iniciais por um método de detecção de pico e clusterização, classificar cada grupo como uma das fontes, eliminando (filtrando) de cada agrupamento os *outliers*. A clusterização prévia normalmente é feita por alguma técnica de clusterização, porém nesta tese usou-se um algoritmo de classificação do tipo “vizinho mais próximo” (1-NN) [37], que representa uma máscara de distância euclideana entre os elementos do vetor de características e os coeficientes inicialmente estimados. A filtragem após a clusterização é dada por

$$\tilde{\underline{R}}_k = \underline{X}_k \text{ se } \left| \frac{\underline{X}_k - \overline{\underline{X}}_k}{\sigma_k} \right| < L \quad (2.15)$$

onde L é o limite a partir do qual a amostra é considerada *outlier*¹, $\tilde{\underline{R}}_k$ e \underline{X}_k representam os vetores de características original e filtrado de seus *outliers* no domínio da transformada associados à k -ésima fonte, e $\overline{\underline{X}}_k$ e σ_k representam a média e o desvio padrão do vetor \underline{X}_k .

Uma característica dessa abordagem é que o número de *outliers* em cada grupo pode ser distinto e independente da sua cardinalidade. Portanto, ao se recalcular a média de cada conjunto após a filtragem, ou seja, sem os elementos “estranhos” ao agrupamento, espera-se uma melhoria na estimativa. Outra distinção desta aborda-

¹Na Eq. (2.15), o limite estabelecido é igual para todos os conjuntos, no entanto, pode-se pensar em limites distintos para cada agrupamento.

gem é que, ao “clusterizar” as amostras em função das fontes a partir de um par de misturas, permite-se que cada coeficiente associado a um par fonte-sensor possa ser estimado independentemente, tornando a estimativa mais robusta.

Filtro SSP

O Filtro SSP foi proposto inicialmente para o caso instantâneo [19]. Ele busca os pontos no espaço T-F onde há apenas uma fonte ativa (*Single Source Points*), a partir das razões das partes reais e das partes imaginárias das transformadas dos sinais de misturas. Definindo

$$\alpha_r(\tau, \kappa) = \arctan\left(\frac{\text{Real}(X_2(\tau, \kappa))}{\text{Real}(X_1(\tau, \kappa))}\right) \quad (2.16)$$

$$\alpha_i(\tau, \kappa) = \arctan\left(\frac{\text{Imag}(X_2(\tau, \kappa))}{\text{Imag}(X_1(\tau, \kappa))}\right) \quad (2.17)$$

É fácil mostrar que, para misturas instantâneas sem a presença de ruído, nas regiões do plano T-F em que somente uma fonte está ativa, tem-se

$$\alpha_r(\tau, \kappa) = \alpha_i(\tau, \kappa) \quad (2.18)$$

Nas regiões onde ocorreram misturas com mais de uma fonte ativa, a igualdade acima não é verificada.

Portanto, os pontos SSP podem ser facilmente encontrados comparando-se os coeficientes α_r e α_i de cada amostra no domínio T-F. Na prática, devido à presença de ruído ou erros de medição, a condição da Eq. (2.18) deve ser relaxada, de forma que $\theta = \alpha_r - \alpha_i \approx 0$, ou seja, procuram-se os pontos para os quais a diferença angular θ seja pequena em módulo. O resultado desta busca incluirá pontos correspondentes a regiões onde ocorra uma forte prevalência de uma das fontes, os quais podem também ser usados para estimar os coeficientes da mistura. A condição utilizada para obter os SSPs é

$$\cos(\theta) > \cos(\theta_L) = c_{tol} \quad (2.19)$$

onde $\theta_L \ll 1$ [19] e $c_{tol} \approx 1$, sendo a função cosseno usada por mapear diferenças angulares pequenas positivas ou negativas em valores positivos próximos de 1. Através de manipulações algébricas, usando-se igualdades trigonométricas, obtém-se

$$\cos(\theta) = \frac{\text{Real}(X_1)\text{Imag}(X_1) + \text{Real}(X_2)\text{Imag}(X_2)}{\sqrt{(\text{Real}(X_1)^2 + \text{Real}(X_2)^2) \times (\text{Imag}(X_1)^2 + \text{Imag}(X_2)^2)}} > c_{tol} \quad (2.20)$$

que é a condição SSP normalmente encontrada na literatura [19].

Filtro de Variância Mínima

O Filtro de Variância Mínima foi proposto originalmente em [36], como parte do algoritmo TIFROM [36]. Nessa abordagem, obtêm-se os pontos SSP através da busca de regiões onde a variância da razão das misturas é pequena. Num cenário ideal de misturas instantâneas sem ruídos, a variância da razão dos sinais de cada par de misturas nas regiões (onde somente uma das fontes está ativa) é nula. Caso haja mais de uma fonte, a razão das misturas passa a variar em função das amplitudes dos sinais, como pode ser facilmente mostrado no domínio do tempo, para o caso com duas fontes. Assumindo a primeira mistura como referência, a razão das misturas no instante k é

$$r(k) = \frac{x_2(k)}{x_1(k)} = \frac{\alpha_1 s_1(k) + \alpha_2 s_2(k)}{s_1(k) + s_2(k)} \quad (2.21)$$

Quando $s_2(k) = 0$ e $s_1(k) \neq 0$ ($k_1 < k < k_2$) então $r(k) = \alpha_1$ e sua variância é zero. Supondo-se que $s_1(k) = s_2(k)$ e $s_2(k+1) = p s_1(k+1)$, onde $p \neq 1$, temos $r(k) = \frac{\alpha_1 + \alpha_2}{2}$ e $r(k+1) = \frac{\alpha_1 + p\alpha_2}{1+p}$, os quais são valores diferentes, sendo, portanto, a variância da razão diferente de zero. Como em geral as fontes não são disjuntas no domínio temporal, normalmente se aplica uma transformação do tipo STFT, visando esparsificar os sinais e aumentar o número de regiões onde apenas uma fonte está presente. Assim, a busca é feita para todas as raiais, eliminando-se os pontos em que a variância é não nula. Na prática, o algoritmo relaxa essa condição e seleciona as n regiões com menores variâncias que possuam razão entre as misturas com valores distintos, sendo n o número máximo de fontes presentes no cenário.

2.2.3 Estimação dos Coeficientes do Sistema de Mistura

Os coeficientes da matriz de mistura são estimados através de um processo de clusterização ou de detecção de picos, a partir de histogramas [29, 35]. O processo de clusterização estima os coeficientes calculando a média dos elementos dos vetores de características de cada grupo (ou *cluster*). Quando as variâncias desses elementos são baixas, as estimativas costumam ser boas [24].

No entanto, quando o agrupamento possui variâncias não muito pequenas, as estimativas podem apresentar distorções. Neste caso, é preferível usar uma técnica de detecção de picos, a qual procura o valor de um parâmetro (por exemplo o DOA) que teve o maior número de ocorrências dentro do conjunto, a partir de uma resolução escolhida (*bin*). Importa ressaltar que o conceito de esparsidade aqui empregado é menos forte do que o utilizado em álgebra linear, já que é suficiente que as amostras das fontes (ou os coeficientes de sua transformada) sejam, em sua maior parte, próximas de zero, ou seja, que poucas amostras das fontes concentrem a maior parte da energia, permitindo o uso de técnicas baseadas em histogramas

para a estimação dos coeficientes.

2.3 Medidas de Avaliação das Técnicas de Separação

Há duas formas distintas de avaliar os desempenhos de algoritmos de separação de fontes de áudio: por meio de avaliações objetivas ou subjetivas. As avaliações objetivas são constituídas por medidas matemáticas que visam determinar o grau de proximidade entre a estimativa obtida pelo método de separação e a fonte original. As avaliações subjetivas podem ser feitas visualmente, comparando-se os espectrogramas ou as formas de onda dos sinais das fontes e das estimativas obtidas, ou auditivamente, atribuindo-se notas aos sinais resultantes em relação à redução de interferências e à qualidade do som, considerando-se a presença de distorções e ruídos.

Foram propostas na literatura diversas medidas para se avaliar a qualidade das estimativas dos sinais das fontes em relação aos sinais originais [38–40], uma vez que a comparação direta entre a estimativa e o sinal original em geral não retrata de forma adequada a qualidade percebida pelo sistema auditivo humano. Descreveremos, nesta seção, as principais medidas empregadas recentemente na avaliação de resultados de algoritmos de separação de áudio, que são a SIR (*Source to Interference Ratio*), a SAR (*Source to Artifact Ratio*), a SDR (*Signal to Distortion Ratio*) e a SNR (*Source to Noise Ratio*), propostas em [40].

Uma vez obtida a estimativa $\hat{\underline{s}}$ de uma dada fonte \underline{s} , o erro entre a estimativa e o sinal da fonte pode ser decomposto como

$$\epsilon_{total} \triangleq \hat{\underline{s}} - \underline{s} = \epsilon_{interf} + \epsilon_{noise} + \epsilon_{artif} \quad (2.22)$$

onde ϵ_{interf} representa o erro na estimativa devido à presença de interferência de uma das outras fontes presentes nas misturas, ϵ_{noise} representa o erro na estimativa devido à presença de componentes de ruído, e ϵ_{artif} representa o erro devido à presença de artefatos na estimativa. Essa decomposição é feita considerando disponíveis os sinais originais das fontes e de ruídos, e permitindo quatro diferentes tipos de distorções: ganhos invariantes no tempo, filtros invariantes no tempo, ganhos variantes no tempo e filtros variantes no tempo. Os componentes do erro são obtidos a partir das projeções das estimativas nos espaços das fontes e de ruídos [40].

Após obterem-se os componentes do erro da Eq. (2.22), as medidas de desempenho são definidas através de razões de energias em decibéis (dB). A distorção entre

cada estimativa $\hat{\underline{s}}$ e o sinal original \underline{s} é medida pela razão

$$\text{SDR} = 10 \log_{10} \frac{\|\underline{s}\|^2}{\|\hat{\underline{s}} - \underline{s}\|^2} = 10 \log_{10} \frac{\|\underline{s}\|^2}{\|\epsilon_{interf} + \epsilon_{noise} + \epsilon_{artif}\|^2} \quad (2.23)$$

Já a quantidade de interferência das outras fontes presente numa dada estimativa é medida por

$$\text{SIR} = 10 \log_{10} \frac{\|\underline{s}\|^2}{\|\epsilon_{interf}\|^2} \quad (2.24)$$

A medida utilizada para quantificar a presença de artefatos na estimativa é dada por

$$\text{SAR} = 10 \log_{10} \frac{\|\underline{s}\|^2}{\|\epsilon_{artif}\|^2} \quad (2.25)$$

A quantidade de ruído presente na estimativa é estimada pela razão

$$\text{SNR} = 10 \log_{10} \frac{\|\underline{s}\|^2}{\|\epsilon_{noise}\|^2} \quad (2.26)$$

Em VINCENT *et al.* [41], é disponibilizado um algoritmo que decompõe os erros e calcula a SIR, a SAR, a SDR e a SNR de cada fonte a partir das fontes originais, das suas estimativas e das misturas. A SNR é somente utilizada na presença de ruído aditivo.

2.4 Revisão Bibliográfica

Esta seção contém uma revisão de alguns dos principais trabalhos desenvolvidos na área de separação cega de sinais. Foram incluídos neste resumo alguns trabalhos que não estão restritos ao caso subdeterminado, mas que podem também ser aplicados neste caso ou que serviram de base para o desenvolvimento de algoritmos para este cenário. Neste ponto, é importante ressaltar que, nesta pesquisa, não consideramos o caso subdeterminado monocanal, em que há apenas uma mistura. O problema de separação de sinais utilizando um único sensor é normalmente tratado à parte e requer a utilização de técnicas específicas, estando, portanto, fora do escopo do presente trabalho.

Em 1993, BELOUHRANI *et al.* [42] apresentaram um método de separação de fontes para misturas instantâneas para o caso sobredeterminado, que aborda o problema de separação cega de fontes explorando as possíveis coerências temporais das fontes. A abordagem proposta neste trabalho, denominada SOBI (*Second-Order Blind Identification*), se baseia apenas em estatísticas de segunda ordem, sendo a solução obtida através da diagonalização conjunta de matrizes de correlação para diferentes defasagens temporais (*lags*) τ . A modelagem empregada no artigo pres-

supõe que as misturas foram corrompidas por ruído branco decorrelacionado com as fontes. Como consequência, a matriz de correlação das misturas é função da matriz de mistura \underline{A} , da matriz de correlação das fontes $\underline{S}(\tau)$ e da potência do ruído aditivo σ^2 , conforme a seguinte relação $\underline{R}(\tau) = E(x(t + \tau)x^*(t)) = \underline{A}\underline{S}(\tau)\underline{A}^H + \delta(\tau)\sigma\underline{I}$. A técnica estima as fontes $\hat{s}(t)$ estimando a matriz de branqueamento $\underline{\hat{W}}$ e a matriz unitária $\underline{\hat{U}}$ conforme a seguinte relação $\hat{s}(t) = \underline{\hat{U}}^H \underline{\hat{W}}x(t)$ ². Para se calcular a matriz de branqueamento $\underline{\hat{W}}$ é necessário primeiro decompor a matriz de covariância $\underline{\hat{R}}(0)$ em suas matrizes de autovalores e autovetores segundo a seguinte expressão $\underline{\hat{R}}(0) = \underline{L}\underline{\Delta}\underline{L}^H$, onde \underline{L} representa a matriz de autovetores formada pelos autovetores ($\underline{L} = [l_1, \dots, l_m]$) e $\underline{\Delta}$ representa a matriz de autovalores ($\underline{\Delta} = \text{diag}(\lambda_1, \dots, \lambda_m)$). Após se determinar as n componentes principais (o número de fontes presentes no cenário) através dos n maiores autovalores de $\underline{\Delta}$, é estimado a potência do ruído $\hat{\sigma}$ pela média dos $m - n$ menores autovalores da matriz $\underline{\Delta}$. Então a matriz $\underline{\hat{W}}$ é calculada pela expressão $\underline{\hat{W}} = \underline{\Delta}'^{-\frac{1}{2}}\underline{L}'^H$, onde $\underline{\Delta}' = \text{diag}(\lambda_1 - \hat{\sigma}, \dots, \lambda_n - \hat{\sigma})$ e $\underline{L}' = [l_1, \dots, l_n]$. Por fim, a matriz unitária $\underline{\hat{U}}$ é estimada através da diagonalização conjunta aproximada.

Em 2000, YEREDOR [43] propôs um método de separação para misturas de fontes gaussianas, no qual a solução é obtida, desde que as fontes tenham diversidade espectral [42]. Um método iterativo foi proposto para derivar os pesos ótimos a serem aplicados no processo de diagonalização das matrizes de correlação, sendo por esse motivo denominado algoritmo SOBI ponderado, ou WASoBI (*weights-adjusted SOBI*). A principal vantagem desta proposta é que a matriz de transformação $\underline{\hat{U}}$, antes unitária, estimada pelo algoritmo SOBI, pode ser melhor estimada quando a diagonalização conjunta é transformada em um problema de mínimos quadrados ponderado não linear (*non-linear WLS*), cuja solução é possível sob o pressuposto de que as fontes são gaussianas com correlações de comprimento finito, como em processos de média móvel (MA, *moving average*).

Também em 2000, JOURJINE *et al.* [44] apresentaram um método de separação cega de fontes, que emprega somente duas misturas e pode ser empregado para um número qualquer de fontes. A principal característica deste trabalho é a sua validade para o caso subdeterminado. O método foi batizado pelos autores como DUET (*Degenerate Unmixing Estimation Technique*), sendo aplicável quando as fontes são disjuntas WDO.

Em 2001, BOFILL e ZIBULEVSKY [29] abordaram a separação de sinais sonoros no caso subdeterminado, sendo descrita uma técnica para estimar a matriz de mistura no domínio tempo-frequência (T-F), a partir de uma função potencial elaborada para formar um histograma dos ângulos de chegada estimados dos sinais. Posteriormente, é sugerido que, para reconstrução das fontes, seja utilizado o método de decomposição do caminho mais curto, através do princípio da decomposição es-

²O índice H representa o transposto conjugado.

estatística esparsa (SSDP, *Statistically Sparse Decomposition Principle*), no lugar de somente minimizar a norma l_1 . Após a passagem das misturas para o domínio T-F (através de uma STFT com janela *Hanning* e superposição), os valores complexos obtidos da razão das amostras de cada mistura em relação a uma mistura de referência são agrupados (clusterizados) por meio de histogramas, com o objetivo de encontrar os coeficientes da matriz de mistura. Portanto, é necessário pressupor que haja um bom grau de esparsidade entre as fontes (preferencialmente disjuntas) no plano T-F, para que os agrupamentos possam, adequadamente, representar as colunas da matriz de mistura. Várias simulações foram apresentadas para o caso subdeterminado com sinais de instrumentos musicais.

Também em 2001, ABRARD *et al.* [45] apresentaram uma proposta para separação de misturas instantâneas nos casos determinado e subdeterminado, que pode ser aplicada a fontes que não necessariamente satisfazem as condições de não-gaussianidade, estacionariedade e independência estatística, usualmente empregadas no métodos de separação. O método, projetado para fontes não-estacionárias, se aplica a sinais gaussianos desde que possuam representações distintas no domínio T-F. A abordagem utilizada se baseia em duas suposições: a transformada T-F de cada fonte deve ser diferente nas adjacências temporais de cada janela, e existem algumas janelas onde somente uma fonte está presente. A ideia principal deste método é usar a informação no domínio T-F para cancelar uma das fontes presentes nas observações obtidas pelos sensores. Para isso, o método encontra uma combinação linear dos sinais de duas misturas de forma a obter um terceiro sinal com o cancelamento de uma das fontes. Os coeficientes da combinação linear correspondem às razões dos elementos de cada coluna da matriz de separação.

Em 2003, LUO e CHAMBERS [46] apresentaram uma solução para o problema de determinação automática do número de fontes ativas e separação no caso subdeterminado. Nesta proposta, também é usada uma abordagem tempo-frequência para discriminar nas misturas a estrutura de cada fonte. Para isso, é utilizada a representação discreta de Cohen [47], combinada com uma função *kernel* previamente selecionada, para passar as observações para o domínio T-F. O número de fontes ativas é obtido através de uma técnica de agrupamento, baseada em estatísticas estimadas durante um intervalo de observação. A técnica proposta foi comparada a uma implementação clássica de SCA (*Sparse Component Analysis*), tendo obtido bons resultados tanto na separação das fontes quanto na determinação do seu número. No entanto, a sua implementação apresenta um elevado custo computacional. Também em 2003, LUO e CHAMBERS [48] apresentaram uma nova solução baseada em aprendizagem competitiva, cuja principal vantagem é poder trabalhar com uma maior quantidade de dados, uma vez que apresenta um custo computacional reduzido. Esse método foi denominado *Self-Splitting Competitive Learning*

(SSCL). O problema desta última abordagem, observado em [49], é que uma sinapse [50] pode representar mais de um *cluster* se o número de sinapses, ao final, for menor que o número real de agrupamentos, e, como resultado, nenhum dos grupos pode estar corretamente identificado.

Em 2004, LUO *et al.* [51] apresentaram uma solução para o problema de separação cega de fontes, para o caso subdeterminado, na qual é relaxada a condição de esparsidade das fontes, uma vez que tal característica não pode ser garantida na maioria das aplicações práticas. Para isso é feita a introdução de uma técnica de agrupamento baseada na auto-divisão dos agrupamentos a partir da aprendizagem competitiva (SSCL). Duas hipóteses usuais são assumidas: independência estatística das fontes, e número de sensores (m) ao menos igual ao número de fontes ativas em cada bloco. Essa restrição é mais relaxada do que a de que as fontes sejam disjuntas no plano T-F (WDO). Métodos para separar um número arbitrário de fontes a partir de duas misturas e que empregam a restrição WDO foram propostos em [44] e em [52].

Em 2005, FLEGO *et al.* [53] propuseram um método para separar sinais de voz de misturas anecoicas e reverberantes no caso subdeterminado. Nos métodos clássicos de separação de fontes no caso subdeterminado, são usadas as características de esparsidade dos sinais de voz combinadas com a aplicação de máscaras binárias no domínio T-F para recuperar os sinais. Essas máscaras binárias provocam cortes abruptos quando estabelecem que somente uma das fontes é diferente de zero. Portanto, os sinais recuperados podem conter alto nível de ruído musical. Para atenuar esse efeito, é proposta uma abordagem combinada de máscaras contínuas com um método de pós-processamento. A partir da estimativa da frequência fundamental (f_0), filtros do tipo *comb* adaptativos são ajustados e utilizados para melhorar o desempenho tanto da separação quanto da qualidade do som. São usados dois filtros: o primeiro, do tipo FIR, é responsável pelo aprimoramento dos harmônicos do sinal de voz ativo; o segundo, do tipo IIR, suprime a interferência causada pelos demais sinais presentes na mistura. Os resultados apresentados mostraram que o método proposto reduz o ruído musical tanto em condições anecoicas quanto em reverberantes.

Em 2007, ZHANG *et al.* [54] apresentaram uma nova abordagem, em dois estágios, para resolver o problema de separação cega no caso subdeterminado. No primeiro estágio, tanto a matriz de mistura quanto o número de fontes são estimados por um método de clusterização a partir de uma função ponderada definida no trabalho como *weighted potential function* (WPF), construída com uma janela laplaciana [29]. No segundo estágio, a fim de superar a desvantagem da solução que minimiza a norma l_1 , uma nova representação esparsa, denominada SSCA (*Statistical Sparse Component Analysis*) e baseada em estatísticas de alta ordem no

domínio da transformada, é usada para recuperar as fontes. As hipóteses utilizadas nesta abordagem são: o número de misturas (m) é menor que o de fontes (n), as fontes são estatisticamente independentes e esparsas em certo grau no domínio da transformada, a matriz de mistura (\underline{A}) é de posto linha completo (igual a m), e o ruído é aditivo e independente das fontes. A técnica de recuperação das fontes considera que existem no máximo m fontes ativas para cada amostra das misturas no domínio da transformada e que as fontes são estatisticamente independentes. O artigo ao final faz uma comparação entre a norma l_1 , o SSDP e o método proposto, sendo que este apresenta um desempenho ligeiramente superior quando comparadas as SNRs resultantes.

Em 2009, REJU *et al.* [19] apresentaram um método de estimação da matriz de mistura para o caso instantâneo, desenvolvido a partir de um algoritmo simples de detecção dos pontos SSPs (*Single Source Points*) no plano T-F das misturas, nos quais há contribuição de uma única fonte. Os SSPs são obtidos comparando-se as fases dos coeficientes das transformadas de Fourier das misturas e agrupado-as usando um algoritmo de clusterização hierárquica. Para o caso determinado 2×2 , o desempenho médio, em relação à SIR, do algoritmo proposto, foi muito superior aos de algoritmos clássicos encontrados no ICALAB³ [21]. No caso subdeterminado, o método foi comparado ao algoritmo apresentado em [55], que é uma extensão dos algoritmos TIFROM (*Time Frequency Ratio of Mixtures*) [17, 35] e DUET, apresentando resultados superiores em todos os casos testados.

Também em 2009, TICHAVSKY *et al.* [56] propuseram um novo algoritmo para o caso determinado, denominado BARBI (*Block AutoRegressive Blind Identification*), o qual combina dois princípios: não-estacionariedade [57] e diversidade espectral [58–60]. O algoritmo BARBI é eficiente na separação de misturas instantâneas de processos AR (*AutoRegressive model*) estacionários em blocos e Gaussianos. O algoritmo *Block WASoBI* [61] é uma generalização do algoritmo WASoBI para um número arbitrário de blocos, sendo computacionalmente mais simples, pois não requer o cálculo das matrizes de peso usadas no WASoBI. Em vez disso, ele utiliza a estimativa ótima [62]. O algoritmo se mostrou robusto e com bom desempenho computacional.

Em 2010, LI *et al.* [63] apresentaram um algoritmo robusto de clusterização no espaço K -dimensional para o caso subdeterminado. O artigo considera a presença de ruído e considera as fontes insuficientemente esparsas, o que dificulta a obtenção de uma solução satisfatória. Os hiperplanos (representando o espaço dos sensores, $m - 1$) de separação no espaço K -dimensional são encontrados e usados para estimar a matriz de mistura. Em seguida, os sinais das fontes são recuperados através do método proposto em [64]. O conceito de suficiência de esparsidade empregado, que

³Toolbox de separação de fontes para o MATLAB disponível na internet[21].

preconiza que somente uma fonte está ativa na maioria das amostras é dispensado, contudo, o método restringe o sinal de cada mistura em cada instante de tempo a ter contribuição significativa de no máximo $m - 1$ fontes, sendo m o número de misturas. Uma função objetivo com restrições é definida, sendo a solução ótima obtida pelo método *Steepest Descent*. O problema de otimização possui um parâmetro de entrada arbitrário C , que representa o número de hiperplanos, o qual é inicializado com um valor maior que o número de fontes presentes e reduzido após a aplicação de um algoritmo de clusterização (por exemplo, o *K-means*) [65], que estabelece o número de agrupamentos como sendo o número de fontes. A recuperação do sinal é alcançada com simples inversão de matrizes, uma vez que o sistema se torna determinado, após a submatriz quadrada ser identificada. Os resultados de simulações mostram que o algoritmo apresenta bom desempenho mesmo quando as fontes são insuficientemente esparsas.

Em 2011, RAFII e PARDO [32] apresentaram um método para o caso anecoico que utiliza uma técnica de estimação baseada no algoritmo DUET. A principal ideia do trabalho consiste em aprimorar a discriminação dos picos nos histogramas de tempo-frequência através do uso de representações baseadas na “transformada” constante Q (*Constant Q transform*, CQT [66]), no lugar da transformada de Fourier. A CQT tem como característica uma resolução de frequência logarítmica combinada com espaçamento geométrico das notas na escala musical ocidental, apresentando uma discriminação superior em baixas frequências quando comparada à transformada de Fourier. Devido ao fato de a CQT não possuir uma transformação inversa, ela é empregada somente na fase de análise, ou seja, na estimação dos coeficientes da matriz de mistura. A sua versão janelada, denominada STQQT (*Short Time constant Q Transform*) e apresentada em [67], foi utilizada. Na fase de síntese, ou de reconstrução dos sinais, é utilizada a STFT, uma vez que essa transformação é inversível. O artigo propõe o uso de filtros de Wiener [68] para melhorar a reconstrução das fontes a partir de um método apresentado em [69]. Para testes realizados empregando misturas de sinais de instrumentos musicais, o algoritmo proposto apresentou um desempenho superior quando comparado ao DUET clássico, sendo a diferença entre os desempenhos acentuada com o aumento do número de fontes.

Em 2011, AYLLON *et al.* [70] propuseram um novo algoritmo para a separação de sinais de voz em misturas anecoicas, o qual corresponde a uma versão modificada do método de clusterização MS (*Mean Shift*) [71]. Um método automático de identificação do número de fontes, que combina técnicas de agrupamento com o mascaramento no plano T-F, é descrito. Este algoritmo pressupõe que as fontes sejam WDO, sendo esta hipótese normalmente satisfeita para sinais de voz uma vez que a energia em cada raia no plano T-F de uma das fontes predomina sobre a das outras. Assim, pode-se identificar cada raia tempo-frequência de uma fonte e isolar

a sua informação a partir de uma mistura através da aplicação de uma máscara. A técnica de agrupamento MS é não-paramétrica e estima os modos da densidade de probabilidade desconhecida no espaço de características multidimensional. O método é baseado na janela Parzen [72] e é executado independentemente da forma e do número de agrupamentos. O espaço de características empregado no trabalho é bidimensional e composto pelas estimativas da amplitude e da fase obtidos a partir das funções ILD (*Interaural Level Difference*) e ITD (*Interaural Time Difference*). Os resultados apresentados mostraram uma pequena melhora, quando comparados aos obtidos pelo algoritmo DUET.

Em 2012, PUIGT *et al.* [73], propuseram um novo método para estimar mapeamentos para o problema de separação de fontes não lineares. Foram empregados vários métodos, usando a mesma estrutura, mas com critérios distintos, para estimar as não-linearidades do problema de separação de fontes não lineares. Diferentemente dos métodos do estado-da-arte, a abordagem proposta usou uma suposição de esparsidade fraca nas fontes (regiões do sinal onde somente uma fonte está ativa): para isso, são procuradas pequenas regiões temporais, onde apenas uma fonte está ativa (SSZ). O método foi indicado, pelos autores, para sinais não estacionários, tais como sinais de voz. O trabalho representa uma generalização de não-linearidades nas misturas de um trabalho anterior dos mesmos autores. Foram investigadas várias medidas de confiança para determinar a unicidade da fonte ativa e, especificamente, algumas medidas inspiradas em técnicas de aprendizagem. Também foram estudados vários métodos de agrupamentos funcionais e validados os seus desempenhos com simulações. A abordagem empregada pode ser considerada uma extensão do caso linear instantâneo na análise de componentes esparsos para misturas não-lineares. Os experimentos demonstraram a eficácia e a relevância da abordagem proposta.

Em 2012, MA *et al.* [15] apresentaram um trabalho sobre sinais eletromagnéticos e receptores radar, onde foi mostrado que devido à complexidade do ambiente eletromagnético e a limitação dos receptores radar que, os sinais radares, para o caso subdeterminado, representam uma tarefa de difícil solução e de alto custo computacional. Nesse trabalho, foi analisada a aplicabilidade da técnica de análise de componentes esparsos (SCA) bem como a esparsidade dos sinais de modulação intrapulso dos radares. Foi apresentado um método de classificação cego subdeterminado com base no SCA para sinais radar no domínio do tempo. Os experimentos dos sinais de modulação intrapulso de radares mostraram, comparada a dificuldade de outros métodos de classificação tradicionais, a validade da abordagem em alguns casos de difíceis soluções. Foi empregada para a reconstrução das fontes, a partir da matriz de mistura estimada “A”, para o caso subdeterminado, a abordagem usual para a SCA, ou seja, a solução que minimiza a norma l_1 . A análise teórica da esparsidade da modulação intrapulso de sinais radares juntamente com os experimentos mostraram

que a SCA pode ser aplicada à triagem cega (processo pelo qual se estabelece quais sinais serão analisados) . Embora alguns ruídos ainda permanecessem presentes, o método mencionado se mostrou válido.

Em 2013, SHAH e PAPADIAS [12] apresentaram um estudo sobre sinais cardiorespiratórios, ou seja, sons cardíacos e respiratórios também chamados de ausculta. Tais sinais fornecem informações úteis sobre o comportamento do coração e do pulmão. O problema desses sons, cardíacos e respiratórios, é que eles interferem uns com os outros, bem como com outros sons, como ronco, fala ou ruído do tráfego, o que compromete a eficácia da ausculta. Esses sons cardiorrespiratórios possuem um alto nível de esparsidade podendo ser reconstruídos por técnicas cegas. O artigo apresentado aborda o problema da ausculta em ambientes auditivos complexos, inspirado no modelo de detecção de coincidências, o que é obtido pela localização do som por meio da estimação da diferença de nível interaural (ILD) (estimativa de amplitude) e pela diferença de tempo interaural (ITD) (estimativa de atraso). O método proposto, explora a esparsidade dos sons cardíacos e respiratórios e faz uso da técnica DUET, utilizando apenas duas observações para recuperar um número maior de fontes. A abordagem utiliza a análise no domínio tempo-frequência (DUET) para produzir um histograma bidimensional de estimativas de atenuação e atraso, onde os picos do histograma indicam as fontes em uma mistura. A máscara é calculada usando parâmetros da mistura e empregada para recuperar as fontes originais. É mostrado que existem máscaras tempo-frequência com resultados satisfatórios para sons cardíacos e respiratórios. O desempenho do método proposto é demonstrado por meio de uma série de experimentos com dados reais, exibindo uma melhor recuperação das fontes do que empregando outras técnicas. A abordagem DUET com STFT não apresenta bons desempenhos quando os parâmetros de mistura são muito baixos. O uso da CQT fornece uma melhor representação esparsa, apresentando alta resolução no domínio da frequência em baixas frequências. Essa característica em baixa frequência se adapta bem aos sinais cardiorrespiratórios. Usando a CQT com a abordagem DUET, a recuperação dos sinais sonoros são conseguidos com uma alta cardio SNR. O método proposto melhora a ausculta em ambientes auditivos; até quatro fontes são recuperadas com apenas dois sensores, uma capacidade que, normalmente não é possível com outros métodos existentes.

Em 2014, ZHANG *et al.* [74] apresentaram um algoritmo (processamento em blocos) adaptativo para um ambiente dinâmico, para o caso subdeterminado. Diferentemente do DUET, que se baseia na estimativa de parâmetros da mistura a partir de pontos no domínio tempo-frequência, o método proposto agrupa os parâmetros das estimativas da mistura em cada quadro a partir da potência total combinada com um algoritmo de clusterização FCM (Fuzzy C-Means) [75].

Para acompanhar os parâmetros da mistura variável, os coeficientes da matriz

de mistura estimados no quadro anterior são usados para inicializar FCM no quadro atual (em vez de uma inicialização aleatória como normalmente se dá no caso da FCM convencional [76]). Assim, um algoritmo adaptativo subdeterminado com a matriz de mistura variável no tempo foi realizado. Foram estudados os efeitos da variação contínua em vez do processo de mistura subdeterminada, no desempenho do algoritmo de processamento em blocos. A ideia principal foi a decomposição das misturas em curtos intervalos de tempo enquanto a matriz de mistura foi considerada constante durante um intervalo, mas diferente entre intervalos. Assim, um algoritmo *offline* pode ser usado para fontes separadas em um curto intervalo de tempo. No experimento, foram combinados quatro intervalos em um quadro para obter amostras suficientes para o algoritmo de agrupamento FCM, sendo os parâmetros do quadro atual inicializados como os parâmetros estimados do quadro anterior, considerando-se que há algumas correlações entre intervalos adjacentes. A combinação do processamento em lote por blocos e a melhoria empregada no agrupamento dos parâmetros, contidos no algoritmo proposto, realizaram uma separação cega de fontes subdeterminada e dinâmica eficaz.

Em 2015, BOBIN *et al.* [77] trataram o caso de fontes parcialmente correlacionadas, uma vez que, normalmente, as abordagens padrão geralmente fazem a suposição de que as fontes são estatisticamente independentes, ou pelo menos não correlacionadas. No entanto, nem sempre isso é verdade, e muitas vezes as fontes apresentam características parcialmente correlacionadas. Em contraste com os métodos padrão BSS esparsos, o ingrediente chave empregado neste trabalho para a separação das fontes esparsas e parcialmente correlacionadas (*sparse and partially correlated*, s.p.c.) foram estimativas de amostras discriminativas⁴, favorecendo o processo de separação. Para esse efeito, o algoritmo proposto AMCA (*Adaptive Morphological Component Analysis*) se baseou em um sistema de ponderação que iterativamente avalia o poder de discriminação dos dados da amostra com base na distribuição esparsa das fontes estimadas. Portanto, o algoritmo se aproveita de um esquema de ponderação adaptativa para favorecer as amostras (mais discriminativas) com base em seu nível de correlação. Vários cenários experimentais utilizando simulações de Monte-Carlo foram efetuados, e mostraram que o método proposto é robusto para a correlação parcial das fontes, situações que as técnicas padrão BSS costumam falhar. O algoritmo de AMCA obteve um desempenho satisfatório na maioria dos experimentos realizados com fontes parcialmente correlacionadas. O mesmo desempenho foi observado quando as fontes empregadas se encontravam altamente correlacionadas (isto é, quando cerca de 80% das amostras ativas são comuns a todas as fontes). Contudo, se mostra competitivo quando o número de fontes a ser recuperado é grande. O desempenho do algoritmo proposto foi ilustrado

⁴São as amostras das fontes que contém informação mais relevante para o processo de separação.

com simulações dentro do contexto da astrofísica por microondas. O algoritmo AMCA se mostrou adequado para estimar os componentes físicos, que costumam, por natureza, ser parcialmente correlacionados. Os resultados numéricos obtidos demonstraram a robustez da AMCA à correlação parcial das fontes.

2.5 Algoritmos Clássicos com Esparsidade

2.5.1 Algoritmo SCA

A técnica SCA proposta em [28, 29] para misturas instantâneas foi o primeiro método que utilizou a hipótese de esparsidade entre as fontes para estimar a matriz de misturas, sendo até hoje um dos mais competitivos. Ela também permite estimar o número n de fontes.

Este algoritmo SCA segue os três passos descritos no início da Seção 2.2. Inicialmente, é aplicada a transformada STFT ou wavelet aos sinais das misturas $x_i(t)$, gerando as representações \tilde{X}_i no domínio da transformada, e visando esparsificar essas representações. Em seguida, obtém-se o vetor de características, calculando-se a razão da parte real de cada sinal dos sensores em relação a um dos sinais das misturas (sinal de referência). O filtro de norma é aplicado aos vetores obtidos. Então, recorre-se a um algoritmo de clusterização ou de detecção de picos.

No método de clusterização, as razões das partes reais ou imaginárias das amostras de cada mistura no domínio da transformada, em relação a uma mistura de referência, são obtidas para formar os vetores características. Em seguida, um algoritmo de clusterização é aplicado a esses vetores e n centroides são obtidos, sendo cada centroide definido por um vetor de dimensão $m - 1$. A matriz de mistura, de dimensão $m \times n$, é formada justapondo-se os n vetores dos centroides e acrescentando-se uma primeira linha formada por coeficientes unitários (devido ao fator de escala das fontes originais não ser relevante para o problema de separação).

Alternativamente ao processo de clusterização, foi proposto em [29, 31] o uso de um método de detecção de picos para estimar os coeficientes da matriz de mistura. Esse estimador consiste na formação de um histograma para a busca dos ângulos do gráfico $\tilde{X}_2 \times \tilde{X}_1$ associados às maiores densidades de pontos. Para tal, é feita uma varredura angular, sendo escolhidos ângulos-teste (θ_t) uniformemente espaçados no intervalo $[0, \pi]$. O histograma é formado usando a função potencial local, definida como

$$\Phi(\theta_t, \theta_n, \tilde{\underline{X}}) = \begin{cases} \lambda |\tilde{\underline{X}}| (1 - \frac{\Delta}{\pi/4}), & \text{para } |\Delta| < \frac{\pi}{4}, \\ 0, & \text{para } |\Delta| \geq \frac{\pi}{4} \end{cases} \quad (2.27)$$

onde $\Delta = \theta_t - \theta_n$, $\theta_n = \arctan \frac{\tilde{X}_2}{\tilde{X}_1}$ e λ é um parâmetro de ajuste da abertura angular, representando a resolução das contribuições das vizinhanças. A função

Φ é nula se θ_n não se encontrar na vizinhança de θ_t ($|\Delta| \geq \frac{\pi}{4}$), sendo atribuídos pesos maiores para as medidas θ_n dentro da vizinhança que apresentam maiores módulos $|\tilde{X}|$. Esse critério de peso em função dos valores dos sinais das misturas funciona como um filtro de norma, desprezando as amostras que possuem módulos pequenos, uma vez que essas são mais suscetíveis a serem contaminadas por ruídos ou interferências de outras fontes. Os picos do histograma ocorrem nos ângulos cujas tangentes formam as estimativas dos coeficientes de cada linha da matriz de mistura. Quanto mais esparsas as fontes se apresentarem nas misturas no domínio da transformada, melhores serão as estimativas obtidas; da mesma forma, quanto mais afastados forem os valores dos coeficientes, mais fácil será a distinção de seus picos.

Para exemplificar os métodos descritos nessa seção, consideramos o cenário sem ruído contendo 3 fontes e 2 misturas. As fontes foram sintetizadas por sequências de notas de instrumentos da classe palhetas. Os sinais originalmente foram gravados a taxa de 44,1 kHz, e foram reamostrados para a taxa de 8 kHz, no intuito de reduzir o custo computacional.

Na Fig. 2.1 são apresentados os sinais das fontes e na Fig. 2.2 os sinais das misturas instantâneas formadas pela matriz de mistura \underline{A} , de dimensão 2×3 , com coeficientes unitários na primeira linha e com a segunda linha dada por $[0, 3710 - 1, 3573 - 0, 3898]$.

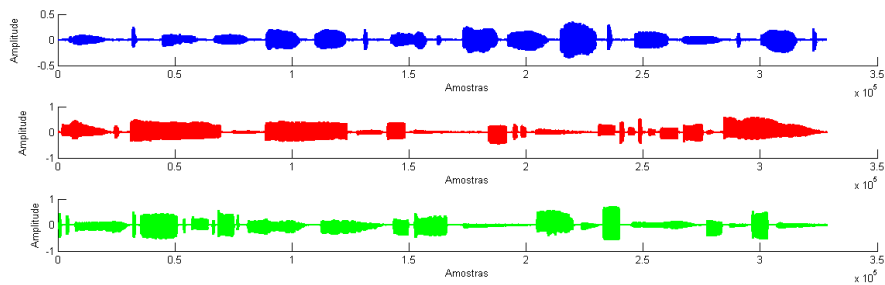


Figura 2.1: Sequência de Notas de Palhetas

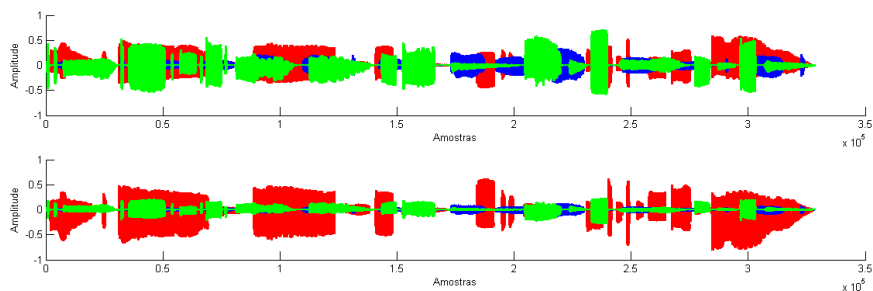


Figura 2.2: Misturas Instantâneas da Sequência de Notas de Palhetas.

O algoritmo SCA com clusterização pelo método *K-means* [65] foi implementado, usando-se como transformada esparsificadora uma STFT de 1024 pontos com janela *Hanning* e com um salto de 256 amostras. O vetor de características empregado foi formado pela razão entre as partes reais das amostras das misturas no domínio da T-F, eliminando-se os coeficientes de módulos menores que 0,001 e maiores que 3. Os coeficientes obtidos foram $[0,4260 \quad -1,3340 \quad -0,4384]$, ou seja, todos próximos dos valores corretos. Este algoritmo, por empregar um método de clusterização, apresenta os problemas dessas técnicas, ou seja, agrupamentos com centroides próximos tendem a formar um único agrupamento com valor médio posicionado no baricentro dos dois grupos, enquanto misturas com poucos componentes esparsos tendem a formar falsos centroides.

Uma segunda implementação da técnica SCA foi realizada, utilizando-se, no lugar da técnica de clusterização, a abordagem por detecção de pico. A Fig. 2.3 apresenta o histograma com 1000 *bins*, formado a partir das mesmas amostras usadas no processo de clusterização *K-means*. Os valores obtidos para os coeficientes

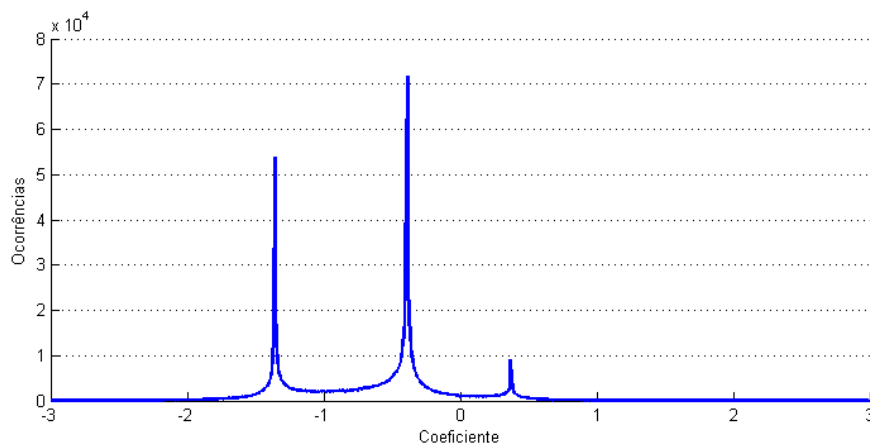


Figura 2.3: Histograma de Razão das Partes Reais das Misturas no Domínio da Transformada.

com o algoritmo de detecção de pico foram $[0,3688 \quad -1,3590 \quad -0,3871]$, ou seja, bem próximos dos valores corretos. A melhoria na estimativa em relação ao método de clusterização é perceptível; no entanto, com o aumento do número de sensores, as colunas associadas as fontes passarão a ter mais coeficientes. Assim, a técnica de detecção de picos no histograma unidimensional (que resolve o problema para matrizes de misturas com duas linhas, vetor de característica unidimensional) deixará de ser válida, necessitando que ocorra um aumento na dimensão do histograma (vetor de características bidimensional) para que a técnica de detecção continue válida, tornando o método mais complexo, embora continue válido.

Pode-se também empregar os ângulos obtidos da razão dos sinais da mistura no domínio da transformada para formar os histogramas. Nesta implementação, usu-

se um filtro de norma, eliminando todas as amostras com normas inferiores a 0,01, e o algoritmo de clusterização *K-means*. As estimativas dos coeficientes obtidas com este método foram $[0,4860 - 1,4788 - 0,4083]$, valores próximos aos obtidos pela mesma técnica usando as razões das misturas.

Na Fig. 2.4 são apresentados os histogramas angulares formados pela função da Eq. (2.27) (em azul) e por uma função histograma convencional (em vermelho), sendo o número de *bins* igual a 1000, como no exemplo anterior. Nessa figura podemos observar que existem 3 picos em cada histograma (cada pico associado à atenuação de uma das fontes a um dado sensor), e que, praticamente, não há diferença nas suas posições. Contudo, a função de Zibulevski tem uma forma mais suave, com picos menos evidenciados. Os coeficientes estimados pela função de

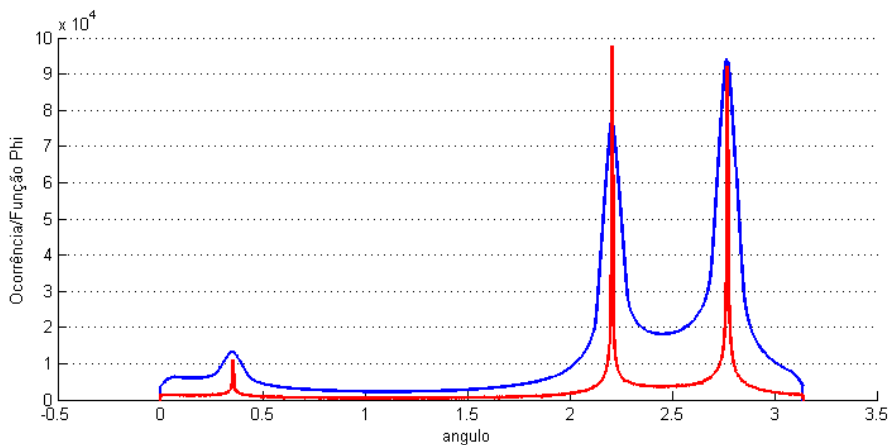


Figura 2.4: Histogramas Angulares Convencional (em vermelho) e Obtido pela Função de Zibulevsky (em azul).

Zibulevski foram $[0,3707 - 1,3583 - 0,3887]$, ao passo que os coeficientes estimados pelo histograma simples foram $[0,3725 - 1,3538 - 0,3905]$, apresentando erros quadráticos médios similares. Aqui podemos notar que o uso de histogramas pode ser um caminho válido para se estimar o número de fontes presentes num dado cenário.

Nos testes efetuados, os métodos de detecção de picos apresentaram resultados melhores do que a técnica de clusterização *K-means*.

2.5.2 Algoritmo DUET

O algoritmo DUET (*Degenerate Unmixing Estimation Technique*) [44] foi concebido especificamente para o caso anecoico. Trata-se de uma generalização do algoritmo SCA para esse cenário. Contudo, possui algumas restrições inerentes às equações formadoras do vetor de características. O algoritmo utiliza um histograma bidimensional de atenuação e atraso para determinar os coeficientes que estabelecem a

mistura anecoica, sendo as estimativas das atenuações α e dos atrasos δ obtidas por

$$\left(\hat{\alpha}, \hat{\delta}\right) = \left(|R(\tau, \kappa)|, -\frac{\text{Imag}(\log(R(\tau, \kappa)))}{\kappa}\right) \quad (2.28)$$

onde $R(\tau, \kappa)$ corresponde à razão das representações das misturas no domínio T-F.

Para exemplificar o comportamento desse algoritmo, usaram-se as mesmas fontes dos testes com o algoritmo SCA, formadas por sinais de instrumentos da classe palhetas. Contudo, devido à restrição da utilização dessa técnica apenas para atenuações positivas e à necessidade de se incluírem atrasos nos sinais, fez-se necessário gerar novas matrizes de atenuação e atraso. A Fig. 2.5 ilustra o histograma bidimensional utilizado na técnica do DUET, para um cenário com 3 fontes e 2 misturas, geradas com os seguintes pares de atenuação e atraso: $(0, 70, 0, 50)$, $(1, 50, -1, 50)$, $(0, 40, 1, 50)$, que devem ser estimados na fase de análise do algoritmo.

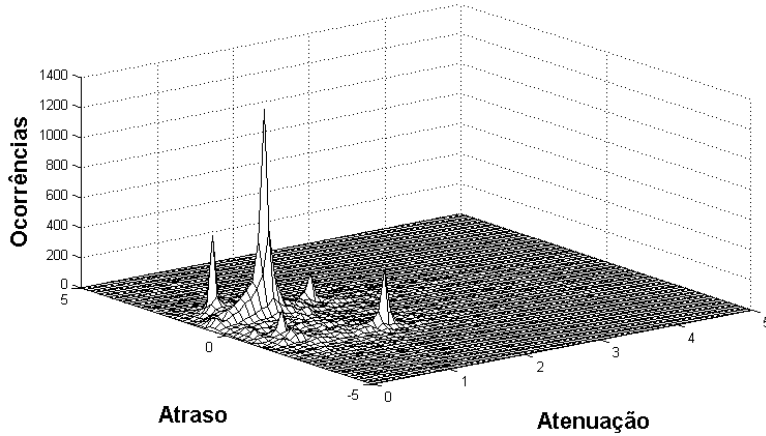


Figura 2.5: Histograma Bidimensional de Atenuação e Atraso para 3 Fontes e 2 Misturas.

A implementação do algoritmo DUET disponibilizada em [23] não realiza, de forma automática, a busca dos picos no histograma bidimensional. Portanto, a fim de suprir essa ausência e reduzir o custo computacional de uma busca de picos num espaço bidimensional, a implementação empregada neste trabalho (batizada de DUETm) sofreu algumas adaptações, sendo o algoritmo resultante usado como referência para as técnicas de detecção dos coeficientes através de histogramas. Para evitar a busca no espaço bidimensional, foram empregados dois histogramas unidimensionais, um de atraso e outro de atenuação, obtendo-se um número de picos superior ao número de fontes presentes no cenário, para cada histograma. Em seguida, encontram-se todas as combinações de pares com o valor de atenuação e atraso obtidos dos picos dos histogramas unidimensionais, formando um conjunto de candidatos dos pares verdadeiros. A determinação dos picos verdadeiros pode

ser feita através de um algoritmo de classificação, como o vizinho mais próximo ($1 - NN$), que associe cada par (atenuação,atraso), obtido pela Eq. (2.28), a um dos pares candidatos, desde que os mesmos estejam dentro de um raio de vizinhança pré-estabelecido, evitando interferências provenientes de outras fontes. Ao final, os pares candidatos mais votados são selecionados como pares verdadeiros.

A Fig. 2.6 ilustra o histograma combinado de atenuações e atrasos utilizado nesta técnica. Neste exemplo usou-se uma janela *Hanning* de 1024 amostras com um salto de 512 amostras. Foram usadas 50 discretizações para os valores de atenuação e 50 para os de atraso, gerando uma grade equivalente de 2500 *bins* para o histograma bidimensional. Buscou-se um candidato a mais na atenuação e também um candidato a mais no atraso, em relação ao número de fontes, totalizando 16 candidatos a serem investigados pelo algoritmo de classificação. Além disso, o algoritmo de busca usou um raio de exclusão de 0,1 (resolução), ou seja, todos os picos que caírem dentro desse raio serão considerados como sendo um mesmo pico. Pode-se

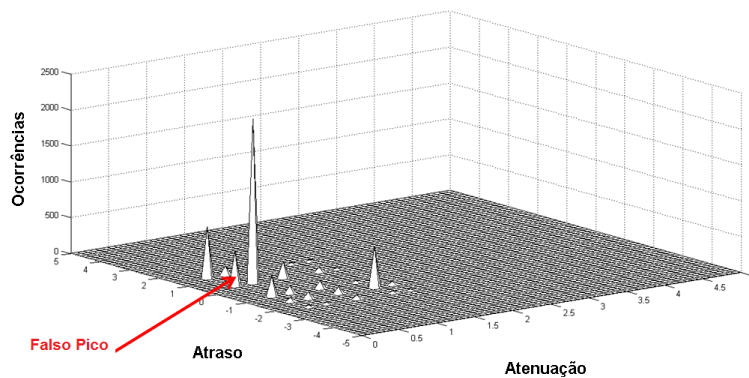


Figura 2.6: Histograma Combinado de Atenuação e Atraso para 3 Fontes e 2 Misturas.

observar na Fig. 2.6 que os 3 picos com maiores ocorrências correspondem aos pares (atenuação,atraso) iguais a $(0, 65, 0, 56)$, $(1, 45, -1, 52)$ e $(0, 45, 1, 52)$, aproximadamente iguais às localizações dos picos verdadeiros do histograma bidimensional. A desvantagem deste procedimento é a possibilidade de ocorrerem falsos pares candidatos próximos aos pares verdadeiros ou do histograma bidimensional apresentar picos poucos proeminentes, o que pode conduzir a estimativas erradas dos picos nos histogramas unidimensionais. Sua principal restrição é a necessidade de que as fontes sejam razoavelmente WDO (*W-Disjoint Orthogonality*) no domínio da transformada para que os picos fiquem separados, isto é, quando as fontes no domínio da transformada (FFT, *Fast Fourier Transform* [78], ou STFT, *Short-Time Fourier Transform* [79], etc) para quaisquer duas misturas são conjuntos disjuntos (não ocorrem simultaneamente).

A etapa de reconstrução do algoritmo, responsável pela obtenção das estimativas

das fontes no domínio da frequência, será abordada no Cap. 5.

2.5.3 Algoritmo TIFROM

O algoritmo TIFROM (*Time-Frequency Ratio Of Mixtures*) foi originalmente proposto em [17] para misturas instantâneas e subdeterminadas. Nele foi proposto o uso de filtro de variância mínima, ou seja, é feita uma busca por regiões, contendo um número predeterminado de amostras, que possua pouca variação na razão entre as misturas, no domínio da transformada. Essa busca é feita a partir de um par de misturas, assumindo que as regiões encontradas mantêm as mesmas características (baixa variância) nos demais pares. Essa abordagem evita uma clusterização no espaço de dimensão $m - 1$. Para cada região encontrada, o coeficiente de atenuação é estimado calculando a média da razão das partes reais das misturas. O processo é repetido até se obter tantos coeficientes distintos quanto o número de fontes. A principal restrição dessa técnica é assumir que existem regiões na representação das misturas no domínio T-F em que somente uma das fontes está ativa, sendo necessário que isso ocorra com todas as fontes presentes no cenário, para que se possam estimar corretamente os coeficientes de todas elas.

Para ilustrar o desempenho dessa técnica, usaremos o mesmo exemplo numérico descrito para o algoritmo SCA. A implementação do algoritmo TIFROM [36] para misturas instantâneas utilizada, obtida de [22], emprega a transformada STFT com 128 amostras e salto de 32 amostras, com busca de regiões de baixa variância contendo 10 STFT adjacentes. Podemos observar na Fig. 2.2 que mesmo no domínio temporal, há regiões com predominância de uma das fontes. No domínio T-F o algoritmo encontrou as regiões de baixa variância nas raias 3, 10 e 12, representadas pelas retas destacadas em vermelho na Fig. 2.7. Os coeficientes estimados pelo

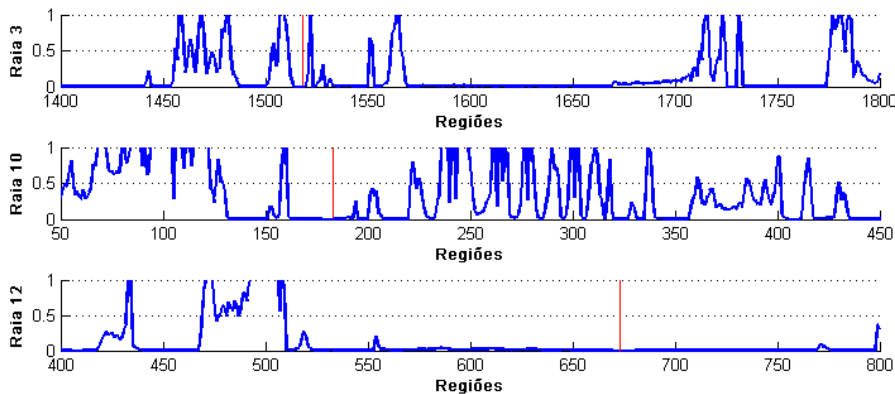


Figura 2.7: Variância da Razão das Partes Reais das Misturas nas Raias 3, 10 e 12.

TIFROM foram $[0,3710 - 1,3573 - 0,3898]$, exatamente iguais aos coeficientes

do sistema de mistura, demonstrando a eficiência da abordagem em misturas que contêm regiões com apenas uma das fontes ativa.

Há duas versões distintas do algoritmo TIFROM para o caso anecoico [24]. Na versão denominada AD-TIFROM-CF, a busca das regiões de baixa variância ocorre no domínio da frequência, ao passo que na versão AD-TIFROM-CT, a busca das constantes ocorre no domínio do tempo. Uma vez que as implementações do TIFROM para o caso anecoico não foram disponibilizadas pelos autores, foi feita, nesta tese, uma adaptação do algoritmo TIFROM instantâneo para esse caso, sendo denominada de TIFROM_Anec. Nessa implementação, usou-se o módulo da razão das misturas no domínio da transformada (primeiro componente da Eq. (2.28)) para determinar as regiões de baixa variância. Em seguida, o valor do atraso é calculado, usando o segundo componente da Eq. (2.28), sobre essa mesma região.

Para avaliar o desempenho do algoritmo TIFROM_Anec, usaremos os mesmos sinais empregados nos experimentos com o DUET, mantendo os pares de coeficientes de atenuação e atraso iguais a $(0, 70, 0, 50)$, $(1, 50, -1, 50)$ e $(0, 40, 1, 50)$. A implementação TIFROM_Anec obteve os pares estimados $(0, 7008, 0, 4972)$, $(1, 5008, -1, 4816)$ e $(0, 3999, 1, 4574)$, os quais estão bem próximos das atenuações e atrasos reais. Pode-se concluir, desses experimentos, que o algoritmo TIFROM apresenta bom desempenho tanto para misturas instantâneas, quanto para misturas anecoicas, quando as misturas apresentam regiões com uma única fonte, chamadas de *Single Source Zones* (SSZ) [24, 36].

Portanto, as técnicas descritas pelo método esparso descrevem basicamente duas abordagens: uma empregando um processo de clusterização e/ou detecção de picos; e outra empregando uma busca em regiões de baixa variância. O principal denominador comum em ambas as técnicas é que elas só serão efetivas caso o domínio aplicado seja esparso. Portanto, os picos só serão representativos a um par fonte-sensor, em uma mistura, se o número de ocorrências em uma dada discretização, em que a referida fonte aparece unicamente ativa, for maior que o número de ocorrências nas demais discretizações em que a mesma fonte aparece combinada com outras fontes quaisquer presentes no cenário. Em igual medida, as regiões de baixa variância só poderão ser empregadas para estimar os coeficientes da matriz de mistura se e somente se ocorrerem, ao menos, n regiões com fontes unicamente ativas e distintas. Essa condição obriga que existam n regiões com $n - 1$ fontes inativas em cada uma delas. Portanto, a efetividade dessas técnicas só ocorrerá se o sinal for realmente esparso. Logo, o domínio onde a técnica é aplicada é fundamental, de forma a atender as condições descritas anteriormente. Infelizmente, as transformadas esparificadoras nem sempre garantem, de antemão, que tais condições sejam satisfeitas independentemente do sinal. Portanto, transformações matemáticas e procedimentos que visem esparsificar o sinal representam caminhos úteis para essas técnicas.

Capítulo 3

Métodos de Estimação da Matriz de Mistura para o Caso Instantâneo

Neste capítulo apresentaremos três novos algoritmos para a estimação da matriz de misturas para o cenário de misturas instantâneas determinadas e subdeterminadas. Conforme descrito no Cap. 1, nas misturas instantâneas os sinais provenientes de uma dada fonte chegam nos diversos receptores simultaneamente ou, na prática, com diferenças de tempos de chegada desprezíveis. Assim, pode-se considerar que as misturas ocorrem sem atrasos nos sinais. Essa forma de mistura é comum quando os receptores se encontram próximos das fontes e em ambientes onde não há anteparos que possam propiciar reflexões ou reverberações dos sinais e sem ruído. A relação entre os sinais das fontes e dos receptores, considerando misturas lineares, é dada por

$$\begin{bmatrix} x_1(t) \\ \vdots \\ x_m(t) \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ & \vdots & \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} s_1(t) \\ \vdots \\ s_n(t) \end{bmatrix} \quad (3.1)$$

onde m é o número de sensores, n é o número de fontes presentes no cenário, $x_j(t)$ é a mistura percebida pelo j -ésimo sensor, $s_k(t)$ é a k -ésima fonte, e a_{jk} representa o coeficiente de atenuação da fonte k na mistura percebida pelo j -ésimo sensor.

No caso determinado, no qual o número de fontes n é igual ao número de sensores m , normalmente a reconstrução é feita invertendo-se a matriz de mistura estimada. No entanto, outras formas de reconstrução serão discutidas nessa tese, considerando soluções mais gerais obtidas para o caso subdeterminado, como por exemplo, o método baseado nas avaliações das estimativas candidatas pelo caminho mínimo, proposto por Zibulevsky [29, 31]. No caso subdeterminado, o número de fontes n

é superior ao número de sensores m , sendo as matrizes de misturas, de dimensão $m \times n$, portanto, não inversíveis. Já os métodos de estimação da matriz de mistura elaborados nesse trabalho obtêm os coeficientes das misturas sensor por sensor, e podem, portanto, ser aplicados sem modificações nos cenários determinados e subdeterminados.

Este capítulo está dividido em quatro seções: a primeira aborda os métodos propostos, a segunda avalia os resultados desses algoritmos sem a presença de ruído por meio do MSE das estimativas da matriz de mistura, a terceira avalia os mesmos resultados da seção anterior, com a presença do ruído no cenário, e a quarta avalia algumas características dos algoritmos propostos por meio das medidas SDR, SIR e SAR. Nessa última seção, considerou-se o caso determinado, sendo as estimativas das fontes (necessárias para se calcularem as medidas acima) obtidas aplicando-se a inversa da matriz estimada ao vetor de misturas. Já a utilização do MSE, como medida para avaliar o desempenho dos algoritmos de estimação dos coeficientes de mistura não requer a reconstrução das fontes e pode ser aplicada ao caso subdeterminado. Deve-se observar que as medidas SDR, SIR e SAR sofrem influência das fontes empregadas, sendo, portanto, dependentes das posições em que ocorrem os maiores erros nos coeficientes da matriz estimada. Assim, diferentes matrizes estimadas para um mesmo cenário que apresentam o mesmo MSE podem, dependendo das fontes empregadas, apresentar valores distintos de SDR, SIR e SAR, mesmo que se mantenha o método de reconstrução.

3.1 Algoritmos Propostos

Nesta tese foram elaborados inicialmente dois algoritmos distintos, para cenários com misturas instantâneas, denominados SCAM e SCAz, e uma variação do algoritmo clássico SCA, denominado SCAC. Além disso, nas Seções 3.2 a 3.5, onde são apresentados os resultados de simulações, foi utilizado um quarto algoritmo, denominado STUE (*Sparsifying Transform Unmixing Estimation*), elaborado originalmente para o caso anecoico; por esse motivo, sua descrição mais detalhada se encontra no Cap. 4. Conforme descrito no Cap. 2, existem diversos algoritmos propostos na literatura para resolver o problema de separação cega de fontes para o caso instantâneo, sendo alguns deles usados como referência para avaliar o desempenho dos algoritmos propostos nesta seção.

Ao se estudar os algoritmos que usam esparsidade como princípio (independentemente do tipo de mistura), verificou-se que a estimação dos coeficientes da matriz de mistura pode ser feita de diversas formas: clusterização cega (p.ex., *K-means*) [80, 81]; mascaramento [23]; e clusterização hierárquica [19], dentre outras. Pode-se também empregar técnicas de detecção de picos comuns ou mais elaboradas,

como por meio de construções de funções histograma (p.ex., Varredura “Zibulevsky” [29, 31]). Todas essas metodologias, quando aplicadas na técnica SCA [28, 29], pressupõem a formação de agrupamentos ou a detecção de picos em espaços vetoriais de dimensão $m - 1$. De forma diversa, e coerentemente com um dos objetivos desta tese, que é o de obter métodos menos complexos, o algoritmo TIFROM faz uma busca num espaço unidimensional; no entanto, esse algoritmo pressupõe a existência de regiões onde somente uma única fonte está presente e que as amostras chegam simultaneamente nos sensores, sendo tais assunções nem sempre necessárias nos algoritmos que apresentamos a seguir.

3.1.1 Algoritmo SCAM

No método SCAM (SCA modificado), originalmente apresentado em [33], a determinação dos coeficientes da matriz de mistura é feita através de técnicas de clusterização utilizando-se pares de misturas formados a partir de uma mistura de referência (previamente escolhida). No algoritmo SCA convencional [28, 29], todas as misturas são usadas simultaneamente.

A ideia do algoritmo proposto é desacoplar as misturas, estimando os coeficientes de cada linha da matriz de mistura independentemente. Tal abordagem, permite a generalização do método para o caso anecoico, uma vez que os instantes das amostras, dos sinais das fontes, usados para estimar os coeficientes de uma dada linha da matriz de mistura (associada à observação de um dos sensores) não necessariamente serão os mesmos instantes a serem empregados para estimar os coeficientes de outra linha da matriz de mistura (associada a outro sensor). Além disso, o desacoplamento permite que cada par de mistura possa sofrer um tratamento diferenciado, para melhor se estimar os coeficientes da matriz de mistura.

Um problema que surge com essa abordagem é a obtenção das colunas da matriz de mistura, uma vez que os coeficientes de uma dada linha podem ser estimados numa ordem diferente em relação aos coeficientes de outra linha. Portanto, ao final do processo de estimação dos coeficientes obteremos uma matriz (\underline{R}) com elementos possivelmente posicionados de modo equivocado. Como no caso instantâneo e determinado, a reconstrução das fontes normalmente é feita a partir da inversa da matriz de mistura; assim, é importante que os coeficientes estejam posicionados corretamente, tornando necessário utilizar um procedimento para posicionar os coeficientes de forma casada nas diferentes linhas da matriz de mistura.

Neste trabalho, serão investigadas três possíveis funções objetivo (\mathcal{F}_{obj}) a serem minimizadas a fim de obter a posição correta dos coeficientes estimados pelo método de clusterização. As funções empregadas e os princípios em que se baseiam são os seguintes:

1. \mathcal{F}_{CEF} : Correlação entre as Estimativas das Fontes;
2. \mathcal{F}_{KEF} : Curtose das Estimativas das Fontes;
3. \mathcal{F}_{CFR} : Correlação das Fontes Reconstruídas.

Na primeira abordagem, válida para o caso determinado, dado o conjunto de todas as possíveis matrizes de misturas candidatas ($\hat{\mathbf{W}}^{-1} = \hat{\mathbf{H}} = \{\hat{H}_1, \dots, \hat{H}_C\}$) formadas pela permutação dos elementos de cada linha da matriz \underline{R} , determina-se a matriz ótima ($\hat{H}_{ot} \in \hat{\mathbf{H}}$) que minimiza uma das funções objetivo \mathcal{F}_{CEF} ou \mathcal{F}_{KEF} . Podemos normalizar a matriz de mistura de forma que todos os elementos da primeira linha sejam iguais a 1, uma vez que o fator de escala original das fontes não é importante na reconstrução. As posições dos coeficientes da segunda linha determinam apenas em qual ordem as fontes serão reconstruídas e, conseqüentemente, podem ser arbitrárias. Portanto, o número total de matrizes candidatas é igual a $C = (m - 2) \times n!$. Como, nessa abordagem, busca-se a solução ótima dentro de um número finito de matrizes candidatas, trata-se de um problema de otimização inteira. É possível construir um conjunto de matrizes candidatas com mais que C possibilidades, bastando, para isso, que se detecte um número maior de picos para cada vetor de característica associado a um par de mistura. Tal procedimento torna a solução mais robusta, mas apresenta um custo computacional maior.

Na segunda abordagem, a função correlação das estimativas das fontes (\mathcal{F}_{CEF}) é definida por

$$\mathcal{F}_{CEF}(p) = \left(\sum_{i=1}^n \sum_{j=1}^n |corr[\hat{s}_{i,p}, \hat{s}_{j,p}]| \right) - n \quad (3.2)$$

onde $p \in \mathcal{C} = \{1, \dots, C\}$, $\hat{s}_{i,p}$ é a estimativa da i -ésima fonte da matriz candidata p , e $corr[x, y]$ é o coeficiente de correlação de Pearson dado por

$$corr[x, y] = \frac{E[xy] - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (3.3)$$

onde $\mu_x = E[x]$ e $\sigma_x = \sqrt{E[x^2] - \mu_x^2}$. Para cada matriz candidata são obtidas as estimativas das fontes ($\hat{\underline{s}}_p = \hat{H}_p^{-1} \underline{x}$) e calculado o valor da função $\mathcal{F}_{CEF}(p)$. Em seguida, é obtido o índice p da solução que estabelece a matriz candidata correta (\hat{H}_{ot}), dado por

$$ot = \min_{p \in \mathcal{C}} \mathcal{F}_{CEF}(p) \quad (3.4)$$

A hipótese considerada neste procedimento é que diferentes fontes possuem baixa correlação. Um posicionamento incorreto dos coeficientes resultará em estimativas das fontes contendo trechos de outras fontes, o que, conseqüentemente, aumentará as correlações entre as estimativas das fontes em relação às correlações entre os sinais originais.

A função curtose das estimativas das fontes \mathcal{F}_{KEF} (3.5) é definida como

$$\mathcal{F}_{KEF}(p) = \sum_{i=1}^n \|\text{kur}(\hat{s}_{i,p})\| \quad (3.5)$$

onde o coeficiente referente à curtose $\text{kur}(x)$ é dado por

$$\text{kur}(x) = \frac{E[(x - \mu_x)^4]}{\sigma_x^4} \quad (3.6)$$

A hipótese assumida neste caso, é que as fontes são supergaussianas, e o processo de otimização corresponde à maximização da função acima, ou seja

$$ot = \max_{p \in \mathcal{C}} \mathcal{F}_{KEF}(p) \quad (3.7)$$

Portanto, ao se maximizar a função curtose das estimativas das fontes, encontra-se o índice da matriz que gera, na média, as estimativas mais supergaussianas. Tal hipótese é verdadeira para os sinais de voz e sequências de notas descritas no Cap. 2. No entanto, para alguns sinais de áudio, como por exemplo os de flautas, esta condição pode não ser respeitada, não sendo a função objetivo \mathcal{F}_{KEF} indicada para esse cenário. Aqui, novamente, a matriz de mistura estimada $\hat{\underline{A}}$ é dada por $\hat{\underline{H}}_{ot}$. Cabe ressaltar que outras funções custo podem ser utilizadas em outros cenários. Por exemplo, quando as fontes são subgaussianas, pode-se definir o problema de otimização como a minimização da curtose das estimativas.

Na terceira abordagem, é utilizada uma técnica de reconstrução de fontes para o caso subdeterminado, tal como o mascaramento *HARD*, representada pelo operador \mathcal{F}_r [23]. Assim, obtêm-se estimativas das fontes a partir do conjunto de matrizes $\mathbf{I} = \{\underline{I}_0, \dots, \underline{I}_{m-2}\}$, de dimensão $2 \times n$, formadas pela primeira linha e uma das demais linhas da matriz \underline{R} . A matriz \underline{I}_0 é formada pelas duas primeiras linhas da matriz \underline{R} , sendo usada para obter as estimativas de referência, dadas por

$$\hat{\underline{s}}_0 = \mathcal{F}_r(\underline{I}_0, n) \quad (3.8)$$

Para cada uma das matrizes \underline{I}_p , com $p \in \mathcal{B} = \{1, \dots, m-2\}$, obtêm-se novas estimativas das fontes, através da operação

$$\hat{\underline{s}}_p = \mathcal{F}_r(\underline{I}_p, n) \quad (3.9)$$

Em seguida, calculam-se as correlações entre cada sinal obtido através da Eq. (3.9) e os sinais de referência obtidos pela Eq. (3.8), definindo a função Correlação

das Fontes Reconstruídas como

$$\mathcal{F}_{CFR}(i, p, k) = |E[\hat{s}_{0,i}\hat{s}_{p,k}]| \quad (3.10)$$

sendo $\hat{s}_{0,i}$ a i -ésima fonte de referência, $\hat{s}_{p,k}$ a k -ésima estimativa candidata à i -ésima fonte de referência para uma dada submatriz \underline{I}_p , e $i, k \in \mathcal{S} = \{1, \dots, n\}$, determina-se a matriz de mistura com os coeficientes posicionados corretamente $\hat{\underline{A}}$ através da otimização inteira

$$ot(i, p) = \max_{k \in \mathcal{S}} \mathcal{F}_{CFR}(i, p, k) \quad (3.11)$$

Assim, os elementos da primeira linha de $\hat{\underline{A}}$ serão todos iguais a 1, os da segunda linha serão iguais aos da segunda linha de \underline{R} , e os das demais linhas serão $\hat{\underline{A}}(p+2, i) = \underline{R}(2, ot(i, p))$.

Esse processo de busca da matriz de mistura é um diferencial do algoritmo SCA proposto em relação aos demais algoritmos que usam a técnica SCA, não possuindo uma distinção clara entre a fase de análise (estimação da matriz de mistura) e a de reconstrução das fontes. Assim, o método desenvolvido recorre a dois princípios: esparsidade das fontes (sendo que esta propriedade pode ser obtida através de uma transformação esparsificadora, como a STFT ou a wavelet) e baixa correlação ou não-gaussianidade (necessárias para reposicionar os coeficientes de mistura estimados para cada sensor independentemente) por meio da otimização de uma função objetivo \mathcal{F}_{obj} . Na Fig. 3.1 apresentamos o diagrama de blocos das etapas do método.

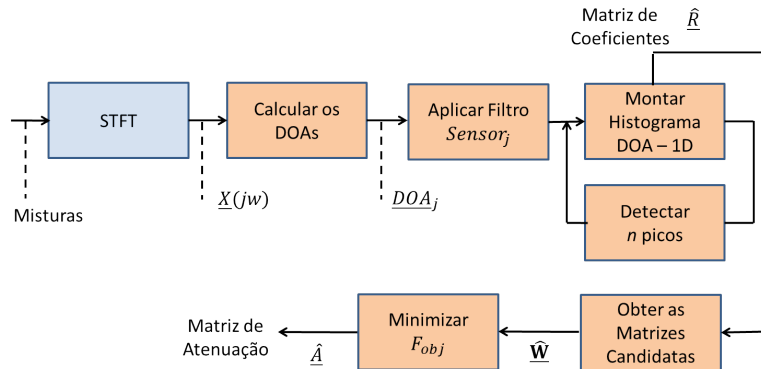


Figura 3.1: Esquema Geral do Algoritmo SCAM.

Verificou-se ser possível empregar o método no domínio temporal (sem apelo ao primeiro passo), desde que se faça um branqueamento prévio das misturas, de modo que a estimação dos coeficientes ocorra neste novo espaço. Após a estimação dos coeficientes, é possível voltar ao espaço original, a fim de obter uma estimativa da matriz de mistura. Essa abordagem será exemplificada na seção de simulações.

A principal vantagem de desacoplar as observações dos diferentes sensores é que

podem ser aplicados “filtros” para selecionar as melhores amostras de cada sensor a serem utilizadas na determinação de cada coeficiente da matriz de mistura. Isso resulta numa estimação linha a linha dessa matriz, permitindo-se que as estimativas dos coeficientes se tornem mais robustas, uma vez que perturbações específicas em um dos sensores podem ser filtradas, sem que essas amostras sejam também eliminadas do sinal observado em outro sensor. Portanto, pode-se fazer uma escolha seletiva das amostras por par de sensores para se estimar os coeficientes de uma determinada linha independentemente da estimação dos coeficientes de outra linha, o que não é possível no método de clusterização direta. A desvantagem dessa abordagem é a sua complexidade computacional, visto que a determinação da equivalência entre as fontes correspondentes aos coeficientes de diferentes linhas se dá a partir de uma busca exaustiva entre todas as matrizes candidatas, ou por meio de correlação entre estimativas oriundas de diversas reconstruções.

Assim, o método proposto difere de outros métodos baseados na técnica SCA, que, normalmente, procuram a solução exclusivamente no espaço das matrizes de misturas e somente empregam o princípio da esparsidade, por meio de técnicas de clusterização ou de detecção de pico num espaço de dimensão $m - 1$, assumindo que as amostras chegam simultaneamente aos sensores. Também se diferencia de outros métodos, como a Análise de Componentes Independentes (ICA, do inglês *Independent Component Analysis*), que empregam independência estatística e algoritmos iterativos, como o *steepest descent*, para efetuar a busca da solução no espaço das matrizes de separação. O método SCAM proposto resolve o mesmo problema com $m - 1$ clusterizações no espaço unidimensional das matrizes de misturas, obtendo a solução por um processo de otimização inteira, que não requer análise de convergência.

3.1.2 Algoritmo SCAC

O algoritmo SCAC (SCA com classificador), elaborado, nesta tese, para o caso determinado, é uma variante do algoritmo SCA original [28, 29]. Além de empregar o conceito de esparsidade, explora o fato das misturas serem instantâneas, ou seja, de haver simultaneidade de sinais das fontes nos sensores. Nenhum outro princípio é empregado (tal como baixa correlação dos sinais das fontes) o que torna o método proposto vantajoso em algumas aplicações.

Nessa abordagem, cada representação das misturas no plano tempo-frequência é classificada em relação a uma fonte, a partir da clusterização dos vetores de características obtidos dos sinais de um par de sensores escolhido (e.g., microfones 1 e 2). A mesma classificação é utilizada para as amostras, no domínio tempo-frequência, dos sinais dos demais sensores, associando-as às mesmas fontes das amostras do par

de misturas escolhido como referência. Os vetores de características são formados para os outros pares de sensores (e.g., microfones 1 e 3), mantendo-se, para todos os pares, sempre um mesmo sensor como referência (e.g., microfone 1). A partir do *cluster* formado pelos vetores de características associados a cada fonte e a cada par de misturas, estimam-se os coeficientes correspondentes da matriz de misturas.

Uma abordagem semelhante de reconstrução da matriz de mistura, sem recorrer à clusterização no espaço de dimensão $m - 1$ e sem ter que usar uma função objetivo para reposicionar os coeficientes foi utilizada pelo algoritmo TIFROM. No entanto, no algoritmo TIFROM, este método é utilizado apenas nas regiões de baixa variância (sem classificar todas as amostras), o que requer a busca das regiões SSZ para estimar os coeficientes. Diferentemente do SCA clássico (descrito na Seção 2.2), que emprega uma clusterização no espaço de dimensão $m - 1$, e do SCAM, que emprega $m - 1$ clusterizações no espaço unidimensional, o algoritmo SCAC estima os coeficientes usando uma única clusterização no espaço unidimensional. Assim, esse algoritmo pode ser visto como uma combinação dos métodos SCAM, que estima os coeficientes a partir do DOA com detecção de pico, e do TIFROM, que utiliza o princípio da simultaneidade das amostras nos sensores para estimar os coeficientes da matriz de mistura, a partir das características oriundas de um par de misturas.

Uma vantagem dessa abordagem híbrida, em que se emprega uma classificação seguida de uma clusterização, é o uso de filtros de *Outlier* de forma similar à abordagem de dupla clusterização [19]. Naquela proposta, a primeira clusterização é feita por um mascaramento cego, apenas para determinar os principais agrupamentos, ao passo que, na segunda clusterização, ocorre a eliminação dos *outliers* de cada *cluster*, visando determinar com maior precisão a posição dos coeficientes.

Na nossa abordagem, diferentemente do usual, a clusterização não ocorre de forma cega, mas funciona como um mascaramento ou classificação, e é anterior à fase de análise, fazendo com que os agrupamentos sejam formados a partir de um critério de similaridade com os coeficientes previamente estimados, e não com as características de homogeneidade entre as amostras, normalmente empregadas num processo de clusterização cega. Por fim, os coeficientes da mistura são reestimados empregando somente as melhores amostras associadas a cada par fonte-sensor (agrupamento). Isso também representa um diferencial em relação aos algoritmos TIFROM e SCA, permitindo a inclusão de outros “filtros”, além do de *outlier*, para retirar de forma distinta as amostras indesejadas para os diferentes pares fonte-sensor. Na Fig. 3.2 apresentamos um diagrama de blocos com as etapas do método SCAC.

O algoritmo SCAC diferencia-se de outros métodos, como o ICA, por não empregar a independência estatística entre os sinais das fontes e por não utilizar algoritmos iterativos para efetuar a busca da solução no espaço das matrizes de separação. Esse

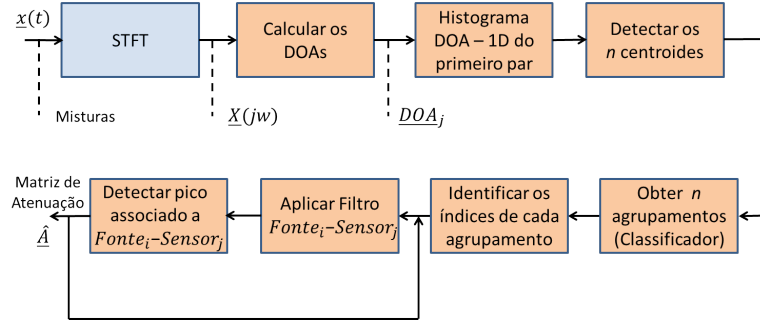


Figura 3.2: Esquema Geral do Algoritmo SCAc.

método emprega somente o princípio da esparsidade e a simultaneidade entre amostras em misturas instantâneas, resolvendo o problema de separação com uma única etapa de classificação/clusterização unidimensional no espaço das matrizes de misturas, sendo obtidos $n(m-1)$ agrupamentos associados aos coeficientes das matrizes de misturas. Portanto, esse método não requer análise de convergência.

Visando exemplificar a sua viabilidade em sinais que possuam alguma esparsidade temporal, apresentamos alguns exemplos, na Seção 3.4.1, nos quais o vetor de características é construído utilizando amostras das misturas no domínio do tempo. Tal abordagem é possível em cenários com misturas de sinais de voz e com poucas fontes presentes.

3.1.3 Algoritmo SCAz

O algoritmo SCAz (SCA modificado com SSZ) combina a abordagem de desacoplamento dos sensores, utilizada no algoritmo SCAM (baseado no SCA clássico [28, 29]), e a da estimação dos coeficientes a partir da determinação das regiões de baixa variância (SSZ), utilizada pelo TIFROM. Portanto, assim como no SCAM, nesse método é necessário o uso de uma das funções objetivo descritas na Seção 3.1.1 para reposicionar corretamente os coeficientes estimados por cada sensor. Na Fig. 3.3 apresentamos um diagrama de blocos com as etapas do método SCAz.

O algoritmo SCAz diferencia-se dos métodos baseados na técnica SCA por não empregar clusterização ou detecção de pico no espaço de dimensão $m-1$, e não utilizar algoritmos iterativos para efetuar a busca da solução no espaço das matrizes de separação. O método proposto utiliza dois princípios, o da esparsidade e um segundo, o da baixa correlação mútua das fontes, resolvendo o mesmo problema com $m-1$ buscas de n regiões SSZ no espaço unidimensional das matrizes de misturas, e obtendo a solução numa busca por exaustão num espaço finito das matrizes de separação candidatas. Portanto, não requer análise de convergência.

O SCAz também difere dos algoritmos TIFROM e SCAc, uma vez que, para estimar os coeficientes, é efetuado um desacoplamento dos pares de sensores, não

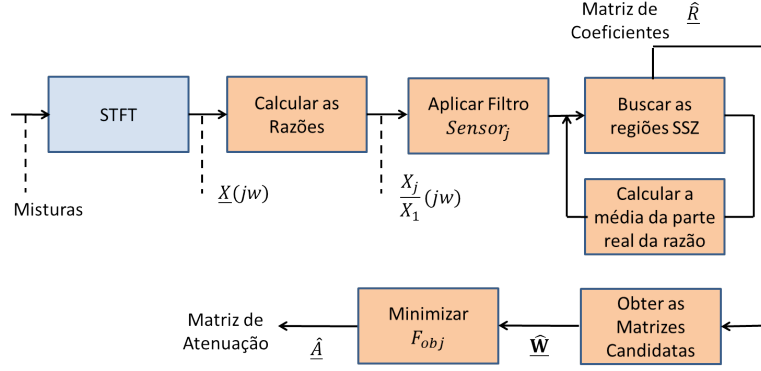


Figura 3.3: Esquema Geral do Algoritmo SCAz.

utilizando a hipótese de que as amostras chegam simultaneamente nos sensores. O emprego de uma função objetivo para reposicionar os coeficientes no algoritmo SCAz não ocorre nos algoritmos SCA, TIFROM e SCAC. Enquanto os algoritmos SCAM e SCAC utilizam vetores de características baseados em DOA, combinados com histogramas e detecção de picos para obter os coeficientes da matriz de mistura, o método proposto emprega a média das razões nas regiões SSZ para estimar esses mesmos coeficientes.

3.1.4 Algoritmo STUE

O algoritmo STUE (*Sparsifying Transform Unmixing Estimation*), para o caso instantâneo, é um caso particular da sua implementação para o caso anecoico, portanto, este método é mais detalhado no Cap. 4. Neste contexto, ele representa uma alteração sobre o algoritmo SCAz, porém, ele é baseado no uso da estimativa das fontes, visando obter estimativas mais precisas dos coeficientes da matriz de mistura.

Assim, igualmente ao TIFROM e ao SCAz, o algoritmo STUE emprega o princípio da esparsidade a partir de amostras ou de representações discriminantes, dependendo do domínio empregado. Contudo, diferentemente do TIFROM, o algoritmo STUE, da mesma forma que o algoritmo SCAz, aplica o princípio do desacoplamento dos sensores, empregando portanto as funções objetivo (\mathcal{F}_{CEF} , \mathcal{F}_{KEF} e \mathcal{F}_{CFR}) previamente discutidas neste capítulo, e usadas para reposicionar os coeficientes estimados para cada par de sensores da matriz de mistura. De forma diversa aos algoritmos TIFROM e SCAz, que procuram as regiões de baixa variância diretamente sobre as misturas para estimar seus coeficientes, o algoritmo STUE faz a busca das regiões SSZ nas estimativas das fontes. Para isso, o algoritmo STUE, depende de uma estimação prévia (grosseira) dos coeficientes da matriz de mistura, podendo ser obtidos por qualquer algoritmo de separação cega de fontes para o caso instantâneo (algoritmo preliminar), e que disponibilize os coeficientes da matriz de

mistura.

Uma vez estimado os coeficientes da matriz de mistura pelo algoritmo preliminar, se reconstrói duas estimativas da mesma fonte associadas a sensores distintos (por meio de uma técnica de reconstrução, podendo ser um mascaramento *HARD*), preservando os valores de atenuação associados aos sensores distintos empregados, e que foram obtidos pelo algoritmo preliminar.

Tendo se obtido o par de estimativas das fontes com diferentes fatores de atenuação, a partir do método de reconstrução escolhido, procura-se, sobre esse par de estimativas, as regiões de baixa variância, e se recalcula os coeficientes estimados pelo algoritmo preliminar, empregando por exemplo, a técnica prevista pelo método TIFROM.

3.2 Desempenho dos Algoritmos Propostos em Misturas sem Ruído

Nesta seção, apresentamos os resultados obtidos com os métodos propostos para misturas de sinais de voz e de áudio (descritas no Apêndice A) com três fontes e três sensores sem a presença de ruído ambiente ou erros de medições. Esses resultados foram comparados aos dos quatro algoritmos usados como referência, FastICA [82], SOBI [42, 83–85], JADE [86–88] e TIFROM [17, 22, 24, 35], cujas implementações foram disponibilizadas por seus autores. Os resultados das simulações são apresentados em duas formas distintas: a primeira, na forma de gráficos, contendo os erros quadráticos médios (em dB) dos diferentes algoritmos obtidos com cada uma das 30 misturas simuladas e a segunda, forma de tabela, contendo os valores mínimos, máximos, médios e desvio padrão dos erros quadráticos obtidos para cada algoritmo. No gráfico, pode-se averiguar o desempenho do algoritmo para cada cenário, caracterizado pela matriz de mistura e tipo de fonte, ao passo que, na tabela, pode-se avaliar a robustez do algoritmo para cada tipo de sinal de voz, através da variação do seu desempenho com a matriz de mistura.

A apresentação dos resultados está dividida conforme os cinco diferentes grupos de algoritmos. No primeiro grupo, estão os algoritmos usados como referências neste trabalho, chamados de Algoritmos Clássicos. No segundo grupo, estão os algoritmos SCAM e SCAC, considerando suas implementações no domínio do tempo com o pré-branqueamento dos sinais das misturas. O terceiro grupo é composto pelo algoritmo SCAM no domínio da frequência, empregando três tipos distintos de filtragens. O quarto contém o algoritmo SCAC no domínio da frequência, empregando as mesmas três filtragens utilizadas no conjunto anterior. Por fim, no quinto grupo, estão os algoritmos SCAZ e STUE, implementados no domínio da frequência.

3.2.1 Algoritmos Clássicos

As Figs. 3.4 e 3.5 mostram os erros quadráticos médios (em dB) das soluções obtidas para a estimativa da matriz de atenuação¹ com os algoritmos JADE, TIFROM, SOBI e FastICA, a fim de estabelecer a faixa de aceitabilidade a ser empregada nos gráficos contendo os algoritmos elaborados com novos princípios combinados com a esparsidade, introduzidos nesta tese. Dessa forma, as seguintes configurações foram empregadas respeitando o critério previamente estabelecido na Seção 1.2. O algoritmo TIFROM empregou os seguintes parâmetros: número de amostras na janela igual a 128; fator de superposição de 75%; número de STFTs adjacentes igual a 10. O algoritmo SOBI empregou número de matrizes correlação a serem diagonalizadas igual a 4. O FastICA empregou os seguintes parâmetros: não linearidade $g(u) = u^3$, número máximo de iterações igual a 1000, estimação sequencial das componentes e número de autovalores menor ou igual a m . Todos os algoritmos citados anteriormente foram exercitados no cenário com misturas instantâneas formadas por sinais de voz e áudio (sinais de flautas). Cada ponto do eixo x representa um caso distinto de matriz de atenuação simulado.

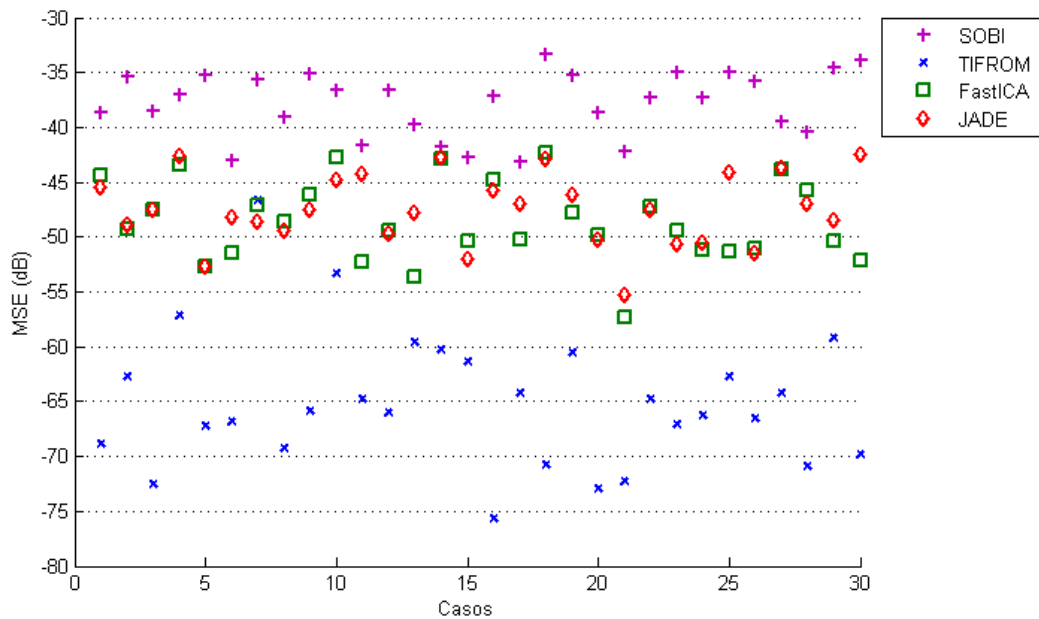


Figura 3.4: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Voz.

Nas figuras apresentadas nas seções seguintes, contendo os resultados obtidos com os algoritmos propostos, incluímos retas horizontais iguais aos erros médios quadráticos (em dB) correspondentes às piores e as melhores soluções obtidas pelos

¹Em todos os algoritmos que estimam a matriz de separação, obteve-se a matriz de mistura invertendo-se a matriz de separação.

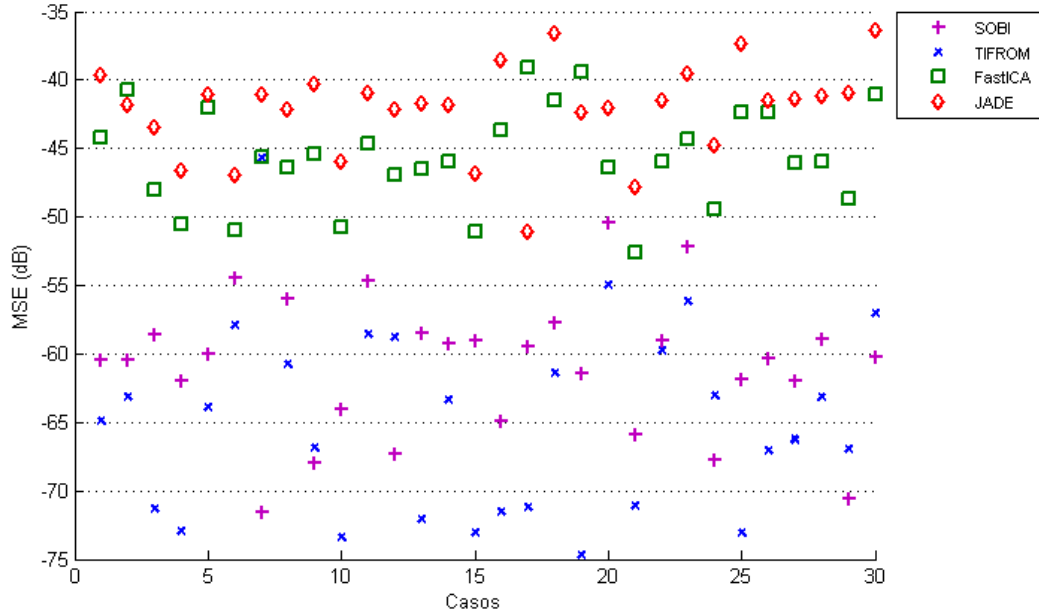


Figura 3.5: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Áudio.

Algoritmos Clássicos e os melhores resultados obtidos pelo algoritmo TIFROM (que também emprega o princípio da esparsidade). Esses valores passarão a representar os limiares para que uma solução seja considerada aceitável. Quando o MSE obtido com algum dos algoritmos propostos for superior ao limiar para o dado cenário, o resultado será objeto de uma análise mais detalhada. Assim sendo, o limiar (superior) aceitável para o erro quadrático médio para sinais de voz será $-33,32$ dB, obtido pelo algoritmo SOBI no seu pior caso, ao passo que esse limiar para sinais de áudio será $-36,39$ dB, obtido pelo algoritmo JADE também no seu pior caso (ver Fig. 3.4). No cenário de pior desempenho do algoritmo SOBI com sinais de voz, a SDR obtida foi $30,11$ dB para a fonte estimada correspondente ao maior MSE dentre as três, enquanto a pior estimativa de fonte obtida com o algoritmo JADE, usando sinais de áudio, obteve SDR de $34,89$ dB. Ambos os casos representam excelentes resultados, sendo difícil, sem uma boa amplificação dos sinais, diferenciar subjetivamente os sinais originais das suas respectivas estimativas. Portanto, apesar de serem os piores resultados, os mesmos apresentam um excelente desempenho, garantindo que resultados inferiores a esses limiares também apresentem excelentes desempenhos.

3.2.2 Algoritmos SCAM e SCAC no Domínio do Tempo

Os algoritmos propostos nesta seção, o número de *bins* usados na construção dos histogramas foi 36000. Todos os algoritmos propostos nesta seção utilizaram o filtro de norma e empregaram o fator $L = 0,09$. Esses parâmetros foram estabelecidos

após várias simulações empregando os sinais de voz e de áudio descritos no Apêndice A.

A Fig. 3.6 contém os resultados dos erros médios quadráticos (em dB) das matrizes de mistura obtidas com as implementações no domínio do tempo dos algoritmos SCAM e SCAC, e dos algoritmos SOBI e TIFROM, para sinais de voz. Os limiares de referência são representados pelas linhas horizontais tracejadas, correspondendo ao melhor MSE obtido com o algoritmo TIFROM (em vermelho) e ao pior MSE obtido com o algoritmo SOBI (em preto).

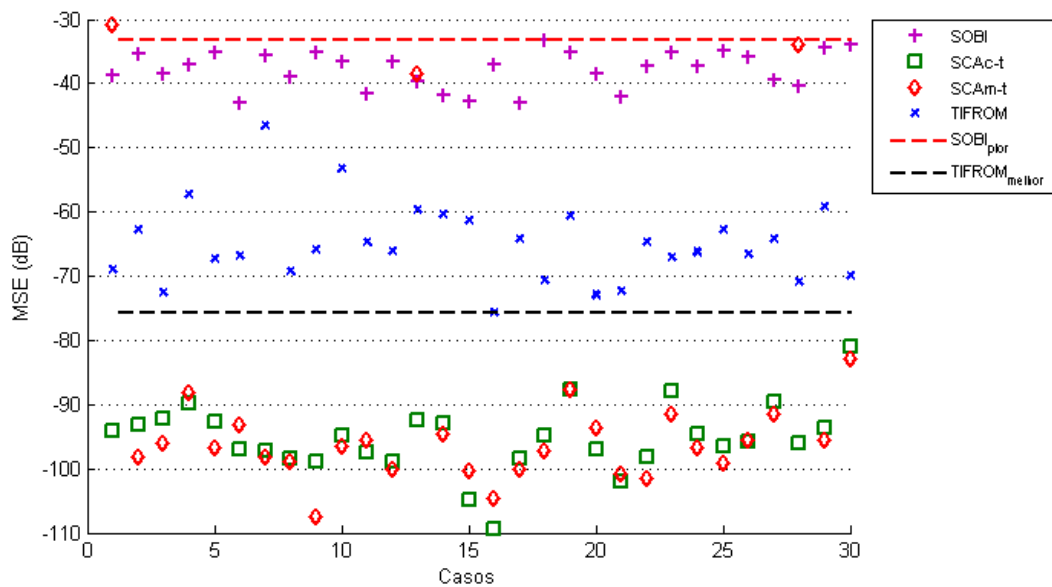


Figura 3.6: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Propostos SCAM-t e SCAC-t, e dos Clássicos SOBI e TIFROM - Sinal de Voz.

Na Fig. 3.6, pode-se observar que somente em um caso o resultado obtido foi ligeiramente pior que o limiar. Para esse cenário específico, o algoritmo SCAM obteve SDR igual a 19,05 dB para a fonte com pior estimaco e 25,42 dB para a fonte melhor estimada, sendo estes valores inferiores aos obtidos com o algoritmo SOBI. J o algoritmo SCAC teve todas as suas soluoes com valores bem superiores aos demais algoritmos clssicos, demonstrando a validade da abordagem.

As duas principais diferenas entre os algoritmos SCAM e SCAC so o conjunto de amostras empregado para a deteco dos picos utilizados para obter os coeficientes de atenuao e, a forma como os coeficientes estimados de cada linha da matriz de mistura so correlacionados para formar uma coluna que seja corretamente associada a um determinado sensor.

Comparando a matriz de mistura (no espaço branqueado) do Caso 1, dada por

$$\begin{bmatrix} 1 & 1 & 1 \\ 21,1389 & -0,0066 & -0,1185 \\ -1,1819 & 0,7373 & -1,3648 \end{bmatrix} \quad (3.12)$$

com a matriz estimada pelo algoritmo SCAM, dada por

$$\begin{bmatrix} 1 & 1 & 1 \\ -0,1185 & 21,1472 & -0,0066 \\ 0,7369 & 0,7372 & -1,3648 \end{bmatrix} \quad (3.13)$$

podemos observar que na terceira linha houve uma estimativa errada, ocasionada por um falso pico próximo ao pico verdadeiro. Tal ocorrência levou a um reposicionamento dos coeficientes incorretos pela função \mathcal{F}_{CEF} , causando o baixo desempenho da separação. Para contornar esse problema, é possível adotar duas abordagens distintas. A primeira é estimar um número de picos superior ao número de fontes, e avaliar a função \mathcal{F}_{CEF} para as matrizes candidatas, com número de linhas igual ao número de fontes, formadas pelas combinações de todas as linhas com coeficientes obtidos dos picos estimados. Assim, o resultado total de possibilidades possui um custo computacional combinatorial, uma vez que o número de matrizes candidatas será $(C_{n_p}^m)^{m-1}$, onde n_p é o número de picos estimados. A segunda solução é estabelecer um afastamento mínimo entre os picos estimados no espaço da transformada, eliminando-se picos próximos aos de valores maiores. Assim, a complexidade do algoritmo continua sendo combinatorial, porém não mais dependente do aumento do número de picos estimados, embora aumente-se a complexidade do algoritmo de busca de picos. Neste último caso a complexidade do algoritmo se mantém igual a $(n)^{m-1}$, menor do que a primeira abordagem.

Devido aos sinais de áudio empregados neste trabalho não apresentarem esparsidade temporal, espera-se que os desempenhos dos algoritmos propostos quando aplicados aos sinais no domínio do tempo não sejam satisfatórios, uma vez que o pré-branqueamento dos sinais não é suficiente para torná-los esparsos. Em outras palavras, o pré-branqueamento não evita que haja presença de mais de uma fonte na maioria das amostras das misturas. Assim, num cenário pouco esparsos, os picos formados pelos histogramas, aparentemente, não são mais representativos dos coeficientes da matriz de mistura (conforme se pode verificar na Fig. 3.26), mas apenas das combinações lineares dos coeficientes de misturas das diversas fontes num dado sensor.

Os resultados das simulações com esses sinais de áudio (com pouca esparsidade temporal) foram incluídos somente para mostrar a limitação dessa aborda-

gem (técnica temporal). Essa limitação também pode ocorrer no domínio tempo-frequência, mas com uma probabilidade menor, pois naturalmente com aumento de mais uma dimensão (frequência) os sinais ficam mais esparsos. Por esse motivo a maioria dos trabalhos na literatura aplica transformações tempo-frequência objetivando aumentar a esparsidade da mistura.

A Fig. 3.7 contém os resultados dos erros médios quadráticos (em dB) das matrizes de mistura obtidas com as implementações no domínio do tempo dos algoritmos SCAM e SCAC para sinais de áudio.

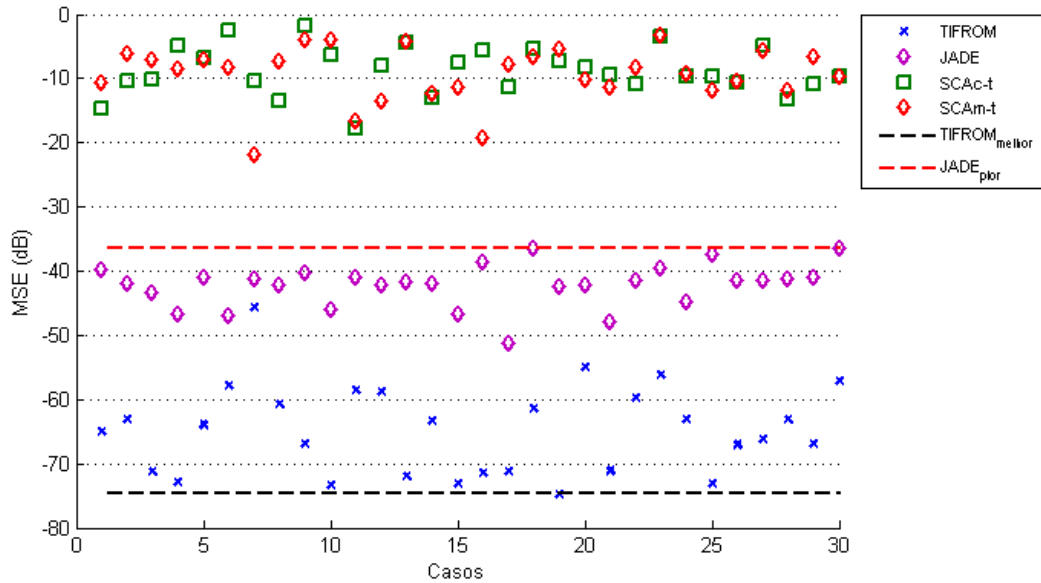


Figura 3.7: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Propostos SCAM-t e SCAC-t, e dos Clássicos TIFROM e JADE - Sinal de Áudio.

Conforme esperado, nenhum dos algoritmos propostos apresenta desempenho satisfatório para sinais de áudio no domínio do tempo. Todas as soluções ficaram acima do limiar estabelecido pelos métodos clássicos.

3.2.3 Algoritmo SCAM no Domínio Tempo-Frequência

Avaliaremos nesta seção o desempenho do algoritmo SCAM no domínio T-F (SCAMTF) com três tipos de filtragens: filtragem de norma (SCAMTF_1), filtragem SSP (SCAMTF_2), e filtragem de *Outlier* (SCAMTF_3).

Os algoritmos propostos nesta seção empregaram 360 *bins*, os que utilizaram o filtro de norma empregaram o fator $L = 0,09$ (SCAMTF_1), os que utilizaram o filtro SSP empregaram $c_{tol} = 0,92$ (SCAMTF_2), e os que empregaram o filtro de *outlier* (SCAMTF_3) consideraram amostras válidas, para determinação do pico, aquelas que ficaram até 2σ em relação à média do agrupamento. Esses parâmetros

foram estabelecidos por meio de simulações realizadas com os sinais descritos no Apêndice A.

A transformação para o domínio T-F foi realizada utilizando a STFT com janela *Hamming* de 1024 pontos e salto de 256 amostras. Os tamanhos da janela e do salto foram fixados conforme os valores presentes no algoritmo DUET [23], que também emprega histograma, com intuito de permitir comparações entre os métodos que empregam histogramas.

A Fig. 3.8 contém os MSE obtidos com as três versões do algoritmo SCAmTF e o limiar de referência (reta horizontal) obtido com o algoritmo SOBI para sinais de voz. Da Fig. 3.8, pode-se observar que, para todos os casos simulados, os algoritmos propostos apresentaram MSEs menores do que o limiar, sendo o melhor desempenho médio obtido com o filtro de *Outlier* (SCAmTF_3).

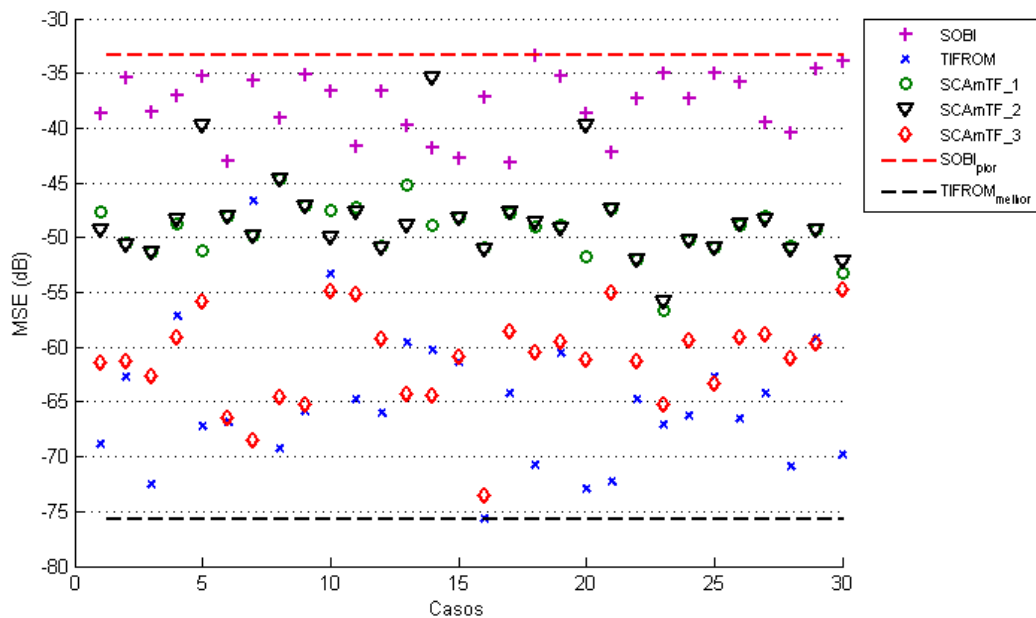


Figura 3.8: MSE dos Elementos das Matrizes de Mistura Estimadas pelo Algoritmo SCAmTF com 3 Tipos de Filtragem e pelos Algoritmos SOBI e TIFROM - Sinal de Voz.

A Fig. 3.9 contém os MSEs obtidos com as três versões do algoritmo SCAmTF e o limiar de referência (reta horizontal) obtido com o algoritmo JADE para sinais de áudio. Da Fig. 3.9, pode-se observar que para todos os casos simulados os algoritmos propostos apresentaram MSEs significativamente menores do que o limiar de referência, sendo os desempenhos médios das três versões do algoritmo SCAmTF semelhantes. Nesse cenário não houve soluções consideradas *outliers*.

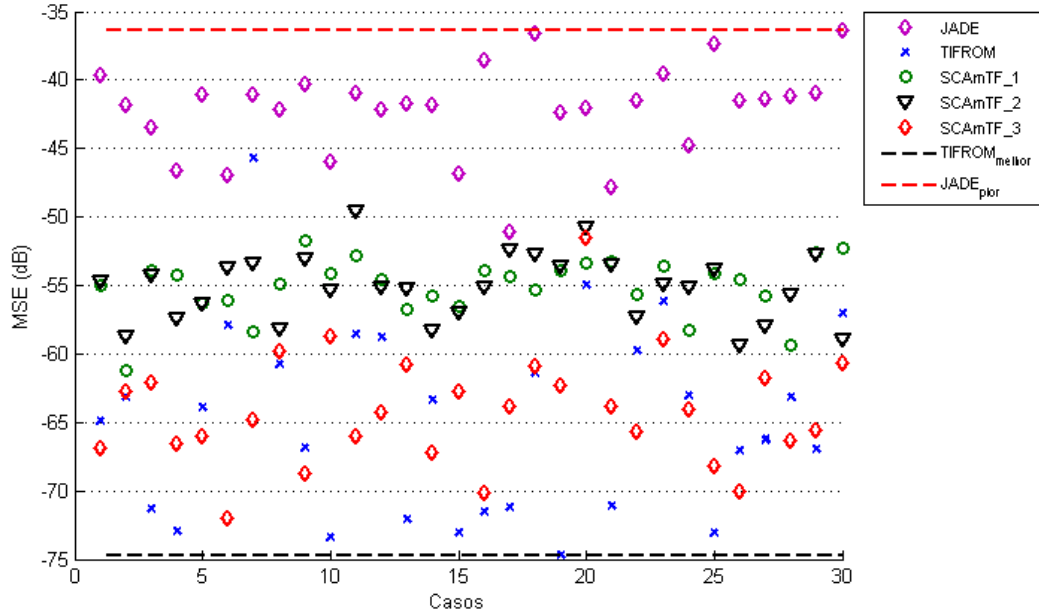


Figura 3.9: MSE dos Elementos das Matrizes de Mistura Estimadas pelo Algoritmo SCAmTF com 3 Tipos de Filtragem - Sinal de Áudio.

3.2.4 Algoritmo SCAc no Domínio Tempo-Frequência

Avaliaremos nesta seção o desempenho do algoritmo SCAc no domínio T-F (SCAcTF) com os mesmos três tipos de filtrações empregados na seção, ou seja, filtração de norma (SCAcTF_1), filtração SSP (SCAcTF_2) e filtração de *Outlier* (SCAcTF_3).

Os algoritmos propostos nesta seção empregaram 360 *bins*, os que utilizaram o filtro de norma empregaram o fator $L = 0,09$ (SCAcTF_1), os que utilizaram o filtro SSP empregaram $c_{tol} = 0,92$ (SCAcTF_2), e os que empregaram o filtro de *outlier* (SCAcTF_3) consideraram amostras válidas, para determinação do pico, aquelas que ficaram até 2σ em relação à média do agrupamento. Os valores estabelecidos neste parágrafo seguiram o mesmo procedimento descrito na seção anterior.

A transformação para o domínio T-F empregada foi a descrita na seção anterior. Da mesma forma que na seção anterior, foram igualmente fixados os valores do tamanho da janela e do salto empregado em relação ao que fora usado na seção anterior, possibilitando comparações diretas entre este método e o método descrito na seção anterior.

A Fig. 3.10 contém os MSEs obtidos com as três versões do algoritmo SCAcTF e o limiar de referência (reta horizontal), obtido com o algoritmo SOBI, para sinais de voz.

Da Fig. 3.10, pode-se observar que somente para o Caso 10, as soluções obtidas pelas três versões do algoritmo SCAcTF não foram satisfatórias. Ao analisarmos as

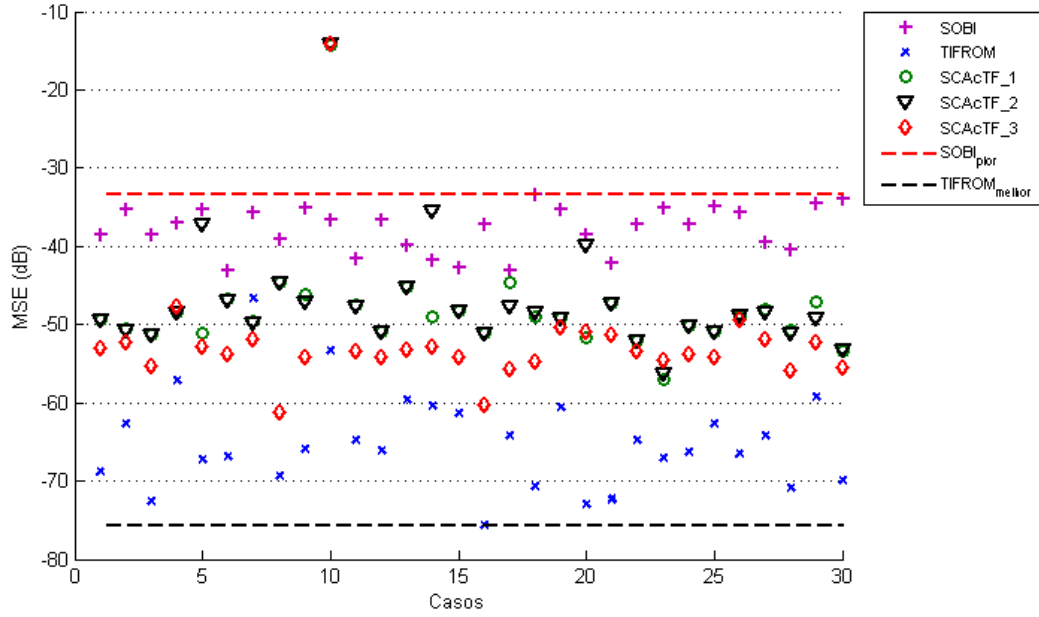


Figura 3.10: MSE dos Elementos das Matrizes de Mistura Estimadas pelo Algoritmo SCAcTF - Sinal de Voz.

soluções obtidas, podemos observar que, similarmente ao ocorrido com o algoritmo SCAM no domínio do tempo, em todos os casos um dos picos do histograma foi um valor falso e, novamente, próximo de um dos picos verdadeiros. Assim, a abordagem sugerida na Seção 3.2.2 para contornar esse problema pode aqui ser novamente adotada.

A Fig. 3.11 contém os MSEs obtidos com as três versões do algoritmo SCAcTF e o limiar de referência (reta horizontal), obtido com o algoritmo JADE, para sinais de áudio.

Da Fig. 3.11, pode-se observar que, somente para o Caso 3, a solução obtida pelo algoritmo SCAcTF_3 não foi satisfatória. Comparando a matriz de mistura desse caso, dada por

$$\begin{bmatrix} 1 & 1 & 1 \\ -1,1162 & 0,5512 & -2,0046 \\ 0,6353 & -1,0998 & -0,4931 \end{bmatrix} \quad (3.14)$$

com a sua estimativa obtida pelo algoritmo SCAcTF_3, dada por

$$\begin{bmatrix} 1 & 1 & 1 \\ -1,1147 & -2,0145 & -1,4847 \\ 0,6312 & -0,4945 & 0,7337 \end{bmatrix} \quad (3.15)$$

pode-se observar que os coeficientes da matriz de mistura correspondentes à terceira fonte foram erroneamente estimados. Para resolver esse problema, pode-se utilizar

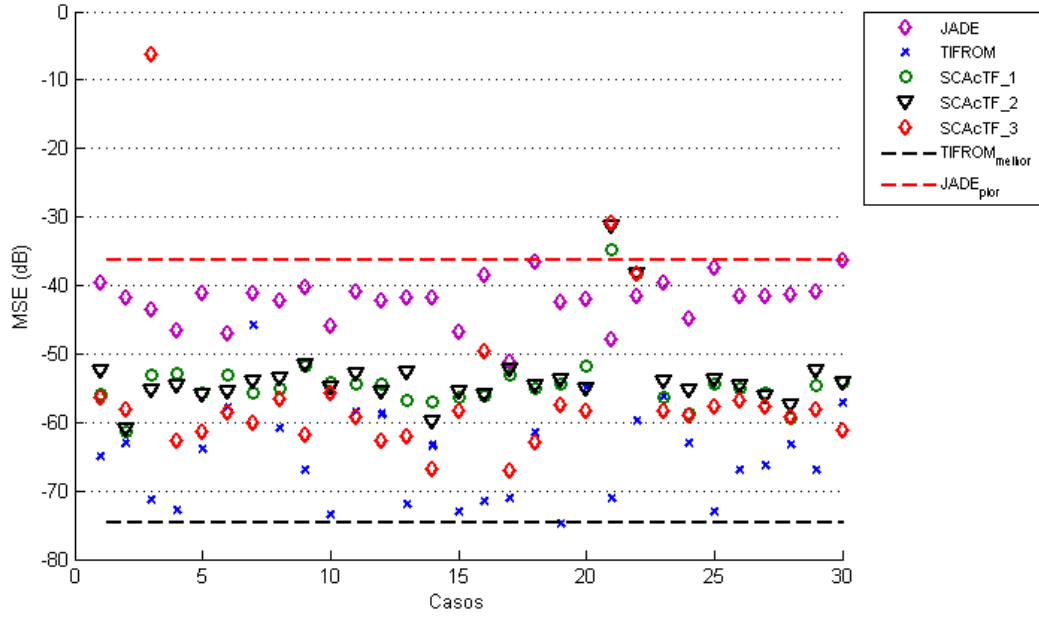


Figura 3.11: MSE dos Elementos das Matrizes de Mistura Estimadas pelo Algoritmo SCAcTF - Sinal de Áudio.

uma das soluções propostas para o algoritmo SCAM, que consiste na detecção de um número de agrupamentos maior que o número de fontes, sendo empregada a função \mathcal{F}_{CEF} para determinar quais das colunas candidatas obtidas desses agrupamentos correspondem às colunas da matriz de mistura verdadeira.

Pode-se observar que apenas para o Caso 21 os algoritmos SCAcTF apresentaram MSEs maiores do que o limiar. Analisando-se as matrizes de mistura estimadas para esse caso, verificou-se para todas as versões do algoritmo, uma discrepância em um dos coeficientes da terceira linha da matriz, sendo esse erro de natureza diversa dos observados anteriormente.

3.2.5 Algoritmos SCAz e STUE

Avaliaremos nesta seção os desempenhos dos algoritmos SCAz no domínio da frequência (SCAzTF) e STUE, descrito no Cap. 4, para sinais de voz e áudio.

Ambos os algoritmos empregaram uma janela de tamanho igual a 128 com um salto de 96 (75%) e número de STFTs adjacentes na busca das regiões de baixa variância = 10. Tal configuração obedeceu o critério previamente estabelecido pelo algoritmo TIFROM [22], a fim de permitir comparações diretas entre esses método e o algoritmo TIFROM.

A Fig. 3.12 contém os MSEs obtidos com esses dois algoritmos e o limiar de referência (reta horizontal), obtido com o algoritmo SOBI, para sinais de voz. Da Fig. 3.12, pode-se observar que somente para o Caso 15 as soluções obtidas pelos algoritmos

mos SCAzTF e STUE não foram satisfatórias. Ao analisarmos as soluções obtidas, verificamos que ambos os algoritmos estimaram corretamente todos os coeficientes do primeira mistura, e erroneamente um dos coeficientes da segunda mistura. Esse erro fez com que a função \mathcal{F}_{CEF} alinhasse incorretamente os coeficientes da terceira linha em relação aos da segunda linha da matriz de mistura.

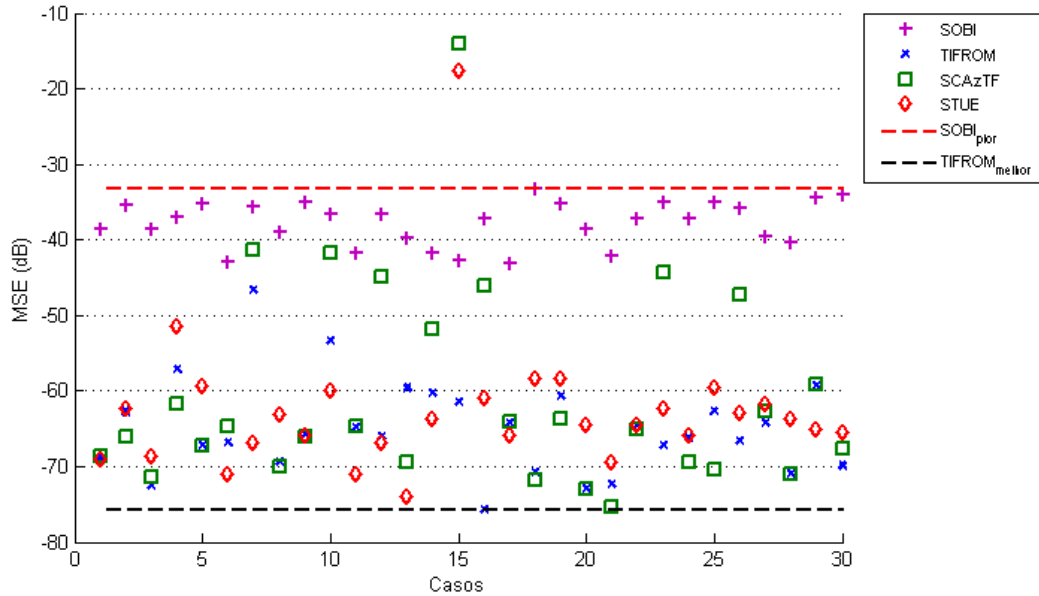


Figura 3.12: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos SCAzTF e STUE - Sinal de Voz.

Pode-se observar que para todos os casos, exceto o 15, os algoritmos propostos apresentaram MSEs consideravelmente menores do que o limiar.

A Fig. 3.13 contém os MSEs obtidos com os algoritmos SCAzTF e STUE, e o limiar de referência (reta horizontal), obtido com o algoritmo JADE, para sinais de áudio. Pode-se observar nessa figura que em todos os casos os MSEs obtidos com os dois métodos ficaram abaixo do limiar de referência.

A Tabela 3.1 apresenta as estatísticas dos algoritmos propostos e os algoritmos clássicos, correspondentes para os 30 casos simulados, com sinais de voz.

Pode-se observar da Tabela 3.1 que os algoritmos testados apresentaram bons resultados, tendo como destaque os algoritmos SCAMTF_3 e TIFROM. A Tabela 3.2 apresenta as estatísticas dos algoritmos propostos e os algoritmos clássicos, correspondentes para os 30 casos simulados com sinais de áudio.

Pode-se observar da Tabela 3.2 que os algoritmos testados apresentaram bons resultados, tendo como destaque os algoritmos STUE e SCAMTF_3.

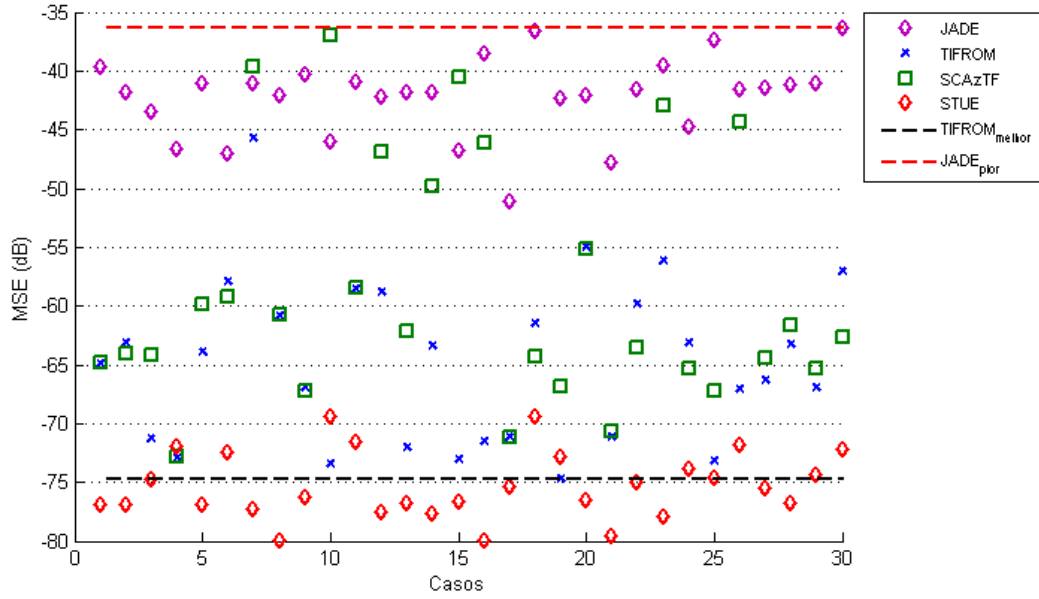


Figura 3.13: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos SCAzTF e STUE - Sinal de Áudio.

	Algoritmo	Média \pm Desvio	Mínimo	Máximo
Sinais de Voz	SCAmTF_1	$1,31 \times 10^{-5} \pm 6,95 \times 10^{-6}$	$2,19 \times 10^{-6}$	$3,44 \times 10^{-5}$
	SCAmTF_2	$2,77 \times 10^{-5} \pm 5,43 \times 10^{-5}$	$2,63 \times 10^{-6}$	$2,85 \times 10^{-4}$
	SCAmTF_3	$1,13 \times 10^{-6} \pm 9,76 \times 10^{-7}$	$4,40 \times 10^{-8}$	$3,32 \times 10^{-6}$
	SCAcTF_1	$1,25 \times 10^{-3} \pm 6,80 \times 10^{-3}$	$2,03 \times 10^{-6}$	$3,70 \times 10^{-2}$
	SCAcTF_2	$1,34 \times 10^{-3} \pm 7,10 \times 10^{-3}$	$2,35 \times 10^{-6}$	$3,92 \times 10^{-2}$
	SCAcTF_3	$1,28 \times 10^{-3} \pm 7,00 \times 10^{-3}$	$7,67 \times 10^{-7}$	$3,82 \times 10^{-2}$
	SCAzTF	$1,30 \times 10^{-3} \pm 7,09 \times 10^{-3}$	$2,91 \times 10^{-8}$	$3,88 \times 10^{-2}$
	STUE	$5,84 \times 10^{-4} \pm 3,19 \times 10^{-3}$	$3,97 \times 10^{-8}$	$1,75 \times 10^{-2}$
	FastICA	$1,95 \times 10^{-5} \pm 1,64 \times 10^{-5}$	$1,88 \times 10^{-6}$	$5,82 \times 10^{-5}$
	JADE	$2,28 \times 10^{-5} \pm 1,60 \times 10^{-5}$	$3,01 \times 10^{-6}$	$5,60 \times 10^{-5}$
	SOBI	$2,00 \times 10^{-4} \pm 1,15 \times 10^{-4}$	$4,91 \times 10^{-5}$	$4,65 \times 10^{-4}$
	TIFROM	$1,27 \times 10^{-6} \pm 3,99 \times 10^{-6}$	$2,70 \times 10^{-8}$	$2,19 \times 10^{-5}$

Tabela 3.1: Estatísticas dos MSEs Obtidos com os Algoritmos Propostos e Clássicos para Sinais de Voz.

	Algoritmo	Média \pmDesvio	Mínimo	Máximo
Sinais de Áudio	SCAmTF_1	$3,43 \times 10^{-6} \pm 1,41 \times 10^{-6}$	$7,55 \times 10^{-7}$	$6,70 \times 10^{-6}$
	SCAmTF_2	$3,59 \times 10^{-6} \pm 2,17 \times 10^{-6}$	$1,15 \times 10^{-6}$	$1,11 \times 10^{-5}$
	SCAmTF_3	$6,87 \times 10^{-7} \pm 1,26 \times 10^{-6}$	$6,35 \times 10^{-8}$	$7,12 \times 10^{-6}$
	SCAcTF_1	$8,49 \times 10^{-6} \pm 2,77 \times 10^{-5}$	$7,15 \times 10^{-7}$	$1,52 \times 10^{-4}$
	SCAcTF_2	$8,72 \times 10^{-6} \pm 2,74 \times 10^{-5}$	$8,01 \times 10^{-7}$	$1,51 \times 10^{-4}$
	SCAcTF_3	$6,96 \times 10^{-6} \pm 2,83 \times 10^{-5}$	$2,02 \times 10^{-7}$	$1,51 \times 10^{-4}$
	SCAzTF	$1,84 \times 10^{-5} \pm 4,31 \times 10^{-5}$	$5,28 \times 10^{-8}$	$1,98 \times 10^{-4}$
	STUE	$3,71 \times 10^{-8} \pm 2,78 \times 10^{-8}$	$1,01 \times 10^{-8}$	$1,17 \times 10^{-7}$
	FastICA	$3,78 \times 10^{-5} \pm 3,12 \times 10^{-5}$	$5,55 \times 10^{-6}$	$1,24 \times 10^{-4}$
	JADE	$7,76 \times 10^{-5} \pm 5,48 \times 10^{-5}$	$7,75 \times 10^{-6}$	$2,30 \times 10^{-4}$
	SOBI	$1,48 \times 10^{-6} \pm 1,92 \times 10^{-6}$	$6,97 \times 10^{-8}$	$9,13 \times 10^{-6}$
	TIFROM	$1,51 \times 10^{-6} \pm 4,85 \times 10^{-6}$	$3,38 \times 10^{-8}$	$2,68 \times 10^{-5}$

Tabela 3.2: Estatísticas dos MSEs Obtidos com os Algoritmos Propostos e Clássicos para Sinais de Áudio.

3.3 Desempenho dos Algoritmos Propostos em Misturas com Ruído

Nesta seção, apresentamos os resultados obtidos com os métodos propostos para misturas de sinais de voz e de áudio com três fontes e três sensores na presença de ruído. Não foi utilizada nenhuma técnica de remoção de ruído ou feita modificação nos algoritmos para otimizá-los para os cenários simulados, o que exigiria um conhecimento prévio das características do ruído.

Dependendo da natureza do ruído, seu efeito pode ser melhor modelado adicionando-se o ruído diretamente aos sinais das fontes ou aos sinais dos sensores. Segundo a literatura [10], a presença de ruído nas fontes não causa erros significativos na estimação dos coeficientes da matriz de mistura; contudo, afeta a recuperação dos sinais originais, uma vez que o algoritmo de reconstrução produzirá estimativas de versões ruidosas dos sinais das fontes. Portanto, é necessário um tratamento diferenciado para a remoção do ruído, podendo ser feito após a obtenção das estimativas das fontes. A presença dos ruídos independentes inseridos diretamente nos sensores é um fator complicador maior do que a dos inseridos nas fontes, uma vez que, neste caso, as razões entre os sinais nos sensores não se mantêm constantes mesmo nas regiões onde só existe uma única fonte ativa.

Como o objetivo desta seção é avaliar os desempenhos dos algoritmos de estimação da matriz de mistura na presença de ruídos, consideraremos somente o caso de ruídos inseridos diretamente nos sensores. Cada sensor é perturbado com uma amostra de ruído distinto, sendo, no entanto, mantido o nível (energia) de ruído para todos os sensores. Os dois tipos de ruído considerados foram ruído branco gaussiano e de *babble*, com SNR variando de 5 a 25 dB, com passo de 5 dB. Serão apresentadas a seguir as estatísticas dos erros médios quadráticos obtidos pelos algoritmos propostos STUE, SCAzTF, SCAmTF_2 e SCACTF_2, os quais obtiveram os melhores desempenhos (medidos pela SDR) sem a presença de ruído. Serão também apresentados os resultados obtidos com os algoritmos clássicos, para fins de comparação.

3.3.1 Resultados com Ruído Branco

Para evitar o excesso de tabelas e gráficos, apresentaremos os resultados obtidos pelos algoritmos clássicos e propostos com ruído branco para cada uma das 30 matrizes de misturas utilizadas apenas para SNR de 10 dB. Ao final, incluiremos um gráfico com os valores médios de MSE obtidos por todos os algoritmos para os demais valores de SNR simulados.

As Figs. 3.14 e 3.15 apresentam os erros médios quadráticos (em dB) das estimativas das matrizes de mistura para misturas com ruído branco e SNR de 10 dB

para cada um dos 30 casos simulados, com sinais de áudio e voz, respectivamente.

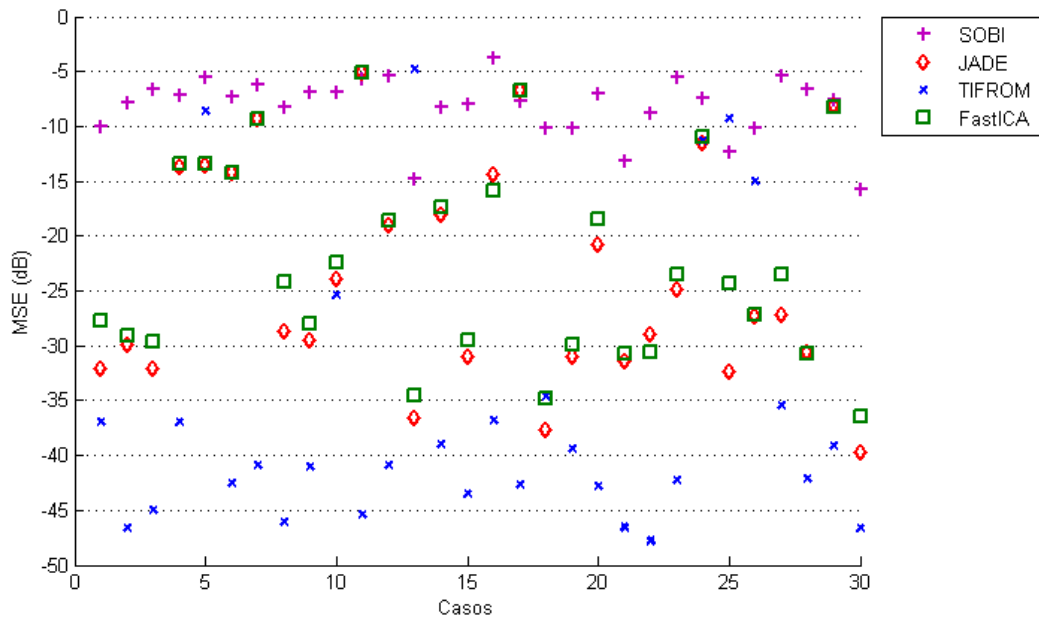


Figura 3.14: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Voz - Ruído Branco, SNR = 10 dB.

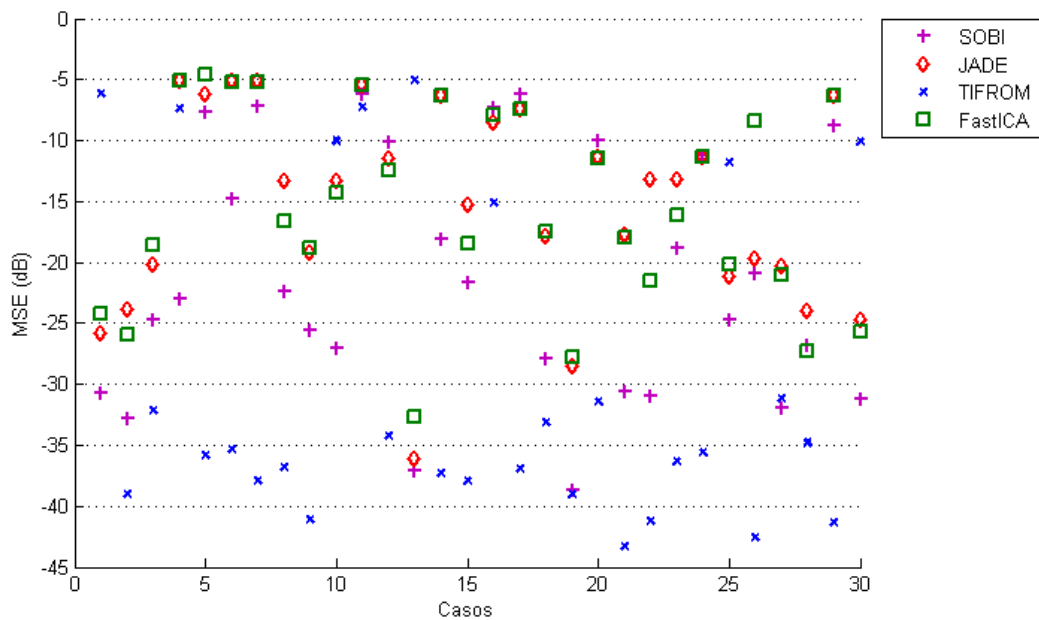


Figura 3.15: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Áudio - Ruído Branco, SNR = 10 dB.

Podemos observar, como esperado, que os valores obtidos pelos algoritmos clássicos, foram piores do que os obtidos sem a presença do ruído, sendo que, para

esse nível de ruído, o algoritmo TIFROM apresentou a menor degradação.

As Figs. 3.16 e 3.17 apresentam os resultados obtidos com os métodos propostos e com o algoritmo de referência SOBI para sinais de voz e áudio, respectivamente. Pode-se observar que os algoritmos SCAzTF e STUE obtiveram soluções com valores de MSE inferiores aos limiares estabelecidos tanto para sinais de voz quanto para sinais de áudio. O mesmo comportamento não foi observado com os algoritmos SCAcTF_2 e SCAMTF_2, para sinais de voz e áudio, respectivamente. As Tabelas 3.3 e 3.4 apresentam os valores médios, máximos e mínimos, e desvios padrões, dos 30 casos simulados para sinais de voz e áudio, respectivamente.

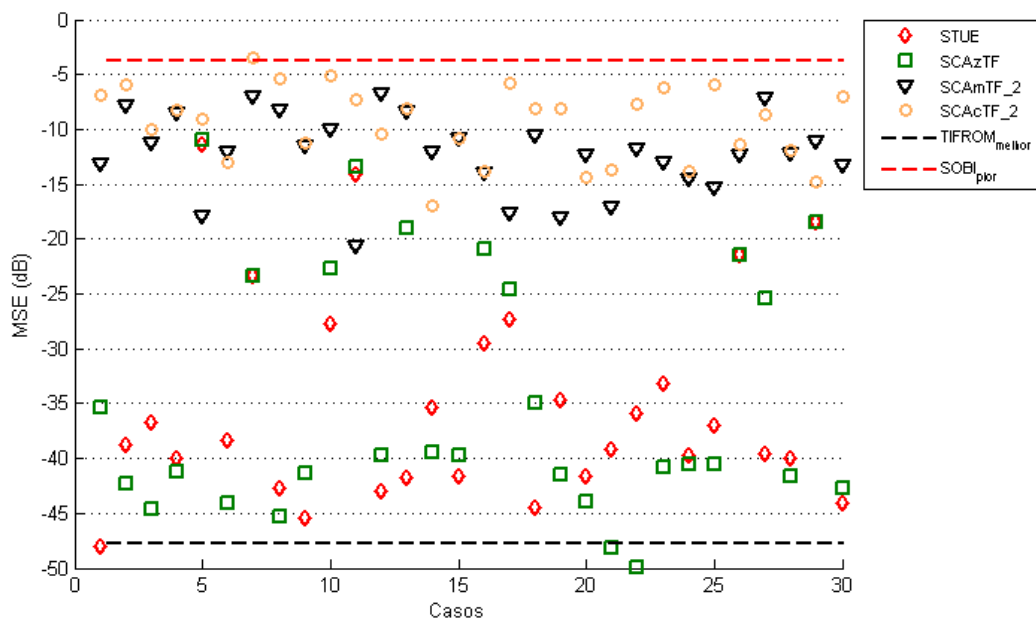


Figura 3.16: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Propostos - Sinal de Voz - Ruído Branco, SNR = 10 dB.

As Figs. 3.18 e 3.19 apresentam os valores médios dos MSEs dos algoritmos clássicos e propostos em função da SNR para sinais de voz e áudio, respectivamente. Pode-se observar na Fig. 3.18 que os algoritmos STUE e SCAzTF apresentaram os menores valores médios de MSE para $SNR \leq 15$ dB. O algoritmo TIFROM apresentou melhores valores médios de MSE para $SNR \geq 15$ dB; já para valores de SNR menores, seu desempenho foi mais afetado pela presença de ruído. Os algoritmos JADE e FastICA apresentaram valores de MSE praticamente iguais para todas as SNRs, e maiores do que os dos algoritmos STUE e SCAzTF. Os algoritmos SOBI, SCAcTF_2 e SCAMTF_2 resultaram em MSEs médio significativamente maiores do que os outros métodos para sinais de voz com todas as SNRs testadas. Na análise de incerteza (linhas tracejadas) podemos destacar que, o algoritmo STUE apresentou valores de MSE acrescidos do desvio inferiores ao valor de MSE médio do algoritmo TIFROM,

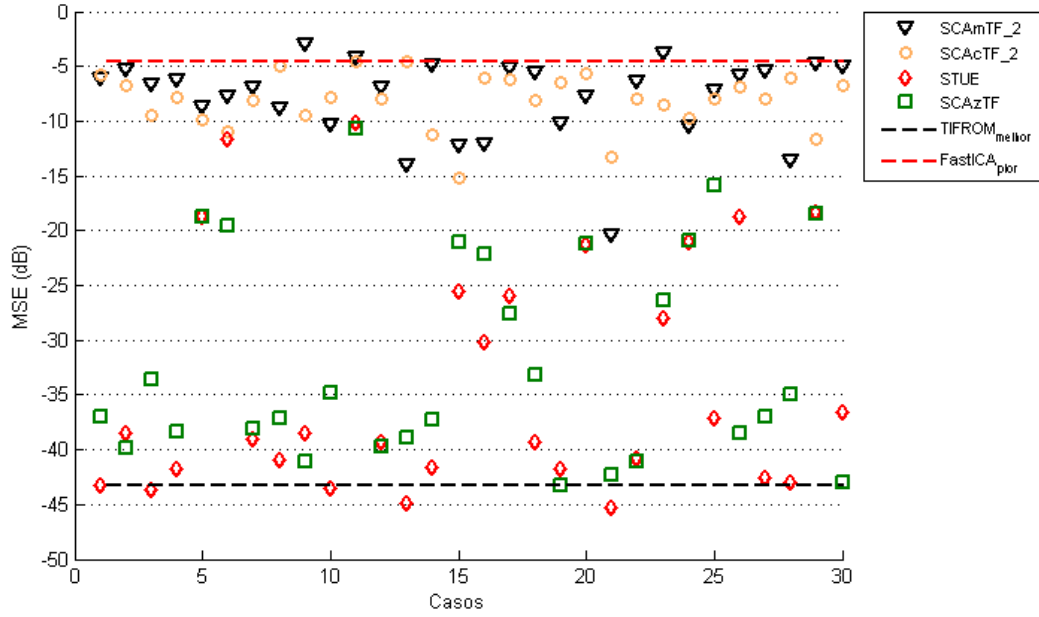


Figura 3.17: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Áudio - Ruído Branco, SNR = 10 dB.

	Algoritmo	Média \pm Desvio	Mínimo	Máximo
Sinais de Voz	SCAmTF_1	$1,93 \times 10^{-1} \pm 1,11 \times 10^{-1}$	$1,16 \times 10^{-4}$	$4,24 \times 10^{-1}$
	SCAmTF_2	$7,97 \times 10^{-2} \pm 5,77 \times 10^{-2}$	$8,60 \times 10^{-3}$	$2,10 \times 10^{-1}$
	SCAmTF_3	$1,77 \times 10^{-1} \pm 1,03 \times 10^{-1}$	$2,44 \times 10^{-4}$	$4,24 \times 10^{-1}$
	SCAcTF_1	$1,84 \times 10^{-1} \pm 1,22 \times 10^{-1}$	$3,84 \times 10^{-5}$	$6,26 \times 10^{-1}$
	SCAcTF_2	$1,48 \times 10^{-1} \pm 1,02 \times 10^{-1}$	$1,98 \times 10^{-2}$	$4,47 \times 10^{-1}$
	SCAcTF_3	$1,67 \times 10^{-1} \pm 9,19 \times 10^{-2}$	$2,24 \times 10^{-4}$	$4,17 \times 10^{-1}$
	SCAzTF	$6,10 \times 10^{-3} \pm 1,63 \times 10^{-2}$	$1,03 \times 10^{-5}$	$7,91 \times 10^{-2}$
	STUE	$4,90 \times 10^{-3} \pm 1,50 \times 10^{-2}$	$1,56 \times 10^{-5}$	$7,37 \times 10^{-2}$
	FastICA	$3,67 \times 10^{-2} \pm 7,13 \times 10^{-2}$	$2,24 \times 10^{-4}$	$3,05 \times 10^{-1}$
	JADE	$3,59 \times 10^{-2} \pm 7,16 \times 10^{-2}$	$1,05 \times 10^{-4}$	$3,05 \times 10^{-1}$
	SOBI	$1,79 \times 10^{-1} \pm 8,96 \times 10^{-2}$	$2,69 \times 10^{-2}$	$4,28 \times 10^{-1}$
	TIFROM	$2,33 \times 10^{-2} \pm 6,80 \times 10^{-2}$	$1,67 \times 10^{-5}$	$3,33 \times 10^{-1}$

Tabela 3.3: Estatísticas dos MSEs Obtidos pelos Algoritmos Propostos e Clássicos para Sinais de Voz com Ruído Branco (SNR=10 dB).

para SNR ≤ 10 dB. Isso se traduz em um desempenho superior do algoritmo STUE em relação ao TIFROM, para cenários com sinais de voz contaminados com ruído branco nos sensores e SNR ≤ 10 dB.

Na Fig. 3.19 observa-se que os algoritmos STUE e SCAzTF também apresentaram as menores degradações nas estimativas dos coeficientes da matriz de mistura para sinais de áudio, exceto para SNR = 5 dB, para a qual o algoritmo SOBI apre-

	Algoritmo	Média \pm Desvio	Mínimo	Máximo
Sinais de Áudio	SCAmTF_1	$2,15 \times 10^{-1} \pm 1,19 \times 10^{-1}$	$1,17 \times 10^{-2}$	$4,20 \times 10^{-1}$
	SCAmTF_2	$2,13 \times 10^{-1} \pm 1,22 \times 10^{-1}$	$9,20 \times 10^{-3}$	$5,06 \times 10^{-1}$
	SCAmTF_3	$2,11 \times 10^{-1} \pm 1,16 \times 10^{-1}$	$1,08 \times 10^{-2}$	$4,19 \times 10^{-1}$
	SCAcTF_1	$2,03 \times 10^{-1} \pm 1,22 \times 10^{-1}$	$9,86 \times 10^{-5}$	$4,30 \times 10^{-1}$
	SCAcTF_2	$1,77 \times 10^{-1} \pm 8,48 \times 10^{-2}$	$3,03 \times 10^{-2}$	$3,51 \times 10^{-1}$
	SCAcTF_3	$1,89 \times 10^{-1} \pm 1,23 \times 10^{-1}$	$1,26 \times 10^{-4}$	$4,28 \times 10^{-1}$
	SCAzTF	$6,30 \times 10^{-3} \pm 1,63 \times 10^{-2}$	$4,79 \times 10^{-5}$	$8,64 \times 10^{-2}$
	STUE	$7,70 \times 10^{-3} \pm 2,09 \times 10^{-2}$	$2,98 \times 10^{-5}$	$9,58 \times 10^{-2}$
	FastICA	$9,65 \times 10^{-2} \pm 1,18 \times 10^{-1}$	$5,39 \times 10^{-4}$	$3,44 \times 10^{-1}$
	JADE	$9,25 \times 10^{-2} \pm 1,11 \times 10^{-1}$	$2,42 \times 10^{-4}$	$3,13 \times 10^{-1}$
	SOBI	$5,15 \times 10^{-2} \pm 7,94 \times 10^{-2}$	$1,34 \times 10^{-4}$	$2,43 \times 10^{-1}$
	TIFROM	$4,10 \times 10^{-2} \pm 8,34 \times 10^{-2}$	$4,68 \times 10^{-5}$	$3,16 \times 10^{-1}$

Tabela 3.4: Estatísticas dos MSEs Obtidos pelos Algoritmos Propostos e Clássicos para Sinais de Áudio com Ruído Branco (SNR=10 dB).

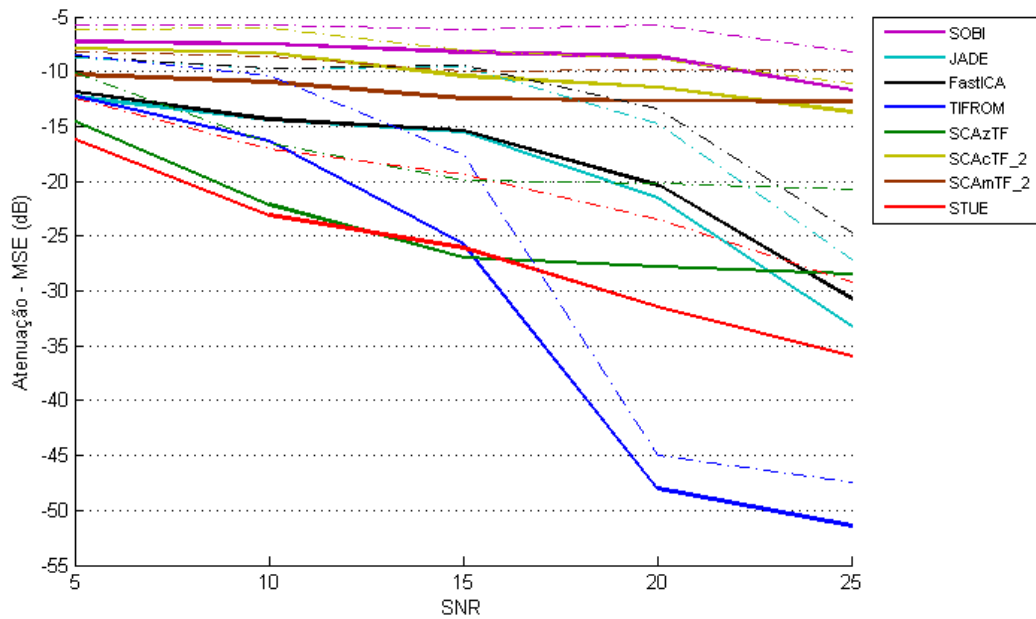


Figura 3.18: Valores Médios do MSE em Função da SNR para Sinais de Voz com Ruído Branco.

sentou valores médios de MSE ligeiramente menores. O algoritmo SOBI apresentou um desempenho consideravelmente melhor para sinais de áudio, em relação ao seu desempenho com sinais de voz. Já os algoritmos SCAmTF_2 e SCAcTF_2 voltaram a apresentar desempenhos semelhantes.

Vale ressaltar que, tanto para sinais de voz quanto de áudio, os desempenhos médios dos algoritmos STUE e SCAzTF mantêm-se praticamente insensíveis à pre-

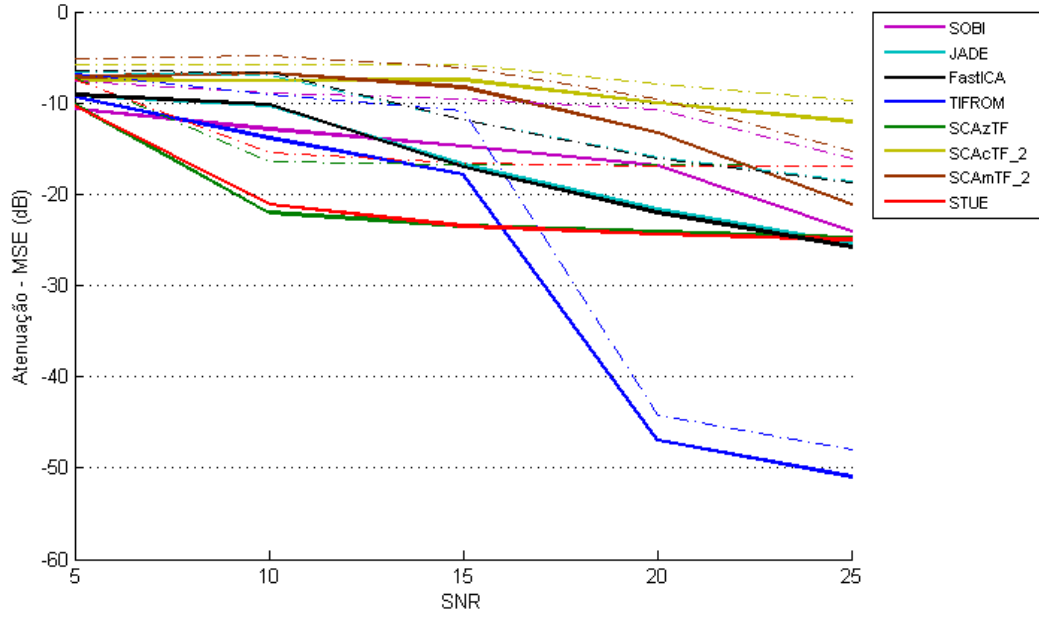


Figura 3.19: Valores Médios do MSE em Função da SNR para Sinais de Áudio com Ruído Branco.

sença de ruído nos sensores para $SNR \geq 10$ dB. A análise de incerteza mostra que esses algoritmos são melhores que o algoritmo TIFROM na faixa de SNR entre 7 e 13 dB.

3.3.2 Resultados com Ruído *Babble*

Nesta seção, apresentamos os MSEs obtidos com os algoritmos clássicos e propostos com ruído *babble*. Como nas simulações com ruído branco, consideramos valores de SNR na faixa 5 dB a 25 dB, com passo de 5 dB, sendo amostras independentes de ruído adicionadas a cada sensor. Novamente, para evitar o excesso de tabelas e gráficos, apresentaremos os resultados obtidos para cada uma das 30 matrizes de misturas utilizadas apenas para SNR de 10 dB. Ao final, incluiremos um gráfico com os valores médios de MSE (em dB) obtidos por todos os algoritmos para os demais valores de SNR simulados.

As Figs. 3.20 e 3.21 apresentam os erros médios quadráticos das estimativas das matrizes de mistura para misturas com ruído *babble* e SNR de 10 dB para cada um dos 30 casos simulados, com sinais de áudio e voz, respectivamente. As Figs. 3.22 e 3.23 apresentam os resultados obtidos com os algoritmos SCAzTF, SCAmTF_2, SCAcTF_2, STUE e SOBI. As Tabelas 3.5 e 3.6 apresentam os valores médios, mínimos e máximos e desvios padrões dos 30 casos simulados para sinais de voz e áudio, respectivamente.

Das Figs. 3.22 e 3.23 podemos observar que, diferentemente do que ocorreu com

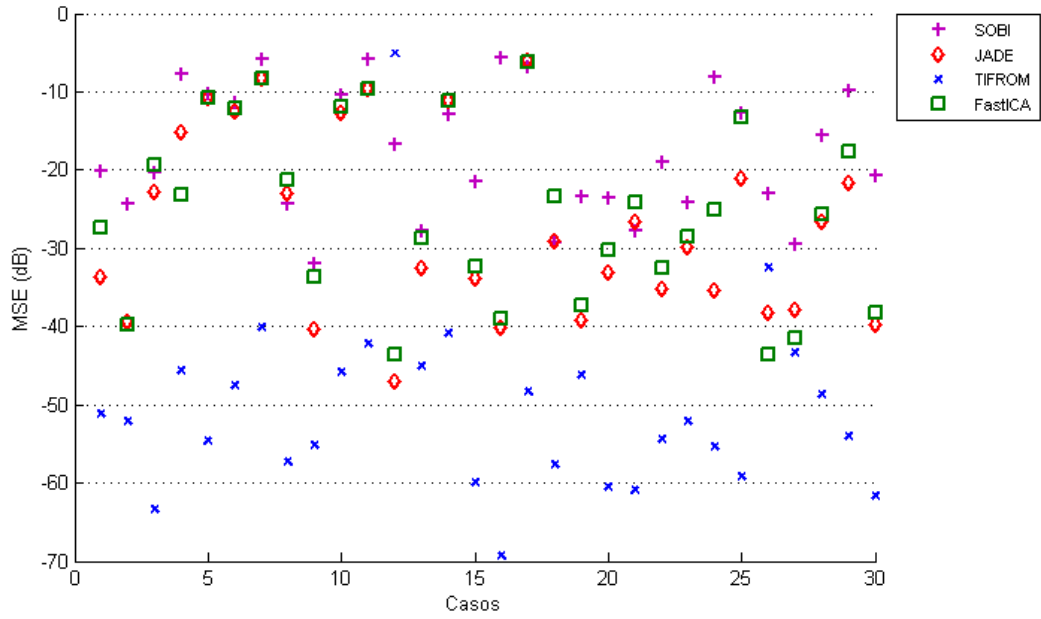


Figura 3.20: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Voz - Ruído *Babble*, SNR = 10 dB.

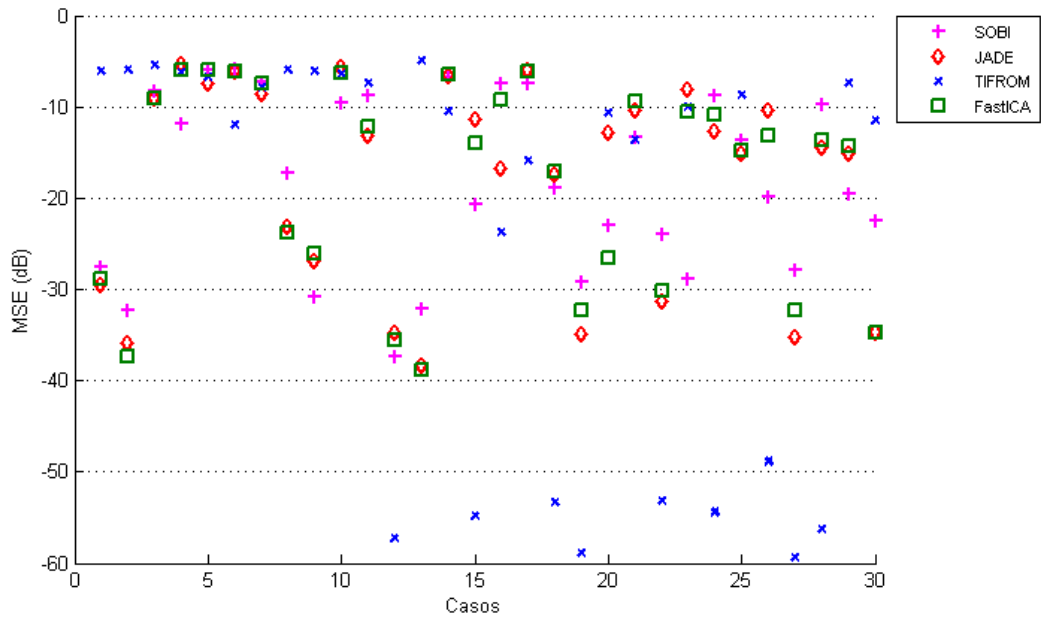


Figura 3.21: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Clássicos - Sinal de Áudio - Ruído *Babble*, SNR = 10 dB.

o ruído branco, todas as soluções obtidas pelos algoritmos propostos resultaram em valores de MSE dentro da faixa de aceitabilidade, tanto para sinais de áudio quanto para sinais de voz. Cabe destacar novamente, o desempenho do algoritmo STUE que apresentou, em alguns casos, valores de MSE inferiores ao limiar mínimo

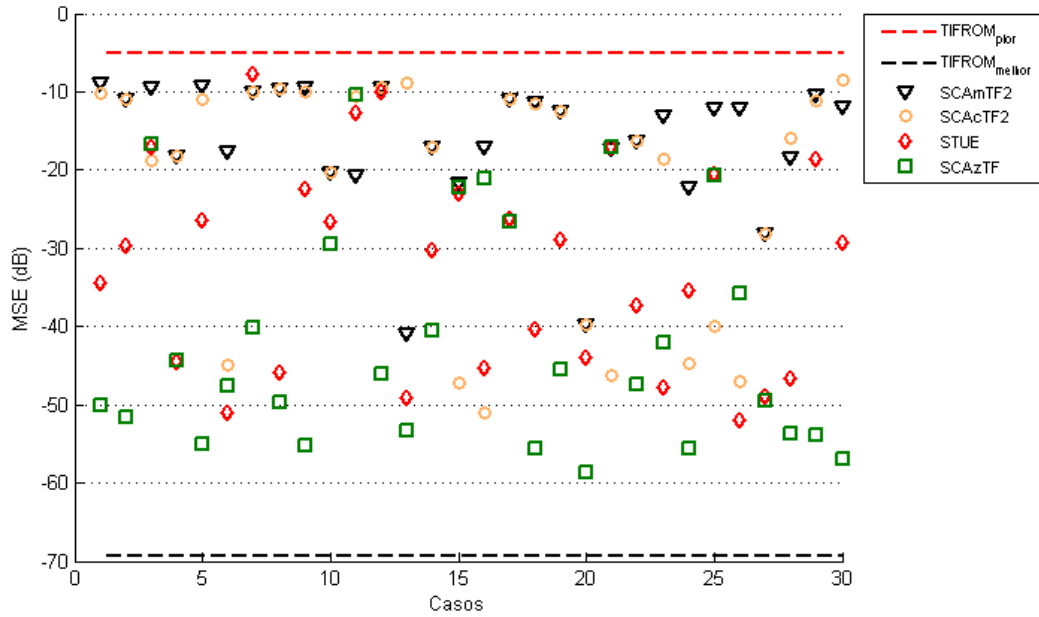


Figura 3.22: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Propostos - Sinal de Voz - Ruído *Babble*, SNR = 10 dB.

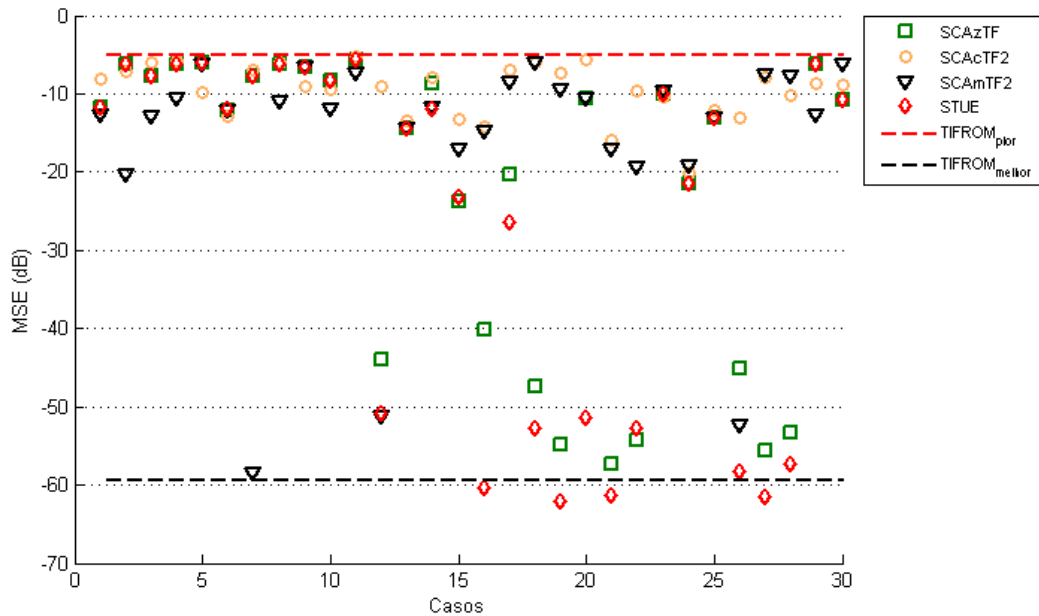


Figura 3.23: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos Propostos - Sinal de Áudio - Ruído *Babble*, SNR = 10 dB.

estabelecido pela faixa definida pelos algoritmos clássicos.

As Figs. 3.24 e 3.25 apresentam os valores médios dos MSEs dos algoritmos clássicos e propostos em função da SNR para sinais de voz e áudio, respectivamente.

	Algoritmo	Média \pm Desvio	Mínimo	Máximo
Sinais de Voz	SCAmTF_1	$1,49 \times 10^{-1} \pm 1,01 \times 10^{-1}$	$2,40 \times 10^{-5}$	$3,50 \times 10^{-1}$
	SCAmTF_2	$5,25 \times 10^{-2} \pm 4,36 \times 10^{-2}$	$8,24 \times 10^{-5}$	$1,31 \times 10^{-1}$
	SCAmTF_3	$1,51 \times 10^{-1} \pm 1,01 \times 10^{-1}$	$7,51 \times 10^{-6}$	$3,64 \times 10^{-1}$
	SCAcTF_1	$1,27 \times 10^{-1} \pm 1,06 \times 10^{-1}$	$7,82 \times 10^{-6}$	$3,44 \times 10^{-1}$
	SCAcTF_2	$4,80 \times 10^{-2} \pm 4,70 \times 10^{-2}$	$7,85 \times 10^{-6}$	$1,43 \times 10^{-1}$
	SCAcTF_3	$1,18 \times 10^{-1} \pm 1,03 \times 10^{-1}$	$9,50 \times 10^{-6}$	$3,01 \times 10^{-1}$
	SCAzTF	$5,30 \times 10^{-3} \pm 1,74 \times 10^{-2}$	$1,38 \times 10^{-6}$	$9,30 \times 10^{-2}$
	STUE	$1,35 \times 10^{-2} \pm 3,53 \times 10^{-2}$	$6,44 \times 10^{-6}$	$1,67 \times 10^{-1}$
	FastICA	$2,98 \times 10^{-2} \pm 5,53 \times 10^{-2}$	$4,36 \times 10^{-5}$	$2,40 \times 10^{-1}$
	JADE	$2,82 \times 10^{-2} \pm 5,71 \times 10^{-2}$	$1,97 \times 10^{-5}$	$2,54 \times 10^{-1}$
	SOBI	$6,44 \times 10^{-2} \pm 8,85 \times 10^{-2}$	$6,27 \times 10^{-4}$	$2,79 \times 10^{-1}$
TIFROM	$1,04 \times 10^{-2} \pm 5,68 \times 10^{-2}$	$1,16 \times 10^{-7}$	$3,11 \times 10^{-1}$	

Tabela 3.5: Estatísticas dos MSEs Obtidos pelos Algoritmos Propostos e Clássicos para Sinais de Voz com Ruído *Babble* (SNR=10 dB).

	Algoritmo	Média \pm Desvio	Mínimo	Máximo
Sinais de Áudio	SCAmTF_1	$1,18 \times 10^{-1} \pm 8,20 \times 10^{-2}$	$9,40 \times 10^{-3}$	$3,18 \times 10^{-1}$
	SCAmTF_2	$8,76 \times 10^{-2} \pm 7,82 \times 10^{-2}$	$1,46 \times 10^{-6}$	$2,56 \times 10^{-1}$
	SCAmTF_3	$1,12 \times 10^{-1} \pm 8,52 \times 10^{-2}$	$1,12 \times 10^{-5}$	$3,19 \times 10^{-1}$
	SCAcTF_1	$1,38 \times 10^{-1} \pm 1,00 \times 10^{-1}$	$2,80 \times 10^{-3}$	$3,18 \times 10^{-1}$
	SCAcTF_2	$1,40 \times 10^{-1} \pm 8,20 \times 10^{-2}$	$9,70 \times 10^{-3}$	$2,94 \times 10^{-1}$
	SCAcTF_3	$1,29 \times 10^{-1} \pm 1,03 \times 10^{-1}$	$5,37 \times 10^{-6}$	$3,18 \times 10^{-1}$
	SCAzTF	$9,43 \times 10^{-2} \pm 9,87 \times 10^{-2}$	$1,88 \times 10^{-6}$	$2,75 \times 10^{-1}$
	STUE	$8,90 \times 10^{-2} \pm 1,00 \times 10^{-1}$	$6,10 \times 10^{-7}$	$2,76 \times 10^{-1}$
	FastICA	$8,20 \times 10^{-2} \pm 9,37 \times 10^{-2}$	$1,29 \times 10^{-4}$	$2,53 \times 10^{-1}$
	JADE	$8,11 \times 10^{-2} \pm 9,44 \times 10^{-2}$	$1,45 \times 10^{-4}$	$2,90 \times 10^{-1}$
	SOBI	$7,18 \times 10^{-2} \pm 8,65 \times 10^{-2}$	$1,83 \times 10^{-4}$	$2,66 \times 10^{-1}$
	TIFROM	$1,15 \times 10^{-1} \pm 1,09 \times 10^{-1}$	$1,14 \times 10^{-6}$	$3,18 \times 10^{-1}$

Tabela 3.6: Estatísticas dos MSEs Obtidos pelos Algoritmos Propostos e Clássicos para Sinais de Áudio com Ruído *Babble* (SNR=10 dB).

Na Fig. 3.24 pode-se observar que, semelhantemente ao verificado com sinal de voz com ruído branco, os algoritmos STUE e SCAzTF apresentaram os menores valores de MSE entre os algoritmos propostos. Os algoritmos SCAmTF_2 e SCAcTF_2, igualmente, se mostraram um pouco mais sensíveis à presença de ruído *babble*, enquanto o algoritmo SOBI apresentou o pior desempenho para sinais de voz com baixo SNR.

Diferentemente do que ocorreu no caso com ruído branco, a análise de incerteza demonstra equivalência entre os algoritmos TIFROM, SCAzTF e STUE para SNR

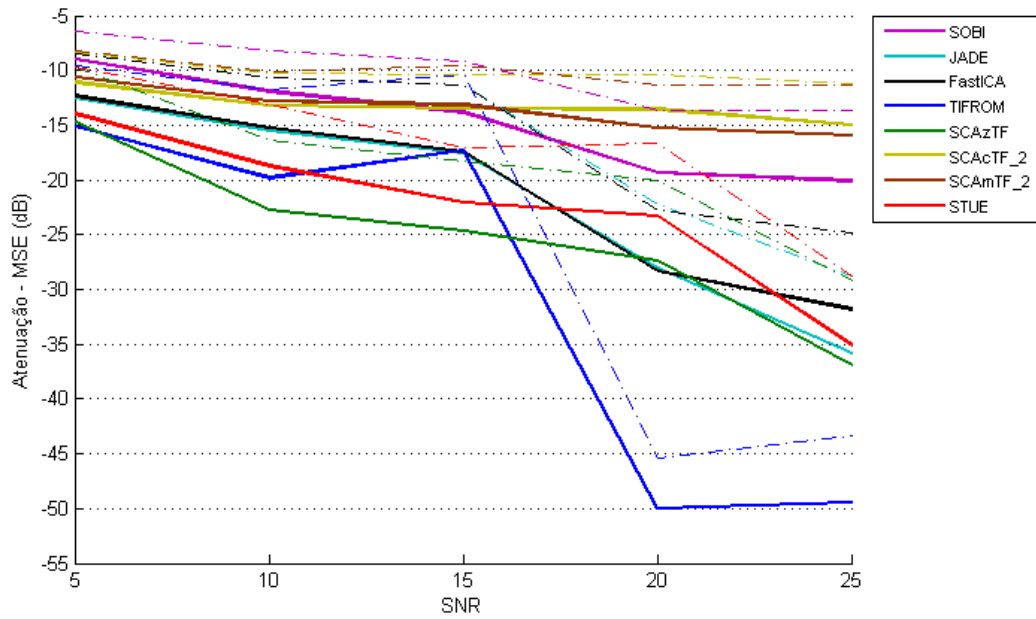


Figura 3.24: Valores Médios do MSE em Função da SNR para Sinais de Voz com Ruído *Babble*.

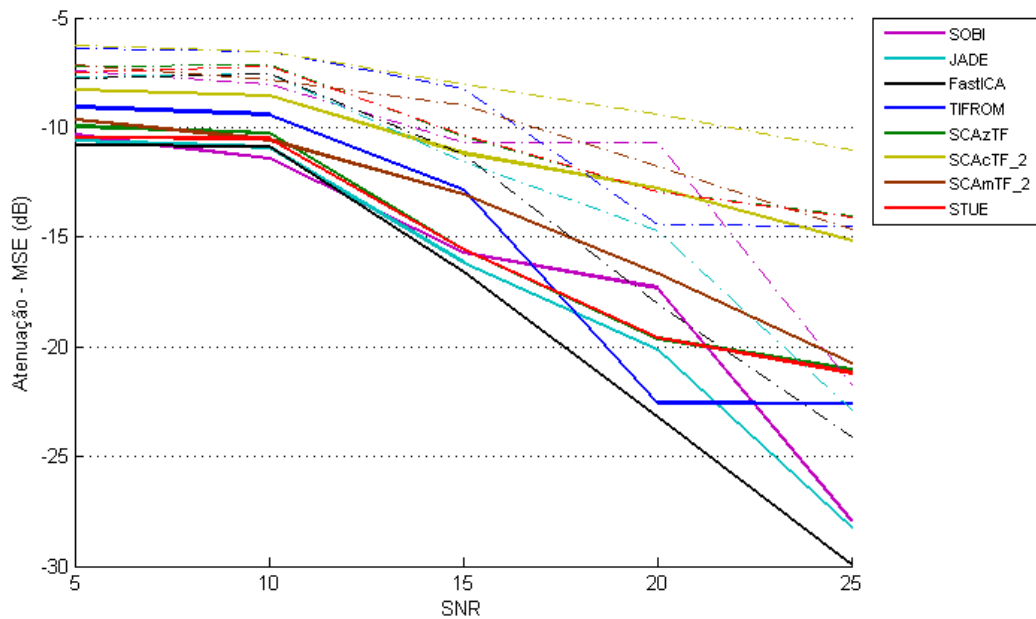


Figura 3.25: Valores Médios do MSE em Função da SNR para Sinais de Áudio com Ruído *Babble*.

≤ 15 dB, e uma preferência pelo algoritmo TIFROM para $SNR > 15$ dB.

Já na Fig. 3.25, a qual contém os resultados para sinais de áudio, observa-se uma degradação maior no desempenho dos algoritmos STUE e SCAzTF em relação ao cenário de ruído branco. Ainda assim, esses algoritmos estão entre os que obtiveram

os valores de MSE médio mais baixos. Para um cenário muito ruidoso, com SNR de 5 e 10 dB, os algoritmos TIFROM e SCAC_{TF_2} apresentaram os piores desempenhos. Já na faixa de SNR entre 5 e 15 dB, o algoritmo SCAM_{TF_2} apresentou desempenho melhor do que os algoritmos TIFROM e SCAC_{TF_2}.

A análise de incerteza mostra uma equivalência, num melhor patamar, dos algoritmos STUE, SCAzTF, SCAM_{TF_2}, FastICA, JADE e SOBI, e equivalência dos algoritmos TIFROM e SCAC_{TF_2}, num patamar ligeiramente pior.

3.4 Estudo dos Algoritmos Propostos Empregando Medidas Obtidas após a Reconstrução das Fontes

Nesta seção, avaliamos os desempenhos dos métodos de estimação da matriz de misturas através das medidas SDR, SIR e SAR, descritas na Seção 2.3 e calculadas a partir dos sinais das fontes reconstruídas. Para obtermos tais sinais, a matriz de separação é estimada pela inversa da matriz de mistura obtida por cada método, sendo, em seguida, aplicada ao vetor de misturas, gerando o vetor com as estimativas das fontes.

Esta seção foi dividida em duas subseções, a primeira contendo experimentos que ilustram algumas características importantes dos algoritmos elaborados, e a segunda contendo um resumo dos resultados obtidos com a presença de ruído aditivo.

3.4.1 Resultados de Experimentos Individuais

Foram geradas 30 matrizes de misturas com diferentes atenuações, conforme descrito na Seção B.1. As medidas SDR, SIR e SAR, para cada estimativa de matriz de mistura, foram obtidas pelo algoritmo desenvolvido em [40]. Nas tabelas abaixo, os valores dessas medidas correspondem às médias dos valores obtidos para todas as estimativas dos 30 casos testados.

Avaliaremos os desempenhos dos algoritmos SCAM e SCAC em suas implementações no domínio do tempo e no domínio tempo-frequência (T-F). Essas duas formas de implementação também foram propostas para o algoritmo TIFROM.

A Tabela 3.7 resume as características dos algoritmos avaliados nesta seção, contendo as informações dos domínios dos dados (Tempo ou T-F), do tipo de busca (por Detecção de Picos, D.P., ou região SSZ), e da utilização ou não de função objetivo (\mathcal{F}_{obj}) para alinhar os coeficientes da matriz de mistura. Nas tabelas e figuras a seguir, sempre que aparecer um algoritmo que utilize uma função objetivo para reposicionar os coeficientes e essa função não estiver especificada, empregou-se

a função \mathcal{F}_{CEF} .

#	Nome	Algoritmo	Dominio	Busca	\mathcal{F}_{obj}
01	SCAm-t	SCAm	Tempo	D.P.	×
02	SCAmTF	SCAm	T-F	D.P.	×
03	SCAc-t	SCAc	Tempo	D.P.	-
04	SCAcTF	SCAc	T-F	D.P.	-
05	SCAz	SCAz	T-F	SSZ	×

Tabela 3.7: Características dos Algoritmos Avaliados.

Experimento 1

O objetivo desse experimento é avaliar o efeito da matriz de branqueamento nos algoritmos SCAm e SCAc (portanto, não está incluso o resultado dos clássicos) quando a estimação dos coeficientes se der no domínio do tempo sem a presença de ruído. Nas Tabelas 3.8 a 3.11 são apresentadas as medidas SDR, SIR e SAR, em dB, resultantes de simulações com sinais de voz em inglês e japonês, sequência de notas de instrumentos musicais metálicos, e sinal de áudio, respectivamente, com e sem a utilização da matriz de branqueamento nas misturas.

Algoritmo	Pré-Bran.	SDR	SIR	SAR
SCAm-t	Sim	87,13	87,13	246,51
SCAm-t	Não	14,09	14,09	240,35
SCAc-t	Sim	93,77	93,77	248,90
SCAc-t	Não	34,85	34,85	238,72

Tabela 3.8: Avaliação dos Algoritmos SCAm-t e SCAc-t com e sem Pré-Branqueamento para Sinais de Voz em Inglês.

Algoritmo	Pré-Bran.	SDR	SIR	SAR
SCAm-t	Sim	86,30	86,30	247,74
SCAm-t	Não	14,23	14,23	242,69
SCAc-t	Sim	92,11	92,11	247,58
SCAc-t	Não	28,35	28,35	241,39

Tabela 3.9: Avaliação dos Algoritmos SCAm-t e SCAc-t com e sem Pré-Branqueamento para Sinais de Voz em Japonês.

Pode-se observar nas Tabelas 3.8 e 3.9 que o uso da matriz de branqueamento resultou em ganhos significativos para sinais de voz. O mesmo não é observado

Algoritmo	Pré-Bran.	SDR	SIR	SAR
SCAm-t	Sim	90,47	90,47	221,07
SCAm-t	Não	82,66	82,66	221,04
SCAc-t	Sim	92,40	92,40	221,18
SCAc-t	Não	86,10	86,10	221,16

Tabela 3.10: Avaliação dos Algoritmos SCAm-t e SCAc-t com e sem Pré-Branqueamento para Sequências de Notas Musicais de Instrumentos Metálicos.

Algoritmo	Pré-Bran.	SDR	SIR	SAR
SCAm-t	Sim	2,68	2,68	235,91
SCAm-t	Não	1,41	1,41	235,63
SCAc-t	Sim	1,31	1,31	234,90
SCAc-t	Não	1,69	1,69	234,75

Tabela 3.11: Avaliação dos Algoritmos SCAm-t e SCAc-t com e sem Pré-Branqueamento para Sinais de Áudio.

nas Tabelas 3.10 e 3.11 para misturas de sequências de notas de instrumentos musicais, cujas fontes naturalmente já apresentam um bom grau de esparsidade no domínio temporal, e para sinais de áudio, cujas fontes apresentam um baixo grau de esparsidade no domínio temporal. Nesse último caso, o emprego da matriz de branqueamento na etapa de pré-processamento não é suficiente para compensar a não esparsidade das fontes no domínio temporal.

Para ilustrar o efeito da matriz de branqueamento nos histogramas de DOA em sinais de voz (dois locutores em inglês, voz masculina e feminina, e um terceiro locutor em japonês voz masculina), na Fig. 3.26, são mostrados os histogramas de DOA com branqueamento (a) e sem branqueamento (b). Pode-se verificar na Fig. 3.26(a) que os picos do histograma estão localizados nas posições corretas, ao passo que na Fig. 3.26(b) somente um pico está corretamente localizado. Além de corrigir o posicionamento dos picos, o efeito ocasionado pela matriz de branqueamento dos sinais das misturas reduziu o espalhamento dos valores dos DOAs, resultando em um número maior de ocorrências nos picos verdadeiros.

Experimento 2

Neste experimento avaliaremos o desempenho do algoritmo SCAm no domínio T-F com as três funções objetivo, propostas nesta tese, nos 30 diferentes cenários. Os resultados obtidos pelo algoritmo SCAcTF (que não utiliza função objetivo para obter a solução, uma vez que não emprega o desacoplamento entre os sinais dos sensores) são também apresentados, para fins de comparação. A Fig. 3.27 apresenta os

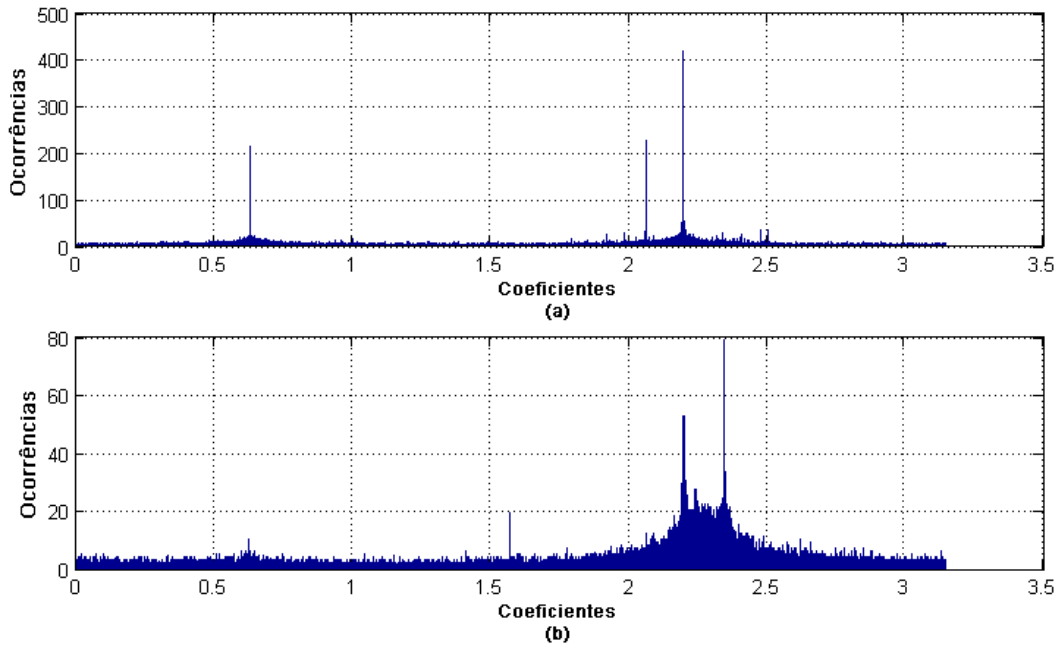


Figura 3.26: Histogramas de DOA para Sinais de Voz.

valores de SDR (em dB) obtidos pelos dois algoritmos com sinal de voz em inglês. Conforme pode ser visto nas Figs. 3.27(a) e (b), as funções objetivo \mathcal{F}_{CEF} e \mathcal{F}_{KEF}

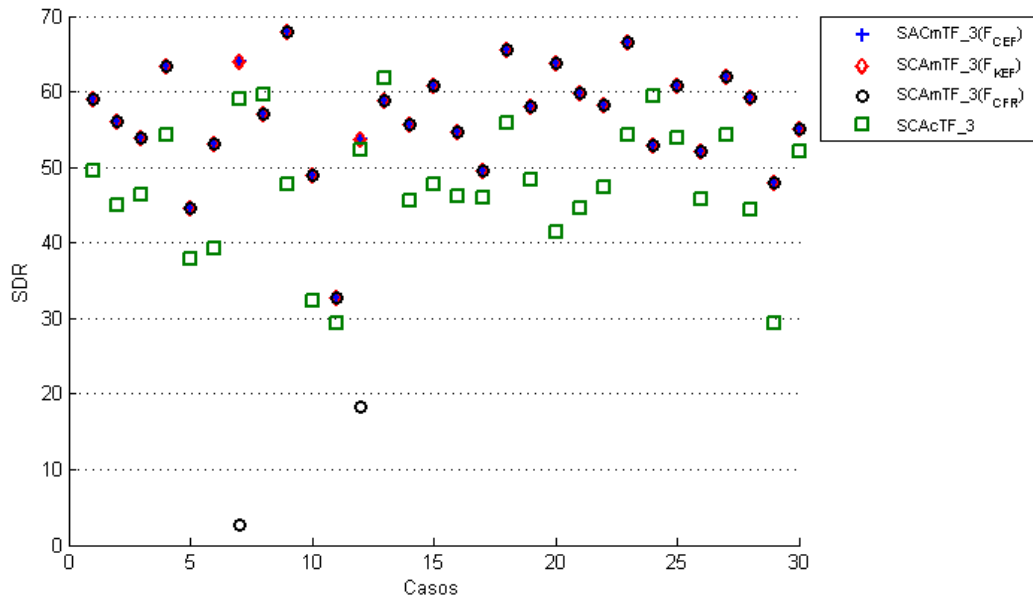


Figura 3.27: SDR (em dB) do Algoritmo SCAmTF para Diferentes Funções Objetivo e do Algoritmo SCAcTF com Sinal de Voz em Inglês para Cada Cenário.

apresentaram resultados muito semelhantes, indicando que, para sinais de voz, não existe uma vantagem clara entre minimizar a correlação mútua ou maximizar a cur-

toose. Podemos observar na Fig. 3.27(c) que o método baseado na função objetivo \mathcal{F}_{CFR} apresentou problemas nos Cenários 7 e 12, o que pode ser explicado pelo fato do mascaramento usar apenas duas misturas, e nem sempre fornecer estimativas das fontes satisfatórias, independentemente de quão boa seja a estimativa dos coeficientes da matriz de mistura. Por esse motivo, essa técnica deve somente ser usada pelo algoritmo SCAm quando as fontes não apresentarem características definidas em relação à curtose nem baixa correlação, pois trata-se de um mascaramento cego.

A Fig. 3.28 apresenta os resultados do algoritmo SCAmTF com as 3 funções objetivo distintas e do algoritmo SCAcTF para sinais de áudio (flautas) nos 30 cenários distintos. Conforme pode ser visto nessa figura, o emprego da função \mathcal{F}_{KEF}

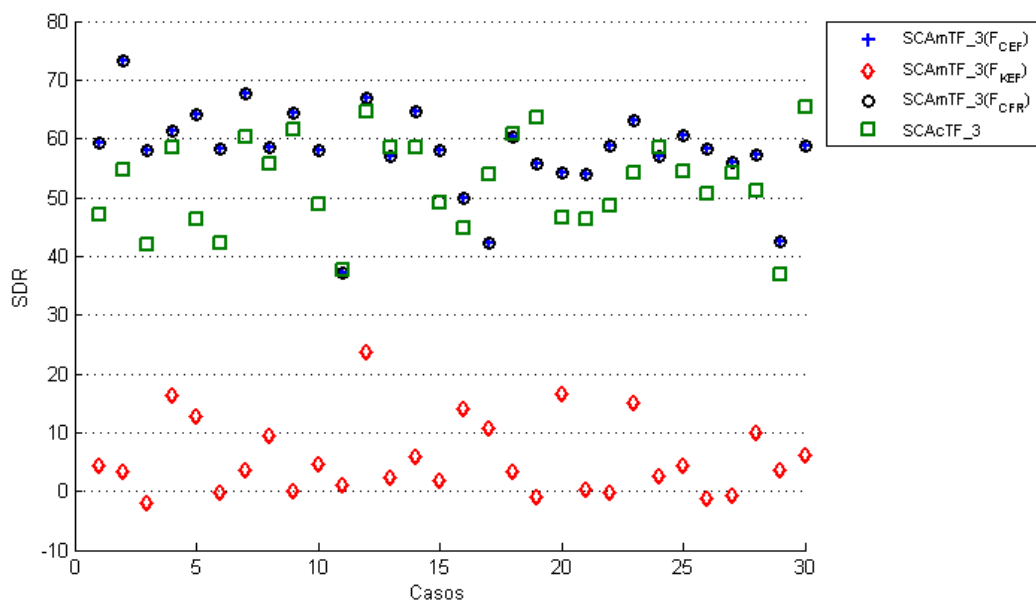


Figura 3.28: SDR (em dB) do Algoritmo SCAmTF para Diferentes Funções Objetivo e do Algoritmo SCAcTF com Sinal de Áudio (Flautas) para Cada Cenário.

resultou em valores muito baixos de SDR, como esperado, uma vez que sinais de áudio não apresentam valores de curtose altos, como visto na Tabela A.1. Para esse conjunto de sinais as funções \mathcal{F}_{CEF} e \mathcal{F}_{CFR} apresentaram resultados semelhantes.

Das Figs. 3.27 e 3.28, pode-se observar que exceto para sinais de áudio com a função \mathcal{F}_{KEF} , as soluções obtidas pelo algoritmo SCAmTF apresentaram SDRs próximas das do algoritmo SCAcTF, com uma ligeira vantagem do algoritmo SCAmTF.

Experimento 3

Neste experimento, avaliaremos os desempenhos dos algoritmos SCAmTF e SCAcTF em função do tipo de filtro aplicado ao vetor de características. Na Fig. 3.29,

apresentamos os valores da SDR obtidos com os filtros de Norma, SSP e *Outlier*, para sinais de voz em inglês. Conforme pode se observado nessa figura, o filtro de

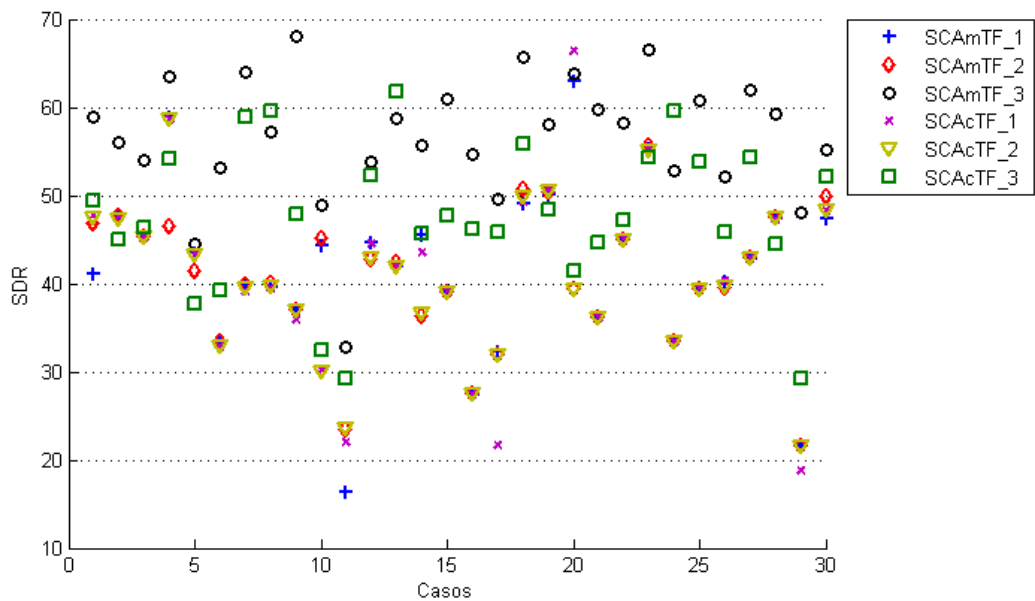


Figura 3.29: SDR (em dB) dos Algoritmos SCAmTF e SCAcTF com Diferentes Filtragens do Vetor de Características para Sinal de Voz em Inglês.

Norma e o filtro SSP apresentaram resultados semelhantes, inferiores aos do filtro de *Outlier*.

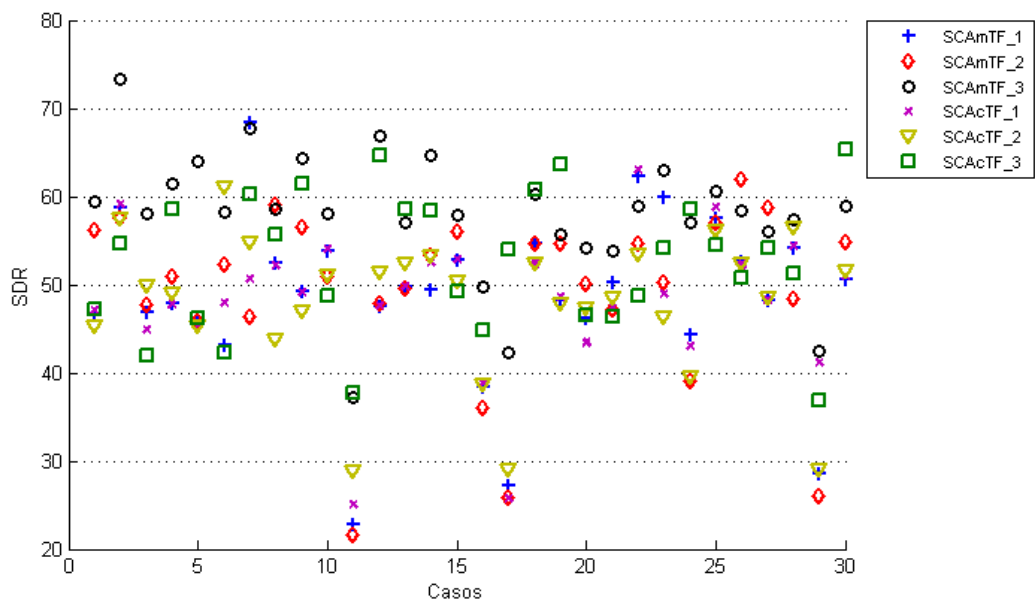


Figura 3.30: SDR (em dB) dos Algoritmos SCAmTF e SCAcTF com Diferentes Filtragens do Vetor de Características para Sinal de Áudio (Flautas).

A Fig. 3.30 apresenta os resultados obtidos com os algoritmos SCAmTF e SCAcTF com as diferentes filtragens do vetor de características para sinal de áudio. Nesse caso os algoritmos apresentaram resultados mais próximos, sendo que o filtro de *Outlier* apresentou SDR com variância ligeiramente menor.

Experimento 4

A proposta deste experimento é ilustrar os desempenhos dos algoritmos SCAc e SCAm no domínio do tempo utilizando a moda (ou seja, o valor que detém o maior número de observações) da DOA, ao invés da detecção de picos. Dessa forma, cada coeficiente associado a um par fonte-sensor é estimado independentemente, através da moda das DOAs. Neste experimento foram usados 3 sinais de voz: uma masculina em inglês, uma feminina em inglês e uma masculina em japonês.

A Fig. 3.31 apresenta os resultados obtidos com a detecção de pico e com a moda para os algoritmos SCAm-t e SCAc-t, ambos usando filtro de *Outlier*. Os resultados obtidos pelos algoritmos JADE e SOBI foram também incluídos para fins de comparação. Dessa figura, pode-se observar que os algoritmos SCAm-t e

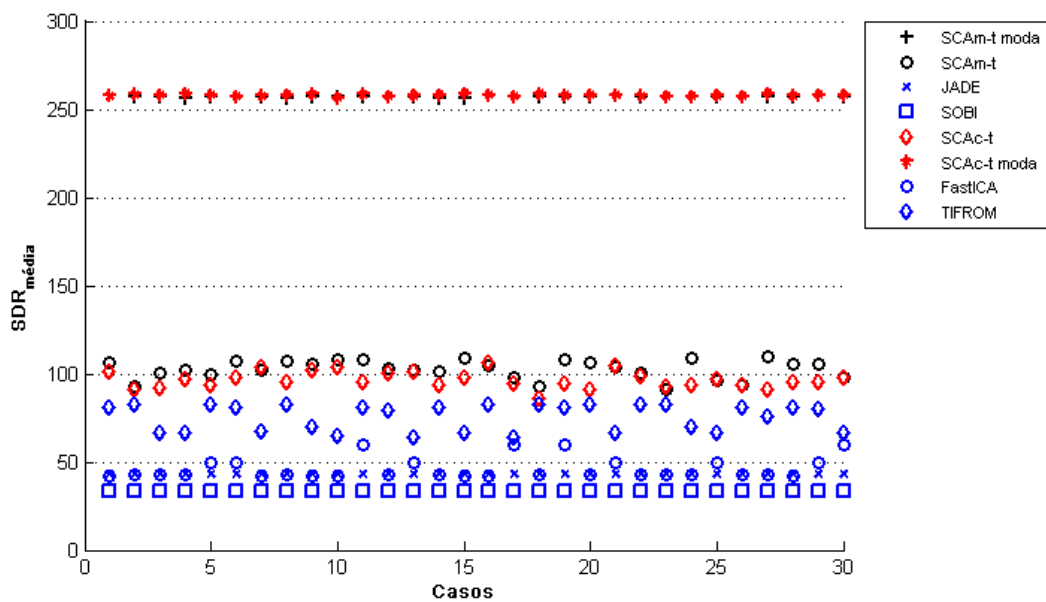


Figura 3.31: SDR (em dB) dos Algoritmos SCAm-t e SCAc-t com Emprego da Detecção de Pico e da Moda, e dos Algoritmos JADE e SOBI, para Sinais de Voz.

SCAc-t, com emprego da moda, combinado com o princípio de esparsidade e filtro de *Outlier*, apresentaram excelentes desempenhos para sinais de voz (supergaussianos) em cenários com poucas fontes, bastante superiores aos algoritmos clássicos. Infelizmente, esse ganho não foi verificado para sinais de áudio.

3.5 Resultados Gerais na Presença de Ruído

Nesta seção são apresentados os resultados obtidos com os métodos de estimação da matriz de mistura propostos e clássicos para sinais de voz e áudio na presença de ruído branco e *babble* com diferentes SNRs. Nas Tabelas 3.12 a 3.15 são apresentadas as medidas SDR, SIR e SAR, em dB, calculadas a partir dos 30 casos de misturas instantâneas formadas por 3 fontes, com valores de SNR variando de 5 a 25 dB, com passo de 5 dB.

A principal intenção de apresentar esses resultados, neste capítulo, é mostrar que os resultados obtidos por essas medidas, nem sempre acompanham os resultados do erro quadrático médio entre os elementos da matriz de mistura estimada e da matriz de mistura real. Isso ocorre devido ao erro introduzido pelo método de reconstrução produzir distorções distintas nas estimativas, de acordo com os sinais originais e/ou com a posição da coluna da matriz de mistura estimada onde ocorreram os erros. Outro motivo é que, neste trabalho, o ruído foi somado aos sensores e, portanto, após a separação, estará distribuído entre as estimativas, que intencionalmente foram comparadas com as estimativas originais, uma vez que não há estimativa original contaminada com ruído para ser usada como referência na obtenção da SDR, SIR e SAR. Portanto, as tabelas a seguir mostram, principalmente, o quanto os algoritmos testados são diferentemente afetados na reconstrução das estimativas na presença do ruído, independentemente da acurácia em que os coeficientes foram estimados.

Na Tabela 3.12 por exemplo, pode-se observar que os algoritmos JADE, FastICA e STUE apresentaram os melhores desempenhos em relação à SDR para $SNR = 5$ dB; no entanto, para a mesma SNR e com os mesmos sinais e ruídos, se observarmos os MSEs na Fig. 3.18 os algoritmos que apresentaram melhores desempenhos foram o STUE, o SCAzTF e o JADE, o que mostra que a SDR sofre influência da forma como a reconstrução das fontes é realizada. Métodos de reconstrução das fontes serão discutidos no Cap. 5.

Conforme se pode observar, os resultados obtidos na presença de ruído branco foram diferentes dos resultados obtidos na presença de ruído *babble*. O mesmo ocorreu quando mudamos o tipo de sinal, ou seja, voz e áudio.

Sinais de Voz	SNR = 05 dB			SNR = 10 dB			SNR = 15 dB			SNR = 20 dB			SNR = 25 dB			SNR = ∞		
	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
JADE	2,69	18,21	3,97	6,96	23,88	8,21	11,43	29,78	12,36	16,13	35,00	16,68	20,96	38,57	21,11	43,31	43,31	253,61
SOBI	0,55	8,61	5,04	4,13	10,28	9,96	7,90	13,31	14,70	11,85	18,12	19,20	17,22	24,88	23,43	33,71	33,71	253,65
TIFROM	-0,22	20,28	1,75	3,63	26,49	5,92	10,98	31,90	11,05	15,99	36,89	16,03	20,97	41,42	21,03	79,49	79,49	253,88
FastICA	2,60	18,34	4,10	6,93	24,10	8,27	11,41	29,62	12,45	16,09	34,86	16,73	20,96	38,96	21,14	53,98	53,98	253,64
STUE	0,88	20,31	2,08	6,13	26,21	6,61	11,07	31,59	11,36	15,94	36,43	16,09	20,76	40,88	20,98	69,33	69,33	253,53
SCAcTF_1	-2,07	10,58	3,76	2,69	17,54	8,35	6,99	25,78	13,01	14,42	34,33	17,75	20,29	38,95	21,64	53,11	53,11	253,23
SCAcTF_2	-0,77	10,35	4,36	3,08	12,41	9,02	6,49	22,63	13,23	11,36	28,64	17,90	16,55	32,64	21,69	47,64	47,64	253,44
SCAcTF_3	-2,15	8,95	3,83	2,30	15,34	8,21	6,72	23,42	13,20	12,53	30,79	17,69	17,99	36,68	22,21	53,37	53,37	253,40
SCAmTF_1	-1,45	10,24	3,30	2,00	12,17	7,97	7,24	24,64	12,25	14,48	33,69	16,85	20,42	38,61	20,97	50,91	50,91	253,31
SCAmTF_2	-0,53	10,35	3,00	3,37	15,31	7,46	8,07	20,65	12,80	11,43	24,48	17,56	15,18	29,15	22,78	45,82	45,82	253,37
SCAmTF_3	-1,68	10,37	3,33	1,98	15,35	7,91	7,56	25,57	12,10	14,44	34,78	16,67	20,49	41,26	21,03	60,32	60,32	253,81
SCAm-t	0,57	11,53	5,14	3,94	11,97	10,26	7,67	18,90	14,84	11,04	20,38	19,09	12,95	23,82	24,31	104,92	104,92	253,78
SCAc-t	-0,04	9,80	4,80	3,60	14,16	9,84	7,22	19,97	14,51	10,80	21,51	19,04	13,62	23,35	24,43	99,20	99,20	253,67
SCAc-t(modal)	-8,65	-1,02	-2,34	-5,90	-0,33	1,52	-3,27	0,03	6,11	-1,55	0,30	11,10	0,74	2,46	16,96	258,14	268,38	253,68
SCAm-t(modal)	0,04	8,44	4,40	3,70	11,83	9,33	7,80	16,21	13,62	10,55	20,37	18,26	12,81	17,45	23,37	258,32	268,96	253,77
SCAzTF	0,61	19,82	1,96	6,07	26,00	6,77	11,07	31,65	11,43	15,97	36,33	16,32	20,68	40,75	21,06	79,30	79,30	253,66

Tabela 3.12: Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Voz na Presença de Ruído Branco com Diferentes SNRs.

Sinais de Áudio	SNR = 05 dB			SNR = 10 dB			SNR = 15 dB			SNR = 20 dB			SNR = 25 dB			SNR = ∞		
	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
JADE	2,14	10,80	4,66	6,36	15,50	8,87	10,98	20,90	12,89	15,94	26,52	17,17	20,93	31,80	21,77	51,53	51,53	255,17
SOBI	2,95	12,68	4,81	7,30	17,83	9,02	11,80	23,25	13,22	16,52	28,62	17,52	21,41	33,83	21,84	64,19	64,19	264,69
TIFROM	-0,40	11,45	3,00	5,34	17,68	6,98	11,11	23,56	11,83	16,31	28,79	16,57	21,30	33,72	21,56	71,06	71,06	263,96
FastICA	2,19	10,73	4,89	6,53	15,58	8,82	11,28	21,26	12,92	16,21	27,02	17,16	21,22	32,35	21,77	53,55	53,55	258,83
STUE	0,95	11,27	3,62	6,70	18,59	7,61	11,51	23,76	12,31	16,39	28,94	17,03	21,31	33,91	21,89	74,65	74,65	263,75
SCAcTF_1	-1,33	7,39	3,92	2,61	12,62	8,25	7,98	18,97	12,93	13,99	25,81	16,75	21,00	33,28	21,64	53,03	53,03	261,79
SCAcTF_2	-2,57	1,51	5,46	0,24	5,57	10,37	3,58	10,72	15,20	8,36	15,07	19,24	16,34	25,77	22,85	52,34	52,34	262,00
SCAcTF_3	-1,28	7,30	3,95	2,14	11,71	8,39	7,35	17,95	12,93	13,94	25,67	17,11	20,22	32,35	21,59	57,71	57,71	262,15
SCAmTF_1	-0,62	7,59	3,80	3,01	11,16	8,51	6,13	14,68	13,01	11,27	20,81	17,79	20,12	32,35	22,23	56,45	56,45	261,90
SCAmTF_2	-0,81	7,71	3,43	2,65	10,30	8,27	7,04	16,06	12,85	14,52	26,00	17,44	21,07	33,38	21,96	54,18	54,18	262,12
SCAmTF_3	-1,23	6,72	3,63	2,60	10,59	8,49	5,65	13,58	13,03	11,03	20,66	17,77	20,10	32,26	22,25	62,68	62,68	263,18
SCAm-t	-0,54	5,95	3,64	0,63	4,84	8,47	4,62	7,72	13,17	6,05	8,69	17,28	7,57	9,33	22,59	15,88	15,88	246,77
SCAc-t	-1,85	4,58	3,16	0,71	7,45	7,89	3,08	5,42	11,65	3,86	6,01	15,51	8,83	11,84	21,69	6,46	6,46	239,25
SCAc-t(modal)	-7,36	-0,64	-1,01	-5,22	-0,29	2,43	-3,28	-0,26	6,99	-1,67	0,10	11,88	-0,67	0,32	16,84	0,68	0,68	235,50
SCAm-t(modal)	-1,26	5,05	3,84	1,89	6,78	8,22	4,46	8,62	13,13	5,10	8,06	17,43	8,20	9,81	22,45	15,48	15,48	248,19
SCAzTF	0,64	10,89	3,71	6,66	18,44	7,61	11,50	23,56	12,32	16,36	28,49	17,13	21,25	33,30	21,99	65,30	65,30	263,15

Tabela 3.13: Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Áudio na Presença de Ruído Branco com Diferentes SNRs.

Sinais de Voz	SNR = 05 dB			SNR = 10 dB			SNR = 15 dB			SNR = 20 dB			SNR = 25 dB			SNR = ∞		
	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
JADE	13,09	25,89	15,89	16,96	33,00	18,56	21,21	35,34	22,61	25,55	38,28	26,64	29,54	40,31	31,18	43,31	43,31	253,61
SOBI	7,86	14,59	16,27	12,63	20,48	20,15	18,63	26,76	23,98	22,16	29,64	28,36	25,27	31,19	33,25	33,71	33,71	253,65
TIFROM	10,94	31,39	15,94	15,91	35,91	17,09	20,86	39,98	22,35	26,06	44,95	26,19	31,09	49,46	31,19	79,49	79,49	253,88
FastICA	12,75	25,88	16,44	16,64	32,45	18,81	20,97	36,11	22,65	25,57	39,67	26,98	29,94	41,70	31,23	53,98	53,98	253,64
STUE	10,49	28,39	12,89	15,68	33,30	16,90	20,81	38,19	21,47	25,19	41,93	26,17	29,69	48,36	31,13	69,33	69,33	253,53
SCAcTF_1	9,67	25,41	18,01	14,52	29,80	22,49	20,25	36,08	22,01	24,82	39,61	26,01	28,94	42,27	31,16	53,11	53,11	253,23
SCAcTF_2	8,31	26,11	15,33	14,31	32,16	16,02	18,13	34,03	21,98	22,34	36,22	26,86	26,62	38,25	31,83	47,64	47,64	253,44
SCAcTF_3	10,22	27,13	20,11	13,17	32,36	17,71	20,41	38,35	21,90	25,15	41,58	26,99	30,39	43,91	31,19	53,37	53,37	253,40
SCAmTF_1	9,21	19,73	18,33	12,58	23,26	23,39	14,98	33,30	21,86	24,23	39,55	26,01	28,97	42,44	31,13	50,91	50,91	253,31
SCAmTF_2	5,39	18,40	11,77	11,57	25,47	15,73	16,37	30,09	20,43	20,88	32,95	25,22	26,29	38,24	31,04	45,82	45,82	253,37
SCAmTF_3	8,46	15,15	18,44	9,75	23,71	21,44	15,23	34,28	22,63	25,58	44,39	26,09	29,98	45,56	31,22	60,32	60,32	253,81
SCAm-t	6,68	16,74	19,82	11,42	23,60	25,05	11,39	19,05	25,95	15,32	23,65	31,25	22,14	33,32	35,92	104,92	104,92	253,78
SCAc-t	6,62	18,75	21,78	11,81	22,70	26,93	12,22	16,49	27,58	18,08	22,10	32,01	19,32	26,04	35,27	99,20	99,20	253,67
SCAc-t(modal)	-2,28	2,80	9,34	2,53	5,88	14,91	9,43	11,60	17,20	7,72	8,29	23,79	10,12	10,40	28,22	258,14	268,38	253,68
SCAm-t(modal)	7,94	15,87	18,19	9,96	21,93	21,74	11,71	16,09	23,98	14,63	20,26	28,81	17,49	22,16	32,84	258,32	268,96	253,77
SCAzTF	10,62	29,00	13,35	15,36	34,40	16,74	20,12	38,54	21,84	25,58	42,55	26,93	30,52	47,53	31,36	79,30	79,30	253,66

Tabela 3.14: Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Voz na Presença de Ruído *Babble* com Diferentes SNRs.

Sinais de Voz	SNR = 05 dB			SNR = 10 dB			SNR = 15 dB			SNR = 20 dB			SNR = 25 dB			SNR = ∞		
	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
JADE	13,84	19,56	19,26	18,02	25,58	21,86	21,76	29,56	26,29	25,28	34,07	30,89	28,42	37,14	34,37	51,53	51,53	255,17
SOBI	13,94	20,08	19,26	19,14	28,14	22,51	23,87	34,76	26,41	28,48	38,37	30,74	32,86	43,00	34,47	64,19	64,19	264,69
TIFROM	13,90	22,65	23,30	18,46	30,49	28,24	23,29	35,75	30,94	28,49	40,59	30,75	33,38	44,93	36,17	71,06	71,06	263,96
FastICA	13,61	19,36	18,78	17,49	25,29	21,62	22,75	30,85	26,02	26,50	34,49	30,27	24,65	36,90	34,43	53,55	53,55	258,83
STUE	13,64	24,43	18,96	17,30	29,60	23,73	23,08	35,72	27,40	28,44	40,89	32,46	33,53	46,31	35,81	74,65	74,65	263,75
SCAcTF_1	11,73	20,53	22,49	14,49	24,43	27,25	22,34	32,95	29,26	27,88	38,16	29,55	32,22	41,55	33,69	53,03	53,03	261,79
SCAcTF_2	8,15	18,06	18,17	12,40	21,18	23,47	19,29	29,44	23,43	23,53	33,22	27,56	29,07	38,00	32,11	52,34	52,34	262,00
SCAcTF_3	12,11	20,48	22,69	13,31	24,06	26,73	22,07	32,59	29,47	26,98	37,94	30,40	32,49	42,40	34,58	57,71	57,71	262,15
SCAmTF_1	7,88	14,07	21,01	10,90	16,78	25,73	20,26	30,50	28,57	26,86	36,05	29,74	31,61	40,05	34,65	56,45	56,45	261,90
SCAmTF_2	8,33	18,09	20,21	11,59	22,90	24,35	22,33	33,04	27,14	27,01	36,96	31,42	31,44	41,02	34,59	54,18	54,18	262,12
SCAmTF_3	8,00	14,42	21,36	8,18	16,50	24,80	20,43	29,74	28,00	27,33	39,05	29,82	32,24	42,51	34,66	62,68	62,68	263,18
SCAm-t	3,50	6,92	11,58	1,91	5,61	17,23	6,99	15,88	19,63	5,87	7,52	30,93	14,64	19,35	32,41	15,88	15,88	246,77
SCAc-t	1,44	4,47	15,43	2,03	4,60	17,72	4,31	6,69	22,61	6,02	9,61	27,97	7,52	8,48	28,63	6,46	6,46	239,25
SCAc-t(modal)	-0,78	2,66	11,60	-0,85	1,39	14,15	-0,50	0,82	17,95	-0,30	0,48	22,84	-0,12	0,34	28,04	0,68	0,68	235,50
SCAm-t(modal)	2,72	7,10	14,32	3,63	7,58	16,22	4,88	7,25	20,67	5,18	6,29	26,42	7,85	9,07	33,72	15,48	15,48	248,19
SCAzTF	12,61	19,02	19,93	17,13	26,67	23,78	22,59	33,12	27,19	27,71	38,31	32,43	33,02	44,38	36,32	65,30	65,30	263,15

Tabela 3.15: Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Áudio na Presença de Ruído *Babble* com Diferentes SNRs.

Na Tabela 3.12, ao analisarmos a presença de ruído branco nos sinais de voz, verificamos na coluna com $\text{SNR} = 5$ dB, que os algoritmos que apresentaram melhores desempenhos, ou seja, que foram menos afetados pela presença do ruído, em relação à SDR foram, respectivamente, o JADE, o FastICA, o STUE, o SCAzTF, o SCAM-t e o SOBI, enquanto os demais algoritmos ficaram com valores de SDR negativos ou bem próximos de zero. Ainda para $\text{SNR} = 5$ dB, tanto o algoritmo STUE quanto o algoritmo SCAzTF apresentaram desempenhos superiores ao obtido pelo algoritmo TIFROM, que serve de paradigma de comparação nesta tese para os algoritmos que buscam os coeficientes da matriz de mistura nas regiões de baixa variância. Portanto, as hipóteses de desacoplamento dos sensores e de uso das estimativas na busca das regiões de baixa variância não foram negadas neste cenário. Deve-se ressaltar que os resultados aqui presentes são dependentes da etapa de reconstrução das fontes, o que não é a melhor abordagem, uma vez que a reconstrução é um passo posterior à estimação dos coeficientes.

É possível também observarmos, ainda na mesma tabela que, em relação à SDR, tanto o algoritmo STUE quanto o algoritmo SCAzTF para $\text{SNR} \leq 15$ dB apresentaram, consistentemente, resultados ligeiramente superiores aos obtidos pelo algoritmo TIFROM, o que demonstra uma tendência a favor do desacoplamento dos sensores para esse cenário com o aumento do ruído. Da mesma forma, se compararmos os resultados em relação à SDR, obtidos pelos algoritmos que empregam o desacoplamento dos sensores, SCAMTF_1, SCAMTF_2 e SCAMTF_3, contra os algoritmos respectivos que não empregam o desacoplamento dos sensores, SCAcTF_1, SCAcTF_2 e SCAcTF_3, verificamos que, para todas as SNRs, sempre um algoritmo que emprega o desacoplamento obteve a melhor solução.

Em relação à SIR, na Tabela 3.12, para $\text{SNR} = 5$ dB, os algoritmos que apresentaram melhores desempenhos foram o STUE, o TIFROM, o SCAzTF, o FastICA e o JADE, sendo que os demais apresentaram valores inferiores a 15 dB. É interessante observar que o resultado obtido pelo algoritmo STUE foi ligeiramente melhor do que o obtido pelo algoritmo TIFROM (o que não ocorreu para o algoritmo SCAzTF) e que o resultado obtido pelo algoritmo STUE sempre foi ligeiramente melhor do que o resultado obtido pelo algoritmo SCAzTF (exceto para $\text{SNR} = \infty$), o que mostra uma tendência positiva para o uso das estimativas na busca das regiões de baixa variância em relação à SIR.

Na Tabela 3.13, ao analisarmos a SDR na presença de ruído branco em misturas com sinais de áudio, verificamos na coluna da $\text{SNR} = 5$ dB que os algoritmos que apresentaram os melhores desempenhos foram, respectivamente, o SOBI, o FastICA, o JADE, o STUE e o SCAzTF, tendo os demais algoritmos apresentado resultados negativos. Já em relação à SIR os algoritmos que apresentaram melhores desempenhos foram, respectivamente, o SOBI, o TIFROM, o STUE, o SCAzTF, o JADE e

o FastICA, tendo os demais algoritmos apresentado valores inferiores a 10 dB.

Nesse cenário, para a SDR, o algoritmo STUE sempre apresentou resultados superiores ao algoritmo TIFROM. Já em relação à SIR, apresentou resultado superior para todos os valores de SNR, exceto para $\text{SNR} = 5$ dB. Já o algoritmo SCAzTF, em relação à SDR, apresentou resultados superiores ao algoritmo TIFROM para $\text{SNR} \leq 20$ dB, sendo que o mesmo não ocorreu em relação à SIR, já que foi superior ao TIFROM somente para $\text{SNR} = 10$ dB.

Na comparação entre o algoritmo SCAmTF_2 e o algoritmo SCAcTF_2, podemos novamente observar a tendência favorável ao desacoplamento dos sensores, uma vez que o algoritmo SCAmTF_2 sempre apresentou resultados superiores ao algoritmo SCAcTF_2, tanto para SDR quanto para SIR. Essa tendência, nesse cenário, não foi observada para os demais algoritmos da classe SCAmTF, como pode ser verificado comparando-se os algoritmos SCAmTF_3 e SCAmTF_1 com os algoritmos SCAcTF_3 e SCAcTF_1, respectivamente.

Na Tabela 3.14, ao analisarmos a presença de ruído *babble* com sinais de voz, verificamos na coluna da $\text{SNR} = 5$ dB, que os algoritmos que apresentaram melhores desempenhos em relação à SDR foram, respectivamente, o JADE, o FastICA, o TIFROM, o SCAzTF, o STUE e o SCAcTF_3, sendo que os demais algoritmos ficaram com valores de SDR abaixo de 10 dB. Já os algoritmos que apresentaram melhores desempenhos em relação à SIR foram, respectivamente, o TIFROM, o SCAzTF, o STUE, o SCAcTF_3, o SCAcTF_2 e o JADE, tendo os demais (excetuando os algoritmos FastICA e SCAcTF_1) um desempenho inferior a 20 dB. Logo, podemos inferir que o desacoplamento não se mostrou vantajoso na forte presença de ruído *babble*, uma vez que o algoritmo TIFROM foi melhor que os algoritmos STUE e SCAzTF tanto para SDR quanto para SIR e que os algoritmos da classe SCAcTF foram melhores que os algoritmos da classe SCAmTF.

O algoritmo SCAmTF_2 não apresentou resultados superiores ao algoritmo SCAcTF_2, nesse cenário. No entanto, para $\text{SNR} = \infty$, o algoritmo SCAmTF_3 obteve o melhor desempenho em relação à SDR e à SIR entre todos os algoritmos da classe SCAcTF.

Na Tabela 3.15, ao analisarmos a presença de ruído tipo *babble* com sinais de áudio, verificamos na coluna da $\text{SNR} = 5$ dB, que os algoritmos que obtiveram os melhores desempenhos em relação à SDR foram, respectivamente, o SOBI, o TIFROM, o JADE, o STUE, o FastICA e o SCAzTF, tendo os demais algoritmos (com exceção do algoritmo SCAcTF_3) apresentado desempenhos inferiores a 12 dB. Os algoritmos que obtiveram os melhores desempenhos em relação à SIR foram, respectivamente, o STUE, o TIFROM, o SCAcTF_1, o SCAcTF_3, o SOBI e o JADE, tendo os demais algoritmos (com exceção do FastICA e do SCAzTF) resultados inferiores a 19 dB.

Ao compararmos o algoritmo TIFROM com o algoritmo STUE, verificamos a mesma tendência observada na Tabela 3.14, ou seja, de que a redução da presença de ruído *babble* favorece o algoritmo STUE (conforme se pode observar nos resultados obtidos tanto para a SDR quanto para a SIR) a partir da $SNR = 25$ dB.

Podemos destacar que o desacoplamento presente no algoritmo SCAzTF não foi suficiente para garantir um desempenho superior em relação ao algoritmo TIFROM. No entanto, a combinação do desacoplamento com o uso das estimativas na busca das regiões de baixa variância permitiu ao algoritmo STUE apresentar um bom desempenho. De fato, o algoritmo STUE, para $SNR = \infty$ até $SNR = 25$ dB, foi o algoritmo com o melhor desempenho entre todos os algoritmos avaliados nesta tese para este cenário.

Ao analisarmos a SAR na condição de ausência de ruído, percebe-se que todos os algoritmos apresentaram excelentes resultados, tanto para sinais de voz quanto para sinais de áudio. Isso era de certa forma esperado, uma vez que o método de reconstrução efetuado para esses casos foi a inversa da matriz de mistura estimada, e como os coeficientes foram bem estimados e a mistura não possui ruído, deduz-se que as estimativas não apresentarão artefatos.

Na condição presente na Tabela 3.12 percebemos que os algoritmos, na presença de ruído branco, que apresentaram menos artefato foram o SCAM-t, o SOBI e o SCAC-t. Cabe ressaltar que, esses algoritmos não são baseados numa transformação T-F. Contudo, se observarmos a Tabela 3.14 para $SNR = 5$ dB, contendo os mesmos sinais contaminados com ruído *babble*, verificamos que novamente os algoritmos SCAM-t e SCAC-t apresentam desempenhos superiores em relação à SAR.

Finalmente, na Tabela 3.13, contendo sinais de áudio contaminados com ruído branco, podemos observar que os algoritmos com melhores desempenhos para $SNR = 5$ dB foram o SCACTF_2, o FastICA e o SOBI. De fato, para este cenário, o algoritmo SCACTF_2 sempre apresentou o melhor desempenho em relação à SAR. Quando os sinais de áudio estão contaminados com ruído *babble*, conforme os resultados mostrados na Tabela 3.15, percebemos que o algoritmo TIFROM quase sempre obteve o melhor desempenho nas SNRs, salvo para o caso com $SNR = 20$ dB e para o caso onde não havia presença de qualquer ruído. Cabe destacar que para os sinais de áudio, os algoritmos da classe SCACTF sempre apresentaram bons resultados, sendo que somente os algoritmos SCAM-t, SCAC-t, SCAC-t (moda) e SCAM-t (moda) apresentaram resultados muito ruins. Tais desempenhos ruins já eram esperados, uma vez que esses algoritmos necessitam que os sinais contenham alguma esparsidade temporal, o que praticamente não ocorre nos sinais de áudio testados.

Capítulo 4

Métodos de Estimação da Matriz de Mistura para o Caso Anecoico

Nas misturas anecoicas os sinais provenientes de uma dada fonte chegam aos diversos receptores em momentos distintos. Assim, as misturas contêm versões atrasadas de cada fonte, com atrasos possivelmente diferentes. Essa forma de mistura é comum quando os receptores se encontram mais afastados das fontes e em ambientes onde não há anteparos que possam propiciar reflexões ou reverberações dos sinais. Na ausência de ruído, esse tipo de fenômeno é descrito pela equação:

$$\begin{bmatrix} x_1(t) \\ \vdots \\ x_m(t) \end{bmatrix} = \begin{bmatrix} a_{11}\delta(t - d_{11}) & \dots & a_{1n}\delta(t - d_{1n}) \\ & \vdots & \\ a_{m1}\delta(t - d_{m1}) & \dots & a_{mn}\delta(t - d_{mn}) \end{bmatrix} * \begin{bmatrix} s_1(t) \\ \vdots \\ s_n(t) \end{bmatrix} \quad (4.1)$$

onde $x_j(t)$ é a j -ésima mistura, $s_k(t)$ é a k -ésima fonte, e a_{jk} e d_{jk} são, respectivamente, os coeficientes de atenuação e atraso da fonte k na mistura j , e “*” representa a operação de convolução.

As abordagens investigadas nesta tese para a reconstrução foram baseadas em dois caminhos distintos, ou seja, com o emprego de máscaras ou partir de técnicas de subtração temporal. Em ambas, na maioria das vezes, parte-se de uma estimativa inicial de solução (obtida, por exemplo, via mascaramento) visando a geração de novas estimativas que, potencialmente, possam superar, em qualidade, as estimativas iniciais.

Este capítulo está dividido em quatro seções. A Seção 4.1 contempla os métodos propostos para misturas anecoicas. A Seção 4.2 apresenta os resultados de erro quadrático médio na ausência de ruído, enquanto que a Seção 4.3 apresenta os resultados para casos onde ruídos encontram-se presentes. Por fim, a Seção 4.4 exhibe os resultados obtidos pelos métodos de estimação propostos neste capítulo combinados com o mascaramento [23] como forma de reconstrução das fontes. Tal

combinação será avaliada por meio das medidas objetivas SDR, SIR e SAR.

4.1 Algoritmos Propostos para Misturas Anecoicas

Foram elaborados, originalmente, dois algoritmos para misturas anecoicas, o STUE e o IMUE, além de uma variação do algoritmo clássico AD-TIFROM-CF [24] denominado ATIFROM. O algoritmo STUE foi também generalizado para o cenário instantâneo. Ao se pesquisar na literatura algoritmos específicos para esse cenário, verificou-se um número bem menor de propostas, em relação aos desenvolvidos para misturas instantâneas.

4.1.1 Algoritmo ATIFROM

O algoritmo ATIFROM obtém os coeficientes de atraso e atenuação de modo sequenciado. Contudo, ele mantém um casamento na estimação dos pares atenuação e atraso da mesma forma que o TIFROM Anecoico (TIFROM_Anec, Seção 2.5.3). Esse acoplamento evita o inconveniente de uma busca num espaço bidimensional, se valendo, na maioria dos casos, da técnica de SSZ¹ para estimação dos coeficientes de atenuação e atraso.

Assim, o algoritmo ATIFROM, empregado para estimar os coeficientes de atenuação, é baseado no TIFROM_Anec, podendo empregar uma transformada esparificadora qualquer (como STFT, wavelet, CQT ou DFT). Porém, o algoritmo proposto não se confunde com o AD-TIFROM-CF, pois, diferentemente deste, contém uma etapa intermediária de reconstrução das estimativas das fontes para cada razão das misturas, com o intuito de estimar o atraso das fontes.

Na Fig. 4.1 apresentamos o diagrama de blocos das etapas do algoritmo ATIFROM.

Da mesma forma que o algoritmo TIFROM_Anec, o algoritmo estima um coeficiente de atenuação a partir do ILD descrito na Eq. (2.28). Assim, pelo mascaramento, é reconstruído um par de estimativas das fontes, a partir de cada uma das misturas. Com isso são obtidas duas estimativas, \hat{y}_{1k} e \hat{y}_{2k} , para cada fonte, ou seja

$$\hat{y}_{1k}(t) = \hat{s}_k(t) \quad (4.2)$$

$$\hat{y}_{2k}(t) = \hat{\alpha}_k \hat{s}_k(t - \hat{\Delta}_k) \quad (4.3)$$

¹Pode-se também combinar histograma ITD (*Interaural Time Difference*) ou ILD (*Interaural Level Difference*, [44]) para estimar os coeficientes iniciais de atraso ou atenuação.

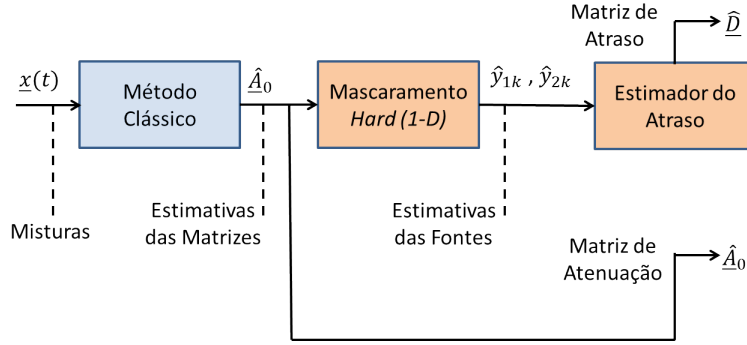


Figura 4.1: Esquema Geral do Algoritmo ATIFROM.

onde $\hat{\alpha}_k$ representa o valor de atenuação e $\hat{\Delta}_k$ o atraso da k -ésima fonte na segunda estimativa. Tais parâmetros são estabelecidos pelo processo de reconstrução das fontes via mascaramento binário bidimensional.

Ao se reproduzir a mesma fonte com e sem atraso a partir das misturas, torna-se possível estimar o coeficiente de atraso (posteriormente à estimação dos coeficientes de atenuação) por um processo especializado, mantendo-se a vantagem de poder continuar estimando de forma casada os coeficientes de atenuação. Outra característica dessa abordagem reside na possibilidade da aplicação de transformadas distintas nos processos de estimação dos coeficientes de atenuação e de atraso.

Discutiremos a seguir o uso de bancos de filtros como forma alternativa de se esparsificar o sinal além do uso da STFT [24], wavelet [31] e CQT [32], que são as transformadas usualmente encontradas na literatura. Isso possibilita projetos de bancos de filtros específicos, uniformes e não uniformes, para sinais dos quais se possua algum conhecimento prévio das raias de frequência mais significativas, ou para situações onde se queira avaliar somente sinais que possuam determinadas frequências de interesse. Por exemplo, no caso de sistemas específicos de identificação automática conhecidos na área de *WARFARE* como SEI (*Specific Emitter Identification*) [89], [90], é comum ter-se conhecimento prévio do conjunto de frequências e do tipo de modulação empregado; ou ainda no caso de sinais de voz, se sabe de antemão que o sinal possui uma maior energia em baixas frequências.

Banco de Filtros

Tendo em vista as considerações pregressas, além da STFT, avaliaremos os seguintes banco de filtros como transformada esparsificadora para facilitar a estimação dos coeficientes da matriz de mistura:

1. Banco de filtros DFT uniforme (DFTU) [91];
2. Banco de filtros DFT não uniforme (DFTNU) [92, 93];

3. Banco de filtros *Warped* DFT (WDFT) [94, 95].

O banco de filtros DFTU foi implementado a partir de sua decomposição polifásica com fator de decimação igual ao número de canais ($M = 128$), conforme a topologia apresentada na Fig. 4.2. As funções $E_i(z)$ representam as componentes polifásicas da função de transferência de um filtro protótipo passa-baixa [91]. Na Fig. 4.3 é apresentada a resposta em frequência dos filtros que compõem um banco de filtros DFTU.

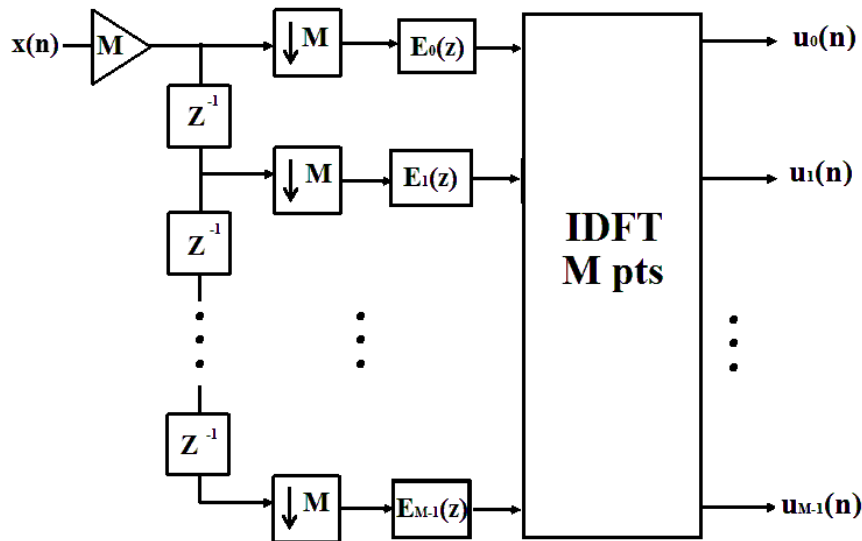


Figura 4.2: Diagrama de um Banco de Filtros DFT Polifásico.

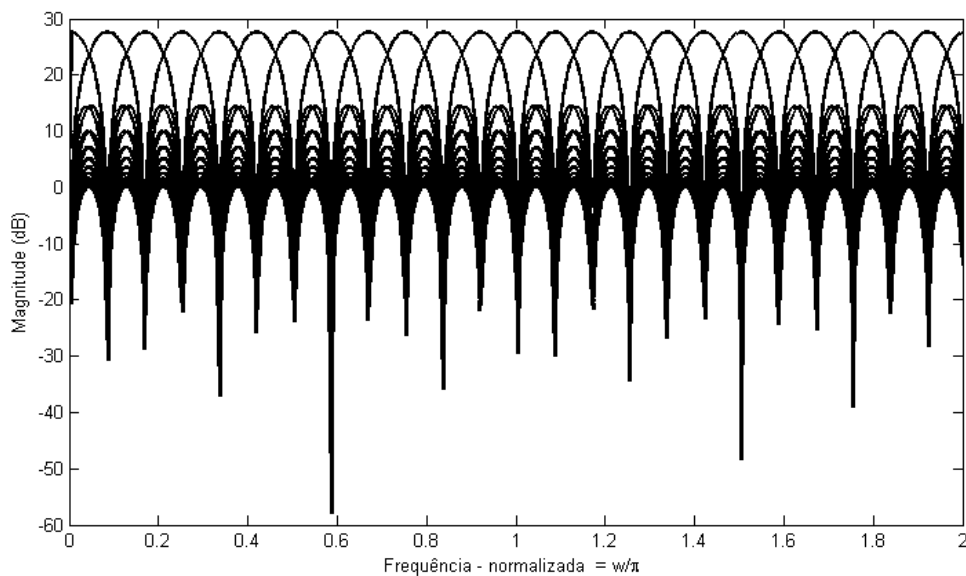


Figura 4.3: Resposta em Frequência dos Filtros de um Banco DFT Uniforme.

O banco de filtros DFTNU, baseado em [92, 93], seguiu a topologia apresentada na Fig. 4.4 usando a mesma razão entre o fator de decimação e o número de canais ($M = 128$). Nesta figura, $G(z)$ é a função de transferência de um filtro passa-tudo de primeira ordem, dada por:

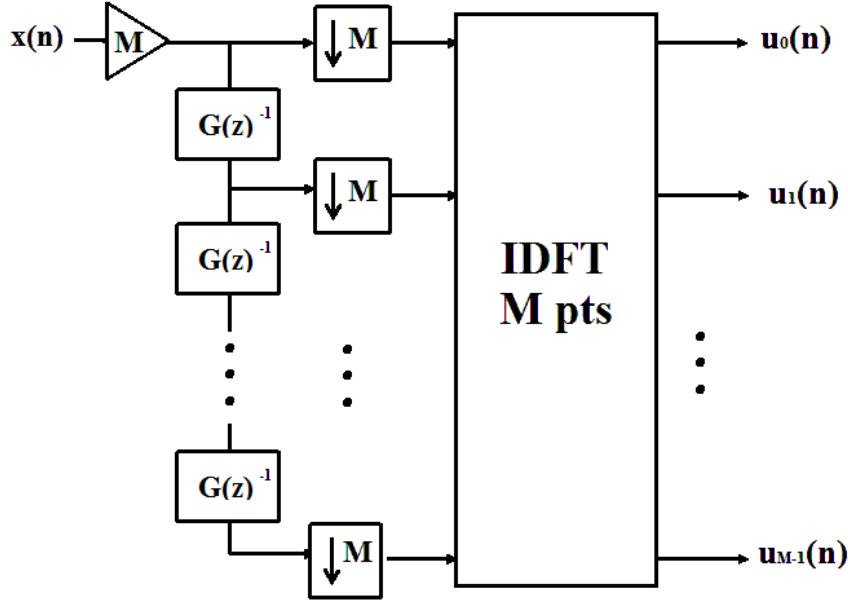


Figura 4.4: Diagrama de um Banco de Filtros DFTNU Polifásico.

$$G(z) = \frac{z - \lambda}{1 - \lambda z} \quad (4.4)$$

A Fig. 4.5 apresenta a resposta em frequência desse banco de filtros (com 128 canais) para o caso de $\lambda = 0,5$, no qual se pode observar a melhoria da resolução em baixas frequências.

Nos testes realizados, usou-se fator λ variável com a taxa de amostragem, dado por

$$\lambda = 0,1957 - 1,048 \sqrt{\frac{2}{\pi} \tan^{-1}\left(0,07212 \times \frac{f_s}{1000}\right)} \quad (4.5)$$

de sorte a aumentar a resolução em baixas frequências.

O banco de filtros DFT não uniforme *warped* baseado em [94, 95], aqui denominado WDFT, apresenta a topologia mostrada na Fig. 4.6, contendo o mesmo número de canais que as topologias anteriores ($M = 128$).

Na Fig. 4.7 é apresentada a resposta em frequência do banco de filtros WDFT, para $\lambda = -0,5$, apresentando melhores resoluções em baixas frequências. A principal diferença que essa topologia apresenta em relação à anterior, é a introdução de *gaps* entre as bandas passantes dos filtros.

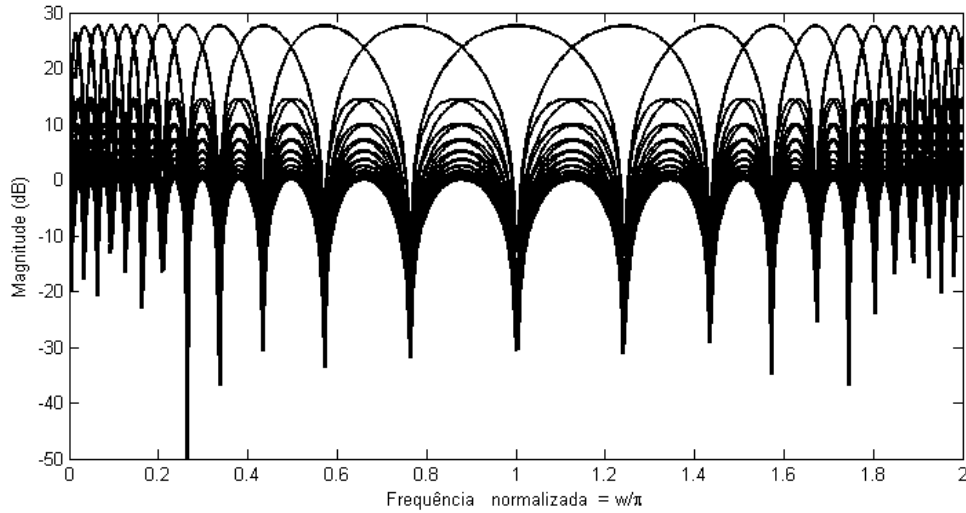


Figura 4.5: Resposta em Frequência do Banco de Filtros DFTNU - $\lambda = 0, 5$.

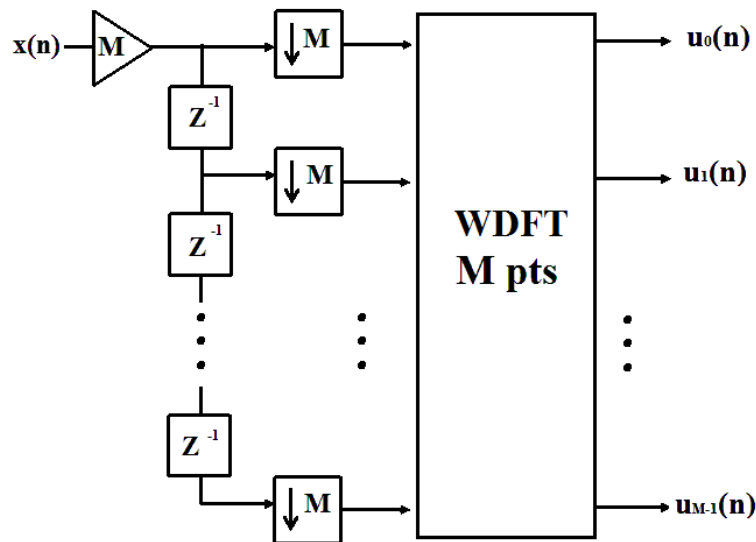


Figura 4.6: Estrutura de um Banco de Filtros WDF com Componentes Polifásicas.

Estimação do Atraso

A estimação do atraso foi feita, nesta tese, de duas formas distintas. Na primeira delas, usou-se um algoritmo específico que, partindo dos pares de fontes estimadas, \hat{y}_1 e \hat{y}_2 , estima os coeficientes de atraso. O algoritmo calcula a correlação entre dois sinais em função do atraso (vide Fig. 4.1), retornando o *lag* onde ocorre o máximo da correlação. Dada a possibilidade do atraso ser fracionário, efetua-se a reamostragem dos sinais para uma taxa mais alta, antes de se calcular a correlação cruzada. Após determinado o ponto (*lag*) de máximo, o valor do atraso é obtido dividindo-o pelo fator de *upsampling* empregado na reamostragem, escolhido arbitrariamente. A

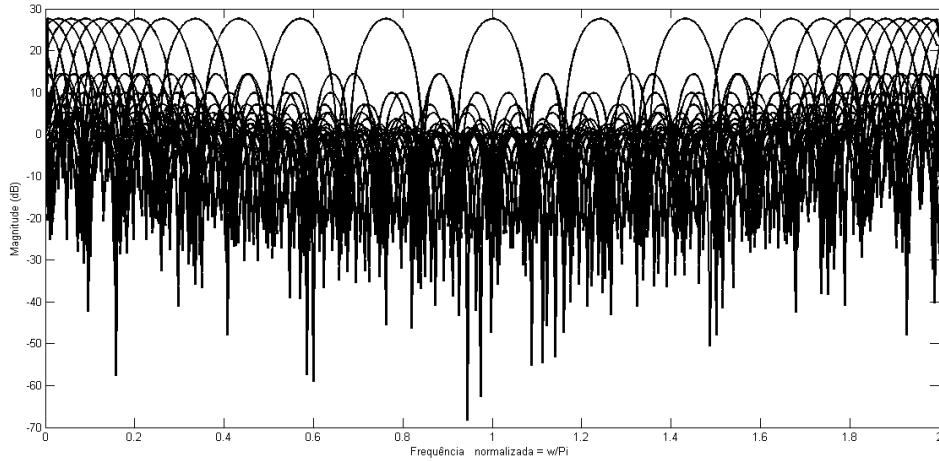


Figura 4.7: Resposta em Frequência do Banco de Filtros WDF com $\lambda = -0,5$.

Fig. 4.8 ilustra o gráfico de correlação gerado pelo algoritmo empregado a partir do método elaborado por [96], para um atraso de $-1,5$, onde reamostragem (*upsampling*) por fator 20 foi empregado.

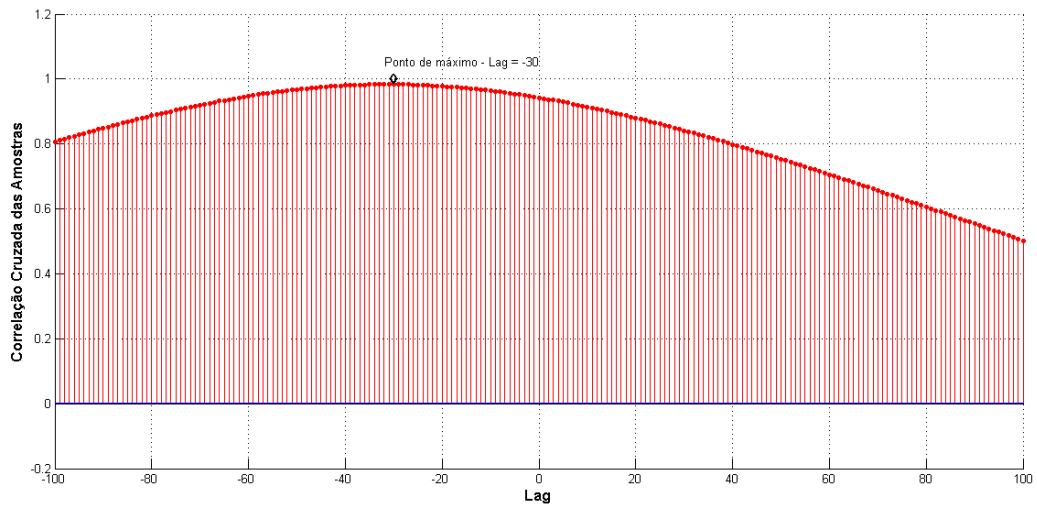


Figura 4.8: Exemplo de Determinação do Atraso por Meio do Método de Correlação Cruzada. Máximo em $Lag = -30$, Resultando em $Atraso = Lag / Upsampling = -1,5$.

A segunda abordagem se utiliza do segundo componente da Eq. (2.28) para calcular o atraso nas regiões previamente determinadas. Essa abordagem é exclusiva dos algoritmos TIFROM_AneC e DUETm.

O algoritmo proposto ATIFROM pode ser entendido como uma variante do TIFROM_AneC, no qual o coeficiente de atenuação é estimado exatamente da mesma forma, contudo os coeficientes de atraso são estimados pelo método da correlação.

Assim, nesse algoritmo, diferentemente dos demais propostos, a busca das regiões de baixa variância ocorre a partir do primeiro componente da Eq. (2.28), ou seja, os coeficientes são calculados diretamente das misturas, e não das estimativas das fontes.

4.1.2 Algoritmo STUE

O principal diferencial do algoritmo STUE consiste em explorar a esparsificação ocasionada pela reconstrução das estimativas das fontes obtidas pelo mascaramento binário bidimensional², e reestimar os coeficientes de atenuação e de atraso a partir dessas fontes estimadas, empregando alguns algoritmos especializados para calcular esses coeficientes.

Espera-se, ao aplicar essa técnica, que ocorra uma melhora porque, caso as estimativas das fontes obtidas das misturas sejam razoavelmente correlacionadas com as fontes originais, os coeficientes de atraso e atenuação serão mais facilmente estimados, devido à possibilidade maior das estimativas das fontes apresentarem mais regiões SSZ do que as misturas.

Portanto, a aplicação do método de busca de regiões de baixa variância fica facilitada, uma vez que será efetuada sobre as estimativas das fontes, a qual, são em geral, menos contaminadas por outras fontes, do que os sinais das misturas. Na Fig. 4.9 apresentamos o diagrama de blocos das etapas do algoritmo STUE.

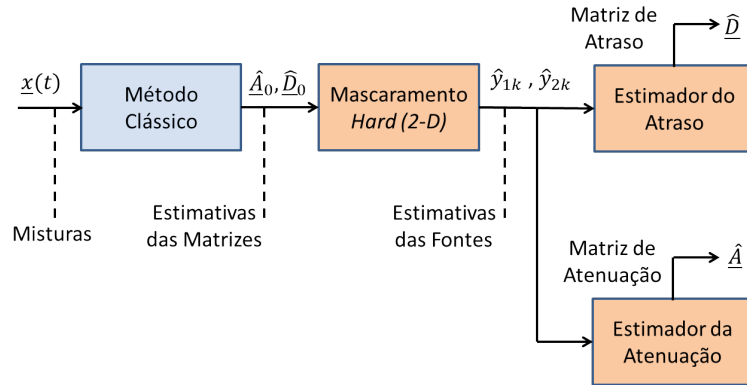


Figura 4.9: Esquema Geral do Algoritmo STUE.

O mascaramento bidimensional para se obter as estimativas \hat{y}_1 e \hat{y}_2 é feito através da minimização da função

$$score_i(\tau, \kappa) = \frac{1}{1 + \alpha_i^2} |\alpha_i e^{-j\kappa\delta_i} X_1(\tau, \kappa) - X_2(\tau, \kappa)|^2 \quad (4.6)$$

onde α_i e δ_i representam a atenuação³ e o atraso da i -ésima fonte na mistura 1

²Tal mascaramento é obtido a partir de matrizes iniciais de atenuação e atraso.

³Nesta abordagem, assume-se que as atenuações são positivas.

em relação à mistura 2, enquanto $X_1(\tau, \kappa)$ e $X_2(\tau, \kappa)$ representam as misturas no domínio T-F.

4.1.3 Algoritmo IMUE

A possibilidade de estimar os coeficientes, reconstruir os pares das fontes estimadas, e novamente reestimar os coeficientes (e assim sucessivamente), levou à implementação de um terceiro algoritmo, denominado nesta tese de IMUE [97]. Tal algoritmo parte de um par de matrizes iniciais de atenuação (\underline{A}_0) e atraso (\underline{D}_0), seguindo o fluxo de decisão descrito na Fig. 4.10.

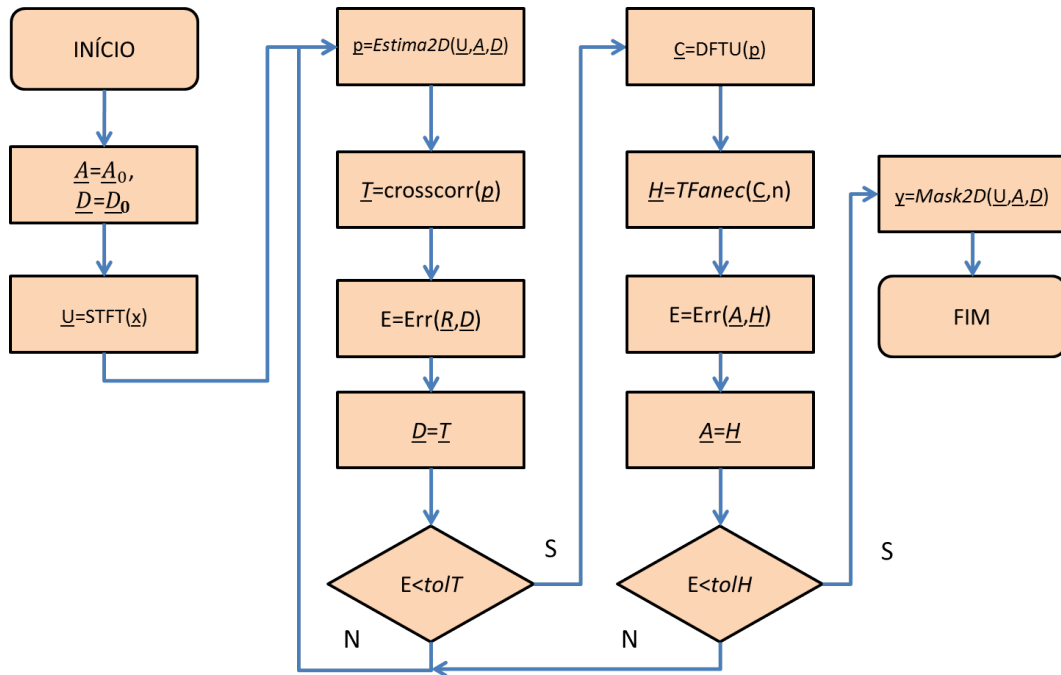


Figura 4.10: Fluxo de Decisão do Algoritmo IMUE.

Nesse fluxograma, \underline{T} e \underline{D} representam matrizes de atraso e \underline{A} e \underline{H} representam matrizes de atenuação. A matriz \underline{T} é obtida pelo método de correlação descrito no item Estimação do Atraso na Seção 4.1.1. O algoritmo *Estima2D* representa um mascaramento binário bidimensional, e o algoritmo *TFAnec* representa a busca das regiões de baixa variância aplicada ao vetor \underline{C} obtido na etapa anterior. Esse algoritmo (TFAnec) é parte do algoritmo TIFROM [22], sendo reaproveitado pelo IMUE para o caso anecoico. Contudo, neste caso, fez-se a modificação de realizar a busca das regiões de baixa variância sobre as componentes ILD, no lugar da razão das misturas que normalmente é adotada no caso instantâneo.

A função $Err(\cdot)$ calcula a soma do erro quadrático dos elementos de duas matrizes (a estimada no passo atual e a estimada no passo anterior), assumindo que as matrizes têm somente as colunas fora da posição correta. É por meio da minimização

do erro quadrático entre os coeficientes das linhas das matrizes a serem comparadas que se obtém a ordem correta das colunas de uma das matrizes. Para isso, a matriz a ter suas colunas reposicionadas tem os coeficientes de cada uma de suas linhas permutados, gerando linhas candidatas que representam possibilidades distintas de arranjo das colunas. Em seguida, calcula-se a soma do erro quadrático entre os coeficientes da linha de uma matriz com cada linha candidata associada àquela linha da outra matriz. Assim, ao minimizarmos esses erros quadráticos de uma dada linha, teremos um provável reagrupamento das colunas. Por fim, estabelece-se que o reagrupamento correto das colunas é obtido pelo arranjo que forneceu o menor erro quadrático entre todas as linhas. A matriz a ter as colunas reposicionadas é rearrumada conforme o arranjo estabelecido pela linha que ofereceu o menor erro quadrático entre todas. Finalmente, o erro quadrático entre todos os coeficientes das duas matrizes é calculado. $TolT$ e $TolH$ representam tolerâncias pré-estabelecidas para a diferença entre iterações sucessivas relativas às matrizes de atraso e de atenuação, respectivamente.

A principal vantagem dessa abordagem se dá em situações nas quais as estimativas iniciais da matriz de atraso e atenuação estão distantes dos seus valores reais, desde que o processo de reconstrução consiga gerar estimativas das fontes correlacionadas com as fontes originais. Assim, caso o mascaramento gere pares de fontes estimadas mais próximas das fontes originais, o processo iterativo tende a convergir.

Para a estimação inicial dos coeficientes de atenuação e atraso, nesta tese se empregará, em algumas abordagens, o método `TIFROM_Anec`, com uma transformada esparsificadora. No entanto, esses coeficientes podem ser obtidos por algum outro método ou mesmo serem arbitrados de modo aleatório, como já visto em uma das abordagens apresentadas.

Assim, esse método não exige que as fontes sejam WDO (*W-disjoint orthogonal*), nem que as misturas possuam regiões SSZ no domínio da transformada. Contudo, ele é dependente do mascaramento binário, do qual se espera que reconstrua estimativas distintas e correlacionadas com as fontes originais. Afinal, ele exige que os pares das fontes estimadas possuam regiões SSZ com razões entre as estimativas distintas.

A seleção dos valores iniciais é crítica para esse algoritmo, já que uma escolha ruim pode gerar pares de fontes estimadas distintos com razões idênticas (ou muito próximas) nas suas regiões SSZ. Isso pode ocorrer caso as fontes estimadas possuam ainda muita interferência proveniente de outras fontes. Uma possibilidade para contornar essa restrição é comunicar as razões estimadas pelos pares já avaliados aos pares de fontes estimadas ainda não avaliados, de forma semelhante ao feito pelo `TIFROM`, evitando que se obtenham razões idênticas de fontes estimadas distintas.

A seguir, exemplificamos o algoritmo `IMUE` num cenário em que a hipótese considerada de um valor inicial próximo dos valores corretos não é satisfeita. Assim,

arbitram-se valores de atenuação e atraso sem o uso de qualquer algoritmo para se obter estimativas razoáveis. Dessa forma, podemos verificar a possibilidade do algoritmo convergir para valores próximos dos corretos, a despeito de uma escolha arbitrária inicial.

Um problema dessa abordagem reside na falta de garantia de convergência. Devido a esse problema, foram inseridas duas condições de paradas que não estão representadas no fluxograma da Fig. 4.10. A primeira condição estabelece um número máximo de iterações nas quais o processo deve convergir, e a segunda estabelece um número máximo de vezes consecutivas para que a condição $\underline{E} < TolD$ não seja satisfeita.

Utilizou-se um caso anecoico subdeterminado com 3 fontes e 2 misturas de voz em inglês, gerando misturas a partir dos coeficientes de atenuação $A = [0, 3 \ 1, 5 \ 2, 4]$ e atraso $D = [3, 5 \ -1, 5 \ 1, 5]$. O valor inicial empregado nos coeficientes de atenuação foi $A_0 = [1, 0 \ 2, 0 \ 3, 0]$ e de atraso $D_0 = [1, 0 \ 1, 0 \ 1, 0]$.

Os valores obtidos após 12 iterações foram $H = [0, 36 \ 1, 50 \ 2, 39]$ e $T = [3, 45 \ -1, 45 \ 1, 50]$, ou seja, próximos dos valores corretos. As Figs. 4.11 e 4.12 ilustram as evoluções dos coeficientes de atenuação e de atraso ao longo das iterações.

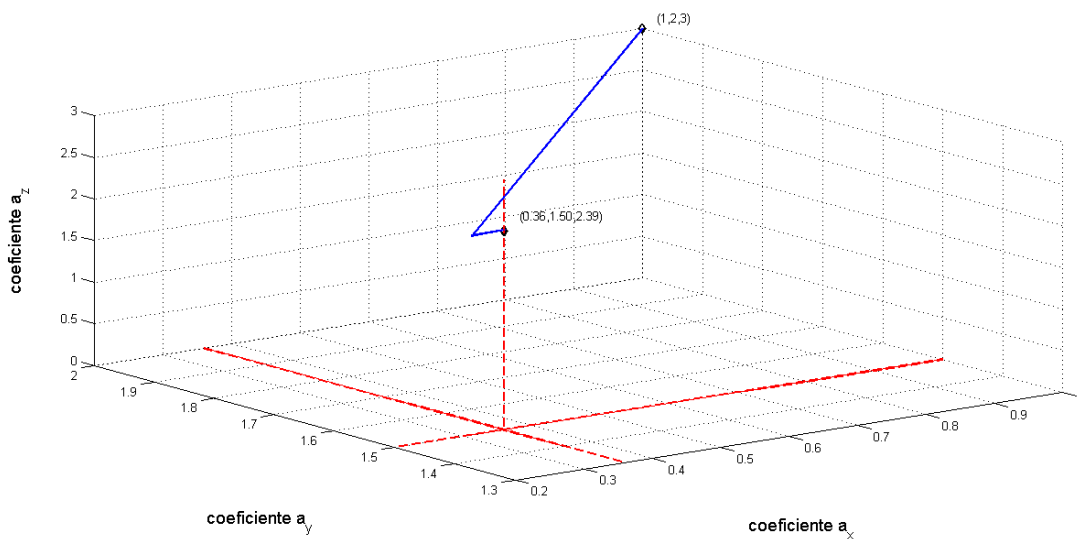


Figura 4.11: Evolução dos Coeficientes de Atenuação do Algoritmo IMUE.

Um procedimento empregado nas simulações para que a convergência fosse alcançada para quase todos os cenários foi empregar uma solução inicial razoavelmente próxima da solução correta, utilizando-se outra técnica (mais grosseira) para estimar as matrizes de atenuação e atraso.

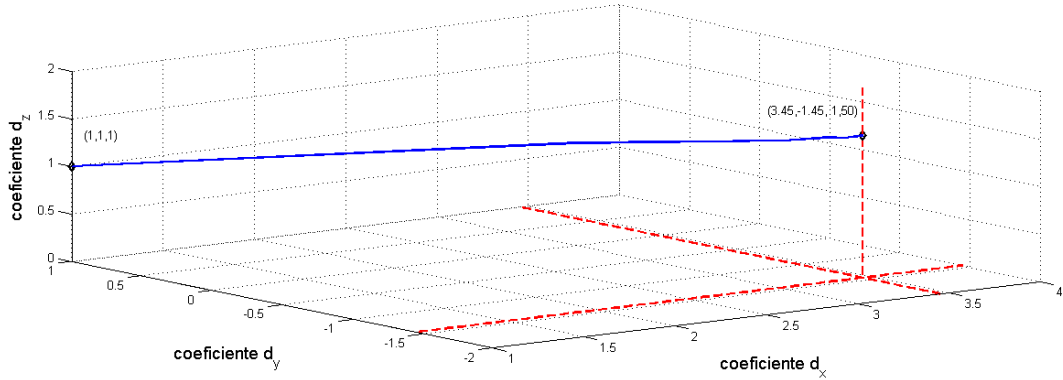


Figura 4.12: Evolução dos Coeficientes de Atraso do Algoritmo IMUE.

4.2 Desempenho dos Algoritmos Sem a Presença de Ruído

Nesta seção apresentaremos os resultados obtidos com os algoritmos propostos para o cenário subdeterminado de misturas anecoicas com 3 fontes e 2 sensores. As atenuações empregadas foram positivas e inferiores a 5, enquanto os atrasos ficaram restritos ao intervalo de -4 a $+4$ amostras. Portanto, a técnica foi exercitada para atrasos pequenos. No entanto, os métodos propostos nesta tese fazem *upsampling* por um fator 20 nas estimativas reconstruídas das fontes, para estimar os atrasos a partir de duas estimativas para cada fonte (uma com atraso e outra sem atraso). Portanto, as medidas são feitas sobre atrasos que variam de -80 a $+80$ amostras.

Os dados empregados nesta seção são compostos por dois conjuntos distintos: sinais de voz compostos por dois locutores (em inglês e um em japonês), de 10 s de duração e amostrados em 16 kHz [98], e sinais de áudio (flautas), com 2 s de duração e amostrados em 8 kHz [34], descritos no Apêndice A.

Novamente começaremos analisando os resultados obtidos pelos dois algoritmos de referência, o DUET modificado (DUETm) e o TIFROM_Anec (TIFROM_Anec). Por ser necessário obtermos estimativas para as matrizes de atenuação e de atraso de forma acoplada, geraremos dois gráficos de erro: um associado ao MSE dos coeficientes de atenuação e outro ao MSE dos coeficientes de atraso. Nas tabelas de estatísticas também apresentaremos os resultados para os dois conjuntos de coeficientes.

Os algoritmos de mascaramento binário bidimensional, presentes no IMUE e no STUE, e unidimensional, empregado pelo ATIFROM, utilizaram STFT com janela *Hamming* de 1024 amostras e salto de 256 amostras. A configuração usada pelo algoritmo DUETm foi a mesma, exceto pela janela empregada, que neste caso foi a *Hanning*, conforme estabelecido na fase de análise disponibilizado em [23]. Contudo,

a configuração usada para se fazer a busca das regiões de baixa variância no domínio T-F foi diferente, empregando-se a janela retangular de 128 amostras e superposição de 96 amostras.

As figuras a seguir mostram os erros médios quadráticos dos coeficientes de atenuação e de atraso dos algoritmos DUETm e TIFROM_Anec para misturas anecoicas com sinais de voz e áudio, respectivamente. Além disso, são representados dois limites, correspondentes a $\frac{1}{12}$ e $\frac{1}{3}$ do valor máximo do erro médio quadrático, visando facilitar a visualização da dispersão das soluções obtidas pelos algoritmos.

Pode-se observar que os algoritmos clássicos, implementados para o caso anecoico, apresentam uma dispersão bem maior do que os algoritmos clássicos disponibilizados para o caso instantâneo. Por esse motivo, se adotará o valor médio das soluções do DUETm como referência, já que a pior solução não caracteriza adequadamente uma solução válida.

Além disso, é possível observar que os valores de MSE da medida de atraso são piores que os da medida de atenuação, tanto no método do DUETm quanto no método do TIFROM_Anec.

Seguindo o procedimento, apresentaremos a seguir os resultados obtidos pelo algoritmo proposto ATIFROM, usando os três tipos de transformações esparsificadoras já mencionadas, ou seja, os bancos de filtros DFTU, DFTNU e WDFT. As Figs. 4.13 e 4.14 mostram esses resultados para sinais de voz e áudio, respectivamente.

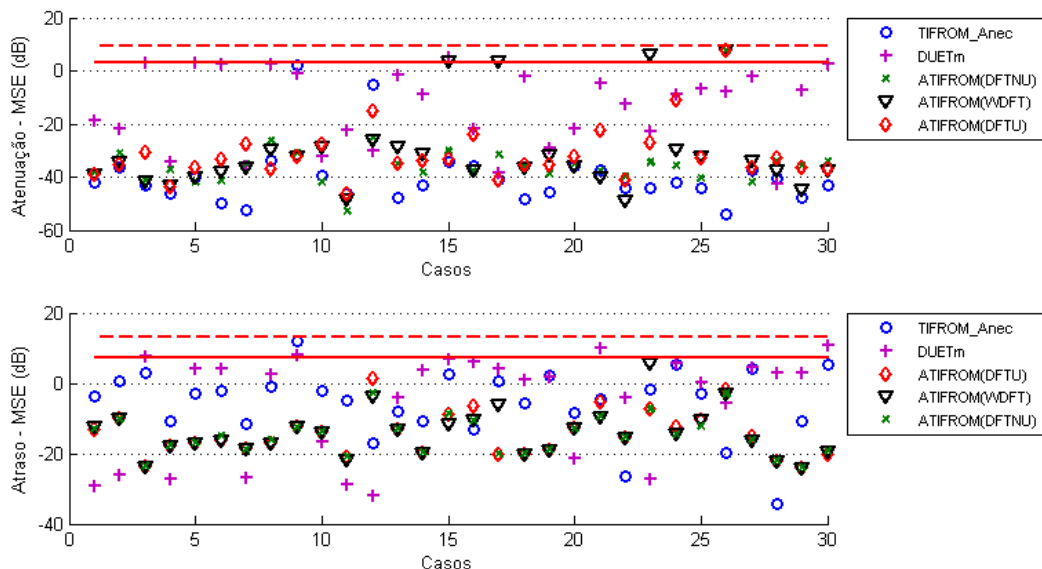


Figura 4.13: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM - Voz.

Nos algoritmos IMUE e STUE o termo entre parênteses representa o algoritmo

que forneceu os coeficientes da matriz de mistura inicial, ou seja, STUE(TIFROM_Anec) significa que o algoritmo empregou os coeficientes do algoritmo TIFROM_Anec como parâmetros iniciais em seu algoritmo. As Figs. 4.15 e 4.16 apresentam os resultados obtidos pelo algoritmo STUE, usando três soluções iniciais distintas, ou seja, fornecidas pelos algoritmos DUETm, TIFROM_Anec e ATIFROM(DFTU).

Por fim, apresentamos os resultados obtidos pelo algoritmo IMUE, usando as três soluções iniciais distintas. As Figs. 4.17 e 4.18 mostram esses resultados.

Das Figs. 4.13 a 4.18 podemos verificar que o algoritmo IMUE é superior ao algoritmo STUE, pois nota-se que, para as mesmas matrizes de inicializações, a dispersão ficou quase sempre menor, exceto para o atraso no caso das matrizes de inicializações provenientes do algoritmo ATIFROM. No entanto, mesmo para esse caso, os valores médios ficaram próximos dos do STUE. A principal desvantagem desse algoritmo é o custo computacional, uma vez que não há garantia de convergência com um número pré-definido de iterações.

As Tabelas 4.1 e 4.2 apresentam um resumo dos resultados obtidos, com os algoritmos propostos e os algoritmos de referência, nas misturas anecoicas avaliadas, para sinais de voz e para sinais de áudio, respectivamente.

		Atenuação - MSE	Atraso - MSE
Algoritmo		Média \pm Desvio	Média \pm Desvio
Voz	ATIFROM(DFTU)	$1,90 \times 10^{-1} \pm 1,04$	$7,35 \times 10^{-2} \pm 1,25 \times 10^{-1}$
	ATIFROM(DFTNU)	$1,94 \times 10^{-1} \pm 1,03$	$1,22 \times 10^{-1} \pm 2,74 \times 10^{-1}$
	ATIFROM(WDFT)	$4,85 \times 10^{-1} \pm 1,37$	$1,92 \times 10^{-1} \pm 6,63 \times 10^{-1}$
	STUE(DUETm)	$7,30 \times 10^{-1} \pm 1,47$	$1,09 \pm 2,37$
	STUE(TIFROM_Anec)	$8,73 \times 10^{-2} \pm 3,03 \times 10^{-1}$	$4,71 \times 10^{-1} \pm 2,12$
	STUE(ATIFROM(DFTU))	$5,79 \times 10^{-4} \pm 1,50 \times 10^{-3}$	$1,30 \times 10^{-3} \pm 9,58 \times 10^{-4}$
	IMUE(DUETm)	$1,75 \times 10^{-1} \pm 6,47 \times 10^{-1}$	$1,59 \times 10^{-1} \pm 5,31 \times 10^{-1}$
	IMUE(TIFROM_Anec)	$1,69 \times 10^{-2} \pm 9,25 \times 10^{-2}$	$5,12 \times 10^{-2} \pm 2,72 \times 10^{-1}$
	IMUE(ATIFROM(DFTU))	$4,54 \times 10^{-5} \pm 6,34 \times 10^{-5}$	$1,60 \times 10^{-3} \pm 2,50 \times 10^{-3}$
	DUETm	$5,52 \times 10^{-1} \pm 8,42 \times 10^{-1}$	$2,33 \pm 2,93$
	TIFROM_Anec	$6,39 \times 10^{-2} \pm 2,95 \times 10^{-1}$	$1,27 \pm 2,84$

Tabela 4.1: Estatísticas dos Algoritmos para Sinais de Voz.

Podemos verificar nas Tabelas 4.1 e 4.2 que o algoritmo IMUE(ATIFROM(DFTU)) apresentou destaque tanto em sinais de voz quanto em sinais de áudio, e que, tanto o algoritmo IMUE quanto o algoritmo STUE sempre apresentaram como pior solução aquela que empregou como solução original o DUETm, e que o algoritmo ATIFROM sempre apresentou como melhor solução aquela que empregou o banco de filtros DFTU.

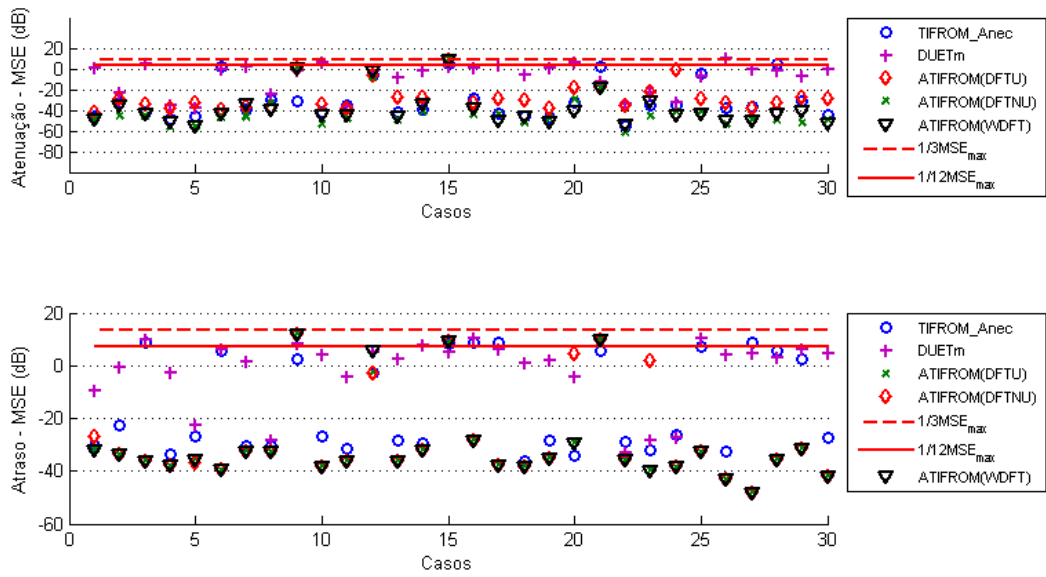


Figura 4.14: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM - Áudio.

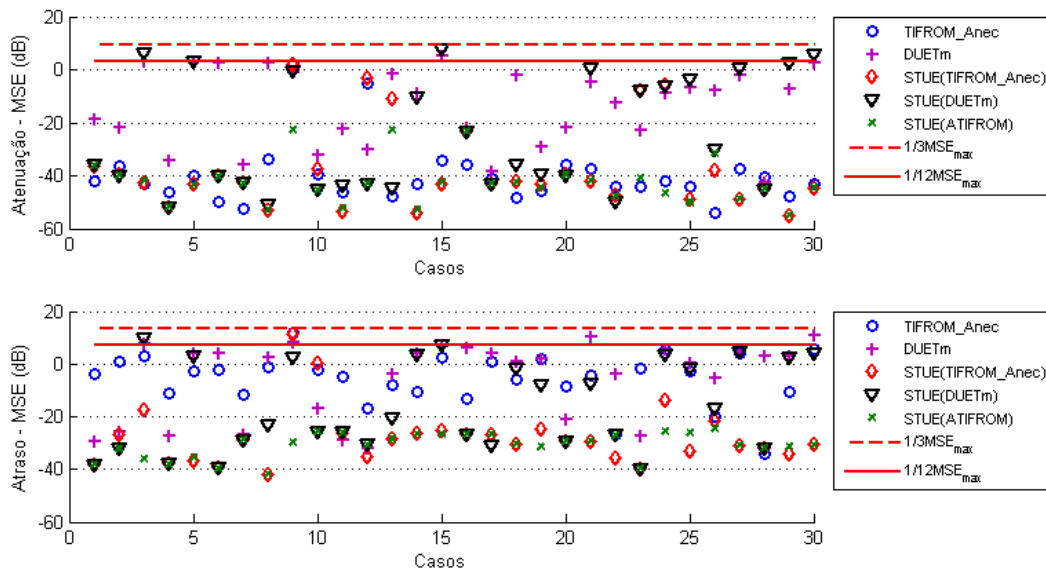


Figura 4.15: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE - Voz.

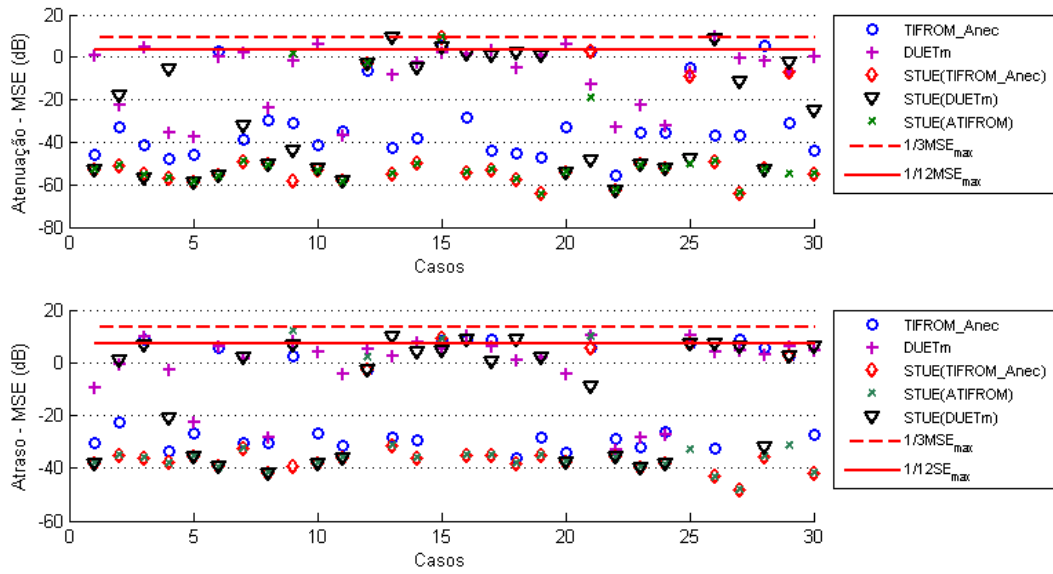


Figura 4.16: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE - Áudio.

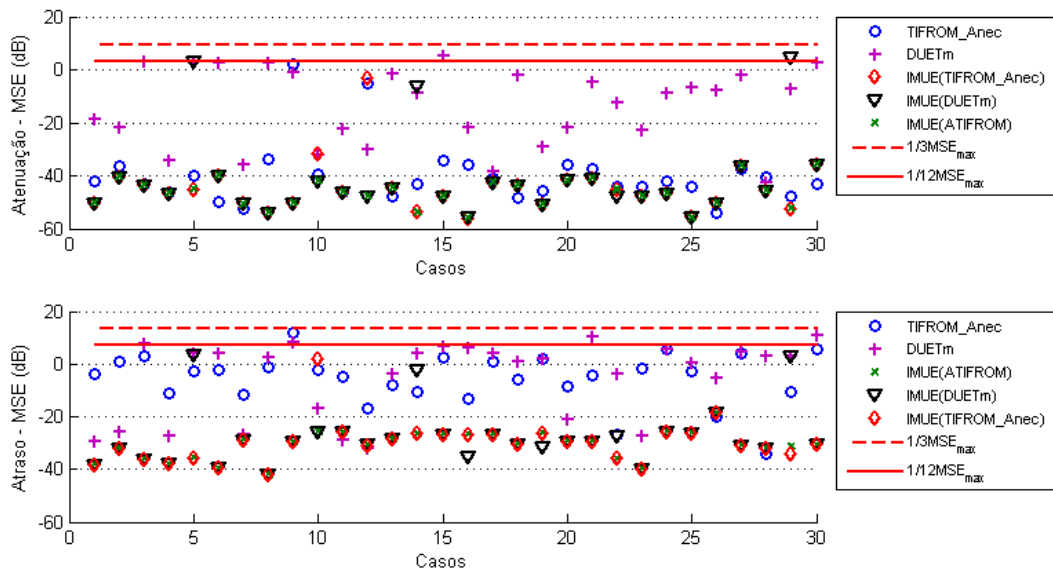


Figura 4.17: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE - Voz.

		Atenuação - MSE	Atraso - MSE
Algoritmo		Média \pm Desvio	Média \pm Desvio
Áudio	ATIFROM(DFTU)	$3,56 \times 10^{-1} \pm 1,66$	$1,14 \pm 3,57$
	ATIFROM(DFTNU)	$3,95 \times 10^{-1} \pm 1,66$	$1,29 \pm 3,57$
	ATIFROM(WDFT)	$3,71 \times 10^{-1} \pm 1,66$	$1,24 \pm 3,60$
	STUE(DUETm)	$8,77 \times 10^{-1} \pm 2,06$	$2,20 \pm 2,77$
	STUE(TIFROM_Anec)	$3,97 \times 10^{-1} \pm 1,67$	$6,46 \times 10^{-1} \pm 1,85$
	STUE(ATIFROM(DFTU))	$3,67 \times 10^{-1} \pm 1,66$	$1,17 \pm 3,56$
	IMUE(DUETm)	$8,86 \times 10^{-1} \pm 1,74$	$2,18 \pm 3,69$
	IMUE(TIFROM_Anec)	$3,62 \times 10^{-1} \pm 1,65$	$6,56 \times 10^{-1} \pm 1,85$
	IMUE(ATIFROM(DFTU))	$3,71 \times 10^{-1} \pm 1,16$	$1,92 \times 10^{-1} \pm 3,56$
	DUETm	$1,22 \pm 1,98$	$3,15 \pm 3,40$
TIFROM_Anec	$3,73 \times 10^{-1} \pm 9,49 \times 10^{-1}$	$1,87 \pm 2,78$	

Tabela 4.2: Estatísticas dos Algoritmos para Sinais de Áudio.

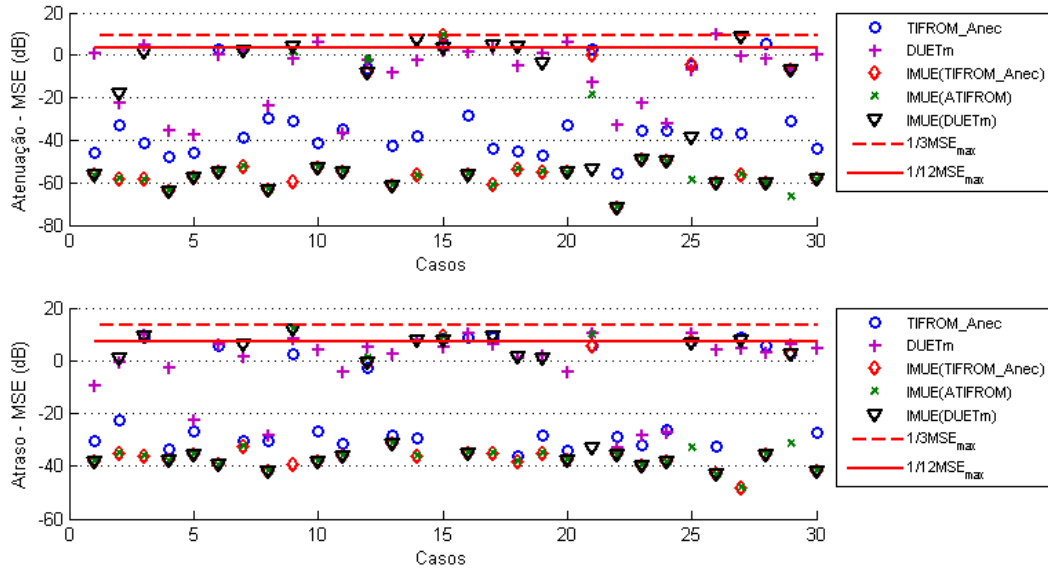


Figura 4.18: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE - Áudio.

4.3 Desempenho dos Algoritmos na Presença de Ruído

Nesta seção, apresentaremos os resultados da estimação dos coeficientes, obtidos com a presença de ruído nas misturas. Avaliaremos o efeito da presença do ruído em seis configurações: ATIFROM(DFTU), ATIFROM(DFTNU), STUE(TIFROM_Anec), STUE(ATIFROM), IMUE(TIFROM_Anec) e IMUE(ATIFROM), que apresentaram os melhores desempenhos nas simulações sem ruído, e ATIFROM(DFTNU)) para verificarmos se o comportamento apresentado que privilegiou o banco DFTU, na ausência de ruído, será preservado.

Como nas simulações realizadas sem a presença de ruído, os algoritmos de mascaramento binário bidimensional, presentes no IMUE e STUE, e unidimensional, empregado pelo ATIFROM, utilizaram STFT com janela *Hanning* de 1024 amostras e salto de 256 amostras. A configuração usada pelo algoritmo DUETm foi a mesma. E, da mesma forma, a configuração usada para se fazer a busca das regiões de baixa variância, no domínio T-F, foi uma janela retangular com 128 amostras e uma superposição de 96 amostras.

Nas tabelas apresentamos apenas os resultados obtidos pelo DUETm e pelo TIFROM_Anec, com SNR igual a 10 dB, para fins de comparação. Nas figuras, entretanto, apresentamos os resultado de duas inicializações distintas de cada algoritmo previsto.

4.3.1 Misturas com Ruído Branco nos Sensores

Nesta seção avaliaremos o comportamento dos algoritmos frente à presença de ruído branco. Da mesma forma como fora efetuado no Cap. 3, variamos o nível do ruído branco nas misturas com SNR de 5 a 25 dB, a passos de 5 dB.

As Figs. 4.19 a 4.24 mostram os erros médios quadráticos das medidas de atenuação e atraso dos algoritmos propostos e de referência para misturas anecoicas com sinais de voz e áudio, com 10 dB de ruído branco inserido nos sensores.

As Tabelas 4.3 e 4.4 apresentam os valores médios e desvios padrões dos coeficientes de atenuação e atraso obtido com cada algoritmo para sinais de voz e áudio, respectivamente.

As Figs. 4.25 a 4.28 apresentam as curvas do erro médio quadrático em função da SNR dos algoritmos propostos e de referência para misturas anecoicas, para sinais de voz e de áudio, respectivamente.

Dessas figuras, podemos observar que somente o algoritmo STUE inicializado com a solução do algoritmo TIFROM_Anec resultou em valores médios de MSE quase sempre abaixo dos obtidos pelos algoritmos de referência para os valores de

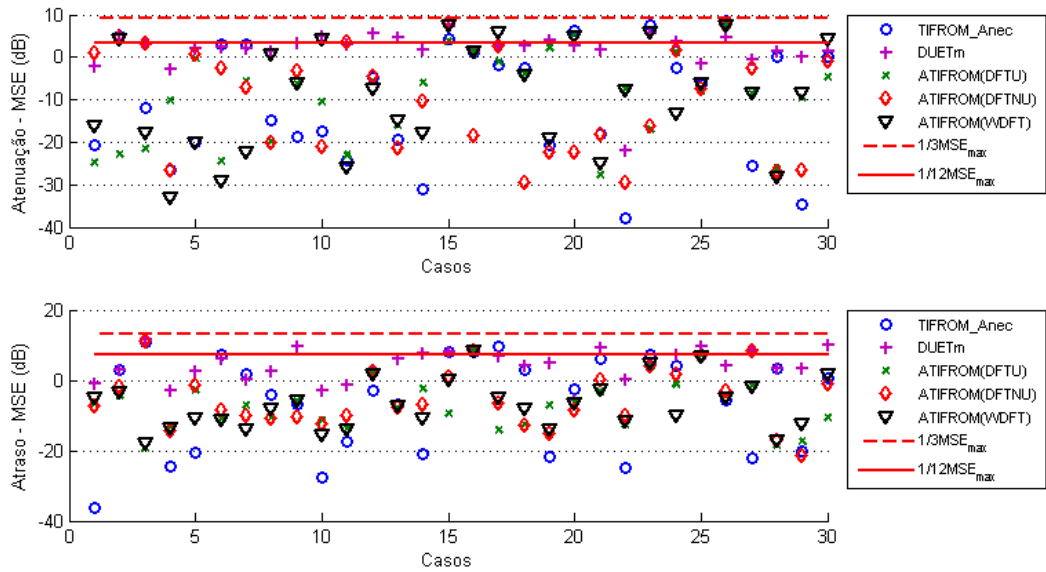


Figura 4.19: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM com Ruído Branco - Voz (SNR = 10 dB).

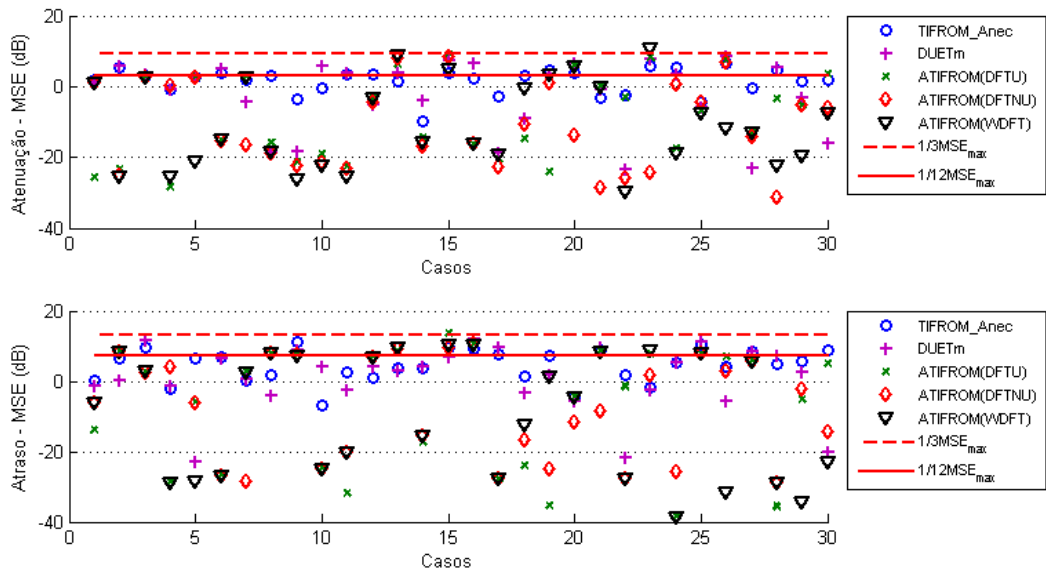


Figura 4.20: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM com Ruído Branco - Áudio (SNR = 10 dB).

SNR testados, e que o algoritmo ATIFROM(DFTNU), para $SNR \leq 15$ dB, sempre obteve valores médios de MSE inferiores aos apresentados pelos algoritmos de referência.

O algoritmo IMUE obteve um bom desempenho médio em quase todos os cenários, exceto para sinais de áudio com SNR inferior a 15 dB. Isso ocorreu devido ao problema de convergência não alcançada. De fato, todos os algoritmos tiveram

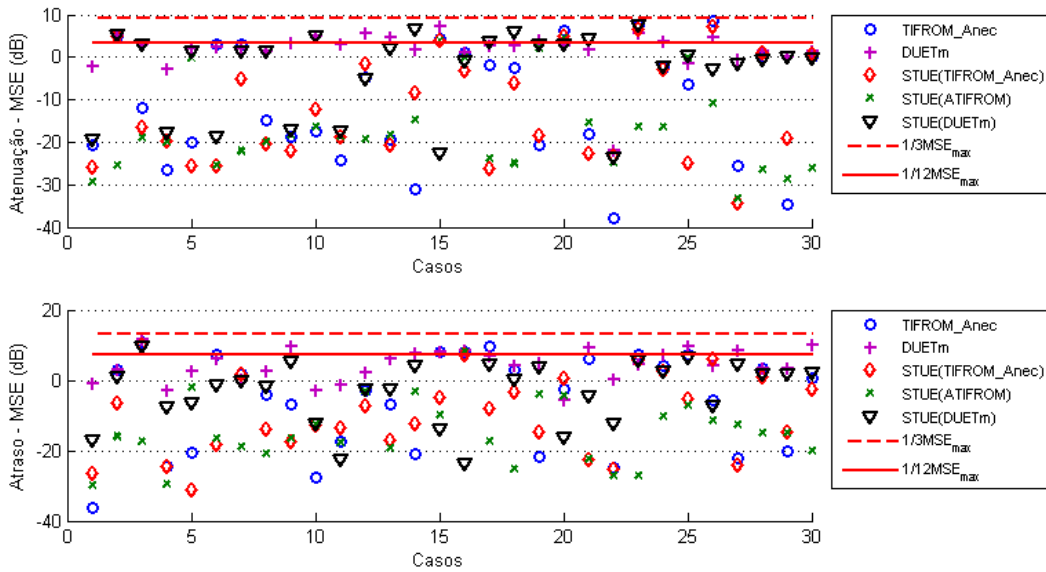


Figura 4.21: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE com Ruído Branco - Voz (SNR = 10 dB).

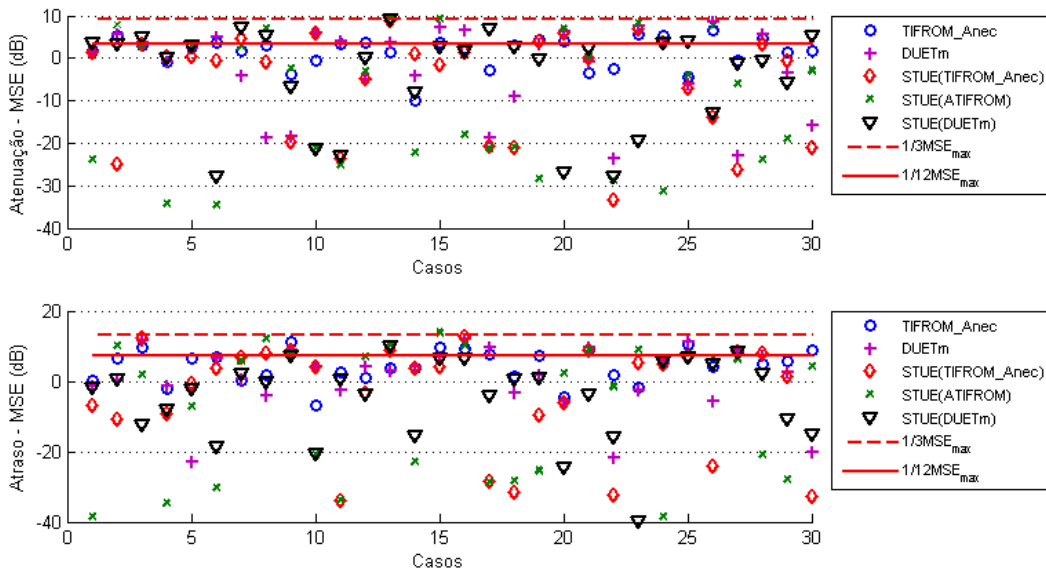


Figura 4.22: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE com Ruído Branco - Áudio (SNR = 10 dB).

dificuldades nos cenários com sinais de áudio, uma vez que nenhum deles obteve MSE inferior a 1 dB para SNR = 5 dB. Tal característica não ocorreu com sinais de voz, onde o algoritmo IMUE, com ambas as inicializações alcançou MSE inferior a 1 dB para a estimação dos coeficientes de atenuação. Da mesma forma, os algoritmos IMUE(ATIFROM) e ATIFROM obtiveram MSEs inferiores a 1 dB para SNR = 5 dB para a estimação dos coeficientes de atraso.

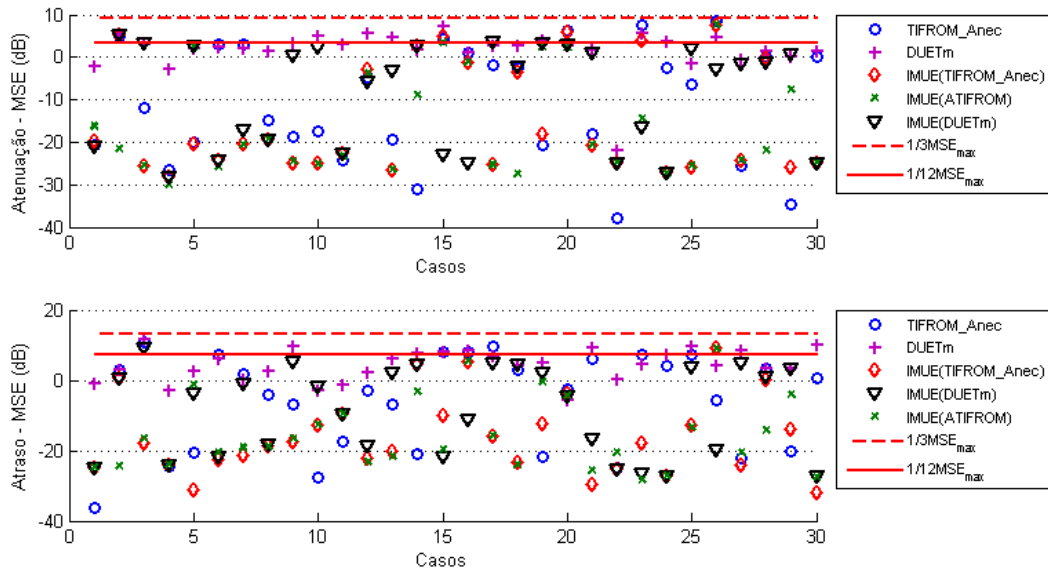


Figura 4.23: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE com Ruído Branco - Voz (SNR = 10 dB).

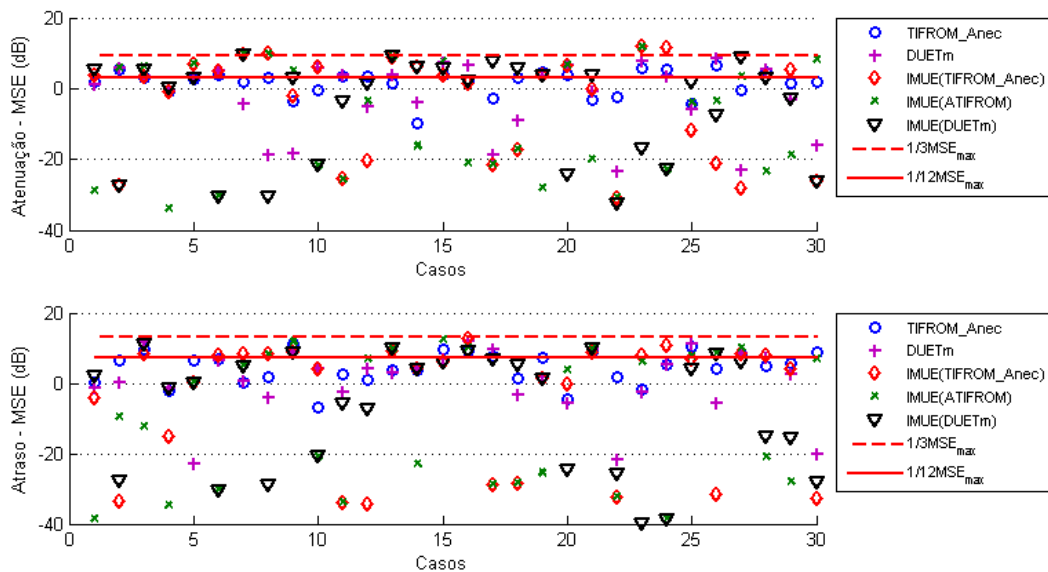


Figura 4.24: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE com Ruído Branco - Áudio (SNR = 10 dB).

O algoritmo ATIFROM(DFTNU) sempre esteve entre os três melhores algoritmos para os 30 cenários testados com SNR = 5 dB, tanto para os coeficientes de atraso quanto para os de atenuação. Tal consistência não foi alcançada por nenhum outro algoritmo testado. Em contrapartida, nenhum algoritmo esteve sempre entre os 3 piores em todos os cenários.

Na estimação dos coeficientes de atenuação, o algoritmo de referência TIFROM_-

		Atenuação - MSE	Atraso - MSE
Algoritmo		Média \pm Desvio	Média \pm Desvio
Voz	ATIFROM(DFTU)	$6,76 \times 10^{-1} \pm 1,25$	$7,41 \times 10^{-1} \pm 1,59$
	ATIFROM(DFTNU)	$9,64 \times 10^{-1} \pm 1,63$	$1,48 \pm 2,85$
	ATIFROM(WDFT)	$1,17 \pm 1,78$	$7,95 \times 10^{-1} \pm 1,63$
	STUE(TIFROM_Anec)	$8,03 \times 10^{-1} \pm 1,48$	$1,02 \pm 2,09$
	STUE(ATIFROM(DFTU))	$3,44 \times 10^{-1} \pm 7,60 \times 10^{-1}$	$3,17 \times 10^{-1} \pm 1,15$
	STUE(DUETm)	$1,48 \pm 1,46$	$1,48 \pm 1,88$
	IMUE(TIFROM_Anec)	$7,96 \times 10^{-1} \pm 1,52$	$5,88 \times 10^{-1} \pm 1,65$
	IMUE(ATIFROM(DFTU))	$5,03 \times 10^{-1} \pm 1,21$	$5,01 \times 10^{-1} \pm 1,59$
	IMUE(DUETm)	$8,82 \times 10^{-1} \pm 9,71 \times 10^{-1}$	$1,20 \pm 1,76$
	DUETm	$1,97 \pm 1,12$	$4,10 \pm 3,51$
	TIFROM_Anec	$1,10 \pm 1,85$	$2,26 \pm 3,12$

Tabela 4.3: Estatísticas dos Algoritmos para Sinais de Voz com Ruído Branco (SNR = 10 dB).

		Atenuação - MSE	Atraso - MSE
Algoritmo		Média \pm Desvio	Média \pm Desvio
Áudio	ATIFROM(DFTU)	$1,34 \pm 2,24$	$3,42 \pm 5,01$
	ATIFROM(DFTNU)	$9,55 \times 10^{-1} \pm 1,88$	$2,40 \pm 3,35$
	ATIFROM(WDFT)	$1,25 \pm 2,68$	$2,81 \pm 3,65$
	STUE(TIFROM_Anec)	$1,46 \pm 1,81$	$3,58 \pm 4,73$
	STUE(ATIFROM(DFTU))	$1,83 \pm 2,78$	$4,09 \pm 5,69$
	STUE(DUETm)	$1,74 \pm 1,93$	$1,83 \pm 2,47$
	IMUE(TIFROM_Anec)	$3,24 \pm 4,22$	$4,00 \pm 4,35$
	IMUE(ATIFROM(DFTU))	$2,68 \pm 3,96$	$3,83 \pm 5,10$
	IMUE(DUETm)	$2,25 \pm 2,64$	$2,85 \pm 3,72$
	DUETm	$1,81 \pm 1,11$	$4,19 \pm 3,45$
	TIFROM_Anec	$1,98 \pm 2,01$	$3,80 \pm 4,57$

Tabela 4.4: Estatísticas dos Algoritmos para Sinais de Áudio com Ruído Branco (SNR = 10 dB).

Anec sempre apresentou bom desempenho para SNR >15 dB e desempenho ruim para SNR \leq 15 dB, enquanto que o algoritmo de referência DUETm sempre teve o pior desempenho para alta SNR, porém nem sempre ficou entre os três piores para baixa SNR, como ocorreu com sinais de áudio. Vale a pena ressaltar que, para sinais de voz, o algoritmo de referência DUETm sempre teve o pior desempenho para todas as SNRs testadas.

A análise do MSE dos coeficientes de atenuação com sinais de voz, mostra os algoritmos TIFROM_Anec e STUE(TIFROM_Anec) com valores de MSE

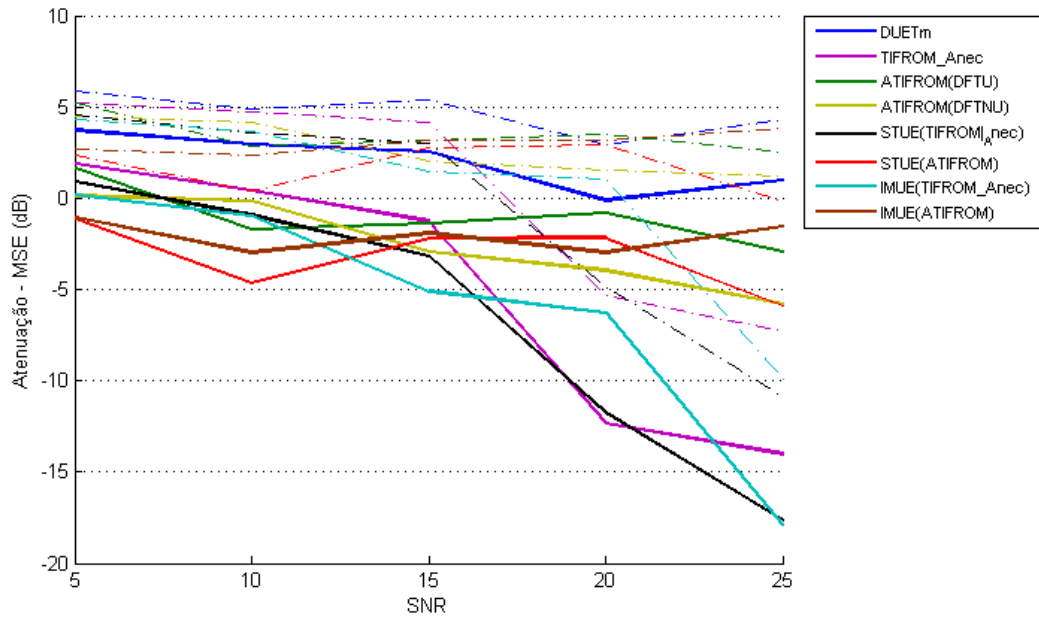


Figura 4.25: MSE dos Coeficientes de Atenuação \times SNR - Voz com Ruído Branco.

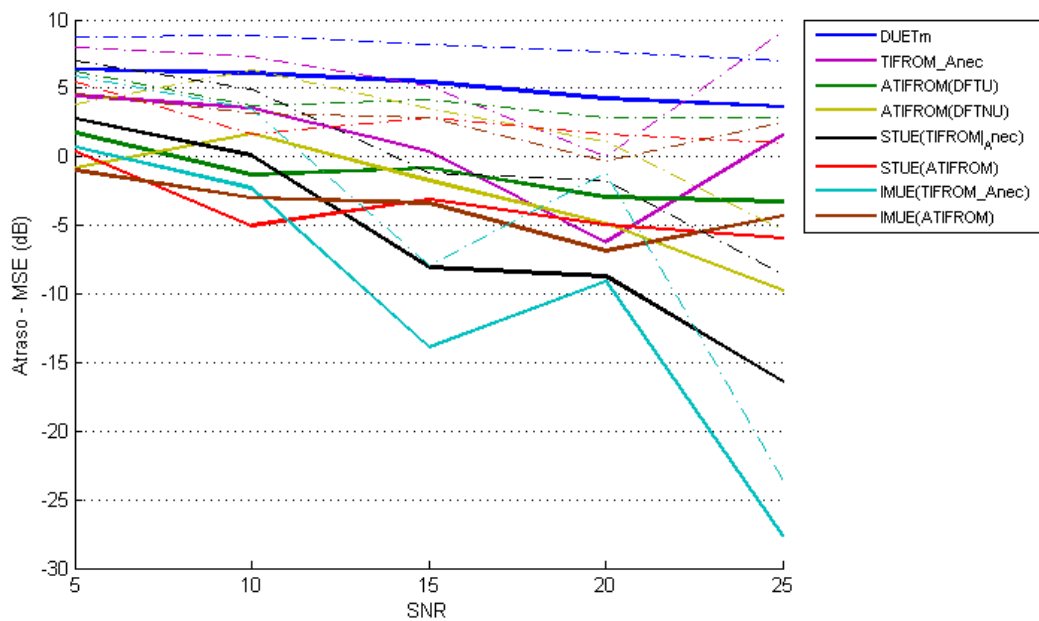


Figura 4.26: MSE dos Coeficientes de Atraso \times SNR - Voz com Ruído Branco.

médio mais um desvio padrão melhores que os dos algoritmos ATIFROM(DFTU), ATIFROM(DFTNU), DUETm e STUE(ATIFROM), para valores de $\text{SNR} \geq 20$ dB. Para $\text{SNR} = 5$ dB, observam-se desempenhos equivalentes dos algoritmos STUE(ATIFROM) e IMUE(ATIFROM), melhores do que os dos demais algoritmos avaliados. E, para baixos valores de SNR, os algoritmos IMUE(ATIFROM) e STUE(ATIFROM) apresentam desempenhos superiores ao do DUETm.

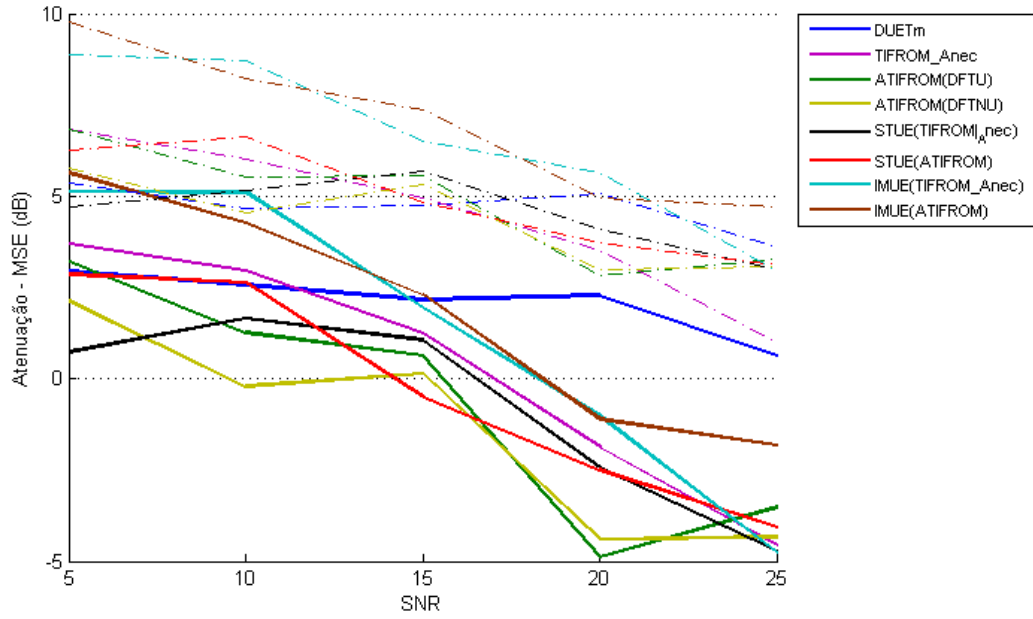


Figura 4.27: MSE dos Coeficientes de Atenuação \times SNR - Áudio com Ruído Branco.

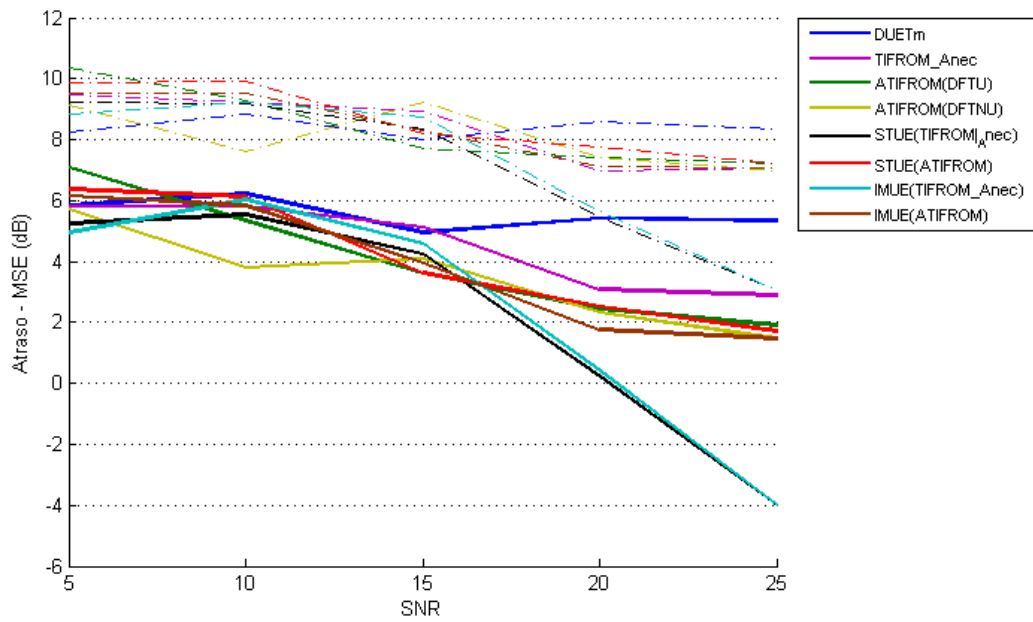


Figura 4.28: MSE dos Coeficientes de Atraso \times SNR - Áudio com Ruído Branco.

A análise do MSE dos coeficientes de atenuação com sinais de áudio não mostra superioridade de nenhum dos algoritmos propostos em relação aos algoritmos de referência.

Na estimação dos coeficientes de atraso para $\text{SNR} = 25$ dB, os algoritmos propostos quase sempre apresentaram desempenhos superiores em relação aos algoritmos de referência, sendo que, para sinais de áudio, essa característica se manteve até

SNR = 15 dB, e para sinais de voz ela só não foi observada para SNR = 20 dB.

A análise do MSE dos coeficientes de atraso com sinais de voz mostra que, para todas as SNRs, a maioria dos algoritmos propostos apresentaram uma superioridade em relação ao DUETm. Esse mesmo desempenho, não foi verificado ao compararmos com o algoritmo TIFROM_Anec. Contudo, o algoritmo IMUE(TIFROM_Anec) apresentou, para várias SNRs, em seus valores médio de MSE mais desvio padrão um desempenho superior ao apresentado pelo TIFROM_Anec.

A análise do MSE dos coeficientes de atraso com sinais de áudio mostra uma equivalência entre os algoritmos, exceto para $SNR \geq 20$ dB, onde o algoritmo IMUE (com ambas inicializações) apresentou desempenho superior ao do algoritmo DUETm.

Por fim, o algoritmo de referência DUETm sempre teve o pior desempenho para SNR = 25 dB, tanto para sinais de áudio quanto para sinais de voz, sendo essa característica ruim com sinais de voz observada para todas as SNRs.

4.3.2 Misturas com Ruído *Babble* nos Sensores

Nesta seção, avaliamos o comportamento dos algoritmos propostos e de referência na presença de ruído *babble*. Da mesma forma como feito no Cap. 3, para misturas instantâneas, variamos o nível de ruído nas misturas para obter os mesmos valores de SNR usados anteriormente.

As Figs. 4.29 a 4.34 mostram os erros médios quadráticos dos coeficientes de atenuação e atraso obtidos com os algoritmos propostos e de referência para misturas anecoicas com sinais de voz e áudio, contaminados por ruído *babble* com SNR = com 10 dB.

As Tabelas 4.5 e 4.6 apresentam os valores médios e desvios padrões dos MSEs obtidos com cada algoritmo na presença de ruído *babble*. As Figs. 4.35 a 4.38 apresentam as curvas dos erros quadráticos médios em função da SNR para sinais de voz e áudio, respectivamente.

Observando as curvas de MSE podemos observar que o algoritmo IMUE inicializado com a solução do TIFROM_Anec foi a única configuração proposta que obteve MSE sempre abaixo dos obtidos pelos algoritmos de referência.

É possível observar que as estimativas de atraso pelos métodos propostos apresentaram sempre menores MSEs do que as dos algoritmos de referência, sendo que o mesmo não ocorreu com as estimativas de atenuação.

Finalmente, independentemente do tipo de ruído presente, também é possível observar que todos os algoritmos tiveram maior dificuldade para estimar os atrasos nos sinais de áudio do que em sinais de voz. Neste caso, somente os algoritmos IMUE e STUE conseguiram para algumas SNRs estimar os atrasos com MSE inferior a 1 dB. Essa dificuldade não se verificou nos demais cenários.

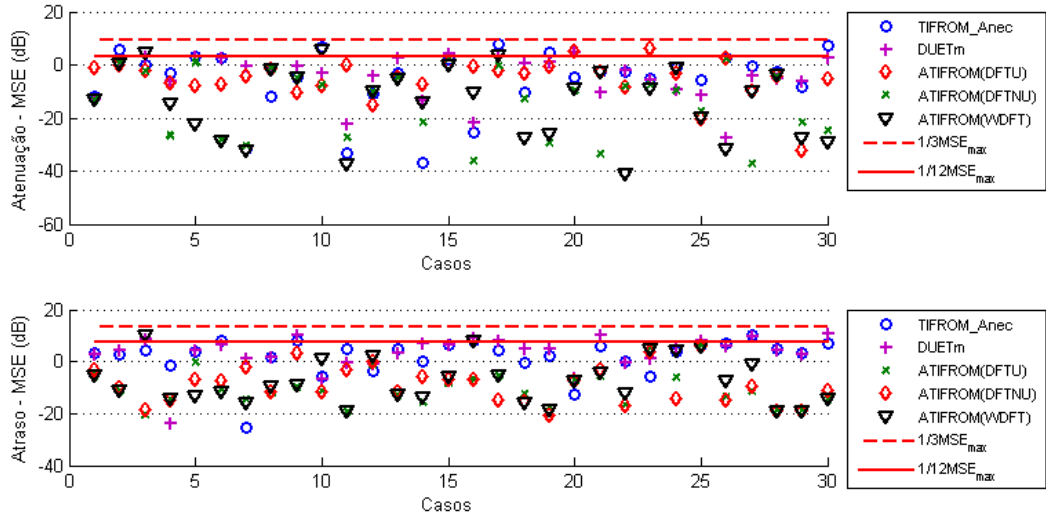


Figura 4.29: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM com Ruído *Babble* - Voz (SNR = 10 dB).

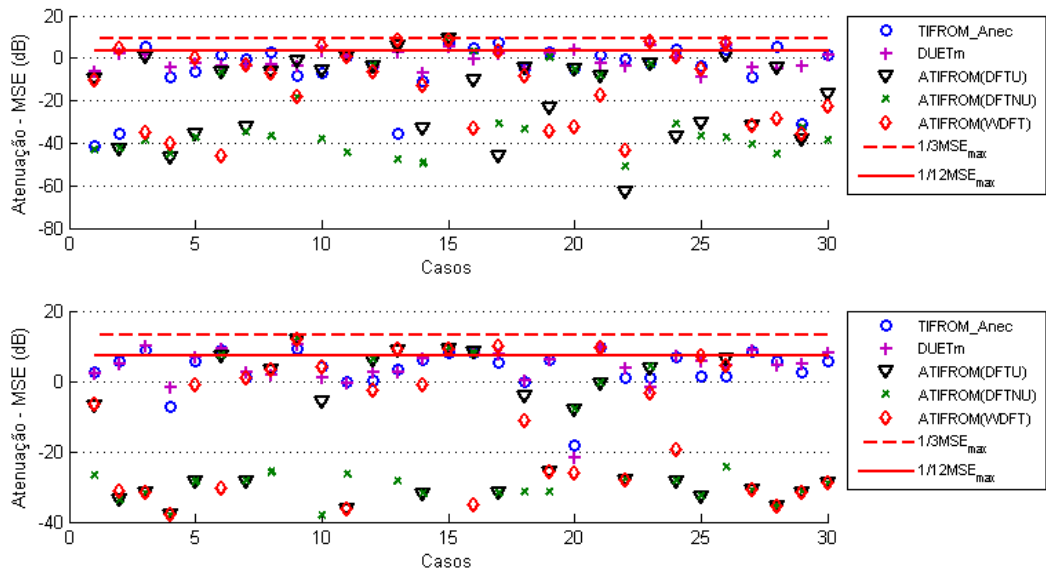


Figura 4.30: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e ATIFROM com Ruído *Babble*- Áudio (SNR = 10 dB).

Uma característica notada na presença de ruído *babble* foi que o banco DFTU quase sempre apresentou melhor desempenho para SNR = 5 dB do que o DFTNU quando empregado pelo algoritmo ATIFROM. No entanto, essa característica se inverte quando a contaminação ocorre com ruído branco.

A análise do MSE dos coeficientes de atenuação com sinais de voz mostra que o algoritmo IMUE(ATIFROM) apresenta um desempenho superior aos dos algoritmos

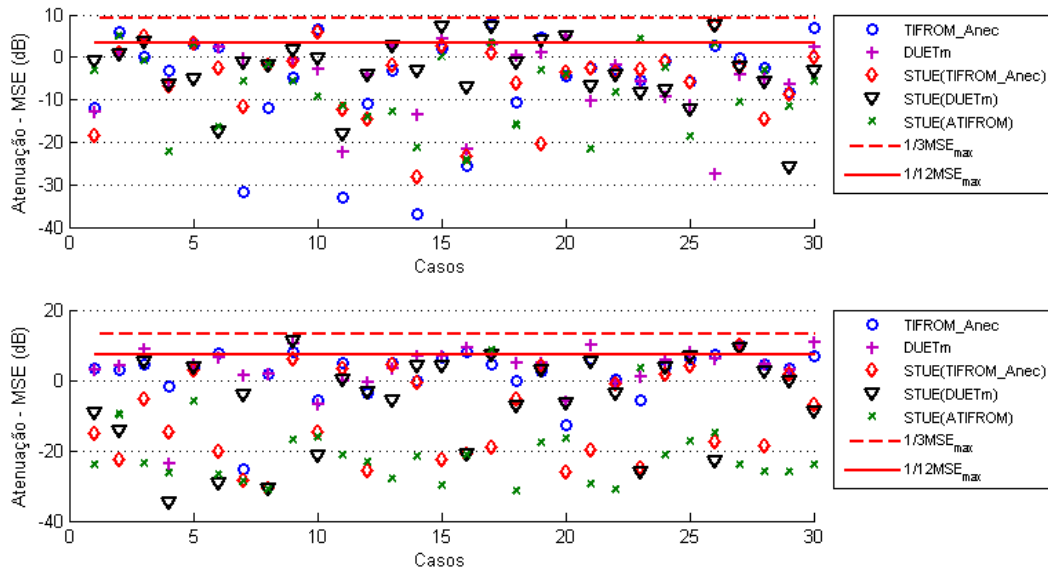


Figura 4.31: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE com Ruído *Babble* - Voz (SNR = 10 dB).

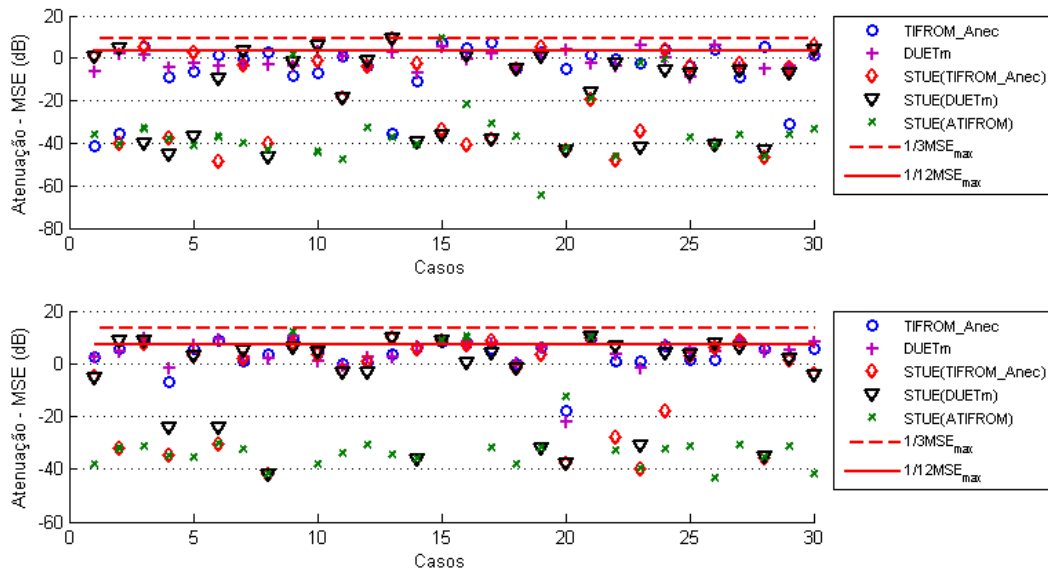


Figura 4.32: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e STUE com Ruído *Babble* - Áudio (SNR = 10 dB).

de referência para algumas SNRs, e que o algoritmo ATIFROM(DFTU) apresenta igual característica em relação aos algoritmos de referência para $SNR \leq 10$ dB.

Já a análise do MSE dos coeficientes de atenuação com sinais de áudio mostra que o algoritmo IMUE(ATIFROM) apresenta um desempenho superior aos dos algoritmos de referência para algumas SNRs, e que os algoritmos IMUE(TIFROM_Anec) e STUE(TIFROM_Anec) apresentaram desempenho superior ao do algoritmo

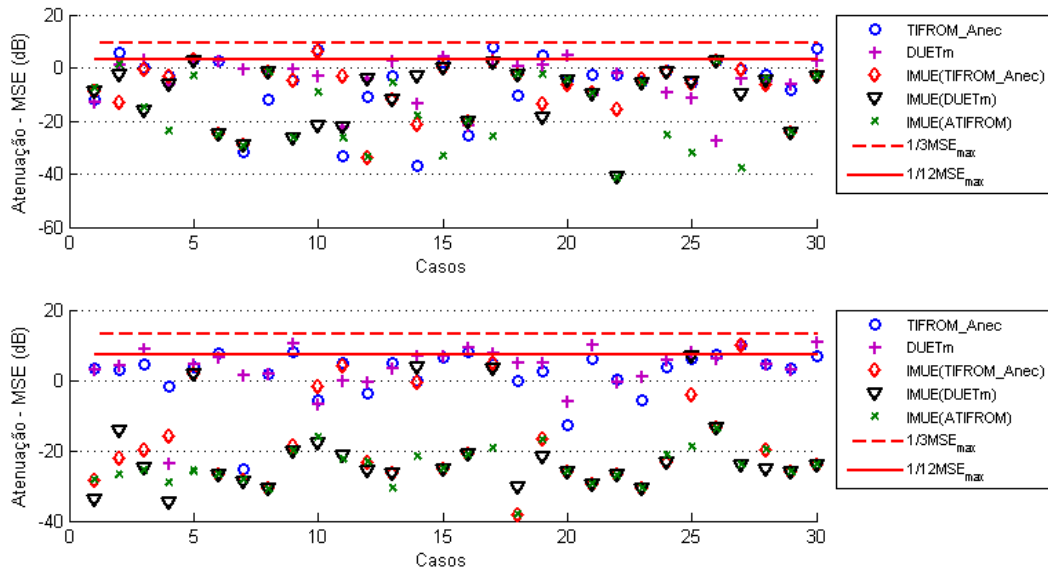


Figura 4.33: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE com Ruído *Babble* - Voz (SNR = 10 dB).

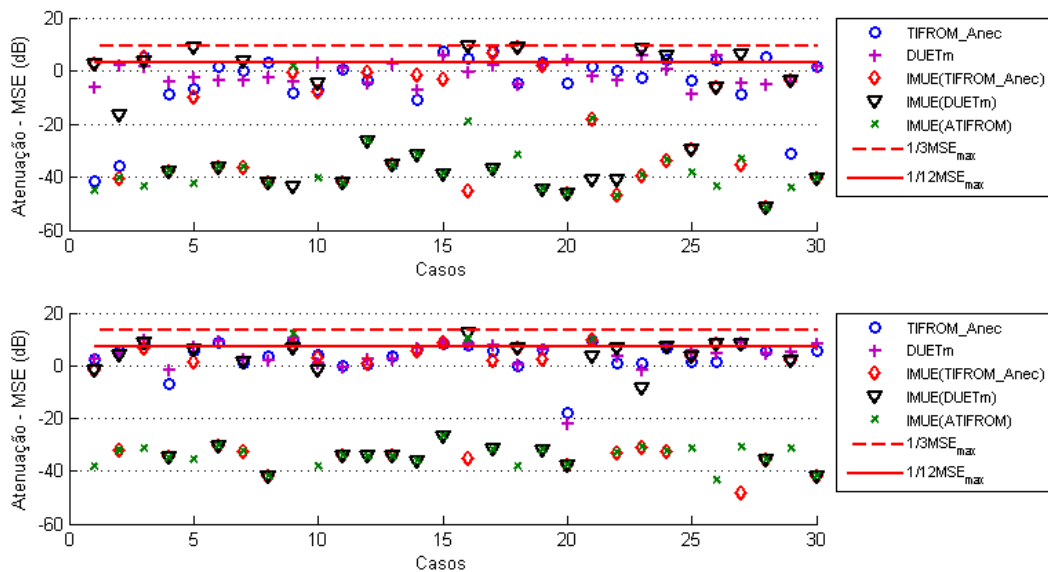


Figura 4.34: MSE dos Elementos das Matrizes de Mistura Estimadas pelos Algoritmos de Referência e IMUE com Ruído *Babble* - Áudio (SNR = 10 dB).

DUETm, para $SNR \geq 20$ dB. Finalmente, com $SNR = 5$ dB, todos os algoritmos apresentaram desempenhos equivalentes.

		Atenuação - MSE	Atraso - MSE
Algoritmo		Média \pm Desvio	Média \pm Desvio
Voz	ATIFROM(DFTU)	$3,35 \times 10^{-1} \pm 5,12 \times 10^{-1}$	$3,88 \times 10^{-1} \pm 9,49 \times 10^{-1}$
	ATIFROM(DFTNU)	$7,13 \times 10^{-1} \pm 9,52 \times 10^{-1}$	$4,50 \times 10^{-1} \pm 8,74 \times 10^{-1}$
	ATIFROM(WDFT)	$4,89 \times 10^{-1} \pm 8,90 \times 10^{-1}$	$1,11 \pm 2,37$
	STUE(TIFROM_Anec)	$9,13 \times 10^{-1} \pm 1,29$	$1,07 \pm 2,07$
	STUE(ATIFROM(DFTU))	$6,24 \times 10^{-1} \pm 8,92 \times 10^{-1}$	$3,28 \times 10^{-1} \pm 1,33$
	STUE(DUETm)	$1,22 \pm 1,62$	$1,90 \pm 2,98$
	IMUE(TIFROM_Anec)	$5,86 \times 10^{-1} \pm 8,51 \times 10^{-1}$	$6,47 \times 10^{-1} \pm 1,97$
	IMUE(ATIFROM(DFTU))	$2,58 \times 10^{-1} \pm 4,21 \times 10^{-1}$	$6,70 \times 10^{-3} \pm 9,00 \times 10^{-3}$
	IMUE(DUETm)	$4,01 \times 10^{-1} \pm 5,18 \times 10^{-1}$	$3,63 \times 10^{-1} \pm 1,03$
	DUETm	$8,95 \times 10^{-1} \pm 8,95 \times 10^{-1}$	$3,99 \pm 3,36$
TIFROM_Anec	$1,24 \pm 1,67$	$2,84 \pm 2,35$	

Tabela 4.5: Estatística dos Algoritmos para Sinais de Voz com Ruído *Babble* (SNR = 10 dB).

		Atenuação - MSE	Atraso - MSE
Algoritmo		Média \pm Desvio	Média \pm Desvio
Áudio	ATIFROM(DFTU)	$4,58 \times 10^{-1} \pm 1,65$	$1,44 \pm 3,43$
	ATIFROM(DFTNU)	$1,36 \pm 2,33$	$2,27 \pm 3,96$
	ATIFROM(WDFT)	$6,96 \times 10^{-1} \pm 1,76$	$1,96 \pm 3,62$
	STUE(TIFROM_Anec)	$9,79 \times 10^{-1} \pm 1,78$	$2,72 \pm 3,18$
	STUE(ATIFROM(DFTU))	$3,96 \times 10^{-1} \pm 1,64$	$1,47 \pm 3,95$
	STUE(DUETm)	$9,19 \times 10^{-1} \pm 1,74$	$2,73 \pm 3,12$
	IMUE(TIFROM_Anec)	$7,68 \times 10^{-1} \pm 1,68$	$1,83 \pm 2,65$
	IMUE(ATIFROM(DFTU))	$4,87 \times 10^{-2} \pm 2,60 \times 10^{-1}$	$1,19 \pm 3,74$
	IMUE(DUETm)	$1,55 \pm 2,75$	$2,46 \pm 3,79$
	DUETm	$1,17 \pm 1,10$	$4,03 \pm 2,99$
TIFROM_Anec	$1,33 \pm 1,51$	$3,45 \pm 2,58$	

Tabela 4.6: Estatística dos Algoritmos para Sinais de Voz e Áudio com Ruído *Babble* (SNR = 10 dB).

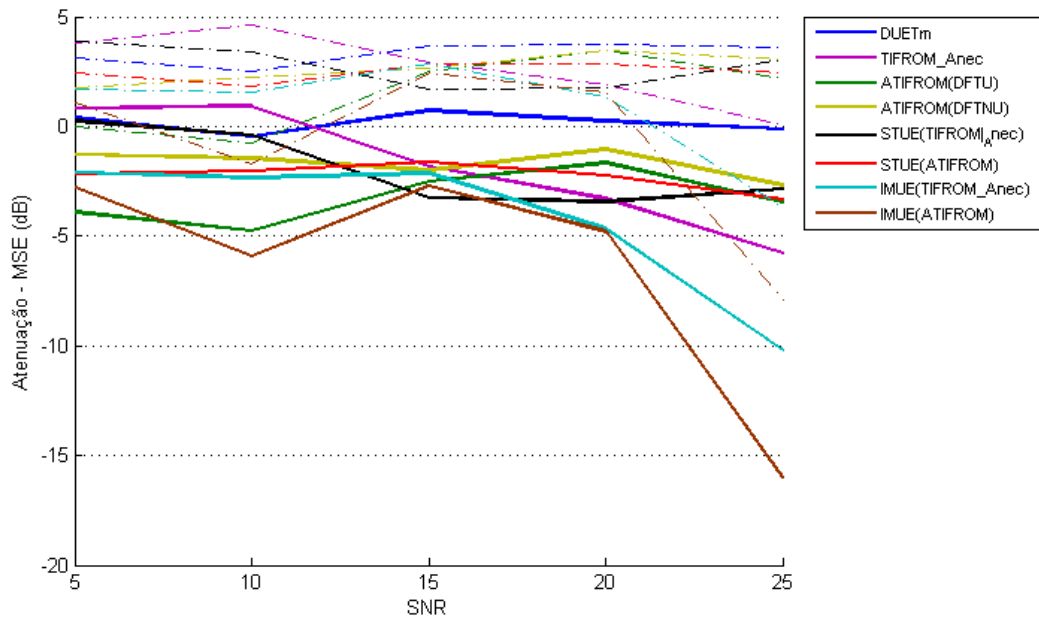


Figura 4.35: MSE dos Coeficientes de Atenuação \times SNR - Voz com Ruído *Babble*.

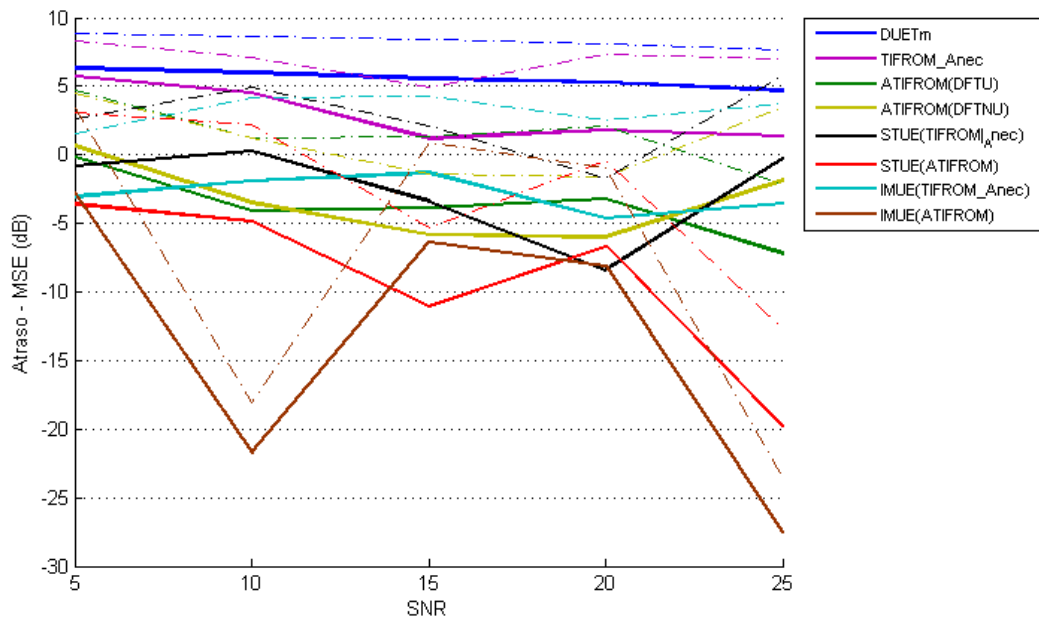


Figura 4.36: MSE dos Coeficientes de Atraso \times SNR - Voz com Ruído *Babble*.

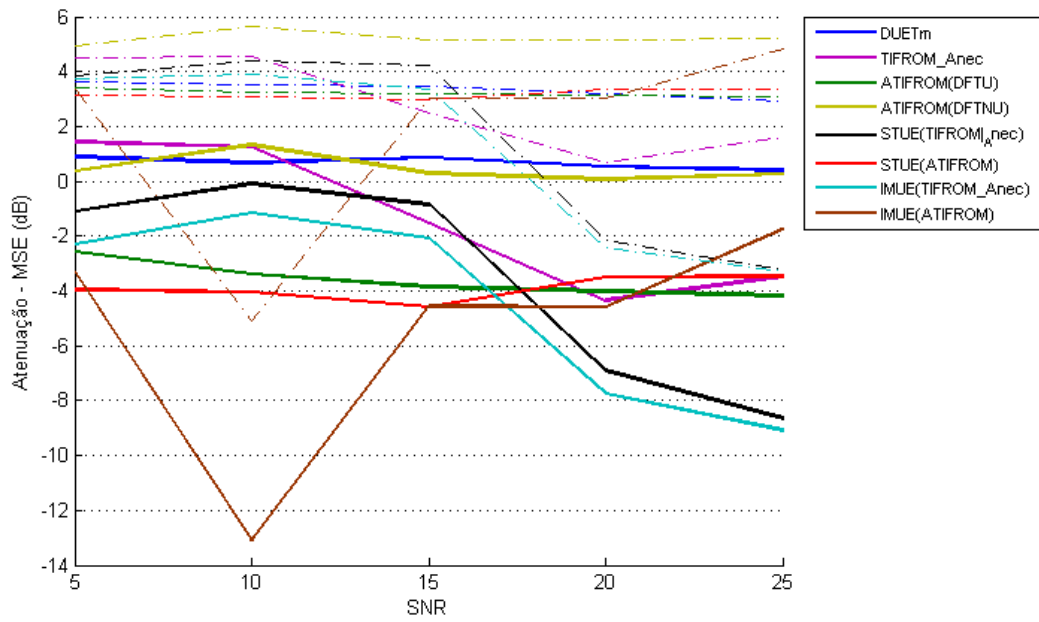


Figura 4.37: MSE dos Coeficientes de Atenuação \times SNR - Áudio com Ruído *Babble*.

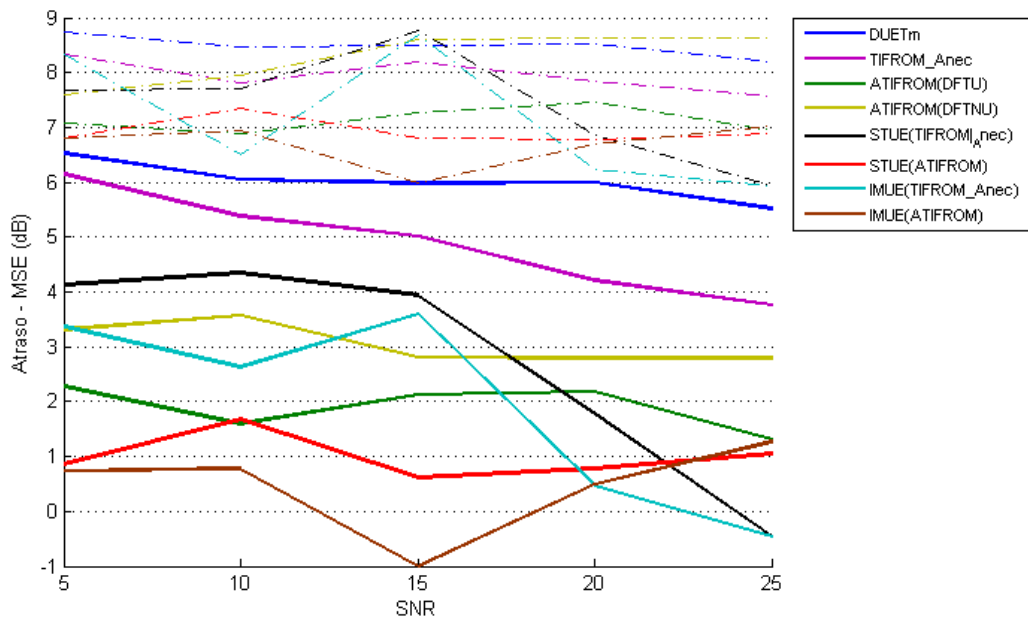


Figura 4.38: MSE dos Coeficientes de Atraso \times SNR - Áudio com Ruído *Babble*.

4.4 Desempenho dos Algoritmos após a Reconstituição das Fontes

Nesta seção as estimativas das fontes são obtidas por meio do mascaramento binário bidimensional (*HARD*) das matrizes de atenuação e de atraso estimadas, obtidas pelos algoritmos propostos neste capítulo. A partir dessas estimativas das fontes, foram formadas as Tabelas 4.7 a 4.10, que contêm as medidas de SDR, SIR e SAR para os cenários com ruído branco e *babble*, e com sinais de voz e de áudio.

Ao analisarmos os resultados mostrados nas tabelas, podemos observar que, para misturas com sinais de voz sem ruído ($\text{SNR} = \infty$), os algoritmos propostos que apresentaram os melhores desempenhos foram os algoritmos IMUE(STUE(ATIFROM)), IMUE(STUE(TIFROM_Anec)) e STUE(ATIFROM), todos eles resultando em $\text{SDR} = 11,13$ dB. Todos os algoritmos propostos e avaliados apresentaram desempenhos superiores aos algoritmos de referência DUETm e TIFROM_Anec. Com relação à SIR, os mesmos algoritmos apresentaram os melhores desempenhos, obtendo valores de SDR em torno de 20,8 dB. Nesse caso, somente o algoritmo STUE(DUET) apresentou um desempenho inferior ao algoritmo de referência (TIFROM_Anec).

Para sinais de voz contaminados por ruído branco com $\text{SNR} = 5$ dB, podemos observar que os algoritmos que apresentaram os melhores desempenhos em relação à SDR foram o ATIFROM(DFTNU), o IMUE(ATIFROM) e o IMUE(STUE(TIFROM)), obtendo 2,26 dB, 2,16 dB e 1,90 dB, respectivamente. Para esse caso, todos os algoritmos propostos apresentaram desempenhos superiores aos algoritmos de referência. Analisando os desempenhos em relação à SIR, podemos verificar que os algoritmos que apresentaram os melhores desempenhos foram o IMUE(ATIFROM), o IMUE(STUE(TIFROM)) e o IMUE(STUE(ATIFROM)), obtendo 15,51 dB, 15,30 dB e 14,97 dB, respectivamente. Nesse caso, todos os algoritmos propostos obtiveram desempenhos superiores aos algoritmos de referência.

Analisando os resultados dos algoritmos em relação à SDR com sinais de voz contaminados por ruído *babble* com $\text{SNR} = 5$ dB, podemos verificar que os algoritmos que apresentaram os melhores desempenhos foram o IMUE(ATIFROM), o STUE(ATIFROM) e o IMUE(TIFROM), obtendo, respectivamente, 4,66 dB, 4,44 dB e 4,40 dB. Com relação à SIR, os algoritmos que obtiveram os melhores desempenhos foram o IMUE(STUE(ATIFROM)), o IMUE(ATIFROM) e o ATIFROM(DFTU), obtendo, respectivamente, 17,74 dB, 17,71 dB e 17,41 dB. Novamente, todos os algoritmos propostos obtiveram desempenhos superiores aos algoritmos de referência.

Sinais de Voz	SNR = 05 dB		SNR = 10 dB		SNR = 15 dB		SNR = 20 dB		SNR = 25 dB		SNR = ∞					
	SDR	SIR	SDR	SIR	SDR	SIR	SDR	SIR	SDR	SIR	SDR	SIR				
Algoritmo																
DUETm	-1,22	10,92	2,01	13,42	5,45	15,07	7,52	17,19	9,30	17,20	8,06	17,20	9,90	9,06	18,61	10,97
TIFROM_Aneq	0,66	13,76	2,62	16,80	6,18	19,04	9,16	20,28	10,72	20,19	10,27	20,19	11,17	9,76	19,49	11,35
ATIFROM(DFTU)	1,63	14,27	3,20	17,51	6,82	18,53	9,38	19,64	11,06	19,87	9,99	19,87	11,81	10,71	20,32	11,63
ATIFROM(DFTNU)	2,26	14,74	3,31	16,68	6,52	19,07	9,18	19,83	10,73	19,99	10,31	19,99	11,20	10,65	20,18	11,64
ATIFROM(WDFT)	1,87	14,23	3,24	17,35	6,77	18,64	9,32	19,46	11,12	19,92	9,99	19,92	12,13	10,37	20,05	12,11
STUE(DUETm)	0,76	14,12	3,05	16,07	6,59	17,59	9,23	19,31	10,55	19,16	9,39	19,16	11,27	9,87	19,32	11,89
STUE(TIFROM_Aneq)	1,28	14,17	3,04	17,15	6,80	19,53	9,53	20,32	10,85	20,58	10,73	20,58	11,47	10,89	20,49	11,78
STUE(ATIFROM)	1,67	14,78	3,20	18,22	7,17	19,26	9,65	20,04	11,29	20,38	10,44	20,38	11,94	11,13	20,81	11,86
IMUE(DUETm)	0,90	13,85	3,21	17,10	7,12	19,22	9,60	20,16	10,90	20,25	10,45	20,25	11,49	10,91	20,57	11,96
IMUE(TIFROM_Aneq)	1,87	14,96	3,34	17,90	7,11	19,68	9,61	20,35	10,88	20,63	10,77	20,63	11,49	11,05	20,74	11,89
IMUE(ATIFROM)	2,16	15,51	3,53	18,15	7,22	19,35	9,70	20,20	11,26	20,25	10,35	20,25	12,04	11,13	20,81	11,86
IMUE((STUE(TIFROM_Aneq)))	1,90	15,30	3,43	18,22	7,14	19,80	9,63	20,35	10,89	20,63	10,77	20,63	11,49	11,05	20,74	11,88
IMUE(STUE(ATIFROM))	1,69	14,97	3,35	18,21	7,24	19,56	9,68	20,34	11,19	20,55	10,65	20,55	11,88	11,13	20,82	11,86

Tabela 4.7: Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Voz na Presença de Ruído Branco com Diferentes SNRs.

Sinais de Áudio	SNR = 05 dB			SNR = 10 dB			SNR = 15 dB			SNR = 20 dB			SNR = 25 dB			SNR = ∞		
	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
DUETm	1,59	11,99	5,18	7,28	18,08	10,21	11,48	21,60	15,02	19,23	16,01	24,40	17,60	24,82	22,50	23,17	28,52	27,47
TIFROM_Aneq	4,35	16,42	6,10	7,71	19,39	10,35	14,76	25,83	16,14	21,26	20,32	30,20	23,82	32,27	24,92	27,67	33,50	29,60
ATIFROM(DFTU)	4,25	16,18	6,40	10,12	21,86	11,72	15,85	26,75	16,66	21,46	20,73	30,45	24,00	32,28	25,06	28,48	34,15	30,12
ATIFROM(DFTNU)	4,83	16,93	6,38	10,77	22,50	11,86	15,99	26,94	16,73	21,38	20,60	30,32	23,96	32,20	25,05	28,30	33,92	30,25
ATIFROM(WDFT)	5,01	16,87	6,67	10,65	22,56	11,89	16,13	27,13	17,00	21,48	20,74	30,52	23,97	32,27	25,04	28,50	34,18	30,14
STUE(DUETm)	2,72	13,43	5,76	9,32	20,71	11,27	13,45	23,96	16,02	20,47	17,87	27,24	21,54	29,78	23,83	25,65	31,08	29,33
STUE(TIFROM_Aneq)	4,83	16,90	6,52	9,07	20,61	11,17	15,59	26,84	16,78	21,57	20,71	30,51	24,26	32,53	25,17	28,23	33,89	29,98
STUE(ATIFROM)	5,01	17,06	6,54	10,79	22,52	12,03	16,48	27,45	17,16	21,51	20,82	30,59	24,14	32,46	25,08	28,46	34,13	30,11
IMUE(DUETm)	3,90	15,64	6,19	9,57	21,58	11,61	14,26	25,09	16,51	19,38	19,38	29,19	22,52	30,86	24,45	26,87	32,58	30,14
IMUE(TIFROM_Aneq)	5,14	17,00	6,55	9,59	21,28	11,58	15,64	26,76	16,83	21,58	20,80	30,61	24,27	32,54	25,17	28,21	33,80	30,06
IMUE(ATIFROM)	5,22	17,10	6,74	10,51	22,32	11,73	16,60	27,58	17,26	21,55	20,86	30,60	24,17	32,48	25,12	28,46	34,13	30,11
IMUE(STUE(TIFROM_Aneq))	5,22	17,28	6,73	9,88	21,79	11,66	15,71	26,84	16,82	21,57	20,80	30,63	24,27	32,54	25,17	28,21	33,80	30,04
IMUE(STUE(ATIFROM))	5,33	17,04	6,83	11,09	22,98	12,34	16,58	27,59	17,25	21,57	20,87	30,62	24,17	32,46	25,10	28,46	34,13	30,11

Tabela 4.8: Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Áudio na Presença de Ruído Branco com Diferentes SNRs.

Sinais de Voz	SNR = 05 dB			SNR = 10 dB			SNR = 15 dB			SNR = 20 dB			SNR = 25 dB			SNR = ∞		
	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
DUETm	1,49	13,18	3,61	3,58	15,12	5,83	5,62	15,95	7,68	16,34	9,12	7,45	17,23	9,94	9,06	18,61	10,97	10,97
TIFROM_Aneec	2,39	14,92	4,93	5,03	16,59	7,37	7,64	18,92	8,87	19,51	10,30	9,87	19,88	11,26	9,76	19,49	11,35	11,35
ATIFROM(DFTU)	4,16	17,41	5,21	6,45	18,72	7,38	7,92	19,14	9,24	19,59	10,64	10,00	19,90	11,19	10,71	20,32	11,63	11,63
ATIFROM(DFTNU)	4,11	16,22	5,36	6,36	18,07	7,30	7,98	19,05	9,11	19,54	11,06	9,77	19,69	11,62	10,65	20,18	11,64	11,64
ATIFROM(WDFT)	3,94	16,97	5,02	6,22	18,35	7,33	7,99	19,45	9,24	19,12	10,86	10,05	19,91	11,63	10,37	20,05	12,11	12,11
STUE(DUETm)	2,99	15,22	4,83	5,24	16,58	7,32	7,21	17,78	9,35	18,28	10,67	9,17	18,97	11,14	9,87	19,32	11,89	11,89
STUE(TIFROM_Aneec)	3,59	16,35	5,42	5,52	17,40	7,54	8,48	19,59	9,61	20,29	10,90	9,85	19,77	11,07	10,89	20,49	11,78	11,78
STUE(ATIFROM)	4,44	17,33	5,39	6,68	18,95	7,45	8,35	19,50	9,43	19,94	10,74	10,16	20,18	11,15	11,13	20,81	11,86	11,86
IMUE(DUETm)	4,13	16,67	5,61	6,73	19,04	7,77	8,22	19,33	9,60	20,07	10,93	10,31	20,35	11,49	10,91	20,57	11,96	11,96
IMUE(TIFROM_Aneec)	4,40	17,24	5,76	6,58	18,72	7,74	8,63	19,74	9,79	20,28	11,01	10,47	20,32	11,58	11,05	20,74	11,89	11,89
IMUE(ATIFROM)	4,66	17,71	5,65	7,03	19,34	7,65	8,65	19,73	9,61	20,35	10,91	10,81	20,67	11,54	11,13	20,81	11,86	11,86
IMUE(STUE(TIFROM_Aneec))	4,29	17,31	5,76	6,52	18,65	7,78	8,55	19,62	9,73	20,52	11,17	10,60	20,45	11,50	11,05	20,74	11,88	11,88
IMUE((STUE(ATIFROM)))	4,37	17,74	5,33	7,11	19,36	7,70	8,48	19,62	9,62	20,43	10,83	10,60	20,51	11,44	11,13	20,82	11,86	11,86

Tabela 4.9: Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Voz na Presença de Ruído *Babble* com Diferentes SNRs.

Sinais de Áudio	SNR = 05 dB		SNR = 10 dB		SNR = 15 dB		SNR = 20 dB		SNR = 25 dB		SNR = ∞	
	SDR	SIR	SDR	SIR	SDR	SIR	SDR	SIR	SDR	SIR	SDR	SIR
DUET _m	7,89	16,56	12,45	19,65	13,82	20,37	17,63	24,14	17,46	23,15	23,17	28,52
TIFROM _{A_{nec}}	15,06	22,98	17,81	24,82	20,50	27,44	22,64	29,48	24,37	31,39	27,67	33,50
ATIFROM(DFTU)	18,21	26,61	20,53	28,30	22,35	29,84	23,75	31,20	25,13	32,31	28,48	34,15
ATIFROM(DFTNU)	17,06	25,29	18,97	27,07	21,39	29,00	22,96	30,32	24,26	31,37	28,30	33,92
ATIFROM(WDFT)	17,82	26,20	19,64	27,38	22,13	29,64	23,62	31,00	25,09	32,36	28,50	34,18
STUE(DUET _m)	14,80	23,45	18,36	25,58	19,96	27,80	21,51	28,53	23,21	30,10	25,65	31,08
STUE(TIFROM _{A_{nec}})	17,25	25,58	18,85	26,32	21,51	28,78	23,39	30,57	25,10	32,23	28,23	33,89
STUE(ATIFROM)	18,43	26,94	20,98	28,82	22,70	30,20	24,08	31,47	25,45	32,59	28,46	34,13
IMUE(DUET _m)	17,66	26,15	18,80	26,39	21,69	28,97	22,68	29,93	22,68	29,93	26,87	32,58
IMUE(TIFROM _{A_{nec}})	17,56	26,13	19,16	27,28	22,01	29,29	23,67	31,01	23,67	31,01	28,21	33,80
IMUE(ATIFROM)	18,71	27,32	20,89	28,77	22,68	30,19	24,13	31,52	24,13	31,52	28,46	34,13
IMUE(STUE(TIFROM))	17,99	26,48	19,38	27,66	21,68	29,10	23,60	30,91	23,60	30,91	28,21	33,80
IMUE(STUE(ATIFROM))	18,24	26,95	20,89	28,77	22,69	30,20	24,13	31,52	24,13	31,52	28,46	34,13

Tabela 4.10: Avaliação dos Algoritmos Propostos e Clássicos para Estimação da Matriz de Mistura com Sinal de Áudio na Presença de Ruído *Babble* com Diferentes SNRs.

Analisando os resultados obtidos para misturas com sinais de áudio sem ruído, os algoritmos que apresentaram os melhores desempenhos em relação à SDR foram os algoritmos ATIFROM(WDFT) com 28,50 dB, ATIFROM(DFTU) com 28,48 dB, e os algoritmos IMUE(STUE(ATIFROM)), IMUE(ATIFROM) e STUE(ATIFROM) empatados com 28,46 dB. Com relação à SIR, os mesmos algoritmos apresentaram os melhores desempenhos, obtendo, respectivamente, 34,18 dB, 34,15 dB e 34,13 dB. Somente os algoritmos propostos STUE(DUETm) e IMUE(DUETm) apresentaram desempenhos inferiores ao algoritmo de referência TIFROM_Anec.

Para sinais de áudio contaminados por ruído branco com $\text{SNR} = 5$ dB, podemos observar que os algoritmos que obtiveram os melhores desempenhos em relação à SDR foram o IMUE(STUE(ATIFROM)) com 5,33 dB, e o IMUE(STUE(TIFROM)) e o IMUE(ATIFROM) empatados com 5,22 dB. Analisando o desempenho em relação à SIR, podemos verificar que, os algoritmos que obtiveram os melhores desempenhos foram o IMUE(STUE(TIFROM)), o IMUE(ATIFROM) e o STUE(ATIFROM), obtendo 17,28 dB, 17,10 dB e 17,06 dB, respectivamente. Os algoritmos propostos ATIFROM(DFTU), STUE(DUETm) e IMUE(DUETm) apresentaram desempenhos inferiores em relação à SDR quando comparados com o algoritmo de referência TIFROM_Anec.

Ao analisarmos os resultados dos algoritmos em relação à SDR, com sinais de áudio contaminados por ruído *babble* com $\text{SNR} = 5$ dB, podemos verificar que os algoritmos que apresentaram os melhores desempenhos foram o IMUE(ATIFROM), o STUE(ATIFROM) e o IMUE(STUE(ATIFROM)), obtendo, respectivamente, 18,71 dB, 18,43 dB e 18,24 dB. Somente o algoritmo proposto STUE(DUETm) apresentou desempenho inferior ao algoritmo de referência TIFROM_Anec. Com relação à SIR, os algoritmos que obtiveram os melhores desempenhos foram o IMUE(ATIFROM), o IMUE(STUE(ATIFROM)) e o STUE(ATIFROM), obtendo, respectivamente, 27,32 dB, 26,95 dB e 26,94 dB. Nesse caso, todos os algoritmos propostos apresentaram desempenhos superiores aos algoritmos de referência.

Ao analisarmos a SAR, nas Tabelas 4.7 a 4.10, na condição de ausência de ruído ($\text{SNR} = \infty$) pode-se observar que nenhum algoritmo apresentou, com misturas anecoicas, desempenho similar ao obtido com misturas instantâneas, independentemente de quão bem os coeficientes tenham sido estimados. A principal razão para que isso ocorresse é que a reconstrução das fontes foi feita por meio de mascaramento *HARD*, acarretando em artefatos, decorrentes do processo natural de escolha da fonte ativa (em cada ponto no domínio da T-F).

Para $\text{SNR} = \infty$, independentemente do tipo de sinal pode-se verificar que os algoritmos IMUE(DUETm) e ATIFROM(WDFT) sempre estiveram entre os três com melhores desempenhos em relação à SAR. Pode-se também constatar que o algoritmo DUETm sempre foi o que apresentou o pior desempenho.

Nas Tabelas 4.7 e Tabela 4.9 para sinal de voz, e 4.8 para sinal de áudio pode-se observar que o algoritmo que apresentou menos artefatos para $\text{SNR} = 5$ dB foi o algoritmo IMUE, sendo que o IMUE(STUE(TIFROM_Anec)) e o IMUE(ATIFROM) estão sempre entre os três melhores. Portanto, os resultados com misturas de sinais de voz ou com ruído branco para o caso da $\text{SNR} = 5$ dB apresentaram consistência em relação aos dos algoritmos que obtiveram as melhores soluções. Para a mistura de sinais de áudio com ruído *babble*, os algoritmos que apresentaram as melhores soluções foram o ATIFROM(WDFT), o STUE(ATIFROM) e o ATIFROM(DFTU). Nesse caso, o efeito do ruído foi bem menos acentuado quando comparado ao dos demais casos, obtendo-se valores de SAR para $\text{SNR} = 5$ dB em torno de 19 dB com a maioria dos algoritmos, ao passo que para os demais casos a SAR ficou abaixo de 6,9 dB.

Capítulo 5

Algoritmos de Reconstrução das Fontes

Uma das técnicas mais disseminadas para a reconstrução das fontes em casos subdeterminados consiste em aplicar a operação de mascaramento. Um aspecto negativo do mascaramento binário (o qual assume que uma fonte encontra-se ativa ou inativa em um determinado ponto de uma dada representação) é que, ainda que se consiga estimar os coeficientes de atenuação e atraso com precisão na fase de análise, não há garantias de desempenho satisfatório na etapa de reconstrução das fontes. Esse problema é decorrente do fato de que nem sempre as fontes são disjuntas, tornando o mascaramento uma decisão que implica erros, não raro drásticos, na reconstrução.

A fim de não se acrescentar o erro proveniente da fase de estimação dos coeficientes da matriz de mistura com o erro oriundo da reconstrução, as matrizes de mistura empregadas neste capítulo são as utilizadas na formação da mistura e não as estimadas.

Um agravante para o caso subdeterminado decorre do fato do sistema ser mal-condicionado e admitir infinitas soluções, salvo em alguns contextos específicos¹, quando então o sistema se torna determinado. Assim, é comum acrescentar restrições ao problema subdeterminado visando reduzir a indeterminação inerente a esse contexto, conseqüentemente restringindo a busca empreendida pelos métodos a um subconjunto de soluções que pertençam a um espaço viável.

Este capítulo está dividido em três seções: a primeira descreve os principais métodos de reconstrução encontrados na literatura e que foram usados como referência nesta tese; a segunda descreve os métodos propostos neste trabalho para obter a reconstrução; e por fim, a terceira contém os resultados obtidos com os métodos propostos.

¹Quando, por exemplo, as fontes são totalmente disjuntas no domínio da transformada.

5.1 Principais Métodos de Reconstrução

Existem vários trabalhos de reconstrução de fontes, principalmente para o caso subdeterminado, eles são baseados nas mais diversas técnicas tais como *Matching Pursuit* [99], mascaramento *HARD* [23], mascaramento *SOFT* [100], minimização de funções [28], reconstrução por blocos [51], entre outras. Neste capítulo, a fim de se efetuar uma comparação justa dos resultados obtidos, elegeram-se, como referências, somente aquelas que são totalmente cegas. Dessas, temos duas técnicas distintas para misturas instantâneas e uma técnica para misturas anecoicas. Assim, foram empregadas como referências as técnicas de mascaramento *HARD* e a de módulo mínimo, para misturas instantâneas, ao passo que para misturas anecoicas somente se usou a técnica de mascaramento binário bidimensional.

5.1.1 Mascaramento

A abordagem mais empregada para o problema de reconstrução é denominada, nesse trabalho, *Mask2D* ou *Mask1D*, para mascaramento bidimensional (anecoico) e unidimensional (instantâneo), respectivamente. Nesta abordagem, assume-se que somente uma fonte se encontra ativa em cada ponto no domínio da transformada (implicando um mascaramento do tipo *HARD*), mesmo que isso não seja verdade. Outras implementações consideram a possibilidade de que mais de uma fonte esteja ativa num dado ponto; neste caso, tem-se um mascaramento *SOFT*.

O mascaramento pode ser entendido como um método de clusterização que possui dois graus de liberdade, a saber: o domínio onde ele ocorre e o critério de mascaramento a ser aplicado. Assim, é possível variar o domínio através de transformações (como STFT, CQT), e/ou empregar técnicas que visem minimizar o erro de decisão. Ambas as abordagens (critério de mascaramento a ser aplicado e domínio onde o mascaramento ocorre) minimizam o efeito causado pela indeterminação inerente do caso subdeterminado. Entretanto, deve-se ressaltar que somente em poucas situações (quando as fontes são completamente disjuntas no domínio da transformada) é possível se obter reconstrução perfeita das fontes.

No mascaramento *HARD*, caso a STFT seja aplicada, assume-se que, em cada ponto (τ, κ) , exista somente uma fonte ativa. Neste caso, é definida uma função *score* que, calculada para todas as fontes, tem o seu mínimo associado à fonte que com maior probabilidade de encontrar-se ativa.

A função custo que integra a solução proposta em [23] e representa um mascaramento binário bidimensional (*Mask2D*) para estimar as fontes presentes na mistura anecoica é dada por:

$$score_i(\tau, \kappa) = \frac{1}{1 + \alpha_i^2} |\alpha_i e^{-j\kappa(\tau, \kappa)\delta_i} X_1(\tau, \kappa) - X_2(\tau, \kappa)|^2, \quad (5.1)$$

onde α_i e δ_i representam a atenuação ² e o atraso da i -ésima fonte na mistura 1 em relação à mistura 2, enquanto que κ , $X_1(\tau, \kappa)$ e $X_2(\tau, \kappa)$ representam a frequência e as misturas no domínio T-F, respectivamente. É possível estabelecer um mascaramento binário unidimensional, *Mask1D*, empregando somente informações de um dos parâmetros (atenuação ou atraso).

A função $score_j$ será mínima se apenas uma das fontes estiver ativa num dado ponto (τ, κ) , e os coeficientes de atraso e atenuação estiverem corretamente estimados.

Após o processo de clusterização das representações no espaço da transformada feito pelo mascaramento binário em n agrupamentos, as fontes são reconstruídas no domínio do tempo através de um algoritmo de síntese [23, 101] aplicado a cada um desses agrupamentos.

A Fig. 5.1 mostra qual fonte está ativa no domínio T-F em uma mistura instantânea (segundo a Eq. (5.1)) formada por 3 fontes distintas. Cada cor está associada a uma fonte distinta, com o preto sinalizando a inatividade da fonte naquela representação. Nesse exemplo, não ocorrem superposições, ou seja, em cada ponto no domínio T-F, somente uma única fonte (representada por uma das cores vermelha, verde ou azul) está ativa.

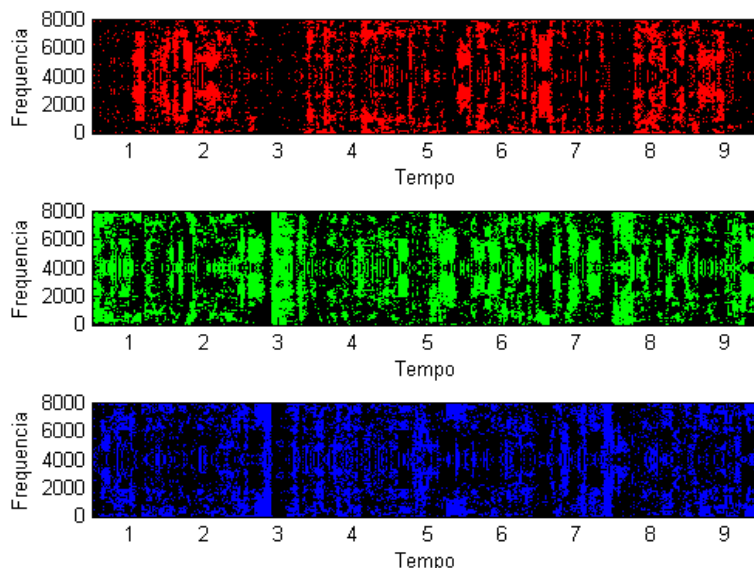


Figura 5.1: Mascaramento *HARD* para uma Mistura com 3 Fontes e 2 Sensores.

²Nesta abordagem, assume-se que as atenuações são positivas.

5.1.2 Módulo Mínimo

Nesta abordagem, para cada representação no espaço da transformada, pressupõe-se a existência de tantas fontes ativas quanto o número de sensores. Na busca da melhor solução, para cada ponto no domínio da transformada, resolvem-se todos os subsistemas de posto completo em busca daquele que minimiza a norma l_1 do conjunto de vetores solução. Essa abordagem também é conhecida como a do caminho mais curto [20].

Da mesma forma que para o mascaramento binário, é possível termos uma implementação *SOFT* e outra *HARD* para a estratégia de menor soma dos módulos. Na implementação *SOFT*, o número de fontes ativas se iguala ao número de sensores, ao passo que em uma implementação *HARD*, uma vez obtida a solução, somente a fonte de maior módulo será suposta ativa, enquanto as demais serão forçadas à inatividade.

Assim, caso seja aplicada a STFT, o índice do vetor de solução ótimo ot para um dado ponto (τ, κ) é obtido por

$$ot = \min_j (\|\underline{B}^j \underline{X}(\tau, \kappa)\|_1) \quad (5.2)$$

onde $\|\cdot\|_1$ representa a norma l_1 e \underline{B}^j representa a j -ésima submatriz de posto m , obtida da matriz de mistura \underline{A} , para $j = 1, \dots, C$, sendo C o número total de submatrizes distintas de posto m obtidas da matriz \underline{A} , e $\underline{X}(\tau, \kappa)$ representa o vetor de mistura num ponto (τ, κ) no domínio T-F.

As estimativas ótimas no domínio T-F são obtidas por

$$\underline{Y}^{ot}(\tau, \kappa) = \underline{B}^{ot} \underline{X}(\tau, \kappa) \quad (5.3)$$

A Fig. 5.2 representa a solução *HARD* obtida pelo algoritmo de módulo mínimo, a partir da mesma mistura com três fontes distintas exemplificada anteriormente para o mascaramento *HARD*. Nesse exemplo, com 3 fontes e 2 sensores, não ocorrem superposições entre as fontes ativas para um dado ponto (τ, κ) .

A Fig. 5.3 representa a solução *SOFT* obtida pelo algoritmo de módulo mínimo, para o mesmo caso. Nesse exemplo, diferentemente dos anteriores, ocorrem superposições entre as fontes ativas para cada ponto (τ, κ) .

Nos experimentos realizados (caso instantâneo), observa-se que, normalmente, as soluções do tipo mascaramento *HARD* apresentam valores de SIR melhores que os obtidos pelo mascaramento *SOFT*, salvo nos pontos onde as fontes ativas foram estimadas corretamente e o número delas é exatamente igual ao número de sensores. Já o mascaramento *SOFT* normalmente apresenta um valor de SDR melhor que o mascaramento *HARD*. Tal fato se deve basicamente à presença de uma fonte

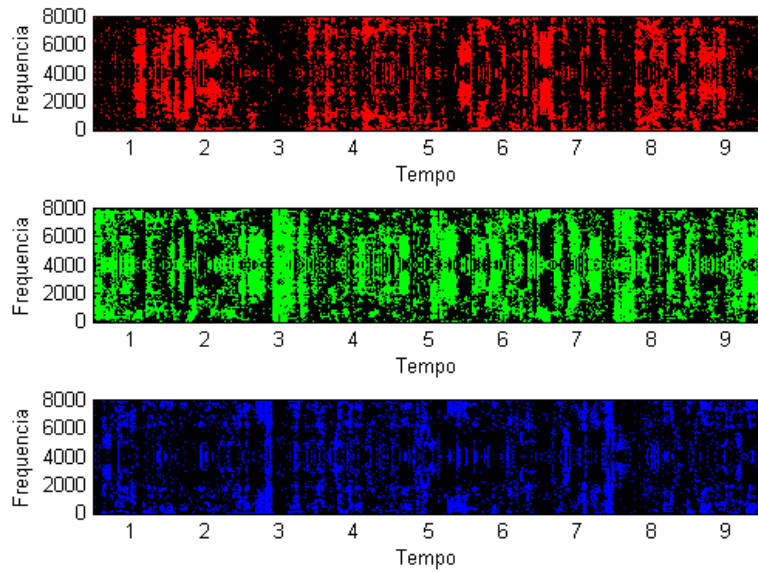


Figura 5.2: Módulo Mínimo do Tipo *HARD* para uma Mistura com 3 Fontes e 2 Sensores.

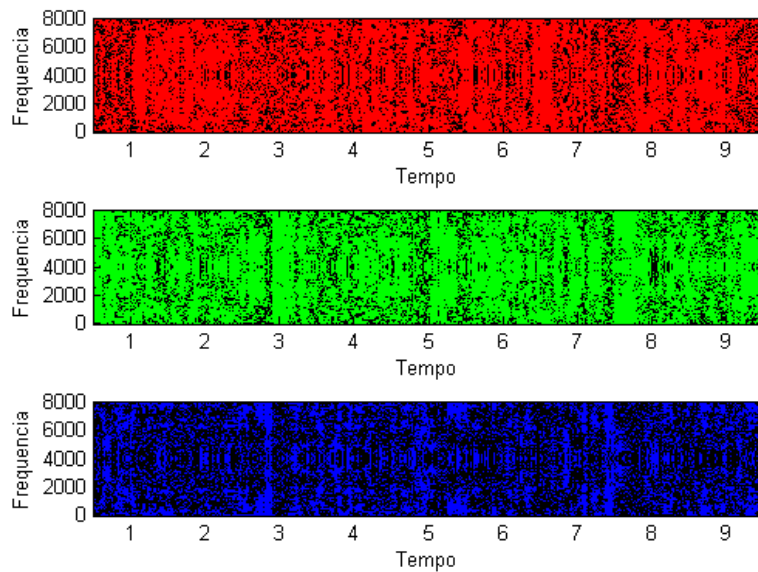


Figura 5.3: Módulo Mínimo do Tipo *SOFT* para uma Mistura com 3 Fontes e 2 Sensores.

secundária nesse processo de reconstrução, o que não ocorre com o mascaramento *HARD*, que na maioria das vezes superestima o valor da magnitude da fonte ativa primária em cada ponto no domínio da transformada.

5.2 Métodos Propostos

Nesta seção, propomos quatro métodos para reconstrução das fontes para o caso subdeterminado. Os métodos denominados deflação na frequência e mascaramento corrigido foram aplicados tanto às misturas anecoicas quanto instantâneas. Já os métodos denominados de correlação mínima e subtração temporal foram implementados somente para misturas instantâneas e anecoicas, respectivamente.

O caso determinado ficou restrito às misturas anecoicas, para matrizes com dimensão 2×2 . Nesse caso, o mascaramento corrigido e a deflação em frequência são avaliados conjuntamente com mais um método proposto para pequenos atrasos, denominado de solução aproximada. Usaram-se os resultados obtidos pelo mascaramento binário e a solução exata como referências de desempenho.

5.2.1 Deflação na Frequência

A primeira proposta de reconstrução, denominada de DEFL_F , consiste em efetuar um processo de deflação em sequência, via mascaramento binário das fontes, após deflacionarmos as misturas originais no domínio T-F, a partir de uma estimativa de uma das fontes.

Após a deflação no domínio da transformada, as misturas são reconstruídas no domínio do tempo, para se aplicar o mesmo procedimento recursivamente, até que o sistema se torne determinado, quando então se pode usar uma técnica distinta, específica ou não ao caso determinado.

Essa abordagem difere da deflação obtida via mascaramento no domínio T-F, a qual assume também que as fontes são WDO (*W-disjoint orthogonal*), ou que pelo menos $n - m$ fontes sejam WDO. A abordagem proposta funciona como um processo de deflação em sequência, em contraposição a um processo de deflação simultâneo, quando para cada agrupamento se zera todos os pontos, no domínio T-F, que não pertencente ao agrupamento (associado a uma dada fonte) em questão (máscara).

Neste contexto, o principal problema dessa proposta consiste em arbitrar qual estimativa de fonte será empregada para deflacionar as misturas. Para contornar esse problema, o critério empregado na escolha da melhor estimativa a ser usada adere ao princípio de que “o conjunto de amostras associadas a cada fonte que detém a menor variância da função *score* será considerado a melhor escolha”.

Assim, o agrupamento associado à fonte que contiver amostras mais homogêneas será usado no início do processo de deflação. Esse critério foi adotado assumindo que os agrupamentos menos homogêneos contêm amostras contaminadas pela presença de outras fontes ou associadas indevidamente ao agrupamento. Em ambas as situações, essas amostras contribuirão para o erro da reconstrução das fontes. Por outro lado, agrupamentos homogêneos que contêm poucas amostras podem ser

resultado do fato de que várias amostras da fonte correspondente ao agrupamento em questão foram associadas aos demais agrupamentos. Apesar de tal ocorrência não ter sido levada em consideração, pode-se pensar na formação de um critério que considere essa possibilidade.

Para se obter o conjunto de amostras associadas a cada fonte, utilizou-se a função *score* (5.1) de mascaramento binário [23], que mede a discrepância de um ponto a cada coeficiente de atenuação³ para determinar a qual fonte a amostra pertence.

Assim, esse critério, apesar de usar a mesma função empregada na fase de análise para estimar os coeficientes de misturas, possui a sua medida menos dependente da estimação correta desses coeficientes, uma vez que a dispersão dos dados se dá em relação às médias dos pares contidos em cada agrupamento e não dos coeficientes de atenuação (e atraso no caso anecoico) estimados.

Portanto, a equação seguinte estipula qual fonte deve ser usada para deflacionar as misturas

$$ot = \min_{j=1,\dots,n} [\sigma^2(score_{\hat{s}_j})] \quad (5.4)$$

onde *ot* representa o índice da fonte a ser deflacionada e $\sigma^2(score_{\hat{s}_j})$ representa a variância (no domínio tempo-frequência) do vetor contendo todos os elementos da função (5.1), associados à *j*-ésima fonte estimada ($score_{\hat{s}_j}$).

A Fig. 5.4 exemplifica a solução obtida pelo algoritmo de Deflação em Frequência para misturas instantâneas, o que representa uma solução do tipo *SOFT*. Isso ocorre porque mais de uma fonte pode estar ativa para um dado ponto no domínio T-F, uma vez que ao se reconstruir a mistura deflacionada por uma das fontes no domínio do tempo, os pontos (τ, κ) que estavam inicialmente assinalados à fonte que foi retirada passarão a ser associados à outra fonte ativa, quando for feito novamente o mascaramento sobre a mistura deflacionada.

Essa diferença sutil entre o método proposto e o mascaramento binário simultâneo representa uma alteração no número de representações a serem distribuídas entre as “fontes” (agrupamentos). Portanto, o mascaramento binário distribui os *P* pontos da mistura no domínio da transformada pelas *n* “fontes”, ao passo que no processo em sequência isso não ocorre, devido ao fato da mistura ser reconstruída no domínio temporal. Assim, após cada retirada de fonte, novamente se terá *P* pontos para distribuir pelas fontes restantes. Esse procedimento, como consequência, faz com que, haja superposições das fontes ativas no domínio da transformada, o que não ocorre no mascaramento do tipo *HARD*.

A principal vantagem esperada pelo processo de deflação em sequência contra

³Para misturas anecoicas, se computa a discrepância de um ponto com relação ao par (atenuação, atraso).

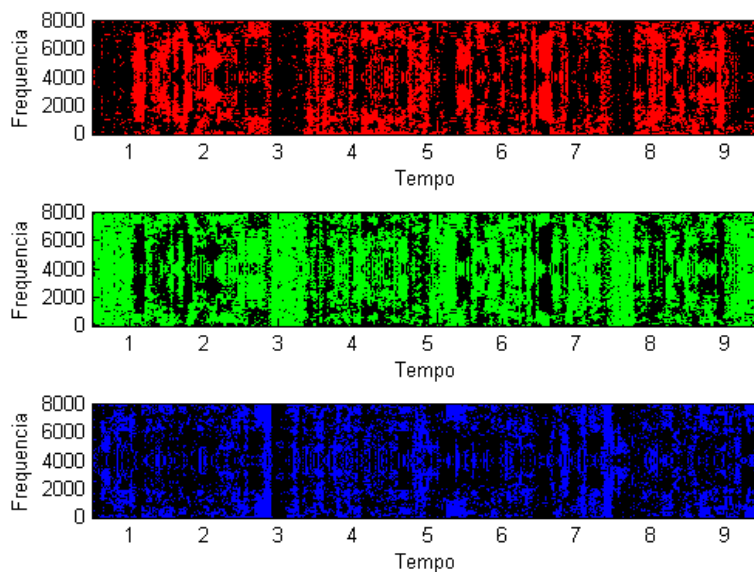


Figura 5.4: Deflação em Frequência para uma Mistura com 3 Fontes e 2 Sensores.

o processo de mascaramento binário é que, por hipótese, uma escolha adequada das fontes que serão deflacionadas, em geral, resultará num sistema reduzido mais próximo do sistema real, sem a presença daquela fonte. Isto possibilitará a obtenção de estimativas mais próximas das originais, a partir do sistema deflacionado. No entanto, caso a escolha inicial seja inadequada, o sistema reduzido poderá ficar muito distante do ideal, o que resultará numa piora nas estimativas em comparação ao mascaramento binário.

5.2.2 Correlação Mínima

Essa técnica foi inspirada no princípio da baixa correlação entre fontes distintas. Portanto, a validade do método está restrita a sinais que respeitam essa propriedade.

O algoritmo proposto reside numa busca de um subconjunto de fontes ativas que melhor representa a mistura no domínio da transformada, por meio da minimização das correlações cruzadas.

O problema de se minimizar a função correlação é que amostras em instantes distintos são parâmetros de entrada dessa função, tornando alto o custo computacional para o seu cálculo, e representando um obstáculo no processo de definição de quais fontes estão ativas num dado ponto no domínio da transformada. Uma forma de diminuir esse custo computacional consiste na redução do número de amostras dos sinais a serem correlacionadas. Esse procedimento pode ser feito por meio da segmentação do sinal em *frames* menores.

Embora o cálculo da correlação entre *frames* se revele mais viável, ainda per-

manece bastante custoso, uma vez que é necessário calcular a correlação para um conjunto de representações temporais para cada raia (se atentarmos para a persistência temporal), e, mesmo não se levando em conta o custo computacional, tal abordagem não garante a obtenção do resultado ótimo.

Sendo assim, experimentou-se o caso extremo baseado em uma única representação no domínio T-F, escolhendo para cada ponto (τ, κ) dentro das soluções obtidas por cada submatriz de posto m , aquela que minimiza a seguinte função, baseada na correlação

$$ot = \min_j (|Y_j^p(Y_j^q)^* + Y_j^q(Y_j^p)^*|) \quad (5.5)$$

onde “*” representa o operador conjugado, p e q representam os índices das duas fontes ativas no ponto (τ, κ) , obtidas por

$$[Y_j^p \ Y_j^q]^T = \underline{B}^j \underline{X}(\tau, \kappa) \quad (5.6)$$

onde \underline{B}^j representa o número de submatrizes de posto completo da matriz de mistura estimada \hat{H} .

Dessa forma, num dado *frame*, raias distintas podem ter distintas fontes ativas. O vetor solução das estimativas das fontes (dimensão n) na representação (τ, κ) é dado por

$$\underline{Y}^{ot}(\tau, \kappa) = \underline{B}^{ot} \underline{X}(\tau, \kappa) \quad (5.7)$$

A Fig. 5.5 representa a solução do tipo *HARD* obtida pelo algoritmo de correlação mínima. Nesse exemplo, não ocorrem superposições entre as fontes ativas para um dado ponto (τ, κ) .

A Fig. 5.6 representa a solução do tipo *SOFT* obtida pelo algoritmo de correlação mínima para o mesmo caso anterior. Nesse exemplo, ocorrem superposições entre as fontes ativas para um dado ponto (τ, κ) .

Conforme se pode observar, para a mesma mistura, os algoritmos obtêm soluções distintas quanto à atividade das fontes nas representações (τ, κ) no domínio T-F, demonstrando as diferentes operações de reconstrução empreendidas pelos métodos propostos.

5.2.3 Solução Exata

Para duas misturas $x_1(t)$ e $x_2(t)$ com $t = 1, \dots, P$, a solução exata para o caso anecoico determinado 2×2 no domínio da frequência⁴ pode ser obtida por

⁴Uma solução equivalente pode ser facilmente demonstrada para o domínio T-F.

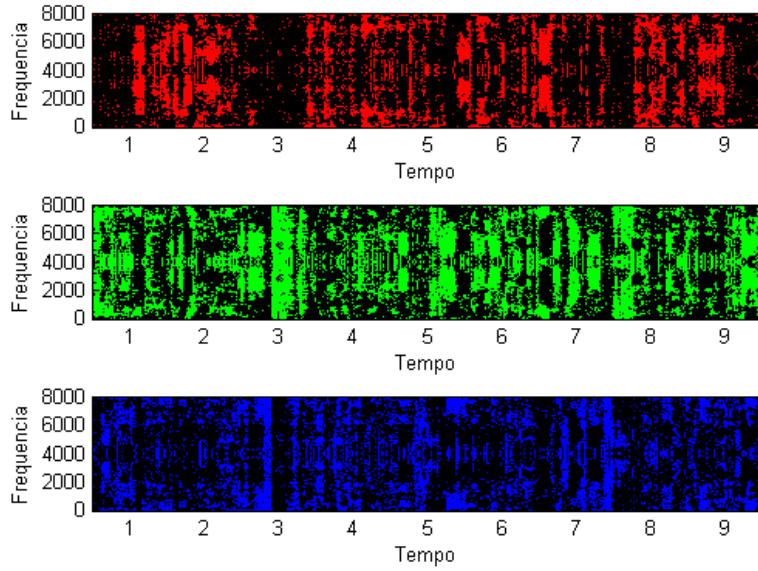


Figura 5.5: Correlação Mínima do Tipo *HARD* para uma Mistura com 3 Fontes e 2 Sensores.

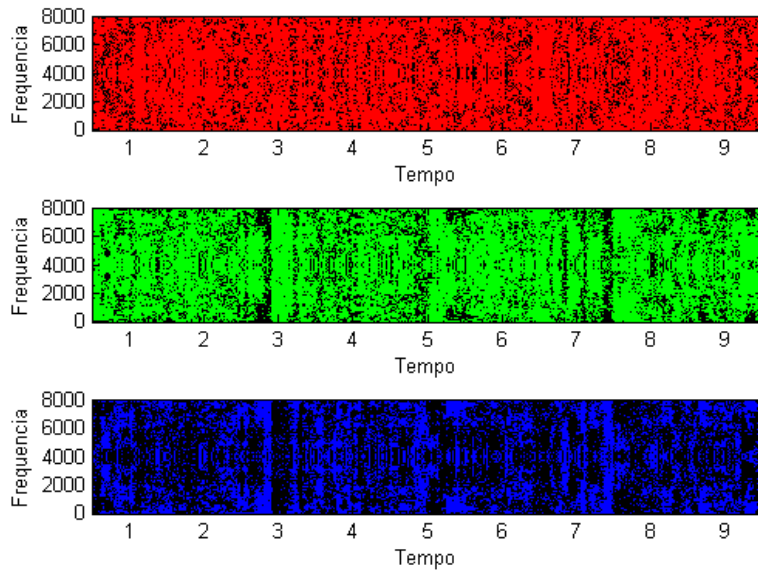


Figura 5.6: Correlação Mínima do Tipo *SOFT* para uma Mistura com 3 Fontes e 2 Sensores.

$$\hat{S}_1(f) = \frac{\alpha_2 e^{-\delta_2 \omega j} X_1(f) - X_2(f)}{\alpha_2 e^{-\delta_2 \omega j} - \alpha_1 e^{-\delta_1 \omega j}} \quad (5.8)$$

$$\hat{S}_2(f) = \frac{X_2(f) - \alpha_1 e^{-\delta_1 \omega j} X_1(f)}{\alpha_2 e^{-\delta_2 \omega j} - \alpha_1 e^{-\delta_1 \omega j}} \quad (5.9)$$

onde $X_1(f)$ e $X_2(f)$ são as transformadas DFTs das misturas.

Portanto, o problema no caso determinado 2×2 é estimar as matrizes de atenuação (\underline{M}) e de atraso (\underline{A}), dadas por

$$\underline{M} = \begin{bmatrix} 1 & 1 \\ \alpha_1 & \alpha_2 \end{bmatrix} \quad (5.10)$$

$$\underline{A} = \begin{bmatrix} 0 & 0 \\ \delta_1 & \delta_2 \end{bmatrix} \quad (5.11)$$

5.2.4 Solução Aproximada

Uma solução algébrica para o caso determinado anecoico (2×2), que evita a transformação dos sinais para o domínio da transformada, considera estimativas aproximadas, de forma que $\hat{s}_1(t)$ seja função somente de $s_1(t)$, ou seja, $\hat{s}_1(t) = f(s_1(t))$, e $\hat{s}_2(t)$ seja função somente de $s_2(t)$, ou seja, $\hat{s}_2(t) = f(s_2(t))$. A partir das misturas, obtêm-se as estimativas aproximadas das fontes dadas por

$$\hat{s}_1(t) = \alpha_2 x_1(t) - x_2(t + \delta_2) \quad (5.12)$$

$$\hat{s}_2(t) = \alpha_1 x_1(t) - x_2(t + \delta_1) \quad (5.13)$$

onde facilmente se chega a

$$\hat{s}_1(t) = \alpha_2 s_1(t) - \alpha_1 s_1(t - (\delta_1 - \delta_2)) \quad (5.14)$$

$$\hat{s}_2(t) = \alpha_1 s_2(t) - \alpha_2 s_2(t - (\delta_2 - \delta_1)) \quad (5.15)$$

Caso a diferença entre os atrasos seja pequena ($|\delta_2 - \delta_1| \approx 0$) e $\alpha_1 \neq \alpha_2$, teremos

$$\hat{s}_1(t) \approx (\alpha_2 - \alpha_1) s_1(t) \quad (5.16)$$

$$\hat{s}_2(t) \approx (\alpha_1 - \alpha_2) s_2(t) \quad (5.17)$$

Portanto, misturas anecoicas que contenham 2 sensores e 2 fontes podem ser resolvidas de forma satisfatória através de métodos algébricos, desde que se tenha uma boa estimativa da matriz de atenuação e da matriz de atraso.

Outra possibilidade apela a um mascaramento binário. Nesse tipo de abordagem é assumido que somente uma fonte se encontra ativa num dado momento no domínio da transformada. Evidentemente que a restrição de que as fontes são WDO

no domínio da transformada introduz uma perda de desempenho na reconstrução das fontes. Contudo, tal restrição permite o emprego desta técnica no caso subdeterminado.

5.2.5 Mascaramento Corrigido

Outra abordagem apresentada neste trabalho, e batizada de mascaramento corrigido (MaskCorr), consiste em, recursivamente, usar as misturas como um estimador do erro obtido pela separação, para tentar melhorar a estimativa obtida pelo mascaramento.

Como a solução via mascaramento não é exata em cada ponto, então as estimativas obtidas do mascaramento (empregando as matrizes de atenuação e atraso) não reproduzem as fontes originais. Essa característica permite que os erros obtidos pela diferença entre as misturas reconstruídas (a partir de suas estimativas) e as misturas originais possam ser reduzidos mascarando-se sucessivamente os erros, e acrescentando-se essas novas estimativas obtidas pelos erros mascarados às estimativas originais.

Assim, inicialmente faz-se um mascaramento a partir das misturas originais, ou seja

$$\hat{y}_1 = \text{Mask2D}(x, \underline{M}, \underline{A}) \quad (5.18)$$

Em seguida, obtêm-se as misturas reconstruídas $\hat{x} = [\hat{x}_1, \dots, \hat{x}_m]^T$, a partir das estimativas, $\hat{y}_1(t), \dots, \hat{y}_n(t)$, obtidas pelo mascaramento, usando-se os coeficientes de atenuação (a_{ij}) e de atraso (d_{ij}) encontrados na fase de análise, ou seja

$$\begin{bmatrix} \hat{x}_1(t) \\ \vdots \\ \hat{x}_m(t) \end{bmatrix} = \begin{bmatrix} a_{11}\hat{y}_1(t - d_{11}) & + \dots + & a_{1n}\hat{y}_n(t - d_{1n}) \\ & \vdots & \\ a_{m1}\hat{y}_1(t - d_{m1}) & + \dots + & a_{mn}\hat{y}_n(t - d_{mn}) \end{bmatrix} \quad (5.19)$$

Os fatores de correção de escala μ_i , $i = 1, \dots, m$ podem ser determinados de diversas formas. Neste trabalho, empregou-se a equação

$$\mu_i = \frac{\sum_{t=1}^P |x_i(t)|}{\sum_{t=1}^P |\hat{x}_i(t)|} \quad (5.20)$$

sendo P o número total de *frames*.

O erro entre as misturas reconstruídas e a mistura original, para o caso determinado com dois sensores, pode ser calculado por

$$\epsilon_i(t) = x_i(t) - \mu_i \hat{x}_i(t) \quad (5.21)$$

para $i = 1, \dots, m$.

A partir desse ponto, é calculado o mascaramento do erro \underline{y}_ϵ , dado por⁵

$$\hat{\underline{y}}_\epsilon = [y_{\epsilon_1}, \dots, y_{\epsilon_n}]^T = \text{mask2D}(\underline{\epsilon}, \underline{M}, \underline{A}) \quad (5.22)$$

onde $\underline{\epsilon} = [\epsilon_1, \dots, \epsilon_m]^T$, \underline{M} é a matriz de atenuação e \underline{A} é a matriz de atraso empregadas na mistura anecoica.

Por fim, a equação de atualização é dada por

$$\hat{\underline{y}}_{k+1} = \hat{\underline{y}}_k + \hat{\underline{y}}_\epsilon \quad (5.23)$$

onde $\hat{\underline{y}}_\epsilon$ contém as estimativas obtidas pelo mascaramento do erro, dado pela Eq. (5.22)

O critério de parada é obtido adotando-se um valor máximo para a média da diferença quadrática entre as estimativas e as misturas, dada por

$$\epsilon_{total} = \frac{1}{P} \sum_{t=1}^P \sum_{i=1}^m \epsilon_i(t)^2 \quad (5.24)$$

5.2.6 Subtração Temporal

Nesta seção, veremos mais três abordagens propostas, denominadas TSOE_{TF} (*Temporal Subtraction using One Estimate in TF domain*), TSAE_{T1} (*Temporal Subtraction using All Estimates in Time domain (mode 1)*) e TSAE_{T2} (*Temporal Subtraction using All Estimates in Time domain (mode 2)*), que fazem subtração no domínio do tempo. O principal ganho dessas abordagens em comparação com a deflação no domínio T-F apresentada anteriormente nesta tese, é que não é necessário forçar, via mascaramento binário, que a fonte a ser deflacionada das misturas seja disjunta em relação às demais fontes. Portanto, o processo de subtração no tempo não é equivalente ao mascaramento binário.

Subtraindo do Sistema Uma Única Estimativa

Nessa abordagem, em um primeiro passo, pode ser usada a função *score* como definida na Eq. (5.1), para estabelecer qual fonte é a melhor para o processo de subtração. A subtração não ocorre no domínio T-F, mas emprega o modelo para a subtração no domínio do tempo para o caso anecoico, descrito por

$$x'_1(t) = x_1(t) - \mu \hat{y}_j(t) \quad (5.25)$$

$$x'_2(t) = x_2(t) - \mu \alpha_j \hat{y}_j(t - \delta_j) \quad (5.26)$$

⁵Para misturas instantâneas é feito um mascaramento unidimensional.

onde μ é um fator de escala a se determinar e j pode ser obtido pela Eq. (5.4). Determinada a fonte \hat{y}_j a ser usada na deflação, ficam também estabelecidos os seus respectivos atrasos (δ_j) e atenuações (α_j).

O problema agora é determinar o melhor fator de escala μ que reduz o sistema. Para isso, monta-se um mapeamento entre o fator de escala μ e a expectativa de baixa correlação das fontes estimadas, combinando-se a função

$$F(\mu) = \sum_{i=1}^n \sum_{j=1}^n |E[\hat{s}_i \hat{s}_j]| - n \quad (5.27)$$

com o procedimento apresentado a seguir, onde $\hat{s}_i = \mu \hat{y}_i$. A partir de um conjunto de estimativas iniciais \hat{y}_i e estabelecida a fonte j a ser subtraída das misturas, temos o mapeamento com a seguinte sequência de passos:

Passo 1 A partir de um dado μ , obter a estimativa da fonte j como $\hat{s}_j = \mu \hat{y}_j$;

Passo 2 Reduzir as misturas, conforme as Eqs. (5.25) e (5.26);

Passo 3 Reduzir as matrizes de atenuação e atraso, eliminando a j -ésima coluna dessas matrizes;

Passo 4 Obter as demais estimativas \hat{s}_i , resolvendo o sistema⁶ com as matrizes reduzidas;

Passo 5 Encontrar o valor de μ mínimo para o mapeamento $F(\mu)$ conforme a Eq. (5.27).

A vantagem dessa abordagem é que, caso existam $n - 2$ estimativas iniciais razoáveis, é possível obter solução para o caso subdeterminado sem necessitar do mascaramento, ou seja, a imposição das fontes serem disjuntas é dispensada.

Logo, o algoritmo proposto de subtração no tempo se vale de uma técnica de busca de mínimo para o mapeamento $F(\mu)$. Em outras palavras, recorre-se a um algoritmo de otimização não linear sem restrições. Para isso, empregou-se um algoritmo que implementa um método de busca direta [102].

O valor inicial do fator de escala μ foi estimado reconstruindo as misturas a partir das estimativas iniciais (\hat{y}_i) junto com os coeficientes de atraso e atenuação, que foram obtidos na fase de análise. Posteriormente, efetuou-se uma busca no domínio da frequência pelos pontos SSP (*Single Source Points*) associados à j -ésima fonte que fora escolhida para ser empregada no processo de deflação. Uma vez obtidos os pontos SSP (a partir das Eqs. (2.16), (2.17), (2.18)), verificou-se qual

⁶Aqui pode ser usado outro método ou um método recursivo, até obter-se um sistema determinado.

ponto apresentava o valor absoluto da razão entre as misturas mais próximo do coeficiente de atenuação associado à fonte j . O valor inicial do fator de escala μ_0 é então determinado, calculando-se, nesse ponto ótimo do espectro, a razão da mistura obtida pelo sensor com a mistura equivalente reconstruída.

Essa proposta combina a baixa correlação entre as fontes e a esparsidade no domínio da transformada com a subtração no tempo para reconstruir as fontes no caso anecoico.

Na abordagem TSOE_{TF} , a primeira redução da ordem do sistema é uma subtração feita no domínio do tempo, ao passo que nas demais reduções ocorrem no domínio tempo-frequência, conforme a abordagem DEFL_{F} . Nesse algoritmo, o fator de escala μ é otimizado conforme o método descrito anteriormente somente para a primeira subtração.

Subtraindo do Sistema Todas as Estimativas

A terceira e a quarta abordagens, denominadas TSAE_{T1} e TSAE_{T2} , não estabelecem o melhor candidato para deflacionar o sistema de misturas, ou seja, não se supõe que nenhuma das fontes seja melhor do que as demais para se deflacionar as misturas. Assim, todas as fontes são empregadas no processo de deflação, uma a uma, de forma que, ao final, somam-se as estimativas correlacionadas obtidas por cada sistema deflacionado formando um novo conjunto de estimativas, assumindo que entre as estimativas equivalentes as diferenças se subtraem e as coincidências se somam.

Na abordagem TSAE_{T1} todas as reduções da ordem do sistema ocorrem no domínio do tempo. Contudo, o fator de escala μ não é otimizado, sendo empregado o valor μ_0 descrito anteriormente.

Na abordagem TSAE_{T2} as subtrações são todas feitas no domínio do tempo. A redução da ordem do sistema ocorre até o sistema se tornar determinado. Isso se dá por meio de deflações recursivas, o que resulta numa implementação cujo custo computacional é elevado. Nesse procedimento o valor de μ é otimizado.

5.3 Desempenhos dos Algoritmos de Reconstrução

Os dados empregados nas simulações apresentadas nesta seção são compostos por dois conjuntos distintos de sinais descritos no Apêndice A: sinais de voz em inglês [98] com 10 s de duração, e amostrados com a taxa de 16.000 amostras/s e sinais de áudio (flautas) [34] amostrados à taxa de 8.000 amostras/s com 2 s de duração, sendo que, na análise de convergência, foram empregadas somente duas das quatro flautas [40]

As medidas de desempenho empregadas nesta seção para analisar o desempenho dos algoritmos foram a média da SDR, SAR e SIR.

5.3.1 Avaliações da Convergência e da Recursividade

Esta seção tem o propósito de descrever características específicas dos métodos de subtração temporal e mascaramento corrigido, mostrando a possibilidade de se melhorar estimativas com recursões, até que se chegue a um determinado limite de desempenho.

Convergência

Nas Figs. 5.7 a 5.9 são apresentadas as evoluções da SDR, SIR e SAR (em função do número de iterações) do método de correção do mascaramento, para sinais de áudio no caso anecoico com dois sensores, tendo atenuações e atrasos de $(0, 3, 1, 5)$ e $(3, 5, -1, 5)$, respectivamente.

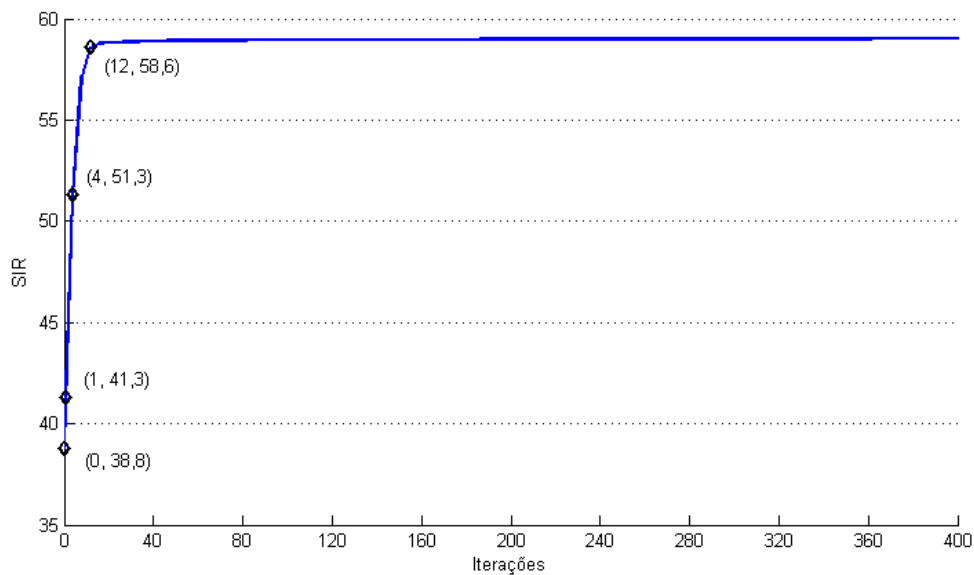


Figura 5.7: Curva de Evolução da SIR Obtida pelo Algoritmo *MaskCorr*.

As curvas mostram evoluções consistentes da SIR, SDR e SAR de 0 a 400 iterações, tendo o maior ganho ocorrido nas primeiras 12 iterações. A solução obtida pelo mascaramento binário resultou em valores de SIR, SDR e SAR, respectivamente, de 38,78 dB, 31,29 dB e 32,15 dB. Logo, o ganho na décima segunda iteração em relação à solução obtida pelo mascaramento foi de 19,80 dB, 13,89 dB e 13,51 dB, respectivamente, ao passo que o ganho até a iteração 400 em relação à iteração 12 foi de 0,44 dB, 0,25 dB e 0,23 dB, respectivamente.

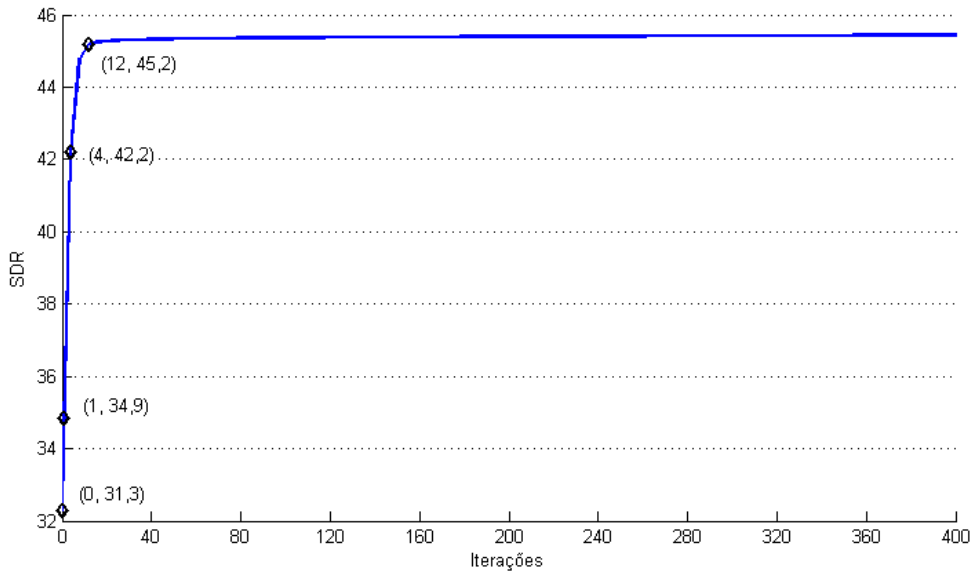


Figura 5.8: Curva de Evolução da SDR Obtida pelo Algoritmo *MaskCorr*.

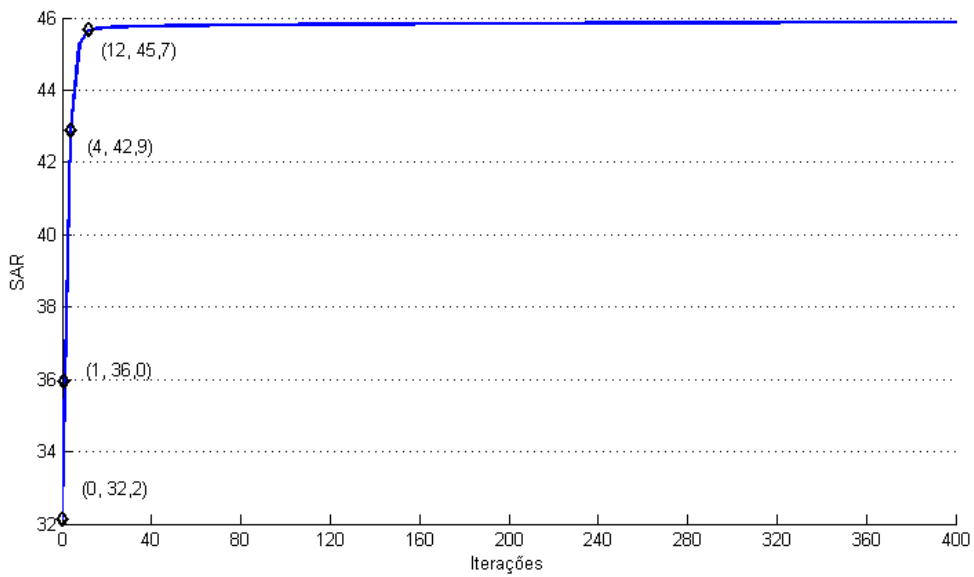


Figura 5.9: Curva de Evolução da SAR Obtida pelo Algoritmo *MaskCorr*.

Recursão

Algumas das técnicas sugeridas neste capítulo podem empregar recursivamente os resultados obtidos como soluções iniciais para uma nova correção nas estimativas. Pode-se usar estimativas iniciais distintas do mascaramento binário, o que pode resultar em um desempenho final com um aumento do custo computacional.

A fim de ilustrar essa possibilidade, as Tabelas 5.1 e 5.2 contêm os valores da SIR, SDR e SAR obtidos com sinal de áudio e voz, respectivamente, utilizando a

recursão supracitada com a inicialização pela solução obtida pelo algoritmo anterior.

#	Algoritmo	Sol. Inicial	SIR	SDR	SAR
01	<i>Mask2D</i>	-	33,87	28,50	30,24
02	TSOE _{TF}	01	34,82	29,85	31,71
03	TSAE _{T1}	02	36,12	30,92	32,80
04	TSAE _{T1}	03	36,67	31,35	33,14
05	TSAE _{T1}	04	37,10	31,56	33,29

Tabela 5.1: Algoritmos com Recursão - Áudio - 3 Fontes.

#	Algoritmo	Sol. Inicial	SIR	SDR	SAR
01	<i>Mask2D</i>	-	17,71	7,88	8,72
02	TSOE _{TF}	01	18,80	8,51	9,09
03	TSAE _{T1}	02	19,33	8,75	9,31
04	TSAE _{T1}	03	19,65	8,83	9,35
05	TSAE _{T1}	04	19,73	8,87	9,39

Tabela 5.2: Algoritmos com Recursão - Voz em Inglês - 3 fontes.

Conforme se pode observar, os ganhos entre as medidas a cada recursão são decrescentes, o que nos leva a concluir que existe um limite a partir do qual novas iterações produzirão ganhos inexpressivos.

5.3.2 Resultados com Misturas Instantâneas - Caso Subdeterminado

As simulações descritas nesta seção foram realizadas com 30 matrizes 2×3 e 2×4 geradas aleatoriamente, cujos coeficientes de atenuação variaram no intervalo $[-3, 0, 5, 0]$. Tanto as misturas quanto as estimativas das fontes foram obtidas a partir dessas matrizes. A ideia é verificar o desempenho do algoritmo de reconstrução caso os coeficientes do sistema de mistura sejam identificados corretamente. Nas Tabelas 5.3, a 5.10 são apresentados os resultados médios obtidos para esse cenário.

A partir da Tabela 5.3, todas as tabelas terão destacados em vermelho os resultados obtidos abaixo do mascaramento binário usado como referência (unidimensional para o caso instantâneo e bidimensional para o caso anecoico), que se encontrará destacada em verde. Além disso, estará também destacado em azul o melhor resultado obtido para uma dada medida. Tal procedimento visa facilitar a leitura das tabelas e auxiliar a análise dos resultados.

Os algoritmos baseados na correção de erro via mascaramento presentes nessas tabelas são basicamente uma simplificação do algoritmo descrito neste capítulo para o caso anecoico. Em todos os casos em que ele foi empregado, utilizaram-se 12 iterações.

#	Algoritmo	Tipo	SDR	SIR	SAR
01	<i>Mask1D</i>	-	11,91	22,05	12,51
02	MaskCorr	-	18,26	23,60	20,53
03	ModMin	<i>HARD</i>	13,20	24,50	13,74
04	ModMin	<i>SOFT</i>	18,58	22,15	21,48
05	DEFL _F	-	11,90	22,08	12,51
06	CorrMin	<i>HARD</i>	13,02	24,35	13,46
07	CorrMin	<i>SOFT</i>	19,07	23,56	21,18

Tabela 5.3: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Voz em Inglês - 3 Fontes.

#	Algoritmo	Tipo	SDR	SIR	SAR
01	<i>Mask1D</i>	-	7,14	18,38	8,23
02	MaskCorr	-	9,40	17,80	10,46
03	ModMin	<i>HARD</i>	8,40	18,72	10,68
04	ModMin	<i>SOFT</i>	10,14	12,74	16,63
05	DEFL _F	-	7,14	18,46	8,22
06	CorrMin	<i>HARD</i>	8,95	19,56	9,70
07	CorrMin	<i>SOFT</i>	13,34	18,68	15,00

Tabela 5.4: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Voz em Inglês - 4 Fontes.

#	Algoritmo	Tipo	SDR	SIR	SAR
01	<i>Mask1D</i>	-	13,55	19,15	15,52
02	MaskCorr	-	18,83	21,27	23,79
03	ModMin	<i>HARD</i>	14,40	20,48	16,24
04	ModMin	<i>SOFT</i>	19,08	21,02	24,23
05	DEFL _F	-	13,65	19,29	15,70
06	CorrMin	<i>HARD</i>	14,44	20,55	16,17
07	CorrMin	<i>SOFT</i>	19,31	21,51	24,13

Tabela 5.5: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Áudio - 3 Fontes.

#	Algoritmo	Tipo	SDR	SIR	SAR
01	<i>Mask1D</i>	-	10,90	15,00	14,08
02	MaskCorr	-	11,99	15,21	15,67
03	ModMin	<i>HARD</i>	11,76	17,19	15,29
04	ModMin	<i>SOFT</i>	12,57	13,75	21,82
05	DEFL _F	-	10,99	14,88	14,36
06	CorrMin	<i>HARD</i>	14,03	19,48	16,33
07	CorrMin	<i>SOFT</i>	16,43	18,62	21,03

Tabela 5.6: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Áudio - 4 Fontes.

#	Algoritmo	Tipo	SDR	SIR	SAR
01	<i>Mask1D</i>	-	12,03	23,69	12,77
02	MaskCorr	-	15,89	25,03	20,14
03	ModMin	<i>HARD</i>	12,73	25,27	13,60
04	ModMin	<i>SOFT</i>	15,69	22,88	20,89
05	DEFL _F	-	12,15	24,02	12,85
06	CorrMin	<i>HARD</i>	12,68	25,21	13,44
07	CorrMin	<i>SOFT</i>	16,62	24,46	20,80

Tabela 5.7: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Voz em Japonês - 3 Fontes.

#	Algoritmo	Tipo	SDR	SIR	SAR
01	<i>Mask1D</i>	-	7,21	18,08	8,01
02	MaskCorr	-	9,22	17,54	10,31
03	ModMin	<i>HARD</i>	8,23	19,71	10,29
04	ModMin	<i>SOFT</i>	10,57	12,72	18,36
05	DEFL _F	-	7,26	18,19	8,03
06	CorrMin	<i>HARD</i>	8,99	20,75	9,64
07	CorrMin	<i>SOFT</i>	13,67	18,68	15,55

Tabela 5.8: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Voz em Japonês - 4 Fontes.

Conforme se pode observar, nas Tabelas 5.3 a 5.10, o algoritmo proposto CorrMin (*HARD* ou *SOFT*) teve um desempenho superior ao mascaramento para todas as medidas e para todos os sinais empregados.

Além disso, pode-se também observar que o algoritmo proposto CorrMin (*SOFT*) apresentou o melhor desempenho médio para a SDR para todos os sinais emprega-

#	Algoritmo	Tipo	SDR	SIR	SAR
01	<i>Mask1D</i>	-	31,67	39,62	32,47
02	MaskCorr	-	37,58	44,31	38,91
03	ModMin	<i>HARD</i>	31,68	39,42	32,54
04	ModMin	<i>SOFT</i>	37,45	43,34	38,90
05	DEFL _F	-	31,64	39,67	32,43
06	CorrMin	<i>HARD</i>	32,00	39,70	32,84
07	CorrMin	<i>SOFT</i>	37,76	44,29	39,12

Tabela 5.9: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Seq. de Notas - 3 Fontes.

#	Algoritmo	Tipo	SDR	SIR	SAR
01	<i>Mask1D</i>	-	22,99	29,57	24,61
02	MaskCorr	-	24,87	30,56	26,60
03	ModMin	<i>HARD</i>	21,94	28,51	23,68
04	ModMin	<i>SOFT</i>	23,13	26,82	28,73
05	DEFL _F	-	23,08	29,65	24,70
06	CorrMin	<i>HARD</i>	23,77	29,97	25,37
07	CorrMin	<i>SOFT</i>	26,82	31,59	28,93

Tabela 5.10: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Instantâneas - Seq. de Notas - 4 Fontes.

dos. O ganho apresentado por esse algoritmo quando comparado ao mascaramento binário unidimensional foi, independentemente do número de fontes e do sinal empregado, da ordem de 6 dB, salvo para sinais de voz (japonês) com 3 fontes, quando o ganho foi de 4,6 dB

Vale a pena ressaltar que o algoritmo MaskCorr teve um excelente desempenho na SDR, pois, salvo o resultado obtido pelo algoritmo CorrMin (*HARD*) para o cenário com sinais de áudio e 4 fontes, obteve sempre resultados superiores aos algoritmos que supõem somente uma única fonte ativa a cada representação, ou seja, os algoritmos *Mask1D*, ModMin (*HARD*) e CorrMin (*HARD*). Este resultado é importante, visto que a correção no mascaramento binário (*HARD*) feita por esse algoritmo tinha como objetivo reduzir o erro do mascaramento *HARD*, uma vez que a hipótese admitia que as misturas reconstruídas pelas fontes estimadas via mascaramento *HARD* apresentavam discrepâncias.

Por fim, o algoritmo DEFL_F nesses cenários apresentou resultados similares ao mascaramento binário unidimensional, com uma diferença menor que 1 dB, não atendendo às expectativas. No entanto, o sistema determinado resultante da deflação

sempre foi resolvido com um novo mascaramento binário, e não através de outro algoritmo. Tal abordagem foi feita para verificarmos se o fato de se reutilizarem as representações já mascaradas para uma fonte em outra fonte (característica inerente do método) afetava de forma positiva ou não quando comparado ao mascaramento binário. Portanto, o método não apresentou ganhos, mas novas investigações empregando outros métodos de reconstrução no sistema deflacionado são necessárias para avaliarmos possíveis ganhos em relação ao mascaramento.

5.3.3 Resultados com Misturas Anecoicas - Caso Subdeterminado

Os resultados para o caso subdeterminado ($n = 3$ ou $n = 4$, e $m = 2$) foram obtidos a partir de 30 matrizes sorteadas aleatoriamente conforme descrito no Apêndice B, sendo apresentados nas Tabelas 5.11 a 5.18.

#	Algoritmo	Sol. Inicial	SDR	SIR	SAR
01	<i>Mask2D</i>	-	8,72	18,54	9,47
02	MaskCorr	01	10,41	17,99	11,64
03	DEFL _F	-	8,73	18,65	9,46
04	TSOE _{TF}	01	8,72	19,43	9,32
05	TSAE _{T1}	01	9,12	19,87	9,68
06	TSAE _{T1}	02	9,74	19,94	10,38
07	TSAE _{T1}	03	9,12	19,89	9,67
08	TSAE _{T2}	02	9,82	19,35	10,57

Tabela 5.11: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Voz em Inglês - 3 Fontes.

#	Algoritmo	Sol. Inicial	SDR	SIR	SAR
01	<i>Mask2D</i>	-	7,13	16,49	7,91
02	MaskCorr	01	7,79	15,77	8,79
03	DEFL _F	-	7,16	16,64	7,94
04	TSOE _{TF}	01	7,04	16,76	7,76
05	TSAE _{T1}	01	7,48	17,58	8,12
06	TSAE _{T1}	02	7,67	17,58	8,34
07	TSAE _{T1}	03	7,23	17,34	8,06
08	TSAE _{T2}	02	7,90	16,95	8,67

Tabela 5.12: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Voz em Inglês - 4 Fontes.

#	Algoritmo	Sol. Inicial	SDR	SIR	SAR
01	<i>Mask2D</i>	-	10,35	20,99	10,92
02	MaskCorr	01	12,03	20,47	12,90
03	DEFL _F	-	10,35	21,08	10,91
04	TSOE _{TF}	01	10,41	21,59	10,90
05	TSAE _{T1}	01	10,76	22,07	11,21
06	TSAE _{T1}	02	11,44	22,32	11,92
07	TSAE _{T1}	03	10,76	22,11	11,21
08	TSAE _{T2}	02	11,48	22,05	12,01

Tabela 5.13: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Voz em Japonês - 3 Fontes.

#	Algoritmo	Sol. Inicial	SDR	SIR	SAR
01	<i>Mask2D</i>	-	6,67	15,83	7,55
02	MaskCorr	01	7,37	14,92	8,49
03	DEFL _F	-	6,74	16,03	7,58
04	TSOE _{TF}	01	6,69	16,48	7,41
05	TSAE _{T1}	01	7,06	16,82	7,78
06	TSAE _{T1}	02	7,28	16,81	8,04
07	TSAE _{T1}	03	7,06	16,86	7,77
08	TSAE _{T2}	02	7,49	16,33	8,32

Tabela 5.14: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Voz em Japonês - 4 Fontes.

#	Algoritmo	Sol. Inicial	SDR	SIR	SAR
01	<i>Mask2D</i>	-	27,85	33,10	29,58
02	MaskCorr	01	28,64	33,21	30,70
03	DEFL _F	-	28,06	33,38	29,73
04	TSOE _{TF}	01	28,14	33,35	29,88
05	TSAE _{T1}	01	28,63	33,80	30,40
06	TSAE _{T1}	02	29,34	34,61	31,07
07	TSAE _{T1}	03	28,75	33,97	30,48
08	TSAE _{T2}	02	29,14	34,30	31,06

Tabela 5.15: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Áudio - 3 Fontes.

Os ganhos obtidos pelos algoritmos propostos para o caso anecoico não foram significativos. A SIR apresentou ganhos inferiores a 1,5 dB, a SDR inferiores a 2,5

#	Algoritmo	Sol. Inicial	SDR	SIR	SAR
01	<i>Mask2D</i>	-	10,30	14,12	13,34
02	MaskCorr	01	9,95	12,99	13,68
03	DEFL _F	-	10,53	14,57	13,44
04	TSOE _{TF}	01	10,25	14,39	13,19
05	TSAE _{T1}	01	11,02	14,98	13,96
06	TSAE _{T1}	02	11,14	14,97	14,17
07	TSAE _{T1}	03	10,89	14,83	13,94
08	TSAE _{T2}	02	10,92	14,44	14,20

Tabela 5.16: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Áudio - 4 Fontes.

#	Algoritmo	Sol. Inicial	SDR	SIR	SAR
01	<i>Mask2D</i>	-	25,74	35,08	26,99
02	MaskCorr	01	27,76	33,76	29,84
03	DEFL _F	-	25,76	35,16	27,01
04	TSOE _{TF}	01	23,57	31,35	26,68
05	TSAE _{T1}	01	22,49	27,29	26,92
06	TSAE _{T1}	02	26,47	34,64	28,12
07	TSAE _{T1}	03	22,29	27,10	26,78
08	TSAE _{T2}	02	26,48	34,63	28,11

Tabela 5.17: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Seq. de Notas - 3 Fontes.

#	Algoritmo	Sol. Inicial	SDR	SIR	SAR
01	<i>Mask2D</i>	-	18,98	25,28	22,37
02	MaskCorr	01	19,29	23,64	23,37
03	DEFL _F	-	19,03	25,37	22,40
04	TSOE _{TF}	01	18,87	24,83	22,35
05	TSAE _{T1}	01	17,03	21,41	22,51
06	TSAE _{T1}	02	19,23	25,08	22,89
07	TSAE _{T1}	03	16,99	21,38	22,52
08	TSAE _{T2}	02	19,14	24,36	23,09

Tabela 5.18: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Misturas Anecoicas - Seq. de Notas - 4 Fontes.

dB e a SAR inferiores a 3,0 dB.

Apesar desse resultado insatisfatório, é possível observar que o algoritmo DEF_L_F

sempre apresentou ganho de desempenho na SDR e SIR quando comparado ao mascaramento binário, perdendo na SAR em somente dois cenários (voz em inglês e japonês com 3 fontes) por apenas 0,01 dB.

Da mesma forma, os algoritmos 06 e 08 sempre apresentaram desempenhos melhores em relação à SDR e à SAR que o mascaramento binário, independentemente do cenário testado.

Por fim, foi também observado que o cenário das misturas anecoicas com sequência de notas foi mais difícil para os algoritmos propostos do que os demais cenários contendo outros tipos de sinais.

5.3.4 Resultados com Misturas Anecoicas - Caso Determinado

Esse experimento pretende comparar o resultado da solução exata com os da aproximada e do mascaramento binário para misturas anecoicas no caso de duas fontes e duas misturas. As matrizes de atenuação e atraso usadas para a reconstrução foram as reais. Portanto, atua-se num contexto ideal de forma proposital, em que não há erro de estimação das matrizes de mistura, para que o enfoque seja a reconstrução nesse cenário.

Os resultados para o caso determinado foram obtidos a partir de 50 matrizes de misturas sorteadas aleatoriamente. O mascaramento corrigido foi finalizado com 12 iterações, de acordo com o resultado obtido na Seção 5.3.1.

Os algoritmos implementados, numerados de 01 a 05, estão descritos nas Tabelas 5.19 a 5.21. Na implementação do algoritmo 02, usou-se na síntese a mesma janela *Hamming* com 1024 amostras, com salto de 256 amostras entre janelas consecutivas.

Nas Tabelas 5.19 a 5.22 apresentamos os resultados obtidos usando sinais de voz em inglês e japonês, de áudio e sequência de notas de instrumentos musicais, respectivamente.

#	Algoritmo	SDR	SIR	SAR
01	Exata na Freq.	62,70	72,28	68,50
02	Exata em T-F	55,31	70,06	57,65
03	<i>Mask1D</i>	13,85	23,72	14,41
04	<i>Mask2D</i>	17,58	27,21	18,29
05	Aproximada	53,31	55,41	60,67
06	MaskCorr	36,61	41,87	54,96
07	DEFL _F	17,58	27,21	18,29

Tabela 5.19: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Voz em Inglês.

#	Algoritmo	SDR	SIR	SAR
01	Exata na Freq.	66,63	72,17	69,31
02	Exata em T-F	55,51	70,54	57,47
03	<i>Mask1D</i>	13,57	24,51	14,01
04	<i>Mask2D</i>	17,28	27,41	17,85
05	Aproximada	53,93	57,13	61,00
06	MaskCorr	38,19	40,06	61,72
07	DEFL _F	17,28	27,41	17,85

Tabela 5.20: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Voz em Japonês.

#	Algoritmo	SDR	SIR	SAR
01	Exata na Freq.	38,97	49,25	40,41
02	Exata em T-F	42,13	49,93	45,05
03	<i>Mask1D</i>	27,92	34,87	28,96
04	<i>Mask2D</i>	30,86	39,36	31,69
05	Aproximada	35,69	43,41	37,97
06	MaskCorr	41,82	52,98	42,69
07	DEFL _F	30,79	39,35	31,62

Tabela 5.21: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Áudio.

#	Algoritmo	SDR	SIR	SAR
01	Exata na Freq.	50,25	59,55	57,31
02	Exata em T-F	49,21	59,39	54,09
03	<i>Mask1D</i>	22,06	29,05	23,35
04	<i>Mask2D</i>	27,52	37,03	28,75
05	Aproximada	48,53	50,39	58,85
06	MaskCorr	51,93	59,46	61,11
07	DEFL _F	27,52	37,02	28,75

Tabela 5.22: SDR, SIR e SAR (em dB) dos Algoritmos de Reconstrução para Sequência de Notas.

Pode-se observar das Tabelas 5.19 e 5.20 que para sinais de voz os algoritmos apresentaram a seguinte ordem decrescente de desempenho em relação à SDR: 01, 02, 05, 06, 04/07, 03. No entanto, para sinais de áudio os algoritmos resultaram em uma ordem decrescente de desempenho diferente da obtida com sinais de voz: 02,

06, 01, 05, 04/07, 03. Para sequência de notas a ordem obtida foi: 06, 01, 02, 05, 04/07, 03.

Para esse cenário em particular (duas fontes e dois sensores), o mascaramento corrigido e a solução aproximada sempre obtiveram resultados superiores ao mascaramento simples, independentemente do sinal empregado. A solução empregando a deflação em frequência obteve resultados similares ao mascaramento binário.

Capítulo 6

Avaliação de Algoritmos de Separação de Fontes Aplicados no Reconhecimento Automático de Instrumentos Musicais em Misturas Polifônicas

O propósito deste capítulo é descrever e avaliar o emprego das técnicas de separação cega propostas em um sistema de reconhecimento automático de instrumentos musicais quando aplicadas a misturas polifônicas. Neste contexto, a taxa de acerto do sistema serve como indicador da qualidade da separação, uma vez que a taxa de acerto do instrumento que compõe uma dada sequência monofônica para todas as famílias de instrumentos testadas é alta¹, e superior à taxa de acerto de uma nota específica de um instrumento perante o classificador. Assim, espera-se que uma boa separação preserve as características utilizadas pelos classificadores da maioria das notas associadas ao instrumento que compõem a sequência monofônica, mantendo alta a taxa de acerto do sistema de reconhecimento automático de instrumentos musicais (SRAIM), ao passo que uma separação ruim possivelmente alterará as características das maiorias das notas que compõem a sequência monofônica, reduzindo de forma significativa a taxa de acerto do SRAIM.

Este capítulo é composto por três seções. A primeira o banco de classificadores empregado pelo SRAIM ; a segunda descreve o sistema de reconhecimento automático empregado; e finalmente, a última descreve o experimento realizado, incluindo os sinais utilizados e os algoritmos empregados e os resultados obtidos.

¹100% caso a sequência seja avaliada diretamente sem passar pelo algoritmo de separação.

6.1 Classificador de Instrumentos Musicais

O classificador [18] empregado recorre a um banco de classificadores, conforme mostra a Fig. 6.1. O desempenho do banco de classificadores é superior ao de cada classificador utilizado de forma isolada. O classificador foi projetado para o reconhecimento de notas isoladas a partir de segmentos das mesmas.

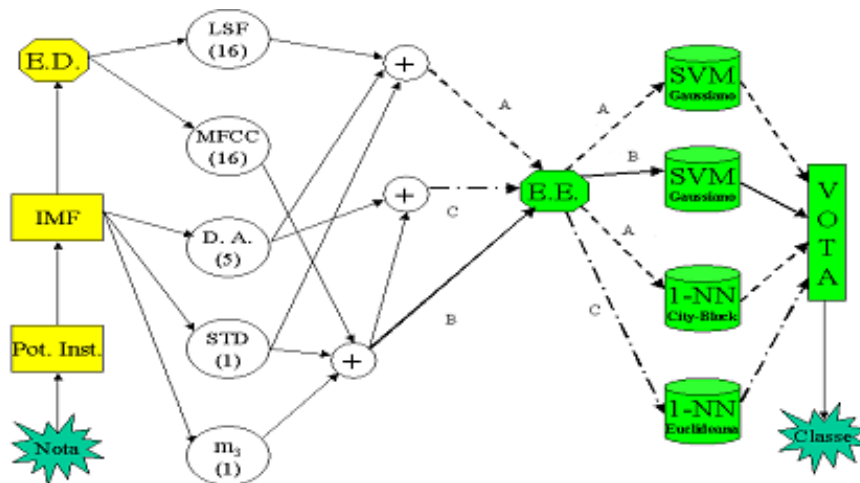


Figura 6.1: Classificador Empregado no SRAIM.

Conforme pode ser visto na Fig. 6.1, os três primeiros blocos (Pot.Inst., IMF, E.D.) correspondem à etapa de pré-processamento aplicado ao sinal da nota de um instrumento musical. O bloco Pot.Inst. obtém a potência instantânea do sinal; já o bloco IMF é um segmentador, que retira o segmento intermediário da potência instantânea do sinal através de um limiar definido como 90% da média da potência instantânea da nota musical; por fim, o segmento extraído passa por um processo de escalamento dinâmico representado pelo bloco E.D., visando a normalização desse segmento. A partir desse momento, são extraídos os vetores de características da nota, que são formados, basicamente, por dois tipos de codificadores: coeficientes LSF (do inglês *Line Spectral Frequencies*) [103] e MFCC (do inglês *Mel Frequency Cepstral Coefficients*) [104, 105]. Ambos os codificadores utilizam 16 coeficientes representativos. Além dos codificadores, o vetor de características também tem como elementos o desvio padrão (STD), o *skewness* (m_3) [18] e mais cinco descritores de áudio: a taxa de cruzamento por zero, o fluxo espectral, o valor RMS do *frame*, o centroide espectral e a largura do centroide espectral. Essas medidas são combinadas de três diferentes formas, resultando em vetores de características distintos (A, B e C), passando, posteriormente, por um processo de escalamento estatístico (E.E.) e servindo de entrada para 4 classificadores. Foram utilizados como classificadores dois SVM's (do inglês *Support Vector Machine*) [106–108] e dois K -NN (do inglês *K-Nearest Neighbor*, com $K = 1$) [109]. Ao final, a classe identificada é obtida como a

classe mais votada entre os quatro classificadores (caso haja empate, há um sorteio). O classificador foi treinado com um subconjunto de notas oriundas de três bases de dados: *Real World Computing*. (RWC) [110], *Musical Instruments Samples*. (MIS) [111] e *McGill University Master Samples*. (MUMS) [112].

O banco de classificadores adotado é capaz de classificar 20 instrumentos distintos. O conjunto usado na fase de treinamento (90% do total de notas das bases de dados MUMS, RWC e MIS) não possui notas em comum com o conjunto de notas usado para formar as sequências monofônicas de teste (10% restante do total das notas da base de dados RWC). As classes de instrumentos musicais que o classificador desenvolvido está preparado para reconhecer se encontra na Tabela 6.1.

Classe	#	Instrumento
Metais	1	Trompa
	2	Trombone
	3	Trompete
	4	Tuba
Flautas	5	Flauta Contralto
	6	Flauta Baixo
	7	Flauta
Palhetas	8	Fagote
	9	Saxofone Contralto
	10	Saxofone Tenor
	11	Saxofone Soprano
	12	Saxofone Barítono
	13	Clarinete Bb
	14	Clarinete Eb
	15	Oboé
Percussão	16	Glockenspiel
	17	Xilofone
	18	Vibrafone
Cordas	19	Violoncelo
	20	Violino

Tabela 6.1: Classes de Instrumentos Musicais do Classificador.

6.2 Sistema de Reconhecimento Automático de Instrumentos Musicais

O SRAIM foi elaborado originalmente em [18] e testado, naquela ocasião, com um algoritmo de separação de fontes não-cego (ou seja, supervisionado), por meio de um oráculo (que possui algum conhecimento prévio da fonte dominante no domínio T-F). Posteriormente, o SRAIM foi avaliado com alguns algoritmos de separação cega em [113, 114].

A arquitetura do SRAIM foi construída a partir de três grandes blocos: um bloco separador, um detector/extrator de notas e um terceiro bloco codificador/classificador de notas, conforme ilustrado na Fig. 6.2

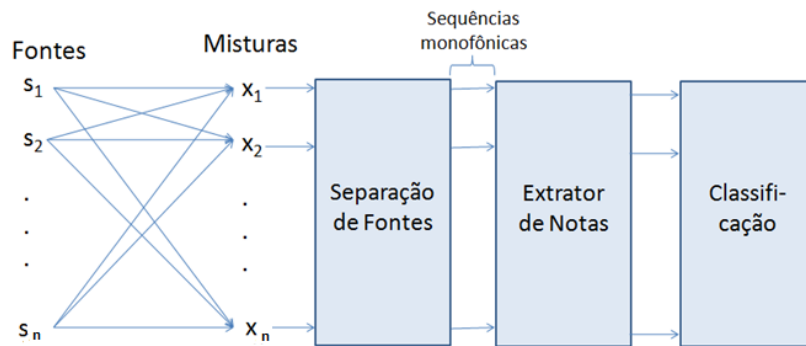


Figura 6.2: SRAIM Proposto.

O sistema proposto foi exercitado com duas fontes ($n = 2$) e dois sensores ($m = 2$) considerando as misturas: instantâneas e anecoicas.

As sequências monofônicas foram construídas a partir dos sinais das notas musicais de um conjunto de teste, obtidos da região central da tessitura de cada instrumento musical. Essas sequências foram agrupadas com intervalos (*gaps*) entre elas com duração variando entre 0,045 ms e 0,3 ms. Após a formação das sequências monofônicas, estas foram combinadas através de matrizes de mistura (2×2) para gerar os sinais polifônicos.

Foram geradas dez matrizes de misturas, sendo cada mistura formada por pares de sequências monofônicas pertencentes à mesma família de instrumentos (Palhetas, Cordas, Metais e Percussão).

Para geração das sequências monofônicas utilizaram-se os seguintes instrumentos: Trompa e Trombone, da família Metais; Glockenspiel e Vibrafone, da família Percussão; Saxofone Soprano e Saxofone Tenor, da família Palhetas; Violoncelo e Violino, da família Cordas.

Na etapa 1, além dos algoritmos de separação cega propostos nesse trabalho,

foram usados, como referências para misturas instantâneas, o algoritmo JADE, e para as misturas anecoicas, o algoritmo TIFROM_Anec.

Na etapa 2, foi empregado um algoritmo extrator de notas que utiliza a média e o desvio padrão [18] das amostras de uma janela deslizante aplicada a cada estimativa da fonte no domínio do tempo, de tamanho fixo, para determinar o início e o fim de cada nota.

Na etapa 3, após a separação das fontes, cada sequência monofônica é formada por um conjunto de notas que, em tese, pertencem a mesma fonte. Assim, empregou-se um critério de contabilização do instrumento classificado pelo banco de classificadores para cada nota extraída da sequência monofônica, visando determinar o instrumento presente nesta sequência.

A classificação da nota efetuada pelo banco de classificadores é feita também através de uma contabilização por voto majoritário entre os quatro classificadores. Portanto, a classe estimada do instrumento musical deve receber ao menos dois votos vindos de classificadores distintos. Caso ocorra um empate na votação entre duas ou mais classes, normalmente é feito um sorteio entre essas classes. Contudo, na implementação adotada escolheu-se um critério de desempate diferente, ou seja, entre as classes empatadas optou-se pela classe que recebeu mais votos entre as demais notas da sequência monofônica.

Como experimento de controle para avaliar o desempenho do sistema independentemente do algoritmo de separação, foram introduzidas as sequências monofônicas originais, antes de passar pelo misturador, e avaliou-se o desempenho do restante do sistema para cada uma das oito sequências monofônicas originais.

A Fig. 6.3 mostra um histograma de classificação típico de uma sequência original do Saxofone Soprano, que teve o maior número de ocorrências de falsas estimativas entre os instrumentos considerados. Conforme se pode observar, o histograma apresentou um número maior de ocorrências para a classe certa, entretanto, apresentou estimativas erradas, com uma ocorrência apontando para as classes 5, 8, 9 e 20, e duas para a classe 10. Na Tabela 6.2 apresentamos as taxas de acerto obtidas pelo sistema nas sequências monofônicas empregadas para formar as misturas.

Cabe ressaltar que a taxa de acerto apresentada não é do classificador isoladamente mas do SRAIM completo, empregando um subconjunto de notas bem menor que o total de notas empregadas para avaliar o Classificador isoladamente em [18]. Além disso, os classificadores foram treinados e testados com as notas já separadas, ao passo que na Tabela 6.2 existem as perdas decorrentes do módulo extrator.

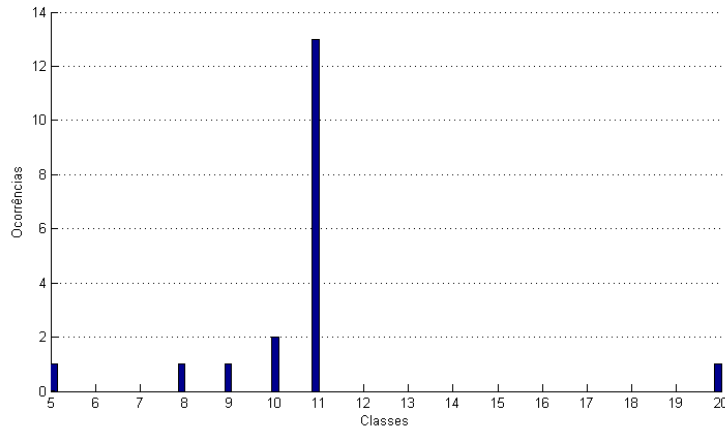


Figura 6.3: Histograma de Ocorrências do Saxofone Soprano.

Instrumento	Taxa de Acerto
Trompa	92,3%
Trombone	100,0%
Saxofone Tenor	100,0%
Saxofone Soprano	68,0%
Glockenspiel	100,0%
Vibrafone	100,0%
Violoncelo	98,6%
Violino	91,9%

Tabela 6.2: Taxa de Acerto do SRAIM.

6.3 Resultados do Experimento

Nesta seção apresentaremos os resultados do experimento em duas etapas. Na primeira será apresentado o resultado do bloco separador, e na segunda serão apresentadas as taxas de acertos do banco de classificadores para cada nota extraída e do instrumento estimado na sequência monofônica.

Foram sorteadas aleatoriamente 10 matrizes de misturas com coeficientes de atenuação positivos e limitados ao intervalo $[0, 2, 5, 0]$. Os mesmos coeficientes foram usados nas misturas instantâneas e anecoicas.

Os coeficientes de atraso foram sorteados aleatoriamente (distribuição uniforme) dentro do intervalo $[-4, 0, 4, 0]$.

O número de notas presentes em cada sequência monofônica variou em função do instrumento. Isso ocorreu porque se fixou um percentual do total de notas do conjunto de testes de cada instrumento para compor a sequência monofônica. Consequentemente o número de notas na sequência variou, uma vez que cada instrumento musical na base de dados RWC continha uma quantidade distinta de notas.

Todos os sinais foram gerados com uma taxa de amostragem de 44,1 kHz. A duração dos sinais dos Metais foi de 37.8 s, das Palhetas 41.1 s e da Percussão de 33.1 s.

6.3.1 Misturas Instantâneas

Nesta seção foram empregados os algoritmos de separação cega que trabalham no domínio temporal, uma vez que as sequências de notas são construídas com presença de uma esparsidade temporal. Portanto, técnicas que exploram esparsidade nesse domínio levam vantagem em relação as que exploram a esparsidade em outro domínio. Assim, os algoritmos propostos exercitados nesse experimento foram o SCAM-t, o SCAc-t, o SCAM-t (moda) e o SCAc-t (moda), além do algoritmo clássico JADE, usado como referência. Os sinais foram reconstruídos a partir da inversão da matriz de mistura estimada.

Desempenho da Separação

Nas Figs. 6.4 e 6.5, apresentamos os valores médios da SDR e da SIR, respectivamente, para a família de Metais, obtidos para cada par de fontes estimadas, com intuito de avaliarmos os desempenhos dos algoritmos de separação para as misturas formadas pelas sequências monofônicas. A partir de testes e resultados, podemos avaliar a correspondência entre os desempenhos médios obtidos pelos algoritmos na etapa de separação e na etapa de classificação.

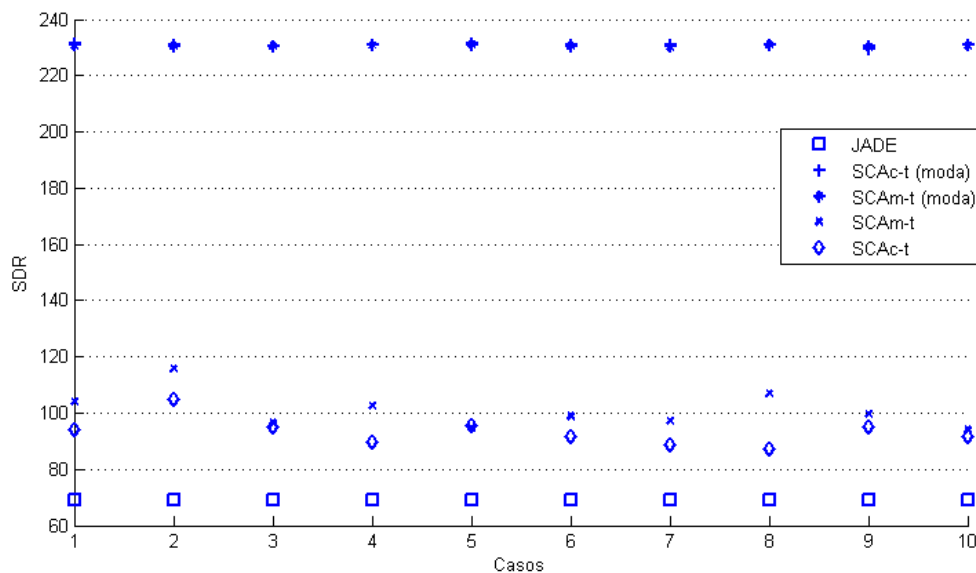


Figura 6.4: Resultados da SDR com Metais.

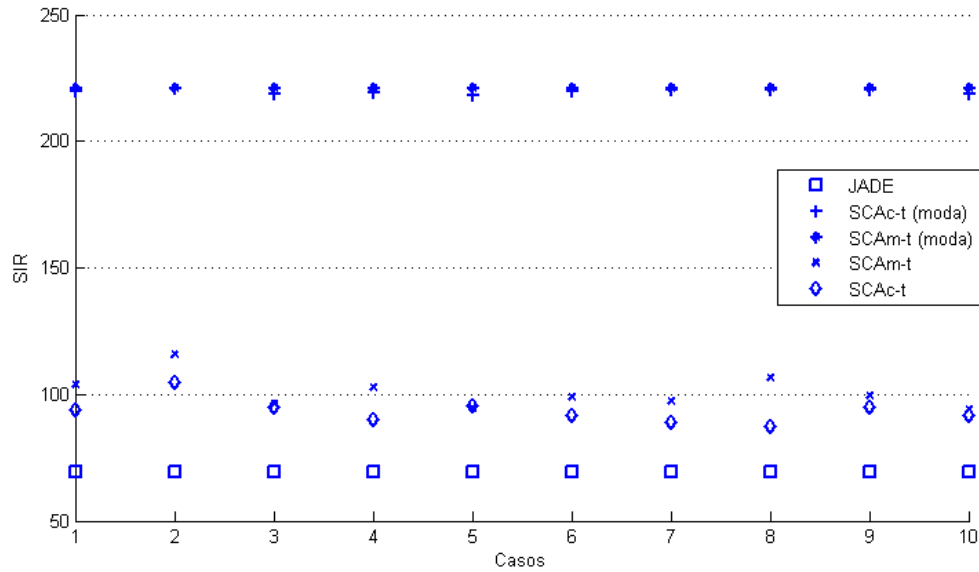


Figura 6.5: Resultados da SIR com Metais.

Todos os algoritmos apresentaram excelentes resultados em relação à SAR, sendo o menor valor obtido igual a 216,89 dB e o maior igual a 220,77 dB. As soluções obtidas pelos métodos SCAM-t (moda) e SCAC-t (moda) foram semelhantes e apresentaram os melhores valores para todas as medidas, seguidas pelas soluções SCAC-t e SCAM-t, e pela solução do algoritmo JADE.

Assim, face ao excelente desempenho obtido por todos os algoritmos exercitados nas misturas instantâneas em relação a todas as medidas (SDR, SIR e SAR), espera-se que o classificador para misturas instantâneas apresente um bom desempenho, independentemente do algoritmo empregado na separação cega das sequências monofônicas.

Desempenho do Classificador

Nas Figs. 6.6 a 6.9 apresentamos o desempenho do classificador para cada algoritmo de separação cega para misturas instantâneas, onde cada caso representa uma matriz de mistura instantânea diferente. O limiar de 50% na taxa de acerto encontra-se indicado com uma linha tracejada em vermelho, para visualizarmos em quais casos os instrumentos foram identificados corretamente. Algoritmos com desempenhos superiores a 50% tiveram a maioria das notas da sequência monofônica classificadas corretamente, indicando que o instrumento foi corretamente identificado pelo SRAIM. Eventualmente um percentual inferior a 50% pode representar uma identificação correta, bastando que esse percentual corresponda à maior votação no instrumento correto.

Na Fig. 6.6, contendo instrumentos da família Metais, pode-se observar que

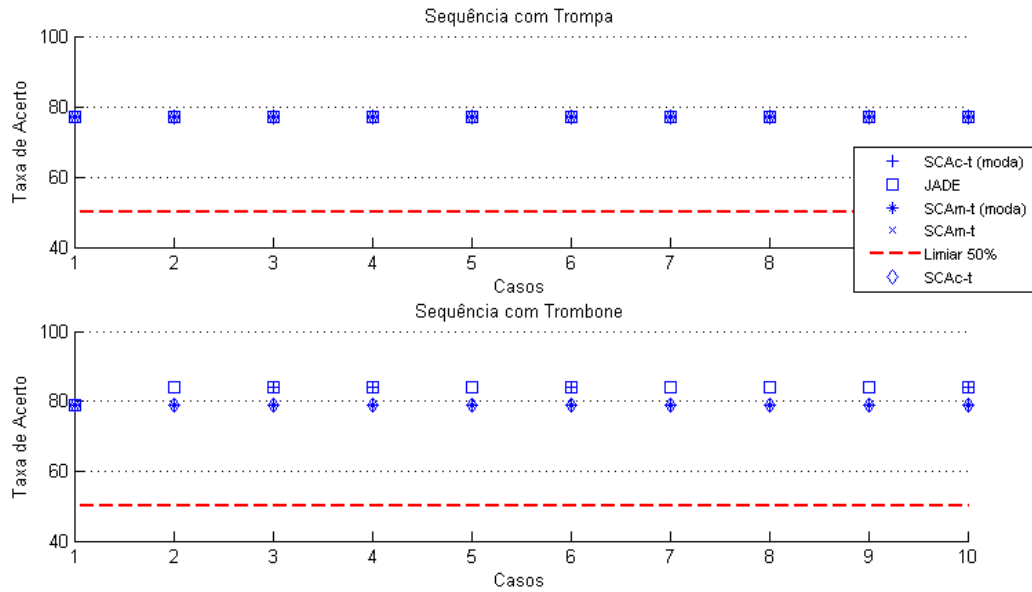


Figura 6.6: Taxa de Acerto para Misturas Monofônicas - Metais.

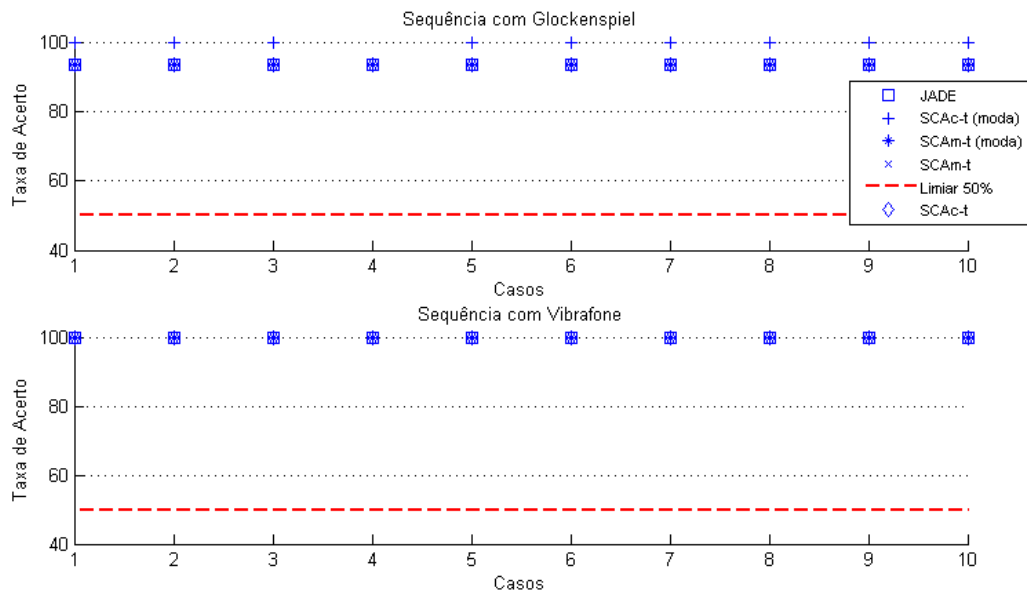


Figura 6.7: Taxa de Acerto para Misturas Monofônicas - Percussão.

o algoritmo JADE apresentou um desempenho ligeiramente superior, seguido do algoritmo SCAc-t (moda), e que todos os algoritmos exercitados resultaram na correta identificação dos instrumentos que compunham as misturas polifônicas, ou seja, houve 100% de acerto. Pode-se verificar que o desempenho obtido no instrumento Trombone foi ligeiramente superior ao obtido no instrumento Trompa, conforme se esperava a partir do experimento de controle, com resultados sintetizados na Tabela 6.2.

Na Fig. 6.7, contendo instrumentos da família Percussão, pode-se observar que

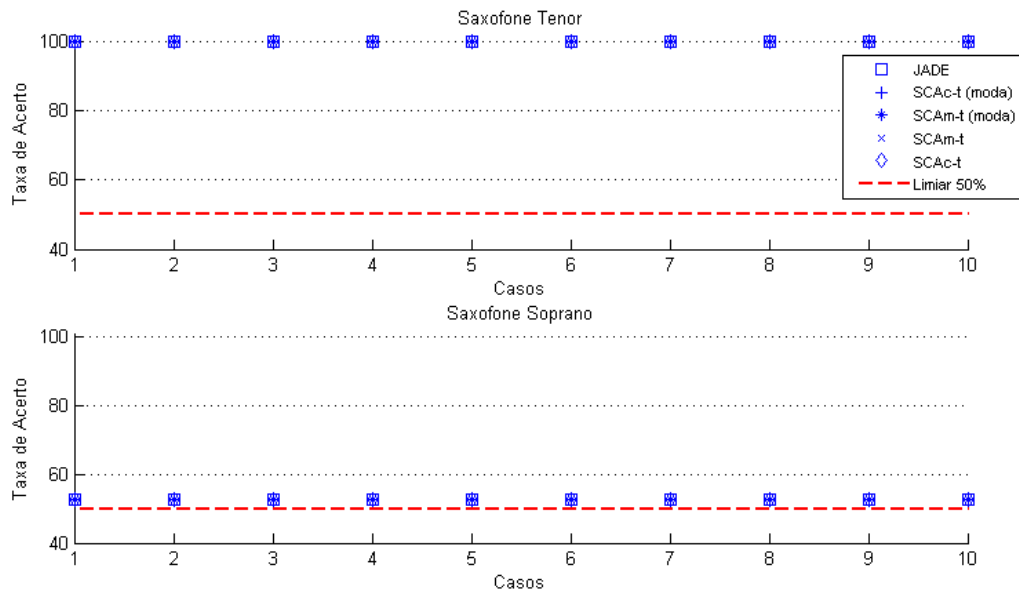


Figura 6.8: Taxa de Acerto para Misturas Monofônicas - Palhetas.

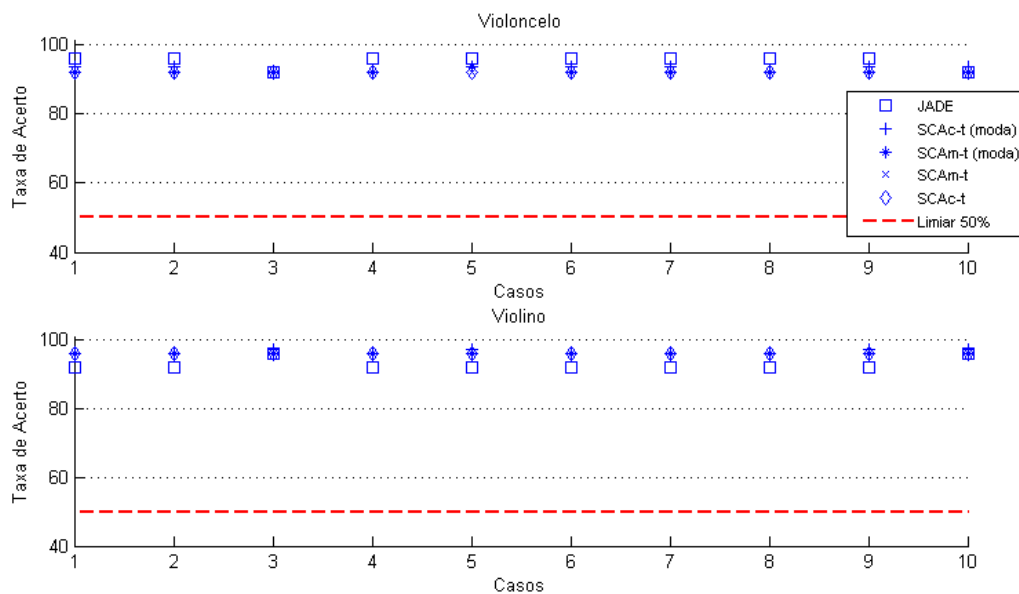


Figura 6.9: Taxa de Acerto para Misturas Monofônicas - Cordas.

todos os algoritmos apresentaram o mesmo desempenho, sendo que o desempenho obtido para os instrumentos Vibrafone e Glockenspiel seguiu a mesma tendência (mantendo 100% para o Vibrafone) apresentada pelo experimento de controle com resultados apresentados na Tabela 6.2.

Na Fig. 6.8, contendo instrumentos da família Palhetas, pode-se observar que todos os algoritmos apresentaram desempenhos semelhantes, mantendo-se a tendência apresentada no experimento de controle, ou seja, um desempenho melhor para o instrumento Saxofone Tenor em comparação ao obtido pelo instrumento Saxofone

Soprano. A taxa de acerto de 100% obtida para o instrumento Saxofone Tenor apresentada no experimento de controle foi mantida, e o desempenho obtido com o instrumento Saxofone Soprano apresentou uma piora.

Por fim, na Fig. 6.9, contendo instrumentos da família Cordas, pode-se observar que o algoritmo o JADE obteve um desempenho ligeiramente melhor para o instrumento Violoncelo do que os demais algoritmos e ligeiramente pior que os demais algoritmos para o instrumento Violino. O desempenho obtido pelo algoritmo JADE manteve a tendência obtida no experimento de controle para os instrumentos dessa família, ou seja, apresentando uma ligeira piora para o instrumento Violoncelo.

6.3.2 Misturas Anecoicas

Nesta seção, diferentemente da anterior, foram empregados algoritmos de separação cega que trabalham no domínio tempo-frequência, uma vez que todas as técnicas elaboradas nesta tese para resolver o problema de separação cega no caso anecoico atuam nesse domínio.

Optou-se por usar o algoritmo TIFROM_Anec como referência devido ao fato de que esse algoritmo obteve, em geral, desempenho superior à versão do DUET implementada para esta tese. Assim, os algoritmos propostos neste trabalho e exercitados nesse experimento foram o TIFROM_Anec com reconstrução via mascaramento binário, o STUE(TIFROM_Anec) com reconstrução via mascaramento binário e com reconstrução via solução aproximada ($STUE(TIFROM_Anec)_{ap}$), e o STUE(ATIFROM(DFTNU)) também com ambos os tipos de reconstrução.

Desempenho da Separação

Nas Figs. 6.10 a 6.12 apresentamos os valores médios da SDR, SIR e SAR, respectivamente, para a família Metais obtidos para cada par de fontes estimadas, para avaliarmos a correspondência entre o desempenho médio do algoritmo obtido na etapa de separação e o resultado da classificação.

Alguns algoritmos tiveram problemas nos Casos 1, 4 e 10. Nota-se que as soluções obtidas pelo algoritmo STUE(TIFROM_Anec) foram equivalentes às do TIFROM_Anec, ganhando em algumas ocasiões e perdendo em outras. Como já havíamos visto no Cap. 4, o algoritmo STUE apresenta em média uma solução com MSE menor do que a do algoritmo TIFROM_Anec. Entretanto, isso não é adequadamente aproveitado pelo mascaramento binário bidimensional (por ser um processo de clusterização). Portanto, aproveitaremos melhor o menor MSE obtido pelo algoritmo STUE ao empregarmos na reconstrução o algoritmo com a solução Aproximada.

Avaliando as Figs. 6.10 a 6.12 pode-se verificar que as reconstruções que empregaram a solução Aproximada obtiveram melhores desempenhos, principalmente em

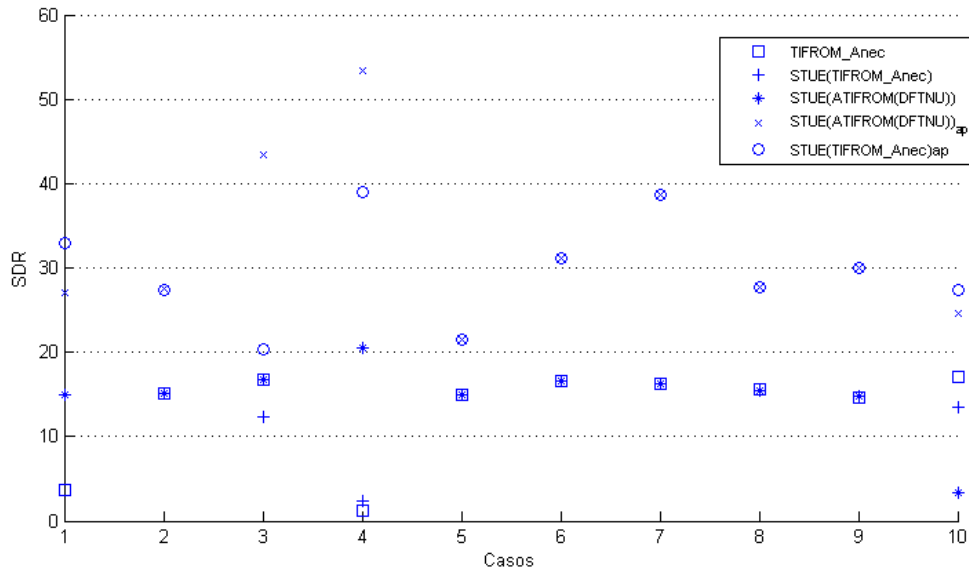


Figura 6.10: Resultados da SDR dos Algoritmos TIFROM_Anec e STUE para Metais.

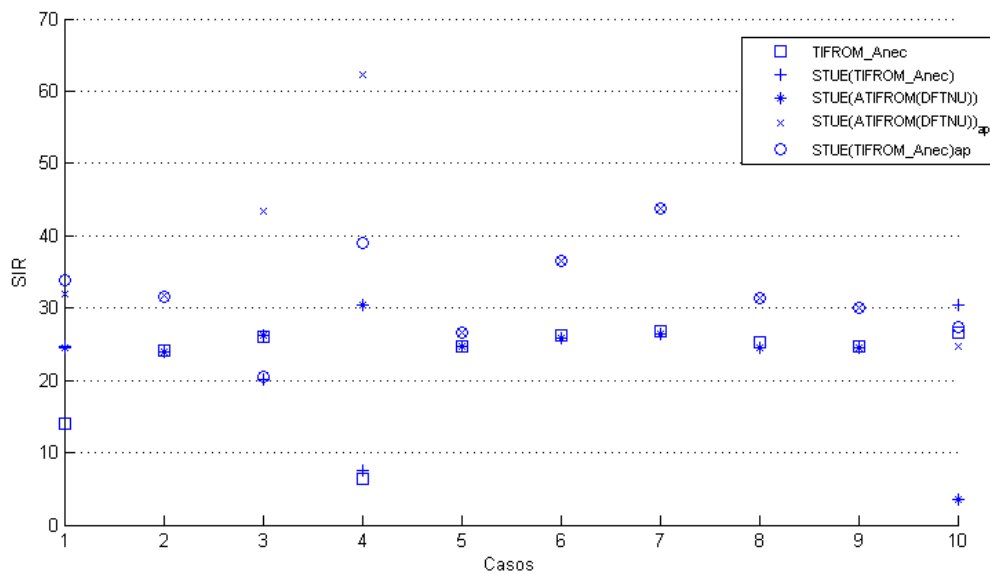


Figura 6.11: Resultados da SIR dos Algoritmos TIFROM_Anec e STUE para Metais.

relação à SAR.

Desempenho do Classificador

Nesta seção apresentaremos o desempenho do classificador para cada algoritmo de separação cega para misturas anecoicas. Nas figuras apresentadas a seguir, novamente indicaremos o limiar de 50% com uma linha tracejada em vermelho, para fácil visualização dos casos em que o instrumento pertencente à mistura polifônica

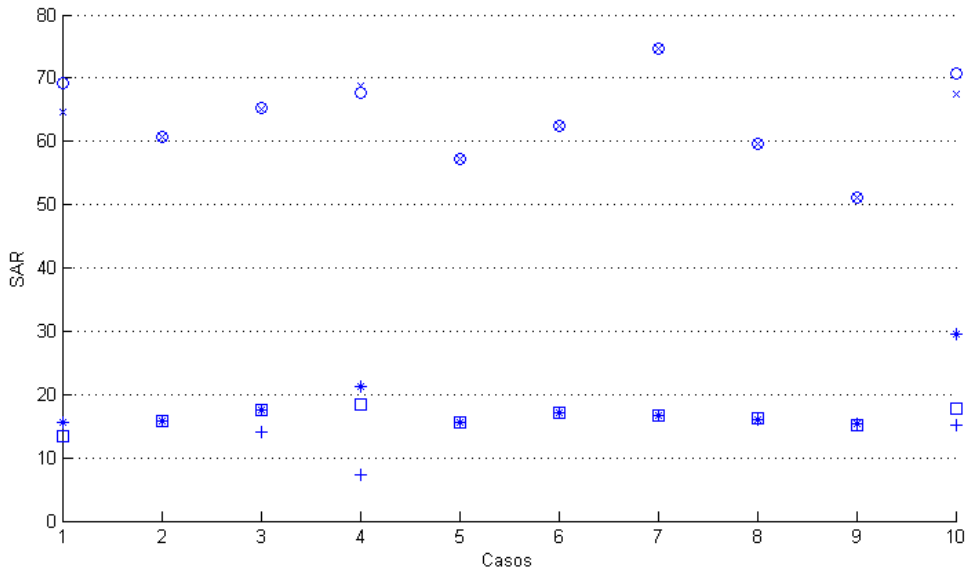


Figura 6.12: Resultados da SAR dos Algoritmos TIFROM_Anec e STUE para Metais.

foi corretamente identificado.

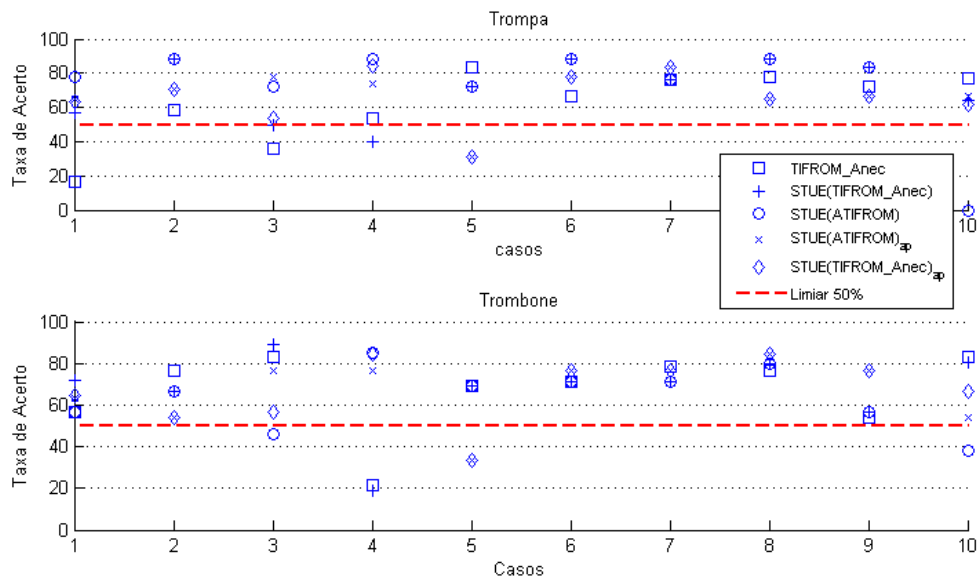


Figura 6.13: Taxa de Acerto para Misturas Monofônicas - Metais.

Diferentemente do experimento com misturas instantâneas, em alguns casos nem todos os algoritmos obtiveram desempenhos superiores a 50%. Na Tabela 6.3 apresentamos as porcentagens dos casos em que houve identificação correta de ambos os instrumentos contidos nas misturas para os diferentes algoritmos. Os casos em que o SRAIM identificou corretamente apenas um dos instrumentos foram contabilizados

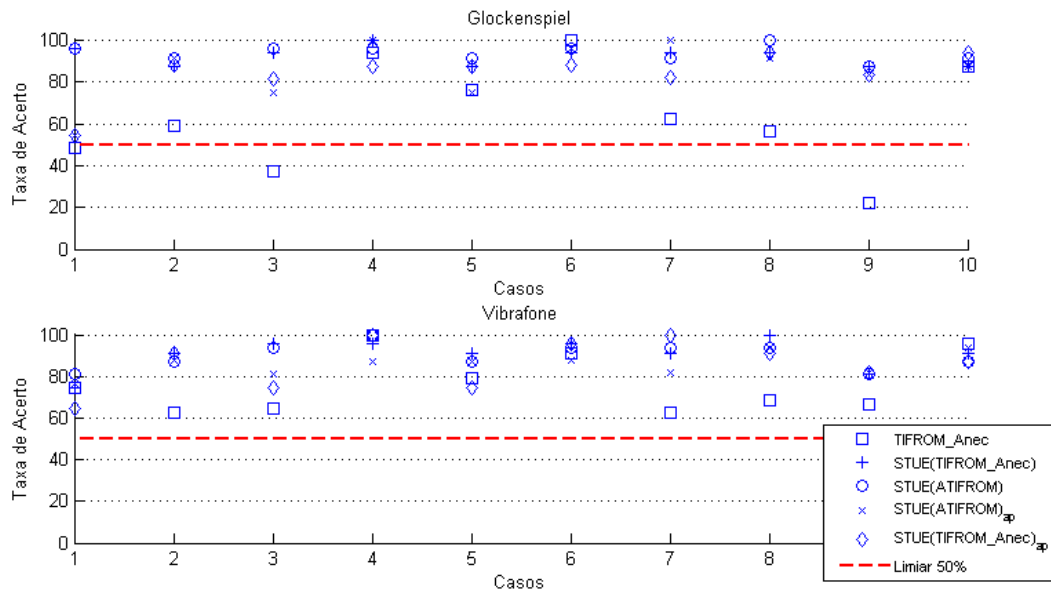


Figura 6.14: Taxa de Acerto para Misturas Monofônicas - Percussão.

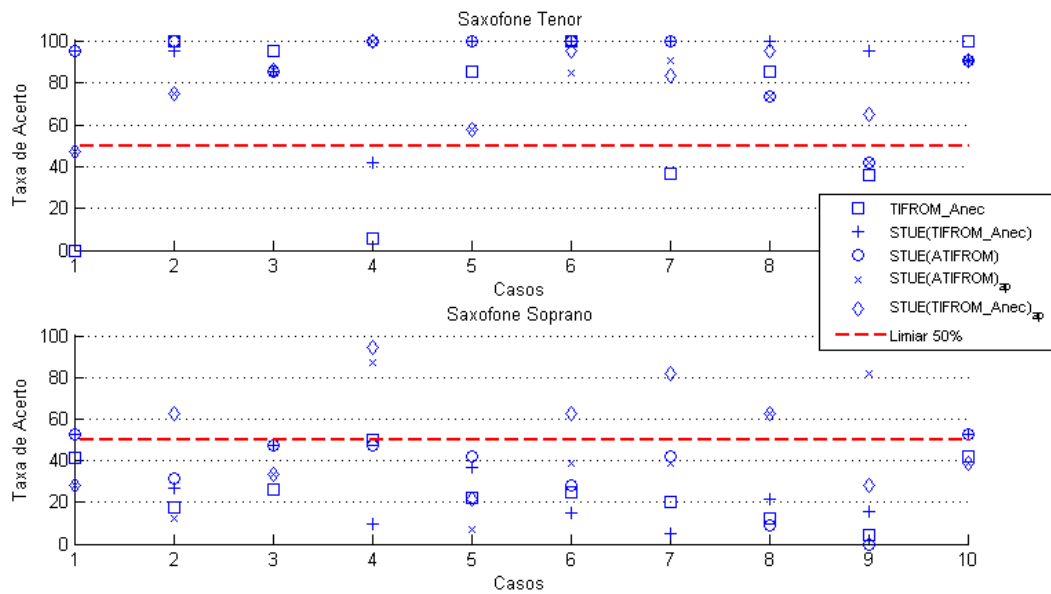


Figura 6.15: Taxa de Acerto para Misturas Monofônicas - Palhetas.

como classificação incorreta. Como as misturas foram formadas por instrumentos da mesma família, a Tabela 6.3 discrimina o percentual de acerto nas misturas indicadas pelas diferentes famílias.

O baixo desempenho obtido para as misturas formadas pelos instrumentos da classe Palhetas pode ser explicado observando na Tabela 6.2 (antes de se formarem as misturas) e na Fig. 6.8 (para misturas instantâneas) o baixo desempenho obtido na identificação do Saxofone Soprano, que foi o instrumento responsável pela baixa taxa de acerto obtida para essa classe. Essa piora ocorreu quando a sequência monofônica

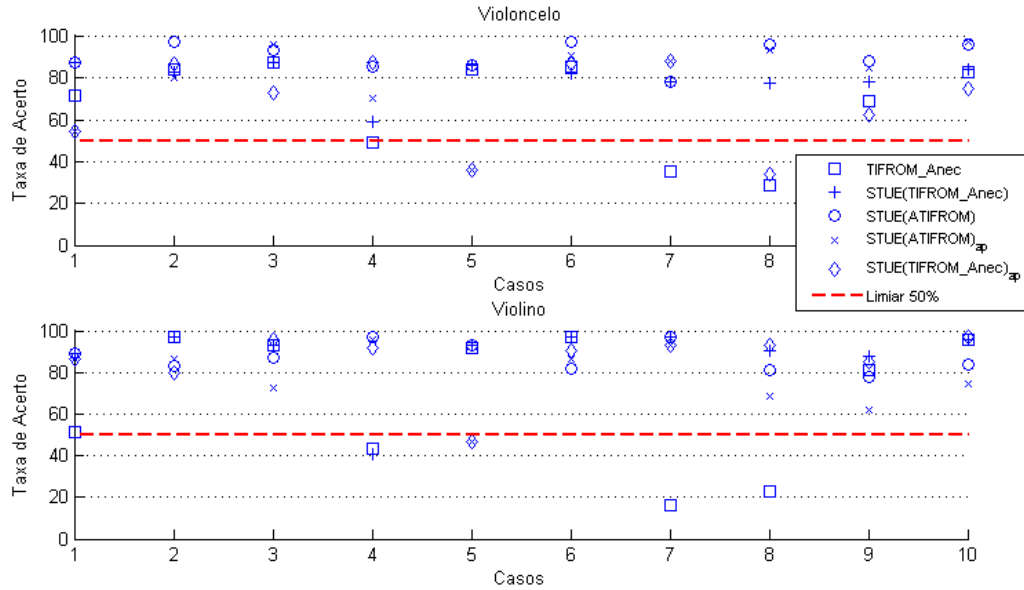


Figura 6.16: Taxa de Acerto para Misturas Monofônicas - Cordas.

Algoritmo	Metais	Percussão	Palhetas	Cordas
TIFROM_Anec	70%	70%	0%	70%
STUE(TIFROM_Anec)	80%	100%	20%	90%
STUE(ATIFROM(DFTNU))	80%	100%	20%	100%
STUE(TIFROM_Anec) _{ap}	90%	100%	50%	80%
STUE(ATIFROM(DFTNU)) _{ap}	90%	100%	20%	90%

Tabela 6.3: Porcentagem da Identificação Correta dos Instrumentos nas Misturas.

gerada por esse instrumento passou a compor uma mistura anecoica, que apresenta grau de dificuldade superior à mistura instantânea. O SRAIM teve dificuldades para identificar esse instrumento, classificando-o como outro instrumento da mesma família.

Por outro lado, se compararmos o algoritmo TIFROM_Anec com o algoritmo STUE(TIFROM_Anec), sabendo que a reconstrução é a mesma e que o método de estimação dos coeficientes difere, podemos verificar que, quando o algoritmo STUE(TIFROM_Anec) foi empregado, a taxa de acerto obtida pelo SRAIM sempre foi superior, independentemente da família de instrumentos avaliada.

Por fim, se compararmos o algoritmo STUE(TIFROM_Anec) e o STUE(ATIFROM(DFTNU)) com seus respectivos algoritmos STUE(TIFROM_Anec)_{ap} e STUE(ATIFROM(DFTNU))_{ap}, sabendo que a principal diferença foi a reconstrução, percebemos que, houve uma melhora perceptível do desempenho do SRAIM para as classes Metais e Palhetas com a reconstrução via solução Aproximada. Já para a classe Cordas, diferentemente das demais classes, percebe-se que

o desempenho do SRAIM foi melhor com a reconstrução via mascaramento binário bidimensional.

Capítulo 7

Conclusões e Trabalhos Futuros

Este capítulo apresenta as principais contribuições resultantes do estudo realizado nesta tese e descreve os principais caminhos que foram propostos, bem como os trabalhos futuros que se pretende investigar, decorrentes dos resultados desta pesquisa.

Este capítulo está dividido em quatro seções: a primeira descreve as conclusões relativas aos métodos de estimação dos coeficientes de misturas, tanto para os cenários de misturas instantâneas quanto para os de misturas anecoicas; a segunda apresenta as contribuições e conclusões decorrentes dos métodos de reconstrução propostos; a terceira apresenta as conclusões do uso de algumas técnicas propostas nesta tese quando combinadas num sistema de reconhecimento automático de instrumentos musicais; e por fim, a quarta propõe alguns tópicos de pesquisa para trabalhos futuros.

7.1 Métodos de Estimação dos Coeficientes do Sistema de Mistura

Duas hipóteses independentes foram consideradas para o desenvolvimento dos algoritmos propostos para a estimação dos coeficientes de mistura empregando a esparsidade como princípio. A primeira supõe que o desacoplamento dos sensores (caso instantâneo) permite um tratamento diferenciado em cada sensor, tornando os métodos de estimação que usam esparsidade mais robustos na presença de ruído ou de sinais interferentes, quando comparado com os métodos de separação que empregam somente o princípio da simultaneidade das amostras das fontes. Para explorar essa hipótese, foram desenvolvidas funções-objetivo para reposicionar os coeficientes estimados independentemente em cada sensor. Para o caso das misturas instantâneas, três algoritmos atendem a esse requisito de desacoplamento dos sensores: o SCAM, o STUE e o SCAz, tendo como algoritmos correlatos que não desacoplam os sensores o SCAC (para o SCAM) e o TIFROM (para o STUE e o

SCAz).

Para avaliar essa hipótese, cumpre comparar os resultados obtidos pelo algoritmo SCAM com os obtidos pelo algoritmo SCAC, e os do algoritmo SCAzTF com os do algoritmo TIFROM¹, uma vez que eles apresentam abordagens semelhantes, tendo como principal diferença o emprego do desacoplamento. Dos quatro cenários avaliados com presença de ruído (branco e *babble*) correspondendo a 20 valores distintos de SNRs, o algoritmo SCAMTF_2 foi melhor (em MSE médio) que o algoritmo SCACTF_2 em 14 casos (ou seja, 70% das vezes), enquanto o algoritmo SCAzTF foi superior em desempenho (MSE médio) ao TIFROM em 11 casos (ou seja, 55% das vezes). Todos esses 11 casos ocorreram nos cenários com menor SNR (≤ 15 dB), independentemente do tipo de ruído presente na mistura.

Portanto, a partir dos resultados dos experimentos efetuados nesta tese, podemos afirmar a validade da hipótese de que o uso do desacoplamento dos sensores torna as estimativas dos coeficientes de mistura mais robustas em relação ao MSE médio, no que tange à presença de ruído. Embora, o desvio mostre uma certa equivalência dos algoritmos, há o indicativo claro de preferência ao algoritmo TIFROM para $\text{SNR} \geq 20$ dB e, uma ligeira tendência ao desacoplamento das fontes para $\text{SNR} \leq 15$ dB.

A segunda hipótese supõe que estimativas grosseiras das fontes obtidas ou reconstruídas a partir de amostras de um dado sensor (desde que contenham correlação com as fontes originais) são preferíveis de serem usadas na busca de regiões SSZ do que o emprego das misturas para esse mesmo fim. Para explorar essa hipótese, foi empregado o mascaramento *HARD*, sendo que outras formas de reconstrução podem também ser empregadas. O algoritmo STUE, baseado nesse princípio, foi também desenvolvido para misturas instantâneas. Importa salientar que o algoritmo STUE tem por correlato o algoritmo clássico TIFROM, cujo processo de estimação não se vale das estimativas das fontes, mas das misturas. Para validar essa hipótese, novamente simulações com ruídos nos sensores foram empregadas, na expectativa de que o algoritmo que emprega as estimativas das fontes (ao invés das misturas) apresentasse maior robustez perante o ruído, já que este tende a promover uma redução das regiões SSZ. Essa hipótese foi testada para os casos instantâneo e anecoico, sendo que, para o caso anecoico, além do algoritmo STUE, derivou-se o algoritmo IMUE, que emprega esse mesmo princípio.

Para misturas instantâneas, os algoritmos SCAzTF e STUE apresentaram bons desempenhos, principalmente com o aumento do nível de ruído, sendo que em três dos quatro cenários avaliados eles apresentaram, para $\text{SNR} = 5$ dB, resultados superiores ao do algoritmo TIFROM. Nas misturas anecoicas, ao se analisarem as figuras

¹O algoritmo STUE emprega além do desacoplamento, o uso das estimativas das fontes, portanto, foi retirado da comparação.

que contêm as curvas da $MSE \times SNR$, pode-se verificar que o algoritmo STUE (com as diferentes inicializações) apresentou, para $SNR = 5$ dB, um desempenho superior aos dos algoritmos TIFROM_Anec e DUETm, exceto no cenário com sinais de áudio contaminados com ruído branco para estimação dos coeficientes de atraso, e que o algoritmo IMUE apresentou desempenho superior aos dos algoritmos TIFROM_Anec e DUETm, exceto nos cenários com sinais de áudio contaminados com ruído branco.

Se observarmos o desempenho a partir dos valores de MSE médio dos coeficientes de atenuação, podemos notar que, a técnica SSZ, utilizada para obter as estimativas reconstruídas pelo mascaramento binário nos algoritmos STUE e IMUE, comprovou ser eficiente, superando as técnicas que utilizam histogramas e busca das SSZ nas várias situações em que foram confrontadas, principalmente, nos cenários com menores valores de SNR. Portanto, as implementações propostas obtiveram desempenhos comparáveis aos das técnicas clássicas, tendo apresentado resultados ligeiramente superiores em alguns casos. Contudo, a análise da incerteza mostrou que para $SNR = 10$ dB o algoritmo IMUE(ATIFROM), em apenas 2 casos dos 20, apresentou desempenho superior ao dos algoritmos de referência.

Portanto, podemos afirmar que a segunda hipótese não foi completamente validada, ou seja, a obtenção dos coeficientes a partir das estimativas das fontes foi melhor do que a partir das misturas somente na média e para misturas que estão contaminadas com baixa SNR.

Além das hipóteses assumidas anteriormente, especificamente para o caso anecoico, foi elaborado um método para obter os coeficientes de atraso a partir de duas estimativas das fontes associadas ao mesmo sensor, as quais foram reconstruídas pelo mascaramento *HARD*. Novamente, essa hipótese foi validada por simulações. Para efeitos de comparação, usou-se o método de estimação dos coeficientes de atraso adotado pelo DUET nas implementações DUETm e TIFROM_Anec. Os resultados obtidos na estimação dos coeficientes de atraso evidenciam que, em três dos quatro cenários exercitados (áudio e voz combinados com ruído branco e *babble*), os algoritmos que utilizaram o método proposto obtiveram sempre desempenhos superiores aos algoritmos de referência DUETm e TIFROM_Anec para todos os valores de SNR empregados. Além disso, as implementações STUE(TIFROM_Anec) e ATIFROM(DFTNU) sempre obtiveram desempenhos superiores aos das implementações de referência, em todos os casos e para todas as SNRs testadas.

A análise da incerteza mostrou que para sinais de voz com ruído branco e *babble* os algoritmos propostos apresentaram desempenho superior ao apresentado pelos algoritmos de referência. Portanto, ficou evidenciada a superioridade do método proposto para estimar o atraso nas misturas anecoicas quando comparado com o método usado pelo DUET ou TIFROM_Anec, principalmente para sinais de voz.

Transformadas tempo-frequência não uniformes foram empregadas como transformações esparsificadoras no processo de estimação dos coeficientes de mistura, tanto no caso instantâneo quanto no caso anecoico. Algumas implementações dos algoritmos STUE, ATIFROM e IMUE empregaram essas transformadas. Comparando os resultados obtidos na presença de ruído, verificou-se que a implementação ATIFROM(DFTNU) apresentou resultados melhores em relação ao algoritmo TIFROM_Anec para baixas SNRs, independentemente da natureza do ruído. Assim, fica evidenciado que os bancos de filtros não uniformes podem efetivamente ser empregadas como transformadas esparsificadoras, e que possuem o potencial de melhorar o desempenho dos algoritmos para misturas ruidosas.

7.2 Métodos de Reconstrução das Fontes

Com relação à reconstrução das fontes, nesta tese foram apresentadas propostas baseadas em deflação na frequência, subtração temporal e correlação mínima, além de uma técnica de correção de erro a partir do mascaramento *HARD*.

O mascaramento corrigido, quando comparado ao mascaramento binário bidimensional, apresentou desempenho superior em todas as medidas para o caso com duas fontes de áudio, e ganhos nas medidas SDR e SAR para três e quatro fontes de voz em inglês.

O emprego da técnica de deflação em frequência (DEF_{L_F}) apresentou um desempenho médio equivalente ao mascaramento binário bidimensional.

O emprego da técnica de subtração temporal, $TSOE_{TF}$ não apresentou sempre desempenho superior ao do mascaramento binário bidimensional. Como a redução da indeterminação do sistema ocorre no domínio do tempo, possivelmente o critério adotado no domínio tempo-frequência para a determinação da fonte a ser subtraída pode não ter sido o mais indicado para a subtração temporal. Tal fato é corroborado ao verificarmos que a não utilização desse critério nos algoritmos de subtração temporal $TSAE_{T1}$ e $TSAE_{T2}$ resulta em desempenho médio superior.

O emprego da técnica de subtração temporal ($TSAE_{T1}$) apresentou desempenho superior ao do mascaramento binário bidimensional em relação à SDR e à SAR em todas as simulações executadas, nas quais a estimativa inicial empregada foi obtida pelo mascaramento corrigido.

O emprego da técnica de subtração temporal ($TSAE_{T2}$) com estimativa inicial pelo mascaramento corrigido também apresentou desempenho superior ao do mascaramento binário bidimensional em todas as simulações executadas em relação à SDR e à SAR. Contudo, seu desempenho foi prejudicado quando comparado com o da técnica de subtração temporal, $TSAE_{T1}$, devido, provavelmente, à simplificação feita em não se otimizar o fator de escala μ para reduzir o custo computacional.

A técnica de reconstrução para misturas instantâneas que minimiza a correlação, CorrMin (*SOFT*), apresentou o melhor desempenho em relação ao parâmetro SDR em todas as simulações realizadas. Além disso, sempre apresentou desempenho superior em relação à SIR quando comparado à técnica de minimização da norma l_1 , ModMin (*SOFT*).

A técnica CorrMin apresentou desempenho melhor do que o mascaramento binário para as misturas instantâneas, sendo a única técnica que conseguiu valores médios superiores em relação a todos os parâmetros para todos os cenários.

Portanto, podemos concluir que a técnica CorrMin apresenta tendência a obter desempenho médio superior ao do mascaramento e ao do módulo mínimo para misturas instantâneas, e que as técnicas de subtração temporal, TSAE_{T1} e TSAE_{T2}, apresentaram desempenhos médios superiores ao do mascaramento binário bidimensional nas misturas anecoicas, demonstrando a validade das abordagens propostas.

7.3 Emprego no SRAIM

Nesse experimento, avaliou-se o emprego de algumas técnicas propostas em misturas instantâneas e anecoicas para um sistema de reconhecimento automático de instrumentos musicais. Também foi avaliada a técnica de reconstrução aproximada combinada com o algoritmo STUE.

Nas misturas instantâneas os resultados obtidos na separação das sequências monofônicas se mostraram sempre válidos, ou seja, sempre foi possível estimar corretamente os instrumentos que estavam presentes nessas sequências. Os algoritmos empregados nesse exercício foram as técnicas de separação cega que trabalham com esparsidade no domínio temporal, uma vez que os sinais empregados são esparsos nesse domínio.

Nas misturas anecoicas os resultados obtidos pelo SRAIM com os algoritmos propostos nesta tese podem ser considerados satisfatórios (exceto para o instrumento Saxofone Soprano), onde a taxa de acerto para a maioria dos instrumentos ficou acima de 80%.

Foi possível também constatar que o uso da técnica de reconstrução aproximada quando combinada com o algoritmo STUE apresentou resultados satisfatórios. O uso das estimativas das fontes no lugar das misturas para a estimação dos coeficientes foi benéfico para o SRAIM, visto que o algoritmo STUE sempre apresentou ganhos nas taxas de acerto quando comparado com o algoritmo TIFROM_Anec.

7.4 Trabalhos Futuros

Este trabalho abordou a pesquisa de métodos que empregam o princípio da esparsidade nas diversas fases do processo de separação cega das fontes. Apresentaremos nesta seção, possibilidades de trabalhos futuros nas etapas de pré-processamento, de estimação dos coeficientes de mistura e reconstrução das fontes. Não incluiremos os trabalhos futuros que podem ocorrer no emprego das técnicas elaboradas nesta tese em outras áreas ou combinadas com outros métodos visando resolver problemas variados.

Na fase de pré-processamento, uma abordagem que foi pouco explorada foi o uso de transformadas tempo-frequência, tais como a CQT, wavelet, wavelet *packet* (WPD) [115], Choi-Williams (CWD) [116] e Wigner-Ville (VWD) [117] combinadas com os algoritmos propostos. Dessa forma, pretende-se avaliar a técnica SSZ com os sensores desacoplados ou com o uso das estimativas das fontes empregando essas transformadas.

Na fase de estimação, é possível introduzir uma melhoria na técnica que utiliza as estimativas das fontes na busca das regiões SSZ, de forma que haja uma comunicação entre as razões estimadas pelos pares já avaliados aos pares de fontes estimadas ainda não avaliados, visando garantir uma distinção mínima das razões.

Na fase de estimação dos coeficientes nas misturas anecoicas, as técnicas propostas nesta tese para obter os atrasos seguem um caminho diferente do método proposto pelo DUET. Essas técnicas foram validadas apenas para pequenos atrasos. Como um fator de *upsampling* é utilizado para calcular o atraso, resultados preliminares indicam que é possível obter bons resultados com atrasos maiores. Além disso, é possível estabelecer um fator de *upsampling* variável com a frequência de amostragem, visando refinar a medida de atraso.

O algoritmo IMUE pode facilmente ser generalizado para o caso instantâneo. Devido ao bom desempenho obtido pelo algoritmo STUE nas misturas instantâneas na presença de ruído, o algoritmo IMUE para essas misturas pode vir a se tornar uma opção interessante, principalmente quando as fontes apresentam características esparsas.

O algoritmo SCAM pode ser generalizado para o caso anecoico, já que a estimação dos coeficientes ocorre de forma desacoplada em relação aos sensores. Diferentemente do DUET, o algoritmo SCAM não utiliza histogramas bidimensionais, podendo-se estimar de forma casada os coeficientes de atraso e atenuação, bastando empregarmos estes últimos coeficientes (via mascaramento binário unidimensional) para obtermos o par de estimativas associados a uma fonte para estimarmos o atraso. Contudo, um problema que deve ser levado em consideração é que, para misturas anecoicas, o DUET, ao fazer um histograma bidimensional, naturalmente elimina os

falsos picos das suas projeções (somente do atraso ou da atenuação). Portanto, para contornar esse problema potencial, essa nova abordagem deve estimar um número maior de picos na dimensão atenuação, e usar algum outro princípio, tal como a baixa correlação entre as estimativas das fontes, para reduzir os falsos picos nessa dimensão antes de estimar os coeficientes de atraso.

Já na fase de reconstrução das fontes, os métodos de subtração temporal podem apresentar ganhos mais significativos com o uso da recursão. Assim, um trabalho futuro a ser realizado é determinar o limite desse ganho. Por fim, ainda na fase da reconstrução, o algoritmo CorrMin adotou o critério de minimização da Eq. (5.5) para todas as possibilidades a cada ponto, resultando em um algoritmo guloso. Uma abordagem potencialmente melhor, mas com um custo computacional superior, seria para cada raia usar um critério que incorporasse mais pontos no cálculo da função correlação a ser minimizada.

O algoritmo de deflação em frequência empregou um critério baseado na variância para determinar as componentes que serão deflacionadas, que podem corresponder a um agrupamento homogêneo, porém, contendo poucas representações no conjunto. Portanto, um caminho a ser investigado é estabelecer uma medida para detectar essas anomalias, e a partir disso estabelecer um outro critério, que nesses casos (contendo poucas representações), formaria agrupamentos com mais representações adequadas da fonte a ser deflacionada.

No experimento final empregando o SRAIM, pode-se, em trabalhos futuros, avaliar o desempenho desse sistema com os diferentes algoritmos de separação cega perante misturas contaminadas por ruído, a fim de verificar a sua robustez. Além disso, o SRAIM foi treinado com um conjunto específico de fabricantes de instrumentos musicais e de intérpretes, estando portanto, limitado com relação a esses elementos. Logo, é possível aumentar seu grau de generalização nesses elementos, principalmente a partir da base de dados RWC, que contém conjuntos com variações nesses elementos que não foram usados.

Referências Bibliográficas

- [1] CONGEDO, M., GOUYPAILLER, C., JUTTEN, C. “On the Blind Source Separation of Human Electroencephalogram by Approximate Joint Diagonalization of Second Order Statistics”, *Clinical Neurophysiology*, v. 119, n. 12, pp. 2677–2686, Dec. 2008.
- [2] FORNASIER, M., PITOLLI, F., PIZZELLA, V. “Blind Source Separation with Sparsity Constraints for Magnetoencephalography”, *Communications to SIMAI Congress*, v. 1, May 2006.
- [3] VAYÁ, C., RIETA, J. J., SÁNCHEZ, C., et al. “Performance Study of Convolutional BSS Algorithms Applied to the Electrocardiogram of Atrial Fibrillation”. In: *Independent Component Analysis and Blind Signal Separation, 6th International Conference, ICA 2006*, pp. 495–502, SC, USA, Mar. 2006.
- [4] LEI, T., UDUPA, J. K. “Blind Source Separation (BSS) for fMRI Analysis”. v. 4321, pp. 312–320, May 2001.
- [5] FREISLEBEN, B., HAGEN, C., BORSCHBACH, M. “A Neural Network for Blind Acoustic Signal Separation”. In: *Proceedings of the 1996 International Conference on Robotics Vision and Parallel Processing for Industrial Automation*, v. 1, pp. 67–72, Malaysia, Nov. 1996.
- [6] MUKHERJEE, A., MAITI, S., DATTA, A. “Spectrum Sensing for Cognitive Radio Using Blind Source Separation and Hidden Markov Model”. In: *Advanced Computing & Communication Technologies (ACCT), 2014 Fourth International Conference on*, pp. 409–414, India, Feb. 2014.
- [7] BATALHEIRO, P. B. *Novos Algoritmos em Subbandas para Filtragem Adaptativa e Separação Cega de Fontes*. Tese de doutorado em engenharia elétrica, Universidade Federal do Rio de Janeiro, RJ, Brasil, Jul. 2009.
- [8] MEYER, J., SIMMER, K. “Multi-channel Speech Enhancement in Car Environment Using Wiener Filtering and Spectral Subtraction”. In: *Proceedings*

of Acoustics, Speech, and Signal Processing (ICASSP-97, v. 2, pp. 1167–1170, Germany, Apr. 1997.

- [9] VARMA, K. *Time-Delay-Estimate Based Direction-of-Arrival Estimation for Speech in Reverberant Environments*. Masters thesis in electrical engineering, Virginia Polytechnic Institute and State University, VA, USA, Oct. 2002.
- [10] HYVÄRINEN, A., KARHUNEN, J., OJA, E. *Independent Component Analysis*. New York, USA, John Wiley and Sons Inc., 2001.
- [11] PIENIEZNY, A. “Intrapulse Analysis of Radar Signal by the Use of Hough Transform”. In: *2008 Proceedings of International Conference on Date of Conference*, pp. 306 – 309, Ukraine, Feb. 2008.
- [12] SHAH, G., PAPADIAS, C. “Separation of Cardiorespiratory Sounds Using Time-Frequency Masking and Sparsity”. In: *Digital Signal Processing (DSP), 2013 18th International Conference on*, pp. 1–6, Greece, Jul. 2013.
- [13] PUIGT, M., BERNÉ, O., GUIDARA, R., et al. “Cross-validation of Blindly Separated Interstellar Dust Spectra”. In: *Proceedings of ECMS 2009*, pp. 41–48, Spain, Jul. 2009.
- [14] MONIY, A., BEAN, C. J., LOKMER, I., et al. “Seismic Signal Source Separation”. In: *22nd IET Irish Signals and Systems Conference*, Irish, Jun. 2011.
- [15] MA, J., HUANG, G., ZHOU, D., et al. “Underdetermined Blind Sorting of Radar Signals Based on Sparse Component Analysis”. In: *Communication Technology (ICCT), 2012 IEEE 14th International Conference*, pp. 1296–1300, China, Nov. 2012.
- [16] KLAPURI, A. “Introduction to Music Transcription”. In: Klapuri, A., Davy, M. (Eds.), *Signal Processing Methods for Music Transcription*, Springer US, pp. 3–20, 2006.
- [17] ABRARD, F., DEVILLE, Y., WHITE, P. “A New Source Separation Approach for Instantaneous Mixtures Based on Time-Frequency Analysis”. In: *In, 5th Workshop on Electronics, Control, Modelling, Measurement and Signals*, pp. 259–267, France, Jun. 2001.
- [18] PIRES FILHO, J. C. *Classificação de Instrumentos Musicais em Configurações Monfônicas e Polifônicas*. Dissertação de mestrado, Universidade Federal do Rio de Janeiro, COPPE/UFRJ, RJ, Brasil, Setembro 2009.

- [19] REJU, V. G., KOH, S. N., SOON, I. Y. “An Algorithm for Mixing Matrix Estimation in Instantaneous Blind Source Separation”, *Signal Processing, Elsevier*, v. 89, pp. 1762–1773, Mar. 2009.
- [20] BOFILL, P., ZIBULEVSKY, M. *Sparse Underdetermined ICA: Estimating the Mixing Matrix and the Sources Separately*. Tech. report, Joint Speech Research Unit, Ruislip, England, 2000.
- [21] CICHOCKI, A., S. AMARI, K. S. “ICALAB Toolboxes”, <http://www.bsp.brain.riken.jp/ICALAB>, Mar. 2007.
- [22] PUIGT, M. “matlab code for LLTIFROM”, <http://www.ast.obs-mip.fr/bss-softwares>, Oct. 2008.
- [23] RICKARD, S. “The DUET Blind Source Separation Algorithm”. In: Makino, S., Sawada, H., Lee, T.-W. (Eds.), *Blind Speech Separation*, Signals and Communication Technology, Springer, Netherlands, pp. 217–241, Netherland, 2007.
- [24] PUIGT, M., DEVILLE, Y. “Time-Frequency Ratio-Based Blind Separation Methods for Attenuated and Time-Delayed Sources”, *Mechanical Systems and Signal Processing*, v. 19, n. 6, pp. 1348–1379, Nov. 2005.
- [25] LI, H., ADALI, T. “A Class of Complex ICA Algorithms Based on the Kurtosis Cost Function”, *IEEE Transactions on Neural Networks*, v. 19, n. 3, pp. 408–420, Mar. 2008.
- [26] BUCHNER, H., KELLERMAN, W. “A Fundamental Relation Between Blind and Supervised Adaptive Filtering Illustrated For Blind Source Separation and Acoustic Echo Cancellation”. In: *Hands-Free Speech Communication and Microphone Arrays, (HSCMA)*, pp. 17–20, Italy, May 2008.
- [27] MAKINO, S., LEE, T.-W., SAWADA, H. *Blind Speech Separation*. 1 ed. New York, USA, Springer: Series: Signals and Communication Technology, 2007.
- [28] ZIBULEVSKY, M., PEARLMUTTER, B. *Blind Source Separation by Sparse Decomposition*. Technical report no. cs99-1, University of New Mexico, Albuquerque, Jul. 1999.
- [29] BOFILL, P., ZIBULEVSKY, M. “Underdetermined Blind Source Separation Using Sparse Representations”, *Signal Processing*, v. 81, pp. 2353–2362, Nov. 2001.

- [30] GRIBONVAL, R., LESAGE, S. “A Survey of Sparse Components Analysis for Blind Source Separation: principles, perspectives and new challenges”. In: *European Symposium on Artificial Neural Networks - ESANN2006 proceedings*, pp. 323–330, Belgium, Apr. 2006.
- [31] KISILEV, P., ZIBULEVSKY, M., ZEEVI, Y. Y. “A Multiscale Framework for Blind Separation of Linearly Mixed Signals”, *Journal of Machine Learning Research*, v. 4, pp. 1339–1363, Dec. 2003.
- [32] RAFII, Z., PARDO, B. “Degenerate Unmixing Estimation Technique Using The Constant Q Transform”. In: *Proceedings on 36th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Czech Republic, May 2011.
- [33] PIRES FILHO, J. C., PETRAGLIA, M. R., HADDAD, D. B. “Evaluation of Techniques for Blind Sources Separation in the Identification of Musical Instruments”. In: *17th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 372–375, RJ, Brazil, Jun. 2010.
- [34] BOFILL, P., ZIBULEVSKY, M. “Sound Examples of Sparse Underdetermined Instantaneous BSS”, <http://people.ac.upc.es/pau/shpica/instant.html>, Jul. 2011.
- [35] DEVILLE, Y., PUIGT, M., ALBOUY, B. “Time-Frequency Blind Signal Separation: Extended Methods, Performance Evaluation for Speech Sources”. In: *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2004)*, pp. 255–260, Hungary, Jul. 2004.
- [36] ABRARD, F., DEVILLE, Y. “A Time-Frequency Blind Signal Separation Method Applicable to Underdetermined Mixtures of Dependent Sources”, *Signal Processing*, v. 85, n. 7, pp. 1389–1403, Jul. 2005.
- [37] MITCHELL, T. *Machine Learning*. 1 ed. New York, USA, McGraw Hill, Inc., 1997.
- [38] SCHOBEN, D., TORKKOLA, K., SMARAGDIS, P. “Evaluation of Blind Signal Separation Methods”. In: *Proc. Int. Symposium on ICA and BSS (ICA 99)*, pp. 261–266, France, Jan. 1999.
- [39] LAMBERT, R. “Difficulty Measures and Figures of Merit for Source Separation”. In: *Proc. Int. Symposium on ICA and BSS (ICA 99)*, pp. 133–138, France, Jan. 1999.

- [40] VINCENT, E., GRIBONVAL, R., FÉVOTTE, C. “Performance Measurement in Blind Audio Source Separation”, *IEEE Transactions on Audio, Speech and Language Processing*, v. 14, n. 4, pp. 1462–1469, Jul. 2006.
- [41] VINCENT, E., SAWADA, H., BOFILL, P., et al. “First Stereo Audio Source Separation Evaluation Campaign: Data, Algorithms and Results”. In: *Proceedings of the 7th International Conference on Independent Component Analysis and Blind Source Separation (ICA)*, pp. 552–559, UK, Sep. 2007.
- [42] BELOUHRANI, A., ABED-MERAIM, K., CARDOSO, J. F., et al. “Second Order Blind Separation of Temporally Correlated Sources”. In: *Proc. Int. Conf. on Digital Signal Proc.*, pp. 346–351, Cyprus, May 1993.
- [43] YEREDOR, A. “Blind Separation of Gaussian Sources via Second-Order Statistics with Asymptotically Optimal Weighting”, *IEEE Signal Processing Letters*, v. 7, n. 7, pp. 2353–2362, Jul. 2000.
- [44] JOURJINE, A., RICKARD, S., YILMAZ, O. “Blind Separation of Disjoint Orthogonal Signals: Demixing N Sources from 2 Mixtures”. In: *In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 2985–2988, Turkey, Jun. 2000.
- [45] ABRARD, F., DEVILLE, Y., SABATIER, P. “From Blind Source Separation to Blind Source Cancellation In The Underdetermined Case: A New Approach Based On Time-Frequency Analysis”. In: *3rd Int. Conf. Independent Component Analysis Signal Separation (ICA)*, pp. 734–739, CA, USA, Dec. 2001.
- [46] LUO, Y., CHAMBERS, J. A. “Active Source Selection Using Gap Statistics for Underdetermined Blind Source Separation”. In: *Proceedings of Seventh International Symposium on Signal Processing and Its Applications (ISSPA)*, v. 1, pp. 137–140, France, Jul. 2003.
- [47] COHEN, L. *Time-Frequency Analysis: Theory and Applications*. Upper Saddle River, NJ, USA, Prentice Hall PTR, Inc., 1995.
- [48] LUO, Y., CHAMBERS, J. A. “A Modified Underdetermined Blind Source Separation Algorithm Using Competitive Learning”. In: *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis*, pp. 966–969, Italy, Sep. 2003.

- [49] ZHANG, Y. J., LIU, Z. Q. “Self-Splitting Competitive Learning: A New On-Line Clustering Paradigm”, *IEEE Transactions on Neural Networks*, v. 13, n. 2, pp. 369–380, Mar. 2002.
- [50] HAYKIN, S. *Redes Neurais: Princípios e Práticas*. 2 ed. Porto Alegre, Brasil, Bookman, 2001.
- [51] LUO, Y., LAMBOTHRAN, S., CHAMBERS, J. A. “A New Block Based Time-Frequency Approach for Underdetermined Blind Source Separation”. In: *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, v. 5, pp. 537–540, Canada, May 2004.
- [52] LINHTRUNG, N., BELOUHRANI, A., ABEDMERAIM, K., et al. “Separating More Sources Than Sensors Using TimeFrequency Distributions”. In: *Sixth International Symposium on Signal Processing and its Applications*, v. 2, pp. 583–586, Malaysia, Aug. 2001.
- [53] FLEGO, F., ARAKI, S., SAWADA, H., et al. “Underdetermined Blind Separation for Speech in Real Environments with F0 Adaptive Comb Filtering”. In: *Proceedings of IEEE International Workshop on Acoustic Echo and Noise Control (IWAENC)*, pp. 93–96, Netherland, Sep. 2005.
- [54] ZHANG, W., LIU, J., SUN, J., et al. “A New Two-Stage Approach to Underdetermined Blind Source Separation Using Sparse Representation”. In: *Proceedings of 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 953–956, Hawaii, USA, Apr. 2007.
- [55] LI, Y., AMARI, S., CICHOCKI, A. “Underdetermined Blind Source Separation Based on Sparse Representation”, *IEEE Transactions on Signal Processing*, v. 54, n. 2, pp. 423–437, Feb. 2006.
- [56] TICHAVSKY, P., YEREDOR, A., KOLDOVSKY, Z. “A Fast Asymptotically Efficient Algorithm for Blind Separation of a Linear Mixture of Block-Wise Stationary Autoregressive Processes”. In: *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '09, pp. 3133–3136, DC, USA, Apr. 2009.
- [57] CHOI, S., CICHOCKI, A., BELOUHRANI, A. “Second Order Nonstationary Source Separation”, *Journal of VLSI Signal Processing*, v. 32, n. 1-2, pp. 93–104, Aug. 2002.

- [58] BUCHNER, H., AICHNER, R., KELLERMANN, W. “A Generalization of Blind Source Separation Algorithms for Convolutional Mixtures Based on Second-Order Statistics”, *IEEE Transactions on Audio, Speech, and Language Processing*, v. 13, n. 1, pp. 120–134, Jan. 2005.
- [59] PHAM, D.-T. “Exploiting Source Non Stationary and Coloration in Blind Source Separation”. In: *Proceedings of Digital Signal Processing (DSP), 2002 Conference*, Greece, Jul. 2002.
- [60] MANMONTRI, U., NAYLOR, P. “A Class of Frobenius Norm-Based Algorithms Using Penalty Term and Natural Gradient for Blind Signal Separation”, *IEEE Transactions on Audio, Speech, and Language Processing*, v. 16, n. 6, pp. 1181–1193, Aug. 2008.
- [61] TICHAVSKY, P., YEREDOR, A. “Fast Approximate Joint Diagonalization Incorporating Weight Matrices”, *IEEE Transactions on Signal Processing*, v. 57, n. 3, pp. 878–891, Mar. 2009.
- [62] PHAM, D.-T., GARAT, P. “Blind Separation of Mixtures of Independent Sources Through a Quasi Maximum Likelihood Approach”, *IEEE Transactions on Signal Processing*, v. 45, n. 7, pp. 1712–1725, Jul. 1997.
- [63] LI, F., ZHANG, Y., WU, J., et al. “A Robust K-plane Clustering Algorithm for Blind Separation of Underdetermined Mixtures of Sparse Sources”. In: *Proceedings of 2010 International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, pp. 331–334, China, Mar. 2010.
- [64] GEORGIEV, P., THEIS, F., CICHOCKI, A. “Sparse Component Analysis and Blind Source Separation of Underdetermined Mixtures”, *IEEE Transactions on Neural Networks*, v. 16, n. 4, pp. 992–996, Jul. 2005.
- [65] HARTIGAN, J. *Clustering algorithms*. 1 ed. New York, USA, John Wiley & Sons, 1975.
- [66] BROWN, J. C. “Calculation of Constant Q Spectral Transform”, *The Journal of the Acoustical Society of America (JASA)*, v. 89, n. 1, pp. 425–434, Jan. 1991.
- [67] BROWN, J. C., PUCKETTE, M. S. “An Efficient Algorithm for the Calculation of a Constant Q Transform”, *The Journal of the Acoustical Society of America (JASA)*, v. 92, n. 5, pp. 2698–2701, Nov. 1992.

- [68] BROWN, R. G., HWANG, P. *Introduction to Random Signals and Applied Kalman Filtering*. 3 ed. New York, USA, John Wiley & Sons, 1996.
- [69] BENAROYA, L., BIMBOT, F., GRIBONVAL, R. “Audio Source Separation with a Single Sensor”, *IEEE Transactions on Audio, Speech, Language Processing*, v. 14, n. 1, pp. 191–199, Jan. 2006.
- [70] AYLON, D., GIL-PITA, R., JARABO-AMORES, P., et al. “Energy-Weighted Mean Shift Algorithm for Speech Source Separation”. In: *Proceedings of IEEE Statistical Signal Processing Workshop (SSP)*, pp. 785–788, France, Jun. 2011.
- [71] CHENG, Y. “Mean Shift, Mode Seeking, and Clustering”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 17, n. 8, pp. 790–799, Aug. 1995.
- [72] PARZEN, E. “On Estimation of a Probability Density Function and Mode”, *The Annals of Mathematics Statistics*, v. 33, n. 3, pp. 1065–1076, Sep. 1962.
- [73] PUIGT, M., GRIFFIN, A., MOUCHTARIS, A. “Nonlinear blind mixture identification using local sourcesparsity and functional data clustering”. In: *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2012 IEEE 7th, IEEE Conference Publications*, pp. 481–484, NJ, USA, Jun. 2012.
- [74] ZHANG, Y., WU, K., TAN, G., et al. “An online adaptive algorithm for underdetermined blindsource separation”. In: *Signal Processing (ICSP), 2014 12th International Conference, IEEE Conference Publications*, pp. 467–472, China, Oct. 2014.
- [75] NOCK, R., NIELSEN, F. “On Weighting Clustering”, *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 28, n. 8, pp. 1223–1235, 2006.
- [76] BEZDEK, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Advanced applications in pattern recognition. New York, USA, Plenum Press, 1981.
- [77] BOBIN, J., RAPIN, J., LARUE, A., et al. “Sparsity and Adaptivity for the Blind Separation of Partially Correlated Sources”, *Signal Processing, IEEE Transactions*, v. 63, n. 5, pp. 1199–1213, Jan. 2015.
- [78] VAN LOAN, C. *Computational Frameworks for the Fast Fourier Transform*. PA, USA, Society for Industrial and Applied Mathematics, 1992.

- [79] MITRA, S. K. *Digital Signal Processing. A Computer-Based Approach*. 2 ed. New York, USA, Ed. McGraw-Hill, 2001.
- [80] SPÄTH, H., MUCHA, H. *Cluster Dissection and Analysis: Theory, FORTRAN Programs, Examples. (Translator: Johannes Goldschmidt)*. West Sussex, UK, Ellis Horwood Ltd Wiley, 1985.
- [81] SEBER, G. A. F. *Multivariate Observations*. 1 ed. New York, USA, John Wiley & Sons, 1984.
- [82] GÄVERT, H., HURRI, J., SÄRELÄ, J., et al. “FASTICA(mixedsig) estimates the independent components from given multidimensional signals”, <http://www.cis.hut.fi/projects/ica/fastica/>, Oct. 2005.
- [83] BELOUHRANI, A., ABED-MERAIM, K. “Séparation Aveugle au Second Ordre de Sources Corrélées”. In: *Proc. Grets*, pp. 309–312, France, Sep. 1993.
- [84] BELOUHRANI, A., CICHOCKI, A. “Robust Whitening Procedure in Blind Source Separation Context”, *Electronics Letters*, v. 36, n. 24, pp. 2050–2053, Nov. 2000.
- [85] CICHOCKI, A., AMARI, S. *Adaptive Blind Signal and Image Processing*. New York, USA, John Wiley and Sons Inc., 2002.
- [86] CARDOSO, J.-F. “matlab code for jadeR”, <http://sig.enst.fr/cardoso/stuff.html>, May 2005.
- [87] CARDOSO, J.-F., SOULOUMIAC, A. “Blind Beamforming for Non Gaussian Signals”, *IEE Proceedings-F*, v. 140, n. 6, pp. 362–370, Dec. 1993. <ftp://sig.enst.fr/pub/jfc/Papers/iee.ps.gz>.
- [88] CARDOSO, J.-F. “High-Order Contrasts for Independent Component Analysis”, *Neural Computation*, v. 11, n. 1, pp. 157–192, Jan. 1999.
- [89] YUAN, Y.-J., HUANG, Z.-T., SHA, Z.-C. “Specific Emitter Identification Based on Transient Energy Trajectory”, *Progress In Electromagnetics Research C*, v. 44, pp. 67–82, Sep. 2013.
- [90] SAHMEL, P. H. *Eigenspace Approach to Specific Emitter Identification of Orthogonal Frequency Division Multiplexing Signals*. Master of science in electrical engineering, Faculty of the Virginia Polytechnic Institute and State University, Virginia, USA, Nov. 2011.

- [91] VAIDYANATHAN, P. P. *Multirate Systems and Filter banks*. Upper Saddle River, NJ, USA, Prentice-Hall, 1993.
- [92] BRACCINI, C., OPPENHEIM, A. V. “Unequal Bandwidth Spectral Analysis Using Digital Frequency Warping”, *IEEE Transactions on Acoustic, Speech and Signal Processing*, v. ASSP-22, pp. 236–244, Aug. 1974.
- [93] OPPENHEIM, A., JOHNSON, D., STEIGLITZ, K. “Computation of Spectra with Unequal Resolution Using the Fast Fourier Transform”, *Proceedings of IEEE*, v. 59, n. 6, pp. 299–301, Feb. 1971.
- [94] BAGCHI, S., MITRA, S. K. *The Nonuniform Discrete Fourier Transform and its Applications in Signal Processing*. Boston, Dordrecht, London, Kluwer Academic Publishers, 1999.
- [95] MAKUR, A., MITRA, S. “Warped Discrete-Fourier Transform: Theory and Applications”, *IEEE Transactions on Circuits and Systems*, v. 48, n. 9, pp. 1086–1093, Sep. 2001.
- [96] BOX, G. E. P., JENKINS, G. M., REINSEL, G. C. *Time Series Analysis: Forecasting and Control*. 3 ed. Upper Saddle River, NJ, Prentice-Hall, 1994.
- [97] PIRES FILHO, J. C., PETRAGLIA, M. R. “Improvements in Blind Source Separation of Anechoic Underdetermined Speech Mixtures”. In: *In: Telecommunications Symposium (ITS), 2014 International, IEEE Conference Publications*, pp. 1–4, SP, Brasil, Aug. 2014. doi: 10.1109/ITS.2014.6948009.
- [98] VINCENT, E., SAWADA, H. “Stereo Audio Source Separation Evaluation Campaign”, via <http://www.irisa.fr/metiss/SASSECO7/dev.zip>, Jan. 2010.
- [99] MASSOD, M., AL-NAFFOURI, T. Y. “Sparse Reconstruction Using Distribution Agnostic Bayesian Matching Pursuit”, *IEEE Transactions on Signal Processing*, v. 61, n. 21, pp. 5298–5309, Nov. 2013.
- [100] REDDY, A. M., RAJ, B. “Soft Mask Methods for Single-Channel Speaker Separation”, *IEEE Transactions on Audio, Speech, and Language Processing*, v. 15, pp. 1766–1776, Jul. 2007.
- [101] DE MENEZES LAPORTE, L. V. *Algoritmos de Separação Cega de Sinais de Áudio no Domínio da Frequência em Ambientes Reverberantes: Estudo e*

Comparações. Dissertação de mestrado, Universidade Federal do Rio de Janeiro, COPPE/UFRJ, RJ, Brasil, Outubro 2010.

- [102] LAGARIAS, J. C., REEDS, J. A., WRIGHT, M. H., et al. “Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions”, *SIAM Journal of Optimization*, v. 9, n. 1, pp. 112–147, 1998.
- [103] KABAL, P., RAMACHANDRAN, R. P. “The Computation of Line Spectral Frequencies Using Chebyshev Polynomials”, *IEEE Transactions Acoustics, Speech, Signal Processing*, v. 34, n. 6, pp. 1419–1426, Dec. 1986.
- [104] MERMELSTEIN, P. “Distance Measures for Speech Recognition, Psychological and Instrumental”. In: *Proceedings of the Joint Workshop on Pattern Recognition and Artificial Intelligence*, pp. 374–388, Massachusetts, USA, Jun. 1976.
- [105] DAVIS, S. B., MERMELSTEIN, P. “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 28, n. 4, pp. 357–366, Aug. 1980.
- [106] BOSER, B. E., GUYON, I. M., VAPNIK, V. N. “A Training Algorithm for Optimal Margin Classifiers”. In: *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pp. 144–152, PA, USA, Jul. 1992. ACM Press.
- [107] CORTES, C., VAPNIK, V. “Support-Vector Networks”, *Mach. Learn.*, v. 20, pp. 273–297, Sep. 1995.
- [108] BRIDLE, J. S., BROWN, M. D. *An Experimental Automatic Word Recognition System*. Tech. report, Joint Speech Research Unit, Ruislip, England, Dec. 1974.
- [109] COVER, T. M., HART, P. “Nearest Neighbor Pattern Classification”, *IEEE Transactions on Information Theory*, v. 13, n. 1, pp. 21–27, Jan. 1967.
- [110] GOTO, M., NISHIMURA, T. “RWC Music Database: Music Genre Database and Musical Instrument Sound Database”. In: *ISMIR*, pp. 229–230, Oct. 2003.
- [111] FRITTS, L. “Musical Instruments Samples of IOWA University, MIS”. <http://theremin.music.uiowa.edu/MIS.html>, 1997.
- [112] OPOLKO, F., WAPNICK, J. “McGill University Master Samples”. <http://www.music.mcgill.ca/resources/mums.html>, 1987.

- [113] HADDAD, D. B., PETRAGLIA, M. R., BATALHEIRO, P. B., et al. “Análise de Componentes Esparsos: Separação Cega de Fontes e Aplicações”, *Learning and Nonlinear Models*, v. 10, pp. 19–35, 2012.
- [114] HADDAD, D. B., PETRAGLIA, M. R., LOVISOLO, L., et al. “Emprego de Técnicas de Matching Pursuit para a Identificação dos Parâmetros de um Sistema de Mistura Anecóico”. In: *In: XI Congresso de Engenharia de Áudio da AES Brasil*, pp. 55–62, SP, Brasil, May 2013.
- [115] AKANSU, A. N., HADDAD, P. R. *Multiresolution Signal Decomposition, Second Edition: Transforms, Subbands, and Wavelets (Series in Telecommunications)*. Boston, MA, USA, Academic Press, 1992.
- [116] CHOI, H., WILLIAMS, W. J. “Improved Time-Frequency Representation of Multicomponent Signals Using Exponential Kernels”, *IEEE. Trans. Acoustics, Speech, Signal Processing*, v. 37, n. 6, pp. 862–871, Jun. 1989.
- [117] WIGNER, E. “On the Quantum Correction For Thermodynamic Equilibrium”, *Physical Review*, v. 40, n. 5, pp. 749, 1932.
- [118] HABETS, E. A. P., COHEN, I., GANNOT, S. “Generating Nonstationary Multisensor Signals under a Spatial Coherence Constraint”, *The Journal of the Acoustical Society of America (JASA)*, v. 124, n. 5, pp. 2911–2917, Nov. 2008.

Apêndice A

Fontes Sonoras e Ruídos

Os dados empregados neste trabalho são compostos por quatro conjuntos distintos de sinais de áudio: sinais de voz em inglês e em japonês, tendo cada um deles 10s de duração, amostrados a taxa de 16 kHz e obtidos de [98]; sinais de flautas, com 2 s de duração cada, amostrados a taxa de 8 kHz e obtidos de [34]; e sinais de notas de instrumentos musicais, obtidos de [110], com comprimentos variados, amostrados a taxa de 44,1 kHz, formando sequências de notas musicais com intervalos de silêncio (*gaps*) entre elas [18]¹.

Espectrogramas são gráficos que ilustram, geralmente através de cores, a distribuição da energia dos sinais em suas componentes de frequência ao longo do tempo. Nas Figs. A.1 a A.4 são mostrados os espectrogramas de alguns dos sinais empregados. Em cada uma dessas figuras, são mostrados, nos gráficos (a)-(c), os espectros de três sinais de cada tipo de fonte, ao passo que no gráfico (d) é mostrado o espectro da mistura formada pela soma dos sinais das três fontes.

Ao observarmos os espectrogramas dos sinais de voz, áudio e sequência de notas (trompa, trombone e trompete), podemos verificar que suas características são distintas. Nos espectrogramas de sinais de voz, mostrados nas Figs. A.1 e A.2, as componentes de frequências (médias e altas) dos sinais de voz em japonês apresentam menor duração do que as dos sinais de voz em inglês, havendo em quase todos os sinais de voz intervalos com pouca informação. Nesses sinais, as variações na frequência ocorrem de forma contínua. O sinal de flautas apresenta informação durante praticamente toda a sua duração, e a distribuição de sua energia na frequência varia de forma discreta, havendo descontinuidades nas componentes de frequência. Por fim, na sequência de notas, é possível observar os *gaps* no domínio do tempo, em que não há presença de sinal, e a continuidade das suas componentes de frequência nos demais intervalos.

A Tabela A.1 apresenta os valores da curtose dos sinais das Figs. A.1 a A.4, cal-

¹Na Fig. A.4 os sinais foram reamostrados para a taxa de 16 kHz.

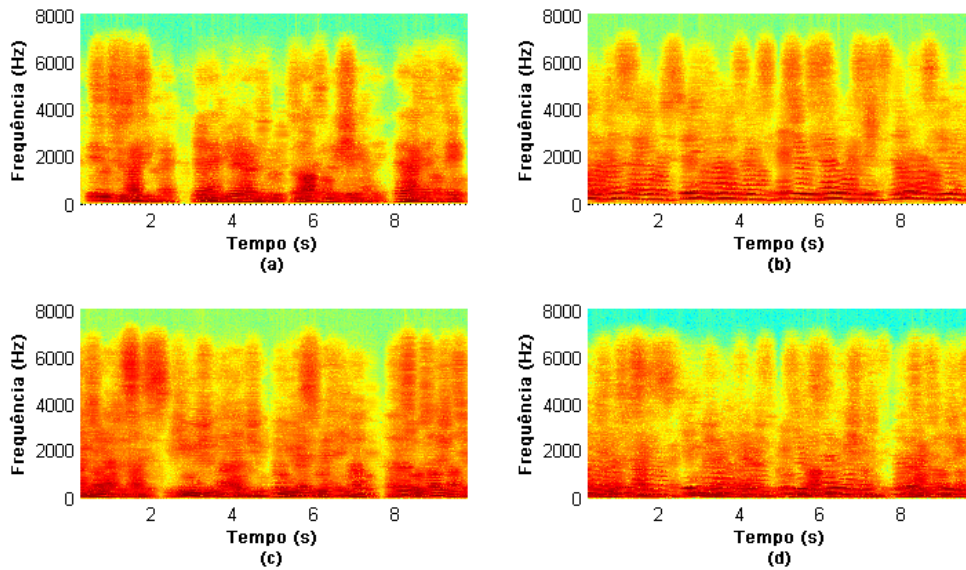


Figura A.1: Espectrogramas de 3 Sinais de Voz em Inglês (a)-(c) e da sua Soma (d).

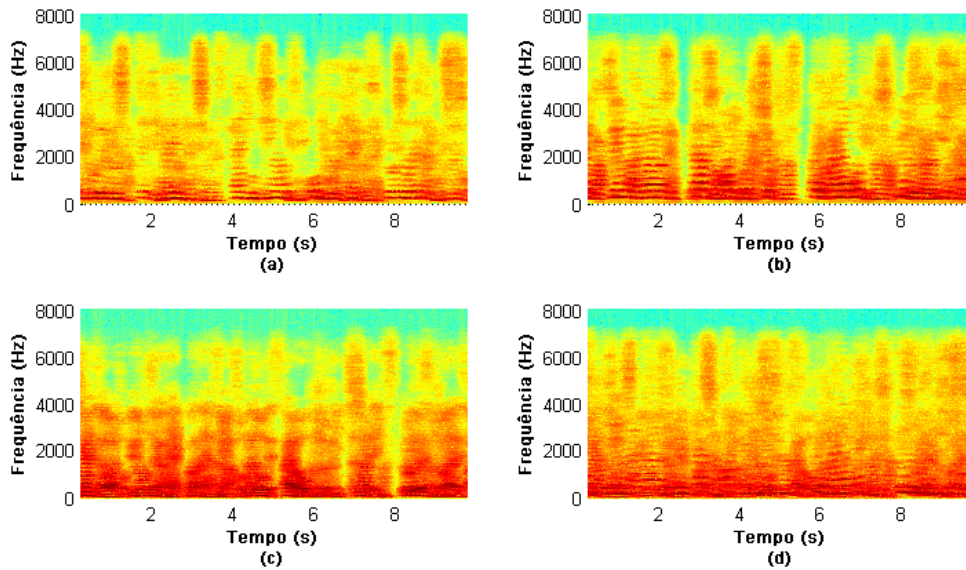


Figura A.2: Espectrogramas de 3 Sinais de Voz em Japonês (a)-(c) e da sua Soma (d).

culados pela Eq. (2.1) e normalizados pelo quarto momento (desvio padrão do sinal elevado à quarta potência). Desta tabela, observa-se que os sinais de flautas apresentam valores de curtose baixos, associados à distribuição subgaussiana, enquanto os demais sinais apresentam valores de curtose altos, associados à distribuição supergaussiana.

Na Tabela A.2 são apresentados os valores das correlações entre as fontes para cada tipo de sinal. Dessa tabela, observa-se que não há uma grande variação dos

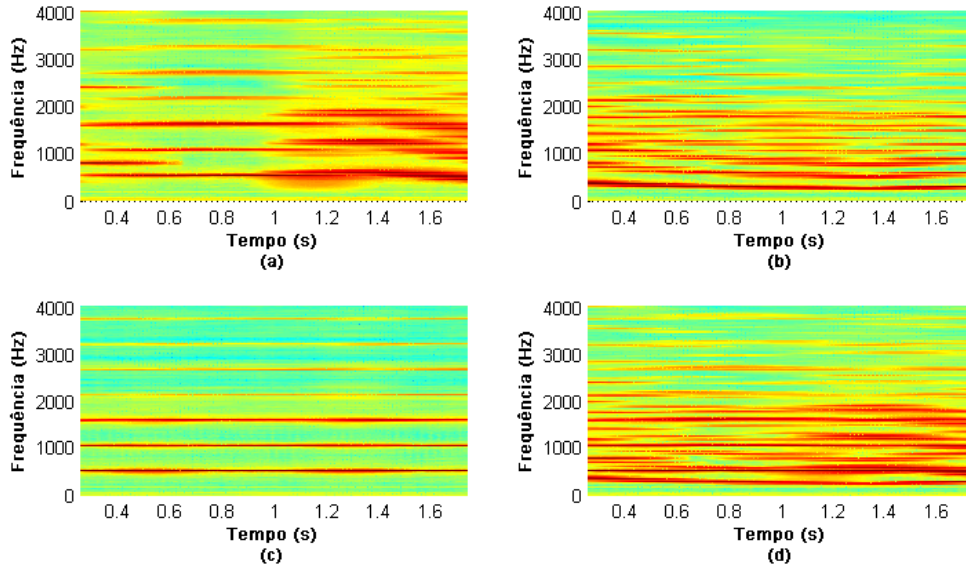


Figura A.3: Espectrogramas de 3 Sinais de Flautas (a)-(c) e da sua Soma (d).

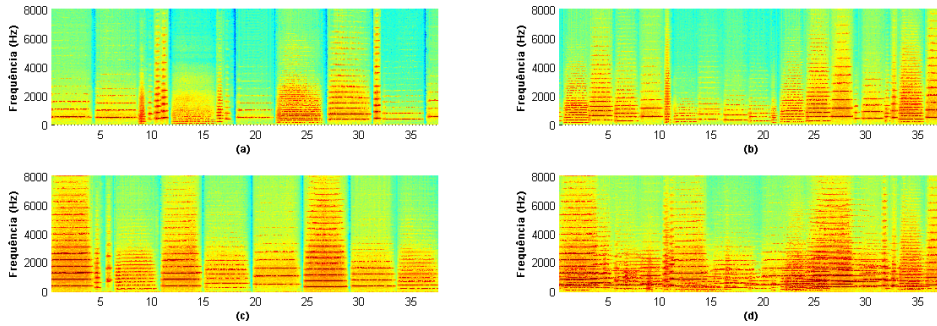


Figura A.4: Espectrogramas de 3 Sinais Formados por Sequências de Notas de Instrumentos Musicais (Metais) (a)-(c) e da sua Soma (d).

#	Tipo	$S_1(n)$	$S_2(n)$	$S_3(n)$	$X_1(n)$
01	Voz-Inglês	15,58	3,97	4,29	2,73
02	Voz-Japonês	2,36	5,63	6,13	1,67
03	Flautas	-0,25	-1,18	-0,86	-0,41
04	Notas-Metais	10,94	9,89	15,70	5,76

Tabela A.1: Curtose dos Sinais Empregados.

valores das correlações para os diferentes tipos de sinais.

Para avaliar os desempenhos dos algoritmos na presença de ruído nos sensores, adicionaram-se, aos sinais das fontes sonoras, ruídos de diferentes tipos e intensidades, descritos a seguir. Dois tipos de ruídos foram empregados: ruído branco gaussiano, gerado em computador, e ruído gerado por [118], correspondente à balbúrdia

#	Tipo	C_{12}	C_{13}	C_{23}
01	Voz-Inglês	$-3,73 \times 10^{-4}$	$2,51 \times 10^{-2}$	$8,80 \times 10^{-3}$
02	Voz-Japonês	$9,90 \times 10^{-3}$	$-8,40 \times 10^{-3}$	$6,00 \times 10^{-3}$
03	Áudio-Flautas	$2,60 \times 10^{-3}$	$4,18 \times 10^{-4}$	$-5,18 \times 10^{-4}$
04	Notas-Metals	$-3,29 \times 10^{-4}$	$-1,75 \times 10^{-4}$	$-3,27 \times 10^{-4}$

Tabela A.2: Correlação entre as Fontes dos Sinais Empregados.

de fundo (*babble*). Para o ruído gaussiano branco, foram gerados 4 sinais distintos de duração de 10 s, para taxa de amostragem equivalente a 16 kHz, correspondendo ao comprimento do maior sinal de áudio empregado nas simulações. Os mesmos sinais foram adicionados às misturas nas simulações com os diversos algoritmos, de modo a testá-los de forma idêntica. Para o ruído do tipo *babble*, foram gerados 4 sinais [118], sob o pressuposto de que o campo de som é difuso (esférico isotrópico). A taxa de amostragem empregada foi de 8 kHz, reamostrada para 16 kHz, sendo os sinais gerados com durações de 20 s. Nesta tese, não foram feitas simulações dos sinais formados por sequência de notas de instrumentos musicais com ruídos.

Sinais de ruído distintos foram adicionados ao sinal de cada sensor com diferentes SNRs. Os mesmos ruídos foram adicionados, alterando-se apenas as suas variâncias. A faixa de valores de SNR considerada foi de 5 a 25 dB, variando ao passo de 5 dB. Esses limites foram estabelecidos a partir de resultados experimentais com ruído branco, observando os resultados dos diversos algoritmos investigados nesta tese para o caso instantâneo.

Na Fig. A.5 são apresentados os espectrogramas de um sinal de cada tipo de ruído, enquanto o ruído branco apresenta uma distribuição uniforme da energia ao longo de frequência, o ruído *babble* apresenta maior concentração nas baixas frequências.

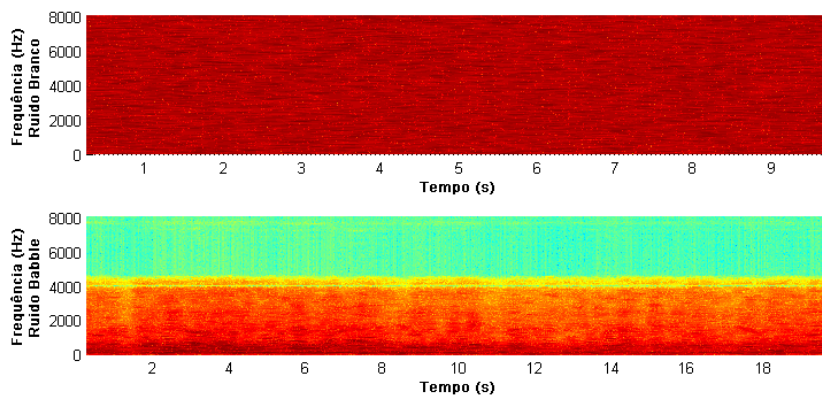


Figura A.5: Espectrogramas de um Sinal de Cada Tipo de Ruído: Branco (a) e *Babble* (b).

Apêndice B

Sistemas de Mistura

O cenário de testes escolhido para a avaliação dos algoritmos desenvolvidos para misturas instantâneas considerou a utilização de 3 microfones e a presença de 3 fontes simultaneamente ativas. Já para avaliação dos algoritmos desenvolvidos para misturas anecoicas, foi considerado o uso de 2 microfones e a presença de até 4 fontes simultâneas. A faixa de valores para ambos os casos empregados (instantâneo e anecoico) foi arbitrada, e, para obter os sistemas de misturas, foram geradas aleatoriamente (a partir de uma distribuição uniforme) 30 matrizes de misturas de dimensões 3×3 para o cenário de misturas instantâneas e 50 matrizes de misturas de dimensões 2×4 para o cenário de misturas anecoicas. Nessas matrizes, cada linha contém os coeficientes da mistura associados a um dos sensores, e cada coluna contém os coeficientes associados a cada uma das fontes. Nas misturas instantâneas, apenas os valores das atenuações são gerados, ao passo que no caso anecoico são também gerados valores de atrasos. A obtenção desses valores e a forma de avaliação das suas estimativas são detalhadas a seguir.

B.1 Atenuações

Para o cenário de misturas instantâneas, os valores limites dos coeficientes da matriz de misturas foram arbitrados na faixa $[-3, 0, 3, 0]$ (distribuição uniforme). Nesse cenário, não se permitiu que coeficientes provenientes de fontes distintas chegassem ao mesmo sensor com diferenças de nível dinâmico inferiores a 0,1. Também não se permitiu que sinais provenientes de uma mesma fonte chegassem nos diversos sensores com uma diferença de nível dinâmico inferior em módulo a 0,05, ou seja, obrigou-se que os sensores tivessem um afastamento mínimo entre eles. A configuração idealizada foi a de que os sensores tenham um afastamento mínimo entre eles igual à metade do afastamento mínimo entre as fontes. Por fim, exigiu-se que o valor mínimo em módulo do determinante das matrizes de misturas geradas fosse maior ou igual a 0,2, para evitar possíveis problemas na obtenção das matrizes

inversas devido a erros numéricos ocasionados por matrizes mal-condicionadas.

Para os cenários com misturas anecoicas, os coeficientes de atenuação utilizados foram todos positivos, devido ao fato do algoritmo DUET e a implementação do algoritmo TIFROM_Anec (que foram utilizados neste trabalho para fins de comparação) serem restritos a este cenário. Na geração dessas matrizes limitou-se os valores dos coeficientes na faixa $[0,05, 5,00]$. No caso anecoico, um coeficiente de atenuação é acompanhado por um coeficiente de atraso, dessa forma, permitiu-se que os coeficientes de atenuação de uma dada matriz de misturas tivessem valores mais próximos, visto que os coeficientes de atraso seriam distintos, tornando os picos no espaço atenuação-atraso mais afastados. Dessa forma, exigiu-se que os coeficientes de atenuação tivessem um afastamento mínimo arbitrário igual a 0,01.

B.2 Atrasos

Nos testes com misturas anecoicas, os sistemas de mistura consideram diferentes atrasos entre as fontes e os sensores. As matrizes de atrasos geradas neste trabalho não possuem restrições em relação a valores negativos, sendo os coeficientes de atrasos obtidos aleatoriamente na faixa $[-4,0, 4,0]$ com distribuição uniforme. Em relação à diferença entre os coeficientes de atraso de uma dada matriz de mistura, arbitrou-se o valor mínimo de 0,01, sendo o valor máximo igual a 8. Considerando que a menor taxa de amostragem empregada foi de 8 kHz, o maior atraso possível entre duas fontes numa mesma mistura, correspondente a 8 amostras, é de 1 ms. Considerando a velocidade do som no nível do mar como aproximadamente 340 m/s, este atraso máximo corresponde a uma distância de 34 cm entre as fontes.

Na geração dos coeficientes do sistema de mistura, feita aleatoriamente dentro das faixas limites acima especificadas, não nos preocupamos em que houvesse uma coerência entre os valores dos coeficientes de atenuação e de atrasos em relação a uma dada configuração dos sensores e fontes, para que a avaliação de desempenho dos algoritmos não ficasse restrita a casos particulares. É possível que alguns conjuntos de matrizes de atenuação e atraso não representem um cenário real. No entanto, existe um certo grau de liberdade na configuração desses coeficientes, uma vez que eles são funções, das distâncias das fontes em relação aos sensores, das posições angulares das fontes em relação aos sensores, dos diagramas de ganhos distintos que os microfones podem apresentar, das potências iniciais das fontes, entre outros fatores.

B.3 Avaliação do Erro da Estimativa da Matriz de Mistura

A medida utilizada, para avaliação dos resultados obtidos pelos métodos de estimação da matriz de misturas, foi o erro médio quadrático dos coeficientes estimados em relação aos seus valores reais. No entanto, devido aos problemas de escalamento e de possíveis permutações entre as colunas das matrizes estimadas, uma comparação direta entre os coeficientes das matrizes não é possível, sem antes considerar esses efeitos. Outro fator importante que deve ser considerado é que, devido à diferença entre as normas das colunas da matriz de mistura, erros nas estimativas dos coeficientes de uma dada coluna podem ter uma influência maior ou menor na reconstrução da fonte, em relação aos erros de mesmo valor em estimativas de outra coluna.

Para corrigir as possíveis distorções acima, primeiramente, as colunas das matrizes de mistura real e estimada são divididas pelos respectivos coeficientes da primeira linha, de modo a equiparar soluções com diferenças de fatores de escala no nível dinâmico das fontes. Em seguida, as colunas das matrizes resultantes são normalizadas (para todas ficarem com normas euclidianas unitárias), de forma a reduzir o efeito ocasionado por erros de diferentes grandezas entre os coeficientes estimados para cada fonte e evitar que pequenos erros sejam desprezados independentemente de seus impacto.

Após serem feitas as correções descritas acima, os erros médios quadráticos entre a matriz de mistura original e as matrizes obtidas fazendo todas as permutações possíveis das colunas da matriz estimada são calculados. O menor valor é utilizado para avaliação dos métodos de obtenção do sistema de mistura.

Apêndice C

Lista dos Métodos Elaborados nesta Tese

Este apêndice foi elaborado visando resumir todas as técnicas propostas nesta tese, a fim de facilitar a consulta das suas principais características.

- SCAm - Algoritmo de estimação dos coeficientes da matriz de mistura, desenvolvido para o caso instantâneo, baseado no princípio da esparsidade. O algoritmo promove um desacoplamento dos sensores e uma busca dos coeficientes da matriz de mistura a partir da detecção dos picos nos histogramas de DOA no domínio da transformada.
- SCAc - Algoritmo de estimação dos coeficientes da matriz de mistura, desenvolvido para o caso instantâneo, baseado no princípio da esparsidade. O algoritmo recorre ao acoplamento entre sensores, combinado com o emprego de um classificador (vizinho mais próximo) para formar agrupamentos associados a cada fonte. A busca dos coeficientes da matriz de mistura é feita a partir da detecção dos picos nos histogramas de DOA no domínio da transformada.
- SCAz - Algoritmo de estimação dos coeficientes da matriz de mistura, desenvolvido para o caso instantâneo, baseado no princípio da esparsidade. O algoritmo efetua um desacoplamento dos sensores, assim como uma busca dos coeficientes da matriz de mistura a partir da média da razão das misturas no domínio da transformada nas regiões de baixa variância (SSZ).
- IMUE - Algoritmo de estimação dos coeficientes da matriz de mistura, desenvolvido para o caso anecoico, iterativo, baseado no princípio da esparsidade. O algoritmo busca os coeficientes de atenuação a partir da média do módulo das razões das estimativas das fontes (reconstruídas via mascaramento *HARD*). Atua nas regiões SSZ no domínio da transformada, e estima os coeficientes

de atraso por meio da correlação das estimativas das fontes, efetuadas a partir de misturas distintas com os coeficientes de atenuação estimados na etapa anterior.

- STUE - Algoritmo de estimação dos coeficientes da matriz de mistura, não iterativo, desenvolvido para os casos instantâneo e anecoico, baseado no princípio da esparsidade. O algoritmo utiliza o princípio do desacoplamento dos sensores, e busca os coeficientes da matriz de mistura a partir da média da razão das estimativas das fontes (reconstruídas via mascaramento) no domínio da transformada (nas regiões SSZ). No caso anecoico, emprega as equações usadas pelo IMUE nas estimativas das fontes, para obter os coeficientes.
- ATIFROM - Algoritmo de estimação dos coeficientes da matriz de mistura, não iterativo, desenvolvido para o caso anecoico, baseado no princípio da esparsidade. O algoritmo emprega uma busca dos coeficientes de atenuação a partir da média do módulo da razão das misturas (no domínio da transformada, nas regiões SSZ), e busca os coeficientes de atraso a partir do *Lag* da correlação das estimativas das fontes (reconstruídas a partir de misturas distintas e dos coeficientes de atenuação já estimados).
- DEFL_f - Método de reconstrução de fontes elaborado para os casos instantâneo e anecoico, similar a um mascaramento *SOFT*, onde cada fonte é deflacionada das misturas, em sequência (após a reconstrução da mistura deflacionada no domínio do tempo), por meio da determinação da fonte deflacionante (correspondente ao agrupamento no domínio T-F com menor variância).
- TSOE - Método de reconstrução de fontes elaborado para o caso anecoico, que emprega subtração temporal a partir de uma estimativa fornecida ou estabelecida (reconstruída via mascaramento binário bidimensional a partir do agrupamento associado à fonte contendo a menor variância).
- TSAE - Método de reconstrução de fontes elaborado para o caso anecoico, que emprega subtração temporal a partir de todas as estimativas de fontes (reconstruídas via mascaramento binário bidimensional).
- MaskCorr - Método de reconstrução de fontes elaborado para os casos instantâneo e anecoico, que se utiliza da mistura reconstruída (a partir das estimativas das fontes obtidas via mascaramento) para estimar o erro destas estimativas das fontes, corrigindo-as iterativamente, mascarando o erro (entre a mistura real e a mistura reconstruída) entre as estimativas das fontes.
- CorrMin - Método de reconstrução de fontes elaborado para o caso instantâneo que estabelece que as duas fontes ativas num determinado ponto no domínio T-

F são obtidas ao se minimizar, para todas possíveis soluções dos pares de fontes ativas, o módulo da soma do produto da parte real com a parte imaginária das respectivas fontes ativas.

- Solução Aproximada - Método de reconstrução de fontes elaborado para o caso anecoico determinado, com $n = 2$, representando uma aproximação algébrica, sendo válida somente se os coeficientes de atenuação forem distintos e os coeficientes de atraso forem próximos.