

SUPRESSÃO DE RUÍDO, DETECÇÃO E CLASSIFICAÇÃO DE SINAIS DE  
DESCARGAS PARCIAIS EM TRANSFORMADORES DE POTÊNCIA

Filipe Castello da Costa Beltrão Diniz

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS  
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE  
FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM  
CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

---

Prof. Sergio Lima Netto, Ph.D.

---

Prof. Luiz Wagner Pereira Biscaíno, D.Sc.

---

Eng. Marco Antonio Macciola Rodrigues, D.Sc.

---

Prof. Jacques Szczupak, Ph.D.

RIO DE JANEIRO, RJ - BRASIL  
FEVEREIRO DE 2005

DINIZ, FILIPE CASTELLO DA COSTA BELTRÃO

Supressão de Ruído, Detecção e  
Classificação de Sinais de Descargas  
Parciais em Transformadores de Potência  
[Rio de Janeiro] 2005

XVIII, 155 pp., 29,7 cm (COPPE/UFRJ,  
M.Sc., Engenharia Elétrica, 2005)

Tese - Universidade Federal do Rio de  
Janeiro, COPPE

1.Descargas parciais 2.Supressão  
de ruído 3.Detecção de sinais  
4.Classificação de padrões

I.COPPE/UFRJ      II.Título (série)

## Agradecimentos

Agradeço a todas as pessoas que direta ou indiretamente contribuíram para que este trabalho tenha sido concluído:

- À minha família que sempre apoiou minha idéia de apostar em um Mestrado;
- Ao professor Sergio Lima Netto, por ter se disposto a enfrentar o desafio de uma área nova e ter tanto me orientado técnica e academicamente quanto também me apoiado com conversas bastante incentivadoras;
- Aos professores Luiz Wagner Pereira Biscainho, Jacques Szczupak e ao engenheiro Marco Antonio Macciola Rodrigues por se disporem a compor a banca avaliadora;
- Ao pessoal do Cepel e do Laboratório de Diagnóstico de Equipamentos Elétricos, principalmente Helvio Martins, Alexandre Neves, Roberto Menezes, Marlon, Leonardo e Tiago, por me fornecer o tema, a bolsa, o equipamento necessário e um ambiente favorável durante o processo de desenvolvimento deste trabalho;
- Aos grandes amigos e amigas do Laboratório de Processamento de Sinais (LPS), do Laboratório de Processamento Analógico e Digital de Sinais (PADS), da Coppe, do grupo CGTotal e da turma de Engenharia Eletrônica e de Computação da UFRJ: alunos, professores e funcionários. Vocês foram essenciais a cada passo deste projeto e nada disso seria minimamente possível se apenas um de vocês não estivesse presente. Vocês são mais importantes do que podem imaginar;
- A todos os meus amigos extra-trabalho que sempre estiveram a meu lado e me possibilitaram desligar um pouquinho da constante pressão.

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

## SUPRESSÃO DE RUÍDO, DETECÇÃO E CLASSIFICAÇÃO DE SINAIS DE DESCARGAS PARCIAIS EM TRANSFORMADORES DE POTÊNCIA

Filipe Castello da Costa Beltrão Diniz

Fevereiro/2005

Orientador: Sergio Lima Netto

Programa: Engenharia Elétrica

Uma das técnicas atualmente mais utilizadas para se fazer um diagnóstico sobre algum equipamento elétrico é a análise de sinais de descargas parciais (DP). Entretanto, executar medições em equipamentos em operação ou que estejam cercados de outros equipamentos e dispositivos em operação é extremamente dificultado pela grande quantidade de ruído e interferências (tanto conduzidas quanto induzidas). Por isso, torna-se necessário o emprego de técnicas de supressão de ruído de forma a tornar possível a detecção desses sinais. Tendo sido feita a detecção dos sinais de DP, pode-se construir um mapa que descreve o comportamento desse fenômeno em termos da fase em que ocorre, com que amplitude e com que taxa de repetição. Dependendo do tipo de defeito presente no equipamento, há um padrão no mapa, que pode ser classificado de maneira a permitir que se saiba o tipo de defeito e, conseqüentemente, que se providencie medidas adequadas. Esse trabalho visa a desenvolver um sistema que possa detectar sinais de descargas parciais em meio a grande quantidade de ruído de forma a se classificar seu padrão de comportamento.

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

DENOISING, DETECTION AND CLASSIFICATION OF PARTIAL  
DISCHARGE SIGNALS IN POWER TRANSFORMERS.

Filipe Castello da Costa Beltrão Diniz

February/2005

Advisor: Sergio Lima Netto

Department: Electrical Engineering

Currently, one of the techniques more used to execute a diagnosis on some electric equipment is the analysis of partial discharges (PD) signals. However, to perform measurements in equipment under operation or that is surrounded by other equipment and devices under operation is extremely difficult due to the great amount of noise and interference (conducted or induced). Therefore, the use of denoising techniques becomes necessary to make the detection of these signals possible. Once the detection of the PD signals is done, a map that describes the behavior of this phenomenon in terms of occurrence phase, amplitude, and repetition rate can be constructed. Depending on the type of defect in the equipment, it has a distinct pattern in the map. Such map can be classified allowing the type of defect to be known and the proper actions to take place. This work aims at developing a system that can detect partial discharge signals buried in great amount of noise to classify its behavior pattern.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Organização da dissertação . . . . .	2
<b>2</b>	<b>Descargas Parciais</b>	<b>4</b>
2.1	Introdução . . . . .	4
2.2	Conceitos básicos . . . . .	4
2.3	Relação entre descargas parciais e a vida útil dos materiais dielétricos	6
2.4	Técnica elétrica de detecção de descargas parciais . . . . .	7
2.5	Instrumentos digitais de medição . . . . .	8
2.5.1	ICM <i>System</i> . . . . .	8
2.5.2	Osciloscópio digital . . . . .	9
2.6	Modelo matemático . . . . .	12
2.7	Ruído presente no sinal de descargas parciais . . . . .	14
2.8	Conclusão . . . . .	15
<b>3</b>	<b>Métodos de Supressão de Ruído</b>	<b>16</b>
3.1	Introdução . . . . .	16
3.2	Subtração espectral . . . . .	17
3.2.1	Algoritmo do método . . . . .	18
3.3	Filtro de Wiener . . . . .	21
3.3.1	Algoritmo do método . . . . .	24
3.4	Transformada <i>Wavelet</i> . . . . .	25
3.4.1	A STFT . . . . .	25
3.4.2	<i>Wavelet shrinkage</i> . . . . .	31

3.4.3	Método heurístico . . . . .	32
3.4.4	Método <i>VISU Shrink</i> ou Minimax . . . . .	32
3.4.5	Método <i>SURE Shrink</i> . . . . .	33
3.4.6	Método híbrido . . . . .	35
3.5	Filtragem adaptativa . . . . .	35
3.5.1	Algoritmo <i>Least-Mean Squares</i> (LMS) . . . . .	38
3.5.2	Algoritmo <i>Normalized Least-Mean Squares</i> (NLMS) . . . . .	40
3.5.3	Algoritmo <i>Recursive Least Squares</i> (RLS) . . . . .	42
3.5.4	Algoritmo <i>Set-Membership Affine Projection</i> . . . . .	45
3.6	Análise de componentes independentes . . . . .	47
3.6.1	Curtose . . . . .	50
3.6.2	Negentropia . . . . .	51
3.6.3	Aproximações para negentropia . . . . .	53
3.6.4	Pré-processamento . . . . .	53
3.6.5	Algoritmo de <i>FastICA</i> . . . . .	54
3.6.6	<i>Blind source separation</i> (BSS) . . . . .	56
3.7	Conclusão . . . . .	57
<b>4</b>	<b>Resultados</b>	<b>58</b>
4.1	Introdução . . . . .	58
4.2	Critérios de avaliação . . . . .	58
4.2.1	SNR . . . . .	59
4.2.2	Correlação cruzada . . . . .	59
4.2.3	Taxa de redução de amplitude . . . . .	59
4.3	Geração e aquisição de sinais . . . . .	59
4.4	Resultados da filtragem de Wiener . . . . .	62
4.4.1	Ajuste dos parâmetros . . . . .	62
4.4.2	Testes de desempenho . . . . .	65
4.5	Resultados da transformada <i>wavelet</i> . . . . .	68
4.5.1	Ajuste dos parâmetros . . . . .	68
4.5.2	Testes de desempenho . . . . .	71
4.6	Resultados da filtragem adaptativa . . . . .	75
4.7	Resultados da separação cega por ICA . . . . .	83

4.8	Resultados da combinação de filtragem de Wiener com transformada <i>wavelet</i> . . . . .	83
4.9	Conclusão . . . . .	87
<b>5</b>	<b>Detecção de Sinais</b>	<b>89</b>
5.1	Introdução . . . . .	89
5.2	Teoria de decisão e teste de hipótese . . . . .	89
5.3	Filtro casado . . . . .	92
5.4	Filtro casado adaptado ao caso das DP's . . . . .	93
5.4.1	Determinação do limiar de detecção . . . . .	94
5.4.2	Algoritmo do método . . . . .	95
5.5	Geração de mapas . . . . .	98
5.6	Resultados da detecção . . . . .	100
5.6.1	Detecção com pré-processamento feito por filtro de Wiener . .	100
5.6.2	Detecção com pré-processamento feito por transformada <i>wavelet</i>	101
5.6.3	Detecção com pré-processamento feito por filtro adaptativo . .	101
5.6.4	Detecção com pré-processamento feito por combinação de filtro de Wiener com transformada <i>wavelet</i> . . . . .	110
5.7	Conclusão . . . . .	113
<b>6</b>	<b>Classificação Neural de Padrões de Descargas Parciais</b>	<b>114</b>
6.1	Introdução . . . . .	114
6.2	Aquisição de sinais . . . . .	114
6.3	Componentes principais não lineares . . . . .	116
6.3.1	Extração de componentes . . . . .	117
6.4	Resultados da extração de componentes . . . . .	119
6.4.1	Dimensionamento da rede . . . . .	119
6.4.2	Configuração dos testes . . . . .	120
6.4.3	Resultados . . . . .	121
6.5	Resultados quanto à classificação . . . . .	125
6.6	Conclusão . . . . .	126
<b>7</b>	<b>Implementação do Sistema</b>	<b>129</b>
7.1	Introdução . . . . .	129



7.2	Visão geral do sistema . . . . .	129
7.3	<i>Software</i> desenvolvido . . . . .	130
7.3.1	Tela principal . . . . .	130
7.3.2	Ajuste de parâmetros para métodos de supressão de ruído . . .	132
7.3.3	Ajuste de parâmetros para detecção . . . . .	142
7.3.4	Avaliação do sistema . . . . .	145
7.4	Conclusão . . . . .	147
<b>8</b>	<b>Conclusão</b>	<b>148</b>
8.1	Resultados . . . . .	148
8.2	Trabalhos Futuros . . . . .	149
	<b>Referências Bibliográficas</b>	<b>151</b>

# Lista de Figuras

2.1	Distribuição $(\phi, q, N)$ correspondente a descargas parciais em um transformador de potência de 53,33 MVA, 345/15 KV [1]. . . . .	10
2.2	Característica $(t, i)$ de uma medição de DP's em célula de ensaio ponta plano com ar ambiente, como visto em [1]. . . . .	11
2.3	Característica $(t, i)$ de várias medições de DP's em célula de ensaio ponta plano com óleo isolante, como visto em [1]. . . . .	11
2.4	Pulso típico de DP juntamente com os parâmetros que o descrevem. . . . .	13
2.5	(a) Pulso típico do tipo DEP; (b) Pulso típico do tipo DOP. . . . .	14
2.6	Comparação entre o modelo matemático e o sinal medido. . . . .	15
3.1	(a) Sinal original. (b) A DFT convencional do sinal original. (c) Gráfico gerado pela STFT: o eixo X (horizontal) mostra os índices das janelas, o eixo Y (vertical) mostra as frequências sobre as quais foram calculados os espectros e os tons de cinza indicam os módulos de cada componente frequencial numa determinada janela. . . . .	26
3.2	Interpretação da STFT como um banco de filtros. . . . .	27
3.3	Comparação da ação de janelas de comprimentos iguais para sinais de diferentes frequências. . . . .	27
3.4	Banco de filtros usado na transformada <i>wavelet</i> (larguras das bandas diferentes). . . . .	29
3.5	(a) Estrutura de árvore binária com 3 níveis. (b) Sistema equivalente de 4 canais. . . . .	30
3.6	Configuração de filtragem adaptativa para supressão de ruído. . . . .	36

4.1	(a) Sinal medido pelo osciloscópio digital TDS460, com 120.000 pontos numa frequência de amostragem de 5 MHz. (b) Detalhe do sinal mostrado em (a). . . . .	61
4.2	(a) Trem de pulsos gerado por 5 cópias do sinal de DP. (b) Detalhe do sinal mostrado em (a). . . . .	61
4.3	Correlação cruzada e taxa de redução de amplitude para as famílias Daubechies e Symlets. . . . .	71
4.4	Curvas de convergência do algoritmo LMS variando-se o valor de $\mu$ . . . . .	79
4.5	Curvas de convergência para os algoritmos estudados quando se submete o sinal corrompido pela primeira vez ao sistema de supressão adaptativa de ruído: (a) LMS; (b) NLMS; (c) RLS; (d) SMAP . . . . .	82
4.6	Curvas de convergência para os algoritmos estudados quando se submete o sinal corrompido pela segunda vez ao sistema de supressão adaptativa de ruído: (a) LMS; (b) NLMS; (c) RLS; (d) SMAP. . . . .	82
4.7	Sinais de misturas usados para a separação cega baseada em ICA. . . . .	84
4.8	Resultados da separação cega baseada em ICA. . . . .	84
4.9	Supressão de ruído utilizando conjuntamente filtro de Wiener e transformada <i>wavelet</i> : (a) Sinal corrompido com sinal ideal concentrado na amostra 60000; (b) Sinal resultante da supressão de ruído empregando filtro de Wiener; (c) A seguir, emprega-se a transformada <i>wavelet</i> ; (d) A seguir, ajustam-se os limiars manualmente. . . . .	86
5.1	Configuração básica para o uso de filtros casados. . . . .	93
5.2	As duas pdf's condicionais: a da esquerda corresponde à resposta do filtro casado aos eventos que são constituídos apenas de ruído e a da direita de eventos de sinal corrompido por ruído. . . . .	95
5.3	Curva ROC referente a Figura 5.2 . . . . .	96
5.4	(a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem de Wiener e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 60 dB. . . . .	102

5.5	(a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem de Wiener e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 40 dB. . . . .	103
5.6	(a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem de Wiener e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB. . . . .	104
5.7	(a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por transformada <i>wavelet</i> e o padrão de DP deslocado. Para ambos os casos, o ruído branco presente no sinal corrompido apresentava a mesma amplitude do sinal de interesse. . . . .	105
5.8	(a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem adaptativa (algoritmo LMS) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB. .	106
5.9	(a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem adaptativa (algoritmo NLMS) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB. .	107
5.10	(a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem adaptativa (algoritmo RLS) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB. .	108

5.11	(a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem adaptativa (algoritmo <i>set-membership affine projection</i> ) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB. . . . .	109
5.12	(a) Função de correlação cruzada entre o sinal processado pelo filtro de Wiener e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído (formado pela filtragem de Wiener em série com a transformada <i>wavelet</i> ) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 60 dB. . . . .	111
5.13	(a) Função de correlação cruzada entre o sinal processado pelo filtro de Wiener e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído (formado pela filtragem de Wiener em série com a transformada <i>wavelet</i> ) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB. . . . .	112
6.1	Aparência do mapa gerado para os sinais de DP [2]. . . . .	115
6.2	Redes que implementam as funções de codificação e de decodificação. . . . .	118
6.3	Rede auto-associativa usada para a extração de componentes principais não-lineares. . . . .	118
6.4	Correlação entre cada evento e sua versão reconstruída por PCA e NLPCA. . . . .	124
6.5	Matriz de histogramas que verifica a real separação das classes. . . . .	127
7.1	Tela principal do programa CleanDetect, o qual tem as funções de controlar os métodos de supressão de ruído, detecção e classificação e mostrar os resultados. . . . .	131
7.2	Caixa de diálogo para ajuste dos parâmetros da subtração espectral. . . . .	134
7.3	Interface para visualização dos efeitos da variação dos parâmetros sobre a subtração espectral e do filtro de Wiener. . . . .	136

7.4	Caixa de diálogo para ajuste dos parâmetros da transformada <i>wavelet</i> .	137
7.5	Interface para visualização dos efeitos da variação dos parâmetros sobre a supressão de ruído com base em <i>wavelet shrinkage</i> . . . . .	139
7.6	Caixa de diálogo para ajuste dos parâmetros da filtragem adaptativa.	140
7.7	Interface para visualização dos efeitos da variação dos parâmetros sobre a filtragem adaptativa . . . . .	141
7.8	Interface gráfica que implementa o algoritmo de FastICA. . . . .	143
7.9	Interface gráfica que auxilia no ajuste dos parâmetros do filtro casado.	146

# Lista de Tabelas

3.1	Vantagens e desvantagens dos métodos de supressão de ruído estudados.	57
4.1	Comportamento da filtragem de Wiener ao tratar ruído branco apresentando uma janela para estimativa do espectro do sinal com 10000 amostras. A SNR original (do sinal corrompido) é de -21,105 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,0879. . . . .	64
4.2	Comportamento da filtragem de Wiener ao tratar ruído branco apresentando uma janela para estimativa do espectro do sinal com 10000 amostras, forçando o espectro do ruído a ser constante. A SNR original (do sinal corrompido) é de -21,105 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,0879. . . . .	64
4.3	Comportamento da filtragem de Wiener ao tratar DSI apresentando uma janela para estimativa do espectro do sinal com 10000 amostras A SNR original (do sinal corrompido) é de -46,105 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,009736. . . . .	66
4.4	Estudo mais detalhado do comportamento da filtragem de Wiener ao tratar DSI apresentando uma janela para estimativa do ruído com 20000 amostras. A SNR original (do sinal corrompido) é de -46,6424 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,009736.	67
4.5	Comportamento da filtragem de Wiener ao tratar DSI contaminado com ruído branco apresentando uma janela para estimativa do ruído com 20000 amostras. . . . .	69
4.6	Comportamento da filtragem de Wiener ao tratar DSI contaminado com ruído branco apresentando uma janela para estimativa do ruído com 20000 amostras. INR = 20 dB. . . . .	69
4.7	Comparação entre filtragem de Wiener e subtração espectral. . . . .	70

4.8	Comportamento da transformada <i>wavelet</i> ao tratar ruído branco usando abordagem heurística. . . . .	73
4.9	Comportamento da transformada <i>wavelet</i> ao tratar ruído branco usando <i>visu shrink</i> com <i>hard thresholding</i> . . . . .	73
4.10	Comportamento da transformada <i>wavelet</i> ao tratar ruído branco usando <i>visu shrink</i> com <i>soft thresholding</i> . . . . .	74
4.11	Comportamento da transformada <i>wavelet</i> ao tratar ruído branco usando <i>sure shrink</i> com <i>hard thresholding</i> . . . . .	74
4.12	Comportamento da transformada <i>wavelet</i> ao tratar DSI contanimada por ruído branco usando abordagem heurística. . . . .	76
4.13	Comportamento da transformada <i>wavelet</i> ao tratar DSI contanimada por ruído branco usando <i>visu shrink</i> com <i>hard thresholding</i> . . . . .	76
4.14	Comportamento da transformada <i>wavelet</i> ao tratar DSI contanimada por ruído branco usando <i>visu shrink</i> com <i>soft thresholding</i> . . . . .	76
4.15	Comportamento da transformada <i>wavelet</i> ao tratar DSI contanimada por ruído branco usando <i>sure shrink</i> com <i>hard thresholding</i> . . . . .	77
4.16	Comportamento da filtragem adaptativa ao tratar DSI (INR = 20 dB) e ruído branco de mesma amplitude que sinal de interesse usando algoritmo LMS. A SNR original (do sinal corrompido) é de -46,79 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,00957. . . . .	79
4.17	Comportamento da filtragem adaptativa ao tratar DSI (INR = 20 dB) e ruído branco de mesma amplitude que sinal de interesse usando algoritmo NLMS. A SNR original (do sinal corrompido) é de -46,79 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,00957. . . . .	80
4.18	Comportamento da filtragem adaptativa ao tratar DSI (INR = 20 dB) e ruído branco de mesma amplitude que sinal de interesse usando algoritmo RLS. A SNR original (do sinal corrompido) é de -46,79 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,00957. . . . .	80



4.19	Comportamento da filtragem adaptativa ao tratar DSI (INR = 20 dB) e ruído branco de mesma amplitude que sinal de interesse usando algoritmo <i>Set-membership Affine Projection</i> . A SNR original (do sinal corrompido) é de -46,79 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,00957. . . . .	81
4.20	Comportamento da supressão de ruído empregando-se filtro de Wiener e transformada <i>wavelet</i> . A SNR original (do sinal corrompido) é de -46,6425 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,00972 . . . . .	87
5.1	Possibilidades quanto à forma com a qual o sistema de detecção irá responder a um dado sinal de entrada. . . . .	90
6.1	Energia acumulada para 5 diferentes combinações do conjunto de dados usando-se 10 componentes principais lineares. . . . .	122
6.2	Acúmulo de energia (informação) para PCA e NLPCA, variando-se o número de componentes. . . . .	123
6.3	Médias geométricas das correlações entre cada evento e sua versão reconstruída. . . . .	123
6.4	Comparação dos resultados de eficiência de classificação obtidos na prospecção de dados (PD) e por validação cruzada (VC). Em todos os casos, usou-se o conjunto de teste. . . . .	125
6.5	Comparação da eficiência de classificação usando-se as técnicas de PCA e de NLPCA. . . . .	126

# Lista de Algoritmos

3.1	Estimação do espectro de densidade de potência do ruído. . . . .	19
3.2	Procedimentos para a subtração espectral. . . . .	20
3.3	Procedimentos para a filtragem de Wiener. . . . .	24
3.4	Procedimentos para o LMS. . . . .	39
3.5	Procedimentos para o NLMS. . . . .	41
3.6	Procedimentos para o RLS. . . . .	44
5.1	Procedimentos para a detecção baseada em filtros casados. . . . .	97
5.2	Procedimentos para geração do mapa segundo o ICM System. . . . .	99

# Capítulo 1

## Introdução

### 1.1 Motivação

Atualmente, as empresas responsáveis por prover energia à população têm-se visto obrigadas a colocar seus equipamentos sob um pesado regime de funcionamento (muitas vezes com sobrecarga), sendo que os mesmos já vêm sofrendo os efeitos a isto relacionados. Uma das conseqüências é a perda das propriedades elétricas e mecânicas por parte dos materiais isolantes. Tudo isso forma o cenário adequado para a intensificação do fenômeno conhecido como descargas parciais [3].

As descargas parciais ou, simplesmente, DP são sucessões de descargas elétricas incompletas, rápidas e intermitentes na ordem de nanossegundos. Estas ocorrem pela proximidade entre duas partes energizadas e um meio isolante, pelo efeito de ionização em cavidades gasosas no interior dos materiais isolantes sólidos e líquidos.

Assim, para que se evitem tanto graves acidentes (como explosões) quanto desligamentos não programados, deve-se adotar alguma forma de monitoração e diagnóstico para equipamentos elétricos, como os transformadores de potência. Uma maneira de se conseguir isso é através de medições de sinais de DP, o que é considerada atualmente a técnica mais eficaz para avaliação de dielétricos em operação [4].

Contudo, para se medirem tais sinais, não se pode retirar o transformador da subestação e nem desligá-lo. Isso acarreta uma grande dificuldade, pois, nessas condições, o equipamento em questão estaria submetido a grande quantidade de ruído eletromagnético. Tal ruído vem tanto do equipamento sob medição quanto daqueles que o cercam (incluindo outros transformadores e linhas de transmissão),

além de todo tipo de interferências conduzidas e induzidas (como sinais de modulação AM). Dessa forma, torna-se necessário algum tipo de técnica que possa detectar o sinal apesar da imensa interferência a que está submetido.

Cada tipo de defeito resulta em um comportamento específico para os sinais de DP. A forma do sinal em si não se altera. O que é diferenciado é a fase na qual o sinal ocorre dentro do ciclo de potência, a sua amplitude e a taxa de repetição. Cada comportamento é característico de uma classe de DP. Classificando-se o tipo de DP presente em um determinado equipamento, pode-se ter uma idéia de que tipo de defeito está ocorrendo.

Esse trabalho tem como objetivo estudar e implementar um sistema de detecção de sinais de DP que possa atuar em ambientes bastante ruidosos e que seja capaz de, em seguida, classificar o comportamento do sinal. Dessa forma, o que se obtém é um sistema em três etapas: supressão de ruídos, detecção de sinais e classificação de padrões.

Devido ao fato de este ser um estudo de um problema bastante prático, o sistema será implementado em Matlab<sup>TM</sup>, de forma a apresentar uma interface amigável com o usuário. No futuro, todo o processamento pode e deve ser feito de forma automatizada, através do emprego de processadores específicos, como DSPs [5].

## 1.2 Organização da dissertação

No Capítulo 2, faz-se uma introdução ao conceito de descargas parciais, é estabelecida uma relação entre a vida útil dos equipamentos elétricos e as descargas parciais. Técnicas de medição de descargas parciais são comentadas, ressaltando-se o método elétrico. São vistos os instrumentos usados para medir e avaliar as descargas parciais. O modelo matemático deste tipo de sinal é visto e os ruídos presentes na descarga são comentados.

O Capítulo 3 tem como objetivo dar uma visão geral sobre várias técnicas de supressão de ruído. São apresentados métodos de subtração espectral, filtragem de Wiener, transformada *wavelet*, filtragem adaptativa e ICA (análise de componentes independentes).

O Capítulo 4 apresenta os resultados obtidos ao se empregar os métodos de supressão de ruído discutidos no Capítulo 3. São discutidos os critérios de avaliação utilizados para verificar a eficiência do sistema. É relatado o modo através do qual foi feita a aquisição dos sinais usados para obter tais resultados. São ainda feitas uma análise qualitativa sobre a combinação de métodos e uma comparação de todos os resultados.

No Capítulo 5, é feita a detecção das DP's utilizando-se um método baseado em filtros casados, e são mostrados os resultados deste método quando em conjunto com o que foi apresentado no Capítulo 3. É introduzido o conceito de detecção binária, e é vista a estrutura dos filtros casados. Este método adaptado para a detecção de DP's é explicado, assim como o modo de se determinar um limiar de detecção. Por fim, será mostrado como se gerar os mapas de acordo com o ICM System (assim como é explicado no Capítulo 2).

No Capítulo 6, propõe-se a implementação de um classificador neural que opere sobre os componentes principais não lineares de modo a indicar qual tipo de DP está ocorrendo e é feita uma análise do mesmo. É mostrada como foi feita a aquisição de eventos para treinamento do classificador e são apresentados os componentes principais não lineares. Será visto também como se faz a extração destes componentes. Por último, mostra-se como foi feita a classificação dos eventos de DP.

O Capítulo 7 tem como objetivo descrever o *software* desenvolvido para a execução das tarefas de supressão de ruído, detecção e classificação. Cada uma das partes será explicada, assim como a forma através da qual estas partes se integram. É dada uma idéia geral de como será o uso do sistema por um usuário comum. Também será descrito detalhadamente o *software* desenvolvido, mostrando-se todas as suas telas e etapas do processamento.

O Capítulo 8 discute os resultados obtidos pelo sistema como um todo, além de mencionar as contribuições do presente trabalho e propostas para trabalhos futuros.

# Capítulo 2

## Descargas Parciais

### 2.1 Introdução

O monitoramento de descargas parciais tem se firmado cada vez mais como um dos principais meios para se avaliar a “saúde” de um equipamento elétrico [4]. Neste capítulo, serão vistas as suas características básicas e algumas técnicas de medição associadas a este tipo de fenômeno. Esse capítulo tem um caráter mais didático a fim de auxiliar a quem não tiver uma maior familiaridade com o assunto. A Seção 2.2 faz uma introdução ao conceito de descargas parciais. Na Seção 2.3, é estabelecida uma relação entre a vida útil dos equipamentos elétricos e as descargas parciais. A Seção 2.4 comenta técnicas de medição de descargas parciais, ressaltando o método elétrico. Na Seção 2.5, são vistos os instrumentos usados para medir e avaliar as descargas parciais. O modelo matemático deste tipo de sinal é visto na Seção 2.6, e os ruídos presentes na descarga são comentados na Seção 2.7.

### 2.2 Conceitos básicos

Um material isolante ou dielétrico, quando submetido a grandes valores de tensão, pode permitir a ocorrência de descargas elétricas causadas por uma avalanche de elétrons ionizados. Se essas descargas ocorrem no ar e em torno de um elemento condutor, são denominadas de efeito corona (coroa em latim) [1].

Quando essas descargas não são visíveis e estão no interior de algum equipamento são denominadas descargas parciais (DP's), que podem ocorrer tanto nas

cavidades ou bolhas existentes no interior de um material dielétrico quanto em superfícies condutoras. O termo descargas parciais é utilizado porque as descargas são localizadas, ou seja, não chegam a percorrer todo o caminho dentro de um material isolante colocado entre dois condutores submetidos a uma diferença de potencial.

DP's são extremamente prejudiciais tanto para o material dielétrico como também para o equipamento que o envolve. Elas podem gerar calor, agentes oxidantes como ozônio, ondas eletromagnéticas interferindo nas recepções de rádio e deterioração do material dielétrico, sendo uma das principais causas de falhas no isolamento. De acordo com a intensidade das descargas, a vida útil do material pode ser reduzida. Por todas essas razões, pode-se entender por que DP's são indesejáveis. Logo, sua medição permite uma avaliação importante da qualidade e do desempenho do dielétrico dos equipamentos e instalações de alta tensão.

Segundo a norma ABNT NBR 6940 / 81 [4], a medição das descargas parciais é feita por três motivos:

- Para verificar se o material isolante, numa tensão especificada, não apresenta descargas parciais acima dos valores especificados.
- Para determinar os valores de tensão nos quais descargas de uma intensidade baixa especificada se iniciam com tensão crescente e cessam com tensão decrescente.
- Para verificar a magnitude das descargas parciais numa tensão especificada.

Existem várias técnicas de detecção de DP's: elétrica, acústica, óptica, química etc. Porém, as detecções não elétricas não fornecem medidas quantitativas satisfatórias e são usadas apenas para complementar a detecção elétrica, por exemplo localizando onde ocorreram as descargas no equipamento [6]. Este trabalho, portanto, enfatiza o processamento realizado junto à detecção elétrica, cuja técnica é a que mais cresceu em importância nos últimos anos e dispõe de procedimentos e normas mais completos.

A unidade mais utilizada para medir descargas parciais é o picocoulomb (pC). Isto porque a carga é proporcional à energia destrutiva liberada no local da descarga.

DP's podem ocorrer em qualquer ponto do material dielétrico, na junção de dois materiais dielétricos diferentes ou adjacentes ao eletrodo, seguidamente em

vários pontos do dielétrico ou em uma cavidade do material dielétrico. Além disso, DP's são de curta duração em relação ao período da senóide de tensão aplicada (da ordem de nanossegundos), são repetitivas e têm o tempo de subida muito curto, podendo ser consideradas como uma função impulso.

## **2.3 Relação entre descargas parciais e a vida útil dos materiais dielétricos**

A partir das medições das DP's, surge um problema importante: como interpretar os valores medidos com respeito à expectativa de vida dos objetos sob ensaio?

A relação qualitativa entre uma medição de descargas parciais e a vida de um material dielétrico ainda não é completamente conhecida, particularmente devido à grande diversidade dos equipamentos de alta tensão. As experiências práticas somente revelam que DP's durante longos períodos de tempo alteram a estrutura de um material dielétrico e podem causar a destruição de todo o equipamento sob alta tensão.

A estimativa da expectativa de vida baseada nas medições de DP's pode ser possível para equipamentos de alta tensão que tiveram o seu comportamento avaliado através de medições sistemáticas conduzidas durante vários anos. No entanto, esses resultados não podem ser convertidos para outros tipos de equipamentos. Atualmente, cada equipamento possui uma norma específica que informa qual é o seu nível máximo de DP admissível. Na maioria deles, pode-se considerar uma carga aparente máxima admissível em torno de 5 a 15 pC, sendo desejável que os equipamentos, na sua máxima tensão de operação, apresentem níveis de DP's menores que 10 pC e próximos dos ruídos ambientes dos sistemas de medição.

Os valores máximos de carga aparente podem variar devido às diferentes constituições dos equipamentos, aos tipos de dielétrico e à maneira como o material dielétrico é cuidado ao longo da sua vida útil. Por exemplo, equipamentos grandes, com grande quantidade de material dielétrico, admitem um nível máximo de DP maior que equipamentos pequenos, pois nestes, uma pequena descarga pode causar danos mais sérios. DP's em equipamentos com materiais dielétricos sólidos



geram conseqüências diferentes daquelas em equipamentos com materiais dielétricos líquidos ou gasosos, pois o dielétrico sólido não é auto-recuperável, o que implica necessidade de um nível máximo de descargas mais baixo.

## 2.4 Técnica elétrica de detecção de descargas parciais

O método de medição elétrico faz com que o instrumento de detecção de descargas se torne parte de um circuito elétrico, incluindo o equipamento onde as descargas estão ocorrendo. Esse procedimento de medição é o mais utilizado em cabos de alta tensão, capacitores e transformadores, entre outros. Uma impedância de detecção RLC (resistiva, capacitiva e indutiva) ou RC (resistiva e capacitiva) pode ser usada, sendo a RLC para um modo de detecção de banda estreita de freqüência e a RC para um modo de detecção de banda larga de freqüência [4].

Durante a medição de DP's podem ocorrer vários fatores que dificultam a sua detecção. As fontes de distúrbio podem interferir nos resultados quantitativos de DP's do objeto sob ensaio. As interferências estão classificadas em duas categorias: as que ocorrem quando o circuito de ensaio não está energizado e as que ocorrem quando o circuito está energizado, porém não ocorrem no objeto sob ensaio.

Quando o circuito de ensaio não está energizado, os distúrbios podem ser causados, por exemplo, por chaveamentos em outros circuitos, ensaios de alta tensão nas proximidades, transmissões de rádio e até mesmo por ruídos inerentes ao instrumento de medição.

Se o circuito de ensaio está energizado, os distúrbios geralmente crescem com o aumento do nível de tensão. Esses distúrbios podem ser provenientes de descargas parciais no transformador ou nos condutores de alta tensão. Também podem ocorrer distúrbios devido às falhas no aterramento de equipamentos nas proximidades e a harmônicos da fonte de tensão que estejam próximos à banda de freqüência utilizada na medição.

Portanto, para que as medições de descargas parciais forneçam resultados satisfatórios, alguns cuidados especiais devem ser tomados [1]:

- As superfícies dos objetos sob ensaio devem ser limpas e secas, pois a umidade

ou partículas de pó nas superfícies isolantes podem causar DP's.

- As partes externas do objeto sob ensaio que possam gerar corona, tais como saliências pontiagudas, devem ser protegidas eletricamente por meio de toróides.
- O objeto sob ensaio deve estar à temperatura ambiente.
- Deve-se dar um intervalo de tempo mínimo entre um ensaio e outro, pois o resultado dos ensaios pode ser afetado pelas solicitações mecânicas, térmicas ou elétricas associadas.
- O local do ensaio deve ser isento de quaisquer perturbações e de transmissões radiofônicas que interfiram no resultado, se possível em recintos blindados.
- A fonte de alimentação deve ser isenta de quaisquer DP's indesejáveis.

Algumas dessas especificações não podem ser satisfeitas em situações reais, estando restritas ao ambiente de laboratório. Isso aumenta a necessidade de existir algum tipo de tratamento em relação a interferências.

## 2.5 Instrumentos digitais de medição

Os instrumentos digitais permitem uma gama maior de informações a respeito das descargas parciais. Estes são capazes de armazenar dados estatísticos das descargas parciais. A partir desses dados, é possível obter uma distribuição estatística das descargas parciais em função da sua posição relativa ao ângulo de fase da tensão aplicada e da sua amplitude, durante um período de tempo pré-selecionado.

Existem vários instrumentos digitais específicos para medição de DP's, entre eles: ICM System [2], TETEX, Robinson Instruments, SEFEMI etc.

### 2.5.1 ICM *System*

O ICM System, da Power Diagnostix [2], consiste em uma unidade de aquisição de sinais, pré-amplificadores externos e um computador portátil com interface IEEE 488 [7] (que é a conhecida interface GPIB - *General Purpose Interface Bus*).

As DP's medidas podem ser caracterizadas pela fase da tensão aplicada em que elas ocorrem ( $\phi$ ), amplitude dos pulsos ( $q$ ) e taxa de repetição ( $N$ ). A distribuição dos sinais medidos forma um mapa característico que se apresenta em duas dimensões, porém permite uma leitura em três dimensões através de um código de cores correspondente ao terceiro eixo.

Para compreender melhor como é feita a leitura de descargas parciais com o ICM System, pode-se observar a Figura 2.1, que mostra o resultado de uma medição de descargas parciais em um transformador.

Como pode ser visto no gráfico, o eixo horizontal corresponde à fase da tensão aplicada e o eixo vertical representa a amplitude da carga aparente medida. O terceiro eixo corresponde ao número de descargas medidas, identificadas através de um código de cores. As partes mais escuras significam um número maior de descargas e o cinza, um número menor. As partes mais claras representam uma menor ocorrência de descargas. A distribuição de cores entre os extremos permite uma visualização aproximada do número de descargas ocorridas em um período da tensão aplicada.

O ICM System realiza a aquisição dos sinais em um tempo determinado, geralmente de 60 s. O resultado da medição fica sendo apresentado na tela do computador intermitentemente, sendo acrescentadas mais descargas com o decorrer do tempo, alterando as cores do mapa.

Os mapas resultantes desse sistema de medição fornecem parâmetros importantes que podem ser utilizados em várias aplicações: reconhecimento de padrões de descargas, determinação de ruídos na medição etc. Porém, esses mapas não fornecem as características da evolução momentânea das descargas no decorrer do tempo em que elas ocorrem. Para obter informações mais completas, ou seja, como as descargas evoluem em vários períodos de aquisição e a sua forma sobreposta à senóide de tensão, é preciso realizar a medição através do osciloscópio digital.

## 2.5.2 Osciloscópio digital

O osciloscópio digital é inserido no circuito de medição da mesma maneira que o ICM System, com exceção dos pré-amplificadores, que não são necessários. Os sinais apresentados na tela do instrumento devem ser integrados no tempo para que

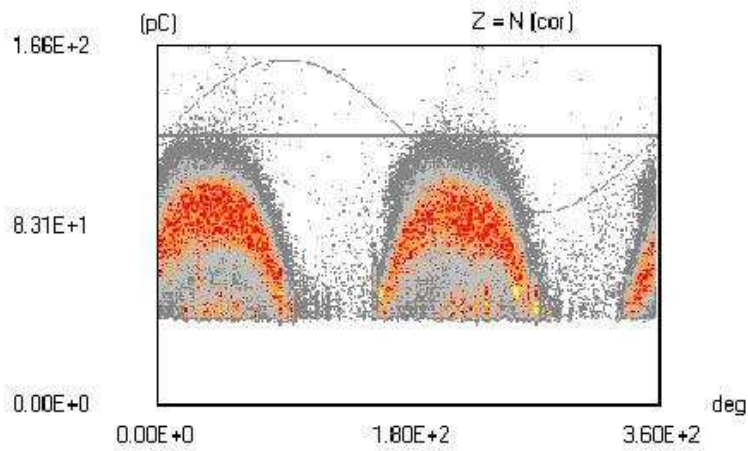


Figura 2.1: Distribuição  $(\phi, q, N)$  correspondente a descargas parciais em um transformador de potência de 53,33 MVA, 345/15 KV [1].

possam fornecer os valores dos pulsos da descarga equivalentes aos valores fornecidos pelo ICM System.

Para que seja possível a comunicação com o instrumento, um computador deve ser munido de uma placa GPIB e de um *software* de comunicação (escrito em LabView, Visual Basic, C etc.). Neste projeto, foi utilizado um programa construído em LabView no Cepel (Centro de Pesquisas de Energia Elétrica). Atualmente, com a facilidade de acesso aos *softwares* e *hardwares* disponíveis para realizar uma comunicação eficiente entre o computador e os instrumentos de aquisição, os instrumentos digitais tendem a ser cada vez mais utilizados.

Pode-se observar que o processamento do sinal medido através do osciloscópio digital é mais trabalhoso do que o do sinal lido diretamente no ICM System. Porém, com o sinal das descargas no decorrer do tempo, é possível fazer análises mais profundas sobre o processo de formação e a evolução das descargas no material dielétrico. Além disso, é possível executar técnicas de supressão de ruído e detecção de sinais.

Para análise do comportamento das descargas, foram realizadas medições com células de ensaio contendo materiais dielétricos diferentes, e os gráficos resultantes estão nas Figuras 2.2 e 2.3.

A Figura 2.2, por exemplo, mostra que as descargas medidas no ar ambiente têm um tempo de descida aproximadamente cem vezes maior que as descargas medidas no óleo (Figura 2.3). Esta informação não é obtida quando se realizam medições

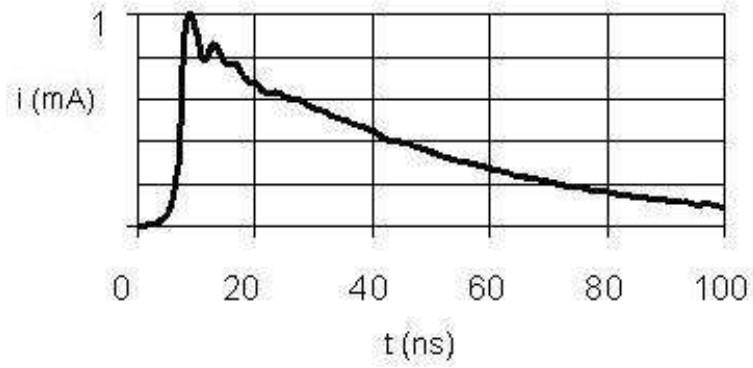


Figura 2.2: Característica  $(t, i)$  de uma medição de DP's em célula de ensaio ponta plano com ar ambiente, como visto em [1].

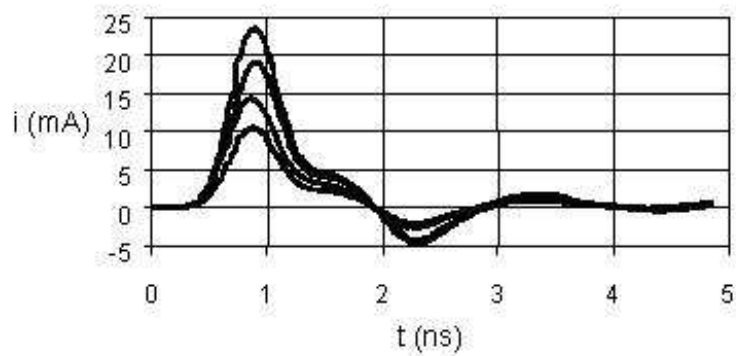


Figura 2.3: Característica  $(t, i)$  de várias medições de DP's em célula de ensaio ponta plano com óleo isolante, como visto em [1].

de DP's através dos outros sistemas de medição, sejam eles analógicos ou digitais.

As medições na célula de ensaio contendo óleo isolante foram realizadas várias vezes, uma após a outra. Alguns desses registros estão representados no gráfico da Figura 2.3. Isso foi feito para mostrar que um sinal de descarga pode apresentar amplitudes diferentes entre medições consecutivas.

Outra vantagem do osciloscópio digital é a sua sensibilidade. Várias medições foram realizadas nos laboratórios do Cepel, em Adrianópolis, onde o osciloscópio digital foi inserido juntamente com o ICM System no circuito de medição. Quando os sinais de descarga tinham uma amplitude muito pequena, o ICM System não apresentava nenhum resultado, enquanto que o osciloscópio digital mostrava o sinal da descarga. Isto é função do próprio circuito de ensaio e medição, no qual a sensibilidade é baixa.

## 2.6 Modelo matemático

A Figura 2.4 demonstra um típico pulso de DP [8], o qual pode ser caracterizado por uma série de parâmetros:

- $A$ : valor de pico do pulso.
- $t_r$ : tempo de subida do pulso (medido entre as amplitudes de 10% e 90% do valor de pico).
- $t_w$ : largura do pulso (medida entre as amplitudes de 50% do valor de pico).
- $t_d$ : tempo de descida do pulso (medido entre as amplitudes de 90% e 10% do valor de pico).

Na prática, sinais de tensão de DP's são medidos com o auxílio de um circuito de detecção. Dessa forma, os sinais detectados tendem a variar a sua forma de acordo com a configuração do circuito de detecção utilizado. Tal circuito pode ter, como já foi explicado, uma impedância do tipo RC ou RLC. As funções de transferência desses circuitos estão descritas nas equações (2.1) e (2.2), respectivamente.

$$G_{RC}(s) = \frac{1}{C} \frac{1}{s + \frac{1}{\tau}} \quad (2.1)$$

$$G_{RLC}(s) = \frac{1}{C} \frac{s}{s^2 + \frac{s}{\tau} + \omega_0^2}, \quad (2.2)$$

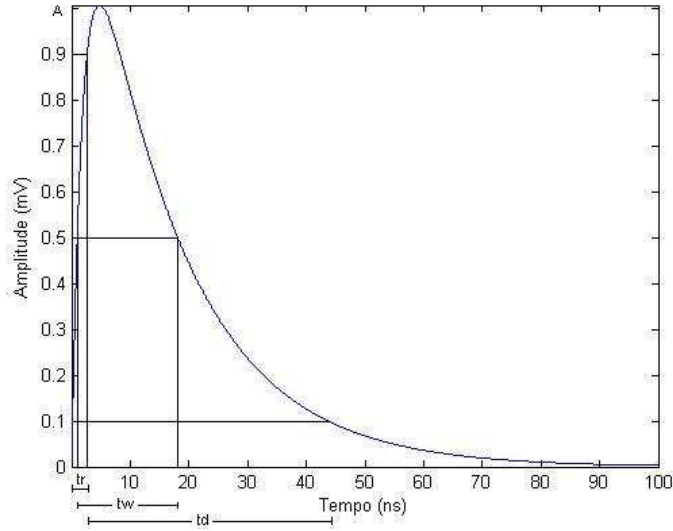


Figura 2.4: Pulso típico de DP juntamente com os parâmetros que o descrevem.

onde  $\tau = RC$  e  $\omega_0 = 1/\sqrt{LC}$ . A resposta impulsional dos circuitos representados por essas funções de transferência apresentam, respectivamente, a forma de uma exponencial decrescente ou de uma senóide modulada por uma exponencial decrescente. O valor de  $\tau$  controla o tempo de decaimento de ambos os pulsos e  $\omega_0$  determina a frequência de oscilação do pulso relativo ao circuito RLC.

Na realidade, o pulso de corrente causado pela DP não é um impulso de Dirac ideal. Quando um pulso real de DP passa pelo circuito de detecção, o pulso de saída terá sempre um tempo de subida finito. Considerando a forma de um pulso de DP e as características do circuito de detecção, o pulso exponencial decrescente (DEP - *damped exponential pulse*) e o pulso oscilatório decrescente (DOP - *damped oscillatory pulse*) podem ser numericamente simulados pelos modelos matemáticos mostrados nas equações (2.3) e (2.4), respectivamente. Os gráficos são mostrados nas Figuras 2.5.

$$DEP(t) = A (e^{-t/t_1} - e^{-t/t_2}) u(t) \quad (2.3)$$

$$DOP(t) = A \sin(2\pi f_c t) (e^{-t/t_1} - e^{-t/t_2}) u(t), \quad (2.4)$$

onde  $A$  dá o valor de pico do pulso,  $t_1$  e  $t_2$  determinam os demais parâmetros ( $t_r$ ,  $t_w$  e  $t_d$ ) e  $f_c$  é a frequência de oscilação do pulso do tipo DOP.

Para se verificar a validade do modelo proposto, pode-se utilizar a Figura 2.6. É possível perceber que o sinal medido tem grande semelhança com o modelo. A

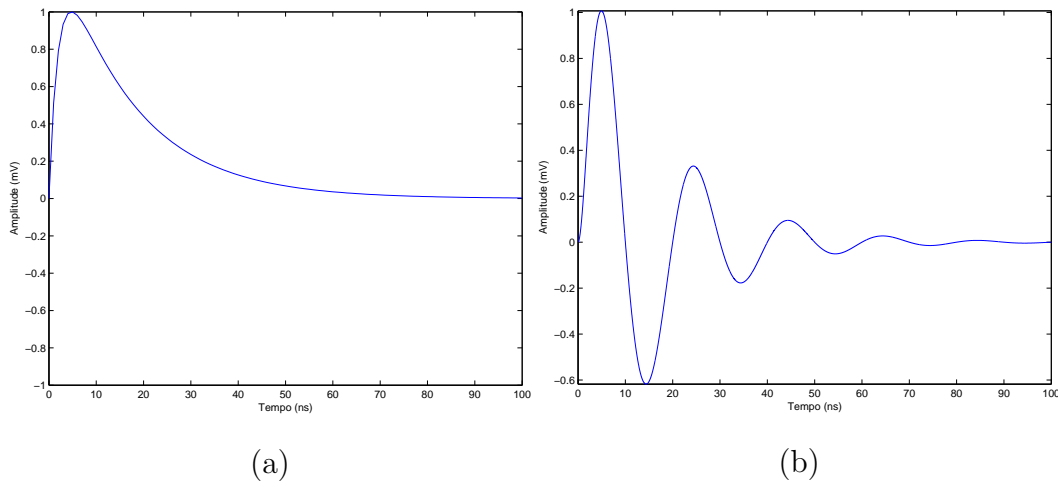


Figura 2.5: (a) Pulso típico do tipo DEP; (b) Pulso típico do tipo DOP.

correlação cruzada entre eles é de 0,9003. Os parâmetros usados neste gráfico são:  $t_1 = 0,83$ ,  $t_2 = 0,1$  e  $f_c = 5$  MHz. A taxa de amostragem empregada foi de 10 MHz.

## 2.7 Ruído presente no sinal de descargas parciais

Contudo, há uma imensa dificuldade a se enfrentar: a presença de elevado nível de ruído no ambiente de campo [9], que pode ser resultado de:

- Interferência de espectro discreto (DSI - *Discrete Spectrum Interference*) oriunda de transmissões de rádio e de sistemas de comunicação com portadoras baseadas em linha de transmissão.
- Interferência com formato de pulsos periódicos oriunda de sistemas de eletrônica de potência ou outro tipo de chaveamento periódico.
- Interferência com formato de pulsos estocásticos, advindos de operações de chaveamento esporádico ou raios, além de efeito corona, que pode ser oriundo de outro equipamento, afetando o equipamento que está sendo testado. Este ruído é geralmente modelado por ruído branco.

Esses ruídos e interferências, que acabam por contaminar e corromper a aquisição do sinal de DP, podem ser propagados tanto por condução como por indução. Nos Capítulos 3 e 4, será feito um estudo acerca dos métodos a serem usados na redução destes métodos.



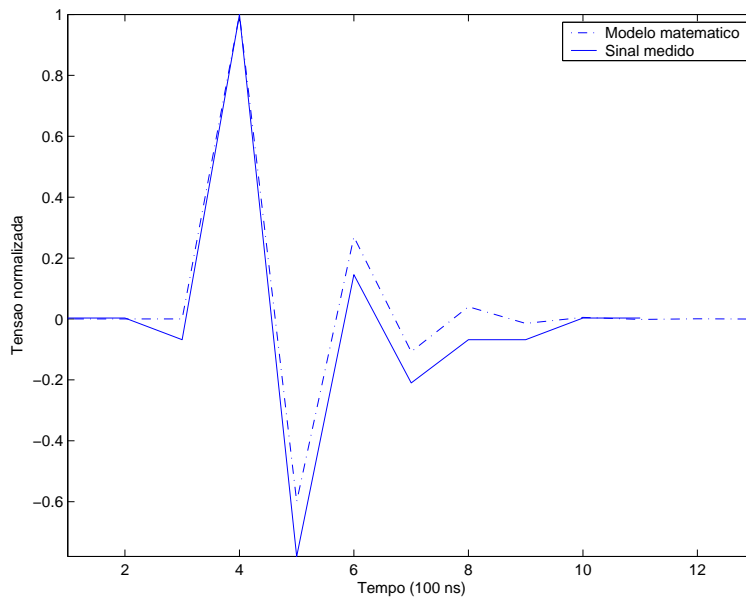


Figura 2.6: Comparação entre o modelo matemático e o sinal medido.

## 2.8 Conclusão

Neste capítulo, foram introduzidos conceitos básicos acerca das descargas parciais e, com isso, foi possível traçar uma relação entre tal fenômeno e a “saúde” de um equipamento elétrico. Foi observado também que a técnica mais eficiente para a medição de descargas parciais é a elétrica, onde o dispositivo a ser medido é inserido em um circuito elétrico. O sinal deve ser então capturado por um instrumento digital que pode ser um osciloscópio digital ou algo mais sofisticado, como o ICM System. Foram também apresentados modelos matemáticos que têm como objetivo facilitar tratamento desse tipo de sinal. Contudo, concluiu-se que a detecção é bastante dificultada pela presença de considerável quantidade de ruído, o que faz com que seja necessário usar técnicas de supressão de ruído, que serão vistas no próximo capítulo.

# Capítulo 3

## Métodos de Supressão de Ruído

### 3.1 Introdução

Este capítulo tem como objetivo dar uma visão geral sobre várias técnicas de supressão de ruído. Basicamente, pode-se separá-las em dois grandes grupos. O primeiro grupo engloba as técnicas que utilizam apenas um canal de informação, ou seja, usam o próprio sinal corrompido como base para a supressão do ruído. Neste grupo, estão, por exemplo, a subtração espectral (Seção 3.2), o filtro de Wiener (Seção 3.3) e a transformada *wavelet* (3.4). O segundo grupo é formado pelas técnicas que usam mais de um canal de informação, isto é, utilizam outros sinais relacionados com o sinal corrompido como ferramentas para a restauração do sinal original. Neste outro grupo, estão, por exemplo, a filtragem adaptativa (Seção 3.5) e ICA (análise de componentes independentes) (Seção 3.6).

Os fundamentos teóricos relacionados a processamento de sinais necessários para se entender tais métodos podem ser encontrados em [5] e [10].

A primeira coisa que se deve conhecer é o tipo de ruído com o qual se está trabalhando. No presente trabalho, o ruído é considerado aditivo, o que significa que o ruído pode ser modelado pela por

$$x(k) = s(k) + n(k), \quad (3.1)$$

onde  $s(k)$  é o sinal original livre de qualquer tipo de ruído (possivelmente apenas o ruído de medição) e  $n(k)$  é o ruído aditivo.

É usual assumir que tal ruído seja (além de aditivo) branco e gaussiano, o que é denotado pela sigla AWGN (do inglês *aditive white Gaussian noise*). Um

ruído é branco quando apresenta seu espectro constante ao longo de toda a faixa de frequência sobre a qual se está trabalhando. Isso também é válido (de forma aproximada) para quando a banda do ruído é muito mais larga que a frequência de amostragem do sistema. Isso significa que este tipo de ruído é formado por componentes relacionados a todas as frequências, de forma a ser igualmente distribuído no espectro. O fato de ser branco é uma analogia com o fato de a cor branca ser formada por uma combinação igualmente distribuída de todas as cores. Um ruído é gaussiano quando a função de densidade de probabilidade (ou simplesmente *pdf*) da variável aleatória que o representa segue uma curva gaussiana. Essa consideração é bastante plausível para os casos em que existem várias fontes de ruído independentes atuando simultaneamente, o que é o caso dos sinais de descargas parciais. Esse fato é justificado pelo Teorema do Limite Central [11].

## 3.2 Subtração espectral

Pode-se calcular o espectro de densidade de potência  $S_{xx}(\omega)$  do sinal  $x(n)$  descrito na equação (3.1), chegando-se a [12]

$$S_{xx}(\omega) = S_{ss}(\omega) + S_{nn}(\omega), \quad (3.2)$$

onde  $S_{ss}(\omega)$  é o espectro de densidade de potência do sinal original e  $S_{nn}(\omega)$  é o espectro de densidade de potência do ruído.

Com essa equação em mente, pode-se obter o espectro de densidade de potência do sinal original  $S_{ss}(\omega)$  apenas subtraindo-se o espectro do ruído  $S_{nn}(\omega)$  do espectro do sinal corrompido  $S_{xx}(\omega)$ . A partir deste espectro, pode-se tentar calcular o sinal de interesse. Essa é a idéia básica por trás da subtração espectral. Nesse ponto, duas perguntas podem aparecer:

1. Como obter o espectro de densidade de potência do ruído?
2. Como obter o sinal desejado a partir de seu espectro de densidade de potência?

Respondendo a primeira pergunta, o espectro do ruído pode ser estimado a partir de um intervalo da sequência de entrada onde se sabe que o sinal não é significativo, ou mesmo não está presente [13]. Esse intervalo é comumente chamado

de “silêncio” devido à analogia com o processo de *denoising* em sinais de áudio. Uma outra forma de estimar  $S_{nn}(\omega)$  seria a partir de uma faixa de frequência onde se sabe que o sinal não é significativo. Com base nisso, pode-se extrapolar esse espectro para toda a extensão da seqüência de entrada, considerando-se que o ruído seja também estacionário no sentido amplo.

A segunda pergunta é respondida em [12], onde é afirmado que o espectro de um sinal  $p(k)$  pode ser estimado (a menos de uma constante) se forem assumidas propriedades como estacionaridade e ergodicidade durante janelas de  $K$  amostras. Seu espectro de densidade de potência  $S_{pp}(\omega)$  pode ser estimado pelo periodograma  $|P(\omega)|^2$  de cada janela. A versão amostrada desse periodograma pode ser obtida pelo quadrado do módulo da DFT do sinal.

### 3.2.1 Algoritmo do método

Para esse método, deve-se estimar o espectro de densidade de potência do ruído. Isso é feito segundo o Algoritmo 3.1. Os procedimentos para a subtração espectral (executado sobre cada janela do sinal de entrada com o devido fator de superposição) são resumidos no Algoritmo 3.2.

1. Estimar o intervalo da seqüência de entrada onde o sinal não é significativo.
2. Multiplicar esse intervalo de “silêncio” por uma janela de Hamming [5] definida por

$$w(n) = \begin{cases} 0.54 + 0.46\cos\left(\frac{2\pi n}{M}\right), & |n| \leq \frac{M}{2} \\ 0, & |n| \geq \frac{M}{2}, \end{cases} \quad (3.3)$$

onde  $M$  é o tamanho da janela.

3. Calcular o módulo da DFT deste sinal janelado, forçando para que essa DFT já seja do comprimento da seqüência de entrada. Isso pode ser feito fazendo um *zero-padding* antes do cálculo da DFT, definida por

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}. \quad (3.4)$$

4. Calcular o quadrado do módulo da DFT.
5. Se o ruído for considerado branco, para se dar maior precisão ao cálculo, pode-se tirar a média do que foi calculado no passo anterior e considerar esse valor como constante por toda a extensão da seqüência de entrada.

**Algoritmo 3.1:** Estimação do espectro de densidade de potência do ruído.

1. Estimar o espectro do ruído segundo o Algoritmo 3.1.
2. Estimar o espectro do sinal corrompido de forma análoga ao Algoritmo 3.1.
3. Subtrair o espectro do ruído do espectro do sinal corrompido. Denomina-se essa diferença de espectro do sinal estimado. Se alguma amostra desse sinal estimado for menor que zero, tal amostra deverá receber o valor zero.
4. Calcular a DFT  $S(\omega_i)$  do sinal estimado, da seguinte forma: para cada amostra  $\hat{S}_{ss}(\omega_i)$  do espectro do sinal estimado

- Se a amostra for positiva,

$$S(\omega_i) = \sqrt{\hat{S}_{ss}(\omega_i)} e^{j\angle X(\omega_i)}, \quad (3.5)$$

onde  $\angle X(\omega_i)$  é a fase de uma amostra da DFT do sinal corrompido.

- Se a amostra for negativa,

$$S(\omega_i) = 0 \quad (3.6)$$

5. Calcular a parte real da DFT inversa do que foi calculado no passo anterior, chegando-se ao sinal estimado
6. Como isso deve ser feito para cada janela do sinal de entrada, o sinal reconstruído deve ser montado superpondo-se as janelas do sinal estimado com o mesmo fator de superposição.

**Algoritmo 3.2:** Procedimentos para a subtração espectral.

### 3.3 Filtro de Wiener

Na teoria de estimação de sinais em meio a ruído aditivo, existe um índice de desempenho largamente empregado que é o MSE (*mean square error*). Esse é calculado através da média dos quadrados das diferenças entre o sinal ideal e o sinal estimado, a ser definido na equação (3.9). O filtro que se obtém quando se tenta minimizar este índice é denominado Filtro de Wiener e pode ser visto em [10], [14] e [15]. Dessa forma, diz-se que o filtro de Wiener é ótimo no sentido do MSE. Uma grande vantagem desta técnica, como será visto mais adiante, é o fato de necessitar apenas das estatísticas de primeira e segunda ordem.

Considere um sinal estacionário no sentido amplo  $x(n)$ . O sinal na saída  $y(n)$  de um filtro linear invariante no tempo com resposta ao impulso  $h(n)$  é dado por

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k). \quad (3.7)$$

Esse filtro, definido por  $h(n)$ , deve gerar, a partir de  $x(n)$  uma saída que se aproxime ao máximo de  $s(n)$ . Dessa forma, pode ser definido um sinal de erro ou resíduo  $e(n)$  por

$$e(n) = d(n) - y(n). \quad (3.8)$$

onde  $d(n)$  é o sinal desejado, que, no caso, é  $s(n)$ . Este sinal  $e(n)$  é chamado sinal de erro ou resíduo. O objetivo no momento é mantê-lo o menor possível do ponto de vista estatístico. Assim, pode-se definir seu valor médio quadrático como

$$\begin{aligned} \epsilon &= E \left[ e^2(n) \right] \\ &= E \left[ (d(n) - y(n))^2 \right] \\ &= E \left[ d^2(n) \right] - 2E \left[ d(n)y(n) \right] + E \left[ y^2(n) \right] \\ &= A + B + C \end{aligned} \quad (3.9)$$

Com:

$$A = E \left[ d^2(n) \right] \quad (3.10)$$

$$B = -2 \sum_{k=0}^{\infty} h(k) E \left[ d(n)x(n-k) \right] \quad (3.11)$$

$$C = \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} h(k)h(m) E \left[ x(n-m)x(n-k) \right] \quad (3.12)$$

O que se assume aqui é que o sinal de entrada do filtro  $x(n)$  e o sinal desejado  $d(n)$  são conjuntamente estacionários no sentido amplo, o que significa que, além de ambos serem estacionários no sentido amplo separadamente, a correlação cruzada entre eles deve ser independente do tempo  $n$ . Assim, pode-se interpretar os termos da equação (3.9) da seguinte forma:

- O termo  $A$  pode ser visto como o valor médio quadrático do sinal desejado que é igual a  $E\left[d(n)d(n-0)\right]$  e, por isso, é igual a  $r_d(0)$ , ou seja, é a autocorrelação do sinal desejado para  $lag$  igual a zero.
- O termo  $B$  pode ser visto, de forma análoga a  $A$ , em função da correlação cruzada entre o sinal de entrada do filtro e o sinal desejado, para  $lags$  iguais a  $k$  (que varia de zero ao infinito) e, por isso, é denotado  $r_{dx}(k)$ . Seguindo a dedução,  $B = -2 \sum_{k=0}^{\infty} h(k)r_{dx}(k)$ .
- O termo  $C$  pode ser visto, de forma análoga a  $B$ , em função da autocorrelação do sinal de entrada do filtro para  $lags$  iguais a  $(k-m)$  e, por isso, é denominado  $r_x(k-m)$ . Seguindo a dedução,  $C = \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} h(k)h(m)r_x(k-m)$ .

Tendo em mente essas interpretações, pode-se reescrever o erro médio quadrático como

$$\epsilon = r_d(0) - 2 \sum_{k=0}^{\infty} h(k)r_{dx}(k) + \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} h(k)h(m)r_x(k-m). \quad (3.13)$$

Resta então calcular o mínimo dessa equação em função dos valores dos coeficientes do filtro  $h(k)$ , o que é feito derivando-se a equação (3.13) em relação a  $h(k)$ , obtendo-se

$$\frac{\partial \epsilon}{\partial h(k)} = -2r_{dx}(k) + 2 \sum_{m=0}^{\infty} h(m)r_x(k-m). \quad (3.14)$$

Agora basta igualar a equação (3.14) a zero, obtendo-se

$$\begin{aligned} -2r_{dx}(k) + 2 \sum_{m=0}^{\infty} h_o(m)r_x(k-m) &= 0 \\ \Rightarrow \sum_{m=0}^{\infty} h_o(m)r_x(k-m) &= r_{dx}(k) \end{aligned} \quad (3.15)$$

A equação (3.15) é denominada equação de Wiener-Hopf e as únicas quantidades conhecidas são a autocorrelação do sinal de entrada do filtro e a correlação



cruzada entre o sinal desejado e o sinal de entrada do filtro. A variável  $h_o(k)$  são os coeficientes do filtro que minimizam o erro médio quadrático.

À primeira vista, pode-se pensar em resolver essa equação através da transformada  $\mathcal{Z}$  [10]. Contudo, isso não é possível devido a essa equação só ser válida para valores de  $k$  maiores ou iguais a zero, ao passo que, para se poder utilizar a transformada  $\mathcal{Z}$ , deveria valer para valores negativos também. O filtro definido por  $h_o(k)$  é causal, isto é, a resposta impulsional deste filtro é zero para todo valor de  $k$  menor que zero. Essa restrição permite que esse sistema seja implementado em tempo real. Contudo, se for admissível um funcionamento de forma *off-line*, pode-se modificar a equação (3.15) chegando-se à equação que define o filtro de Wiener não realizável, definido por

$$\sum_{m=-\infty}^{\infty} h_{uo}(m)r_x(k-m) = r_{dx}(k). \quad (3.16)$$

Com essa nova equação, pode-se utilizar a transformada  $Z$  e chegar a uma fórmula fechada para o filtro de Wiener (não realizável), como mostrado a seguir

$$H_{ou}(z)S_x(z) = S_{dx}(z) \Rightarrow H_{ou}(z) = \frac{S_{dx}(z)}{S_x(z)}, \quad (3.17)$$

onde  $S_{dx}$  é a transformada  $\mathcal{Z}$  da correlação cruzada entre o sinal desejado e o sinal corrompido por ruído. Como o sinal desejado é descorrelacionado do ruído,  $r_{dn}$  é zero e  $S_{dx}$  passa a ser apenas a transformada  $\mathcal{Z}$  da autocorrelação do sinal desejado.

Assim,

$$H_{ou}(z) = \frac{S_{dx}(z)}{S_x(z)} = \frac{Z\{r_d(k) + r_{dn}(k)\}}{S_x(z)} = \frac{Z\{r_d(k)\}}{S_x(z)} = \frac{S_d(z)}{S_x(z)} \quad (3.18)$$

Pode-se, agora, escrever a equação (3.18) em função da razão sinal ruído. Isso é feito dividindo-se esta equação (tanto no numerador quanto no denominador) por  $S_d(z)$ , e fazendo-se

$$H_{ou}(z) = \frac{S_d(z)}{S_d(z) + S_n(z)} = \frac{\frac{S_d(z)}{S_d(z)}}{\frac{S_d(z)}{S_d(z)} + \frac{S_n(z)}{S_d(z)}} = \frac{1}{1 + \frac{1}{SNR(z)}}, \quad (3.19)$$

que pode ser visto como um filtro que atua de forma mais intensa nas frequências onde a SNR é pior, isto é, mais baixa.

Portanto, o filtro de Wiener pode ser aproximado, segundo [12], se se considerar que  $S_d(z) = S_x(z) - S_n(z)$ . Isso significa que

$$H_{ou}(z) = \frac{S_x(z) - S_n(z)}{S_x(z)} \quad (3.20)$$

Realizando-se as mesmas aproximações assumidas na Seção 3.2, conclui-se que, na prática, o Filtro de Wiener é construído da seguinte forma:

$$H_{ou}(z) = \frac{|X(z)|^2 - |N(z)|^2}{|X(z)|^2}. \quad (3.21)$$

### 3.3.1 Algoritmo do método

Para esse método, deve-se estimar o espectro de densidade de potência de um sinal. Isso é feito de forma semelhante ao que foi feito na subtração espectral. Os procedimentos para a filtragem de Wiener (executado sobre cada janela do sinal de entrada com o devido fator de superposição) são resumidos no Algoritmo 3.3.

1. Estimar o espectro do ruído segundo o algoritmo visto na subtração espectral.
2. Estimar o espectro do sinal corrompido de forma análoga ao algoritmo visto na subtração espectral.
3. Subtrair o espectro do ruído do espectro do sinal corrompido. Denomina-se essa diferença de espectro do sinal estimado. Se alguma amostra desse sinal estimado for menor que zero, tal amostra deverá receber o valor zero.
4. Calcular o filtro de Wiener dividindo-se o espectro do sinal estimado pelo espectro do sinal corrompido calculado no passo 2.
5. Calcular o sinal estimado multiplicando-se ponto a ponto o filtro de Wiener com a DFT do sinal corrompido.
6. Calcular a parte real da DFT inversa do que foi calculado no passo anterior, chegando ao sinal estimado.
7. Como isso deve ser feito para cada janela do sinal de entrada, o sinal reconstruído deve ser montado somando-se as janelas do sinal estimado com o mesmo fator de superposição.

**Algoritmo 3.3:** Procedimentos para a filtragem de Wiener.

## 3.4 Transformada *Wavelet*

### 3.4.1 A STFT

Quando se pensa em *denoising* no domínio da frequência, pensa-se logo em trabalhar com a transformada de Fourier e separar, através de um filtro, as componentes relacionadas ao sinal das componentes do ruído. Contudo, a transformada de Fourier apresenta uma grande desvantagem para sinais como os de DP's, isto é, sinais bem rápidos e bem definidos no tempo: as funções base dessa transformada (senos e cossenos complexos) possuem suporte infinito. Dessa forma, torna-se extremamente difícil encontrar uma combinação linear de senos e cossenos complexos cujo resultado seja algo muito localizado no tempo.

A primeira tentativa de se contornar esse problema foi a técnica denominada *short time Fourier transform* (STFT). Isso consiste em janelar o sinal e calcular a Transformada de Fourier convencional para cada uma dessas janelas. A STFT é dada pela expressão mostrada na equação (3.22). E como, nessa equação,  $v(n)$  tem duração finita, essa soma sempre irá convergir, isto é, a STFT sempre existe.

$$X_{STFT}(e^{j\omega}, m) = \sum_{n=-\infty}^{\infty} x(n)v(n-m)e^{-j\omega n}. \quad (3.22)$$

Com isso, ter-se-á um gráfico tridimensional onde o eixo X denota o índice da janela, o eixo Y denota as frequências e o eixo Z (representado por uma escala de tons de cinza) denota o módulo de cada componente frequencial. Com isso, é possível perceber, no diagrama de tempo-frequência descrito acima, onde um sinal rápido ocorre. Um exemplo deste diagrama é mostrado no terceiro gráfico da Figura 3.1. Nessa figura, pode-se ver que no segundo gráfico, não se tem idéia sobre que ponto ocorreu o sinal apresentado no primeiro gráfico.

Com base nesta visualização, verifica-se que há um *trade-off* entre a resolução no tempo e a resolução na frequência. Se forem usadas janelas muito curtas, será obtida uma resolução temporal bastante significativa pois a janela se moverá muito rapidamente. Contudo, se a janela for muito curta, o espectro de frequência (que apresenta o mesmo comprimento do sinal no tempo) também será curto (terá poucas amostras), ou seja, de baixa resolução, e vice-versa.

Pode-se ainda interpretar a STFT como um banco de filtros, o que é visto na

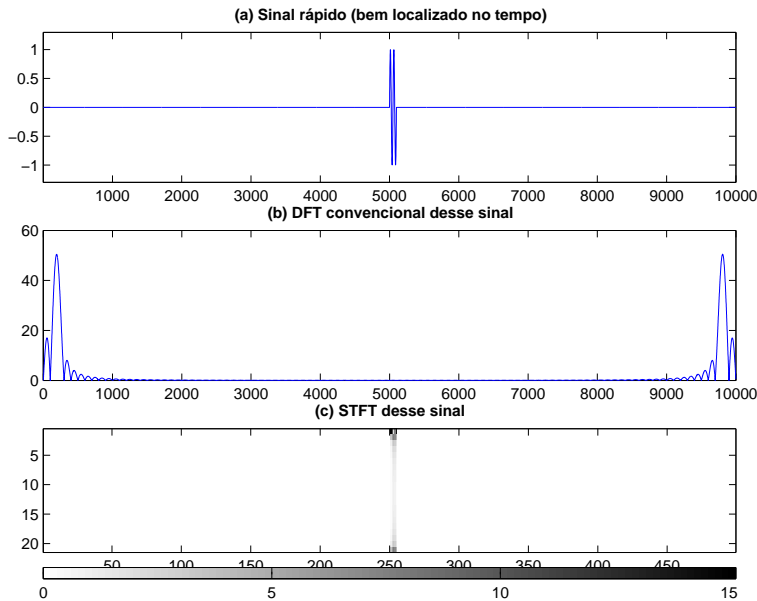


Figura 3.1: (a) Sinal original. (b) A DFT convencional do sinal original. (c) Gráfico gerado pela STFT: o eixo X (horizontal) mostra os índices das janelas, o eixo Y (vertical) mostra as freqüências sobre as quais foram calculados os espectros e os tons de cinza indicam os módulos de cada componente frequencial numa determinada janela.

Figura 3.2. No gráfico da Figura 3.1, cada linha horizontal (e, portanto, paralela ao eixo X) representa a saída de um filtro deste banco. A expressão que denota essa interpretação é da forma

$$x_k(n) = \sum_{m=-\infty}^{\infty} x(m)h_k(n_k n - m), \quad (3.23)$$

onde  $k$  varia de 0 a  $M - 1$ ,  $h_k$  são os vários filtros passa-banda do banco de filtros,  $n_k$  é o fator de decimação (neste caso, onde todas as bandas são iguais,  $n_k = M$ ) e  $x_k(n)$  é a saída de cada filtro. À luz desta interpretação, percebe-se ainda outra desvantagem na STFT: como todos os filtros apresentam o mesmo comprimento, os componentes de baixa freqüência tendem a ser mal estimados. Isso pode ser visto na Figura 3.3.

Na Figura 3.3a, pode-se ver que, quando a freqüência do sinal é alta, o espectro do sinal pode ser estimado com facilidade pois há muitos ciclos dentro da janela. Entretanto, para um sinal de baixa freqüência, como na Figura 3.3b, a janela irá conter poucos ciclos (e, às vezes, pode nem chegar a conter um ciclo inteiro) dificultando a estimação do espectro deste sinal.

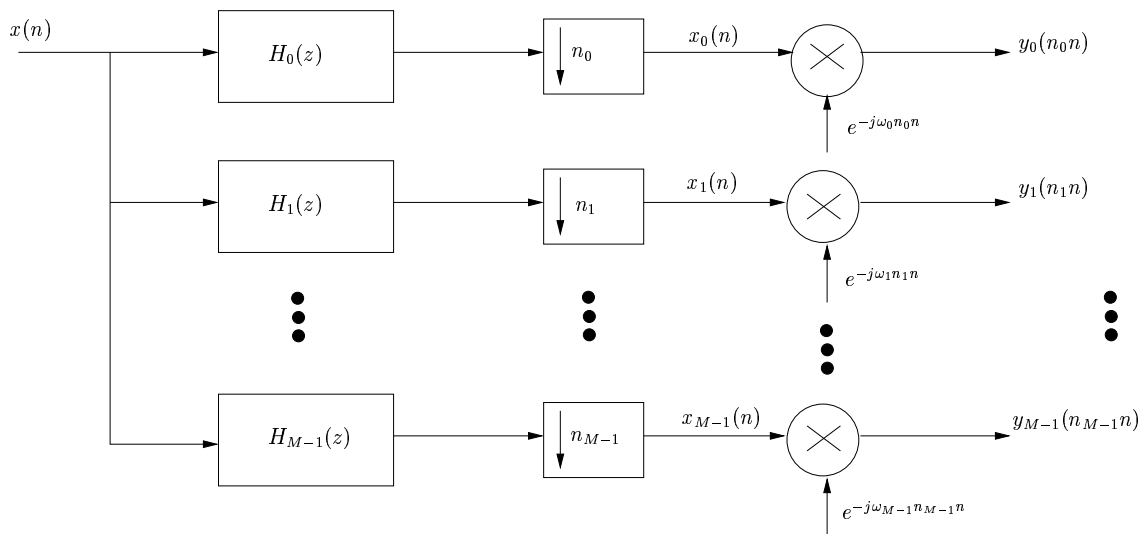


Figura 3.2: Interpretação da STFT como um banco de filtros.

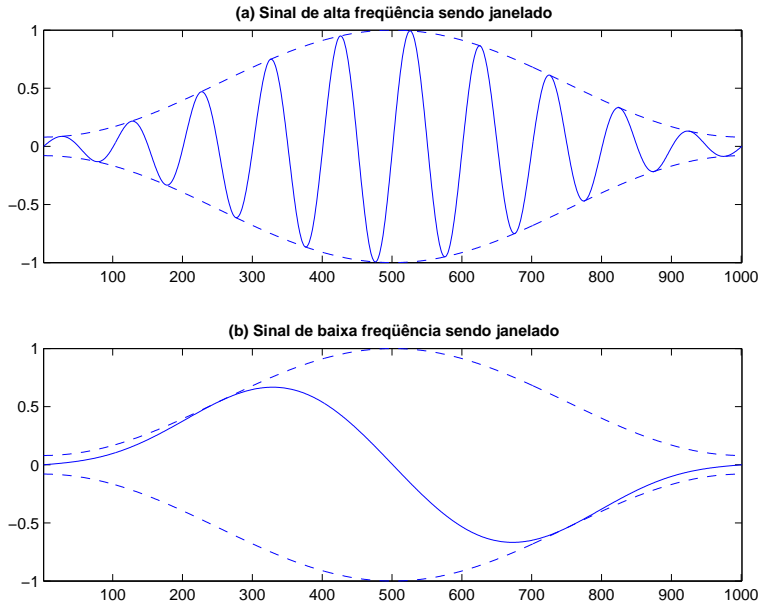


Figura 3.3: Comparação da ação de janelas de comprimentos iguais para sinais de diferentes frequências.

Uma outra questão que é levantada pela STFT é o fato de que, quando se usam  $M$  filtros nesse esquema do banco de filtros, o número de amostras é multiplicado por  $M$ . A teoria de sistemas multitaxa garante [16] [5] que, se a saída de cada um desses  $M$  filtros for decimada por um fator igual ao número de filtros, é possível, sob certas condições de reconstrução perfeita (realizadas por uma relação entre os filtros de análise e de síntese), recuperar integralmente o sinal original [5]. Dessa maneira, são justificados os decimadores presentes na estrutura da Figura 3.2.

Para contornar estes problemas, pode-se usar a transformada *wavelet*, que pode ser vista como uma STFT melhorada. Na verdade, a transformada *wavelet* é definida no domínio contínuo, como pode ser visto em [16]. A explicação que se segue é uma forma intuitiva de se entender essa transformada para se poder utilizá-la para os fins desse trabalho. A evolução da STFT para a transformada *wavelet* se dá através de dois passos:

1. A largura da banda dos filtros passa a ser função da frequência central dessa banda. Dessa forma, para os filtros com frequências centrais maiores, os filtros serão representados por uma janela mais estreita no tempo, o que implica uma largura de banda maior. Do mesmo modo, os filtros com frequências centrais mais baixas serão representados por uma janela mais larga no tempo, implicando larguras de banda menores. Com essa abordagem, o número de ciclos dentro de cada janela será o mesmo, independente da frequência central do filtro (Figura 3.4).
2. O fator de decimação passa a ser variável para cada filtro [16].

A adaptação descrita no item 1 é obtida ao se adotar a seguinte regra para divisão da banda original:

1. Divide-se inicialmente a banda original em duas através de um filtro passa-baixas e de um filtro passa-altas de forma que se tenha duas bandas de larguras aproximadamente iguais. Portanto, apenas se está dividindo a banda original à metade. Chamemos o filtro passa-baixas de  $G(z)$  e o filtro passa-altas de  $H(z)$ .
2. Conforme tem sido feito, faz-se a decimação dos sinais gerados na saída de cada um desses filtros por um fator de dois.

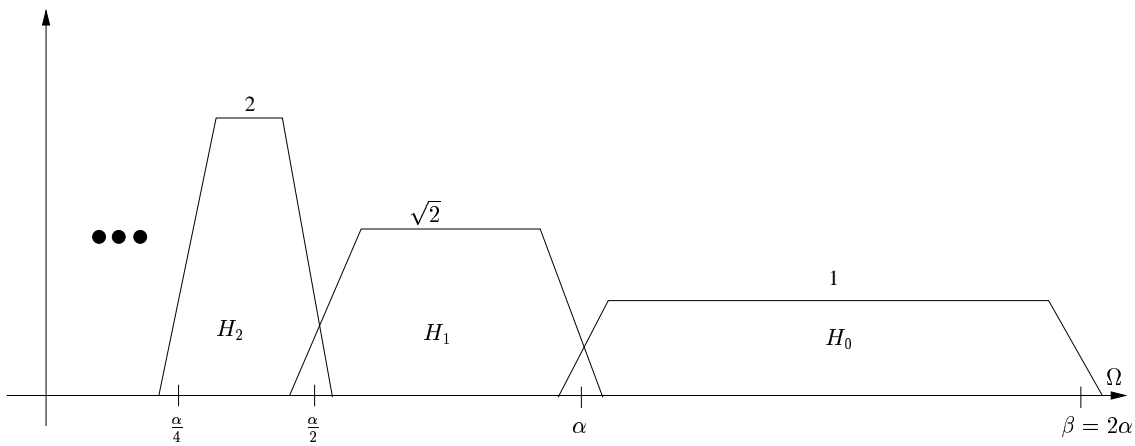


Figura 3.4: Banco de filtros usado na transformada *wavelet* (larguras das bandas diferentes).

3. Repete-se, então, o procedimento descrito nos itens 1 e 2 acima sobre o sinal gerado pela decimação da saída do filtro passa-baixas. Assim, ter-se-ão três bandas, sendo que duas (relativas às frequências mais baixas) terão larguras iguais a um quarto da largura original, enquanto que a terceira apresentará uma largura igual à metade da largura original (e representará as frequências mais altas).
4. O item 3 é, então, repetido até se atingir o número de bandas desejado.

Para representar esse algoritmo, recorre-se à Figura 3.5. O que é visto na Figura 3.5(a) pode ser simplificado de forma a se obter a Figura 3.5(b). É claro que, ao fim desse processo, a adaptação relativa a fatores de decimação variável é atendida automaticamente.

Pela Figura 3.5, pode-se ver que, se a árvore apresenta  $L$  níveis, então o número de canais será  $M = (L + 1)$ . Isso significa que a banda original será dividida entre  $M$  bandas de forma não-uniforme. Mais precisamente, as bandas resultantes estarão distribuídas segundo as potências de dois: a banda de frequência central maior terá metade da banda original, a seguinte terá um quarto, a próxima, um oitavo e assim por diante até a última, que terá uma largura igual à da penúltima (devido à construção da estrutura em árvore). Dessa forma, chega-se às seguintes

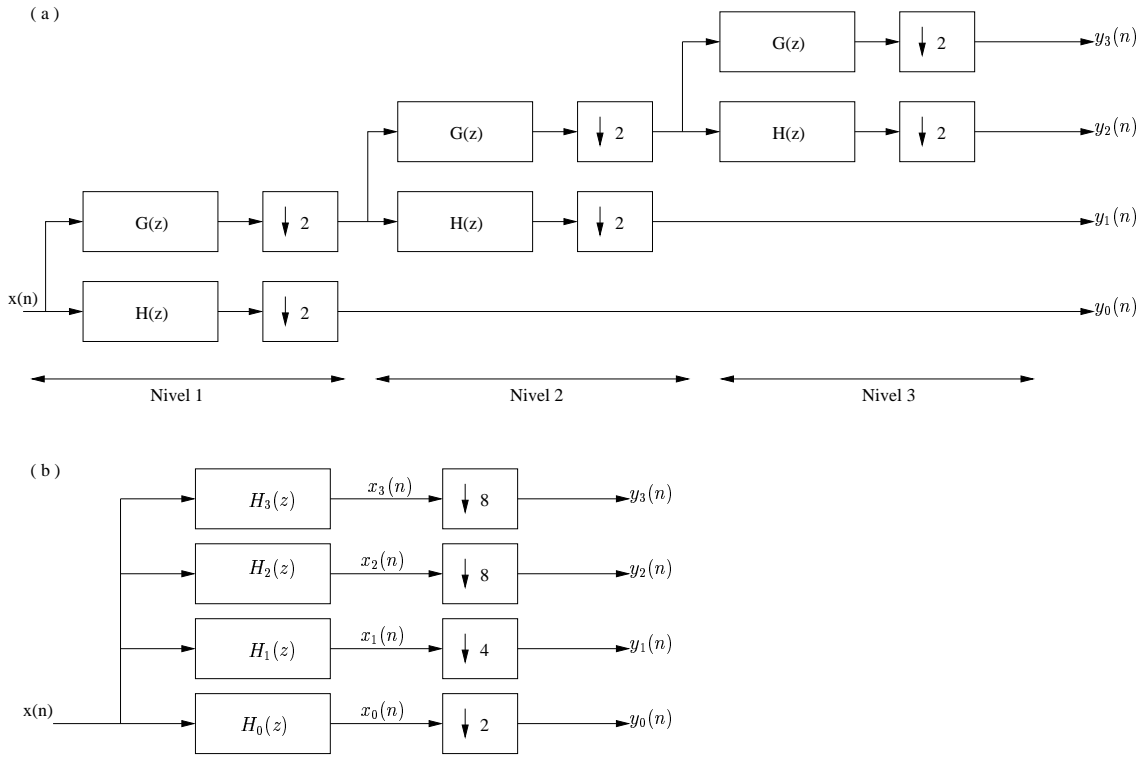


Figura 3.5: (a) Estrutura de árvore binária com 3 níveis. (b) Sistema equivalente de 4 canais.

equações:

$$y_k(n) = \sum_{m=-\infty}^{\infty} x(m)h_k(2^{k+1}n - m), \quad 0 \leq k \leq M - 2 \quad (3.24)$$

$$y_{M-1}(n) = \sum_{m=-\infty}^{\infty} x(m)h_{M-1}(2^{M-1}n - m) \quad (3.25)$$

As equações (3.24) e (3.25) mostram que, na verdade, a transformada *wavelet* faz somente a correlação cruzada entre o sinal de entrada e um outro sinal, representado por  $h_k(2^{k+1}n - m)$ . Esse outro sinal pode ser visto como uma versão escalada (ou comprimida), devido ao fator que multiplica a variável  $n$ , e deslocada (em  $m$  amostras) de um sinal  $h_k(n)$ , devido ao fato de que um *off-set*  $m$  é somado ao índice desse sinal.

Com todos esses fatos em mente, pode-se dizer que a transformada *wavelet* é calculada através de uma série de correlações cruzadas entre o sinal de entrada e versões deslocadas e escaladas de um outro sinal e que resultado de cada uma dessas correlações cruzadas é um coeficiente da transformada *wavelet*. Assim, a cada coeficiente está associado um par de valores formado por um valor de escala



e um de deslocamento. Esses dois valores são comumente associados à idéia de frequência e de tempo, respectivamente. Daí diz-se que a transformada *wavelet* trabalha no domínio conjunto tempo-frequência, diferentemente da transformada de Fourier convencional, que atua somente no domínio da frequência.

Esse sinal que é deslocado e escalado é denominado *wavelet*-mãe (ou, em inglês, *mother-wavelet*). Existem várias famílias de *wavelets*, cada uma com sua *wavelet*-mãe. Elas diferem em forma e comprimento. A forma e comprimento da *wavelet*-mãe devem ser escolhidos de forma específica para cada aplicação, ou seja, cada sinal que se deseja analisar é melhor tratado por uma certa *wavelet*-mãe. Nesse ponto está uma das principais dificuldades na utilização da transformada *wavelet*: a correta escolha quanto à *wavelet*-mãe. De uma forma geral, tende-se a escolher a *wavelet*-mãe que possui a forma mais similar ao sinal que se deseja analisar.

Tendo selecionado corretamente a *wavelet*-mãe para o sinal que se deseja analisar e submetido tal sinal à transformada *wavelet*, vê-se que há uma tendência de concentração de coeficientes não-nulos. Tal fato permite a utilização dessa transformada em compressão de sinais.

### 3.4.2 *Wavelet shrinkage*

O grande poder de concentração de coeficientes da transformada *wavelet* também pode ser aproveitado na supressão de ruído. Se a *wavelet*-mãe for bem escolhida e o sinal a ser submetido a essa transformada apresentar ruído aditivo, percebe-se que os coeficientes da *wavelet* relativos ao sinal estarão bem comprimidos nos primeiros coeficientes enquanto que os do ruído estarão bem distribuídos ao longo de todos os coeficientes.

O método de *denoising* conhecido como *wavelet shrinkage* (ou encolhimento dos coeficientes) é bem simples e intuitivo. Consiste apenas em submeter o sinal ruidoso à transformada *wavelet*, zerar (ou multiplicar por algum fator de encolhimento) os coeficientes relativos ao ruído e, finalmente, passar a transformada inversa.

Esse encolhimento (ou anulação) é feito submetendo cada coeficiente a um limiar (ou, em inglês, *threshold*). O que acontece a seguir depende de que método for adotado: *hard-thresholding* ou *soft-thresholding*. No primeiro método, todos os coeficientes cujos valores estejam abaixo do limiar serão zerados. No segundo, estes

mesmos coeficientes seriam apenas multiplicados por um fator de encolhimento.

A maior dificuldade, no entanto, está na correta escolha desses limiares. Para cumprir tal objetivo, existem várias metodologias. As que foram abordadas neste trabalho são: heurística, *Visu Shrink* (ou minimax), *Sure Shrink* e híbrida. Há ainda a possibilidade de se utilizar redes neurais, assim como foi feito em [13], mas que não será visto aqui.

### 3.4.3 Método heurístico

Esse método consiste em:

1. Submeter o sinal limpo à *wavelet*-mãe selecionada, usando-se um número de graus de decomposição menor ou igual ao logaritmo na base 2 do comprimento do sinal. Isso garante que haja pelo menos um coeficiente em cada escala da *wavelet*.
2. Observar em que banda ou bandas da *wavelet* estão os coeficientes de maior módulo.
3. Fazer os limiares dessas bandas iguais a zero.
4. Fazer os limiares das demais bandas iguais ao valor do maior coeficiente de cada banda (ou do maior coeficiente absoluto).

Esse método faz com que sejam descartados os coeficientes correspondentes a informações que não sejam parte do sinal e que, portanto, seriam parte do ruído.

### 3.4.4 Método *VISU Shrink* ou Minimax

Esse método busca uma estimativa mais acurada dos valores ótimos para os limiares baseando-se na estimativa da variância do ruído. Assume-se que, se o ruído for branco (assim como sua transformada *wavelet*), então, na banda referente às frequências mais altas, estará uma boa aproximação do ruído.

Segundo [17], essa estimativa da variância pode ser obtida por

$$\hat{\sigma}^2 = \left( \frac{m}{0.6745} \right)^2 \quad (3.26)$$

onde  $m$  é calculado como a mediana do valor absoluto dos coeficientes da transformada. Há duas formas de se abordar este problema: limiar dependente do nível e limiar independente do nível. Se o limiar é independente do nível, ou seja, é o mesmo para todas as bandas da transformada,  $m$  é calculado apenas sobre os coeficientes da banda de maior resolução. Caso contrário, cada banda tem seu limiar e  $m$  é calculado com base nos coeficientes de cada banda individualmente.

Neste método, os limiares são calculados da seguinte forma [17]

$$t_j = \hat{\sigma}^2 \sqrt{2 \log_{10} n} \quad (3.27)$$

onde  $n$  é o número de coeficientes na banda em questão.

### 3.4.5 Método *SURE Shrink*

Define-se o  $k$ -ésimo coeficiente de uma banda da transformada *wavelet* (de comprimento  $N$ ) de um sinal contaminado com ruído aditivo através de

$$Z_k = \beta_k + \sigma_k \xi_k \quad (3.28)$$

onde  $k = 1, 2, \dots, N$ ;  $\beta_k$  é o valor do coeficiente *wavelet* do sinal puro;  $\sigma_k > 0$  são as variâncias de cada coeficiente da transformada *wavelet* referentes ao ruído e  $\xi_k$  são variáveis aleatórias gaussianas independentes de média zero e variância 1.

Define-se, também, a função-risco do estimador de  $\beta_k$  através do erro médio quadrático. Este indica a penalidade por errar o valor de  $\beta_k$  (que é o valor real do coeficiente), e é dado por

$$R(\hat{\beta}_k, \beta_k) = \sum_{k=1}^N E \left[ (\hat{\beta}_k - \beta_k)^2 \right], \quad (3.29)$$

onde  $\hat{\beta}_k = Z_k + H_t(Z_k)$  é o estimador de  $H_t[\cdot]$ , que é uma função real diferenciável para qualquer valor de  $t$ . Essa função deve se aproximar o máximo possível de  $\sigma_k \xi_k$ . Assim, vê-se que o valor do parâmetro  $t$  deve ser escolhido de forma a ser aquele que minimiza a função risco do estimador  $R(\hat{\beta}_k, \beta_k)$ .

Como não se conhece o valor de  $\beta_k$ , o que se pode fazer é selecionar uma estimativa  $\hat{t}$  para  $t$  que minimiza uma estimativa não polarizada  $\hat{R}(\hat{\beta}_k, \beta_k)$  da função risco  $R(\hat{\beta}_k, \beta_k)$ . Para construir  $\hat{R}(\hat{\beta}_k, \beta_k)$ , deve se fazer o seguinte:

$$\begin{aligned} E \left[ (\hat{\beta}_k - \beta_k)^2 \right] &= E \left[ \hat{\beta}_k^2 \right] - 2E \left[ \hat{\beta}_k \beta_k \right] + E \left[ \beta_k^2 \right] \\ &= E \left[ H_t^2(Z_k) \right] + 2\sigma_k E \left[ \xi_k H_t(Z_k) \right] + \sigma_k^2, \end{aligned} \quad (3.30)$$

onde  $E [\xi_k H_t(Z_k)]$  pode ser calculado como em [13]

$$\begin{aligned} E [\xi_k H_t(\beta_k + \sigma_k \xi_k)] &= \frac{1}{\sqrt{2\pi}} \int \xi_k H_t(\beta_k + \sigma_k \xi_k) e^{-\frac{\xi_k^2}{2}} d\xi_k \\ &= \sigma_k E \left[ \left. \frac{dH_t(x)}{dx} \right|_{x=Z_k} \right]. \end{aligned} \quad (3.31)$$

Das equações (3.30) e (3.31), tem-se que

$$E [(\hat{\beta}_k - \beta_k)^2] = E [R(\sigma_k, Z_k, t)], \quad (3.32)$$

onde  $R(\sigma, x, t) = H_t^2(x) + 2\sigma^2 \frac{\partial}{\partial x} H_t(x) + \sigma^2$

Dessa forma, a equação (3.29) pode ser escrita como

$$R(\hat{\beta}_k, \beta_k) = E [\hat{R}(\hat{\beta}_k, \beta_k)], \quad (3.33)$$

onde

$$\hat{R}(\hat{\beta}_k, \beta_k) = \sum_{k=1}^N R(\sigma_k, Z_k, t) \quad (3.34)$$

é um estimador de risco e é chamado *Stein's Unbiased Risk Estimator* (SURE). O princípio de Stein é minimizar  $\hat{R}(\hat{\beta}_k, \beta_k)$  com respeito a  $t$  e usar aquele minimizador como um estimador do limiar. Portanto, o estimador do limiar é dado por

$$\hat{t} = \min_{t \geq 0} \sum_{k=1}^N R(\sigma_k, Z_k, t). \quad (3.35)$$

Este estimador depende da função de limiar,  $H_t(k)$ , selecionada. Para o *soft-thresholding*, tem-se  $H_t(x) = -xI\{|x| < t\} - tI\{|x| \geq t\} \text{sign}(x)$ , onde  $I\{\cdot\}$  é chamada de função indicadora e definida como 1, se  $x$  for verdadeiro e 0, se  $x$  for falso. Assim, pode-se escrever a equação (3.33) como

$$\begin{aligned} R(\sigma, x, t) &= (x^2 - \sigma^2)I\{|x| < t\} + (\sigma^2 + t^2)I\{|x| \geq t\} \\ &= [x^2 - \sigma^2] + (2\sigma^2 - x^2 + t^2)I\{|x| \geq t\}, \end{aligned} \quad (3.36)$$

o que resulta, para *soft-thresholding*, em [13]:

$$\hat{t} = \min_{t \geq 0} \sum_{k=1}^N (2\sigma^2 - x^2 + t^2)I\{|Z_k| \geq t\}. \quad (3.37)$$

### 3.4.6 Método híbrido

O método SURE não apresenta bons resultados quando a razão sinal-ruído é muito baixa. A causa disso está no fato de o ruído exercer grande influência sobre o valor da função a ser minimizada. Como resultado, não há uma boa estimativa dos limiares, que tendem a receber os valores máximos dentro de cada escala e o sinal resultante é nulo em toda a sua extensão. Para contornar tal problema, alterou-se a forma de cálculo de limiares para que se use o método SURE quando a razão sinal-ruído for alta, deixando-se para se usar o método VISU quando a razão sinal-ruído for baixa. Mas como pode se saber quando a razão sinal ruído é alta ou baixa? Em [12] adota-se uma razão sinal-ruído crítica heurística de valor dada pela Equação 3.38.

$$SNR_{critica} = \frac{(\log_2 K)^{3/2}}{\sqrt{K}} \quad (3.38)$$

A estimativa da razão sinal-ruído necessita da estimativa da variância do ruído, já inerente ao algoritmo, pois é necessária para a determinação do limiar pelo método VISU.

## 3.5 Filtragem adaptativa

Esse método faz parte do grupo dos métodos denominados multi-canal. Esse nome é justificado pelo fato de ser necessário ter acesso a mais de um sinal de informação (e não apenas ao sinal corrompido, como nos métodos estudados até aqui). A idéia por trás desse método é, ao invés de tentar impedir a captura do ruído que corrompe o sinal, permitir que o sinal seja capturado com ruído, captando também em um outro sensor outro ruído que tenha alguma correlação com o primeiro. Dessa forma, espera-se que este segundo ruído forneça alguma informação que possibilite a identificação e a remoção do primeiro, restaurando o sinal de interesse e obtendo-se um resultado melhor do que aquele que seria obtido ao não se permitir a captura do ruído.

Uma das formas de se fazer isso é através da filtragem adaptativa. Um filtro adaptativo [14] é um sistema cujo comportamento muda de forma a minimizar o MSE entre seu sinal de saída e algum sinal de referência. A configuração do sistema

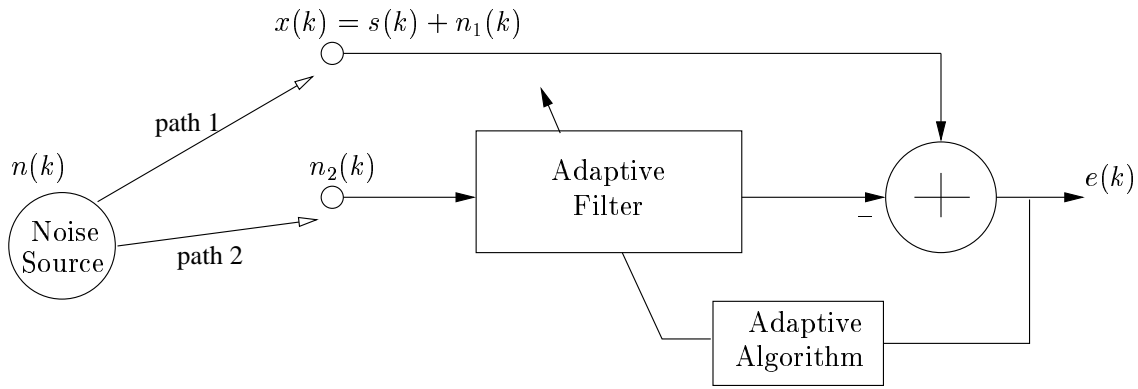


Figura 3.6: Configuração de filtragem adaptativa para supressão de ruído.

é montada de acordo com a aplicação em questão. Para supressão de ruído [12], a configuração utilizada pode ser explicada pelos seguintes fatos (ilustrados pela Figura 3.6):

1. Existe um sinal de interesse  $s(k)$ , o qual deve ser estimado.
2. Existe uma fonte de ruído em algum lugar no ambiente, emitindo  $n(k)$ .
3. Existem dois sensores. O primeiro tem como objetivo medir unicamente o ruído, e está afastado da fonte de ruído através do “caminho 2”. O outro sensor deve captar o sinal. Contudo, esse sinal é corrompido pelo ruído que sai da fonte de ruído e chega ao sensor através do “caminho 1”.
4. Cada caminho corresponde a uma função de transferência ( $H_1(z)$  e  $H_2(z)$ ). Assim, o ruído que emitido pela fonte de ruído se torna  $n_1(k)$  ao fim do “caminho 1” (alterado por  $H_1(z)$ ) e é adicionado ao sinal de interesse  $s(k)$ , formando  $x(k)$ . Além disso, o ruído emitido pela fonte de ruído se torna  $n_2(k)$  quando alterado por  $H_2(z)$  no fim do “caminho 2”.

O objetivo do sistema é estimar  $n_1(k)$  a partir de  $n_2(k)$  e subtrair essa estimativa de  $x(k)$  resultando em  $s(k)$ . O sinal de referência é  $x(k)$ . A entrada do filtro é  $n_2(k)$ . Como esse sinal de entrada não tem nenhuma informação sobre o sinal de interesse, o filtro adaptativo só poderá modelar o componente ruidoso de  $x(k)$ . Esse componente aparecerá na saída do filtro adaptativo e, assim, será removido limpando o sinal corrompido e recuperando o sinal de interesse.

O sistema evolui dependendo do sinal de entrada e do sinal de referência. O sinal de erro, isto é, a diferença entre o sinal de referência e a saída do filtro adaptativo é submetido a um algoritmo adaptativo que faz com que os coeficientes do filtro tenham seus valores alterados de forma a minimizar o MSE, dado por

$$\epsilon = E \left[ e^2(k) \right] = E \left[ (d(k) - y(k))^2 \right] = E \left[ d^2(k) - 2d(k)y(k) + y^2(k) \right] \quad (3.39)$$

Contudo,  $y(k)$  é a saída do filtro adaptativo, que, por enquanto, será tratado simplesmente como um combinador linear. Além disso, pode-se definir  $\mathbf{x}(k) = [x_0(k)x_1(k)\dots x_N(k)]^T$  como o vetor de entradas. Analogamente, define-se  $\mathbf{w}(k) = [w_0(k)w_1(k)\dots w_N(k)]^T$  como sendo o vetor coluna onde cada elemento é o valor de um coeficiente do filtro. Nota-se que estes coeficientes variam ao longo do tempo, conforme se espera de um filtro adaptativo. É importante perceber que o número de elementos nos dois vetores é o mesmo. O motivo será visto mais adiante.

A saída será dada por uma combinação linear dos elementos de  $\mathbf{x}(k)$  comandada por  $\mathbf{w}(k)$ . Isso significa que os valores dos elementos do vetor  $\mathbf{x}(k)$  são os valores que serão multiplicados pelos coeficientes do filtro e somados para gerar a saída, que será dada por

$$y(k) = \sum_{i=0}^N w_i(k)x_i(k) = \mathbf{w}^T(k)\mathbf{x}(k) \quad (3.40)$$

A equação (3.40) pode ser reescrita com base na estrutura de um filtro FIR, transformando o vetor de entradas em uma *delay-line*, chegando-se a

$$y(k) = \sum_{i=0}^N w_i(k)x(k-i) = \mathbf{w}^T(k)\mathbf{x}(k) \quad (3.41)$$

Nessa equação, percebe-se que  $\mathbf{x}(k) = [x(k)x(k-1)x(k-2)\dots x(k-N)]^T$  é o vetor coluna onde cada elemento é uma amostra do sinal de entrada atrasada pelo número correspondente a seu índice, o que explicita ainda mais a denominação de *delay-line*. Os valores variam ao longo do tempo de modo a atuarem como uma fila do tipo FIFO (*first-in first-out*). Usando-se a equação (3.41) na equação (3.39), tem-se

$$\epsilon(k) = E \left[ d^2(k) \right] - 2E \left[ d(k)\mathbf{w}^T(k)\mathbf{x}(k) \right] + E \left[ \mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) \right]. \quad (3.42)$$

Removendo os fatores relativos a  $\mathbf{w}$  de dentro do operador média, chega-se a

$$\epsilon(k) = E \left[ d^2(k) \right] - 2\mathbf{w}^T E \left[ d(k)\mathbf{x}(k) \right] + \mathbf{w}^T E \left[ \mathbf{x}(k)\mathbf{x}^T(k) \right] \mathbf{w} \quad (3.43)$$

Contudo, pode-se definir o vetor de correlação cruzada  $\mathbf{p}$  entre o sinal de entrada do filtro  $\mathbf{x}(k)$  e o sinal de referência  $d(k)$ . Além disso, denomina-se  $\mathbf{R}$  a matriz de auto-correlação relativa ao sinal de entrada  $\mathbf{x}(k)$ . Portanto têm-se as definições

$$\mathbf{p} = E \left[ d(k)\mathbf{x}(k) \right] \quad (3.44)$$

$$\mathbf{R} = E \left[ \mathbf{x}(k)\mathbf{x}^T(k) \right] \quad (3.45)$$

Substituindo as equações (3.44) e (3.45) em (3.43), tem-se

$$\epsilon(k) = E \left[ d^2(k) \right] - 2\mathbf{w}^T \mathbf{p} + \mathbf{w}^T \mathbf{R} \mathbf{w} \quad (3.46)$$

Resta, então, calcular o valor dos coeficientes que minimizam o erro médio quadrático. Isso é feito derivando-se a expressão para o MSE em relação a  $\mathbf{w}$ , o que é dado por

$$\mathbf{g}_{\mathbf{w}} = \frac{\partial \epsilon}{\partial \mathbf{w}} = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}. \quad (3.47)$$

Igualando-se o resultado obtido a zero e resolvendo-se a equação com relação a  $\mathbf{w}$ , chega-se à solução de Wiener-Hopf

$$\mathbf{w}_0 = \mathbf{R}^{-1}\mathbf{p} \quad (3.48)$$

Infelizmente, segundo [14], na prática, estimativas precisas de  $\mathbf{R}$  e  $\mathbf{p}$  não estão disponíveis. Quando o sinal de entrada e o sinal de referência são ergódicos, pode-se usar médias temporais para estimar  $\mathbf{R}$  e  $\mathbf{p}$ , que é o que é feito implicitamente pelos algoritmos adaptativos que serão vistos nas próximas seções.

### 3.5.1 Algoritmo *Least-Mean Squares* (LMS)

O objetivo de um algoritmo adaptativo é chegar na solução de Wiener, ou pelo menos se aproximar dela. Nesse momento, pode surgir a pergunta: “por que não usar diretamente a solução de Wiener?” A resposta é simples: porque não é possível, na grande parte das aplicações, ter os valores de  $\mathbf{R}$  e  $\mathbf{p}$ , porque precisar-se-ia de uma quantidade infinita de dados e os processos podem ser não estacionários. Dessa forma, os algoritmos adaptativos vieram como uma alternativa para contornar



esse problema. Eles usam processos iterativos para minimizar uma função custo e, com isso, obter valores ótimos para os coeficientes do filtro adaptativo.

Um dos mais simples algoritmos e, por isso, um dos mais utilizados é o LMS (*Least Mean Square*), que, como o nome indica, visa a minimizar o erro médio quadráticos. Ele faz uso do algoritmo *Steepest Descent* [14], que poderia ser traduzido como “descida mais íngreme”. Esse algoritmo estima que os valores dos coeficientes na próxima iteração serão função dos coeficientes na iteração atual e dados por

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \mu \hat{\mathbf{g}}_{\mathbf{w}}(k) \quad (3.49)$$

onde  $\hat{\mathbf{g}}_{\mathbf{w}}(k)$  é uma estimativa do gradiente do MSE em função do valor dos coeficientes do filtro e  $\mu$  é um fator que serve para limitar o movimento de forma a não deixar que o movimento divirja do ponto de mínimo. Por essa razão, esse fator recebe o nome de fator de convergência. O sinal de menos que multiplica o gradiente é devido ao fato de o gradiente apontar a direção de maior crescimento da superfície de erro. Dessa forma, o novo coeficiente será o coeficiente atual somado de uma parcela na direção de maior decrescimento da superfície de erro.

O vetor  $\hat{\mathbf{g}}_{\mathbf{w}}(k)$  pode ser calculado por

$$\begin{aligned} \hat{\mathbf{g}}_{\mathbf{w}}(k) &= \nabla_w e^2(k) \\ &= 2\mathbf{e}(k)\nabla_w \mathbf{e}(k) \\ &= 2\mathbf{e}(k)\nabla_w [\mathbf{d}(k) - \mathbf{x}^T(k)\mathbf{w}(k)] \\ &= -2\mathbf{e}(k)\mathbf{x}(k) \end{aligned} \quad (3.50)$$

Os procedimentos para o LMS são sintetizados no Algoritmo 3.4.

**Inicialização**

$$\mathbf{x}(0) = \mathbf{w}(0) = [0 \ 0 \ \dots \ 0]^T$$

**Para**  $k \geq 0$

$$e(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}(k)$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu e(k)\mathbf{x}(k)$$

**Algoritmo 3.4:** Procedimentos para o LMS.

### 3.5.2 Algoritmo *Normalized Least-Mean Squares* (NLMS)

Um das principais desvantagens do algoritmo LMS, é o fato de o fator de convergência  $\mu$  ser constante. Isso propicia o aparecimento do fenômeno conhecido como desajuste, o que significa que o MSE ficara oscilando em torno do mínimo sem alcançá-lo, aumentando o tempo de convergência e, portanto, diminuindo a velocidade de convergência. Além disso, o fato de  $\mu$  ser constante significa que o algoritmo está assumindo que a superfície MSE apresenta um aspecto “uniforme”, o que nem sempre é verdade. Isso gera uma grande diminuição na velocidade de convergência pois esse tipo de fator de convergência, ironicamente, nem sempre poderá garantir a convergência. Isso acontece pois, à medida que há uma maior aproximação do ponto ótimo na superfície de desempenho, o passo precisa ser reduzido. Se isso não acontecer, o ponto ótimo pode estar num lugar da superfície onde os coeficientes do filtro irão sempre passar mais nunca iriam se aproximar.

Uma primeira tentativa de contornar esse problema é através de fatores de convergência variáveis. Dessa forma, ter-se-á a expressão modificada do algoritmo LMS como mostrada por

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\mu_k \mathbf{x}(k)e(k) = \mathbf{w}(k) + \Delta \mathbf{w}(k) \quad (3.51)$$

A filosofia por trás disso é tentar minimizar o erro quadrático instantâneo, que, além de ser muito simples de se calcular, é uma boa estimativa para o MSE. O erro quadrático instantâneo é dado pela equação (3.52).

$$\begin{aligned} e^2(k) &= [d(k) - y(k)]^2 \\ &= [d(k) - \mathbf{w}^T(k)\mathbf{x}(k)]^2 \\ &= d^2(k) + \mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) - 2d(k)\mathbf{w}^T(k)\mathbf{x}(k) \end{aligned} \quad (3.52)$$

Substituindo a equação (3.51) em (3.52), tem-se

$$\begin{aligned} e^2(k+1) &= e^2(k) + 2\Delta \mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) \\ &\quad + \Delta \mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta \mathbf{w}(k) \\ &\quad - 2d(k)\mathbf{w}^T(k)\mathbf{x}(k) \end{aligned} \quad (3.53)$$

Define-se, então,  $\Delta e^2(k) \triangleq [e^2(k+1) - e^2(k)]$  e chega-se a

$$\Delta e^2(k) = -2\Delta \mathbf{w}^T(k)\mathbf{x}(k)e(k) + \Delta \mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Delta \mathbf{w}(k) \quad (3.54)$$

Como dito anteriormente, deve-se fazer com que o erro quadrático instantâneo seja o menor possível, escolhendo de forma adequada o valor para  $\mu_k$ . Substituindo  $\Delta \mathbf{w}(k) = 2\mu_k \mathbf{x}(k)e(k)$ , em (3.54), chega-se a

$$\Delta e^2(k) = -4\mu_k e^2(k) \mathbf{x}^T(k) \mathbf{x}(k) + 4\mu_k^2 e^2(k) [\mathbf{x}^T(k) \mathbf{x}(k)]^2 \quad (3.55)$$

Derivando essa equação e igualando seu resultado a zero, tem-se que o valor de  $\mu_k$  que minimiza o erro instantâneo quadrático é dado por

$$\mu_k = \frac{1}{2\mathbf{x}^T(k) \mathbf{x}(k)} \quad (3.56)$$

Contudo, o denominador da fração contida nessa equação pode tender a zero, o que faria que os coeficientes tivessem seus valores levados a infinito. Para contornar esse problema, acrescenta-se uma pequena parcela de módulo muito menor que a potência do sinal de entrada, a qual é denotada por  $\mathbf{x}^T(k) \mathbf{x}(k)$ . Além disso, multiplica-se  $\Delta \mathbf{w}(k)$  por um fator de convergência fixo para controlar o desajuste, devido ao fato de que as derivadas são agora baseadas no erro instantâneo (o que geraria um erro maior), e não mais no MSE. Assim, a equação de atualização de coeficientes para o NLMS fica da forma

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mu_n}{\gamma + \mathbf{x}^T(k) \mathbf{x}(k)} \mathbf{x}(k) e(k) \quad (3.57)$$

Os procedimentos para o NLMS estão sintetizados no Algoritmo 3.5.

Inicialização

$$\mathbf{x}(0) = \mathbf{w}(0) = [0 \ 0 \ \dots \ 0]^T$$

selecione  $\mu_n$  entre 0 e 2

Para  $k \geq 0$

$$e(k) = d(k) - \mathbf{x}^T(k) \mathbf{w}(k)$$

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\mu_n}{\gamma + \mathbf{x}^T(k) \mathbf{x}(k)} e(k) \mathbf{x}(k)$$

**Algoritmo 3.5:** Procedimentos para o NLMS.

### 3.5.3 Algoritmo *Recursive Least Squares* (RLS)

Os algoritmos adaptativos vistos até aqui têm um caráter essencialmente estatístico. Isso significa que tudo é feito como uma aproximação para valores que deveriam ser calculados levando em consideração todo o *ensemble* do processo. O algoritmo RLS tem como uma de suas principais características o fato de ser de caráter determinístico. Isso significa que não será considerada nenhuma formulação estatística como a média, usada nos algoritmos anteriores. Assim, ter-se-á uma função custo determinística para ser minimizada.

O objetivo deste algoritmo é fazer com que os coeficientes do filtro sejam definidos de modo a fazer com que a saída do filtro  $y(k)$  seja aproximada do sinal de referência  $d(k)$  no sentido dos mínimos quadrados. Na prática, a função-custo a ser minimizada será dada por

$$\epsilon(k) = \sum_{i=0}^k \lambda^{k-i} e^2(i) = \sum_{i=0}^k \lambda^{k-i} [d(i) - \mathbf{x}^T(i)\mathbf{w}(k)]^2 \quad (3.58)$$

Com relação à equação (3.58), algumas considerações devem ser feitas:

1.  $\lambda$  é um fator de ponderação e indica quanto um erro do passado influencia na função custo a ser minimizada. Ele deve estar no intervalo  $0 \ll \lambda \leq 1$ . Por esse motivo, quanto maior for seu expoente, menor será seu valor. Dessa forma, segundo o que se vê na função-custo, quanto mais para o passado for o erro, menor será seu peso. Por esse motivo,  $\lambda$  também recebe o nome de fator de esquecimento.
2. Outra diferença em relação à função custo dos algoritmos anteriores é o fato de  $\epsilon$  ser o erro *a posteriori*. Isso significa que o erro é calculado a partir de valores atualizados para os coeficientes do filtro adaptativo. No caso do LMS, por exemplo, onde o erro é *a priori*, o erro que é usado na função custo é calculado para os valores atuais dos coeficientes do filtro e, a partir deste erro substituído na regra de atualização do LMS, os coeficientes serão atualizados. Entretanto, no caso do RLS, os coeficientes são atualizados através da regra de atualização do RLS e só então o erro é calculado para entrar na função custo. Para evitar ambigüidade, no caso do RLS, o erro *a priori* será denotado por  $e'(k)$  enquanto que, como já foi visto, o erro *a posteriori* será representado por  $e(k)$ .

Tendo em mente essas considerações, pode-se analisar a equação (3.58) como sendo o somatório de erros *a posteriori* do presente e do passado ponderados por um fator de esquecimento. Mas novamente é bom lembrar que todos esses erros são calculados usando-se apenas o último conjunto atualizado de coeficientes do filtro adaptativo.

Como sempre, precisa-se calcular a derivada da função custo e igualar seu resultado a zero para se obter os coeficientes que minimizam seu valor. Fazendo isso, chega-se a

$$\begin{aligned} \frac{\partial \epsilon(k)}{\partial \mathbf{w}(k)} &= -2 \sum_{i=0}^k \lambda^{k-i} \mathbf{x}(i) [d(i) - \mathbf{x}^T(i) \mathbf{w}(k)] \\ &= \left[ \sum_{i=0}^k \lambda^{k-i} \mathbf{x}(i) \mathbf{x}^T(i) \right] \mathbf{w}(k) - \left[ \sum_{i=0}^k \lambda^{k-i} \mathbf{x}(i) d(i) \right] = \mathbf{0} \end{aligned} \quad (3.59)$$

Resolvendo a equação (3.59) em relação a  $\mathbf{w}(k)$ , chega-se a

$$\mathbf{w}(k) = \left[ \sum_{i=0}^k \lambda^{k-i} \mathbf{x}(i) \mathbf{x}^T(i) \right]^{-1} \left[ \sum_{i=0}^k \lambda^{k-i} \mathbf{x}(i) d(i) \right] = \mathbf{R}_D^{-1}(k) \mathbf{p}_D(k) \quad (3.60)$$

Na equação (3.60),  $\mathbf{R}_D(k)$  é denominado matriz de correlação determinística do vetor de entrada e  $\mathbf{p}_D(k)$  é denominado vetor de correlação cruzada determinística entre o vetor de entrada e o sinal desejado. Contudo, o cálculo da matriz inversa de  $\mathbf{R}_D(k)$  na forma convencional pode ser extremamente custoso computacionalmente. A complexidade é de  $O[N^3]$ . Para aliviar tamanho esforço computacional, pode-se recorrer ao lema de inversão de matrizes [14], apresentado na equação (3.61). A complexidade desse cálculo é de  $O[N^2]$ .

$$\mathbf{S}_D(k) = \mathbf{R}_D^{-1}(k) = \frac{1}{\lambda} \left[ \mathbf{S}_D(k-1) - \frac{\mathbf{S}_D(k-1) \mathbf{x}(k) \mathbf{x}^T(k) \mathbf{S}_D(k-1)}{\lambda + \mathbf{x}^T(k) \mathbf{S}_D(k-1) \mathbf{x}(k)} \right] \quad (3.61)$$

Usando-se a equação (3.61), chega-se a uma primeira formulação do RLS, dada pelo Algoritmo 3.6.

Inicialização

$$\mathbf{S}_D(-1) = \delta \mathbf{I}$$

onde  $\delta$  pode ser o inverso da estimativa  
da potência do sinal de entrada

$$\mathbf{p}_D(-1) = \mathbf{x}(-1) = [0 \ 0 \ \dots \ 0]^T$$

Para  $k \geq 0$

$$\mathbf{S}_D(k) = \mathbf{R}_D^{-1}(k) = \frac{1}{\lambda} \left[ \mathbf{S}_D(k-1) - \frac{\mathbf{S}_D(k-1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{S}_D(k-1)}{\lambda + \mathbf{x}^T(k)\mathbf{S}_D(k-1)\mathbf{x}(k)} \right]$$

$$\mathbf{p}_D(k) = \lambda \mathbf{p}_D(k-1) + d(k)\mathbf{x}(k)$$

$$\mathbf{w}(k) = \mathbf{S}_D(k) \mathbf{p}_D(k)$$

$$y(k) = \mathbf{w}^T(k)\mathbf{x}(k)$$

$$e(k) = d(k) - y(k)$$

**Algoritmo 3.6:** Procedimentos para o RLS.

### 3.5.4 Algoritmo *Set-Membership Affine Projection*

O RLS, visto na seção anterior, apresenta, para sinais altamente correlacionados, uma convergência mais rápida do que a do LMS e do NLMS. Contudo, o custo dessa maior velocidade é uma maior complexidade computacional. Para se contornar este problema, foram criados os algoritmos de reutilização de informação (ou *data-reuse* em inglês), que, por sua vez, tendem a aumentar o desajuste final. De forma geral, tenta-se contornar o problema de desajuste sacrificando-se a velocidade de convergência. Isso é feito multiplicando-se a atualização dos coeficientes por um fator de atenuação denominado *step-size*.

É possível melhorar ainda mais o desempenho, segundo [18], submetendo-se o algoritmo de *data-reuse* à técnica conhecida como *set-membership filtering* ou SMF. A filosofia por trás dessa técnica é estabelecer um limite superior  $\gamma$  para o erro entre o sinal desejado e a saída do filtro adaptativo. Dessa forma, para cada instante de tempo  $k$ , define-se um conjunto de vetores de coeficientes do filtro para os quais o erro será limitado superiormente por  $\gamma$ . Esse conjunto recebe o nome de conjunto de restrição ou *constraint set* e é denotado por  $\mathbf{H}_k$ . A interseção de todos esses conjuntos (cada um relativo a um instante  $k$ ) é denominada *membership set* e é denotada por  $\Psi_k$ . Dessa forma, o próximo coeficiente deve estar contido na interseção dos conjuntos de restrição relativos aos últimos  $P$  instantes de tempo. Esse conjunto será denotado por  $\Psi_k^P$ .  $P$  é um parâmetro livre do sistema, depende da natureza da aplicação e indica o número de reusos. Através dessa propriedade de discernimento sobre a informação utilizada, esse tipo de algoritmo reduz na média a complexidade computacional.

Deve-se lembrar que  $\mathbf{w}_k$  está necessariamente, por definição, na interseção dos últimos  $P$  conjuntos de restrição com a exceção do último. Este último conjunto é relativo ao último par “saída - sinal de referência” (no instante  $k$ ). Uma primeira hipótese seria aquela na qual erro entre a saída no instante  $k$  e o sinal de referência resultam em um erro menor que  $\gamma$ . Nessa hipótese,  $\mathbf{w}_k \in \mathbf{H}_k$ , o que implica que  $\mathbf{w}_k \in \Psi_k^P$ . Estando satisfeita essa condição, não há necessidade de atualizar o vetor de coeficientes do filtro.

A segunda hipótese a se analisar é a situação onde o erro entre a saída no instante  $k$  e o sinal de referência resultam em um erro maior que  $\gamma$ . Nessa hipótese,

$\mathbf{w}_k \notin \mathbf{H}_k$ , o que implica que  $\mathbf{w}_k \notin \Psi_k^P$ . Nessa condição, o objetivo deste algoritmo é chegar a um novo vetor de coeficientes  $\mathbf{w}_{k+1}$  que possa garantir um erro inferior a  $\gamma$  de modo a fazer com que  $\mathbf{w}_{k+1} \in \Psi_k^P$ .

Isso é feito de forma que a norma quadrada da distância entre o próximo vetor de coeficientes e o atual seja mínima. Portanto, deve-se minimizar  $\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2$  sujeito à restrição  $\mathbf{d}_k - \mathbf{X}_k^T \mathbf{w}_{k+1} = \mathbf{g}_k$ . Nessa restrição,  $\mathbf{d}_k \in R^{P \times 1}$  e contém as saídas desejadas para os últimos  $P$  instantes de tempo;  $\mathbf{g}_k \in R^{P \times 1}$  e especifica um vetor em  $\Psi_k^P$  que tenha módulo inferior a  $\gamma$ ;  $\mathbf{X}_k \in R^{N \times P}$  e contém os vetores de entrada  $\mathbf{x}_k = [x_k x_{k-1} \dots x_{k-N+1}]^T$ .

Usando multiplicadores de Langrange, chega-se a

$$f(\mathbf{w}_{k+1}) = \|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2 + \lambda_k^T [\mathbf{d}_k - \mathbf{X}_k^T \mathbf{w}_{k+1} - \mathbf{g}_k]. \quad (3.62)$$

Depois de se calcular o gradiente de  $f(\mathbf{w}_{k+1})$  em relação a  $\mathbf{w}_{k+1}$  e se igualar o resultado a zero, obtém-se

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{X}_k \frac{\lambda_k}{2}. \quad (3.63)$$

Lembrando-se das restrições sobre esse problema de minimização, tem-se

$$\mathbf{X}_k^T \mathbf{X}_k \frac{\lambda_k}{2} = \mathbf{d}_k - \mathbf{X}_k^T \mathbf{w}_k - \mathbf{g}_k = \mathbf{e}_k - \mathbf{g}_k. \quad (3.64)$$

A partir da equação (3.64), chega-se à equação de atualização dos coeficientes do algoritmo, representada por

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{X}_k (\mathbf{X}_k^T \mathbf{X}_k)^{-1} (\mathbf{e}_k - \mathbf{g}_k) \text{ se } |\mathbf{e}_k| > \gamma. \quad (3.65)$$

O único parâmetro que ainda não ficou bem definido foi  $\mathbf{g}_k$ . Esse é um vetor qualquer pertencente a  $\Psi_k^P$  cuja única restrição (pelo menos até agora) era ter módulo inferior a  $\gamma$ . Existem infinitos valores válidos. Contudo, duas escolhas serão analisadas.

A escolha mais natural seria  $\mathbf{g}_k = \underline{0}$ , pois, além de 0 ser obviamente menor que  $\gamma$ , usar valores nulos geralmente trazem uma simplicidade ao algoritmo. Pode-se ver que, tomando-se esse caminho, o erro *a posteriori* nos últimos  $P$  instantes de tempo será forçado a ser zero. Isso pode ser verificado pois estar-se-ia fazendo com que  $\mathbf{d}_k - \mathbf{X}_k^T \mathbf{w}_{k+1} = 0$ . Daí, chegar-se-ia a

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{X}_k (\mathbf{X}_k^T \mathbf{X}_k)^{-1} (\mathbf{e}_k) \text{ se } |\mathbf{e}_k| > \gamma \quad (3.66)$$



Essa abordagem é idêntica ao algoritmo convencional de *affine projection* com *step-size* unitário sempre que  $\mathbf{w}_k \in \mathbf{H}_k$ . Contudo, há uma grande redução no custo computacional devido à seletividade da informação.

Uma outra escolha para o valor de  $\mathbf{g}_k$  seria baseada no fato de que  $\mathbf{w}_k$  está necessariamente, por definição, na interseção dos últimos  $P$  conjuntos de restrição, com a exceção do último, (cada um sendo relativo a um instante de tempo). Com isso em mente, pode-se associar os valores dos erros para cada um desses instantes ao vetor  $\mathbf{g}_k$ . Por definição, todos esses erros são menores que o limite estabelecido  $\gamma$ . O que resta agora é escolher o valor do elemento de  $\mathbf{g}_k$  associado ao último instante, ou seja,  $k$ . Isso deve ser feito de modo que o novo vetor de coeficientes caia no ponto pertencente a  $\mathbf{H}_k$  mais próximo da interseção dos outros conjuntos de restrição. Para isso, escolhe-se o valor de  $\mathbf{g}_k = \gamma \text{sign}(e_k)$ , de acordo com [18], o que faz com que

$$\mathbf{X}_k^T \mathbf{X}_k \lambda_k = \alpha_k e_k \mathbf{u}_1 \quad (3.67)$$

onde  $\alpha_k = 1 - (\gamma/|e_k|)$  e  $\mathbf{u}_1 = [1, 0, \dots, 0]^T$ . Nessa equação, houve a absorção do fator meio pois este apenas multiplicava o multiplicador de Lagrange. Isso, então, leva à equação de atualização deste algoritmo dada por

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{X}_k (\mathbf{X}_k^T \mathbf{X}_k)^{-1} \alpha_k e_k \mathbf{u}_1, \quad (3.68)$$

$$\text{onde } \alpha_k = \begin{cases} 1 - (\gamma/|e_k|), & \text{se } |e_k| > \gamma; \\ 0, & \text{em caso contrário.} \end{cases}$$

### 3.6 Análise de componentes independentes

Outra técnica multi-canal é baseada na análise de componentes independentes (ou, da sigla em inglês, ICA - *independent components analysis*). Para entender tal método, considera-se que existam duas fontes de sinal  $s_1$  e  $s_2$ , a princípio sem nenhuma restrição quanto a sua natureza. Agora, considera-se que existam dois sensores. Desprezam-se aqui as diferenças de fase quanto aos sinais que chegam nestes sensores. Em cada sensor, irá chegar um sinal que será uma mistura de  $s_1$  com  $s_2$ . Essas misturas serão denominadas  $x_1$  e  $x_2$ . Contudo, para cada sensor a mistura se dará de forma diferente. O que irá diferenciar essas misturas será a

proporção do sinal de cada fonte. Dessa forma, define-se uma matriz de mistura  $\mathbf{A}$  da forma

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, \quad (3.69)$$

ou

$$\mathbf{x} = \mathbf{A} \cdot \mathbf{s}. \quad (3.70)$$

Se a matriz de mistura  $\mathbf{A}$  fosse conhecida, bastaria resolver o sistema dado pela equação (3.70) segundo algum método como a regra de Kramer ou algo baseado em eliminação gaussiana. Contudo, em um caso real, a matriz de mistura não é conhecida, o que torna o problema bastante complexo.

Segundo [19], surpreendentemente, o único requisito, a princípio, para que este tipo de problema possa ser resolvido é o fato de as fontes de sinal sejam independentes. Será visto mais adiante que as fontes devem apresentar também uma distribuição não-gaussiana. Contudo, essas distribuições são consideradas desconhecidas, de forma que o modelo se torne o mais próximo do real.

O objetivo da ICA é estimar a matriz de mistura  $A$ , para, em seguida, computar sua inversa  $W$  e, assim, ser possível calcular os sinais emitidos pelas fontes independentes  $s_1$  e  $s_2$ . Tudo isso é feito unicamente através das observações das misturas propriamente ditas, ou seja,  $x_1$  e  $x_2$ .

Antes de descrever o funcionamento deste método, é preciso saber suas limitações. Os resultados por ele obtidos apresentam as seguintes ambigüidades:

- Não é possível determinar a variância dos sinais emitidos pelas fontes. Isso pode ser explicado pelo fato de se um desses sinais for multiplicado por um escalar, a matriz de mistura pode ser multiplicada pelo inverso deste escalar de forma a não alterar o sistema como um todo.
- Não é possível determinar a ordem das fontes. Isso pode ser explicado pelo fato de ser possível que permutações sejam feitas de modo que  $x = AP^{-1}Ps$  de modo que o sistema não se altere.

A chave para a estimação do modelo ICA é a não-gaussianidade e, relacionado a esta, o teorema do limite central. Os componentes independentes são considerados

não-gaussianos pois, segundo [19], a matriz de mistura não pode ser estimada se os componentes independentes forem gaussianos. O teorema do limite central diz que a soma de variáveis aleatórias independentes tende a ser uma variável com distribuição gaussiana, sob certas condições. Assim, pode-se dizer que a soma de duas variáveis aleatórias independentes geralmente tem uma distribuição que é mais gaussiana do que qualquer uma das duas variáveis aleatórias originais.

Considera-se agora que o vetor  $\mathbf{x}$  é formado segundo a equação (3.70), ou seja, esse vetor é uma mistura de componentes independentes (que, por simplicidade, terão a mesma distribuição). Para estimar um dos componentes independentes, segue-se a seguinte regra.

1. Considera-se uma combinação linear dos elementos de  $\mathbf{x}$ , denotados por  $x_i$ . Dessa forma, tem-se:  $y = \mathbf{w}^T \mathbf{x} = \sum w_i x_i$ , onde  $\mathbf{w}$  é um vetor a ser determinado. Se  $\mathbf{w}$  for uma das linhas da matriz inversa de  $\mathbf{A}$ , essa combinação linear seria igual a um dos componentes independentes.
2. Define-se  $\mathbf{z} = \mathbf{A}^T \mathbf{w}$ . Isso resulta que  $y = \mathbf{w}^T \mathbf{x} = \mathbf{w}^T \mathbf{A} \mathbf{s} = \mathbf{z}^T \mathbf{s}$ , ou seja,  $y$  é uma combinação linear dos componentes independentes  $s_i$ .
3. Como, pelo teorema do limite central, a soma de duas variáveis aleatórias independentes é mais gaussiana que as variáveis aleatórias,  $\mathbf{z}^T \mathbf{s}$  é mais gaussiana que qualquer um dos  $s_i$ . Além disso, essa soma se torna menos gaussiana se esta se torna igual a um dos  $s_i$ . É claro que, nesse caso, apenas um dos elementos de  $\mathbf{z}$  é não-nulo.
4. Dessa maneira,  $\mathbf{w}$  é o vetor que deve ser estimado maximizando a não-gaussianidade de  $\mathbf{w}^T \mathbf{x}$ . Esse vetor corresponde necessariamente a um vetor  $\mathbf{z}$  que apresenta apenas um elemento não-nulo. Isso significa que  $y = \mathbf{w}^T \mathbf{x} = \mathbf{z}^T \mathbf{s}$  será igual a um dos componentes independentes.

Nesse ponto, então, o problema se torna encontrar uma forma de medir a não-gaussianidade de uma variável aleatória, que, no caso, será  $y$ . As formas mais comuns são:

- Curtose

- Negentropia
- Aproximações para negentropia

### 3.6.1 Curtose

A medida clássica de não-gaussianidade é o curtose ou cumulante de quarta ordem, que é definido por

$$kurt(y) = E[y^4] - 3\{E[y^2]\}^2 \quad (3.71)$$

Se  $y$  tem variância unitária, seu curtose é simplificado para  $(E[y^4] - 3)$ , o que mostra que essa medida é apenas uma versão normalizada do quarto momento  $E[y^4]$ . Se  $y$  for gaussiana, o quarto momento pode ser simplificado para  $3\{E[y^2]\}^2$ . Assim, o curtose é igual a zero para uma variável aleatória gaussiana.

O valor do curtose pode ser positivo ou negativo. Quando é positivo, a variável é dita super-gaussiana e sua pdf tem o pico mais acentuado, como numa distribuição de Laplace. Quando é negativo, a variável é dita sub-gaussiana e sua pdf tem o pico mais achatado, aproximando-se de uma distribuição uniforme.

Tipicamente, a não-gaussianidade é medida pelo valor absoluto do curtose ou por seu valor ao quadrado. A principal razão para o uso desse valor como medida de não-gaussianidade é a simplicidade tanto computacional quanto teórica. Computacionalmente, o curtose pode ser obtido pelo simples uso do quarto momento dos dados amostrais. A análise teórica pode ser simplificada devido às propriedades de linearidade e de multiplicação por escalar, dadas, respectivamente, por

$$kurt(x_1 + x_2) = kurt(x_1) + kurt(x_2) \quad (3.72)$$

$$kurt(\alpha x_1) = \alpha^4 kurt(x_1). \quad (3.73)$$

O funcionamento do curtose para a estimação dos componentes independentes se dá da seguinte forma:

1. Assume-se um modelo de dimensão 2 dado pela equação (3.70), ou seja,  $\mathbf{x} = \mathbf{A}s$ .
2. Assume-se que os componentes independentes  $s_1$  e  $s_2$  são não gaussianos e, portanto, apresentam valores de curtose diferentes de zero.

3. O objetivo é estimar um dos componentes independentes, que é dado por  $y = \mathbf{w}^T \mathbf{x}$ .
4. Realizando-se as transformações propostas anteriormente, tem-se que  $y = \mathbf{w}^T \mathbf{x} = \mathbf{w}^T \mathbf{A} \mathbf{s} = \mathbf{z}^T \mathbf{s} = z_1 s_1 + z_2 s_2$ .
5. Com base, então, nas propriedades do curtose, tem-se que  $kurt(y) = kurt(z_1 s_1) + kurt(z_2 s_2) = z_1^4 kurt(s_1) + z_2^4 kurt(s_2)$ .
6. Assume-se que a variância de  $y$  é unitária, o que implica que há mais uma restrição sobre  $\mathbf{z}$ :  $E[y^2] = z_1^2 + z_2^2 = 1$ . Geometricamente, isso significa que o vetor está no círculo unitário do plano bidimensional. O problema se reduz então a: qual é o máximo da função  $|kurt(y)| = |z_1^4 kurt(s_1) + z_2^4 kurt(s_2)|$  sobre o círculo unitário?
7. Pode-se mostrar que os máximos dessa função são os pontos onde exatamente um dos elementos do vetor  $\mathbf{z}$  é zero e os outros são não-nulos. Devido à restrição de os pontos deverem estar sobre o círculo unitário, o valor não-nulo só poderá ser 1 ou  $-1$ . Esses pontos são exatamente aqueles onde  $y$  é igual a um dos componentes independentes com exceção do sinal, ou seja,  $y = \pm s_i$  e o problema foi resolvido.

Na prática, usa-se algum algoritmo de otimização sobre a superfície de desempenho dada pelo curtose de  $y$  para se atualizar os valores dos elementos do vetor  $\mathbf{w}$ . O principal problema do curtose é sua grande sensibilidade a pontos fora da curva (os chamados *outliers*). Por causa disso, o valor do curtose pode depender só de alguns poucos pontos no fim das caudas da distribuição, os quais podem ser errôneos ou irrelevantes. Desse modo, vê-se que o curtose não é uma medida robusta de não-gaussianidade.

### 3.6.2 Negentropia

A negentropia é baseada na entropia, conceito básico da teoria da informação [20] e que pode ser definida como

$$H(Y) = - \sum_i P(Y = a_i) \log P(Y = a_i), \quad (3.74)$$

onde  $a_i$  são os possíveis valores de  $Y$ . Essa definição pode ser generalizada para variáveis aleatórias contínuas. Nesse caso, define-se a entropia diferencial  $h$  de uma variável aleatória contínua  $y$  com densidade  $f(y)$  de forma

$$h(y) = - \int f(y) \log f(y) dy. \quad (3.75)$$

Um resultado fundamental de teoria da informação é o fato de a distribuição gaussiana apresentar a maior entropia diferencial entre todas as distribuições de mesma variância. Dessa forma, a entropia diferencial pode então ser usada como uma medida de não-gaussianidade. Para se obter uma medida de não gaussianidade que seja zero para uma variável gaussiana e sempre não-negativa, é comum usar uma variação da definição de entropia diferencial denominada negentropia, definida por

$$J(y) = H(y_{gauss}) - H(y), \quad (3.76)$$

onde  $y_{gauss}$  é uma variável aleatória gaussiana com a mesma matriz de covariância de  $y$ . Devido às propriedades acima mencionadas, a negentropia é sempre não-negativa e é zero só se  $y$  apresentar distribuição gaussiana.

A negentropia tem as vantagens de ser invariante para transformações lineares inversíveis e ser o estimador ótimo de não-gaussianidade, segundo as propriedades estatísticas [19]. O problema de se usar negentropia para esse fim é a grande dificuldade computacional, pois é necessário estimar a pdf da variável que se deseja medir. Assim, torna-se necessário utilizar aproximações para a negentropia que sejam mais simples computacionalmente.

É ainda possível mostrar que maximizando a negentropia é equivalente a se minimizar a informação mútua entre os componentes independentes. Isso pode ser justificado pelo fato de que a informação mútua é uma medida de independência entre variáveis aleatórias. Se duas variáveis aleatórias são independentes, a informação mútua entre elas é zero. Tal medida corresponde à conhecida distância de Kullback-Leibler que se calcula entre as densidades de probabilidade de variáveis aleatórias.

### 3.6.3 Aproximações para negentropia

A método clássico para aproximação da negentropia é usar momentos de ordem superior, como, por exemplo [19]

$$J(y) \approx \frac{1}{12} E [y^3]^2 + \frac{1}{48} kurt^2(y), \quad (3.77)$$

Contudo, essa aproximação sofre da mesma falta de robustez encontrada na utilização do curtose. Para contornar esse problema, novas aproximações foram desenvolvidas baseadas no princípio da máxima entropia, chegando-se a

$$J(y) \approx \sum_{i=1}^p k_i \{E [G_i(y)] - E [G_i(v)]\}^2, \quad (3.78)$$

onde  $K_i$  são constantes positivas,  $v$  é uma variável gaussiana de médio zero e variância unitária. A variável aleatória  $y$  também é considerada como sendo de média zero e variância unitária e as funções  $G_i$  são funções não-quadráticas. No caso de se usar apenas uma função não-quadrática  $G$ , essa aproximação é dada por

$$J(y) \propto \{E [G(y)] - E [G(v)]\}^2. \quad (3.79)$$

Escolhendo a função não-quadrática  $G$  convenientemente, pode-se obter aproximações bem robustas para a negentropia. Particularmente, selecionando-se  $G$  de forma que não haja um crescimento muito acelerado, pode-se chegar a estimadores bastante robustos. As escolhas dadas abaixo tem mostrado bons resultados segundo [19]:

$$G_1(u) = \frac{1}{a_1} \log (\cosh (a_1 u)) \quad (3.80)$$

$$G_2(u) = -\exp(-u^2/2), \quad (3.81)$$

onde  $1 \leq a_1 \leq 2$ . Dessa forma, é possível construir uma medida de não-gaussianidade que apresenta um bom compromisso entre a simplicidade de cálculo e a robustez dos dois métodos clássicos dados por curtose e negentropia.

### 3.6.4 Pré-processamento

Para facilitar o cálculo do modelo ICA, é necessário que seja feito algum tipo de pré-processamento. Essa etapa é dividida fundamentalmente em duas: centralização e branqueamento.

A centralização força que as variáveis de observação, denotadas por  $x_i$ , tenham médias iguais a zero. Isso pode ser conseguido através da simples subtração da média. Isso não significa que a média deva ficar de fora do resultado final. Esta pode ser somada ao resultado final. Considera-se que o vetor de médias de  $\mathbf{x}$  seja o vetor  $\mathbf{m}$ . Quando já se tiver uma estimativa de  $\mathbf{A}$ , pode-se somar a média transformada  $\mathbf{A}^{-1}\mathbf{m}$  aos componentes independentes  $s_i$ .

O branqueamento faz com que o vetor de observação  $\mathbf{x}$  seja transformado para  $\tilde{\mathbf{x}}$ , que, por sua vez, possui os elementos descorrelacionados e cada um com variância unitária, o que pode ser denotado por  $E[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T] = \mathbf{I}$ .

Uma das formas mais populares de se realizar o pré-processamento é através da decomposição em auto-valores da matriz de autocorrelação, ou seja,  $E[\mathbf{x}\mathbf{x}^T] = \mathbf{V}\mathbf{D}\mathbf{V}^T$ . Faz-se então que  $\tilde{\mathbf{x}} = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T\mathbf{x}$ . Dessa forma, pode-se provar que  $E[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T] = \mathbf{I}$  da seguinte maneira:

$$\begin{aligned}\tilde{\mathbf{x}} &= \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T\mathbf{x} \\ E[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T] &= E[\mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T\mathbf{x}\mathbf{x}^T\mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T] \\ &= \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T E[\mathbf{x}\mathbf{x}^T] \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T \\ &= \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T \mathbf{R} \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T \\ &= \mathbf{R}^{-1/2} \mathbf{R} \mathbf{R}^{-1/2} \\ &= \mathbf{I}\end{aligned}$$

É também possível provar que a matriz de mistura que se obtém após o branqueamento é ortogonal. Dessa forma, executando o branqueamento, o número de parâmetros a serem estimados é reduzido. Ao invés de se ter que calcular  $n^2$  parâmetros para a matriz original  $\mathbf{A}$ , deve-se, agora, calcular apenas  $n(n-1)/2$  parâmetros da nova matriz  $\tilde{\mathbf{A}}$  (estimativa da matriz  $\mathbf{A}$ ), que é ortogonal.

### 3.6.5 Algoritmo de *FastICA*

O primeiro passo a ser dado é selecionar uma função não-linear  $g$ , que será a derivada da função não-quadrática  $G$  usada em (3.79). Pode-se usar as derivadas das funções dadas pelas equações (3.80) ou (3.81) para dar uma aproximação mais robusta da negentropia. Alternativamente, pode-se usar derivadas correspondentes



à quarta potência, como no curtose. Assim, pode-se escolher uma das seguintes funções:

$$g_1(y) = \tanh(a_1 y) \Rightarrow g_1'(y) = a_1(1 - \tanh^2(a_1 y)) \quad (3.82)$$

$$g_2(y) = y \cdot \exp(-y^2/2) \Rightarrow g_2'(y) = (1 - y^2) \cdot \exp(-y^2/2) \quad (3.83)$$

$$g_3(y) = y^3 \Rightarrow g_3'(y) = 3y^2 \quad (3.84)$$

Assim, tem-se o seguinte algoritmo

1. Escolher um valor inicial para o vetor  $\mathbf{w}$  com norma unitária.
2. Fazer  $\mathbf{w}^+ = E [\tilde{\mathbf{x}}g(\mathbf{w}^T \tilde{\mathbf{x}})] - E [g'(\mathbf{w}^T \tilde{\mathbf{x}})] \mathbf{w}$ .
3. Fazer  $\mathbf{w} = \mathbf{w}^+ / \|\mathbf{w}^+\|$ .
4. Se não convergiu, voltar ao passo 2.

Contudo, esse algoritmo tem como objetivo a extração de apenas um componente independente. Para extrair todos, é necessário que cada vetor  $\mathbf{w}$  seja decorrelacionado após cada iteração do algoritmo. Isso deve ser feito para garantir que cada vetor convirja para um máximo diferente. Para isso, pode-se utilizar, por exemplo, o método de ortogonalização de Gram-Schmidt. Nesse método, estimam-se os componentes independentes um a um e quando se obtiver  $p$  componentes, calcula-se o vetor de índice  $(p + 1)$  subtraindo deste as projeções dos demais vetores sobre ele, seguido de uma normalização:

1. Fazer com que  $\mathbf{w}_{p+1} = \mathbf{w}_{p+1} - \sum_{j=1}^p \mathbf{w}_{p+1}^T \mathbf{w}_j \mathbf{w}_j$ .
2. Fazer  $\mathbf{w}_{p+1} = \mathbf{w}_{p+1} / \sqrt{\mathbf{w}_{p+1}^T \mathbf{w}_{p+1}}$ .

Contudo, em algumas aplicações, deseja-se que nenhum componente seja privilegiado sobre outros, como acontece quando se usa Gram-Schmidt, onde um componente é calculado e os demais dependem deste. Quando, se busca uma maior simetria entre os componentes independentes, pode-se usar o seguinte método:

1. Fazer com que  $\mathbf{W}$  seja a matriz que possui os vetores  $w$  em suas linhas.
2. Fazer  $\mathbf{W} = \mathbf{W} / \sqrt{\|\mathbf{W}\mathbf{W}^T\|}$ .

3. Fazer  $\mathbf{W} = \frac{3}{2}\mathbf{W} - \frac{1}{2}\mathbf{W}\mathbf{W}^T\mathbf{W}$ .
4. Repetir passo 3 até a convergência.

Esse algoritmo FastICA [19] tem as seguintes propriedades:

1. A convergência é cúbica (no mínimo, quadrada).
2. Não há parâmetros livres do tipo *step-size*, o que facilita seu uso.
3. O algoritmo encontra os componentes independentes usando-se qualquer função não-linear  $g$ , ao contrário de outros métodos que precisam de estimativas de pdf's e funções não-lineares que estejam de acordo com tais estimativas.
4. Os componentes independentes podem ser calculados um a um, o que ajuda na redução do custo computacional.

Sua única desvantagem é o fato de seu desempenho ser fraco em ambientes não estacionários, onde uma adaptatividade mais rápida é necessária.

### 3.6.6 *Blind source separation (BSS)*

Um dos objetivos da análise de componentes independentes é a separação de fontes. Isso pode ser usado para separar um sinal de interesse de alguma espécie de ruído aditivo. É claro que, nesse caso, assume-se que o ruído é independente do sinal, o que é geralmente o caso.

Contudo, devido às limitações citadas, alguns cuidados precisam ser tomados:

1. Como não se sabe qual é a ordem na qual os componentes independentes serão calculados, é necessário verificar qual é aquele que se aproxima mais de um ruído branco. Isso pode ser feito comparando-se os vetores de autocorrelação. O que se aproximar mais de um impulso em zero será o ruído. Assim, o outro componente será o sinal de interesse.
2. Como a informação sobre amplitude é perdida, é necessário associar algum outro método de supressão de ruído que apresente pouca redução de amplitude, para que o sinal seja totalmente recuperado.

## 3.7 Conclusão

Neste capítulo, foi feita uma revisão geral sobre várias ferramentas usadas na supressão de ruído (ou *denoise*) como a subtração espectral (Seção 3.2), o filtro de Wiener (Seção 3.3), a transformada *wavelet* (3.4), a filtragem adaptativa (Seção 3.5) e a análise de componentes independentes (Seção 3.6).

Suas vantagens e desvantagens podem ser resumidas na Tabela 3.1.

Tabela 3.1: Vantagens e desvantagens dos métodos de supressão de ruído estudados.

Método	Vantagens	Desvantagens
<b>subtração espectral</b>	Utiliza apenas um canal. Não é necessária qualquer informação sobre o sinal propriamente dito.	É necessário estimar o período de silêncio. O <i>denoising</i> depende da estimativa do silêncio.
<b>filtragem de Wiener</b>	Utiliza apenas um canal. Não é necessária qualquer informação sobre o sinal propriamente dito. Além disso, o filtro atua de forma mais intensa de acordo com a SNR.	É necessário estimar o período de silêncio. O <i>denoising</i> depende da estimativa do silêncio.
<i>wavelet shrinkage</i>	Utiliza apenas um canal. Não é necessário estimar o espectro do ruído.	É necessário ter uma noção da forma do sinal para seleção da <i>wavelet</i> -mãe. Além disso, a seleção dos limiares ainda é uma grande dificuldade.
<b>filtragem adaptativa</b>	Não é necessário ter informações sobre o sinal ou sobre o ruído.	São necessários dois canais, é um método iterativo e é dependente da estacionariedade do processo.
<b>ICA</b>	Se o ruído é totalmente independente do sinal, a separação é próxima da ótima.	São necessários dois canais, além de não fornecer informação sobre amplitude.

# Capítulo 4

## Resultados

### 4.1 Introdução

Esse capítulo tem como objetivo apresentar os resultados obtidos ao se empregarem os métodos de supressão de ruído discutidos no Capítulo 3. A Seção 4.2 discute os critérios de avaliação utilizados para verificar a eficiência do sistema. A Seção 4.3 relata como foi feita a aquisição dos sinais usados para obter tais resultados. Os resultados obtidos com o filtro de Wiener e com a subtração espectral são mostrados na Seção 4.4. Os resultados referentes ao uso da transformada *wavelet* são discutidos na Seção 4.5 e os referentes à filtragem adaptativa, na Seção 4.6. É ainda feita uma análise qualitativa sobre o uso da separação cega baseada em ICA (Seção 4.7) e sobre a combinação de métodos (Seção 4.8). A comparação de todos os resultados é feita na Seção 4.9.

### 4.2 Critérios de avaliação

Os critérios usados para avaliar o desempenho do sistema são os mesmos empregados em [9]. A utilização de apenas um critério, seja este qual for, não fornece informação suficiente para se ter uma completa noção da eficiência do sistema. Dessa forma, usaram-se, de forma integrada, a SNR, a correlação cruzada e a taxa de redução de amplitude.

Se a SNR for usada individualmente, não se tem informação sobre a forma de onda do sinal processado. Se a correlação cruzada for usada sozinha, não se tem

informação sobre a amplitude do sinal processado, o que faz com que seja necessário recorrer à taxa de redução de amplitude.

### 4.2.1 SNR

Essa medida bem conhecida de quanto o sinal de interesse está corrompido é calculada pela razão entre a potência do sinal e a potência do ruído, ou seja,

$$SNR = 10 \cdot \log \left\{ \frac{\sum_{k=1}^L s^2(k)}{\sum_{k=1}^L [x(k) - s(k)]^2} \right\}, \quad (4.1)$$

onde  $x(k)$  é o sinal corrompido, formado pela adição do sinal original  $s(k)$  a um ruído aditivo  $n(k)$ , e  $L$  é o comprimento total do sinal. A unidade é dB.

### 4.2.2 Correlação cruzada

Essa medida, em sua forma normalizada, é descrita por

$$R = \frac{1}{L} \cdot \frac{\sum_{n=1}^L \{ [Y(n) - E[Y]] \cdot [S(n) - E[S]] \}}{\sigma_Y \cdot \sigma_S} \quad (4.2)$$

onde  $E[\cdot]$  denota o operador valor esperado,  $\sigma_Y$  e  $\sigma_S$  denotam os desvios padrões do sinal processado e do sinal ideal. O seu valor varia no intervalo  $-1 \leq R \leq 1$ .

### 4.2.3 Taxa de redução de amplitude

Essa medida é a diferença percentual entre o pico máximo contido no sinal original e o valor da amostra correspondente no sinal processado. Isso denota a quantidade de sinal que é perdida no processo de supressão de ruído.

## 4.3 Geração e aquisição de sinais

Os sinais utilizados para os testes de desempenho neste trabalho foram gerados artificialmente com base no princípio de que o sinal de descargas parciais é a corrente medida numa impedância de detecção quando esta for submetida a uma tensão que se aproxime de um impulso. A impedância de detecção considerada no presente trabalho é um transformador de corrente (ou TC). Outros tipos de impedância poderiam ter sido usados, mas, para simplificar os testes, decidiu-se usar

apenas o TC (fornecido pelo fabricante do ICMSystem, Power Diagnostix), devido ao fato de que tal tipo de impedância é de fácil acesso no Cepel e é, de certa forma, padronizada.

Dessa forma, através de um gerador de sinais, obteve-se um trem de pulsos com frequência de 60 Hz. O uso dessa frequência é devido ao fato de proporcionar um pulso por ciclo de potência da rede elétrica. Esse pulso apresentava uma amplitude de 10 volts, para que o ruído presente no laboratório pudesse ser bem menor que o sinal. A largura do sinal foi ajustada para 20 ns, o que é consideravelmente menor que a largura do sinal de DP [9].

Dessa forma, obteve-se um sinal como na Figura 4.1. Esse sinal foi adquirido com um osciloscópio digital da Tektronix TDS460. Este equipamento permite uma aquisição de 120.000 pontos. Como um ciclo de potência apresenta 16,67 ms, foi tomada uma janela de 24 ms. Isso significa uma frequência de amostragem de 5 MHz. Para se ter um padrão mais limpo, após a aquisição, as amostras que não correspondiam ao sinal de DP propriamente dito foram zeradas. Isso, no exemplo, corresponde às amostras antes da de índice 60000 e das amostras após a de índice 60025.

A partir deste sinal, foi gerado um trem de pulsos de DP, assim como proposto em [21]. Para isso, copiou-se o sinal de DP 5 vezes e essas cópias foram colocadas com espaçadas de 10  $\mu$ s duas a duas. Dessa forma, obteve-se o sinal visto na figura 4.2.

Como visto no Capítulo 2, o ruído que corrompe o sinal de DP consiste basicamente de ruído de fundo de natureza puramente estocástica, interferência de espectro discreto (DSI) e interferências impulsivas. Assim como em [9], os testes serão feitos apenas com os dois primeiros tipos de ruído. Para gerar o sinal corrompido, o ruído foi modelado segundo [21]. Para isso, foram executados os seguintes passos:

1. Gerar os sinais de DSI. Estes consistem em 8 sinais de AM com 40% de modulação cuja informação é uma senóide de 1 kHz. A razão entre as potências de qualquer DSI e a do ruído de fundo é denominada razão interferência-ruído ou INR. Optou-se por definir suas frequências portadoras de forma mais distribuída ao longo da faixa de frequência disponível. Por isso, as frequências

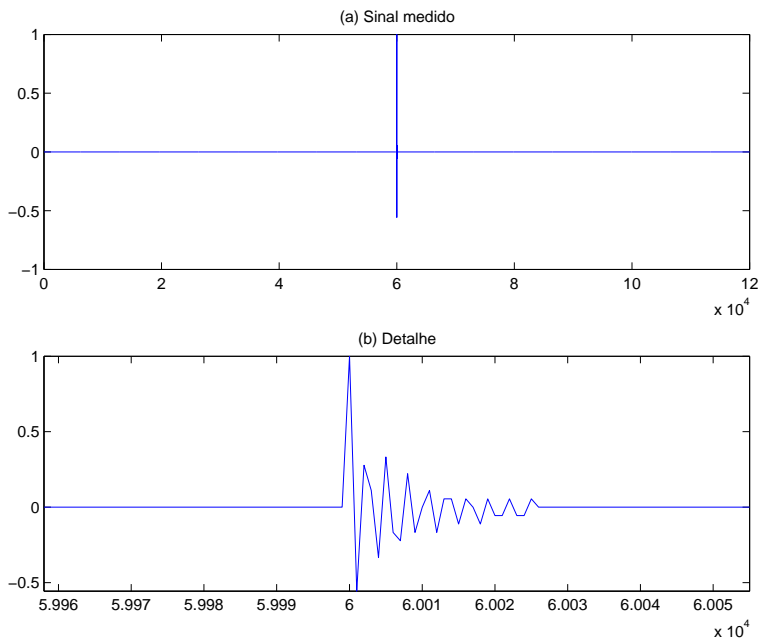


Figura 4.1: (a) Sinal medido pelo osciloscópio digital TDS460, com 120.000 pontos numa frequência de amostragem de 5 MHz. (b) Detalhe do sinal mostrado em (a).

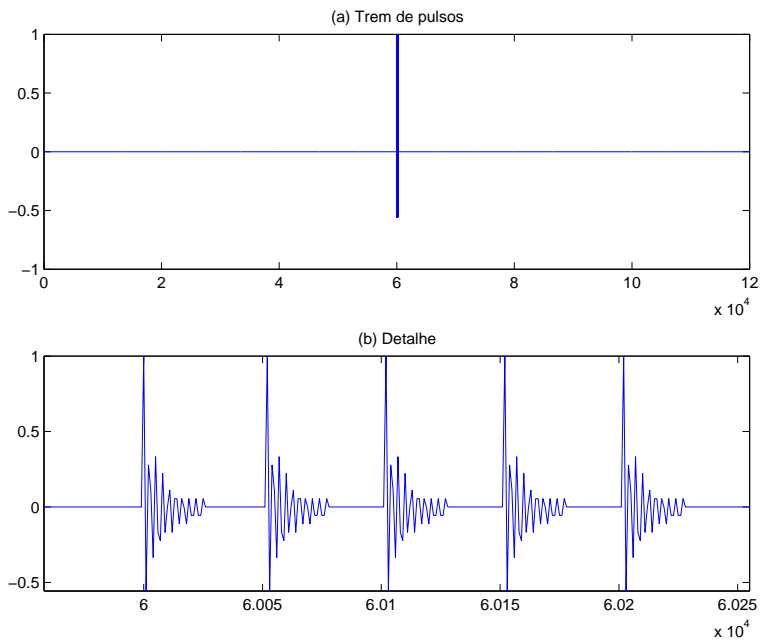


Figura 4.2: (a) Trem de pulsos gerado por 5 cópias do sinal de DP. (b) Detalhe do sinal mostrado em (a).

variam de 200 Hz a 90% da metade da frequência de amostragem.

2. Adicionar ruído branco gaussiano de variância a ser especificada em cada experimento, para simular o ruído do ambiente externo.

## 4.4 Resultados da filtragem de Wiener

A filtragem de Wiener pode ser vista como uma evolução da subtração espectral, pois é uma versão desse método ponderada pela SNR. Desse modo, a subtração espectral será avaliada como um caso particular da filtragem de Wiener.

### 4.4.1 Ajuste dos parâmetros

A princípio, dois parâmetros devem ser ajustados: a largura da janela que é usada para a estimativa do espectro do ruído e o tamanho da janela do sinal a tratar. O primeiro parâmetro determina a precisão da estimativa do ruído que corrompe o sinal. Quanto mais larga, mais precisa será esta estimativa. Contudo, a partir de um certo ponto, não será mais necessário aumentá-la, pois não haverá mais influência no resultado final. Do mesmo modo, ajusta-se o tamanho da janela. Este influencia na ordem do filtro e na precisão da estimativa do espectro do sinal. É claro que, aqui, o objetivo é buscar os menores valores possíveis, para aliviar o processamento.

Para isso, testa-se o sistema de supressão de ruído através da filtragem de Wiener usando-se uma faixa de valores para a largura do intervalo de estimativa do ruído e outra para a largura da janela de sinal. Como existem dois tipos principais de ruído, ou seja, ruído branco e DSI, os parâmetros do sistema devem ser ajustados para cada caso.

Para o caso do ruído branco, fez-se com que o ruído branco apresentasse um terço da “amplitude” do sinal de interesse, segundo é feito em [21]. Nesse caso, amplitude significa três desvios padrões (considerando que o ruído apresenta distribuição gaussiana). Para o caso do DSI, fez-se com que cada interferência AM apresentasse amplitude igual à do sinal.

Primeiramente, estudou-se o comportamento do sistema ao se tratar ruído branco. Testou-se o sistema para janelas de 100, 1000 e 10000 amostras. Para cada uma dessas situações, usaram-se janelas para estimação do ruído de 1000, 5000,



10000, 20000, 30000, 40000 e 50000 amostras. Para esses casos, a SNR original (do sinal corrompido) era de -21,105 dB e a correlação cruzada original era de 0,088. Para uma janela de 100 amostras, sem forçar o espectro a ser constante, a SNR do sinal processado foi sempre de -17,862 dB, a correlação cruzada foi sempre 0,12 e a redução de amplitude foi de 8,89 %. Para uma janela de 100 amostras, se o espectro for forçado a ser constante, a SNR do sinal processado foi sempre de -14,972 dB, a correlação cruzada foi sempre 0,16 e a redução de amplitude foi de 11,30 %. Para uma janela de 1000 amostras, sem forçar o espectro a ser constante, a SNR do sinal processado foi sempre de -17,028 dB, a correlação cruzada foi sempre 0,12 e a redução de amplitude foi de 19,87 %. Para uma janela de 1000 amostras, se o espectro for forçado a ser constante, a SNR do sinal processado foi sempre de -14,972 dB, a correlação cruzada foi sempre 0,16 e a redução de amplitude foi de 11,30 %. Os resultados obtidos quando se usam 10000 amostras para a janela de estimativa do ruído estão mostrados nas Tabelas 4.1 e 4.2.

Analisando-se os resultados para ruído branco, percebe-se que, se a janela para estimativa do espectro do ruído for muito curta, o fato de se forçar o espectro estimado do ruído a ser constante tende a melhorar o desempenho da supressão do ruído. Isso foi deduzido devido a uma maior SNR e uma maior correlação cruzada depois do processamento. À medida que se aumenta a janela para estimativa do espectro do ruído, já não há tanta influência em forçar o espectro do ruído a ser constante. Outra conclusão importante, para o caso de se tratar ruído branco, é o fato de o tamanho da janela para estimativa do espectro do sinal não exercer grande influência no resultado final.

O próximo passo seria tratar o ruído formado por DSI. Para isso, repete-se o que foi feito em relação ao ruído branco, isto é, testes com janelas de 100, 1000 e 10000 amostras para estimação do espectro do sinal. Para cada uma dessas situações, usaram-se janelas para estimação do ruído de 1000, 5000, 10000, 20000, 30000, 40000 e 50000. Para esses casos, a SNR original (do sinal corrompido) era de -46,642 dB e a correlação cruzada original era de 0,0097. Para uma janela de 100 amostras, sem forçar o espectro a ser constante, a SNR do sinal processado foi sempre de -38,631 dB, a correlação cruzada foi sempre 0,0096 e a redução de amplitude foi de 37,261 %. Para uma janela de 1000 amostras, sem forçar o espectro a ser constante, a SNR

Tabela 4.1: Comportamento da filtragem de Wiener ao tratar ruído branco apresentando uma janela para estimativa do espectro do sinal com 10000 amostras. A SNR original (do sinal corrompido) é de -21,105 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,0879.

<b>Largura da janela de ruído</b>	<b>SNR processado</b>	<b>Corr. processado</b>	<b>Red. amplitude</b>
1000	-20,567200 dB	0,093262	-4,680500
5000	-18,876600 dB	0,092233	14,494200
$\geq 10000$	-16,736800 dB	0,087252	37,296800

Tabela 4.2: Comportamento da filtragem de Wiener ao tratar ruído branco apresentando uma janela para estimativa do espectro do sinal com 10000 amostras, forçando o espectro do ruído a ser constante. A SNR original (do sinal corrompido) é de -21,105 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,0879.

<b>Largura da janela de ruído</b>	<b>SNR processado</b>	<b>Corr. processado</b>	<b>Red. amplitude</b>
1000	-20,567200 dB	0,093262	-4,680500
5000	-18,876600 dB	0,092233	14,494200
10000	-16,736800 dB	0,087252	37,296800
$\geq 20000$	-16,700300	0,087705	36,105900

do sinal processado foi sempre de  $-27,749$  dB, a correlação cruzada foi sempre  $0,023$  e a redução de amplitude foi de  $64,79$  %. Para uma janela de  $10000$  amostras, os resultados obtidos estão mostrados na Tabela 4.3.

A primeira conclusão deste estudo é o fato de que a supressão de ruído é muito prejudicada quando a janela de estimativa de ruído (ou do espectro do sinal) é curta. A causa disso é que, para estimar o espectro de um ruído formado por DSI, é preciso que estejam contidos, dentro da janela, vários ciclos de senóides. Dessa forma, uma estimativa mais precisa requer uma janela mais longa.

Percebe-se que o filtro de Wiener tem grande capacidade de tratar DSIs. Vê-se que o melhor é usar uma janela de  $10000$  amostras e estimar o ruído com uma janela de  $20000$  amostras. Esse tamanho para a janela também leva em consideração os resultados obtidos para ruído branco. Contudo, como foram testadas poucas janelas de diferentes comprimentos, faz-se necessário um estudo mais detalhado em relação à forma com a qual o filtro de Wiener trata a DSI. Na Tabela 4.4, esse estudo é feito usando-se uma janela de  $20000$  amostras para a estimativa do espectro do ruído.

Primeiramente, percebe-se, ao analisar a Tabela 4.4, que há uma certa periodicidade para os valores de pico. Isso ocorre em  $5000$ ,  $10000$ ,  $15000$  e  $20000$ . Nessas situações, a “grade” de pontos dos espectros usados no cálculo do filtro de Wiener coincide com os picos do espectro do ruído, o que faz com que esses sejam anulados por aqueles, fazendo com que o desempenho do filtro seja melhorado. Nas demais situações, os picos devem cair em algum lugar entre dois pontos consecutivos, o que faz com que não seja possível um bom alinhamento, piorando o desempenho do filtro. Dessa forma, percebe-se que uma boa configuração é  $15000$  amostras para a janela de estimativa do sinal e  $20000$  amostras para o intervalo de estimativa do ruído.

#### **4.4.2 Testes de desempenho**

Existem basicamente dois tipos de ruído: o ruído branco e a DSI, constituído por sinais de modulação AM ruidosos. Como o filtro de Wiener se mostrou muito mais adequado para a supressão de ruído formado por DSI, os testes de desempenho serão focados neste tipo de interferência. A metodologia a ser empregada aqui é

Tabela 4.3: Comportamento da filtragem de Wiener ao tratar DSI apresentando uma janela para estimativa do espectro do sinal com 10000 amostras A SNR original (do sinal corrompido) é de -46,105 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,009736.

<b>Largura da janela de ruído</b>	<b>SNR processado</b>	<b>Corr. processado</b>	<b>Red. amplitude</b>
1000	-46,811500 dB	0,007901	-43,007000
5000	-43,584000 dB	0,010722	3,110900
$\geq 10000$	1,426400 dB	0,648080	60,676300

primeiramente variar a INR referente a cada componente DSI, o que é mostrado na Tabela 4.5. Depois de analisar o comportamento do sistema neste ambiente, varia-se a amplitude da DSI, o que é visto na Tabela 4.6.

Ao se variar a INR dos sinais de DSI, deduz-se que, logicamente, quanto maior a INR, mais eficaz é a supressão de ruído. Esta foi bem sucedida até chegar a INR a 10 dB. Abaixo desse patamar, a correlação cruzada após o processamento começou a diminuir. Contudo, o processamento só apresentou uma melhora perceptível quando a INR foi maior ou igual a 20 dB. Nesse nível de ruído, a correlação cruzada após o processamento chegou a 4 vezes o que era originalmente.

O próximo teste a ser feito foi, tomar o sinal de DSI com INR igual a 20 dB e variar sua amplitude. Verifica-se que, mesmo com a amplitude do sinal de DSI aumentada em 2 vezes, é possível, através da filtragem de Wiener, aumentar a correlação cruzada em cerca de 4 vezes.

Por fim, testa-se, como caso particular, a subtração espectral. Como o filtro de Wiener se mostrou propício à remoção de DSI, testa-se a subtração espectral com DSI de INR igual a 50 dB. O resultado pode ser resumido na Tabela 4.7. Pode-se ver que o filtro de Wiener propriamente dito é bem superior à subtração espectral. Deve-se lembrar que, nesta configuração, o sinal corrompido apresenta SNR inicial igual a -46,6425 dB e correlação cruzada inicial igual a 0,00972. Essa grande diferença pode ser explicada devido ao fato de que o filtro de Wiener pode ser visto como uma subtração espectral ponderada pela razão sinal ruído. Dessa

Tabela 4.4: Estudo mais detalhado do comportamento da filtragem de Wiener ao tratar DSI apresentando uma janela para estimativa do ruído com 20000 amostras. A SNR original (do sinal corrompido) é de -46,6424 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,009736.

<b>Largura da janela</b>	<b>SNR processado</b>	<b>Corr. processado</b>	<b>Red. amplitude</b>
5000	-2,437500	0,409920	53,807800
6000	-28,268900	0,021646	57,134700
7000	-16,206500	0,160110	-3,330100
8000	-26,405000	0,045150	38,503800
9000	-25,384600	0,038059	43,036300
10000	1,426400	0,648080	60,676300
11000	-25,064600	0,034295	44,187800
12000	-24,991200	0,031222	42,549300
13000	-22,741500	0,057111	38,106400
14000	-7,790000	0,329760	29,649400
15000	4,716200	0,816570	46,209800
16000	-21,797600	0,051948	53,348600
17000	-21,414700	0,060537	51,990300
18000	-21,983700	0,053771	26,685100
19000	-21,603900	0,066546	33,008500
20000	4,280300	0,791750	44,954500
30000	-38,710200	0,012506	8,879300
40000	-42,162400	0,010160	-28,176000
50000	-43,232000	0,010014	-35,975900

forma, ele é mais intenso onde o sinal é mais fraco e vice-versa, o que lhe dá maior flexibilidade.

## 4.5 Resultados da transformada *wavelet*

Esse é o método mais encontrado na literatura especializada para a solução do problema de supressão de ruído em sinais de DP, como visto em [8, 13, 22–26].

### 4.5.1 Ajuste dos parâmetros

O primeiro passo a ser tomado é definir qual família *wavelet* utilizar. Como já foi visto, isso depende da aplicação em questão. Para isso, a seguinte metodologia foi empregada:

1. A primeira família a ser avaliada foi a Daubechies, que é mencionada em diversos artigos como sendo a mais adequada para esse tipo de problema.
2. Definiu-se a ordem da *wavelet* como sendo, a princípio, igual a 1 (o que, no caso da Daubechies, corresponde à família Haar).
3. O número de níveis de decomposição foi definido como o maior inteiro menor que o logaritmo na base 2 do número de amostras do sinal de entrada. Isso pode ser justificado pelo fato de que um sinal de  $N$  amostras após um nível de decomposição (na forma de oitavas) será dividido em dois sinais de  $N/2$  amostras. Isso é devido à decimação inerente ao processo de decomposição. Assim, pode-se ver que o número máximo de decomposições é o maior inteiro menor que o logaritmo na base 2 do número de amostras. Como o sinal de entrada neste trabalho apresenta 100 mil pontos, o número de níveis de decomposição deve ser igual a 16, o que indica que o número de bandas no domínio *wavelet* será igual a 17.
4. Definiu-se o sinal de entrada como sendo o sinal de DP.
5. Tomou-se cada banda *wavelet* individualmente e, para cada uma delas, executou-se a transformada *wavelet* inversa. Assim, pôde-se medir a correlação cruzada

Tabela 4.5: Comportamento da filtragem de Wiener ao tratar DSI contaminado com ruído branco apresentando uma janela para estimativa do ruído com 20000 amostras.

<b>INR</b>	<b>SNR original</b>	<b>SNR processado</b>	<b>Corr. original</b>	<b>Corr. processado</b>	<b>Red. amplitude</b>
100	-46,642400	4,716300	0,009736	0,816570	46,210700
90	-46,642400	4,716100	0,009736	0,816570	46,218400
80	-46,642400	4,713400	0,009736	0,816470	46,221900
70	-46,642400	4,679600	0,009735	0,815070	46,260000
60	-46,642400	4,294300	0,009745	0,799170	46,271300
50	-46,642400	1,954700	0,009734	0,692150	46,365400
40	-46,643100	-4,762000	0,009721	0,378950	44,883100
30	-46,647100	-13,301500	0,009837	0,135330	49,481100
20	-46,683000	-22,328500	0,009834	0,040838	35,491700
10	-47,050100	-32,096500	0,009982	0,019419	66,618300
0	-49,673000	-41,963200	0,007303	0,006515	98,189800

Tabela 4.6: Comportamento da filtragem de Wiener ao tratar DSI contaminado com ruído branco apresentando uma janela para estimativa do ruído com 20000 amostras. INR = 20 dB.

<b>Fator</b>	<b>SNR original</b>	<b>SNR processado</b>	<b>Corr. original</b>	<b>Corr. processado</b>	<b>Red. amplitude</b>
0,1	-26,685900	-2,843400	0,051530	0,543270	23,120900
0,2	-32,710900	-8,516600	0,028221	0,272380	37,028600
0,5	-40,666600	-16,393800	0,014514	0,096668	50,425100
0,8	-44,750700	-20,452500	0,010984	0,053490	34,104700
1,0	-46,682200	-22,367000	0,010399	0,051997	42,084500
2,0	-52,703600	-28,400700	0,007201	0,025314	92,200900
3,0	-56,229300	-31,952200	0,006319	0,008495	61,472800

Tabela 4.7: Comparação entre filtragem de Wiener e subtração espectral.

	Filtro de Wiener	Subtração Espectral
SNR final	1,8876 dB	-14,5146 dB
Correlação cruzada final	0,68811	0,11541
Redução de amplitude	46,8262%	45,5864%

entre o sinal reconstruído (a partir de cada banda individualmente) e o sinal ideal, bem como a taxa de redução de amplitude entre esses dois sinais.

6. Verificou-se, então, qual era a banda que apresentava a maior correlação cruzada com o sinal ideal. Foi visto que era a última, ou seja, a décima sétima (aquela relativa às frequências mais elevadas). A partir deste ponto, trabalhar-se-ia apenas com esta banda.
7. A seguir, até uma dada ordem permitida pela família em questão (no caso da família Daubechies, esse número foi variado de 1 a 45), mediu-se a correlação cruzada entre o sinal reconstruído (a partir apenas da última banda) e o sinal ideal, bem como a taxa de redução de amplitude entre esses dois sinais. Esses valores era, assim, guardados para futura análise.
8. Repetiu-se este processo para as outras famílias *wavelet* disponíveis.

Foi verificado que as melhores famílias para a aplicação em questão eram a família Daubechies e a symlets, devido ao fato de apresentarem maior correlação com o sinal de interesse. Também foram testadas as coiflets e família biortogonal. A partir dos valores relativos a estas famílias, foi possível montar a Figura 4.3. Baseando-se nestes gráficos, pode-se dizer que a melhor família dentre essas é a Daubechies de ordem 4. Apesar de esta não apresentar a menor taxa de redução de amplitude (cerca de 23%), apresenta a maior correlação cruzada com o sinal ideal quando a reconstrução é feita a partir da última banda (cerca de 0,96). Isso significa dizer que esta família é a que apresenta a maior capacidade de concentração de coeficientes e, por esta razão, é a mais propícia ao processo de supressão de ruído.

Contudo, como apenas a última banda será utilizada, após o primeiro nível de decomposição não é mais necessário dividir o sinal. Isso é devido ao fato de que a última banda é formada exatamente no primeiro nível de decomposição. Assim,



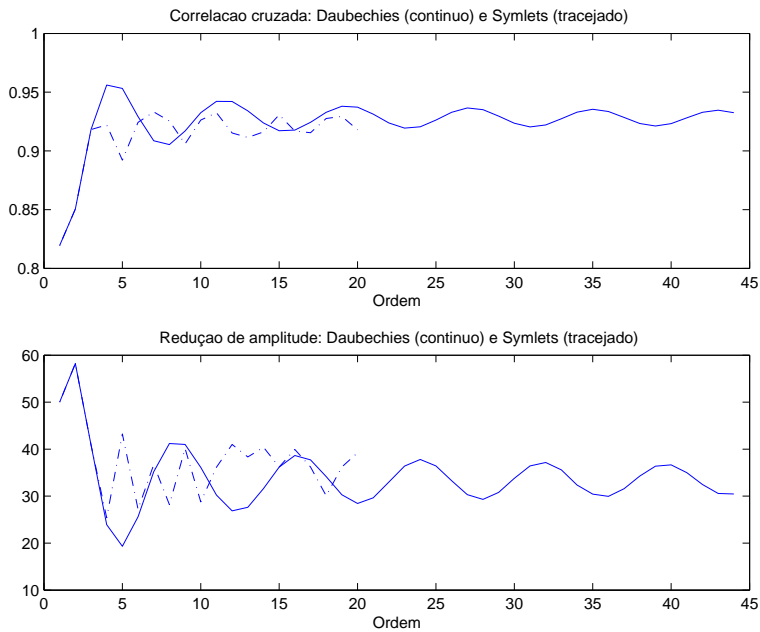


Figura 4.3: Correlação cruzada e taxa de redução de amplitude para as famílias Daubechies e Symlets.

será usado apenas um nível de decomposição, de forma a acelerar o processamento. Nesse caso, somente a segunda banda *wavelet* será usada.

Assim sendo, os parâmetros básicos necessários para a utilização da transformada *wavelet* já estão definidos: o número de níveis de decomposição é igual a 1, a família *wavelet* será a Daubechies de ordem 4 e a banda *wavelet* a ser usada na reconstrução será a última, ou seja, a segunda.

## 4.5.2 Testes de desempenho

Existem basicamente dois tipos de ruído, o ruído branco e a interferência de espectro discreto (DSI), constituído de sinais de modulação AM ruidosos. Os testes serão feitos para cada tipo de ruído separadamente e, em seguida, para a combinação desses dois tipos.

O método de *shrinkage* será feito em duas etapas de forma seqüencial. A primeira etapa será o método heurístico e a segunda será um dos demais métodos. O método heurístico consiste em realizar o que foi discutido na seção anterior, isto é, tomar apenas a banda *wavelet* onde se encontra a maior concentração de informação referente ao sinal de DP. A segunda etapa será executada sobre o sinal resultante

da primeira etapa.

#### 4.5.2.1 Tratamento do ruído branco

A metodologia a ser aplicada aqui para avaliação da capacidade da *wavelet* de tratar ruído branco é a seguinte:

1. Corromper o sinal ideal com vários níveis de ruído branco. Na verdade, o que irá variar é a razão entre a altura do nível do sinal de interesse e do ruído. Novamente, por altura, entenda-se três vezes o desvio padrão.
2. Realizar a supressão de ruído apenas com o método heurístico, analisando os valores referentes a cada um dos critérios de avaliação discutidos no início do capítulo.
3. Realizar a supressão de ruído do sinal resultante do método heurístico usando-se um dos métodos de *shrinkage*, analisando também os critérios de avaliação.
4. Comparar os resultados também variando o método de *thresholding*.

Na abordagem heurística (Tabela 4.8), foi possível obter alguma melhora no sinal corrompido. Isso se deu para todas as razões de amplitude entre sinal e ruído testadas. Contudo, a melhora não é significativa, visto que, ainda existe muito ruído na banda *wavelet* onde se encontra o sinal de DP. Realizando-se algum tipo de *shrinkage*, obtiveram-se resultados mais satisfatórios. Com o *visu shrink*, usando-se *hard thresholding* (Tabela 4.9), conseguiu-se aumentar bastante a correlação cruzada em relação ao que foi conseguido usando-se apenas a abordagem heurística. Uma melhora ainda maior foi obtida usando-se *visu shrink* com *soft thresholding* (Tabela 4.10), A aplicação de *sure shrink* (Tabelas 4.11) não se mostrou satisfatória devido ao fato de esta abordagem não ser adequada para sinais com grande presença de ruído. Por essa mesma razão, a abordagem híbrida obteve resultados iguais aos verificados quando se utiliza *visu shrink*.

Algo que se deve notar é o fato de que, em alguns casos, a redução de amplitude é de 100%. Isso não significa que os sinais de interesse foram “apagados”. Uma das possíveis causas é o fato de existir um seqüência de sinais e apenas o primeiro deles ter sido “ignorado”.

Tabela 4.8: Comportamento da transformada *wavelet* ao tratar ruído branco usando abordagem heurística.

Razão de amplitudes	SNR original	SNR processado	Corr. original	Corr. processado	Red. amplitude
0,10	-10,749600	-7,773300	0,278680	0,349750	18,624400
0,20	-16,780600	-13,794600	0,140850	0,181020	23,763200
0,50	-24,731800	-21,714200	0,056130	0,074532	30,654500
0,75	-28,240900	-25,214000	0,038809	0,051551	-6,395100
1,00	-30,760400	-27,759300	0,030872	0,040084	25,290900
1,50	-34,260300	-31,255300	0,016374	0,022217	33,085600
2,00	-36,774000	-33,752400	0,018971	0,024043	28,469700
3,00	-40,253600	-37,219700	0,010199	0,012710	20,074900

Tabela 4.9: Comportamento da transformada *wavelet* ao tratar ruído branco usando *visu shrink* com *hard thresholding*.

Razão de amplitudes	SNR original	SNR processado	Corr. original	Corr. processado	Red. amplitude
0,10	-10,749600	5,636600	0,278680	0,869010	18,624400
0,20	-16,780600	1,149600	0,140850	0,694910	26,890500
0,50	-24,731800	-5,852100	0,056130	0,345260	36,852200
0,75	-28,240900	-8,545500	0,038809	0,280540	8,827800
1,00	-30,760400	-12,144700	0,030872	0,112900	100,000000
1,50	-34,260300	-15,591500	0,016374	0,033662	100,000000
2,00	-36,774000	-18,282600	0,018971	0,038609	100,000000
3,00	-40,253600	-20,957700	0,010199	0,000000	100,000000

Tabela 4.10: Comportamento da transformada *wavelet* ao tratar ruído branco usando *visu shrink* com *soft thresholding*.

Razão de amplitudes	SNR original	SNR processado	Corr. original	Corr. processado	Red. amplitude
0,10	-10,749600	7,926300	0,278680	0,923310	28,960200
0,20	-16,780600	5,672400	0,140850	0,875760	43,897500
0,50	-24,731800	2,426700	0,056130	0,702510	69,310900
0,75	-28,240900	1,408100	0,038809	0,529210	54,822400
1,00	-30,760400	-0,206050	0,030872	0,163580	100,000000
1,50	-34,260300	-1,704000	0,016374	0,030489	100,000000
2,00	-36,774000	-2,550000	0,018971	0,079347	100,000000
3,00	-40,253600	-4,011900	0,010199	0,000000	100,000000

Tabela 4.11: Comportamento da transformada *wavelet* ao tratar ruído branco usando *sure shrink* com *hard thresholding*.

Razão de amplitudes	SNR original	SNR processado	Corr. original	Corr. processado	Red. amplitude
0,10	-10,749600	0,589510	0,278680	0,357090	26,410600
0,20	-16,780600	0,589240	0,140850	0,357090	100,000000
0,50	-24,731800	0,572350	0,056130	0,357090	100,000000
0,75	-28,240900	1,881300	0,038809	0,614610	6,191500
1,00	-30,760400	-0,351180	0,030872	0,257150	100,000000
1,50	-34,260300	-18,496900	0,016374	0,063420	100,000000
2,00	-36,774000	-27,041000	0,018971	0,017929	100,000000
3,00	-40,253600	-34,749300	0,010199	0,010106	100,000000

#### 4.5.2.2 Tratamento da DSI

A metodologia a ser aplicada aqui para avaliação da capacidade da *wavelet* de tratar DSI é a seguinte:

1. Corromper o sinal ideal um conjunto de sinais de DSI uniformemente distribuídos, assim como foi explicado anteriormente.
2. Realizar a supressão de ruído apenas com o método heurístico, analisando os valores referentes a cada um dos critérios de avaliação discutidos no início do capítulo.
3. Realizar a supressão de ruído do sinal resultante do método heurístico usando-se um dos métodos de *shrinkage*, analisando também os critérios de avaliação.
4. Comparar os resultados também variando o método de *thresholding*.

A transformada *wavelet* se mostrou mais adequada para tratar ruído branco do que para tratar ruído formado por DSI. Com a abordagem heurística (Tabela 4.12) foi possível apenas uma pequena melhora em relação ao sinal original. A abordagem *visu* como um todo se mostrou pouco eficiente: enquanto que ao se fazer *hard thresholding*, já se obtia um resultado bom para ruído branco, este método (*visu shrink*) se mostrou pouco eficiente para DSI. Esse quadro só se alterou ao se empregar o *soft thresholding*. Contudo o lucro que se obtém, mesmo ao se usar o *soft thresholding*, é bem menor do que o relativo a ruído branco. O método *sure*, pelo mesmo motivo apresentado para ruído branco, também não se mostrou adequado, o que implica um efeito semelhante ao que ocorreu com ruído branco relacionado à abordagem híbrida.

## 4.6 Resultados da filtragem adaptativa

A filtragem adaptativa também é bastante citada na literatura especializada [21, 27–32]. Contudo, não é tão usada devido a sua principal desvantagem: a necessidade de mais um canal contendo um ruído de referência que tenha sido gerado pela mesma fonte que gerou o ruído que corrompe o sinal de interesse.

Tabela 4.12: Comportamento da transformada *wavelet* ao tratar DSI contanimada por ruído branco usando abordagem heurística.

<b>INR</b>	<b>SNR original</b>	<b>SNR processado</b>	<b>Corr. original</b>	<b>Corr. processado</b>	<b>Red. amplitude</b>
100	-46,642500	-31,868900	0,013398	0,036844	-10,012100
50	-46,642700	-31,869800	0,013390	0,036798	-10,972100
25	-46,654800	-32,067300	0,013397	0,036383	-15,699200
20	-46,686700	-32,461200	0,013150	0,033891	1,831700
15	-46,770800	-33,548700	0,013452	0,030920	30,937300

Tabela 4.13: Comportamento da transformada *wavelet* ao tratar DSI contanimada por ruído branco usando *visu shrink* com *hard thresholding*.

<b>INR</b>	<b>SNR original</b>	<b>SNR processado</b>	<b>Corr. original</b>	<b>Corr. processado</b>	<b>Red. amplitude</b>
100	-46,642500	-15,712300	0,013398	0,099072	-37,318800
50	-46,642700	-15,429500	0,013390	0,102470	-37,983500
25	-46,654800	-16,725800	0,013397	0,088372	-42,915100
20	-46,686700	-17,279100	0,013150	0,079657	-20,909600
15	-46,770800	-17,987400	0,013452	0,047647	100,000000

Tabela 4.14: Comportamento da transformada *wavelet* ao tratar DSI contanimada por ruído branco usando *visu shrink* com *soft thresholding*.

<b>INR</b>	<b>SNR original</b>	<b>SNR processado</b>	<b>Corr. original</b>	<b>Corr. processado</b>	<b>Red. amplitude</b>
100	-46,642500	0,094341	0,013398	0,292660	61,425200
50	-46,642700	0,094947	0,013390	0,293400	60,768800
25	-46,654800	-0,359840	0,013397	0,236490	59,307500
20	-46,686700	-0,944340	0,013150	0,148020	86,341500
15	-46,770800	-1,242100	0,013452	0,067520	100,000000

Tabela 4.15: Comportamento da transformada *wavelet* ao tratar DSI contanimada por ruído branco usando *sure shrink* com *hard thresholding*.

INR	SNR original	SNR processado	Corr. original	Corr. processado	Red. amplitude
100	-46,642500	-20,717800	0,013398	0,062965	-32,349100
50	-46,642700	-20,858800	0,013390	0,061999	-33,034600
25	-46,654800	-22,192400	0,013397	0,066605	-37,744100
20	-46,686700	-23,705600	0,013150	0,065636	-20,909600
15	-46,770800	-26,676700	0,013452	0,039204	-8,367500

Para estes testes, o sinal desejado é constituído pelo sinal de interesse somado um ruído. O sinal do segundo sensor é esse mesmo ruído submetido a um filtro IIR de ordem 3.

Os testes foram feitos usando-se um sinal que é corrompido por DSI com INR igual a 20 dB e ruído branco de mesma amplitude que o sinal de interesse. Todos os resultados foram obtidos calculando-se a média dos resultados referentes a 10 tentativas de modo a se ter resultados mais confiáveis. Isso não seria feito na prática, mas apenas para se ter uma informação estatística teórica mais completa.

Ao se realizar os primeiros testes, percebeu-se que, ao se passar o sinal corrompido pelo sistema adaptativo, a supressão de ruído era bastante prejudicada nas primeiras amostras. Para evitar essa região de pouca eficiência, o sinal corrompido é submetido duas vezes ao sistema, sendo que, na segunda vez, os coeficientes são inicializados com os valores obtidos no fim da primeira vez.

O primeiro algoritmo a ser implementado foi o LMS. Este algoritmo apresenta um parâmetro livre, denominado  $\mu$ , que precisa ser estimado. Por isso, foram feito vários testes variando-se o valor de  $\mu$ . Esses resultados são mostrados na Tabela 4.16.

Pode-se ver nesta tabela que o melhor valor para  $\mu$  é 0,001 pois apresenta a maior melhora na correlação cruzada e na razão sinal ruído, mantendo uma redução de amplitude baixa. Contudo, o LMS é conhecido por apresentar uma curva de convergência muito lenta. Sua principal vantagem é a simplicidade do algoritmo. Na

Figura 4.4, pode ser vista de que forma as curvas de convergência são modificadas ao se variar o valor de  $\mu$ . Essas curvas são referentes à segunda vez que o sinal corrompido é submetido ao sistema adaptativo.

Vê-se que, na Figura 4.4, ao se usar o valor de  $\mu$  igual a 0,001, a convergência foi mais rápida e o excesso foi visualmente menor. Dessa forma, percebe-se que o desajuste para este valor de  $\mu$  é também menor, provendo uma melhor convergência.

O NLMS tende a ter uma curva de convergência mais rápida devido ao uso do fator de convergência variável. Entretanto, o preço a ser pago por essa rapidez é o aumento do desajuste [14]. Novamente, existe um parâmetro a ser ajustado,  $\mu_n$ . Para isso, foram feitos testes variando-se o valor de  $\mu_n$  e os resultados são mostrados na Tabela 4.17. O melhor valor para este parâmetro é 0,4, pois apresenta o maior aumento na correlação cruzada e na razão sinal-ruído, mantendo uma redução de amplitude baixa.

Para aumentar ainda mais a velocidade de convergência, recorre-se ao algoritmo RLS. Os resultados relativos a este algoritmo são mostrados na Tabela 4.18. O parâmetro a ser ajustado é o fator de esquecimento, denotado por  $\lambda$ . Pelos mesmos critérios usados anteriormente, o melhor valor para  $\lambda$  é 0,9999.

O RLS apresenta uma correlação cruzada final bem próxima de 1. Porém, esse aumento na velocidade de convergência requer uma grande elevação na complexidade computacional e apresenta alguns problemas de estabilidade [14].

Como uma alternativa para aliviar o custo computacional, porém mantendo a velocidade e a qualidade da convergência, emprega-se o algoritmo *Set-membership affine projection*. Os resultados relativos a este algoritmo são mostrados na Tabela 4.19. O parâmetro livre deste algoritmo é o  $\gamma$ . Deve-se notar uma nova coluna nesta tabela, que é a porcentagem de *updates*. Vê-se que, para  $\gamma$  igual a 0,005, tem-se uma correlação cruzada final maior do que aquelas obtidas no LMS ou NLMS. Seu desempenho é quase tão bom quanto o do RLS (enquanto que no RLS, obteve-se perto de 0,99, para o SMAP, obteve-se 0,92). Mas, segundo a Tabela 4.19, em apenas 15% das iterações houve a necessidade de atualizar os coeficientes do filtro, implicando uma drástica redução na complexidade.

Para se ter uma idéia geral do que ocorre em termos de qualidade e velocidade da convergência, pode-se verificar, na Figura 4.5, as curvas de convergência relativas



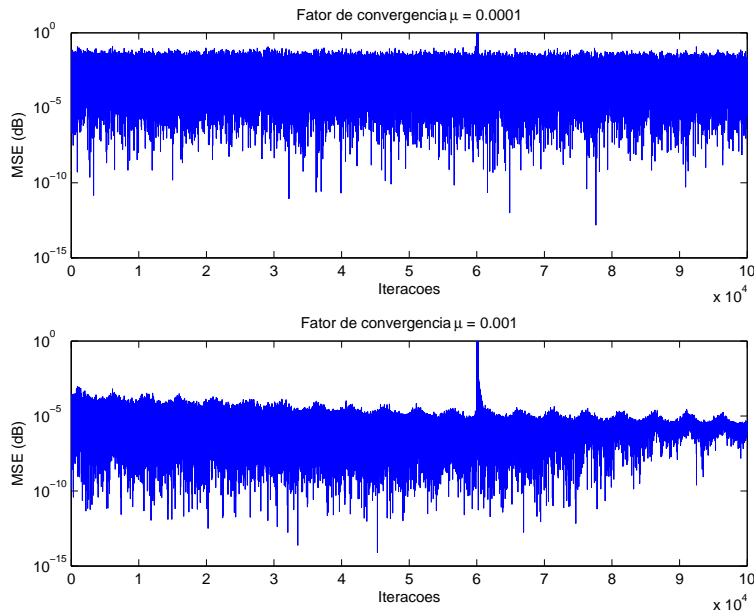


Figura 4.4: Curvas de convergência do algoritmo LMS variando-se o valor de  $\mu$ .

Tabela 4.16: Comportamento da filtragem adaptativa ao tratar DSI (INR = 20 dB) e ruído branco de mesma amplitude que sinal de interesse usando algoritmo LMS. A SNR original (do sinal corrompido) é de -46,79 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,00957.

$\mu$	MSE final	SNR processado	Corr. processado	Red. amplitude
0,000100	0,005110	-18,400000	0,121450	1,560000
0,000200	0,001962	-15,430000	0,172620	1,090000
0,000300	0,000712	-12,400000	0,244650	0,700000
0,000400	0,000243	-9,290000	0,342860	0,430000
0,000500	0,000078	-6,130000	0,467650	0,250000
0,000600	0,000024	-3,010000	0,606110	0,130000
0,000700	0,000008	-0,110000	0,728970	0,050000
0,000800	0,000004	2,130000	0,808960	0,010000
0,000900	0,000003	3,310000	0,842810	-0,010000
0,001000	0,000003	3,500000	0,844960	-0,020000

Tabela 4.17: Comportamento da filtragem adaptativa ao tratar DSI (INR = 20 dB) e ruído branco de mesma amplitude que sinal de interesse usando algoritmo NLMS. A SNR original (do sinal corrompido) é de -46,79 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,00957.

$\mu_n$	<b>MSE final</b>	<b>SNR processado</b>	<b>Corr. processado</b>	<b>Red. amplitude</b>
0,100000	0,001151	-13,770000	0,206380	-0,910000
0,200000	0,000081	-6,080000	0,465100	-0,300000
0,300000	0,000005	1,280000	0,781170	-0,070000
0,400000	0,000001	5,470000	0,899330	0,000000
0,500000	0,000001	5,200000	0,892520	0,020000
0,600000	0,000001	3,910000	0,858330	0,030000
0,700000	0,000002	2,700000	0,817150	0,030000
0,800000	0,000002	1,650000	0,772160	0,040000
0,900000	0,000002	0,710000	0,724370	0,040000
1,000000	0,000003	-0,120000	0,674570	0,050000

Tabela 4.18: Comportamento da filtragem adaptativa ao tratar DSI (INR = 20 dB) e ruído branco de mesma amplitude que sinal de interesse usando algoritmo RLS. A SNR original (do sinal corrompido) é de -46,79 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,00957.

$\lambda$	<b>MSE final</b>	<b>SNR processado</b>	<b>Corr. processado</b>	<b>Red. amplitude</b>
0,999900	0,000001	33,370000	0,999750	0,029000
0,999990	0,000001	26,430000	0,998840	0,040000
0,999999	0,000002	20,590000	0,995640	0,047000
1,000000	0,000002	19,980000	0,995000	0,048000

Tabela 4.19: Comportamento da filtragem adaptativa ao tratar DSI (INR = 20 dB) e ruído branco de mesma amplitude que sinal de interesse usando algoritmo *Set-membership Affine Projection*. A SNR original (do sinal corrompido) é de -46,79 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,00957.

$\gamma$	MSE final	% <i>Up- dates</i>	SNR processado	Corr. processado	Red. amplitude
0,100000	0,001296	5	-12,100000	0,280630	-0,411000
0,010000	0,000015	12	3,810000	0,891570	0,005000
0,005000	0,000005	15	5,240000	0,921590	0,037000
0,001000	0,000005	75	5,650000	0,928800	0,080000
0,000100	0,000006	99	5,630000	0,928380	0,103000
0,000000	0,000006	100	5,620000	0,928310	0,106000

a cada um dos algoritmos. Deve-se lembrar que estes gráficos são relativos à primeira vez que o sinal corrompido é submetido ao sistema de supressão de ruído. As curvas relativas à segunda vez são mostradas na Figura 4.6.

Percebe-se que, o algoritmo LMS é o mais lento e que o NLMS apresenta, na aplicação em questão, uma leve melhora em relação ao primeiro. O algoritmo RLS é o que apresenta uma curva mais rápida, contudo o SMAP, apesar de ser ligeiramente mais lento que o RLS, apresenta uma curva mais suave indicando que o excesso de MSE é menor e, por conseqüência, o desajuste também (o que é mostrado principalmente pela Figura 4.6).

O critério para saber se o algoritmo já convergiu foi o seguinte: testou-se submeter o sinal a ser tratado uma vez. Como já foi visto, isso gerou resultados ruins pois o filtro ainda estava convergindo. Ao se submeter a segunda vez, os resultados foram bem superiores. Ao se submeter uma terceira vez, não houve melhora nos resultados. Desse modo, considerou-se o fim da convergência no final da segunda submissão.

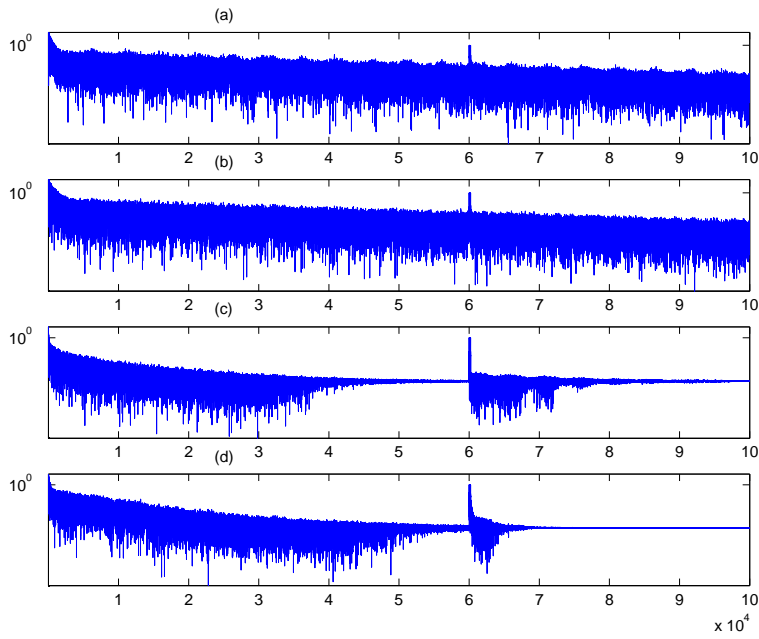


Figura 4.5: Curvas de convergência para os algoritmos estudados quando se submete o sinal corrompido pela primeira vez ao sistema de supressão adaptativa de ruído: (a) LMS; (b) NLMS; (c) RLS; (d) SMAP

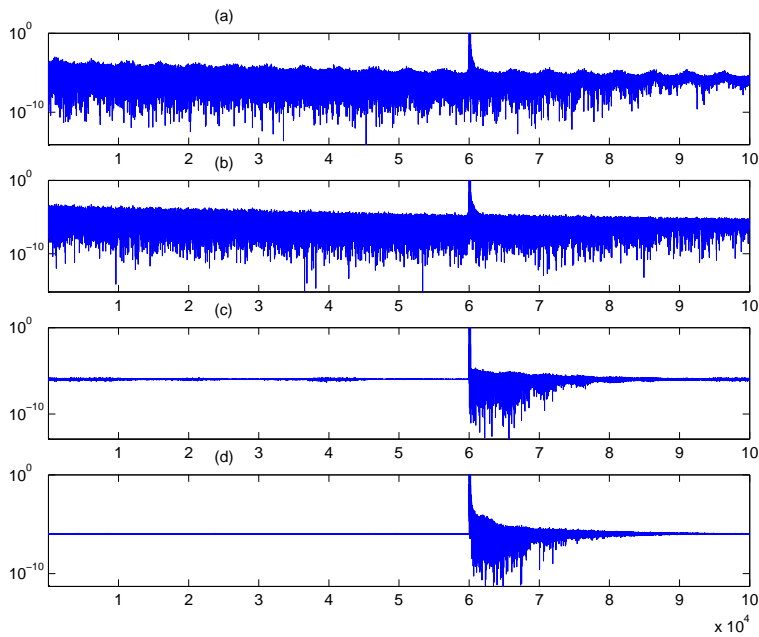


Figura 4.6: Curvas de convergência para os algoritmos estudados quando se submete o sinal corrompido pela segunda vez ao sistema de supressão adaptativa de ruído: (a) LMS; (b) NLMS; (c) RLS; (d) SMAP.

## 4.7 Resultados da separação cega por ICA

Este método necessita de dois canais de informação. Esse número é justificado pelo fato de serem necessários tantos sinais de mistura quanto forem os sinais a serem separados. Desse modo, foi definido um sinal de ruído assim como no caso da filtragem adaptativa e uma matriz de mistura de forma arbitrária, como por exemplo

$$A = \begin{bmatrix} 1 & 1 \\ 0,1 & 10 \end{bmatrix} \quad (4.3)$$

Dessa forma, obtiveram-se os sinais de mistura apresentados na Figura 4.7. Ao se realizar o processo de separação cega baseado em ICA, obtiveram-se os sinais mostrados na Figura 4.8. O sinal de interesse é recuperado de forma quase perfeita.

É importante lembrar que, neste método, a informação quanto à amplitude do sinal é totalmente perdida, sendo possível que o sinal recuperado seja até mesmo o oposto do original, ou seja, tenha sua fase alterada em -180 graus.

A outra limitação deste método, isto é, o fato de não se saber a ordem na qual os sinais originais serão calculados, pode ser contornada se for calculada, para cada sinal obtido, a função de autocorrelação. Nesse caso, o que se faz é comparar a função de autocorrelação para cada sinal para se verificar qual é a que tem o aspecto mais impulsivo. Esta seria a que representa o sinal de ruído indicando que a outra representaria o sinal de interesse.

## 4.8 Resultados da combinação de filtragem de Wiener com transformada *wavelet*

Ao se analisar os resultados dos métodos apresentados, uma primeira idéia que se tem é combinar métodos para verificar se o poder de supressão de ruído é aumentado. Os métodos mais propícios a serem combinados são aqueles baseados em apenas um canal. Isso é devido a questões práticas: para os métodos baseados em mais canais, toda a configuração de aquisição dos sinais é diferenciada. A razão principal de se tentar combinar métodos é fazer com que um método possa tratar o que outro método não foi capaz de eliminar.

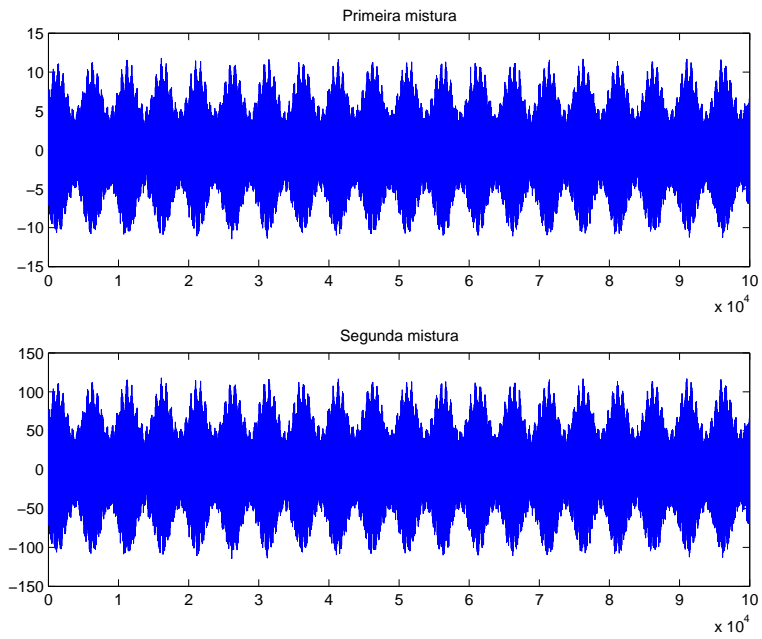


Figura 4.7: Sinais de misturas usados para a separação cega baseada em ICA.

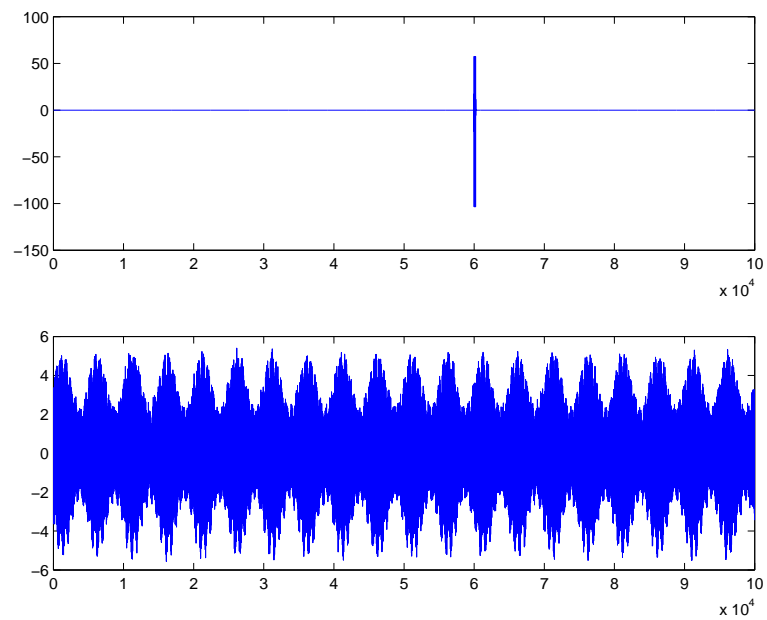


Figura 4.8: Resultados da separação cega baseada em ICA.

Como exemplo dessa abordagem, submete-se o sinal corrompido (Figura 4.9a) ao filtro de Wiener. Isso é mostrado na Figura 4.9b e pode-se ver que o sinal resultante apresenta perto de 0,64 de correlação cruzada. Ainda restam, no sinal resultante, resíduos de ruído que se assemelham a impulsos ditribuídos ao longo do sinal. Esse sinal pode, então, ser submetido à transformada *wavelet*, o que é mostrado na Figura 4.9c. Ao se fazer isso, a nova correlação cruzada resultante atinge um valor próximo de 0,88. Isso ocorre porque a transformada *wavelet* irá atacar justamente esses resíduos. Contudo, ainda se percebe que este resíduos apresentam uma amplitude significativa. Isso acontece basicamente pelo fato de que os limiares calculados para se realizar o *wavelet shrinkage* não estão acima destes resíduos. Ajustando manualmente os limiares, assim como é feito na Figura 4.9d, obtêm-se resultados ainda melhores, chegando-se a uma correlação cruzada final de cerca de 0,92. Pode-se ver que os resíduos estão bem mais suavizados devido ao melhor posicionamento dos limiares. Os resultados são enfatizados na Tabela 4.20.

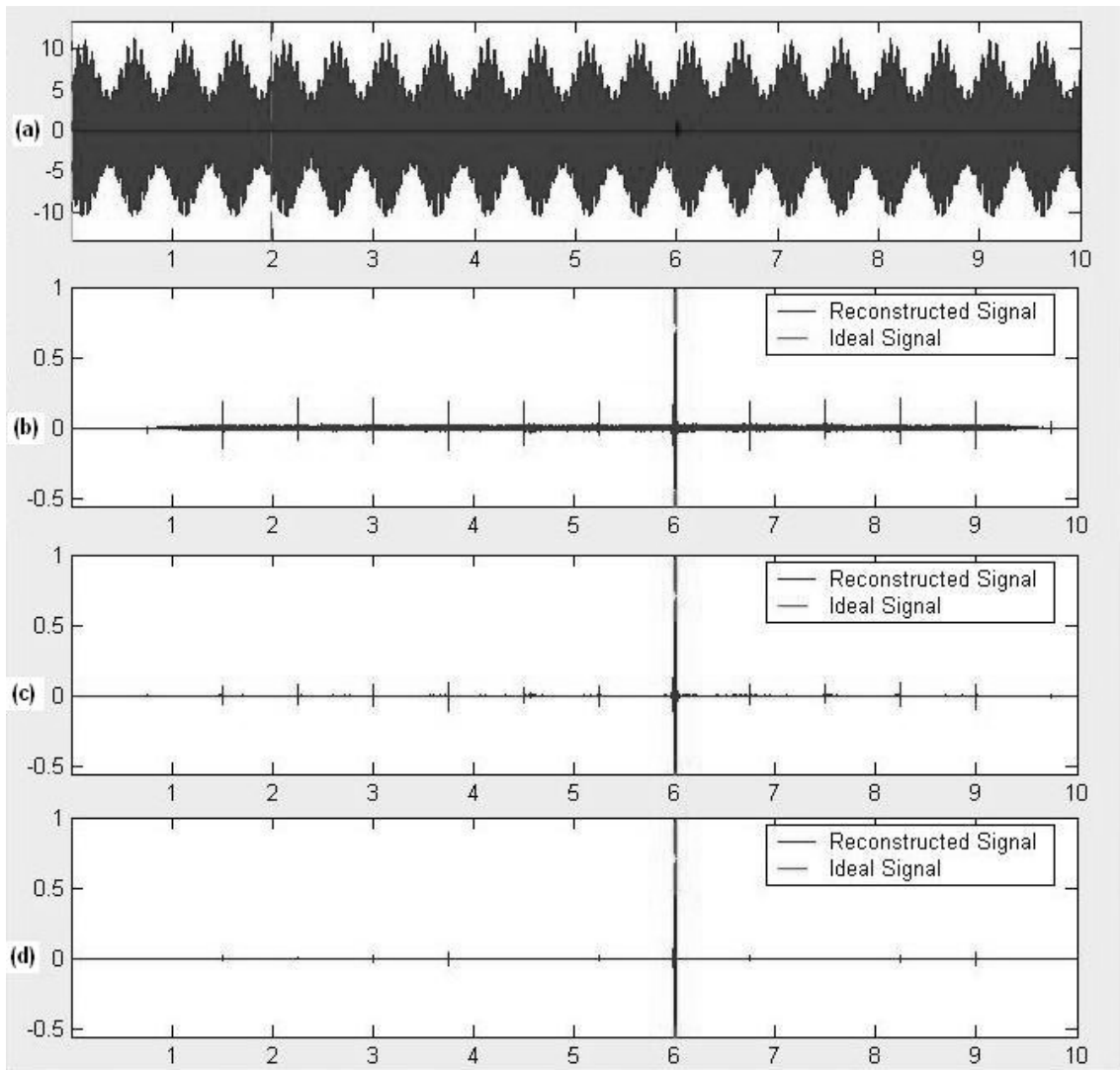


Figura 4.9: Supressão de ruído utilizando conjuntamente filtro de Wiener e transformada *wavelet*: (a) Sinal corrompido com sinal ideal concentrado na amostra 60000; (b) Sinal resultante da supressão de ruído empregando filtro de Wiener; (c) A seguir, emprega-se a transformada *wavelet*; (d) A seguir, ajustam-se os limiares manualmente.



Tabela 4.20: Comportamento da supressão de ruído empregando-se filtro de Wiener e transformada *wavelet*. A SNR original (do sinal corrompido) é de -46,6425 dB e a correlação cruzada entre este sinal e o sinal limpo é de 0,00972

Método	Ajuste manual dos limiares	SNR	Corr. Cruzada	Red. amplitude
Corrompido	–	-46,6425	0,00972	–
Wiener	–	1,8876	0,68811	46,8262
<i>wavelet</i>	não	6,1725	0,88558	58,9544
<i>wavelet</i>	sim	5,5752	0,92096	63,1795

## 4.9 Conclusão

Neste capítulo, foram vistos os resultados obtidos ao se empregar os métodos de supressão de ruído debatidos no Capítulo 3.

O filtro de Wiener se mostrou muito adequado para o tratamento de DSI. Isso acontece porque, como o espectro da interferência deste tipo é discreto, o filtro de Wiener (de forma análoga ao que acontece na subtração espectral) é capaz de subtrair este espectro sem que isso afete a DP propriamente dita. Isso, por sua vez, é justificado pelo fato de o espectro do sinal de interesse ser bem mais “distribuído” que o espectro do ruído. Assim, mesmo após a remoção dos espectros relativos à DSI (que pode não ser tão precisa), o espectro do sinal de interesse se mantém bem próximo do que era originalmente. A subtração espectral se mostrou inferior ao filtro de Wiener, pois esse pode ser visto como uma subtração espectral ponderada pela SNR, o que faz com que a remoção das interferências, principalmente as de espectro discreto, seja mais suave.

O uso da transformada *wavelet* se mostrou eficiente ao se tratar ruído branco empregando-se a abordagem heurística seguida da abordagem *visu shrink*. Devido ao fato de haver muito ruído, o *sure shrink* se mostrou ineficiente enquanto que o método híbrido obteve resultados similares ao *visu*. A utilização deste tipo de transformada apresentou bons resultados quando em conjunto com a filtragem de Wiener, o que gerou resultados superiores aos obtidos com os métodos individuais.

A filtragem adaptativa obteve resultados superiores aos demais métodos.

Contudo, esta necessita de mais de um canal de informação, o que é uma dificuldade grande em termos práticos. Foram apresentados resultados referentes aos diversos algoritmos. O LMS, por apresentar o passo fixo, mostrava uma convergência muito lenta. O NLMS possui passo variável e, por isso, tendia a ter uma convergência mais rápida. O RLS, ao custo de um aumento da complexidade computacional, apresentava uma convergência ainda mais rápida. O *set-membership affine projection* apresentava uma convergência quase tão rápida quanto o RLS, um desajuste menor e uma complexidade computacional bem mais baixa devido ao *set-membership filtering*, em que os coeficientes só eram atualizados quando estavam abaixo de um certo limiar.

O uso de separação cega baseada em ICA leva a resultados próximos do ótimo. As únicas limitações são aquelas do próprio ICA, isto é, perda da informação sobre amplitude e desconhecimento em relação à ordem na qual os sinais ideal e de ruído são calculados. Esse método também sofre da mesma dificuldade da filtragem adaptativa, isto é, a necessidade de mais de um canal de informação.

Tendo sido feita a supressão de ruído presente no sinal de descargas parciais, pode-se dar início à etapa de detecção do sinal, o que será tratado no Capítulo 5.

# Capítulo 5

## Detecção de Sinais

### 5.1 Introdução

Tendo sido realizada a supressão do ruído presente no sinal de DP, como foi discutido no capítulo anterior, pode-se executar o processo de detecção com maior probabilidade de sucesso. Existem várias formas de se realizar essa tarefa. O ICM System, citado no Capítulo 2, faz a detecção com base na amplitude do sinal: se este exceder um certo limiar, considera-se que ocorreu a descarga. Contudo, este método não é robusto, visto que, mesmo após a utilização de métodos de *denoising*, há a possibilidade de existir algum ruído no sinal que apresentaria uma amplitude similar à da DP, sendo, então, mascarado como descarga parcial. O que é proposto no presente capítulo é a utilização de um método baseado em filtros casados, devido ao fato de que esta abordagem leva em consideração a forma do sinal a ser detectado. A Seção 5.2 introduz o conceito de detecção binária. Na Seção 5.3, é vista a estrutura dos filtros casados. Este método adaptado para a detecção de DP's é explicado na Seção 5.4, assim como o modo de se determinar o limiar de detecção. Por fim, será mostrado, na Seção 5.5, como se gerar os mapas de acordo com o ICM System (assim como foi explicado no Capítulo 2).

### 5.2 Teoria de decisão e teste de hipótese

Em alguns problemas de detecção de sinais, existem duas hipóteses  $H_0$  e  $H_1$ , onde  $H_0$  é a hipótese de o sinal na entrada do sistema de detecção ser um sinal  $s_0(k)$

corrompido por ruído e  $H_1$  é a hipótese de o sinal na entrada do sistema de detecção ser um sinal  $s_1(k)$  corrompido por ruído. Isso é sintetizado a seguir:

$$H_0 : x(k) = s_0(k) + n(k)$$

$$H_1 : x(k) = s_1(k) + n(k).$$

Tendo isso em mente, existem 4 possibilidades quanto à forma com a qual o sistema de detecção irá responder a um dado sinal de entrada. Tais possibilidades são resumidas na Tabela 5.1.

Assim, pode-se definir probabilidades associadas a tais eventos da forma que se segue:

$$Prob(\text{detecção}) = \frac{\text{eventos detectados como } H_1}{\text{total de eventos } H_1} \quad (5.1)$$

$$Prob(\text{falso alarme}) = \frac{\text{eventos detectados como } H_1}{\text{total de eventos } H_0} \quad (5.2)$$

$$Prob(\text{miss}) = \frac{\text{eventos detectados como } H_0}{\text{total de eventos } H_1} \quad (5.3)$$

$$(5.4)$$

Tal decisão (entre as hipóteses  $H_0$  e  $H_1$ ) deve ser feita com base na probabilidade *a posteriori*  $P(H_i|y)$ , onde  $y$  é um valor particular de uma variável aleatória  $Y$  obtido a partir da observação do sinal  $x(k)$ . Assim,  $P(H_i|y)$  denota a probabilidade de a hipótese  $H_i$  ser verdadeira dado que um valor particular  $y$  foi obtido. Dessa forma, fica claro que o sistema deve escolher como verdadeira  $H_1$  se  $P(H_1|y)$  for maior que  $P(H_0|y)$  e vice-versa. De uma forma mais resumida, tem-se

$$\frac{P(H_1|y)}{P(H_0|y)} \underset{H_0}{\overset{H_1}{\geq}} 1 \quad (5.5)$$

Tabela 5.1: Possibilidades quanto à forma com a qual o sistema de detecção irá responder a um dado sinal de entrada.

Resposta do sistema	Resposta correta	Classificação
$H_1$	$H_1$	Detecção
$H_0$	$H_1$	Perda do alvo ou <i>miss</i>
$H_1$	$H_0$	Falso alarme
$H_0$	$H_0$	-

A equação (5.5) é denominada regra de decisão MAP (*maximum a posteriori*). É ainda possível desenvolver essa fórmula ao se invocar a regra de Bayes, onde pode-se escrever  $P(H_i|y)$  como

$$P(H_i|y) = \frac{f_{Y|H_i}(y|H_i)P(H_i)}{f_Y(y)}, \quad (5.6)$$

o que permite que se defina uma nova regra de decisão como

$$\frac{f_{Y|H_1}(y|H_1)P(H_1)}{f_{Y|H_0}(y|H_0)P(H_0)} \underset{H_0}{\underset{H_1}{\geq}} 1 \quad (5.7)$$

$$\frac{f_{Y|H_1}(y|H_1)}{f_{Y|H_0}(y|H_0)} \underset{H_0}{\underset{H_1}{\geq}} \frac{P(H_0)}{P(H_1)} \quad (5.8)$$

A equação (5.8) é uma forma mais prática da regra de decisão MAP. A partir desse desenvolvimento, pode-se definir um valor muito importante na teoria de decisão denominado razão de semelhança, dada por

$$L(y) = \frac{f_{Y|H_1}(y|H_1)}{f_{Y|H_0}(y|H_0)} \quad (5.9)$$

Em alguns problemas nessa área, são definidos valores denominados custos de erros. Esses valores têm como objetivo estimar a importância de cada uma das linhas da Tabela 5.1. Dessa forma, é possível obter-se uma regra de decisão que retrate de forma mais fiel a realidade [15]. Assim, a regra de decisão, agora denominada regra de decisão de Bayes, é dada por

$$L(y) = \frac{f_{Y|H_1}(y|H_1)}{f_{Y|H_0}(y|H_0)} \underset{H_0}{\underset{H_1}{\geq}} \frac{P(H_0)(C_{10} - C_{00})}{P(H_1)(C_{01} - C_{11})}, \quad (5.10)$$

onde  $C_{ij}$  é o custo (ou penalidade) que se obtém ao selecionar a hipótese  $H_i$  como verdadeira quando a verdadeira é a  $H_j$ .

Quando os custos não estão disponíveis (como é geralmente o caso), é necessário chegar a regras alternativas, como a regra *minimax* ou o critério de Neyman-Pearson [15]. Neste trabalho, foi utilizado o critério de Neyman-Pearson por este não necessitar nem das probabilidades *a priori*  $P(H_i)$  e nem dos custos. Nesse

critério, a regra é manter a probabilidade de falso alarme,  $P(D_1|H_0)$  abaixo de um certo nível e minimizar a probabilidade de *miss*. Isso será visto com mais detalhes mais adiante.

### 5.3 Filtro casado

Filtro casado é um filtro que apresenta uma associação com um sinal específico. Mais precisamente, sua resposta impulsional tem a forma deste sinal invertida no tempo e deslocada. Sua formação pode ser justificada pelo fato de que, quando se submete algum sinal a este filtro, na verdade o que é feito é a correlação cruzada entre o sinal de entrada e o sinal para o qual o filtro foi casado. A inversão é justificada pois a filtragem é descrita por uma convolução, e esta é feita invertendo-se a resposta impulsional do filtro. Assim, realizando esta inversão, a convolução estaria, na verdade, levando o sinal casado ao seu formato original. Dessa forma, percebe-se que, quando o sinal de entrada é o mesmo sinal para o qual o filtro foi casado, obtém-se a correlação cruzada máxima. Assim, do ponto de vista digital, o filtro casado nada mais é do que um correlacionador.

Deve-se submeter o sinal de entrada,  $x(k)$  a dois filtros casados em paralelo: um casado com  $s_0(k)$  e outro casado com  $s_1(k)$ . A saída de cada filtro então deverá ser amostrada com um período igual ao comprimento do sinal para o qual o filtro foi casado. A seguir, as saídas dos filtros são comparadas (subtraídas), gerando um valor que pode determinar com qual sinal o sinal de entrada possui a maior correlação cruzada. Isso é feito comparando-se este valor com um limiar que é calculado com base nas probabilidades de detecção e falso alarme que se deseja. Toda essa explicação pode se resumida pela Figura 5.1.

Pela configuração apresentada na Figura 5.1, pode-se deduzir, segundo [15], a regra de decisão dada por

$$\int_0^T y(t)s_1(t)dt - \int_0^T y(t)s_0(t)dt \underset{H_0}{\overset{H_1}{\geq}} \gamma \quad (5.11)$$

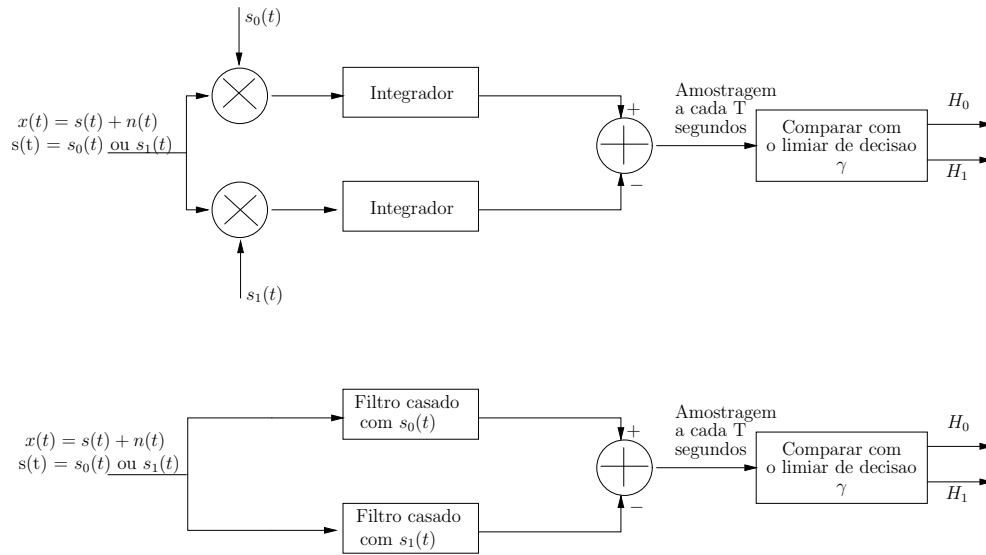


Figura 5.1: Configuração básica para o uso de filtros casados.

## 5.4 Filtro casado adaptado ao caso das DP's

No caso das DP's, as hipóteses que serão testadas apresentam uma diferença básica em relação ao que foi apresentado. Enquanto a hipótese  $H_1$  corresponde ao sinal de interesse mais ruído, a hipótese  $H_0$  corresponde a somente ruído. Isso pode ser resumido como

$$H_0 : x(k) = n(k) \quad (5.12)$$

$$H_1 : x(k) = s(k) + n(k) \quad (5.13)$$

Dessa forma, a estrutura geral é alterada para acomodar essa nova condição.

Para a estrutura de utilização do filtro casado convencional, supõe-se que o sinal está sincronizado com o sistema de detecção. Isso significa que o sinal ocorre apenas em instantes definidos (regularmente espaçados), de forma que se saiba exatamente os instantes iniciais nos quais o sinal de interesse pode ocorrer.

Contudo, a maior peculiaridade em relação às descargas parciais é o fato de não se ter informação sobre onde elas ocorrerão. Elas podem ocorrer em qualquer fase (ou ângulo) dentro do ciclo de potência. Essa informação, isto é, a fase de ocorrência da DP, é uma das mais importantes informações para o correto diagnóstico do equipamento elétrico em questão.

Assim, não haverá a etapa de amostragem da saída do filtro casado mostrada

na Figura 5.1. A comparação deverá ser feita em todas as amostras da saída do filtro. Isso é devido ao fato de que a DP pode ocorrer em qualquer instante. Desse modo, se alguma amostra do sinal de saída do filtro casado superar um determinado limiar, a DP será detectada.

É importante lembrar que, neste caso, será usado o conceito de filtros casados determinísticos. Isso significa que o sinal de interesse tem uma forma definida. Essa forma será o modelo matemático da descarga parcial.

#### 5.4.1 Determinação do limiar de detecção

Na Figura 5.2, estão esquematizados todos os conceitos vistos aqui:

- No eixo horizontal estão os possíveis valores que a saída do filtro casado pode assumir (tanto para eventos ruidosos quanto para eventos de sinal mais ruído).
- Há duas pdf's condicionais. A da esquerda representa a pdf da variável aleatória que constitui a saída do filtro casado para o caso de o evento ser apenas de ruído. A da direita, analogamente, representa a pdf da variável aleatória que constitui a saída do filtro casado para o caso de o evento ser de sinal corrompido por ruído.
- Está traçado o limiar de detecção. Se a saída do filtro casado for maior que este limiar o evento será considerado como de sinal mais ruído. Caso contrário, será tido como um evento ruidoso.
- Com base nesse limiar, é possível estabelecer os valores para probabilidade de detecção, probabilidade de falso alarme e probabilidade de *miss*. Essas probabilidades são as áreas representadas na figura.

Como já foi dito, o limiar de detecção é, neste trabalho, determinado pelo critério de Neyman-Pearson. O que se fez foi determinar um valor para a probabilidade de detecção. É importante notar que, tendo estabelecido um valor para a probabilidade de detecção, a probabilidade de *miss* estará definida (como 100% menos a probabilidade de detecção). Contudo, a probabilidade de falso alarme irá depender das pdf's condicionais definidas nas equações (5.8) e (5.9). Dessa forma, definindo-se uma probabilidade de detecção, tendo-se as pdf's condicionais, é possível se chegar



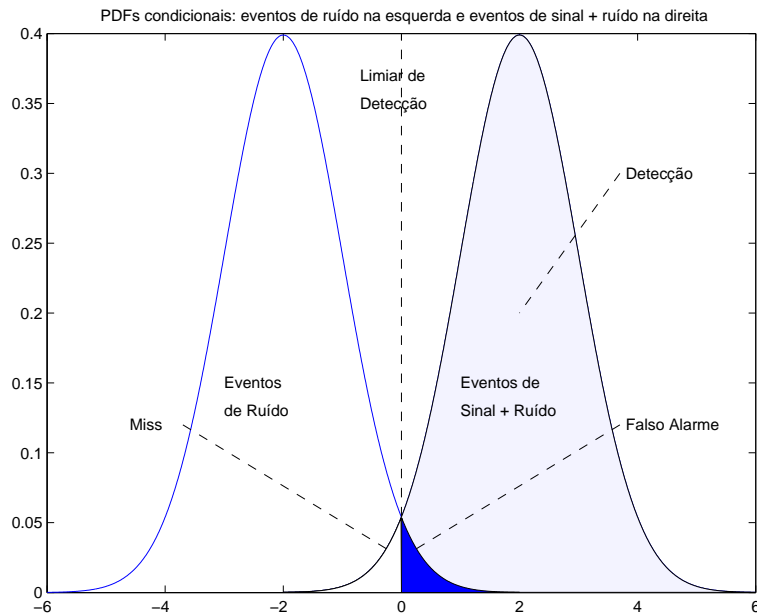


Figura 5.2: As duas pdf's condicionais: a da esquerda corresponde à resposta do filtro casado aos eventos que são constituídos apenas de ruído e a da direita de eventos de sinal corrompido por ruído.

a um valor para a probabilidade de falso alarme. Tendo vários pares de valores “probabilidade de detecção - probabilidade de falso alarme”, gera-se um gráfico que possui no eixo horizontal os valores de falso alarme e no eixo vertical os valores correspondentes para a probabilidade de detecção. A curva obtida é denominada curva ROC (*receiver operating characteristic*), cuja nomenclatura é oriunda de sistemas de radar. Examinando-se esta curva, como na Figura 5.3, é possível chegar a valores de detecção e falso alarme que apresentem um bom compromisso e, assim, obter um valor para o limiar de detecção que seja o mais próximo possível do ótimo.

### 5.4.2 Algoritmo do método

Os procedimentos para a detecção baseada em filtros casados são resumidos no Algoritmo 5.1.

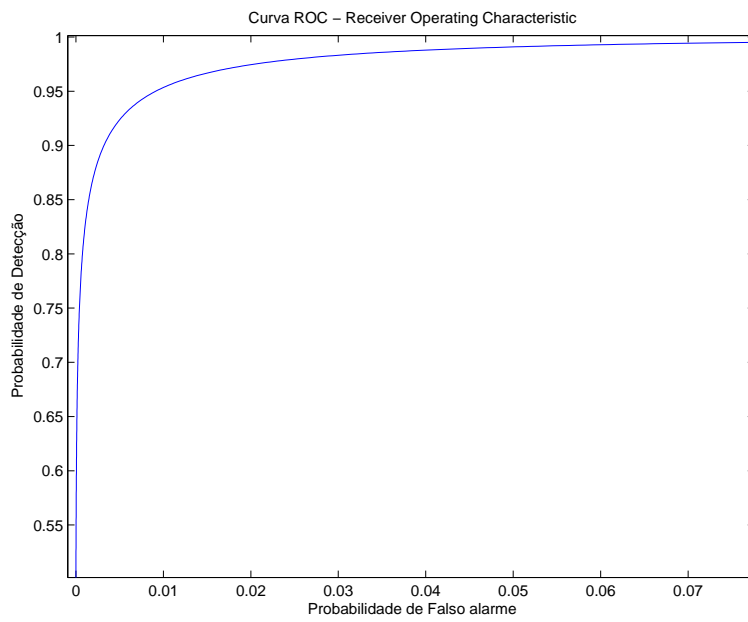


Figura 5.3: Curva ROC referente a Figura 5.2

1. Definir o sinal de interesse. No caso, será o modelo matemático da descarga parcial.
2. Definir a resposta impulsional filtro casado como o sinal de interesse revertido no tempo.
3. Estimar que razão sinal-ruído será obtida nas medições reais.
4. Gerar uma certa quantidade de eventos de sinal corrompido por ruído apresentando a SNR estimada no passo anterior. Gerar a mesma quantidade de eventos apenas de ruído. Pode-se aqui usar um ruído com a mesma variância do ruído que foi adicionado ao sinal.
5. Submeter esses eventos ao filtro casado e definir as pdf's condicionais através de histogramas.
6. Gerar a curva ROC.
7. Definir a probabilidade de detecção que fornece o melhor compromisso em relação à probabilidade de falso alarme.
8. Estabelecer, com base nessa probabilidade de detecção, o limiar de detecção.
9. Tendo sido definidos o filtro casado em si e o limiar de detecção, o sistema de detecção está pronto para ser utilizado.
10. Submeter um vetor contendo possíveis eventos de DP ao filtro casado. Isso significa fazer a convolução deste vetor (sinal de entrada) com a resposta impulsional do filtro casado (que é definida como o sinal de interesse revertido no tempo).
11. Deslocar a saída do filtro casado em  $L$  amostras para a esquerda, onde  $L$  é o comprimento do filtro casado.
12. Submeter a saída deslocada do filtro casado ao limiar de detecção.
13. As amostras que forem maiores que o limiar de detecção serão tidas como instantes iniciais de eventos de DP.

**Algoritmo 5.1:** Procedimentos para a detecção baseada em filtros casados.

## 5.5 Geração de mapas

Os mapas gerados pelo ICM System nada mais são do que histogramas. Esses histogramas são estruturados conforme explicado no Capítulo 2. Para gerá-los, criou-se uma função que tem como argumentos o sinal de entrada (que pode ter sido ou não submetido a processos de *denoising*), o padrão do filtro casado e o limiar de detecção. Essa função retorna as amostras nas quais foram detectados instantes iniciais de eventos de DP. Esses instantes são denominados como *marcas*. Esse vetor de marcas é processado da seguinte forma:

1. Removem-se os eventuais valores nulos.
2. Define-se um vetor-linha de zeros com o comprimento do sinal de entrada.
3. As amostras do vetor de zeros que tiverem índices contidos no vetor de marcas terão seus valores modificados para 1. Esse é o novo vetor de marcas.
4. Faz-se a multiplicação ponto-a-ponto deste novo vetor de marcas com o sinal de entrada. Desta multiplicação, obter-se-á um vetor que é todo zerado, exceto nas posições onde houve um início de um evento de DP. O valor de cada uma destas amostras será a amplitude do vetor de entrada no instante do início do evento de DP. Esse é o novo vetor de marcas.
5. Concatenando-se verticalmente todos esses vetores de marcas, obtém-se uma matriz de marcas.

Para se obter o mapa de acordo com o ICM System, procede-se segundo o Algoritmo 5.2. A matriz de marcas é denominada **Marks\_Hist** e o mapa final é denominado **Hist**. **SignalVector** é o sinal de entrada do sistema de detecção.

```

Max_Hist = max(max(Marks_Hist));
Min_Hist = min(min(Marks_Hist));
Bins_Amp = linspace(Min_Hist, Max_Hist, 256+1);
Bins_Pha = 1 : ((floor(length(SignalVector)/257))*257);
Bins_Pha = reshape(Bins_Pha, length(Bins_Pha)/257, 257)';
Bins_Pha = Bins_Pha(:, 1)';

Hist = zeros(length(Bins_Amp)-1, length(Bins_Pha)-1);

for i = 1 : size(Marks_Hist, 1),
    for j = 2 : length(Bins_Pha)-1,
        for k = Bins_Pha(j-1) : Bins_Pha(j),
            if (Marks_Hist(i, k) ~= 0)
                I = max(find(Bins_Amp < Marks_Hist(i, k)));
                Hist(I, j) = Hist(I, j) + 1;
            end
        end
    end
end
end

```

**Algoritmo 5.2:** Procedimentos para geração do mapa segundo o ICM System.

## 5.6 Resultados da detecção

Deve-se, então, verificar o desempenho do sistema de detecção e os efeitos da supressão de ruído sobre o mesmo. Como o sistema de detecção é baseado em filtro casado, que, por sua vez, é baseado em correlação cruzada, plota-se um gráfico da correlação cruzada do sinal corrompido com o sinal limpo de DP. Assim, existirão picos de correlação nas posições onde se encontram os sinais de DP. Isso deve ser feito tanto para o sinal corrompido quanto para o sinal processado pelo sistema de supressão de ruído. Executar esse procedimento antes e depois desse processamento é justificado pelo fato de que, antes do processamento, estar-se-ia testando o desempenho do sistema de detecção antes da supressão do ruído, e, depois, estar-se-ia testando a influência da supressão de ruído na detecção dos sinais de DP. Os sinais de DP têm início na amostra de índice 60000. O objetivo de todos esses procedimentos de detecção é localizar picos de correlação.

### 5.6.1 Detecção com pré-processamento feito por filtro de Wiener

Para testar o desempenho do sistema de detecção quando o sinal corrompido for pré-processado por uma etapa de supressão de ruído realizada com filtro de Wiener, usam-se, assim como no Capítulo 4, ruídos formados por DSI. Isso é justificado pelo fato de o filtro de Wiener ter se mostrado mais adequado para tratar esse tipo de interferência. Os parâmetros empregados são os mesmos que foram estipulados na análise do Capítulo 4. Assim, varia-se a INR desde um valor alto até um valor baixo. Os valores utilizados para a INR foram 60, 40 e 20 dB.

Para cada um desses valores de INR, foi calculada a função de correlação cruzada com o sinal padrão de DP cujos procedimentos de geração e aquisição foram descritos também no Capítulo 4. A Figura 5.4a mostra a função de correlação cruzada entre o sinal corrompido e o sinal padrão de DP quando a INR é igual a 60 dB. A Figura 5.4b mostra a função de correlação cruzada entre o sinal processado e o sinal padrão de DP na mesma situação. A Figura 5.5a mostra a função de correlação cruzada entre o sinal corrompido e o sinal padrão de DP quando a INR é igual a 40 dB. A Figura 5.5b mostra a função de correlação cruzada entre o sinal processado

e o sinal padrão de DP na mesma situação. A Figura 5.6a mostra a função de correlação cruzada entre o sinal corrompido e o sinal padrão de DP quando a INR é igual a 20 dB. A Figura 5.6b mostra a função de correlação cruzada entre o sinal processado e o sinal-padrão de DP na mesma situação.

Pode-se ver que, em nenhum dos casos apresentados, o filtro casado, representado pela função de correlação sem processamento, é capaz de detectar a posição dos sinais de DP. A etapa de pré-processamento de supressão de ruído se mostrou essencial, nesse caso, para que fosse possível a detecção.

### **5.6.2 Detecção com pré-processamento feito por transformada *wavelet***

Para testar o desempenho do sistema de detecção quando o sinal corrompido for pré-processado por uma etapa de supressão de ruído realizada com transformada *wavelet*, usa-se, assim como no Capítulo 4, ruído branco para corromper o sinal de interesse. Os parâmetros empregados são os mesmos que foram estipulados na análise do Capítulo 4. Isso é justificado pelo fato de a transformada *wavelet* ter se mostrado mais adequada para tratar esse tipo de interferência.

Para o caso onde o ruído branco apresenta altura igual à do sinal de interesse, o filtro casado é capaz de apontar a posição dos pulsos de DP, o que é mostrado na Figura 5.7a. Contudo, percebe-se que há uma considerável probabilidade de erro, já que há outros pontos onde a correlação cruzada é alta. Ao se processar o sinal corrompido pela transformada *wavelet*, o risco de se cometer um erro é diminuído pelo fato de os picos referentes aos verdadeiros pulsos de DP serem muito mais destacados em relação aos demais, o que é visto na Figura 5.7b.

### **5.6.3 Detecção com pré-processamento feito por filtro adaptativo**

Para estudar o efeito da filtragem adaptativa sobre a detecção dos sinais de DP, corrompeu-se o sinal de interesse com DSI contaminada com ruído branco (INR igual a 20 dB). A Figura 5.8a mostra a função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. A 5.8b mostra a função de correlação

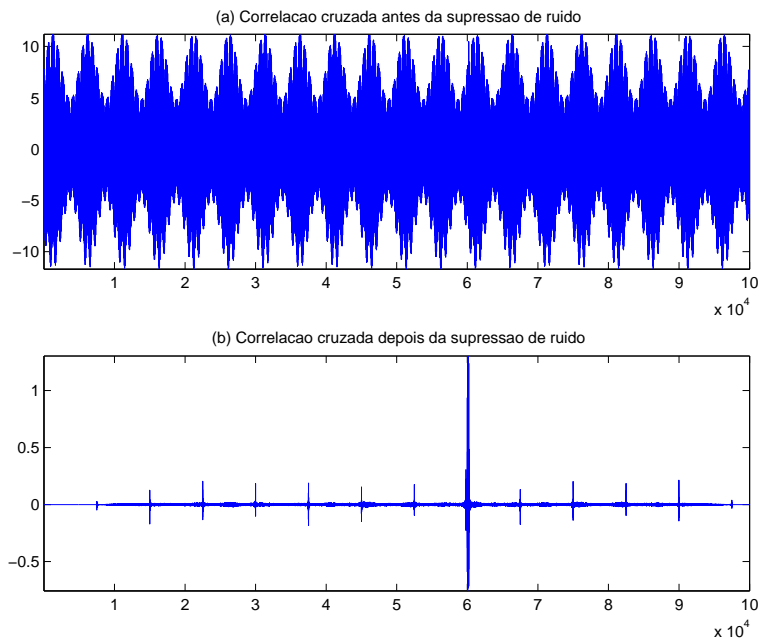


Figura 5.4: (a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem de Wiener e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 60 dB.



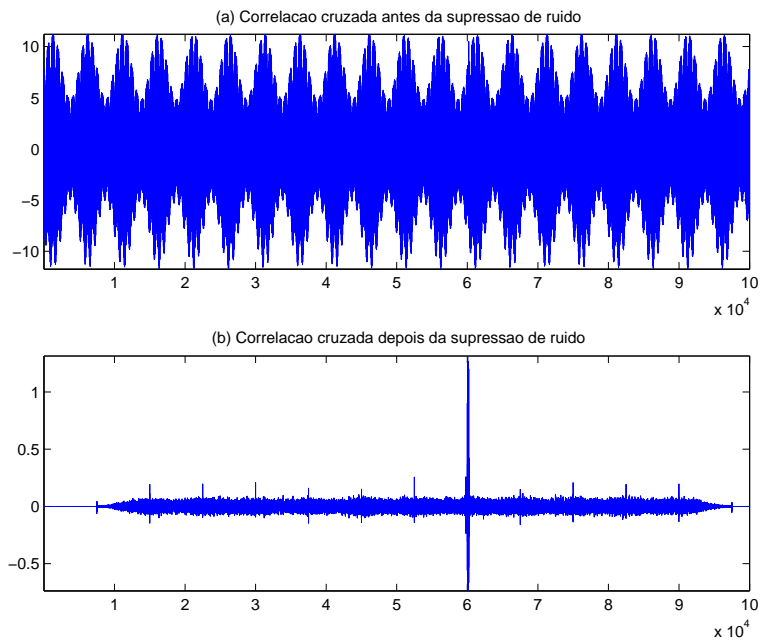


Figura 5.5: (a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem de Wiener e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 40 dB.

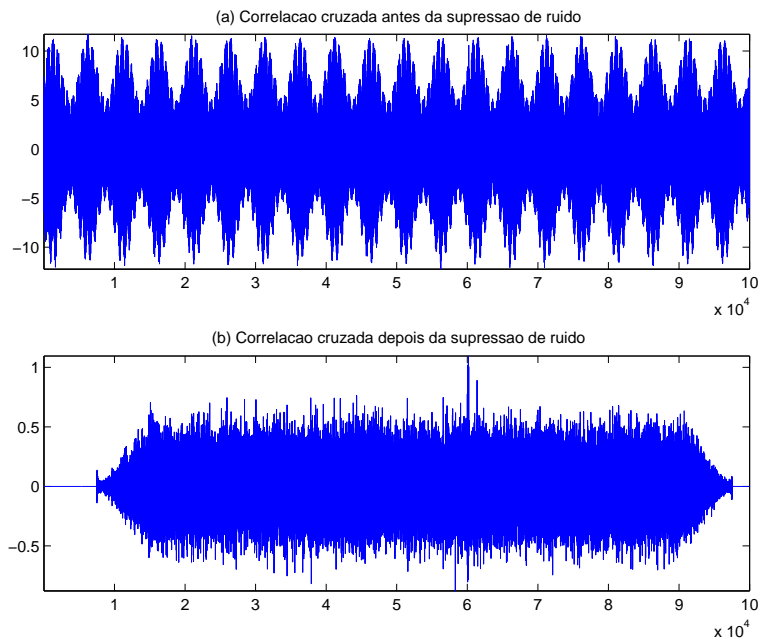


Figura 5.6: (a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem de Wiener e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB.

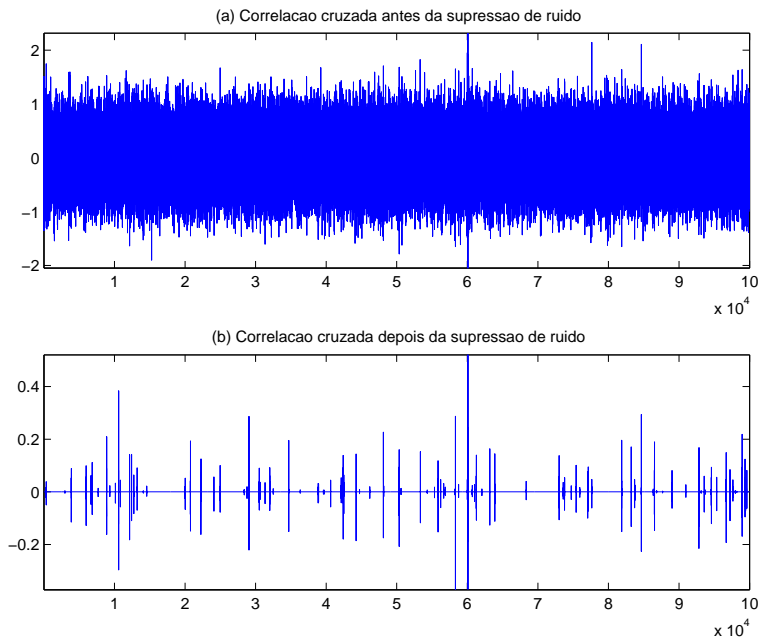


Figura 5.7: (a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por transformada *wavelet* e o padrão de DP deslocado. Para ambos os casos, o ruído branco presente no sinal corrompido apresentava a mesma amplitude do sinal de interesse.

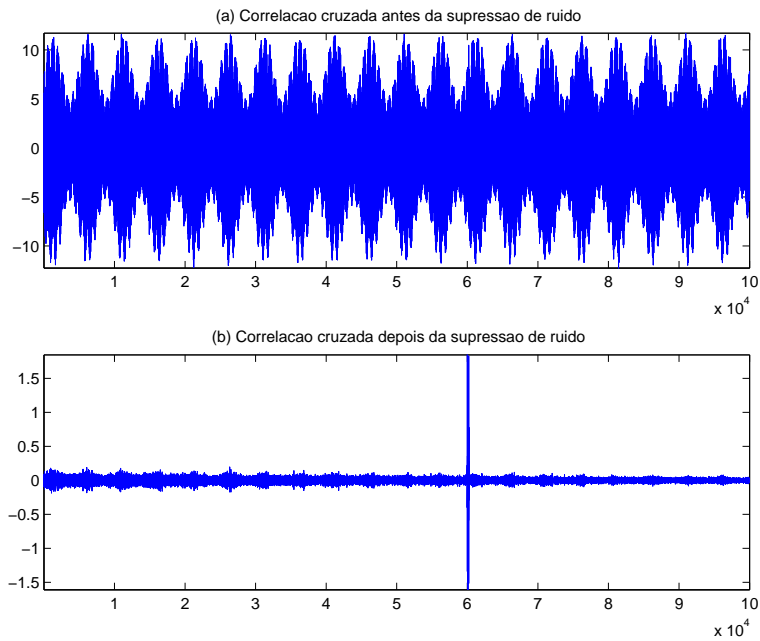


Figura 5.8: (a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem adaptativa (algoritmo LMS) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB.

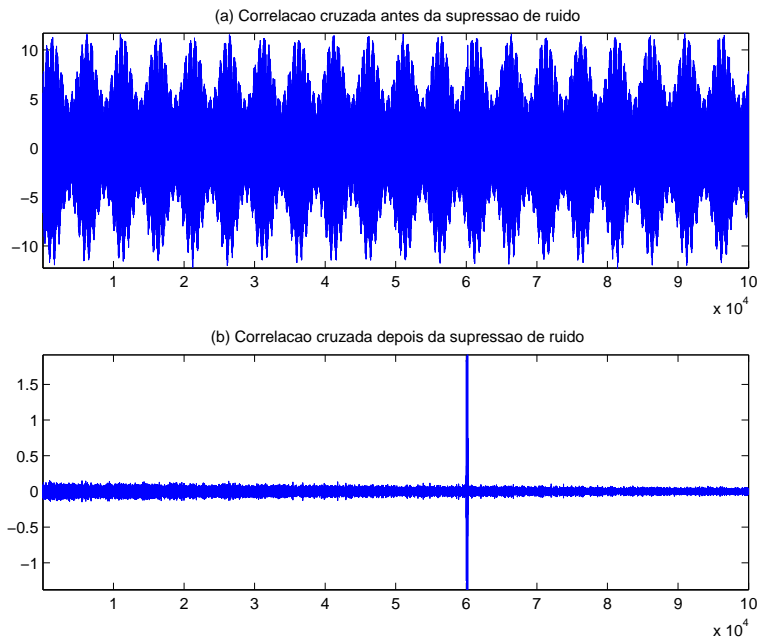


Figura 5.9: (a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem adaptativa (algoritmo NLMS) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB.

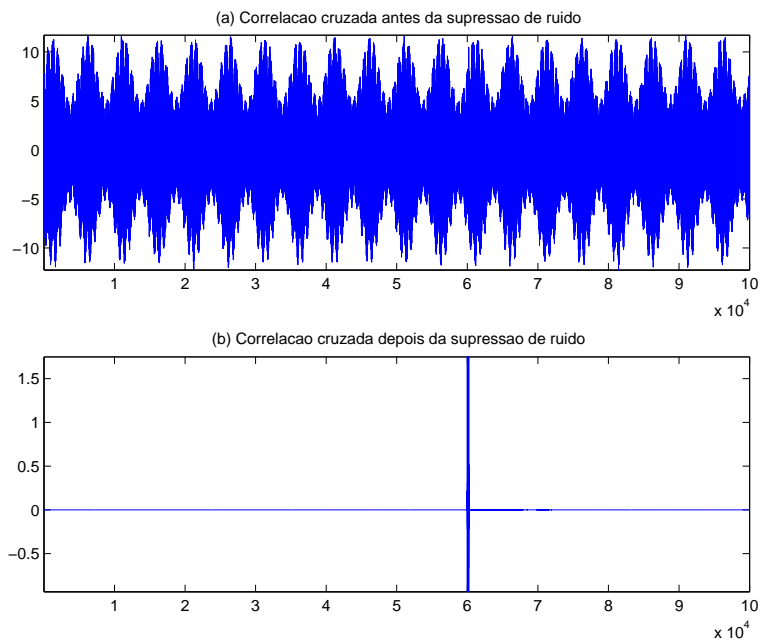


Figura 5.10: (a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem adaptativa (algoritmo RLS) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB.

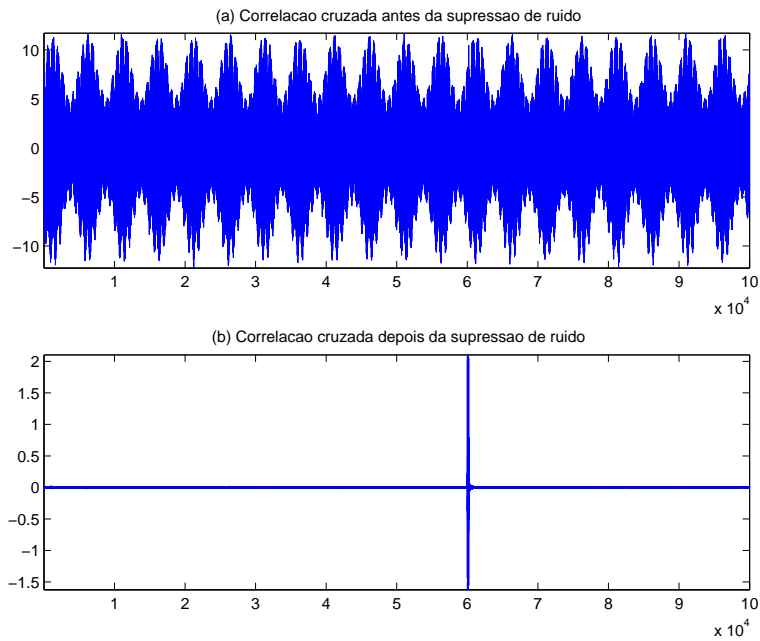


Figura 5.11: (a) Função de correlação cruzada entre o sinal corrompido e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído por filtragem adaptativa (algoritmo *set-membership affine projection*) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB.

cruzada entre o sinal processado pela etapa de supressão de ruído realizada com filtragem adaptativa (algoritmo LMS). As Figuras 5.9, 5.10 e 5.11 mostram as mesmas condições para os algoritmos NLMS, RLS e *set-membership affine projection*, respectivamente.

Pode-se ver que, em nenhum dos casos apresentados, o filtro casado, representado pela função de correlação sem processamento, é capaz de detectar a posição dos sinais de DP. Para cada algoritmo adaptativo, usaram-se os parâmetros considerados ótimos na análise feita no capítulo 4. Variando-se o algoritmo, o desempenho apresentado na etapa de detecção pode ser melhorado. A etapa de pré-processamento de supressão de ruído se mostrou essencial, nesse caso, para que fosse possível a detecção.

#### 5.6.4 Detecção com pré-processamento feito por combinação de filtro de Wiener com transformada *wavelet*

Para se estudar o efeito da combinação da filtragem de Wiener com a transformada *wavelet* para fins de detecção, corrompeu-se o sinal de interesse com DSI contaminada com ruído branco. Foram usados dois níveis de INR (60 e 20 dB).

A Figura 5.12a mostra a função de correlação cruzada entre o sinal corrompido e o sinal padrão de DP quando a INR é igual a 60 dB. A Figura 5.12b mostra a função de correlação cruzada entre o sinal processado e o sinal padrão de DP na mesma situação. A Figura 5.13a mostra a função de correlação cruzada entre o sinal corrompido e o sinal padrão de DP quando a INR é igual a 20 dB. A Figura 5.13b mostra a função de correlação cruzada entre o sinal processado e o sinal padrão de DP na mesma situação.

Percebe-se que, com DSI apresentando INR igual a 60 dB, ao se usar somente filtro de Wiener, existirão, após o processamento, pulsos falsos secundários, como se vê na Figura 5.12a. Este pulsos poderão ser suavizados com a utilização da transformada *wavelet*, como pode se ver na Figura 5.12b. Vê-se que, agora os pulsos verdadeiros de DP são praticamente os únicos a serem detectados. Quando a DSI apresentar INR igual a 20 dB, vê-se que, após a filtragem de Wiener, ainda restará bastante ruído (5.13a). Este poderia ser amenizado com a transformada *wavelet*, como visto na Figura 5.13b.



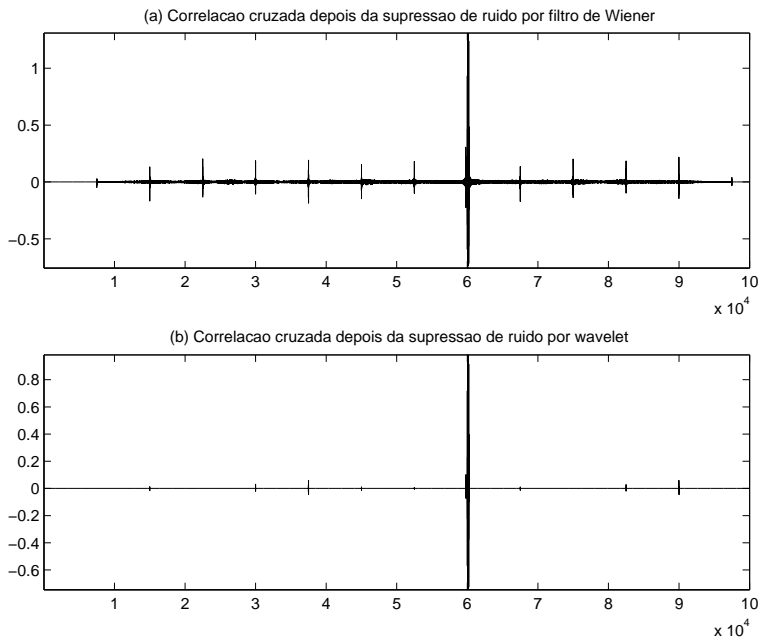


Figura 5.12: (a) Função de correlação cruzada entre o sinal processado pelo filtro de Wiener e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído (formado pela filtragem de Wiener em série com a transformada *wavelet*) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 60 dB.

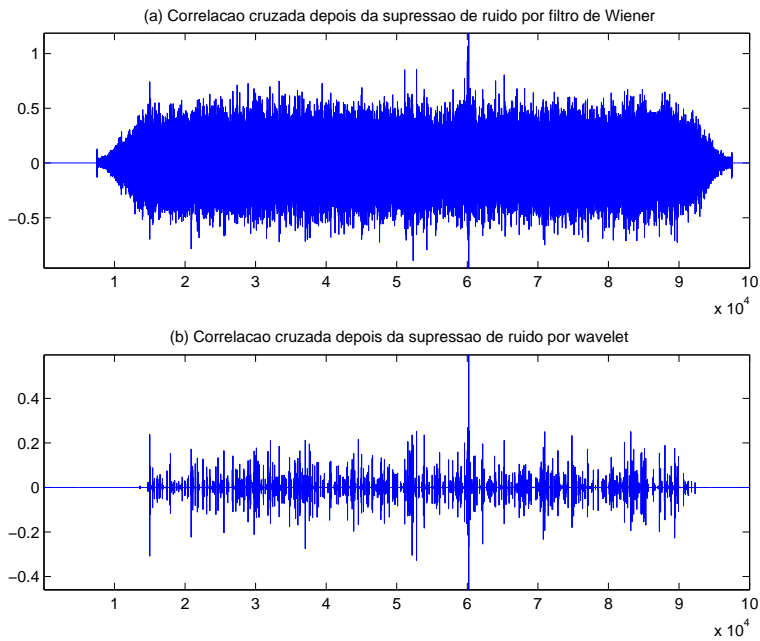


Figura 5.13: (a) Função de correlação cruzada entre o sinal processado pelo filtro de Wiener e o padrão de DP deslocado. (b) Função de correlação cruzada entre o sinal processado pelo sistema de supressão de ruído (formado pela filtragem de Wiener em série com a transformada *wavelet*) e o padrão de DP deslocado. Para ambos os casos, a DSI presente no sinal corrompido apresentava INR igual a 20 dB.

## 5.7 Conclusão

Neste capítulo, foi visto um método de detecção de sinais baseado em filtros casados. Este método tende a ser mais robusto que o utilizado pelo ICM System, visto que este utiliza apenas limiar de detecção por amplitude. Por se basear em filtros casados, o método proposto leva em consideração a forma do sinal de interesse para realizar a detecção.

Os métodos apresentados no Capítulo 4 são analisados como uma etapa de pré-processamento dos sinais de descargas parciais. Pode-se dizer que o mais vantajoso, do ponto de vista de resultado de detecção, é a filtragem adaptativa. Contudo, este método sofre da grande desvantagem de necessitar de dois canais de informação. Os resultados relativos aos demais métodos são bastante dependentes do modelo para a descarga parcial adotado.

Com isso, é possível gerar mapas similares ao que o ICM System. Contudo, o mapa gerado desta forma terá uma imunidade maior a ruídos. Esses mapas serão as entradas para o sistema de classificação proposto no próximo capítulo, que representará a última etapa do processamento dos sinais de descargas parciais. Sabendo-se a classe da DP que está ocorrendo, saber-se-á também que tipo de defeito está presente no equipamento elétrico sob avaliação.

# Capítulo 6

## Classificação Neural de Padrões de Descargas Parciais

### 6.1 Introdução

O que se propõe neste capítulo é a implementação de um classificador neural que opere sobre os componentes principais não-lineares de modo a indicar qual tipo de DP está ocorrendo, apesar das interferências, propagadas por condução ou indução [33]. Sabendo-se o tipo de DP, é possível saber o tipo de defeito que está ocorrendo no equipamento elétrico. O uso de componentes principais não-lineares permite a aceleração do processamento e a diminuição do custo computacional. Também serão feitas comparações de desempenho com os componentes principais lineares e um estudo sobre conjuntos de treino e teste, tendo em vista a classificação destes sinais e referindo-se ao trabalho de prospecção de dados, realizado em [3]. A Seção 6.2 mostra como foi feita a aquisição de eventos para treinamento do classificador. A Seção 6.3 apresenta os componentes principais não-lineares, enquanto que, na Seção 6.4, será visto como se faz a extração destes componentes. Por último, a Seção 6.5 mostra como foi feita a classificação dos eventos de DP.

### 6.2 Aquisição de sinais

As técnicas e ferramentas existentes para classificação de sinais de DP são baseadas em modelos experimentais [3]. Para se obter tais modelos, são feitos testes

em células experimentais, que são dispositivos capazes de forçar o aparecimento de DP. É possível, então, gerar experimentalmente alguns padrões típicos similares à realidade encontrada nos equipamentos. Assim, será possível saber em que parte do sistema a ser medido está realmente ocorrendo o fenômeno de DP.

Este trabalho é baseado em dados fornecidos pelo Cepel, o qual adota três células com configurações elétricas básicas denominadas Ponta Flutuante (FL), Ponta Plano (PP) e Ponta Terra (PT). Da configuração FL, foram gerados 35 eventos, da PP, 55, e da PT, 25. Tais eventos foram, então, divididos em dois conjuntos para o projeto de classificador: de treino e de teste. O conjunto de treino serve para que o sistema possa ser ajustado. O de teste se presta a validar o que foi desenvolvido usando-se o conjunto de treino. Dessa forma, deve-se ter, nesses dois conjuntos, caracterizações estatísticas bem próximas, apesar da restrição prática dos dados experimentais colhidos.

A medição dos sinais de DP é obtida através de três parâmetros: amplitude da descarga, fase onde ocorre a descarga dentro do ciclo de potência e taxa de repetição. Estes três parâmetros são então colocados em um mapa bidimensional, conforme é mostrado na Figura 6.1, ao longo de um período de senoíde, que serve de referência para se saber em que ponto do ciclo de potência houve a descarga. O eixo horizontal, desta forma, indica o ângulo (ou fase) no qual a descarga ocorre. O eixo vertical indica a amplitude da descarga, e a taxa de repetição da descarga com certa amplitude ocorrendo em uma determinada fase é visualizada pela intensidade do ponto em uma escala de cinza.

Esse mapa é uma matriz que apresenta 256 linhas e 256 colunas, o que requer

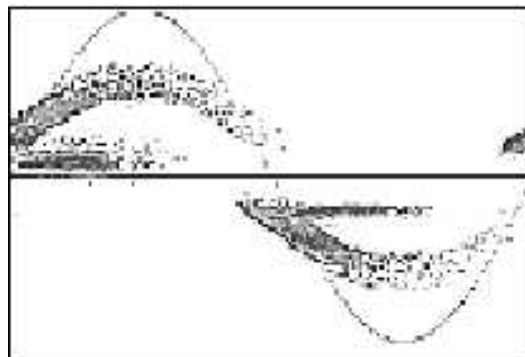


Figura 6.1: Aparência do mapa gerado para os sinais de DP [2].

um esforço computacional significativo para o processamento da informação. Como uma primeira tentativa de aliviar os cálculos, o que foi feito foi agrupar os elementos dessa matriz em grupos de  $8 \times 8$  elementos e somá-los, obtendo-se uma nova matriz com 32 linhas e 32 colunas.

### 6.3 Componentes principais não lineares

Mesmo assim reduzido, o espaço de dados de entrada é de dimensão bastante elevada e a compactação da informação pode se mostrar atraente. Tal compactação em geral é possível porque há dois tipos de dimensionalidade: a superficial e a intrínseca [34]. A dimensionalidade superficial dos dados pode ser definida como o número de variáveis aparentes de um conjunto de dados. A dimensionalidade intrínseca poderia, então, ser definida como o número de variáveis independentes que regem as variações não-aleatórias nas observações, ou seja, é o número de variáveis que estão realmente por trás do processo que se deseja estudar, lembrando-se que estas nem sempre estão visíveis como aquelas que foram descritas como superficiais.

A análise de componentes principais (PCA) tem sido extensamente utilizada na compactação de sinais, que se dá na retenção de componentes de maior energia e na eliminação daquelas de menor energia. Estes componentes principais correspondem aos autovetores da matriz de correlação relativa ao conjunto de dados [35].

Contudo, em muitos casos, as variáveis que compõem o vetor de informação de entrada apresentam correlação não-linear, de tal modo que uma análise de componentes principais não-lineares (ou NLPCA) pode ser mais eficiente. Esta é análoga à PCA, mas com a inserção de mapeamentos não-lineares entre os espaços de dimensão ( $\mathbf{Y}$ ) original e de dimensão ( $\mathbf{T}$ ) reduzida. Dessa forma, se houver correlação não-linear entre as variáveis, a NLPCA descreverá os dados com maior acurácia e com um número menor de componentes que a PCA, a menos que não haja dados suficientes para permitir a formulação de funções mais complexas.

Em NLPCA, o mapeamento no espaço de componentes (ou fatores) é generalizado para permitir não-linearidades arbitrárias

$$\mathbf{T} = \mathbf{G}(\mathbf{Y}), \quad (6.1)$$

onde  $\mathbf{Y}$  é a matriz de dados (onde as linhas são os eventos e as colunas são as

variáveis) e  $\mathbf{G}$  é um vetor de funções não-lineares, isto é,  $\mathbf{G} = \{G_1, G_2, \dots, G_f\}$ , de forma que, se  $T_i$  representa o  $i$ -ésimo elemento de  $\mathbf{T}$ ,  $T_i = G_i(\mathbf{Y})$ . Neste caso,  $G_i$  refere-se ao  $i$ -ésimo fator não-linear de  $\mathbf{Y}$ . É possível ainda definir a transformada inversa, que restaura a dimensionalidade original dos dados a partir do espaço de dimensão reduzida. Isso é feito novamente por meio de um vetor de funções não-lineares  $\mathbf{H} = \{H_1, H_2, \dots, H_m\}$ :

$$\mathbf{Y}'_j = \mathbf{H}_j(\mathbf{T}), \quad (6.2)$$

onde  $\mathbf{Y}'$  é a matriz de dados reconstruída com a dimensionalidade original. A perda de informação é, então, medida através de  $\mathbf{E} = \mathbf{Y} - \mathbf{Y}'$ . As funções  $G$  e  $H$  devem, desta forma, ser escolhidas de forma a minimizar o erro médio quadrático (MSE) relativo a  $E$ .

### 6.3.1 Extração de componentes

Para que uma rede neural possa modelar uma função não-linear arbitrária, é necessário que a rede possua pelo menos uma camada escondida de neurônios com funções de ativação não-lineares [35]. Assim, para a extração de componentes principais não-lineares, adotam-se as arquiteturas mostradas na Figura 6.2. Estas redes neurais modelam as transformadas direta e inversa, a partir da NLPCA.

Assume-se que se deseja extrair  $f$  componentes principais não-lineares.  $G$  opera sobre cada linha de  $Y$  (ou seja, sobre cada evento) e tem  $m$  entradas (o mesmo número de colunas de  $Y$ , isto é, o número de variáveis apresentadas pelo conjunto original). A camada escondida de  $G$ , que será chamada de camada de codificação, apresenta  $M_1$  neurônios não-lineares, onde  $M_1 > f$ . A saída dessa rede é a projeção do conjunto de dados original no espaço da NLPCA e, por isso, a camada de saída da rede relativa a  $G$  deve apresentar  $f$  neurônios, os quais podem ou não ser lineares. De forma similar, é definida a rede relativa à transformada inversa  $H$ . Sua camada escondida é, portanto, denominada camada de decodificação, e sua estrutura geral pode ser vista da mesma forma que a rede relativa a  $G$ . A entrada dessa rede é a representação do conjunto de dados original no espaço da NLPCA, e a saída é a restauração do conjunto à sua dimensionalidade original. Essas duas redes (relativas a  $G$  e a  $H$ ) podem ser vistas na Figura 6.2.

Estas redes neurais multicamadas usam tipicamente treinamento supervisi-

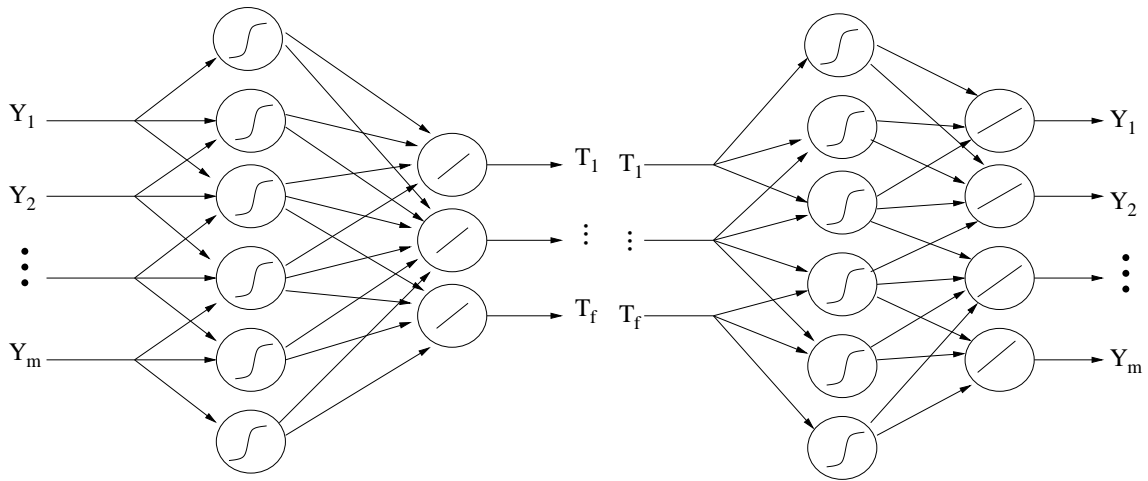


Figura 6.2: Redes que implementam as funções de codificação e de decodificação.

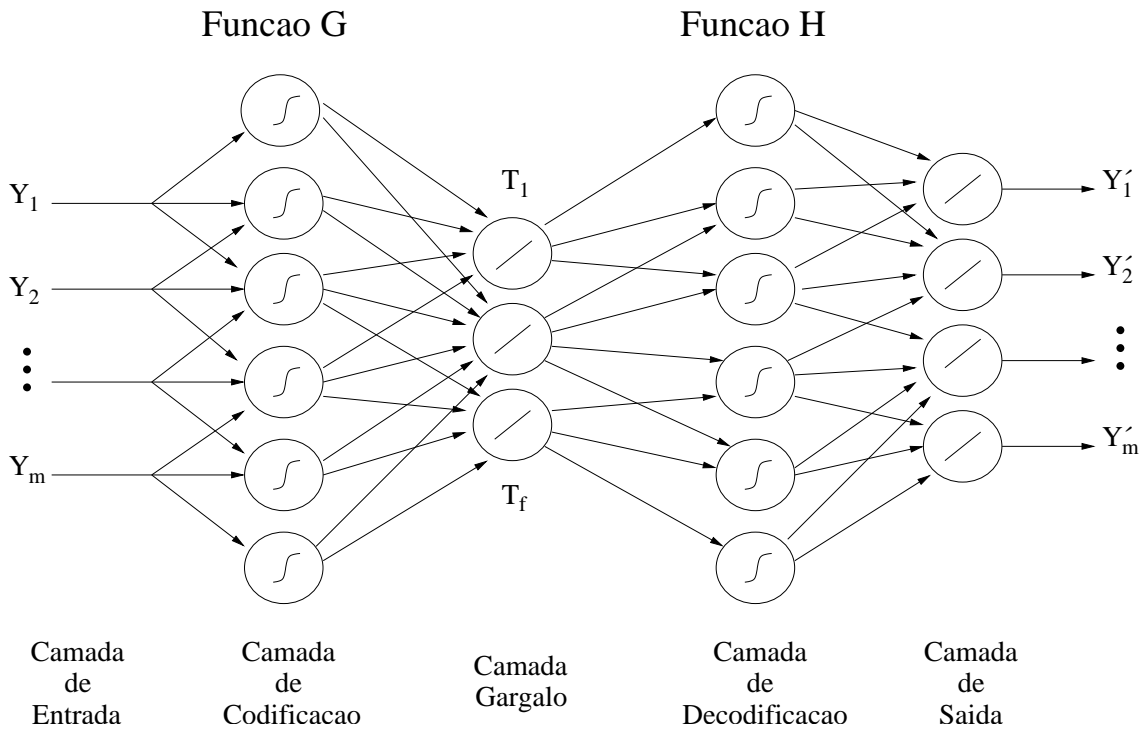


Figura 6.3: Rede auto-associativa usada para a extração de componentes principais não-lineares.



onado, com a definição de vetores-alvo de saída. Contudo, para a rede  $G$ , não se sabe a saída desejada e, para a rede  $H$ , não se sabe a entrada. Observa-se, assim, que  $T$  é tanto a saída de  $G$  quanto a entrada de  $H$ . Pode-se então combinar as duas redes formando-se o que se vê na Figura 6.3. Esse tipo de rede é denominada autoassociativa, pois tanto a entrada da rede quanto sua saída desejada são formadas pelo mesmo conjunto de dados. A função da rede é, portanto, mapear o conjunto original no conjunto de fatores NLPCA e, a partir destes, reconstruir o conjunto original com o menor erro possível [36]. A rede combinada contém três camadas escondidas: a camada de codificação, a camada central e, por fim, a camada de decodificação. A camada central, isto é, a segunda camada escondida, é denominada camada gargalo por ter a menor dimensão. O número de neurônios nesta camada determina o número de componentes principais a serem extraídos. Neste trabalho, as camadas gargalo e de saída usam neurônios lineares.

A extração propriamente dita pode ocorrer de duas formas: simultânea ou seqüencial [34]. Na extração simultânea, todos os componentes principais são extraídos colocando-se na camada gargalo um número de neurônios correspondente ao número de componentes que se deseja extrair. No modo seqüencial, usa-se apenas um neurônio na camada gargalo. Ao se fazer a extração, apenas o primeiro componente (o de maior energia) será extraído. O que se faz, então, é o processo conhecido como deflexão: reconstrói-se o conjunto original e subtraindo tal reconstrução do conjunto original. Ao se realizar novamente a extração com este conjunto de dados resultante, na verdade, estar-se-á extraíndo o segundo componente. Pode-se repetir esse processo até atingir-se o número desejado de componentes, sendo que a deflexão pode ser feita no conjunto de entrada, no de saída ou em ambos. Neste trabalho, foi usada a extração simultânea, por apresentar uma maior facilidade de implementação.

## 6.4 Resultados da extração de componentes

### 6.4.1 Dimensionamento da rede

Para se realizar a extração dos componentes principais não-lineares, usou-se uma rede neural multicamadas que foi treinada segundo o algoritmo de *back-*

*propagation* resiliente [37]. Esta rede apresenta a estrutura explicada anteriormente e as funções de ativação não-lineares dos neurônios são tangentes hiperbólicas deslocadas para a faixa entre zero e um. O número de neurônios nas camadas de mapeamento da rede extratora de componentes principais não-lineares é um parâmetro livre. Se este valor for subestimado, a acurácia da extração pode ficar prejudicada. Se este valor for superestimado, há uma maior tendência ao *over-fitting*, quando a rede assimila as flutuações estatísticas do conjunto de treino e não as funções inerentes ao processo em si. Como não existem técnicas analíticas definitivas para determinar o número de neurônios nas camadas de mapeamento, foi feita uma busca extensiva, cobrindo-se a faixa de 16 a 512 neurônios. Para cada número de neurônios, extraíram-se os componentes e fez-se a classificação com a rede que será explicada mais adiante. O melhor resultado quanto à classificação se deu ao se utilizarem 100 neurônios.

## 6.4.2 Configuração dos testes

No caso das DPs, devido à dificuldade de se medir em eventos reais, a base de dados conta com poucos eventos (apenas 115), fazendo com que a análise fosse dificultada. O maior problema decorrente desta estatística reduzida é a difícil escolha dos conjuntos de treino e de teste, necessários para a extração de componentes e subsequente classificação. Um critério de separação desses conjuntos é se basear em conhecimento especialista. Isso consiste em considerar alguma informação, fundamento teórico ou experiência prática para essa separação. Uma das aplicações deste critério é obter características dos padrões, isto é, fazer a prospecção de dados. Um modo de fazer isso é separar para o conjunto de treino os eventos que apresentam baixo nível de ruído e os demais para o conjunto de teste [3]. Nesse caso, treinava-se com padrões limpos para se testar a capacidade da rede de classificar eventos corrompidos por ruído a partir das características propriamente ditas. Assim, o conjunto de treino ficou, nesta aplicação de referência, com 61 eventos e o de teste com 54.

Usando-se esse critério, obteve-se uma discrepância muito grande para o acúmulo de energia entre o conjunto de treino e de teste, considerando-se a expansão PCA. Por energia, entende-se quantidade de informação. Enquanto que 10

componentes principais lineares representavam cerca de 79% da energia total do conjunto de treino, este mesmo número de componentes representava aproximadamente 8% da energia do conjunto de teste. Isto confirma que os conjuntos de treino e de teste apresentam estatísticas muito diferentes, fazendo com que o conjunto de teste não pudesse generalizar as características contidas no conjunto de treino. Isto é problemático quando se desenvolve um classificador, como é nosso objetivo.

Dessa forma, foi necessário escolher novos conjuntos de treino e de teste, de modo a representar melhor o universo de dados de interesse, além de deixá-los com estatísticas semelhantes, o que possibilitaria que o conjunto de teste pudesse ser uma generalização do conjunto de treino. Isso foi feito por meio de validação cruzada [38], distribuindo de forma aleatória todos os eventos disponíveis entre blocos que apresentassem aproximadamente o mesmo número de elementos. Em seguida, formaram-se diferentes agrupamentos de blocos, constituindo diferentes conjuntos de treino e teste. Para cada agrupamento, mediu-se o acúmulo de energia para um certo número de componentes extraídos. Como figura de mérito para medição do acúmulo de energia, foram usados os dez primeiros componentes principais lineares. Deve-se lembrar que os componentes são extraídos apenas do conjunto de treino. Para o presente trabalho, foram feitos 5 agrupamentos, sendo que o conjunto de treino apresentava 70 eventos e o conjunto de teste, 45.

A Tabela 6.1 mostra os resultados. Nota-se que a combinação 3 foi a que obteve uma diferença menor entre a energia representada por 10 componentes no conjunto de treino e de teste. Dessa forma, conclui-se que, dentre todas as combinações, nesta tais conjuntos apresentam as estatísticas mais próximas. Assim, esses conjuntos são escolhidos como novos conjuntos de treino e de teste, sendo, então, utilizados por toda a extensão do trabalho, incluindo o projeto de classificador.

### 6.4.3 Resultados

Fazendo a extração dos componentes lineares e não-lineares, pode-se obter os percentuais de acumulação de energia mostrados na Tabela 6.2. Esses percentuais mostram o quanto da energia (informação) total do processo foi possível reconstruir, usando-se um certo número de componentes principais. A qualidade da reconstrução

Tabela 6.1: Energia acumulada para 5 diferentes combinações do conjunto de dados usando-se 10 componentes principais lineares.

Combinação	Treino	Teste
1 <sup>a</sup>	74%	49%
2 <sup>a</sup>	74%	54%
3 <sup>a</sup>	<b>72%</b>	<b>63%</b>
4 <sup>a</sup>	73%	53%
5 <sup>a</sup>	76%	42%

é ainda corroborada se a correlação entre os eventos (pertencentes aos conjuntos de treino e de teste) e suas versões reconstruídas é significativa. Isso pode ser visto na Tabela 6.3, que mostra as médias geométricas das correlações entre cada evento e sua versão reconstruída. Pode-se ver que, para todos os casos, a correlação cresce com o número de componentes. Além disso, a correlação no conjunto de teste é sempre menor que para o conjunto de treino e que a NLPCA apresenta, de forma geral, uma maior correlação do que a PCA. Esses valores de correlação, para o caso de 9 componentes, são ilustrados na Figura 6.4. Cada linha em cada gráfico nessa figura representa a correlação entre um evento e a versão reconstruída de cada evento (inclusive ele mesmo). Assim, espera-se que a correlação entre um evento e sua própria versão reconstruída seja alta (próxima a 1 - branco) e que a correlação entre um evento e a versão reconstruída de outro evento seja baixa (próxima de 0 - preto). Espera-se que, se houver uma boa qualidade de reconstrução, as figuras relativas aos conjuntos de treino serão totalmente escuras com exceção da diagonal principal, que deve ser o mais próximo possível do branco. Do mesmo modo, se houver uma boa generalização, o mesmo deve ocorrer com o conjunto de teste. Essas condições podem ser comprovadas visualmente pela Figura 6.4. Pode-se ver também que a NLPCA é capaz de fornecer uma melhor reconstrução e uma melhor generalização, em relação à PCA.

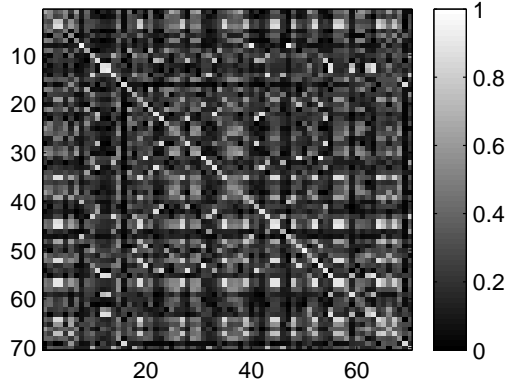
Tabela 6.2: Acúmulo de energia (informação) para PCA e NLPCA, variando-se o número de componentes.

<b>Técnica</b>	<b>PCA</b>		<b>NLPCA</b>	
<b>Nº Comp.</b>	<b>Treino</b>	<b>Teste</b>	<b>Treino</b>	<b>Teste</b>
<b>3</b>	41%	36%	85%	82%
<b>5</b>	54%	45%	91%	83%
<b>7</b>	62%	49%	93%	88%
<b>9</b>	69%	59%	94%	91%

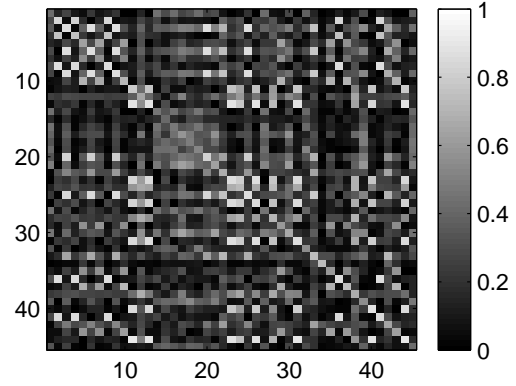
Tabela 6.3: Médias geométricas das correlações entre cada evento e sua versão re-construída.

<b>Técnica</b>	<b>PCA</b>		<b>NLPCA</b>	
<b>Nº Comp.</b>	<b>Treino</b>	<b>Teste</b>	<b>Treino</b>	<b>Teste</b>
<b>3</b>	0,46	0,36	0,95	0,62
<b>5</b>	0,61	0,53	0,97	0,70
<b>7</b>	0,66	0,59	0,98	0,79
<b>9</b>	0,72	0,67	0,98	0,82

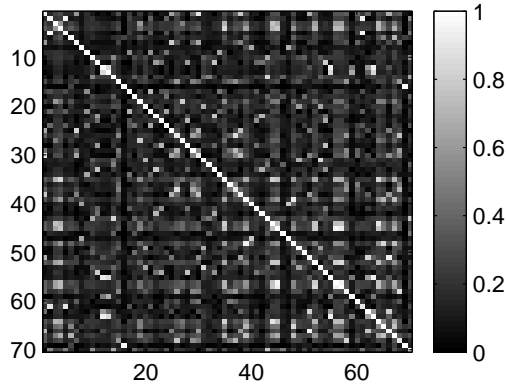
Correlação para o conjunto de treino (PCA)



Correlação para o conjunto de teste (PCA)



Correlação para o conjunto de treino (NLPCA)



Correlação para o conjunto de teste (NLPCA)

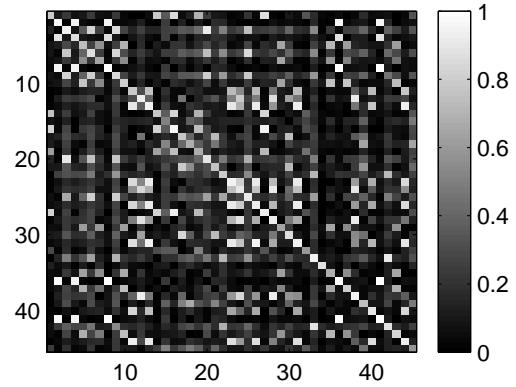


Figura 6.4: Correlação entre cada evento e sua versão reconstruída por PCA e NLPCA.

## 6.5 Resultados quanto à classificação

A classificação dos padrões foi feita através de uma rede neural de duas camadas (ambas usando tangente hiperbólica como função de ativação), treinada pelo algoritmo de *back-propagation* resiliente. A camada escondida apresentava dois neurônios. Para se utilizar os componentes principais não-lineares, alimentou-se a entrada desta rede com a saída da rede neural formada pelas duas primeiras camadas da rede responsável pela extração destes componentes (ou seja, as camadas de codificação e gargalo). Dessa forma, o classificador de DPs foi alimentado com a informação pré-processada, resultante da projeção dos eventos de DP nos componentes principais não-lineares. A saída da rede neural classificadora usou três neurônios, cada um associado a uma classe de DP. Deste modo, para se treinar a rede, o vetor-alvo de saída era formado com uma codificação 1 para saída que se deseja ativa e  $-1$  para as demais.

Usando-se conhecimento especialista [3], fez-se uma prospecção de dados para os sinais de DP usando-se PCA, tendo a classificação como sub-produto. Contudo, como os conjuntos de treino e teste se mostraram com características estatísticas diferentes, empregou-se a terceira combinação da validação cruzada, chegando-se aos resultados expostos na Tabela 6.4. Pode-se ver que a escolha dos conjuntos de treino e de teste a partir de validação cruzada possibilitou uma grande melhora na eficiência de classificação, pois a informação extraída do conjunto de treino pôde generalizar-se de forma mais eficaz.

Para comparar a técnica de PCA com a de NLPCA em termos de classificação, foram usados 3, 5, 7 e 9 componentes. A Tabela 6.5 mostra a eficiência de classificação para o conjunto de teste. Essa eficiência é calculada como a média

Tabela 6.4: Comparação dos resultados de eficiência de classificação obtidos na prospecção de dados (PD) e por validação cruzada (VC). Em todos os casos, usou-se o conjunto de teste.

<b>Componentes</b>	<b>P.D.</b>	<b>V.C.</b>
<b>5</b>	73%	95%
<b>7</b>	86%	100%
<b>9</b>	91%	100%

geométrica das probabilidades de detecção de cada classe. É possível ver que, para 3 e 5 componentes, a rede que usa PCA apresenta uma eficiência de 95 %, enquanto que para todos os números de componentes testados, foi alcançada uma eficiência de 100 % usando-se NLPCA. Como existem apenas 45 eventos no conjunto de teste, os valores na Tabela 6.5 garantem que, com NLPCA, uma eficiência de classificação superior a 97 % é alcançada, devido à incerteza neste cálculo.

Para se verificar a qualidade da classificação, foi gerada uma matriz  $3 \times 3$  (mostrada na Figura 6.5), onde cada elemento é um histograma que mostra a distribuição de cada saída (1, 2 e 3) para cada classe. Por exemplo, o primeiro histograma mostra como é a distribuição da saída do neurônio 1 da última camada da rede classificadora, quando os eventos a serem classificados pertencem à classe FL. Pode-se ver que, para cada classe, existe sempre um neurônio com valores muito próximos de 1 e os demais com valores muito próximos de  $-1$ , indicando que há uma boa separação das classes. Pode-se ver também que os histogramas que apresentam valores próximos de 1 são aqueles que estão na diagonal principal da matriz, indicando que as classes foram bem identificadas.

## 6.6 Conclusão

Com o ICM System, é necessário que haja um especialista para fazer a classificação do tipo do DP apenas olhando para o mapa. Desse modo, percebe-se que o sistema proposto, além de implementar a função básica do ICM System (plotar o mapa de comportamento de DP), ainda se propõe a realizar a classificação de modo automático. É claro que um sistema de classificação automática deve agir como um

<b>Técnica</b>	<b>PCA</b>	<b>NLPCA</b>
<b>3 comp.</b>	95 %	100 %
<b>5 comp.</b>	95 %	100 %
<b>7 comp.</b>	100 %	100 %
<b>9 comp.</b>	100 %	100 %

Tabela 6.5: Comparação da eficiência de classificação usando-se as técnicas de PCA e de NLPCA.



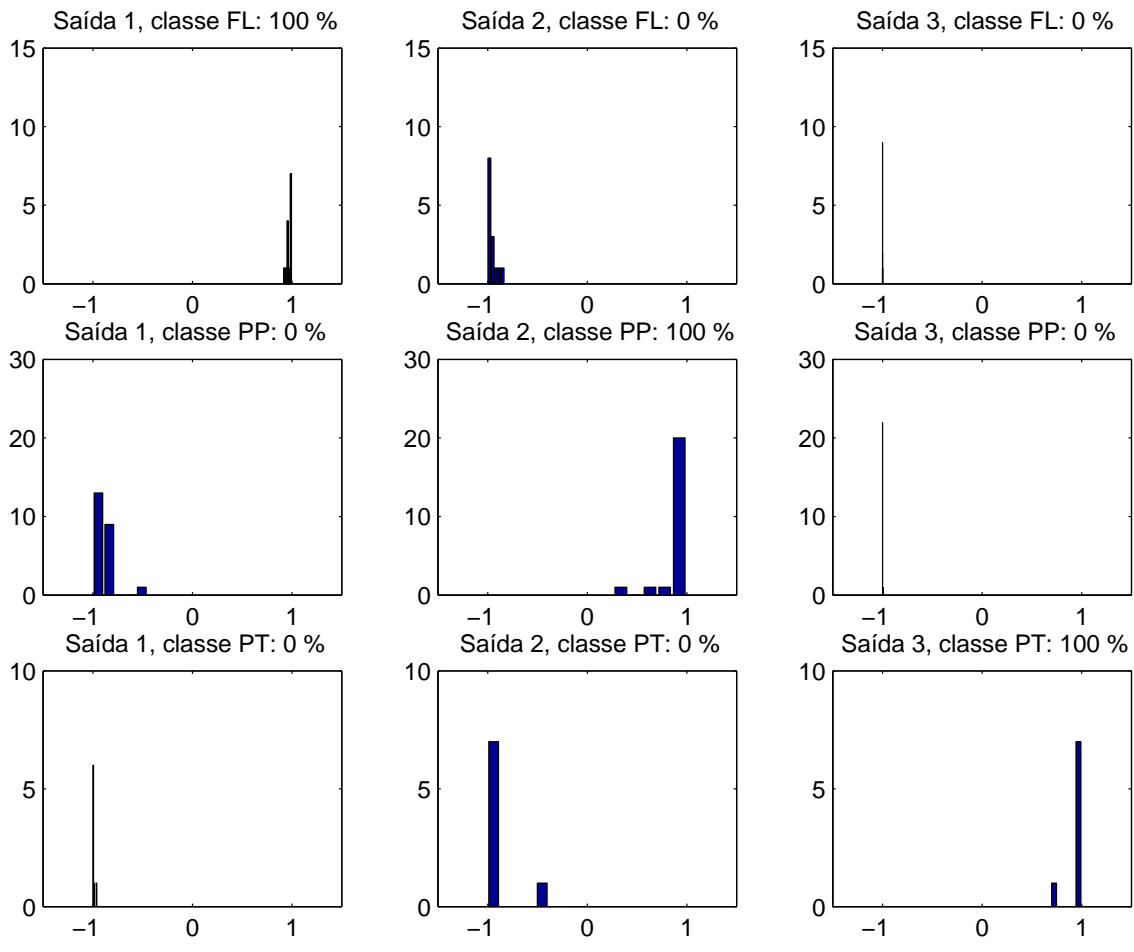


Figura 6.5: Matriz de histogramas que verifica a real separação das classes.

apoio à tomada de decisão, mas nunca deve atuar sozinho.

Obteve-se uma eficiência de classificação maior do que 97% ao se usar NLPCA para a compactação da informação do processo relativo às DPs. Com essa compactação, foi possível usar apenas 3 componentes (de um total de 1024 variáveis originalmente disponíveis). Com a PCA, por outro lado, foram necessários 7 componentes para se obter esse mesmo nível de eficiência de classificação. A técnica de NLPCA apresenta uma melhor reconstrução dos eventos do processo analisado. Isso pode ser visto tanto em termos de percentual de energia reconstruído, quanto em termos da correlação entre cada evento e sua versão reconstruída. Isso indica que o NLPCA possui um maior poder de compactação, pois, com poucos componentes, preserva a maior parte da informação (energia) necessária para se classificar os sinais de DP.

# Capítulo 7

## Implementação do Sistema

### 7.1 Introdução

Este capítulo tem como objetivo descrever o *software* desenvolvido para a execução das tarefas de supressão de ruído, detecção e classificação. Cada uma das partes será explicada, bem como a forma através da qual estas partes se integram. A Seção 7.2 dá uma idéia geral de como será o uso do sistema por um usuário comum. A Seção 7.3 descreve detalhadamente o *software* desenvolvido, mostrando todas as suas telas e etapas do processamento.

### 7.2 Visão geral do sistema

O sistema deve ser utilizado da seguinte forma:

1. Ajusta-se o osciloscópio digital de modo a capturar um ciclo de potência. Como isso equivaleria a aproximadamente 16,67 ms e não é possível definir este tamanho de janela na prática, a janela deve ser de 20 ms. Em alguns tipos de osciloscópios, isso também já define a frequência de amostragem. Deve-se tomar o cuidado de ajustar, através do *trigger* do osciloscópio, para que o começo do ciclo esteja o mais próximo possível do começo da janela de aquisição.
2. Grava-se, então o maior número possível de sinais, armazenando-os em arquivos individuais.

3. Executa-se o programa CleanDetect.
4. Ajustam-se os parâmetros de maneira que as tarefas desejadas sejam efetuadas.
5. Após o processamento (que envolve supressão de ruído, detecção e classificação) tem-se o mapa que descreve o comportamento do fenômeno de descargas parciais no equipamento e a classe a que pertence o padrão medido.

## 7.3 *Software* desenvolvido

O *software* foi integralmente desenvolvido em Matlab<sup>TM</sup> com o auxílio da ferramenta de criação de GUIs (*Graphic User Interfaces*) denominada Guide. O sistema apresenta uma interface amigável com o usuário, de forma que todos os comandos são acionados através do *mouse*.

### 7.3.1 Tela principal

Esta tela (mostrada na Figura 7.1) é a responsável por ativar todos os principais comandos e atuar como visualizador dos principais resultados (relativos à supressão de ruído, detecção e classificação). Apresenta os seguintes componentes:

1. Janela do tipo *browser* para selecionar o diretório onde estarão armazenados os arquivos de dados de sinais de DP.
2. Quadro de seleção de métodos para supressão de ruído. Os métodos listados são: subtração espectral, filtragem de Wiener, *wavelet shrinkage*, filtragem adaptativa e ICA. É possível marcar qualquer um destes métodos (ou um conjunto deles). Para cada método, existe um botão (com a etiqueta “*Settings...*”) para acionar uma tela de configuração de parâmetros.
3. Quadro para ativação da função de detecção baseada em filtros casados. Neste quadro, há também um botão (com a etiqueta “*Settings...*”) para acionar uma tela de configuração de parâmetros.
4. Quadro para ativação da função de classificação baseada em redes neurais. Neste quadro, há também um botão (com a etiqueta “*Settings...*”) para acionar uma tela de configuração de parâmetros.

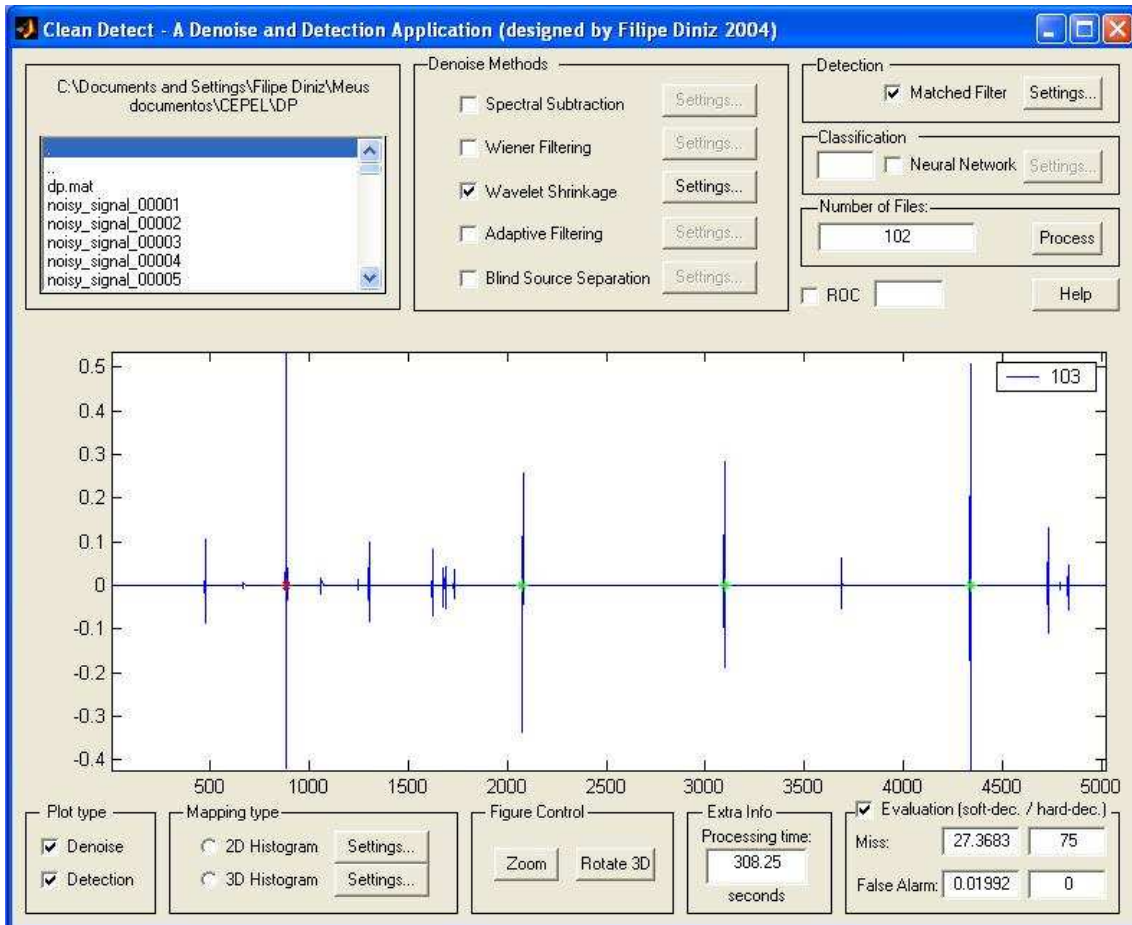


Figura 7.1: Tela principal do programa CleanDetect, o qual tem as funções de controlar os métodos de supressão de ruído, detecção e classificação e mostrar os resultados.

5. Quadro para informar o número de arquivos a serem processados. Esse quadro é, a princípio, preenchido com o número total de arquivos no diretório atual (selecionado no *browser*) do item 1. Há nesse quadro um botão (com a etiqueta “*Process*”). Este botão é responsável por ativar toda a cadeia de processamento desejada.
6. Janela de gráfico para visualizar gráficos referentes à supressão de ruído e detecção. É também nessa janela que será mostrado o mapa similar ao do ICM System.
7. Quadro para ativar e desativar a visualização dos gráficos de supressão de ruído e detecção.
8. Quadro para selecionar se o mapa será mostrado no formato 2D (através de cores representando a taxa de repetição) ou no formato 3D.
9. Quadro com controles para a janela de gráficos do item 6. Há um botão para *zoom* e outro para rotação (usada no caso o mapa ser visualizado no formato 3D).
10. Quadro para informar o tempo de processamento em segundos.
11. Quadro para avaliação do sistema de detecção. Aqui, há uma *check-box* com etiqueta “*Evaluation (soft dec. / hard dec.)*”. Se esta *check-box* estiver marcada, serão exibidos valores para *miss* e falso alarme (baseados em métodos *soft* e *hard* - o que foi visto no capítulo 4).

### **7.3.2 Ajuste de parâmetros para métodos de supressão de ruído**

Para cada método de supressão de ruído, há uma tela para ajuste de parâmetros. Para que o botão que possibilita o acionamento desta tela seja ativado, é necessário marcar a *check-box* relativa e este método.

Em todas essas telas, há dois componentes em comum. O primeiro é a janela de descrição: se o usuário clicar com o botão direito do *mouse* sobre qualquer parâmetro, aparecerá, nesta janela, uma curta definição do parâmetro em questão.

O segundo é uma interface onde será possível visualizar o efeito da variação de um parâmetro no resultado final da supressão de ruído. Cada método de supressão de ruído possui sua própria interface, contudo todas têm componentes em comum. São estes:

1. **Caixa de texto para carregar o sinal corrompido.** Este sinal é originalmente armazenado em uma variável de Matlab a pode ser carregado em sua forma original ou transposta. Ao ser carregado, digitando-se o nome da variável e clicando-se no botão *Load*, o número total de amostras é mostrado.
2. **Caixa de texto para carregar o sinal original.** Esse sinal não estará disponível em uma situação real, mas é crucial para determinar o desempenho do sistema. Quando este sinal estiver disponível, como é o caso se o sinal corrompido for sintético, o *check-box Evaluation* deve ser marcado.
3. **Critérios para avaliação de desempenho.** Os critérios (que foram descritos no Capítulo 4) são mostrados ao usuário se o *check-box Evaluation* for marcado. A SNR é medida antes e depois do *denoising* assim como a diferença entre estes valores. O mesmo é feito para correlação cruzada com o sinal original.
4. **Gráfico para o sinal original (ideal) e para o sinal processado.** Esse gráfico é atualizado tão logo um parâmetro tenha seu valor alterado.

#### 7.3.2.1 Subtração espectral e filtro de Wiener

Nesta tela (mostrada na Figura 7.2), os parâmetros vistos nas Seções 3.2 e 3.3 do Capítulo 3 poderão ser ajustados. São estes:

1. ***Window size*:** tamanho da janela que será usada para segmentar o sinal de entrada.
2. ***Overlapping factor*:** fator de superposição que será aplicado na janela do item acima.
3. ***No signal zone: begin*:** começo da região do sinal de entrada onde o sinal de interesse não é significativo ou é ausente.

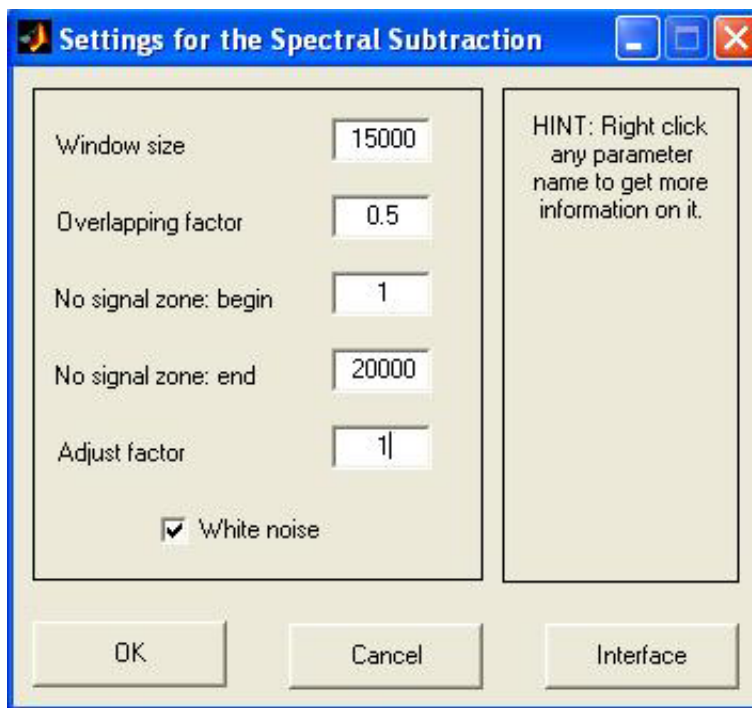


Figura 7.2: Caixa de diálogo para ajuste dos parâmetros da subtração espectral.

4. **No signal zone: end:** fim da região definida no item acima.
5. **Adjust factor:** é o fator de correção para a estimação do espectro do ruído. Essa correção é devida ao fato de que o número de amostras é pequeno demais para que a estimativa do espectro do ruído seja acurada.
6. **White noise:** se essa *check-box* for marcada, o espectro do ruído será forçado a ser constante, o que é mais adequado para estimação de ruído considerado branco.

Se o botão *Interface* for acionado, será chamada uma tela para visualizar o efeito da variação de um parâmetro no resultado final da subtração espectral (ou do filtro de Wiener). Essa tela faz parte de um *toolbox* de supressão de ruído descrita em [39], está mostrada na Figura 7.3 e apresenta os seguintes componentes:

1. **Controle de janelamento:** É possível definir o número de janelas e seu fator de superposição. Essas janelas são usadas para a estimação da PSD do sinal. Ao se definirem estes dois valores, o número de janelas é mostrado.
2. **Opção de “White noise”.** Se o *radiobutton* “White Noise” for marcado, a estimação do PSD do ruído será forçada a ser constante.



3. **Definição de intervalo onde o sinal não é significativo.** Usando os dois “*sliders*”, é possível ajustar o intervalo onde a presença do sinal de interesse não é significativa e, por isso, o sinal é formado basicamente por ruído. Esse intervalo é usado para estimar a PSD do ruído.
4. **Controle do fator de ajuste.** A estimação do PSD é afetada pelo pequeno número de amostras contidas no intervalo definido no item acima. Para ajustar o valor da PSD do ruído manualmente, a caixa de texto *Adjust factor* pode ser usada.
5. **Opção de subtração espectral.** Se o *radiobutton Spectral subtraction* estiver marcado, o método de supressão de ruído é definido como subtração espectral. Caso contrário, este é definido como filtro de Wiener.

### 7.3.2.2 *Wavelet shrinkage*

Esta tela (mostrada na Figura 7.4), os parâmetros vistos na Seção 3.4 do Capítulo 3 poderão ser ajustados. São estes:

1. ***Number of levels***: é o número de níveis de decomposição na árvore binária da *wavelet*. O número de bandas no domínio *wavelet* é esse número incrementado em 1.
2. ***Wavelet size***: é a ordem da *wavelet*-mãe.
3. ***Wavelet family***: é a forma da *wavelet*-mãe. Esse parâmetro depende da forma do sinal de interesse. As opções são: Daubechies, Symlets, Coiflets e Biortogonais.
4. ***Shrinkage method***: é o método usado para o processamento dos coeficientes *wavelet*, o que resultará em supressão de ruído. Isso indica o modo através do qual os limiares para os coeficientes serão escolhidos. As opções são: método heurístico, *VisuShrink*, *SureShrink* e híbrido.
5. ***Thresholding***: é o modo através do qual os coeficientes *wavelet* serão modificados pelos limiares calculados pelo método selecionado no parâmetro anterior. As opções são: *hard-thresholding* e *soft-thresholding*.

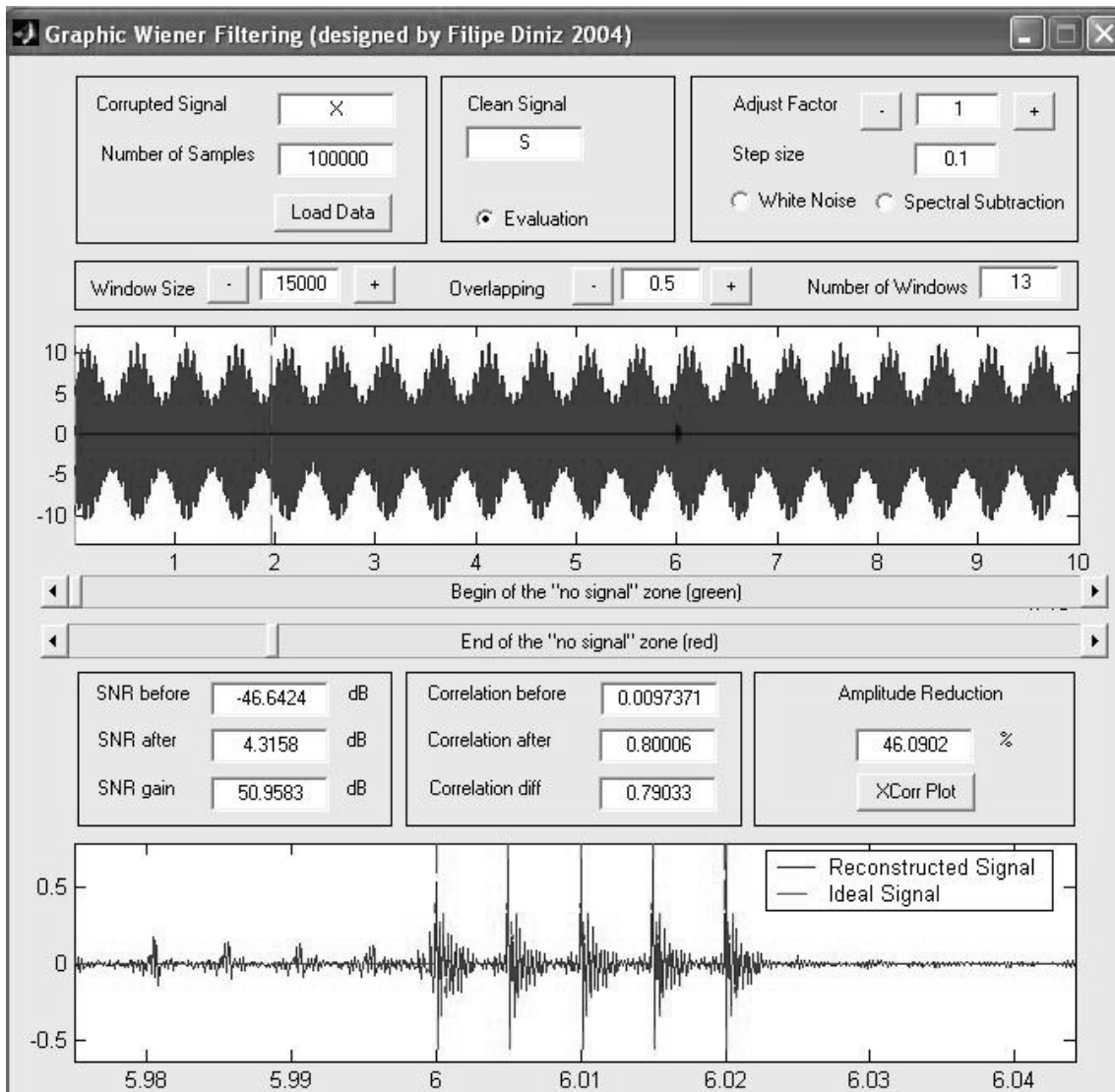


Figura 7.3: Interface para visualização dos efeitos da variação dos parâmetros sobre a subtração espectral e do filtro de Wiener.

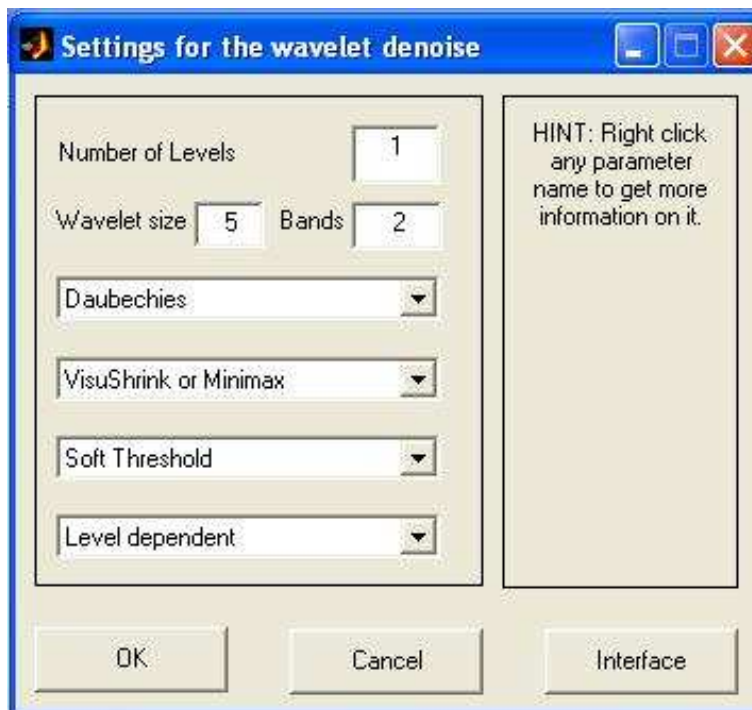


Figura 7.4: Caixa de diálogo para ajuste dos parâmetros da transformada *wavelet*.

6. **Level dependency:** independência do nível significa que os limiares são calculados considerando que o ruído é o mesmo para todas as bandas do domínio *wavelet*. Neste caso, é considerado o nível de ruído presente na última banda. Se houver uma dependência de nível, os limiares são calculados considerando que cada banda *wavelet* tem seu próprio nível de ruído. Isso é válido apenas para o método *VisuShrink*.

Se o botão *Interface* for acionado, será chamada uma tela para visualizar o efeito da variação de um parâmetro no resultado final da supressão de ruído baseada em *wavelet shrinkage*. Essa tela faz parte de um *toolbox* de supressão de ruído descrita em [39], está mostrada na Figura 7.5 e apresenta os seguintes componentes:

1. **Controle da *wavelet*-mãe.** O usuário pode selecionar a família da *wavelet*, bem como o número de níveis de decomposição e a ordem.
2. **Controle do método de *shrinkage*.** É o método para cálculo dos limiares.
3. **Controle de *thresholding*.** Escolhe-se entre *hard-thresholding* e *soft-thresholding*.
4. **Ajuste manual dos limiares.** Depois do *denoising*, o usuário pode manualmente selecionar qualquer uma das bandas no *combo-box Band* e, usando o

*slider*, colocar o limiar em qualquer altura.

5. **Gráfico das bandas *wavelet*.** Esse gráfico mostra as bandas *wavelet* separadas por linhas verdes tracejadas. As linhas vermelhas tracejadas indicam os limiares para cada banda *wavelet*. Se o usuário quiser verificar o efeito do *denoising* sob a configuração atual, o *radiobutton* ***Shrink*** deve ser marcado.

### 7.3.2.3 Filtragem adaptativa

Esta tela (mostrada na Figura 7.6), os parâmetros vistos na Seção 3.5 do Capítulo 3 poderão ser ajustados. São estes:

1. ***Number of iterations***: é o número máximo de iterações que o algoritmo adaptativo realiza para chegar à solução. Se esse número for alcançado, o algoritmo pára.
2. ***Number of coefficients***: é o número de coeficientes do filtro adaptativo.
3. ***Delay***: é o tamanho do atraso ao qual o sinal de referência é submetido.
4. ***Tolerance***: é a tolerância relativa ao MSE sob a qual o algoritmo adaptativo pára.
5. ***Adaptive algorithm***: é o algoritmo usado para fazer com que os coeficientes do filtro adaptativo convirjam para uma solução. As opções são: *Least Mean Squares* (LMS), *Normalized LMS*, *Recursive Least Squares*, e *LMS-based set-membership affine projection* [18].
6. ***Algorithm parameters***: os parâmetros que poderão ser ajustados dependerão de qual algoritmo adaptativo foi selecionado.

Se o botão *Interface* for acionado, será chamada uma tela para visualizar o efeito da variação de um parâmetro no resultado final da supressão de ruído baseada em filtragem adaptativa. Essa tela faz parte de um *toolbox* de supressão de ruído descrita em [39], está mostrada na Figura 7.7 e apresenta os seguintes componentes:

1. **Controle do sinal de referência.** Esse sinal pode ser carregado da mesma forma como foi feito com o sinal corrompido.

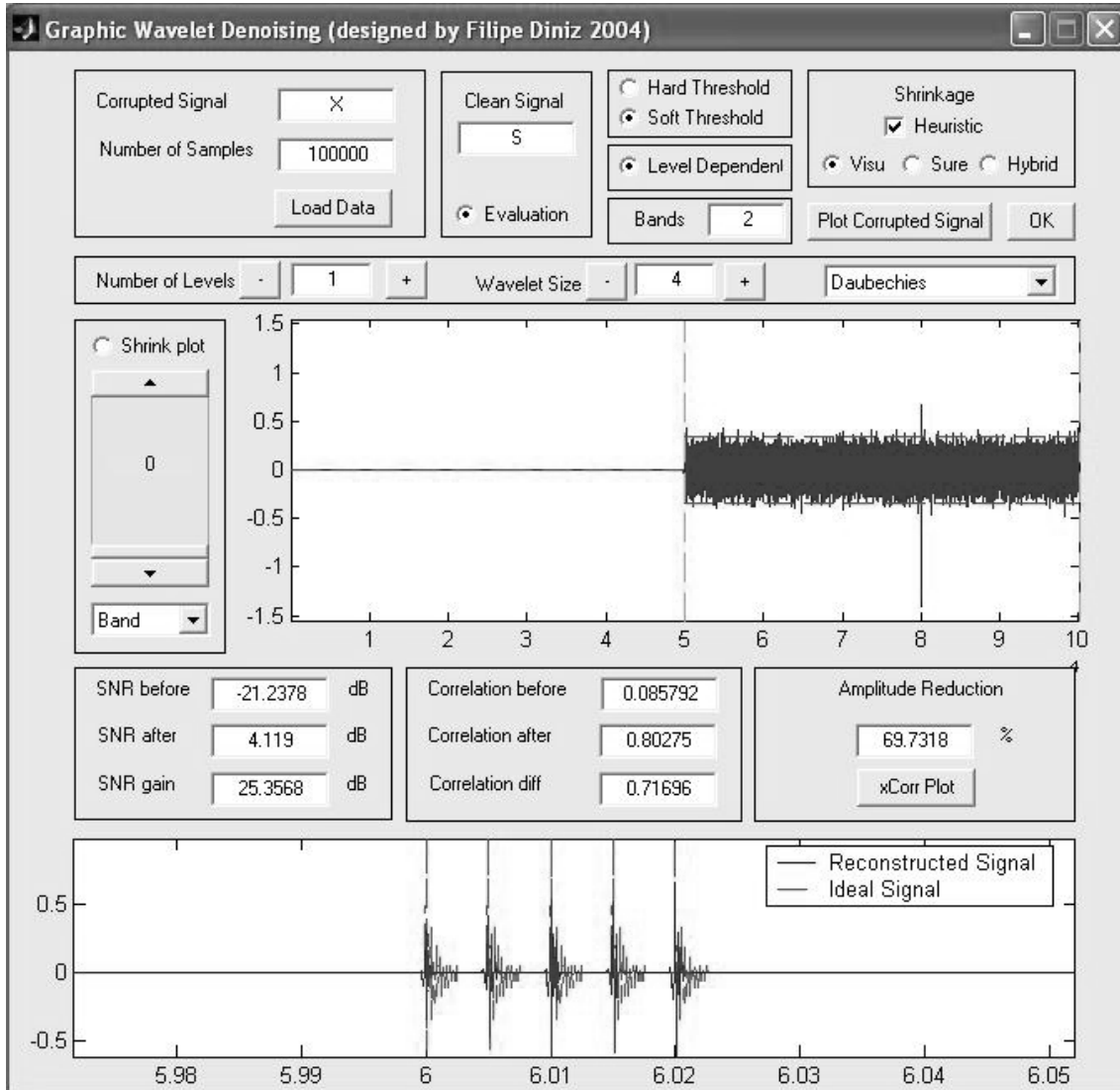


Figura 7.5: Interface para visualização dos efeitos da variação dos parâmetros sobre a supressão de ruído com base em *wavelet shrinkage*.

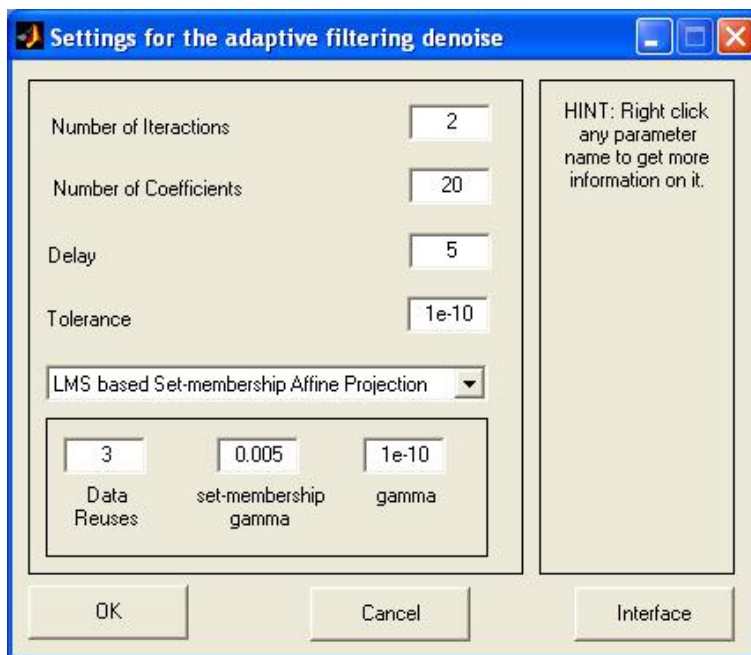


Figura 7.6: Caixa de diálogo para ajuste dos parâmetros da filtragem adaptativa.

2. **Controle do algoritmo adaptativo.** O usuário pode selecionar, através de uma *combo-box*, o algoritmo adaptativo. A tolerância em relação ao MSE também pode ser determinada. Os parâmetros relativos a cada algoritmo também podem ser ajustados.
3. **Características do filtro.** O usuário pode definir o número de coeficientes e o atraso para o sinal de referência.
4. **Número de iterações.** O número de iterações também pode ser determinado pelo usuário.
5. **Gráficos.** O gráfico do MSE (em dB) e o gráfico de visualização dos coeficientes do filtro podem ser habilitados ou desabilitados.

#### 7.3.2.4 ICA

Para implementar a separação cega baseada em ICA, usou-se o *software* contido no portal citado em [19]. O *script* de extensão “.m” é denominado *fastica.m* e a interface gráfica (mostrada na Figura 7.8) é *fasticag.m*.

O *script* básico, o qual é manipulado pela interface gráfica, tem como parâmetros, em sua forma mais simples, apenas a matriz de misturas descrita na Seção

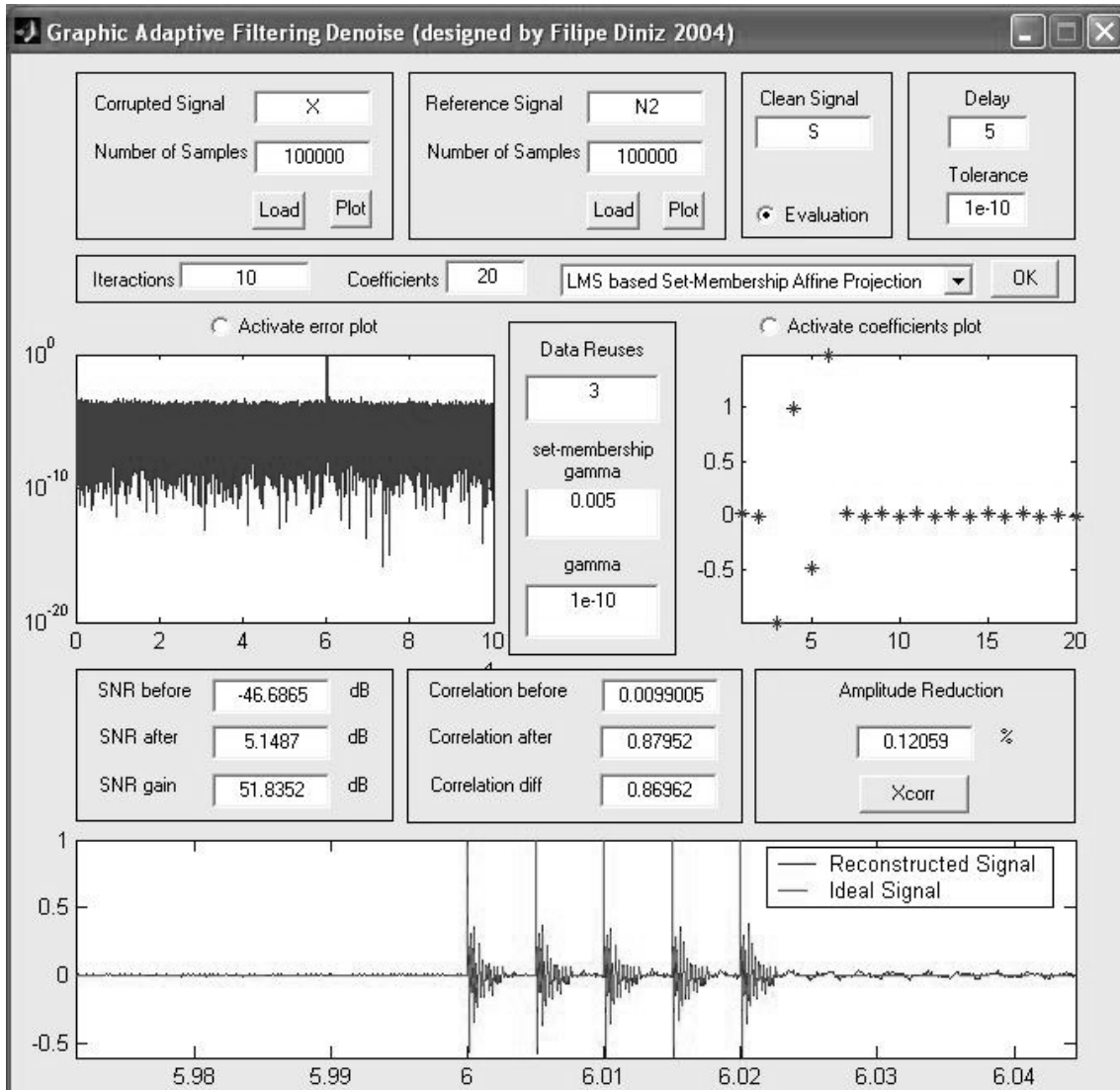


Figura 7.7: Interface para visualização dos efeitos da variação dos parâmetros sobre a filtragem adaptativa

3.6 do Capítulo 3. Outras opções são o tipo de não-linearidade (que pode ser uma parábola, função cúbica, tangente hiperbólica ou uma curva gaussiana) e parâmetros de controle de convergência citados em [19].

### 7.3.3 Ajuste de parâmetros para detecção

A tela (mostrada na Figura 7.9) apresenta os seguintes componentes:

1. Janela do tipo *browser* para selecionar o arquivo de extensão “.dat” onde está armazenado o sinal padrão que se deseja detectar.
2. Caixa de texto denominada *Matched filter signal filename*, onde é mostrado o nome do arquivo selecionado. Juntamente com essa caixa de texto, estão o botão *Plot* para plotar esse sinal e o botão *Load* para carregar o sinal selecionado para ser utilizado pelo programa.
3. Caixa de texto denominada *Number of signal events*, onde é mostrado o número de eventos que contém sinal mais ruído que devem ser usados para plotar a curva ROC.
4. Caixa de texto denominada *Number of noisy events*, onde é mostrado o número de eventos que contém ruído que devem ser usados para plotar a curva ROC.
5. Caixa de texto denominada *Desired detection probability*, onde é mostrada a probabilidade de detecção desejada. Esse valor será usado para calcular, sobre a curva ROC, o ponto em que o limiar de detecção oferece tal probabilidade.
6. Caixa de texto denominada *Number of Bins*, onde é mostrado o número de canais para os histogramas utilizados para montar as curvas ROC.
7. Caixa de texto denominada *SNR*, onde é mostrado o valor para a razão-sinal ruído dos eventos que são constituídos de sinal mais ruído. Juntamente com essa caixa de texto, existe um botão (*Plot noisy event*) para que um exemplo de evento constituído de sinal mais ruído seja plotado.
8. Janela de gráfico (*Matched Filter outputs*), para mostrar os histogramas que representam as saídas do filtro casado para quando o evento de entrada apresenta o sinal de interesse (histograma escuro) ou não (histograma claro). Nesse



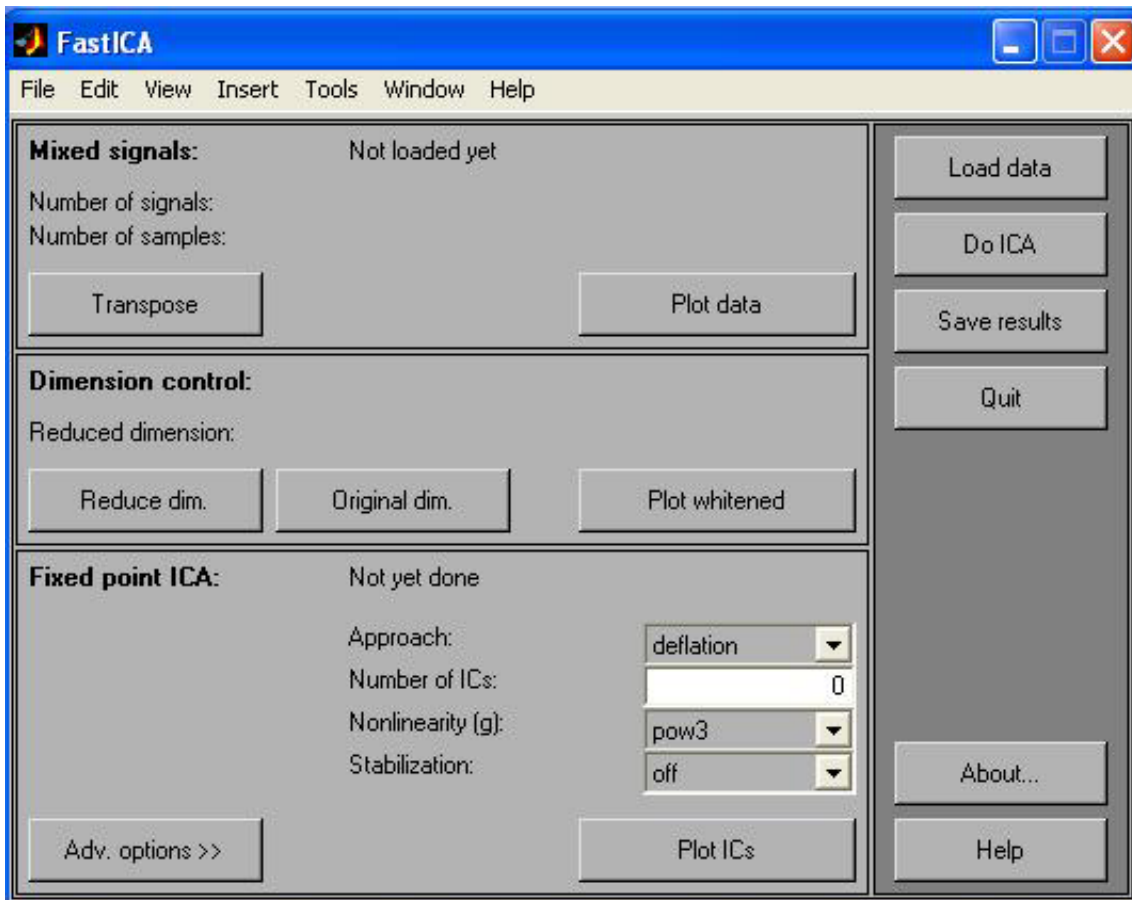


Figura 7.8: Interface gráfica que implementa o algoritmo de FastICA.

gráfico, também é mostrada uma linha pontilhada que representa o limiar de detecção.

9. O limiar de detecção mostrado no gráfico do item anterior pode ser controlado pelo *slider* denominado *Neyman Pearson decision threshold*. Nesse *slider*, também é mostrado o valor desse limiar.
10. Janela de gráfico denominada *ROC curve - Receiver Operating Characteristic*, onde são mostradas as curvas ROC de treino e de teste. Sobre a curva ROC, o limiar de detecção é calculado.
11. Caixa de texto denominada *Detection probability*, onde é mostrada a probabilidade de detecção obtida pela curva ROC de forma a ser a mais próxima possível da probabilidade de detecção desejada.
12. Caixa de texto denominada *False alarm probability*, onde é mostrada a probabilidade de falso alarme obtido pela curva ROC.
13. Caixa de texto denominada *Error probability*, onde é mostrada a probabilidade de erro obtida pela curva ROC. Esse probabilidade de erro é a combinação das probabilidades de falso alarme e de perda do alvo (*miss*).
14. Botão *Process* para que todo o cálculo do limiar de detecção seja realizado.

O funcionamento desta interface se dá da seguinte forma:

1. Seleciona-se o arquivo que contém o padrão do sinal de interesse a ser detectado, clicando-se duas vezes em cima de seu nome.
2. O nome desse arquivo aparece na caixa de texto *Matched filter signal filename*.
3. Clica-se em *Load*. O sinal é, então, carregado para uma variável interna.
4. Selecionam-se os valores para o número de eventos de sinais ruidosos e de ruídos, para a probabilidade de detecção, para o número de canais dos histogramas e o valor da razão sinal ruído.
5. Clica-se em *Process* para que o cálculo do limiar de detecção seja realizado. Baseado na probabilidade de detecção desejada e na curva ROC obtida, calcula-se o ponto mais próximo do desejado.

6. Ao final desse cálculo, serão plotadas as curvas ROC de treino e de teste e serão mostradas as probabilidades reais de detecção, de falso alarme e de erro.
7. Por fim, será colocada uma linha pontilhada no gráfico dos histogramas. Essa linha denota a posição do limiar de detecção. Seu valor também é mostrado sobre o *slider*.

### 7.3.4 Avaliação do sistema

Para avaliar o desempenho do sistema de detecção, não se poderiam usar os métodos convencionais. O que intuitivamente se espera de um sistema de detecção é que esse seja capaz de apontar se um evento é composto por sinal mais ruído ou apenas ruído. No caso das DPs, o sistema deve ser capaz de apontar a posição dentro do ciclo de potência, onde ocorreu o evento de sinal mais ruído. O ciclo de potência é denotado pelo sinal de entrada do sistema.

Desse modo, o sistema de detecção deve receber o sinal que representa um ciclo de potência e especificar em que amostra começa um evento de DP. Contudo, é extremamente difícil se dizer exatamente a posição dessa amostra. Na prática, não é necessário acertar exatamente, pois há uma margem de erro. Essa margem é definida pela largura dos canais do histograma ou mapa descrito no Capítulo 5.

Com base nisso, estabeleceram-se dois tipos de critérios de avaliação: um denominado *soft decision* e outro, *hard decision*. Esses critérios são baseados nas seguintes definições: cada ponto onde realmente começa um evento de DP é denominado “resposta certa” e cada ponto onde o sistema considera o começo de um evento de DP (estando ele certo ou não) é denominado “ponto de detecção”.

O critério por *soft decision* é dividido em *miss* e falso alarme. Para o *miss*, calcula-se, para cada “resposta certa”, a distância para o “ponto de detecção” mais próximo. Em seguida, somam-se essas distâncias e divide-se o resultado pelo número de “respostas certas”. Isso dá uma idéia da média da distância entre cada evento real e o ponto onde se considerou começar um evento. Quanto menor, mais próximos os eventos reais estarão dos eventos detectados. Se houver um evento real de DP que não foi detectado (ou seja, não há “pontos de detecção” próximos), sua distância será grande e isso fará com que o valor deste critério seja elevado.

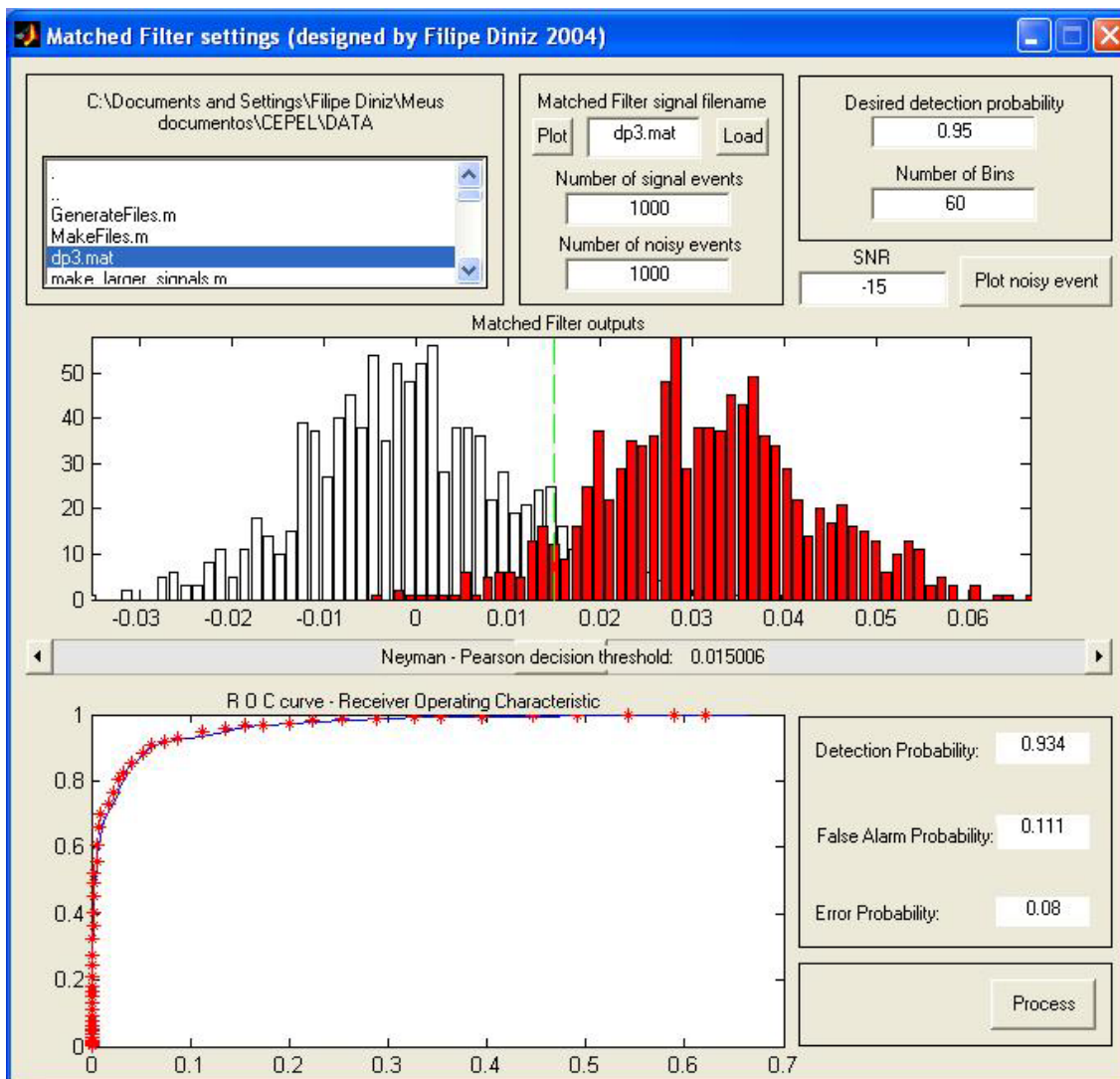


Figura 7.9: Interface gráfica que auxilia no ajuste dos parâmetros do filtro casado.

Para o falso alarme, repete-se o mesmo procedimento de forma invertida. calcula-se, para cada “ponto de detecção”, a distância para a “resposta certa” mais próxima. Em seguida, somam-se essas distâncias e divide-se o resultado pelo número de “pontos de detecção”. Isso dá uma idéia da média da distância entre cada evento detectado e o ponto onde realmente começa um evento. Quanto menor, mais próximos os eventos reais estarão dos eventos detectados. Se houver um ponto em que se considerou haver um evento sem que um evento real de DP tenha ocorrido (ou seja, não há “respostas certas” próximas), sua distância será grande e isso fará com que o valor deste critério seja elevado.

Para o critério de *hard decision*, foi acrescentado à definição do critério de avaliação o fato de haver uma margem de erro para a detecção. Inicialmente, o valor foi definido como o comprimento do sinal de entrada (que representa um ciclo de potência) dividido por 256, por ser esse o número de canais do histograma do mapa do ICM System. Isso pode ser justificado pelo fato de que, se houver um erro menor que esse, o mapa gerado não será alterado, pois o evento em questão cairia no mesmo canal. Entretanto, com base em conhecimento especialista, essa margem de erro foi definida como o comprimento do sinal de entrada dividido por 64. Isso foi feito porque, com essa margem de erro, os resultados ainda seriam satisfatórios.

## 7.4 Conclusão

Neste capítulo, foram descritos o *software* e as interfaces gráficas desenvolvidas para controlar a executar todos os métodos de supressão de ruído e detecção de sinais discutidas nos Capítulos 3 e 5.

Foram criados também critérios de avaliação para se verificar o desempenho do sistema. Tais critérios foram desenvolvidos com base no fato de que critérios convencionais de avaliação de sistemas de detecção não puderam ser utilizados devido a peculiaridades da aplicação.

# Capítulo 8

## Conclusão

### 8.1 Resultados

O sistema implementado neste trabalho é executado em três etapas: supressão de ruído, detecção de sinais e classificação de padrões.

A supressão de ruído foi implementada usando-se as técnicas mais conhecidas para este fim: subtração espectral, filtragem de Wiener, transformada *wavelet*, filtragem adaptativa e separação cega por ICA. Cada uma apresenta vantagens e desvantagens, as quais foram enumeradas no final do Capítulo 3. Quando tais técnicas foram efetivamente usadas, cada uma delas se mostrou mais propícia a um tipo específico de interferência. Por exemplo, o filtro de Wiener se mostrou mais adequado no tratamento de ruído formado por DSI. A transformada *wavelet* se mostrou mais adequada à supressão de ruído branco. O filtro adaptativo e a separação cega por ICA se mostraram bastante robustos. Contudo, possuem a desvantagem de precisarem de mais de um canal de informação, o que representa uma grande dificuldade em termos práticos.

Tais resultados deram margem ao surgimento da idéia de se combinar métodos. Isso foi realizado, e os resultados relativos à combinação de filtragem de Wiener com a transformada *wavelet* foram vistos. Como se suspeitava, a combinação destes métodos se mostrou mais vantajosa do que o uso de um método de forma individual.

A detecção de sinais foi implementada usando-se uma técnica baseada em filtros casados. Isso foi feito desta forma pois, se for usado, por exemplo, um limiar

de amplitude (como é implementado no ICMSystem), o sistema se tornaria bastante vulnerável a ruídos. A grande vantagem de se basear em filtros casados é o fato de esta técnica ter como fundamento a correlação cruzada com o padrão que se deseja detectar. Quando o ruído que corrompe o sinal de interesse apresenta uma amplitude relativamente pequena, o uso do filtro casado, por si só, já garante que a detecção seja feita de maneira correta. Quando o ruído passa a ser bem mais significativo que o sinal, o uso de uma etapa de pré-processamento com supressão de ruído se mostrou essencial para que a detecção fosse possível (ou pelo menos não tão sujeita a erros). Após a etapa da detecção de sinais, foi possível plotar um gráfico, como no ICMSystem, contendo informações de fase, carga e taxa de repetição. Esse tipo de gráfico descreve o comportamento do sinal de DP no equipamento em questão. Esse gráfico é, então, submetido a uma etapa de classificação.

A classificação é feita, então, sobre o gráfico gerado na etapa de detecção. Como este gráfico é composto por uma matriz de 256 linhas e 256 colunas, torna-se necessário uma etapa de pré-processamento para redução da dimensionalidade. Isso é necessário para aliviar o custo computacional. Para esta tarefa, usou-se a análise por componentes principais não lineares. Isso permitiu que se trabalhasse com apenas 3 componentes sem que o desempenho do classificador fosse prejudicado. Esses 3 componentes foram, então, submetidos a uma rede neural para que se pudesse classificar o comportamento do sinal de DP e, assim, se pudesse ter uma idéia do tipo de defeito que o equipamento apresenta.

## 8.2 Trabalhos Futuros

Essa tese não se propõe a esgotar o tema em questão. Como idéias para trabalhos futuros, pode-se citar:

1. Implementação do sistema em alguma linguagem compilada de modo que o processamento seja executado de forma mais rápida. Com isso teria-se um *software* no modo *stand-alone*, que poderia ser usado sem a presença de qualquer outro programa, como, por exemplo, o Matlab<sup>TM</sup>.
2. Implementação do sistema em DSP. Com isso, seria possível desenvolver um *hardware* como todas as funcionalidades do programa proposto neste trabalho.

3. Testes com sinais reais extraídos de sub-estações em operação. Ter-se-iam, dessa forma, resultados mais próximos da realidade de utilização (que são as medições de sinais de DP no campo).
4. Realizar um trabalho de prospecção de dados com NLPCA do mesmo modo como foi feito anteriormente usando-se PCA. Espera-se que esse método possa modelar com maior fidelidade os padrões relacionados ao processo de DP. Além disso, resta ainda verificar a eficiência quanto à extração seqüencial dos componentes.



# Referências Bibliográficas

- [1] RODRIGUES, F. R. F., *Estudo para implementação de medição de descargas parciais utilizando osciloscópio digital*. M.Sc. dissertation, UFRJ, January 2000.
- [2] *ICM: Digital Partial Discharge Measurement System Operation Manual vs. 2.20*.
- [3] CUENCA, W. M., SEIXAS, J. M., LEVY, A. F., “Análise de Componentes Principais para Identificação de Descargas Parciais em Transformadores de Potência”, *VI Brazilian Symposium on Neural Networks*, , 2000.
- [4] *Norma IEC 270 - Partial Discharge Measurements - Committee Draft*, September 1995.
- [5] DINIZ, P. S. R., SILVA, E. A. B. D., NETTO, S. L., *Digital Signal Processing*. Cambridge University Press, 2002.
- [6] *Acoustic Diagnosis of Gas Insulated Substation (GIS) in Normal Operation - Instrument from Transinor AS - AIA-1 , User's Guide, Version 5.0.1*, May 2001.
- [7] TUTORIAL, H. G., “<http://www.techsoft.de/htbasic/tutgpib.htm>”, Página na Internet, 2004.
- [8] MA, X., ZHOU, C., KEMP, I. J., “Interpretation of Wavelet Analysis and Its Application in Partial Discharge Detection”, *IEEE Transaction on Dielectrics and Eletrical Insulation*, v. 9, n. 3, pp. 446–457, June 2002.
- [9] SATISH, L., NAZNEEN, B., “Wavelet-based denoising of partial discharge signals buried in excessive noise and interference”, *IEEE Transaction on Dielectrics and Eletrical Insulation*, v. 10, n. 2, pp. 354–367, April 2003.

- [10] HAYKIN, S., *Modern Filters*. MacMillan, 1989.
- [11] HAYKIN, S., *Communications Systems*. Wiley and Sons, 2001.
- [12] BISCAINHO, L. W. P., *Restauração Digital de Sinais de Áudio Provenientes de Gravações Musicais Degradadas*. Ph.D. dissertation, COPPE-UFRJ, December 2000.
- [13] SOTOMAYOR, C. A. M., *Realce de Voz Aplicado à Verificação Automática de Locutor*. M.Sc. dissertation, IME, June 2003.
- [14] DINIZ, P. S. R., *Adaptive Filtering*. Kluwer Academic Publishers, 2002.
- [15] SHANMUGAN, K. S., BREIPOHL, A. M., *Random Signals: Detection, Estimation and Data Analysis*. John Wiley and Sons, 1988.
- [16] VAIDYANATHAN, P. P., *Multirate Systems and Filter Banks*. Prentice Hall, 1993.
- [17] DONOHO, D. L., JOHNSTONE, I. M., “Threshold selection for wavelet shrinkage of noisy data”, *Proceedings of the 16th Annual Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 24a–25a, November 1994.
- [18] WERNER, S., DINIZ, P. S. R., “Set-membership affine projection algorithm”, *IEEE Signal Processing Letters*, v. 8, pp. 231–235, August 2001.
- [19] HYVÄRINEN, A., KARHUNEN, J., OJA, E., *Independent Component Analysis*. John Wiley and Sons, 2001.
- [20] COVER, T. M., THOMAS, J. A., *Elements of Information Theory*. John Wiley and Sons, 1991.
- [21] NAGESH, V., GURURAJ, B. I., “Evaluation of Digital Filters for Rejecting Discrete Spectral Interference in On-Site PD Measurements”, *IEEE Transaction on Electrical Insulation*, v. 28, n. 1, pp. 73–85, February 1993.
- [22] MA, X., ZHOU, C., KEMP, I. J., “Investigation into the use of wavelet theory for partial discharge pulse extraction in electrically noisy environments”, *IEE Conference on Dielectric Materials, Measurements and Applications*, pp. 123–126, 2000.

- [23] XIAORONG, W., ZONGJUN, G., YONG, S., *et al.*, “Extraction of Partial Discharge Pulse via Wavelet Shrinkage”, *Proceedings of the 6th International Conference on Properties and Applications of Dielectrics Materials*, pp. 685–688, June 2000.
- [24] WEN, Z., YIGANG, Z., WEIYONG, Y., *et al.*, “Wavelet-based partial discharge denoising using Hidden Markov Model”, *International Conference on Power System Technology. Proceedings. PowerCon.*, pp. 2239–2242, October 2002.
- [25] HU, M., JIANG, X., XIE, H., *et al.*, “A New Technique for extracting Partial Discharge Signals in Online Monitoring With Wavelets Analysis”, *Proceedings of 1998 International Symposium on Electrical Insulating Materials in conjunction with 1998 Asian International Conference on Dielectrics and Electrical Insulation and the 30th Symposium on Electrical Insulating Materials*, pp. 677–680, September 1998.
- [26] SHIM, I., SORAGHAN, J. J., SIEW, W. H., “Detection of PD utilizing digital signal processing methods. Part 3: Open-loop noise reduction”, *IEEE Electrical Insulation Magazine*, pp. 6–13, January 2001.
- [27] SU, Q., “Application of digital signal processing techniques for noise suppression in partial discharge measurements”, *Proceedings of the 4th International Conference on Properties and Applications of Dielectric Materials*, v. 2, pp. 602–605, July 1994.
- [28] KöPF, U., FESER, K., “Rejection of Narrow-band Noise and Repetitive Pulses in On-site PD Measurements”, *IEEE Transaction on Dielectrics and Electrical Insulation*, v. 2, n. 3, pp. 433–447, June 1995.
- [29] KHAN, S. Z., DEHENG, Z., XIANHE, J., *et al.*, “A New Adaptive Technique for On-line Partial Discharge Monitoring”, *IEEE Transactions on Dielectrics and Electrical Insulation*, v. 2, n. 4, pp. 700–707, August 1995.
- [30] ZARGARI, A., BLACKBURN, T. R., “Application of adaptive filters for the estimation of partial discharge signals in noisy environments”, *Proceedings of*

*the 5th International Conference on Properties and Applications of Dielectric Materials*, v. 1, pp. 212–215, May 1997.

- [31] KOSAKO, M., TIAN, Z., OKUBO, H., *et al.*, “Noise Reduction and PD Measurements Using Digital Filters And Signal Processing Technique in HV Substations”, *Proceedings of 1998 International Symposium on Electrical Insulating Materials in conjunction with 1998 Asian International Conference on Dielectrics and Electrical Insulation and the 30th Symposium on Electrical Insulating Materials*, v. 1, pp. 561–564, September 1998.
- [32] SCHINLER, U., BORSI, H., GOCKENBACH, E., “Application of Digital Filters for On-line Partial Discharge Detection on High Voltage Cables”, *Proceedings of the 4th International Conference on Properties and Applications of Dielectrics Materials*, v. 1, pp. 618–621, July 1994.
- [33] DINIZ, F. C. D. C. B., SEIXAS, J. M. D., MARTINS, H. J. A., “Análise por Componentes Principais Não Lineares para a Classificação Neural de Sinais de Descargas Parciais em Transformadores de Potência”, *Congresso Brasileiro de Automática*, , 2004.
- [34] KRAMER, M. A., “Nonlinear principal component analysis using autoassociative neural networks”, *AIChE Journal*, v. 37, n. 2, pp. 233–243, February 1991.
- [35] HAYKIN, S., *Redes Neurais: Princípios e Práticas*. Bookman, 2001.
- [36] HSIEH, W. W., “Nonlinear principal component analysis by neural networks”, *Tellus*, v. 53A, pp. 599–615, 2001.
- [37] RIEDMILLER, M., BRAUN, H., “A direct adaptive method for faster back-propagation learning: The RPROP algorithm”, *Proceedings of The IEEE*, , 1993.
- [38] STONE, M., “Asymptotics for and against cross-validation”, *Biometrika*, v. 64, pp. 29–35, 1977.

- [39] DINIZ, F. C. D. C. B., NETTO, S. L., “A Package Tool for General-Purpose Signal Denoising”, *to appear in IEEE International Conference on Acoustics, Speech and Signal Processing*, , 2005.