

UNIVERSIDADE FEDERAL DE ITAJUBÁ

Valéria Cristina Maria Nascimento Leite

**SEPARAÇÃO CEGA DE SINAIS:
ANÁLISE COMPARATIVA ENTRE ALGORITMOS**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica como requisito parcial à obtenção do título de *Mestre em Engenharia Elétrica*.

Orientador: Prof. Luiz Eduardo Borges da Silva, PhD.

Itajubá, junho de 2004

Ficha Catalográfica elaborada pela biblioteca Mauá
Bibliotecária Cristiane N. C. Carpinteiro – CRB6 / 1702

L 533 s

Leite, Valéria Cristina Maria Nascimento.

Separação Cega de Sinais: Análise Comparativa entre Algoritmos /
por Valéria Cristina Maria Nascimento Leite. – Itajubá , MG : [s.n.],
2004.

182p. il.

Orientador: Prof. Dr.Luiz Eduardo Borges da Silva
Dissertação (mestrado) – Universidade Federal de Itajubá.

1. Separação cega de sinais 2. Análise de componentes independen-
tes 3. Processamento cego de sinais 4. Filtragem adaptativa. I. Silva,
Luiz Eduardo Borges. orient. II. Universidade Federal de Itajubá.
III. Título

CDU 621.391: 004.021

UNIVERSIDADE FEDERAL DE ITAJUBÁ

Valéria Cristina Maria Nascimento Leite

**SEPARAÇÃO CEGA DE SINAIS:
ANÁLISE COMPARATIVA ENTRE
ALGORITMOS**

Dissertação aprovada por banca examinadora em IX de junho de 2004, conferindo a autora o título de *Mestre em Engenharia Elétrica*.

Banca Examinadora:

Prof. Luiz Eduardo Borges da Silva, Ph.D. (Orientador)

Prof. Germano Lambert Torres, Ph.D.

Cynthia Cristina Martins Junqueira, Dra.

Itajubá, junho de 2004

Ao amigo Walter Pereira Junior[†] ...

Carinho,...

Saudade,...

E esta dissertação .

AGRADECIMENTOS

Ao meu orientador professor Luiz Eduardo Borges da Silva, por sua amizade, apoio e valiosa orientação ao longo desta dissertação.

Aos professores Allan Kardec Barros, Andrej Cichocki, Seungjin Choi e Jose Juan Fuentes pelas valiosas contribuições.

Ao Instituto de Aeronáutica e Espaço (IAE) do Centro Técnico Aeroespacial (CTA) pela confiança e suporte decisivos para o desenvolvimento de trabalhos científicos.

Aos colegas da Divisão de Eletrônica do IAE, em especial aos membros da Comissão de Recursos Humanos, ao Tecn. Sergio Costa, Eng. Francisco Visconti, Eng. Cynthia, Tec. Walter pela confiança, incentivo e apoio indispensáveis para a conclusão desta dissertação.

Ao amigo Douglas Barzon pelo inestimável apoio e pela correção ortográfica.

Ao amigo Alison de Oliveira Moraes pelo incentivo ímpar e pelas discussões técnicas que muito contribuíram para realização desta dissertação.

Ao meu querido Manoel, pelo companheirismo, pelo carinho, pelo estímulo e principalmente pela paciência.

Um agradecimento especial a toda minha família.

A todos que contribuíram para realização desta etapa de minha vida, que foram e são muito importantes para mim, meu sincero muito obrigada.

Á Deus toda glória e louvor.

SUMÁRIO

Dedicatória	iii
Agradecimentos	iv
Sumário	v
Resumo	viii
Abstract	ix
Lista de Figuras	x
Lista de Tabelas	xiii
Lista de Símbolos e Abreviaturas	xiv
1. INTRODUÇÃO	1
1.1. Objetivo	1
1.2. Motivação	1
1.3. Organização e Metodologia	5
2. PROCESSAMENTO CEGO DE SINAIS: PRINCIPAIS CONCEITOS E APLICAÇÕES	8
2.1. Introdução	8
2.2. Definição do Problema	9
2.3. Separação Cega de Fontes - BSS	12
2.3.1. BSS versus ICA	13
2.3.2. Técnicas para Separação Cega de Fontes	14
2.4. Análise de Componentes Independentes - ICA	15
2.5. Deconvolução Cega de Múltiplos Usuários	16
2.6. Extração Cega de Fontes – BSE	18
2.7. Principais Aplicações das Técnicas de Processamento Cego de Sinais	19
2.7.1. Processamento de Sinais Biomédicos	20
2.7.2. Sistemas Acústicos	21
2.7.3. Sistemas de Comunicação	22
2.7.4. Monitoramento das Condições de Máquinas	24
3. IDENTIFICAÇÃO CEGA DE FONTES BASEADA EM ESTATÍSTICAS DE SEGUNDA ORDEM	26
3.1. Introdução	26
3.2. Separação Cega Utilizando SOS: Considerações Iniciais	27
3.3. Pré-Processamento: Branqueamento ou Descorrelação	30
3.3.1. Relação entre Separação Cega de Fontes e Branqueamento	33
3.4. Separação Cega Baseada em Diagonalizações Simultâneas	34
3.4.1. Princípios Básicos	34
3.4.2. AMUSE – Algoritmo para Extração de Múltiplos Sinais	36
3.4.3. Considerações Adicionais	38
3.5. Separação Cega Baseada em Diagonalização Conjunta	39
3.5.1. Diagonalização Conjunta Aproximada	39
3.5.2. Algoritmos Baseados em Diagonalização Conjunta	41
3.6. Comparação entre Algoritmos de JAD e EVD	44
4. ANÁLISE DE COMPONENTES INDEPENDENTES	45
4.1. Introdução	45
4.1.1. Revisão Histórica	46
4.2. Modelo e Definição	49
4.3. Restrições e Ambigüidades para ICA	51
4.3.1. Independência Estatística e Descorrelação	51

4.3.2.	Variáveis Não Gaussianas	52
4.3.3.	Ambigüidades da ICA	54
4.4.	Técnicas de Pré-Processamento	55
4.4.1.	Centralização	56
4.4.2.	Branqueamento	56
4.4.3.	Pré-Processamento Adicional	58
4.5.	Princípios para Estimativa de Componentes Independentes	59
4.5.1.	Maximização da Não Gaussianidade	60
4.5.1.1.	Medida de Não Gaussianidade por <i>Kurtosis</i>	60
4.5.1.2.	Negentropia	62
4.5.1.3.	Aproximação da Negentropia	63
4.5.1.4.	Divergência de Kullback-Leibler	64
4.5.2.	Estimativa de Máxima Verossimilhança	66
4.5.3.	Princípio de INFOMAX	67
4.5.4.	Informação Mútua	68
4.5.5.	Relação entre os Princípios para Estimativa de Componentes Independentes	70
4.6.	Principais Algoritmos para ICA	71
4.6.1.	Algoritmo de Héroult-Jutten	71
4.6.2.	Algoritmo de Descorrelação Não Linear	73
4.6.2.1.	Algoritmo EASI	74
4.6.3.	Algoritmo INFOMAX	75
4.6.4.	Algoritmos Baseados em Gradiente Natural	79
4.6.5.	Algoritmo Rápido de Ponto Fixo (FastICA)	80
4.6.6.	Algoritmos Baseados em Estatísticas de Ordem Superior	83
4.6.6.1.	Algoritmo de Comon	85
4.6.7.	PCA Não Linear	86
4.6.8.	Algoritmos com Regras Neurais	87
4.6.9.	Escolha do Algoritmo	88
4.7.	ICA na Presença de Ruído	88
4.8.	Considerações Finais	89
5.	AMUSE	91
5.1.	Introdução	91
5.2.	Modelo e Considerações Gerais	91
5.2.1.	Espaço de Identificação e Relação de Preservação da Forma de Onda	92
5.2.2.	Identificidade	94
5.3.	Identificação de Fontes Descorrelacionadas	94
5.4.	Descrição do AMUSE	97
5.4.1.	Princípio Fundamental	97
5.4.2.	Passos para Implementação do AMUSE	98
5.4.3.	Decomposição em Valores Singulares	100
5.5.	Descrição do AMUSE	102
6.	ALGORITMO JADE	103
6.1.	Introdução	103
6.2.	Notações e Considerações Iniciais	104
6.2.1.	Cumulantes: Definição e Propriedades	105
6.2.2.	Matriz de Cumulantes	106
6.3.	Separação Cega Utilizando Estruturas Algébricas	108
6.3.1.	Descorrelação Espacial	108
6.3.2.	Técnicas Mal Condicionadas	110
6.3.3.	Solucionando os Problemas – Técnica Bem Condicionada	111
6.4.	Algoritmos Jacobianos	113

6.5.	JADE: Um Algoritmo Baseado em Estatística	115
6.6.	Considerações Finais	118
7.	ALGORITMO FlexICA	119
7.1.	Introdução	119
7.2.	Considerações Iniciais	119
7.3.	Algoritmos para ICA e Gradiente Natural	120
7.3.1.	NG com Restrição de Ortogonalidade	121
7.4.	Modelo Generalizado para Distribuição Gaussiana	122
7.4.1.	<i>Kurtosis</i> e Expoente Gaussiano	123
7.5.	O Algoritmo FlexICA	125
7.5.1.	Passos para Implementação do FlexICA	127
7.5.2.	Análise de Estabilidade	129
7.6.	Considerações Finais	131
8.	SIMULAÇÕES COMPUTACIONAIS	133
8.1.	Conclusão	133
8.1.1.	Índices de Desempenho	133
8.2.	Simulações com Sinais de Teste	136
8.2.1.	Separação para um Problema Bem Condicionado	136
8.2.2.	Separação para um Problema Mal Condicionado	140
8.2.3.	Separação na Presença de Ruído	143
8.2.4.	Separação de Senóides com Frequências Múltiplas de 60Hz	146
8.2.5.	Separação de Fontes Esparsas	150
8.3.	Aplicação a Sinais de Telecomunicações	153
8.3.1.	Sinais com Modulação Analógica	154
8.3.1.1.	Sinais com Modulação em Frequência	157
8.3.2.	Sinais com Modulação Digital	161
8.3.2.1.	FlexICA e sinais complexos	163
8.4.	Aplicação a Sinais Biomédicos	164
8.4.1.	Sinais de Eletrocardiograma Fetal (FECG)	164
8.5.	Considerações Finais	168
9.	CONCLUSÕES E COMENTÁRIOS	170
9.1.	Recomendações para Futuros Trabalhos	173
	REFERÊNCIAS BIBLIOGRÁFICAS	175

RESUMO

A separação cega ou o problema da separação de sinais (fontes) consiste na recuperação de um conjunto de sinais ou fontes desconhecidos a partir de observações feitas por sensores das misturas destes sinais. Este é, sem dúvida, um problema de grande interesse dentro da área de processamento de sinais, uma vez que para ser solucionado é necessário que se cumpram um conjunto de hipóteses pouco restritivas. Assim sendo, as técnicas de separação cega de fontes encontram inúmeras aplicações: processamento de conjunto de dados; comunicações multiusuários; reconhecimento de voz e imagem; processamento de sinais biomédicos. Através das técnicas de separação, pode-se, portanto, recuperar uma ou todas as fontes com base apenas nas informações nas observações ou medidas feitas por um conjunto de sensores. O adjetivo cego foi incorporado para caracterizar a falta de informação inerente ao processo de separação. Para suprir esta falta de informação são levadas em consideração no processo de separação algumas propriedades sobre a natureza das fontes, da mistura e sobre o ruído adicionado ao processo. Assim, os algoritmos para separação cega procuram restaurar na saída do sistema de separação uma propriedade conhecida das fontes.

Uma das principais ferramentas utilizadas para solucionar o problema da separação cega tem sido a Análise de Componentes Independentes. É importante mencionar que os termos separação cega e Análise de Componente Independentes são freqüentemente confundidos ou trocados, embora eles se refiram a um modelo similar ou igual e sejam resolvidos com algoritmos similares ou iguais, sob a restrição de que as fontes originais são estatisticamente independentes. Entretanto, principalmente em problemas reais, os objetivos da Análise de Componentes Independentes e da separação cega são um pouco diferentes: o objetivo da separação cega é estimar os sinais originais mesmo se eles não forem completamente independentes; por sua vez, o objetivo da análise de componentes independentes é determinar uma transformação que assegure que os sinais estimados sejam tão independentes quanto possível. Deve-se notar ainda que os métodos para análise de componentes independentes utilizam, na maioria dos casos, estatísticas de ordem superior, enquanto que os métodos para separação cega são aptos a utilizar somente estatísticas de segunda ordem.

Com base nas considerações supracitadas, esta dissertação apresenta uma revisão do estado da arte e das principais técnicas que tratam o problema da separação cega, através da análise comparativa entre três algoritmos: AMUSE, JADE e FLEXICA. Para realizar a referida comparação, os algoritmos são aplicados a sinais de teste, sinais de comunicações e sinais biomédicos reais.

ABSTRACT

Nowadays the blind signal processing is one of the areas of greater highlight in the signal processing. The signal processing techniques do not make use of any training sequence nor any information on the mixture of the system to which the signals are subjected to; being the blind separation one of the main areas of the blind processing.

The blind separation or the blind signal (source) separation problem consists of retrieval a set of unknown signals or sources by the observations done by sensors of mixture of this signals. It's no shadow of doubt, a problem of great interest in the signal processing area, once to be solved it is necessary that a set of hypotheses a bit restrictive be carried out. Being that, the blind source separation techniques run across countless applications: data set processing; multi-users communications; voice and image recognition; biomedical signal processing. By means of separation techniques, one can, therefore, retrieve one or all the sources just basing on the information on observations or measurement done by the set of sensors. The blind adjective was incorporated to characterize the lack of information inherent in the separation process. To fulfill this lack of information some properties of the sources nature, mixture, and noise are taken into account for separation process. This way, the algorithms of blind separation try and restore at the exit of separation system one property known by the sources.

One of the main tools used to solve the blind separation problem has been the Independent Component Analysis. It is important to mention that the blind separation and the Independent Component Analysis terms are often mixed up or used like synonym, although they refer to a similar or equal pattern and are solved with similar or equal algorithms, under the restriction on the original sources be statistically independent. However, mainly in real problems, the independent component analysis and blind separation goals are a bit different: the blind separation goal is to estimate the original signals even if they are not completely independent; whereas, the independent component analysis goal is to determine one transformation that ensures the estimate signals are as independent as possible. One may yet observe that the independent component analysis methods use, in most cases, statistics of superior order, while the blind separation methods are apt to use just statistics of second order.

Based on the above-mentioned considerations, this dissertation presents one state-of-art review of the main techniques that deal with the separation problem; by means of comparison of three algorithms: AMUSE, JADE and FLEXICA that were compared through the application of them into test signals, telecommunication signals and a real world biomedical signal.

LISTA DE FIGURAS

Figura 2.1 - Problema do processamento cego de sinais	9
Figura 2.2 – Diagrama em blocos ilustrando o modelo utilizado no processamento cego de sinais	10
Figura 2.3 - Diagrama em blocos ilustrando o modelo utilizado na deconvolução cega de múltiplos usuários	18
Figura 2.4 – Problema do coquetel (<i>cocktail party</i>) e reconhecimento de voz	22
Figura 2.5 – Ambiente de comunicação digital	23
Figura 2.6 – BSS aplicada a sistemas de espectro espalhado	24
Figura 2.7 – Esquema de teste para detecção de falhas em bombas submersas utilizando ICA	25
Figura 3.1 – Diagrama em blocos para o problema da separação cega utilizando SOS	27
Figura 3.2 – Ilustração do branqueamento de dois sinais \mathbf{x}_1 e \mathbf{x}_2	30
Figura 4.1 – Diagrama em blocos do modelo de mistura de sinais para ICA	49
Figura 4.2 – Exemplo de distribuição gaussiana, super-gaussiana e sub-gaussiana	54
Figura 4.3 – Efeito do branqueamento: representação da f.d.p. para $n=2$	58
Figura 4.4 – Relação entre algoritmo, função custo e modelo para ICA	59
Figura 4.5 – Rede de Heráult-Jutten para duas entradas e duas saídas	72
Figura 4.6 – Rede de Bell-Sejnowski	76
Figura 4.7 – Função não linear e sua derivada	77
Figura 4.8 – Diagrama em blocos do FASTICA	82
Figura 6.1 – Modelo para ICA com branqueamento	109
Figura 7.1 – <i>Kurtosis</i> versus expoente gaussiano	124
Figura 7.2 – Funções densidade de probabilidade resultantes para diferentes valores de α e funções de ativação não lineares correspondentes	126
Figura 8.1 – Sinais originais para o experimento 8.2.1	137
Figura 8.2 – Sinais medidos pelos sensores para o experimento 8.2.1	137
Figura 8.3 – Sinais estimados com o AMUSE para o experimento 8.2.1	138
Figura 8.4 – Sinais estimados com o algoritmo JADE para o experimento 8.2.1	139
Figura 8.5 – Sinais estimados com o algoritmo FLEXICA para o experimento 8.2.1	138
Figura 8.6 – Índices de desempenho para o experimento 8.2.1	139
Figura 8.7 – Sinais medidos pelos sensores para o experimento 8.2.2	140

Figura 8.8 – Sinais estimados com o AMUSE para o experimento 8.2.2	141
Figura 8.9 – Sinais estimados com o algoritmo JADE para o experimento 8.2.2	141
Figura 8.10 – Sinais estimados com o algoritmo FLEXICA para o experimento 8.2.2	142
Figura 8.11 – Índices de desempenho para o experimento 8.2.2	142
Figura 8.12 – Sinais originais para o experimento 8.2.3	143
Figura 8.13 – Sinais medidos pelos sensores para o experimento 8.2.3	144
Figura 8.14 – Sinais estimados com o AMUSE para o experimento 8.2.3	144
Figura 8.15 – Sinais estimados com o algoritmo JADE para o experimento 8.2.3	145
Figura 8.16 – Sinais estimados com o algoritmo FLEXICA para o experimento 8.2.3	145
Figura 8.17 – Índices de desempenho para o experimento 8.2.3	146
Figura 8.18 – Sinais originais para o experimento 8.2.4	147
Figura 8.19 – Sinais medidos pelos sensores para o experimento 8.2.4	147
Figura 8.20 – Sinais estimados com o AMUSE para o experimento 8.2.4	148
Figura 8.21 – Sinais estimados com o algoritmo JADE para o experimento 8.2.4	148
Figura 8.22 – Sinais estimados com o algoritmo FLEXICA para o experimento 8.2.4	149
Figura 8.23 – Índices de desempenho para o experimento 8.2.4	150
Figura 8.24 – Sinais originais para o experimento 8.2.5	150
Figura 8.25 – Sinais medidos pelos sensores para o experimento 8.2.5	151
Figura 8.26 – Sinais estimados com o AMUSE para o experimento 8.2.5	151
Figura 8.27 – Sinais estimados com o algoritmo JADE para o experimento 8.2.5	152
Figura 8.28 – Sinais estimados com o algoritmo FLEXICA para o experimento 8.2.5	152
Figura 8.29 – Índices de desempenho para o experimento 8.2.5	153
Figura 8.30 – Sinais originais para o experimento 8.3.1	154
Figura 8.31 – Sinais medidos pelos sensores para o experimento 8.3.1	155
Figura 8.32 – Sinais estimados com o AMUSE para o experimento 8.3.1	155
Figura 8.33 – Sinais estimados com o algoritmo JADE para o experimento 8.3.1	156
Figura 8.34 – Sinais estimados com o algoritmo FLEXICA para o experimento 8.3.1	156
Figura 8.35 – Índices de desempenho para o experimento 8.3.1	157
Figura 8.36 – Sinais originais para o experimento 8.3.1.1	158
Figura 8.37 – Sinais medidos pelos sensores para o experimento 8.3.1.1	158
Figura 8.38 – Sinais estimados com o AMUSE para o experimento 8.3.1.1	159
Figura 8.39 – Sinais estimados com o algoritmo JADE para o experimento 8.3.1.1	159
Figura 8.40 – Sinais estimados com o algoritmo FLEXICA para o experimento 8.3.1.1	160
Figura 8.41 – Índices de desempenho para o experimento 8.3.1.1	160

Figura 8.42 – Constelações dos sinais 4QAM originais	161
Figura 8.43 – Sinais medidos pelos sensores	162
Figura 8.44 – Sinais estimados com o algoritmo JADE para o experimento 8.3.2	162
Figura 8.45 – Sinais estimados com o algoritmo FLEXICA para o experimento 8.3.2	163
Figura 8.46 – FECG – disposição dos sensores	165
Figura 8.47 – Sinais de FECG medidos pelos sensores	166
Figura 8.48 – Sinais estimados com o algoritmo JADE para o experimento 8.4.1	167
Figura 8.49 – Sinais estimados com o algoritmo FLEXICA para o experimento 8.4.1	167
Figura 8.50 – Sinais estimados com o algoritmo FLEXICA - detalhe	168

LISTA DE TABELAS

Tabela 2.1 – ICA aplicada a MEG	20
Tabela 8.1 – Índices de desempenho para o experimento 8.2.1	139
Tabela 8.2 – Índices de desempenho para o experimento 8.2.2	142
Tabela 8.3 – Índices de desempenho para o experimento 8.2.3	145
Tabela 8.4 – Índices de desempenho para o experimento 8.2.4	149
Tabela 8.5 – Índices de desempenho para o experimento 8.2.5	152
Tabela 8.6 – Índices de desempenho para o experimento 8.3.1	156
Tabela 8.7 – Índices de desempenho para o experimento 8.3.1.1	160

LISTA DE SÍMBOLOS E ABREVIATURAS

A	Matriz de mistura
a_{ij}	ij-ésimo elemento da matriz A
b_i	i-ésimo elemento do vetor b
$\det(\mathbf{A})$	determinante da matriz A
$\text{diag}(d_1, d_2, \dots, d_n)$	matriz diagonal com elementos d_1, d_2, \dots, d_n na diagonal principal
$\exp(\cdot)$	exponencial
$E[\cdot]$	esperança
$\mathbf{f}(\mathbf{y}) = [f_1(y_1), \dots, f_n(y_n)]^T$	transformação não linear do vetor y
$\mathbf{g}(\mathbf{y}) = [g_1(y_1), \dots, g_n(y_n)]^T$	transformação não linear do vetor y
\mathbf{A}^{-1}	inversa de uma matriz não singular A
\mathbf{A}^+	pseudo-inversa da matriz A
$\mathcal{H}(\cdot)$	entropia
I	matriz identidade
$\text{KL}(\mathbf{y} \parallel \mathbf{s}) = \left\langle \log \frac{p_y(\mathbf{y})}{p_s(\mathbf{y})} \right\rangle$	divergência de Kullback-Leibler, entropia relativa
$\text{KL}(p_y(\mathbf{y}) \parallel p_s(\mathbf{s}))$	divergência de Kullback-Leibler, entropia relativa
$\log(\cdot)$	logaritmo
$\det(\cdot)$	determinante
n	número de entradas e saídas
m	número de sensores
k	amostra ou número de iterações
$p(\mathbf{x})$ ou $p_x(\mathbf{x})$	função densidade de probabilidade (f.d.p.) de x
P	matriz de permutação
Q	matriz de branqueamento
\mathbf{R}_x ou \mathbf{R}_{xx}	matriz de covariância de x
\mathbf{R}_{xy}	matriz de covariância de x e y
R^M	parâmetro real M-dimensional
s	vetor dos sinais fontes ou originais
$s(t)$	sinal contínuo
$\mathbf{s}(k) = [s_1(k), \dots, s_n(k)]^T$	vetor dos sinais fontes ou originais na k -ésima amostra
$\text{tr}(\mathbf{A})$	traço da matriz A
$\mathbf{W} = w_{ij}$	matriz de separação
$\mathbf{x}(k)$	dados discretos observados pelos sensores ou dados misturados
$ x $	valor absoluto de x
$\ \mathbf{x}\ $	comprimento do vetor x
δ_{ij}	delta de Kronecker
μ ou η	passo de adaptação ou taxa de aprendizagem
λ_{\max}	autovalor máximo da matriz de correlação R
λ_{\min}	autovalor mínimo da matriz de correlação R
$\mathbf{\Lambda}$ ou D	matriz diagonal
$\text{kurt}(y)$	<i>kurtosis</i> da variável aleatória y
$\text{cum}(y)$	cumulante da variável aleatória y
$\varphi(\cdot)$	função não linear de ativação
σ^2	variância

\hat{s}	valor estimado de s
Δw_i	variação aplicada a w_i
∇	gradiente
$\Sigma(\cdot)$ ou $sum(\cdot)$	somatória
$(\cdot)^+$	matriz pseudo inversa
$(\cdot)^T$	matriz transposta
$(\cdot)^*$	valor ótimo
$\hat{\cdot}$	estimador
i.i.d.	distribuição independente e idêntica
c.d.f.	função densidade cumulativa
f.d.p.	função densidade de probabilidade
BSE	Extração Cega de Fontes ou Sinais (<i>Blind Source (Signal) Extraction</i>)
BPSK	<i>Binary Phase Shift Keying</i>
BSS	Separação Cega de Fontes (<i>Blind Source (Signal) Separation</i>)
BSD	Deconvolução Cega de Sinais (<i>Blind Signal Deconvolution</i>)
BSP	Processamento Cego de Sinais (<i>Blind Signal Processing</i>)
EEG	Eletro encefalograma
EVD	Decomposição em Autovalores (<i>Eign Value Decomposition</i>)
FECG	Eletrocardiograma fetal
GPS	Sistema de Posicionamento Global
HOS	Estatística de Ordem Superior (<i>High Order Statistics</i>)
ICA	Análise de Componentes Independentes (<i>Independent Component Analysis</i>)
LTI	Linear e Invariante no Tempo
MBD	Deconvolução Cega de Múltiplos Canais ou Usuários (<i>Multichannel Blind Deconvolution</i>)
MBE	Equalização Cega de Múltiplos Canais (<i>Multichannel Blind Equalization</i>)
MEG	Magneto encefalograma
MECG	Eletrocardiograma materno
MI	Informação Mútua (<i>Mutual Information</i>)
MLE	Estimativa de Máxima Verossimilhança (<i>Maximum Likelihood Estimation</i>)
NG	Gradiente Natural (<i>Natural Gradient</i>)
PCA	Análise de Componentes Principais (<i>Principal Component Analysis</i>)
QAM	Modulação em Quadratura (<i>Quadrature Amplitude Modulation</i>)
SNR	Relação Sinal Ruído
SOS	Estatística de Segunda Ordem (<i>Second Order Statistics</i>)
SVD	Decomposição em Valores Singulares (<i>Singular Value Decomposition</i>)

Capítulo 1 – Introdução

Este capítulo apresenta uma explanação sobre os tópicos que serão discutidos ao longo desta dissertação e assinala sua relevância e seus objetivos. No final do capítulo é apresentada a forma como esta dissertação foi organizada.

1.1 – Objetivo

O principal objetivo desta dissertação é realizar uma análise comparativa entre algoritmos, visando abordar um dos mais novos problemas na área de processamento de sinais, mais especificamente inserido na área de filtragem adaptativa: a separação cega de sinais ou fontes (BSS - *Blind Signal Separation*) (Cichocki, 2002; Kofidis, 2001; Sánchez, 2002; Hyvärinen, 2001; Mansur, 2000; Lee, 2000; Alvarez, 1999; Cardoso, 1998; Chan, 1997; Karhunen, 1996; Cao, 1996; Pope, 1996; Smaragdis, 1995).

Os estudos a cerca do problema da separação cega de fontes têm pouco mais de 15 anos e são motivados por situações práticas que envolvem multi-fontes e multi-canais e tem por objetivo separar, estimar ou recuperar as fontes ou sinais originais a partir unicamente de medidas obtidas por um conjunto de sensores.

Para realizar a citada análise comparativa, foram selecionados três algoritmos dentre os inúmeros algoritmos utilizados para solucionar o referido problema: AMUSE (Cichocki, 2002; Tong, 1991; Tong, 1990), JADE (Cardoso, 1993; Cardoso, 1999) e FlexICA (Choi, 2000c; Choi, 1998). Estes algoritmos foram analisados e comparados para diferentes situações. Os dois primeiros algoritmos foram selecionados por serem marcos no estudo da separação cega, já o algoritmo FlexICA foi selecionado por sua abordagem neural e por sua habilidade em lidar com sinais sub e super-gaussianos graças à utilização de uma não linearidade adaptativa.

1.2 – Motivação

Em processamento de sinais e áreas correlatas, são encontradas frequentemente aplicações envolvendo multi-canais e/ou multi-usuários; sendo que tais aplicações são vistas

como um desafio para pesquisadores. Os dados a serem processados são obtidos de diferentes formas de acordo com a aplicação. Por exemplo, os dados podem ser obtidos por meio de um conjunto de antenas, vários microfones espalhados em uma sala, sensores biomédicos, entre outros. Nesta dissertação, independente do sistema utilizado para captar, medir ou observar os dados de interesse ele será chamado de sensor. Cada sensor adquire um conjunto de dados diferentes, uma vez que cada sensor apresenta propriedades diferentes (como, por exemplo, a localização). Mas, embora diferentes, as medidas dos sensores podem ser relacionadas, visto que eles estão medindo o mesmo evento ou o mesmo conjunto de sinais originais. Em muitos casos práticos, diversas fontes estão presentes ao mesmo tempo e muitas vezes as características ou propriedades dos sinais originais e/ou como eles influem na medida de cada sensor são desconhecidas. Em outras palavras, pode-se dizer que, nestes casos, nem os sinais originais, nem o sistema físico ou o meio no qual estes sinais estão inseridos é conhecido; têm-se apenas as informações presentes nas saídas dos sensores. Neste contexto, o processamento dos sinais medidos pelos sensores recebe o adjetivo cego.

Na última década, o processamento cego de sinais, com seu sólido embasamento teórico e suas inúmeras aplicações em potencial, vem se tornando uma das áreas mais promissoras de pesquisa dentro do universo do processamento de sinais. A principal diferença entre as técnicas utilizadas pelo processamento cego e as convencionais é que elas não necessitam de dados ou seqüência de treinamento e desconsideram o conhecimento sobre os sinais originais e sobre os parâmetros de convolução, filtragem ou sistema de mistura ao qual estes sinais foram submetidos (Cichocki, 2002, Hyvärinen, 1999a; Comon, 1994).

O processamento cego inclui duas grandes áreas de pesquisa: a separação cega de sinais – que constitui o principal objetivo desta dissertação – e a equalização ou deconvolução cega de multi-canal (Cichocki, 2002; Haykin, 1998; Neves, 2001; Godard, 1980).

A separação cega ou problema da separação cega consiste na recuperação ou identificação de um conjunto de sinais não observáveis ou desconhecidos a partir de observações de suas misturas instantâneas (Alvarez, 1999). Este problema tem despertado grande interesse na comunidade científica principalmente por suas inúmeras aplicações em diferentes áreas, entre as quais estão: sistemas de comunicação multi-usuários; processamento de conjunto de sinais (*array signal processing*); cancelamento de ruído; identificação de falhas em sistemas elétricos; análise de sinais biomédicos; processamento e recuperação de imagens, entre outros (Cardoso, 2003; Ypma, 2002; Ypma, 2001; Fuentes, 2001; Hyvärinen, 2001; Belouchrani, 2000; Vigário, 2000; Hyvärinen, 2000; Barros, 1999; Castedo, 1997; Makeig, 1996; Torkkola, 1996; Bell, 1995; Cardoso, 1993).

O problema da separação cega foi inicialmente proposto na seguinte situação: imagine uma sala onde um grupo de pessoas conversa animadamente, como acontece, por exemplo, em uma festa ou coquetel. Além das diversas vozes, podem estar presentes outras fontes de ruído, como música de fundo, carros passando do lado de fora, telefones tocando, etc. Cada pessoa presente na festa escuta uma mistura de sons vindos de várias direções, sendo que esta mistura é composta por diferentes sinais de voz, música e diversos sinais de ruído. O problema de focar a atenção ou audição em um locutor específico dentro deste universo de sons é conhecido como problema do coquetel (*cocktail party problem*) (Haykin, 2003; Chan, 1997; Torkkola, 1999). Este problema consiste, então, em separar sinais de voz com diferentes características a partir da mistura destes sinais, sendo que, possivelmente foi adicionado ruído à mistura. Os sinais ou fontes originais não são conhecidos a priori; escuta-se apenas a combinação ou a mistura destes sinais. Além disso, a forma como os sinais são misturados também não é conhecida.

O problema descrito acima é uma excelente analogia de muitas outras situações que necessitam da separação ou recuperação de sinais que foram misturados, considerando que nem os sinais originais nem o sistema de mistura ao qual estes sinais foram submetidos são conhecidos. Por causa desta analogia, o problema do coquetel motivou e serviu de base para implementação de outras aplicações ligadas ao problema da separação cega.

Para solucionar o problema da separação cega é necessário suprir a falta de informações com relação aos sinais originais e ao sistema de mistura. Para suprir esta falta de informação, são feitas algumas considerações pouco restritivas sobre as propriedades estatísticas e/ou estruturais dos sinais originais e do sistema de mistura. Por exemplo, em um sistema de comunicação digital multi-usuário, como o GPS (Sistema de Posicionamento Global), pode-se considerar que os sinais originais são mutuamente independentes; independe e identicamente distribuídos (i.i.d.), utilizam modulação BPSK, portanto são sub-gaussianos e que o canal ou meio de transmissão é linear, invariante no tempo, embora desconhecido. Nesta dissertação, o sistema de mistura será considerado linear e invariante no tempo (LTI). Outra consideração feita nesta dissertação é que o número de fontes originalmente transmitidas é igual ao número de sensores, de tal forma que a matriz de mistura é quadrada.

Com relação às fontes, as considerações recaem sobre suas propriedades estatísticas. Assim, pode-se considerar que as fontes são descorrelacionadas – como será descrito no Capítulo 3 – ou estatisticamente independentes – como será abordado no Capítulo 4. Independente da consideração feita em relação às propriedades estatísticas das fontes, a

separação somente será bem sucedida se as fontes recuperadas ou separadas satisfizerem a consideração ou o critério pré-estabelecido.

A consideração de que as fontes são descorrelacionadas leva a algoritmos simples, rápidos e eficientes, que utilizam apenas estatísticas de segunda ordem (Cichocki, 2002; Choi, 2002; Choi, 2001; Belouchrani, 1993; Tong, 1991). Estes algoritmos se baseiam ainda na restrição de que as fontes não apresentam o mesmo espectro de potência. Estes algoritmos também não são aplicáveis quando as fontes são consideradas brancas ou mesmo i.i.d. - o que é muito comum em sistemas de comunicação digital (Kofidis, 2001). Nestes algoritmos as fontes não têm que ser independentes, mas apenas o fato de que são descorrelacionadas é levado em consideração. Isto sugere que utilizando apenas estatísticas de segunda ordem não está se explorando toda a informação contida nas fontes. Por outro lado, estes algoritmos são especialmente úteis quando se trabalha com fontes gaussianas, uma vez que estas são completamente descritas por meio de estatísticas de segunda ordem, ou seja, média e covariância. Além disso, para fontes gaussianas, os conceitos de descorrelação e independência coincidem (Papoulis, 1991). Entretanto, a maioria das fontes encontradas em aplicações práticas, por exemplo, áudio, dados, imagem, são sinais não gaussianos. Nestes casos, é necessário utilizar uma transformação que extraia mais informações dos sinais do que as estatísticas de segunda ordem.

A consideração de que as fontes são estatisticamente independentes, por sua vez, é muito pouco restritiva, podendo ser aplicada a um grande número de casos práticos, como por exemplo, sinais de áudio, dados, imagens, sinais de telecomunicações e sinais biomédicos. Dizer que as fontes são estatisticamente independentes implica que conhecida uma fonte, nada pode ser dito ou estimado a respeito das demais fontes que compõem o sistema. A independência estatística leva a utilização de um método estatístico denominado Análise de Componentes Independentes (ICA – *Independent Component Analysis*), que surgiu como um novo paradigma na área de processamento cego e análise de dados em geral e vem sendo utilizada como a principal ferramenta para solucionar o problema da separação cega (Karvanen, 2002; Hyvärinen, 2001; Cardoso, 1998; Comon, 1994).

Em termos matemáticos, a ICA consiste em encontrar uma transformação linear que minimize a dependência estatística de um determinado conjunto de dados. Neste sentido, e em se tratando de misturas lineares, a solução do problema da ICA leva à solução do problema da separação cega. Devido a esta relação muitas vezes os termos separação cega de fontes e análise de componentes independentes são usados como sinônimos, embora exista uma diferença conceitual entre eles. O objetivo da separação cega é recuperar uma ou mais fontes

que foram submetidas a algum tipo de mistura, a partir unicamente das medidas ou observações feitas por um conjunto de sensores; já a Análise de Componentes Independentes tem por objetivo encontrar, dentro de um determinado conjunto de dados, componentes que sejam tão estatisticamente independentes quanto possível.

Com base nas afirmações supra-citadas, pode-se dizer que o problema da separação cega não só parece, mas é um problema de difícil solução. Nos últimos anos, porém, vem sendo realizado um grande progresso nas pesquisas voltadas à separação cega e um grande número de métodos vem sendo desenvolvidos, principalmente com base na consideração de que as fontes são estaticamente independentes (Hyvärinen,2000; Lee, 2000; Barros, 1999; Alvarez, 1999, Amari, 1996; Cardoso,1996a; Bell, 1995; Delfosse, 1995; Comon, 1994). O grande número de métodos disponíveis para a solução do problema da separação cega faz com que este seja um tópico vasto e, portanto, impossível de ser abordado em um único trabalho. Por este motivo, foram selecionados alguns tópicos para serem discutidos nesta dissertação, de tal forma que ela pudesse despertar interesse e servir de base para o desenvolvimento de futuros trabalhos na área de separação cega de fontes. Feita a seleção de tópicos mais relevantes, a dissertação foi organizada conforme descrito na seção seguinte.

1.3 - Organização e metodologia

Como apresentado na seção anterior, a teoria desenvolvida acerca da separação cega de sinais ou fontes é extensa e suas aplicações inúmeras, sendo praticamente impossível abordá-las como um todo. Por este motivo, foi realizada uma seleção dos tópicos mais relevantes para a solução do problema da separação cega, inicialmente com base na consideração de que as fontes são decorrelacionadas e depois considerando fontes estatisticamente independentes.

A partir da seleção dos tópicos mais importantes para a solução do problema da separação cega, esta dissertação foi dividida em quatro partes: introdução ao problema do processamento e da separação cega; estudo de soluções para o problema da separação cega; descrição dos algoritmos selecionados para comparação e simulações computacionais; conclusões e comentários finais.

A primeira parte é composta pelos Capítulos 1 e 2, sendo que no Capítulo 2 é feita uma introdução ao problema do processamento cego de sinais. São apresentados os principais conceitos teóricos e as abordagens utilizadas na solução do referido problema, sendo dada

especial ênfase à separação cega e à ICA. No Capítulo 2 são apresentadas ainda as principais aplicações do processamento cego nas mais diferentes áreas de pesquisa.

Os Capítulos 3 e 4 compõem a segunda parte desta dissertação, na qual são descritas duas das principais ferramentas para solução do problema da separação cega: as estatísticas de segunda ordem e a análise de componentes independentes. No Capítulo 3, são apresentadas as principais técnicas para separação cega utilizando apenas estatísticas de segunda ordem. De forma sucinta, é feita uma revisão dos conceitos teóricos envolvidos e são apresentados os principais algoritmos baseados nestes conceitos.

No Capítulo 4, faz-se uma revisão do estado da arte de umas das mais importantes e mais utilizadas ferramentas para separação cega: a análise de componentes independentes.

A terceira parte desta dissertação dedica-se ao estudo, aplicação e comparação de três algoritmos selecionados dentre os inúmeros existentes para a implementação da separação cega e é composta pelos Capítulos 5 a 8. O Capítulo 5 descreve o AMUSE (*Algorithm for Multiple Unknown Signals Extraction*), que trabalha apenas com estatísticas de segunda ordem proporcionando uma abordagem rápida e simples para o problema da separação cega, sendo que sua eficiência está bem documentada em (Tong, 1991; Cichocki, 2002).

No Capítulo 6, é feita a descrição do algoritmo JADE (*Joint Approximate Diagonalization of Eigenmatrices*), o qual trata de componentes independentes através de uma abordagem algébrica (Cardoso, 1993; Cardoso, 1999). A base dos algoritmos algébricos, que utilizam quantidades estatísticas, é a exploração do fato de que a independência dos componentes de um vetor aleatório é equivalente a fazer com que todos os seus cumulantes cruzados sejam nulos. A principal vantagem desta abordagem é que praticamente não há necessidade de ajuste de nenhum parâmetro para que se consiga um bom desempenho. A principal desvantagem é sua complexidade computacional.

O Capítulo 7 descreve o algoritmo FlexICA (*Flexible Independent Component Analysis*). O FlexICA é um algoritmo adaptativo capaz de separar sinais super e sub-gaussianos através da utilização de uma não linearidade auto-adaptativa (Cichocki, 2002; Choi, 2000; Choi, 1998).

Finalmente, no Capítulo 8 são apresentadas várias simulações computacionais, realizadas com o auxílio do MATLAB, para verificar e comparar o desempenho dos algoritmos selecionados. Além das simulações com sinais de teste, os algoritmos JADE e FlexICA são testados para problemas típicos da literatura de Análise de Componentes Independentes: sinais de telecomunicações e sinais biomédicos.

Na quarta e última parte desta dissertação, são apresentadas as principais conclusões e comentários, além de sugestões para futuros trabalhos. Nesta parte final, são apresentadas ainda as referências bibliográficas que serviram de base para elaboração desta dissertação e que podem auxiliar no direcionamento de futuras pesquisas.

Capítulo 2: Processamento Cego de Sinais: Principais Conceitos e Aplicações

2.1 – Introdução

O problema do reconhecimento ou recuperação de sinais a partir da mistura de várias fontes tem se apresentado como um desafio para as áreas de processamento de sinais, pois embora, para o cérebro humano, esta seja uma tarefa simples, é de extrema complexidade para uma máquina.

As técnicas para o processamento e análise de dados, com o objetivo de recuperar, reconhecer ou separar fontes desconhecidas a partir unicamente dos sinais observados, obtidos por um conjunto de sensores, antenas ou microfones, recebem o nome de **Processamento Cego de Sinais** (BSP – *Blind Signal Processing*).

O adjetivo **cego** vem do fato de que os sinais originais não são observados e de que não se tem nenhuma informação, a priori, sobre a mistura a que estes sinais foram submetidos, ou sobre o canal pelo qual os sinais estão se propagando. Do ponto de vista dos algoritmos pode-se dizer que o adjetivo cego é derivado do fato de não haver nenhum sinal de referência, seqüência de treinamento ou supervisão para guiar o sistema de separação ou recuperação na execução de sua tarefa. Por isso, os algoritmos utilizados para solucionar o problema da separação cega são também conhecidos como **autodidatas** ou **não supervisionados**.

De fato, é difícil imaginar como é possível estimar os sinais originais sem o conhecimento dos parâmetros do processo da mistura ou do canal, sem considerar que há algo de mágico no processamento cego de sinais (Cichocki, 2002). Efetivamente, sem nenhum conhecimento do processo é impossível estimar os sinais fontes. Por isso, costuma-se dizer que a denominação, cego, é um pouco exagerada, uma vez que a definição de um modelo para mistura é um dos requisitos iniciais para a recuperação das fontes; além disso, é necessária que sejam impostas restrições ou considerações a respeito das fontes que se deseja recuperar. E, se for o caso, são feitas até restrições com relação ao ruído presente no sistema.

Este capítulo tem por objetivo descrever, de forma sintetizada, diferentes abordagens, métodos ou técnicas de processamento cego de sinais, em especial separação cega de sinais, análise de componentes independentes e deconvolução e equalização de múltiplos canais quando as medidas destes sinais feitas por sensores são contaminadas por misturas,

interferências ou ruído. São apresentadas ainda as principais áreas onde as técnicas apresentadas são aplicadas.

2.2 – Definição do Problema

Considere, inicialmente, o modelo para a solução do problema do processamento cego de sinais apresentado na Figura 2.1, onde se considera que um determinado número de sinais, $\mathbf{s} = \mathbf{s}(k) = [s_1(k), s_2(k), \dots, s_n(k)]^T$, está sendo emitido, ou transmitido, por alguma fonte física através de um canal \mathbf{A} . Estas fontes podem representar sinais elétricos emitidos pelo cérebro, sinais de voz ou ondas de rádio frequência. Considere, a seguir, que existam vários sensores, receptores ou antenas, em diferentes posições que medem e gravam uma mistura das fontes originalmente transmitida com pesos levemente diferentes. No canal de transmissão, além da mistura, os sinais originais sofrem o efeito do ruído e de interferências $\mathbf{v}(k)$. Os sinais misturados que chegam aos sensores são representados por $\mathbf{x} = \mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_m(k)]^T$.

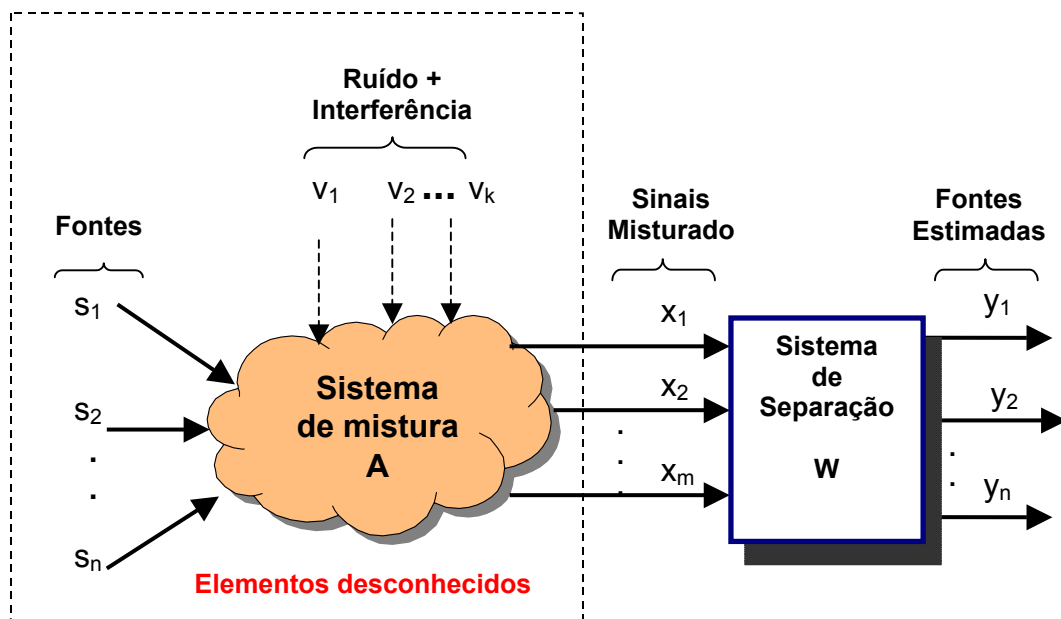


Figura 2.1 – Problema do processamento cego de sinais (Cichocki, 2002)

O problema do processamento cego de sinal é encontrar os sinais originais ($\mathbf{s}(k)$) a partir das medidas dos sensores ($\mathbf{x}(k)$).

A solução é extrair os sinais originais da mistura através de um sistema \mathbf{W} que forneça uma estimativa, $\mathbf{y} = \mathbf{y}(k) = [y_1(k), y_2(k), \dots, y_n(k)]^T$, destes sinais.

No caso mais geral tratado pelo processamento cego de sinais, consideram-se, portanto, m sinais misturados $\mathbf{x}(k)$. Estes sinais são combinados linearmente a partir de n sinais fontes $\mathbf{s}(k)$, e do canal $\mathbf{A}_{m \times n}$. Considera-se ainda a presença de ruído $\mathbf{v} = \mathbf{v}(k) = [v_1(k), v_2(k), \dots, v_m(k)]^T$. Em geral, tem-se que $m \geq n$. O sistema, representado de forma esquemática e matricial, é apresentado na Figura 2.2.

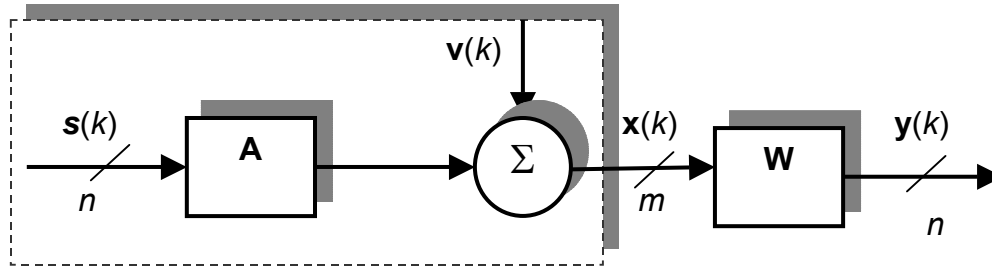


Figura 2.2 – Diagrama em blocos ilustrando o problema básico do processamento cego de sinais

Os sinais fontes $\mathbf{s}(k)$ são recuperados, por meio de um sistema de separação $\mathbf{W}_{n \times m}$, que fornece uma estimativa, $\mathbf{y}(k)$, dos sinais originais.

A formulação matemática do sistema da Figura 2.2, na forma matricial, é a mostrada em (2.1).

$$\begin{cases} \mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{v}(k) \\ \mathbf{y}(k) = \mathbf{W}\mathbf{x}(k) \end{cases} \quad (2.1)$$

onde:

$$\mathbf{x}(k) = [x_1(k), x_2(k), \dots, x_m(k)]^T \quad (2.2)$$

é vetor dos sinais medidos pelos sensores. E

$$\mathbf{s}(k) = [s_1(k), s_2(k), \dots, s_n(k)]^T \quad (2.3)$$

é o vetor dos sinais fontes. E ainda,

$$\mathbf{v}(k) = [v_1(k), v_2(k), \dots, v_m(k)]^T \quad (2.4)$$

é o vetor ruído. Os sinais recuperados ou estimados são representados por:

$$\mathbf{y}(k) = [y_1(k), y_2(k), \dots, y_n(k)]^T \quad (2.5)$$

As fontes $\mathbf{s}(k)$ são consideradas, em geral, variantes no tempo, com média zero e totalmente desconhecidas. O número n de fontes também é desconhecido e pode variar com o tempo, salvo alguma consideração inicial. O usuário tem disponíveis apenas medidas dos sensores $\mathbf{x}(k)$.

Assim, para solucionar este problema é preciso adaptar, ou estimar, os pesos w_{ij} da matriz \mathbf{W} para que as medidas dos sensores $\mathbf{x}(k)$ possam gerar estimativas $\mathbf{y}(k)$ dos sinais, ou em termos matemáticos:

$$y_j(k) = \hat{s}_j(k) = \sum_{i=1}^m w_{ji} x_i(k) \quad (j = 1, 2, 3, \dots, n) \quad (2.6)$$

Os pesos ótimos são encontrados quando os sinais de saída $\mathbf{y}(k)$ atendem uma das restrições estabelecidas, como por exemplo, ser tão estatisticamente independentes quanto possíveis.

Este problema freqüentemente é descrito como **Separação Cega de Sinais** (BSS–*Blind Source Separation*) (Kofidis, 2001; Cardoso, 1998; Hyvärinen, 2001; Mansur, 2000)

O sistema de mistura \mathbf{A} neste trabalho será representado por um sistema linear invariante no tempo (LTI – *Linear Time Invariant*) e, quanto ao nível de complexidade, \mathbf{A} será representada por uma matriz. Este tipo é denominado **modelo instantâneo de mistura**, pelo fato de que somente são consideradas as atenuações relativas à distância das fontes aos sensores. Neste trabalho, \mathbf{A} será considerada uma matriz quadrada, o que implica que o número de fontes é igual ao número de sensores.

Uma outra consideração em relação ao sistema de mistura diz respeito à presença ou não de ruído. A definição de ruído é sempre problemática e, por isso, existem basicamente duas formas de considerar sua presença no sistema. A primeira trata o ruído como uma fonte a ser separada das demais, considerando que ele é captado pelos sensores da mesma forma que demais fontes (Bell, 1995; Cichocki, 2002). Outra abordagem trata o ruído como elemento

que deteriora e mistura e que, portanto, não deve ser recuperado (Cardoso, 1993; Hyvärinen, 2001; Hyvärinen, 1999b; Douglas, 1998). Ao longo deste trabalho, a abordagem adotada para o tratamento do ruído será sempre descrita.

Os sinais de saída $y(k)$ são estimados a partir de certas indeterminações ou ambigüidades que, em termos matemáticos, podem ser expressas por parâmetros estatísticos arbitrários, permutações e atrasos dos sinais fontes estimados, mas que preservam a forma de onda dos sinais originais (Hyvärinen, 2001; Tong, 1991);

Considerando ainda o modelo da Figura 2.1 e os princípios supracitados, várias técnicas foram empregadas com o objetivo de tentar solucionar o problema do processamento cego de sinais. Dentre estas técnicas, as principais são:

- Análise de Componentes Independentes (**ICA** – *Independent Component Analysis*);
- Separação Cega de Fontes (**BSS** – *Blind Source Separation* ou *Blind Signal Separation*);
- Extração Cega de Sinais (**BSE** – *Blind Signal Extraction*);
- Deconvolução ou Equalização de Múltiplos Canais (**MBD** – *Multichannel Blind Deconvolution* ou **MBE** – *Multichannel Blind Equalization*).

Nas seções seguintes será apresentada uma descrição sucinta de cada uma destas técnicas.

2.3 – Separação Cega de Fontes - BSS

A **Separação Cega de Fontes** é um processo geral que tem por objetivo separar sinais ou fontes originais a partir de misturas observadas destes sinais (Cichocki, 2002; Hyvärinen, 1999a; Alvarez, 1999; Chan, 1997).

As pesquisas na área de separação cega de fontes são motivadas pelas inúmeras aplicações práticas que envolvem multi-fontes e multi-sensores sem o conhecimento das características do canal de transmissão.

As numerosas aplicações em campos tão diversos, que vão desde processamento de um conjunto de dados a recuperação de voz e imagem, provêm do fato de que para solucionar o problema da separação cega são necessários que se verifiquem um conjunto de hipóteses muito pouco restritivas.

Inerentes ao problema de separação, existem várias ambigüidades ou indeterminações. Uma delas diz respeito à ordem com que as fontes são recuperadas. Tal ordem é desconhecida

e qualquer permutação entre as fontes na saída é vista como aceitável. Outra ambigüidade está relacionada à amplitude dos sinais recuperados. Estas ambigüidades não são, de fato, importantes para a solucionar o problema da separação cega de fontes; o único interesse é com relação à preservação da forma de onda. Em (Tong, 1991) foi proposto o conceito de preservação da forma de onda e definido que todos os possíveis conjuntos de sinais recuperados devem pertencer à mesma classe de forma de onda, o que, formalmente, define uma família de soluções aceitáveis para o problema da separação cega. A incapacidade de se definir uma única solução para o problema ocorre devido ao fato de que não se têm informações sobre a distribuição de probabilidade das fontes e nem sobre as características do canal de transmissão

Em alguns casos, especialmente em aplicações práticas, considera-se que o número de fontes é desconhecido, sendo necessário, portanto, a utilização de técnicas para estimar o número de fontes (Cichocki, 1999; Cichocki, 1997).

Outra dificuldade encontrada ao se implementar um sistema de separação, e, talvez, a maior delas, é a complexidade computacional apresentada pela maioria dos métodos, o que torna estes sistemas difíceis de serem implementados em aplicações em tempo real. Assim sendo, o método ou algoritmo de separação para aplicações práticas deve apresentar robustez para lidar com ruído, além de ser capaz de recuperar as fontes desejadas de forma eficiente e com implementação computacional simples.

2.3.1 – BSS versus ICA

Na literatura, os problemas BSS e ICA são freqüentemente confundidos ou usados como sinônimo. Embora eles se refiram a modelos e algoritmos similares - considerando que as fontes originais são mutuamente independentes - nos casos gerais, especialmente em problemas do mundo real, os objetivos da BSS e da ICA são um pouco diferentes.

O objetivo da ICA é encontrar transformações que assegurem que os sinais de saída do sistema sejam tão independentes quanto possíveis, através da maximização da independência estatística entre os componentes observados. Procura-se, portanto, representar os dados de uma forma eficiente e que preserve sua informação. A ICA não precisa necessariamente da existência de uma matriz de mistura.

O objetivo da BSS, por outro lado, é estimar os sinais fonte originais mesmo que eles não sejam completa, mútua e estatisticamente independentes.

Deve-se observar que os métodos para ICA utilizam, na maioria dos casos, estatística de ordem superior (HOS – *High Order Statistics*), enquanto que os métodos para BSS são capazes de utilizar estatística de segunda ordem (SOS – *Second Order Statistics*) (Cichocki, 2002; Tong, 1990; Alvarez, 1999).

Os métodos de segunda ordem consideram que as fontes têm uma estrutura temporal, enquanto que os métodos de ordem superior levam em consideração sua independência mútua.

Uma outra, e não menos importante, diferença é que os métodos estatísticos de ordem superior não podem ser aplicados a sinais gaussianos, enquanto que os métodos de segunda ordem não apresentam tal restrição (Tong, 1990; Tong, 1991)

Deste ponto de vista, os métodos para BSS não substituem os para ICA e vice e versa, uma vez que cada abordagem é baseada em diferentes suposições e, frequentemente, diferentes objetivos (Cichocki, 2002).

2.3.2 – Técnicas para Separação Cega de Fontes

Embora existam inúmeras técnicas para solucionar os problemas de separação cega como descrito em (Cichocki, 2002), a abordagem mais popular adota como função custo alguma medida de independência entre os sinais. Quando os sinais são considerados estatisticamente independentes sem uma estrutura temporal, a estatística de ordem superior é essencial para solucionar o problema de BSS. Neste caso, os métodos não permitem mais que uma fonte gaussiana e utilizam a ICA como principal ferramenta (Hyvärinen, 2001).

Se as fontes têm estrutura temporal, podem ser usadas condições menos restritivas que a independência estatística, a saber, a estatística de segunda ordem que é suficiente para estimar a matriz de mistura (Tong, 1990; Amari, 2000; Degerine, 2000; Belouchrani, 1993). É importante salientar que os métodos de segunda ordem não permitem a separação de fontes com espectros de potência idênticos ou fontes independentes e identicamente distribuídas (i.i.d.). Em comparação com os métodos de HOS, os métodos de SOS não precisam inferir na distribuição de probabilidade das fontes ou utilizar funções não lineares de ativação. A simplicidade computacional, a velocidade de convergência e o bom desempenho fazem deste um método adequado para lidar com o problema da separação cega.

Alguns pesquisadores têm utilizado também técnicas de inteligência artificial, como as redes neurais para solucionar os problemas de BSS (Amari, 1996; Amari, 1998; Cardoso, 1996; Choi, 2000c; Karhunen, 1997).

Existem ainda técnicas que exploram a diversidade dos sinais. A diversidade está relacionada as diferentes características ou atributos de um determinado sinal. Nestas técnicas são utilizadas as diversidades de tempo, frequência ou, de forma generalizada, a diversidade espaço-temporal (Cichocki, 2002).

Técnicas mais avançadas são obtidas pela combinação ou integração das técnicas acima mencionadas, com o objetivo de separar ou extrair fontes com propriedades estatísticas variadas e mitigar a influência de ruído ou interferências indesejáveis (Hyvärinen, 2001; Chen, 1997).

2.4 – Análise de Componentes Independentes – ICA

Na literatura, são encontradas várias definições ou métodos para ICA, de acordo com o modelo e aplicação a ser utilizada. De uma forma geral, pode-se dizer que a Análise de Componentes Independentes é um método matemático que procura determinar fatores ou componentes a partir de dados estatísticos multivariáveis. O que diferencia de outros métodos é que este método trata componentes estatisticamente independentes e não gaussianos. A independência entre os componentes implica que o valor de um componente não fornece nenhuma informação com relação a outros componentes.

A Análise de Componentes Independentes pode ser vista ainda como uma extensão da técnica de Análise de Componentes Principais (PCA – *Principal Component Analysis*), a qual trabalha somente com a independência de segunda ordem e, conseqüentemente, define direções ortogonais. Atualmente, na ICA, as técnicas de PCA são empregadas para branquear os sinais observados (Hyvärinen, 2001; Fuentes, 2001).

Aplicações potenciais da ICA incluem análise e compressão de dados, processamento de imagens e, atualmente, a técnicas de ICA tem sido o método favorito para solucionar o problema da separação cega de sinais mutuamente independentes e não gaussianos.

Em situações práticas, não é possível encontrar componentes completamente independentes, mas, sim, componentes tão independentes quanto possível.

Para se estabelecer um modelo para solucionar o problema de ICA, considera-se, inicialmente, um conjunto de variáveis $[x_1(k), \dots, x_m(k)]^T$, onde k representa tempo ou um índice das amostras. Considerando que estas variáveis são geradas a partir de uma mistura linear de componentes independentes, ou matematicamente:

$$\begin{bmatrix} x_1(k) \\ \vdots \\ x_m(k) \end{bmatrix} = \mathbf{A} \begin{bmatrix} s_1(k) \\ \vdots \\ s_n(k) \end{bmatrix} \quad (2.7)$$

Onde \mathbf{A} é a matriz de mistura desconhecida.

A análise de componentes independentes consiste em estimar \mathbf{A} e $\mathbf{s}(k)$ a partir de $\mathbf{x}(k)$, ou de uma forma alternativa pode-se dizer que a técnica ICA consiste em encontrar uma transformação linear representada por uma matriz \mathbf{W} , tal que as variáveis de saída y_i , $i=1, \dots, n$, sejam tão independentes quanto possíveis, sendo que, $y_i(k) = \mathbf{W}x_i(k)$.

Os dados de saída só podem ser estimados se os componentes de s_i forem não gaussianos.

Embora vários artigos proponham novos métodos ou algoritmos para ICA, o modelo matemático basicamente é o mesmo em todos os casos. As fontes são modeladas como processos aleatórios e a independência estatística é explorada. De forma geral, o critério para se proceder a separação é a medida de independência, representada como uma função custo. A otimização desta função custo, com relação a algum dos parâmetros do sistema de separação, resultará na determinação dos componentes independentes, o que reduz o problema da separação cega a um problema de otimização matemática.

Mais detalhes sobre ICA serão apresentados no Capítulo 4 desta dissertação.

2.5 – Deconvolução Cega de Múltiplos Usuários

Outra técnica estudada pelo processamento cego de sinais é a **Deconvolução Cega de Sinais** (BSD – *Blind Signal Deconvolution*) (Haykin, 1994; Hyvärinen, 2001; Bell, 1995; Amari, 1997). Dá-se o nome de deconvolução ao processo pelo qual os efeitos da convolução, causados por um sistema LTI em um sinal de entrada, são mitigados.

Historicamente, muitas idéias usadas em ICA foram originalmente propostas para a deconvolução cega. Assim, a deconvolução cega pode ser encarada como o antecessor intelectual da ICA.

Na deconvolução cega somente o sinal de saída é conhecido e o objetivo é procurar determinar o sistema de convolução e o sinal de entrada.

As técnicas de deconvolução cega são exploradas em ambientes externamente não estacionários. No caso cego, o equalizador (ou filtro adaptativo) opera sem supervisão, ou seqüência de treinamento.

Vários algoritmos foram desenvolvidos para solucionar o problema da deconvolução cega baseados em diferentes restrições ou considerações, sendo que os dois principais grupos de algoritmos são os baseados em estatísticas de ordem superior e os ciclo-estacionários (Haykin, 1994).

Os algoritmos baseados em HOS podem ser divididos em dois grupos: algoritmos implícitos e explícitos. Os algoritmos implícitos exploram as estatísticas de ordem superior do sinal recebido de uma forma implícita. Neste grupo estão os algoritmos de Bussgang, que recebem esta denominação pelo fato de que a seqüência estimada de saída, após a convergência do algoritmo, segue as estatísticas de Bussgang (Neves, 2001). Os algoritmos explícitos utilizam HOS e suas transformadas de Fourier (Cichoki, 2002).

Existem ainda os algoritmos ciclo-estacionários que exploram as estatísticas de segunda ordem do sinal de entrada. A propriedade da ciclo-estacionaridade é comum a sinais modulados em amplitude, fase ou frequência (Cichoki, 2002).

No caso em que são considerados vários sinais de entrada e saída, o processo de deconvolução é denominado **deconvolução de múltiplos usuários** ou **múltiplos canais** (MBD ou MU-BD – *Multichannel* ou *Multi-User Blind Deconvolution*)

O problema de deconvolução cega de múltiplos usuários (ou canais) pode ser encarado como uma extensão, ou uma generalização, do problema da separação cega de sinais, como pode ser visto na Figura 2.3.

Na deconvolução multidimensional, um vetor m -dimensional de sinais discretos recebidos $\mathbf{x}(k)$, no instante k , é gerado a partir de um vetor n -dimensional $\mathbf{s}(k)$, $m \geq n$, utilizando um modelo de mistura estável (Amari, 1997; Haykin, 1994; Cichocki, 2002).

Matematicamente o problema da deconvolução de múltiplos usuários é descrito em (2.8).

$$\mathbf{x}(k) = \sum_{p=-\infty}^{\infty} \mathbf{A}_p \mathbf{s}(k-p) = \mathbf{A}_p * \mathbf{s}(k) \quad (2.8)$$

Onde $*$ significa convolução e \mathbf{A}_p é uma matriz $m \times n$ dos coeficientes dos coeficientes da mistura no instante $(k-p)$.

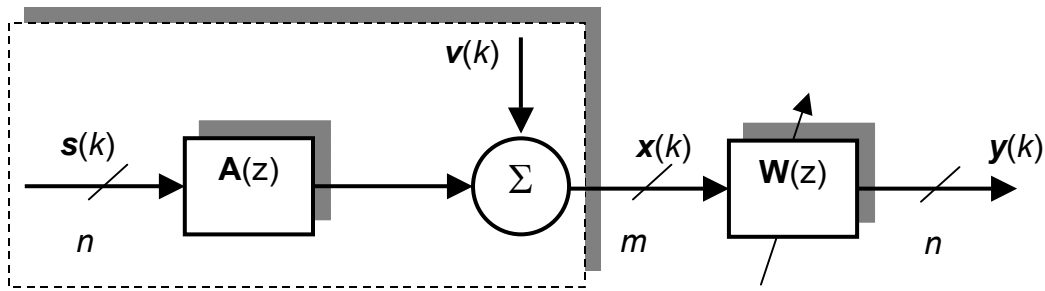


Figura 2.3 – Diagrama em blocos ilustrando o modelo utilizado na deconvolução cega de múltiplos usuários.

O sistema de convolução ou mistura $\mathbf{A}(z)$ pode ser definido como:

$$\mathbf{A}(z) = \sum_{p=-\infty}^{\infty} \mathbf{A}_p z^{-p} \quad (2.9)$$

Onde z^{-1} é a unidade de atraso. Ou seja, $\mathbf{A}(z)$ é a Transformada Z de \mathbf{A}_p .

Com base nas equações (2.8) e (2.9) pode-se escrever:

$$\mathbf{x}(k) = [\mathbf{A}(z)]\mathbf{s}(k) \quad (2.10)$$

O objetivo da MBD é, portanto determinar uma versão filtrada dos sinais fontes ($\mathbf{y}(k)$), definido por (2.11) a partir dos sinais recebidos, utilizando um conhecimento aproximado da distribuição e estatísticas das fontes.

$$\mathbf{y}(k) = \mathbf{W}_p * \mathbf{x}(k) = [\mathbf{W}(z, k)]\mathbf{x}(k) \quad (2.11)$$

Em geral, os sinais fonte são considerados seqüências i.i.d, independente de todas as outras seqüências fontes (Neves, 2001).

2.6 – Extração Cega de Fontes – BSE

Os problemas da separação cega e deconvolução de múltiplos usuários têm atraído muito interesse, devido às suas inúmeras aplicações nas mais diferentes áreas. Entretanto, a

solução destes problemas pode depender um tempo de execução demasiadamente alto, pois em algumas aplicações é necessário o emprego de um grande número de sensores, como no caso de aplicações biomédicas. Por outro lado, em geral, apenas um pequeno número de fontes é de interesse; as demais fontes são interferências ou ruído.

Assim sendo, é mais adequado recuperar apenas o pequeno conjunto de fontes de interesse. A técnica que explora a extração de apenas algumas fontes de forma cega é conhecida como **Extração Cega de Fontes** (BSE – *Blind Source Extraction*).

Na BSE os sinais podem ser extraídos, ou separados, em uma ordem específica de acordo com as características estatísticas das fontes. A BSE pode ainda ser considerada uma generalização de Análise de Componentes Principais, onde os sinais são extraídos de acordo com a ordem crescente de suas variâncias. Os algoritmos de BSE disponíveis são, em geral, mais simples que os para BSS instantânea (Cichocki, 2002; Hyvärinen, 1997a; Hyvärinen, 1997b; Hyvärinen, 2000). Porém, como os algoritmos de BSE extraem as fontes uma a uma, é despendido um enorme esforço computacional para evitar que haja replicação de fontes.

Quando combinados com procedimento de remoção ou deflatação (*deflation*) (Delfosse, 1995), que realiza a sub-extração sequencial de fontes, os algoritmos para BSE podem realizar a extração sequencial de todas as fontes independentes.

De uma forma geral, pode-se dizer que a BSE é uma excelente abordagem quando o objetivo é extrair fontes com características estatísticas específicas a partir de uma determinada mistura.

2.7 – Principais Aplicações das Técnicas de Processamento Cego

Nas seções seguintes são descritas as principais áreas e aplicações nas quais as são empregadas técnicas de processamento cego de sinais, dentre as quais se encontram processamento e análise de sinais biomédicos (EEG, MEG, ECG), processamento de dados geofísicos, reconhecimento de voz e imagem, busca de dados, processamento de sinais acústicos e comunicações sem fio, tais aplicações podem ser encontradas em (Hyvärinen, 2001; Cardoso, 1996; 1993; Lee, 1997; Torkkola, 1999; Fuentes, 2001; Knaack, 2001; Ristaniemi, 2002a; Ristaniemi, 2002b; Sanchez, 2002). Nas próximas seções serão descritas algumas destas aplicações.

2.7.1 – Processamento de Sinais Biomédicos

A literatura apresenta inúmeras aplicações do BSP na área biomédica. Nestas aplicações é comum encontrar situações nas quais o sinal (ou sinais) de interesse é fraco, não estacionário e distorcido por ruído e interferências.

As principais aplicações na área biomédica, na atualidade, encontram-se no monitoramento das variações que podem ocorrer nos órgãos internos do corpo humano.

Entre outras, as aplicações de BSP na área de análise de sinais biomédicos incluem: a melhoria dos baixos níveis dos componentes de ECG (eletrocardiograma); extração de eletrocardiograma fetal (FECG) e materno (MECG); cancelamento de ruído, eliminação e detecção de potenciais incorretos em monitoramento por múltiplos sensores em eletroencefalograma (EEG) e magnetoencefalograma (MEG); melhoria nos sinais de eletromiografia (EMG) além de reconhecimento e processamento de imagens biomédicas (Cichocki, 2002; Makeig, 1996; Barros, 2003; 2000; 1999; Vigário, 2000).

O BSP tem se mostrado especialmente útil nas áreas de processamento de imagens para cerebrais como acontece com as medidas de MEG e EEG. Nas medidas de MEG, por exemplo, a utilização de ICA tem se mostrado mais eficiente que os métodos convencionais de processamento.

As medidas de MEG fornecem informações semelhantes às medidas de EEG, mas com uma melhor resolução espacial. Estas medidas são utilizadas, em geral, para o monitoramento das atividades cerebrais. Nestas medidas a ICA é utilizada para decompor as saídas dos eletrodos em componentes originadas no cérebro e componentes de ruído e artefatos. A Tabela 2.1 exemplifica a aplicação de ICA em sinais de MEG, onde se tem que (Parra, 2002):

x	=	A	.	s
Campo magnético medido pelos sensores		Acoplamento magnético ou atenuações		Corrente efetiva dos neurônios

Tabela 2.1 – ICA aplicada a MEG.

Para a aplicação da ICA em sinais de EEG e MEG são necessárias que se verifiquem várias condições, entre as quais estão: a existência de sinais estatisticamente independentes;

mistura linear e instantânea destes sinais nos sensores e deve-se considerar ainda que os sinais e o sistema de mistura são estacionários (Hyvärinen, 2001).

O critério de independência considera apenas as relações estatísticas entre os sinais envolvidos, e não a morfologia ou a fisiologia neural envolvida. Assim, este critério não pode ser utilizado como regra geral, pois sua validade deste critério depende de resultados experimentais. A propagação dos sinais no interior no cérebro é imediata, ou seja, não é necessário considerar qualquer valor de atraso e mistura instantânea pode ser utilizada sem perda de generalidade.

Quanto às fontes são consideradas processos estocásticos, em teoria, os sinais devem ser consideradas estacionários para garantir que sua distribuição seja representativa. Além disso, considerar que o sistema de mistura (**A**) é estacionária é crucial para a utilização das técnicas de ICA. Felizmente, esta consideração está de acordo com os modelos neurais comumente utilizados.

2.7.2 – Sistemas Acústicos

As aplicações das técnicas de BSP a sinais acústicos são de especial importância. Nesta área incluem-se aplicações clássicas como a extração de voz a partir de sinais captados por vários microfones espalhados em uma sala onde várias pessoas estão falando simultaneamente, problema este conhecido como *cocktail party*, e aplicações que envolvem sinais de sonar e sinais de vibração (Haykin, 2003; Torkkola, 1999).

Na Figura 2.4, representa-se o problema clássico da recuperação de sinais de voz, onde cada microfone recebe uma cópia dos sinais fontes bem como os sinais refletidos e corrompidos destas fontes. Este problema é solucionado por técnicas de deconvolução cega de múltiplos usuários, que procuram deconvoluir as fontes originais a partir das misturas recebidas pelos sensores ou mais precisamente por microfones. Pode-se dizer que estas técnicas procuram imitar a habilidade humana de focar a atenção em um único som desprezando os demais (Smaragdís, 1997).

Outras aplicações na área de processamento de sinais de áudio incluem projeto de dispositivos controlados por voz e identificadores de voz onde as máquinas precisam reconhecer o comando de voz em um ambiente com outras vozes e ruído.

Em situações onde o sinal desejado é corrompido por ruído, o sinal desejado pode ser recuperado tratando cada sinal de ruído como uma fonte e assim, separando o sinal desejado e removendo ou extraindo as fontes de ruído (Chan, 1997). Esta técnica é útil principalmente

quando o sinal desejado encontra-se em um meio com fortes interferências, como por exemplo, sinais de ultra-som para análise de fluidos e sistemas de sonar.

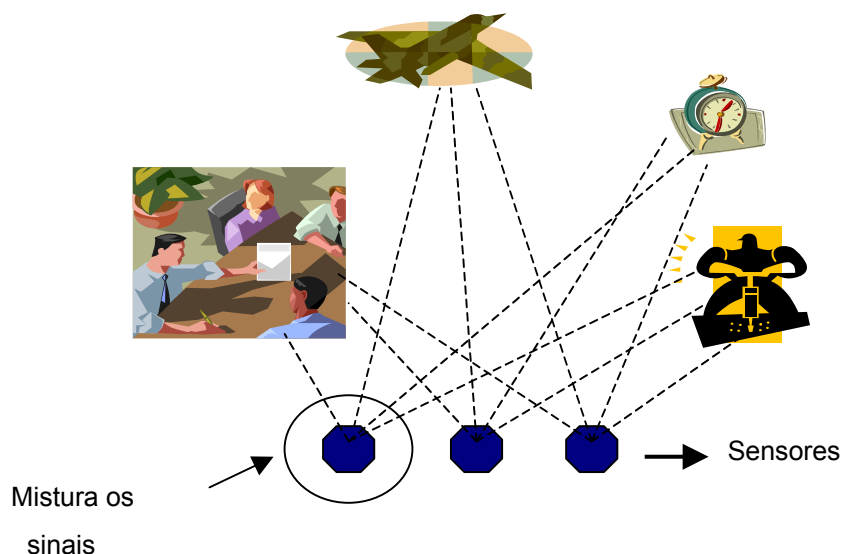


Figura 2.4 – Problema do coquetel (*cocktail party*) e reconhecimento de voz.

2.7.3 – Sistemas de Comunicação

Os sistemas de comunicações sem fio e comunicações via satélite são exemplos de sistemas de comunicação por rádio frequência que tem despertado grande interesse na atualidade.

A necessidade crescente de se aumentar a capacidade dos sistemas, faz com que muitas vezes haja necessidade de compartilhamento de espectro de frequências por um número cada vez maior de usuários. Conseqüentemente surge a necessidade do emprego de novas técnicas para processamento de sinais que permitam recuperar os sinais desejados. As técnicas de BSP são especialmente úteis nestes casos.

As principais aplicações das técnicas de BSP, na área de comunicações, encontram-se em rádio digital com diversidade, canais de rádio com polarização dual, linhas digitais de assinantes de alta velocidade, sistemas de comunicação com multiusuário/multiacesso, sistemas de radar com múltiplos sensores e sistemas de posicionamento via satélite (Talwar,1996; Hyvärinen, 2001).

Os algoritmos cegos vêm ganhando especial atenção na área de comunicações devido ao fato de que, entre outras características, evita a necessidade de seqüência de treinamento

que é um problema principalmente em ambientes sujeitos a interferências e/ou que variam com o tempo e em sistemas que necessitam de sincronismo. Com a eliminação ou redução da seqüência de treinamento pode-se aumentar a capacidade do sistema como um todo.

A Figura 2.5 ilustra um sistema típico de comunicação onde os sinais dos usuários são recebidos, através de múltiplos caminhos, por um conjunto de antenas, sendo que cada sinal recebido esteve sujeito a diferentes interferências, atenuações e ruído aditivo.

Como pode ser observado, os sinais são distorcidos pelos objetos antes de atingir as antenas, criando um problema conhecido como desvanecimento por múltiplos caminhos.

Atualmente, as atenções dos pesquisadores na área de telecomunicações estão voltadas principalmente para trabalhos que envolvem a redução ou mitigação de interferências (*jamming mitigation*) em sistemas de comunicação de espectro espalhado (*spread spectrum*) (Ristaniemi, 2002a; Ristaniemi, 2002b; Raju, 2002). Nestas aplicações, as técnicas de BSS são utilizadas na tentativa de separar os sinais desejados dos interferentes, produzindo, como consequência, uma redução das interferências nos sinais.

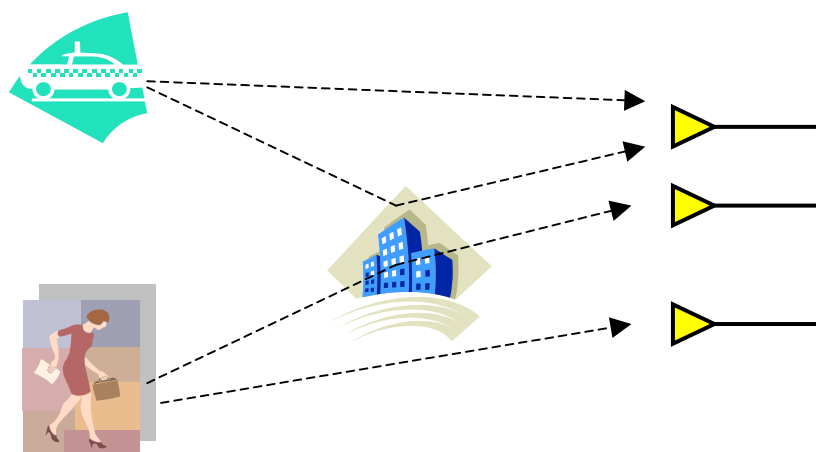


Figura 2.5 – Ambiente de comunicação digital

A Figura 2.6 apresenta um sistema para redução de interferências em sistemas de espectro espalhado baseado em técnicas de separação cega, como proposto em (Belouchrani, 2000). O sistema proposto é robusto com relação à multi-caminho e imperfeições no conjunto de sensores. Na Figura 2.6, a demodulação dos sinais em espectro espalhado é realizada por um pré-processador, que consiste em um separador seguido por um seletor. O bloco **separador**, implementado utilizando técnicas de separação cega de sinais, age separando a interferência do sinal desejado, através da utilização da diversidade espacial proporcionada pelo conjunto sensores. Assim, o bloco de **seleção** é necessário para classificar os sinais

separados em *jammer* ou sinal desejado. Esta seleção pode ser implementada com base em uma característica específica do sinal desejado reconhecido pelo receptor, remetendo a um problema de classificação de sinais. O processo de demodulação recupera, então, os sinais originais desespalhando os sinais desejados e mitiga os interferentes.

Outra aplicação também de grande interesse, diz respeito à estimativa de direção de chegada (DOA – *Directional Of Arrival*) dos sinais desejados (Fuentes, 2001).

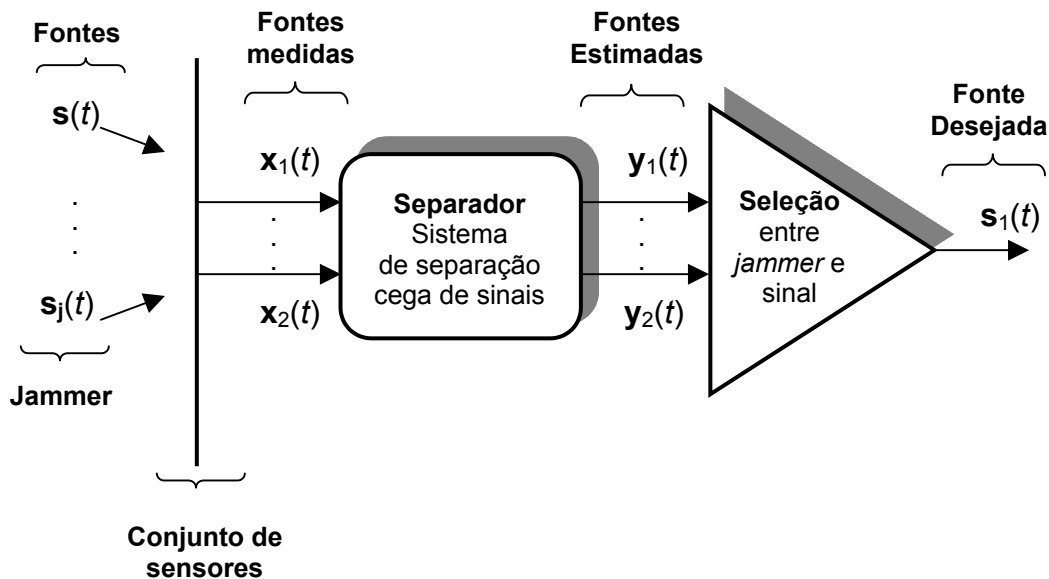


Figura 2.6 - BSS aplicada a sistema de espectro espalhado (Belouchrani,2000).

Podem, ainda, ser citados os problemas de BSP relacionado à separação e equalização cega espaço-temporal dos sinais transmitidos, que procuram estimar as fontes e seus canais de propagação na presença de ruído e interferência co-canal (Hyvärinen, 2001; Cichocki, 2002).

2.7.4 - Monitoramento das Condições de Máquinas

O Monitoramento das Condições das Máquinas (MCM) é uma área que vem apresentando resultados cada vez mais confiáveis dentre as pesquisas em processamento de sinais e visa principalmente à detecção de falhas em máquinas rotativas.

Estatísticas de ordem superior podem ser utilizadas para agir como medidores sensíveis da presença de falhas, com índice de sucesso superior a 95% (Nandi, 2002).

Um dos maiores problemas no processamento de dados de sistemas de MCM está no fato de que é praticamente impossível prever limites (*bounds*) nos dados; além disso, é muito difícil obter dados suficientes das condições de uma falha para caracterizá-la completamente.

Os dados relativos ao MCM contêm, em geral, dimensões extremamente altas e dados de alta dimensão tendem a ter limites não lineares, o que dificulta sua análise, criando um problema para os pesquisadores da área. Por esta razão, Redes Neurais e técnicas de Separação Cega de Fontes vêm sendo empregadas com sucesso na solução deste problema. Através do uso de transformações não lineares em espaços de altas dimensões é possível obter um grau de separação que é impossível de se obter utilizando outras técnicas.

Em (Knaak, 2001) é proposto um sistema para monitoramento e diagnóstico de falhas de máquinas rotativas baseado no som produzido pela máquina, levando em consideração que uma falha modifica o som característico da máquina. Como em ambientes reais o som da máquina é corrompido por interferências e ruído, tem-se, freqüentemente uma relação sinal-ruído (SNR), em dB, negativa. O sistema proposto utiliza, então, separação cega de fontes para recuperar o som original da máquina.

Em (Ypma, 2001) também são aplicadas técnicas para extrair uma assinatura ou característica peculiar de uma máquina rotativa a partir de medidas distorcidas ou contaminadas pelo ambiente onde a máquina se encontra. São realizados, durante os experimentos, monitoramento acústico e de vibração. A Figura 2.7 mostra o esquema utilizado para detecção de falhas em bombas submersas, onde cada IC (Componente Independente) fornece uma das características da máquina.

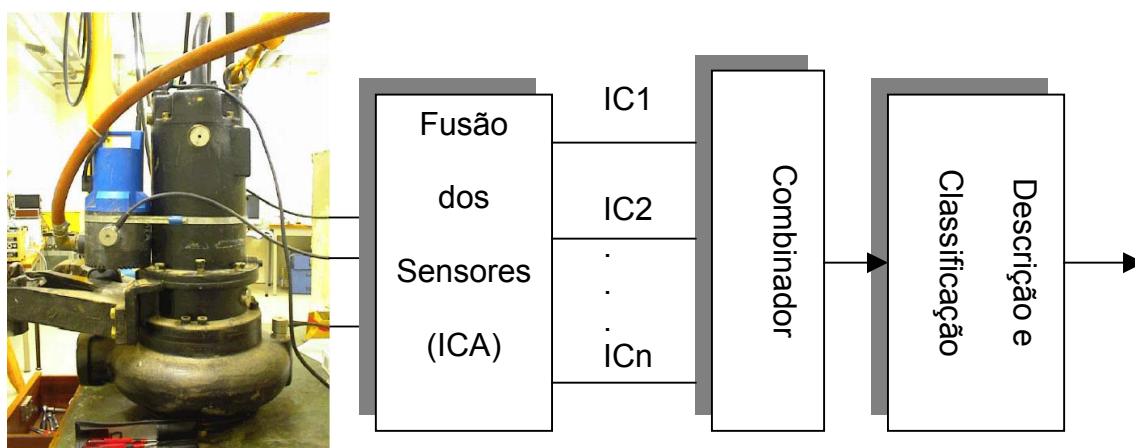


Figura 2.7 – Esquema de teste para detecção de falhas em bombas submersas utilizando ICA.

Capítulo 3 - Identificação Cega de Fontes Baseada em Estatística de Segunda Ordem

3.1- Introdução

Como visto no Capítulo 1, o objetivo da separação cega é identificar os sinais ou fontes originais a partir de um conjunto de medidas resultantes da mistura linear destas fontes, sem o conhecimento destas ou do sistema de mistura a que foram submetidas.

Como a separação é feita apenas com base na informação dos sinais recebidos pelos sensores, o desempenho das técnicas, ditas cegas, não é afetado por erros no modelo de propagação ou na calibração do conjunto de sensores; o que não ocorre quando são utilizadas técnicas convencionais de processamento de sinais.

Nos últimos anos, várias abordagens têm sido propostas para solucionar o problema da separação cega (Comon, 1994; Cardoso, 1993; Bell, 1995; Pope, 1996; Jutten, 1991; Tong, 1991). Uma destas abordagens considera que os sinais originais são espacialmente decorrelacionados ou brancos, com diferentes funções de autocorrelação, mas temporalmente correlacionados ou coloridos. Neste caso, a identificação cega é realizada utilizando apenas estatísticas de segunda ordem (Belouchnari, 1993; Tong, 1990; Belouchnari, 1997; Belouchrani, 1998).

Outras abordagens exploram ainda a não-estacionaridade de segunda ordem das fontes para estimar a matriz de mistura. A consideração de não-estacionaridade é viável uma vez que muitos sinais reais são inerentemente não-estacionários. Trabalhos que utilizam não-estacionaridade para solucionar o problema da separação cega podem ser encontrados em (Choi, 2002; Choi, 2001; Choi, 2000a; Chang, 2000; Pham, 2000).

As técnicas que utilizam estatísticas de segunda ordem podem ser divididas em dois grupos principais:

- a) os baseados em diagonalizações simultâneas implementadas por meio de autodecomposição (EVD – decomposição em autovalores, SVD – decomposição em valores singulares ou GEVD – decomposição generalizada em autovalores);
- b) e os baseados em diagonalização conjunta.

Em comparação com os métodos que utilizam estatísticas de ordem superior, os métodos baseados em estatísticas de segunda ordem não precisam estimar a função densidade de probabilidade (f.d.p.) das fontes ou uma função de adaptação não linear.

Neste capítulo serão apresentados alguns dos principais algoritmos, que trabalham com estatísticas de segunda ordem, considerando algoritmos baseados em diagonalizações simultâneas e em diagonalização conjunta.

3.2 - Separação Cega Utilizando SOS: Considerações Iniciais

Os sistemas de separação ou identificação cega de fontes são constituídos por multi-fontes e multi-sensores e têm por objetivo estimar, separar ou identificar as fontes originais sem o conhecimento destas e sem o conhecimento das características do meio através do qual elas se propagam. O problema da identificação pode ser representado pela Figura 3.1.

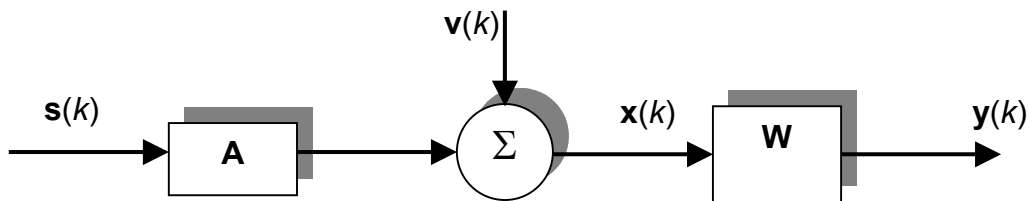


Figura 3.1- Diagrama em blocos para o problema da separação cega utilizando SOS.

Matematicamente, pode-se escrever que:

$$\mathbf{x}(k) = \mathbf{A} \mathbf{s}(k) + \mathbf{v}(k) \quad (3.1)$$

Onde $\mathbf{x}(k) \in \mathbf{R}^m$ é o vetor das fontes observadas, $\mathbf{s}(k) \in \mathbf{R}^n$ é o vetor dos sinais originais e $\mathbf{v}(k) \in \mathbf{R}^m$ é o ruído aditivo e $\mathbf{A} \in \mathbf{R}^{m \times n}$ é a matriz de mistura que caracteriza o meio ou o canal de transmissão.

O objetivo da identificação ou separação cega é identificar a matriz de mistura \mathbf{A} e/ou a matriz de separação \mathbf{W} , a partir de $\mathbf{x}(k)$.

É importante lembrar que a amplitude das fontes originais e a ordem com que elas foram transmitidas não podem ser estimadas diretamente pelos algoritmos que realizam a

separação cega. A amplitude e ordem ou permutação das fontes constituem as duas principais ambigüidades ou indeterminações do sistema de separação.

O que se procura, então, por meio do sistema de separação, é a preservação da forma de onda como definida por Tong et al. (1993), que diz que se a matriz de mistura estimada $\hat{\mathbf{A}}$ satisfaz a relação:

$$\mathbf{G} = \mathbf{W}\mathbf{A} = \hat{\mathbf{A}}^+ \mathbf{A} = \mathbf{P}\mathbf{D} \quad (3.2)$$

Onde \mathbf{G} representa a transformação global que combina os sistemas de mistura e separação, \mathbf{A}^+ é a pseudo-inversa de \mathbf{A} , - se a matriz de mistura é quadrada e não singular pode-se utilizar a matriz inversa ao invés da pseudo-inversa -, \mathbf{P} é uma matriz de permuta e \mathbf{D} é uma matriz diagonal não singular de escalonamento. Então, pode-se dizer que:

$$(\hat{\mathbf{A}}, \hat{\mathbf{s}}) = (\mathbf{A}, \mathbf{s}) \quad (3.3)$$

estão de acordo com a relação de preservação da forma onda, sendo que $\hat{\mathbf{s}}$ indica as fontes estimadas.

Além da necessidade da preservação da forma de onda, visando à utilização de métodos de SOS para a solução do problema da separação cega, é necessário levar em conta algumas considerações, conforme mostrado por (Tong, 1990; Cichocki, 2002; Belouchnari, 1997):

- A matriz de mistura \mathbf{A} é de posto completo (*full column rank*);
- As fontes $\mathbf{s}(k)$ e o ruído $\mathbf{v}(k)$ são estatística e mutuamente independentes;
- As fontes são espacialmente descorrelacionadas com diferentes funções de autocorrelação, mas são sinais estocásticos temporalmente correlacionados (coloridas);
- As fontes são sinais estacionários e/ou não estacionários de segunda ordem no sentido que suas variâncias variam com o tempo;
- No caso de se considerar ruído aditivo, este é independente dos sinais fontes, podendo ser espacialmente correlacionado, mas temporalmente branco.

Retornando ao modelo linear apresentado na equação (3.1), a matriz de correlação \mathbf{R}_x , para $\mathbf{x}(k) = \mathbf{x}$, pode ser definida como:

$$\begin{aligned}
 \mathbf{R}_x &= \mathbf{E}[\mathbf{x}\mathbf{x}^T] = \mathbf{E}[(\mathbf{A}\mathbf{s} + \mathbf{v})(\mathbf{A}\mathbf{s} + \mathbf{v})] \\
 &= \mathbf{E}[(\mathbf{A}\mathbf{s}\mathbf{s}^T \mathbf{A}^T)] + \mathbf{E}[(\mathbf{A}\mathbf{s}\mathbf{v}^T)] + \mathbf{E}[(\mathbf{v}\mathbf{s}^T \mathbf{A}^T)] + \mathbf{E}[(\mathbf{v}\mathbf{v}^T)] \\
 &= \mathbf{A}\mathbf{E}[(\mathbf{s}\mathbf{s}^T)]\mathbf{A}^T + \mathbf{A}\mathbf{E}[(\mathbf{s}\mathbf{v}^T)] + \mathbf{E}[(\mathbf{v}\mathbf{s}^T)]\mathbf{A}^T + \mathbf{E}[(\mathbf{v}\mathbf{v}^T)] \\
 &= \mathbf{A}\mathbf{R}_s\mathbf{A}^T + \mathbf{A}\mathbf{R}_{sv} + \mathbf{R}_{vs}\mathbf{A}^T + \mathbf{R}_v
 \end{aligned} \tag{3.4}$$

Considerando $\mathbf{v} = \mathbf{v}(k)$ tem média zero e é decorrelacionado de \mathbf{s} , tem-se que,

$$\mathbf{R}_{sv} = \mathbf{R}_{vs} = 0. \tag{3.5}$$

Logo:

$$\mathbf{R}_x = \mathbf{A}\mathbf{R}_s\mathbf{A}^T + \mathbf{R}_v \tag{3.6}$$

Outra consideração que pode ser feita é que o ruído é branco; neste caso, os componentes de \mathbf{v} são todos decorrelacionados e tem variância σ^2 , tal que,

$$\mathbf{R}_v = \mathbf{R}_v(0) = \sigma^2 \mathbf{I} \tag{3.7}$$

e

$$\mathbf{R}_x = \mathbf{R}_x(0) = \mathbf{A}\mathbf{R}_s\mathbf{A}^T + \sigma^2 \mathbf{I} \tag{3.8}$$

Assim, considerando que as fontes são espacialmente decorrelacionadas, mas temporalmente correlacionadas e, considerando a presença de ruído gaussiano, as matrizes de correlação do vetor de observações $\mathbf{x}(k)$ satisfazem as relações (3.9) e (3.10) para qualquer atraso τ . Em (3.9) e (3.10), $\mathbf{R}_x(0)$ e $\mathbf{R}_x(\tau)$ são matrizes de correlação diagonais, \mathbf{I}_m é uma matriz identidade $m \times m$ e \mathbf{A}^T indica a transposta da matriz \mathbf{A} .

$$\mathbf{R}_x(0) = \mathbf{E}[\mathbf{x}(k)\mathbf{x}(k)]^T = \mathbf{A}\mathbf{R}_s(0)\mathbf{A}^T + \sigma_v^2 \mathbf{I} \tag{3.9}$$

$$\mathbf{R}_x(\tau) = \mathbf{E}[\mathbf{x}(k)\mathbf{x}(k-\tau)]^T = \mathbf{A}\mathbf{R}_s(\tau)\mathbf{A}^T \tag{3.10}$$

No caso de $m > n$, a variância do ruído pode ser estimada a partir dos menores valores singulares de $\mathbf{R}_x(0)$ ou, a partir da média dos menores $m-n$ valores singulares de $\mathbf{R}_x(0)$. Assim, o cálculo da estimativa do ruído não é confiável e apresenta maior complexidade quando o valor da relação sinal-ruído (SNR) é pequena.

Assim, a matriz de separação, \mathbf{W} , pode ser identificada, considerando as ambigüidades inerentes de escala e permutação, por exemplo, através da diagonalização simultânea implementada pela autodecomposição de $\mathbf{R}_x(0)$ e $\mathbf{R}_x(\tau)$, desde que $\mathbf{R}_s^{-1}(0)\mathbf{R}_s(\tau)$ tenham elementos distintos em sua diagonal. Uma alternativa para a utilização de autodecomposição é a aplicação de diagonalização conjunta aproximada.

Com base nas afirmações anteriores, nas seções 3.4 e 3.5 serão apresentados os principais algoritmos baseados em diagonalizações simultâneas e diagonalização conjunta aproximada. Antes, porém, na seção 3.3, será descrita a principal técnica de pré-processamento utilizada por estes algoritmos: o branqueamento ou descorrelação espacial.

3.3 – Pré-Processamento: Branqueamento ou Descorrelação Espacial

Antes da descrição dos algoritmos é importante descrever uma das técnicas mais utilizadas de pré-processamento - o branqueamento (*whitening*) ou descorrelação espacial - uma vez que a maioria dos algoritmos para identificação ou separação cega utiliza este pré-processamento, também conhecido como esferamento (*sphering*). A Figura 3.2 ilustra a transformação realizada pelo branqueamento nos dados obtidos pelos sensores.

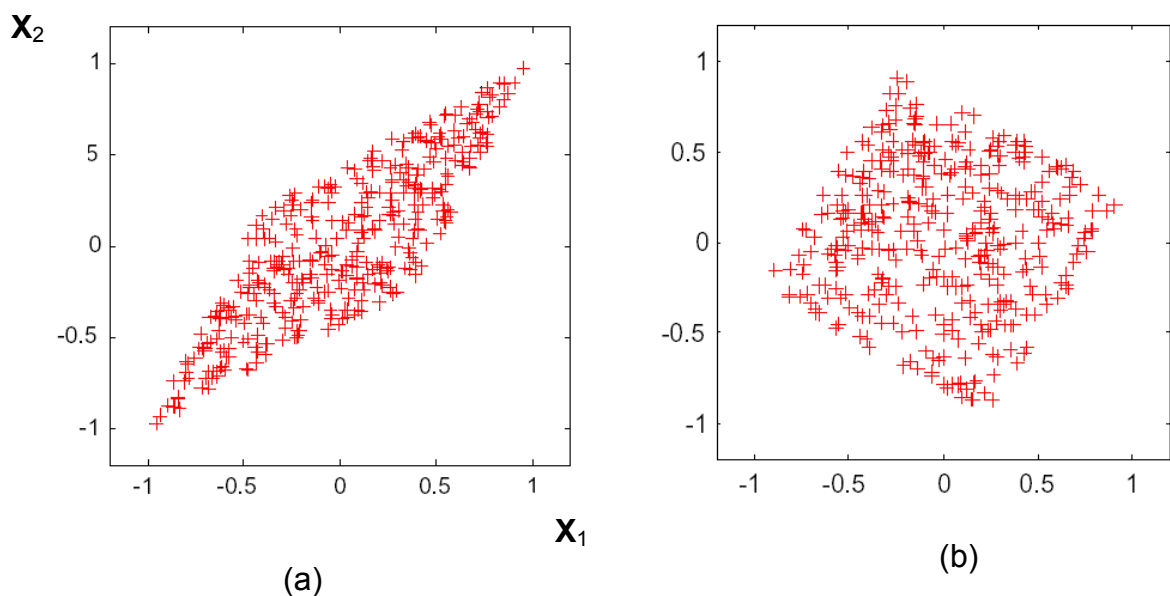


Figura 3.2 – Ilustração do branqueamento de dois sinais x_1 e x_2 :

(a) sinais obtidos pelos sensores (b) sinais branqueados.

O branqueamento é, na maioria dos casos, o primeiro passo para se implementar a separação cega de fontes e consiste tornar branco, o vetor de sinais medidos pelos sensores, $\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k)$. O branqueamento dos sinais é obtido através de uma matriz de branqueamento $\mathbf{Q}_{n \times m}$, tal que:

$$\mathbf{E}[\mathbf{Q}\mathbf{x}(k)\mathbf{x}(k)^T\mathbf{Q}^T] = \mathbf{Q}\mathbf{R}_x(0)\mathbf{Q}^T = \mathbf{Q}\mathbf{A}\mathbf{A}^T\mathbf{Q}^T = \mathbf{I}_n \quad (3.11)$$

Onde \mathbf{I} é uma matriz identidade.

Na equação (3.9) se \mathbf{Q} é uma matriz de branqueamento, então, $\mathbf{Q}\mathbf{A}$ é uma matriz unitária. Segue-se, pois, que para qualquer matriz de branqueamento \mathbf{Q} , existe uma matriz unitária \mathbf{U} , tal que:

$$\mathbf{Q}\mathbf{A} = \mathbf{U} \quad (3.12)$$

Como consequência de (3.12), a matriz de mistura \mathbf{A} pode ser escrita como:

$$\mathbf{A} = \mathbf{Q}^+\mathbf{U} \quad (3.13)$$

Onde \mathbf{Q}^+ indica pseudo-inversa de \mathbf{Q} , que nesta dissertação pode ser substituída pela inversa de \mathbf{Q} .

Nota-se que quando $m > n$, ou seja, número de sensores é maior que o número de fontes, o processo de branqueamento reduz a determinação de uma matriz $\mathbf{A}_{m \times n}$ a determinação de uma matriz unitária $\mathbf{U}_{n \times n}$; ou seja, há uma redução da dimensão do vetor de saída de m -dimensional para n -dimensional.

No branqueamento a matriz \mathbf{Q} faz com que a matriz de covariância dos dados de saída, $\mathbf{z}(k) = \mathbf{Q}\mathbf{x}(k)$, seja tal que $\mathbf{E}[\mathbf{z}(k)\mathbf{z}(k)^T] = \mathbf{I}$. Desta forma, os componentes do vetor branqueado $\mathbf{z}(k)$ são mutuamente descorrelacionados e tem covariância unitária.

É importante salientar também que o vetor resultante do branqueamento $\mathbf{z}(k) = \mathbf{Q}\mathbf{x}(k)$ conserva o modelo linear para separação, ou seja:

$$\begin{aligned} \mathbf{z}(k) &= \mathbf{Q}[\mathbf{x}(k)\mathbf{v}(k)] = \mathbf{Q}[\mathbf{A}\mathbf{s}(k) + \mathbf{v}(k)] = \mathbf{Q}\mathbf{A}\mathbf{s}(k) + \mathbf{Q}\mathbf{v}(k) \\ \mathbf{z}(k) &= \mathbf{U}\mathbf{s}(k) + \mathbf{Q}\mathbf{v}(k) \end{aligned} \quad (3.14)$$

A equação (3.14) corresponde, portanto, ao modelo branqueado para a separação de fontes (Belouchrani, 1997).

Para o cálculo de $\mathbf{Q}_{n \times m}$, deve-se ter em mente que esta matriz não é única, visto que, se ela for multiplicada por uma matriz ortogonal arbitrária, a relação em (3.9) é preservada (Cichocki, 2002).

É importante observar ainda que toda informação de segunda ordem é explorada pelo processo de branqueamento, uma vez que, substituindo \mathbf{U} por qualquer matriz unitária, a covariância de $\mathbf{z}(k)$ fica inalterada.

Como a matriz de covariância de $\mathbf{x}(k)$ é, na maioria das aplicações, simétrica e com elementos positivos, ela pode ser decomposta em seus autovetores e autovalores como mostrado em (3.15).

$$\mathbf{R}_x(0) = \mathbf{V}_x \mathbf{D}_x \mathbf{V}_x^T = \mathbf{V}_x \mathbf{D}_x^{1/2} \mathbf{D}_x^{1/2} \mathbf{V}_x^T \quad (3.15)$$

Onde \mathbf{V}_x é uma matriz ortogonal e $\mathbf{D}_x = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_n]$ é uma matriz com autovalores positivos, ou seja, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$.

Assim, sob a condição de que a matriz de covariância é positiva, a matriz de branqueamento \mathbf{Q} pode ser calculada como:

$$\mathbf{Q} = \mathbf{D}_x^{-1/2} \mathbf{V}_x^T \quad (3.16)$$

Ou de forma geral,

$$\mathbf{Q} = \mathbf{U} \mathbf{D}_x^{-1/2} \mathbf{V}_x^T \quad (3.17)$$

Onde \mathbf{U} é uma matriz ortogonal arbitrária e \mathbf{V}_x^T é a transposta de \mathbf{V}_x .

Outro método muito utilizado de branqueamento considera matrizes de correlação atrasadas no tempo, ou seja, ao invés de trabalhar com $\mathbf{R}_x^{-1/2}(0)$, trabalha-se com $\mathbf{R}_x^{-1/2}(\tau)$, que branqueia os dados $\mathbf{x}(k)$, sem considerar o efeito do ruído $\mathbf{v}(k)$. Vários algoritmos utilizam este tipo de branqueamento, como por exemplo, os propostos por Belouchrani et al. (2000b) e Choi et al. (2000a).

Para sinais corrompidos por ruído $\mathbf{v}(k)$ deve-se aplicar um método de branqueamento robusto, como o apresentado em (Belouchrani, 2000b).

Um outro método de branqueamento robusto para sinais corrompidos por ruído é apresentado em (Cichoki, 2002). Utilizando a técnica de subespaço, estima-se a variância do ruído σ_v^2 e o número de fontes para o caso em que a matriz de covariância do ruído é definida como $\mathbf{R}_v(0) = \sigma_v^2 \mathbf{I}_m$ e a variância do ruído é pequena, ou seja, a relação sinal ruído é relativamente alta. A seguir, de forma simplificada, são apresentados os passos para implementação deste algoritmo de branqueamento robusto:

- Inicialmente, deve-se calcular a matriz de covariância $\mathbf{R}_x(0) = \mathbf{A}\mathbf{R}_s(0)^T + \sigma_v^2 \mathbf{I}_m = \mathbf{A}\mathbf{A}^T + \sigma_v^2 \mathbf{I}_m$, que contém os sinais fontes e o ruído branco.
- $\mathbf{R}_x(0)$ é então decomposta em seus autovalores, isto é:

$$\begin{aligned} \mathbf{R}_x(0) &= \mathbf{V}_x \mathbf{D}_x \mathbf{V}_x^T = [\mathbf{V}_s, \mathbf{V}_v] \begin{bmatrix} \mathbf{D}_s & 0 \\ 0 & \mathbf{D}_v \end{bmatrix} [\mathbf{V}_s, \mathbf{V}_v]^T \\ \mathbf{R}_x(0) &= \mathbf{V}_s \mathbf{D}_s \mathbf{V}_s^T + \mathbf{V}_v \mathbf{D}_v \mathbf{V}_v^T \end{aligned} \quad (3.18)$$

Onde \mathbf{V}_s , contém os autovetores associados com os n autovalores principais da diagonal \mathbf{D}_s em ordem decrescente. \mathbf{V}_v contém os $m-n$ autovetores de ruído que correspondem aos autovalores de ruído \mathbf{D}_v .

- Estima-se a seguir a variância do ruído σ_v^2 calculando o valor médio dos $m-n$ menores autovalores e a ordem da matriz \mathbf{A} , ou utilizando algum outro método descrito na literatura. Finalmente, define-se a matriz de branqueamento como:

$$\mathbf{Q} = \mathbf{D}_s^{-1/2} \mathbf{V}_s^T = (\mathbf{D}_s - \hat{\sigma}_v^2 \mathbf{I}_n)^{-1/2} \mathbf{V}_s^T \quad (3.19)$$

3.3.1 – Relação entre Separação Cega de Fontes e Branqueamento

As técnicas de pré-processamento, em especial de branqueamento, podem ser utilizadas em algoritmos para separação cega de sinais.

Considerando que \mathbf{Q} é uma matriz simétrica, seu valor ótimo (\mathbf{Q}^*) é:

$$\mathbf{Q}^* = \mathbf{R}_x^{1/2} = \mathbf{V}_x \mathbf{D}_x^{-1/2} \mathbf{V}_x^T \quad (3.20)$$

Onde \mathbf{V}_x é uma matriz ortogonal, \mathbf{D}_x é uma matriz diagonal obtida pela decomposição em autovalores da matriz de covariância \mathbf{R}_x . Isto significa que os sinais de saída do sistema de separação $\mathbf{y}(k)$ são mutuamente ortogonais com variância unitária. Em geral, porém, a decorrelação espacial não é suficiente para realizar a separação de fontes a partir de uma mistura linear (Cichocki, 2002).

Por outro lado, para o caso especial em que a matriz de mistura \mathbf{A} é não singular e simétrica, a decorrelação espacial cega (branqueamento) com uma matriz de branqueamento \mathbf{Q} pode realizar diretamente a separação de fontes, desde que $\mathbf{R}_x = \mathbf{A}^2$, sob a condição de que $\mathbf{R}_s = \mathbf{I}$, ou seja, as fontes são espacialmente decorrelacionadas e tem variância unitária. Neste caso, tem-se então:

$$\mathbf{W} = \mathbf{Q}^* = \hat{\mathbf{A}}^{-1} = \hat{\mathbf{R}}_{xx}^{-1/2} = \mathbf{V}_x \mathbf{D}_x^{-1/2} \mathbf{V}_x^T \quad (3.21)$$

3.4 – Separação Cega Baseada em Diagonalizações Simultâneas

Nesta seção será discutido como a separação cega pode ser implementada através de diagonalizações simultâneas (SD – *Simultaneous Diagonalization*) que são implementadas por meio de decomposição em autovalores (EVD – *EigenValue Decomposition*), decomposição em autovalores generalizada (GEVD) ou decomposição em valores singulares (SVD – *Singular Value Decomposition*).

3.4.1 – Princípios Básicos

A partir das equações (3.9) e (3.10) pode-se determinar a matriz de separação \mathbf{W} considerando as ambigüidades inerentes de escala e permutação, através da diagonalização simultânea de duas matrizes de covariância $\mathbf{R}_x(0)$ e $\mathbf{R}_x(\tau)$, sendo que, em geral, pode-se

encontrar uma transformação linear que simultaneamente diagonalize estas duas matrizes (Hyvärinen, 2001).

Para simplificar, a diagonalização simultânea será apresentada para o caso em $m=n$, ou seja, número de sensores igual ao número de fontes, e não existindo ruído adicionado à mistura.

As técnicas de diagonalização simultânea trabalham com a matrizes de correlação $\mathbf{R}_x(0)$ e matrizes de correlação atrasadas no tempo $\mathbf{R}_x(\tau)$ consistindo basicamente de dois passos: branqueamento seguido por uma transformação unitária, ou seja:

(1) Branquear a matriz de covariância $\mathbf{R}_x(0)$, através de:

$$\mathbf{z}(k) = \mathbf{D}_x^{-1/2} \mathbf{V}_x^T \mathbf{x}(k) = \mathbf{Q} \mathbf{x}(k) \quad (3.22)$$

Onde \mathbf{D}_x e \mathbf{V}_x são os obtidos a partir da decomposição em autovalores (EVD) da matriz $\mathbf{R}_x(0)$, tal que:

$$\mathbf{R}_x(0) = \mathbf{V}_x \mathbf{D}_x \mathbf{V}_x^T \quad (3.23)$$

Tem-se então que:

$$\begin{aligned} \mathbf{R}_z(0) &= \mathbf{D}_x^{-1/2} \mathbf{V}_x^T \mathbf{R}_x(0) \mathbf{V}_x \mathbf{D}_x^{-1/2} = \mathbf{I} \\ \mathbf{R}_z(\tau) &= \mathbf{D}_x^{-1/2} \mathbf{V}_x^T \mathbf{R}_x(\tau) \mathbf{V}_x \mathbf{D}_x^{-1/2} \end{aligned} \quad (3.24)$$

(2) Aplicar uma transformação unitária para diagonalizar a matriz $\mathbf{R}_z(\tau)$, sendo que esta diagonalização é obtida através da autodecomposição de $\mathbf{R}_z(\tau)$, tal que:

$$\mathbf{R}_z(0) = \mathbf{V}_z \mathbf{D}_z \mathbf{V}_z^T \quad (3.25)$$

Assim, os sinais de saída estimados,

$$\mathbf{y}(k) = \mathbf{V}_z^{-1/2} \mathbf{z}(k) \quad (3.26)$$

satisfarão a relação:

$$\begin{aligned}\mathbf{R}_y(0) &= \mathbf{V}_z^T \mathbf{R}_z(0) \mathbf{V}_z = \mathbf{I} \\ \mathbf{R}_y(\tau) &= \mathbf{V}_z^T \mathbf{R}_z(\tau) \mathbf{V}_z = \mathbf{D}_z\end{aligned}\quad (3.27)$$

Isto posto, ambas as matrizes de correlação $\mathbf{R}_x(0)$ e de correlação atrasadas no tempo $\mathbf{R}_x(\tau)$ são simultaneamente diagonalizadas por uma transformação linear \mathbf{W} , tal que:

$$\mathbf{W} = \mathbf{U}_z^T \mathbf{D}_z^{-1/2} \mathbf{U}_x^T \quad (3.28)$$

Sendo que \mathbf{W} é uma matriz de separação válida, desde que todos os elementos da diagonal \mathbf{D}_z sejam distintos (Choi, 2001).

É importante salientar que ao invés da decomposição em autovalores padrão (EVD) pode-se utilizar a GEVD ou a SVD conforme apresentado em (Cichocki, 2002).

As equações (3.22) - (3.28) serviram de base para o desenvolvimento de algoritmos como o **AMUSE** (*Algorithm for Multiple Unknown Signals Extraction*), inicialmente proposto por Tong et al. (1990) e modificado e otimizado em (Cichocki, 2002). Na seção 3.5.2 será descrita uma versão modificada e otimizada do AMUSE conforme proposto em (Cichocki, 2002).

3.4.2 – AMUSE - Algoritmo para Extração de Múltiplos Sinais Desconhecidos

Motivado pela deficiência de algoritmos que não podem lidar com fontes gaussianas - uma vez que os momentos de quarta ordem de sinais gaussianos são completamente especificados por momentos de segunda ordem - desenvolveu-se o algoritmo **AMUSE**, que explora as estatísticas de segunda ordem das fontes originais.

De forma sumária, os passos para implementação do AMUSE são:

- Estimar a matriz de correlação $\mathbf{R}_x(0)$ dos sinais dos sensores $\mathbf{x}(k)$;

$$\mathbf{R}_x(0) = E[\mathbf{x}(0)\mathbf{x}(0)^T] = \frac{1}{N} \sum_{k=1}^n \mathbf{x}(k)\mathbf{x}^T(k) \quad (3.29)$$

- Calcular a SVD ou ainda decomposição em autovalores (EVD) da matriz $\mathbf{R}_x(0)$;

$$\mathbf{R}_x(0) = \mathbf{U}_x \sum_x \mathbf{V}_x \mathbf{V}_x^T = \mathbf{V}_x \mathbf{\Lambda}_x \mathbf{V}_x^T \quad (3.30)$$

Onde $\mathbf{V}_s = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$ contém os autovalores associados com os autovalores de $\mathbf{\Lambda}_s = \text{diag}[\lambda_1 \geq \lambda_2 \dots \geq \lambda_n]$ em ordem decendente. Se for considerado ruído aditivo, os últimos $m-n$ autovalores não significativos correspondem ao ruído e os primeiros autovalores significativos correspondem ao sinal mais ruído. Deve-se, pois, estimar o número n de fontes a partir dos dados mais significantes.

- Caso seja considerado ruído aditivo, estimar a variância do ruído branco como a média dos valores de $m-n$ autovalores menos significativos;
- Realizar o branqueamento do vetor $\mathbf{x}(k)$ através da matriz de branqueamento \mathbf{Q} , de forma a se obter o vetor branco:

$$\mathbf{z}(k) = \mathbf{Q}\mathbf{x}(k) \quad (3.31)$$

- Selecionar um valor de atraso $\tau \neq 0$ e estimar a matriz de covariância atrasada do vetor branqueado, $\mathbf{z}(k)$;
- Calcular a SVD da matriz de covariância atrasada $\mathbf{R}_z(\tau)$, tal que:

$$\mathbf{R}_z(\tau) = \frac{1}{N} \sum_{k=1}^N \mathbf{z}(k) \mathbf{z}^T(k-\tau) = \mathbf{U}_z \mathbf{D}_z \mathbf{V}_z^T; \quad (3.32)$$

- Estimar os vetores originais:

$$\mathbf{y}(k) = \mathbf{U}_z^T \mathbf{z}(k); \quad (3.33)$$

Em geral, para um atraso $\tau=1$, este algoritmo separa com sucesso fontes coloridas com diferentes espectros de potência, considerando ainda um número de sensores maior ou igual ao número de fontes. Por outro lado, ele apresenta como principal desvantagem o fato que sua precisão é deteriorada quando da presença de ruído aditivo (Cichoki, 2002).

O algoritmo **AMUSE** para separação cega de fontes coloridas pode ser facilmente adaptado para Análise de Componentes Independentes (ICA) de fontes não Gaussianas se, ao

invés de da matriz de correlação com atraso $\mathbf{R}_x(\tau)$, for utilizada a matriz de quadricovariância, cuja decomposição em autovalores é dada por:

$$\mathbf{C}\bar{\mathbf{x}}(\mathbf{E}) = \mathbf{U}\mathbf{\Lambda}_E\mathbf{U}^T = \widehat{\mathbf{A}}\mathbf{C}_s(\mathbf{E})\widehat{\mathbf{A}}^T \quad (3.34)$$

Onde, em geral, \mathbf{E} é tipicamente igual à matriz identidade, com:

$$\mathbf{\Lambda}_E = \text{diag}\{\lambda_1 \mathbf{u}_1^T \mathbf{E} \mathbf{u}_1, \dots, \lambda_n \mathbf{u}_n^T \mathbf{E} \mathbf{u}_n\} \quad (3.35)$$

e

$$\lambda_i = k_4(s_i) = E\{s_i^4\} - 3E^2\{s_i^2\} \quad (3.36)$$

Onde λ_i é o *kurtosis* (Papoulis, 1991) da i -ésima fonte e \mathbf{u}_i é a i -ésima coluna da matriz de autovetores \mathbf{U} .

Assim, se $\mathbf{C}\bar{\mathbf{x}}$ é único, ou seja, todos os autovalores de $\mathbf{\Lambda}_E$ são distintos, pode-se estimar a matriz de mistura $\mathbf{W} = \hat{\mathbf{A}} = \mathbf{Q}\mathbf{H} = \mathbf{U}$. Para $\mathbf{E}=\mathbf{I}$ estas condições são satisfeitas se os sinais fontes têm diferentes valores de *kurtosis*, estes procedimentos levam ao algoritmo denominado **FOBI** (*Forth Order Blind Identification*) (Cardoso, 1989).

Informações mais detalhadas e simulações computacionais realizadas utilizando o AMUSE serão apresentadas no Capítulo 5.

3.4.3 – Considerações Adicionais

De forma geral, pode-se dizer que os algoritmos que utilizam EVD/SVD e GEVD são os mais simples para a realização de identificação e separação cega de fontes com estrutura temporal, sendo úteis quando se trabalha com um número reduzido de amostras, uma vez que a matriz de covariância – informação de segunda ordem - pode ser estimada com precisão a partir de um número de dados menor do que os usados em estatísticas de ordem superior. Por outro lado, os algoritmos que utilizam estatísticas de segunda ordem apresentam, em geral, baixa robustez com relação a ruído e desempenho.

Outros algoritmos baseados em diagonalizações simultâneas podem ser encontrados em (Cichocki, 2002; Chang, 2000).

3.5 – Separação Cega Baseada em Diagonalização Conjunta

Nesta seção será discutido como a separação cega pode ser implementada através de diagonalização conjunta. Com este objetivo, em 3.5.1 serão definidas diagonalização conjunta e diagonalização conjunta aproximada e em 3.5.2 serão apresentados alguns algoritmos que utilizam estes métodos.

3.5.1 – Diagonalização Conjunta Aproximada

Uma alternativa ao cálculo da EVD / SVD / GEVD é a utilização da diagonalização conjunta (JD – *Joint Diagonalization*) como descrita em (Cardoso, 1993; Cardoso, 1996b; Belouchnari, 1997). O objetivo desta técnica é encontrar uma matriz \mathbf{U} que diagonalize um dado conjunto de matrizes, tal que:

$$\mathbf{M}_i = \mathbf{U} \mathbf{D}_i \mathbf{U}^T \quad (3.37)$$

Onde \mathbf{M}_i é um conjunto de matrizes, por exemplo, matrizes de covariância e/ou matrizes de cumulantes; \mathbf{D}_i são matrizes diagonais.

Esta abordagem procura ainda reduzir a probabilidade de uma escolha incorreta do valor do atraso (τ) possa impedir a determinação dos sinais de saída. Além disso, com este método há, em geral, um aumento da eficiência estatística, uma vez que \mathbf{U} é determinada a partir de um grande conjunto de matrizes.

Assim, com o objetivo de encontrar uma matriz que diagonalize um dado conjunto de matrizes faz-se uso de um operador numérico denominado **off**(.). O **off**(.) de uma determinada matriz $\mathbf{M}_{n \times n}$ é definido por:

$$\mathbf{off}(\mathbf{M}) \stackrel{def}{=} \sum_{1 \leq i \neq j \leq n} |M_{ij}|^2 \quad (3.38)$$

Onde M_{ij} são os elementos da matriz \mathbf{M} .

A diagonalização unitária de \mathbf{M} é, então, obtida quando **off**($\mathbf{V}^T \mathbf{M} \mathbf{V}$) é zerado por uma matriz unitária \mathbf{V} . Como descrito em (Belouchnari, 1997), somente matrizes normais podem ser diagonalizadas por uma matriz unitária. Além disso, se a matriz \mathbf{M} pode ser decomposta

como $\mathbf{M}=\mathbf{U}\mathbf{D}\mathbf{U}^T$, onde \mathbf{U} é unitária e \mathbf{D} é diagonal, com elementos da diagonal distintos, então \mathbf{M} pode ser diagonalizada somente por uma matriz que seja essencialmente igual a \mathbf{U} , ou seja, se $\text{off}(\mathbf{V}^T\mathbf{M}\mathbf{V})=0$, então \mathbf{V} é essencialmente igual a \mathbf{U} .

Considerando um conjunto de matrizes $\mathbf{M} = \mathbf{M}_i = [\mathbf{M}_1, \dots, \mathbf{M}_L]$ com L matrizes $n \times n$, o critério para diagonalização conjunta (JD) é definido para qualquer matriz \mathbf{V} , tal que:

$$\mathcal{C}(\mathbf{M}, \mathbf{V}) = \sum_{p=1, L} \text{off}(\mathbf{V}^T \mathbf{M}_L \mathbf{V}) \quad (3.39)$$

Uma matriz unitária é dita diagonalizadora conjunta das matrizes de \mathbf{M} , se, e somente se, ela minimizar a função (3.39) em relação ao conjunto de todas as matrizes unitárias (Belouchrani, 1997). Considerando o caso no qual cada matriz do conjunto \mathbf{M} está na forma:

$$\mathbf{M}_k = \mathbf{U}\mathbf{D}_k\mathbf{U}^T \quad (3.40)$$

Onde \mathbf{D}_k é uma matriz diagonal, então, $\mathcal{C}(\mathbf{M}, \mathbf{U})=0$ e este será o mínimo global de (3.39) desde que $\mathcal{C}(\mathbf{M}, \mathbf{V}) \geq 0$ para qualquer matriz \mathbf{V} . Deste modo, se cada uma das matrizes de \mathbf{M} puder ser diagonalizada pela matriz unitária \mathbf{U} , então \mathbf{U} é um diagonalizador conjunto de \mathbf{M} .

Mais importante do que definir diagonalizador conjunto é determinar a singularidade da matriz diagonalizadora, a qual é definida da seguinte forma:

Sendo $\mathbf{M} = [\mathbf{M}_1, \dots, \mathbf{M}_L]$ um conjunto de L matrizes, tal que, para $1 \leq l \leq L$, a matriz \mathbf{M}_L , pode ser escrita como $\mathbf{M}_l = \mathbf{U}\mathbf{D}_l\mathbf{U}^T$, sendo \mathbf{U} uma matriz unitária e $\mathbf{D}_p = \text{diag}[D_1(l), \dots, D_n(l)]$. Qualquer matriz diagonalizadora de \mathbf{M} é essencialmente igual a \mathbf{U} se, e somente se:

$$\forall 1 \leq i \neq j \leq n \quad \exists \quad p, 1 \leq l \leq L \quad D_i(l) \neq D_j(l) \quad (3.41)$$

É importante salientar que a restrição (3.41) é muito mais severa do que a (3.39). Uma característica importante desta definição de diagonalização conjunta é que ela não requer que o conjunto de matrizes \mathbf{M} , sob estudo, seja diagonalizado simultaneamente por uma única matriz unitária. De fato, não é nem necessário que as matrizes do conjunto \mathbf{M} sejam

individualmente diagonalizadas por uma matriz unitária. Isto ocorre uma vez que não é necessário que o $\text{off}(\cdot)$ de todas as matrizes de \mathbf{M} sejam iguais a zero para que uma transformação unitária seja implementada. O objetivo do diagonalizador conjunto é simplesmente minimizar o critério definido em (3.39).

Se as matrizes de \mathbf{M} não estiverem de acordo com (3.41), o critério JD não pode ser zerado e as matrizes podem somente ser diagonalizadas conjuntamente de forma **aproximada** e defini-se, então, a Diagonalização Conjunta Aproximada (JAD – *Joint Approximate Diagonalization*). Assim, ao invés da diagonalização exata, a diagonalização aproximada, permite que a informação contida no conjunto de matrizes seja integrada em uma única matriz unitária.

Uma importante característica da diagonalização conjunta aproximada é a existência de diversos métodos matemáticos que podem ser utilizados para seu cálculo incluindo, técnicas Jacobianas, ALS (*Alternative Least Square*), PARAFAC (*Parallel Factor Analysis*), entre outros, como mostrado em (Belouchrani, 1997; Cichocki, 2002).

Um dos métodos mais conhecidos para a implementação da diagonalização conjunta aproximada é baseado na técnica de Jacobi (Golub, 1989). Esta técnica consiste em se calcular o diagonalizador unitário \mathbf{U} como um produto das rotações de Givens. Os parâmetros das rotações de Givens podem ser calculados de forma simples, mesmo quando as matrizes a serem diagonalizadas conjuntamente não apresentam qualquer propriedade de simetria. Esta característica é importante quando são processadas matrizes de covariância amostradas que não são exatamente normais.

Outra importante observação com relação à Diagonalização Conjunta Aproximada é que se ao invés de utilizar matrizes de covariância forem empregadas, por exemplo, matrizes de cumulantes de ordem superior, pode-se obter o algoritmo de Diagonalização Conjunta Aproximada de Automatrizes (JADE) para ICA (Cardoso, 1993).

3.5.2 – Algoritmos Baseados em Diagonalização Conjunta

Os dois principais algoritmos que utilizam a técnica de diagonalização conjunta são o **SOBI** (*Second Order Blind Identification*) (Belouchrani, 1997; Belouchrani, 1993) e o **SONS** (*Second Order Nonstationary Source separation*) (Choi, 2000a; Choi, 2001).

O algoritmo SOBI baseia-se na diagonalização conjunta aproximada de múltiplas matrizes de correlação com atrasos. Dado um vetor branqueado $\mathbf{z}(k)$, toma-se um conjunto de

matrizes de correlação com atraso de $\mathbf{z}(k)$, $\mathbf{R}_z(\tau)$, $i=1\dots k$. O objetivo do algoritmo SOBI é encontrar uma transformação unitária \mathbf{V} , tal que:

$$\mathbf{V}^T \mathbf{R}_z(\tau_i) \mathbf{V} = \mathbf{D}_i, \quad (3.42)$$

Para $i=1,\dots,k$, onde \mathbf{D}_i é um conjunto de matrizes diagonais.

Como o SOBI trabalha com várias matrizes de correlação, ele reduz a probabilidade de que uma escolha incorreta do atraso τ impeça a implementação da separação cega.

De forma simplificada os passos para implementação do SOBI, são:

- (1) Estimar a matriz de covariância amostral $\mathbf{R}_x(0)$ a partir de n amostras;
- (2) Decompor $\mathbf{R}_x(0)$ em $\lambda_1, \dots, \lambda_n$, os n maiores autovalores e h_1, \dots, h_n , autovetores;
- (3) Caso seja considerado ruído branco adicionado ao processo, uma estimativa da variância do ruído σ_v^2 pode ser feita pela média dos $m-n$ autovalores de $\mathbf{R}_x(0)$;

- (4) Calcular a matriz de branqueamento como, por exemplo:

$$\mathbf{Q} = [(\lambda_1 - \sigma_v^2)^{-1/2} h_1, \dots, (\lambda_n - \sigma_v^2)^{-1/2} h_n]^T \quad (3.43)$$

- (5) Branquear os dados, determinando o vetor $\mathbf{z}(k)$;
- (6) Formar a matriz de correlação com atraso $\mathbf{R}_z(\tau)$ através de amostras das matrizes de covariância de $\mathbf{z}(k)$ para um conjunto fixo de atrasos τ ;

- (7) Obter a matriz unitária \mathbf{U} , que é a diagonalizadora conjunta de $\{\mathbf{R}_z(\tau_j) | j=1, \dots, F\}$

- (8) Estimar os sinais de saída conforme (3.44).

$$\mathbf{y}(k) = \mathbf{U}^T \mathbf{Q} \mathbf{x}(k) \quad (3.44)$$

e/ou a matriz de mistura

$$\mathbf{A} = \mathbf{Q}^+ \mathbf{U} \quad (3.45)$$

Experimentos mostraram a utilidade de se utilizar várias matrizes de covariância em ambientes onde há baixa relação sinal ruído e/ou fontes com pequena diferença espectral.

Em (Belouchrani, 2000) é apresentada uma alteração do algoritmo de SOBI, denominado **R-SOBI** (SOBI Robusto); este algoritmo utiliza uma técnica robusta de branqueamento, especialmente útil quando da presença de ruído aditivo.

O algoritmo SOBI convencional estima a matriz de branqueamento a partir de uma matriz de correlação com um atraso (τ) diferente de zero. Como não há garantias de que a matriz será positiva, uma escolha incorreta de τ , pode comprometer o processo de branqueamento. Para solucionar este problema, o R-SOBI utiliza uma combinação linear positiva do conjunto de matrizes de correlação, considerando atrasos diferentes de zero para o cálculo da matriz de branqueamento. Os coeficientes desta combinação são calculados, por exemplo, utilizando um algoritmo de convergência global.

Outro algoritmo também muito utilizado para separação cega, baseado em estatísticas de segunda ordem, é o **SONS** ou **SEONS** (*Second Order Nonstationary Source separation*) desenvolvido por Choi et al (Choi, 2000a).

Este algoritmo nada mais é do que uma modificação do algoritmo SOBI para fontes não-estacionárias que explora, ao mesmo tempo, a não estacionaridade de segunda ordem e a estrutura temporal das fontes, considerando que o ruído é branco ou que as interferências e o ruído são sinais estacionários. Neste sentido pode-se dizer que o SONS inclui o SOBI como um caso especial.

A seguir, de forma sumária, são apresentados os passos para implementação do SONS:

- (1) Empregar um dos métodos de branqueamento robusto e utilizá-lo para calcular o vetor branqueado $\mathbf{z}(k)=\mathbf{Q}\mathbf{x}(k)$;
- (2) Dividir os dados branqueados $\mathbf{z}(k)$ em B blocos não sobrepostos;
- (3) Calcular $\mathbf{M}_z(k_p, \tau_j)$, para $l=1, \dots, L$ e $j=1, \dots, J$, ou seja, para cada janela ou bloco, calcula-se J diferentes matrizes de correlação atrasadas de $\mathbf{z}(k)$;
- (4) Encontrar uma matriz diagonalizadora conjunta de $\mathbf{M}_z(k_l, \tau_j)$, utilizando o método de diagonalização conjunta aproximada;
- (5) Estimar os dados de saída e/ou a matriz de mistura.

O SONS tem apresentado um bom desempenho, principalmente em ambientes onde há presença de ruído aditivo. Para valores de SNR maiores ou iguais a 10dB seu desempenho é superior ao do algoritmo SOBI. Além disso, o SONS é aplicável às fontes não-estacionárias,

inclusive i.i.d não-estacionárias e temporariamente coloridas, enquanto que o SOBI, é hábil somente para separar ou extrair fontes coloridas.

3.6 – Comparação entre os Algoritmos de JAD e EVD

Embora a abordagem proposta pelos algoritmos de Diagonalização Conjunta Aproximada (JAD), em geral, apresentem um desempenho melhor, especialmente para dados corrompidos por ruído com baixa SNR, a abordagem de decomposição em autovalores (EVD) simétrica apresenta importantes vantagens (Cichocki, 2002):

- Ambos EVD e JAD são algoritmos para processamento em blocos (*batch*), no sentido que o conjunto dos dados de entrada ou blocos de dados são coletados e processados de uma vez. Porém, os algoritmos baseados em EVD têm uma complexidade computacional menor, pois processam somente uma diagonalização média, ao invés de processarem várias diagonalizações conjuntas das matrizes de dados simultaneamente. Pode-se dizer, então, que os algoritmos baseados em EVD têm complexidade computacional menor que os algoritmos baseados em JAD;
- A EVD permite que se controle explicitamente se a separação das fontes está sendo realizada corretamente através do monitoramento dos autovalores obtidos, que devem ser diferentes;
- Vários algoritmos eficientes vêm sendo desenvolvidos para EVD com alta velocidade de convergência.

Capítulo 4 - Análise de Componentes Independentes

4.1 – Introdução

Um dos principais e mais interessantes problemas nas áreas de estatística, análise de dados, processamento de sinais e redes neurais é encontrar uma representação adequada para dados multivariáveis. Por simplicidade computacional e conceitual, tal representação é feita por meio de uma transformação linear. Nesta transformação, cada componente é uma combinação linear das variáveis originais. Para encontrar esta transformação empregam-se métodos bem conhecidos como: Análise de Componentes Principais (PCA - *Principal Component Analysis*); Análise de Fator (FA - *Factor Analysis*) (Hyvärinen, 2001); Projeção de Busca (*Projection Pursuit*) (Cichocki, 2002; Schell, 1992; Hyvärinen, 2001); e a pouco mais de 15 anos, emprega-se o método de Análise de Componentes Independentes (ICA – *Independent Component Analysis*) (Cao, 1996; Comon, 1994; Hyvärinen, 2001; Kofidis, 2001; Cardoso, 1998).

O principal objetivo da ICA é encontrar uma representação linear de dados não gaussianos, minimizando a dependência estatística entre eles, de forma que os componentes resultantes sejam estatisticamente independentes, ou tão independentes, quanto possível. O que diferencia a ICA das outras técnicas existentes é justamente o fato dela trabalhar com componentes que são ao mesmo tempo não gaussianos e estatisticamente independentes. Esta representação pode ser usada para capturar a estrutura dos componentes, ou dados, em várias aplicações como extração de características ou separação de sinais.

Neste capítulo será realizada uma revisão do estado da arte dos princípios da Análise de Componentes Independentes, bem como uma revisão dos principais algoritmos que utilizam a ICA com o objetivo de realizar a separação cega de sinais. É apresentada ainda uma revisão de conceitos estatísticos indispensáveis para a compreensão do problema da ICA. Com esta finalidade este capítulo está organizado da seguinte forma: na seção 4.1 é feita uma introdução ao problema da ICA, na seção 4.2 é apresentada uma revisão histórica do estado da arte das técnicas que tratam da separação cega utilizando ICA, não levando em consideração os conceitos matemáticos associado a cada uma destas técnicas que serão apresentados em 4.3. A seção 4.4 apresenta as principais técnicas de pré-processamento utilizadas em ICA. A seção 4.5, por sua vez, discorre sobre as ferramentas para separação cega baseada em ICA, através de um visão baseada em princípios estatísticos e de teoria da informação. Em 4.6 são

apresentados, de forma sumária, os principais algoritmos para ICA baseados nas ferramentas descritas em 4.4 e para finalizar são feitas, na seção 4.7, algumas considerações sobre ICA na presença de ruído.

4.1.1 – Revisão Histórica

As pesquisas na área de processamento cego de fontes (BSP – *Blind Source Processing*) tiveram início na década de 80, principalmente na França onde surgiram os primeiros trabalhos na área.

Em abril de 1986, no encontro sobre Redes Neurais para Computação realizado em Snowbird (Utah, EUA) Jeanny Héroult e Christian Jutten apresentaram um artigo intitulado *Space or time adaptive signal processing by neural network models*, que descrevia uma rede neural recorrente e um algoritmo de treinamento baseado nas regras de Hebb, que segundo os autores era apta para realizar a separação cega de sinais independentes. Como exemplo, foi apresentada a separação de dois sinais misturados e foi mencionada a possibilidade de separação de sinais de estereoscópios com quatro sinais misturados. Este trabalho abriu as portas para uma nova área dentro do processamento de sinais (Hyvärinen, 2000).

Ao longo dos anos 80, as técnicas para ICA, estudadas principalmente por pesquisadores franceses, tiveram pouca influência no cenário internacional. Os poucos trabalhos apresentados em congressos de Redes Neurais em meados dos anos 80 foram ofuscados pelo grande número de trabalhos relacionados a redes de retropropagação, redes de Hopfield e Mapas Auto-Organizáveis de Kohonen (Haykin, 1998).

No final da década de 80, foram apresentados os primeiros artigos relacionados a ICA dos pesquisadores J. F. Cardoso (Cardoso, 1989) e Pierre Comon (Comon, 1989), que foram de grande importância para o avanço das técnicas de separação cega de fontes utilizando estatísticas de ordem superior, em especial cumulantes de quarta ordem, aliadas aos princípios da Teoria da Informação.

Desta forma, pode-se dizer que até meados dos anos 90 poucos trabalhos foram apresentados na área de ICA e os apresentados não atraíram de forma significativa o interesse da comunidade científica. Por isso, em muitos casos, o problema da separação cega foi considerado sem solução.

Em meados de 1994, entretanto, os pesquisadores da área de Processamento de Sinais parecem ter voltado sua atenção para os algoritmos não supervisionados. Em 1994, a estrutura geral apresentada por Jutten e Héroult (Jutten, 1991) foi revisada e aperfeiçoada por Comon

em (Comon, 1994). Neste trabalho, Comon desenvolveu uma teoria matemática sobre o problema da separação cega de fontes utilizando Análise de Componentes Independentes. Comon prova que o critério de independência estatística entre fontes originais é suficiente para a obtenção da separação das fontes sujeitas a misturas lineares. A definição de Análise de Componentes Independentes apresentada neste artigo depende de uma função custo, sendo que uma das funções propostas é obtida com base na Informação Mútua das observações.

Embora a aplicação da ICA à separação cega de fontes apresente algumas limitações, como, por exemplo, há necessidade de conhecer ou estimar a função densidade de probabilidade das fontes originais, o trabalho de Comon consagrou a Análise de Componentes Independentes como uma importante ferramenta para a separação cega de fontes.

Em 1995, A.J. Bell e T. J. Sejnowski (Bell, 1995) publicaram um artigo onde foi apresentado um algoritmo baseado no princípio de Maximização da Informação (INFOMAX) que atraiu as atenções para a área de ICA por colocar o problema da separação cega de fontes dentro de uma estrutura neural baseada em princípios de Teoria da Informação.

Em 1996, Tony Bell colaborou a distância com Te-Won Lee em seu trabalho para separação cega de fontes acústicas gravadas levando em consideração atrasos de tempo (*time delays*) (Lee, 1998).

Foi proposta ainda em 1996, por Cardoso e Laheld (Cardoso, 1996a), uma melhoria no algoritmo desenvolvido por Bell e Sejnowski (Bell, 1995), permitindo que este algoritmo se tornasse prático com aplicações em vários problemas reais. O trabalho de Cardoso e Laheld (Cardoso, 1996a) introduziu uma classe de algoritmos adaptativos, que implementam uma versão da estimativa de equivariância e foi denominado EASI (*Equivariant Adaptive Separation via Independence*). Esta classe de algoritmos é baseada na idéia de atualização serial (*serial updating*) da matriz de separação, que geram algoritmos com estruturas simples. Outro importante ponto deste algoritmo é que o seu desempenho independe da matriz de mistura, dependendo somente da distribuição normalizada das fontes.

Em 1996, S. I. Amari et al. (Amari, 1996), de forma independente, refinaram o algoritmo desenvolvido por Bell e Sejnowski utilizando gradiente natural, o que aumentou a velocidade de convergência do algoritmo, pois eliminou a inversão de matriz.

Um ano mais tarde, os pesquisadores Hyvärinen, Kahunen e Oja, da Universidade de Tecnologia Helsinque na Finlândia, apresentaram o algoritmo de ponto fixo para ICA ou FASTICA, que contribui para a utilização da ICA em diversas aplicações devido a sua eficiência computacional (Hyvärinen, 2000; Hyvärinen, 2001, Hyvärinen, 1997). Este

algoritmo foi proposto com base nas técnicas desenvolvidas pelos pesquisadores para estimar de forma simples a f.d.p. dos sinais originais a partir das amostras recebidas pelos sensores.

Outro algoritmo também de grande importância foi proposto por Te-Won Lee et al. (Lee, 1997), que desenvolveram uma versão generalizada ou estendida do INFOMAX, denominado *Extendend-Infomax* (ou E-Infomax), que estimava os momentos das fontes originais e chaveava o algoritmo de acordo com o tipo de fonte super ou sub-gaussiana. As principais aplicações deste algoritmo encontram-se na área biomédica, especialmente para o processamento de sinais de eletroencefalograma (EEG), onde o algoritmo tem se mostrado adequado para separação dos sinais de EEG de artefatos.

Em 1998, Cardoso em (Cardoso, 1998) apresentou uma importante revisão do ponto de vista estatístico, das principais técnicas que haviam sido desenvolvidas até o momento para solucionar o problema da separação cega de sinais utilizando ICA. Este trabalho teve grande repercussão e serve como uma boa introdução matemática ao problema. Outra boa introdução ao problema da ICA foi apresentada em 2000 por Hyvärinen e Oja em (Hyvärinen, 2000). Este trabalho descreve de forma simples os principais conceitos da ICA, bem como suas principais aplicações.

Em 1999, foi criado na Europa um projeto, não governamental, de pesquisa denominado BLISS (*Blind Source Separation and Applications*) (BLISS, 2003). Como o próprio nome sugere, o projeto é voltado para pesquisas em BSS e ICA com o objetivo de desenvolver trabalhos considerando misturas instantâneas e convolutivas, presença de ruído, métodos adaptativos e/ou não lineares. Os trabalhos desenvolvidos visam, em geral, aplicações nas áreas de telecomunicações e engenharia biomédica. Este projeto, ou consórcio, é composto atualmente por cinco países incluindo Portugal (INESC), Finlândia (Helsinki University of Technology), França (Institute National Polytechnique de Grenoble), Alemanha (GMD) e Canadá (McMaster University).

Portanto, desde meados dos anos 90, tem crescido o número de publicações e congressos destinados a ICA. O primeiro congresso internacional sobre ICA aconteceu em Janeiro de 1999 em Aussois, França e o segundo em Junho de 2000 em Helsinki, Finlândia. Estes congressos contaram com mais de 100 pesquisadores que trabalham em ICA e BSS e que contribuíram para transformar ICA em uma área de pesquisa bem definida.

Em 2003, a Conferência Internacional sobre Análise de Componentes Independentes e Separação Cega (ICA'2003) foi realizado de 1-4 de abril em Nara, Japão, atraindo pesquisadores de diferentes áreas de pesquisa, entre as quais se destacaram as áreas de engenharia biomédica, processamento de voz e comunicações móveis. Além das aplicações,

foram apresentadas dezenas de novos algoritmos ou versões modificadas dos algoritmos existentes (ICA, 2003). A 5ª Conferência Internacional sobre Análise de Componentes Independentes e Separação Cega (ICA'2004), será realizada em Granada, Espanha, de 22-24 de setembro (ICA, 2004).

Em junho de 2003, foi organizada em Berlim, pelo projeto BLISS, uma escola de verão que discutiu a teoria e as aplicações da ICA e BSS (ICA_SS, 2003). A escola contou com a participação de importantes pesquisadores na área como Luis Almeida (INESC, Portugal), J. F. Cardoso (ENST, França), Simon Haykin (McMaster, Canadá), Christian Jutten (INPG, França), T-W. Lee (Salk Institute, EUA), J. Karhunen e E. Oja (HUT, Finlândia) entre outros.

4.2 – Modelo e Definição de ICA

Com o objetivo de se estimar quantidades desconhecidas, sejam elas parâmetros, seqüências de dados ou sinais, a partir de medidas conhecidas (obtidas por meio de sensores) e de alguma restrição estatística sobre os dados originais, deve-se utilizar um modelo de representação que seja capaz de expressar a relação entre as grandezas desconhecidas.

O modelo mais utilizado é o modelo linear expresso, por:

$$\mathbf{x}(k) = \mathbf{A} \mathbf{s}(k) + \mathbf{v}(k) \quad (4.1)$$

Onde $\mathbf{x}(k)$ é o vetor das medidas realizadas pelos sensores e \mathbf{A} é a matriz de mistura. A matriz \mathbf{A} corresponde aos parâmetros de combinação dos sinais ou dados, sendo que os coeficientes ou pesos desta matriz são considerados constantes e desconhecidos, uma vez que não é possível determiná-los sem o conhecimento das propriedades do sistema de mistura. O vetor $\mathbf{s}(k)$ é aquele que contém as fontes sinais ou dados originais e $\mathbf{v}(k)$ corresponde ao ruído adicionado ao processo.

Um diagrama em blocos do modelo da Equação (4.1) é apresentado na Figura 4.1.

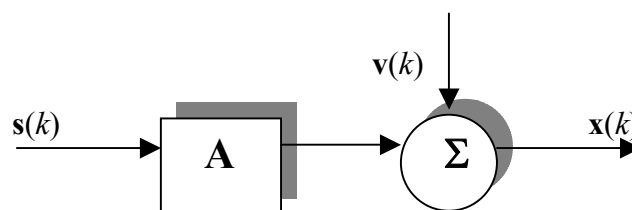


Figura 4.1 – Diagrama em blocos do modelo de mistura de sinais para ICA.

Devido a sua simplicidade, o modelo linear invariante no tempo, com fontes estacionárias, tem sido utilizado para equacionar o problema da Análise de Componentes Independentes.

Outras considerações feitas ao modelo da equação (4.1) para sua utilização em ICA são:

- $\mathbf{s}(k)$ e $\mathbf{v}(k)$ são estacionários e de média zero;
- Os componentes de $\mathbf{s}(k)$ são estatisticamente independentes;
- Os sinais de ruído aditivo $\mathbf{v}(k)$, são estatisticamente independentes entre si e independentes das fontes $\mathbf{s}(k)$;
- O número de sensores é maior ou igual ao número de fontes ($m \geq n$).

Assim, partindo do modelo linear (4.1) e do princípio de que as fontes originais são estatisticamente independentes, a ICA procura estimar um conjunto de fontes $\mathbf{y}(k)$ também independentes, através de um sistema de separação \mathbf{W} quando somente as medidas dos sensores $\mathbf{x}(k)$ são conhecidas (Hyvärinen, 2001; Kofidis, 2001; Comon, 1994). Para isso, a restrição estatística utilizada é que a função densidade de probabilidade das fontes sejam não-gaussianas, ou que no máximo uma das fontes tenha f.d.p. gaussiana. Esta restrição é utilizada, visto que, conforme descreve o Teorema do Limite Central (Papoulis, 1991), a soma de variáveis aleatórias gaussianas fornece uma distribuição de probabilidade conjunta também gaussiana, o que inviabiliza qualquer inferência sobre as fontes originais a partir das observações dos sensores, ou seja das misturas.

O modelo de separação utilizado na ICA pode, então, ser escrito como:

$$\mathbf{y}(k) = \hat{\mathbf{s}}(k) = \mathbf{W} \mathbf{x}(k) \quad (4.2)$$

Na realidade, as formulações das equações (4.1) e (4.2) não são diferentes, uma vez que \mathbf{W} pode ser considerada a inversa de \mathbf{A} .

A partir da equação (4.2) o problema passa a ser a determinação do sistema ou matriz de separação \mathbf{W} , tal que os componentes de $\mathbf{y}(k)$ sejam independentes ou tão independentes quanto possível.

Mais uma vez vale a pena frisar que este problema só será resolvido se, e somente se, as fontes originais forem consideradas não-gaussianas, pois só assim a restrição de

independência estatística será suficiente para a determinação dos coeficientes de \mathbf{W} e dos componentes $\mathbf{y}(k)$. Desta forma, pode-se dizer que a não-gaussianidade das fontes originais é o ponto de partida para a elaboração de métodos para a solução do problema da ICA, que tem por objetivo encontrar componentes independentes para o caso especial em que as fontes são não-gaussianas.

Para melhor compreensão de como são feitas medidas de não-gaussianidade e como estas medidas são utilizadas para a separação de fontes estatisticamente independentes serão apresentados, ao longo deste capítulo, os princípios matemáticos da ICA e os principais algoritmos desenvolvidos com base nestes princípios.

4.3 – Restrições Estatísticas e Ambigüidades para ICA

Nas seções seguintes serão apresentados as principais considerações estatísticas para solucionar o problema da Análise de Componentes Independentes.

4.3.1 - Independência Estatística e Descorrelação

Para que se possa utilizar ICA é necessário que os sinais originais sejam estatisticamente independentes. Assim sendo, a matriz de separação \mathbf{W} é determinada de tal forma que os componentes de $\mathbf{y}(k)$ também sejam estatisticamente independentes. Isto significa que o valor de um componente não fornece nenhuma informação sobre o valor dos outros componentes. Em termos estatísticos, as variáveis são mutuamente independentes se a f.d.p. conjunta de $\mathbf{y}(k)$ puder ser fatorada no produto de suas f.d.p.'s marginais, como mostra a equação (4.3) (Papoulis, 1991).

$$p(y_1, \dots, y_n) = \prod_{i=1}^n p_{y_i}(y_i) \quad (4.3)$$

A equação (4.3) pode ser utilizada para definir a propriedade mais importante das variáveis aleatórias independentes, ou seja, (Hyvärinen, 2000):

$$\mathbf{E}[g(y_1), \dots, g(y_n)] = \mathbf{E}[g(y_1)] \mathbf{E}[g(y_2)] \dots \mathbf{E}[g(y_n)] \quad (4.4)$$

Onde $g(\cdot)$ são funções não lineares quaisquer.

Através da equação (4.4) pode-se verificar que se uma determinada v.a. independente sofrer uma transformação não linear a condição de independência não é afetada.

Uma forma fraca de independência é a decorrelação. Um conjunto de n v.a. (y_1, \dots, y_n) é dito ser decorrelacionado se sua covariância é igual a zero; como definido pela equação (4.5).

$$\mathbf{E}[y_1, \dots, y_n] - \mathbf{E}[y_1] \mathbf{E}[y_2] \dots \mathbf{E}[y_n] = 0 \quad \text{para } y_1 \neq y_2 \neq \dots \neq y_n \quad (4.5)$$

Como base em (4.3) e (4.5) pode-se comprovar que se as v.a. são estatisticamente independentes, elas são decorrelacionadas, mas a decorrelação não implica em independência. O único caso em que decorrelação implica em independência estatística ocorre quando as v.a. são gaussianas, uma vez que estas variáveis são completamente descritas por suas estatísticas de segunda ordem (Papoulis, 1991). Como a decorrelação não garante a independência, ela também não pode garantir a separação em componentes independentes. Assim, para garantir a separação é necessário recorrer a estatísticas de ordem superior. A utilização de estatísticas de ordem superior, por sua vez, faz com que a separação, utilizando ICA, somente seja garantida para no máximo uma fonte gaussiana, como será apresentado na seção 4.3.2..

Não obstante, como a independência estatística implica em decorrelação, a maioria dos métodos de ICA restringe o problema da estimativa de fontes originais de tal forma que os componentes independentes estimados sejam sempre decorrelacionados. O procedimento de decorrelação reduz o número de parâmetros livres a serem determinados na separação, simplificando o problema.

4.3.2 – Variáveis Não Gaussianas

Uma das principais restrições para a realização da ICA é que os componentes independentes sejam não gaussianos ou que no máximo um dos componentes seja gaussiano. Nesta seção são apresentadas as principais razões que justificam esta restrição.

Considere um sistema de mistura (2×2) ortogonal e duas fontes gaussianas $(s_1$ e $s_2)$. Assim, os sinais misturados resultantes $(x_1$ e $x_2)$ são gaussianos, decorrelacionados e tem variância unitária. Como a f.d.p. conjunta de $(x_1$ e $x_2)$ é simétrica, ela não contém nenhuma informação sobre a matriz de mistura \mathbf{A} . Desta forma, pode-se dizer que a matriz \mathbf{A} não pode

ser estimada, se mais do que uma das fontes originais for gaussiana (Hyvärinen, 2000). De forma mais rigorosa, pode-se comprovar que a distribuição das variáveis gaussianas não é afetada por qualquer transformação ortogonal e, que as variáveis são independentes. Assim, para variáveis gaussianas, pode-se apenas estimar o modelo para ICA a menos de uma transformação ortogonal. Isto confirma e reforça a afirmação de que a separação em componentes independentes só é possível se no máximo uma fonte original for gaussiana. Além disso, é importante reforçar a idéia de que a separação somente é possível através da utilização de estatísticas de ordem superior e que, para variáveis gaussianas, as estatísticas de ordem maior que dois são zero, uma vez que as variáveis gaussianas são totalmente caracterizadas por suas estatísticas de segunda ordem.

Por outro lado, para variáveis não gaussianas, a independência estatística é a principal garantia de que as fontes estimadas serão distintas e com base neste critério são definidas a maioria das estratégias para a separação cega utilizando ICA.

A seguir são apresentadas mais informações sobre as f.d.p.'s gaussianas e não gaussianas para melhor compreensão das definições.

A função densidade de probabilidade para a distribuição gaussiana ou normal é definida em (4.6).

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4.6)$$

Onde μ e σ são, respectivamente, a média e o desvio padrão.

Uma densidade de probabilidade sub-gaussiana ou *Platykurt* é tipicamente mais plana que a distribuição gaussiana. Um exemplo é a f.d.p. uniforme.

Já a f.d.p. super-gaussiana ou *Leptokurtic* tem tipicamente extremidades mais prolongadas e picos mais proeminentes que os da distribuição gaussiana. Um exemplo é distribuição Laplaciana. Sinais de voz e de música apresentam distribuição super-gaussiana.

A Figura 4.2, mostra exemplos de variáveis sub, super e gaussianas.

Assim, vários métodos foram desenvolvidos, baseados no princípio de que para variáveis não gaussianas a independência é suficiente para garantir que as fontes estimadas serão distintas, validando desta forma a separação das fontes.

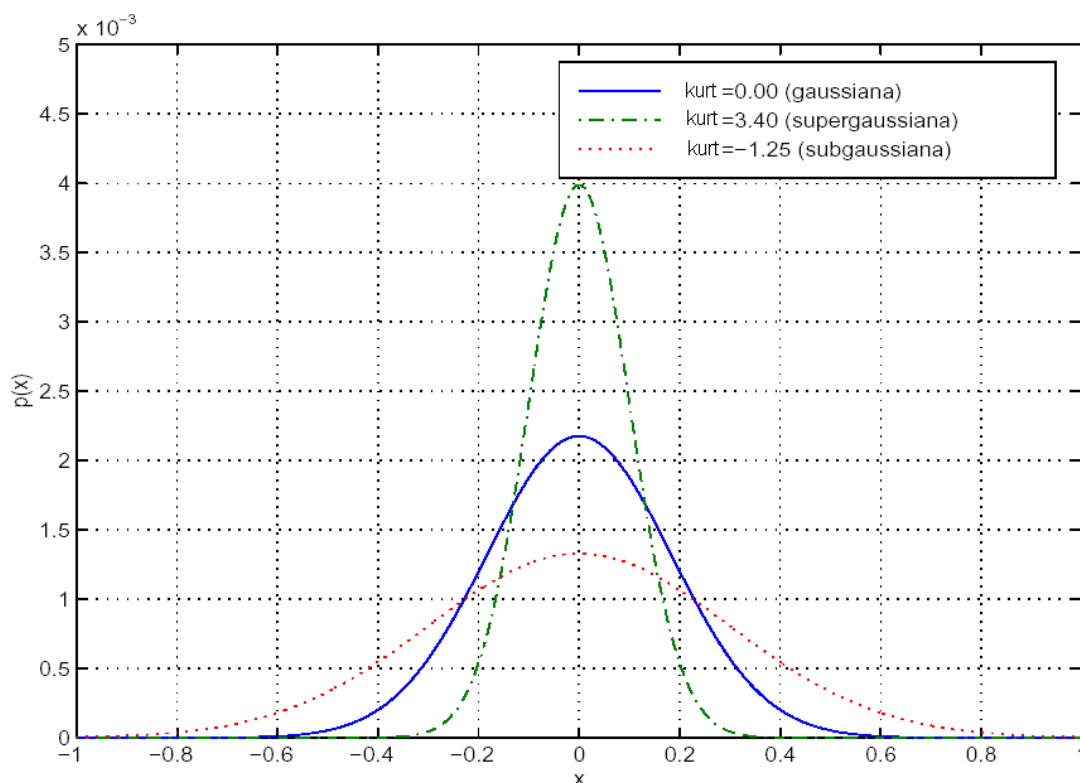


Figura 4.2 – Exemplo de distribuições gaussianas, super-gaussianas e sub-gaussianas

Na seção 4.4, serão apresentados os principais métodos utilizados para medidas de independência que permitem a separação dos sinais recebidos pelos sensores em componentes independentes.

4.3.3 – Ambigüidades da ICA

Inerentes ao problema da separação cega existem algumas ambigüidades ou indeterminações, que são mantidas pelo modelo geral de ICA da equação (4.1). As principais ambigüidades ou indeterminações são:

1. Não é possível determinar a variância (energia) dos componentes independentes. Como consequência não é possível fixar a amplitude dos componentes independentes. Mas como eles são variáveis aleatórias, a maneira mais simples de fazer isto é considerando que cada um dos componentes tem variância unitária, ou seja, $E[s(k)^2]=1$. Assim, a matriz de mistura \mathbf{A} pode ser estimada pelo método de ICA levando em conta esta restrição. Nota-se ainda, uma ambigüidade em relação ao sinal (positivo ou negativo) das fontes recuperadas. Este problema pode ser

solucionado multiplicando-se os componentes independentes recuperados por -1, sem comprometer o modelo original;

2. Não é possível determinar a ordem dos componentes independentes. A razão está no fato de que ambos $\mathbf{s}(k)$ e \mathbf{A} são desconhecidos; portanto, pode-se tranquilamente trocar a ordem dos elementos destas matrizes. A representação formal desta afirmação pode ser feita através da matriz de permutação \mathbf{P} e de sua inversa \mathbf{P}^{-1} , que podem ser substituídas no modelo da equação (4.1) resultando em (4.7) - onde os elementos de $\mathbf{Ps}(k)$ são as variáveis independentes originais, mas em uma outra ordem e a matriz \mathbf{AP}^{-1} é a nova matriz de mistura desconhecida, que pode ser encontrada através dos métodos de ICA.

A partir das indeterminações supracitadas o modelo para ICA pode ser escrito como:

$$\mathbf{x}(k) = \mathbf{AP}^T \mathbf{\Lambda}^{-1} \mathbf{\Lambda} \mathbf{Ps}(k) \quad (4.7)$$

Onde \mathbf{P} é a matriz de permuta e $\mathbf{\Lambda}$ é qualquer matriz de posto completo (*full rank matrix*) diagonal que está relacionada à amplitude das fontes estimadas. \mathbf{P}^T é a transposta de \mathbf{P} e $\mathbf{\Lambda}^{-1}$ é a inversa de $\mathbf{\Lambda}$.

Assim, a separação de sinais utilizando análise de componentes independentes somente é possível sob as condições de permutação e escalonamento.

Embora, não apareça na literatura como ambigüidade, é importante observar que alguns algoritmos causam ainda uma alteração na fase do sinal recuperado.

Nenhuma destas indeterminações, porém, compromete a recuperação das fontes, além disto elas podem ser corrigidas através de técnicas simples, como controle automático de ganho (CAG), ordenação pela máxima entropia e ajuste de fase.

4.4 – Técnicas de Pré-Processamento

A maioria dos métodos para ICA necessita que os dados passem por pré-processamento. Os principais métodos de pré-processamento empregados pelas técnicas de análise de componentes independentes são a Centralização (*Centering*) e o Branqueamento (*Whitening*). Outros métodos como, por exemplo, a filtragem tipo passa faixa

(Cichocki, 2002; Hyvärinen, 2001), também podem ser empregados, mas não serão discutidos neste trabalho.

4.4.1 – Centralização

O processamento inicial é feito através da centralização do vetor $\mathbf{x}(k)$. Isto é feito subtraindo a média $\mathbf{E}[\mathbf{x}(k)]$ de cada elemento de $\mathbf{x}(k)$, fazendo com que $\mathbf{x}(k)$ se torne uma variável de média zero. Isto implica que $\mathbf{s}(k)$ também tenha média zero (Hyvärinen, 2001).

Este tipo de pré-processamento não é obrigatório, mas simplifica os algoritmos. Após a estimativa da matriz \mathbf{A} , com os dados centralizados, pode-se completar a estimativa adicionando vetor média de $\mathbf{s}(k)$ de volta ao vetor $\mathbf{s}(k)$ centralizado.

4.4.2 – Branqueamento

Como visto no Capítulo 3, as informações estatísticas de segunda ordem não permitem a separação em componentes independentes. Mas, por outro lado, a decorrelação espacial ou independência de segunda ordem permite expressar o espaço dos sinais medidos em um sistema ortogonal, ou branco, que ainda não implica na independência estatística destas medidas, mas que irá simplificar o problema (Cardoso, 1998).

No branqueamento, também denominado esferamento, o vetor de observações $\mathbf{x}(k)$ é linearmente transformado em um novo vetor $\mathbf{z}(k)$ que é branco e é definido como: $\mathbf{z}(k) = \mathbf{Q}\mathbf{x}(k)$, onde \mathbf{Q} é uma matriz de transformação linear.

O branqueamento faz com que os componentes de $\mathbf{z}(k)$ sejam decorrelacionados e que sua matriz de covariância seja unitária, ou seja, igual à identidade.

Portanto, se \mathbf{Q} é uma matriz de branqueamento para $\mathbf{x}(k)$, então $\mathbf{z}(k) = \mathbf{Q}\mathbf{x}(k)$, é espacialmente branco, seus componentes são decorrelacionados e sua variância é unitária.

O branqueamento é aplicado, portanto, após a centralização e antes do algoritmo principal e sua aplicação é sempre possível.

Um dos métodos mais simples para branqueamento – como foi apresentado no Capítulo 3 e também pode ser empregado com algoritmos de ICA - utiliza a decomposição em autovetores (EVD) da matriz de covariância dos dados medidos, ou seja,

$$\mathbf{E}[\mathbf{x}(k)\mathbf{x}(k)^T] = \mathbf{M}\mathbf{D}\mathbf{M}^T \quad (4.8)$$

Onde \mathbf{M} é a matriz ortogonal de autovetores de $\mathbf{E}[\mathbf{x}(k)\mathbf{x}(k)^T]$ e \mathbf{D} é a matriz diagonal de seus autovalores.

Os dados branqueados $\mathbf{z}(k)$ são então obtidos, tais que:

$$\mathbf{z}(k) = \mathbf{M}\mathbf{D}^{-1/2}\mathbf{M} \mathbf{x}(k) \quad (4.9)$$

Onde a matriz $\mathbf{D}^{-1/2}$, é calculada como $\mathbf{D}^{-1/2} = \text{diag}(d_1^{-1/2}, \dots, d_n^{-1/2})$. É fácil, então, verificar que $\mathbf{E}[\mathbf{z}(k) \mathbf{z}(k)^T] = \mathbf{I}$; onde \mathbf{I} é a matriz identidade (Hyvärinen, 2000).

O branqueamento também transforma a matriz de mistura em uma nova matriz \mathbf{A}_b , tal que:

$$\begin{aligned} \mathbf{z}(k) &= \mathbf{M}\mathbf{D}^{-1/2}\mathbf{M} \mathbf{x}(k) \\ \mathbf{z}(k) &= \underbrace{\mathbf{M}\mathbf{D}^{-1/2}\mathbf{M} \mathbf{A}}_{\mathbf{A}_b} \mathbf{s}(k) \\ \mathbf{z}(k) &= \mathbf{A}_b \mathbf{s}(k) \end{aligned} \quad (4.10)$$

Sendo que, a matriz branqueada \mathbf{A}_b é uma matriz ortogonal. Por isso, muitas vezes o branqueamento é descrito como uma transformação ou decomposição ortogonal (Cardoso, 1996b).

O fato de \mathbf{A}_b ser ortogonal reduz o número de variáveis desconhecidas na matriz de mistura \mathbf{A} de n^2 para $n(n-1)/2$. Muitos algoritmos se beneficiam desta afirmação e estimam a matriz de mistura \mathbf{A}_b ao invés da matriz de mistura \mathbf{A} . Em sistemas de grandes dimensões, uma matriz ortogonal contém aproximadamente metade do número de parâmetros de uma matriz arbitrária. Para grandes dimensões, pode-se dizer que o branqueamento reduz metade do problema da ICA.

Uma ilustração gráfica do efeito do branqueamento pode ser visto na Figura 4.3. Na Figura 4.3(a) tem-se a função de distribuição de probabilidade conjunta dos componentes independentes $s_1(k)$ e $s_2(k)$, em 4.3(b) pode-se ver a distribuição conjunta dos sinais medidos $x_1(k)$ e $x_2(k)$ e finalmente em 4.3(c) tem-se a distribuição conjunta dos sinais $z_1(k)$ e $z_2(k)$, após o branqueamento.

Por meio da Figura 4.3 pode-se claramente verificar que a distribuição resultante branqueada é uma versão rotacionada da distribuição original. Seria necessário, por exemplo, estimar o ângulo que determina esta inclinação para encontrar as fontes originais.

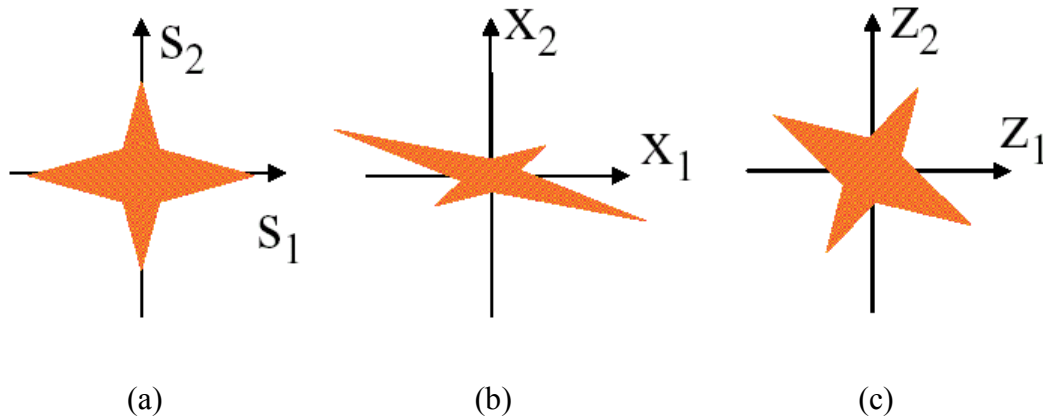


Figura 4.3 - Efeito do branqueamento: representação da f.d.p. para $n = 2$, sendo: (a) fontes originais, (b) medidas dos sensores, (c) dados branqueados (Fuentes, 2001).

4.4.3 – Pré-Processamento Adicional

Para simplificar ainda mais o problema da ICA alguns algoritmos utilizam métodos de pré-processamento adicionais.

Em alguns casos, por exemplo, pode ser útil reduzir a dimensão dos dados, ao mesmo tempo em que é feito o branqueamento, descartando os autovalores de $\mathbf{E}[\mathbf{x}(k) \mathbf{x}(k)^T]$ que sejam muito pequenos (da ordem de 10^{-10}), resultando, em geral, na redução do ruído. A redução de dimensão pode prevenir ainda a sobreposição que algumas vezes ocorre na ICA (Hyvärinen, 1999c).

Além disso, a filtragem pode ser útil, já que não altera a matriz de mistura, uma vez que os componentes independentes são filtrados pelo mesmo filtro que irá filtrar a mistura.

O filtro ótimo a ser utilizado é aquele que aumenta a independência dos componentes enquanto reduz o ruído, ou seja, é aquele que une as vantagens dos filtros passa-alta e passa-baixa enquanto minimiza suas desvantagens. Isto leva a filtragem passa-faixa, na qual as frequências mais altas e mais baixas são filtradas, deixando uma faixa de frequência adequada (Hyvärinen, 2001).

4.5 – Princípios para Estimativa de Componentes Independentes

A estimativa dos dados independentes é realizada, na maioria dos casos, pela minimização ou maximização de uma dada função custo, de contraste ou objeto (Comon,1994). Neste trabalho, será utilizado o termo função custo para definir qualquer função cuja otimização permite a estimativa de componentes independentes. Além disso, não será considerada a presença de ruído aditivo. Considerações sobre ICA na presença de ruído aditivo serão feitas na seção 4.7.

Em (Hyvärinen, 1999a), como neste trabalho, é feita uma distinção entre a função custo e o algoritmo para implementar a ICA, sendo que eles estão relacionados como mostra a Figura 4.4.

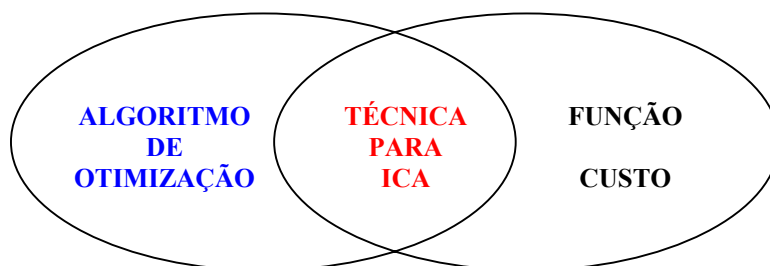


Figura 4.4 – Relação entre algoritmo, função custo e modelo para ICA.

Desta forma, o modelo para ICA depende de ambos, algoritmo de otimização e função custo, ou, em outras palavras, as propriedades estatísticas - como consistência e robustez - do modelo ICA dependem da escolha da função custo e as propriedades computacionais - como velocidades de convergência, requisitos de memória e estabilidade - dependem da escolha do algoritmo de otimização, sendo que algoritmo e função custo são independentes, ou seja, podem-se utilizar diferentes algoritmos de otimização para a mesma função custo e vice e versa.

As principais funções custo para estimativa dos componentes independentes são baseadas na medida e maximização da não-gaussianidade das variáveis medidas, principalmente por meio de HOS, minimização da informação mútua e estimativa da máxima verossimilhança.

Embora conceitualmente diferentes, pode-se demonstrar que matematicamente estas funções são equivalentes (Cardoso, 1997).

A partir destas funções custos foram desenvolvidos inúmeros algoritmos como os apresentados em (Cardoso, 1999; Amari, 1998; Karhunen, 1997; Bell, 1995; Comon, 1994; Jutten, 1991; Hyvärinen, 1997; Alvarez, 1999).

Além da abordagem estatística, existem estimativas para ICA, que são baseadas em formulações algébricas ou geométricas como as propostas em (Carvalho, 2000; Mansur, 2001;), que não fazem parte do escopo deste trabalho.

Serão apresentadas a seguir as principais funções custo utilizadas para Análise de Componentes Independentes baseadas na abordagem estatística.

4.5.1 – Maximização da Não Gaussianidade

Uma das formas de estimar os componentes independentes é através da medida da não gaussianidade dos componentes, considerando que estes são centralizados (média zero) e de variância unitária.

A medida da não gaussianidade pode ser feita por meio do *kurtosis* ou achatamento da curva da distribuição de probabilidade dos componentes sob estudo ou utilizando o princípio da negentropia como será apresentado nas seções seguintes (Hyvärinen, 2001).

4.5.1.1 – Medida de Não Gaussianidade por *Kurtosis*

As funções custo mais simples para ICA são expressas através de cumulantes quarta ordem, ou *kurtosis*.

O *kurtosis* ($kurt(\cdot)$) de uma variável aleatória $\mathbf{y}(k)$ é definido como:

$$kurt(\mathbf{y}(k)) = \mathbf{E}[\mathbf{y}(k)^4] - 3(\mathbf{E}[\mathbf{y}(k)^2])^2 \quad (4.11)$$

Considerando que $\mathbf{y}(k)$ tem variância unitária (4.11) pode ser escrita por (4.12):

$$kurt(\mathbf{y}(k)) = \mathbf{E}[\mathbf{y}(k)^4] - 3 \quad (4.12)$$

A equação (4.12) mostra que o *kurtosis* é simplesmente uma normalização do momento de quarta ordem, ou seja, $\mathbf{E}[\mathbf{y}^4(k)]$.

Existe uma propriedade muito importante com relação aos *kurtosis* e que justifica sua utilização em ICA, ou seja, para uma variável gaussiana, o momento de quarta ordem é dado por $3\mathbf{E}[\mathbf{s}^2(k)]^2$, como a variância $\mathbf{E}[\mathbf{s}^2(k)]=1$; conseqüentemente, o *kurtosis* é nulo, o que reforça a idéia de que variáveis gaussianas são proibidas para ICA.

O *kurtosis* de uma variável aleatória pode ser positivo, negativo ou nulo, sendo que:

$Kurt > 0$	Variável sub-gaussiana	Ex. f.d.p. plana, uniforme.
$Kurt < 0$	Variável super-gaussiana	Ex. f.d.p. Laplaciana.
$Kurt = 0$	Variável gaussiana	Ex. f.d.p. gaussiana ou normal

A não gaussianidade pode ser medida através do valor absoluto do *kurtosis* ou $(kurtosis)^2$, sendo que:

$ kurt = 0$	variáveis gaussianas
$ kurt > 0$	variáveis não gaussianas

Apenas em alguns casos especiais podem-se encontrar variáveis não gaussianas com $|kurt| = 0$ (Papoulis, 1991).

Os componentes independentes são encontrados maximizando ou minimizando o *kurtosis*, de acordo com algoritmo de otimização empregado.

Na prática, para se encontrar a matriz de separação \mathbf{W} que fornecerá os componentes independentes $\mathbf{y}(k)$, tais que, $\mathbf{y}(k) = \hat{\mathbf{s}}(k) = \mathbf{W}^T \mathbf{x}(k)$, com base na medida dos *kurtosis* das fontes, deve-se, como descrito em (Hyvärinen, 2000):

- Branquear o sinal medido $\mathbf{x}(k)$, de forma a se obter $\mathbf{z}(k)$;
- Inicializar \mathbf{W} com um valor qualquer;
- Calcular a direção em que *kurtosis* de $\mathbf{y}(k)$ está crescendo (ou decrescendo) mais fortemente;
- Baseado nas medidas disponíveis $\mathbf{x}(k)$, utilizar um método de gradiente, ou equivalente, para encontrar um novo vetor \mathbf{W} .

- O processo acima descrito continua até que todas as fontes sejam estimadas.

Embora existam vários algoritmos que utilizam *kurtosis* para solucionar o problema da ICA, na prática, eles apresentam algumas desvantagens, pois sua implementação computacional é complexa, não é uma medida robusta de não gaussianidade e é sensível a *outliers* (amostras inconsistentes que não se adaptam a curva). Além disso, como se devem estimar os *kurtosis* a partir de dados observados, ou amostrados, os valores podem variar bastante em função do número de amostras e alterar a estimativa das fontes.

4.5.1.2 – Negentropia

Uma outra forma de se medir a não gaussianidade de uma função é através da negentropia ou entropia diferencial.

A entropia de uma variável aleatória pode ser interpretada como o grau de desordem ou desorientação deste sistema; assim, quanto mais imprevisível e desestruturada for a variável, maior é sua entropia. Para a mesma variância, as variáveis gaussianas são as que apresentam a maior entropia (Papoulis, 1991); o que significa que a distribuição gaussiana é mais aleatória ou menos estruturada de todas as distribuições e, portanto, a entropia pode ser utilizada como medida de não gaussianidade.

Com base na teoria da informação e sob determinadas restrições, a entropia \mathcal{H} definida pela equação (4.13) define o comprimento do código de uma variável aleatória. Em outras palavras, a entropia define o grau de informação de uma dada variável aleatória (Hyvärinen, 2001).

$$\mathcal{H}(\mathbf{y}) = -\sum_i P(\mathbf{y} = a_i) \log(P(\mathbf{y} = a_i)) \quad (4.13)$$

Onde $P(\mathbf{y} = a_i)$ é a probabilidade de $\mathbf{y}(k) = \mathbf{y}$ ser igual a a_i .

A definição (4.13) pode ser generalizada para variáveis aleatórias contínuas e vetores e, nestes casos, ela passa a ser denominada entropia diferencial. Matematicamente, a entropia diferencial pode então ser definida por:

$$\mathcal{H}(\mathbf{y}) = -\int f(\mathbf{y}) \cdot \log(f(\mathbf{y})) \cdot d\mathbf{y} \quad (4.14)$$

Onde: $f(\mathbf{y})$ é a f.d.p. de $\mathbf{y}(k)$.

Por simplicidade, para se obter uma medida de não gaussianidade, utiliza-se frequentemente uma modificação da entropia diferencial denominada negentropia (J), que é equacionada em (4.15) (Hyvärinen, 2000; Comon, 1994)

$$J(\mathbf{y}) = \mathcal{H}(\mathbf{y}_{GAUSS}) - \mathcal{H}(\mathbf{y}) \quad (4.15)$$

Onde: \mathbf{y}_{GAUSS} = variável aleatória gaussiana com a mesma matriz de covariância que \mathbf{y} .

Da equação (4.15) obtém-se:

$J(\mathbf{y}) = 0$	Variável gaussiana
$J(\mathbf{y}) > 0$	Variável não-gaussiana

A vantagem de se utilizar a negentropia como medida de não gaussianidade é sua base teórica e matemática bem definida. Outra característica importante da negentropia é que ela é invariante para transformações lineares inversíveis (Comon, 1994).

O inconveniente de se utilizar a negentropia como função custo, é que ela requer um grande esforço computacional, pois necessita de uma estimativa da f.d.p. da função analisada. Uma solução é utilizar uma aproximação da negentropia.

4.5.1.3 – Aproximação da Negentropia

Um método clássico para o cálculo da negentropia aproximada é através de momentos de ordem superior, como mostra a equação (4.16) (Hyvärinen, 2000).

$$J(\mathbf{y}) = \frac{1}{12} E[\mathbf{y}^3]^2 + \frac{1}{48} kurtosis(\mathbf{y})^2 \quad (4.16)$$

Em (4.16) considera-se que \mathbf{y} tem média zero e variância unitária e a validade desta aproximação é limitada, sofrendo com a não robustez do *kurtosis*, chegando algumas vezes a ser imprecisa e em muitos casos sensível a *outliers*.

Para evitar problemas como estes, novas técnicas foram propostas. No caso mais simples, utiliza-se uma função não-quadrática, $G(\cdot)$, resultando na aproximação mostrada em (4.17).

$$J_G(\mathbf{y}) = \alpha [E[G(\mathbf{y})] - E[G(\mathbf{b})]]^2 \quad (4.17)$$

A equação (4.17) é válida para praticamente qualquer função $G(\cdot)$ não-quadrática e considerando que \mathbf{b} é uma variável aleatória gaussiana com média zero e variância unitária.

A escolha adequada de $G(\cdot)$, faz com que (4.17) seja uma aproximação conceitualmente simples, computacionalmente rápida e com propriedades estatísticas interessantes, como a robustez. Valores de $G(\cdot)$, que se mostram adequados, foram:

$$G_1(u) = \frac{1}{a_1} \log(\cosh(a_1 u)), \quad G_2(u) = -\exp\left(\frac{-u^2}{2}\right) \quad (4.18)$$

Um dos principais algoritmos que utilizam a aproximação da negentropia para medida de não-gaussianidade é o FastICA (Hyvärinen, 1997a).

4.5.1.4 – Divergência de Kullback-Leibler

A divergência de Kullback-Leibler (KLD) proporciona uma medida da divergência entre duas funções de densidade de probabilidade $f_y(\mathbf{y})$ e $g_y(\mathbf{y})$, sendo algumas vezes denominada entropia cruzada. A KLD é definida matematicamente como:

$$KLD(f_y(\mathbf{y}) \| g_y(\mathbf{y})) = \int_{-\infty}^{\infty} \mathbf{f}_y(\mathbf{y}) \log\left(\frac{\mathbf{f}_y(\mathbf{y})}{\mathbf{g}_y(\mathbf{y})}\right) \cdot d\mathbf{y} \quad (4.19)$$

Ainda que a KLD tenha algumas propriedades de distância é importante observar que $KLD(f_y(\mathbf{y}) \| g_y(\mathbf{y})) \neq KLD(g_y(\mathbf{y}) \| f_y(\mathbf{y}))$; portanto, ela é matematicamente falando uma pseudo-distância, uma vez que ela não comutativa.

Na prática, a divergência KL é muito utilizada tomando-se $g_y(\mathbf{y})$ como uma função de referência.

Existem algumas propriedades importantes com a relação a KLD e que, de certa forma, justificam sua utilização em problemas de ICA (Cardoso, 1997):

(KLD_1) É sempre positiva; e é zero se, e somente se, as funções comparadas forem idênticas;

(KLD_2) É invariante frente a transformações inversíveis do espaço amostral, tais como, permutação da ordem e escalonamento da amplitude;

(KLD_3) É uma função convexa;

(KLD_4) A entropia diferencial $\mathcal{H}(\mathbf{z})$ de uma distribuição N-dimensional $p_{\mathbf{z}}(\mathbf{z})$ coincide com o oposto da divergência KL entre esta distribuição e a distribuição uniforme $p_{\mathbf{u}}(\cdot)$, tal que:

$$\mathcal{H}(\mathbf{z}) = -KL(p_{\mathbf{z}}(\mathbf{z}) \parallel p_{\mathbf{u}}(\mathbf{z})) \quad (4.20)$$

(KLD_5) A regra da cadeia para a divergência KL é escrita como:

$$KL(f_{\mathbf{y}}(\mathbf{x}, \mathbf{y}) \parallel g_{\mathbf{y}}(\mathbf{x}, \mathbf{y})) = KL(f_{\mathbf{x}}(\mathbf{x}) \parallel g_{\mathbf{x}}(\mathbf{x})) + KL(f(\mathbf{y}|\mathbf{x}) \parallel g(\mathbf{y}|\mathbf{x})) \quad (4.21)$$

(KLD_6) A relação entre três f.d.p.s distintas $p_{\mathbf{x}}(\mathbf{x})$, $p_{\mathbf{z}}(\mathbf{x})$ e $p_{\mathbf{y}}(\mathbf{y})$ pode ser escrita como:

$$KL(p_{\mathbf{y}}(\mathbf{y}) \parallel p_{\mathbf{x}}(\mathbf{x})) = KL(p_{\mathbf{y}}(\mathbf{y}) \parallel p_{\mathbf{z}}(\mathbf{x})) + KL(p_{\mathbf{z}}(\mathbf{x}) \parallel p_{\mathbf{x}}(\mathbf{x})) \quad (4.22)$$

Sendo que esta igualdade é válida se $\mathbf{y}(k) = \mathbf{W}\mathbf{x}(k)$, onde $\mathbf{x}(k)$ e $\mathbf{y}(k)$ são vetores e \mathbf{W} é uma matriz não diagonal e se $p_{\mathbf{z}}(\mathbf{x})$ for a f.d.p. marginal de cada elemento x_i . Esta propriedade também é conhecida como Decomposição Pitagórica da Divergência KL (Kofidis, 2001).

A propriedade (KLD_1) é sem dúvida, a mais importante, uma vez que, a partir dela pode-se construir uma classe de funções de otimização para solucionar o problema de ICA.

A KLD também é muito útil quando se deseja relacionar diferentes princípios para ICA (Cardoso, 1997).

4.5.2 - Estimativa de Máxima Verossimilhança

Uma das abordagens mais conhecidas para Análise de Componentes Independentes é através da estimativa da Máxima Verossimilhança (MLE - *Maximum Likelihood Estimation*), a qual está ligada muito de perto ao princípio INFOMAX, além de ser equivalente à minimização da informação mútua (Cardoso, 1997; Hyvärinen, 2001).

Os estimadores baseados em ML são muito utilizados por sua eficiência e variação mínima. Mas, por outro lado, a ML requer que as funções de densidade de probabilidade dos sinais de entrada sejam conhecidas, o que nem sempre é possível e tem sido um desafio para os pesquisadores. Contudo, o princípio da ML pode ser aplicado para solucionar o problema de separação cega utilizando ICA. Para um modelo sem ruído aditivo, pode-se determinar a verossimilhança e então estimar os componentes independentes pela maximização da verossimilhança como descrito em (Hyvärinen, 2001; Hyvärinen, 1999a; Alvarez, 1999; Pham, 1992).

O estimador ML é calculado buscando parâmetros que maximizem a verossimilhança das observações e frequentemente é mais prático e mais simples utilizar o logaritmo da verossimilhança. A utilização do logaritmo não causa nenhuma alteração uma vez que o máximo do logaritmo é obtido no mesmo ponto da máxima verossimilhança. Assim, sendo $\mathbf{W} = (w_1, w_2, \dots, w_m)^T$, a inversa da matriz de mistura \mathbf{A} , o logaritmo da verossimilhança é dado por:

$$L = \sum_{k=1}^T \sum_{i=1}^m \log f_i(\mathbf{w}_i^T \mathbf{x}(k)) + N \log |\det \mathbf{W}|, \quad (4.23)$$

Onde $f_i(\cdot)$ é a função densidade de probabilidade de $\mathbf{s}(k)$, considerada conhecida, e $\mathbf{x}(k)$, $k=1, \dots, N$, as observações ou medidas de $\mathbf{s}(k)$. O termo $\log |\det \mathbf{W}|$ é derivado da regra clássica para transformação das variáveis aleatórias e suas densidades (Papoulis, 1991).

Em geral, para qualquer vetor aleatório $\mathbf{x}(k)$ com densidade de probabilidade $p_{\mathbf{x}}$ e para uma qualquer matriz \mathbf{W} a densidade de $\mathbf{y}(k)=\mathbf{W}\mathbf{x}(k)$ é dada por (4.24).

$$p_{\mathbf{y}} = p_{\mathbf{x}}(\mathbf{W}^{-1}\mathbf{y}) \log |\det \mathbf{W}^{-1}| \quad (4.24)$$

A abordagem de ICA baseada em MLE propõe calcular a matriz \mathbf{W} que maximize (4.23).

A vantagem dos métodos que utilizam ML, é que, sob condições regulares, eles produzem resultados eficientes, bem definidos pela teoria da estimação (Hyvärinen, 1999a).

Por outro lado, estes métodos apresentam também algumas desvantagens, como:

- Necessita-se do conhecimento *a priori* da densidade de probabilidade dos componentes independentes, o que leva ao emprego de métodos consideravelmente complexos;
- A solução pode ser sensível a *outlier*, dependendo da forma da f.d.p. escolhida.

4.5.3 – Princípio de INFOMAX

Um outro princípio que pode ser utilizado como função custo para solucionar o problema da ICA é de maximização da quantidade de informação ou entropia de uma rede neural com saídas não lineares, denominado **INFOMAX** (*Information Maximization*).

Este princípio foi inicialmente proposto em (Linsker, 1988), como um método de treinamento não supervisionado de uma rede neural baseado na maximização da informação entre as entradas e as saídas da rede neural. Este trabalho serviu de inspiração para o trabalho de Bell e Sejnowski (Bell, 1995), que é um dos principais exemplos de aplicação do princípio de INFOMAX à separação de fontes.

Considerando que $\mathbf{x}(k)$ é a entrada da rede neural cujas saídas são:

$$\mathbf{y}(k) = \mathbf{g}_i(\mathbf{w}_i^T \mathbf{x}) \quad (4.25)$$

onde $g_i(\cdot)$ são funções não lineares, \mathbf{w}_i os pesos sinápticos e \mathbf{w}_i^T é o transposto de \mathbf{w}_i . A entropia das saídas é dada por:

$$L = \mathcal{H}(\mathbf{g}(\mathbf{y})) = \mathcal{H}(g_1(\mathbf{w}_1^T \mathbf{x}), \dots, g_m(\mathbf{w}_m^T \mathbf{x})) \quad (4.26)$$

Onde $\mathcal{H}(\cdot)$ é entropia conjunta e, $g_i(\cdot)$ é uma função não linear que, quando escolhida corretamente, permite estimar o modelo de ICA.

Em (Cardoso, 1997) foi provado que o princípio de INFOMAX é equivalente à estimativa da máxima verossimilhança, desde que, a função não linear $g(\cdot)$ corresponda à função de distribuição cumulativa das fontes, de tal forma que:

$$g_i'(\cdot) = f_i(\cdot) \quad (4.27)$$

4.5.4 – Informação Mútua

Uma outra abordagem para ICA, também baseada na teoria da informação, é feita através da minimização da informação mútua (MI – *Mutual Information*), que é uma forma natural de se medir a independência entre variáveis aleatórias (Hyvärinen, 2001).

A informação mútua (I) entre m variáveis aleatórias $y_i(k)$, $i=1\dots m$, pode ser definida a partir do conceito de entropia diferencial como mostrado em (4.28).

$$I(y_1, \dots, y_m) = \sum_{i=1}^m \mathcal{H}(y_i) - \mathcal{H}(\mathbf{y}) \quad (4.28)$$

Onde $\mathcal{H}(\cdot)$ é a entropia diferencial (Hyvärinen, 2001). A informação mútua pode ser entendida utilizando a interpretação de entropia como comprimento de código, que diz que a entropia fornece, a grosso modo, o comprimento médio mínimo do código de uma variável aleatória. Desta forma, $\mathcal{H}(y_i)$ fornece o comprimento do código para y_i , quando estas variáveis são codificadas separadamente e $\mathcal{H}(\mathbf{y})$ fornece o comprimento do código quando y_i é codificado como um vetor aleatório, ou seja, todos, os componentes de y_i são codificados sob o mesmo código.

Com base na equação (4.28), pode-se definir a informação mútua entre os sinais medidos $\mathbf{x}(k)$ e os sinais a serem estimados $\mathbf{y}(k)$ pela equação (4.29).

$$I(\mathbf{x}, \mathbf{y}) = \mathcal{H}(\mathbf{x}) - \mathcal{H}(\mathbf{x}|\mathbf{y}) \quad (4.29)$$

A informação mútua é uma forma natural de medida da dependência entre variáveis aleatórias sendo sempre não negativa e zero se, e somente se, as variáveis são estatisticamente independentes. Efetivamente isto ocorre uma vez que a observação de uma variável independente não fornece nenhuma informação sobre outra variável independente.

Pode-se verificar, a partir de (4.29) que para $I(\mathbf{x}, \mathbf{y}) = 0$, $\mathcal{H}(\mathbf{x}) = \mathcal{H}(\mathbf{x}|\mathbf{y})$. Desta forma, pode-se utilizar a informação mútua como um critério confiável para estimar um modelo para ICA minimizando a informação mútua entre os componentes (Hyvärinen, 1999a; Kofidis, 2001).

Uma propriedade importante da MI (Papoulis, 1991) é que para uma transformação linear inversível como $\mathbf{y}(k) = \mathbf{W}\mathbf{x}(k)$, pode-se escrever (4.30).

$$I(y_1, \dots, y_m) = \sum_{i=1}^m \mathcal{H}(y_i) - \mathcal{H}(\mathbf{x}) - \log|\det \mathbf{W}| \quad (4.30)$$

Se $\mathbf{y}(k)$ for considerada descorrelacionada e com variância unitária, tem-se que:

$$\mathbf{E}[\mathbf{y}\mathbf{y}^T] = \mathbf{W}\mathbf{E}[\mathbf{x}\mathbf{x}^T]\mathbf{W}^T = \mathbf{I} \quad (4.31)$$

Onde \mathbf{I} é a matriz identidade.

A equação (4.31) implica em:

$$\det \mathbf{I} = 1 = (\det \mathbf{W}\mathbf{E}[\mathbf{x}\mathbf{x}^T]\mathbf{W}^T) = (\det \mathbf{W})(\det \mathbf{E}[\mathbf{x}\mathbf{x}^T])(\det \mathbf{W}^T) \quad (4.32)$$

A partir de (4.32), conclui-se que \mathbf{W} é constante. Além disso, para $\mathbf{y}(k)$ com variância unitária, a entropia e a negentropia diferem somente por uma constante. Assim, pode-se escrever (4.30) como:

$$I(y_1, \dots, y_n) = C - \sum_i J(y_i) \quad (4.33)$$

Onde C é uma constante que não depende de \mathbf{W} e $J(\cdot)$ é a negentropia.

A equação (4.33) define a principal relação entre negentropia e MI (Hyvärinen, 2000). Pela equação (4.33) pode-se concluir que encontrar a matriz \mathbf{W} que minimize a MI é, a grosso modo, equivalente a encontrar a direção na qual a negentropia é maximizada. Pode-se observar ainda - a partir de (4.33) - que a estimativa dos componentes independentes através da minimização da MI é equivalente à maximização do somatório das não gaussianidades das estimativas, quando estas são consideradas descorrelacionadas; trata-se mais uma justificativa da idéia de encontrar a maximização da não gaussianidade para solucionar o problema da ICA.

4.5.5 – Relação entre os Princípios para Estimativa de Componentes Independentes

Como apresentado na seção 4.5.3, existe uma equivalência entre os critérios de Informação Mútua e Negentropia. Nesta seção, de forma sumária, serão apresentadas outras relações entre os critérios de informação mútua, máxima verossimilhança e INFOMAX, para solucionar o problema de ICA, conforme proposto em (Cardoso, 1997; Alvarez, 1999; Kofidis, 2001) entre outros.

A informação mútua pode ser expressa por meio da divergência de Kullback-Leibler entre a f.d.p. conjunta $p_{\mathbf{y}}(\mathbf{y})$ e o produto das f.d.p.'s marginais $\prod_{i=1}^n p_{y_i}(y_i)$; pode-se então escrever a informação mútua como:

$$I(\mathbf{x}, \mathbf{y}) = \int p_{\mathbf{x}, \mathbf{y}}(\mathbf{x}, \mathbf{y}) \cdot \ln \left(\frac{p_{\mathbf{x}, \mathbf{y}}(\mathbf{x}, \mathbf{y})}{p_{\mathbf{x}}(\mathbf{x})p_{\mathbf{y}}(\mathbf{y})} \right) d\mathbf{x}d\mathbf{y} = KL(p_{\mathbf{x}, \mathbf{y}}(\mathbf{x}, \mathbf{y}) \| p_{\mathbf{x}}(\mathbf{x})p_{\mathbf{y}}(\mathbf{y})) \quad (4.34)$$

A partir de (4.34), pode-se dizer que $I(\mathbf{x}, \mathbf{y})$ também é uma medida de independência estatística.

A relação com a divergência KL mostra também a estreita relação entre a minimização da Informação Mútua e Maximização da Verossimilhança, uma vez que a verossimilhança pode ser representada como a divergência KL entre a variável observada e densidade fatorada. De uma forma geral, pode-se dizer que todos estes métodos minimizam a divergência KL entre a densidade observada e a fatorada (Hyvärinen, 1999a).

A relação entre a verossimilhança e a informação mútua pode ser vista considerando a média do logaritmo da verossimilhança, ou seja,

$$\frac{1}{N} E[L] = \sum_{i=1}^n E\{\log f_i(\mathbf{w}_i^T \mathbf{x})\} + \log|\det \mathbf{W}| \quad (4.35)$$

Na verdade, se $f_i(\cdot)$ for igual a função distribuição de $\mathbf{w}_i^T \mathbf{x}$, o primeiro termo da equação anterior será igual a $-\Sigma \mathcal{H}(\mathbf{w}_i^T \mathbf{x})$. Assim a verossimilhança será igual, a menos de uma constante, ao negativo da informação mútua dada pela equação (4.30).

Na prática, esta relação é muito forte, pois não se conhece a distribuições dos componentes independentes.

Uma abordagem razoável seria estimar a densidade de $\mathbf{w}_i^T \mathbf{x}$ como parte da estimativa da ML e utilizar esta como uma aproximação da densidade das fontes originais $\mathbf{s}(k)$. Neste caso, os critérios de ML e MI são equivalentes (Hyvärinen, 2000).

Contudo, existe uma pequena diferença que deve ser considerada em casos práticos. O problema com a estimativa da ML é que as densidades $f_i(\cdot)$ devem ser corretamente determinadas; não necessariamente de forma precisa, mas precisa-se definir se a distribuição é sub ou super-gaussiana. Na maioria dos casos, tem-se algum conhecimento *a priori* dos componentes independentes, não sendo necessário, portanto, estimar a natureza dos dados.

Em alguns casos, porém, se a informação sobre a natureza dos componentes independentes não for correta, a estimativa da ML fornecerá resultados completamente incorretos, o que faz com que muito cuidado e atenção deva ser tomada com este tipo de estimativa. Este problema não ocorre quando se utilizam técnicas que medem a não gaussianidade.

Em (Cardoso, 1997) é provado que o princípio de INFOMAX é equivalente a MLE, sendo que esta equivalência requer que as não linearidades $g_i(\cdot)$ utilizadas na rede neural sejam iguais a função de distribuição cumulativa correspondente as funções de densidade $f_i(\cdot)$.

4.6 – Principais Algoritmos para ICA

Após a escolha do princípio de estimativa para ICA é necessário escolher um método de otimização para sua implementação. Nesta seção serão apresentados, de forma sumária, os principais algoritmos para ICA.

4.6.1 – Algoritmo de Héroult-Jutten

Os trabalhos pioneiros apresentados por Héroult e Jutten (Jutten, 1991) foram baseados em redes neurais associada a uma regra de treinamento adaptativa, com estrutura semelhante à apresentada na Figura 4.5 para duas entradas e duas saídas.

O algoritmo apresentado em (Jutten, 1991) trabalha com n entradas e n saídas, ou seja, as matrizes de mistura e separação são quadradas. Uma outra consideração sobre a rede de Héroult - Jutten é que ela necessita que as entradas tenham média zero.

A rede neural de Héroult – Jutten (H-J) pode ser equacionada através do seguinte sistema:

$$\begin{aligned} \mathbf{y}(k) &= \mathbf{x}(k) - \hat{\mathbf{W}}(k) \mathbf{y}(k) \\ \mathbf{x}(k) &= \mathbf{A} \mathbf{s}(k) \end{aligned} \tag{4.36}$$

Onde $\hat{\mathbf{W}} = \begin{bmatrix} 0 & \dots & \dots & \hat{W}_{1n} \\ \hat{W}_{21} & 0 & \dots & \hat{W}_{2n} \\ \dots & \dots & \dots & \dots \\ \hat{W}_{n1} & \dots & \dots & 0 \end{bmatrix}$.

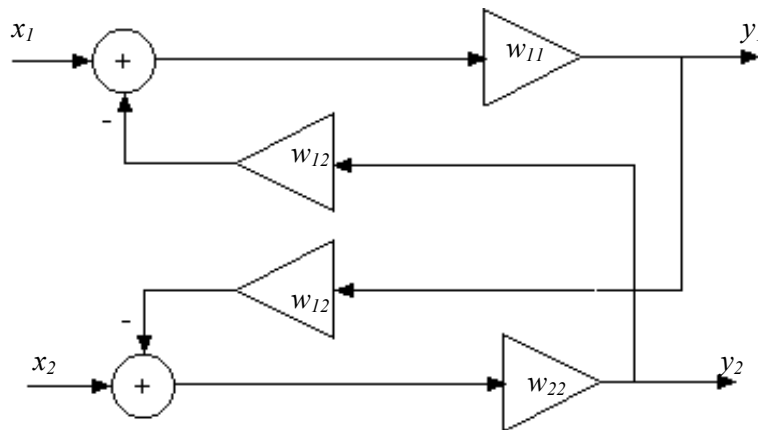


Figura 4.5 – Rede de Héault-Jutten para duas entradas e duas saídas.

O objetivo do algoritmo de Héault-Jutten é adaptar os pesos sinápticos de $\hat{\mathbf{W}}$ de forma a se obter uma estimativa das fontes originais, tal que, $\mathbf{y}(k) = \hat{\mathbf{s}}(k)$.

Os pesos ótimos são obtidos quando os elementos de $\mathbf{y}(k)$ forem estatisticamente independentes.

Então com base na condição de independência estatística, os pesos da matriz \mathbf{W} são atualizados de acordo com a equação (4.37).

$$\Delta \mathbf{W}_{ij} = \mu(k) f(y_i(k)) g(y_j(k)), \text{ para } i \neq j \tag{4.37}$$

Onde $f(\cdot)$ e $g(\cdot)$ são funções de ativação diferentes, onde pelo menos uma delas é não linear e $\mu(k) > 0$ é passo de adaptação. Para fontes sub-gaussianas utiliza-se em geral:

$$f(\mathbf{y}(k)) = \mathbf{y}(k)^3 \text{ e } g(\mathbf{y}(k)) = \tanh(10\mathbf{y}(k)) \tag{4.38}$$

E para fontes super-gaussianas utiliza-se:

$$f(\mathbf{y}(k)) = \tanh(10\mathbf{y}(k)) \text{ e } g(\mathbf{y}(k)) = \mathbf{y}(k)^3 \quad (4.39)$$

Sendo que devem ser respeitadas as condições $\mathbf{E}[f(\mathbf{y}(k))] = 0$ e $\mathbf{E}[g(\mathbf{y}(k))] = 0$ (Cichoki, 2002).

O valor de y_i , calculado a cada iteração, é dado por:

$$y_i = (\mathbf{I} + \hat{\mathbf{W}})^{-1} x_i \quad (4.40)$$

Os valores de $\mathbf{y}(k)$ resultantes após a convergência do algoritmo correspondem aos componentes independentes.

Embora o algoritmo de H-J seja simples e tenha sido pioneiro na solução do problema da ICA, na prática, ele apresenta algumas desvantagens. Ele pode trabalhar de forma imperfeita e até mesmo falhar se, por exemplo, a matriz de mistura for mal condicionada. Além disso, o número de fontes que o método pode separar é limitado e, embora sua estabilidade local tenha sido verificada a convergência global não é garantida. Neste algoritmo deve-se ainda considerar que a distribuição de probabilidade das fontes é conhecida. Também a inversão de matrizes apresentada em (4.40) pode ser computacionalmente pesada principalmente quando se trabalha com mais do que duas fontes e duas misturas (Hyvärinen, 2001).

4.6.2 – Algoritmos de Descorrelação Não Linear

Algoritmos baseados no cancelamento de decorrelação cruzada foram introduzidos independentemente por Cichocki et al. (Cichocki, 2002; Hyvärinen, 1999), Cardoso e Laheld (Cardoso, 1996a).

Nestes algoritmos, o esforço computacional é reduzido evitando inversão de matrizes, além de serem mais estáveis, quando comparados com o algoritmo de H-J.

No algoritmo proposto por Cichocki et al., os pesos da matriz de separação são atualizados por:

$$\Delta \mathbf{W}_{ij} = (\mathbf{I} - f(\mathbf{y}(k))g(\mathbf{y}^T(k)))\mathbf{W} \quad (4.41)$$

Onde $\mathbf{y}(k) = \mathbf{W}\mathbf{x}(k)$ e $f(\cdot)$ e $g(\cdot)$ são não linearidades aplicadas separadamente em cada componente do vetor $\mathbf{y}(k)$. A matriz \mathbf{I} (identidade) pode ser substituída por qualquer matriz diagonal positiva.

4.6.2.1 – Algoritmo EASI

Em (Cardoso, 1996a) é apresentada uma família de algoritmos denominada EASI (*Equivariant Adaptive Separation via Independence*). Este algoritmo sem dúvida é um dos mais interessantes da literatura, por realizar separação equivariante e adaptativa utilizando a propriedade da independência estatística. A implementação do algoritmo é feita, portanto, com base na estimativa da equivariância e gradiente relativo.

Os algoritmos da família EASI seguem a idéia de atualização serial (*serial updating*). Esta forma de atualização gera algoritmos com estruturas simples, tanto para misturas reais, quanto para complexas.

Outra característica importante destes algoritmos é que seu desempenho não depende da matriz de mistura. De uma forma geral, a taxa de convergência, estabilidade e grau de rejeição a interferências depende somente da distribuição dos sinais fontes.

A adaptação serial para uma matriz de separação $\mathbf{W}_{n \times n}$ - principal ponto apresentado em (Cardoso, 1996a) – consiste em uma regra de treinamento, tal que:

$$\mathbf{W}_{n+1} = \mathbf{W}_n - \mu T(\mathbf{y})\mathbf{W}_n \quad (4.42)$$

Onde μ é o passo de adaptação e $T(\mathbf{y})$ é uma função matricial $n \times n$ das saídas, denominada função de estimativa.

A regra de adaptação mostrada em (4.42) é denominada serial, pois pode ser escrita na forma multiplicativa como: $\mathbf{W}_{n+1} = (\mathbf{I} - \mu T(\mathbf{y}))\mathbf{W}_n$, na qual \mathbf{W}_n é atualizada através da multiplicação de $(\mathbf{I} - \mu T(\mathbf{y}))$, na saída do sistema atual \mathbf{W}_n , para se obter o sistema atualizado \mathbf{W}_{n+1} (Cardoso, 1996a).

A adaptação série é ainda dita ser equivariante ou bem condicionada; sendo que um estimador $\hat{\mathbf{A}}$ para a matriz de mistura \mathbf{A} é dito ser equivariante se satisfizer à relação:

$$\hat{\mathbf{A}}(\mathbf{M}\mathbf{x}(k)) = \mathbf{M}\hat{\mathbf{A}}(\mathbf{x}(k)) \quad (4.43)$$

Onde $\mathbf{x}(k)$ é o vetor de observações e $\mathbf{M}_{n \times n}$ é qualquer matriz inversível.

Uma grande vantagem da propriedade da equivariância em BSS é que ela leva a estimadores cujo comportamento é uniforme, ou seja, é independente dos parâmetros da matriz de mistura, desde que a matriz seja não singular.

Assim, pode-se dizer que a adaptação serial é uma ferramenta necessária para transferir o comportamento uniforme dos algoritmos equivariantes para um algoritmo adaptativo (Cardoso, 1996a).

Partindo então das propriedades da equivariância e atualização serial e baseando-se na técnica de gradiente relativo, Cardoso e Laheld desenvolveram o algoritmo EASI, onde a matriz de separação é atualizada por:

$$\mathbf{W}_{n+1} = \mathbf{W}_n - \mu [\mathbf{y}(k)\mathbf{y}(k)^T - \mathbf{I} + \mathbf{g}(\mathbf{y}(k))\mathbf{y}(k)^T - \mathbf{y}(k)\mathbf{g}(\mathbf{y}(k))^T] \mathbf{W}_n \quad (4.44)$$

Onde $\mathbf{g}(\cdot)$ é uma função não linear e μ é o passo de adaptação do algoritmo.

Pode-se ainda escrever uma forma normalizada do algoritmo (4.44), com o objetivo de garantir a convergência do algoritmo e evitar *outliers*; assim, a forma normalizada do algoritmo EASI é dada por:

$$\mathbf{W}_{n+1} = \mathbf{W}_n - \mu \left(\frac{\mathbf{y}(k)\mathbf{y}(k)^T - \mathbf{I}}{1 + \mu \mathbf{y}^T(k)\mathbf{y}(k)} + \frac{\mathbf{g}(\mathbf{y}(k))\mathbf{y}^T(k) - \mathbf{y}(k)\mathbf{g}^T(\mathbf{y}(k))}{1 + \mu |\mathbf{y}^T(k)\mathbf{g}(\mathbf{y}(k))|} \right) \mathbf{W}_n \quad (4.45)$$

O algoritmo (4.45) é denominado EASI normalizado e é muito parecido com o algoritmo LMS modificado ou LMS normalizado (Cardoso, 1996a).

4.6.3 – Algoritmo INFOMAX

Bell e Sejnowski em (Bell, 1995) propuseram um algoritmo que maximiza a informação transferida em uma rede neural artificial (RNA) tipo *feedforward* com unidades não lineares. Estas não linearidades, presentes na função de transferência da rede neural, são capazes de captar os momentos de ordem superior das distribuições das entradas da rede e de realizar uma redução de redundância entre os componentes da saída. Desta forma a rede está

apta para realizar a separação os sinais de entrada em componentes que serão apresentados na saída.

Os autores relacionaram, assim, a separação cega à teoria da informação, de tal forma que a matriz de separação pode ser determinada pela maximização da informação (entropia) do vetor de observações $\mathbf{x}(k)$ seguida por não linearidades.

A maximização da informação $\mathcal{H}(\mathbf{y})$ em uma rede neural não linear tende a minimizar a informação mútua entre os componentes de saída $y_i(g(\mathbf{u}_i))$ quando a otimização é realizada sobre os pesos sinápticos \mathbf{W} sobre a função $g(\mathbf{u}_i)$, onde $g(\mathbf{u}_i)$ é uma não linearidade monotônica inversível e $\mathbf{u}(k)=\mathbf{W}\mathbf{x}(k)$.

A Figura 4.6 mostra de forma simplificada a rede proposta por Bell-Sejnowski.

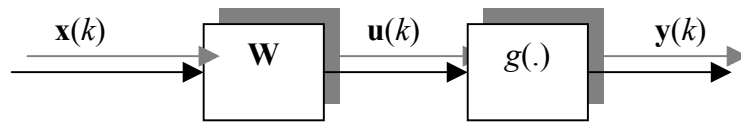


Figura 4.6 – Rede de Bell-Sejnowski

A entropia conjunta $\mathcal{H}(\mathbf{y}_1, \dots, \mathbf{y}_n)$ das saídas da rede neural é dada pela somatória das entropias marginais $\mathcal{H}(\mathbf{y}_i)$ menos a informação mútua entre as saídas $I(\mathbf{y}_1, \dots, \mathbf{y}_n)$, ou de forma matemática:

$$\mathcal{H}(\mathbf{y}_1, \dots, \mathbf{y}_n) = \mathcal{H}(\mathbf{y}_1) + \dots + \mathcal{H}(\mathbf{y}_n) - I(\mathbf{y}_1, \dots, \mathbf{y}_n) \quad (4.46)$$

A expressão (4.46) deixa claro que a maximização da entropia conjunta corresponde à minimização da informação mútua, uma vez que esta é sempre positiva. Assim, para $I(\mathbf{y}_1, \dots, \mathbf{y}_n) = 0$, a entropia conjunta é máxima e igual à soma das entropias marginais, condição que só é atingida se $\mathbf{u}(k) = \mathbf{s}(k)$.

Quanto a não linearidade $g(\cdot)$, ela deve corresponder a função densidade de distribuição cumulativa (c.d.f.) das fontes, que neste caso, é considerada conhecida.

Se a informação mútua entre as saídas é zero $I(\mathbf{y}_1, \dots, \mathbf{y}_n) = 0$, a informação mútua antes da não linearidade $I(\mathbf{u}_1, \dots, \mathbf{u}_n)$ também deve ser zero, uma vez que a não linearidade não introduz qualquer dependência. A relação entre y_i e u_i é tal que:

$$p(y_i) = \frac{p(u_i)}{\left| \frac{\partial g(u_i)}{\partial u_i} \right|} \tag{4.47}$$

Onde $p(\cdot)$ é a função de distribuição.

Através de (4.47) pode-se dizer que u_i é uma variável independente, com uma distribuição aproximadamente igual à derivada da não linearidade $g(\cdot)$. Para o caso da função sigmoïdal, a f.d.p. correspondente é mostrada na Figura 4.7.

A escolha correta da não linearidade é, portanto, de grande importância. Os experimentos mostram que quando ocorre grande descasamento entre a não linearidade e a função densidade cumulativa das fontes originais, o algoritmo não consegue minimizar a informação mútua entre os componentes da saída.

Uma outra motivação para a escolha da não linearidade $g(\cdot)$ é que ela proporciona uma combinação de estatísticas de ordem superior através de sua expansão em séries de Taylor; esta expansão é essencial para minimizar correlações de ordem superior que possam existir nos sinais medidos.

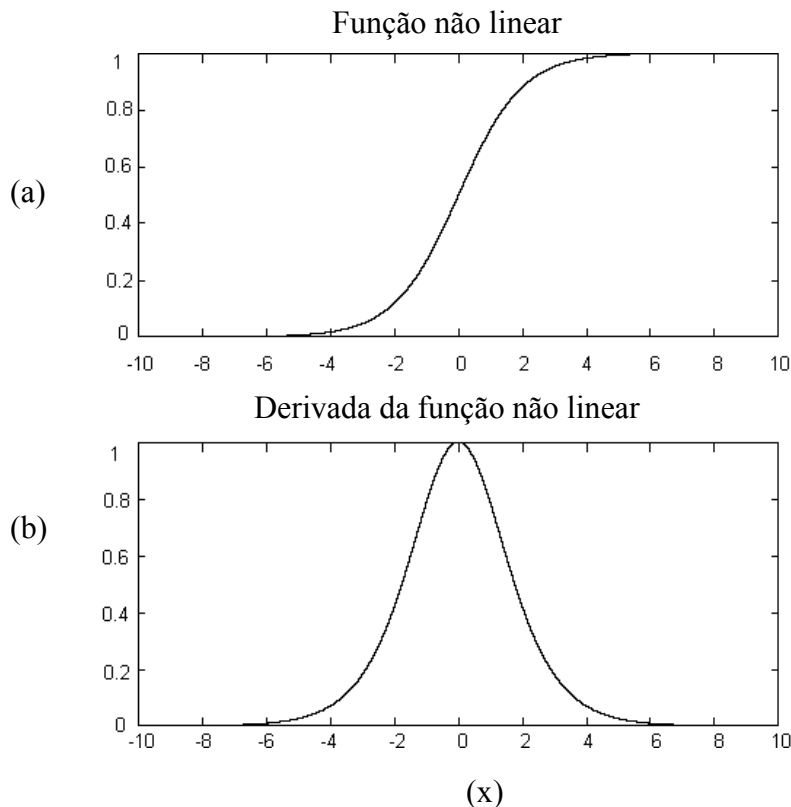


Figura 4.7 – (a) Função não linear ($g(\mathbf{x}) = 1/(1+\exp(-\mathbf{x}))$)
 (b) derivada da função não linear ($\partial g/\partial \mathbf{x} = \mathbf{y}(1-\mathbf{y})$)

A regra de treinamento da rede neural é definida em função da maximização da entropia de saída $\mathcal{H}(\mathbf{y})$ da rede é dada por:

$$\Delta \mathbf{W} = \frac{\partial \mathcal{H}(\mathbf{y})}{\partial \mathbf{W}} = [(\mathbf{W}^T)^{-1} + \varphi(\mathbf{u}) \mathbf{x}^T] \quad (4.48)$$

Quando a função não linear $g(\mathbf{u})$ utilizada é uma função sigmoidal, dada por:

$$g(\mathbf{u}) = (1 + \exp(\mathbf{u}))^{-1} \quad (4.49)$$

Então,

$$\varphi(\mathbf{u}) = 1 - 2 * (1 + \exp(-\mathbf{u}))^{-1} \quad (4.50)$$

Por outro lado, se a função não linear utilizada for $g(\mathbf{u}) = \tanh(\mathbf{u})$, então, $\varphi(\mathbf{u}) = -2 * (\tanh(\mathbf{u}))$.

A função sigmoidal, assim como a função tangente hiperbólica, é utilizada com fontes super-gaussianas como, por exemplo, sinais de áudio e voz. No caso de fontes sub-gaussianas outras funções devem ser utilizadas como, por exemplo, a função cúbica.

Embora o algoritmo INFOMAX seja um marco no desenvolvimento das técnicas para separação cega, apresentando várias vantagens, sua convergência é muito lenta, embora possa ser melhorada pelo branqueamento e principalmente com o uso de gradiente natural (Amari, 1996).

Outra desvantagem do algoritmo é que se deve conhecer, *a priori*, a distribuição de probabilidade das fontes originais. Para tentar solucionar este problema e possibilitar a separação de fontes super e sub-gaussianas foram desenvolvidas extensões do algoritmo INFOMAX (Lee, 1999), que procuram adaptar a função não linear utilizada de acordo com os *kurtosis* das fontes estimadas. O problema destas extensões é que a estimativa dos *kurtosis* nem sempre é exata, principalmente quando não se dispõe de muitas amostras dos sinais observados.

4.6.4 – Algoritmos Baseados em Gradiente Natural

Amari et al. (Amari, 1996; Amari, 1997) alteraram o algoritmo INFOMAX utilizando Gradiente Natural (NG) ao invés de Gradiente Estocástico, reduzindo a complexidade computacional do algoritmo e aumentando significativamente sua convergência.

O conceito de NG foi introduzido de forma independente em (Amari, 1996) e (Cardoso, 1996a), onde o gradiente recebeu a denominação Gradiente Relativo. De forma geral, o gradiente natural difere do gradiente relativo, no ponto em que o primeiro é definido para qualquer grupo de transformações, enquanto que o segundo é definido para alguns modelos estatísticos. Para o problema da separação cega de fontes, porém, pode-se dizer que estas técnicas são equivalentes (Fuentes, 2000).

O algoritmo desenvolvido por Amari et al. (Amari, 1996) mede a dependência entre as saídas através da informação mútua das saídas e utiliza a técnica de gradiente natural para minimizá-la; minimizando assim, a dependência entre os componentes.

Para expressar a independência entre os componentes pode-se utilizar a divergência de Kullback-Leibler entre a distribuição conjunta e o produto das distribuições marginais dos componentes de saída. A minimização da divergência KL leva, então, ao algoritmo para estimativa da matriz de separação dado por:

$$\mathbf{W}_{n+1} = \mathbf{W}_n + \eta[\mathbf{I} - g(\mathbf{y}(k))\mathbf{y}^T(k)]\mathbf{W}_n \quad (4.51)$$

Onde: $\mathbf{y}(k) = \mathbf{W}\mathbf{x}(k)$, η é o passo de adaptação do algoritmo e $g(\cdot)$ são não linearidades adequadamente escolhidas de acordo com os sinais de saída; em geral utiliza-se para $g(\cdot)$ os seguintes valores:

$$g(\mathbf{y}) \begin{cases} \tanh(\mathbf{y}) \Rightarrow \text{componentes super-gaussianos} \\ \mathbf{y}^3 \Rightarrow \text{componentes sub-gaussianas} \end{cases}$$

Pode-se dizer que o algoritmo NG é um caso especial dos algoritmos de decorrelação linear (relacionado ao algoritmo EASI), uma vez que o NG apresenta a propriedade da equivariância e pode ser facilmente implementado por uma rede neural.

Em (Amari, 1996), são realizadas várias simulações computacionais com o objetivo de validar o algoritmo proposto.

Por ser uma derivação do INFOMAX, o NG necessita do conhecimento, *a priori*, da distribuição das fontes originais. Para solucionar esta limitação, foram propostos extensões deste algoritmo que lidam com fontes super e sub-gaussianas. Um destes algoritmos é *Flexible ICA* (ou FlexICA) (Choi, 2000c; 1998a), que calcula o valor *kurtosis* das fontes estimadas e utiliza este valor para ajustar a regra de treinamento para estimativa da matriz de separação. O algoritmo FlexICA será apresentado em detalhes no Capítulo 7.

4.6.5 - Algoritmo Rápido de Ponto Fixo (FastICA)

Os algoritmos adaptativos baseados em gradiente estocástico podem apresentar problemas quando utilizados em casos práticos, pois sua convergência torna-se lenta, além de ser dependente de uma escolha adequada do passo de adaptação. Uma escolha não adequada do passo de adaptação pode levar a problemas na convergência do algoritmo. Para tentar minimizar este problema foi desenvolvida uma família de algoritmos baseados em iterações de ponto fixo denominada FastICA (Hyvärinen, 1997a, 1997b; 2000; 2001).

Os algoritmos desta família são diferenciados pela abordagem e pela função custo utilizada; todos eles, porém visam encontrar componentes independentes através da maximização da negentropia.

Com relação à abordagem, foram desenvolvidas duas versões do FastICA: a primeira permite a recuperação de todas as fontes (*multi-unit*) e a segunda, encontra um dos componentes (*one-unit*). Ambas abordagens podem utilizar qualquer função custo não paramétrica para estimar a negentropia das fontes (Hyvärinen, 1997b).

O algoritmo inicialmente proposto utilizava *kurtosis*, sendo, posteriormente, generalizado para outras funções custo (Hyvärinen, 2001).

O algoritmo FastICA procura encontrar uma direção, ou seja, um vetor, cuja projeção $\mathbf{W}^T \mathbf{x}(k)$ maximiza a função $J_G(\mathbf{W})$; ou seja, uma aproximação da negentropia dada por (4.52).

$$J_G(\mathbf{W}) = [\mathbf{E}[G(\mathbf{W}^T \mathbf{x}(k))] - \mathbf{E}[G(\mathbf{v})]]^2 \quad (4.52)$$

Onde $G(\cdot)$ é qualquer função não quadrática e \mathbf{v} é uma variável gaussiana de média zero e variância unitária. Na prática, as esperanças $\mathbf{E}[\cdot]$ são substituídas por médias amostrais.

A regra de treinamento do algoritmo consiste de um método de decorrelação que estima os componentes um a um (*one-unit*), sendo que para dados branqueados, o algoritmo FastICA *one-unit* pode ser equacionado da seguinte forma:

$$\mathbf{w}(k)^* = \mathbf{E}[\mathbf{x}g(\mathbf{w}(k-1)^T \mathbf{x})] - \mathbf{E}[g'(\mathbf{w}(k-1)^T \mathbf{x})]\mathbf{w}(k-1) \quad (4.53)$$

$$\mathbf{w}(k) = \frac{\mathbf{w}^*(k)}{\sqrt{\mathbf{w}^{*T}(k) \cdot \mathbf{C} \cdot \mathbf{w}^*(k)}} \quad (4.54)$$

Onde $g(\cdot)$ é a derivada da função $G(\cdot)$. \mathbf{C} é a matriz de covariância dos sinais observados e $\mathbf{w}(k)$ é o vetor de separação normalizado após cada iteração.

As esperanças $\mathbf{E}[\cdot]$ são estimadas através da média amostral de um número suficientemente grande de amostras dos dados de entrada para garantir a precisão. A Figura 4.8 mostra o diagrama em blocos para implementação do algoritmo FASTICA.

Quando é necessária a estimativa de várias (ou de todas) fontes, deve-se processar o algoritmo FastICA *one-unit* utilizando várias unidades (neurônios) com vetores de peso $\mathbf{w}_1(k)$, $\mathbf{w}_2(k), \dots, \mathbf{w}_n(k)$, onde n é o número total de componentes que se deseja recuperar.

Para evitar que vários os resultados convirjam para o mesmo ponto, devem-se descorrelacionar as saídas $\mathbf{w}_1^T(k)\mathbf{x}(k)$, $\mathbf{w}_2^T(k)\mathbf{x}(k), \dots, \mathbf{w}_n^T(k)\mathbf{x}(k)$, a cada iteração (Hyvärinen, 2000). Para se obter esta descorrelação pode-se utilizar um método denominado *Deflation* (Hyvärinen, 1997; Cichocki, 2002). A aplicação deste método leva a estimativa das fontes uma a uma sem repetição.

As principais características dos algoritmos FastICA são:

- Na maioria dos casos, um pequeno número de iterações - em geral, de 5 a 10, são suficientes para se obter a máxima precisão. Uma consequência desta característica é uma convergência muito rápida, de onde provém o adjetivo FAST;
- Comparado com algoritmos baseados em gradiente, o FASTICA é mais simples, uma vez que não necessita de ajuste no passo de adaptação;
- Pode ser considerado um algoritmo neural no sentido que sua estrutura é paralela e distribuída, mas não é adaptativa;
- Ao invés de utilizar todos os dados imediatamente para treinamento, o FASTICA utiliza médias amostrais calculadas sobre uma amostra grande de dados;

- O algoritmo encontra diretamente as fontes para praticamente qualquer função não-gaussiana, ou seja, não é necessário que seja feita uma distinção entre as fontes com *kurtosis* positivos ou negativos;
- O desempenho do algoritmo pode ser melhorado de acordo com a escolha da função não linear a ser utilizada;
- Pode ser utilizado para identificar os componentes independentes um a um (extração cega de fontes).

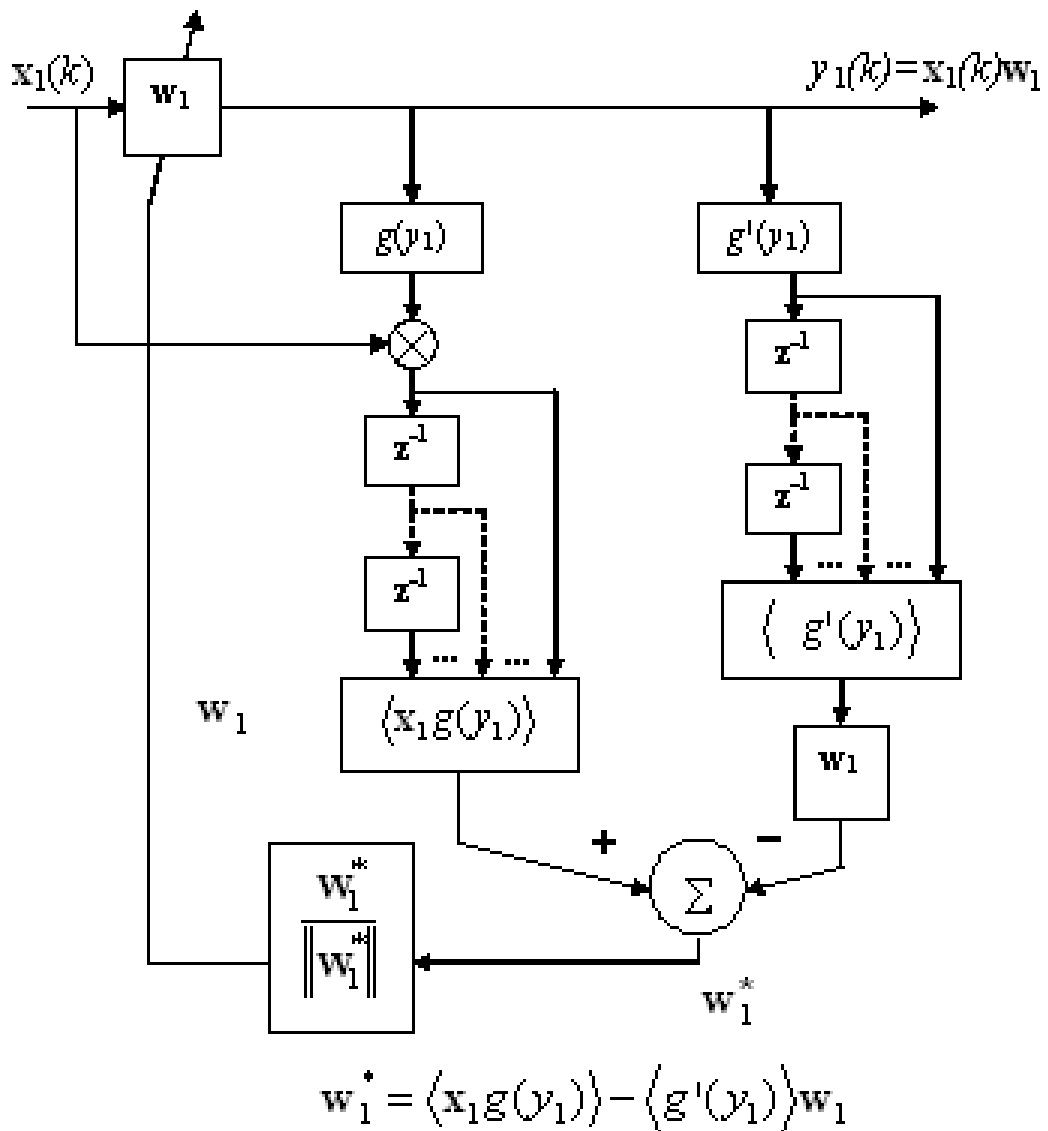


Figura 4.8 - Diagrama em blocos do FastICA (Hyvärinen, 2001).

4.6.6 – Algoritmos Baseados em Estatísticas de Ordem Superior

A base dos algoritmos que utilizam estatísticas de ordem superior - cumulantes de ordem superior ou *kurtosis* - é o fato de que a independência dos componentes de um vetor aleatório ocorre quando seus cumulantes cruzados são nulos, ou de forma equivalente, o módulo de seus cumulantes é máximo. Isto implica que a BSS pode ser obtida, por exemplo, fazendo com que os cumulantes cruzados dos vetores de saída $\mathbf{y}(k)$ sejam nulos.

Um grande número de algoritmos tem utilizado cumulantes de quarta ordem para estimar componentes independentes (Cardoso, 1993; Cardoso, 1999, Comon, 1994; Alvarez, 1999; Cichoki, 2002; Fuentes, 2001; Hyvärinen, 2001).

Para variáveis aleatórias de média zero (y_i, y_j, y_l, y_k) os cumulantes de quarta ordem são definidos por:

$$\begin{aligned} Cum_{ijkl}[\mathbf{y}] &= Cum [y_i, y_j, y_k, y_l] \\ &= \mathbf{E}[y_i y_j y_l y_k] - \mathbf{E}[y_i y_j] \mathbf{E}[y_k y_l] - \mathbf{E}[y_i y_k] \mathbf{E}[y_j y_l] - \mathbf{E}[y_i y_l] \mathbf{E}[y_j y_k] \end{aligned} \quad (4.55)$$

Ou ainda,

$$Cum_{ijkl}[\mathbf{s}] = kurt_i (\delta_{ijkl}) \quad (4.56)$$

Onde δ é o delta de Kronecker e $kurt_i$ é *kurtosis* ou cumulante de 4ª ordem da i -ésima fonte.

Com base na estimativa da máxima verossimilhança (ML), pode-se definir uma função custo, através da medida do erro quadrático entre os cumulantes das fontes originais $\mathbf{s}(k)$ e das estimadas $\mathbf{y}(k)$, tal que:

$$\phi_4[\mathbf{y}] = \sum_{ijkl} (Cum_{ijkl}[\mathbf{y}] - Cum_{ijkl}[\mathbf{s}])^2 \sum_{ijkl} (Cum_{ijkl}[\mathbf{y}] - kurt_i^2 \delta_{ijkl})^2 \quad (4.57)$$

A função (4.57) custo é definida se todos as fontes tem *kurtosis* não nulos, ou seja, ela não pode ser aplicada à fontes gaussianas, que tem *kurtosis* nulo, como já mostrado anteriormente. Ainda para a utilização desta função é necessário que se tenha algum conhecimento a respeito das fontes originais, mais especificamente sobre o *kurtosis* das fontes originais.

Vale ainda ressaltar que a minimização da função baseada em ML, com relação às distribuições das fontes, leva a função baseada em maximização da informação. Assim, a minimização da função (4.57), com relação aos *kurtosis* das fontes, leva a uma aproximação da função baseada nos princípios de informação mútua ou máxima informação, definida por (Fuentes, 2001):

$$\phi_{MI}[\mathbf{y}] = \sum_{ijkl \neq iiii} (\text{Cum}_{ijkl}[\mathbf{y}])^2 = -\sum_i (\text{kurt}_i[\mathbf{y}])^2 \quad (4.58)$$

Portanto, o problema da BSS pode ser solucionado pela maximização da soma das *kurtosis* ao quadrado de $\mathbf{y}(k)$, traduzindo a independência dos elementos do vetor de saída em estatísticas de quarta ordem. É importante observar ainda que a função (4.58) é uma aproximação da função (4.57) e dispensa o conhecimento a respeito das f.d.p. das fontes originais. Este resultado foi inicialmente apresentado em (Comon, 1994).

Em (Cardoso, 1993) é proposto algoritmo JADE (*Joint Approximate Diagonalization of Eigemtrices*) que tem por objetivo minimizar uma função custo que corresponde a soma dos cumulantes cruzados, dada por (4.59).

$$\phi_{JADE}^0[\mathbf{y}(k)] = \sum_{ijkl \neq ijkk} (\text{Cum}_{ijkl}[\mathbf{y}(k)])^2 \quad (4.59)$$

A equação (4.59) remete a idéia de minimizar os cumulantes cruzados para se determinar os componentes independentes, além de ser um critério de diagonalização de cumulantes.

Quando o número de fontes é igual a 2, a minimização da função custo pode ser obtida através de rotações de Givens, porém quando o número de fontes é maior que dois, é necessária a utilização de métodos jacobianos para minimizar a função custo; por isso, o JADE é conhecido como um algoritmo jacobiano. Mais detalhes sobre o algoritmo JADE serão apresentados no Capítulo 6.

Embora os algoritmos baseados nas funções (4.58) e (4.59) dispensem o conhecimento a respeito dos *kurtosis* das fontes originais, quando se conhecem estes *kurtosis* as funções custo podem ser simplificadas (Fuentes, 2001). Por exemplo, se as fontes são sub-gaussianas, (*kurtosis*<0), pode-se escrever a função custo (4.60).

$$\phi_{kurt}^0[\mathbf{y}(k)] = -\sum_i (Cum_{iiii}[\mathbf{y}(k)])^2 = -\sum_{i=1}^n \mathbf{E}^4[y_i(k)] \quad (4.60)$$

Esta função tem por objetivo maximizar o *kurtosis* das saídas e este algoritmo recebe o nome de MaxKur.

Os algoritmos que utilizam cumulantes de ordem superior ou *kurtosis* apresentam bons resultados para sistemas de pequenas dimensões; porém para sistemas de grandes dimensões, os requisitos de memória tornam a utilização destes algoritmos praticamente inviável. Isto ocorre porque para armazenar os coeficientes de 4ª ordem são necessárias m^4 unidades de memória, onde m é o número de sensores (Hyvärinen, 1999).

4.6.6.1 – Algoritmo de Comon

Em (Comon, 1994) detalhou-se matematicamente o problema da ICA, o qual é visto como uma extensão da Análise de Componentes Principais.

Comon apresentou várias considerações para que o problema da separação cega utilizando ICA pudesse ser tratável: o sistema de mistura foi considerado sem ruído aditivo; foi considerado no máximo um componente gaussiano e matriz de mistura foi considerada ortogonal. Feitas estas considerações iniciais, os sinais medidos pelos sensores são branqueados para garantir sua decorrelação.

Segundo Comon, a solução do problema da ICA depende de uma função custo. Uma das funções propostas foi construída a partir da informação mútua das observações. Para implementações práticas, esta função foi aproximada pela expansão de Edgeworth (Weisstein, 2000) da informação mútua e consiste basicamente de uma combinação de cumulantes de ordem superior. A vantagem de utilizar a expansão de Edgeworth é que seus termos são apresentados em ordem decrescente de significância. A partir desta expansão, Comon definiu uma função custo simplificada, a qual era igual a soma dos cumulantes de quarta ordem e cuja maximização implicava na minimização da informação mútua dos componentes observados (Comon, 1994; Basu, 2000). A partir desta função custo, foi proposto um algoritmo de otimização que é implementado através de uma seqüência de rotações de polinômios de quarta ordem.

Com relação ao algoritmo proposto, Comon demonstrou, empiricamente, que $1 + \sqrt{n}$ iterações são suficientes para a convergência do algoritmo, com n igual ao número de sinais.

Ainda em testes empíricos, mostrou-se que o algoritmo apresenta bons resultados mesmo quando a relação sinal ruído esta em torno de 0 dB, para ruído não gaussiano e dependente do número de amostras consideradas (embora ele apenas mostre o valor de erro entre a matriz de mistura e matriz de separação estimada e não o valor real da relação sinal ruído dos sinais estimados).

Comon sugeriu a utilização do algoritmo proposto para o processamento de sinais capturados por um conjunto de antenas (*antenna array processing*); separação e reconhecimento de fontes; mitigação de *jammer*; entre outros.

Não foi comprovado que o algoritmo não converge para um máximo local. De fato, Comon somente garante que o algoritmo alcança o máximo global no caso de duas fontes na presença de ruído não gaussiano.

A principal desvantagem deste algoritmo é a complexidade para elaboração do código, que exige sofisticadas manipulações algébricas.

4.6.7 – PCA Não Linear

Extensões do algoritmo para Análise de Componentes Principais (PCA) foram propostas inicialmente por pesquisadores finlandeses (Pajunen, 1998; Karhunen, 1997; 1998).

No algoritmo PCA não linear os sinais medidos devem ser inicialmente branqueados, gerando assim, $\mathbf{z}(k)$. A regra de treinamento do algoritmo é obtida através do gradiente estocástico, que minimiza a medida de erro médio quadrático gerado na representação de um vetor por uma projeção não linear $g(\mathbf{W}\mathbf{z}(k))$ em uma base de dimensões reduzidas (Lee, 2000).

O próximo passo é a minimização de uma função custo (4.61), que irá determinar a matriz de separação \mathbf{W} , tal que, $\mathbf{y}(k) = \mathbf{W}\mathbf{z}(k)$, onde $\mathbf{y}(k)$ corresponde a estimativa das fontes originais e \mathbf{W} é ortonormal, ou seja, $\mathbf{W}^T\mathbf{W} = \mathbf{I}$.

$$\mathbf{C}(\mathbf{W}) = \mathbf{ones}(1,n)^T \mathbf{E}[\mathbf{z}(k) - \mathbf{W}g(\mathbf{W}\mathbf{z}(k))^2] \quad (4.61)$$

Onde $\mathbf{C}(\cdot)$ é a função custo a ser minimizada, $\mathbf{ones}(1,n)$ é um vetor linha de comprimento n com elementos iguais a 1.

Reescrevendo (4.61) em sua forma transposta, tem-se que:

$$\mathbf{C}(\mathbf{W}) = \mathbf{E}[(\mathbf{z}^T(k)\mathbf{z}(k) - g^T(\mathbf{W}\mathbf{z}(k)) - \mathbf{z}^T(k)\mathbf{W}g(\mathbf{W}\mathbf{z}(k)) + g^T(\mathbf{W}\mathbf{z}(k))\mathbf{W}^T\mathbf{W}g(\mathbf{W}\mathbf{z}(k)))] \quad (4.62)$$

Uma vez que $\mathbf{z}(k)$ é espacialmente branco isto implica que $\mathbf{E}[\mathbf{W}^T\mathbf{z}(k)\mathbf{z}^T(k)\mathbf{W}] = \mathbf{I}$, considerando ainda que os componentes de $\mathbf{y}(k)$ tem variância unitária, pode-se reescrever (4.62) como:

$$\mathbf{C}(\mathbf{W}) = m + \mathbf{E}[(g^T(\mathbf{y}(k))\mathbf{y}(k) - \mathbf{y}^T(k)g(\mathbf{y}(k)) + g^T(\mathbf{y}(k))g(\mathbf{y}(k)))] \quad (4.63)$$

Onde m é o número de fontes, $g(\mathbf{y}(k)) = \mathbf{y}(k)^3/3$ ou $g(\mathbf{y}(k)) = -\mathbf{y}(k)^3/3$ ou uma função hiperbólica não linear como uma cúbica.

4.6.8 – Algoritmos com Regras Neurais

Pode-se desenvolver algoritmos simples utilizando o princípio de gradiente estocástico descendente (Hyvärinen, 1999; 1997). Por exemplo, pode-se, considerando dados branqueados, calcular o gradiente instantâneo de uma função custo generalizada em função de \mathbf{w} , e normalizando $\|\mathbf{w}\| = 1$, obter-se a regra de treinamento Hebbiana:

$$\Delta \mathbf{w} \propto r \mathbf{x} g(\mathbf{w}^T \mathbf{x}), \text{ para } \mathbf{w} \text{ normalizado,} \quad (4.64)$$

Onde r é uma constante, sendo que sua correta estimativa é um ponto importante para a convergência do algoritmo. Pode-se definir $g(\cdot)$, como qualquer função não linear.

Este tipo de algoritmo foi apresentado pela primeira vez por Delfosse et al. (Delfosse, 1995), utilizando *kurtosis* e com $g(\mathbf{y}) = \mathbf{y}^4$.

Para estimar vários componentes independentes é necessário que se utilize um sistema com várias unidades, ou neurônios, sendo que cada um deles será treinado de acordo com uma regra de treinamento. O sistema deve ainda conter algum mecanismo de realimentação entre as unidades.

4.6.9 – Escolha do Algoritmo

A escolha do algoritmo para solucionar o problema de ICA deve ser feita levando em consideração dois pontos principais como discutido em (Hyvärinen, 1999):

- Escolher entre estimar todos os componentes independentes ao mesmo tempo ou estimar um conjunto de componentes, um a um.

A maioria dos pesquisadores tem optado por estimar todos os componentes simultaneamente, embora, na prática, a segunda opção seja computacionalmente mais adequada.

- Escolher entre algoritmos adaptativos e algoritmos tipo *batch*, ou em bloco.

Os algoritmos adaptativos têm sido os mais empregados, embora computacionalmente, os algoritmos tipo *batch* sejam mais recomendados devido a sua velocidade de processamento.

Para o caso adaptativo, os algoritmos são obtidos, na maioria dos casos, por métodos de gradiente estocástico. No caso onde todos os componentes são estimados ao mesmo tempo, os algoritmos mais populares nesta classe são os baseados em gradiente natural, ou relacionados, como INFOMAX (Bell, 1995; Amari, 1996; Cardoso, 1996a; Hyvärinen, 1999; Cichocki, 2002).

Métodos de gradiente estocástico resultam também em algoritmos adaptativos que maximizam a negentropia ou suas aproximações (Hyvärinen, 1999; 2001).

Para o caso onde os cálculos serão feitos em blocos, podem-se encontrar algoritmos mais eficientes, como os baseados em cumulantes (Cardoso, 1993; Comon, 1994) para pequenas dimensões e o FASTICA, baseado em iterações de ponto fixo (Hyvärinen, 1999; 2000).

4.7 – ICA na Presença de Ruído

A estimativa da matriz de separação e/ou mistura torna-se ainda mais difícil quando há presença de ruído aditivo no modelo para ICA, tal que a maioria dos algoritmos consideram que o ruído aditivo pode ser desprezado, ou seja, $\mathbf{v}(k)=0$.

Alguns métodos consideram ainda o ruído como uma fonte desconhecida que deve ser separada ou recuperada como as demais; neste caso, no máximo, uma das fontes de ruído deve ser gaussiana e as demais devem ser não-gaussianas. O problema aqui é com relação ao esforço computacional necessário para separar fontes indesejáveis.

Para solucionar esta questão cai-se em um problema clássico de cancelamento de ruído, que, em geral, é complexo e muitas vezes sem solução. Isto ocorre porque se trabalha com $(n + v)$ fontes desconhecidas, onde n é o número de fontes de interesse e v é o número de fontes de ruído, o que corresponde a problemas onde se têm mais fontes do que sensores.

A maioria dos algoritmos que consideram ruído aditivo assume que o ruído é gaussiano e que fontes e ruído são independentes. Estas considerações restringem a solução do problema a algoritmos que trabalham com estatísticas de ordem superior e que não são afetadas por ruído gaussiano (Hyvärinen, 1999; Cichocki, 2002). Métodos baseados em cumulantes podem, por exemplo, ser utilizados no caso de ICA na presença de ruído aditivo, embora a falta de robustez destes algoritmos possa, em alguns casos, comprometer o resultado do processo de separação.

Embora existam vários algoritmos relacionados a ICA na presença de ruído (Cichocki, 2002; Douglas, 1995; Hyvärinen, 2001), diferentes extensões destas estruturas básicas indicam direções importantes para pesquisas futuras, como por exemplo, a estimativa de modelos com mais fontes do que sensores.

4.8 – Considerações Finais

Neste capítulo foi apresentada uma revisão do estado da arte sobre a ICA, sendo dado especial destaque às funções custo e aos algoritmos.

Como descrito, a ICA é um método estatístico no qual dados aleatórios observados são linearmente transformados em componentes que são estatisticamente independentes um dos outros e ao mesmo tempo são tão não gaussianos quanto possível (Hyvärinen, 2000). Esta idéia de maximizar a não gaussianidade foi utilizada para derivar diferentes funções custo, cuja otimização permite estimar o modelo para ICA. Além desta idéia, métodos clássicos como a estimativa da máxima verossimilhança ou minimização da informação mútua também podem ser utilizados para estimar componentes independentes. A ICA encontra aplicações em diferentes áreas, sendo importante ressaltar que ela é a principal ferramenta utilizada para solucionar o problema da BSS.

Mostrou-se ainda que a escolha da técnica de ICA a ser utilizada depende basicamente de dois fatores. Primeiramente, deve-se escolher entre estimar todos os componentes de uma vez ou estimar um conjunto de componentes, possivelmente um a um. É importante destacar que a maioria dos algoritmos utiliza a primeira opção embora, na prática, a segunda opção pareça ser interessante (Hyvärinen, 1999). O segundo ponto que deve ser considerado na escolha da técnica para ICA é se será utilizado um algoritmo adaptativo ou um que realize processamento em blocos (*batch mode*). Mais uma vez a primeira opção é a mais adotada, principalmente por sua simplicidade computacional.

As pesquisas na área de ICA já atingiram certa maturidade, tanto no que diz respeito aos conceitos teóricos, quanto ao desenvolvimento de algoritmos, o que não significa que as pesquisas nesta área estão esgotadas. Embora a probabilidade de avanços notáveis nas técnicas para estimar o modelo básico de ICA seja pequena; as extensões das técnicas existentes mostram direções para novos trabalhos. Além disso, existem questões relativas à implementação dos algoritmos existentes que necessitam de melhor compreensão. Novas aplicações também vêm sendo estudadas nas áreas de telecomunicações, engenharia biomédica, processamento de imagens, entre outras, abrindo promissoras frentes de pesquisa (Hyvärinen, 2001). Desta forma, justifica-se o sentimento de que a ICA/BSS continuará sendo objeto de pesquisa ainda nos próximos anos.

Capítulo 5 – AMUSE

5.1 – Introdução

O Algoritmo para Extração de Múltiplos Sinais Desconhecidos (AMUSE – *Algorithm for Multiple Unknown Signals Extraction*) foi inicialmente proposto em (Tong, 1990) para solucionar o problema da separação ou identificação cega de sinais através de diagonalizações simultâneas de \mathbf{R}_x e $\mathbf{R}_x(\tau)$ - que são respectivamente a matriz de covariância e a matriz de covariância atrasada no tempo dos sinais medidos pelos sensores -, desde que $\mathbf{R}_s^{-1}\mathbf{R}_s(\tau)$ (respectivamente, o inverso da matriz de covariância e a matriz de covariância atrasada no tempo dos sinais originais) tenham elementos distintos em sua diagonal.

A motivação para o desenvolvimento do AMUSE surgiu das desvantagens apresentadas por algoritmos que consideram que os sinais fontes são estatisticamente independentes. Tais algoritmos são incapazes de trabalhar com mais do que um sinal gaussiano, uma vez que os momentos de ordem superior de sinais gaussianos são completamente especificados através de momentos de segunda ordem. Assim, para mitigar as desvantagens dos algoritmos que utilizam HOS, o AMUSE utiliza apenas SOS. Para encontrar a matriz de separação, ao invés de calcular momentos de ordem superior, o AMUSE decompõe a matriz de covariância das medidas dos sensores correlacionada com sua versão atrasada no tempo.

A principal condição ou restrição para a utilização do AMUSE é que os sinais originais sejam temporalmente descorrelacionados; tal restrição é menos rigorosa do que a restrição de independência estatística (Cichocki, 2002; Tong, 1991). Para compreender, então, como o AMUSE realiza a separação cega, este capítulo foi dividido em quatro seções, sendo que a seção 5.2 apresenta o modelo e as considerações estatísticas para a solução do problema da separação utilizando o AMUSE; a seção 5.3 discorre sobre a identificação de fontes descorrelacionadas; a seção 5.4 descreve como o AMUSE é implementado.

5.2 – Modelo e Considerações Gerais

O modelo utilizado pelo AMUSE é matematicamente descrito pela equação (5.1).

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{v}(k) \quad (5.1)$$

Onde $\mathbf{s}(k) = \mathbf{s} = [s_1(k), \dots, s_n(k)]$ é o vetor que representa os sinais originais, os quais se deseja recuperar, $\mathbf{x}(k) = \mathbf{x} = [x_1(k), \dots, x_m(k)]$ as observações ou medidas feitas pelos sensores e $\mathbf{y}(k) = \mathbf{y} = [y_1(k), \dots, y_n(k)]$ as fontes resultantes ou estimadas pelo processo de separação; o vetor $\mathbf{v}(k)$ representa o ruído adicionado ao processo; \mathbf{A} é a matriz de parâmetros que caracteriza o meio ou canal de transmissão e \mathbf{W} é a matriz de separação dos sinais lidos pelos sensores. Assim, como nos capítulos anteriores, o número de fontes será considerado igual ao número de sensores o que implica que $m=n$.

O canal de transmissão ou a matriz de mistura é considerada linear e sem memória, uma vez que modelo é relativamente preciso para inúmeras aplicações, dentre elas, o processamento de conjunto de sinais. Além desta, outras considerações iniciais devem ser feitas sobre o modelo de separação:

- (C1) A matriz \mathbf{A} é de posto completo;
- (C2) Os sinais originais $\mathbf{s}(k)$ são processos estacionários de média zero com matriz de covariância dada por $\mathbf{R}_s = \mathbf{E}[\mathbf{s}(k)\mathbf{s}^T(k)]$;
- (C3) O ruído $\mathbf{v}(k)$ é um processo estacionário branco gaussiano com matriz de covariância $\mathbf{R}_v = \sigma^2 \mathbf{I}_m$;
- (C4) O sinais $\mathbf{s}(k)$ e $\mathbf{v}(k)$ são estatística e mutuamente independentes.

5.2.1 – Espaço de Identificação e Relação de Preservação da Forma de Onda

Em (Tong, 1990) foi introduzido o conceito de espaço de identificação, que inclui todas as fontes $\mathbf{s}(k)$ e todas as matrizes de mistura \mathbf{A} capazes de gerar o mesmo conjunto de observações $\mathbf{x}(k)$. Assim, dadas as medidas $\mathbf{x}(k)$ e o ruído $\mathbf{v}(k)$, o espaço de identificação $I_{(\mathbf{x},\mathbf{v})}$ é definido em (5.2).

$$I_{(\mathbf{x},\mathbf{v})} = \{(\mathbf{A}, \mathbf{s}) \mid \mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{v}\} \quad (5.2)$$

Desta forma, $I_{(\mathbf{x},\mathbf{v})}$ descreve o conjunto de todas as potenciais soluções para o problema da separação dados $\mathbf{x}(k)$ e $\mathbf{v}(k)$.

Assim, se (\mathbf{A}, \mathbf{s}) são os valores reais da matriz de mistura e dos sinais originais que produziram $\mathbf{x}(k)$ na presença de $\mathbf{v}(k)$ então, $(\hat{\mathbf{A}}, \hat{\mathbf{s}}) \in I_{(\mathbf{x}, \mathbf{v})}$ se e somente se, satisfizer o modelo (5.1) e suas restrições e se $(\hat{\mathbf{A}}, \hat{\mathbf{s}}) = (\mathbf{A}, \mathbf{s})$.

Embora todo elemento pertencente ao espaço de identificação seja uma estimativa legítima, deve-se levar em consideração as ambigüidades ou indeterminações inerentes ao problema da separação cega. Isto significa que nem todo conjunto $(\hat{\mathbf{A}}, \hat{\mathbf{s}}) \in I_{(\mathbf{x}, \mathbf{v})}$ é uma estimativa adequada de (\mathbf{A}, \mathbf{s}) , uma vez que nem a forma de onda nem as propriedades estatísticas de $\hat{\mathbf{s}}(k)$ podem estar de acordo com $\mathbf{s}(k)$.

Com o objetivo de minimizar as ambigüidades do processo de separação e considerando que, em aplicações práticas, a ambigüidade com relação à amplitude dos sinais estimados é tolerável, contanto que a sua forma de onda seja preservada e, a ordem com que os sinais são estimados não é importante, foi definida a Relação de Preservação da Forma de Onda (WPR – *Waveform Preservation Relation*) no espaço $I_{(\mathbf{x}, \mathbf{v})}$ (Tong, 1991; 1993).

A WPR é definida para os pares $(\hat{\mathbf{A}}, \hat{\mathbf{s}}), (\mathbf{A}, \mathbf{s})$ se:

$$\hat{\mathbf{A}} = \mathbf{A}\mathbf{\Lambda}^{-1}\mathbf{P}^T \quad (5.3)$$

$$\hat{\mathbf{s}}(k) = \mathbf{\Lambda}\mathbf{P}\mathbf{s}(k) \quad (5.4)$$

Sendo que (5.3) e (5.4) são válidas para qualquer matriz de permutação \mathbf{P} e qualquer matriz não singular diagonal $\mathbf{\Lambda}$. Por conveniência quando $(\hat{\mathbf{A}}, \hat{\mathbf{s}})$ está relacionado pela WPR com (\mathbf{A}, \mathbf{s}) utiliza-se a notação $(\hat{\mathbf{A}}, \hat{\mathbf{s}}) \sim (\mathbf{A}, \mathbf{s})$

Como mostrado em (Tong, 1991), a WPR nada mais é do que uma relação de equivalência, a partir da qual é possível definir uma classe de sinais equivalentes a (\mathbf{A}, \mathbf{s}) .

Ainda sobre a WPR, é importante observar que ela não garante somente a preservação da forma de onda dos sinais, mas também algumas propriedades estatísticas dos sinais originais e propriedades algébricas da matriz de mistura. Assim, se $(\hat{\mathbf{A}}, \hat{\mathbf{s}}) \sim (\mathbf{A}, \mathbf{s})$, têm-se que:

- $\hat{\mathbf{R}}_s(\tau)$ é diagonal se $\mathbf{R}_s(\tau)$ for diagonal; onde $\mathbf{R}_s(\tau) = \mathbf{E}[\mathbf{s}(k)\mathbf{s}(k-\tau)^T]$ para qualquer atraso τ .
- Os vetores coluna de $\hat{\mathbf{A}}$ são ortogonais se os vetores de \mathbf{A} forem ortogonais.

A partir das afirmações supra-citadas pode-se dizer que o objetivo da separação cega é encontrar uma classe de equivalências que contenha (\mathbf{A}, \mathbf{s}) dentro do espaço $I_{(\mathbf{x}, \mathbf{v})}$.

5.2.2 – Identificidade

Todo conjunto de propriedades atribuído a (\mathbf{A}, \mathbf{s}) reduzem o número de soluções $(\hat{\mathbf{A}}, \hat{\mathbf{s}})$ aceitáveis, desde que $(\hat{\mathbf{A}}, \hat{\mathbf{s}})$ também apresentem estas propriedades. Surge então a questão sobre a existência de um conjunto de propriedades ou um modelo que reduza o conjunto de soluções, mantendo ainda a WPR. Se tal modelo existir, então se pode dizer que (\mathbf{A}, \mathbf{s}) é identificável.

O conceito de identificidade (*identifiability*) é definido da seguinte forma (Tong, 1991; Buccigrossi, 1997):

- Seja \mathbf{M} um conjunto de $(\hat{\mathbf{A}}, \hat{\mathbf{s}})$ que satisfaz o modelo proposto em (5.1) e suas restrições. Se para todo $(\hat{\mathbf{A}}, \hat{\mathbf{s}}) \in I_{(x,v)} \cap \mathbf{M}, (\hat{\mathbf{A}}, \hat{\mathbf{s}}) \sim (\mathbf{A}, \mathbf{s})$ então, $(\mathbf{A}, \mathbf{s}) \in I_{(x,v)}$ é identificável com relação a \mathbf{M} .

É importante observar que o conceito de identificidade é definido em relação a um determinado modelo. Por exemplo, com base no conceito de identificidade, se for feita uma análise para verificar se (\mathbf{A}, \mathbf{s}) é identificável a partir do modelo (5.1) e suas restrições, a resposta será negativa. Isto ocorre porque para qualquer matriz não singular \mathbf{H} , $(\mathbf{A}\mathbf{H}^{-1}, \mathbf{H}\mathbf{s}) \in I_{(x,v)}$ e está de acordo com o modelo (5.1) e suas restrições; porém, $(\mathbf{A}\mathbf{H}^{-1}, \mathbf{H}\mathbf{s}) \neq (\mathbf{A}, \mathbf{s})$ para todo \mathbf{H} . Assim, (\mathbf{A}, \mathbf{s}) não é identificável para o modelo (5.1). Para que o modelo (5.1) passe a ser identificável é necessário que sejam incluídas novas restrições como será apresentado na seção (5.3).

5.3 - Identificação de Fontes Descorrelacionadas

A matriz de mistura original \mathbf{A} e as fontes originais $\mathbf{s}(k)$ são identificáveis com relação a \mathbf{M} , onde \mathbf{M} é o conjunto de todos os valores (\mathbf{A}, \mathbf{s}) que satisfazem as condições (C5) e (C6) (Tong, 1991).

(C5) As fontes $\{s_i(k), i = 1, 2, \dots, n\}$ são descorrelacionadas;

(C6) Existe uma constante $\tau > 0$, tal que:

$$\frac{\mathbf{E}[s_i(k)s_i(k-\tau)]}{\mathbf{E}[s_i(k)^2]} \neq \frac{\mathbf{E}[s_j(k)s_j(k-\tau)]}{\mathbf{E}[s_j(k)^2]}, \text{ para } i \neq j \quad (5.5)$$

As restrições (C5) e (C6) compõe o chamado Teorema da Identificação de Fontes Descorrelacionadas definido em (Tong,1991).

A seguir será apresentada a formulação matemática que justifica a utilização das restrições (C5) e (C6).

Para comprovar que as restrições (C5) e (C6) tornam o modelo identificável, deve-se observar que para qualquer $(\hat{\mathbf{A}}, \hat{\mathbf{s}}) \in I_{(x,v)} \cap \mathbf{M}, (\hat{\mathbf{A}}, \hat{\mathbf{s}}) \sim (\mathbf{A}, \mathbf{s})$, a partir da definição de $I_{(x,v)}$, para qualquer $(\hat{\mathbf{A}}, \hat{\mathbf{s}}) \in I_{(x,v)} \cap \mathbf{M}$ tem-se que:

$$\hat{\mathbf{A}}\hat{\mathbf{s}}(k) = \mathbf{A}\mathbf{s}(k) \quad (5.6)$$

Considerando agora que $\mathbf{R}_{\hat{\mathbf{s}}} = \mathbf{E}[\hat{\mathbf{s}}(k)\hat{\mathbf{s}}^T(k)]$ e $\mathbf{R}_{\mathbf{s}} = \mathbf{E}[\mathbf{s}(k)\mathbf{s}^T(k)]$, tem-se que:

$$\hat{\mathbf{A}}\mathbf{R}_{\hat{\mathbf{s}}}\hat{\mathbf{A}}^T = \mathbf{A}\mathbf{R}_{\mathbf{s}}\mathbf{A} \quad (5.7)$$

Onde $\mathbf{R}_{\hat{\mathbf{s}}}$ e $\mathbf{R}_{\mathbf{s}}$ são diagonais, devido ao fato das fontes serem decorrelacionadas. Considerando que:

$$\mathbf{R} \equiv \mathbf{A}\mathbf{R}_{\mathbf{s}}\mathbf{A}^T, \quad (5.8)$$

A decomposição singular (SVD) de \mathbf{R} é dada por:

$$\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \quad (5.9)$$

Onde \mathbf{U} é uma matriz ortonormal $n \times m$ ($n=m$) e $\mathbf{\Lambda}$ é uma matriz diagonal positiva. A partir das equações (5.7), (5.8) e (5.9) pode-se escrever:

$$\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T = \mathbf{A}\mathbf{R}_{\mathbf{s}}\mathbf{A}^T = \hat{\mathbf{A}}\mathbf{R}_{\hat{\mathbf{s}}}\hat{\mathbf{A}}^T \quad (5.10)$$

Considerando agora que:

$$\mathbf{T} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T \quad (5.11)$$

Multiplicando-se o lado esquerdo de (5.10) por \mathbf{T} e o lado direito por \mathbf{T}^T obtém-se (5.12) e (5.13).

$$\mathbf{TAR}_s\mathbf{A}^T\mathbf{T}^T = \mathbf{I} \quad (5.12)$$

$$\mathbf{TAR}_\hat{s}\mathbf{A}^T\mathbf{T}^T = \mathbf{I} \quad (5.13)$$

As igualdades (5.12) e (5.13) implicam que as matrizes \mathbf{V} e $\hat{\mathbf{V}}$, definidas em (5.14) e (5.15), respectivamente, são matrizes ortogonais.

$$\mathbf{V} = \mathbf{TAR}_s^{1/2} \quad (5.14)$$

$$\hat{\mathbf{V}} = \mathbf{T}\hat{\mathbf{A}}\mathbf{R}_\hat{s}^{1/2} \quad (5.15)$$

Desde que ambos \mathbf{R}_s e $\mathbf{R}_\hat{s}$ são diagonais, \mathbf{TA} e $\mathbf{T}\hat{\mathbf{A}}$ pertencem a uma classe de equivalência e \mathbf{T} induz uma transformação ortogonal. Como a transformação ortogonal preserva a identificidade pode-se mostrar que $\mathbf{TA} \sim \mathbf{T}\hat{\mathbf{A}}$ ou, de forma equivalente, $\mathbf{V} \sim \hat{\mathbf{V}}$.

Seja agora um valor τ , tal que a condição (C6) seja satisfeita, tem-se então, a partir de (5.10), que:

$$\mathbf{AR}_s(\tau)\mathbf{A}^T = \hat{\mathbf{A}}\mathbf{R}_\hat{s}(\tau)\hat{\mathbf{A}}^T \quad (5.16)$$

Onde $\mathbf{R}_s = \mathbf{E}[\mathbf{s}(k)\mathbf{s}^T(k-\tau)]$ e $\mathbf{R}_\hat{s} = \mathbf{E}[\hat{\mathbf{s}}(k)\hat{\mathbf{s}}^T(k-\tau)]$ são matrizes diagonais. Conseqüentemente tem-se ainda que:

$$\mathbf{VR}_s^{-1}(\tau)\mathbf{R}_s(\tau)\mathbf{V}^T = \hat{\mathbf{V}}\mathbf{R}_\hat{s}^{-1}(\tau)\mathbf{R}_\hat{s}(\tau)\hat{\mathbf{V}}^T \quad (5.17)$$

É importante observar que o vetor coluna das matrizes ortogonais $\hat{\mathbf{V}}$ e \mathbf{V} são os autovetores da matriz \mathbf{R}_τ como mostrado em (5.18). E os componentes das diagonais de ambos $\mathbf{R}_s^{-1}\mathbf{R}_s(\tau)$ e $\mathbf{R}_\hat{s}^{-1}\mathbf{R}_\hat{s}(\tau)$ são os autovalores de \mathbf{R}_τ .

$$\mathbf{R}_p \equiv \mathbf{VR}_s^{-1}\mathbf{R}_s(\tau)\mathbf{V}^T \quad (5.18)$$

Desde que todos os autovalores de \mathbf{R}_τ são distintos, existe uma matriz de permutação \mathbf{P} e uma matriz diagonal $\mathbf{\Lambda}$ (com 1 ou -1 como componentes de sua diagonal), tal que:

$$\hat{\mathbf{V}} = \mathbf{VPA} \quad (5.20)$$

Ou, de outra forma, $\hat{\mathbf{V}} \sim \mathbf{V}_0$.

Conseqüentemente, $\mathbf{TA} \sim \mathbf{T}\hat{\mathbf{A}}$, e, conforme o Teorema da Preservação da Identificidade - que diz que $(\mathbf{A}, \mathbf{s}) \in I_{(\mathbf{x}, \mathbf{v})}$ é identificável com relação a algum modelo \mathbf{M} se, e somente se, $(\mathbf{B}, \mathbf{s}) \in I_{(\mathbf{x}, \mathbf{v})}$ for identificável com relação a \mathbf{M} , sendo \mathbf{B} , uma transformação ortogonal de \mathbf{A} - pode-se dizer que $\mathbf{A} \sim \hat{\mathbf{A}}$ (Cichocki, 2002).

5.4 – Descrição do AMUSE

5.4.1 – Princípio Fundamental

Nesta seção os conceitos apresentados nas seções anteriores serão condensados de forma a que o processo de separação realizado pelo AMUSE possa ser melhor compreendido.

Com base nas considerações (C1) a (C6) pode-se verificar que o AMUSE trabalha com fontes que são espacialmente descorrelacionadas, mas temporalmente correlacionadas e considera ainda que o ruído é branco gaussiano. Além disso, a matriz de correlação de $\mathbf{x}(k)$ e a sua atrasada no tempo satisfazem as relações (5.21) e (5.22).

$$\mathbf{R}_x - \sigma_v^2 \mathbf{I}_m = \mathbf{A} \mathbf{R}_s \mathbf{A}^T \quad (5.21)$$

$$\mathbf{R}_x(\tau) = \mathbf{A} \mathbf{R}_s(\tau) \mathbf{A}^T \quad (5.22)$$

Onde τ é qualquer valor de atraso diferente de zero e \mathbf{R}_s e $\mathbf{R}_s(\tau)$ são matrizes diagonais, desde que as fontes sejam consideradas espacialmente descorrelacionadas, ou seja, passaram por algum processo de descorrelação espacial ou branqueamento.

Considerando que $\tilde{\mathbf{R}}_x = \mathbf{R}_x - \sigma_v^2 \mathbf{I}_m$, a matriz de separação pode ser encontrada através de diagonalizações simultâneas de \mathbf{R}_x e $\mathbf{R}_x(\tau)$, contanto que $\mathbf{R}_s^{-1} \mathbf{R}_s(\tau)$ tenha elementos da diagonal distintos. Esta é a idéia fundamental do AMUSE.

Como os demais algoritmos para separação cega, o AMUSE realiza antes da separação propriamente dita um branqueamento nos dados medidos pelos sensores como descrito no capítulo 3.

A seguir, serão apresentados os passos para implementação do AMUSE (Cichocki, 2002; Tong, 1990).

5.4.2 – Passos para Implementação do AMUSE

Para sua implementação o AMUSE segue os seguintes passos:

1 – Pré-Processamento: Branqueamento

1.1 - Calcular a matriz de correlação dos sinais dos sensores $\mathbf{x}(k)$;

$$\mathbf{R}_x = \mathbf{E}[\mathbf{x}(k)\mathbf{x}^T(k)] = \frac{1}{N} \sum_{k=1}^n \mathbf{x}(k)\mathbf{x}^T(k) \quad (5.23)$$

1.2 - Calcular a SVD da matriz \mathbf{R}_x ;

1.3 – Estimar o número de fontes n , - caso não seja conhecido -, a partir dos valores singulares mais significativos;

1.4 - Caso seja considerado ruído aditivo e quando o número de fontes é conhecido ou foi estimado a partir dos valores singulares mais significativos, estimar a variância do ruído como o valor médio dos $(m-n)$ autovalores (ou valores singulares) menos significativos. A seguir, calcular a matriz de branqueamento \mathbf{Q} ;

1.5 – Caso não seja considerado ruído aditivo e o número de fontes seja conhecido, como nesta dissertação, pode-se calcular a matriz de branqueamento \mathbf{Q} a partir de:

$$\mathbf{Q} = (\mathbf{R}_x)^{-1/2}; \quad (5.24)$$

1.6 - Executar o branqueamento, tal que $\mathbf{z}(k)$ seja um vetor branco;

$$\mathbf{z}(k) = \mathbf{Q} * \mathbf{x} \quad (5.25)$$

2 – Separação Utilizando EVD

2.1 - Estimar a matriz de covariância do vetor $\mathbf{z}(k)$, com média zero, para um atraso $\tau \neq 0$. Em geral $\tau=1$ dá bons resultados, mas podem ser tentados outros valores:

$$\mathbf{R}_z(\tau) = \frac{1}{N} \sum_{k=1}^N \mathbf{z}(k) \mathbf{z}^T(k-\tau) \quad (5.26)$$

2.2 – Calcular a EVD de $\mathbf{R}_z(\tau)$:

$$\mathbf{R}_z(\tau) = \mathbf{V}_z \mathbf{D}_z \mathbf{V}_z^T \quad (5.27)$$

2.3- Verificar se para o atraso τ escolhido todos os valores da matriz diagonal resultante da decomposição realizada em (5.30) são distintos, senão repetir a partir do passo 2.1 para um novo valor de τ . Por outro lado, se os autovalores são distintos e suficientemente distantes um dos outros, então pode-se estimar corretamente a matriz de mistura.

2.4 – Calcular a matriz de separação \mathbf{W} e/ou os componentes estimados $\mathbf{y}(k)$;

$$\mathbf{W} = \mathbf{V}_z^T * \mathbf{Q} \quad (5.28)$$

$$\mathbf{y}(k) = \mathbf{W} * \mathbf{x}(k) \quad (5.29)$$

Em geral, para um atraso $\tau=1$ o algoritmo separa com sucesso fontes coloridas, isto é, sinais com estrutura temporal, com diferentes espectros de potência.

Pela descrição do algoritmo pode-se comprovar que ele pertence à classe dos métodos de SOS para separação cega, uma vez que ele utiliza apenas a matriz de covariância atrasada no tempo para realizar a separação das fontes. A matriz de covariância (SOS) pode ser estimada com precisão a partir de um número de dados menor do que os em estatísticas de alta ordem. Assim sendo, quando o número de dados observados é relativamente pequeno é mais vantajoso utilizar métodos baseados em estatísticas de segunda ordem (Cichocki, 2002).

A principal desvantagem do AMUSE é que sua precisão é seriamente deteriorada quando da presença de ruído aditivo. A fim de melhorar o desempenho do algoritmo, para dados

corrompidos por ruído, pode-se tentar modificar o algoritmo utilizando diferentes valores de atraso (p) ou adotando um método de branqueamento mais robusto.

5.4.3 – Decomposição em Valores Singulares

Para realizar a separação cega o AMUSE utiliza uma ferramenta matemática denominada SVD (*Singular Value Decomposition*). Esta ferramenta é muito utilizada em problemas de processamento de sinais e identificação.

Nesta seção será apresentado o procedimento para a implementação da SVD de uma matriz real \mathbf{X} . Este procedimento é útil quando se trabalha com *softwares* que não implementam a SVD de forma direta.

A SVD de uma matriz real $\mathbf{X} = [x(1), \dots, x(N)] \in \mathbf{R}^{m \times N}$ é tal que:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (5.30)$$

Onde $\mathbf{U} \in \mathbf{R}^{m \times N}$ e $\mathbf{V} \in \mathbf{R}^{N \times N}$ são matrizes ortogonais e $\mathbf{\Sigma} \in \mathbf{R}^{m \times N}$ é uma matriz pseudo-diagonal (Golub, 1989; Cichocki, 2002).

Para sinais não corrompidos por ruído, pode-se escrever (5.30) através da seguinte decomposição:

$$\mathbf{X} = (\mathbf{U}_S, \mathbf{U}_N) \begin{pmatrix} \mathbf{\Sigma}_S & 0 \\ 0 & 0 \end{pmatrix} (\mathbf{V}_S, \mathbf{V}_N)^T \quad (5.31)$$

Onde $\mathbf{U}_S = (u_1, \dots, u_n)$, $\mathbf{\Sigma}_S = \text{diag}(\sigma_1, \dots, \sigma_n)$ e $\mathbf{U}_N = (u_{n+1}, \dots, u_m)$.

O conjunto de matrizes $(\mathbf{U}_S, \mathbf{\Sigma}_S, \mathbf{V}_S)$ representa o subespaço do sinal e o conjunto de matrizes $(\mathbf{U}_N, \mathbf{\Sigma}_N, \mathbf{V}_N)$ representa o subespaço nulo ou, para sinais corrompidos por ruído, o subespaço do ruído.

Com base em (5.1) as colunas de \mathbf{U} correspondem aos valores singulares diferentes de zero que expandem o espaço das colunas de \mathbf{X} e são chamados vetores singulares esquerdos; de forma similar, as colunas de \mathbf{V} são chamadas vetores singulares direitos e expandem o espaço linha de \mathbf{X} . Assim sendo, a SVD de \mathbf{X} pode ser escrita como:

$$\mathbf{X} = \mathbf{U}_S \mathbf{\Sigma}_S \mathbf{V}_S^T = \sum_{i=1}^n \sigma_i u_i v_i^T \quad (5.33)$$

Considerando que a SDV $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ tem posto $n \leq m \leq N$, tem-se que (Cichocki, 2002):

$$\mathbf{X}\mathbf{X}^T = \mathbf{U}\mathbf{\Sigma}_1^2\mathbf{V}^T \quad (5.33)$$

$$\mathbf{X}^T\mathbf{X} = \mathbf{U}\mathbf{\Sigma}_2^2\mathbf{V}^T \quad (5.34)$$

Onde $\mathbf{\Sigma}_1 = \text{diag}(\sigma_1, \dots, \sigma_m)$ e $\mathbf{\Sigma}_2 = \text{diag}(\sigma_1, \dots, \sigma_N)$.

As equações anteriores mostram que os valores singulares de \mathbf{X} são iguais à raiz quadrada dos autovalores de $\mathbf{X}\mathbf{X}^T$. É importante salientar ainda que a PCA ou EVD da matriz de covariância de \mathbf{R}_x pode ser implementada através de algoritmos de SVD; sendo que somente nos casos em que \mathbf{X} é uma matriz de grandes dimensões os algoritmos para SVD têm custo computacional maior que os algoritmos para PCA.

No caso de \mathbf{X} ser corrompido por ruído aditivo \mathbf{N} – branco, com variância σ_N^2 , tem-se que $\tilde{\mathbf{X}} = \mathbf{X} + \mathbf{N}$ e a matriz de covariância do ruído é dada por:

$$\mathbf{E}\left[\frac{\mathbf{N}\mathbf{N}^T}{N}\right] = \sigma_N^2\mathbf{I}_m \quad (5.35)$$

A partir da equação (5.35) tem-se:

$$\mathbf{E}\left[\frac{\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T}{N}\right] = \mathbf{E}\left[\frac{\mathbf{X}\mathbf{X}^T}{N}\right] + \sigma_N^2\mathbf{I}_m \quad (5.36)$$

A partir de (5.36) pode-se afirmar que para um grande valor de atraso p , a SVD de $\tilde{\mathbf{X}}$ é aproximadamente igual a:

$$\tilde{\mathbf{X}} \approx \mathbf{U}(\mathbf{\Sigma}_1^2 + N\sigma_N^2\mathbf{I}_m)^{1/2}\tilde{\mathbf{V}}^T \quad (5.37)$$

Onde $\tilde{\mathbf{V}}^T$ é qualquer matriz ortogonal.

A partir da equação (5.37), nota-se que, para grandes valores de N e pequenas variâncias do ruído, os valores singulares da matriz de covariância de $\tilde{\mathbf{X}}$, são praticamente insensíveis às perturbações adicionadas aos elementos de \mathbf{X} . Esta consideração mostra que a SVD é um método confiável e robusto (Cichocki, 2002).

Na teoria, é possível recuperar a matriz \mathbf{XX}^T , sem ruído, subtraindo $N\sigma_N^2 \mathbf{I}_m$ de Σ_1 .

5.5 – Considerações Finais

Neste capítulo foi descrito o Algoritmo para Extração de múltiplos Sinais Desconhecidos (AMUSE). Foi feita uma descrição matemática dos principais conceitos que serviram de base para o desenvolvimento do AMUSE. Realizou-se ainda uma descrição detalhada para a implementação computacional do algoritmo.

A motivação para o desenvolvimento do AMUSE surgiu das desvantagens apresentadas por algoritmos que consideram que os sinais fontes são estatisticamente independentes e que são incapazes de trabalhar com mais do que um sinal gaussiano. Assim, ao invés de trabalhar com a restrição de independência estatística das fontes, o AMUSE considera que os sinais originais são temporalmente descorrelacionados, que é uma restrição menos rigorosa do que a primeira. As simulações computacionais apresentadas em (Tong,1990) mostram o bom desempenho do AMUSE para diferentes aplicações mesmo quando da presença de ruído aditivo.

Capítulo 6 – Algoritmo JADE

6.1 – Introdução

Neste capítulo será descrito o algoritmo de Diagonalização Conjunta Aproximada de Automatrizes (JADE – *Joint Approximate Diagonalization of Eigen-matrices*) proposto por Cardoso et al. (Cardoso, 1993) para solucionar o problema da conformação cega de feixe (*blind beamforming*). Este algoritmo foi escolhido para ser estudado de forma mais detalhada principalmente por suas aplicações em telecomunicações e por ser um bom exemplo de algoritmo para BSS baseado em estatísticas de ordem superior.

A principal restrição utilizada pelo algoritmo JADE para solucionar o problema da separação cega é que as fontes a serem separadas apresentem-se estatisticamente independentes. O algoritmo explora esta independência estatística através de cumulantes de quarta ordem, ou melhor, através da diagonalização conjunta das matrizes de cumulantes de 4ª ordem dos dados observados. Desta forma, o JADE reduz o conjunto de dados observados a um pequeno conjunto de estatísticas. Para realizar a diagonalização conjunta, o algoritmo emprega uma técnica semelhante ao método jacobiano de diagonalização e por este motivo ele é conhecido como algoritmo jacobiano.

Uma das principais vantagens apresentadas pelo JADE é que ele pode se mover em passos macroscópicos através do espaço de parâmetros, solucionando o problema causado pela escolha incorreta do passo de adaptação, comum em algoritmo que utilizam técnicas como o do gradiente natural (Cichocki, 2002). Por outro lado, a implementação computacional do JADE apresenta grande complexidade, principalmente no que diz respeito aos cálculos das estatísticas de ordem superior.

Assim, com o objetivo de descrever o algoritmo JADE pode-se dividir este capítulo em duas partes. Na primeira, é feita uma introdução e uma breve revisão dos conceitos matemáticos utilizados na implementação do algoritmo e na segunda parte, é feita uma descrição do algoritmo propriamente dito.

6.2 – Notações e Considerações Iniciais

Considere um sistema no qual $\mathbf{s}(k)$ sinais ou fontes são emitidos por um meio ou canal de transmissão representado pela matriz de mistura \mathbf{A} e que possivelmente há ruído $\mathbf{v}(k)$ adicionado ao sistema. Para observar este sistema são empregados sensores cujas saídas são representadas por um vetor $\mathbf{x}(k)$, que pode ser descrito matematicamente da seguinte forma:

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{v}(k) \quad (6.1)$$

O objetivo da separação cega é identificar a matriz de mistura \mathbf{A} ou a matriz de separação \mathbf{W} , de tal sorte que os sinais originais $\mathbf{s}(k)$ possam ser estimados. Os sinais estimados $\mathbf{y}(k)$ são tais que:

$$\mathbf{y}(k) = \mathbf{W}\mathbf{x}(k) \quad (6.2)$$

A partir do sistema das equações (6.1) e (6.2), os algoritmos baseados em Análise de Componentes Independentes - como o JADE - procuram realizar a separação cega considerando que:

- $\mathbf{v}(k)$ e $\mathbf{s}(k)$ são estacionários;
- Os vetores a_1, \dots, a_n que compõem a matriz de mistura \mathbf{A} são linearmente independentes, embora arbitrários;
- As fontes $s_1(k), \dots, s_n(k)$, são estatisticamente independentes;
- O ruído aditivo é normalmente distribuído e independente de $\mathbf{s}(k)$;
- O ruído aditivo é espacialmente branco, com variância conhecida;
- O número de fontes é maior ou igual ao número de sensores, sendo que, neste estudo - como citado anteriormente - será considerado que o número de fontes é igual ao número de sensores.

Com base nas considerações supra-citadas, o algoritmo JADE realiza a separação através da diagonalização conjunta das matrizes de cumulantes de 4ª ordem dos dados observados. Portanto, para que se compreenda como o JADE realiza a separação de fontes é

necessário que sejam feitas algumas considerações sobre a teoria estatística envolvida, principalmente no que diz respeito à cumulantes e matriz de cumulantes.

6.2.1 – Cumulantes: Definição e Propriedades

Nesta subseção serão apresentadas a definição de cumulantes e suas principais propriedades. Estes conceitos são importantes uma vez que o JADE se baseia nestas estatísticas para realizar a estimativa dos componentes independentes.

Considere uma variável aleatória (v.a.) r , cuja função característica $\psi(t)$ é definida por:

$$\psi(t) = \mathbf{E}[\exp(rt)] \quad (6.3)$$

Onde $\mathbf{E}[\cdot]$ é a esperança matemática.

Os coeficientes da expansão em séries de Taylor do logaritmo de $\psi(t)$ são denominados cumulantes (Papoulis, 1991). Para a v.a. r os quatro primeiros cumulantes são dados por:

$$\begin{aligned} Cum_1(r) &= \mathbf{E}[r] \\ Cum_2(r) &= \mathbf{E}[r^2] - \mathbf{E}[r]^2 \\ Cum_3(r) &= \mathbf{E}[r^3] - 3\mathbf{E}[r]\mathbf{E}[r^2] + 2\mathbf{E}[r]^3 \\ Cum_4(r) &= \mathbf{E}[r^4] - 4\mathbf{E}[r]\mathbf{E}[r^3] - 3\mathbf{E}[r^2]^2 + 12\mathbf{E}[r]^2\mathbf{E}[r^2] - 6\mathbf{E}[r]^4 \end{aligned} \quad (6.4)$$

Para ICA, os cumulantes de 4ª ordem, também denominados *kurtosis*, são os mais importantes, visto que o cumulante de 4ª ordem de uma variável gaussiana é zero. Considerando r , uma v.a. com média zero os cumulantes de 4ª ordem são definidos por (Mansur, 1998):

$$Cum_4(r) = kurt(r) = \mathbf{E}[r^4] - 3(\mathbf{E}[r^2])^2 \quad (6.5)$$

Os cumulantes são muito utilizados em análises estatísticas, pois apresentam importantes propriedades, tais como:

- Os cumulantes são lineares. Se r_1 e r_2 são duas v.a. independentes verificam-se as relações (6.6) e (6.7); onde α é um escalar.

$$Cum_4(r_1 + r_2) = Cum_4(r_1) + Cum_4(r_2) \quad (6.6)$$

$$Cum_4(\alpha r_1) = \alpha^4 Cum_4(r_1) \quad (6.7)$$

- Cumulante de 4ª ordem de uma variável gaussiana é zero;
- Os cumulantes cruzados (*cross-cumulants*) de variáveis aleatórias independentes são zero. Os cumulantes cruzados são aqueles que envolvem pelo menos duas variáveis diferentes em seu cálculo. Os cumulantes cruzados $cum(r_{i1}, r_{i2}, \dots, r_{ik})$ para qualquer conjunto de índices i_j são definidos pelos coeficientes dos termos t_{i1}, \dots, t_{ik} da expansão de $\log(\mathbf{E}[\exp(i \sum t_j \mathbf{r}_j)])$ em séries de Taylor. Sendo $\mathbf{r} = (r_1, \dots, r_m)^T$ (Hyvärinen, 1999).

Outra importante característica dos cumulante é que através deles pode-se definir várias funções custo para ICA (INFOMAX, informação mútua, mínima entropia e outras), como demonstrado em (Alvarez,1999), sendo que a função custo para o algoritmo JADE, ϕ^{JADE} , corresponde a soma do quadrado dos cumulantes cruzados de \mathbf{y} , sendo definida por (Cardoso, 1999):

$$\phi^{JADE}(\mathbf{y}) = \sum_{ijkl \neq iikk} (Q_{ijkl}^y)^2 \quad (6.8)$$

Onde $Q_{ijkl}^y = cum(y_i, y_j, y_k, y_l)$ é a matriz de cumulantes de 4ª ordem de \mathbf{y} com $i \neq j$.

6.2.2 – Matriz de Cumulantes

A natureza algébrica dos cumulantes é tensorial, mas como os estudos relacionados à ICA abrangem principalmente estatísticas de segunda e quarta ordens, a notação matricial é suficiente para representar os cumulantes, como será apresentado nesta seção.

Considerando um vetor aleatório $\mathbf{x}_{n \times 1}$ e uma matriz $\mathbf{M}_{n \times n}$, a matriz de cumulantes de 4ª ordem associada a \mathbf{x} e \mathbf{M} , $Q_x(\mathbf{M})$, é uma matriz $n \times n$ tal que:

$$[Q_x(\mathbf{M})]_{ij} = \sum_{k,l=1}^n Cum(x_i, x_j, x_k, x_l) \mathbf{M}_{kl} \quad (6.9)$$

A partir da equação (6.9), pode-se dizer que a matriz de cumulantes é uma combinação linear de partes bidimensionais de Q_x , cujos pesos são determinados pela matriz \mathbf{M} (Cardoso, 1996b).

Se a variável \mathbf{x} for centralizada, pela definição de cumulantes de 4ª ordem, tem-se que:

$$Q_x(\mathbf{M}) = E\left[\left(\left[\mathbf{x}^T \mathbf{M} \mathbf{x}\right]\right) \mathbf{x} \mathbf{x}^T\right] - \mathbf{R}_x \cdot tr(\mathbf{M} \mathbf{R}_x) - \mathbf{R}_x \mathbf{M} \mathbf{R}_x - \mathbf{R}_x \mathbf{M}^T \mathbf{R}_x \quad (6.10)$$

Onde $tr(\cdot)$ denomina-se traço e \mathbf{R}_x é a matriz de covariância de \mathbf{x} , ou seja,

$$(\mathbf{R}_x)_{ij} = Cum(x_i, x_j) \quad (6.11)$$

Através da equação (6.10) pode-se dizer que uma dada matriz de cumulantes pode ser calculada com um custo computacional similar àquele para o cálculo da matriz de covariância. Uma outra observação importante é de que não há necessidade de calcular todo o conjunto de cumulantes de 4ª ordem para que se obtenha o valor de $Q_x(\mathbf{M})$ para uma dada matriz \mathbf{M} ; pode-se obter um valor aproximado da matriz de cumulantes, que contenha apenas uma parte da informação de 4ª ordem de $\mathbf{x}(k)$. Desta forma, realiza-se uma simplificação computacional, uma vez que, para extração de toda informação de 4ª ordem de $\mathbf{x}(k)$ seria necessário o cálculo de n^4 cumulantes de 4ª ordem (Cardoso, 1999).

Para o caso da ICA e considerando o modelo $\mathbf{x}(k) = \mathbf{A} \mathbf{s}(k)$, a estrutura da matriz de cumulantes pode ser determinada tomando-se como base as o seguinte equacionamento:

$$\begin{aligned} [Q_x(\mathbf{M})]_{ij} &= \sum_{k,l=1}^n Cum(x_i, x_j, x_k, x_l) \mathbf{M}_{kl} \\ [Q_x(\mathbf{M})]_{ij} &= \sum_{k,l=1}^n Cum(a_i \mathbf{s}, a_j \mathbf{s}, a_k \mathbf{s}, a_l \mathbf{s}) m_{lk} \\ [Q_x(\mathbf{M})]_{ij} &= \sum_{k,l=1}^n m_{lk} \sum_{abcd=1}^m Cum(a_{ai} \mathbf{s}_a, a_{jb} \mathbf{s}_b, a_{kc} \mathbf{s}_c, a_{ld} \mathbf{s}_d) \\ [Q_x(\mathbf{M})]_{ij} &= \sum_{k,l=1}^n m_{lk} \sum_{abcd=1}^m a_{ai} a_{jb} a_{kc} a_{ld} Cum(\mathbf{s}_a, \mathbf{s}_b, \mathbf{s}_c, \mathbf{s}_d) \\ [Q_x(\mathbf{M})]_{ij} &= \sum_{k,l=1}^n m_{lk} \sum_{a=1}^m a_{ia} a_{ja} a_{ka} a_{da} Cum(\mathbf{s}_a, \mathbf{s}_a, \mathbf{s}_a, \mathbf{s}_a) \\ [Q_x(\mathbf{M})]_{ij} &= \sum_{k,l=1}^n m_{lk} \sum_{a=1}^m a_{ia} a_{ja} a_{ka} a_{da} kurt(\mathbf{s}_a) \\ [Q_x(\mathbf{M})]_{ij} &= \sum_{a=1}^m kurt(\mathbf{s}_a) a_{ia} a_{ja} \mathbf{A}_a^T \mathbf{M} \mathbf{A}_a \end{aligned} \quad (6.12)$$

Sendo: $\Lambda_{\mathbf{M}} = \text{diag}_{i=1}^m(\text{kurt}(s_i)\mathbf{A}_i^T\mathbf{M}\mathbf{A}_i)$, a equação (6.12), pode ser escrita pela equação (6.13).

$$[\mathcal{Q}_{\mathbf{x}}(\mathbf{M})]_{ij} = \mathbf{A}\Lambda_{\mathbf{M}}\mathbf{A}^T \quad (6.13)$$

Onde a_i é a i -ésima coluna de \mathbf{A} . Neste tipo de fatoração o valor do *kurtosis* aparece apenas na matriz diagonal $\Lambda_{\mathbf{M}}$.

6.3 – Separação Cega Utilizando Estruturas Algébricas

Nesta seção será discutido como o algoritmo JADE realiza a separação cega baseado em estruturas algébricas. Antes, porém, será descrito o método utilizado pelo JADE para decorrelacionar os sinais medidos pelos sensores.

6.3.1 – Decorrelação Espacial

O JADE, como a maioria dos algoritmos para ICA, realiza a decorrelação espacial ou branqueamento dos componentes observados pelos sensores antes de implementar a separação propriamente dita. É importante observar que os componentes independentes não são, necessariamente, decorrelacionados.

O branqueamento é implementado por meio de uma matriz de branqueamento \mathbf{Q} , de tal forma que $\mathbf{z}(k) = \mathbf{Q}\mathbf{x}(k)$ seja espacialmente branco. Desta forma, o branqueamento faz com que $\mathbf{Q}\mathbf{A}$ seja uma matriz unitária (Cardoso, 1996b). Em outras palavras, pode-se dizer que para qualquer matriz de branqueamento \mathbf{Q} existe uma matriz unitária \mathbf{U} , tal que:

$$\mathbf{Q}\mathbf{A} = \mathbf{U} \quad (6.14)$$

A partir de (6.14) pode-se escrever:

$$\mathbf{z}(k) = \mathbf{Q}\mathbf{x}(k) = \mathbf{U}\mathbf{s}(k) + \mathbf{Q}\mathbf{v}(k) \quad (6.15)$$

Tomando como base a equação (6.15), pode-se afirmar que o branqueamento, além de decorrelacionar os dados medidos pelos sensores, também reduz a solução do problema da

separação cega a inversão de uma matriz unitária $n \times n$ mantendo o modelo linear inicialmente proposto. Ainda com base em (6.15), define-se, para a qualquer matriz \mathbf{M} , a matriz de cumulantes de $\mathbf{z}(k)$ como:

$$Q_z(\mathbf{M}) = \mathbf{U}\Lambda_{\mathbf{M}}\mathbf{U}^T \quad (6.16)$$

Considerando o modelo $\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k)$, antes da identificação cega pode ser realizado o branqueamento do vetor $\mathbf{x}(k)$, através da matriz de branqueamento \mathbf{Q} , gerando o vetor branco $\mathbf{z}(k) = \mathbf{Q}\mathbf{x}(k) = \mathbf{Q}\mathbf{A}\mathbf{s}(k)$. Sendo $\mathbf{z}(k)$ branco, a matriz $\mathbf{U} = \mathbf{Q}\mathbf{A}$, deve ser unitária, ou seja, $\mathbf{U}\mathbf{U}^T = \mathbf{I}$, onde \mathbf{I} é a matriz identidade. O modelo para identificação passa, então, a ser o da Figura 6.1.

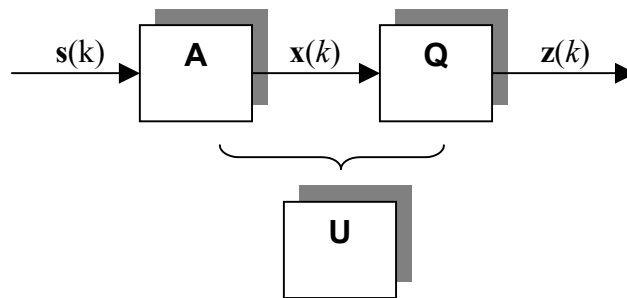


Figura 6.1 – Modelo para ICA com branqueamento

O modelo branqueado $\mathbf{z}(k) = \mathbf{A}\mathbf{s}(k)$ continua sendo um modelo de componentes independentes e a equação (6.13) passa a ser escrita como:

$$Q_z(\mathbf{M}) = \mathbf{U}\Lambda_{\mathbf{M}}\mathbf{U}^T \quad (6.17)$$

Onde $\Lambda_{\mathbf{M}} = \text{dig}(kurt(s_1)u_1^T M u_1, \dots, kurt(s_n)u_n^T M u_n)$ e u_i é a i -ésima coluna de \mathbf{U} .

Portanto, na prática, para se estimar os componentes independentes deve-se seguir os seguintes passos:

- Estimar a matriz de branqueamento \mathbf{Q} ;
- Estimar alguns cumulantes de $\mathbf{z}(k)$ e, por fim,
- Estimar a matriz $\mathbf{A} = \mathbf{Q}^{-1} \mathbf{U}$ ou a matriz de separação $\mathbf{W} = \mathbf{U}^{-1} \mathbf{Q}$.

A seguir, serão enunciadas técnicas para identificação da matriz \mathbf{A} através da determinação de uma matriz unitária \mathbf{U} . Conforme descrito em (Cardoso, 1999) estas técnicas podem ser divididas em mal condicionadas ou não equivariantes e bem condicionadas ou equivariantes. Esta descrição é importante uma vez que o algoritmo JADE procura unir as vantagens destas técnicas ao mesmo tempo em que busca minimizar suas desvantagens.

6.3.2 – Técnicas Mal Condicionadas

As técnicas mal condicionadas ou não-equivariantes podem ser não ortogonais ou ortogonais. Na primeira técnica para determinação de \mathbf{U} , baseada em uma abordagem não ortogonal, \mathbf{M}_1 e \mathbf{M}_2 são duas matrizes aleatórias $n \times n$, tais que suas matrizes de cumulantes são definidas por:

$$Q_1 = Q_x(\mathbf{M}_1) \quad (6.18)$$

$$Q_2 = Q_x(\mathbf{M}_2) \quad (6.19)$$

Considerando o modelo $\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k)$ e a equação (6.13) pode-se deduzir as equações (6.20) e (6.21).

$$Q_1 = \mathbf{A}\mathbf{\Lambda}_1\mathbf{A}^T \quad (6.20)$$

$$Q_2 = \mathbf{A}\mathbf{\Lambda}_2\mathbf{A}^T \quad (6.21)$$

Onde $\mathbf{\Lambda}_1$ e $\mathbf{\Lambda}_2$ são matrizes diagonais.

Desta forma tem-se que:

$$Q_{12} = Q_1 Q_2^{-1} = (\mathbf{A}\mathbf{\Lambda}_1\mathbf{A}^T) \cdot (\mathbf{A}\mathbf{\Lambda}_2\mathbf{A}^T)^{-1} = \mathbf{A}\mathbf{\Lambda}\mathbf{A}^T \quad (6.22)$$

Onde $\mathbf{\Lambda} = \mathbf{\Lambda}_1\mathbf{\Lambda}_2^{-1}$.

A partir de (6.22) pode-se verificar que $Q_{12}\mathbf{A} = \mathbf{A}\mathbf{\Lambda}$, ou seja, as colunas de \mathbf{A} nada mais são do que os autovalores de Q_{12} .

O problema desta técnica está no fato de que se deve considerar que as matrizes utilizadas no processo são inversíveis, o que, algumas vezes, pode levar a instabilidade do algoritmo.

A outra técnica, baseada em abordagem ortogonal, parte do princípio de que a equação (6.16) realiza uma autodecomposição, ou seja, as colunas de \mathbf{U} serão os autovetores de $Q_z(\mathbf{M})$. Desta forma a matriz unitária \mathbf{U} e, conseqüentemente, a matriz de mistura \mathbf{A} podem ser determinados considerando que \mathbf{U} é um diagonalizador de $Q_z(\mathbf{M})$. Como os dados utilizados são branqueados os problemas associados a inversão de matrizes, que ocorrem na abordagem não ortogonal, desaparecem. Por outro lado, surge um problema associado à própria estrutura algébrica desta técnica: os autovetores são únicos, se e somente se, todos os autovalores da matriz são distintos. Portanto, é preciso garantir que os autovalores de $Q_z(\mathbf{M})$ sejam todos distintos para que se possa realizar a identificação cega (Cardoso, 1999).

Os autovalores de $Q_z(\mathbf{M})$ dependem do sistema sob estudo, o qual é por definição desconhecido. Assim, não se pode determinar *a priori* se para dada matriz \mathbf{M} os autovalores de $Q_z(\mathbf{M})$ serão distintos. Na prática, um pequeno erro na estimativa de $Q_z(\mathbf{M})$ pode levar a um grande desvio no valor dos autovetores, se os autovalores de $Q_z(\mathbf{M})$ não forem distintos.

Em suma, pode-se dizer que, embora a diagonalização de uma única matriz de cumulantes seja, do ponto de vista computacional, simples e atrativa, na prática não é satisfatória, pois pode causar uma degeneração no resultado final, comprometendo a separação. Assim sendo, a precisão da estimativa dependerá das características de um sistema desconhecido, o que, na prática, dificilmente será aceita. Um outro problema associado a esta técnica é que pouca informação de 4ª ordem é explorada, uma vez que esta é condensada em uma única matriz $n \times n$. Com o objetivo de solucionar estes problemas pode-se utilizar a diagonalização conjunta de várias matrizes de cumulantes e uma técnica bem condicionada como, por exemplo, a baseada na otimização de uma função custo.

6.3.3 – Solucionando os Problemas – Técnica Bem Condicionada

A técnica descrita nesta subseção procura solucionar os problemas que ocorrem quando são utilizadas técnicas mal condicionadas, como descrito na seção 6.3.2. Para a utilização desta técnica considera-se, inicialmente, um conjunto de l matrizes $n \times n$ $\mathbf{M} = [\mathbf{M}_1, \dots, \mathbf{M}_l]$ e as matrizes de cumulantes associadas aos dados branqueados $Q_i = Q_z(\mathbf{M}_i)$, $1 \leq i \leq l$. Conforme a equação (6.16), para todo i , $Q_i = \mathbf{U} \Lambda_i \mathbf{U}^T$, com Λ_i sendo a matriz diagonal.

Para garantir que Q_i será completamente diagonalizado foi utilizado em (Cardoso, 1999; 1996c) o operador $\mathbf{off}(\cdot)$ como medida de diagonalização. A partir deste operador e da equação (6.16) pode-se dizer que $\mathbf{off}(\mathbf{U}^T Q_i \mathbf{U}) = \mathbf{off}(\Lambda_i) = 0$, se e somente se

$Q_i = \mathbf{U}\mathbf{\Lambda}_i\mathbf{U}^T$ e $\mathbf{U}\mathbf{U}^T = \mathbf{I}$, sendo \mathbf{I} a matriz identidade. Desta forma, quando $\mathbf{off}(\mathbf{U}^T Q_i \mathbf{U}) = 0$, pode-se definir a matriz \mathbf{U} unitária e, conseqüentemente, a matriz de mistura \mathbf{A} . Assim, define-se, para qualquer conjunto \mathbf{M} e qualquer matriz unitária \mathbf{V} , o seguinte critério de diagonalização conjunta (Cardoso, 1996c):

$$D_{\mathbf{M}}(\mathbf{U}) = \sum_{\mathbf{M}_i \in \mathbf{M}} \mathbf{off}(\mathbf{U}^T Q_z(\mathbf{M}_i) \mathbf{U}) \quad (6.23)$$

A equação (6.23) irá medir, portanto, quanto próximo da diagonalidade uma matriz unitária \mathbf{U} pode levar as matrizes de cumulantes geradas por \mathbf{M} . Para se obter a separação cega é necessário, então, minimizar o critério (6.23). Associa-se, assim, a cada conjunto de matrizes \mathbf{M} , um algoritmo para separação cega que segue os seguintes passos:

- Encontrar a matriz de branqueamento \mathbf{Q} ;
- Branquear os dados coletados, fazendo $\mathbf{z}(k) = \mathbf{Q}\mathbf{x}(k)$;
- Estimar as matrizes de cumulantes $Q_z(\mathbf{M})$ para toda matriz $\mathbf{M} \in \mathbf{M}$;
- Minimizar o critério de diagonalização conjunta (6.23), ou seja, tornar as matrizes de cumulantes tão diagonais quanto possível, por meio de uma transformação unitária \mathbf{U} ;
- Determinar a matriz de mistura \mathbf{A} , ou sua inversa \mathbf{W} , ou ainda, estimar os componentes independentes $\mathbf{y}(k)$, dados $\mathbf{y}(k) = \mathbf{U}^T \mathbf{z}(k)$.

O algoritmo delineado acima é capaz de reduzir as desvantagens dos algoritmos descritos na seção 6.3.2.

Encontrar uma transformação unitária \mathbf{U} , que minimize o critério (6.23), é uma técnica aparentemente adequada para solucionar do problema da identificação cega, uma vez que envolve um grande número de estatísticas de 4ª ordem e, além disso, diminui a probabilidade de degeneração dos componentes independentes. Esta consideração torna-se mais rígida na medida em que se aumenta o conjunto de matrizes de cumulantes. Por definição, o conjunto de matrizes de cumulantes é obtido quando \mathbf{M} é uma base ortonormal para um espaço linear de matrizes $n \times n$. (Cardoso, 1999). Esta base contém n^2 matrizes, tal que, a matriz total de

cumulantes corresponde a $n^2 \times n^2 = n^4$ componentes, isto é, todo conjunto de cumulantes de 4ª ordem.

Para qualquer conjunto máximo de cumulantes, foi definido em (Cardoso, 1993) que:

$$DM(\mathbf{U}) = \phi^{JADE}(\mathbf{y}(k)) \quad (6.24)$$

Onde $\mathbf{y}(k) = \mathbf{U}^T \mathbf{z}(k)$ e ϕ^{JADE} é função custo definida em (6.8).

A principal motivação de se utilizar o critério (6.24) é sua ligação com a autodecomposição, o que permite sua otimização por meio de diagonalizações conjuntas.

O ponto chave do algoritmo JADE é que ele transforma a propriedade algébrica da diagonalidade das matrizes de cumulantes em uma função custo ϕ^{JADE} , garantindo desta forma a equivariância dos resultados obtidos.

Este algoritmo apresenta, porém, algumas desvantagens, sendo que a principal delas é o custo computacional para processar o conjunto de matrizes de cumulantes e para realizar a diagonalização conjunta de $l=n^2$ matrizes. Por outro lado, este custo computacional é similar ao dos algoritmos adaptativos, uma vez que as matrizes são calculadas apenas uma vez e, além disso, existem técnicas eficientes para a diagonalização conjunta, como o algoritmo jacobiano, como será discutido na seção 6.4.

A eficiência da diagonalização conjunta pode ainda ser aumentada reduzindo o conjunto de matrizes de cumulantes a um pequeno conjunto que contenha apenas as automatrizes mais significantes de $Q_x(\cdot)$, conforme definido em (Cardoso, 1993). Esta redução no número de matrizes não afeta o desempenho do algoritmo, uma vez que $\frac{n + n(n-1)}{2}$ matrizes de cumulantes são suficientes.

6.4 - Algoritmos Jacobianos

O método jacobiano é uma forma iterativa de otimização aplicada a um conjunto de matrizes ortonormais. A transformação ortonormal resultante é obtida através de uma seqüência de rotações, sendo que cada rotação é aplicada a um par de coordenadas. Se $\mathbf{y}(k)$ é um vetor $n \times 1$, o (i,j) -ésimo plano de rotação, definido por meio de um ângulo θ_{ij} , muda as coordenadas i e j de $\mathbf{y}(k)$ de acordo com (6.25), enquanto as outras coordenadas ficam inalteradas. O ângulo θ_{ij} é denominado ângulo de Givens (Golub, 1989).

$$\begin{bmatrix} y_i \\ y_j \end{bmatrix} \leftarrow \begin{bmatrix} \cos(\theta_{ij}) & \sin(\theta_{ij}) \\ -\sin(\theta_{ij}) & \cos(\theta_{ij}) \end{bmatrix} \begin{bmatrix} y_i \\ y_j \end{bmatrix} \quad (6.25)$$

Uma rotação corresponde a um passo através de todas as $n(n-1)/2$ combinações possíveis de pares de coordenadas.

O método jacobiano pode ser empregado em algoritmos baseados em dados – Maxkurt (Cardoso, 1999; Comon, 1994), em estatísticas – JADE (Cardoso, 1993) ou em algoritmos que são uma mistura de dados e estatísticas – SHIBBS (Cardoso, 1999).

Em problemas de ICA, as técnicas jacobianas foram inicialmente utilizadas por Comon (1994), em um algoritmo baseado em dados que realizava uma seqüência de rotações jacobianas em dados branqueados até que sua função custo $\phi(\mathbf{y}(k))$ fosse otimizada. De uma forma simplificada pode-se descrever este algoritmo jacobiano para separação cega da seguinte forma.:

- 1 - **Inicialização**: Calcular a matriz de branqueamento e branquear os dados de entrada.
- 2 - **Rotação**: Para todos os $n(n-1)/2$ pares, ou seja,

Para $1 \leq i \leq j \leq n$:

 - 2.1 – Calcular o ângulo de Givens θ_{ij} , otimizando a função $\phi(\mathbf{y}(k))$, quando o par de coordenadas (y_i, y_j) é rotacionado;
 - 2.2 – Se $\theta_{ij} < \theta_{min}$, rotacionar o par (y_i, y_j) de acordo com a equação (6.25).
- 3 - **Finalização**: Se nenhum par tiver sido rotacionado no passo anterior, finalizar. Caso contrário, voltar ao passo 2 e realizar mais uma rotação.

Desta forma, pode-se dizer que a técnica jacobiana considera uma seqüência bidimensional (y_i, y_j) do problema de ICA. A atualização do passo 2.2 no par (i, j) parcialmente desfaz o efeito da otimização prévia nos pares contendo ou i ou j . Por esta razão, é necessária a realização de várias rotações antes que a otimização esteja completa (Cardoso, 1999). Na maioria dos casos, porém, os algoritmos jacobianos são muito eficientes e alcançam a convergência com um pequeno número de rotações.

O ponto chave dos algoritmos jacobianos está no fato de que cada plano de rotação depende de um único parâmetro: o ângulo de Givens θ_{ij} ; este ângulo reduz o problema de otimização a um problema unidimensional, em cada passo de rotação.

O ângulo θ_{min} é um ângulo bem pequeno que controla a precisão da otimização e pode ser determinado de forma empírica. Para o caso da ICA, θ_{min} deve ser escolhido de tal forma que as rotações por ângulos pequenos não causem alterações estatísticas. Em (Cardoso, 1999), define-se θ_{min} típico como $\theta_{min} = 10^{-2} / \sqrt{N}$. Onde N é o número de amostras. O valor de θ_{min} , não é, na verdade, um valor crítico, visto que os algoritmos jacobianos são, em geral, rápidos.

No algoritmo JADE, o método jacobiano descrito acima foi estendido para diagonalizar um conjunto arbitrário de $n \times n$ matrizes de cumulantes (Cardoso, 1996c). Desta forma, ao contrário do algoritmo proposto por Comon, no algoritmo JADE as rotações são aplicadas não a dados, mas a um conjunto de matrizes de cumulantes, ou seja, o algoritmo JADE atualiza um conjunto de matrizes estatísticas, não apenas um simples conjunto de dados.

6.5 - JADE: um Algoritmo Baseado em Estatística

Esta seção descreve os passos para implementação do algoritmo de separação cega baseado puramente em estatísticas - o algoritmo JADE inicialmente proposto por Cardoso e Souloumic (Cardoso, 1993).

Com base nas seções anteriores, o algoritmo JADE pode ser implementado seguindo passos seguintes.

- 1 - Inicialização:
 - a. Calcular a matriz de covariância dos dados observados;
 - b. Calcular a matriz de branqueamento;
 - c. Branquear os dados observados;
- 2 - Formação estatística:
 - a. Calcular o conjunto Q_z de matrizes de cumulantes de 4ª ordem dos dados branqueados;
 - b. Calcular os pares n mais significantes;

3 - Otimização da função custo ortogonal:

Encontrar a matriz de rotação \mathbf{U} , tal que as matrizes de cumulantes sejam tão diagonais quanto possível, ou seja, resolver:

$$\mathbf{U} = \arg \min \sum_i \mathbf{off}(\mathbf{U}^T \mathbf{Q}_z \mathbf{U}) \quad (6.26)$$

Este procedimento é descrito como diagonalização conjunta;

4 - Separação dos componentes independentes:

Estimar a matriz de mistura, tal que:

$$\mathbf{A} = \mathbf{U}\mathbf{Q}^{-1} \quad (6.27)$$

e/ou estimar os componentes independentes, tais que:

$$\mathbf{y}(k) = \mathbf{W}\mathbf{x}(k) = \mathbf{V}^T \mathbf{z}(k) \quad (6.28)$$

Onde $\mathbf{W} = \mathbf{U}^T \mathbf{Q}$.

O passo 1 está relacionado a com estatística de segunda ordem, sendo implementado através da decomposição em autovalores da matriz de covariância \mathbf{R}_x . Devido a consideração de ruído branco uma estimativa da variância do ruído é a média dos $m-n$ menores autovalores de \mathbf{R}_x .

No passo 2, o calculo das automatrizes está relacionado a diagonalização de uma matriz $n^2 \times n^2$ dos elementos de \mathbf{Q}_z .

O passo 3 é implementado estendendo-se a técnica jacobiana para várias matrizes. É importante observar ainda quando $n=2$, a técnica jacobiana não é iterativa, ou seja, uma única rotação de Givens é suficiente para a diagonalização conjunta. Mais uma vez fica claro que o algoritmo JADE é um algoritmo jacobiano, visto que o passo 3 é implementado através de uma técnica jacobiana. Além disso, as rotações não são aplicadas aos dados branqueados, mas as suas matrizes de cumulantes. Assim, o algoritmo atualiza não os dados, mas suas matrizes estatísticas.

O ponto chave do algoritmo JADE está na seleção das matrizes de cumulantes envolvidas na estimativa das fontes originais. As matrizes de cumulantes podem ser exata e conjuntamente diagonalizadas quando o modelo permite, o que não ocorre quando são processados dados do mundo real. Como apenas amostras das estatísticas estão disponíveis -

segundo o modelo $\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k)$ - não é possível, em geral, assegurar que o algoritmo encontre uma solução adequada (Cardoso, 1999). Esta é, sem dúvida, mais uma razão que comprova a importância da seleção correta das matrizes de cumulantes, pois no caso de dados do mundo real, a impossibilidade de realizar a diagonalização conjunta exata, corresponde a impossibilidade de encontrar os componentes independentes $\mathbf{y}(k)$. Assim, fazer um conjunto máximo de matrizes de cumulantes tão diagonais quanto possível, coincide com fazer os componentes de $\mathbf{y}(k)$ tão independentes quanto possível (Cardoso, 1999).

Existem várias formas de estimar o conjunto máximo de matrizes de cumulantes. Por exemplo, para bases canônicas, os elementos das matrizes de cumulantes são iguais aos cumulantes de $\mathbf{z}(k)$. Uma outra opção é utilizar uma base simétrica; neste caso, é suficiente estimar e diagonalizar $n + n(n+1)/2$ matrizes de cumulantes simétricas. Existe ainda a opção de reduzir o tamanho das estatísticas necessárias para representar a informação de 4ª ordem, o que somente pode ser aplicado se o modelo de separação permitir. Neste caso, as matrizes de cumulantes devem ter a estrutura mostrada em (6.16). Desta forma, o mapeamento $\mathbf{M} \rightarrow Q_z(\mathbf{M})$ tem *rank* igual a n ou mais precisamente igual ao número de componentes com *kurtosis* diferente de zero. Isto ocorre porque existem n graus de liberdade para matrizes na forma $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, nomeadamente, os n elementos da diagonal de $\mathbf{\Lambda}$ (Cardoso, 1999). A partir desta afirmação e da propriedade de simetria dos cumulantes, pode-se dizer que existem n automatrizes $\mathbf{E}_1, \dots, \mathbf{E}_n$, que são ortonormais e que satisfazem a relação:

$$Q_z(\mathbf{E}_i) = \mu_i \mathbf{E}_i \quad (6.29)$$

Onde o escalar μ_i é o autovetor correspondente.

Com emprego das automatrizes $\mathbf{E}_1, \dots, \mathbf{E}_n$, verifica-se que toda informação contida em $Q_z(\cdot)$ pode ser representada por n automatrizes. Assim, inserindo $\mathbf{M} = u_i u_i^T$ em (6.16) e considerando que \mathbf{U} é unitária, verifica-se que o conjunto de automatrizes é dado por $\mathbf{E}_i = u_i u_i^T$ (Cardoso, 1999).

O algoritmo JADE foi proposto inicialmente para realizar a ICA através da diagonalização conjunta aproximada de automatrizes, onde a diagonalização conjunta foi somente empregada às n mais significantes automatrizes de $Q_z(\mathbf{M})$ como uma forma de reduzir a carga computacional. Desta forma, o número de estatísticas é reduzido de n^4 cumulantes ou $n + n(n+1)/2$ matrizes de cumulantes simétricas, a um conjunto de n automatrizes. Esta redução é obtida sem perdas somente quando o modelo permite, ou seja,

não é recomendada a redução a autmatrizes quando são processados conjunto de dados para os quais não está claro se o modelo $\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k)$ assegura boa precisão.

6.6 – Considerações Finais

As funções custo para ICA baseadas no conceito de entropia podem ser entendidas a partir dos princípios de máxima verossimilhança; e o emprego da divergência de Kullback-Leibler produz aproximações baseadas em cumulantes destas funções custos (Cardoso, 1999).

Com base em uma abordagem ortogonal para ICA, onde a descorrelação é imposta, as funções custo baseadas em cumulantes podem ser otimizadas por meio de técnicas jacobianas aplicadas aos dados ou às estatísticas dos dados, nomeadamente, matrizes de cumulantes.

A estrutura dos cumulantes no modelo para ICA pode ser facilmente explorada por técnicas de identificação algébrica; porém, versões simples destas técnicas são não-equivariantes. Uma possibilidade para solucionar este problema é utilizar a estrutura algébrica conjunta de várias matrizes de cumulantes.

O algoritmo JADE interliga as abordagens baseadas em função custo e as técnicas algébricas, uma vez que, ele objetiva ambos: otimização de uma função custo e a utilização da estrutura dos cumulantes. De uma forma mais geral, pode-se dizer que, através do emprego de técnicas jacobinas, a natureza dos cumulantes pode ser utilizada para facilitar a otimização da função custo baseada em cumulantes.

Outro comentário relevante deve ser feito com relação ao ajuste ou sintonia dos algoritmos. Os algoritmos baseados em técnicas jacobianas, como o JADE, não necessitam de ajustes.

Devem-se ressaltar ainda alguns pontos negativos do algoritmo JADE entre eles os principais são: sua complexidade computacional e o elevado requisito de memória quando se trabalha com um grande número de componentes.

Capítulo 7 – Algoritmo FlexICA

7.1 – Introdução

Este capítulo tem por objetivo descrever o algoritmo para análise de componentes independentes que utiliza uma não linearidade variável ou flexível, controlada por um expoente gaussiano que varia de acordo com o *kurtosis* das fontes. Este algoritmo é denominado *Flexible ICA* ou simplesmente FlexICA e foi desenvolvido por Choi et al. (Choi, 1998). O algoritmo utiliza um modelo generalizado de densidade gaussiana para estimar a distribuição de probabilidade das fontes e tem como base as técnicas de gradiente natural.

7.2 – Considerações Iniciais

Inicialmente, considera-se o sistema de mistura dado por:

$$\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k) + \mathbf{v}(k) \quad (7.1)$$

Onde $\mathbf{x}(k)=[x_1(k), \dots, x_m(k)]^T$ é o vetor que corresponde às medidas feitas pelos sensores corrompidas por ruído aditivo $\mathbf{v}(k)=[v_1(k), \dots, v_m(k)]^T$, $\mathbf{s}(k)=[s_1(k), \dots, s_n(k)]^T$ é o vetor das fontes originais e $\mathbf{A}_{m \times n}$ é a matriz de mistura. Considera-se neste problema que apenas $\mathbf{x}(k)$ é conhecido.

Neste capítulo, assim como nos demais, o número de fontes é considerado conhecido e igual ao número de sensores, ou seja, $n=m$, de tal forma que \mathbf{A} e \mathbf{W} são matrizes quadradas $n \times n$ e, além disso, as fontes $\mathbf{s}(k)$ são consideradas mutuamente independentes e com média zero. Considera-se ainda que o ruído aditivo pode ser desprezado ou reduzido a níveis desprezíveis.

O desafio deste problema está em desenvolver uma rede neural e um algoritmo de treinamento adaptativo que sejam capazes de estimar as fontes originais e/ou identificar a matriz de mistura \mathbf{A} , ou de separação \mathbf{W} e que possam, até certo ponto, acompanhar as variações do sistema em observação.

7.3 – Algoritmos para ICA e Gradiente Natural

Amari et al. (Cichoki, 2002; Choi, 2000; Choi, 1998; Amari, 1996) propuseram o conceito de Gradiente Natural e mostram que este é um método eficiente para treinamento de redes neurais aplicadas na solução do problema de ICA.

É importante salientar que o método de Gradiente Relativo, proposto de forma independente por Cardoso e Laheld (Cardoso, 1996a), é idêntico ao Gradiente Natural (NG), para a solução do problema de ICA.

Nesta seção será feita uma breve revisão sobre os algoritmos para ICA baseados em NG.

Inicialmente, deve-se considerar uma rede linear cujas saídas são dadas por:

$$\mathbf{y}(k) = \mathbf{W}\mathbf{x}(k) \quad (7.2)$$

Onde $\mathbf{y}(k) = [y_1(k), \dots, y_n(k)]^T$ é vetor que contém as fontes estimadas, ou seja, $\mathbf{y}(k) = \hat{\mathbf{s}}(k)$ e os elementos da matriz de separação \mathbf{W} (w_{ij}) representam os pesos sinápticos entre $y_i(k)$ e $x_j(k)$.

Para o problema de ICA quadrada ($n=m$), os conceitos de máxima verossimilhança ou informação mútua levam a seguinte função custo:

$$L(\mathbf{W}) = -\log|\det(\mathbf{W})| - \sum_{i=1}^n \log p_i(y_i) \quad (7.3)$$

Onde $p_i(\cdot)$ representa a função densidade de probabilidade. E define-se ainda a seguinte função de ativação não linear:

$$g_i(y_i) = -\frac{d \log p_i(y_i)}{dy_i} \quad (7.4)$$

Como o objetivo é minimizar a função custo de (7.3), pode-se, utilizando a função (7.4), escrever:

$$\nabla L(\mathbf{W}) = \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}} = -\mathbf{W}^{-T} + \mathbf{g}(\mathbf{y}(k))\mathbf{x}(k)^T \quad (7.5)$$

Onde $g(\mathbf{y}(k))$ é a função cujo i -ésimo elemento é $g_i(y_i(k))$ e \mathbf{W}^{-T} é transposta da inversa de \mathbf{W} .

Assim, com algumas manipulações matriciais pode-se chegar ao algoritmo:

$$\Delta \mathbf{W}(k) = \mu(\mathbf{W}^{-T} - g(\mathbf{y}(k))\mathbf{x}(k)^T) \quad (7.6)$$

Onde μ é o passo de adaptação.

Amari et al. (Amari, 1996) propuseram uma modificação no algoritmo (7.6) através da utilização do Gradiente Natural de Riemannian ($\tilde{\nabla}L(\mathbf{W})$), que levou ao seguinte algoritmo:

$$\begin{aligned} \mathbf{W}(k+1) &= \mathbf{W}(k) - \mu \tilde{\nabla}L(\mathbf{W}) \\ \mathbf{W}(k+1) &= \mathbf{W}(k) - \mu \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}} \mathbf{W}^T(k) \mathbf{W}(k) \\ \mathbf{W}(k+1) &= \mathbf{W}(k) + \mu(\mathbf{I} - g(\mathbf{y}(k))\mathbf{y}(k)^T) \mathbf{W}(k) \\ \Delta \mathbf{W}(k) &= \mu(\mathbf{I} - g(\mathbf{y}(k))\mathbf{y}(k)^T) \mathbf{W}(k) \end{aligned} \quad (7.7)$$

O algoritmo (7.7) é essencialmente igual ao obtido utilizando o gradiente relativo proposto por Cardoso e Laheld (Cichoki, 2002).

7.3.1 – NG com Restrição de Ortogonalidade

Sendo que o vetor $\mathbf{x}(k)$ foi branqueado e os sinais fontes foram normalizados, ou seja:

$$\mathbf{E}[\mathbf{x}(k)\mathbf{x}^T(k)] = \mathbf{I}_m \quad (7.8)$$

$$\mathbf{E}[\mathbf{s}(k)\mathbf{s}^T(k)] = \mathbf{I}_n \quad (7.9)$$

Quando $n = m$, tem-se, a partir de (7.8) e (7.9), que $\mathbf{A}\mathbf{A}^T = \mathbf{I}$. Isto implica que as linhas do vetor \mathbf{A} são vetores unitários n -dimensionais, de outra forma que $\mathbf{Q}\mathbf{A} \in \mathbf{R}^{n \times n}$ é ortogonal, onde \mathbf{Q} é a matriz de branqueamento. Neste caso, procura-se, então, uma matriz de separação \mathbf{W} que seja ortogonal, tal que, $\mathbf{W}^{-1} = \mathbf{W}^T$.

O algoritmo para determinação da matriz de separação \mathbf{W} utilizando gradiente natural, no caso em que os dados observados são branqueados e considerando que \mathbf{W} é aproximadamente ortogonal, tal que $\mathbf{x}(k)=\mathbf{W}^T\mathbf{y}(k)$, é dado por:

$$\mathbf{W}(k+1) = \mathbf{W}(k) - \mu[\mathbf{g}(\mathbf{y}(k))\mathbf{y}^T(k) - \mathbf{y}(k)\mathbf{g}^T(\mathbf{y}(k))]\mathbf{W}(k) \quad (7.10)$$

A matriz \mathbf{W} é dita aproximadamente ortogonal. Uma matriz é precisamente ortogonal, para a versão contínua do algoritmo, se, e somente se, $\mathbf{W}(0)$ é ortogonal. Para o caso discreto, a matriz é aproximadamente ortogonal sob a condição de que o passo de adaptação μ é um valor suficientemente pequeno (algo da ordem de 10^{-2}) (Cichocki, 2002).

Na prática, devido a anti-simetria do termo $\mathbf{g}(\mathbf{y}(k))\mathbf{y}^T(k) - \mathbf{y}(k)\mathbf{g}^T(\mathbf{y}(k))$, o branqueamento pode ser realizado juntamente com o processo de separação. A partir desta idéia, Cardoso e Laheld desenvolveram o algoritmo conhecido como EASI (*Equivariant Adaptive Separation via Independence*) dado por (Cardoso, 1996a):

$$\Delta\mathbf{W}(k) = \mu[\mathbf{I} - \mathbf{y}(k)\mathbf{y}^T(k) - \mathbf{g}(\mathbf{y}(k))\mathbf{y}^T(k) + \mathbf{y}(k)\mathbf{g}^T(\mathbf{y}(k))]\mathbf{W}(k) \quad (7.11)$$

Pode se considerar ainda a versão em blocos (*batch*) do algoritmo (7.11), dada por:

$$\Delta\mathbf{W}(k) = \mu[\mathbf{I} - \langle \mathbf{y}(k)\mathbf{y}^T(k) \rangle - \langle \mathbf{g}(\mathbf{y}(k))\mathbf{y}^T(k) \rangle + \langle \mathbf{y}(k)\mathbf{g}^T(\mathbf{y}(k)) \rangle]\mathbf{W}(k) \quad (7.12)$$

Onde $\langle . \rangle$ é o valor médio.

7.4 – Modelo Generalizado para Distribuição Gaussiana

Como a escolha e utilização da função não linear ótima, dada pela equação (7.4) depende do conhecimento *a priori* da distribuição de probabilidade das fontes originais, o que, em muitos casos, não é possível, tem-se um problema. Para tentar solucionar este problema são utilizadas funções hipotéticas ou aproximadas que tentam representar a distribuição original. Por exemplo, para fontes super-gaussianas, como sinais de áudio, a distribuição laplaciana leva a função não linear dada por, $\mathbf{g}(\mathbf{y}(k))=\text{sign}(\mathbf{y}(k))$, onde $\text{sign}(\cdot)$ é a função sigmóide. Para fontes sub-gaussianas, como sinais modulados, a função cúbica, dada por $\mathbf{g}(\mathbf{y}(k)) = \mathbf{y}^3(k)$, tem sido a mais utilizada.

Por outro lado, quando ocorre a mistura de fontes com distribuições diferentes, ou seja, super e sub-gaussianas, ou quando não se conhece a distribuição das fontes, a função não linear pode ser escolhida, ou adaptada, de acordo com uma estimativa do *kurtosis* da fonte separada. A partir desta idéia, vários trabalhos foram apresentados (Cichocki, 2002; Hyvärinen, 2001).

Neste capítulo, será descrito o trabalho apresentado por Choi et al. (Choi, 1998) que apresenta uma função não linear, auto-adaptável e controlada por um expoente gaussiano, que é obtida a partir do modelo generalizado de distribuição gaussiana.

O modelo de distribuição de probabilidade gaussiano nada mais é do que um conjunto de distribuições parametrizadas por um número real positivo α , o qual é geralmente referenciado como o expoente gaussiano da distribuição que irá controlar o achatamento da curva de distribuição.

Assim sendo, a função densidade de probabilidade para uma função gaussiana generalizada é representada por (Choi, 1998; Choi, 2000):

$$q_i(y_i) = \frac{\alpha_i}{2\sigma_i \Gamma\left(\frac{1}{\alpha_i}\right)} \exp\left(-\left|\frac{y_i}{\sigma_i}\right|^{\alpha_i}\right) \quad (7.13)$$

Onde $\Gamma(\cdot)$ é a função Gama, $\alpha_i > 0$ é um parâmetro variável e σ_i é a medida de variância, ou dispersão da distribuição.

O valor de α pode variar de zero até infinito, sendo que $\alpha=1$ corresponde à distribuição laplaciana; $\alpha=2$, à distribuição gaussiana padrão e $\alpha \rightarrow \infty$ corresponde à distribuição uniforme.

7.4.1 – Kurtosis e Expoente Gaussiano

Como visto no Capítulo 3, o *kurtosis* é uma grandeza adimensional que mede o achatamento da curva de distribuição de probabilidade. Se a distribuição apresenta um *kurtosis* positivo ela é denominada super-gaussiana ou *leptkurtic* e se apresenta *kurtosis* negativo é denominada sub-gaussiana ou *platykurtic*, sendo que o *kurtosis* de uma distribuição pode ser definido em termos dos momentos de 2ª e 4ª ordem como:

$$kurt(y) = \frac{m_4}{m_2^2} - 3 \tag{7.14}$$

A constante -3 na equação (7.14) faz com que o *kurtosis* seja zero para distribuição normal. No aplicativo MATLAB, por outro lado, a função *kurtosis*(x) não leva em consideração a constante -3; neste caso, o valor de *kurtosis* que o software retorna para a distribuição normal é 3.

Para a distribuição gaussiana generalizada mostrada na equação (7.13), o *kurtosis* pode ser expresso em termos do expoente gaussiano, ou seja:

$$kurt(y) = \frac{\Gamma\left(\frac{5}{\alpha}\right)\Gamma\left(\frac{1}{\alpha}\right)}{\Gamma^2\left(\frac{3}{\alpha}\right)} - 3 \tag{7.15}$$

A Figura 7.1 mostra o gráfico de *kurtosis*(.) versus expoente gaussiano α para sinais super e sub-gaussianos (Choi, 2000).

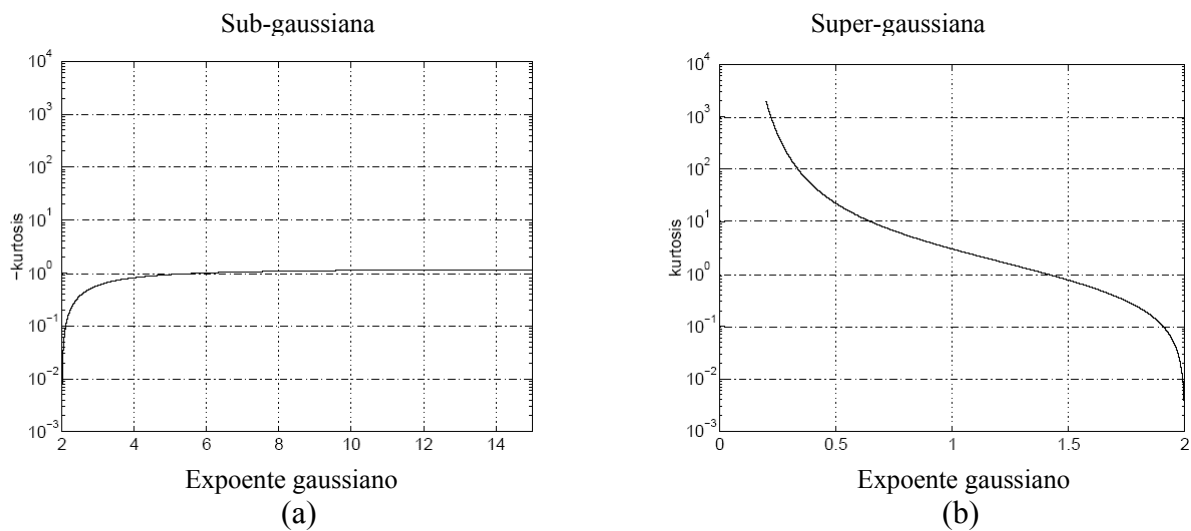


Figura 7.1 – *Kurtosis* versus expoente gaussiano para sinais (a) sub-gaussianos e (b) super-gaussianos.

Pela Figura 7.1(a) pode-se observar que o valor do *kurtosis* praticamente não varia com o expoente gaussiano α , por isso, a função cúbica $g_i(y_i(k)) = |y_i(k)|^3 \text{sign}(y_i(k))$, é a mais utilizada para separação de fontes sub-gaussianas. Embora não seja mostrado no gráfico,

podem ser utilizados valores de $\alpha \gg 2$, mas estes valores, em geral, não acrescentam melhoria ao processo de separação.

Na Figura 7.1(b), por outro lado, pode-se observar uma variação relativamente grande do *kurtosis* em função do expoente α .

Na prática, devem-se estimar os *kurtosis* das fontes extraídas e, então, selecionar o valor de α mais adequado como será descrito nas próximas seções.

7.5 – O Algoritmo FlexICA

Com base no modelo de distribuição gaussiana generalizado, a função não linear de ativação $g_i(y_i)$ para os algoritmos descritos em (7.11) é escrita como:

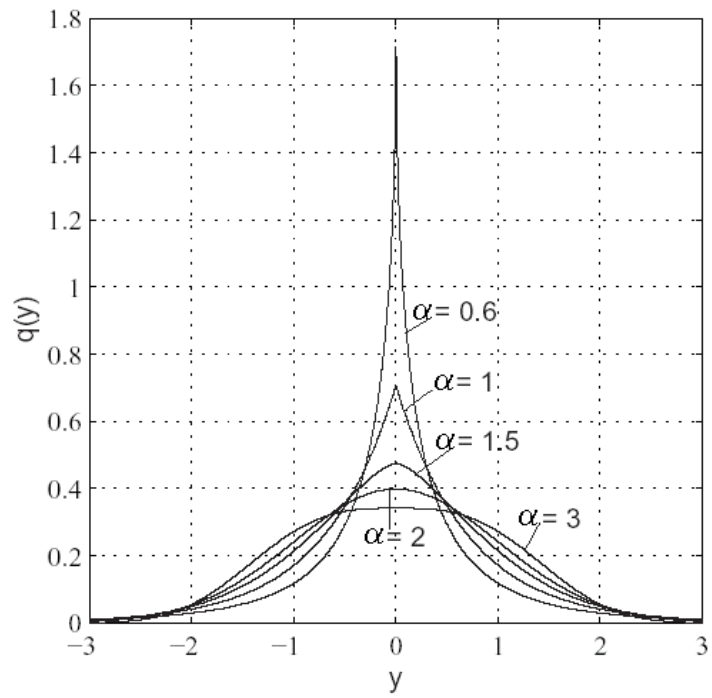
$$g_i(y_i) = |y_i|^{\alpha_i-1} \text{sign}(y_i) \quad (7.16)$$

Na Figura 7.2(a) podem ser vista as diferentes distribuições de probabilidade correspondentes aos valores de α , considerando $\sigma=1$ e na Figura 7.1(b) são mostradas as não linearidades resultantes, obtidas a partir da equação (7.16).

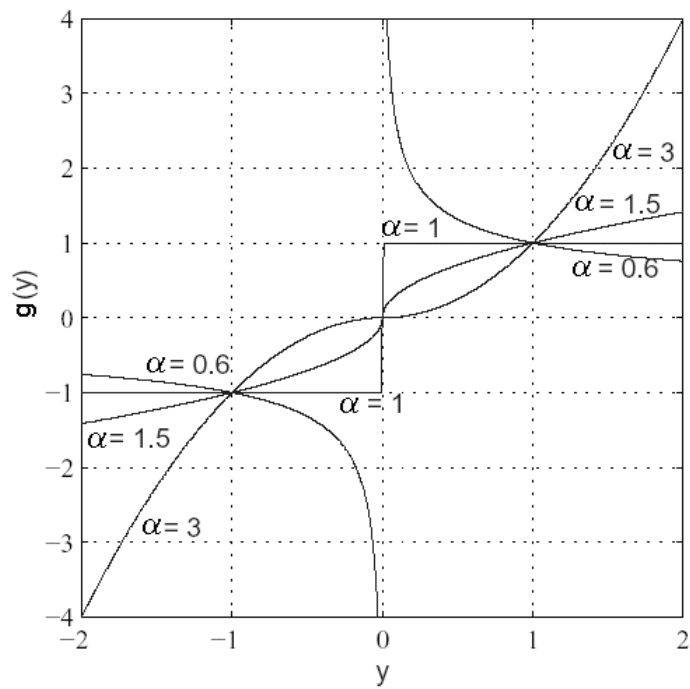
Para o caso em que se trabalha com sinais como impulsos ou *spikes*, os parâmetros α_i podem assumir valores entre $0 < \alpha_i < 1$ e a função de ativação não linear pode ser escrita da seguinte forma:

$$g_i(y_i) = \frac{y_i}{\left[|y_i|^{2-\alpha} + \varepsilon \right]} \quad (7.17)$$

Onde ε assume um valor positivo da ordem de (10^{-4}) , que evita que a singularidade da função quando $y_i=0$.



(a)



(b)

Figura 7.2 – (a) Funções densidade de probabilidade resultantes para diferentes valores de α , (b) Funções de ativação não lineares correspondentes.

A equação (7.17), para $\alpha=2$, torna-se uma função sigmóide, adequada para sinais super-gaussianos, e para $\alpha=4$, torna-se uma função cúbica, adequada para sinais sub-gaussianos.

Choi et al. (Choi, 1998; Choi, 2000) mostram através de seus experimentos que pequenas alterações no valor de α não alteram o desempenho do algoritmo.

O ponto de partida do algoritmo FlexICA é o algoritmo mostrado em (7.11), sendo que a função não linear a ser utilizada é determinada pela equação (7.16). O problema está em selecionar um valor adequado para o expoente gaussiano α . Para solucionar este problema deve-se estimar o *kurtosis* do sinal de saída $y_i(k)$ e a seguir selecionar um valor correspondente de α , como mostrado na Figura 7.1. A partir do valor de α pode-se encontrar a função não linear de ativação mais adequada à separação. Em geral, a estimativa do *kurtosis* não representa exatamente o *kurtosis* da fonte original; mas, fornece uma idéia do *kurtosis* real e, através desta idéia é possível saber se a fonte original é sub ou super-gaussiana. Além disso, foi mostrado em (Cardoso, 1997; Amari, 1997) que o desempenho da separação não é degradado mesmo se o *kurtosis* (e, conseqüentemente a densidade de probabilidade estimada) não estiver exatamente de acordo com os originais.

Na seção seguinte serão descritos os principais passos para a implementação do algoritmo FlexICA (Cichocki, 2002; Choi, 2003).

7.5.1 – Passos para Implementação do FlexICA

Primeiramente, devem ser definidos alguns parâmetros iniciais:

- o número máximo de iterações e
- o passo de adaptação.

Para simplificar o processamento pelo MATLAB, as amostras podem ser divididas e processadas em blocos; neste caso, é preciso definir também o número de blocos que serão utilizados. Para o correto desempenho do algoritmo, o número de blocos deve ser escolhido de tal forma que o número de amostras em cada bloco esteja em torno de 3000.

O próximo passo é verificar as dimensões do vetor $\mathbf{x}(k)$.

É realizado, a seguir, um pré-processamento que consiste em centralizar e branquear os dados de $\mathbf{x}(k)$ da mesma forma como descrito para o AMUSE.

O próximo passo consiste em inicializar as variáveis: \mathbf{W} – matriz de separação -, α , o erro inicial entre as verossimilhanças, a tolerância ou erro mínimo e o vetor de saída $\mathbf{y}(k)$ inicial. A partir destes valores iniciais pode-se estimar o valor inicial da verossimilhança, como mostrado pela equação (7.3). Considerando que $\sigma=1$ e que $p_i(\cdot)$ é dada pela equação (7.13), pode-se escrever (7.3) como (7.18).

$$veross = \log(\det(\mathbf{W})) + \sum_{i=1}^n \left(\log \left(\frac{\alpha}{2\Gamma\left(\frac{1}{\alpha}\right)} \exp(-|\mathbf{y}|^\alpha) \right) \right)$$

$$veross = \log(\det(\mathbf{W})) + \sum_{i=1}^n \left(\log(\alpha) - \log\left(2\Gamma\left(\frac{1}{\alpha}\right)\right) - |\mathbf{y}|^\alpha \right) \quad (7.18)$$

Após estes cálculos preliminares, pode-se iniciar o algoritmo propriamente dito, seguindo os passos abaixo:

1. Calcular a função não linear generalizada conforme a equação (7.16);
2. Atualizar a matriz de separação \mathbf{W} conforme a equação (7.12);
3. Atualizar a função não linear $g(y_i)$.
 - 3.1 Calcular os momento de 2ª e 4ª ordens de $\mathbf{y}(k)$.
 - 3.2 A partir dos momentos de 2ª e 4ª ordens de $\mathbf{y}(k)$ estimar o *kurtosis* das fontes.
4. Atualizar o expoente gaussiano α de acordo com o *kurtosis* calculado no passo 3.2 e com base na Figura 7.1.
5. Calcular o novo valor da verossimilhança.
6. Calcular o erro absoluto entre as verossimilhanças calculadas, como:

$$erro = \frac{|veross(iter) - veross(iter - 1)| + |veross(iter - 1) - veross(iter - 2)|}{2} \quad (7.19)$$

Onde $iter$ corresponde ao número da iteração.

Os passos de 1 a 6 devem ser repetidos enquanto o $erro$ for menor que a tolerância escolhida e o número de iterações forem maiores que o número máximo de iterações.

7.5.2 – Análise de Estabilidade

Nesta seção será apresentada uma revisão dos resultados mais importantes obtidos em (Cardoso, 1996a; Cichocki, 2002; Choi, 2000). Com este objetivo, será apresentada uma análise da estabilidade para diferentes funções não lineares derivadas a partir do modelo de densidade gaussiana generalizado, conforme descrito em (Choi, 2000). Assim, será discutida a estabilidade do algoritmo (7.11) para os casos em que:

- $\alpha = 4$ - Distribuição sub-gaussiana;
- $\alpha = 1$ – Distribuição super-gaussiana;
- $\alpha < 1$ – Distribuição esparsa.

Após extensa dedução matemática, foi demonstrado (Choi, 2000; Cichocki, 2002) que uma condição suficiente para a estabilidade é que:

$$\chi_i = \mathbf{E}[\dot{g}_i(y_i)] - \mathbf{E}[g_i(y_i)y_i] > 0 \quad (7.20)$$

Onde $\dot{g}_i(\cdot)$ é a primeira derivada de $g_i(\cdot)$.

- **$\alpha = 4$: Distribuição sub-gaussiana**

Para $\alpha=4$, tem-se uma fonte sub-gaussiana, ou seja, $kurtosis < 0$, e a função não linear correspondente é a função cúbica dada por $g_i(y_i) = |y_i|^2 y_i$.

Nestas condições, χ_i é igual ao *kurtosis* de y_i multiplicado por -1, desta forma, para y_i sub-gaussiano, a condição (7.20) é satisfeita.

- **$\alpha = 1$: Distribuição super-gaussiana**

Com $\alpha = 1$, a função não linear a ser utilizada será a função sigmóide, ou seja, $g_i(y_i) = \text{sign}(y_i) = \frac{y_i}{|y_i|}$, cuja derivada $\dot{g}_i(y_i)$ é dada por:

$$\dot{g}_i(y_i) = 2\delta(y_i) \quad (7.21)$$

Onde δ é o delta de Dirac.

Pode-se, então, calcular:

$$\mathbf{E}[\dot{g}_i(y_i)] = \int_{-\infty}^{\infty} 2\delta(y_i) \frac{1}{2\sigma_i} \exp\left(-\frac{|y_i|}{\sigma_i}\right) dy_i = \frac{1}{\sigma_i} \quad (7.22)$$

E

$$\mathbf{E}[g_i(y_i)y_i] = \mathbf{E}[|y_i|] = \sigma_i \quad (7.23)$$

Considerando ainda que $\mathbf{E}[y_i^2] = 2\sigma_i^2 = 1$, o que implica que, $\sigma_i = \sqrt{1/2}$, então χ_i será dado por:

$$\chi_i = \frac{1 - \sigma_i^2}{\sigma_i} \quad (7.24)$$

Como $\sigma_i = \sqrt{1/2}$, $\chi_i > 0$, satisfazendo a condição de estabilidade.

- **$\alpha < 1$: Distribuição esparsa**

Para fontes esparsas, que apresentam $kurtosis \gg 1$, o valor de α deve ser menor que 1, o que leva a uma função não linear singular em torno da origem. Desta forma, em aplicações práticas, para $y_i \in [-\varepsilon, \varepsilon]$ - onde ε é um valor positivo da ordem de (10^{-4}) -, a função não linear correspondente tem valores constantes.

Através de manipulações algébricas, pode-se verificar que, para $y_i \in [-\varepsilon, \varepsilon]$, $\mathbf{E}[\dot{g}_i(y_i)]$ e $\mathbf{E}[g_i(y_i)]$ são dadas por (7.25) e (7.26) respectivamente.

$$\mathbf{E}[\dot{g}_i(y_i)] = \frac{(\alpha_i - 2)\sigma_i^{\alpha_i - 2}}{\Gamma\left(\frac{1}{\alpha_i}\right)} \Gamma\left(\frac{\alpha_i - 1}{\alpha_i}\right) \quad (7.25)$$

$$\mathbf{E}[g_i(y_i)] = \frac{\alpha_i \sigma_i^{\alpha_i + 1}}{\Gamma\left(\frac{1}{\alpha_i}\right)} \frac{1}{\sigma_i} \Gamma\left(\frac{\alpha_i + 1}{\alpha_i}\right) \quad (7.26)$$

Como pode ser observado em (7.25) e (7.26), deve-se tomar cuidado quando se utiliza $\alpha < 1$, pois, por exemplo, $\alpha = 0.5$ implica em $\Gamma(-1) = 0$, o que não satisfaz a condição de estabilidade. Por outro lado, para $\alpha = 0.8$, a condição de estabilidade é verificada.

7.6 – Considerações Finais

Neste capítulo foi apresentado, dentro da estrutura do gradiente natural, o algoritmo flexível para ICA (*Flexible ICA* ou simplesmente FlexICA). Este algoritmo utiliza uma função não linear auto-adaptável para representar a distribuição de probabilidade dos sinais originais.

A função não linear utilizada é derivada através de um modelo de densidade gaussiana generalizado. Com a utilização de uma não linearidade adaptável ou flexível é possível a separação de misturas de sinais sub e super-gaussianos simultaneamente.

Em comparação com outros métodos, a função não linear utilizada no algoritmo é controlada por um único parâmetro: o expoente gaussiano (α). Desta forma pode-se dizer que a utilização de uma função gaussiana generalizada, controlada pelo expoente α , leva a um

método simples para determinação da função não linear a ser utilizada para a separação dos sinais. Para a correta seleção do expoente α deve-se inicialmente estimar o *kurtosis* do sinal de saída $\mathbf{y}(k)$.

Foi apresentada ainda uma análise de estabilidade do algoritmo para algumas das funções não lineares mais utilizadas na prática.

CAPÍTULO 8 – Simulações Computacionais

8.1 – Introdução

Neste capítulo, a validade e o desempenho dos algoritmos apresentados nos Capítulos 5, 6 e 7 serão analisados para diferentes problemas de separação de sinais.

A comparação entre os algoritmos não é uma tarefa simples, uma vez que existe um grande número de variáveis envolvidas no processo e que devem ser levadas em consideração, entre as quais estão:

- Tipo de mistura: instantânea ou com atraso;
- Condicionamento da matriz de mistura: bem ou mal condicionada;
- Distribuição das fontes: conhecida ou desconhecida;
- Presença de ruído;
- Relação entre número de fontes e número de sensores,...

Neste capítulo, com base nas alternativas supra citadas, procura-se abordar alguns dos principais problemas encontrados na literatura. Para isso, serão feitas simulações com sinais de teste, sinais de telecomunicações e sinais biomédicos que serão misturados através de uma mistura linear e instantânea. Neste trabalho serão apresentados apenas casos em que o número de fontes é igual o número de sensores e sem ruído adicionado ao sistema de mistura. Tanto os sinais de teste como os de telecomunicações foram artificialmente gerados com o auxílio do MATLAB. Por outro lado, os sinais biomédicos são reais (Moor, 1997).

8.1.1 – Índices de Desempenho

Quando as fontes originais são conhecidas, como ocorre quando se trabalha com fontes geradas artificialmente, o desempenho do sistema de separação pode ser medido por vários índices. A seguir serão apresentados alguns destes índices.

Um dos índices de desempenho mais conhecidos e utilizados foi apresentado em (Amari, 1996) e é definido por:

$$E_1 = \sum_{i=1}^n \left(\sum_{k=1}^n \frac{|g_{ik}|}{\max_j |g_{ij}|} - 1 \right) + \sum_{k=1}^n \left(\sum_{i=1}^n \frac{|g_{ik}|}{\max_j |g_{jk}|} - 1 \right) \quad (8.1)$$

Onde g_{ij} é o (i,j) -ésimo elemento da matriz de transformação global $\mathbf{G} = \mathbf{WA}$.

Em (Cichocki, 2002) foi proposto o índice de desempenho dado por:

$$E_{1_norm} = \frac{1}{n(n-1)} \sum_{i=1}^n \left[\left(\sum_{k=1}^n \frac{|g_{ik}|}{\max_j |g_{ij}|} - 1 \right) + \left(\sum_{k=1}^n \frac{|g_{ki}|}{\max_j |g_{ji}|} - 1 \right) \right] \quad (8.2)$$

Onde g_{ij} é o (i,j) -ésimo elemento da matriz de transformação global $\mathbf{G} = \mathbf{WA}$ e $\max_j g_{ij}$ é o máximo valor entre os elementos da i -ésima linha do vetor \mathbf{G} , enquanto que $\max_j(g_{ji})$ representa o máximo valor entre os elementos da i -ésima coluna de \mathbf{G} .

Quando a separação perfeita é obtida, o índice de desempenho é igual a zero. Na prática, porém, um bom índice de desempenho está na ordem de 10^{-2} .

O índice de desempenho (PI – *Performance Index*) proposto em (Choi, 1998) é bem parecido com o apresentado em (8.2). Este índice é definido como:

$$PI = \sum_{i=1}^n \left[\left(\sum_{k=1}^n \frac{|g_{ik}|^2}{\max_j |g_{ij}|} - 1 \right) + \left(\sum_{k=1}^n \frac{|g_{ki}|^2}{\max_j |g_{ji}|} - 1 \right) \right] \quad (8.3)$$

Como pode ser observada, a diferença está no fato de que em (8.2) não é considerado o quadrado de g_{ik} e g_{ki} . Além disso, em (8.3) não é feita a normalização $\frac{1}{n(n-1)}$. Como no índice (8.1), a separação é considerada perfeita quando $PI=0$.

Alvarez (2004) utilizou dois índices de desempenho: PI_1 e PI_2 que se assemelham ao índice proposto em (Amari, 1996). Estes índices são especialmente úteis quando as fontes são normalizadas para possuírem variância unitária. O índice PI_1 fornece uma idéia da capacidade de separação do algoritmo. O índice representa a interferência devido a outras fontes que permanecem nos sinais separados. Quando a separação é perfeita, este índice é zero. Para caso em que o número de fontes é igual ao número de sensores, PI_1 é dado por:

$$PI_1 = \frac{1}{n(n-1)} \text{sum}(\text{sum}(\mathbf{G}_n) - n) \quad (8.4)$$

Onde: $\mathbf{G}_n = \text{pinv}(\text{diag}(\max|\mathbf{G}^T|)) \cdot |\mathbf{G}|$, é a matriz de transferência global com as linhas normalizadas de tal forma que seus elementos máximos sejam iguais a 1; n é o número de fontes ou sensores, considerados iguais; $\text{pinv}(\cdot)$ calcula a matriz pseudo-inversa e $\text{diag}(\cdot)$ monta uma matriz diagonal.

Valores de PI_1 da ordem de 10^{-2} indicam que a separação foi realizada de forma adequada.

PI_1 , por sua vez, não leva em consideração o fato do algoritmo ter separado duas ou mais vezes a mesma fonte. Para solucionar este problema, Cruces propôs o índice PI_2 . Este índice verifica a divergência de \mathbf{G} de uma matriz unitária/semiunitária, após a normalização das linhas de \mathbf{G} para remover qualquer indeterminação com relação a ordem (*scaling indeterminance*) com que as fontes são separadas. Idealmente, PI_2 também deve ser zero. Assim sendo, PI_2 é dado por:

$$PI_2 = \frac{1}{n(n-1)} \text{sum}(\text{sum}(\mathbf{G}\mathbf{G}_n) - n) \quad (8.5)$$

Onde: $\mathbf{G}\mathbf{G}_n = \mathbf{D} \cdot |\mathbf{G} \cdot \mathbf{G}^T| \cdot \mathbf{D}$ e $\mathbf{D} = \text{pinv}(\text{diag}(\text{diag} \cdot |\mathbf{G} \cdot \mathbf{G}^T|)^{1/2})$.

Outro índice que também pode ser utilizado para medir o desempenho do sistema de separação é relação sinal - interferência (SIR – *signal to interference ratio*). Em (Mutihac, 2003), considerando que o número de fontes é igual ao de sensores e igual a n , a SIR é dada por:

$$SIR = -\frac{1}{n} \sum_{i=1}^n 10 \log \left(\frac{\max(\mathbf{g}_i)^2}{\mathbf{g}_i^T \mathbf{g}_i - \max(\mathbf{g}_i)^2} \right) \quad (\text{dB}) \quad (8.6)$$

Ao contrário dos outros índices quanto maior a SIR, melhor a separação. Deve-se notar ainda que esta relação é medida em dB.

Outros índices utilizados para verificar o desempenho da separação podem ser encontrados em (Douglas, 1997; Comon, 1994; Cichocki, 2002; Schobben, 1999; Giannakopoulos, 1999).

Nesta dissertação foram utilizados os índices (8.2), (8.3), (8.4) e (8.5), por serem os mais utilizados na literatura. Além dos índices de desempenho, é importante observar ainda a forma de onda resultante da separação.

8.2 – Simulações com Sinais de Teste

Nesta seção serão apresentados diversos experimentos realizados com sinais padrão. As simulações foram realizadas com auxílio do aplicativo MATLAB 6.1.

8.2.1 – Separação para um Problema Bem Condicionado

Nesta simulação três sinais determinísticos, são misturados através de uma matriz de mistura bem condicionada. Os sinais originais são mostrados na Figura 8.1 e são descritos pelas seguintes equações:

$$s_1(k) = \sin(0.1*(k-1)/2) - \text{senóide} \quad (8.7)$$

$$s_2(k) = \text{sign}(\cos(2*\pi*0.02*(k))) - \text{onda quadrada} \quad (8.8)$$

$$s_3(k) = \text{sawtooth}(2*\pi*0.01*k) - \text{dente de serra} \quad (8.9)$$

Considera-se um número de amostras k variando de 0 a 10000.

Como citado, a matriz de mistura escolhida é uma matriz bem condicionada com $\text{cond}(\mathbf{A}) = 1.87547$ e $\text{det}(\mathbf{A})=1.12727$.

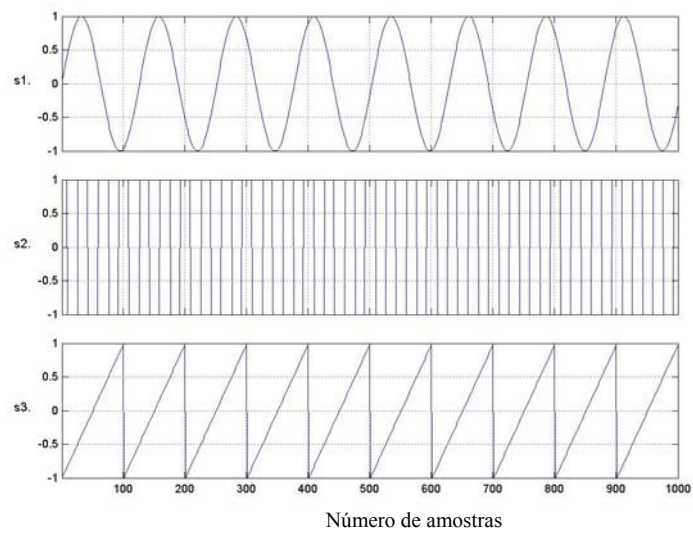


Figura 8.1 - Sinais originais para experimento 8.2.1.

A mistura resultante é apresentada na Figura 8.2.

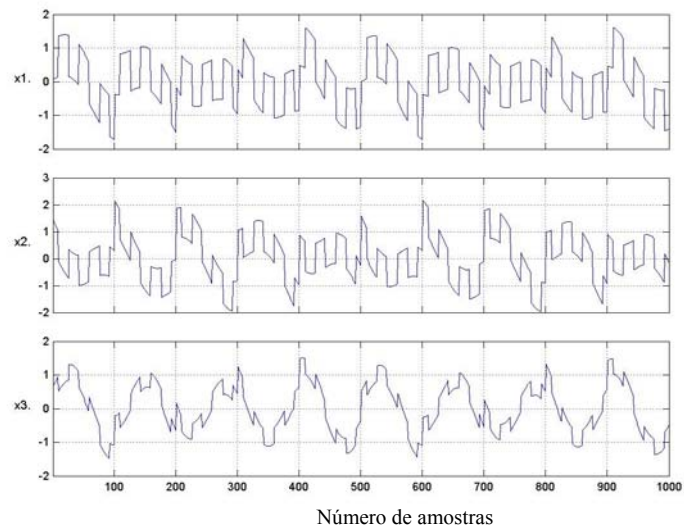


Figura 8.2 – Sinais medidos pelos sensores para experimento 8.2.1.

As Figuras 8.3, 8.4 e 8.5, mostram, respectivamente, as fontes estimadas pelos algoritmos sob estudo.

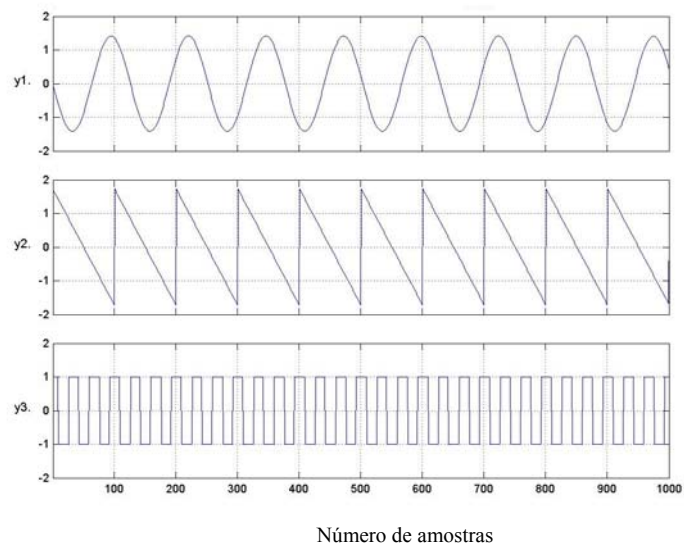


Figura 8.3 – Sinais estimados com o AMUSE para experimento 8.2.1.

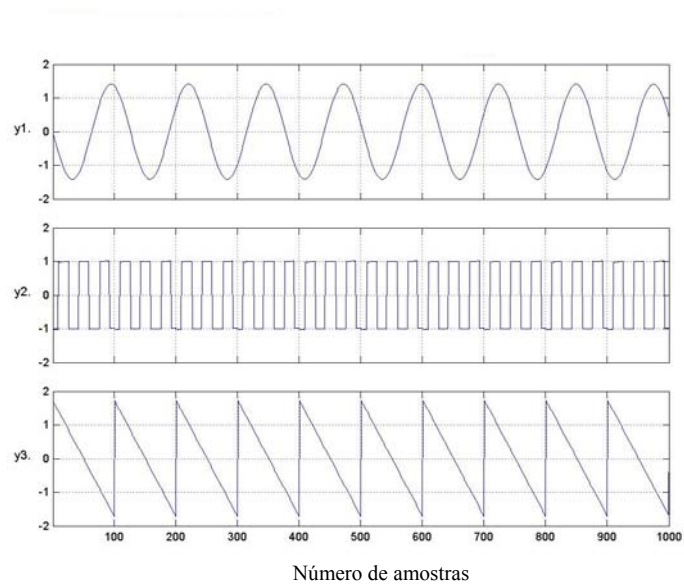


Figura 8.4 – Sinais estimados com o algoritmo JADE para experimento 8.2.1.

O desempenho dos algoritmos, medido pelos Índices de Desempenho, para este problema, são apresentados na Tabela 8.1 e na Figura 8.6.

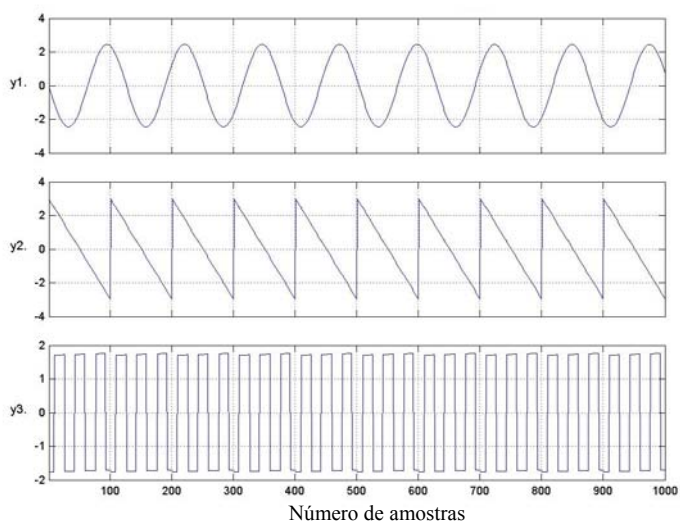


Figura 8.5 - Sinais estimados com o algoritmo FlexICA para experimento 8.2.1.

Algoritmo	PI ₁	PI ₂	PI	E _{1_norm}
AMUSE	8.1686e-4	0.0016	1.8451e-5	0.0016
JADE	0.0036	0.0044	3.1304e-4	0.0069
FlexICA	0.0045	0.0046	6.1951e-4	0.0085

Tabela. 8.1- Índice de desempenho para experimento 8.2.1.

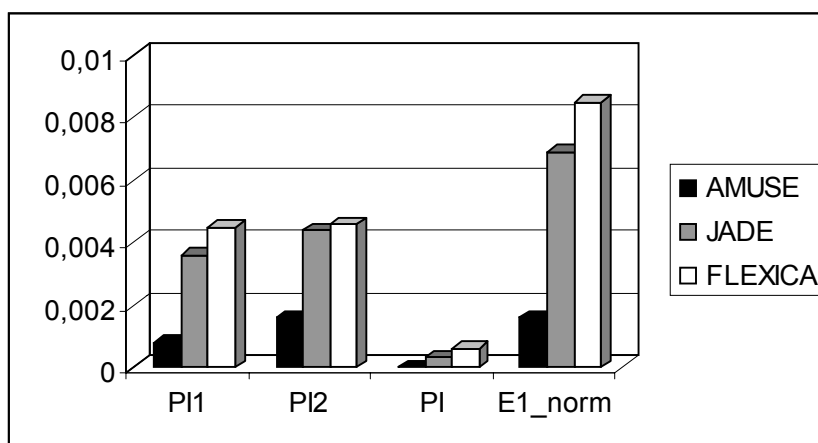


Figura 8.6 - Índice de desempenho para experimento 8.2.1.

A partir dos resultados mostrados pelas Figuras 8.3, 8.4, 8.5 e 8.6 e pela Tabela 8.1 pode-se dizer que os algoritmos apresentaram uma solução adequada.

Neste exemplo, não foi aplicado nenhum tipo de ajuste ou controle de ganho, portanto, como esperado, a amplitude dos sinais recuperados difere da amplitude dos sinais originais. Além disso, também a fase dos sinais recuperados também não é mesma dos sinais originais.

8.2.2 – Separação para um Problema Mal Condicionado

Existem dois casos em que o problema da separação pode ser considerado mal condicionado, um deles ocorre quando a matriz de mistura é mal condicionada ($\text{cond}(\mathbf{A}) \gg 1$), o outro ocorre quando uma das fontes é um ruído. Neste exemplo, será analisado o caso que a matriz de mistura utilizada é mal condicionada, sendo, $\text{cond}(\mathbf{A})=1.08946e+005$ e $\det(\mathbf{A})=-1.43068e-005$, e a matriz é definida por:

$$\mathbf{A} = \begin{pmatrix} 0.97987806683934 & 0.24075667706140 & 0.18481321279432 \\ 0.57559894061671 & 0.13901437514395 & 0.10510608418215 \\ 0.52236350700431 & 0.63563437583109 & 0.83199267804045 \end{pmatrix}$$

Nesta simulação os mesmos sinais de teste empregados no experimento anterior e apresentados na Figura 8.1.

A medida pelos sensores é apresentada na Figura 8.7.

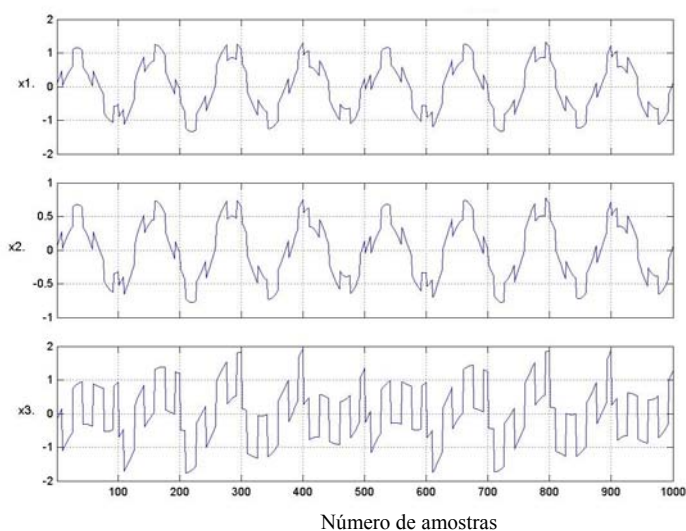


Figura 8.7 – Sinais observados pelos sensores para o experimento 8.2.2.

Mais uma vez as fontes originais não são visíveis a partir das misturas.

As Figuras 8.8, 8.9 e 8.10, mostram, respectivamente, as fontes estimadas pelos algoritmos.

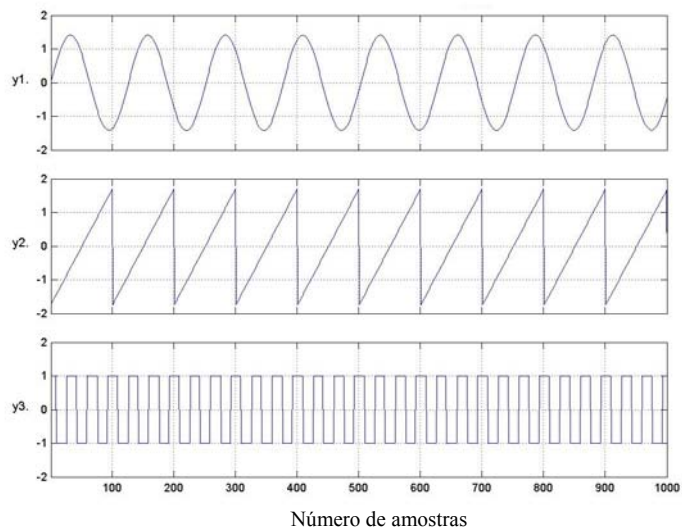


Figura 8.8 – Sinais estimados com o AMUSE para o experimento 8.2.2..

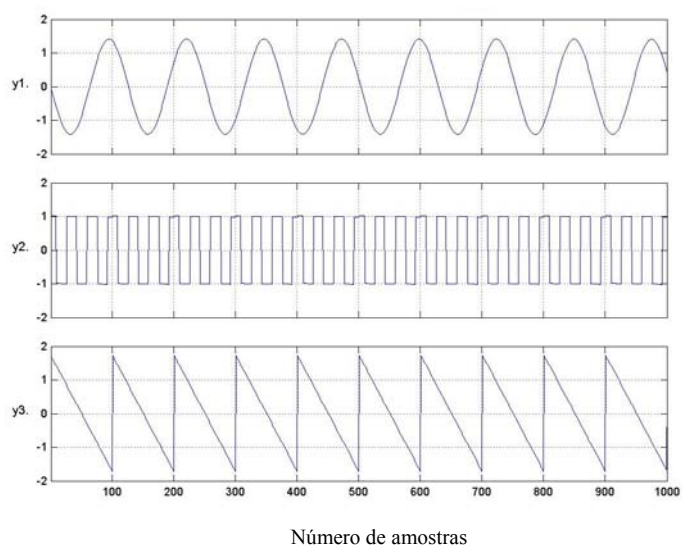


Figura 8.9 - Sinais estimados com o algoritmo JADE para o experimento 8.2.2.

As medidas de desempenho e complexidade dos algoritmos para este exemplo são mostradas na Tabela 8.2 e Figura 8.11.

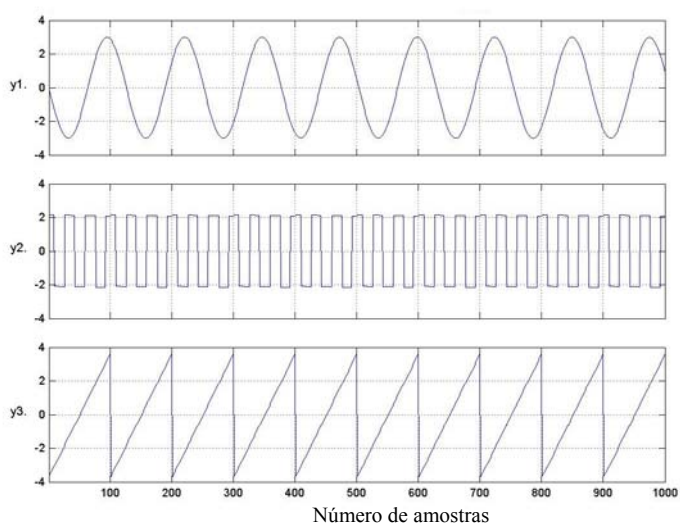


Figura 8.10 - Fontes estimadas com o algoritmo FlexICA para o experimento 8.2.2.

Algoritmo	PI_1	PI_2	PI	E_{1_norm}
AMUSE	8.3465e-4	0.0016	1.8372e-5	0.0016
JADE	0.0036	0.0044	3.1296e-4	0.0069
FlexICA	0.0049	0.0057	6.0714e-4	0.0093

Tabela 8.2 - Índice de desempenho para o experimento 8.2.2..

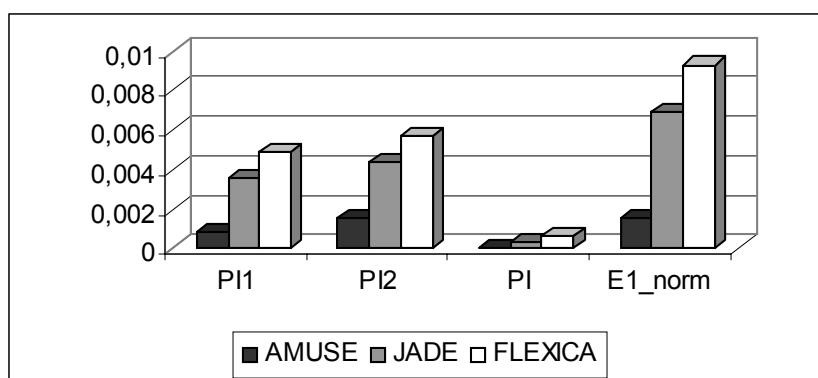


Figura 8.11 - Índice de desempenho para o experimento 8.2.2..

A partir dos resultados mostrados pelas Figuras 8.8, 8.9, 8.10 e 8.11 e pela Tabela 8.2 pode-se dizer que os algoritmos apresentaram uma solução adequada, mesmo considerando uma matriz de mistura mal condicionada. Pode-se concluir, portanto, que o condicionamento da matriz de mistura não afetou o desempenho dos algoritmos para o experimento em questão.

8.2.3 – Separação Cega na Presença de Ruído

Este problema também pode ser considerado um problema mal condicionado, uma vez que uma das fontes é um sinal de ruído. Além disso, como o ruído (gaussiano ou uniforme) é considerado uma fonte ele deve ser recuperado da mesma forma que as demais fontes. Mesmo sendo considerado como fonte o efeito do ruído tende a aumentar a complexidade do problema dificultando a separação das fontes.

Nesta simulação são empregados três sinais de teste, onde um deles é um sinal de ruído gaussiano. Estes sinais são misturados através de uma matriz de mistura bem condicionada. Os sinais originais são mostrados na Figura 8.12 e são descritos pelas seguintes equações:

$$s_1(k) = \sin(0.1*(k-1)/2) - \text{senóide} \quad (8.10)$$

$$s_2(k) = \text{sign}(\cos(2*\pi*0.02*(k))) - \text{onda quadrada} \quad (8.11)$$

$$s_3(k) = \text{rand}(1,k) - \text{ruído} \quad (8.12)$$

$$s_4(k) = \text{sawtooth}(2*\pi*0.01*k) - \text{dente de serra} \quad (8.13)$$

Neste exemplo são também consideradas 1000 amostras, ou seja, k varia de 0 a 1000.

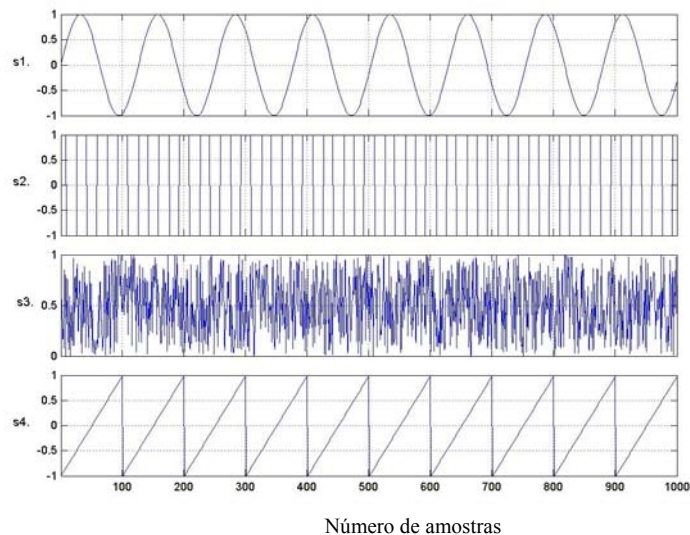


Figura 8.12 - Sinais originais para o experimento 8.2.3.

A matriz de mistura escolhida é uma matriz bem condicionada com $\text{cond}(\mathbf{A}) = 2.9672$ e $\text{det}(\mathbf{A}) = 0.8708$. A mistura resultante ou a medida pelos sensores é apresentada na Figura 8.13.

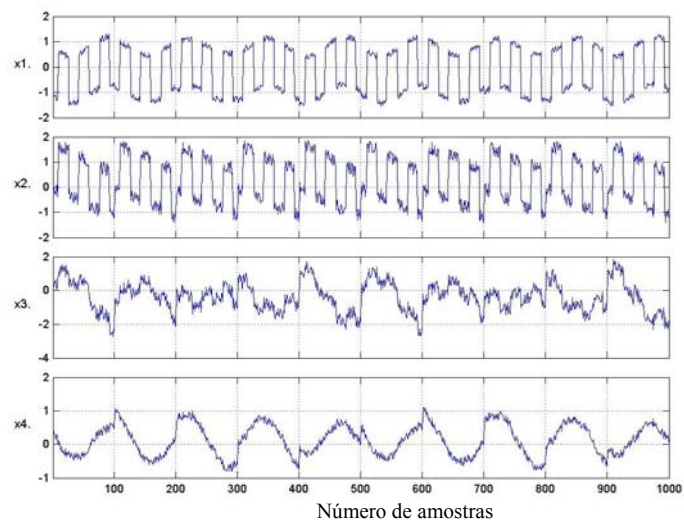


Figura 8.13 – Sinais observados pelos sensores para o experimento 8.2.3.

As Figuras 8.14, 8.15 e 8.16 mostram, respectivamente, as fontes estimadas pelos algoritmos em estudo.

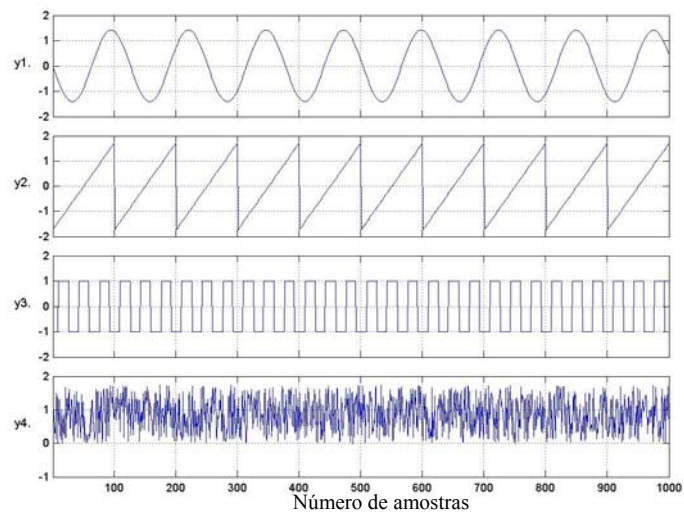


Figura 8.14 – Fontes estimadas com o AMUSE para o experimento 8.2.3.

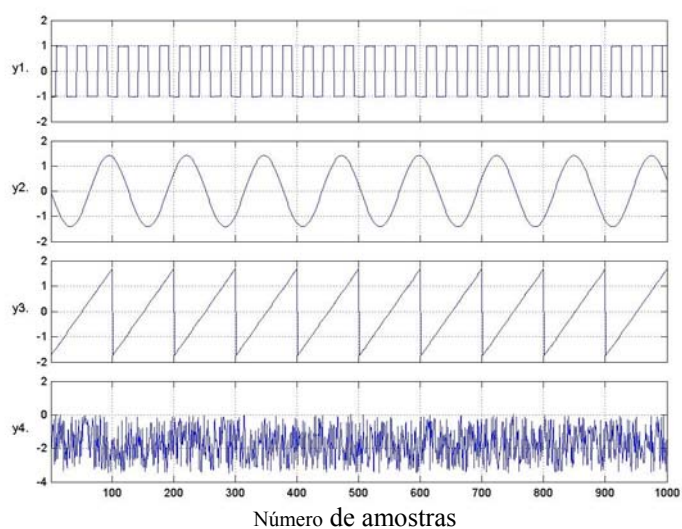


Figura 8.15 – Fontes estimadas com o algoritmo JADE para o experimento 8.2.3.

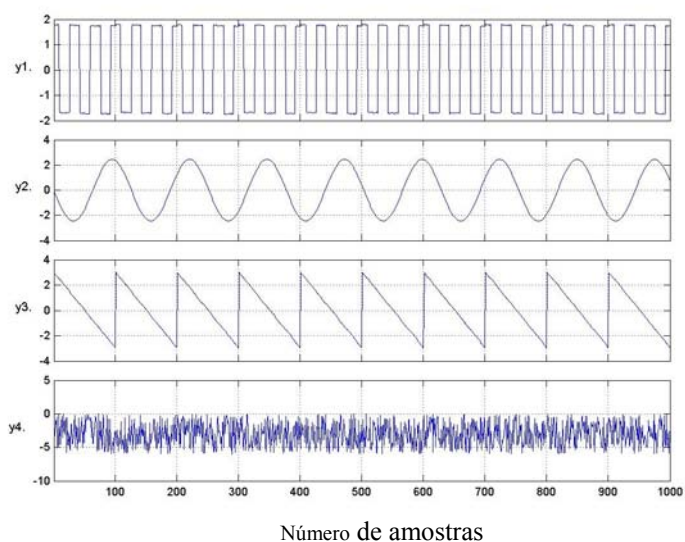


Figura 8.16 - Fontes estimadas com o algoritmo FlexICA para o experimento 8.2.3..

As medidas de desempenho e complexidade dos algoritmos para este exemplo são mostradas na Tabela 8.3 e na Figura 8.17.

Algoritmo	PI1	PI2	PI	E1_norm
AMUSE	0.0035	0.0040	7.1801e-4	0.0072
JADE	0.0060	0.0083	0.0016	0.0119
FlexICA	0.0073	0.0103	0.0024	0.0133

Tabela. 8.3 - Índice de desempenho para o experimento 3.

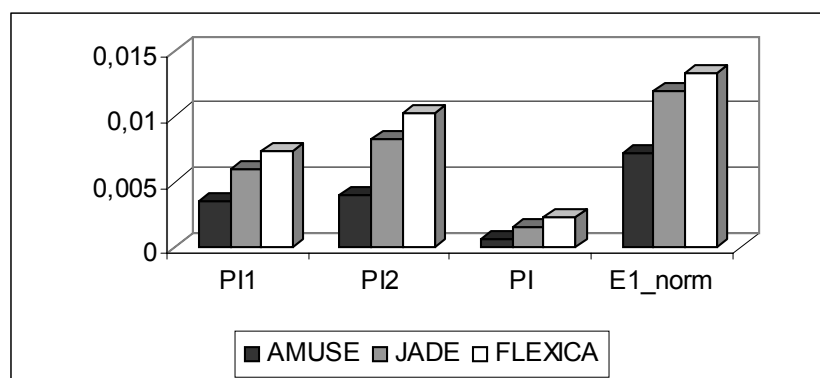


Figura 8.17 - Índice de desempenho para o experimento 3.

A partir dos resultados mostrados pelas Figuras 8.14, 8.15, 8.16 e 8.17 e pela Tabela 8.3 pode-se dizer que os algoritmos apresentaram uma solução adequada.

É importante observar que mesmo sendo s_4 uma fonte bem determinada (ou fonte forte) é praticamente impossível identificá-la visualmente a partir das medidas dos sensores como mostra a Figura 8.13.

Neste exemplo, também não foi aplicado nenhum tipo de ajuste ou controle de ganho, portanto, como esperado, a amplitude dos sinais recuperados difere da amplitude dos sinais originais. Além disso, também a fase dos sinais recuperados também não é mesma dos sinais originais.

8.2.4 – Separação de Senóides com Frequências Múltiplas de 60Hz

Neste experimento foram misturadas várias senóides com frequências que são múltiplas de 60Hz. As fontes utilizadas neste experimento são descritas pelas seguintes equações, onde k varia de 0 a 10000:

$$s_1(k) = \sin(2\pi \cdot 60 \cdot k); \quad (8.9)$$

$$s_2(k) = \sin(2\pi \cdot 120 \cdot k); \quad (8.10)$$

$$s_3(k) = \sin(2\pi \cdot 180 \cdot k); \quad (8.11)$$

$$s_4(k) = \sin(2\pi \cdot 240 \cdot k); \quad (8.12)$$

$$s_5(k) = \sin(2\pi \cdot 300 \cdot k); \quad (8.13)$$

$$s_6(k) = \sin(2\pi \cdot 420 \cdot k); \quad (8.14)$$

$$s_7(k) = \sin(2\pi \cdot 540 \cdot k); \quad (8.15)$$

$$s_8(k) = \sin(2\pi \cdot 900 \cdot k); \quad (8.16)$$

A matriz de mistura \mathbf{A} utilizada, apresenta $\det(\mathbf{A}) = 4.1802$ e $\text{cond}(\mathbf{A}) = 5.8411$.

As Figuras 8.18 e 8.19 mostram as fontes originais e as misturas resultantes medidas pelos sensores. As Figuras 8.20, 8.21 e 8.22 apresentam, respectivamente, os resultados obtidos pelos algoritmos AMUSE, JADE e FlexICA.

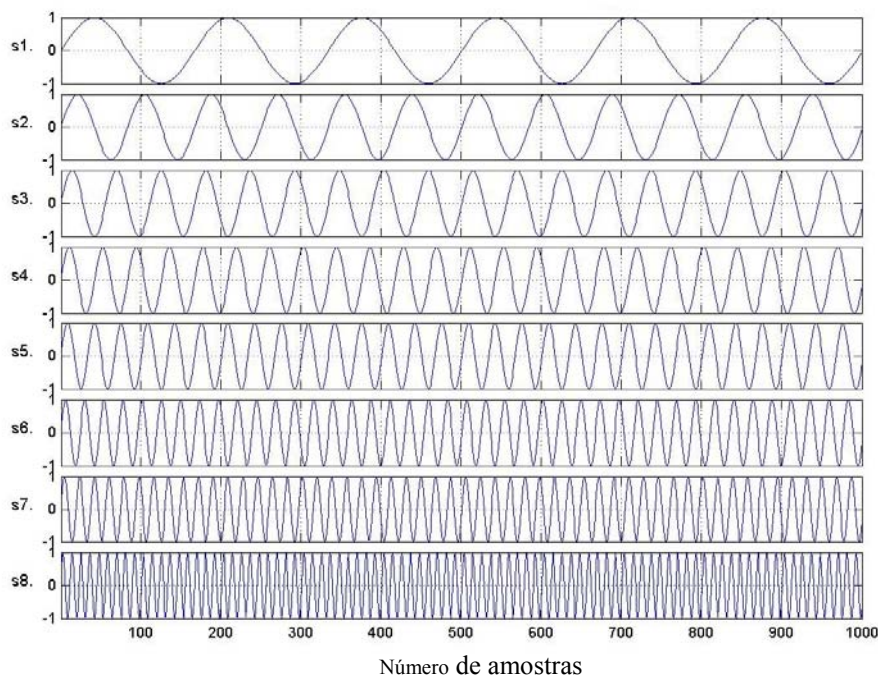
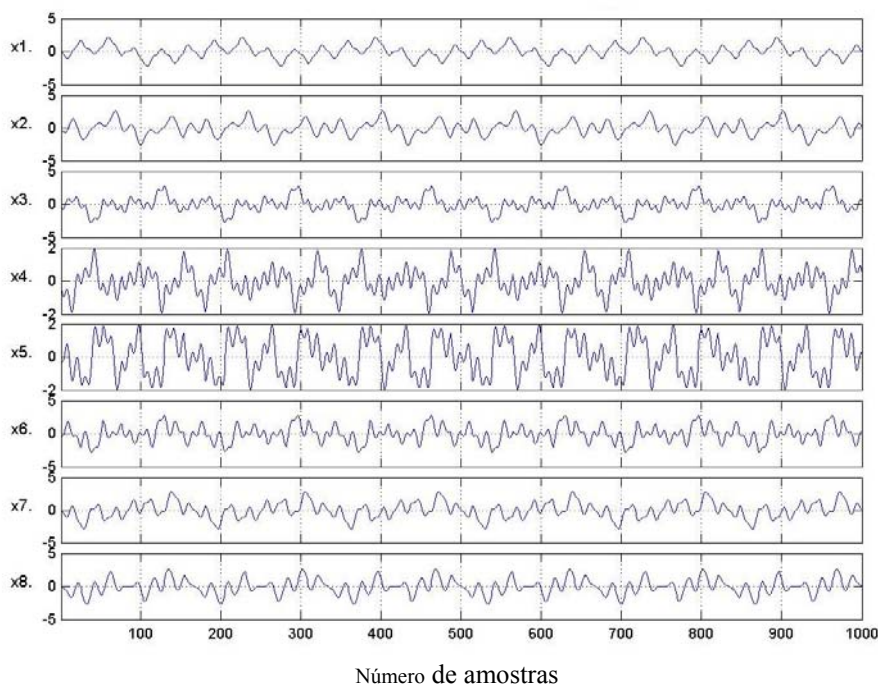
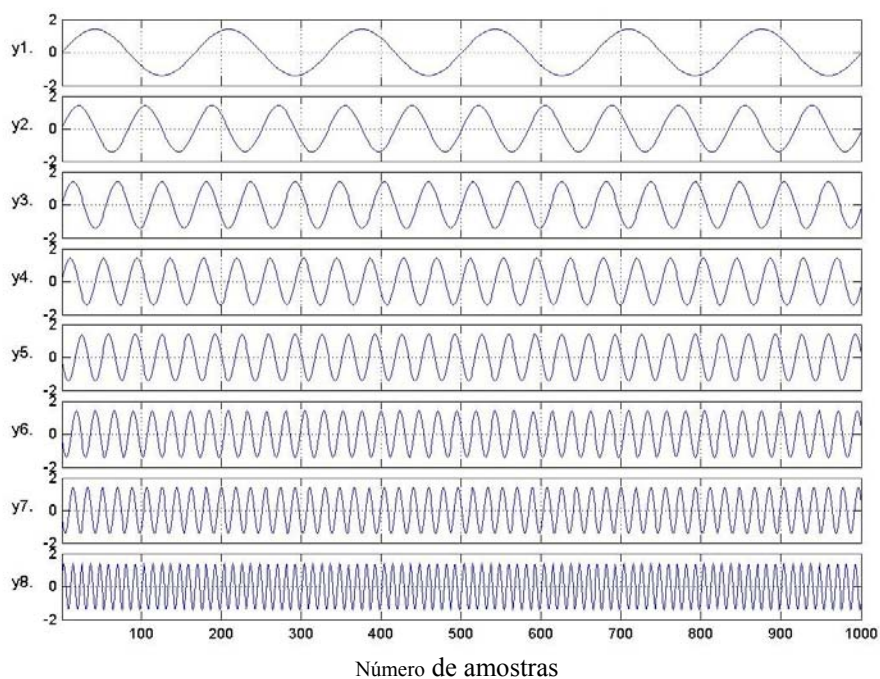


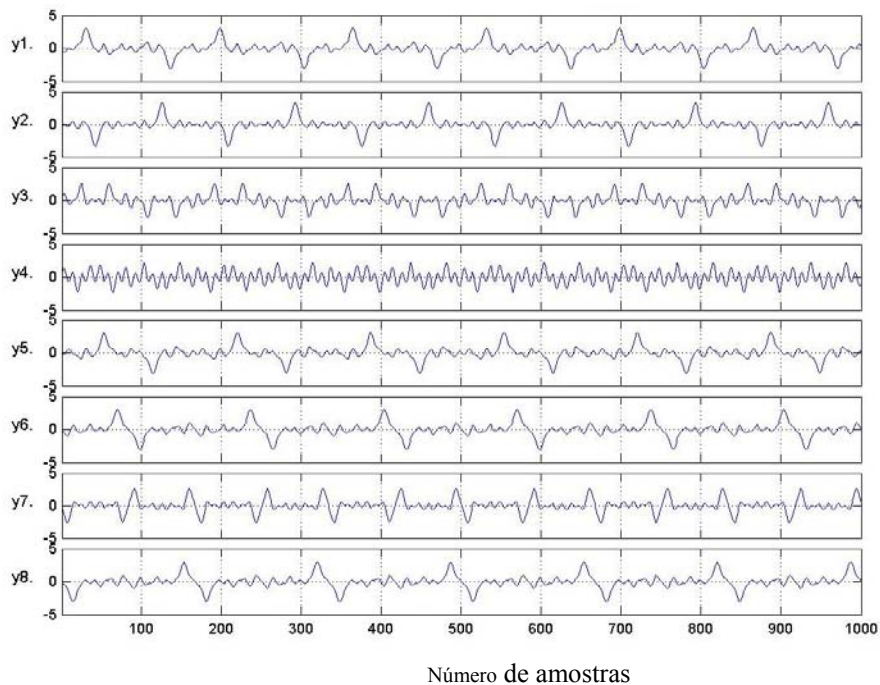
Figura 8.18 – Fontes originais para o experimento 8.2.4.



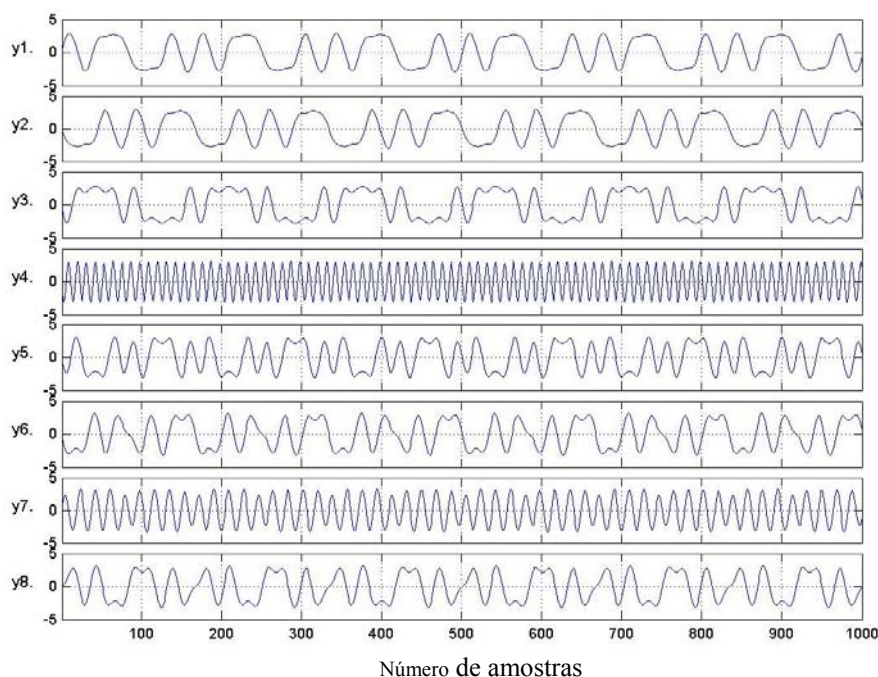
8.19 – Sinais observados pelos sensores para o experimento 8.2.4.



8.20 – Fontes estimadas com o AMUSE para o experimento 8.2.4.



8.21 – Fontes estimadas com o JADE para o experimento 8.2.4.



8.22 – Fontes estimadas com o FlexICA para o experimento 8.2.4.

Finalmente, a Tabela 8.4 e a Figura 8.23 apresentam os resultados dos índices de desempenho obtidos para cada algoritmo.

Algoritmo	PI1	PI2	PI	E1_norm
AMUSE	4.7759e-13	5.5194e-15	0.0000	9.5518e-13
JADE	0.4779	4.8850e-15	36.3437	0.9573
FlexICA	0.3127	9.5548e-4	25.5039	0.6374

Tabela. 8.4 - Índice de desempenho para o experimento 8.2.4.

Como pode ser observado apenas o AMUSE é capaz de separar as fontes propostas neste experimento uma vez que estas são fontes gaussianas.

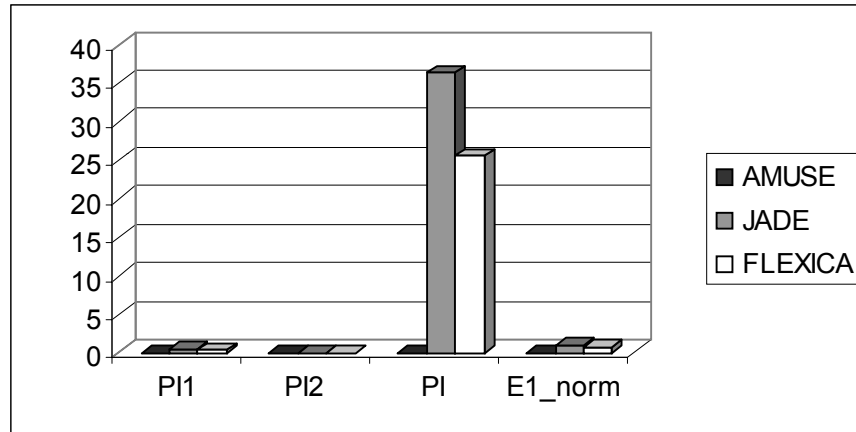


Figura 8.23 - Índice de desempenho para o experimento 8.2.4.

8.2.5 - Separação de Fontes Esparsas

Nesta seção o problema da separação cega recai sobre fontes esparsas, misturadas por meio de uma matriz com elementos não negativos onde $\text{cond}(\mathbf{A}) = 6.5858$ e $\text{det}(\mathbf{A}) = 0.3292$. Fontes esparsas, como as da Figura 8.24, podem ser obtidas utilizando a função $y = \text{filter}(w, 1, k)$ do MATLAB 6.1. Onde w é uma função exponencial e k varia de 0 a 1000.

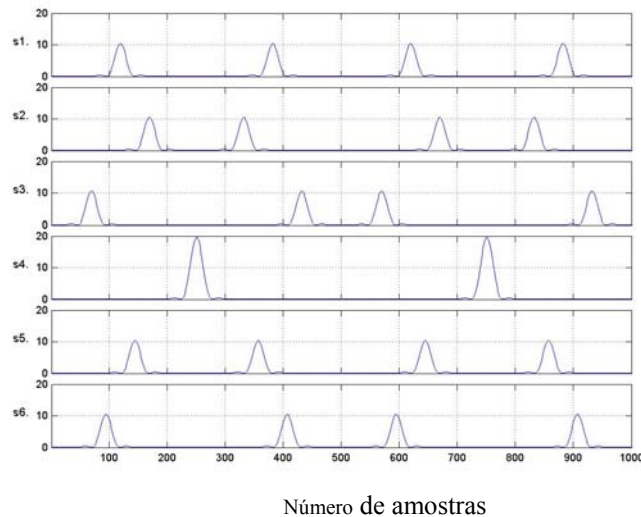


Figura 8.24 - Sinais originais para o experimento 8.2.5.

A medida pelos sensores é apresentada na Figura 8.21.

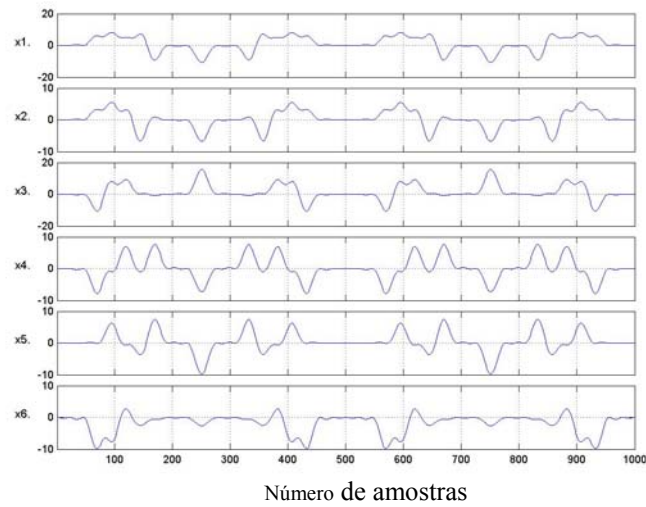


Figura 8.25 – Sinais observados pelos sensores para o experimento 8.2.5.

Mais uma vez as fontes originais não são visíveis a partir das misturas, que parecem apenas sinais de ruído.

As Figuras 8.26, 8.27 e 8.28, mostram, respectivamente, as fontes estimadas pelos algoritmos AMUSE, JADE e FlexICA.

Na Tabela 8.5 e na Figura 8.29 são apresentados os valores do índice de desempenho para os algoritmos.

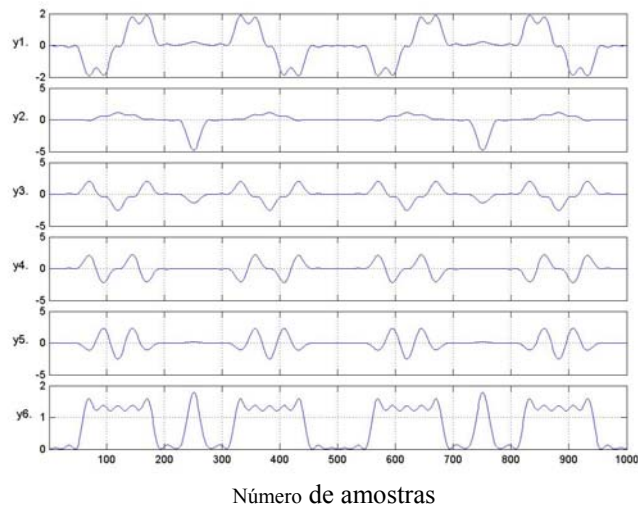


Figura 8.26 – Fontes estimadas com o AMUSE para o experimento 8.2.5.

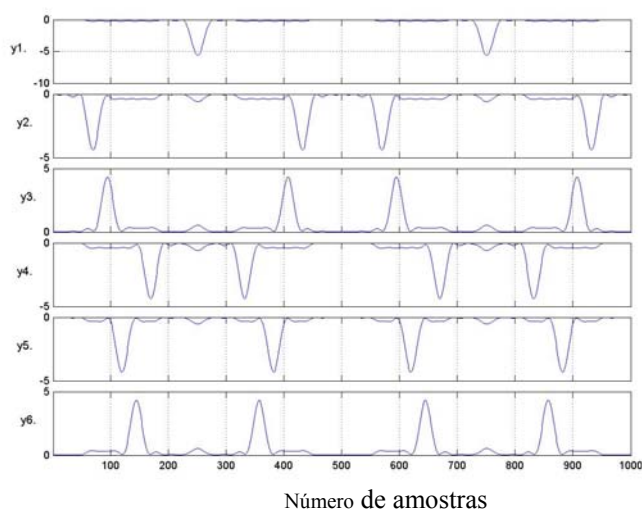


Figura 8.27 – Fontes estimadas com o algoritmo JADE para o experimento 8.2.5.

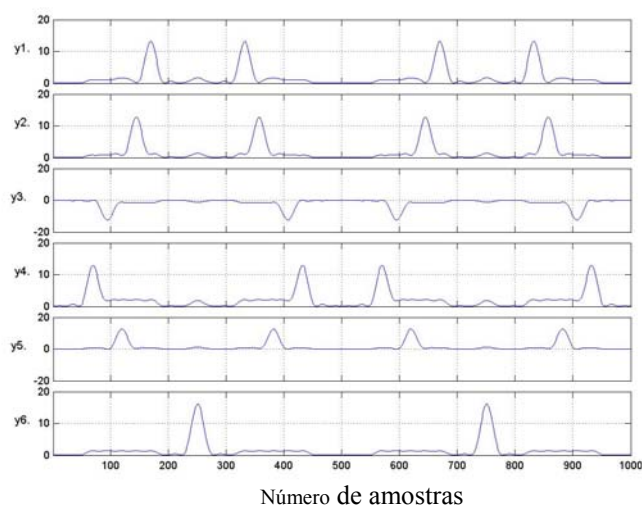


Figura 8.28 - Fontes estimadas com o algoritmo FlexICA para o experimento 8.2.5.

As medidas de desempenho e complexidade dos algoritmos para este exemplo são mostradas na Tabela 8.5 e na Figura 8.29.

Algoritmo	PI1	PI2	PI	E1_norm
AMUSE	0.5371	0.0299	24.0751	1.0312
JADE	0.0608	0.1328	0.2762	0.1232
FlexICA	0.0939	0.2059	0.6473	0.1840

Tabela 8.5 - Índice de desempenho para o experimento 8.2.5.

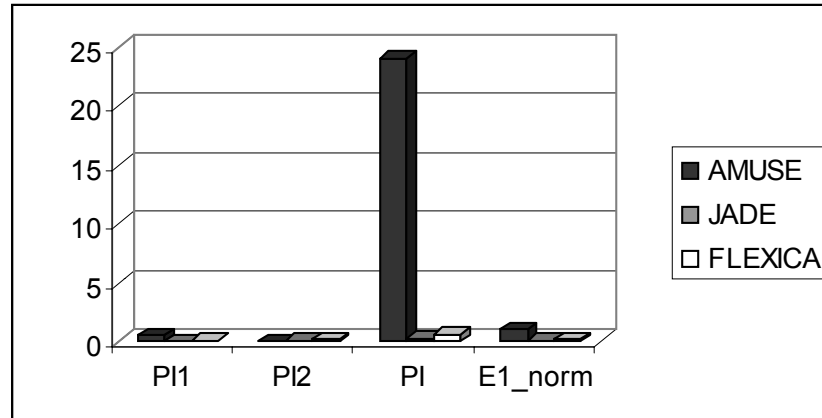


Figura 8.29 - Índice de desempenho para o experimento 8.2.5.

Como pode ser observado, o AMUSE não consegue separar adequadamente fontes esparsas. Pode-se observar através das Figuras 8.27 e 8.28, que algumas fontes estão não positivas, ou seja, estão invertidas. Este fato que pode ser confirmado pela observação da matriz de mistura estimada, por exemplo, pelo algoritmo JADE que apresenta elementos negativos:

$$\mathbf{W}_{\text{JADE}} = \begin{pmatrix} 0.1542 & -0.0374 & -0.1011 & -0.0725 & 0.2521 & 0.2702 \\ 0.0895 & -0.2689 & 0.1475 & -0.3523 & 0.5073 & 0.5304 \\ 0.0830 & -0.1118 & 0.1785 & -0.1853 & 0.3704 & -0.0445 \\ 0.1776 & 0.0414 & 0.0475 & -0.4346 & 0.1145 & 0.4624 \\ 0.0182 & -0.3652 & 0.0083 & -0.5612 & 0.6258 & 0.3532 \\ 0.1352 & -0.4318 & -0.0127 & 0.2242 & -0.0209 & -0.2623 \end{pmatrix}$$

Para solucionar este problema basta multiplicar as colunas de \mathbf{W}_{JADE} com elementos negativos por -1, fazendo com que os pulsos resultantes tenham amplitude positiva, assim como os pulsos originais.

8.3 – Aplicação a Sinais de Telecomunicações

Nesta seção, pretende-se ilustrar o comportamento dos algoritmos sob estudo frente a sinais de telecomunicações. Para isto serão analisados sinais modulados através de modulação analógica e sinais modulados através de modulação digital.

8.3.1 – Modulação Analógica

Neste primeiro experimento são utilizados sinais modulados de forma analógica, sendo que a modulação foi realizada seguindo as seguintes equações:

$$s_1 = (\sin((0.15*(k)+\cos(0.15*k)))); \quad (8.17)$$

$$s_2 = \sin(k/149).*\cos(k/8); \quad (8.18)$$

$$s_3 = (\sin(0.15*k).*\sin(0.15*k)); \quad (8.19)$$

$$s_4 = (\sin(0.4*k).*\sin(0.007*k)); \quad (8.20)$$

Onde k varia de 0 a 10000.

Os sinais originais e os sinais misturados podem ser vistos nas Figuras 8.30 e 8.31 respectivamente.

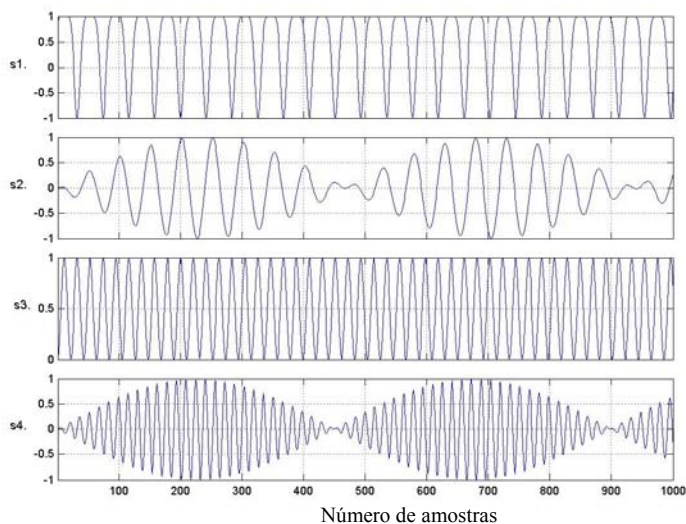


Figura 8.30 – Sinais originais para experimento 8.3.1.

Após a aplicação dos algoritmos para separação foram obtidos os resultados apresentados nas Figuras 8.32, 8.33, 8.34 e valor do índice de desempenho na Tabela 8.6 e na Figura 8.35.

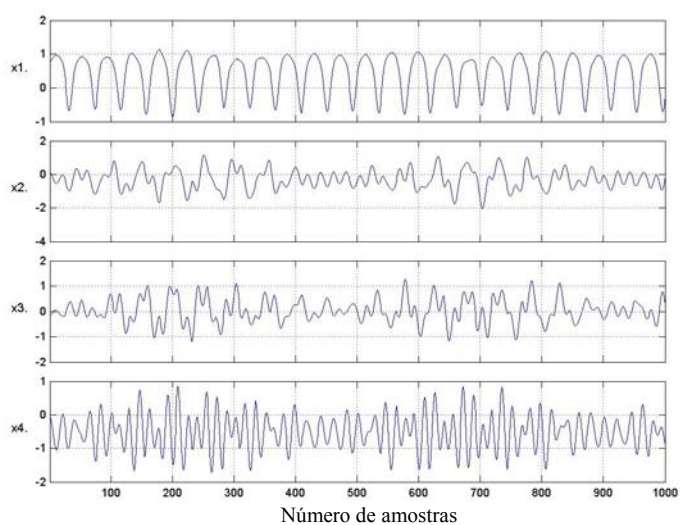


Figura 8.31 – Sinais observados pelos sensores para o experimento 8.3.1.

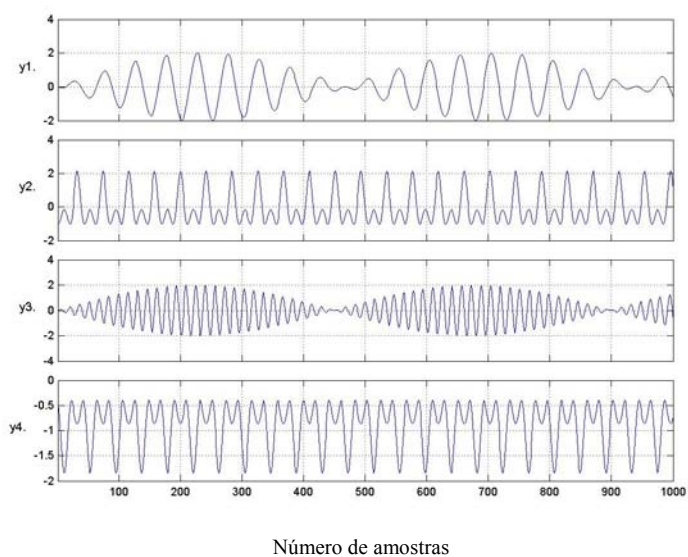


Figura 8.32 – Sinais estimados com o AMUSE para o experimento 8.3.1

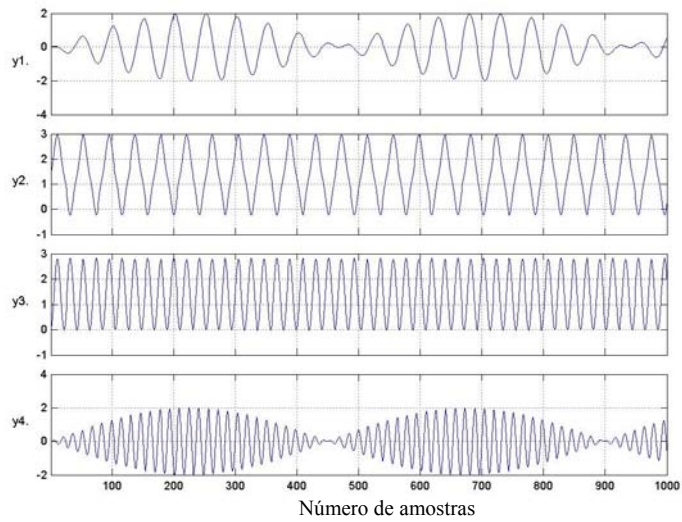


Figura 8.33 – Sinais estimados com o algoritmo JADE para o experimento 8.3.1

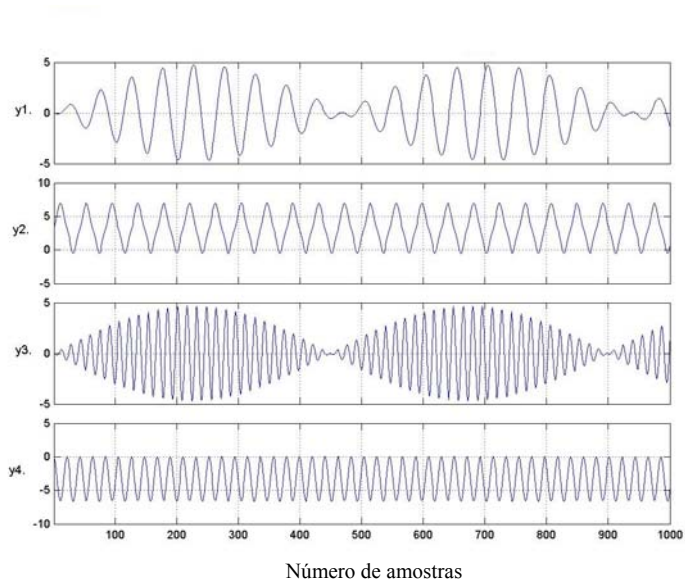


Figura 8.34 - Sinais estimados com o algoritmo FlexICA para o experimento 8.3.1.

Algoritmo	PI1	PI2	PI	E1_norm
AMUSE	0.1050	0.0606	1.6116	0.2045
JADE	0.0747	0.1091	0.9752	0.1186
FlexICA	0.0745	0.1106	0.9860	0.1181

Tabela. 8.6 - Índice de desempenho para o experimento 8.3.1.

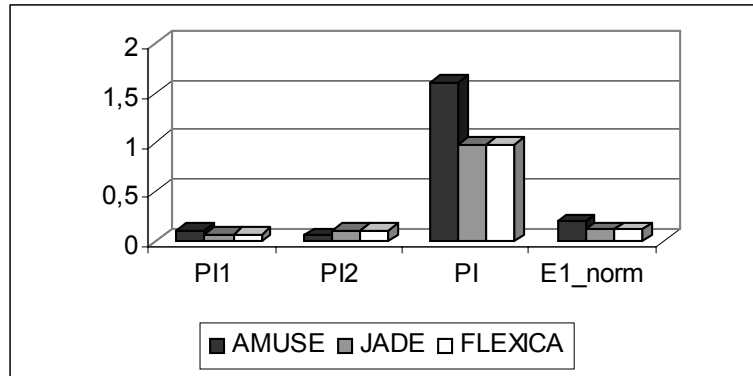


Figura 8.35 - Índice de desempenho para o experimento 8.3.1.

Como pode ser observado pela Figura 8.31, não é possível identificar os sinais originais a partir das medidas feitas pelos sensores, que podem ser representados por um conjunto de antenas.

Analisando os resultados apresentados nas Figuras 8.32, 8.33 e 8.34 e os resultados dos índices de desempenho da Tabela 8.6, pode-se verificar que o AMUSE não apresentou resultados adequados, ou seja, ele não conseguiu recuperar a forma de onda dos sinais originais. Embora o JADE e o FlexICA tenham apresentado resultados melhores do que o AMUSE, no que tange a recuperação da formas de onda originais, seus índices de desempenho também apresentam valores elevados.

8.3.1.1 – Modulação em Frequência

Neste experimento, quatro sinais modulados em frequência foram misturados. Os sinais modulados foram obtidos através da função *modulate*(.) do MATLAB, sendo que:

$$s_1 = \text{modulate}(\sin(2\pi \cdot 13500 \cdot k), 3000, 10000, 'fm'); \quad (8.21)$$

$$s_2 = \text{modulate}(\sin(2\pi \cdot 22270 \cdot k), 1000, 10000, 'fm'); \quad (8.22)$$

$$s_3 = \text{modulate}(\sin(2\pi \cdot 22230 \cdot k), 2000, 10000, 'fm'); \quad (8.23)$$

$$s_4 = \text{modulate}(\sin(2\pi \cdot 22850 \cdot k), 4000, 10000, 'fm'); \quad (8.24)$$

As figuras 8.36 e 8.37 ilustram os sinais fontes e os sinais recebidos pelos sensores respectivamente.

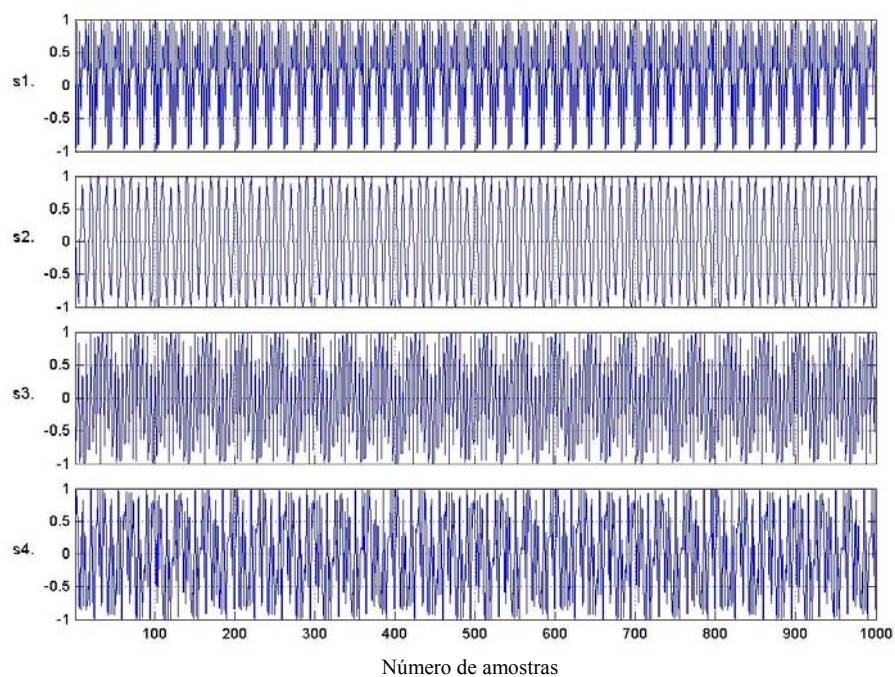


Figura 8.36 – Sinais originais para o experimento 8.3.1.1

Os resultados obtidos pelos algoritmos AMUSE, JADE e FlexICA são apresentados nas Figuras 8.38, 8.39 e 8.40.

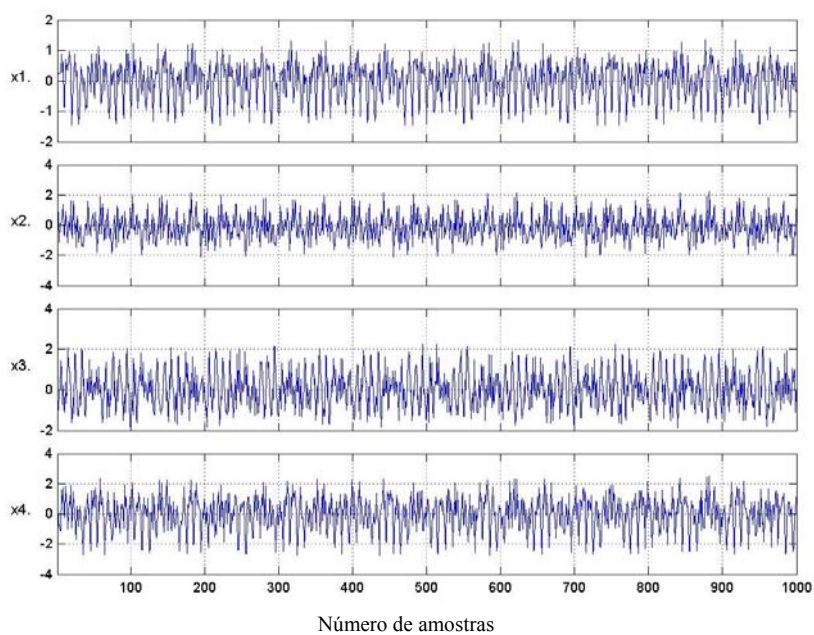


Figura 8.37 – Sinais observados pelos sensores para o experimento 8.3.1.1

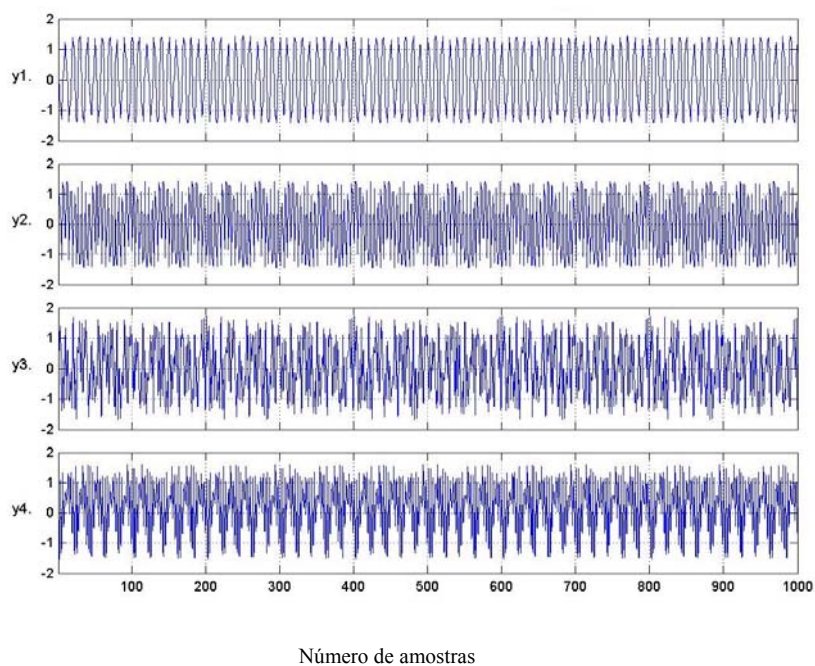


Figura 8.38 - Sinais estimados com o AMUSE para o experimento 8.3.1.1

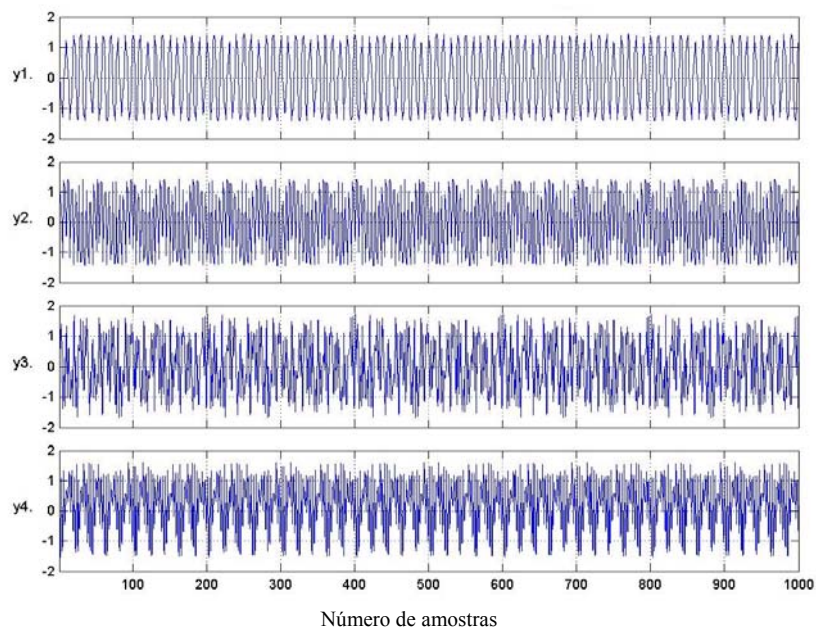


Figura 8.39 - Sinais estimados com o algoritmo JADE para o experimento 8.3.1.1

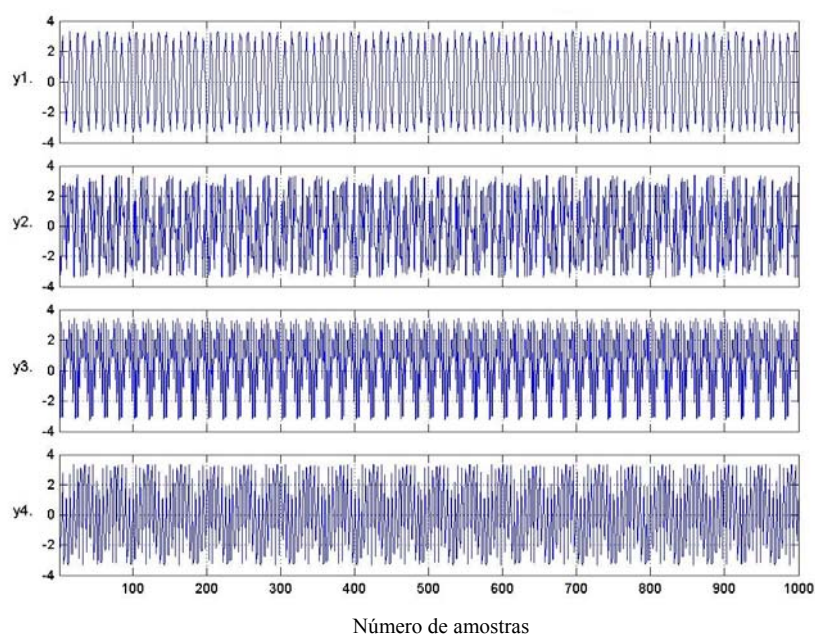


Figura 8.40 - Sinais estimados com o algoritmo FlexICA para o experimento 8.3.1.1

A Tabela 8.7 e a Figura 8.41 apresentam os índices de desempenho obtidos pelos algoritmos AMUSE, JADE e FlexICA, respectivamente.

Algoritmo	PI1	PI2	PI	E1_norm
AMUSE	0.0353	0.0067	0.1679	0.0707
JADE	0.0091	0.0083	0.0038	0.0182
FlexICA	0.0080	0.0082	0.0034	0.0160

Tabela. 8.7 - Índice de desempenho para o experimento 8.3.1.1.

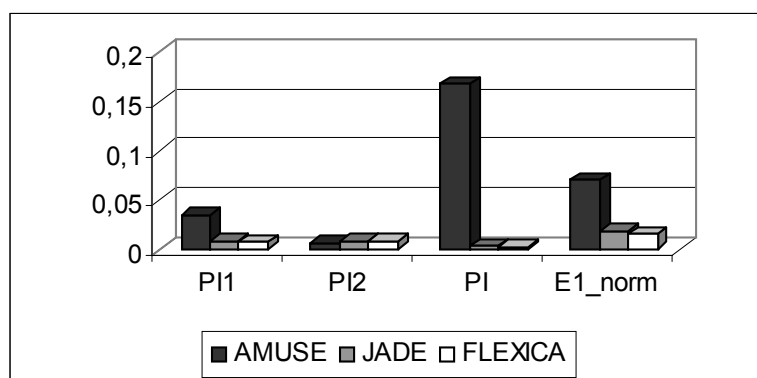


Figura 8.41 - Índice de desempenho para o experimento 8.3.1.1.

Devido ao tipo de modulação utilizada, a análise com relação à preservação da forma de onda torna-se não é simples, por isso, a melhor forma de analisar os resultados obtidos é por meio dos valores dos índices de desempenho apresentados na Tabela 8.7. Por esta tabela pode-se verificar que o AMUSE, não realizou a separação de forma adequada. Por outro lado, o JADE e o FlexICA apresentaram bons resultados, sendo o FlexICA ligeiramente melhor.

8.3.2 – Modulação Digital

Nesta seção será apresentado exemplo de aplicação de BSS/ICA a sinais de telecomunicações, como descrito em (Cardoso, 1996a), no qual se separam misturas de sinais complexos de telecomunicações, um dos problemas típicos de ICA.

Para este experimento são transmitidos quatro sinais modulados digitalmente, sendo que todos eles apresentam *kurtosis* negativo, ou seja, são sinais sub-gaussianos. O canal ou meio de propagação através do qual os sinais são transmitidos é representado por uma matriz de mistura linear e instantânea cujas entradas variam de $[-1,1]$. A Figura 8.42 apresenta as constelações dos sinais 4QAM originais muito utilizados em rádios digitais, enquanto que a Figura 8.43 mostra os sinais misturados que foram captados pelos sensores que podem ser, por exemplo, um conjunto de antenas.

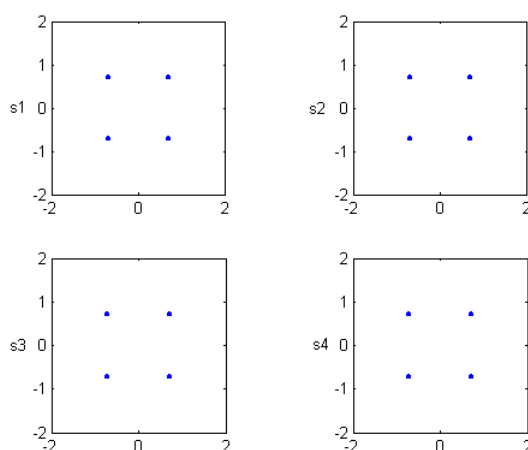


Figura 8.42 – Constatações dos sinais 4QAM originais

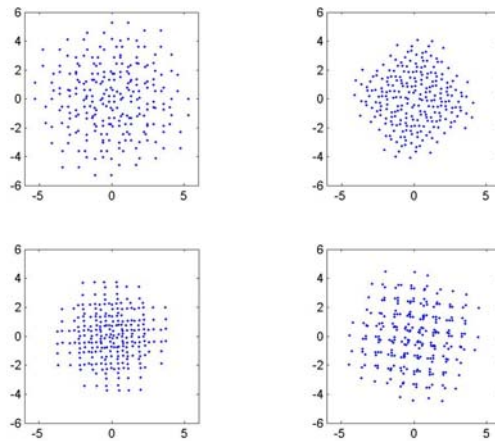


Figura 8.43 – Sinais medidos pelos sensores

Para solucionar este problema e tentar recuperar os sinais originais a partir das medidas dos sensores será utilizada a ICA, ou seja, tentar-se-á separar os sinais obtidos pelos sensores em componentes independentes de tal forma que cada componente corresponda a um sinal original. O algoritmo JADE foi inicialmente desenvolvido para solucionar problemas relacionados a telecomunicações (Cardoso, 1993). Por isso, será o primeiro a ser utilizado para tentar separar os sinais 4QAM misturados. Os sinais recuperados com o auxílio do algoritmo JADE são apresentados na Figura 8.44.

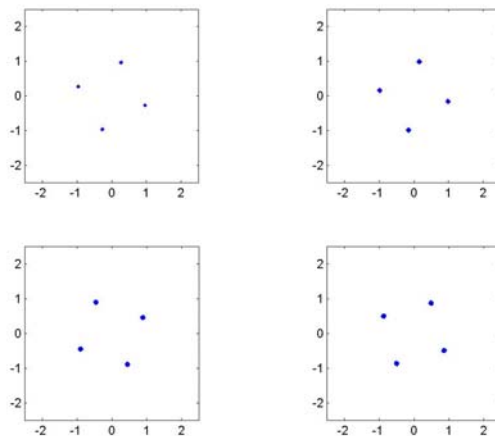


Figura 8.44 – Sinais estimados com o algoritmo JADE para o experimento 8.3.2

Como podem ser observados na Figura 8.44 os sinais foram recuperados, porém com uma alteração na fase. Esta alteração não compromete o sistema de recuperação e pode ser corrigido através de um sistema de recuperação de fase.

8.3.2.1 – FlexICA e sinais complexos

O FlexICA não foi desenvolvido para trabalhar com sinais complexos, por isso, foram necessárias algumas modificações no algoritmo para que ele pudesse separar tais sinais. Em geral, o algoritmo pode ser modificado substituindo y_i transposto em (7.12) por seu valor complexo conjugado transposto.

Além disso, deve-se utilizar uma não linearidade $g_i(y_i)$ adequada. Como os sinais modulados digitalmente são sub-gaussianos, uma função não linear que pode ser utilizada é a função cúbica, dada por:

$$g_i(y_i) = y_i |y_i|^2 \quad (8.25)$$

Desta forma, o algoritmo deixa de ser flexível uma vez que sua não linearidade é fixa. A mudança entre diferentes não linearidades para sinais complexos ainda é um problema não solucionado (Cichocki, 2003). Os resultados obtidos pelo algoritmo FlexICA modificado podem ser vistos na Figura 8.45.

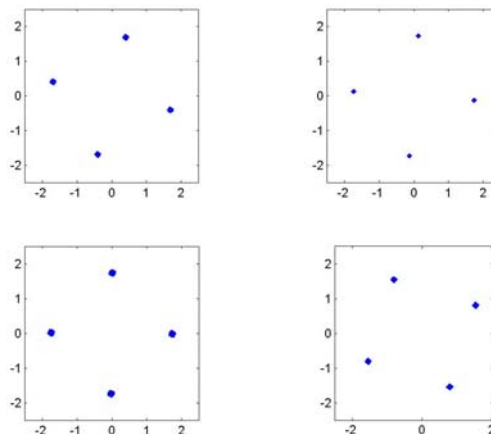


Figura 8.45 – Sinais estimados com o algoritmo FlexICA para o experimento 8.3.2

Como pode ser observado, o algoritmo FlexICA complexo realizou com sucesso a separação dos sinais 4QAM, a menos da alteração na fase. Deve-se observar ainda a alteração na amplitude dos sinais recuperados.

8.4 – Aplicação a Sinais Biomédicos

Nesta seção, os algoritmos sob estudo serão utilizados para o processamento de sinais biomédicos.

Na maioria dos casos, nas medidas de sinais biomédicos os sensores estão situados em locais críticos, nos quais os sensores não registram uma única informação, mas uma mistura de sinais, sendo que cada sensor obtém uma mistura diferente dos sinais medidos. O especialista, então escolhe entre todas as medidas aquela que supostamente apresenta a melhor representação do sinal que se deseja analisar, desprezando os demais (Fuentes, 2000). Em geral, nas medidas de sinais biomédicos estão presentes ainda sinais chamados artefatos, que são, em geral, sinais que representam movimentos musculares ou interferências não patológicas, mas que prejudicam as medidas e, portanto, devem ser separados dos sinais que se deseja medir.

As técnicas de BSS/ICA podem ser utilizadas para o processamento de sinais biomédicos, desde que:

- A mistura das fontes em cada sensor seja linear;
- As misturas sejam independentes no tempo (descorrelacionadas);
- O número de sensores seja no mínimo igual ao número de fontes.

Entre os exemplos de aplicação de separação cega a sinais biomédicos estão: separação de sinais de eletroencefalograma (EEG), eletrocardiograma (ECG), eletrocardiograma fetal (FECG) e materno (MECG), monitoramento de atividade cerebral e comportamento motor através de imagens de fMRI (*functional Magnetic Resonance Imaging*), entre outros.

8.4.1 – Eletrocardiograma Fetal (FECG)

Nesta seção, será reproduzido o experimento proposto em (Choi, 2000c) que trata da separação de sinais de eletrocardiograma fetal. Os dados de eletrocardiograma fetal (FECG) utilizados nesta seção estão disponíveis em (Moor, 1997). Estes dados foram obtidos a partir de 8 sensores cutâneos colocados na mãe e que além do sinal de ECG do feto contêm informações do ECG materno, de órgãos internos e do próprio sistema de medição - artefatos.

A Figura 8.46 apresenta uma representação da disposição dos sensores e dos sinais obtidos durante o teste de FECG e que devem ser separados.

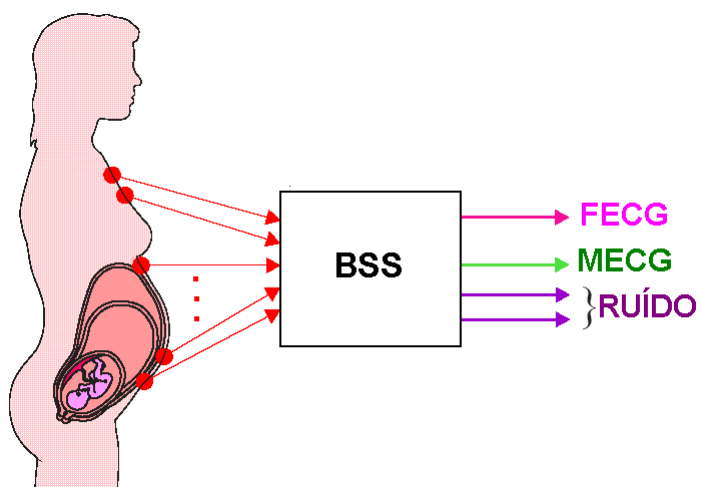


Figura 8.46 – FECG - disposição dos sensores e sinais obtidos.

Foram gravados apenas 5s de teste e a frequência de amostragem utilizada foi de 500Hz. Os sensores são colocados no abdômen e no tórax materno, sendo que:

Sensores 1-5	Abdominal
Sensores 6-8	Torácico

Desta forma, as contribuições mais significativas do sinal do FECG estão contidas principalmente nos sensores de x_1 a x_5 , embora, este não seja um sinal puro, uma vez que o sinal de MECG é predominante. As medidas obtidas no teste de FECG podem ser vistas na Figura 8.47.

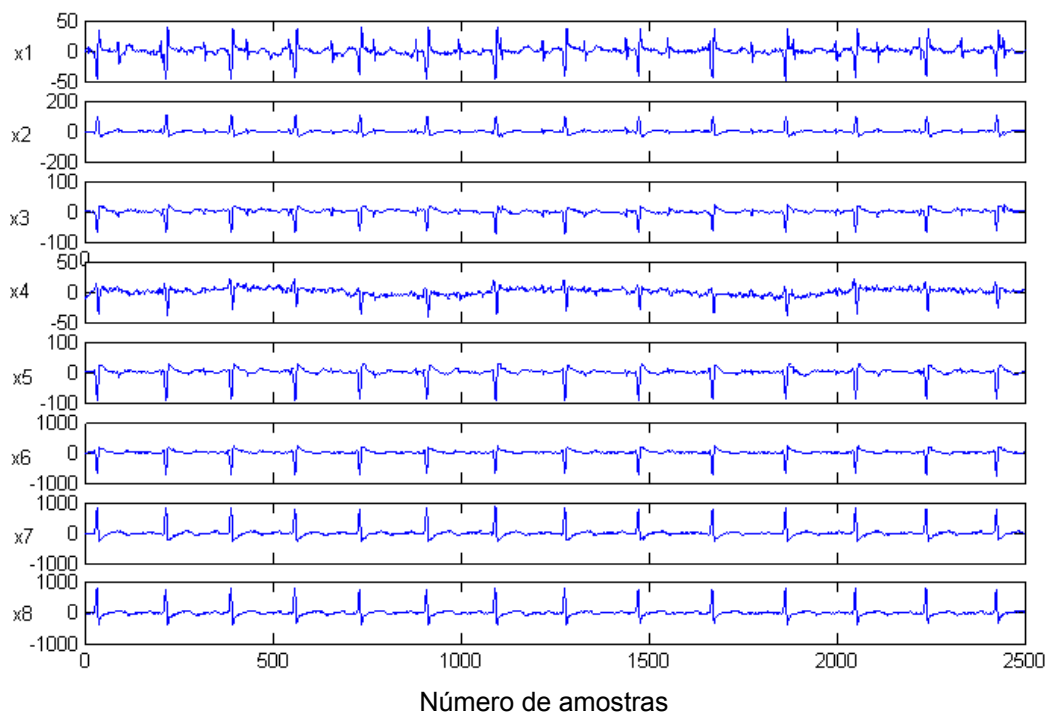


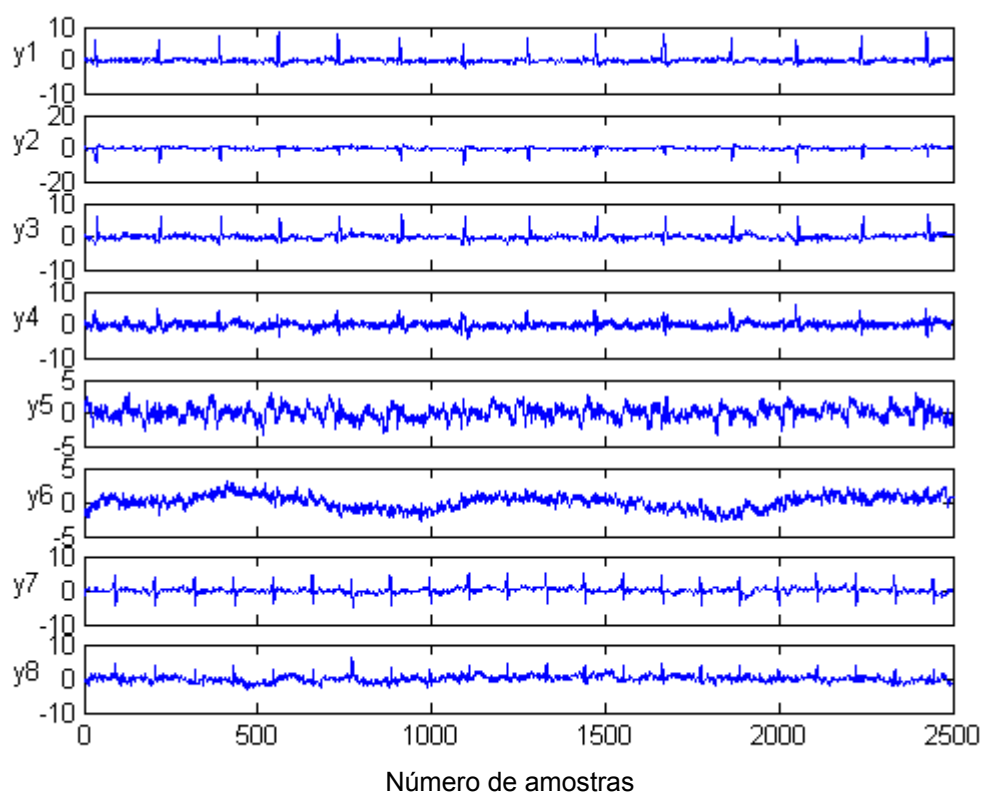
Figura 8.47 – Sinais de FECG medidos pelos sensores.

O objetivo da ICA é separar os sinais medidos pelos sensores x_1 a x_8 em componentes independentes de tal forma que um destes componentes represente o sinal de FECG. Com este objetivo em mente, foram empregados os algoritmos JADE e FlexICA para realizar a separação dos componentes de FECG, sendo que os resultados obtidos por estes algoritmos são mostrados na Figura 8.48 e 8.49 respectivamente.

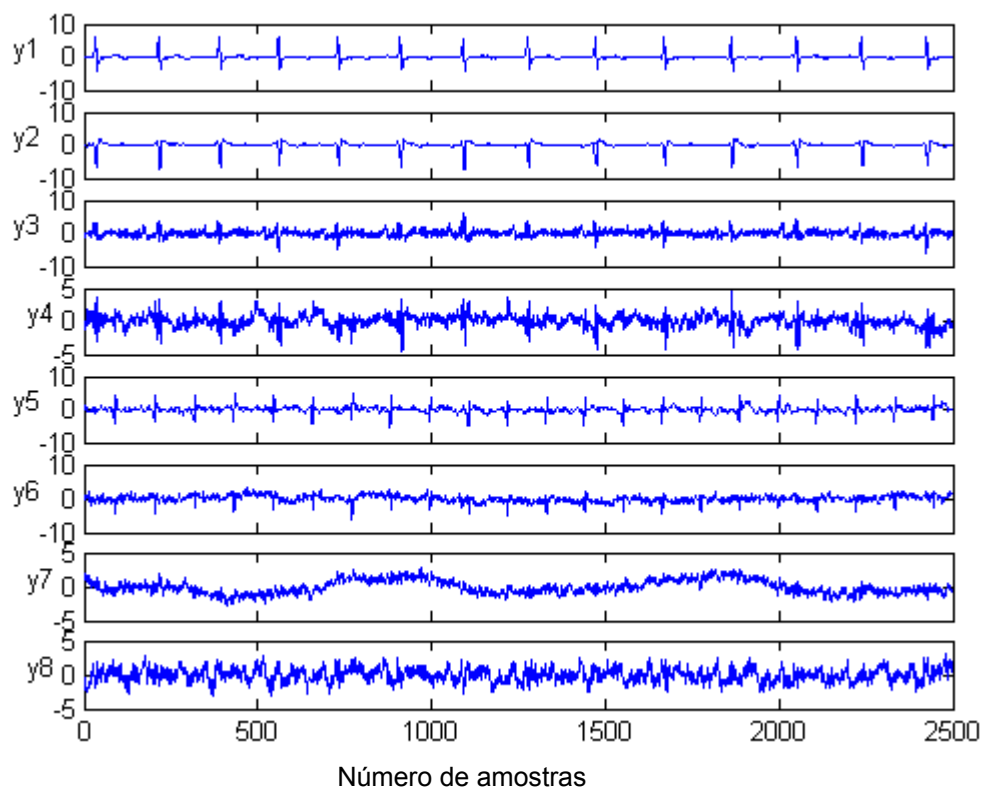
No caso de análise de sinais biomédicos, que são sinais reais, deve-se fazer uma análise gráfica dos sinais estimados, para isto serão levados em consideração os dados obtidos através do FlexICA, uma vez que visualmente apresentam os melhores resultados.

Através da análise da Figura 8.49 e com base em (Choi, 2000c) pode-se dizer que o sinal y_7 corresponde a artefatos causados pela respiração. Os artefatos podem ser causados ainda por efeitos como mau contato nos eletrodos ou sensores ou pela ação de outros órgãos internos do corpo da mãe. Os sinais y_1 e y_2 correspondem ao eletrocardiograma materno, embora y_2 esteja invertido. Finalmente, os sinais y_5 e y_6 correspondem ao eletrocardiograma do feto. Os demais sinais são componentes de ruído.

A partir da análise das Figuras 8.48 e 8.49 pode-se observar que os sinais obtidos pelo FlexICA são bem mais claros, ou seja, as formas de onda obtidas estão mais próximas das originais do que os sinais extraídos pelo JADE.



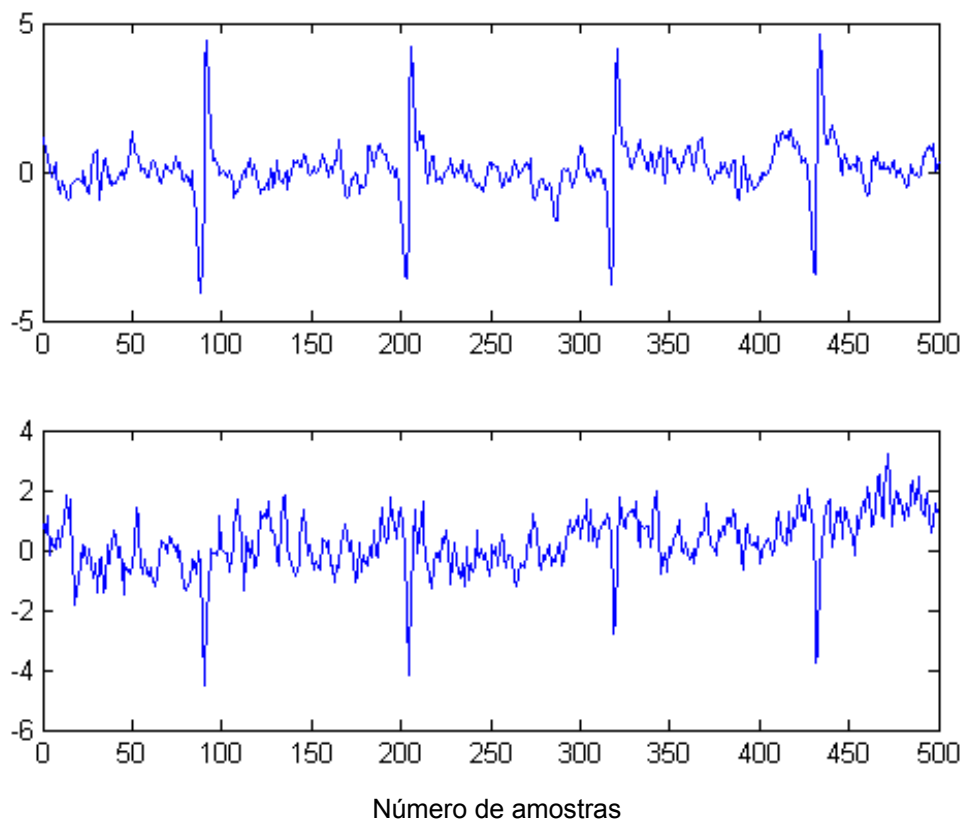
8.48 – Sinais estimados com algoritmo JADE para o experimento 8.4.1



8.49 – Sinais estimados com algoritmo FlexICA para o experimento 8.4.1

Conforme esperado, na Figura 8.49 é possível observar que a amplitude dos sinais de FECG é menor do que a amplitude dos sinais de MECG, por outro lado, a frequência dos sinais de FECG é maior do que a de MECG.

A Figura 8.50 mostra em detalhe os sinais de FECG obtidos pelo FlexICA.



8.50 – Sinais de FECG obtidos pelo FlexICA - Detalhe.

8.5 – Considerações Finais

Neste capítulo foram realizadas simulações computacionais com o objetivo de analisar e comparar o desempenho dos algoritmos AMUSE, JADE e FlexICA. Através desta comparação é possível ter uma visão mais clara dos problemas BSS/ICA.

Os algoritmos foram quantitativamente comparados utilizando quatro índices de desempenho diferentes e são comumente encontrados na literatura.

O algoritmo JADE utilizado nas simulações foi implementado pelo pesquisador que o desenvolveu (Cardoso, 2003b), de tal forma que os resultados obtidos podem ser atribuídos ao algoritmo e não a alguma falha ou idiosincrasia em sua implementação.

Uma diferença fundamental verificada entre os algoritmos é o número de parâmetros a serem ajustados: o JADE não possui parâmetros para ajuste, o FlexICA possui três ou quatro parâmetros e AMUSE apenas um. Em aplicações reais este ajuste de parâmetros pode se tornar um problema, uma vez que uma alteração no sistema de mistura pode levar á contínua atualização nos parâmetros.

Os algoritmos apresentaram resultados adequados mesmo quando mesmo quando foi considerada uma matriz de mistura mal condicionada e quando uma das fontes era um ruído branco gaussiano.

Analisando os experimentos realizados verifica-se que no experimento 8.2.4 apenas o AMUSE obteve o resultado desejado. Isto ocorreu uma vez que os sinais misturados eram gaussianos e, portanto, não podem ser separados por algoritmos de ICA. Por outro lado, no experimento 8.2.5, o AMUSE não obteve um resultado adequado, sendo que o JADE obteve o melhor resultado. A separação de sinais esparsos não é uma tarefa simples e leva a uma área específica dentro da BSS/ICA (Hyvärinen, 2001).

A seção 8.3 teve por objetivo mostrar a viabilidade dos algoritmos em aplicações a problemas de comunicações. Os dois primeiros experimentos desta seção mostraram a separação de sinais com modulação analógica, sendo que o experimento 8.3.1.1 mostra a separação de sinais modulados em frequência, comumente utilizados em sistemas de rádio comunicação e telemetria. O experimento 8.3.2, mostrou um dos problemas atuais de interesse da ICA: a separação de sinais complexos. Neste experimento foram misturados sinais 4QAM, - utilizados frequentemente em rádios digitais -, sendo que, os algoritmos JADE e FlexICA mostraram bom desempenho na separação destes sinais.

Por fim, o experimento 8.4 mostrou uma aplicação típica de ICA: a separação cega de sinais biomédicos, sendo que para os sinais considerados o algoritmo FlexICA apresentou melhor desempenho.

De uma forma geral, pode-se dizer que os algoritmos apresentaram bom desempenho, dentro das limitações teóricas previstas e dentro das limitações do modelo de mistura utilizado. Além disso, pode-se dizer que este capítulo não apresenta apenas comparações, mas mostra importantes aplicações das técnicas de BSS/ICA que podem direcionar futuros trabalhos.

Capítulo 9 – Conclusões e Comentários

O objetivo desta dissertação foi apresentar uma revisão do estado da arte das principais técnicas de separação cega de sinais ou de fontes (BSS), e realizar simulações computacionais comparativas entre algoritmos selecionados dentro do universo de algoritmos existentes. A problemática da separação cega foi apresentada e foram revisados três importantes abordagens para solução deste problema: AMUSE (Tong, 1990); JADE (Cardoso, 1993) e FlexICA (Choi, 1998). Procurou-se reunir estas três abordagens em uma notação comum, de forma a mostrar seus pontos fortes e fracos.

A problemática da BSS consiste em recuperar ou identificar uma ou mais fontes originais quando somente suas misturas estão disponíveis para observação. Nos últimos 15 anos, tem crescido o número de pesquisas sobre a BSS, devido, principalmente, à variedade de aplicações que envolvem múltiplos sensores e múltiplas fontes. Como consequência, cresce o sentimento de que as técnicas para BSS constituem uma ferramenta universal capaz de encontrar aplicações nas mais diferentes áreas.

A separação recebe o adjetivo cega uma vez que nem as fontes originais nem o processo de mistura, ao qual estas fontes foram submetidas, é conhecido (Cardoso, 1999). De fato, tal afirmação é deveras exagerada. Na verdade, é necessário que sejam feitas algumas considerações sobre o modelo de mistura e, em muitos casos, são necessárias o conhecimento de certas estatísticas das fontes originais. Com base neste cenário, nesta dissertação, o sistema de mistura é LTI e, quanto à complexidade, este sistema é representado por uma matriz quadrada. A utilização de uma matriz quadrada é devido ao fato de que o número de fontes foi considerado igual ao número de sensores.

Outro ponto considerado diz respeito ao ruído. Nas simulações computacionais apresentadas o ruído foi considerado uma fonte adicional; embora, ao longo da dissertação, tenham sido feitas considerações ao problema envolvendo ruído aditivo.

Embora muitos pesquisadores venham propondo novos métodos para solucionar o problema da BSS (Cichocki, 2002; Hyvärinen, 2001), a estrutura para sua implementação é freqüentemente a mesma e depende da consideração feita com relação às características das fontes originais. Se as fontes são decorrelacionadas, podem-se utilizar algoritmos simples, rápidos e eficientes, que utilizam, na maioria dos casos, apenas estatísticas de segunda ordem como apresentado no Capítulo 3. Estes algoritmos se baseiam restrição de que as fontes não

apresentam o mesmo espectro de potência e não são aplicáveis quando as fontes são consideradas brancas ou mesmo i.i.d.

Por outro lado, a consideração de que as fontes são estatisticamente independentes é muito pouco restritiva, podendo ser aplicada a um grande número de casos práticos. A independência estatística leva a utilização de um método estatístico denominado Análise de Componentes Independentes como descrito no Capítulo 4. A medida da independência estatística das fontes no Capítulo 4 foi representada por uma função custo, conforme proposto em (Comon, 1994; Hyvärinen, 1999), sendo que a otimização desta função custo permite a estimativa de componentes independentes, que correspondem às estimativas das fontes originais. Desta forma, o problema da BSS é reduzido a um problema de otimização matemática. Como apresentado, as funções custo podem ser baseadas em diferentes critérios: estatísticas de ordem superior (Cardoso, 1993; Comon, 1994; Tong, 1990); informação mútua (Comon, 1994); negentropia (Hyvärinen, 1999); máxima verossimilhança; maximização da informação, entre outros. Foi mostrado em (Cardoso, 1996), que sob certas condições, estas funções custo são equivalentes. Onde, porém, permanecem algumas divergências pode-se escolher uma função custo de acordo com as características da aplicação como, por exemplo, complexidade computacional, características estatísticas das fontes e estabilidade. O mesmo vale para o algoritmo de otimização a ser utilizado, sendo que, no caso do algoritmo, deve-se considerar ainda a robustez frente ao ruído e se serão recuperadas todas as fontes ou apenas algumas.

A formulação para a Análise de Componentes Independentes (ICA) utilizada nesta dissertação, principalmente nas simulações computacionais, foi a padrão, que assume fortes considerações ou simplificações: o número de sensores é igual ao número de fontes, não há ruído adicionado aos sensores e as fontes são modeladas como variável aleatória (v.a). Por outro lado, quando a ICA é aplicada a problemas reais, torna-se evidente que estas restrições são muito severas e, embora o modelo padrão para ICA forneça uma boa aproximação da solução do problema, é necessário ir além das restrições padrão e utilizar algoritmos que possam se adaptar a tais situações. Por este motivo, a maioria dos problemas reais ainda está em aberto e fornecem importantes direções para futuras pesquisas.

Para melhor caracterização e compreensão do referido problema foram comparados três algoritmos: AMUSE, JADE e o FlexICA. Os dois primeiros algoritmos foram escolhidos por serem importantes marcos no desenvolvimento da BSS; sendo que o primeiro utiliza apenas estatísticas de segunda ordem e o segundo estatísticas de ordem superior. O algoritmo

FlexICA foi escolhido por ser um exemplo de algoritmo adaptativo, que permite a separação de sinais sub e super-gaussianos.

O AMUSE (Tong, 1990), discutido no Capítulo 5, foi proposto para solucionar o problema da separação ou identificação cega de sinais através de diagonalizações simultâneas da matriz de covariância e matriz de covariância atrasada no tempo dos sinais medidos pelos sensores. A motivação para o desenvolvimento do AMUSE surgiu devido ao fato de que os algoritmos que consideram que os sinais fontes são estatisticamente independentes são incapazes de trabalhar com mais do que um sinal gaussiano, uma vez que os momentos de quarta ordem de sinais gaussianos são completamente especificados através de momentos de segunda ordem. Assim, o AMUSE utiliza apenas estatística de segunda ordem (SOS); sendo que a principal condição ou restrição para a utilização do AMUSE é que os sinais originais sejam temporalmente decorrelacionados. (Cichocki, 2002).

No Capítulo 6, foi discutido o algoritmo JADE, desenvolvido por J. F. Cardoso (Cardoso, 1993), que soluciona o problema da BSS considerando que as fontes são estatisticamente independentes. O JADE não faz nenhuma restrição quanto aos valores dos *kurtosis* das fontes originais, o que significa que ele trabalha com fontes super e sub-gaussianas sem a necessidade de nenhum ajuste inicial. Desta forma, o JADE remove a restrição de que as fontes originais tenham o mesmo *kurtosis*, como ocorre em algoritmos adaptativos que utilizam não linearidades.

Para realizar a separação cega, o JADE, inicialmente, realiza o branqueamento dos dados medidos pelos sensores, de forma a obter um vetor decorrelacionado. Após o branqueamento, o algoritmo procura encontrar uma matriz de rotação que simultaneamente diagonalize múltiplas matrizes contendo informações estatísticas de 4ª ordem dos dados branqueados.

Em geral, o desempenho dos algoritmos adaptativos para ICA depende da escolha de funções não lineares, que, por sua vez, é escolhida em função da distribuição de probabilidade das fontes a serem estimadas. Como a distribuição das fontes não é conhecida *a priori*, são utilizados modelos hipotéticos de distribuição. Quando são misturadas fontes super e sub-gaussianas a escolha da não linearidade é fundamental para o sucesso da separação. Com este objetivo, vários algoritmos foram desenvolvidos (Lee, 1999). No Capítulo 7, foi analisado o algoritmo FlexICA que emprega um modelo de densidade gaussiano generalizado capaz de lidar com fontes super e sub-gaussianas. Em comparação com outros algoritmos, a função não linear no algoritmo FlexICA é controlada por um único parâmetro: um expoente gaussiano.

Após a apresentação dos algoritmos selecionados foram realizadas, no Capítulo 8, várias simulações para testar desempenho destes. Objetivando comparar o desempenho dos algoritmos utilizou-se uma metodologia que teve como ponto de partida uma condição simples: mistura bem condicionada e sem ruído aditivo. Progressivamente foi-se aumentando o grau de complexidade do problema até se chegar à separação de sinais biomédicos reais. Foi também apresentada separação de sinais de comunicação com modulação analógica e digital.

Estas simulações mostram que os algoritmos testados apresentaram bom desempenho, respeitadas suas limitações teóricas e as limitações do modelo utilizado.

9.1 – Recomendações para Trabalhos Futuros

A teoria envolvida no estudo da separação cega de fontes é recente, suas bases ainda estão sendo consolidadas e, sem dúvida, há um vasto caminho a percorrer. Uma prova disto é que muitas aplicações práticas não estão ao alcance das técnicas desenvolvidas. A dificuldade em se lidar com problemas práticos ocorre uma vez que estes problemas podem apresentar algumas das seguintes propriedades:

- Fontes são não estacionárias;
- O modelo de mistura não é linear nem invariante no tempo;
- Os sinais originais são corrompidos por ruído;
- Existem mais fontes do que sensores;
- O número de amostras é pequeno.

O estudo e o desenvolvimento de técnicas que possam lidar com estas propriedades mostram importantes direções de pesquisa para aprimorar o modelo básico de separação cega (Hyvärinen, 2002; Inki, 2002; Lewicki, 2000; Park, 1999; Taleb, 1999; Vigliano, 2003).

Com relação às áreas de pesquisa aplicada, podem-se delinear algumas direções nas áreas de comunicações, engenharia biomédica, engenharia elétrica entre outras.

O desempenho de sistemas de reconhecimento de voz é fortemente degradado por sinais de ruído e interferência. Os humanos, por sua vez, têm facilidade para reconhecer sinais de voz sob as mais diversas circunstâncias, solucionando de forma simples o problema do coquetel (*cocktail party*). Isto leva, então, ao desenvolvimento de algoritmos para BSS/ICA inspirados em sistemas biológicos.

Na área de engenharia biomédica, importantes direções de pesquisa podem ser delineadas. Entre elas está a separação de sinais de eletro encefalograma (EEG) para mapeamento cerebral e identificação de anomalias. Além disso, é crescente o interesse no aprimoramento de técnicas de pré-processamento para remoção de interferências.

Em sistemas de potência, as técnicas de BSS/ICA se mostram promissoras como estimadores de estado.

Uma das áreas com maior número de aplicações das técnicas de BSS/ICA é, sem dúvida, a área de comunicações. Isto se deve principalmente à crescente demanda dos sistemas de comunicação sem fio (GPS, sistemas de telemetria, telefonia móvel) tanto em aplicações aeroespaciais como em aplicações comerciais. As aplicações em comunicações incluem: a separação de sinais de rádio frequência em canais com desvanecimento (*fading*); determinação da direção de chegada (DOA) para detecção, por exemplo, de sinais SS-CDMA, como os sinais GPS; conformação de feixe para sistemas com diversidade. Outro exemplo de aplicação na área de comunicações é a reconstrução das polarizações multiplexadas em enlaces de microondas, eliminando assim as interferências entre as transmissões.

Quando se trabalha com sinais de comunicações surge, algumas vezes, a necessidade de se trabalhar com sinais complexos. Uma consequência natural desta necessidade é o estudo e a implementação de algoritmos robustos, rápidos e computacionalmente simples, que possam trabalhar com sinais complexos.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALVAREZ, S. A. C., *Performance Index* (comunicação particular). Seville/Spain: University of Seville, 2003.
- ALVAREZ, Sergio A. C., *Una Visión Unificada de los Algoritmos de Separación Ciega de Fuentes*. Tese de Doctorado. Universidad de Vigo, 1999.
- AMARI, S-I., Estimating Functions of Independent Component Analysis for Temporally Correlated Signals. *Neural Computation*, 12(9), p.2083-2107, 2000.
- AMARI, S-I., Douglas, S. C., Cichocki, A., Yang, H. H., Multichannel Blind Deconvolution and Equalization using Natural Gradient. Proc. IEEE Symp. on Wireless Communication, p. 101-107, 1997.
- AMARI, S-I, Cichocki, A., Yang, H. H., A New Learning Algorithm for Blind Signal Separation. In Michael C. Mozer, David S. Toretzky and Michael E. Hasselmo Editors, *Advances in Neural Information Processing Systems*, v. 8, p. 757-763, MIT Press, Cambridge, MA, 1996.
- BARRÉRE, J., Chabriel, G., A Compact Sensor Array for Blind Separation of Sources. *IEEE Trans. on Circuits and Systems - I: Fundamentals, Theory and Applications*, v.49, n.5, p. 565 – 574, Maio 2002.
- BARROS, A. K., Ohnishi, N., *Fetal Heart Rate Variability Using a Frequency Tracking Algorithm*. 3rd International Conference on Independent Component Analysis and Blind Signal Separation (ICA'01), California, Dec. 2001. [on line <http://ica2001.ucsd.edu/>; capturado em 11/Jul/2003]
- BARROS, A. K., Vigário, R., Jousmaki, V., Ohnishi, N., Extration of Event-Related Signals from Multi-Channel Bioelectrical Measurements. *IEEE Trans. on Biomedical Engineering*, v. 47, n.5, p. 583-588, 2000.
- BARROS, A. K., Cichocki, A., Robust batch algorithm for sequencial blind extraction of noisy biomedical signals. In Proc. Symposium on Signal and its Applications (ISSP'99), p. 363-366, Brisbane, Australia, 1999.
- BARROS, A. K., Yoshizawa, M., Yasuda, Y., Filtering Non-Correlated Noise in Impedance Cardiography. *IEEE Trans. on Biomedical Engineering*, v. 42, p. 324-327, 1995.
- BASU, S., ICA: A Critical Review of Three Prominent Approaches. Technical Report , MIT, Apr. 2000.
- BELL, A., Sejnowski, T. J. (1995). An Information Maximization Approach to Blind Separation and Deconvolution. *Neural Computing*, vol. 7, no. 6, pp. 1129-1159, 1995.
- BELOUHRANI, A., Amin, M., Jammer mitigation in spread spectrum communication using blind source separation. *Signal Processing*, 80(4): 723-729, 2000.

- BELOUHRANI, A., Cichocki, A., Robust Withening Procedure in Blind Source Separation Context. *Electronic Letters*, 36(24), p. 2050-2051, Nov. 2000 (b).
- BELOUHRANI, A., Amin, M., Blind Source Separation based on Time-Frequency Signal Representations. *IEEE Trans. on Signal Processing*, 46(11), p.2888-2897, 1998.
- BELOUHRANI, A., Merain, K. A., Cardoso, J-F., Moulines, E., A Blind Separation Techniques Using Second Order Statistics. *IEEE Trans. on Signal Processing*, v.45, p.434-444, Fev. 1997.
- BELOUHRANI, A., Merain, K. A., Cardoso, J-F., Moulines, E., Second Order Blind Separation of Temporally Correlated Sources. In *Proc. of International Conference on Signal Processing*, p.346-351, Cyprus, 1993.
- _____, BLISS Project [*on line* <http://www.bliss-project.org>]
- BUCCIGROSSI, R. W., Breaking up is hard to do: Blind Signal Separation. WPE-II University of Pensilvania, Mar. 1997. [*on line* <http://citseer.nj.nec.com/429031.html>; capturado em 25/Abr./2003].
- CAO, X. -R., Liu, R. -W., General Approach to Blind Source Separation. *IEEE Trans. on Signal Processing*, 44(3), pp. 562-571, 1996.
- CARDOSO, J-F, Independent Component Analysis of the Cosmic Microwave Background, *Proc. International Workshop on Independent Component Analysis and Signal Processing, ICA'03, Nara, Japão, 2003 (a)*.
- CARDOSO, J-F, Blind Source Separation and Independent Component Analysis, Jul. 2003(b). [*on line* <http://www.tsi.enst.fr/~cardoso/guidesepsou.html>; capturado em 06/Jul/2003].
- CARDOSO, J-F., High-Order Contrasts for Independent Component Analysis. *Neural Computation*, v. 11, n.1, p. 157-192, Jan. 1999.
- CARDOSO, J-F., Blind Separation: Statistical Principles. *Proceedings of IEEE*, 86(10), p. 2009-2025, 1998.
- CARDOSO, J-F., Infomax and Maximum Likelihood for Source Separation. *IEEE Letters on Signal Processing*, v.4, n.4, p. 112-114, Abr. 1997.
- CARDOSO, J. -F., Laheld, B., Equivariant Adaptive Source Separation. *IEEE Trans. on Signal Processing*, 44(12): 3017-3030, 1996 (a).
- CARDOSO, J-F., Souloumiac, A., Jacobi Angles for Simultaneous Diagonalization. In *SIAM Journal of Matrix Analysis and Applications*, v. 17, n.1, p. 161-164, Jan. 1996 (b).
- CARDOSO, J-F., Comon, P., Independent Component Analysis: A Survey of Some Algebraic Methods. In *Proc. ISCAS'96*, v.2, p. 93-96, 1996 (c).
- CARDOSO, J-F., Souloumiac, A., A Blind Beam-forming for Non-Gaussian Signals. *IEE, Proc. Part F*, 140(6), p. 362-370, Dez. 1993.

- CARDOSO, J-F., Source Separation using High Order Moments. In International Conference on Acoustics, Speech and Signal Processing (ICASSP 1989), v.4, p. 2109-2112, Glaslow, UK, 1989.
- CARVALHO, C. R., Análise em Componentes Independentes Baseada em um Método Geométrico-Probabilístico. Tese de Mestrado, UFMG, MG, Dez. 2000.
- CASTEDO, L., Escudero, C., Dapena, A., A Blind Signal Separation Method for Multiuser Communication. *IEEE Trans. on Signal Processing*, 45(5): 1343-1348, 1997.
- CHAN, D. C. B., Blind Signal Separation. PhD Thesis, Signal Processing and Communications Lab., Department of Engineering, University of Cambridge, UK, Jan. 1997.
- CHANG, C., Ding, Z., Yau, S. F., Chan, F. H. Y., A Matrix Pencil Approach to Blind Separation of Colored Nonstationary Signals. *IEEE Trans. on Signal Processing*, v.48, p.900-907, Mar. 2000.
- CHEN, S., Wu, Y., McLaughlin, Genetic algorithm optimization for blind channel identification with high order cumulant fitting. *IEEE Trans. Evolutionary Computation*, vol. 1, no. 4, pp. 259-265, 1997.
- CHOI, S., *FlexICA* (comunicação particular). Pohang/Korea: POSTECH, 2003.
- CHOI, S. Cichocki, A., Belouchrani, A., Second Order Nonstationary Source Separation. *Journal of VLSI Signal Processing*, vol. 32, no. 1-2, pp. 93-104, Aug. 2002.
- CHOI, S. Cichocki, A., Belouchrani, A., Blind Separation of Second Order Nonstationary and Temporally Colored Sources. In Proc. IEEE Workshop on Statistical Signal Processing (IEEE-SSP-2001), p. 444-447, Singapore, Ago. 2001.
- CHOI, S. Cichocki, A., Blind Separation of Nonstationary and Temporally Correlated Sources from Noisy Mixtures. *IEEE Workshop on Neural Networks for Signal Processing, NNSP'2000*, Sydney, Australia, p. 405-414, Dec. 2000.
- CHOI, S. Cichocki, A., Blind Separation of Nonstationary in Noisy Mixtures. *Electronic Letters*, 36(9), p. 848-849, Abr. 2000 (b).
- CHOI, S. Cichocki, A., Amari, S-I, Flexible Independent Component Analysis. *Journal of VLSI Signal Processing*, v. 26, n.1/2, p. 25-38, 2000 (c).
- CHOI, S. Cichocki, A., Amari, S-I, Flexible Independent Component Analysis. In Proc. of IEEE Workshop on Neural Network for Signal Processing (NNSP), p. 83-92, Cambridge, UK, 1998.
- CHOI, S., Defferential Hebbian Type Learning Algorithms for Decorrelation and Independent Component Analysis. *Electronic Letters*, v. 34, n. 9, p. 900-901, Apr. 1998 (b).
- CICHOCKI, A., *Complex ICA* (comunicação particular). Saitama/Japan: RIKEN, 2003.
- CICHOCKI, A., Amari, S-I. Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. John Wiley & Sons, 2002.

- CICHOCKI, A., Karhunen, J., Kasprzak, W., Vigário, R., Neural Networks for Blind Separation with Unknown Number of Sources. *Neurocomputing*, 24 (1-3), p. 55-93, Feb. 1999.
- CICHOCKI, A., Sabala, I., Choi, S., Orsier, B., Szupiluk, R., *Self Adaptive Independent Component Analysis for Sub-Gaussian and Super-Gaussian Mixtures with Unknown Number of Sources and Adaptive Noise*. Proc. 1997 International Symposium on Nonlinear Theory and its Applications (NOLTA-97), v.2, p. 731-734, Hawaii, Dec. 1997.
- COMON, P., Independent Component Analysis, A New Concept? *Signal Processing*, Special Issue on High Order Statistics, n. 36, p. 287-314, 1994.
- COMON, P., *Separation of Stochastic Process*. Proc. of Workshop on High Order Spectral Analysis, p. 174-179, Vail, Colorado, Jun. 1989.
- DEGERINE, S., Malki, R., Second Order Blind Separation of Sources Based on Canonical Partial Innovations. *IEEE Trans. on Signal Processing*. 48(3): 629-641, 2000.
- DER, R., *Blind Separation*. Thesis Department of Electrical & Computer Engineering, McGill University, Set.2001.
- DELFOSSÉ, N. Loubaton, P., *Adaptive Separation of Independent Sources: A Deflation Approach*. In International Conference on Acoustics, Speech and Signal Processing, v. 4, p. 41-44, Jul.1994.
- DOUGLAS, S. C., Cichocki, A., Amri, S-I, A Bias Removal Technique for Blind Source Separation with Noisy Measurements. *Electronic Letters*, v. 34, p. 1379-1380, 1998.
- DOUGLAS, S. C., Cichocki, A., Amri, S-I, Multichannel Blind Separation and Deconvolution of Sources with Arbitrary Distributions. Proc. of 1997 IEEE Workshop on Neural Networks for Signal Processing (NNSP-97), p. 436-445, Sept. 1997.
- FUENTES, J. J. M., *Aplicación de Técnicas de Separación Ciega de Fuentes al Procesado de Señal y Comunicaciones*. Tese de Doctorado, Departamento de Teoría de la Señal y Comunicaciones, Escuela Politécnica Superior, Universidad Carlos III, Madrid, 2001.
- GIANNAKOPOULOS, X., *Comparison of Adaptive Independent Component Analysis Algorithm*. Dipl. Eng. Thesis, Helsinki University of Technology, Finland, 1999.
- GIROLAMI, M., *Self Organizing Neural Networks – Independent Component Analysis and Blind Source Separation*. Springer-Verlag, 1999.
- GODARD, D. N., Self-recovering Equalization and Carrier Tracking in two-dimensional Data Communication Systems. *IEEE Trans. on Communications*, 28(11), p. 1867-1875, Nov. 1980.
- GOLUB, G. H., Loan, C. F. V., *Matrix Computation*. John Hopkins, 2nd Ed., 1989.
- HAYKIN, S., *Cocktail Party Phenomenon: What Is It, and How Do We Solve It?* In European Summer School on Independent Component Analysis – From Theory to Applications, Berlin, Germany, Jun. 2003. [on line http://ida.gmd.de/~harmeli/summer_school/; capturado em 15/Set./2003].

- HAYKIN, S., *Neural Networks – A Comprehensive Foundation*. Prentice Hall, 2 ed., 1998.
- HAYKIN, S., *Adaptive Filter Theory*. Prentice Hall, 3 ed., 1996.
- HAYKIN, S., *Blind Deconvolution*. Prentice Hall, Englewood Cliffs, New Jersey, 1994.
- HYVARINEN, A., Oja, E., Karhunen, J., *Independent Component Analysis*. John Wiley & Sons, 2001.
- HYVARINEN, A., Oja, E., *Independent Component Analysis: Algorithms and Applications*. *Neural Networks*, 13(4-5), 411-430, 2000.
- HYVARINEN, A., *Survey on Independent Component Analysis*. *Neural Computing Surveys*, 2, 94-128, 1999.
- HYVÄRINEN, A., *Gaussian Moments for Noisy Independent component analysis*. *IEEE Signal Processing Letters*, v. 6, n. 6, p. 145-147, 1999 (b).
- HYVÄRINEN, A., *Fast and Robust Fixed-Point Algorithms for Independent Component Analysis*. *IEEE Trans. on Neural Networks*, 10(3), p. 626-634, 1999(c).
- HYVÄRINEN, A., Oja, E., *A Fast Fixed Point Algorithm for Independent Component Analysis*. *Neural Computation*, 9(7), p.1483-1492, 1997.
- HYVÄRINEN, A., *One Unit Contrast Function for Independent Component Analysis: A Statistical Analysis*. Proc. IEEE Workshop on Neural Network for Signal Processing, p.388-397, Florida, 1997 (b).
- _____, *ICA and BSS Summer School*, Berlin, Jun. 2003. [*on line* http://www.first.fhg.de/~harmeli/summer_school/].
- _____, *5th International Conference on Independent Component Analysis and Blind Signal Separation (ICA'2004)*, Granada, Set. 22-24, 2004. [*on line* <http://atc.ugr.es/ica2004/>]
- _____, *4th International Conference on Independent Component Analysis and Blind Signal Separation (ICA'2003)*, Nara, Set. 2003. [*on line* <http://ica2003.jp>]
- INKI, M., Hyvärinen, A., *Two approach to estimation of overcomplete independent component bases*. Proc. of 2002 International Joint Conference on Neural Networks, IJCNN'02, vol. 1, pp. 454-459, 12-17 May, 2002.
- JUNG, T-P., Makeig, S., McKeown, M. J., Bell, A.J., Lee, T-W., Sejnowski, T. J., *Imaging Brain Dynamics Using Independent Component Analysis*. Proc. of IEEE, 89(7), p.1107-1122, Jul. 2001.
- JUNG, T-P., Makeig, S., Lee, T-W., McKeown, M. J., Bell, A.J., Sejnowski, T. J., *Independent Component Analysis of Biomedical Signals*. In *Second International Workshop on Independent Component Analysis and Signal Separation*, p. 633-644, Helsinki, 2000.
- JUTTEN, C., Héroult, J., *Blind Separation of Sources: An Adaptive Algorithm based on Neuromimetic Architecture*. *Signal Processing*, v.24, p.1-10, 1991.

- KARHUNEN, J., Pajunen, P., Oja, E., The Nonlinear PCA Criterion in Blind Source Separation: Relations with Other Approaches. *Neurocomputing*, v. 22, p.5-20, Nov. 1998.
- KARHUNEN, J., Oja, E., Wang, L., Koutsensalo, J., A Class of Neural Networks for Independent Component Analysis. *IEEE Trans. on Neural Networks*, v. 8, n.3, p.486-504, May 1997.
- KARVANEN, J., *Adaptive Methods for Score Function Modeling in Blind Source Separation*. Helsinki University of Technology, Signal Processing Laboratory, Report n. 35, Espoo, Finland, 2002.
- KNAAK, M. F., Filbert, D., *Acoustic Machine Monitoring Using Blind Source Separation*. In IV International Conference on Acoustical Vibratory Surveillance Methods and Diagnostics Techniques, França, Set. 2001. [on line, <http://mt.tu-berlin.de/~knaak/frankreich.pdf>; capturado em 21/set/2003].
- KOFIDIS, E., *Blind Source Separation: Fundamentals and Recent Advances*. XIX Simpósio Brasileiro de Telecomunicações (SBrT2001). Mini-curso. Fortaleza, CE, Set. 2001.
- LEE, T-W., Girolami, M., Bell, A. J., Sejnowski, T. J., A Unifying Information-Theoretic Framework for Independent Component Analysis. *Computers & Mathematics with Applications*, 31(11), p.1-21, Mar. 2000.
- LEE, T-W., Girolami, M., Sejnowski, T. J., Independent Component Analysis Using an Extended Infomax Algorithm for Mixed Sub-Gaussian and Super-Gaussian Sources. *Neural Computation*, 11(2), p. 417-441, 1999.
- LEE, T-W., Sejnowski, T., *Independent Component Analysis for Sub-Gaussian and Super-Gaussian Mixture*. In IV Joint Symposium on Neural Computation, v. 7, p.132-140, Institute for Neural Computation, 1997.
- LEWICKI, M., Sejnowski, T. J., Learning overcomplete representations. *Neural Computation*, 12:337-365, 2000.
- MANSUR, A., Barros, A. K., Ohnishi, N., *Blind Separation of Sources: Methods, Assumptions and Applications*, Special Section on Digital Signal Processing, IEICE Trans on Fundamentals, v. E83-A, n.8, Ago. 2000.
- MANSUR, A., Puntonet, C. G., Ohnishi, N., *A Simple ICA Algorithm Based on Geometrical Approach*. International Symposium on Signal Processing and its Applications (ISSPA), Kuala Lumpur, Malasia, Ago. 2001.
- MANSUR, A., Barros, A. K., Ohnishi, N., *Comparison Among Three Estimators for High Order Statistics*. 5th International Conference on Neural Information Processing, p.899-902, Kitakyushu, Japan, Oct. 1998.
- MAKEIG, S., Jung, T-P., Bell, A.J., Ghahremani, D., Sejnowski, T. J., *Blind Separation of Auditory Event-Related Brain Responses into Independent Component Analysis*. *Proc. Natl. Acad. Sci. USA*, v.99, p.10979-10984, Set. 1997.
- MAKEIG, S., Bell, A.J., Jung, T-P., Sejnowski, T. J., Independent Component Analysis of Electroencephalographic Data. In *Advances in Neural Information Processing Systems*, n.8, p.145-151, MIT Press, Cambridge, MA, 1996.

- MUTIHAC, R., Hulle, M. M. V., A Comparative Survey on Adaptive Neural Network Algorithms for Independent Component Analysis. *Romanian Reports in Physics*, v. 55, n. 1, p. 43-67, Romania, 2003.
- NANDI, A. K., Jack, L. B., Zarzoso, V., *Non-linear Non-Gaussian Signal Processing. Proc. of Non-linear Non-Gaussian Signal Processing – N2SP*, Invited Paper, UK, Jul. 2002. [on line <http://www.iee.org/oncomms/pn/signalprocessing/n2sp/workshop.cfm>; capturado em 04/jul/2003].
- NEVES, A. de O, *Uma Abordagem Unificada para Algoritmos de Equalização Autodidata*. Dissertação de Mestrado, FEE/DECOM/UNICAMP, Campinas, 2001.
- PAJUNEN, P., Karhunen, J., Least Square Methods for Blind Source Separation Based on Nonlinear PCA. In *International Journal of Neural Systems*, v. 8, n. 5/6, p.601-612, Oct./Dec. 1998.
- PAPOULIS, A., Probability, Random Variables and Stochastic Process. Mc Graw Hill, 3ed., 1991.
- PARK, H-M, Jung, H-Y., Lee, T-W., Lee, S-Y., Subband-based blind signal separation for noisy speech recognition. *Electronics Letters*, 35(23):2011-2012, Nov. 1999.
- PARRA, L., *Tutorial on Blind Source Separation and Independent Component Analysis*. Fev. 2002. [on line <http://newton.bme.columbia.edu/~lparra/publish/tutorial-bss.pdf>; capturado em 10/maio/2003].
- PHAM, D. T., Cardoso, J-F., *Blind Separation of Instantaneous Mixtures of Nonstationary Sources*. Proc. of International Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000), p. 187-193, Helsinki, 2000.
- POPE, K., Bogner, R., Blind Signal Separation I: Linear, Instantaneous Combinations, *Digital Signal Processing*, 1(6), p.5-16, 1996.
- RAJU, K., Ristaniemi, T., *ICA-RAKE Switching for Jammer Cancellation in DS-CDMA Array Systems*. Proc. of the 2002 IEEE Int. Symp. on Spread Spectrum Techniques and Applications (ISSSTA 2002), p.638-642, Prague, Czech Republic, Set. 2-5, 2002.
- RISTANIEMI, T., Raju, K., Karhunen, J., Oja, E., *Jammer Cancellation in DS-CDMA Array Systems: Pre and Post Switching of ICA and RAKE*. in: Proc. of the 2002 IEEE Int. Symposium on Neural Networks for Signal Processing (NNSP), p. 495-504, Martigny, Switzerland, Set. 4-6, 2002.
- RISTANIEMI, T., Raju, K., Karhunen, J., *Jammer Mitigation in DS-CDMA Using Independent Component Analysis*. Proc. of the IEEE Int. Conference on Communications (ICC 2002), New York City, NY, USA, Abr. 28-Maio 2, 2002.
- SANCHÉZ, V. D., Frontiers of Research in BSS/ICA. *Neurocomputing*, n.49, p.7-23, 2003.
- SCHOBEN, D., Torrkola, K., Smaragdis, P., *Evaluation of Blind Signal Separation Methods*. Proc. of International Workshop on Independent Component Analysis and Blind Signal Separation, Assois, France, Jan. 1999.

- SCHELL, S. V., Gardner, W. A., *Maximum Likelihood and Common Factor Analysis Based Blind Adaptive Spatial Filtering for Cyclostationary Signals*. In Proc. of IEEE ICASSP, v. 4, p. 292-295, 1992.
- SMARAGDIS, P. J., *Information Theoretic Approaches to Source Separation*. MsC Thesis, MIT, Jun. 1997.
- TALEB, A., Jutten, C., Source separation in post-nonlinear mixture. IEEE Trans. on Signal Processing, 47(10):2807-2820, 1999.
- TALWAR, S., Viberg, M., Paulraj, A., Blind Estimation of Synchronous Co-channel Digital Signal Using an Antenna Array. Part I: Algorithms. IEEE Trans. on Signal Processing, 44(5), p. 1184-1197, Maio, 1996.
- TONG, L., Huang, Y. F., Liu, R., Waveform-preserving Blind Estimation of Multiple Independent Sources. IEEE Trans. on Signal Processing, 41(7), p. 2461-2470, Jul. 1993.
- TONG, L., Soon, V. C., Liu, R., Huang, Y., Indeterminacy and Identifiability of Blind Identification. IEEE Trans. on Circuits and Systems, 38(5), p. 499-509, May 1991.
- TONG, L., Soon, V. C., Huang, Y. F., Liu, R., *AMUSE: A New Blind Identification Algorithm*. Proc. IEEE ISCAS, v. 3, p. 1784-1787, New Orleans, LA, 1990.
- TORKKOLA, K., *Blind Separation of Audio Signals – Are We There Yet?* Proc. of The First International Workshop on Independent Component Analysis and Signal Processing, ICA'99, p. 239-244, 1999.
- TORKKOLA, K., *Blind Separation of Delayed Sources Based on Information Maximization*. Proc. of IEEE International Conference Acoustic Speech, Signal Processing (ICASSP), v. 4, p. 3509-3513, NY, 1996.
- VIGÁRIO, R., Särelä, J., Jousmäki, V., Hämäläinen, M., Oja, E., Independent Component Approach to the Analysis of EEG and MEG Recordings. IEEE Trans. on Biomedical Engineering, 47(5), p. 589-593, 2000.
- VIGLIANO, D., Uncini, A., Flexible ICA solution for nonlinear blind source separation problem. *Electronics Letters*, 39(22):1616-1617, Oct., 2003.
- YPMA, A., Leshem, A., Duin, R. P. W., Blind Separation of Rotating Machine Sources: Bilinear Forms and Convolutional Mixtures. *Neurocomputing*, n. 49, p. 349-368, 2002.
- YPMA, A., *Learning Methods for Machine Vibration Analysis and Health Monitoring*. PhD Dissertation, Pattern Recognition Group, Dept. of Applied Physics, Delft University of Technology, Nov. 2001.
- WEISSTEIN, E. W., *Eric Weisstein's World of Mathematics*. [on line <http://mathworld.wolfram.com/2000>]; capturado em 30/maio/2003].
- WESTNER, A. G., *Object-Based Audio Capture: Separating Acoustically Mixed Sounds*. MsC Thesis in Media Arts and Sciences, MIT, Fev. 1999.