

REPRESENTAÇÕES COERENTES DE SINAIS ELÉTRICOS

Lisandro Lovisolo

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. Eduardo Antônio Barros da Silva, Ph.D.

Prof. Paulo Sergio Ramirez Diniz, Ph.D.

Prof. Jacques Szczupak, Ph.D.

Prof. Marco A. Macciola Rodrigues, D.Sc.

Prof. Marcos Roitman, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2001

LOVISOLO, LISANDRO

Representações Coerentes de Sinais
Elétricos [Rio de Janeiro] 2001

XIV,139 pp 29,7 cm (COPPE/UFRJ, M.Sc.,
Engenharia Elétrica, 2001)

Tese - Universidade Federal do Rio de
Janeiro, COPPE

1.Representações de Sinais 2.Represen-
tações Adaptativas 3.Representações Coe-
rentes 4.*Matching Pursuits* 5.Dicionário
de *Gabor* 6.Sinais Elétricos 7.Compres-
são de Sinais Elétricos

I.COPPE/UFRJ II.Título (série)

Agradecimentos

Agradeço a todos que de alguma forma colaboraram comigo nesta jornada.

A todos os companheiros e amigos do LPS pela ajuda e pelo agradável ambiente de trabalho.

A minha mãe, Reina, e meu pai, Hugo, por terem me apoiado e formado em quase tudo que sou.

A meus amigos e irmãos pelos momentos de alegria e lazer ao longo não só deste trabalho mas de toda minha vida.

Ao CEPEL por ter financiado este trabalho, e em especial a Marco Antônio Macciola Rodrigues pelo acompanhamento do meu trabalho e disponibilidade.

Aos meus orientadores Eduardo e Diniz por tudo. Gostaria de agradecer especialmente ao Eduardo por ter me apoiado e acompanhado durante esta jornada, e anteriormente a esta, tendo tido sempre confiança e respeito por mim.

À Gabriela, por ter tido a paciência e compreensão necessária comigo no processo de finalização desta etapa. Por mais do que isso, por me fazer querer ser uma pessoa melhor, e por seu amor e carinho.

Muito obrigado.

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

Representações Coerentes de Sinais Elétricos

Lisandro Lovisolo

Março/2001

Orientadores: Eduardo Antônio Barros da Silva

Paulo Sérgio Ramirez Diniz

Programa: Engenharia Elétrica

Nesta tese se estudará a representação coerente de sinais elétricos usando decomposições adaptativas. As decomposições adaptativas de sinais fornecem ferramentas para identificar fenômenos presentes nos mesmos. Desta forma pode-se identificar as estruturas de um sinal e representá-lo através destas estruturas. O objetivo então não é só obter uma representação do sinal, mas obter uma representação que seja condizente com os fenômenos presentes no mesmo. Utilizamos o algoritmo de *Matching Pursuits*, uma forma de realizar uma representação adaptativa, com o dicionário de *Gabor* para representar sinais elétricos. Vê-se porém que este dicionário não é apropriado pois seus elementos não representam os fenômenos presentes em sinais elétricos. Utilizaremos então um dicionário formado por exponenciais moduladas, que possuem maior relação com fenômenos elétricos. Com estes elementos no *Matching Pursuits* conseguimos identificar grande parte dos fenômenos presentes em sinais elétricos. Por fim estudamos como a reconstrução do sinal a partir dos fenômenos identificados se comporta quando sujeita à quantização dos elementos.

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

Coherent Representations of Electric Signals

Lisandro Lovisolo

March/2001

Advisors: Eduardo Antônio Barros da Silva

Paulo Sérgio Ramirez Diniz

Department: Electrical Engineering

This thesis studies the coherent representations of electric signals using adaptive decompositions. Adaptive signal decompositions provide a tool to identify structures that compose a signal, and an approach to represent a signal from those components. Thus the goal is not just to obtain a representation of the signal, but to obtain a representation that is related to the actual phenomena of the signal. In this thesis we use the *Matching Pursuits* algorithm, a way to accomplish adaptive representations, with the *Gabor* codebook to represent electric signals. We observe that this codebook is not well suited because its elements do not represent the phenomena that compose electric signals. Then we use a codebook composed of modulated exponentials that is closely related to electric phenomena. With these elements in the *Matching Pursuits* algorithm we can achieve the identification of most phenomena present in electric signals. At last we study how the quantization of the identified phenomena influences the reconstruction process.

Sumário

1	Introdução	1
2	Representações de Sinais	4
2.1	Representações de Sinais	5
2.1.1	Transformada de <i>Fourier</i> Discreta	6
2.1.2	Formulação Geral	7
2.1.3	Compacidade da Representação	8
2.2	Bases	9
2.2.1	Bases Ortogonais	10
2.2.2	Bases Biortogonais	10
2.3	<i>Frames</i>	11
2.3.1	Redundância	11
2.3.2	<i>Frames</i> Apertados	12
2.3.3	Operador do <i>Frame</i>	12
2.4	Representações Redundantes	13
2.4.1	Cálculo dos Coeficientes	14
3	Representações Adaptativas	16
3.1	Procurando a Melhor Base	18
3.1.1	Adaptabilidade	18
3.1.2	Diferentes Abordagens	19
3.1.2.1	Método dos <i>Frames</i>	19
3.1.2.2	Melhor Base Ortonormal	19
3.1.2.3	Otimização Massiva	20
3.1.2.4	<i>Basis Pursuit</i>	20

3.1.2.5	<i>Matching Pursuits</i> - Aproximação por Passos	20
3.2	<i>Matching Pursuits</i> em Espaços Finitos	21
3.2.1	Redundância no <i>Matching Pursuits</i>	23
3.2.2	Implementação Rápida	24
3.3	Representações Coerentes	25
3.3.1	Representando Sinais Coerentemente	28
4	Dicionário de <i>Gabor</i>	31
4.1	Representações Tempo-Freqüência	31
4.1.1	Análise Tempo-Freqüência	32
4.1.2	Plano Tempo-Freqüência	34
4.1.3	Representações Tempo-Freqüência Quadráticas	35
4.1.3.1	Princípio da Superposição Quadrática	36
4.1.4	A Distribuição de Wigner-Ville	36
4.2	Átomo de <i>Gabor</i>	37
4.3	Gerando o Dicionário de <i>Gabor</i>	38
4.3.1	Átomos de <i>Gabor</i> Discretos	39
4.3.2	<i>Frame</i> do Dicionário Discreto de <i>Gabor</i>	40
4.3.3	Átomos do Dicionário Discreto de <i>Gabor</i>	41
4.3.4	Átomos Reais	42
4.4	<i>Matching Pursuits</i> com o Dicionário de <i>Gabor</i>	43
4.4.1	Achando a Fase do Átomo	50
4.5	Representação Tempo-Freqüência com o <i>Matching Pursuits</i> e o Dicionário de <i>Gabor</i>	54
4.6	Comportamento Passo a Passo	68
5	Representações Coerentes Com o Dicionário de <i>Gabor</i>	70
5.1	Estudo da Decomposição	70
5.2	Definindo uma medida de Coerência	72
5.3	Diferentes Dimensões	75
6	Modificações no Dicionário de <i>Gabor</i>	79
6.1	Um Dicionário de Parâmetros Contínuos	79
6.2	Modificações no Suporte Temporal	82

6.3	Decompondo Sinais com o Dicionário de <i>Gabor</i> com Parâmetros Con- tínuos e Modificação do Suporte Temporal	85
7	Compressão de Sinais Elétricos Através de Representações Coeren- tes	91
7.1	Dicionário de Átomos Exponenciais	91
7.1.1	Encontrando o Átomo Exponencial	92
7.2	Decompondo Sinais Com Átomos Exponencias	97
7.2.1	Representações Coerentes Com o Dicionário de Exponenciais .	103
7.2.2	Imunidade ao Ruído	105
7.3	Quantização das Estruturas	109
7.3.1	Resultados	113
7.4	Aplicação a Sinais Reais	116
8	Conclusões	119
8.1	Representações Redundantes e Adaptativas	120
8.2	Dicionário de Exponenciais	120
8.3	Quanto ao Critério de Coerência	121
8.4	Quanto às Estruturas	122
8.5	Compressão de Sinais Elétricos	122
8.6	Nossas Contribuições	123
A	Sinais Estudados	125
A.1	Sinais Sintéticos	125
A.2	Sinais Reais	130
A.3	Ruídos Gaussianos	130

Lista de Figuras

2.1	A idéia de representação redundante.	4
2.2	A idéia de representação compacta.	9
3.1	Representação Gráfica de um Passo do <i>Matching Pursuit</i>	22
4.1	Representação Tempo-Frequência de uma átomo $\gamma_{(u,\xi)}$	35
4.2	Resultados do <i>Matching Pursuit</i> com 8 fases possíveis, para o sinal x001.	45
4.3	Resultados do <i>Matching Pursuit</i> com 8 fases possíveis, para o sinal x001.	46
4.4	Resíduo normalizado no passo para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> com diferentes fases para o sinal x001.	48
4.5	RMSE no passo para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> com diferentes fases para o sinal x001.	48
4.6	Energia do ruído pela energia do sinal no passo para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> com diferentes fases para o sinal x001.	49
4.7	Taxa de aproximação no paso para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> com diferentes fases para o sinal x001.	49
4.8	Resultados do <i>Matching Pursuits</i> com fase ótima, para o sinal x001.	53
4.9	Resultados do <i>Matching Pursuits</i> com fase ótima, para o sinal x002.	55
4.10	Resultados do <i>Matching Pursuits</i> com fase ótima, para o sinal x003.	56
4.11	Resultados do <i>Matching Pursuits</i> com fase ótima, para o sinal x004.	57
4.12	Resultados do <i>Matching Pursuits</i> com fase ótima, para o sinal x005.	58
4.13	Resultados do <i>Matching Pursuits</i> com fase ótima, para o sinal x006.	59
4.14	Resultados do <i>Matching Pursuits</i> com fase ótima, para o sinal x007.	60

4.15	Resultados do <i>Matching Pursuits</i> com fase ótima, para o sinal x008.	61
4.16	RMSE no passo corrente para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> e fase ótima para diferentes sinais.	62
4.17	RNS no passo corrente para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> e fase ótima para diferentes sinais.	62
4.18	Taxa de aproximação no passo corrente para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> e fase ótima para diferentes sinais.	63
4.19	Resíduo normalizado após o passo para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> e fase ótima para diferentes sinais.	63
4.20	Taxa de aproximação média até o passo para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> e fase ótima para diferentes sinais.	64
4.21	Sinal de 512 amostras formado a partir da adição <i>chirps</i> , senóides limitadas no tempo e formas de onda com diferentes localizações tempo-freqüência [1, 2].	67
4.22	Distribuição de energia tempo-freüência, $Ef(t, \omega)$, seção 4.5, do sinal mostrado na figura 4.21. O eixo horizontal é o tempo. O vertical a freqüência. As freqüências mais altas estão no topo. Quanto maior $Ef(t, \omega)$ mais escuro o ponto [1, 2].	67
4.23	Comportamento da decomposição com diferentes números de passos para o sinal “x007”, 1 a 4 passos. O sinal original é apresentado em traço cheio e o reconstruído em traço pontilhado.	69
5.1	Resíduo normalizado após o passo para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> para diferentes sinais com 128 amostras.	72
5.2	Taxa de aproximação no passo para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> para diferentes sinais com 128 amostras.	73
5.3	Taxa de aproximação média até o passo para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> para diferentes sinais com 128 amostras.	73
5.4	RMSE no passo para o <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> para diferentes sinais com 128 amostras.	74
5.5	Taxa de aproximação média móvel, $L = 16$, no <i>Matching Pursuits</i> com o dicionário de <i>Gabor</i> para diferentes sinais com 128 amostras.	75

5.6	Taxa de aproximação no passo para sinais de Ruído Gaussiano de comprimentos diferentes	77
5.7	Taxa de aproximação média móvel no passo para sinais de Ruído Gaussiano de comprimentos diferentes	77
6.1	Procura dos parâmetros contínuos para geração dos átomos no algoritmo de <i>Matching Pursuits</i>	83
6.2	Comportamento da decomposição com diferentes números de passos para o sinal “x007”, com 4, 6 e 8 passos, para as modificações introduzidas no dicionário. O sinal original está em linha cheia e o reconstruído em linha pontilhada.	87
6.3	Comportamento da decomposição do sinal “x001”, no primeiro passo, para as modificações introduzidas no dicionário. O sinal original está em linha cheia e o reconstruído em linha pontilhada.	88
6.4	Comportamento da decomposição do sinal “x007”, no primeiro passo, para as modificações introduzidas no dicionário. O sinal original está em linha cheia e o reconstruído em linha pontilhada.	88
6.5	Comportamento da decomposição com diferentes números de passos para o sinal “x006”, com 1, 2, 3 e 4 passos, para as modificações introduzidas no dicionário. O sinal original está em linha cheia e o reconstruído em linha pontilhada.	90
7.1	Comportamento da decomposição do sinal “x002”, no primeiro passo, para o dicionário com átomos exponenciais.	96
7.2	Comportamento da decomposição do sinal “x008”, no primeiro passo, para o dicionário com átomos exponenciais.	98
7.3	Comportamento da reconstrução do sinal “x001”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais .	99
7.4	Comportamento da reconstrução do sinal “x002”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais .	99
7.5	Comportamento da reconstrução do sinal “x003”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais .	101

7.6	Comportamento da reconstrução do sinal “x004”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais .	101
7.7	Comportamento da reconstrução do sinal “x005”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais .	101
7.8	Comportamento da reconstrução do sinal “x006”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais .	102
7.9	Comportamento da reconstrução do sinal “x007”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais .	102
7.10	Comportamento da reconstrução do sinal “x008”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais .	102
7.11	Comportamento da identificação do sinal gerado com uma única exponencial.	105
7.12	Comportamento da identificação do sinal gerado com uma única exponencial.	106
7.13	Comportamento da identificação do sinal gerado com uma única exponencial.	107
7.14	Comportamento da identificação do sinal somado ao ruído com razão entre a energia do sinal e a do ruído de 375,81.	107
7.15	Comportamento da identificação do sinal somado ao ruído com razão entre a energia do sinal e a do ruído de 93,92.	108
7.16	Comportamento da identificação do sinal somado ao ruído com razão entre a energia do sinal e a do ruído de 23,48.	108
7.17	Comportamento da identificação do sinal somado ao ruído com razão entre a energia do sinal e a do ruído de 13,36.	108
7.18	Cabeçalho da compressão de um sinal informando o seu comprimento, e os passos de quantização de: produto interno, λ e ϕ e o número de bits utilizados na quantização de cada um deles.	111
7.19	Cabeçalho da compressão de um sinal.	112
7.20	Cabeçalho completo da compressão de um sinal com uma estrutura. .	113
7.21	SNR \times Taxa de compressão para a quantização do sinal x005.	115
7.22	Melhores quantizadores para as estruturas do sinal x005.	116
7.23	Sinal R001 reconstruído.	117

7.24 Sinal R002 reconstruído.	118
7.25 Sinal R003 reconstruído.	118
7.26 Sinal R004 reconstruído.	118
A.1 Sinal x001	127
A.2 Sinal x002	127
A.3 Sinal x003	127
A.4 Sinal x004	128
A.5 Sinal x005	128
A.6 Sinal x006	128
A.7 Sinal x007	129
A.8 Sinal x008	129
A.9 Sinal R001.	131
A.10 Sinal R002.	131
A.11 Sinal R003.	131
A.12 Sinal R004.	132
A.13 Ruído Gaussiano de 128 Amostras	132
A.14 Ruído Gaussiano de 256 Amostras	133
A.15 Ruído Gaussiano de 512 Amostras	133
A.16 Ruído Gaussiano de 1024 Amostras	133

Lista de Tabelas

4.1	Avaliação dos dicionários com diferentes quantidades de fases.	44
5.1	Redundância do Dicionário de <i>Gabor</i> em diferentes dimensões.	76
5.2	Taxa de aproximação média para sinais de Ruído com diferentes comprimentos	78
6.1	Comparação dos produtos internos sem as modificações no dicionário de <i>Gabor</i> e com elas para os sinais de testem no primeiro passo do <i>Matching Pursuits</i>	85
7.1	Parâmetros extraídos na decomposição dos sinais de teste.	100
7.2	Taxa de aproximação média para sinais de Ruído com diferentes comprimentos, no dicionário de exponenciais.	103
7.3	SNR e número de átomos identificados nos sinais sintéticos.	104
7.4	Decomposição do sinal “x002” com adição de ruído.	107
7.5	SNR e Taxa de compressão obtidas com os sinais x001 e x002.	114
7.6	SNR e Taxa de compressão obtidas para os sinais reais, quantidade de bits utilizados por parâmetro e figura do sinal reconstruído.	117
A.1	Parâmetros utilizados na geração do sinais de testes através do modelo da equação 7.1.	126

Capítulo 1

Introdução

O tema desta tese, Compressão de Sinais Elétricos, tem sido estudado, obtendo-se alguns resultados interessantes [3]. Entretanto ainda não há um padrão estabelecido para a compressão de sinais de oscilografia. Além disso a abordagem utilizada nesta tese: Representações Coerentes de Sinais Elétricos; é o que a torna um pouco diferente.

O que desejamos fazer? A partir de um sinal elétrico extrair suas características fenomenológicas, de forma que ao comprimí-lo possamos a partir da compressão, e não do sinal recuperado após a descompressão, extrair os fenômenos presentes no sinal. É um critério diferente na compressão, pois não estaremos só comprimindo o sinal mas também entendendo-o. Desta maneira queremos interpretar o sinal reduzindo-o a um conjunto de componentes coerentes, ou seja com o mesmo significado, de forma que não só a reconstrução do sinal comprimido possa ser realizada com erro pequeno, mas também extraindo informação do sinal. Desta forma as componentes podem ser utilizadas para analisar o comportamento do sinal, identificando através das mesmas o funcionamento do sistema.

Nesta tese trabalhamos com sinais provenientes de sistemas de transmissão de energia elétrica. Neste âmbito, comprimir o sinal elétrico possui certa relevância, porém com os recentes avanços em sistemas de armazenamento de massa esta relevância tem diminuído, entretanto resta o problema da transmissão destes sinais onde a compressão ainda se faz necessária além do aumento da janela e da taxa de amostragem com o que amostramos sinais cada vez maiores. Por outro lado, encontrar os fenômenos presentes nos sinais gera um interesse especial e desafiador. Não se

trata simplesmente de comprimir o sinal respeitando critérios objetivos e subjetivos, mas além disso, projetar um sistema que além de comprimir com altas taxas seja capaz de, a partir do sinal comprimido, compreender e extrair características dos fenômenos presentes no sinal.

Além do objetivo de obter uma decomposição do sinal com componentes coerentes, desejamos que a extração destas componentes possa ser realizada de forma automática pelo algoritmo de decomposição. É neste compromisso que surgem as representações adaptativas de sinais. Usaremos o conceito de uma representação adaptativa no projeto de um algoritmo capaz de representar coerentemente um sinal elétrico.

Este texto está organizado da seguinte forma:

- No Capítulo 2, trataremos da representação de sinais, chegando por fim às representações adaptativas, que são as representações das quais lançaremos mão na codificação dos sinais elétricos para extrair suas características.
- No Capítulo 3, veremos uma representação adaptativa específica, o algoritmo de *Matching Pursuits*. Este será o nosso ponto de partida para a representação coerente de sinais elétricos.
- No Capítulo 4, estudaremos especificamente o dicionário de *Gabor*, formado de átomos tempo-freqüência, e sua utilização no algoritmo de *Matching Pursuits*, para a decomposição de sinais.
- No Capítulo 5, abordaremos o problema de realizar representações coerentes com o dicionário de *Gabor*.
- No Capítulo 6 trataremos de algumas modificações no dicionário de *Gabor* e as implicações destas modificações na representação coerente de sinais.
- Já no Capítulo 7, trabalharemos especificamente com a representação coerente de sinais elétricos. Com isto faremos modificações ao nosso método de forma a aumentar sua eficiência e coerência na representação de fenômenos presentes em linhas de transmissão. Após a identificação das estruturas coerentes com o sinal veremos os efeitos acarretados ao quantizar os parâmetros que definem as estruturas.

- No Capítulo 8 apresentaremos as conclusões deste trabalho, e sugestões para sua continuidade.

Ao fim deste trabalho estaremos de posse de um algoritmo que não só nos permitirá comprimir um sinal elétrico com altas taxas mas também classificar, a partir do sinal comprimido, as estruturas presentes no mesmo. Com isto teremos além de uma decomposição do sinal, informação sobre as características presentes no mesmo. Teremos então além de uma decomposição do sinal, informação sobre as características ou estruturas do mesmo.

Capítulo 2

Representações de Sinais

Neste capítulo trataremos da base teórica necessária ao entendimento da decomposição redundante de um sinal, ou representação redundante, bem como das representações adaptativas.

Iniciaremos vendo como podemos representar um sinal. A seguir veremos o conceito de bases, na seção 2.2, *frames*, na seção 2.3, e em seguida o conceito de representação redundante. Em todas as formulações com as quais aqui lidamos consideramos espaços de *Hilbert* [2, 4], \mathbb{R}^N ou \mathbb{C}^N , que devem fornecer uma métrica de distância entre os pontos do espaço e serem completos [5, 2, 4]. Neste processo desejamos uma representação de um sinal de dimensão N num conjunto de M coeficientes onde $M \geq N$. Este tipo de representação pode ser vista na figura 2.1. É interessante notar que neste caso pode-se aumentar o número de coeficientes utilizados para representar o sinal em relação ao número de coeficientes do sinal original.

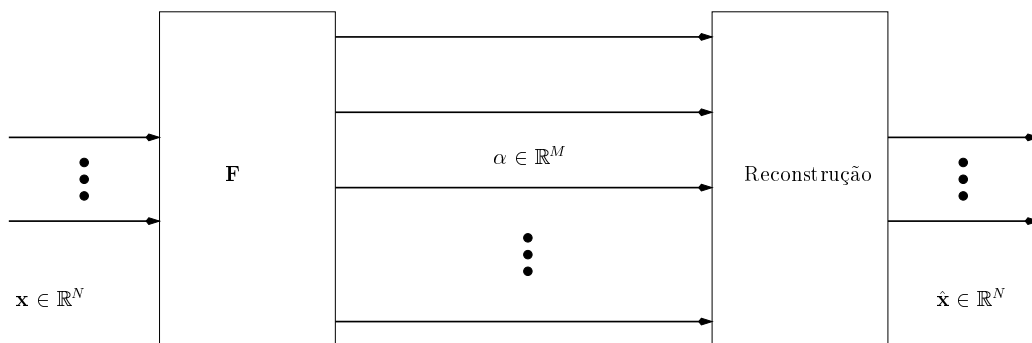


Figura 2.1: A idéia de representação redundante.

Vejamos agora como decompor ou representar sinais. Começemos com bases de exponenciais complexas que são o caso da transformada ou expansão em séries de *Fourier*. Após isto progressivamente iremos expandindo os conceitos até chegarmos aos *frames*, onde passaremos a realizar a nossa representação.

2.1 Representações de Sinais

Um sinal periódico, $x_p(t)$, pode ser representado através de um conjunto de sinais pré-definidos. Este resultado pode ser visto na expansão em séries de exponenciais ou Séries de *Fourier*. Para $x_p(t)$ definido em $(-\frac{T}{2}, \frac{T}{2})$ e fora dele por $x_p(t + T) = x_p(t)$, ou seja $x_p(t)$ é periódico com período T , podemos decompor o sinal $x_p(t)$ através de uma Série Complexa de *Fourier* [6, 5]:

$$x_p(t) = \sum_{n=-\infty}^{\infty} c_n e^{\frac{2\pi j n t}{T}}, \quad (2.1)$$

$$c_n = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} x_p(t) e^{-\frac{2\pi j n t}{T}} dt. \quad (2.2)$$

Um sinal genérico qualquer, $x(t)$, mesmo que não seja periódico, também pode ser representado ao longo de um intervalo de tempo T por uma série de *Fourier*. Neste caso estamos considerando que fora do intervalo de tempo T o sinal se repete, torna-se periódico. É importante notar que esta representação só será válida para o sinal durante o período T . Então, um sinal $x(t)$ definido em (T_1, T_2) , no intervalo $T_1 \leq t \leq T_2$, com período $(T = T_2 - T_1)$, pode ser representado através de [6]:

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{\frac{2\pi j n t}{T}}, \quad (2.3)$$

$$c_n = \frac{1}{T} \int_{T_1}^{T_2} x(t) e^{-\frac{2\pi j n t}{T}} dt. \quad (2.4)$$

Desta forma podemos decompor um sinal num intervalo (T_1, T_2) qualquer, mas a decomposição só será válida neste intervalo. As equações (2.2) e (2.4) decompõem o sinal nos coeficientes c_n , isto é analisam o sinal. Chamamos o processo de cálculo dos coeficientes c_n de **análise**. A reconstrução do sinal a partir dos coeficientes c_n através das equações (2.1) e (2.3) é chamada de **síntese**.

2.1.1 Transformada de *Fourier* Discreta

Passemos do sinal contínuo $x(t)$ explorado anteriormente, para um sinal discreto $x(n)$ formado a partir da amostragem do sinal contínuo $x(t)$. Os valores de $x(n)$ são dados através da amostragem $x(n) = x(n\Delta T)$, sendo ΔT o intervalo de amostragem do sinal $x(t)$. O sinal $x(n)$, que podemos tratar como um vetor de comprimento N , pode ser representado de forma semelhante por um conjunto de coeficientes $X(k)$ dado por[6]:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j(\frac{2\pi}{N})nk}, \quad (2.5)$$

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j(\frac{2\pi}{N})nk}. \quad (2.6)$$

Esta é a Transformada de *Fourier* Discreta do sinal $x(n)$. A equação (2.6) pode ser reescrita de outra forma:

$$X(k) = \langle \mathbf{x}, \mathbf{e}_k \rangle, \quad (2.7)$$

onde \mathbf{x} é o vetor coluna composto pelas amostras $x(n)$, \mathbf{e}_k é vetor coluna para k fixo e $n \in [0 \dots N - 1]$, computado através de $e_{n,k} = e^{j(\frac{2\pi}{N})nk}$, e dado por:

$$\mathbf{e}_k = \begin{bmatrix} e_{0,k} \\ e_{1,k} \\ e_{2,k} \\ \vdots \\ e_{N-2,k} \\ e_{N-1,k} \end{bmatrix}, \quad (2.8)$$

E $\langle \mathbf{u}, \mathbf{v} \rangle$ é o produto interno entre os vetores \mathbf{u} e \mathbf{v} quaisquer. Matricialmente a equação (2.7) fica:

$$\begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_{N-1} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_0^T \\ \mathbf{e}_1^T \\ \dots \\ \mathbf{e}_{N-1}^T \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix}. \quad (2.9)$$

Onde A^T é a matriz A transposta. Reversamente temos o sinal $x(n)$ gerado por uma combinação linear dos vetores \mathbf{f}_k que são gerados a partir de $f_{n,k} = e^{-j(\frac{2\pi}{N})nk}$ com k

fixo e $n \in [0, N - 1]$, que pode ser visto a seguir:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) f_{n,k}^*, \quad (2.10)$$

que pode ser expressa matricialmente através de:

$$\mathbf{x} = \mathbf{F}^{T*} \mathbf{X}, \quad (2.11)$$

onde \mathbf{x} é um vetor coluna de comprimento N , o sinal, \mathbf{X} também é um vetor coluna composto dos elementos $[X(0), X(1), \dots, X(N - 1)]$ e \mathbf{F} a matriz composta dos vetores \mathbf{f}_k dados por:

$$\mathbf{f}_k = [f_{0,k}, f_{1,k}, \dots, f_{N-1,k}]^{T*}. \quad (2.12)$$

Para que uma representação como esta seja válida para analisar e posteriormente sintetizar um sinal, o intervalo de amostragem ΔT deve ser tal que satisfaça a taxa de *Nyquist* [6]: a frequência de amostragem deve ser pelo menos duas vezes maior que a frequência máxima presente no sinal.

2.1.2 Formulação Geral

Genericamente, uma função contínua $x(t) \in \mathbb{F}$ e real pode ser representada por um conjunto finito ou infinito de funções $f_k(t)$, desde que as funções $f_k(t)$ gerem o espaço \mathbb{F} , através de:

$$x(t) = \sum_k \alpha_k f_k(t) \quad (2.13)$$

Com $k \in \mathbb{Z}$, α_k e $t \in \mathbb{R}$. O espaço \mathbb{F} definido como $\mathbb{F} = \text{Span}_k f_k$, é espaço com todos os elementos na forma:

$$f(t) \in \mathbb{F} \Leftrightarrow f(t) = \sum_k \alpha_k f_k(t) \quad (2.14)$$

Assim como um sinal pode ser representado, um vetor $\mathbf{x} \in \mathbb{R}^N$ também pode ser representado por uma combinação linear de vetores \mathbf{g}_i que gerem o espaço \mathbb{R}^N . Ou seja:

$$\mathbf{x} = \sum_i \alpha_i \mathbf{g}_i, \quad (2.15)$$

$$\mathbf{g} \in \mathbb{R}^N \Leftrightarrow \mathbf{g} = \sum_i \alpha_i \mathbf{g}_i. \quad (2.16)$$

As equações (2.13) e (2.15) nos dizem como sintetizar, por isso as chamaremos de equações de síntese, um sinal $x(t)$ ou um vetor \mathbf{x} a partir de coeficientes e funções ou vetores que gerem o espaço ao qual o sinal $x(t)$ ou o vetor \mathbf{x} pertençam. Para tal, as funções $f_k(t)$ ou vetores \mathbf{g}_i deverão satisfazer as equações (2.14) ou (2.16) respectivamente. Mas, em nenhum momento, abordamos o problema do cálculo coeficientes da expansão, ou representação, do sinal ou vetor no conjunto f_k ou \mathbf{g}_i .

Como no caso da Transformada de *Fourier*, abordado na subseção 2.1.1, os coeficientes das expansões (2.13) e (2.15) podem ser calculados através do produto interno do sinal $x(t)$ com as diversas funções ou vetores que compõem o conjunto f_k ou \mathbf{g}_k . Obteremos então os coeficientes através de uma simples projeção do sinal $x(t)$ ou do vetor \mathbf{x} sobre os elementos dos conjuntos f_k ou \mathbf{g}_i . Logo α_k e α_i serão dados por:

$$\alpha_k = \int_t x(t) f_k(t) dt = \langle x(t) f_k(t) \rangle, \quad (2.17)$$

no caso do sinal $x(t)$ contínuo, e:

$$\alpha_k = \langle \mathbf{x}, \mathbf{g}_k \rangle, \quad (2.18)$$

no caso do sinal $x(n)$ discreto. A equação (2.18) é válida tanto para espaços vetoriais quanto para espaços de funções. A diferença encontra-se na definição do produto interno que para o caso contínuo é dado por:

$$\langle h(t), s(t) \rangle = \int_{-\infty}^{\infty} h(t) s(t) dt, \quad (2.19)$$

onde $h(t)$ e $s(t)$ são duas funções quaisquer; e para o caso discreto por:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{n=0}^{N-1} u_n v_n, \quad (2.20)$$

para dois vetores \mathbf{u} e \mathbf{v} quaisquer, onde N é a dimensão do vetor.

2.1.3 Compacidade da Representação

Nas equações (2.13) e (2.15), os limites do somatório propositalmente não foram especificados. Por exemplo, no caso de um vetor \mathbf{x} , equação (2.15), o número de vetores \mathbf{g}_k usados pode ser inclusive maior que a dimensão do vetor. Isto provoca uma redundância da representação. Isto nos remete ao conceito contrário de redundância: de compacidade da representação [7].

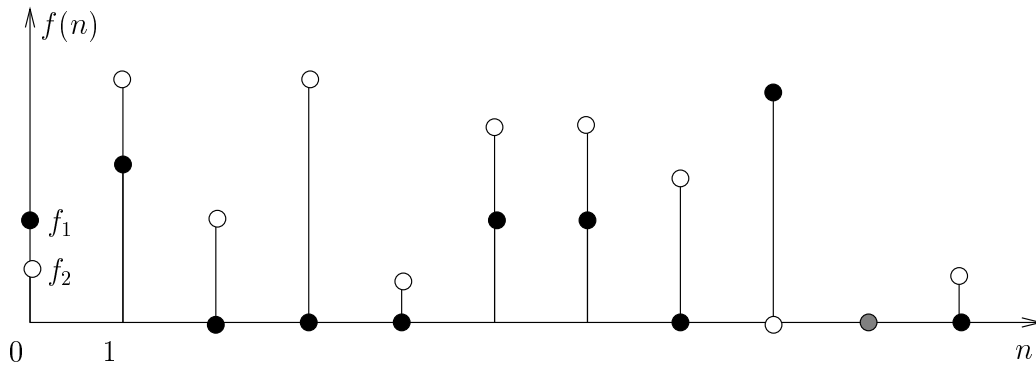


Figura 2.2: A idéia de representação compacta.

Na figura 2.2 podemos ver duas representações de um mesmo sinal em dois conjuntos de funções base $f(n)$. Para o conjunto f_1 as representações são dadas pelos círculos pretos; e para o conjunto f_2 pelo brancos; quando ambos coincidem mostramos este fato com um círculo cinza. Percebe-se que a representação f_1 é mais compacta que a f_2 pois possui um menor número de coeficientes não nulos (5 para os coeficientes da representação sobre o conjunto f_1 e 9 para a representação no conjunto f_2). Ou seja, a energia está compactada num menor número de coeficientes quando utilizamos o conjunto f_1 . Pode-se então associar a idéia de compactidade de uma representação à quantidade de elementos necessários para representar o sinal.

Voltemo-nos ao conjunto de funções f_k e \mathbf{g}_k . Posteriormente trataremos do problema de como obter representações compactas.

2.2 Bases

Um sinal pode ser decomposto através da equação (2.11), repetida a seguir, na forma matricial:

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} f_{0,0} & f_{0,1} & \cdots & f_{0,N-1} \\ f_{1,0} & f_{1,1} & \cdots & f_{1,N-1} \\ \vdots & \ddots & \ddots & \vdots \\ f_{N-1,0} & f_{N-1,1} & \cdots & f_{N-1,N-1} \end{bmatrix} \times \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{N-1} \end{bmatrix}, \quad (2.21)$$

ou compactamente:

$$\mathbf{x} = \mathbf{F}\alpha. \quad (2.22)$$

onde \mathbf{F} será composta dos vetores \mathbf{f}_k , \mathbf{F} é $N \times N$, \mathbf{x} é $N \times 1$ e α é $N \times 1$.

Diremos que \mathbf{F} é uma base do espaço \mathbb{R}^N se o vetor α for único para um vetor particular \mathbf{x} . Analogamente diremos que \mathbf{f}_k ou \mathbf{F} é uma base de um espaço \mathbb{F} , se o conjunto de coeficientes α_k em (2.23) for único para uma função particular $x(t) \in \mathbb{F}$:

$$x(t) \in \mathbb{F} \Leftrightarrow x(t) = \sum_k a_k f_k(t). \quad (2.23)$$

2.2.1 Bases Ortogonais

Aqui nos restringiremos a uma discussão de espaços vetoriais e não de espaços de funções, mas os resultados podem ser transferidos para este último, desde que este seja um espaço de *Hilbert*. Uma base será ortogonal caso o produto interno entre quaisquer dois vetores \mathbf{g}_i que compõem a base seja zero, isto é, $\langle \mathbf{g}_l, \mathbf{g}_k \rangle = 0 \quad \forall k \neq l$. Em qualquer espaço euclidiano, uma base com esta característica seria formada por vetores com ângulos de 90° entre eles.

Diremos que uma base é ortonormal caso, além de ortogonal, o produto interno entre os vetores satisfaça: $\langle \mathbf{g}_l, \mathbf{g}_k \rangle = \delta(l - k) \quad \forall (k, l)$. No caso de $k \neq l$ teremos produto interno igual a zero, incorrendo no caso do parágrafo acima. Porém, caso $k = l$ implica norma unitária para os vetores que formam a base: $\|\mathbf{g}_k\| = 1 \quad \forall k$. A facilidade que um espaço ortonormal nos apresenta é que podemos expressar um vetor $\mathbf{x} \in \mathbb{R}^N$ na base Γ , composta dos vetores \mathbf{g}_i , através de:

$$\mathbf{x} = \sum_i \langle \mathbf{x}, \mathbf{g}_i \rangle \mathbf{g}_i. \quad (2.24)$$

Neste caso os coeficientes, não sendo esta uma característica específica de bases ortogonais, α_i serão dados por $\langle \mathbf{x}, \mathbf{g}_i \rangle$ sem a necessidade de atualização do vetor \mathbf{x} a cada passo, como veremos na seção 2.4.1 para casos mais genéricos.

Teremos que o vetor \mathbf{x} é uma combinação linear dos vetores \mathbf{g}_i . E portanto, os vetores \mathbf{g}_i para compor uma base Γ do espaço \mathbb{R}^N deverão ser linearmente independentes.

2.2.2 Bases Biortogonais

Apesar da facilidade de manipulação de expansões de sinais em bases ortogonais, nem sempre elas se adaptam ou são razoáveis para o problema que desejamos investigar. Em alguns casos quando a base Γ não é ortogonal, lançamos mão da

base dual $\tilde{\Gamma}$, cujos elementos não são ortogonais entre si, mas em relação ao elemento correspondente da base dual Γ . Estas são chamadas de bases biortogonais. Assim teremos:

$$\langle \mathbf{g}_i, \tilde{\mathbf{g}}_k \rangle = \delta(k - l). \quad (2.25)$$

Com isto, poderemos analisar o sinal através dos coeficientes α_i na base Γ e sintetizar através da base dual $\tilde{\Gamma}$:

$$\mathbf{x} = \sum_i \langle \mathbf{x}, \mathbf{g}_i \rangle \tilde{\mathbf{g}}_i, \text{ ou} \quad (2.26)$$

$$\mathbf{x} = \tilde{\Gamma}^T (\Gamma \mathbf{x}). \quad (2.27)$$

2.3 Frames

Nas representações ou expansões de sinais discutidas até aqui, a partir de um vetor \mathbf{x} de N elementos, obtinhamos um vetor α também de N elementos que é uma representação do vetor \mathbf{x} em uma outra base. Em alguns casos - e isto é cada vez mais comum em sistemas de processamento de sinais, principalmente quando queremos trabalhar com o reconhecimento de padrões - temos interesse que o vetor α da figura 2.1 possua dimensão diferente e principalmente maior que N . Neste contexto surgem os *frames* [8, 9, 10, 11, 2].

Definição 2.1 Consideremos o espaço de Hilbert \mathbb{H} , seja $\Phi = \{\phi_m\}$ com $m \in \{1, \dots, M\}$ e $\phi_m \in \mathbb{H}$, Φ é chamado de *frame* se existem $A > 0$ e $B < \infty$ tais que para todo $\mathbf{x} \in \mathbb{H}$ teremos:

$$A \|\mathbf{x}\|^2 \leq \sum_m |\langle \mathbf{x}, \phi_m \rangle|^2 \leq B \|\mathbf{x}\|^2. \quad (2.28)$$

Chamamos M de cardinalidade, o número de elementos sobre o qual projetamos \mathbf{x} . A e B são chamados de *frame bounds*, limites do *frame*. Uma combinação linear dos elementos ϕ_m é capaz de representar qualquer ponto de \mathbb{H} . Isto implica que M seja sempre maior ou igual à dimensão do espaço \mathbb{H} , ou seja $M \geq N$.

2.3.1 Redundância

Chamaremos de r a redundância do *frame* que é dada por:

$$r = \frac{M}{N}. \quad (2.29)$$

Observe que esta é uma medida bastante intuitiva de redundância da representação do espaço \mathbb{H} no *frame* Φ , que necessariamente deverá gerar o espaço \mathbb{H} , já que nos indica quantos vetores a mais utilizamos por dimensão de \mathbb{H} , cuja dimensão é N .

2.3.2 *Frames Apertados*

Um *frame* é apertado¹ se podemos fazer com que os contornos da equação (2.28) sejam iguais, isto é, $A = B$. Isto implica que para um vetor \mathbf{x} teremos a equação (2.28) reduzida a [11, 9, 10]:

$$A \|\mathbf{x}\|^2 = \sum_m |\langle \mathbf{x}, \phi_m \rangle|^2, \quad (2.30)$$

pode-se mostrar que:

$$\mathbf{x} = A^{-1} \sum_m \langle \mathbf{x}, \phi_m \rangle \phi_m. \quad (2.31)$$

Esta é a mesma expressão da representação do vetor numa base Γ ortonormal, equação (2.24), exceto pelo fator A^{-1} que é uma medida do escalamento de energia no conjunto de expansão, o *frame* Φ .

2.3.3 *Operador do Frame*

Para um dado *frame* Ψ no espaço \mathbb{H} podemos associar um operador \mathbf{F} que representa uma série de produtos internos:

$$(\mathbf{F}\mathbf{x})_m = \langle \mathbf{x}, \psi_m \rangle. \quad (2.32)$$

O operador \mathbf{F} será então uma matriz com as colunas compostas por ψ_m . Veja que na verdade o operador é $(\mathbf{F}\cdot)$, a multiplicação do vetor pela matriz \mathbf{F} . Com este operador podemos reformular a equação (2.28) para [9, 10]:

$$A\mathbf{I}_N \leq \mathbf{F}^*\mathbf{F} \leq B\mathbf{I}_N. \quad (2.33)$$

Onde \mathbf{I}_N é a matriz identidade de ordem N . E para o caso de um *frame* apertado, como os limites são iguais, tem-se:

$$A\mathbf{I}_N = \mathbf{F}^*\mathbf{F}. \quad (2.34)$$

¹Do inglês: *tight frame*.

Podemos ainda definir um *frame* dual a \mathbf{F} definido como $\tilde{\Psi} = \{\tilde{\psi}_m\}$ onde:

$$\tilde{\psi}_m = (\mathbf{F}^* \mathbf{F}) \psi_m \quad \forall m \in \{1, \dots, M\}, \quad (2.35)$$

teremos então para $\tilde{\Psi}$ os contornos A^{-1} e B^{-1} .

2.4 Representações Redundantes

Como um conjunto finito de vetores que gera um espaço \mathbb{H} forma um *frame*, o conceito de *frame* pode ser utilizado para representar vetores que pertençam ao espaço \mathbb{H} . Podemos então utilizar os *frames* para deslocar, concentrar, identificar características em/de sinais. Um vetor $\mathbf{x} \in \mathbb{H}$ poderá ser representado através de um *frame* Ψ de maneira eficiente devido à liberdade de escolha do *frame*. Isto é similar à escolha da melhor base [12]. A transformada de *Fourier* pode ser considerada como a expansão num *frame* apertado sem redundância ($M = N$), ver equação (2.31). Porém não podemos simplesmente considerar qualquer expansão de um vetor num *frame* Ψ como uma representação do vetor sem considerar a reconstrução do vetor a partir da expansão [9].

Podemos então representar um vetor \mathbf{x} no *frame* Ψ , no caso de *frame* apertado com contorno $A = 1$, que é o que utilizaremos deste ponto em diante, através de:

$$\mathbf{x} = \sum_m \alpha_m \psi_m, \quad (2.36)$$

onde os coeficientes $\alpha_m \in \mathbb{R}$, $m \in \{1, \dots, M\}$, e são dados por:

$$\alpha_m = \langle \mathbf{x}, \psi_m \rangle. \quad (2.37)$$

Caso $M > N$ os coeficientes α_m não são mais dados pela equação (2.37), e podem não ser únicos. Assim teremos mais de uma solução ou forma de representar \mathbf{x} . A esta idéia chamamos de **Representação Redundante**. O termo redundante vem do fato de termos mais coeficientes que a dimensão do sinal². Apesar disso, através de representações redundantes podemos encontrar representações compactas de sinais, conforme veremos no decorrer deste trabalho.

²Estas representações são chamadas também de *Overcomplete*.

2.4.1 Cálculo dos Coeficientes

O problema a abordar agora é como encontrar os coeficientes da expansão de um sinal num *frame*. É um problema importante pois as funções sobre as quais estamos expandindo $x(t)$ ou o vetor \mathbf{x} , ao utilizarmos *frames* não serão necessariamente L.I.³ nem ortogonais. Com isto a utilização do produto interno como apresentado até aqui para o cálculo dos coeficientes necessita algumas modificações.

Um método que trabalha com todo o conjunto de funções que compõem o *frame* é o método dos *frames* [8]. Este pode ser visto na sub-subseção 3.1.2.1. Os seus problemas são a alta complexidade computacional e o fato de não gerar uma representação compacta. Utilizaremos então uma abordagem em passos sucessivos através da qual possamos garantir que em dado passo utilizamos a função que mais se “parece” com o sinal, para representá-lo, dentre as possíveis funções, os *Matching Pursuits* [1].

Chamemos o conjunto de funções f_k , que compõem o *frame*, de \mathbf{F} . Podemos imaginar uma expansão maximizando o módulo dos coeficientes a cada passo, num processo sucessivo e com isto teremos uma compactação da representação. Para isto, a cada passo i da representação devemos escolher o maior produto interno entre o sinal e o elemento do conjunto \mathbf{F} . Este expediente pode ser visto na equação (2.38). Então a cada passo k do processo de aproximação teremos:

$$\alpha_k = \max_n \int_t x_k(t) f_n(t) dt \quad f_n \in \mathbf{F}. \quad (2.38)$$

Caso as componentes não sejam L.I., o sinal $x_k(t)$ deverá ser atualizado a cada passo k para a realização do cálculo do coeficiente $k + 1$. Teremos que recalculá-lo a cada passo então a parte do sinal que falta representar, pois a cada passo retiramos do sinal a função $f_n(t)$ que mais se assemelha ao sinal. A cada passo teremos então um sinal $x_k(t)$ para o qual calcularemos o coeficiente α_k correspondente. O sinal em cada passo será:

$$x_k(t) = x_{k-1}(t) - \alpha_{k-1} f_{k-1}(t). \quad (2.39)$$

Vejamos algumas conseqüências deste tipo de representação:

³Linearmente Independentes

- Teremos, a cada passo da representação, “informação” decrescente sobre o sinal.
- Podemos parar a representação quando já tivermos atingido um critério de erro determinado.
- Devido ao apresentado acima, podemos dizer que temos uma “aproximação” do sinal.
- Podemos dizer que temos uma representação adaptativa já que “escolhemos” os vetores mais convenientes, que possuem a maior “semelhança” a cada passo.
- Teremos então uma representação mais compacta do sinal.
- Necessitamos, além de armazenar o coeficiente, armazenar também um índice que indique a função utilizada a cada passo da representação. Ou um conjunto de parâmetros que defina a função utilizada.

Os conceitos introduzidos nesta seção serão expandidos no próximo capítulo.

Capítulo 3

Representações Adaptativas

Uma das características que torna interessante o uso de representações redundantes, seção 2.4, é a sua capacidade de adaptação na representação do sinal. Com isto queremos dizer que as funções ou vetores utilizados para representar o sinal são escolhidos por um algoritmo dentre os elementos de um dicionário - o conjunto de funções permitidas para representar o sinal. Estas representações são chamadas de adaptativas pois os elementos utilizados para representar o sinal dependem do mesmo, ou seja, a representação se adapta ao sinal. É interessante notar que o conceito de representação redundante está ligado às representações adaptativas já que se utilizarmos um dicionário redundante podemos escolher a partir do mesmo em quais funções decompor o sinal. Quanto maior a redundância maior será a adaptabilidade permitida na representação, já que nos será possível escolher dentre um número maior de elementos. A possibilidade de adaptação permite que sejamos coerentes com o sinal que desejamos representar, isto é, que escolhamos elementos do dicionário que possuem uma relação fenomenológica com o sinal. Isto nos permite separar os fenômenos que compõem o sinal. Voltemos a atenção ao que se espera de uma representação adaptativa. Segundo Donoho [13] os objetivos de uma representação adaptativa seriam:

- **Velocidade** - Deverá ser possível obter uma representação em um tempo $O(n)^1$ ou $O(n\log(n))$.

¹ $O(\cdot)$ quer dizer a ordem, e n no caso é a dimensão do sinal.

- **Esparsidade** - Deveremos obter a representação mais esparsa do sinal, isto é: com menor número de coeficientes significativos dentre as diversas possibilidades com os elementos do dicionário. É a idéia de compacidade apresentada na seção 2.4.1.
- **Separação Perfeita** - Quando o sinal é composto pela superposição de fenômenos separados, estes devem ser percebidos na representação separadamente.
- **Super-Resolução** - Deveremos obter uma resolução da representação que é muito maior que a possível com as ferramentas tradicionais não adaptativas. Um exemplo é a obtenção de uma discriminação melhor entre componentes de frequências bem próximas.
- **Estabilidade** - Pequenas perturbações do sinal não deverão degradar o desempenho.

O que estas características impõem? O primeiro fator trata da velocidade, pois não adianta termos uma decomposição extremamente lenta, ou seja é necessário ter um algoritmo prático. A esparsidade impõe que façamos uma representação que concentre a informação ao máximo, o que diminui a quantidade de coeficientes necessários para que a informação seja armazenada. A separabilidade impõe que possamos identificar os diferentes fenômenos presentes no sinal a ser armazenado/analizado. A super-resolução é um compromisso: já que gastaremos um poder computacional maior queremos poder analisar o sinal com uma resolução maior que os métodos de menor custo computacional. Finalmente a estabilidade coloca que a representação deve saber discriminar o que é sinal do que é ruído.

Neste capítulo veremos alguns dos problemas encontrados em expansões adaptativas e a seguir um método para a representação adaptativa de sinais, o algoritmo de *Matching Pursuits* [2, 14, 1]. Após ver este método abordaremos a idéia de representações coerentes. A utilização do *Matching Pursuits* com átomos Tempo-Frequência ou dicionário de Gabor será vista no capítulo a seguir.

3.1 Procurando a Melhor Base

Desde que o dicionário D seja um *frame*, seção 2.3, ele pode ser utilizado para decompor um sinal. Antes de vermos propriamente o algoritmo que utilizaremos, vejamos os problemas que surgem ao se tentar encontrar as funções que melhor representam um sinal f a partir de um dicionário D . Esta melhor decomposição está relacionada a duas idéias: a primeira, ser compacta, obtendo o menor número de coeficientes não nulos para representar o sinal; e a segunda, extrair informação dos fenômenos presentes no sinal a decompor. Sendo M o número de elementos do dicionário e N a dimensão do sinal podemos definir a redundância do dicionário conforme a equação (2.29), $r = \frac{M}{N}$. O problema da escolha das funções que utilizaremos na decomposição do sinal, está ligada a três fatores:

- O sinal f e sua dimensão N .
- O dicionário D e a quantidade de elementos do mesmo M .
- A quantidade de funções que utilizaremos para representar um sinal, m (quantidade de coeficientes).

O *Matching Pursuits*, seção 3.2, é um algoritmo sub-ótimo que nos permite ignorar a complexidade de encontrar as m funções de D que melhor representam o sinal f . O problema de encontrar exatamente as m funções de D que melhor representam f é um problema NP-completo [15, 16].

3.1.1 Adaptabilidade

A adaptabilidade destas decomposições está exatamente na possibilidade de escolha das m funções de D que serão utilizadas para representar f . A redundância, nos permite um dicionário com um grande número de funções. Isto é bastante desejável, porém implica na complexidade de cálculo para escolher quais funções do dicionário são as melhores para representar o sinal f . Nos métodos a seguir amostramos o sinal f e geramos seu vetor correspondente \mathbf{f} , e o dicionário passará a ser composto de vetores que são as funções amostradas.

3.1.2 Diferentes Abordagens

Alguns métodos existem para encontrar os vetores do dicionário a utilizar. Veremos alguns destes de forma superficial. O objetivo é encontrar os pontos a favor e contra cada um deles. Quase todos se baseiam na construção de uma matriz Φ cujas colunas são os vetores que compõem o dicionário. Um vetor coluna dos coeficientes das diversas expansões α , e o problema passa a ser:

$$\Phi\alpha = \mathbf{f}. \quad (3.1)$$

O problema então consiste em encontrar uma solução para a equação (3.1).

3.1.2.1 Método dos *Frames*

Neste método [8] acha-se a solução que possua a norma l_2 mínima.

$$\min_{\Phi\alpha=\mathbf{f}} \|\alpha\|_2. \quad (3.2)$$

Esta solução pode ser encontrada em um tempo $O(n \log(n))$. Porém possui alguns problemas [13]: a solução encontrada é uma média de todas as possíveis soluções; sendo assim a representação não é esparsa e possui pouca capacidade de super-resolução.

3.1.2.2 Melhor Base Ortonormal

Proposto em [12], *Best Ortho Basis*, consiste na formação de diversas bases ortogonais. Neste caso, o dicionário, D , será composto de uma vasta coleção de bases ortonormais, \mathbb{B}_I . Sendo $\alpha_{\mathbb{B}_I}$ o vetor de coeficientes da expansão de \mathbf{f} na base \mathbb{B}_I e definindo uma entropia $\varepsilon(\alpha_{\mathbb{B}_I})$ escolhe-se a base \mathbb{B}_I com entropia mínima:

$$\min_{\mathbb{B}_I \subset D} \varepsilon(\alpha_{\mathbb{B}_I}). \quad (3.3)$$

Em [12], impondo-se restrições na entropia ε , é apresentado um algoritmo para solução rápida da equação (3.3).

Segundo [13], o método é rápido, de ordem $O(n \log(n))$, e normalmente fornece soluções esparsas. Porém, tem deficiências quando o sinal f é uma composição de sinais não ortogonais. Além disso o dicionário D de [12] não possui uma resolução

no tempo tão boa, o que dificulta encontrar os instantes de ocorrência de sinais. Devemos considerar, ainda, o fato de que a escolha de uma base e não de cada função por si só diminui a adaptabilidade da representação.

3.1.2.3 Otimização Massiva

Para evitar a limitação dos métodos já apresentados e dos por vir, poderíamos ao invés de realizar uma aproximação por passos, realizar realmente uma otimização global para resolver a equação (3.1). Nesta, pegaríamos todos os possíveis conjuntos com m ou menos elementos de D e os compararíamos com o sinal, \mathbf{f} , pelo método dos mínimos quadrados. Porém esta abordagem é impraticável devido à sua complexidade computacional.

3.1.2.4 *Basis Pursuit*

Em [17] temos o método de *Basis Pursuit*, que é baseado em otimização global. Este é semelhante ao método dos *frames* porém busca a solução com menor norma l_1 . Vale ressaltar que esta solução só é possível graças a recentes avanços em programação linear. Neste método teremos então uma solução com a forma[13]:

$$\min_{\Phi\alpha=\mathbf{f}} \|\alpha\|_1. \quad (3.4)$$

3.1.2.5 *Matching Pursuits* - Aproximação por Passos

Neste caso teremos a representação adaptativa redundante similar à apresentada na seção 2.4. Este método é o *Matching Pursuits*, que será descrito a seguir na seção 3.2. Possui um problema bastante interessante: por ser um algoritmo voraz, após algumas iterações pode vir a consumir grande parte de suas iterações corrigindo erros cometidos nas primeiras iterações [18, 1, 19]. Isto se deve ao fato de retirarmos do sinal a projeção completa da função do dicionário escolhida no passo; entretanto a função extraída pode ter características de outras. Expandiremos estas idéias e veremos uma forma de contornar este problema na seção 3.3.

3.2 Matching Pursuits em Espaços Finitos

O algoritmo *Matching Pursuits* apareceu inicialmente em [1]. É um algoritmo de decomposição de sinais através de passos. Isto quer dizer que progressivamente representamos mais informação do sinal. Ou seja, o erro na reconstrução do sinal a partir da decomposição no passo $n + 1$ será menor que no passo n .

Seja o vetor $\mathbf{x} \in \mathbb{H}$, onde \mathbb{H} é um espaço de *Hilbert*. Definamos um dicionário $D = (\mathbf{g}_\gamma)_{\gamma \in \Gamma}$, tal que os vetores do dicionário possuam módulo unitário, $\|\mathbf{g}_\gamma\| = 1$. Se o dicionário for completo [1], ou seja, expande o espaço \mathbb{H} , podemos representar o vetor \mathbf{x} como uma soma de diferentes \mathbf{g}_{γ_n} pertencentes ao dicionário². Isto é:

$$\mathbf{x} = \sum_n \alpha_n \mathbf{g}_{\gamma_n}. \quad (3.5)$$

Assim como na seção 2.4.1, o problema que surge é como calcular os coeficientes α_n . Escolhemos \mathbf{g}_{γ_0} tal que:

$$\langle \mathbf{x}, \mathbf{g}_{\gamma_0} \rangle = \max_{\gamma \in \Gamma} \langle \mathbf{x}, \mathbf{g}_\gamma \rangle. \quad (3.6)$$

Então \mathbf{x} pode ser decomposto em duas partes:

$$\mathbf{x} = \langle \mathbf{x}, \mathbf{g}_{\gamma_0} \rangle \mathbf{g}_{\gamma_0} + \mathbf{R}_x^0, \quad (3.7)$$

onde é \mathbf{R}_x^0 o resíduo após aproximarmos \mathbf{x} na direção \mathbf{g}_{γ_0} , sendo:

$$\mathbf{R}_x^0 = \mathbf{x} - \langle \mathbf{x}, \mathbf{g}_{\gamma_0} \rangle \mathbf{g}_{\gamma_0}, \quad (3.8)$$

e \mathbf{g}_{γ_0} é tal que:

$$\langle \mathbf{x}, \mathbf{g}_{\gamma_0} \rangle \geq \max_{\gamma \in \Gamma, \gamma \neq \gamma_0} \langle \mathbf{x}, \mathbf{g}_\gamma \rangle. \quad (3.9)$$

Se:

$$\alpha_0 = \max_{\gamma \in \Gamma} \langle \mathbf{x}, \mathbf{g}_\gamma \rangle, \quad (3.10)$$

a equação (3.7) fica:

$$\mathbf{x} = \alpha_0 \mathbf{g}_{\gamma_0} + \mathbf{R}_x^0, \quad (3.11)$$

onde \mathbf{R}_x^0 é dado pela equação (3.8), ou:

$$\mathbf{R}_x^0 = \mathbf{x} - \alpha_0 \mathbf{g}_{\gamma_0}. \quad (3.12)$$

Este tipo de decomposição pode ser vista na figura 3.1.

²Uma combinação linear dos elementos de D pode representar qualquer vetor $\mathbf{y} \in \mathbb{H}$.

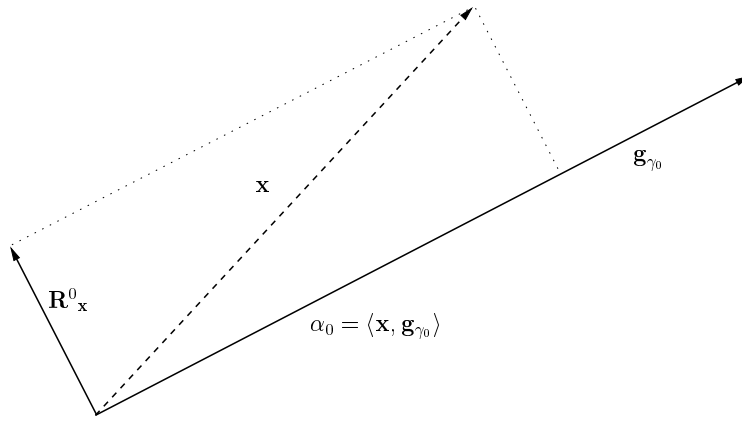


Figura 3.1: Representação Gráfica de um Passo do *Matching Pursuit*.

Podemos ver que $\mathbf{R}^0_{\mathbf{x}}$ e \mathbf{g}_{γ_0} são ortogonais e portanto a energia do sinal é conservada:

$$\|\mathbf{x}\|^2 = \|\langle \mathbf{x}, \mathbf{g}_{\gamma_0} \rangle\|^2 + \|\mathbf{R}^0_{\mathbf{x}}\|^2. \quad (3.13)$$

Tendo $\mathbf{R}^n_{\mathbf{x}}$, suponha que computamos o resíduo de ordem $n + 1$, $\mathbf{R}^{n+1}_{\mathbf{x}}$ através de:

$$\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle = \max_{\gamma \in \Gamma} \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma} \rangle, \quad (3.14)$$

o que implica:

$$\mathbf{R}^n_{\mathbf{x}} = \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle \mathbf{g}_{\gamma_n} + \mathbf{R}^{n+1}_{\mathbf{x}}. \quad (3.15)$$

A equação (3.15) define o resíduo de ordem $n + 1$, $\mathbf{R}^{n+1}_{\mathbf{x}}$.

Assim temos que \mathbf{x} é dado por:

$$\mathbf{x} = \sum_{n=0}^{m-1} (\mathbf{R}^n_{\mathbf{x}} - \mathbf{R}^{n+1}_{\mathbf{x}}) + \mathbf{R}^m_{\mathbf{x}}. \quad (3.16)$$

O processo de decomposição apresentado na equação (3.14) é válido para n , através do mesmo chegamos a $n + 1$, e sendo este válido para $n = 0$, por indução temos:

$$\mathbf{x} = \sum_{n=0}^{m-1} \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle \mathbf{g}_{\gamma_n} + \mathbf{R}^m_{\mathbf{x}}. \quad (3.17)$$

A ortogonalidade entre o vetor do dicionário \mathbf{g}_{γ_n} e $\mathbf{R}^{n+1}_{\mathbf{x}}$ impõe:

$$\|\mathbf{R}^n_{\mathbf{x}}\|^2 = \|\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle\|^2 + \|\mathbf{R}^{n+1}_{\mathbf{x}}\|^2. \quad (3.18)$$

Enquanto a energia do vetor, $\|\mathbf{x}\|^2$, pode ser decomposta pela soma:

$$\|\mathbf{x}\|^2 = \|\mathbf{R}^{-1}_{\mathbf{x}}\|^2 = \sum_{n=0}^{m-1} (\|\mathbf{R}^n_{\mathbf{x}}\|^2 - \|\mathbf{R}^{n+1}_{\mathbf{x}}\|^2) + \|\mathbf{R}^m_{\mathbf{x}}\|^2, \quad (3.19)$$

$\mathbf{R}^{-1}_{\mathbf{x}}$ seria o resíduo antes do primeiro passo, 0. E usando a equação (3.18) teremos a equação de conservação de energia:

$$\|\mathbf{x}\|^2 = \sum_{n=0}^{m-1} \|\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle\|^2 + \|\mathbf{R}^m_{\mathbf{x}}\|^2. \quad (3.20)$$

Em [14] temos um teorema que garante a convergência do algoritmo descrito na expansão de um vetor \mathbf{x} .

Teorema 3.1 *Seja $\mathbf{x} \in \mathbb{H}$. O resíduo $\mathbf{R}^n_{\mathbf{x}}$ definido pela equação (3.15) satisfaz:*

$$\lim_{n \rightarrow +\infty} \|\mathbf{R}^n_{\mathbf{x}}\| = 0, \quad (3.21)$$

com o que:

$$\mathbf{x} = \sum_{n=0}^{+\infty} \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle \mathbf{g}_{\gamma_n}, \quad (3.22)$$

e

$$\|\mathbf{x}\|^2 = \sum_{n=0}^{+\infty} \|\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle\|^2, \quad (3.23)$$

Este é um resultado importante que garante a convergência da decomposição, cuja prova pode ser vista em [20]. O teorema garante que qualquer vetor \mathbf{x} é caracterizado pela seqüência $(\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle, \gamma_n)_{n \in \mathbb{N}}$ comumente chamado de *structure book*. Esta seqüência especifica o coeficiente da expansão e o vetor do dicionário utilizado a cada passo da expansão.

Um ponto interessante é levantado em [18] onde os autores sugerem que uma das questões a entender seria como se comporta o resíduo $\mathbf{R}^n_{\mathbf{x}}$ à medida que n aumenta. Abordaremos isto nos dois próximos capítulos.

3.2.1 Redundância no *Matching Pursuits*

Podemos perceber que qualquer uma das metodologias apresentadas para representações adaptativas se encaixam no conceito de representação redundante apresentado na subseção 2.3.1. A redundância r do dicionário utilizado na representação, será dada pela razão entre o número de componentes do dicionário, M , e a dimensão do espaço do sinal a representar N . Isto é $r = \frac{M}{N}$.

Esta é uma idéia intuitiva, pois na verdade ela nos indica um grau de liberdade por dimensão do sinal na representação do mesmo. Uma base do espaço \mathbb{K} de

dimensão N será formada por N elementos. Se tivermos um dicionário D composto por $4N$ elementos teremos uma redundância $r = 4$ e isto implica que teremos 4 vezes mais vetores para aproximar o sinal num passo específico do algoritmo. Ou seja a liberdade de escolha dos vetores a serem utilizados na representação é 4 vezes maior do que numa base do espaço de dimensão N quando utilizada no mesmo *framework*, o algoritmo de *Matching Pursuits*. Isto permite isolar e identificar sinais com determinadas características ou informações que serão definidas pelos elementos dos dicionários. Então para procurar um conjunto de padrões no sinal basta que estes padrões façam parte do dicionário e estes serão identificados, especialmente no *Matching Pursuits*, através da correlação (produto interno), do padrão presente no dicionário com o sinal no qual procuramos o padrão.

No parágrafo acima comentamos sobre como a utilização de dicionários redundantes nos permite a busca e identificação de padrões no sinal. Porém é importante observar que o dicionário deverá contar com elementos que permitam encontrar estes sinais específicos. No capítulo 4 veremos o dicionário de Gabor que possui algumas características interessantes que nos permitirão localizar padrões e informações conjugadas do sinal tanto no tempo como na frequência, além de fornecer uma representação tempo-frequência [21] adaptativa do sinal.

3.2.2 Implementação Rápida

Além do algoritmo apresentado na seção 3.2 para o resíduo $\mathbf{R}^{n+1}_{\mathbf{x}}$ e a função \mathbf{g}_{γ_n} , *Mallat* e *Zhang* [18, 1] desenvolveram um algoritmo rápido que não depende do cálculo do produto interno a cada iteração do algoritmo de *Matching Pursuits*. Este algoritmo foi estudado também por *Davis, Mallat* e *Zhang* em [14] para o caso de *Orthogonal Pursuits*. Aqui nos restringiremos ao *Matching Pursuits* apresentado na seção 3.2.

No algoritmo apresentado, a cada passo, n , escolhemos um elemento do dicionário \mathbf{g}_{γ_n} para aproximar o sinal de acordo com a equação (3.14). No primeiro passo teremos computados todos os produtos entre o sinal e os elementos do dicionário. Supondo que temos o produto interno $\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma} \rangle$ onde $\gamma \in \Gamma$, podemos então computar o produto interno do resíduo $\mathbf{R}^{n+1}_{\mathbf{x}}$ a ser aproximado no passo seguinte com os elementos do dicionário através de uma fórmula de atualização que pode ser

derivada da equação (3.15) sendo dada por:

$$\langle \mathbf{R}^{n+1}_{\mathbf{x}}, \mathbf{g}_\gamma \rangle = \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_\gamma \rangle - \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle \langle \mathbf{g}_{\gamma_n}, \mathbf{g}_\gamma \rangle. \quad (3.24)$$

Se tivermos tabelado todos os produtos internos entre os elementos do dicionário, $\langle \mathbf{g}_{\gamma_1}, \mathbf{g}_{\gamma_2} \rangle$ onde γ_1 e γ_2 são quaisquer dois elementos diferentes de Γ , não necessitamos fazer o cálculo do produto interno a cada passo entre o resíduo e os elementos do dicionário. Isto diminui consideravelmente o número de operações necessárias a cada passo do algoritmo.

Esta é uma boa solução para agilizar o algoritmo. Foi implementada neste trabalho, porém o seu custo é o alto número de operações necessárias ao iniciar o processamento, quando necessitamos calcular todos os $\langle \mathbf{g}_{\gamma_1}, \mathbf{g}_{\gamma_2} \rangle$, além da memória utilizada para armazenar estes produtos internos. Em nosso trabalho esta não é uma boa solução pois não estamos realizando um processamento *on-line* além de trabalharmos com sinais com números de amostras diferentes, que faz o dicionário mudar e a tabela tem que ser recalculada. Além disso como experimentamos algumas modificações no dicionário, que descreveremos ao longo do texto, a cada uma destas teríamos que recalculamos estes elementos. Este tipo de implementação é entretanto bastante útil e válida em operações *on-line*.

3.3 Representações Coerentes

Até o presente vimos como decompor um sinal através do algoritmo de *Matching Pursuits*. Vimos que esta decomposição escolherá, dentro do dicionário D , a cada passo, o elemento que melhor representa o resíduo decorrente do passo anterior. Entretanto uma decomposição realizada desta forma tende a ter um número infinito de passos. Então, surge a questão: a partir de quantos passos a nossa decomposição é suficiente, ou seja quando aproximamos informação suficiente do ponto de vista dos fenômenos presentes no sinal? A resposta trivial seria que ao atingirmos um erro médio quadrático suficientemente baixo teríamos uma decomposição razoável do sinal. O erro médio quadrático é dado por:

$$\mathbf{mse} = E\{\|\mathbf{x} - \tilde{\mathbf{x}}\|^2\}, \quad (3.25)$$

onde \mathbf{x} é o sinal e $\tilde{\mathbf{x}}$ é o sinal aproximado. Utilizar esta medida para saber quando a representação é característica do sinal seria a resposta trivial já que desta forma minimizamos o erro. Porém para que isto ocorra pode ser necessário um número infinito de passos [14, 20].

Se estamos preocupados em analisar características do sinal, não em representá-lo tal qual, queremos extrair informações sobre o que está presente no sinal, podemos utilizar uma representação redundante e adaptativa. Neste caso se os elementos do dicionário definem características do sinal, ou diferentes estruturas que podem estar presentes no mesmo, podemos limitar o número de estruturas que desejamos analisar ou extrair do sinal. Desta forma podemos estabelecer *a priori* quantos passos realizar e portanto quantos elementos extrair do sinal. Entretanto, esta abordagem é independente do sinal, pois apesar de ela escolher elementos condizentes, coerentes, com o sinal que desejamos analisar, ela é incapaz de saber se esses elementos realmente são suficientes para caracterizar bem o sinal.

Necessitamos então de uma abordagem que nos permita garantir que realmente estamos extraindo as características corretas do sinal e na quantidade apropriada para representá-lo, a partir dos elementos do dicionário. Aqui há dois pontos importantes: entender que estaremos extraindo características do sinal, além de entender que as características que extraíremos do sinal têm que poder ser representadas pelo dicionário!

Tentando resolver este problema tentaremos realizar representações que chamaremos de coerentes. Utilizaremos elementos do dicionário que além de correlacionados sejam coerentes com o sinal. Estudos sobre isto podem ser vistos em [14, 19, 20]. Na verdade o objetivo do conceito de coerência que apresentaremos é permitir identificar a partir de que ponto paramos de representar informação sobre o sinal e passamos a representar ruído.

Definamos primeiro uma taxa de aproximação de \mathbf{x} [18]:

Definição 3.1 *Seja $D = (\mathbf{g}_\gamma)_{\gamma \in \Gamma}$ um dicionário do espaço \mathbb{H} e $\mathbf{x} \in \mathbb{H}$. Chamaremos de taxa de aproximação do sinal \mathbf{x} o valor $\lambda(\mathbf{x})$ definido por:*

$$\lambda(\mathbf{x}) = \max_{\gamma \in \Gamma} \frac{\|\langle \mathbf{x}, \mathbf{g}_\gamma \rangle\|}{\|\mathbf{x}\|}. \quad (3.26)$$

Podemos definir a taxa de aproximação do dicionário D , $\lambda(D)$ como sendo:

$$\lambda(D) = \min_{\mathbf{x} \in \mathbb{H}} \lambda(\mathbf{x}). \quad (3.27)$$

Lema 3.1 *Se D for um dicionário completo de \mathbb{H} teremos que:*

$$\lambda(D) = \min_{\mathbf{x} \in \mathbb{H}} \lambda(\mathbf{x}) > 0. \quad (3.28)$$

A prova deste lema pode ser encontrada em [18]. Vale a pena notar que $\lambda(D)$ será o cosseno do menor ângulo possível entre um vetor $\mathbf{v} \in \mathbb{H}$ e o elemento de D mais próximo de \mathbf{v} . Além do mais podemos encontrar uma relação entre o resíduo e a taxa de aproximação do dicionário, equação (3.29) [18], que nos indica um decaimento exponencial do resíduo de um vetor \mathbf{x} em um espaço finito \mathbb{H} :

$$\| \mathbf{R}^n_{\mathbf{x}} \| \leq \| \mathbf{x} \| (1 - \lambda^2(D))^{\frac{n}{2}}. \quad (3.29)$$

Podemos também definir a taxa de aproximação a cada passo do algoritmo. Isto é a taxa de aproximação de $\mathbf{R}^n_{\mathbf{x}}$ que será dada por:

$$\lambda(\mathbf{R}^n_{\mathbf{x}}) = \max_{\gamma \in \Gamma} \frac{\| \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma} \rangle \|}{\| \mathbf{R}^n_{\mathbf{x}} \|}, \quad (3.30)$$

ou:

$$\lambda(\mathbf{R}^n_{\mathbf{x}}) = \frac{\| \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle \|}{\| \mathbf{R}^n_{\mathbf{x}} \|}, \quad (3.31)$$

esta equação nada mais é que a definição da taxa de aproximação aplicada ao resíduo. Podemos definir, a partir desta, taxas de aproximação médias, que computam a taxa de aproximação em um número de passos M do processo de aproximação do sinal \mathbf{x} no algoritmo apresentado. Podemos definir então a taxa de aproximação média dos resíduos:

$$\lambda_{\text{med}}(\mathbf{R}^M_{\mathbf{x}}) = \frac{1}{M} \sum_{n=0}^{M-1} \frac{\| \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle \|}{\| \mathbf{R}^n_{\mathbf{x}} \|}. \quad (3.32)$$

Em outros casos como um fator de limite da decomposição podemos definir a taxa de aproximação do sinal no passo M que seria dada por:

- Em módulo:

$$\lambda_{\text{tot}}(\mathbf{x}_M) = \frac{\| \sum_{n=0}^{M-1} \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle \mathbf{g}_{\gamma_n} \|}{\| \mathbf{x} \|}; \quad (3.33)$$

- Em energia:

$$\lambda_{\text{tot}}(\mathbf{x}_M) = \frac{\| \sum_{n=0}^{M-1} \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle \|^2}{\| \mathbf{x} \|^2}; \quad (3.34)$$

Ambas são medidas que nos indicam quanto do sinal estamos representando em termos de módulo para a primeira e de energia na segunda. Porém, nenhuma delas nos fornece informação se a representação está parecida ou não com o sinal. Para isto poderíamos utilizar a razão da energia do ruído pela energia do sinal:

$$\mathbf{RNS}(\mathbf{x}_M) = \frac{\|\mathbf{x} - \sum_{n=0}^{M-1} \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle\|^2}{\|\mathbf{x}\|^2}, \quad (3.35)$$

ou o erro normalizado:

$$\epsilon_N(\mathbf{x}_M) = \frac{\|\mathbf{x} - \sum_{n=0}^{M-1} \langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle\|}{\|\mathbf{x}\|}, \quad (3.36)$$

Apesar destas medidas de erro fornecerem uma boa idéia sobre a similaridade da representação do sinal em relação ao sinal original elas não permitem saber se a representação é coerente ou não. Como já mencionamos, para isto devemos primeiro estudar as características do dicionário e encontrar uma forma de saber se num passo n qualquer da aproximação ainda estamos representando o sinal de forma coerente com o dicionário.

3.3.1 Representando Sinais Coerentemente

Imagine um ruído gaussiano \mathbf{g} . Agora decomponha este ruído através do algoritmo de *Matching Pursuits*. Podemos a cada passo computar a taxa de aproximação do ruído:

$$\lambda(\mathbf{R}^n_{\mathbf{g}}) = \frac{\|\langle \mathbf{R}^n_{\mathbf{g}}, \mathbf{g}_{\gamma_n} \rangle\|}{\|\mathbf{R}^n_{\mathbf{g}}\|}. \quad (3.37)$$

Esta taxa indicaria o quanto aproximamos do sinal a cada passo. Em [20] é demonstrado que a partir de uma dada iteração esta taxa permanece aproximadamente constante. Isto não quer dizer que o sinal não está sendo aproximado, simplesmente indica que a partir de um certo ponto a aproximação permanece num fator constante. Esta característica indica que temos para o ruído gaussiano um limiar de taxa de aproximação. Por quê? O ruído gaussiano está distribuído com uma densidade de probabilidade uniforme sobre a superfície da esfera, na verdade tem uma p.d.f.³ independente do ângulo. Então qualquer ponto da esfera será considerado ao utilizarmos o sinal gaussiano como entrada. Quando decomposmos o

³função densidade de probabilidade

ruído gaussiano através do algoritmo de *Matching Pursuits*, o resíduo da decomposição será também ruído com a mesma densidade de probabilidade, e isto se repete a cada passo. Como os vetores do dicionário estão em direções arbitrárias, não serão suficientes para representar caracteristicamente o ruído que possui uma densidade de probabilidade independente do ângulo no espaço de dimensão do sinal [22]. O que ocorre é que sendo $\lambda(D)$ uma medida dos ângulos entre os elementos do dicionário e um vetor qualquer, esta nos indica quão próxima pode ser a direção de um elemento do dicionário de um vetor qualquer, e como o ruído gaussiano terá sua direção uniformemente distribuída teremos um limite na representação do ruído gaussiano. Neste limite podemos dizer que não estamos representando coerentemente o sinal.

De posse deste argumento podemos estipular então um modo de realizar representações coerentes. Como? Compute a taxa de aproximação média para os resíduos da decomposição de um ruído gaussiano:

$$\lambda_{\text{med}}(\mathbf{R}^M_{\mathbf{g}}) = \frac{1}{M} \sum_{n=0}^{M-1} \frac{\|\langle \mathbf{R}^n_{\mathbf{g}}, \mathbf{g}_{\gamma_n} \rangle\|}{\|\mathbf{R}^n_{\mathbf{g}}\|}. \quad (3.38)$$

Agora decomponha o sinal, \mathbf{x} , e compute a taxa de aproximação em cada passo:

$$\lambda(\mathbf{R}^n_{\mathbf{x}}) = \frac{\|\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle\|}{\|\mathbf{R}^n_{\mathbf{x}}\|}. \quad (3.39)$$

Quando a taxa de aproximação do resíduo do sinal for menor que a taxa de aproximação média do ruído:

$$\lambda(\mathbf{R}^n_{\mathbf{x}}) \leq \lambda_{\text{med}}(\mathbf{R}^M_{\mathbf{g}}), \quad (3.40)$$

então paramos a aproximação pois estamos realizando aproximações sem coerência com o dicionário, é como se estivéssemos aproximando ruídos que o dicionário não é capaz de representar.

No capítulo 5 veremos estes conceitos aplicados ao dicionário de *Gabor* que utilizaremos neste trabalho para decompor os sinais. Porém um ponto deve ser observado:

- Não é interessante utilizar exatamente a aproximação do resíduo no passo, equação (3.39), e sim uma média móvel desta:

$$\lambda_{\text{med}}(\mathbf{R}^M_{\mathbf{x}}) = \frac{1}{L} \sum_{n=M-L}^{M-1} \frac{\|\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle\|}{\|\mathbf{R}^n_{\mathbf{x}}\|} \quad (3.41)$$

Com esta média móvel de L passos deixamos o procedimento mais robusto a possíveis desvios do algoritmo de *Matching Pursuits*, como as oscilações na taxa de aproximação do resíduo.

Veremos no capítulo a seguir um dicionário que têm sido amplamente utilizado no algoritmo de *Matching Pursuits*, o dicionário de *Gabor* e sua utilização neste algoritmo para representar sinais elétricos.

Capítulo 4

Dicionário de *Gabor*

Nos capítulos anteriores vimos como representar sinais, e um algoritmo para representar adaptativamente um sinal - o *Matching Pursuits*. Neste capítulo veremos o dicionário de *Gabor*. Este dicionário será utilizado no algoritmo de *Matching Pursuits* para decompor os sinais. O interesse neste dicionário reside também no fato de permitir um mapeamento dos elementos do dicionário utilizados para representar o sinal num plano tempo-freqüência, permitindo assim, além de representar o sinal, estudar as características tempo-freqüência do mesmo. Vejamos primeiro as representações tempo-freqüência.

4.1 Representações Tempo-Freqüência

Representações Tempo-Freqüência de Sinais¹ [21, 23, 2, 24] ou distribuições Tempo-Freqüência, que podemos extrapolar também para análise tempo-freqüência, são ferramentas de decomposição de sinais ou de representação de um sinal num plano que conjuga o tempo em que o sinal acontece e o seu espectro durante um período de tempo. Segundo *Cohen* [21] uma TFR é:

“ uma distribuição que representa a energia ou intensidade de um sinal simultaneamente no tempo e na freqüência.”

Na transformada de *Fourier*, apresentada na seção 2.1.1, obtemos o espectro em freqüência de um sinal para toda sua realização. De forma diferente, nas

¹Chamaremos de TFR do inglês: “Time-Frequency Representations”.

representações tempo-freqüência, obtemos informações conjugadas do espectro do sinal num intervalo de tempo. Salientamos que não obtemos uma informação precisa devido ao princípio da incerteza [2, 21, 23, 24]. Mas de qualquer forma a intenção é representar um sinal num plano tempo-freqüência onde poderemos observar as freqüências presentes no sinal num período de tempo.

Como salientamos, a representação através da transformada de *Fourier* não nos permite fazer uma análise das componentes em freqüência do sinal instantaneamente, ou até mesmo dentro de período menor que aquele composto pelo conjunto com N amostras, no caso discreto. Uma maneira bastante intuitiva de resolver este problema é através da transformada de *Fourier* de tempo curto, ou STFT² que surge ao aplicarmos uma janela g ao sinal, calculando então o espectro do sinal numa região de suporte no tempo. Devemos salientar que esta abordagem realizará uma filtragem do sinal que depende da janela utilizada na implementação da STFT. A STFT é chamada também de Transformada de *Fourier* Janelada [2, 23]. Nas equações (4.1) e (4.2) colocamos sua definição (tanto a análise quanto a síntese):

$$\begin{aligned} x_a(t) &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} SX_a(u, j\Omega) g(t-u) e^{j\Omega t} d\Omega du, \\ SX_a(u, j\Omega) &= \int_{-\infty}^{+\infty} x_a(t) g(t-u) e^{-j\Omega t} dt, \end{aligned} \quad (4.1)$$

e:

$$\begin{aligned} \hat{x}(n) &= \frac{1}{N} \sum_{m=0}^{N-1} \sum_{k=0}^{N-1} S\hat{X}(m, k) g(n-m) e^{j(\frac{2\pi}{N})nk}, \\ S\hat{X}(m, k) &= \sum_{n=0}^{N-1} \hat{x}(n) g(n-m) e^{-j(\frac{2\pi}{N})nk}, \end{aligned} \quad (4.2)$$

para $0 \leq k \leq N-1$ e $0 \leq m \leq N-1$. No caso discreto passamos a chamar a janela $g(t)$ de $g(n)$. Esta representação decompõe o sinal $\hat{x}(n)$ em um conjunto de seqüências $S\hat{X}(m, k)$ que representam as freqüências k presentes no sinal em uma vizinhança do instante m do mesmo.

4.1.1 Análise Tempo-Freqüência

A escolha das funções $g(t)$ ou das seqüências $g(n)$, nas equações (4.1) e (4.2), são o fator que permite uma representação tempo-freqüência dos sinais. Na verdade

²Do inglês: “Short-Time *Fourier* Transform”.

teremos um conjunto de funções $\gamma \in \Gamma$, que é gerado a partir de deslocamentos no tempo e na frequência de uma mesma função, a função a partir da qual podemos fazer a análise em diferentes instantes de tempo t e centrados em diferentes frequências ξ . Estas funções são chamadas de átomos tempo-freqüência [2]. Citamos aqui dois exemplos destes átomos [2]:

1. No caso da STFT [6, 2] teremos os átomos gerados por um deslocamento no tempo e uma modulação na frequência de uma função g e teremos uma família de átomos:

$$\gamma(u, \xi) = g(t - u)e^{j2\pi\xi t}. \quad (4.3)$$

2. No caso de *wavelets* [25, 2, 26] teremos átomos dilatados ou comprimidos por um parâmetro s e transladados de u , resultando:

$$\gamma(u, s) = \frac{1}{\sqrt{s}}g\left(\frac{t - u}{s}\right). \quad (4.4)$$

Dependendo do tipo de análise tempo-freqüência desejada, usaremos átomos diferentes. Algumas características dos átomos devem ser mencionadas, temos:

1. Uma largura de duração do átomo $\sigma_t(\gamma)$ (suporte no tempo), que trataremos adiante, equação (4.8).
2. Uma largura na frequência, que nos restringe a banda, $\sigma_f(\gamma)$ (suporte na frequência), que também trataremos adiante, ver equação (4.10).
3. O produto $\sigma_t(\gamma)\sigma_f(\gamma) \geq \frac{1}{2}$.

Conseqüência:

- Os pontos salientados acima indicam que temos um limite para a resolução conjunta, que é conhecido como teorema da incerteza de *Heisenberg* [21, 2], importado da física [21], característica 3 apresentada. Trataremos melhor este assunto na subseção 4.1.2.

Em [2] podem ser encontrados procedimentos para o projeto dos átomos γ e alguns átomos padrões.

Se reformularmos a representação da equação (4.1) pela apresentada na equação (4.5), que é a formulação genérica de uma TFR, teremos que basta definir a

função γ e podemos realizar diferentes análises tempo-freqüência, simplesmente utilizando a função γ desejada.

$$\begin{aligned} x(t) &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} T_x^{(\gamma)}(u, \xi) \gamma_{(u, \xi)} d\xi du, \\ T_x^{(\gamma)}(u, \xi) &= \int_{-\infty}^{+\infty} x(t) \gamma_{(u, \xi)} dt. \end{aligned} \quad (4.5)$$

4.1.2 Plano Tempo-Freqüência

Podemos num caso genérico pensar então numa família de funções $\gamma_{(u, \xi)}$, gerada a partir de deslocamentos no tempo u e na freqüência ξ de uma mesma função γ . Neste capítulo utilizaremos um conjunto de funções geradas desta forma, o dicionário de *Gabor*. Vejamos agora a informação fornecida pelo produto interno $\langle f, \gamma_{(u, \xi)} \rangle$ onde f é uma função qualquer. A partir deste produto podemos criar uma representação no plano Tempo-Freqüência (t, ω) da função f . Esta representação estará centrada numa posição (d, ξ') , com larguras σ_t no tempo e σ_f na freqüência. Vejamos como chegar a estes quatro valores, (d, ξ') , σ_t e σ_f a partir das funções γ . Temos:

$$\| \gamma_{(u, \xi)} \|^2 = \int_{-\infty}^{\infty} | \gamma_{(u, \xi)}(t) |^2 dt = 1, \quad (4.6)$$

e podemos interpretar $| \gamma_{(u, \xi)}(t) |$ como uma *p.d.f* centrada em:

$$d = \int_{-\infty}^{\infty} t | \gamma_{(u, \xi)}(t) |^2 dt. \quad (4.7)$$

Cujo espalhamento ao redor de d pode ser dado pela variância:

$$\sigma_t^2(\gamma_{(u, \xi)}) = \int_{-\infty}^{\infty} (t - d)^2 | \gamma_{(u, \xi)}(t) |^2 dt. \quad (4.8)$$

Similarmente teremos na freqüência [2]:

$$\xi' = \frac{1}{2\pi} \int_{-\infty}^{\infty} \omega | \hat{\gamma}_{(u, \xi)}(\omega) |^2 d\omega, \quad (4.9)$$

onde $\hat{\gamma}_{(u, \xi)}(\omega)$ é a transformada de *Fourier* de $\gamma_{(u, \xi)}(t)$. E seu espalhamento é dado por:

$$\sigma_\omega^2(\gamma_{(u, \xi)}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} (\omega - \xi')^2 | \hat{\gamma}_{(u, \xi)}(\omega) |^2 d\omega. \quad (4.10)$$

Teremos então um plano tempo-freqüência onde podemos representar os valores da decomposição nos átomos de acordo com o tempo e a freqüência. Este

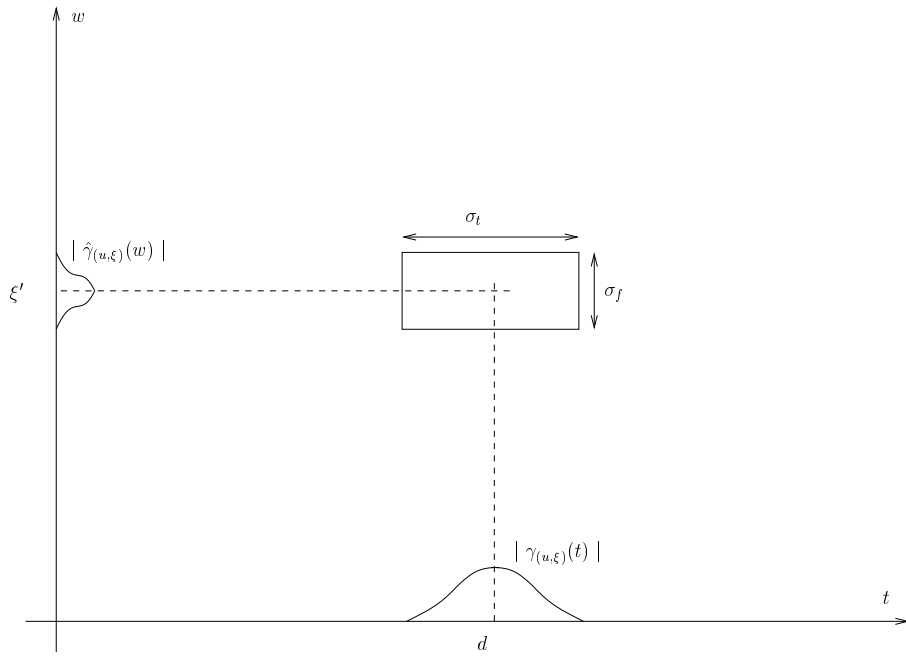


Figura 4.1: Representação Tempo-Freqüência de uma átomo $\gamma_{(u,\xi)}$

plano estará dividido em intervalos de tempo e de freqüência com as larguras σ_t e σ_f mencionadas. No caso da STFT as larguras serão constantes [2]. Porém no caso das *Wavelets*, por exemplo, teremos intervalos de tempo mais curtos em regiões de freqüências maiores e com maior banda [2], e será uma decomposição em oitavas [11]. Um átomo no plano tempo-freqüência está representado na figura 4.1. Devemos observar ainda que a resolução máxima que podemos ter é:

$$\sigma_t(\gamma)\sigma_f(\gamma) \geq \frac{1}{2}. \quad (4.11)$$

Chamamos os retângulos no plano tempo-freqüência de larguras σ_t e σ_f de caixas de *Heisenberg* e sua área mínima é dada pela equação acima. Isto implica então o máximo de separação do período de tempo e das freqüências presentes no sinal que podemos avaliar. Reparamos ainda que o ponto (d, ξ') é só uma indicação do centro do retângulo (σ_t, σ_f) e por si só não fornece nenhuma informação sobre o sinal, isto é: só é relevante a informação fornecida pela caixa de *Heisenberg*.

4.1.3 Representações Tempo-Freqüência Quadráticas

Vimos então maneiras de representar sinais através de sua correlação com os átomos tempo-freqüência. Salientamos também que a resolução que podemos obter

está limitada. Se quiséssemos agora definir uma densidade de energia no tempo e na frequência, poderíamos utilizar o quadrado da TFR [23]³.

Porém ao pensarmos em energia seria interessante satisfazer os conceitos de potência instantânea $p_x(t) = |x(t)|^2$ e de energia espectral de potência $P_x(f) = |X(f)|^2$. Elas são representadas pelas propriedades marginais:

$$\int_f T_x(t, f) df = p_x(t) = |x(t)|^2, \quad (4.12)$$

e

$$\int_t T_x(t, f) dt = P_x(f) = |X(f)|^2. \quad (4.13)$$

4.1.3.1 Princípio da Superposição Quadrática

Devemos observar que uma TFR quadrática de qualquer sinal composto $x(t) = c_1x_1(t) + c_2x_2(t)$ não será a soma simples das TFR quadráticas dos sinais $x_1(t)$ e $x_2(t)$ multiplicados respectivamente pelas constantes c_1 e c_2 . Não o será pois não estamos tratando com TFR lineares. Ao invés da composição linear teremos:

$$x(t) = c_1x_1(t) + c_2x_2(t) \Rightarrow \quad (4.14)$$

$$T_x(t, f) = |c_1|^2 T_{x_1}(t, f) + |c_2|^2 T_{x_2}(t, f) + c_1 c_2^* T_{x_1, x_2}(t, f) + c_2 c_1^* T_{x_2, x_1}(t, f)$$

Chamamos $T_x(t, f)$ de auto-TFR e $T_{x_1, x_2}(t, f)$ de TFR cruzada. O importante neste momento é observar que sempre que o sinal que estivermos representando ou analisando, através de TFR's quadráticas, for formado por uma composição de sinais, teremos termos de interferência dados pelas TFR's cruzadas. Isto acontece porque ao calcular a energia estamos realizando uma operação não linear.

4.1.4 A Distribuição de Wigner-Ville

A distribuição de *Wigner-Ville* [2] é uma TFR quadrática que satisfaz as propriedades marginais descritas nas equações (4.12) e (4.13). Mesmo assim devemos

³O quadrado é utilizado gerando:

- Positividade
- Relação com o conceito de energia

levar em consideração que a mesma não pode ser tomada pontualmente devido ao princípio da incerteza. Definimos a TFR quadrática de *Wigner-Ville*, ou distribuição de *Wigner-Ville*, na equação (4.15).

$$\begin{aligned} WD_{x,y}(t, f) &= \int_{\tau} x\left(t + \frac{\tau}{2}\right) y^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau, \\ WD_{x,y}(t, f) &= \int_{\nu} X\left(f + \frac{\nu}{2}\right) Y^*\left(f - \frac{\nu}{2}\right) e^{j2\pi f\nu} d\nu. \end{aligned} \quad (4.15)$$

Até o momento definimos a distribuição de *Wigner-Ville* para o cruzamento de duas funções $x(t)$ e $y(t)$. No caso do cálculo para uma função $f(t)$ teremos a auto TFR:

$$\begin{aligned} WD_f(t, f) &= \int_{\tau} f\left(t + \frac{\tau}{2}\right) f^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau, \\ WD_f(t, f) &= \int_{\nu} F\left(f + \frac{\nu}{2}\right) F^*\left(f - \frac{\nu}{2}\right) e^{j2\pi f\nu} d\nu. \end{aligned} \quad (4.16)$$

Outro fato interessante é que a distribuição de *Wigner-Ville* é a transformada de *Fourier* dos produtos:

$$f\left(t + \frac{\tau}{2}\right) f^*\left(t - \frac{\tau}{2}\right), \quad (4.17)$$

$$F\left(f + \frac{\nu}{2}\right) F^*\left(f - \frac{\nu}{2}\right). \quad (4.18)$$

No primeiro caso é exatamente a transformada de *Fourier* e no segundo teremos a transformada de *Fourier* inversa escalada por 2π . Este fato nos apresenta uma forma de implementação da WD que é mais importante quando passarmos para o “*universo*” discreto. Utiliza-se este expediente para implementar as TFR’s Quadráticas.

4.2 Átomo de *Gabor*

O dicionário de *Gabor* foi utilizado em [18] no mesmo âmbito que utilizaremos neste trabalho. É gerado a partir de uma coleção de átomos tempo-freqüência. Veremos nesta seção estes átomos e suas características.

Como vimos na seção 4.1 podemos gerar uma família de átomos tempo-freqüência a partir do deslocamento, do escalamento e da modulação de uma função $f(t)$ qualquer. As características dos átomos tempo-freqüência serão dependentes da função $f(t)$. Podemos chamar a escala de s , o deslocamento de u e a freqüência

modulante de ξ , e teremos um conjunto $\gamma = (s, u, \xi)$ que define um átomo tempo-freqüência através de:

$$f_\gamma(t) = \frac{1}{\sqrt{s}} f\left(\frac{t-u}{s}\right) e^{i\xi t}. \quad (4.19)$$

A transformada de *Fourier* $\hat{f}_\gamma(\omega)$ será dada por:

$$\hat{f}_\gamma(\omega) = \sqrt{s} \hat{f}(s(\omega - \xi)) e^{-i(\omega - \xi)t}. \quad (4.20)$$

Como $|\hat{f}_\gamma(\omega)|$ é par, $|\hat{f}_\gamma(\omega)|$ estará centralizada em $\omega = \xi$, com a energia concentrada na vinhança de ξ com largura proporcional a $\frac{1}{s}$, conforme é apresentado na subseção 4.1.2.

O átomo de *Gabor* será gerado permitindo deslocamentos, escalamentos e modulações de uma janela Gaussiana $g(t)$:

$$g(t) = 2^{\frac{1}{4}} e^{-\pi t^2}. \quad (4.21)$$

Teremos então átomos tempo-freqüência $g_\gamma(t)$ dados por:

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) e^{i\xi t}. \quad (4.22)$$

Podemos gerar então uma família de átomos $g_\gamma(t)$ com $\gamma \in \Gamma$.

4.3 Gerando o Dicionário de *Gabor*

De posse do átomo de *Gabor* definido na equação (4.22) podemos então gerar um dicionário a partir deste átomo com $\gamma \in \Gamma = \mathbb{R}^+ \times \mathbb{R}^2$. Neste caso os parâmetros que compõem $\gamma = (s, u, \xi)$ são reais sendo que s é não negativo. Observe que um dicionário gerado através destes átomos é contínuo e completo, além de ser redundante. Isto pode ser visto por que no caso de $\gamma = (s_0, u, \xi)$ teremos o dicionário reduzido à base da transformada janelada de *Fourier* que por sua vez é completa. Outro caso interessante é que no caso de $\gamma = (s, u, \frac{\xi_0}{s})$ teremos uma família de *Wavelets*[2]. Vale ressaltar ainda que os átomos gerados a partir da equação (4.22) são complexos.

O propósito é gerar um dicionário com os átomos de *Gabor*, o dicionário de *Gabor*, para utilizar na decomposição de sinais. Porém muitas vezes não é interessante utilizar o átomo complexo já definido; nestes casos, podemos lançar mão de

um dicionário de átomos reais dados por:

$$g_{(\gamma,\phi)}(t) = \frac{K_{(\gamma,\phi)}}{\sqrt{s}} g\left(\frac{t-u}{s}\right) \cos(\xi t + \phi), \quad (4.23)$$

onde $g(t)$ é definida como na equação (4.21) e $\gamma = (s, u, \xi)$. A constante $K_{(\gamma,\phi)}$ é tal que $\|g_{(\gamma,\phi)}(t)\| = 1$ e a fase $\phi \in [0, \pi)$. E podemos relacionar o átomo real ao complexo através de:

$$g_{(\gamma,\phi)}(t) = \frac{K_{(\gamma,\phi)}}{\sqrt{s}} [e^{i\phi} g_{\gamma^+}(t) + e^{-i\phi} g_{\gamma^-}(t)]. \quad (4.24)$$

Onde $\gamma^+ = [s, u, \xi]$ e $\gamma^- = [s, u, -\xi]$.

Seja agora a família de átomos janelados de *Fourier* $g_{(n,k)}$ e $(n, k) \in \mathbb{Z}^2$. Esta será obtida fazendo: $s = s_0$ constante, e $\gamma = (s_0, nu_0, k\xi_0)$; e podemos definir os átomos:

$$g_{(n,k)} = g(t - nu_0) e^{ik\xi_0 t}. \quad (4.25)$$

Daubechies [8] estudou as condições necessárias a $g(t)$, u_0 , e ξ_0 para que $g_{(n,k)}_{(n,k) \in \mathbb{Z}^2}$ formem um *frame* cujo resultado principal é:

Teorema 4.1 *A família janelada de Fourier $g_{(n,k)}_{(n,k) \in \mathbb{Z}^2}$ é um frame se e somente se:*

$$\frac{2\pi}{u_0 \xi_0} \geq 1, \quad (4.26)$$

e os limites do frame serão dados por:

$$A \leq \frac{2\pi}{u_0 \xi_0} \leq B. \quad (4.27)$$

A razão $\frac{2\pi}{u_0 \xi_0}$ nos fornece uma densidade de átomos janelados de *Fourier* no plano tempo-freqüência. Então estes são parâmetros que podem ser explorados no caso de quisermos especificar uma distribuição no plano tempo-freqüência. Porém, devemos lembrar do limite fornecido pela caixa de *Heisenberg* [2]. Como todos os átomos são normalizados, o segmento será uma base ortogonal somente se $A = B = 1$. Isto é possível somente na amostragem crítica de $u_0 \xi_0 = 2\pi$.

4.3.1 Átomos de *Gabor* Discretos

Assim como discretizamos os átomos gerados a partir da transformada janelada de *Fourier*, podemos discretizar os átomos de *Gabor*, tanto no deslocamento,

na modulação quanto na escala. Vejamos como realizar esta discretização e suas implicações.

Definimos os átomos de *Gabor* como as funções geradas a partir do deslocamento, modulação e escalamento de uma Gaussiana, ou seja:

$$\begin{aligned} g_\gamma(t) &= \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) e^{i\xi t}, \\ g(t) &= 2^{\frac{1}{4}} e^{-\pi t^2}. \end{aligned} \tag{4.28}$$

Onde $\gamma = [s, u, \xi] \in \mathbb{R}^2 \times \mathbb{R}$.

Porém realizar o *Matching Pursuits* com um dicionário de *Gabor* composto pelos átomos definidos acima é impraticável já que teríamos um número infinito de átomos. Podemos então discretizar o conjunto γ . Isto pode ser feito, levando em consideração o resultado sobre a discretização da transformada janelada de *Fourier*, da seguinte forma:

$$\gamma = [a^j, pa^j \Delta u, ka^{-j} \Delta \xi], \quad [j, p, k] \in \mathbb{Z}^3. \tag{4.29}$$

Este procedimento pode ser visto em [19]. Dependendo do fator a , a adequação deste dicionário, formado pelos átomos de *Gabor*, para representar sinais quando utilizado no algoritmo de *Matching Pursuits* pode ser provada. Repare na semelhança entre Δu e $\Delta \xi$ da equação (4.29) e u_0 e ξ_0 que definem a família de átomos da transformada janelada de *Fourier*. O que mais podemos perceber nesta discretização? Devemos estudar as condições de Δu e $\Delta \xi$ para que a partir dos átomos gerados formemos um *frame*.

4.3.2 *Frame* do Dicionário Discreto de *Gabor*

Quando utilizamos o dicionário de *Gabor* para representar sinais através do algoritmo de *Matching Pursuits* estamos realizando uma aproximação do sinal através de passos e não especificamente uma expansão num *frame* como no método dos *Frames* apresentado na seção 3.1.2.1. Porém ainda assim devemos verificar a validade de utilizar o dicionário de *Gabor* para aproximar sinais. Logicamente, se um dicionário formar um *Frame* então o mesmo poderá ser utilizado para representar sinais.

Introduzindo um fator de sobre-amostragem (l) [19] podemos ter os índices de γ dados por:

$$\Delta u = a^{1-l}, \quad \Delta \xi = \pi a^{2-l}. \quad (4.30)$$

Com o que:

$$\frac{2\pi}{\Delta u \Delta \xi} = \frac{2\pi}{a^{1-l} \pi a^{2-l}} = \frac{2}{a^{3-2l}}. \quad (4.31)$$

Então se a e l forem tais que $a^{3-2l} \leq 2$ teremos um *frame* gerado a partir desta amostragem. Isto nos garante poder representar um sinal no *frame* assim gerado. Fazendo $a = 2$ e $l = 1$ satisfazemos a condição de forma crítica e geramos um *frame* apertado. Teremos então que o parâmetro γ será dado por:

$$\gamma = [a^j, p a^{j+1-l}, k \pi a^{2-j-l}], \quad [j, p, k, l] \in \mathbb{Z}^4, \quad (4.32)$$

ou simplesmente:

$$\gamma = [2^j, p 2^j, k \pi 2^{1-j}], \quad [j, p, k] \in \mathbb{Z}^3. \quad (4.33)$$

Podemos representar γ através dos próprios fatores de j , p e k , e teremos o índice discreto:

$$\gamma_d = [j, p, k] \in \mathbb{Z}^3. \quad (4.34)$$

4.3.3 Átomos do Dicionário Discreto de *Gabor*

Estando de posse dos resultados anteriores podemos gerar o dicionário com a certeza de que ele satisfaz as condições necessárias para gerar um *frame*, podendo desta forma ser utilizado para decompor sinais. Vejamos como gerar o dicionário, ou os átomos.

O índice j está relacionado às escalas como mencionado. Porém além da escala devemos considerar o deslocamento, dependente de p , assim como a modulação nas escalas dependente de k , ver equação (4.29). Então teremos para um sinal de comprimento N , e sendo $L = \log_2(N)$, a variação de índices como a seguir:

$$j \in [0, L], \quad (4.35)$$

$$p \in [0, N 2^{l-1-j}), \quad (4.36)$$

$$k \in [0, 2^{l-1+j}). \quad (4.37)$$

Repare que tanto o deslocamento quanto a modulação dependem da escala considerada.

Se além disso amostrarmos os átomos, podemos gerar o dicionário de *Gabor* a partir dos átomos discretos $g_{\gamma_d}(n)$:

$$\begin{aligned}
g_{\gamma_d}(n) &= g_j(n - p2^{j+1-l})e^{ink\pi 2^{2-j-l}}, \\
g_j(n) &= \begin{cases} \delta(n) & , \quad \text{se } j = 0 \\ K_{\gamma_d} g\left(\frac{n}{2^j}\right) & , \quad \text{se } j \in (0, L) \\ \frac{1}{\sqrt{N}} & , \quad \text{se } j = L \end{cases} \quad (4.38) \\
g(n) &= 2^{\frac{1}{4}} e^{-\pi n^2}.
\end{aligned}$$

Esta amostragem, ou discretização, obedecerá a taxa de *Nyquist*, já que realizamos uma amostragem diádica ($a = 2$), ver equação (4.33).

É interessante notar qual a natureza dos vetores que compõem o dicionário em questão. Teremos no dicionário vetores nos eixos do espaço, isto é, impulsos nos diferentes instantes de tempo, no caso de $j = 0$ com a variação de p . No caso de $j = L$ teremos vetores formados por exponenciais complexas que são a base de *Fourier*. E nos casos de $j \neq 0$ e $j \neq L$ teremos uma interpolação entre os impulsos e as exponenciais complexas realizada pela Gaussiana.

4.3.4 Átomos Reais

No caso de desejarmos átomos reais devemos fazer algumas modificações, como no caso de átomos contínuos apresentados nas equações (4.23) e (4.24). Neste caso devemos passar a considerar o fator fase, ϕ . Com isto geraremos os átomos do dicionário através de:

$$\begin{aligned}
g_{(\gamma_d, \phi)}(n) &= g_j(n - p2^{j+1-l}) \cos(nk\pi 2^{2-j-l} + \phi), \\
g_j(n) &= \begin{cases} \delta(n) & , \quad \text{se } j = 0 \\ K_{(\gamma_d, \phi)} g\left(\frac{n}{2^j}\right) & , \quad \text{se } j \in (0, L) \\ \frac{1}{\sqrt{N}} & , \quad \text{se } j = L \end{cases} \quad (4.39) \\
g(n) &= 2^{\frac{1}{4}} e^{-\pi n^2}.
\end{aligned}$$

Normalmente o que fazemos com a fase é amostrá-la [7]. Então podemos considerar o parâmetro de fase entre $[0, \pi)$ podendo assumir M valores distintos e teremos M

possíveis fases dadas por:

$$\phi_m = m \times \frac{\pi}{M}, \quad (4.40)$$

com $m \in [0, M)$. Podemos imaginar que m faz parte do índice γ_d e teremos para γ_d :

$$\gamma_d = [j, p, k, m] \in \mathbb{Z}^4. \quad (4.41)$$

E o átomo dado por:

$$g_{\gamma_d}(n) = g_j(n - p2^{j+1-l}) \cos \left(nk\pi 2^{2-j-l} + \frac{m\pi}{M} \right),$$

$$g_j(n) = \begin{cases} \delta(n) & , \quad \text{se } j = 0 \\ K_{\gamma_d} g \left(\frac{n}{2^j} \right) & , \quad \text{se } j \in (0, L) \\ \frac{1}{\sqrt{N}} & , \quad \text{se } j = L \end{cases} \quad (4.42)$$

$$g(n) = 2^{\frac{1}{4}} e^{-\pi n^2}.$$

É interessante notar a importância da fase no caso da escolha de um átomo para aproximar um sinal. Por exemplo, dois sinais podem ser exatamente os mesmos, ocorrendo no mesmo instante, porém com defasagem de 90° . Isto acarretará um produto interno nulo entre os dois. Então, apesar de serem iguais do ponto de vista comportamental ou fenomenológico não conseguiremos identificar esta semelhança. Logo devemos permitir variações de fase de forma a podermos identificar estas diferenças.

4.4 *Matching Pursuits* com o Dicionário de *Gabor*

O dicionário formado pelos átomos discretos vistos na seção anterior pode ser utilizado no algoritmo de *Matching Pursuits* para decompor sinais. A cada passo do algoritmo de *Matching Pursuits* escolheremos então um elemento do dicionário que maximize o produto interno dele com o sinal de entrada. A partir deste procedimento, que foi visto na seção 3.2, montaremos o *Structure Book* que indicará as estruturas do dicionário presentes no sinal. Cada entrada do *Structure Book* deverá comportar o produto interno entre o sinal no passo e o átomo escolhido do dicionário, bem como o átomo escolhido. Teremos então que o elemento do *Structure Book*

no passo n será dado por:

$$s_n = [\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle, \gamma_n]. \quad (4.43)$$

A nomenclatura seguida é a do capítulo 3 e γ_n são os valores de $[j, p, k, m]$.

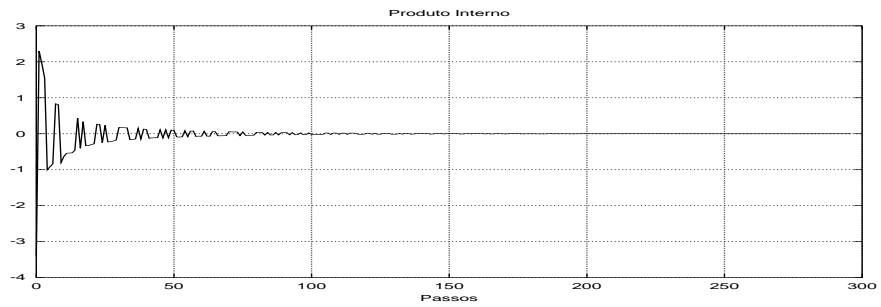
Para o sinal “x001” apresentado na figura A.1, do apêndice A, vejamos o comportamento do *Matching Pursuits* com o dicionário de *Gabor* permitindo diferentes quantidades de variações de fases: 1, 2, 4, 8, 16, 32 e 64. Permitimos que o algoritmo realize 300 passos ou que o processo seja abortado quando a norma do resíduo for menor que 0,001.

Na figura 4.2 vemos a decomposição do sinal “x001” utilizando um dicionário de átomos reais com oito possíveis fases. Nos gráficos apresentados podemos ver o produto a cada passo, a taxa de aproximação no passo, o sinal reconstruído e o erro da reconstrução para as possíveis quantidades de fase permitidas. Na figura 4.3 vemos o resíduo em cada passo, o resíduo normalizado, o RMSE, ver equação (4.47), e a energia do ruído pela energia do sinal.

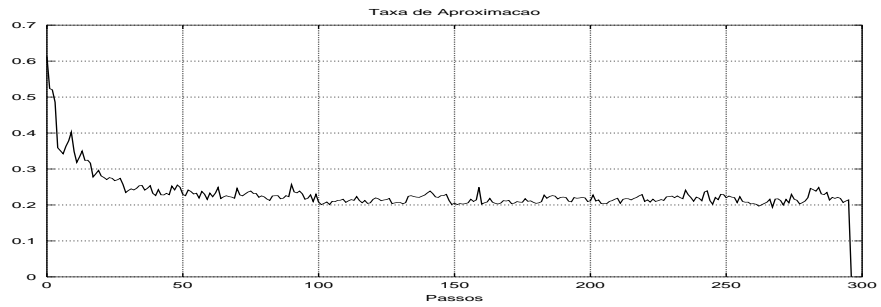
As informações destes gráficos indicam o fato de que podemos utilizar o método de *Matching Pursuits* com o dicionário de *Gabor* para decompor os sinais. Os resultados para diversas quantidades de fases (1,2,4,8,16,32,64) estão resumidos na tabela 4.1. Nesta podemos ver o passo final para a condição de resíduo menor ou igual a 0.001 quando utilizamos o dicionário com diferentes números de variações de fase na decomposição do sinal “x001”, apresentamos também o tamanho do dicionário, a redundância e o resíduo final para cada um dos casos.

Quant. Fases	Tam. do Dicionário	Redundância	Passo Final	Resíduo Final
1	1024	8	300	0.002425
2	1920	15	287	0.000986
4	3712	29	280	0.000979
8	7296	57	296	0.000993
16	14464	113	277	0.000978
32	28800	225	281	0.000986
64	57472	449	282	0.000980
Ótima	1024	8	283	0.000977

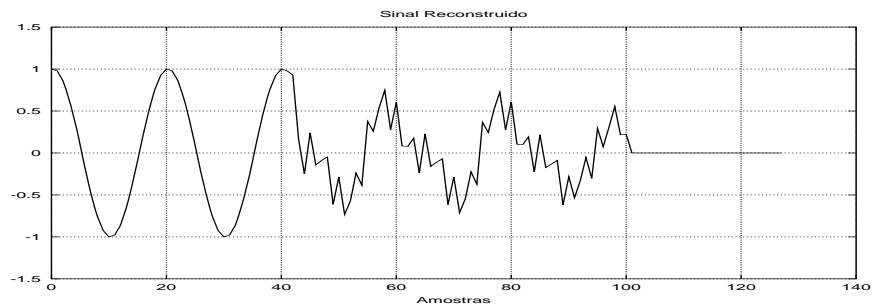
Tabela 4.1: Avaliação dos dicionários com diferentes quantidades de fases.



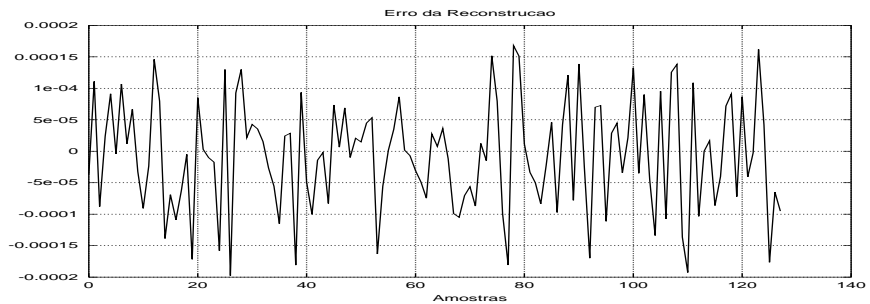
(a) Produto interno no passo



(b) Taxa de aproximação no passo

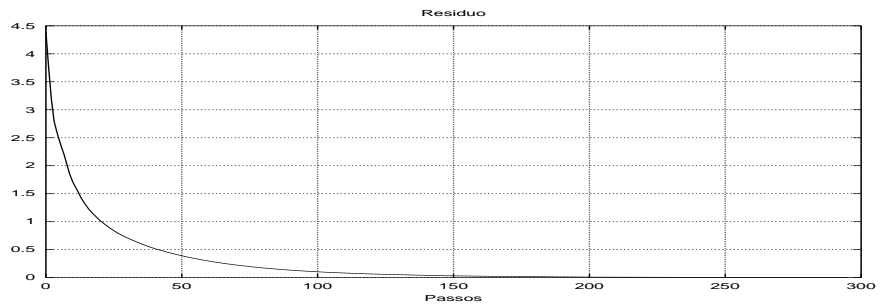


(c) Sinal Reconstruído

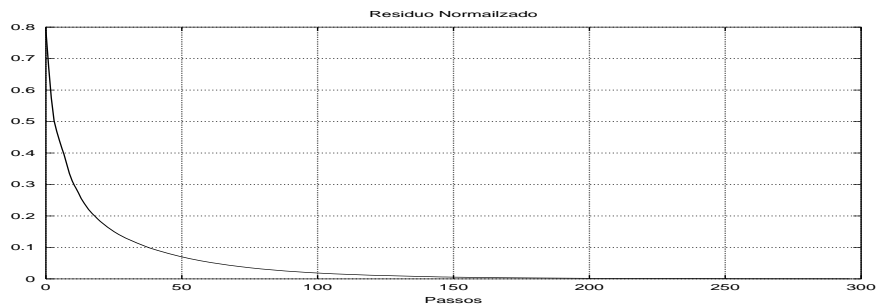


(d) Erro da reconstrução

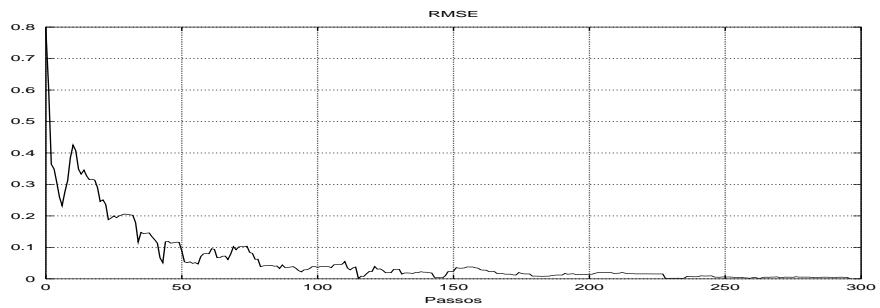
Figura 4.2: Resultados do *Matching Pursuit* com 8 fases possíveis, para o sinal x001.



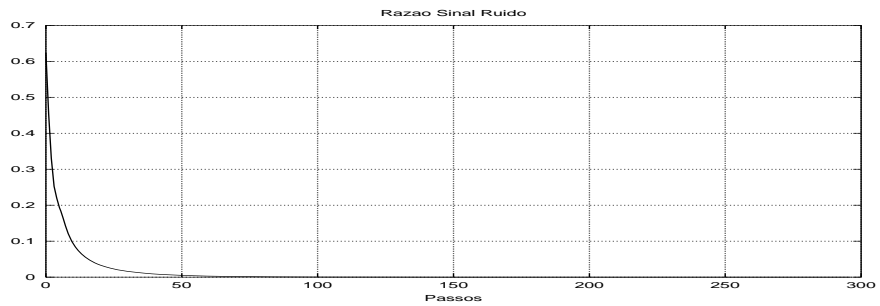
(a) Resíduo no passo



(b) Resíduo normalizado no passo



(c) RMSE



(d) Energia Do Ruído Pela Energia Do Sinal

Figura 4.3: Resultados do *Matching Pursuit* com 8 fases possíveis, para o sinal x001.

Podemos perceber uma pequena influência do número de fases permitidas para o átomo no processo de aproximação, ou convergência, do *Matching Pursuits*. Estas influências podem ser melhor observadas quando agrupamos as diferentes medidas de desempenho: o resíduo normalizado a cada passo, o RMSE, equação (4.47), o RNS, equação (4.48), e a taxa de aproximação no passo. Estes fatores de desempenho podem ser vistos nas figuras 4.4, 4.5, 4.6 e 4.7 respectivamente. Estas simulações foram geradas permitindo um número máximo de 300 passos, ou um resíduo de módulo maior que 0,001 por isso em algumas das curvas observam-se quedas bruscas que identificam quando a decomposição atingiu o segundo critério. As definições das medidas de desempenho mencionadas podem ser vistas a seguir. Nas definições a seguir, denomina-se por q o passo corrente. A taxa de aproximação no passo é definida pela equação (3.26), que repetimos:

$$\lambda(q) = \frac{\| \langle \mathbf{x}, \mathbf{g}_{\gamma_q} \rangle \|}{\| \mathbf{R}_x^q \|}, \quad (4.44)$$

onde γ_q é o índice para o átomo utilizado no passo q do algoritmo de *Matching Pursuits*. O Resíduo normalizado será dado por:

$$\text{Rn}(q) = \frac{\| \mathbf{R}_x^q \|}{\| \mathbf{x} \|}. \quad (4.45)$$

É interessante observar que esta é a mesma medida que o erro máximo relativo, muito utilizado na análise de falhas de potência [19], que é dado por:

$$\epsilon_\infty(q) = \frac{\| \mathbf{x} - \tilde{\mathbf{x}}_q \|_\infty}{\| \mathbf{x} \|_\infty}. \quad (4.46)$$

Além disso consideramos o RMSE (*Root Mean Squared Error*), que é definido como:

$$\text{RMSE}(q) = \sqrt{\frac{\sum_n |x(n) - \tilde{x}_q(n)|^2}{\sum_n |x(n)|^2}}. \quad (4.47)$$

E a energia do ruído pela energia do sinal será dada pela equação (3.35) aqui repetida:

$$\text{RNS}(q) = \frac{\| \mathbf{x} - \tilde{\mathbf{x}}_q \|^2}{\| \mathbf{x} \|^2}. \quad (4.48)$$

Nas equações acima o sinal aproximado no passo q , $\tilde{\mathbf{x}}_q$ é dado por:

$$\tilde{\mathbf{x}}_q = \sum_{n=0}^q \langle \mathbf{R}_x^n, \mathbf{g}_{\gamma_n} \rangle \mathbf{g}_{\gamma_n}. \quad (4.49)$$

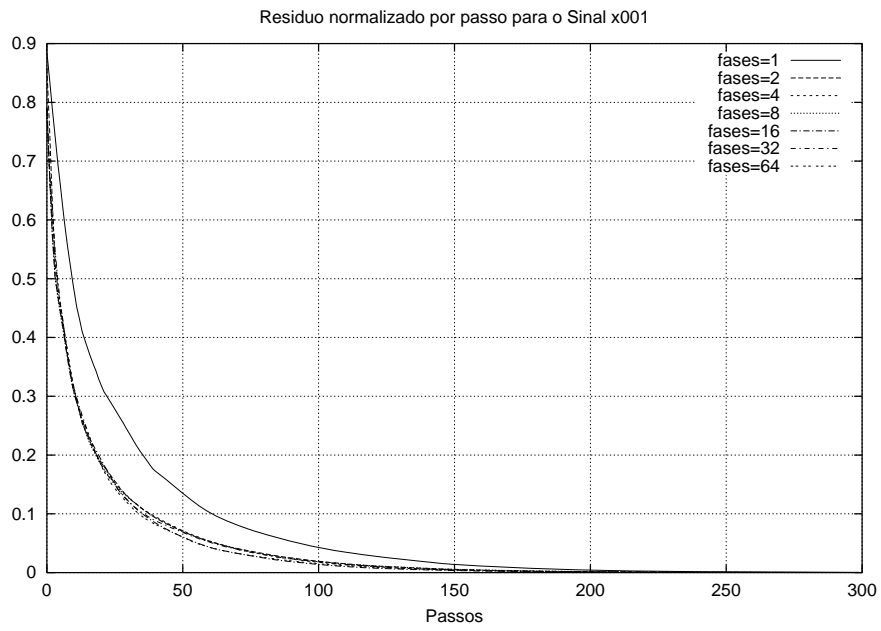


Figura 4.4: Resíduo normalizado no passo para o *Matching Pursuits* com o dicionário de *Gabor* com diferentes fases para o sinal x001.

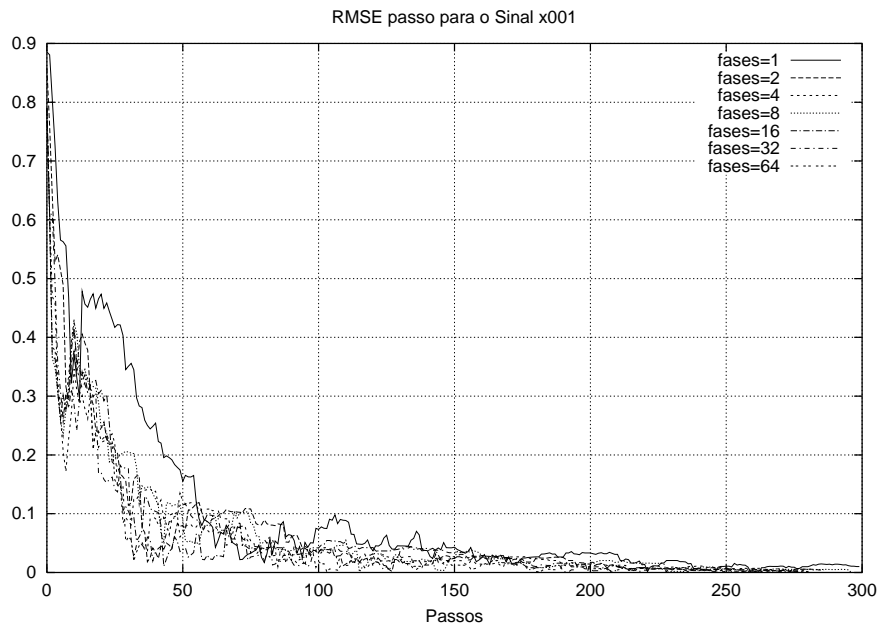


Figura 4.5: RMSE no passo para o *Matching Pursuits* com o dicionário de *Gabor* com diferentes fases para o sinal x001.

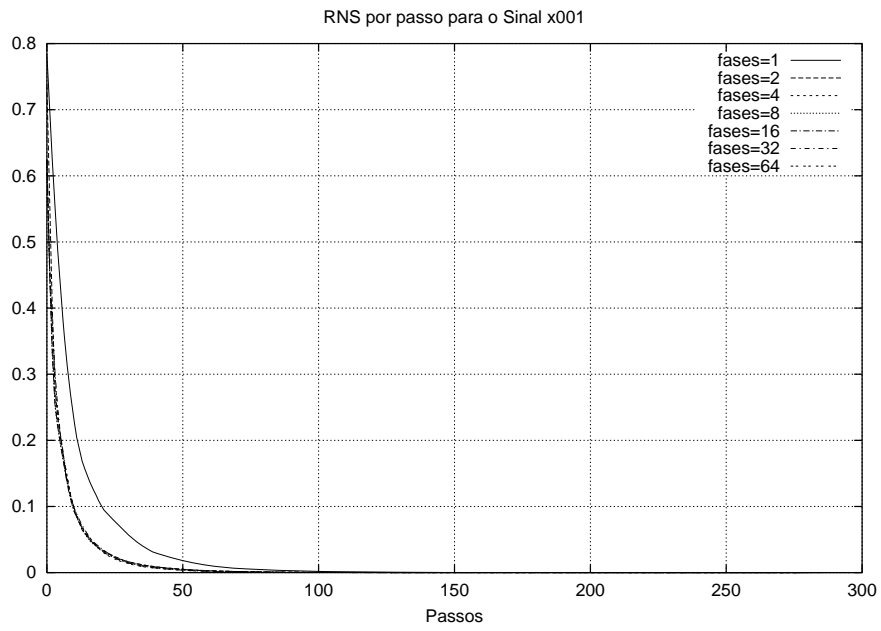


Figura 4.6: Energia do ruído pela energia do sinal no passo para o *Matching Pursuits* com o dicionário de *Gabor* com diferentes fases para o sinal x001.

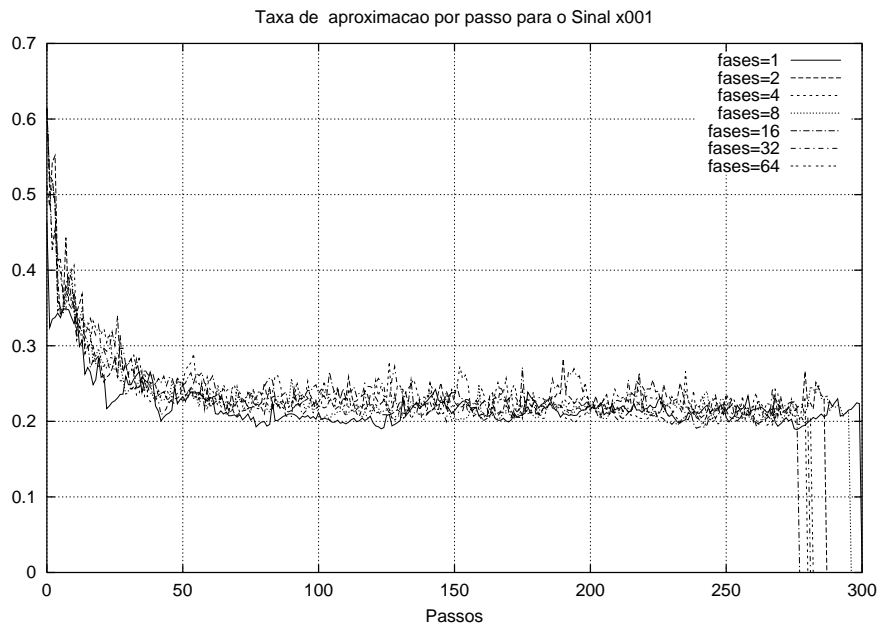


Figura 4.7: Taxa de aproximação no passo para o *Matching Pursuits* com o dicionário de *Gabor* com diferentes fases para o sinal x001.

Podemos perceber que os casos com mais de uma fase possível para os átomos se destacam em relação à utilização de uma única fase, já que necessitamos um número menor de passos para atingir um resíduo tão pequeno, em módulo, do que no caso de uma única fase, o que se observa também nos casos da energia do ruído pela energia do sinal ou do RMSE. Podemos ver que a taxa de aproximação obtida com mais fases para os átomos é maior que com uma, indicando que estamos aproximando melhor o sinal. No caso da energia do ruído pela energia do sinal vemos que esta cai mais rapidamente para diversas fases, assim como o resíduo normalizado e o RMSE.

Devemos ressaltar que no caso complexo, que não é o que está sendo utilizado, a influência do parâmetro fase é minimizada, já que projetamos o sinal real sobre a parte imaginária e a complexa que por definição terão 90° de defasagem. Na subseção a seguir veremos como aproveitar esta característica do dicionário complexo para escolher uma fase apropriada para o átomo a utilizar num passo específico da aproximação. Para isto utiliza-se a projeção em sub-espacos [7].

4.4.1 Achando a Fase do Átomo

Como vimos há uma influência da fase do átomo no processo de aproximação. Para resolver este problema achamos uma fase ótima para o átomo e com esta fase ótima geramos o átomo e calculamos a projeção. O método utilizado para encontrar a fase ótima foi desenvolvido por *Ferrando, Kolasa e Kovacevic* [27] e se baseia em realizar a projeção sobre o átomo complexo e a partir da projeção sobre a parte real e a imaginária encontrar a fase ótima para o átomo. Este procedimento é descrito a seguir.

Podemos definir um átomo real em função dos parâmetros: escalamento, deslocamento, modulação e fase. Teremos então um átomo real em função de $\gamma_\omega = [s, u, \xi, \omega]$, que será definido por:

$$g_{\gamma_\omega}(t) = \frac{\frac{1}{\sqrt{s}}g\left(\frac{t-u}{s}\right) \cos(\xi t + \omega)}{\left\| \frac{1}{\sqrt{s}}g\left(\frac{t-u}{s}\right) \cos(\xi t + \omega) \right\|}. \quad (4.50)$$

Com $\gamma = [s, u, \xi]$ podemos definir as funções:

$$P_\gamma(t) = g\left(\frac{t-u}{s}\right) \cos(\xi t + \omega), \quad Q_\gamma(t) = g\left(\frac{t-u}{s}\right) \sin(\xi t + \omega). \quad (4.51)$$

Podemos definir então:

$$g(\gamma, \omega) = \frac{P_\gamma \cos(\omega) - jQ_\gamma \sin(\omega)}{\|P_\gamma \cos(\omega) - jQ_\gamma \sin(\omega)\|} \quad (4.52)$$

Vimos então como decompor o átomo complexo em dois pedaços, um real e outro imaginário. Lembremo-nos que a cada passo do *Matching Pursuits* procuramos:

$$\max_{g(\gamma, \omega)} |\langle \mathbf{R}^n_{\mathbf{x}}, g(\gamma, \omega) \rangle|. \quad (4.53)$$

A partir da decomposição do átomo mostrada anteriormente podemos encontrar o ω ótimo para um γ e $\mathbf{R}^n_{\mathbf{x}}$ determinados através da proposição a seguir [27]:

Proposição 4.1 *Dados \mathbf{x} e γ temos:*

$$a = \langle \mathbf{x}, P_\gamma \rangle \quad b = \langle \mathbf{x}, Q_\gamma \rangle \quad (4.54)$$

$$a_1 = a \|Q_\gamma\|^2 - b \langle P_\gamma, Q_\gamma \rangle \quad b_1 = b \|P_\gamma\|^2 - a \langle P_\gamma, Q_\gamma \rangle \quad (4.55)$$

Então a fase ótima ω_o , que maximiza $|\langle \mathbf{R}^n_{\mathbf{x}}, g(\gamma, \omega) \rangle|$ será dada nos casos:

1. Se $\xi \neq 0$ e $a_1 \neq 0$, por:

$$\tan \omega_o = -\frac{b_1}{a_1}; \quad (4.56)$$

2. Se $\xi = 0$, por:

$$\omega_o = 0; \quad (4.57)$$

3. Se $a_1 = 0$, por:

$$\omega_o = \frac{\pi}{2}. \quad (4.58)$$

A prova pode ser obtida maximizando a função $h(f) = \langle f, g(\gamma, \omega) \rangle^2$. Teremos na verdade a função:

$$h(x) = \left(\frac{(a - bx)}{\|P - Qx\|} \right)^2 = \frac{(a - bx)^2}{\|P\|^2 + \|Q\|^2 x^2 - 2\langle P, Q \rangle x}, \quad (4.59)$$

onde $x = \tan \omega$. Derivando esta função teremos:

$$\frac{dh(x)}{dx} = \frac{(\|P\|^2 + \|Q\|^2 x^2 - 2\langle P, Q \rangle x)(2b^2 x - 2ab) - (a - bx)(2x \|Q\|^2 - 2\langle P, Q \rangle)}{(\|P\|^2 + \|Q\|^2 x^2 - 2\langle P, Q \rangle x)^2}, \quad (4.60)$$

igualando a zero acharemos o ponto de máximo:

$$0 = 2bx^2(a \|Q\|^2 - b \langle P, Q \rangle) + 2x(b^2 \|P\|^2 - a^2 \|Q\|^2) + 2a(-b \|P\|^2 - a \langle P, Q \rangle), \quad (4.61)$$

mas:

$$a \| Q \|^2 - b \langle P, Q \rangle = a_1, \quad (4.62)$$

$$-b \| P \|^2 - a \langle P, Q \rangle = -b_1, \quad (4.63)$$

$$b^2 \| P \|^2 - a^2 \| Q \|^2 = b_1 b - a_1 a. \quad (4.64)$$

Com isto teremos:

$$0 = 2bx^2a_1 + 2x(b_1b - a_1a) - 2ab_1 \rightarrow x = \frac{-b_1}{a_1}. \quad (4.65)$$

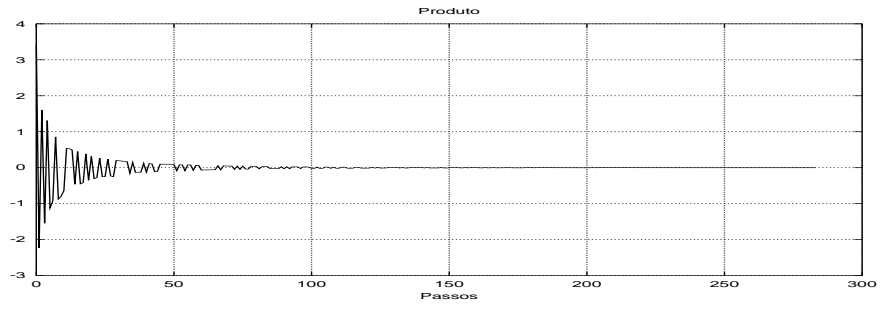
E como $x = \tan \omega$ temos que $\omega = \arctan x$.

É fácil notar cada um dos casos da proposição 4.1: no segundo como não há modulação não há influência da fase; no terceiro quando $a_1 = 0$ teríamos projeção somente sobre a parte imaginária do átomo complexo, então ao utilizarmos o átomo real fazemos a fase de 90° ou $\pi/2$; e no primeiro caso teremos o arco tangente das projeções sobre a parte real e imaginária do átomo complexo.

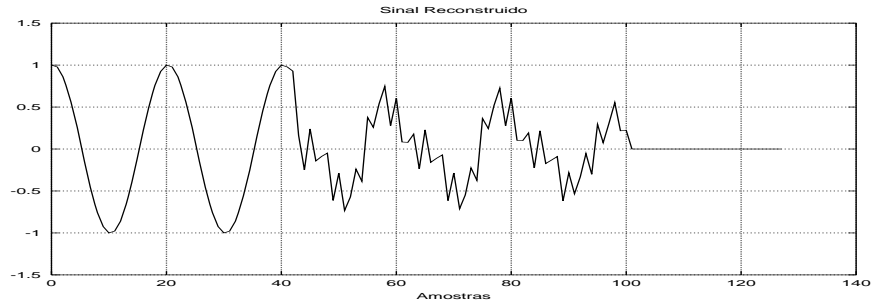
Os mesmos resultados obtidos nos casos anteriores, com diversas fases, para o sinal “x001” podem ser agora obtidos utilizando a fase ótima. Porém a complexidade computacional do algoritmo é aumentada. A cada passo necessitamos realizar 2 produtos internos por elemento do dicionário, porém o tamanho do dicionário permanece igual ao do caso de fase única, reduzindo o número total de produtos internos em comparação aos casos com diversas fases possíveis.

Na figura 4.8 podemos ver os resultados obtidos quando realizamos a decomposição do sinal “x001” utilizando a fase ótima no *Matching Pursuits*. Vemos que o mesmo se comporta bem além do fato de termos um algoritmo independente sem necessitar especificar o número de fases permitidas. Os resultados obtidos para este processo similares aos da tabela 4.1 podem ser vistos na mesma na linha onde o número de fase é dado por **ótima**. Vemos o bom desempenho deste procedimento. A redundância do dicionário não é especificamente 8 já que a fase pode ser qualquer o que levaria a um número infinito como na otimização através do método de *Newton* realizada em γ_d por *Mallat* e *Zhang* em seu trabalho original [1].

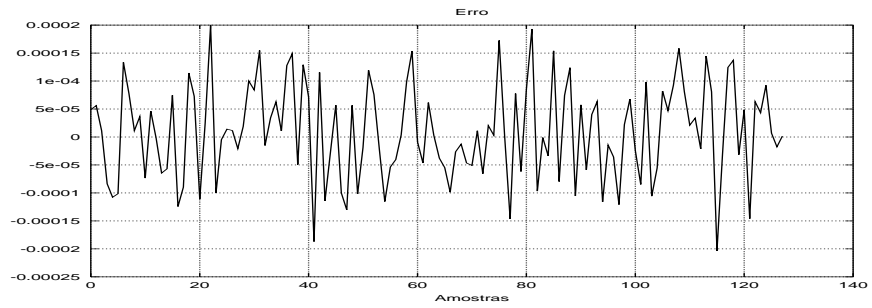
Este algoritmo foi aplicado a uma variedade de sinais de testes. São aqueles apresentados na seção A.1. Para estes sinais podemos ver as suas decomposições nas figuras 4.8 para o sinal “x001” e nas figuras 4.9 à 4.15 para os outros sinais. Apresentamos também as figuras de mérito do algoritmo aplicado aos diversos sinais,



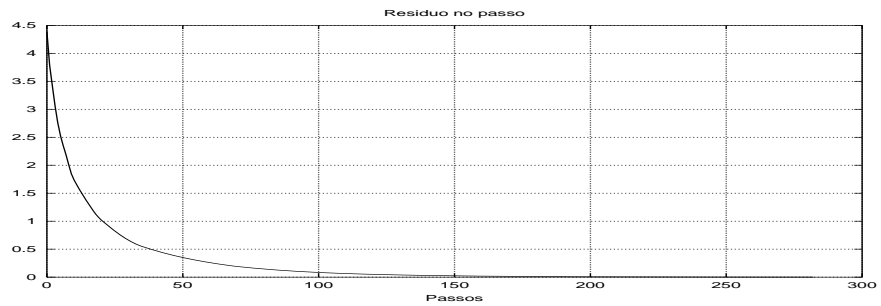
(a) Produto interno no passo



(b) Sinal reconstruído



(c) Erro da reconstrução



(d) Resíduo no passo

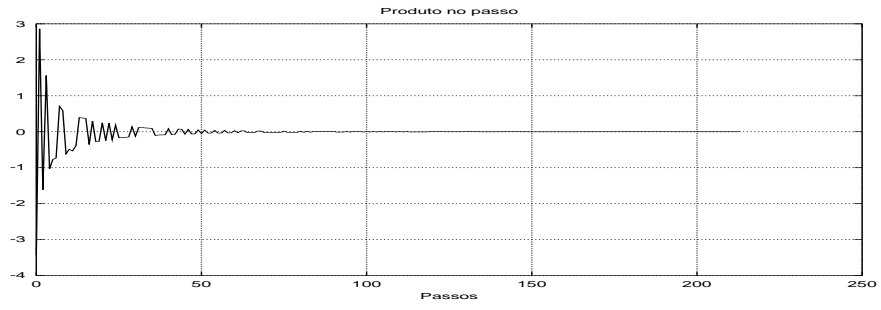
Figura 4.8: Resultados do *Matching Pursuits* com fase ótima, para o sinal x001.

estas são: o RMSE, o RNS, a taxa de aproximação no passo, o resíduo normalizado no passo e a taxa de aproximação média até o passo. Estas podem ser vistas nas figuras 4.16 a 4.20 onde estão agrupados os resultados para os diversos sinais. Em alguns destes gráficos, taxa de aproximação e taxa de aproximação média (figuras 4.18 e 4.20), observam-se quedas bruscas, estas indicam o passo final do processo de decomposição. O processo de decomposição nestes casos foi realizado como na seção 4.4, fixando um número de passos máximo de 300, e o módulo do resíduo como critério de parada em 0,001. Vemos através destes resultados que o desempenho quando utilizada a fase ótima é igual ou superior à quando se amostra a fase, fornecendo uma metodologia automatizada para realizar a decomposição em átomos reais, resolvendo o problema da fase e obtendo o mesmo desempenho ou melhor.

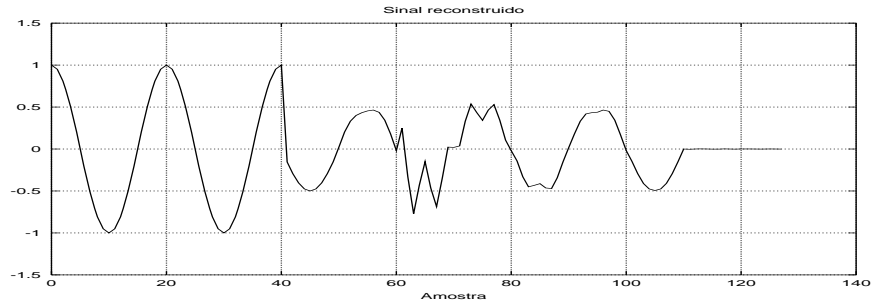
O aumento do custo computacional quando utilizamos o processo de cálculo da fase ótima, além de previsível, é justificável. Se M for o número de elementos do dicionário necessitaremos realizar $2M$ produtos internos para achar o elemento do dicionário que melhor aproxima o sinal no passo, M referentes à parte real do átomo, M referentes à parte complexa, e teremos $2M$ produtos internos a cada passo. No caso de discretizarmos a fase, com 2 possíveis fases já necessitamos realizar aproximadamente $2M$ produtos. Como no caso dos impulsos a fase não faz diferença, na verdade, teremos $2(M - N) + N$ produtos internos para duas fases, e no caso de 4 fases serão $4(M - N) + N$. Então devido à melhora obtida na decomposição vale a pena utilizar o expediente de cálculo da fase ótima ao invés de amostrar a fase no processo de decomposição, já que no caso com duas fases teremos $2M - N$ produtos, no caso de 4 fases $4M - 3N$ e neste caso este número de produtos é maior que o do caso de fase ótima $4M - 3N > 2M$ já que M é N^{L+1} que pode ser obtido a partir da variação dos índices j , p e k definidos na equação (4.37).

4.5 Representação Tempo-Freqüência com o *Matching Pursuits* e o Dicionário de Gabor

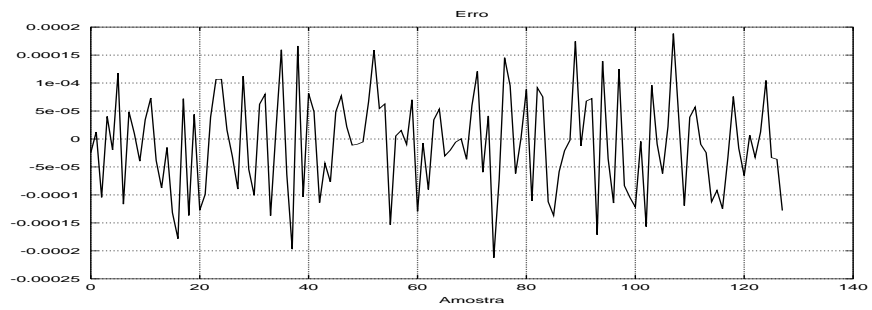
Na seção 3.2 apresentamos o algoritmo de *Matching Pursuits* e um dicionário formado a partir dos átomos tempo-freqüência de *Gabor* na seção 4.4, que utilizamos neste algoritmo. Nesta seção veremos como a partir deste dicionário podemos rea-



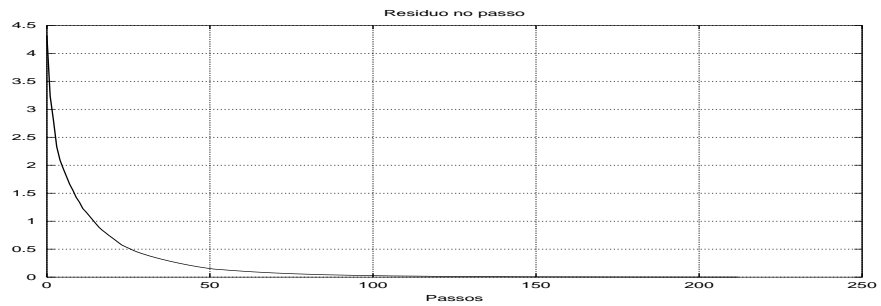
(a) Produto interno no passo



(b) Sinal reconstruído

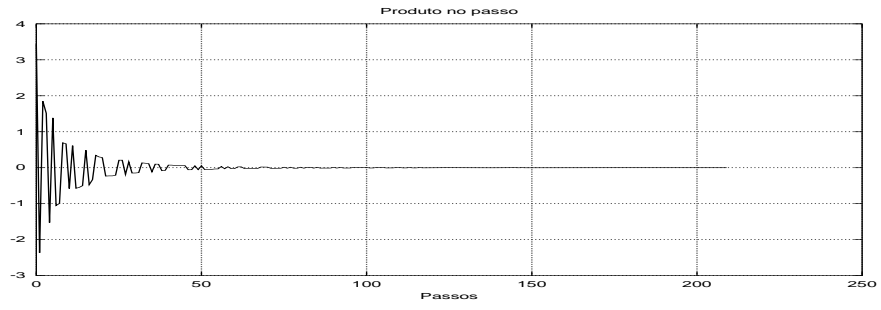


(c) Erro da reconstrução

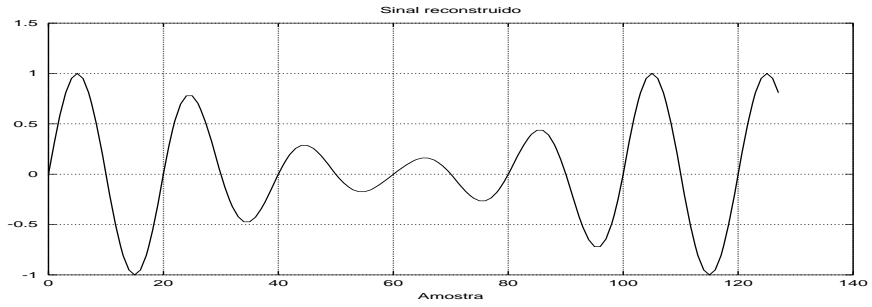


(d) Resíduo no passo

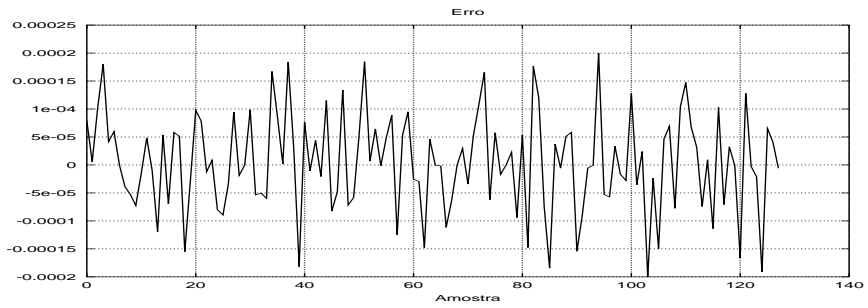
Figura 4.9: Resultados do *Matching Pursuits* com fase ótima, para o sinal x002.



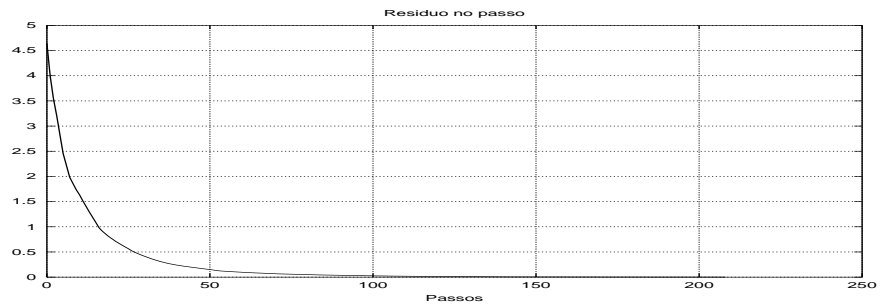
(a) Produto interno no passo



(b) Sinal reconstruído

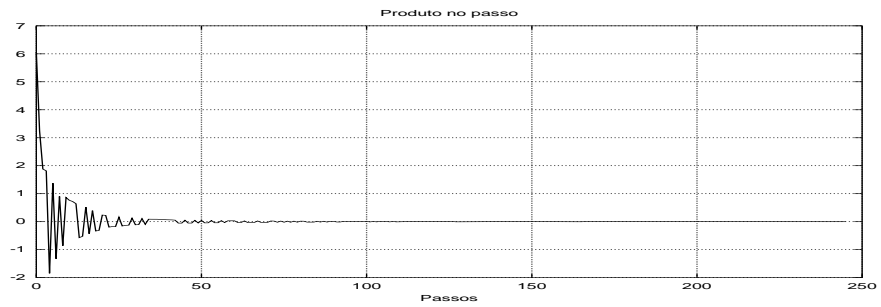


(c) Erro da reconstrução

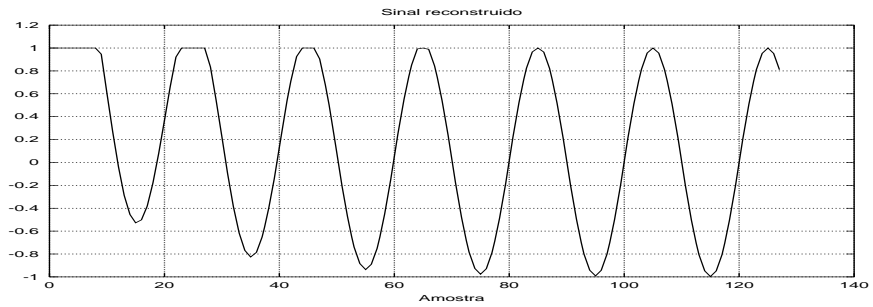


(d) Resíduo no passo

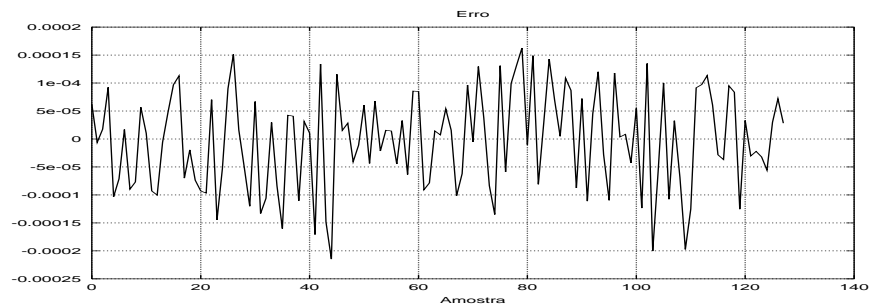
Figura 4.10: Resultados do *Matching Pursuits* com fase ótima, para o sinal x003.



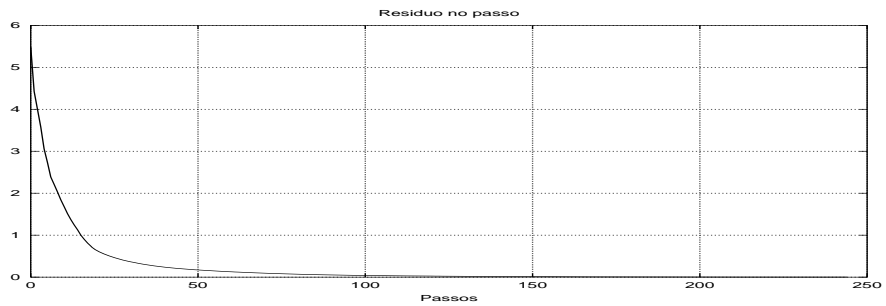
(a) Produto interno no passo



(b) Sinal reconstruído

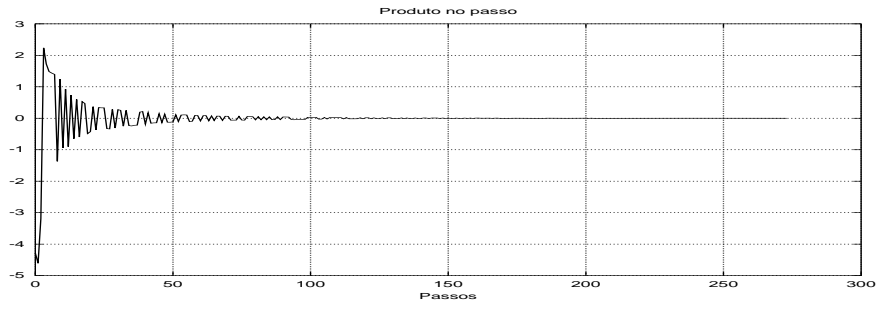


(c) Erro da reconstrução

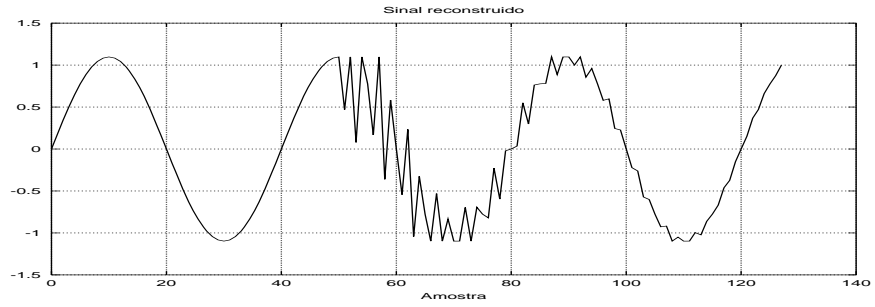


(d) Resíduo no passo

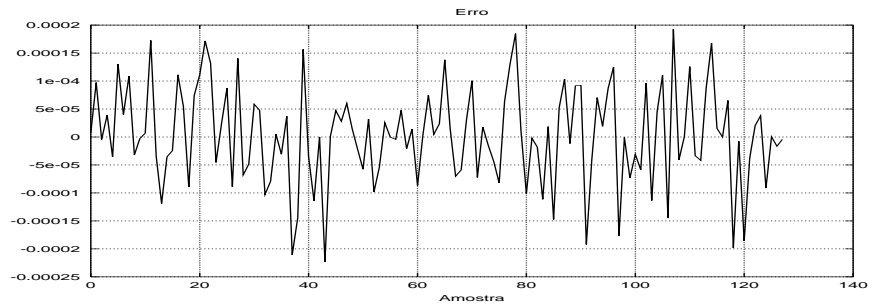
Figura 4.11: Resultados do *Matching Pursuits* com fase ótima, para o sinal x004.



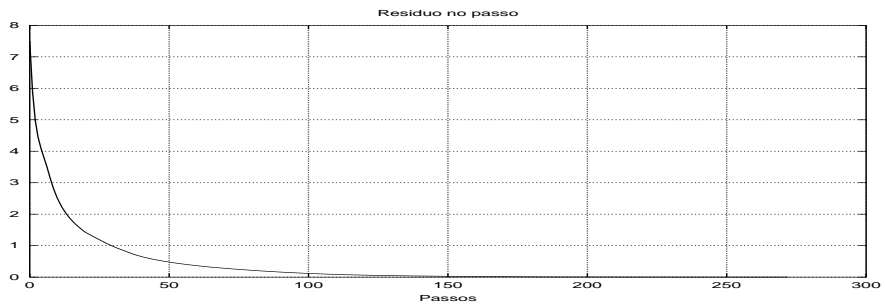
(a) Produto interno no passo



(b) Sinal reconstruído

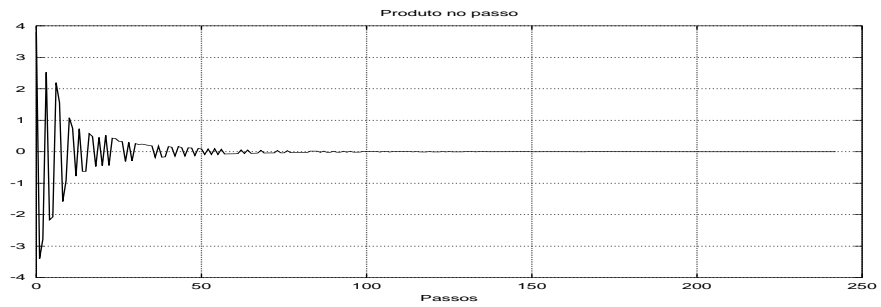


(c) Erro da reconstrução

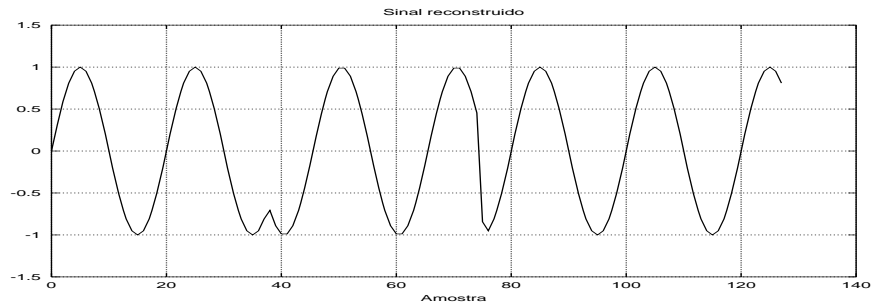


(d) Resíduo no passo

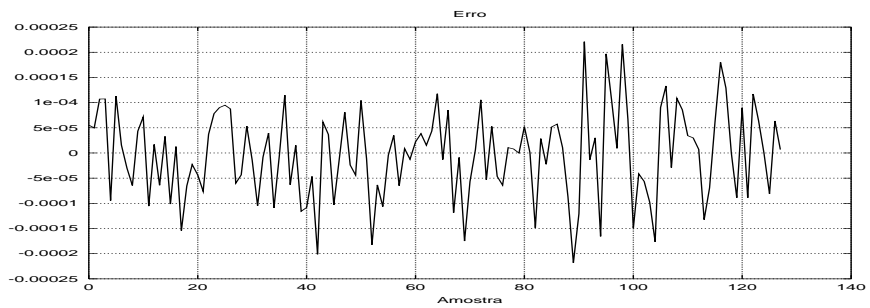
Figura 4.12: Resultados do *Matching Pursuits* com fase ótima, para o sinal x005.



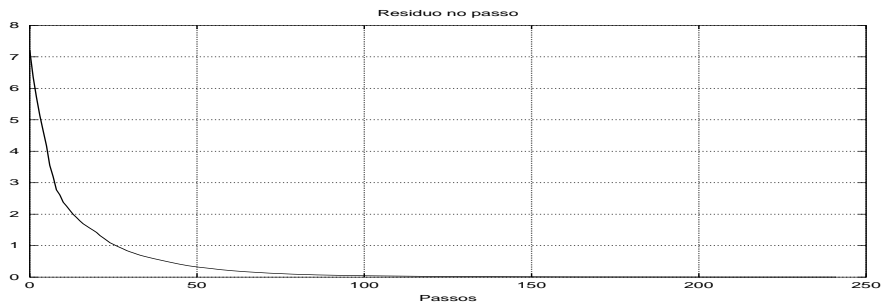
(a) Produto interno no passo



(b) Sinal reconstruído

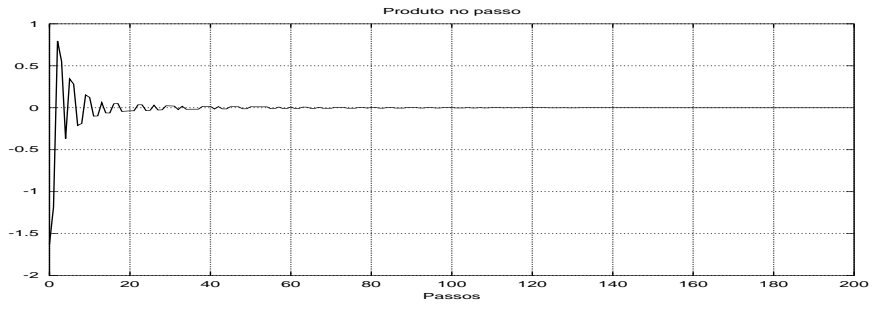


(c) Erro da reconstrução

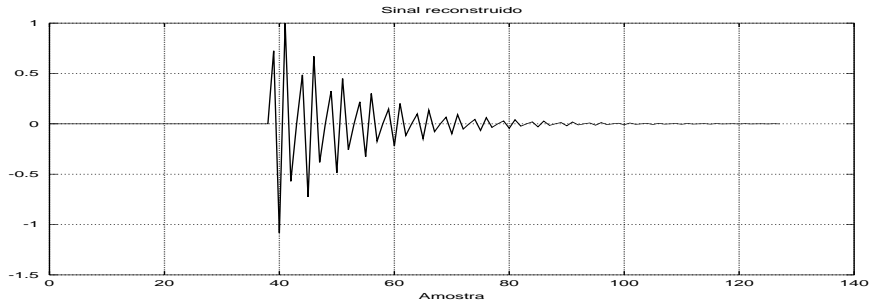


(d) Resíduo no passo

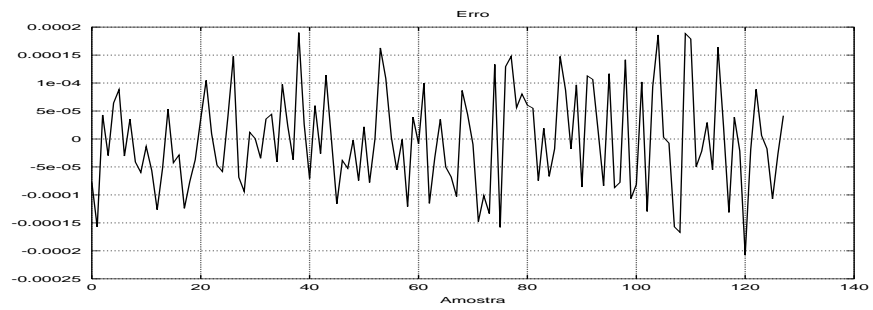
Figura 4.13: Resultados do *Matching Pursuits* com fase ótima, para o sinal x006.



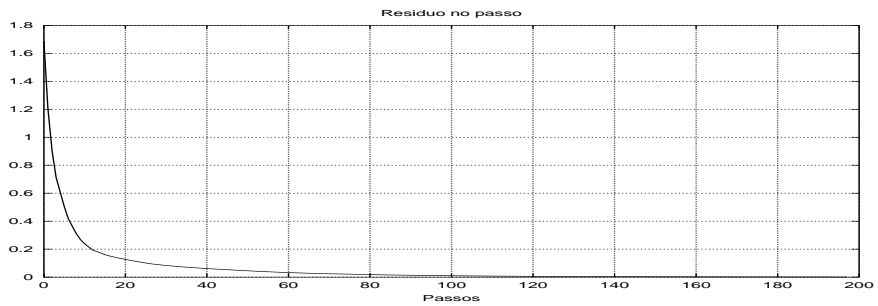
(a) Produto interno no passo



(b) Sinal reconstruído

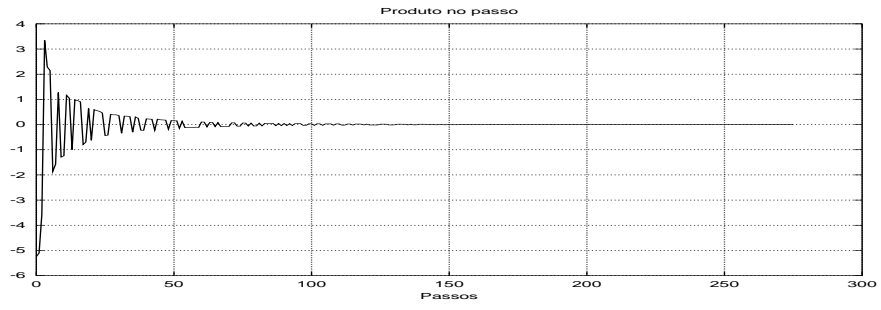


(c) Erro da reconstrução

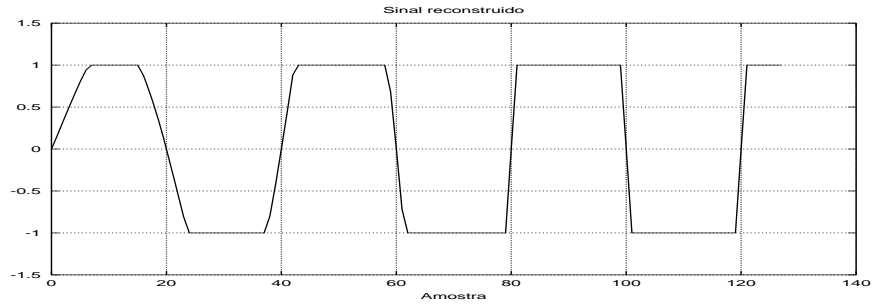


(d) Resíduo no passo

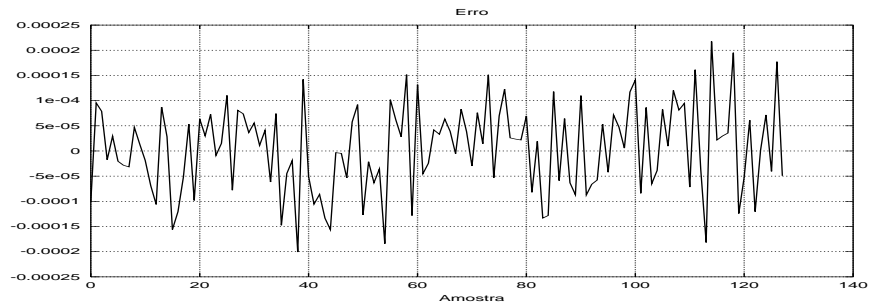
Figura 4.14: Resultados do *Matching Pursuits* com fase ótima, para o sinal x007.



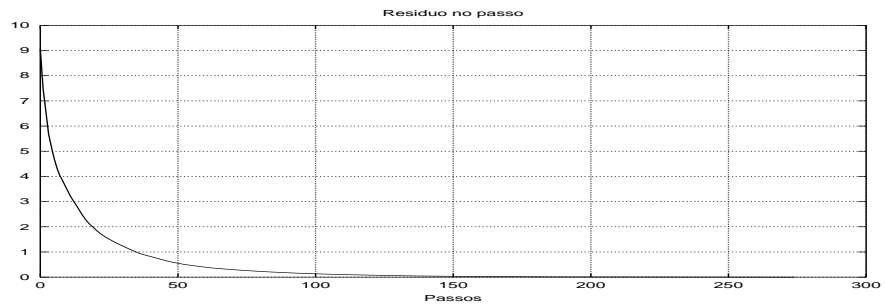
(a) Produto interno no passo



(b) Sinal reconstruído



(c) Erro da reconstrução



(d) Resíduo no passo

Figura 4.15: Resultados do *Matching Pursuits* com fase ótima, para o sinal x008.

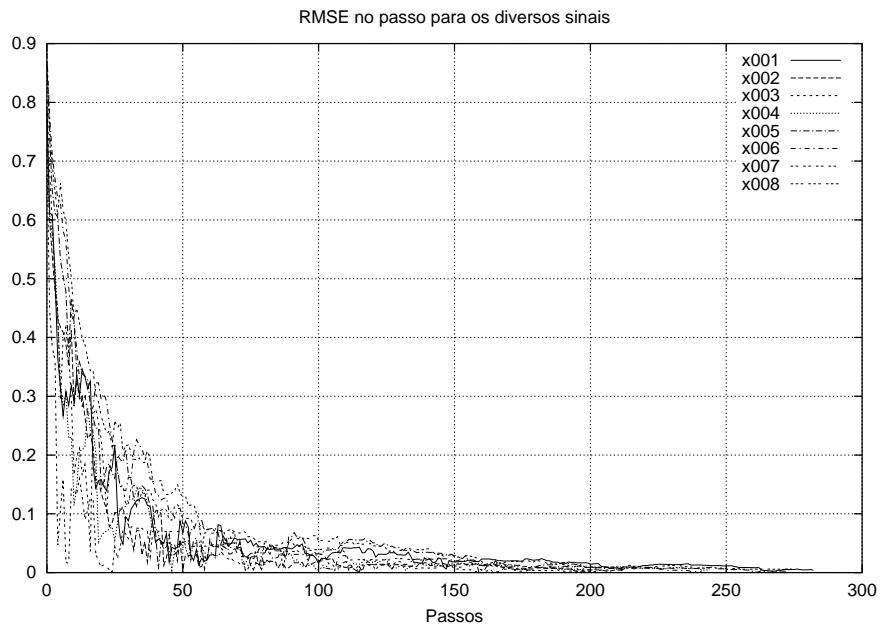


Figura 4.16: RMSE no passo corrente para o *Matching Pursuits* com o dicionário de *Gabor* e fase ótima para diferentes sinais.

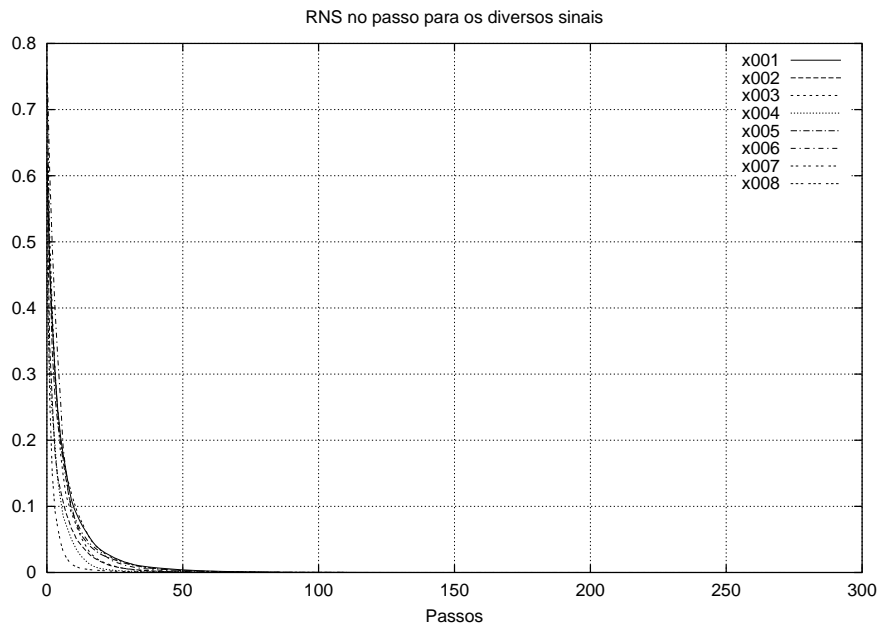


Figura 4.17: RNS no passo corrente para o *Matching Pursuits* com o dicionário de *Gabor* e fase ótima para diferentes sinais.

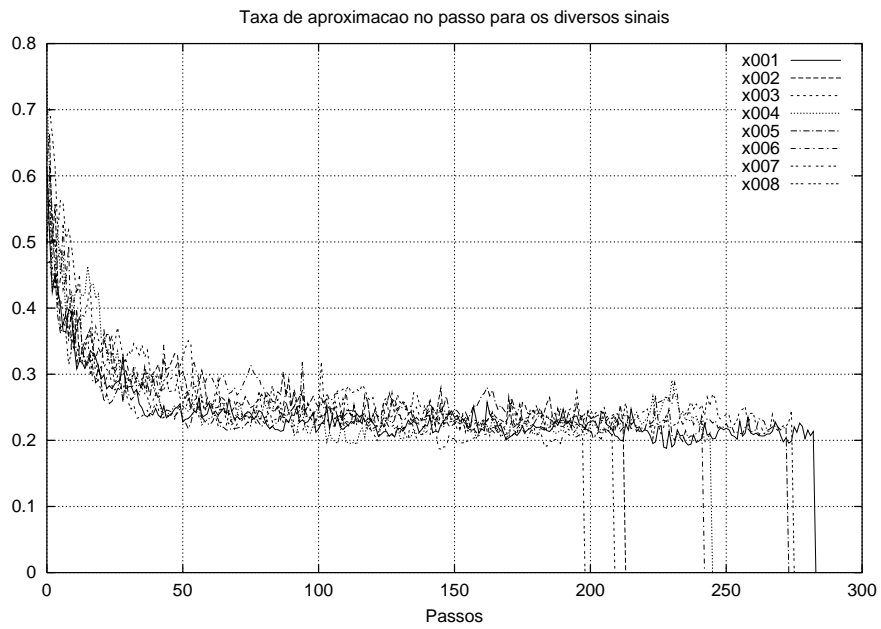


Figura 4.18: Taxa de aproximação no passo corrente para o *Matching Pursuits* com o dicionário de *Gabor* e fase ótima para diferentes sinais.

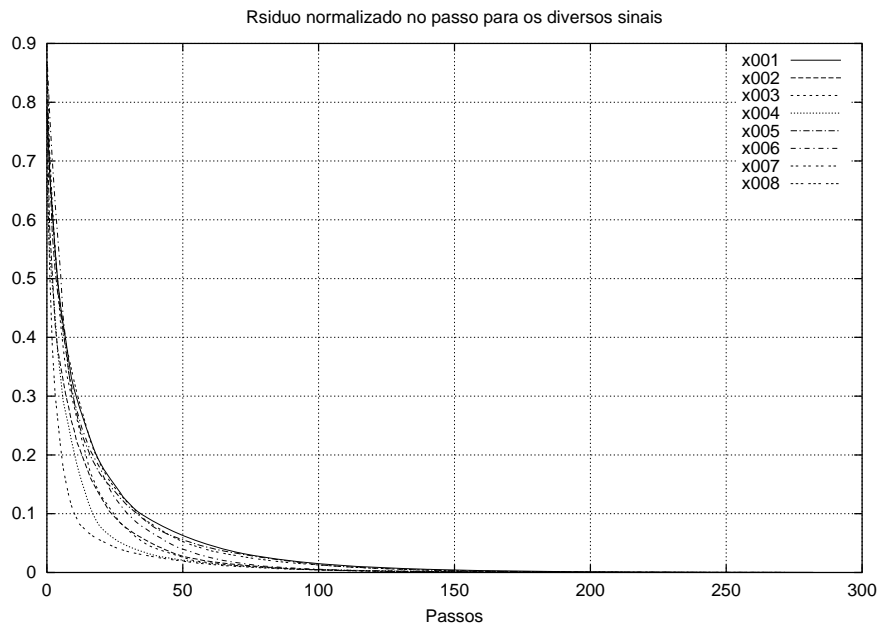


Figura 4.19: Resíduo normalizado após o passo para o *Matching Pursuits* com o dicionário de *Gabor* e fase ótima para diferentes sinais.

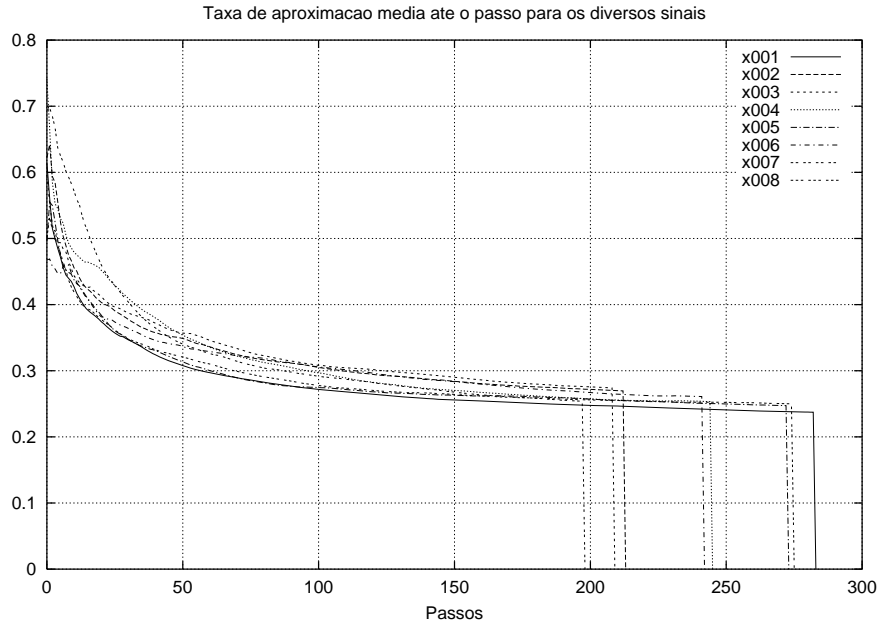


Figura 4.20: Taxa de aproximação média até o passo para o *Matching Pursuits* com o dicionário de *Gabor* e fase ótima para diferentes sinais.

lizar um estudo do sinal no plano tempo-freqüência. Esta distribuição chamaremos de $Ef(t, \omega)$ e sua primeira formulação pode ser vista em [1, 2].

O dicionário apresentado na seção 4.3 é formado a partir da função contínua:

$$g_\gamma(t) = \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) e^{i\xi t}, \quad (4.66)$$

$$g(t) = \sqrt[4]{2} e^{-\pi t^2}.$$

Para uma função $g_\gamma(t)$ podemos calcular a distribuição de *Wigner-Ville* que será dada por:

$$WD_g(t, f) = \int_{\tau} g\left(t + \frac{\tau}{2}\right) g^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau. \quad (4.67)$$

A distribuição de *Wigner-Ville* possui algumas características interessantes dentre as quais podemos destacar:

1. É sempre real.
2. Satisfaz às propriedades de escalamento, e de deslocamento no tempo e na freqüência entre outras [2, 23]:

- (a) Deslocamento no tempo: $T_{x'} = T_x(t - t_0, f)$ para $x'(t) = x(t - t_0)$.

- (b) Deslocamento na frequência: $T_{x'} = T_x(t, f - f_0)$ para $x'(t) = x(t)e^{j2\pi f_0 t}$.
- (c) Marginal temporal: $\int_f T_x(t, f)df = p_x(t) = |x(t)|^2$.
- (d) Marginal na frequência: $\int_t T_x(t, f)dt = P_X(f) = |X(f)|^2$.
- (e) Escalamento tempo-freqüência: $T_{x'}(t, f) = T_x(at, \frac{f}{a})$ para $x'(t) = \sqrt{|a|}x(at)$ com $a \neq 0$.
- (f) Suporte finito no tempo: $T_x(t, f) = 0$ para t fora do intervalo $[t_1, t_2]$ se $x(t) = 0$ fora de $[t_1, t_2]$.
- (g) Suporte finito na frequência: $T_x(t, f) = 0$ para f fora do intervalo $[f_1, f_2]$ se $X(f) = 0$ fora de $[f_1, f_2]$.
- (h) Transformada de *Fourier*: $T_{x'}(t, f) = T_x(\frac{-f}{c}, ct)$ para $x'(t) = \sqrt{|c|}X(ct)$ com $c \neq 0$.

Porém, o princípio da superposição quadrática, sub-subseção 4.1.3.1, é válido nesta distribuição, o que acarreta o surgimento de termos de interferência. Estes termos de interferência impedem um estudo mais preciso das componentes do sinal. Utilizando o algoritmo de *Matching Pursuits* e os átomos de *Gabor* podemos obter uma representação tempo-freqüência do sinal que minimiza os termos de interferência. Isto é feito da seguinte forma: calculamos a distribuição de *Wigner-Ville* da janela Gaussiana, equação (4.28), que será obtida substituindo a equação (4.28) na (4.67). Isto nos dará a distribuição de *Wigner-Ville* da Gaussiana que será dada por:

$$WD_g(t, w) = 2e^{-2\pi[t^2 + (\frac{w}{2\pi})^2]}. \quad (4.68)$$

Devemos reparar que esta expressão define uma região elíptica para $WD_g(t, w)$.

Poderíamos obter a distribuição de *Wigner-Ville* de um sinal x através de sua decomposição no *Matching Pursuits*, equação (3.22), isto é:

$$WD_x(t, w) = WD \left\{ \sum_{n=0}^{n=+\infty} [\langle \mathbf{R}^n_x, \mathbf{g}_{\gamma_n} \rangle \mathbf{g}_{\gamma_n}] (t, w) \right\}. \quad (4.69)$$

Que será dada por [1]:

$$WD_x(t, w) = \sum_{n=0}^{n=+\infty} \langle \mathbf{R}^n_x, g_{\gamma_n} \rangle^2 WD_{g_{\gamma_n}}(t, w) + \sum_{n=0}^{n=+\infty} \sum_{m=0, m \neq n}^{m=+\infty} \langle \mathbf{R}^n_x, g_{\gamma_n} \rangle \langle \mathbf{R}^m_x, g_{\gamma_m} \rangle^* WD_{(g_{\gamma_n}, g_{\gamma_m})}(t, w). \quad (4.70)$$

onde $WD_{(\mathbf{g}_{\gamma_n}, \mathbf{g}_{\gamma_m})}(t, w)$ é a distribuição de *Wigner-Ville* cruzada e pode ser vista na seção 4.1.4. O somatório duplo da equação (4.70) se deve aos termos cruzados ou seja os termos de interferência. Isto pode ser visto já que no somatório “correlacionamos” dois átomos distintos de diferentes passos do processo de aproximação (n, m) . Podemos então eliminar os termos cruzados e obtemos uma distribuição tempo-freqüência que chamaremos de $Ex(t, \omega)$ e será dada por:

$$Ex(t, \omega) = \sum_{n=0}^{n=+\infty} \langle \mathbf{R}^n_x, g_{\gamma_n} \rangle^2 WD_{g_{\gamma_n}}(t, w). \quad (4.71)$$

Com esta abordagem as propriedades de deslocamento no tempo e na freqüência além das propriedades marginais continuam válidas.

A cada passo do *Matching Pursuits* com o dicionário de *Gabor* podemos aumentar o conhecimento do espectro do sinal conjugado ao tempo. Com isto teremos uma distribuição tempo-freqüência do sinal, que nos traz mais informação quanto mais passos realizarmos do *Matching Pursuits*.

Aplicando o escalamento, deslocamento e modulação à janela Gaussiana teremos o átomo de *Gabor* modulado, escalado e transladado, g_γ , $\gamma = [s, u, \xi]$, e a distribuição de *Wigner-Ville* deste átomo será dada por:

$$WD_{g_\gamma}(t, \omega) = WD_g \left(\frac{t-u}{s}, s(\omega - \xi) \right). \quad (4.72)$$

Teremos então que a distribuição $Ex(f, t)$ será dada por:

$$Ex(t, \omega) = \sum_{n=0}^{n=+\infty} \langle \mathbf{R}^n_x, g_{\gamma_n} \rangle^2 WD_g \left(\frac{t-u_n}{s_n}, s_n(\omega - \xi_n) \right), \quad (4.73)$$

onde $\gamma_n = [s_n, u_n, \xi_n]$. Repare que a fase não tem influência sobre a localização desta distribuição.

No caso do átomo real teremos uma pequena modificação na definição de WD_g e a distribuição $Ex(t, \omega)$ será dada por:

$$Ex(t, \omega) = \sum_{n=0}^{n=+\infty} \langle \mathbf{R}^n_x, g_{\gamma_n} \rangle^2 \left[WD_g \left(\frac{t-u_n}{s_n}, s_n(\omega - \xi_n) \right) - WD_g \left(\frac{t-u_n}{s_n}, s_n(\omega + \xi_n) \right) \right]. \quad (4.74)$$

O que teremos então é uma representação tempo-freqüência do sinal baseada na distribuição de *Wigner-Ville* dos átomos. A representação tempo-freqüência do sinal será a soma das representações tempo-freqüência dos átomos através dos quais decomposemos o sinal. Isto é a interpretação das equações (4.73) e (4.74). Um exemplo desta análise é mostrado na figura 4.22 para o sinal da figura 4.21.

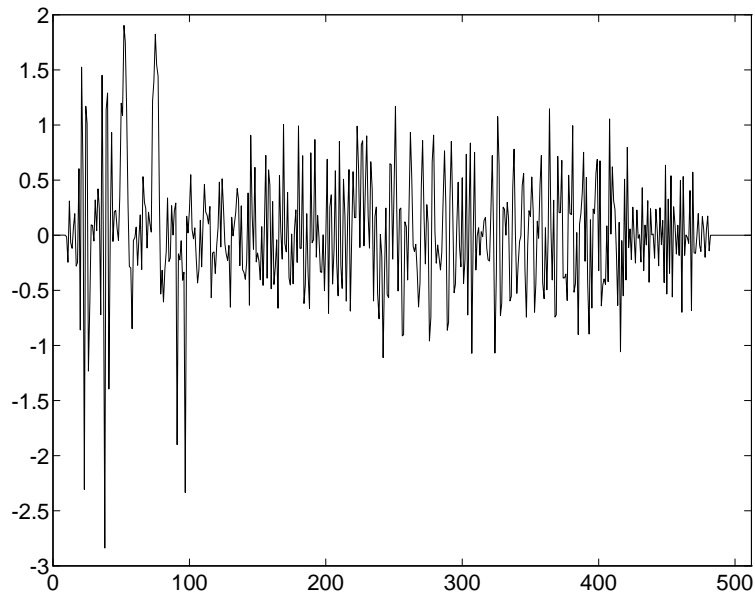


Figura 4.21: Sinal de 512 amostras formado a partir da adição *chirps*, senóides limitadas no tempo e formas de onda com diferentes localizações tempo-freqüência [1, 2].

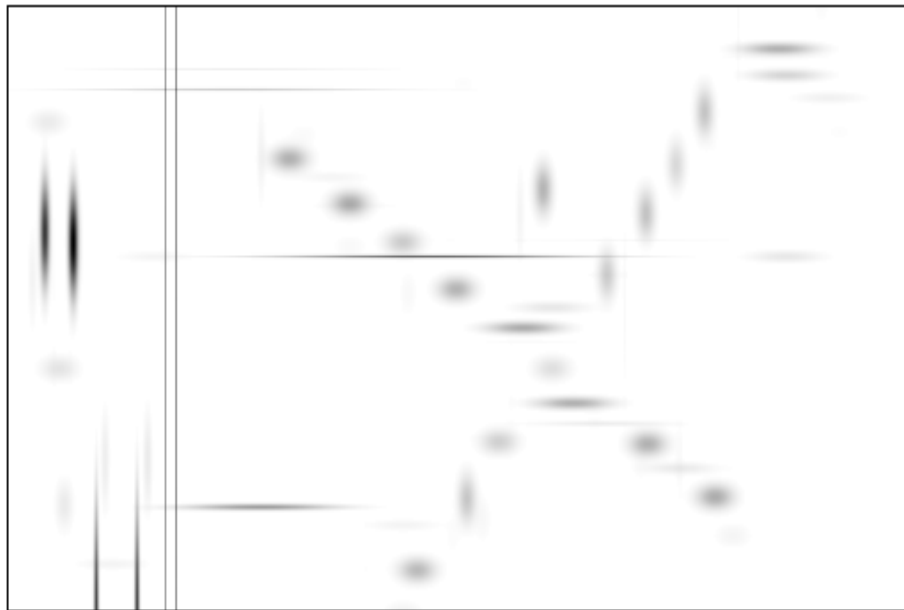


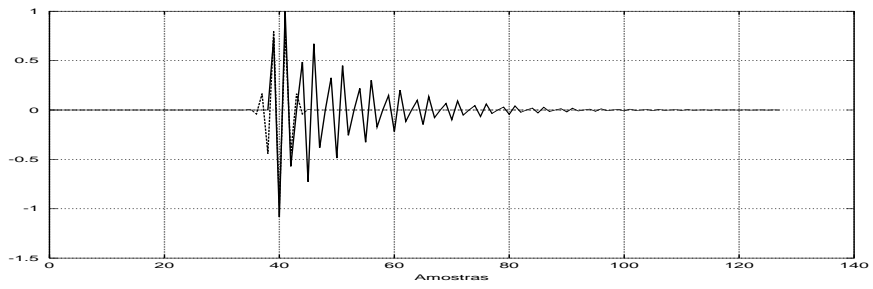
Figura 4.22: Distribuição de energia tempo-freüência, $Ef(t, \omega)$, seção 4.5, do sinal mostrado na figura 4.21. O eixo horizontal é o tempo. O vertical a freqüência. As freqüências mais altas estão no topo. Quanto maior $Ef(t, \omega)$ mais escuro o ponto [1, 2].

4.6 Comportamento Passo a Passo

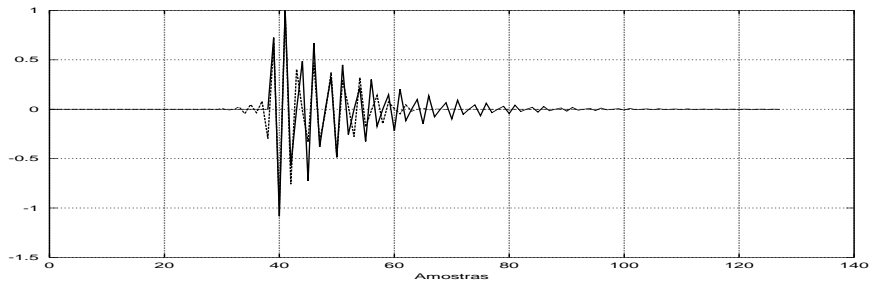
Vejam agora o comportamento da reconstrução de um sinal a partir da sua decomposição através do *Matching Pursuits* com o dicionário de *Gabor*. O sinal escolhido para isto foi o sinal “x007” que pode ser visto na figura A.7.

Na figura 4.23 podemos ver o sinal reconstruído comparado ao original, para o algoritmo de decomposição de 1 a 8 passos, mais especificamente com 1, 2, 4 e 8 passos. O sinal original é apresentado em linha “cheia” enquanto o reconstruído está pontilhado.

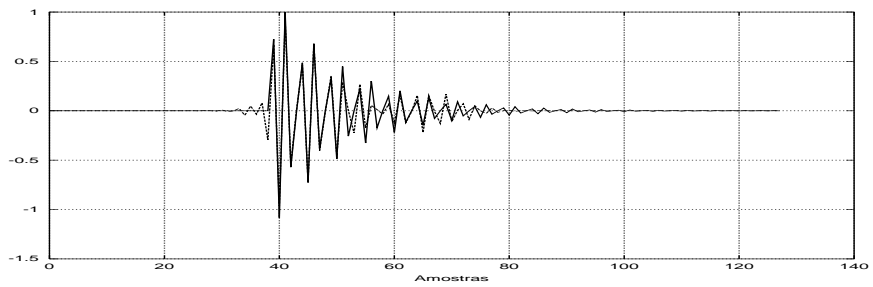
Com oito passos a energia do ruído em relação à energia do sinal já é de 0,023712 ou seja 2,3% o que já é um bom resultado. Mas podemos observar a introdução de “ruído”, pré-eco e pós-eco, ou seja de características que não estão presentes no sinal original. Além disso, como o algoritmo pode saber, automaticamente, se a decomposição é satisfatória? Estas são questões que trataremos nos capítulos a seguir.



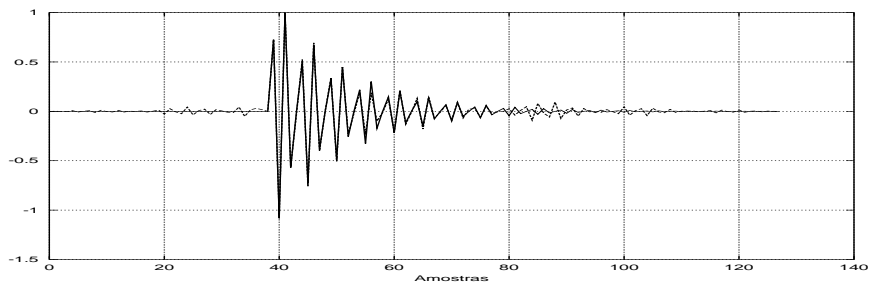
(a) 1 passo



(b) 2 passos



(c) 4 passos



(d) 8 passos

Figura 4.23: Comportamento da decomposição com diferentes números de passos para o sinal “x007”, 1 a 4 passos. O sinal original é apresentado em traço cheio e o reconstruído em traço pontilhado.

Capítulo 5

Representações Coerentes Com o Dicionário de *Gabor*

No capítulo anterior vimos como gerar o dicionário de *Gabor* e sua aplicação no algoritmo de *Matching Pursuits*. Vimos a decomposição de diversos sinais utilizando estas ferramentas. Nas diversas decomposições obtinhamos uma representação boa do sinal. Observamos nas figuras de mérito que podemos utilizar diversas medidas de distorção para decidir em que ponto da decomposição intorromper o processo. Porém estas medidas de desempenho não levam em consideração o dicionário utilizado. Na verdade estas medidas não levam em consideração o comportamento do dicionário na decomposição de ruído nem tampouco a noção de coerência discutida no capítulo 3, mais especificamente na seção 3.3. Além disso o *Matching Pursuits* após algumas iterações passa a codificar erros que foram introduzidos por ele mesmo [1]. Isto nos faz perguntar: como identificar em que ponto passamos a codificar erros? Ou em que ponto a representação deixa de ser coerente?

5.1 Estudo da Decomposição

Como vimos um dos primeiros fatores para podermos desenvolver a noção de coerência é entender como o algoritmo utilizado para representar sinais se comporta quando decompondo ruído. Para este fim utilizamos o *Matching Pursuits* com o dicionário de *Gabor* para decompor os sinais de ruído Gaussiano apresentados na seção A.3.

O que nos interessa agora é comparar o desempenho da decomposição de sinais à decomposição de ruído. Esta comparação será baseada em quatro medidas:

- **A taxa de aproximação no passo** – Esta medida foi apresentada na equação (3.31), que aqui repetimos:

$$\lambda(\mathbf{R}^n_{\mathbf{x}}) = \frac{\|\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle\|}{\|\mathbf{R}^n_{\mathbf{x}}\|}. \quad (5.1)$$

Lembramos que esta medida oferece uma leitura de quanto estamos aproximando do resíduo no passo n .

- **A taxa de aproximação média** – Que é baseada na medida anterior porém leva em consideração todos os passos já realizados. E é dada por:

$$\lambda_{\text{med}}(\mathbf{R}^n_{\mathbf{x}}) = \frac{1}{M} \sum_{n=0}^{M-1} \frac{\|\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle\|}{\|\mathbf{R}^n_{\mathbf{x}}\|}. \quad (5.2)$$

Porém esta medida possui uma deficiência: a alta taxa de aproximação dos passos iniciais é levada em consideração no cálculo da taxa de aproximação média em todos os outros passos.

- **Resíduo normalizado no passo** – Ao contrário das outras medidas, esta nos permite saber quanto do sinal ainda falta ser decomposto. Esta medida é dada por:

$$\text{Rn}(n) = \frac{\|\mathbf{R}^n_{\mathbf{x}}\|}{\|\mathbf{x}\|}. \quad (5.3)$$

Devemos ressaltar que a informação que a mesma fornece sobre “quanto” nos resta decompor é do ponto de vista do módulo, ou seja da energia presente no resíduo, e não sobre a informação do sinal.

- Consideraremos também o **RMSE** (*Root Mean Squared Error*), definido na equação (4.47) que aqui repetimos:

$$\text{RMSE}(q) = \sqrt{\frac{\sum_n |x(n) - \tilde{x}_q(n)|^2}{\sum_n |x(n)|^2}} \quad (5.4)$$

Esta é uma medida interessante pois ela é exatamente capaz de detectar os desvios do algoritmo mostrados na seção 4.6.

Postas estas três medidas de desempenho podemos comparar agora o comportamento delas para sinais e para ruídos. Na figura 5.1 podemos ver a comparação do

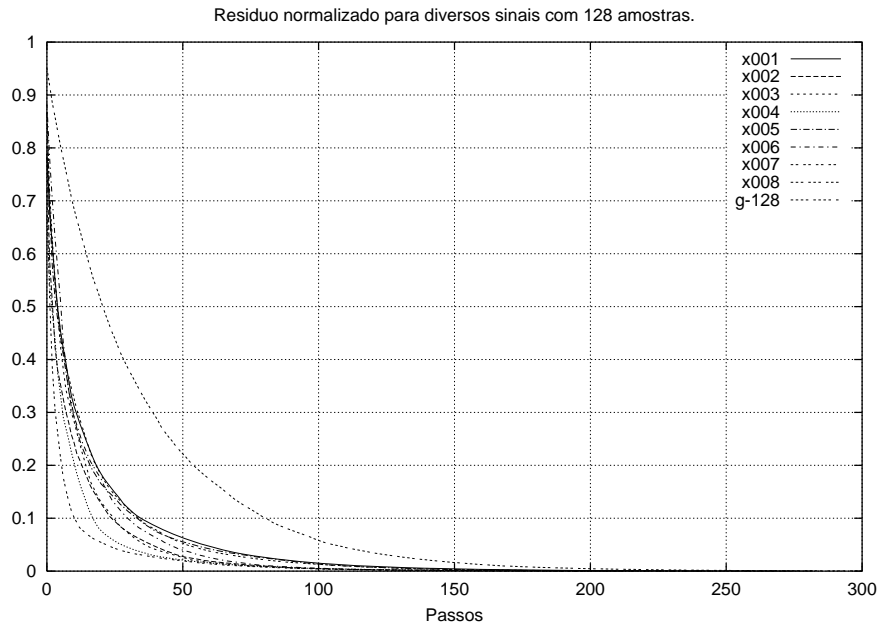


Figura 5.1: Resíduo normalizado após o passo para o *Matching Pursuits* com o dicionário de *Gabor* para diferentes sinais com 128 amostras.

comportamento do resíduo normalizado. Já na figura 5.2 o comportamento da taxa de aproximação no passo, e na figura 5.3 o comportamento da taxa de aproximação média, nestas o decaimento vertical indica o fim do processo de decomposição que como no capítulo anterior foi fixado em 300 passos ou parando quando o módulo do resíduo fosse menor que 0,001. Enquanto na figura 5.4 podemos ver o comportamento do RMSE.

O fator mais interessante a observar é a diferença no comportamento das medidas de desempenho dos sinais “x001” à “x008” em relação ao sinal de ruído “g-128”. Isto nos indica que teremos uma forma de discriminar quando a aproximação do resíduo num certo passo se comporta semelhante à decomposição de um ruído.

5.2 Definindo uma medida de Coerência

Nosso objetivo agora é aprofundar as idéias apresentadas na seção 3.3 e na seção anterior, e com isto chegar a uma forma de definir e realizar representações coerentes. No gráfico da taxa de aproximação no passo, figura 5.2, vemos que a partir de uma dada iteração os valores da taxa de aproximação no passo para os diferentes sinais estão em torno de um mesmo valor.

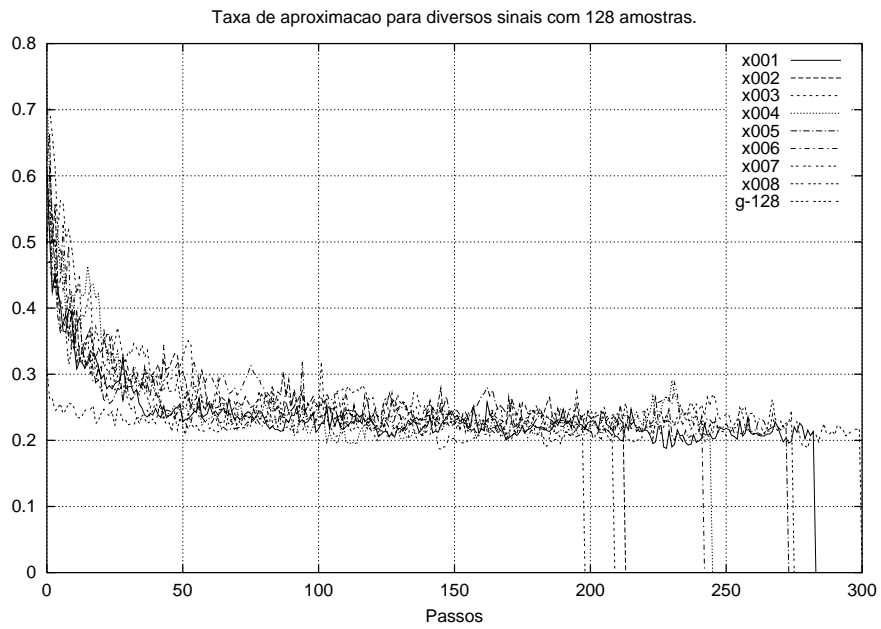


Figura 5.2: Taxa de aproximação no passo para o *Matching Pursuits* com o dicionário de *Gabor* para diferentes sinais com 128 amostras.

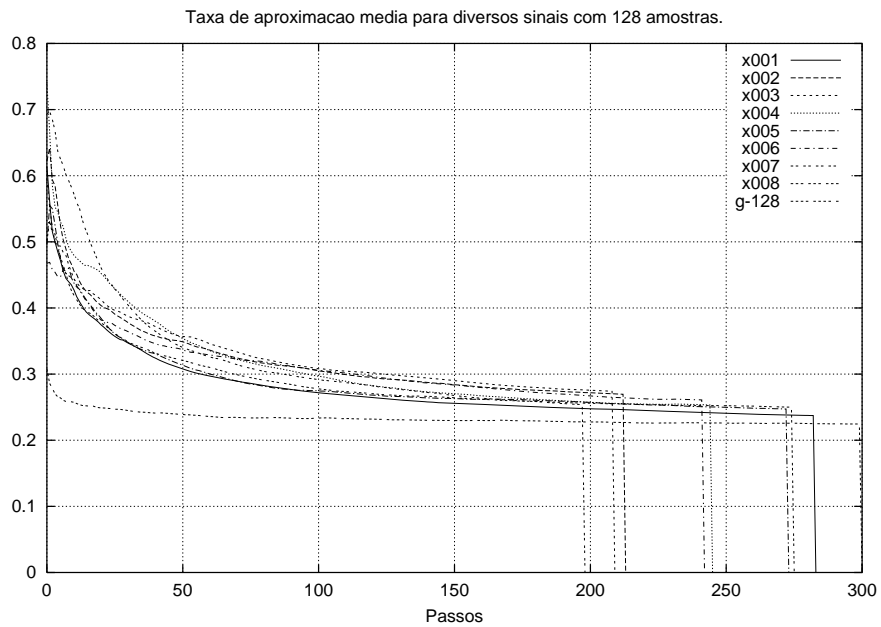


Figura 5.3: Taxa de aproximação média até o passo para o *Matching Pursuits* com o dicionário de *Gabor* para diferentes sinais com 128 amostras.

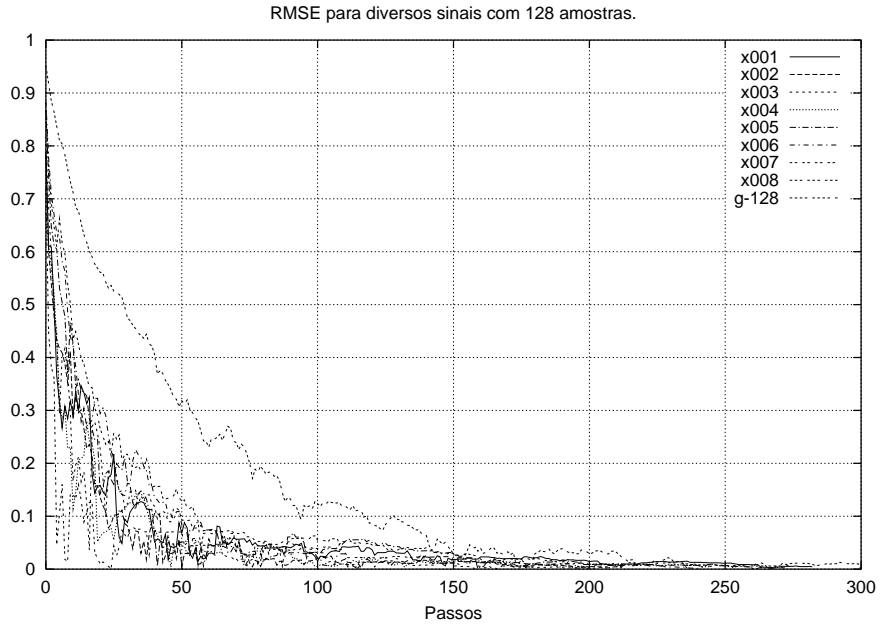


Figura 5.4: RMSE no passo para o *Matching Pursuits* com o dicionário de *Gabor* para diferentes sinais com 128 amostras.

Quando observamos o gráfico da taxa de aproximação média, figura 5.3, vemos que a partir de uma certa iteração esta permanece quase constante para todos os sinais, pelo menos todas são muito semelhantes, somente a do ruído gaussiano se encontra “abaixo” das outras. Este comportamento “inferior” se deve ao fato de estarmos considerando todas as iterações e com isto a taxa de aproximação alta obtida nos primeiros passos dos outros sinais possui alta influência no comportamento deste gráfico. Para libertar-nos da influência das primeiras iterações podemos lançar mão da taxa de aproximação média móvel definida na equação (3.41), que é:

$$\lambda_{\text{med}}(\mathbf{R}^n_{\mathbf{x}}) = \frac{1}{L} \sum_{n=M-L}^{M-1} \frac{\|\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle\|}{\|\mathbf{R}^n_{\mathbf{x}}\|}. \quad (5.5)$$

De posse desta medida podemos então decompor um sinal enquanto sua taxa de aproximação média, definida pela equação (5.5), seja razoavelmente maior que a taxa de aproximação para o ruído gaussiano. Veja que este procedimento permite realizar esta relação sem nos preocuparmos com a energia do sinal, já que a taxa sofre a normalização, além da filtragem. Faremos também uma modificação, ao invés de considerarmos as últimas L iterações consideraremos as próximas L iterações e com isto teremos L elementos a menos do livro de estruturas a armazenar, elementos estes que caracterizam um processo de aproximação de ruído gaussiano no dicionário

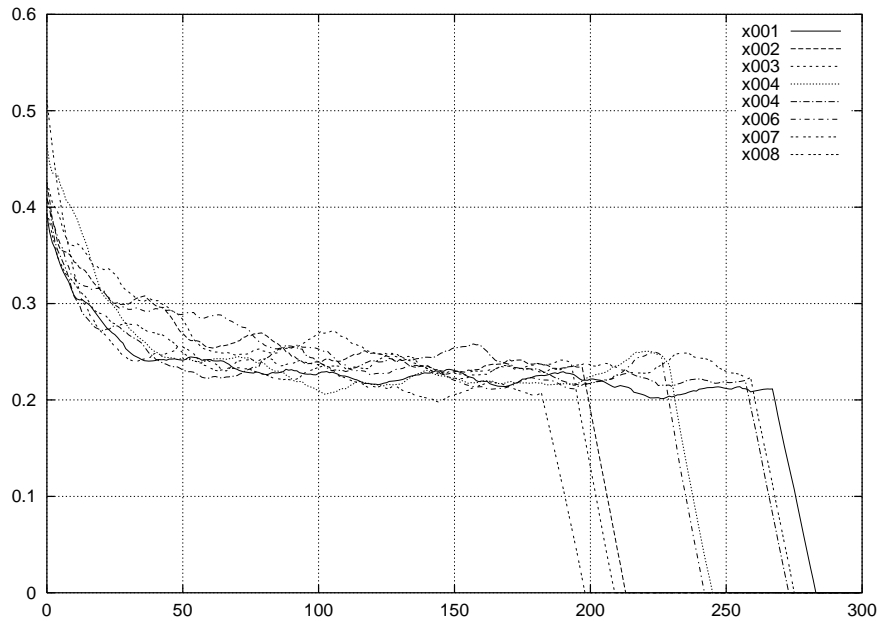


Figura 5.5: Taxa de aproximação média móvel, $L = 16$, no *Matching Pursuits* com o dicionário de *Gabor* para diferentes sinais com 128 amostras.

através do *Matching Pursuits*. Definimos então a taxa de aproximação média móvel como:

$$\lambda_{\text{med}}(\mathbf{R}^n_{\mathbf{x}}) = \frac{1}{L} \sum_{n=M-1}^{M+L} \frac{\|\langle \mathbf{R}^n_{\mathbf{x}}, \mathbf{g}_{\gamma_n} \rangle\|}{\|\mathbf{R}^n_{\mathbf{x}}\|}. \quad (5.6)$$

E teremos a representação coerente até a iteração em que:

$$\lambda_{\text{med}}(\mathbf{R}^n_{\mathbf{x}}) \geq \lambda_{\text{med}}(\mathbf{R}^n_{\mathbf{g}}) + \varepsilon, \quad (5.7)$$

onde \mathbf{g} é um ruído gaussiano e ε uma constante de confiança.

Na figura 5.5 podemos ver esta medida de coerência da aproximação no passo, para os diversos sinais estudados até o momento, com $L = 16$. Podemos reparar que em todos os casos estas taxas são muito próximas e bastante constantes. A queda no final indica que atingimos um resíduo com módulo inferior a 0,001 e o processo foi abortado. Estas são retas pois ainda assim calculamos a taxa de aproximação média porém considerando futuros zeros.

5.3 Diferentes Dimensões

Quando utilizamos sinais de dimensões diferentes, os dicionários de *Gabor* mudam. Passam a ter número de escalas e variações de modulação diferentes, ver

seção 4.3. A redundância do dicionário é alterada mas não no sentido da fase como vimos na subseção 4.4, e sim adquirindo diferentes características. Na tabela 5.1 podemos ver a redundância dos dicionários de *Gabor* em diferentes dimensões.

Dim. do Espaço	Tam. do Dicionário	Redundância
64	448	7
128	1024	8
256	2304	9
431	3569	8.28
512	5120	10
1024	11264	11
2048	24576	12

Tabela 5.1: Redundância do Dicionário de *Gabor* em diferentes dimensões.

Nesta tabela podemos observar alguns fatores interessantes, para dimensões de potências de dois, (2^k), a redundância do dicionário será dada pelo expoente mais um ($k + 1$), o que é fácil observar pela amostragem aplicada aos fatores $[j, p, k]$. Também vemos que para utilizar melhor o dicionário de *Gabor* gerado da forma descrita, é interessante utilizar sinais de comprimento que sejam potências de dois de forma a maximizar a redundância do dicionário. Como se comporta a taxa de aproximação? Na figura 5.6 podemos ver a taxa de aproximação no passo para sinais aleatórios gaussianos, com média zero e variancia unitária, para sinais com os comprimentos indicados na tabela 5.1. A taxa da aproximação média pode ser vista na figura 5.7 utilizando uma média com $L = \log_2(M)$, sendo M o comprimento do sinal. Pode-se observar que em todos os casos há uma certa regularidade da taxa de aproximação. Encontramos então uma taxa de aproximação média para o *Matching Pursuits* com o dicionário de *Gabor* dependente do comprimento do sinal, que estão sumarizadas na tabela 5.2. Vale a pena salientar que a taxa de aproximação não só é muito similar para diversos sinais gaussianos de mesmo comprimento, bem como para os outros sinais com o mesmo comprimento a partir de uma certa iteração. Estes números podem então ser utilizados como um critério para indicar até que ponto estamos realizando uma decomposição coerente com o dicionário.

Quando aplicamos estes limites de taxa de aproximação à decomposição dos

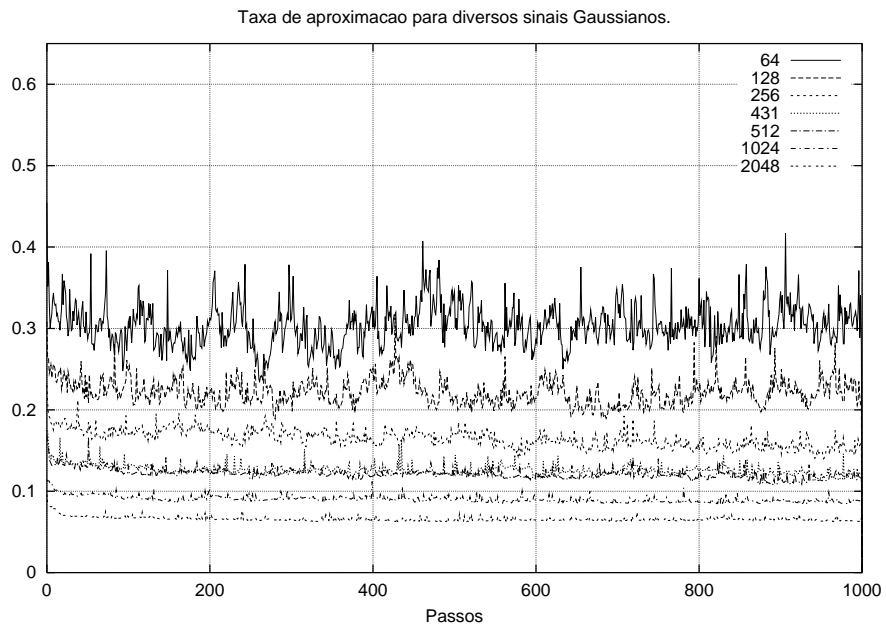


Figura 5.6: Taxa de aproximação no passo para sinais de Ruído Gaussiano de comprimentos diferentes

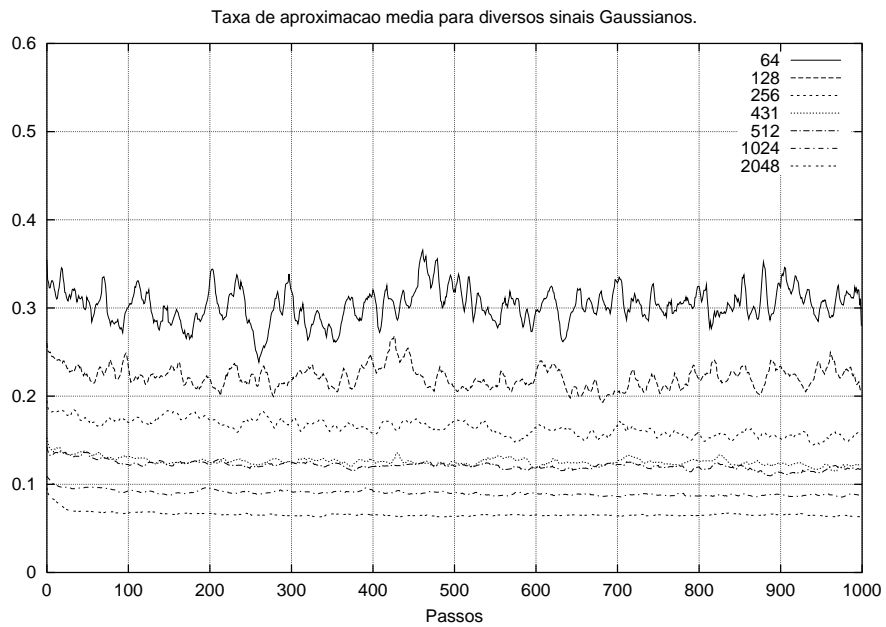


Figura 5.7: Taxa de aproximação média móvel no passo para sinais de Ruído Gaussiano de comprimentos diferentes

Comprimento do Sinal	λ_{med}
64	0.30
128	0.22
256	0.18
431	0.13
512	0.12
1024	0.09
2048	0.065

Tabela 5.2: Taxa de aproximação média para sinais de Ruído com diferentes comprimentos

sinais teremos uma decomposição do sinal coerente com o dicionário. Isto define uma medida de coerência da decomposição num dado passo.

Neste capítulo vimos então um dos problemas levantados no final do capítulo 4: como saber automaticamente se a composição é coerente ou não, definimos então uma medida de coerência da decomposição. No capítulo a seguir trataremos como melhorar a decomposição utilizando os átomos de *Gabor* reduzindo os erros de pré-eco e pós-eco gerados por esta decomposição vistos no capítulo 4, bem como aumentando a semelhança do átomo escolhido com o sinal.

Capítulo 6

Modificações no Dicionário de *Gabor*

Até o presente, vimos como gerar o dicionário de *Gabor* e utilizá-lo no contexto do algoritmo de *Matching Pursuits*, para decompor sinais. Além disso vimos no capítulo anterior como definir se a decomposição ainda é coerente ou não, a partir, do dicionário. Isto foi feito utilizando a taxa de aproximação média. Porém outros problemas como o aparecimento de pré-eco e pós-eco nas representações de sinais através do *Matching Pursuits* com o dicionário de *Gabor* ainda não foram abordadas. Neste capítulo teremos basicamente dois objetos de estudo: a eliminação do pré-eco e pós-eco, e a melhora no “casamento” da função do dicionário com o sinal.

6.1 Um Dicionário de Parâmetros Contínuos

Na seção 4.3 vimos como gerar o dicionário de *Gabor*. Este dicionário é gerado a partir de Gaussianas moduladas por exponenciais complexas, e as funções que compõem este dicionário são:

$$\begin{aligned} g_\gamma(t) &= \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) e^{i\xi t}, \\ g(t) &= 2^{\frac{1}{4}} e^{-\pi t^2}, \end{aligned} \tag{6.1}$$

onde $\gamma = [s, u, \xi]$.

Vimos também como amostrar os parâmetros que compõem o índice γ . Temos que γ , neste caso, é dado por:

$$\gamma = [2^j, p2^j, k\pi 2^{1-j}], \quad [j, p, k] \in \mathbb{Z}^3, \tag{6.2}$$

ou simplesmente:

$$\gamma_d = [j, p, k] \in \mathbb{Z}^3. \quad (6.3)$$

Além disso vimos na seção 4.4.1 como obter a fase ótima do átomo quando fazemos seu casamento com um sinal específico de forma a maximizar o produto interno, e portanto a aproximação do sinal em relação a um dado γ_d .

Porém o dicionário discreto de *Gabor*, gerado a partir de γ_d , não contém muitas das possíveis janelas Gaussianas moduladas por exponenciais. Para isto deveríamos ter um dicionário com todos os possíveis γ dados por $\gamma = [s, u, \xi]$, mas utilizar este dicionário é impraticável já que teria um número infinito de elementos. Podemos então realizar uma decomposição no dicionário discreto onde os elementos são gerados por $\gamma_d = [j, p, k]$ e realizar uma busca em torno de uma vizinhança deste ponto para encontrar γ que maximiza o produto interno. Em [1] é dito que isto é realizado utilizando o método de *Newton*, porém como não temos acesso à função do produto interno, não temos como encontrar o zero da mesma e portanto podemos implementar o método de *Newton* de forma a encontrar somente uma solução sub-ótima do ponto $[s, u, \xi]$, de forma genérica.

O que fazemos é partir de um índice $\gamma_d = [j, p, k]$ que maximiza o produto interno no passo do algoritmo de *Matching Pursuits* encontrar um $\gamma = [s, u, \xi] \in \mathbb{R}^3$ que obtém um produto interno maior numa vizinhança de $\gamma = [s_i, u_i, \xi_i]$ que são dados por:

$$s_i = 2^j, \quad (6.4)$$

$$u_i = p2^j, \quad (6.5)$$

$$\xi_i = k\pi 2^{1-j}. \quad (6.6)$$

Podemos então a partir deste ponto “procurar” próximo a uma vizinhança dele um ponto que maximize a produto interno com o sinal. Isto é feito fixando os passos de busca em função do ponto e da dimensão do sinal, N . Sendo $x(t)$ o sinal a codificar isto é feito iterativamente da seguinte forma:

1. Ache a fase ótima ϕ para o átomo g_{γ_d} , e faça $p = \langle g_{\gamma_d}, x(t) \rangle$.
2. Faça $\Delta s = \frac{s_i}{N/2}$.
3. Faça $\Delta \xi = \frac{\xi_i}{N/2}$.

4. Faça $\Delta u = \frac{u_i}{N/2}$.

5. Faça $\text{vez} = 0$.

6. Faça $\xi_{aux} = \xi_i + \Delta\xi$, $s_{aux} = s_i + \Delta s$, $u_{aux} = u_i + \Delta u$.

7. Caso:

- vez seja 0: gere o átomo de *Gabor* com $\gamma_t = [s_{aux}, u_i, \xi_i]$.
- vez seja 1: gere o átomo de *Gabor* com $\gamma_t = [s_i, u_i, \xi_{aux}]$.
- vez seja 2: gere o átomo de *Gabor* com $\gamma_t = [s_i, u_{aux}, \xi_i]$.

8. Ache a fase ótima, ϕ_{aux} , para o átomo g_{γ_t} e calcule o produto interno deste com o sinal, $p_{aux} = \langle g_{\gamma_t}, x(t) \rangle$.

9. Caso $p_{aux} > p$ então:

- se $\text{vez} = 0$:
 - (a) $s_i = s_{aux}$,
 - (b) $s_{aux} = s_{aux} + \Delta s$,
 - (c) $\phi = \phi_{aux}$ e
 - (d) $p = p_{aux}$.
- se $\text{vez} = 1$:
 - (a) $\xi_i = \xi_{aux}$,
 - (b) $\xi_{aux} = \xi_{aux} + \Delta\xi$,
 - (c) $\phi = \phi_{aux}$ e
 - (d) $p = p_{aux}$.
- se $\text{vez} = 2$:
 - (a) $u_i = u_{aux}$,
 - (b) $u_{aux} = u_{aux} + \Delta u$,
 - (c) $\phi = \phi_{aux}$ e
 - (d) $p = p_{aux}$.

10. Caso $p_{aux} \leq p$ então:

- se $\text{vez} = 0$:
 - (a) $\Delta s = -\frac{\Delta s}{2}$ e
 - (b) $s_{aux} = s_i + \Delta s$.
- se $\text{vez} = 1$:
 - (a) $\Delta \xi = -\frac{\Delta \xi}{2}$ e
 - (b) $\xi_{aux} = \xi_i + \Delta \xi$.
- se $\text{vez} = 2$:
 - (a) $\Delta u = -\frac{\Delta u}{2}$ e
 - (b) $u_{aux} = u_i + \Delta u$.

11. Faça $\text{vez} = \text{vez} + 1 \pmod 3$.

12. Caso os incrementos Δs , $\Delta \xi$ e Δu , sejam menores que as constantes pré-estabelecidas $(s_{lim}, u_{lim}, \xi_{lim})$, que definem até que precisão iterar, pare, senão vá para 7.

Ao final deste processo teremos $\gamma = [\xi_i, s_i, u_i]$ que gera o átomo g_γ que maximiza o produto interno, bem como a fase do átomo em ϕ e o produto interno do átomo com o sinal em p . Veja que desta forma apesar de otimizar cada um dos parâmetros do átomo por vez, levamos em consideração as influências dos outros parâmetros, já que as iterações e atualizações de cada um deles são intercaladas. Este algoritmo pode ser visto, inserido no algoritmo de *Matching Pursuits*, na figura 6.1.

6.2 Modificações no Suporte Temporal

Vimos na seção 4.6, o surgimento de efeitos de pré-eco e pós-eco, quando sintetizamos o sinal a partir da análise através do *Matching Pursuits* com o dicionário de *Gabor*. Para eliminar este efeito, após a escolha de um átomo num passo específico do *Matching Pursuits*, podemos verificar o suporte temporal para o mesmo átomo que mantém ou aumenta o produto interno. Neste caso ao invés de realizarmos o

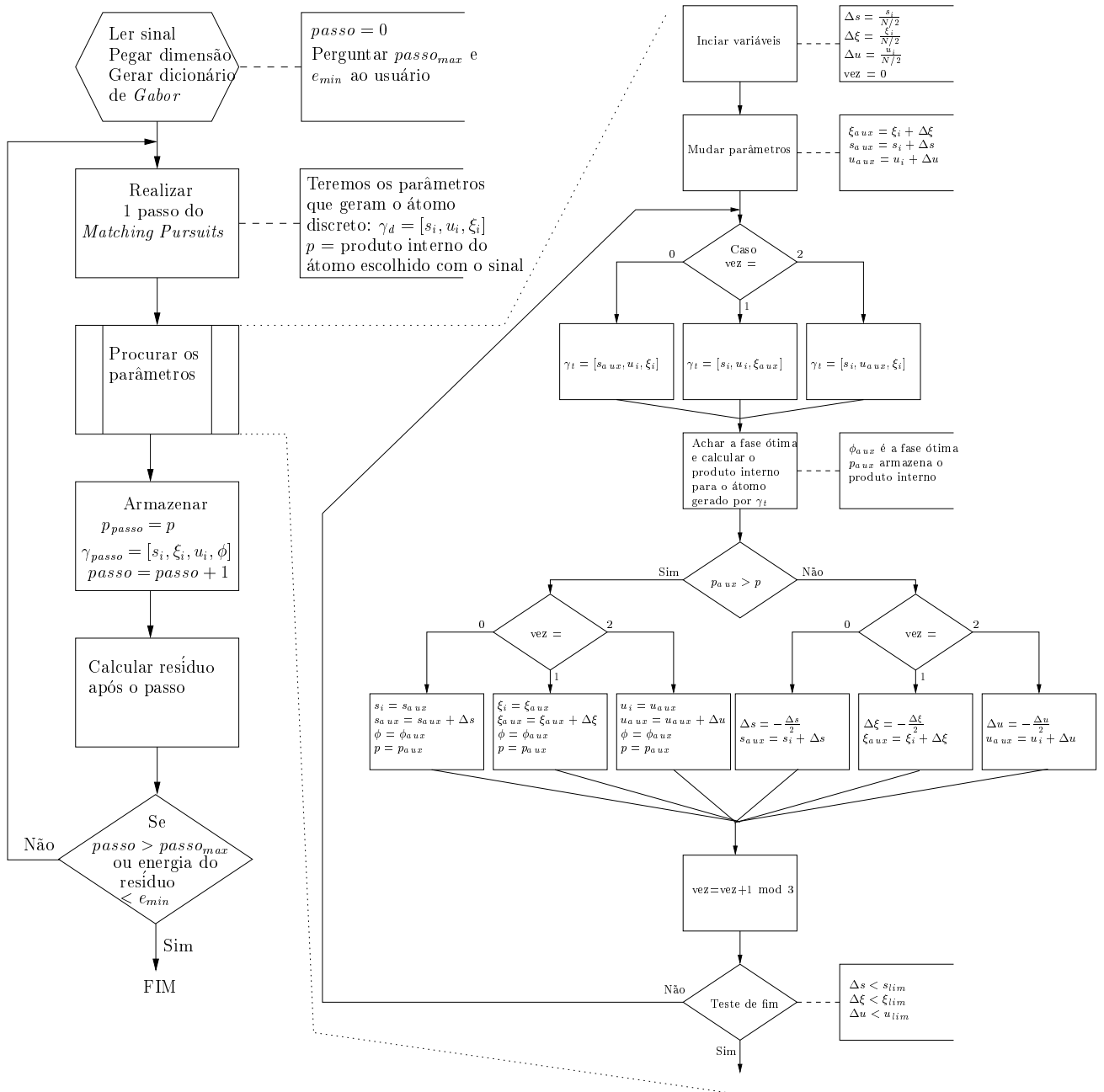


Figura 6.1: Procura dos parâmetros contínuos para geração dos átomos no algoritmo de *Matching Pursuits*.

Matching Pursuits com o átomo:

$$\begin{aligned} g_\gamma(t) &= \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) e^{i\xi t}, \\ g(t) &= 2^{\frac{1}{4}} e^{-\pi t^2}, \end{aligned} \tag{6.7}$$

faremos o *Matching Pursuits* com os átomos:

$$\begin{aligned} g_{\gamma_s}(t) &= \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) e^{i\xi t} (u(t-a) - u(t-b)), \\ g(t) &= 2^{\frac{1}{4}} e^{-\pi t^2}, \end{aligned} \tag{6.8}$$

onde $u(t)$ é a função degrau, e $a < b$, a é o instante de início do átomo e b o de fim.

Repare que não é necessario gerar todos os átomos com estas características; após identificado no *Matching Pursuits* o átomo que possui maior correlação com o sinal, verificamos se seu suporte temporal pode ser menor que o do átomo escolhido. Com isto procuraremos a e b tal que produto interno do sinal com o átomo com o novo suporte seja maior ou igual ao do átomo escolhido. Encontramos então a função $g_{\gamma_s}(t)$, equação (6.8), tal que seu produto interno com o sinal seja maior que o da função $g_\gamma(t)$, equação (6.7). Repare que os parâmetros do átomo $[s, u, \xi]$ são os mesmos fora seu suporte a e b . Teremos então átomos dados pela equação (6.8) indexados através de:

$$\gamma_s = [s, u, \xi, a, b]. \tag{6.9}$$

É verdade que aumentamos o número de parâmetros, porém, em contrapartida, obtemos mais informação sobre o sinal no passo em questão, além de diminuir o erro da representação e eliminar os efeitos de pré-eco e pós-eco.

O processo será então o seguinte:

1. Ache o átomo $g_\gamma(t)$ com fase ótima e parâmetros contínuos conforme apresentado na seção 6.1.
2. Para este átomo procure o átomo $g_{\gamma_s}(t)$, com menor suporte temporal, $b - a$, tal que o produto interno com o sinal a decompor seja maior ou igual a todos os átomos com suporte temporal maior que $b - a$ gerados a partir de $g_\gamma(t)$, ou seja dos parâmetros $\gamma = [s, u, \xi]$. Teremos então os parâmetros para gerar $g_{\gamma_s}(t)$ dados por $\gamma_s = [s, u, \xi, a, b]$.

6.3 Decompondo Sinais com o Dicionário de *Gabor* com Parâmetros Contínuos e Modificação do Suporte Temporal

Quando aplicamos estas modificações no dicionário e no *Matching Pursuits* podemos observar uma melhora na decomposição, ou seja a cada passo aumentamos a nossa taxa de aproximação. Isto pode ser observado através do produto interno no passo entre o sinal a decompor e o elemento escolhido do dicionário. Estes resultados podem ser observados na tabela 6.1. Nela temos os produtos internos no primeiro passo do algoritmo para os oito sinais sintéticos utilizados para testes.

Sinal	Dic. amostrado	Dic. contínuo	com Procura dos Limites
x001	4.457564	5.197896	5.209201
x002	4.307127	4.967297	4.989312
x003	3.729429	4.163669	4.163674
x004	6.065035	7.819225	7.819225
x005	7.811892	8.386336	8.386347
x006	4.641493	4.659904	4.953809
x007	-1.642846	2.087430	2.145189
x008	9.138810	-9.854947	-9.855015

Tabela 6.1: Comparação dos produtos internos sem as modificações no dicionário de *Gabor* e com elas para os sinais de testem no primeiro passo do *Matching Pursuits*.

Espera-se que com as modificações introduzidas no dicionário a taxa de aproximação aumente, o que pode ser observado pelos resultados apresentados. Porém, como esta se comportará para o ruído? A princípio da mesma maneira (aumentando), pois com as modificações estamos concedendo ao dicionário uma adaptabilidade maior, ou melhor estamos aumentando sua redundância. Isto comprovou-se em nossos experimentos, a taxa de aproximação média para um ruído Gaussiano de 128 amostras passa, com as modificações no dicionário, a 0,34 superior ao que tínhamos anteriormente.

Realizando a decomposição do sinal “x007” com estas modificações chegamos

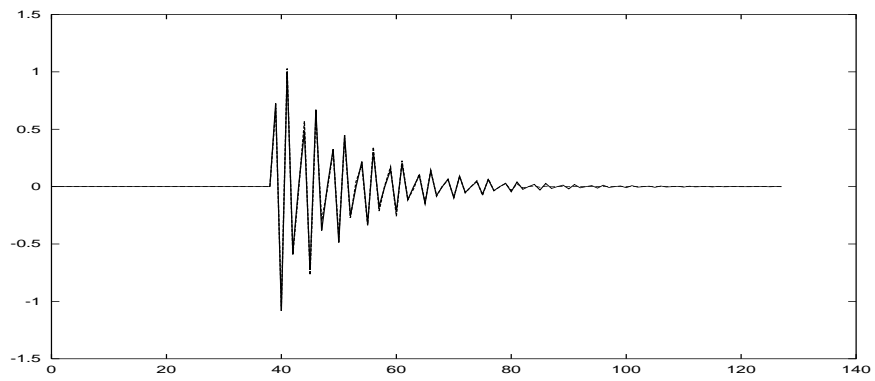
a resultados que podem ser comparados àqueles da seção 4.6. Por exemplo com oito passos de decomposição a energia do ruído em relação à energia do sinal já é de 0,000609 ou seja 0,06%, com seis passos esta já é de 0,001913 ou 0,19% e com 4 é de 0,007797 ou 0,77%. Os sinais reconstruídos nestes casos são vistos na figura 6.2 junto aos originais, sendo os originais em linha “cheia” e os reconstruídos em pontilhada. Introduzimos aqui uma outra medida de desempenho que utilizaremos adiante para comparar o método aqui proposto com outros métodos de compressão. Definindo a relação sinal ruído:

$$\text{SNR} = 10 \log \frac{\|\mathbf{x}\|^2}{\|\mathbf{x} - \tilde{\mathbf{x}}_q\|^2} \text{ dB.} \quad (6.10)$$

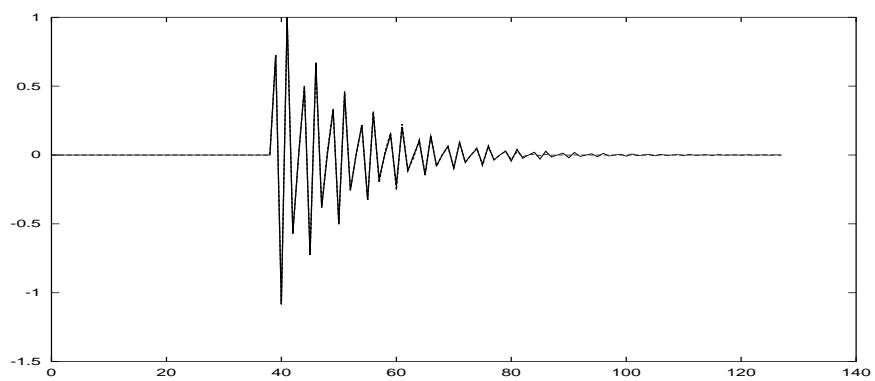
Teremos para estes casos esta medida dada por 32,15 dB, 27,18 dB e 21,08 dB para 8, 6 e 4 passos respectivamente.

Devemos agora tentar entender estas decomposições do ponto de vista fenomenológico. Será que os átomos escolhidos são condizentes com os fenômenos presentes em sinais elétricos? Para verificar a validade deste procedimento utilizaremos os sinais sintéticos apresentados até o momento. Estes foram gerados segundo o modelo apresentado na seção A.1, equação (7.1). Apesar deste modelo não levar em consideração todas as condições presentes em linhas de transmissão (equações diferenciais das linhas), é um modelo que nos permite gerar sinais muito semelhantes aos reais, com a vantagem de que podemos saber *a priori* os fenômenos presentes no sinal. A composição de cada um dos sinais pode ser vista na tabela A.1. Faremos agora um estudo então das decomposições destes sinais realizada pelo *Matching Pursuits* com o dicionário de *Gabor* com as modificações apresentadas neste capítulo. Nos gráficos apresentados os sinais originais estão em linha cheia e os reconstruídos em linhas pontilhadas.

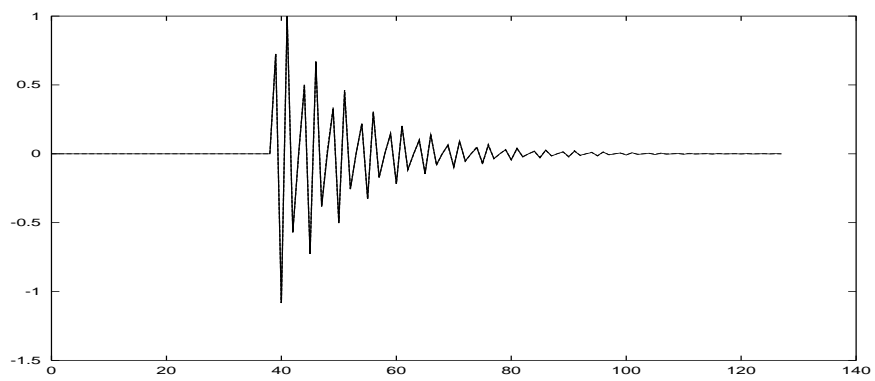
O sinal “x001” possui 4 estruturas, o problema ao decompor este sinal pelo método apresentado até então é que duas destas estruturas são agrupadas em uma única estrutura, no primeiro passo, incorrendo assim em um erro do ponto de vista fenomenológico logo no primeiro passo. Isto pode ser visto na figura 6.3. Veja que apesar de termos uma alta taxa de aproximação, 0,935639, as duas estruturas de mesma frequência, sendo uma delas uma senóide e a outra uma senóide de menor amplitude, são confundidas no mesmo átomo. Este era um resultado esperado já que apesar do dicionário ser capaz de representar senóides, a única forma que o mesmo



(a) 4 passos



(b) 6 passos



(c) 8 passos

Figura 6.2: Comportamento da decomposição com diferentes números de passos para o sinal “x007”, com 4, 6 e 8 passos, para as modificações introduzidas no dicionário. O sinal original está em linha cheia e o reconstruído em linha pontilhada.

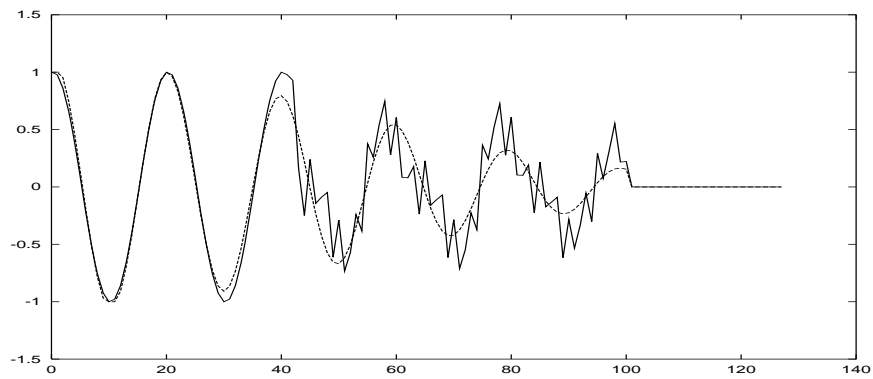


Figura 6.3: Comportamento da decomposição do sinal “x001”, no primeiro passo, para as modificações introduzidas no dicionário. O sinal original está em linha cheia e o reconstruído em linha pontilhada.

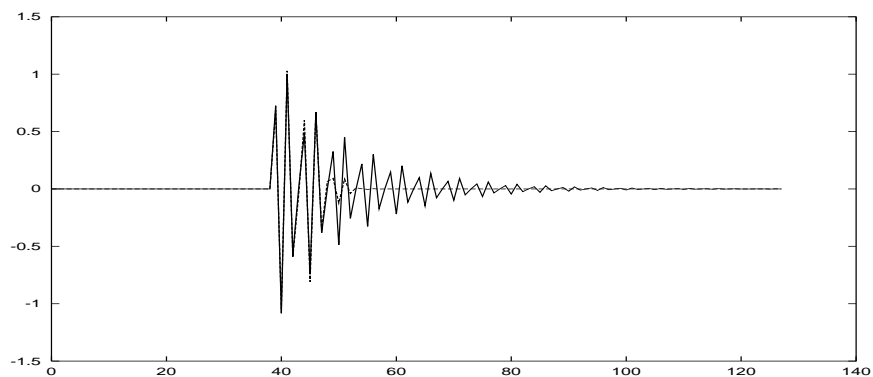


Figura 6.4: Comportamento da decomposição do sinal “x007”, no primeiro passo, para as modificações introduzidas no dicionário. O sinal original está em linha cheia e o reconstruído em linha pontilhada.

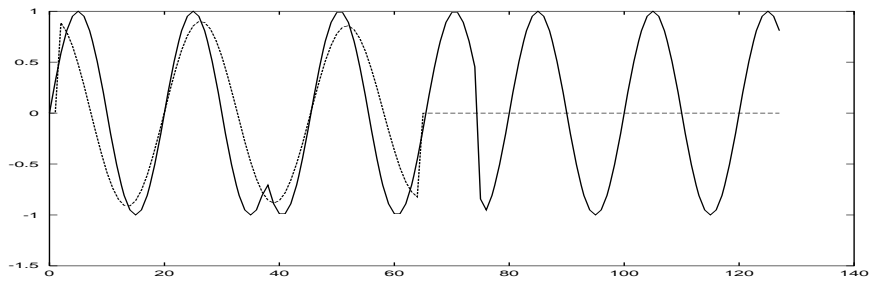
possui de representar exponenciais é através de uma soma de Gaussianas. Como a cada passo maximizamos o produto interno, espera-se que se além das estruturas possuírem a mesma frequência, estejam com a mesma fase. Isto implica que sejam entendidas como um único átomo.

Outro caso interessante é o do sinal “x007” que sabemos ser formado por uma única exponencial, porém, para compô-la o algoritmo necessita utilizar diversas Gaussianas, com o que fica impossível identificar a sua única estrutura. O primeiro passo da decomposição deste sinal pode ser visto na figura 6.3.

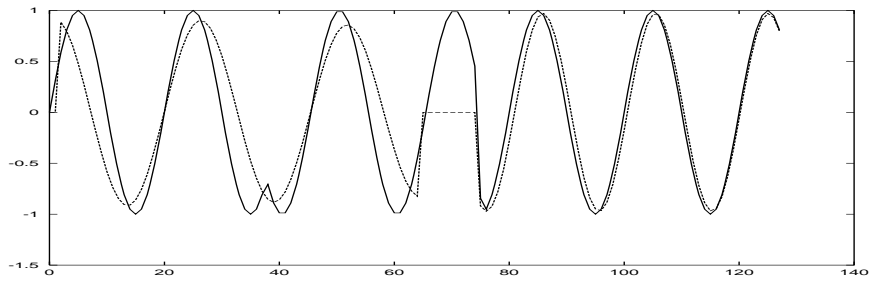
Um problema interessante é entender como se dará o comportamento da decomposição quando o sinal a decompor possui súbitas mudanças de fase. Isto pode ser observado no caso do sinal “x006”. O sinal reconstruído nos 4 primeiros

passos desta decomposição pode ser visto na figura 6.5. Vemos que o algoritmo se perde nesta decomposição, identificando então um problema crítico quando ocorre uma mudança de fase do sinal.

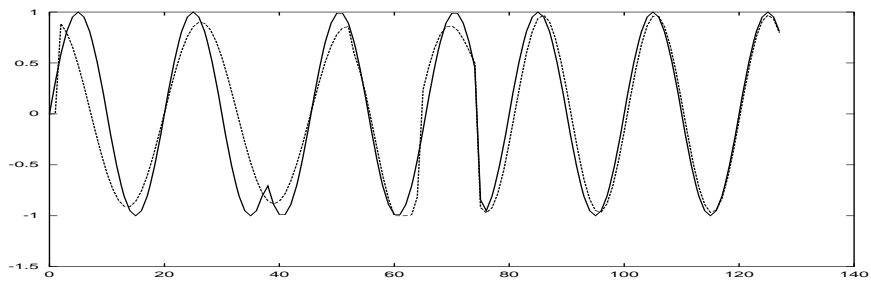
Destes resultados podemos concluir que as modificações introduzidas são válidas e melhoram o desempenho do *Matching Pursuits* com o dicionário de *Gabor*. Porém a decomposição realizada desta forma ainda não é adequada para a representação de sinais elétricos. Pois apesar de identificar as Gaussianas presentes nos sinais, através do dicionário de *Gabor*, estas estruturas não são apropriadas para representar sinais elétricos. Isto se dá por que não há uma relação destas estruturas com os fenômenos presentes em sinais elétricos. Este é o tema de nosso próximo capítulo.



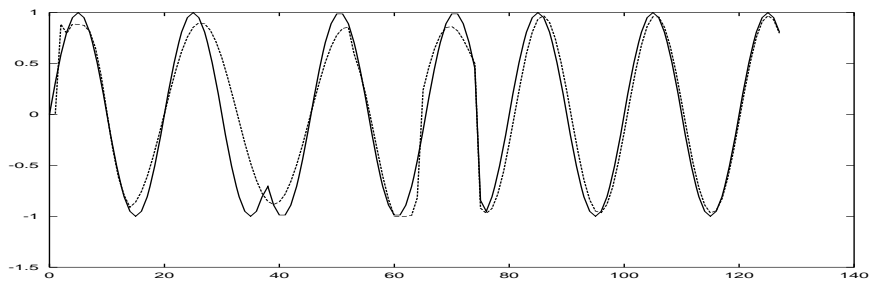
(a) 1 passo



(b) 2 passos



(c) 3 passos



(d) 4 passos

Figura 6.5: Comportamento da decomposição com diferentes números de passos para o sinal “x006”, com 1, 2, 3 e 4 passos, para as modificações introduzidas no dicionário. O sinal original está em linha cheia e o reconstruído em linha pontilhada.

Capítulo 7

Compressão de Sinais Elétricos

Através de Representações Coerentes

Vimos então como obter uma boa representação, do ponto de vista, de um critério de erro de sinais elétricos através do *Matching Pursuits* utilizando um dicionário de *Gabor*, que é formado por Gaussianas moduladas por exponenciais. Vimos inclusive como eliminar os efeitos de pré-eco e pós-eco surgidos ao utilizar átomos de *Gabor*. Porém esta ainda não é uma boa representação de sinais elétricos do ponto de vista das componentes presentes no mesmo.

Segundo *Rodrigues* [19], um modelo razoável para representar sinais elétricos é dado por uma soma de exponenciais moduladas por harmônicos de uma frequência fundamental. Com isto podemos representar sinais elétricos através de:

$$f(t) = \sum_{q=0}^{Q-1} A_q \cos(2\pi F k_q t + \phi_q) e^{-\lambda_q t} [u(t - t_{0_q}) - u(t - t_{f_q})]. \quad (7.1)$$

Onde Q é o número de componentes, F a frequência fundamental (50 ou 60Hz), A_q a amplitude da componente, $k_q \in \mathbb{Z}$ é o harmônico, $u(t)$ é a função degrau e t_{0_q} e t_{f_q} são respectivamente o início e o fim da componente (região de suporte), ϕ_q a fase da componente e λ_q é o coeficiente da exponencial.

7.1 Dicionário de Átomos Exponenciais

Então, seguindo o modelo apresentado, poderíamos gerar um dicionário de exponenciais e realizar o *Matching Pursuits* com este dicionário. Neste dicionário

os átomos seriam dados por:

$$g(t) = A \cos(2\pi\xi t + \phi) e^{-\lambda t} [u(t - t_0) - u(t - t_f)], \quad (7.2)$$

ou complexo:

$$g(t) = A e^{j(2\pi\xi t + \phi)} e^{-\lambda t} [u(t - t_0) - u(t - t_f)]. \quad (7.3)$$

Porém o problema que se apresenta, nesta abordagem, é que não há até o momento como prever uma amostragem do coeficiente λ de forma a garantir que o conjunto de funções formará um *frame* e portanto será capaz de representar sinais, ao contrário do que vimos para o dicionário de átomos de *Gabor* na seção 4.3.

Em [19] foi desenvolvido um agrupamento dos átomos gaussianos que definem um átomo exponencial e a partir deste agrupamento um mapeamento dos grupos de átomos gaussianos em átomos exponenciais. Implementamos este expediente, porém este, por sua vez, confunde algumas componentes, e outras, não identifica. Optamos então por uma abordagem diferente. O que fizemos é, a partir de um átomo Gaussiano, “estimar” um átomo exponencial, e utilizar um processo de procura junto a este átomo similar ao da seção 6.1.

7.1.1 Encontrando o Átomo Exponencial

Partindo do índice discreto $\gamma_d = [j, p, k]$ que maximiza o produto interno no passo do algoritmo de *Matching Pursuits* com o dicionário discreto de *Gabor*, encontramos um $\gamma = [s_k, u_k, \xi_k]$ que é dado por por:

$$s_k = 2^j, \quad (7.4)$$

$$u_k = p2^j, \quad (7.5)$$

$$\xi_k = k\pi 2^{1-j}. \quad (7.6)$$

Depois, otimizamos o átomo conforme apresentado na seção 6.1, obtendo um novo $\gamma = [s_i, u_i, \xi_i]$. Repare que para este novo átomo o processo de cálculo da fase ótima apresentado na subseção 4.4.1 continua válido. A partir destes parâmetros $\gamma = [s_i, u_i, \xi_i]$ otimizados achamos uma exponencial que mais se assemelha da Gaussiana na vizinhança do sinal próxima a u_i . Isto pode ser feito fazendo:

$$\lambda = \frac{\xi_i}{s_i^2}. \quad (7.7)$$

Uma exceção ocorre nos casos quando $s_i < 1$. Nestes fazemos $\lambda = \frac{1}{4}$. Este valor foi obtido experimentalmente.

Após isto fazemos um método de procura no novo átomo dado por:

$$g(t) = K_g e^{-j(2\pi\xi t)} e^{-\lambda(t-t_0)} [u(t-t_0) - u(t-t_f)], \quad (7.8)$$

onde K_g é um fator de normalização do átomo. Nesta procura não nos preocupamos com t_f e o fazemos igual ao comprimento do sinal e $t_0 = u_i$. A procura é feita de forma semelhante à apresentada na seção 6.1, porém com novos parâmetros $\gamma = [\lambda, \xi, u]$, sendo dado por:

1. Ache a fase ótima ϕ para o átomo g_γ , e faça $p = \langle g_\gamma, x(t) \rangle$.
2. Faça $\Delta\lambda = \frac{\lambda}{N/2}$.
3. Faça $\Delta\xi = \frac{\xi}{N/2}$.
4. Faça $\Delta u = \frac{u}{N/2}$.
5. Faça $\text{vez} = 0$.
6. Faça $\xi_{aux} = \xi + \Delta\xi$, $\lambda_{aux} = \lambda + \Delta\lambda$, $u_{aux} = u + \Delta u$.
7. Caso:
 - vez seja 0: gere o átomo de *Gabor* com $\gamma_t = [\lambda_{aux}, u, \xi]$.
 - vez seja 1: gere o átomo de *Gabor* com $\gamma_t = [\lambda, u, \xi_{aux}]$.
 - vez seja 2: gere o átomo de *Gabor* com $\gamma_t = [\lambda, u_{aux}, \xi]$.
8. Ache a fase ótima, ϕ_{aux} , para o átomo g_γ e calcule o produto interno deste com o sinal, $p_{aux} = \langle g_\gamma, x(t) \rangle$.
9. Caso $p_{aux} > p$ então:
 - se $\text{vez} = 0$:
 - (a) $\lambda = \lambda_{aux}$,
 - (b) $\lambda_{aux} = \lambda_{aux} + \Delta\lambda$,
 - (c) $\phi = \phi_{aux}$ e
 - (d) $p = p_{aux}$.

- se $\text{vez} = 1$:
 - (a) $\xi = \xi_{aux}$,
 - (b) $\xi_{aux} = \xi_{aux} + \Delta\xi$,
 - (c) $\phi = \phi_{aux}$ e
 - (d) $p = p_{aux}$.
- se $\text{vez} = 2$:
 - (a) $u = u_{aux}$,
 - (b) $u_{aux} = u_{aux} + \Delta u$,
 - (c) $\phi = \phi_{aux}$ e
 - (d) $p = p_{aux}$.

10. Caso $p_{aux} \leq p$ então:

- se $\text{vez} = 0$:
 - (a) $\Delta\lambda = -\frac{\Delta\lambda}{2}$ e
 - (b) $\lambda_{aux} = \lambda + \Delta\lambda$.
- se $\text{vez} = 1$:
 - (a) $\Delta\xi = -\frac{\Delta\xi}{2}$ e
 - (b) $\xi_{aux} = \xi + \Delta\xi$.
- se $\text{vez} = 2$:
 - (a) $\Delta u = -\frac{\Delta u}{2}$ e
 - (b) $u_{aux} = u + \Delta u$.

11. Faça $\text{vez} = \text{vez} + 1 \pmod{3}$.

12. Caso os incrementos $\Delta\lambda$, $\Delta\xi$ e Δu , sejam menores que as constantes pré-estabelecidas $(\lambda_{lim}, u_{lim}, \xi_{lim})$, que definem até que precisão iterar, pare, senão vá para 7.

Ao final deste processo teremos $\gamma = [\lambda, \xi, u]$ que gera o átomo g_γ , da equação (7.8), que maximiza o produto interno, bem como a fase do átomo em ϕ e o produto interno do átomo com o sinal em p . Veja que desta forma apesar de otimizar cada um dos parâmetros do átomo por vez, levamos em consideração as influências dos outros

parâmetros, já que as iterações e atualizações de cada um deles são intercaladas, e neste caso as otimizações da frequência ξ quase não interferem no processo. Vale ressaltar que o parâmetro u pode ser discretizado e seu limite considerado como o número de amostras do sinal N .

Devemos observar que se encontramos um impulso inicialmente no casamento, *Matching Pursuits*, com átomos de *Gabor* não realizamos esta procura por um átomo exponencial.

É interessante notar que no processo de procura acabamos com átomos exponenciais gerados por $\gamma = [\lambda, \xi, u, \phi]$ que são desreferenciados [7] já que a exponencial e a senóide podem “iniciar” em instantes distintos, sendo o início da exponencial dado por u e o da senóide complexa também, porém a fase ϕ permite que a última inicie em qualquer ângulo, desreferenciando assim uma da outra. Vejamos a definição de um átomo referenciado, este átomo terá a forma [7]:

$$g(t) = e^{\lambda(t-u)} e^{j\xi(t-u)} u(t-u), \quad (7.9)$$

enquanto num átomo desreferenciado não necessariamente a exponencial real e a imaginária iniciam num mesmo ponto e teremos:

$$g(t) = e^{\lambda(t-u)} e^{j\xi(t)} u(t-u). \quad (7.10)$$

Vemos então que as exponenciais e as senóides neste caso iniciam em momentos diferentes, o mesmo ocorre com nosso átomo definido na equação (7.2).

Após isto fazemos uma procura dos limites (t_o, t_f) como foi feita para o átomo Gaussiano na seção 6.2. Assim como fizemos para u podemos considerar t_o e $t_f \in [0, N-1]$. Teremos então um átomo exponencial dado por $\gamma = [\lambda, \xi, u, \phi, t_o, t_f]$. A decomposição feita desta forma já é melhor que a realizada só com o átomo gaussiano, por exemplo para o sinal x002 obtemos um produto interno de 5,040318, maior que os anteriores, no primeiro passo. O sinal reconstruído para o primeiro passo desta decomposição pode ser visto na figura 7.1. Porém vemos que mesmo desta forma as senóides que compõem o sinal “x002” não foram separadas corretamente e sim agrupadas em um mesmo átomo exponencial. Com isto podemos observar que o método ainda possui deficiência para a decomposição em componentes do sinal.

Para resolver o problema mencionado no parágrafo anterior faremos uma procura de possíveis senóides no sinal, pois este é o problema que podemos identificar

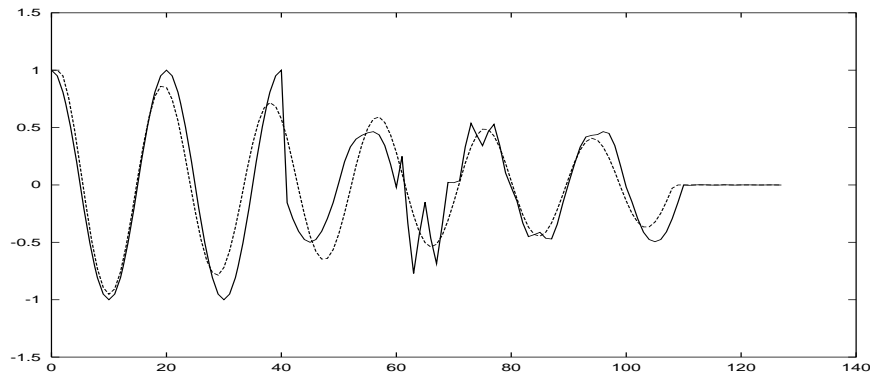


Figura 7.1: Comportamento da decomposição do sinal “x002”, no primeiro passo, para o dicionário com átomos exponenciais.

na figura 7.1, duas senóides confundidas como uma única senóide modulada por exponencial. Para encontrar uma senóide geramos uma senóide com a frequência ξ , obtida até aqui, e verificamos, ao mesmo tempo que realizamos a varredura dos parâmetros (t_o, t_f) encontrando novos parâmetros (t_{os}, t_{fs}) , duas condições:

1. Se o produto interno com a senóide é maior que a metade do obtido com o átomo gerado por $\gamma = [\lambda, \xi, u, \phi, t_o, t_f]$;
2. Se a norma do novo resíduo dividida pelo comprimento do átomo senoidal $(t_{fs} - t_{os})$ é menor que a norma do resíduo com o átomo exponencial sobre seu comprimento $(t_f - t_o)$.

Caso estas duas condições sejam satisfeitas consideramos a possibilidade de um átomo senoidal. O segundo fator nos permite somente considerar os casos em que reduzimos o erro no pedaço (t_{os}, t_{fs}) . Se encontramos uma senóide que satisfaz estas condições fazemos uma nova procura na frequência ξ e o cálculo da fase ótima ϕ .

De posse dos resultados utilizando um átomo senoidal fazemos mais algumas etapas:

1. Verificamos seu comprimento $(t_{fs} - t_{os})$ se este for menor que o do átomo exponencial $(t_f - t_o)$ e a projeção na senóide for maior que 90%, limite obtido experimentalmente, da projeção no átomo exponencial consideramos um átomo senoidal para a decomposição no passo corrente.
2. Se seu comprimento $(t_{fs} - t_{os})$ for maior que o do átomo exponencial $(t_f - t_o)$ e

a projeção na senóide for maior que a projeção no átomo exponencial também consideramos um átomo senoidal.

Se alguma das condições acima for satisfeita, procuramos os limites do átomo senoidal.

Teremos assim uma decomposição do sinal em átomos exponenciais modulados por senóides. Para evitar que uma composição de senóides seja confundida com uma exponencial procuramos uma senóide como explicado nesta seção.

7.2 Decompondo Sinais Com Átomos Exponencias

Os sinais sintéticos apresentados na seção A.1, são agora decompostos com este procedimento. Uma das vantagens de fazê-lo é que nestes sinais temos acesso às suas componentes de acordo com o modelo de sinal elétrico apresentado na equação (7.1), e com isto podemos verificar o comportamento do método na representação dos sinais e na busca de suas estruturas.

Mais um fator foi incluído na decomposição. Muitos sinais elétricos podem ser ceifados. Para lidar com este problema o método de decomposição até aqui apresentado necessitaria uma série de harmônicos para poder representá-lo. De forma a obter uma representação mais compacta e coerente com o sinal, vejamos o que ocorre com o sinal x008, que pode ser observado na figura 7.2. Veja que o algoritmo escolhe uma componente de amplitude maior que o sinal original. O que fazemos então é verificar o maior valor presente no sinal original e utilizá-lo como um limiar no último passo da reconstrução, desta forma ceifamos o sinal reconstruído no mesmo valor e melhoramos o desempenho da decomposição nestes casos. Por exemplo, no caso do sinal “x004” sem limiar de ceifamento obtemos um SNR de 41,46 dB com a reconstrução com dois elementos, quando usamos o limiar de ceifamento obtém-se um SNR de 48,45 dB melhorando portanto o desempenho da reconstrução.

Faremos os *Matching Pursuits* com um dicionário de exponenciais:

$$g(t) = Ae^{j(2\pi\xi t + \phi)}e^{-\lambda t}[u(t - t_0) - u(t - t_f)]. \quad (7.11)$$

Para isto:

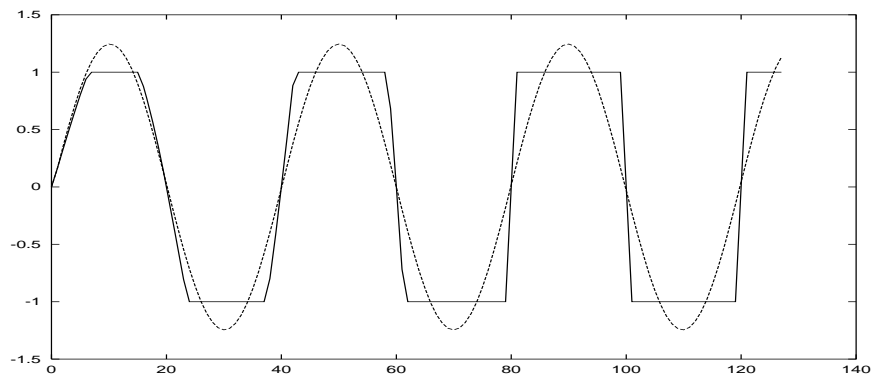


Figura 7.2: Comportamento da decomposição do sinal “x008”, no primeiro passo, para o dicionário com átomos exponenciais.

1. Fazemos os *Matching Pursuits* com o dicionário de *Gabor* com parâmetros contínuos e suporte temporal.
 2. A partir do átomo de *Gabor* escolhido procuramos o átomo exponencial.
 - Isto é feito escolhendo o átomo exponencial que mais se assemelha do Gaussiano na região de maior energia e a seguir uma procura dos parâmetros do átomo exponencial.
 3. Para este átomo achamos os limites temporais de forma a minimizar o resíduo.
 - Os limites podem ser diferentes daqueles do átomo Gaussiano, neste caso além do produto interno consideramos o erro na região de suporte.
 4. Procuramos por uma senóide em vez de uma exponencial.
 - Pois duas senóides de diferentes amplitudes são confundidas por uma exponencial. Os critérios utilizados para decidir por uma senóide ou não serão:
 - (a) Produto interno.
 - (b) Erro na região de suporte da senóide.
5. Como sinais elétricos podem estar ceifados identificamos no sinal um limiar de ceifamento a ser aplicado ao sinal reconstruído após o último passo da reconstrução.

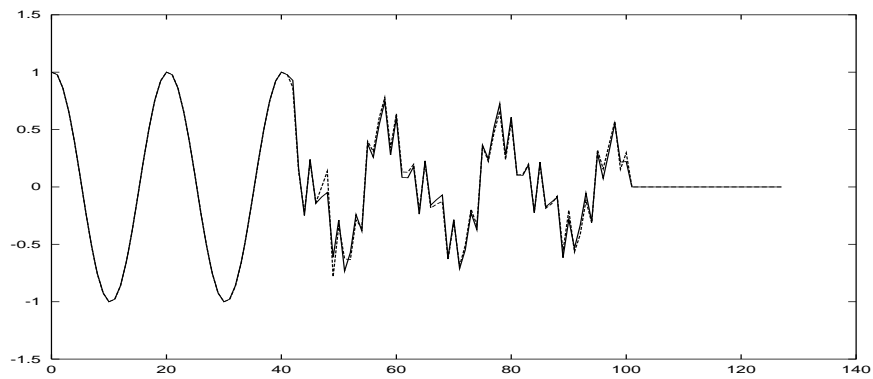


Figura 7.3: Comportamento da reconstrução do sinal “x001”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais

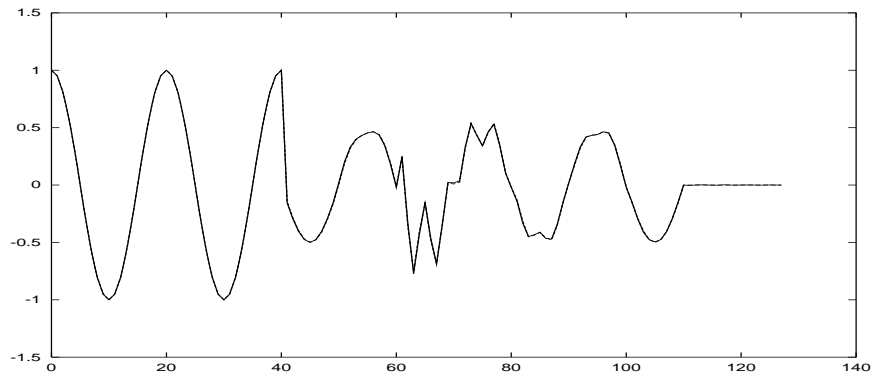


Figura 7.4: Comportamento da reconstrução do sinal “x002”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais

Forçando a decomposição de cada um dos sinais de teste com o mesmo número de estruturas com que foram gerados podemos identificar as estruturas apresentadas na tabela 7.1. Outra alteração realizada no processo é a transferência das frequências no intervalo $[-\pi, 0]$ para $[\pi, 2\pi]$. Na tabela vemos o SNR da reconstrução, o produto de cada estrutura e seus parâmetros. As reconstruções destes sinais usando estes números de componentes, isto é, as utilizadas para gerá-los, podem ser vistos nas figuras 7.3 a 7.10, nestas o sinal original está em linha cheia e o reconstruído em linha pontilhada.

Vê-se que o comportamento e a identificação de estruturas do método proposto é bastante razoável. Os maiores problemas aparecem em mudanças bruscas de fase e quando temos ceifamento do sinal, que podem ser vistos nos sinais “x006” e “x008”, figuras 7.8 e 7.10 respectivamente. Além disso no caso da decomposição do sinal “x003” podemos observar que o método se confunde e identifica batimentos, e

			Parâmetros dos Átomos					
Sinal	SNR dB	Produto	ξ_q	ϕ_q	λ_q	u	t_{0_q}	t_{f_q}
x001	51,664552	4,792623	5,969315	0,082951	0	0	0	42
		2,340427	5,968631	-0,145954	0,009894	49	49	100
		1,247204	3,771522	0,241955	0	41	42	99
		-0,90372	5,656444	1,096688	0,014186	52	52	100
x002	89,015301	4,582636	5,969046	-0,000389	0	0	0	40
		2,956985	5,968820	-1,541265	-0,000041	41	41	109
		0,773385	1,562215	-0,961514	0,094088	61	61	98
		-0,242017	0,944235	-1,301030	-0,000145	50	50	100
x003	53,091694	5,113647	5,969001	1,555935	0	1	1	127
		1,984570	6,026682	1,173911	0	1	1	127
		1,741422	0,365718	-1,531800	0	4	5	126
		-0,375560	5,977884	0,896574	-0,002778	5	5	126
x004	48,476058	7,822828	5,969448	1,536594	-0,000716	0	0	127
		2,145352	0,022461	-0,988498	0,048865	0	0	123
x005	50,575880	8,386618	0,156999	-1,556304	0	1	1	127
		1,891897	3,766125	-1,300815	0,052709	53	53	127
x006	24,195232	-6,042007	0,318646	0,807433	-0,003477	3	3	127
		4,185237	0,255300	-0,379373	-0,016504	2	2	74
		-2,584711	5,903014	-1,490763	0	1	1	76
x007	165,686497	-2,343796	3,769891	-0,313190	0,079967	39	39	127
x008	19,834655	-9,855697	0,157643	1,538846	0	1	1	127

Tabela 7.1: Parâmetros extraídos na decomposição dos sinais de teste.

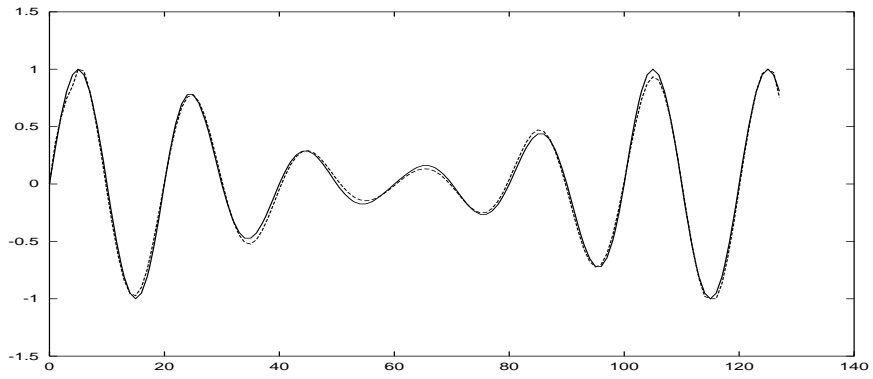


Figura 7.5: Comportamento da reconstrução do sinal “x003”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais

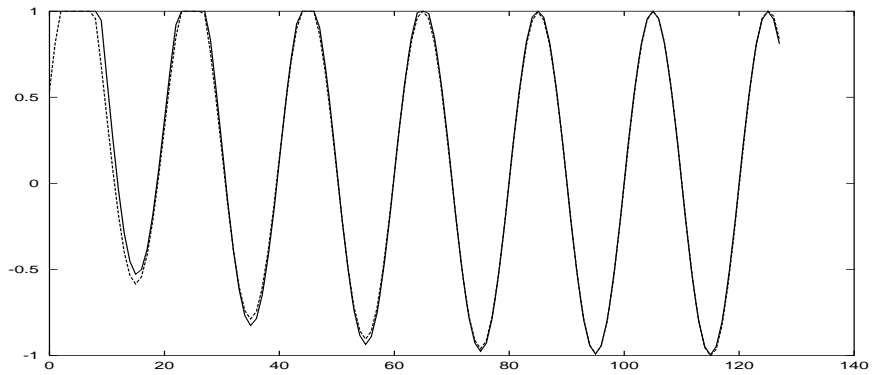


Figura 7.6: Comportamento da reconstrução do sinal “x004”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais

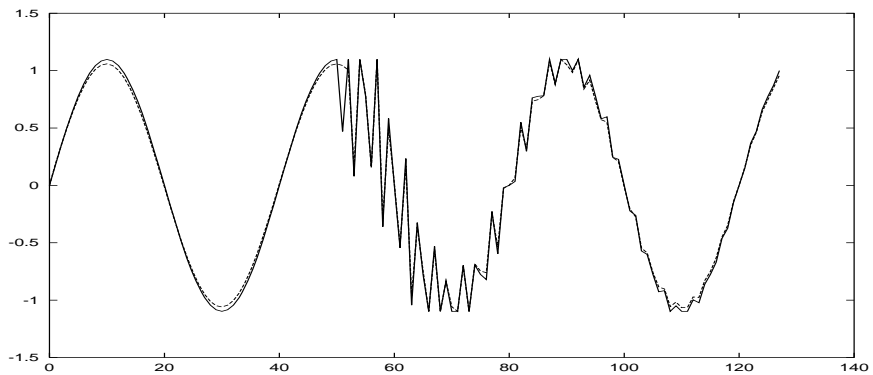


Figura 7.7: Comportamento da reconstrução do sinal “x005”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais

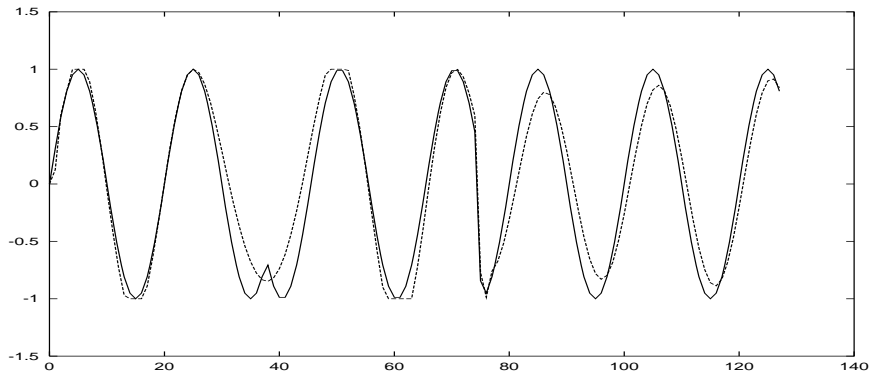


Figura 7.8: Comportamento da reconstrução do sinal “x006”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais

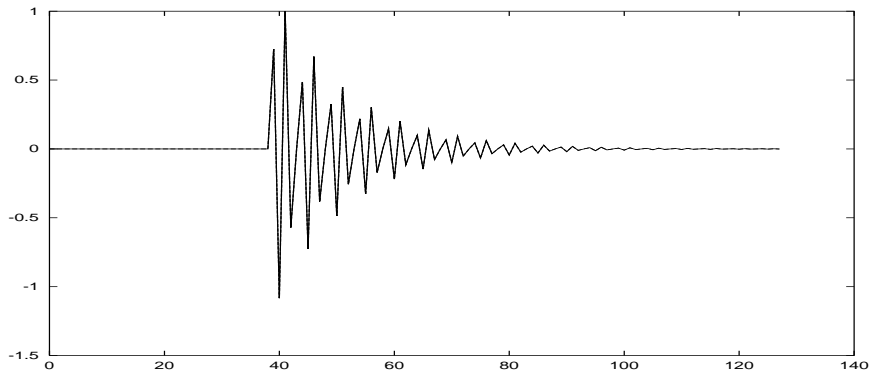


Figura 7.9: Comportamento da reconstrução do sinal “x007”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais

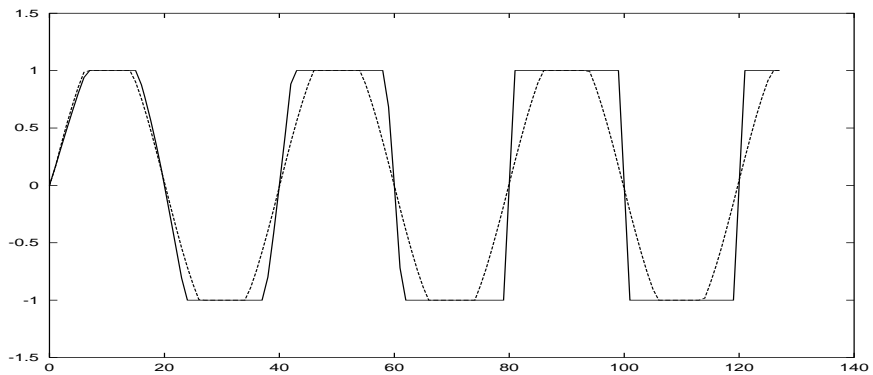


Figura 7.10: Comportamento da reconstrução do sinal “x008”, com o mesmo número de estruturas utilizadas para gerá-lo, com átomos exponenciais

não as componentes utilizadas para gerar o sinal.

7.2.1 Representações Coerentes Com o Dicionário de Exponenciais

Nas decomposições realizadas até aqui indicamos ao algoritmo quantos passos realizar. Podemos entretanto utilizar o conceito de coerência apresentado no capítulo 5 aplicado a este novo dicionário. Ao levantar o comportamento da taxa de aproximação dos sinais de ruído gaussianos para este novo dicionário, observamos também um valor limite para a mesma, como vimos para o dicionário de *Gabor* discreto. Entretanto espera-se que este limiar seja um pouco maior devido à estrutura do dicionário, o que confirmou-se. Estes valores podem ser vistos na tabela 7.2. Com estes podemos realizar então decomposições coerentes utilizando o mesmo conceito utilizado anteriormente para o dicionário de *Gabor*.

Comprimento do Sinal	λ_{med}
64	0.50
128	0.39
256	0.26
431	0.23
512	0.21
1024	0.16
2048	0.13

Tabela 7.2: Taxa de aproximação média para sinais de Ruído com diferentes comprimentos, no dicionário de exponenciais.

Utilizando este critério de coerência os sinais de testes são decompostos conforme a tabela 7.3, para uma média móvel de oito passos, equações (3.41) e (5.5). Nesta tabela vemos também o número de estruturas utilizadas na geração de cada um dos sinais. Repare que o número de estruturas em que o método decompõe os sinais é bem maior que o número de estruturas utilizadas para gerá-lo. Este é um resultado esperado já que o processo de aproximação do sinal só será terminado quando o resíduo tiver um comportamento similar ao do ruído quando aplicado ao

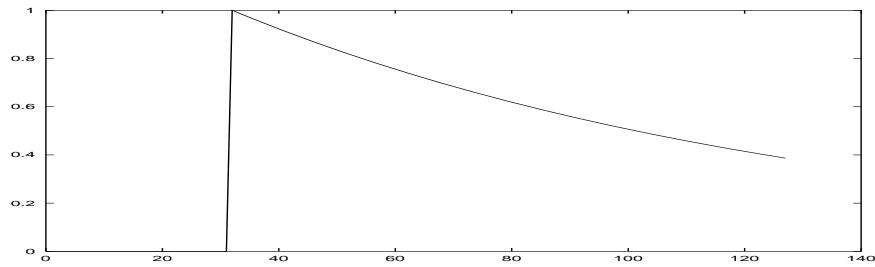
dicionário, e como o processo é uma aproximação isto pode acarretar mais estruturas que o número de estruturas presentes no sinal. Porém ainda assim vemos a capacidade de discernimento do método para identificar estruturas coerentes com o sinal. Veremos depois que grande parte das estruturas identificadas terão amplitude muito pequena e ao quantizarmos as estruturas na seção 7.3 as mesmas desaparecem. Na subseção 7.2.2 veremos como a adição de um ruído Gaussiano de pequena energia pode ser utilizada para tornar esta medida de coerência mais robusta.

Sinal	SNR dB	Num. de Átomos	Estruturas na Geração
x001	141,403556	33	4
x002	147,725354	22	4
x003	106,234750	22	4
x004	117,937663	26	2
x005	134,085335	35	2
x006	56,317937	8	3
x007	244,690827	31	1
x008	134,661621	61	1

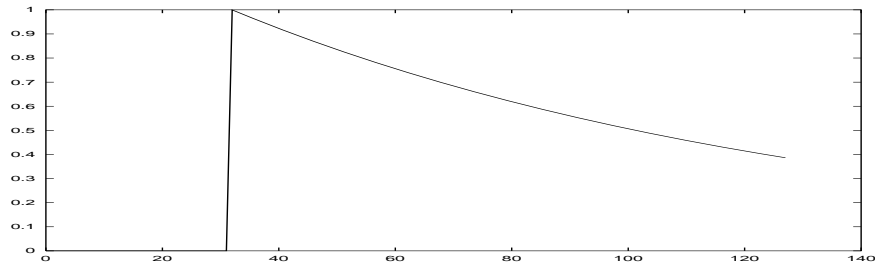
Tabela 7.3: SNR e número de átomos identificados nos sinais sintéticos.

Na tabela 7.3 podemos observar alguns fatores: 1) a incapacidade deste método na representação de sinais com mudança de fase, isto pode ser visto para o caso do sinal “x006”, onde temos duas mudanças de fase, principalmente na cúspide, vemos que além de obtermos uma baixa relação sinal ruído, rapidamente atingimos o critério de coerência, o que indica que rapidamente o resíduo é confundido com ruído neste dicionário; 2) o alto custo na representação de sinais ceifados, são os casos dos sinais “x004” e “x008”, no primeiro como o ceifamento ocorre num suporte limitado sua influência é menor, porém no segundo vemos a demora do algoritmo para atingir a coerência. Este último fator pode ser corrigido implementando o limiar nos passos do algoritmo ao invés de ao final do processo de aproximação, enquanto o problema da fase requer um estudo mais aprofundado.

A identificação de fenômenos através do método proposto é boa, principalmente no caso utópico de fenômenos isolados. Isto pode ser observado não só através do sinal x007, mas também através dos três sinais compostos de uma única estru-



(a) Original



(b) Identificado

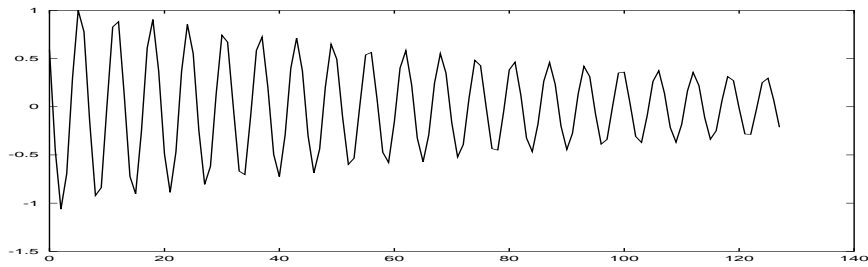
Figura 7.11: Comportamento da identificação do sinal gerado com uma única exponencial.

tura mostrados nas figuras 7.11, 7.12, 7.13. O primeiro é uma exponencial sem modulação, enquanto o segundo e o terceiro são exponenciais moduladas. Nas mesmas figuras podemos ver os sinais reconstruídos/identificados através do método. A reconstrução destes sinais fornece um SNR de 165,94 dB, 159,26 dB e 149,10 dB respectivamente, com uma única estrutura. Devemos observar ainda o coeficiente λ identificado em cada um dos casos: 0,009766, 0,009997 e 0,010015; sendo que todos estes sinais foram gerados com $\lambda = 0,01$, com o que vemos que é realizada uma boa identificação da exponencial.

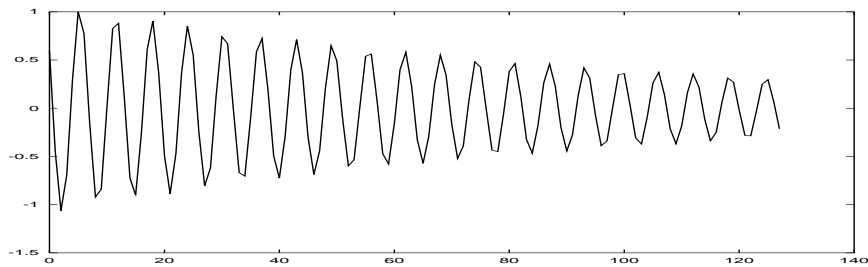
7.2.2 Imunidade ao Ruído

Para avaliar o desempenho deste procedimento podemos somar ruídos de diferentes energias ao sinal original e observar o desempenho do método proposto. Fizemos isto para o sinal “x002” somamos a ele um mesmo ruído gaussiano¹ com diferentes energias e observamos os efeitos causados pela adição destes ruídos. As

¹Um mesma realização de um processo estocástico de ruído gaussiano.



(a) Original

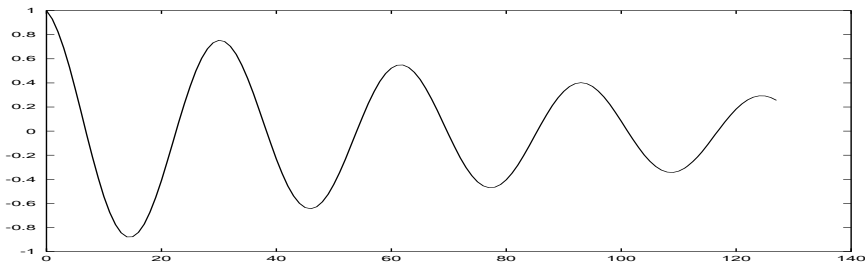


(b) Identificado

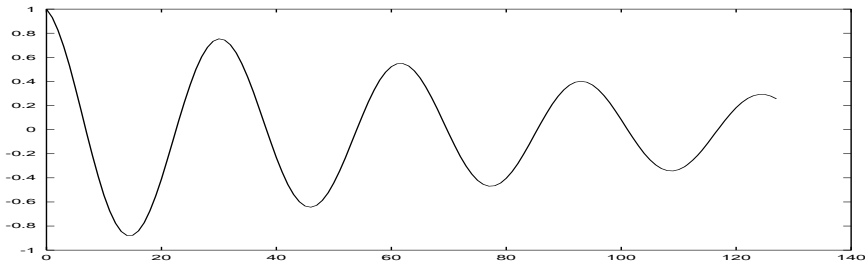
Figura 7.12: Comportamento da identificação do sinal gerado com uma única exponencial.

energias e normas destes ruídos podem ser vistas na tabela 7.4. Nela apresentamos a energia e o módulo dos ruídos, bem como a relação entre a energia do sinal original e o ruído. Podemos ver também o número de estruturas coerentes identificadas e o SNR da reconstrução. Veja que o critério de coerência escolhido funciona muito bem, parando a representação em quatro estruturas até um ruído com energia de aproximadamente 5% da energia do sinal. Há uma pequena confusão na identificação da última estrutura do sinal “x002” mas o desempenho ainda assim é muito bom. Os sinais ruidosos juntamente com os sinais decompostos a partir deles podem ser observados nas figuras indicadas na tabela. Onde novamente o sinal a decompor está em linha cheia e o reconstruído em linha pontilhada.

Esta é uma expansão da idéia de *denoising* [2, 28]. O que vemos é que podemos utilizar um dicionário de exponenciais complexas para representar um sinal elétrico ruidoso e reconstruí-lo com menos ruído, a partir da utilização de uma representação adaptativa através dos *Matching Pursuits* com um dicionário composto de exponenciais complexas.



(a) Original



(b) Identificado

Figura 7.13: Comportamento da identificação do sinal gerado com uma única exponencial.

Ruído					
Energia	Módulo	$\frac{\text{Enrg. Sinal}}{\text{Enrg. Ruído}}$	Num. Estruturas	SNR dB	Figura
0,0809	0,2845	375,81	4	55,175774	7.14
0,3237	0,5689	93,92	4	45,616980	7.15
1,2946	1,1378	23,48	4	34,176268	7.16
2,2756	5,1784	13,36	2	14,923098	7.17

Tabela 7.4: Decomposição do sinal “x002” com adição de ruído.

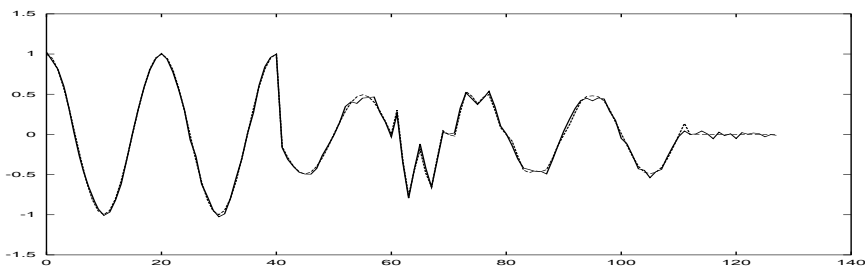


Figura 7.14: Comportamento da identificação do sinal somado ao ruído com razão entre a energia do sinal e a do ruído de 375,81.

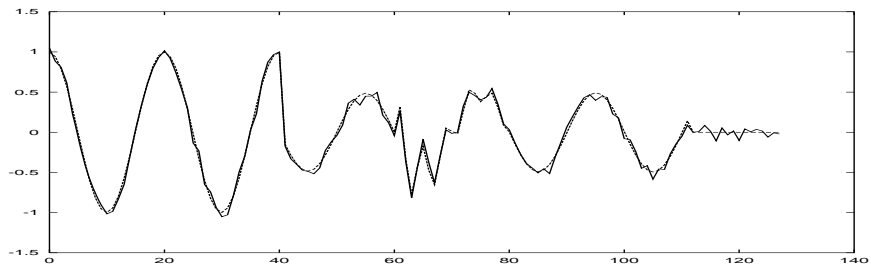


Figura 7.15: Comportamento da identificação do sinal somado ao ruído com razão entre a energia do sinal e a do ruído de 93,92.

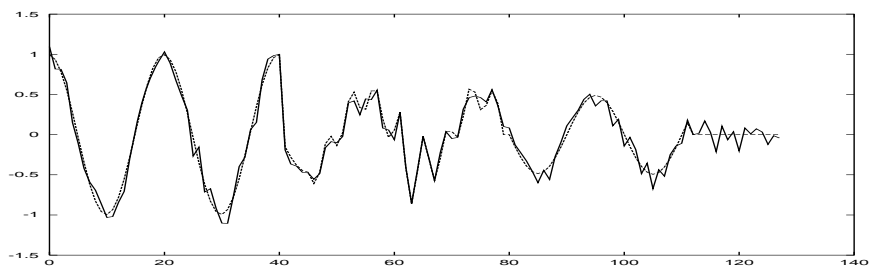


Figura 7.16: Comportamento da identificação do sinal somado ao ruído com razão entre a energia do sinal e a do ruído de 23,48.

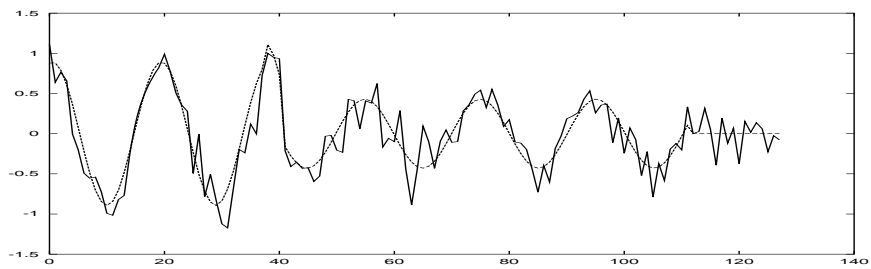


Figura 7.17: Comportamento da identificação do sinal somado ao ruído com razão entre a energia do sinal e a do ruído de 13,36.

7.3 Quantização das Estruturas

Vimos então que o algoritmo proposto é válido para a representação coerente de sinais elétricos, possuindo uma boa capacidade de representação dos sinais do ponto de vista fenomenológico. Porém esta ainda não é uma ferramenta válida para comprimir os sinais a partir destas estruturas, pois necessitamos uma quantidade de bits razoável para o armazenamento dos parâmetros que geram as estruturas.

Para os parâmetros de início e fim das estruturas bem como o parâmetro u , teremos o mesmo número de possibilidades que o comprimento do sinal N , o que pode ser realizado com L bits. Porém para os parâmetros: produto interno, λ , ξ e ϕ ; utilizamos números de 32 bits, o que gera por estrutura um número de $4 \times 32 + 3 \times L$. Repare que para um sinal de 128 amostras com 16 bits, que é o caso dos sinais sintéticos até aqui estudados, teremos uma taxa de compressão de:

$$\frac{128 \times 16}{M(4 \times 32 + 3 \times 7)}, \quad (7.12)$$

onde M é o número de estruturas. Neste caso gastaremos 149 bits por estrutura. Isto nos permite utilizar até 13 estruturas para representar o sinal sem aumentar o número de bits gastos para armazená-lo. Genericamente teremos a taxa dada por:

$$\frac{N \times b}{M(4 \times 32 + 3 \times L)}, \quad (7.13)$$

onde N é o comprimento do sinal, b o número de bits por amostra e L o menor número inteiro maior ou igual a $\log_2 N$.

Podemos ver então que a compressão obtida por esta representação é muito precária. Para resolver este problema podemos quantizar os parâmetros: produto interno, λ , ξ e ϕ . Os parâmetros produto interno, λ e ϕ serão quantizados através de um quantizador escalar. Já a frequência ξ será quantizada em função da frequência de amostragem e da frequência fundamental do sinal elétrico.

Fazemos então quantizadores escalares [29, 30] para λ , ϕ e o produto interno. Estes quantizadores dependerão não só do maior e menor valor que cada um destes três parâmetros podem assumir bem como o número de bits para cada um deles. Sendo w_{max} o maior valor de um destes parâmetros, e w_{min} o menor valor do mesmo parâmetro, definimos então um passo de quantização para o parâmetros w que será dado por:

$$\Delta_{w_q} = \frac{w_{max} - w_{min}}{2^{b_w}}, \quad (7.14)$$

onde b_w é o número de bits permitido para o parâmetro w . Com isto para um dado valor de w que chamaremos w_i teremos seu valor a transmitir/armazenar, i_w , dado por:

$$i_w = \begin{cases} \left\lfloor \frac{w_i + \frac{\Delta_{w_q}}{2}}{\Delta_{w_q}} \right\rfloor, & \text{se } w_i \geq 0 \\ - \left\lfloor \frac{-w_i + \frac{\Delta_{w_q}}{2}}{\Delta_{w_q}} \right\rfloor, & \text{se } w_i < 0 \end{cases}. \quad (7.15)$$

Onde $\lfloor d \rfloor$ é o maior número inteiro menor que d . Desta forma podemos obter o valor a transmitir para w_i . O valor quantizado, w_q , através de:

$$w_q = i_w \times \Delta_{w_q}. \quad (7.16)$$

Então na quantização de um dado parâmetro necessitaremos transmitir além dos índices para reconstrução do parâmetro quantizado o passo de quantização Δ_{w_q} e o número de bits utilizados na quantização do parâmetro. O primeiro faz parte do processo de “desindexação” e o segundo permite entender o *bit-stream* gerado. Teremos então devido ao projeto do quantizador um *over head* a transmitir que pode ser visto na figura 7.18. Salientamos que neste quantizador não enviamos a faixa dinâmica, se esta for enviada podemos obter uma melhora do desempenho do quantizador, porém isto acarreta num custo extra de bits, como estamos interessado em averiguar a possibilidade de quantizar as estruturas este quantizador será suficiente. Além destas informações será necessário enviar também o comprimento do sinal N , o que será feito através dos dois primeiros bytes (com isto podemos codificar sinais de comprimento até 65536). Logo, utilizamos intercaladamente 4 bytes para definir o passo de quantização e um byte para definir o número de bits utilizados para cada um dos parâmetros: produto interno, λ e ϕ (o que fornece a possibilidade de utilizar de 1 a 256 bits por parâmetro). O número de bits a serem utilizados para u , t_f e t_o podem ser obtidos diretamente de N sendo o menor número inteiro maior ou igual a $\log_2 N$. Desta forma já estaremos de posse do modelo dos quantizadores do produto interno, λ , ϕ , u , t_o e t_f bem como do número de bits utilizados na quantização de cada um deles.

Resta-nos agora, quantizar a frequência ξ dos parâmetros de geração dos átomos exponenciais. Inicialmente projetamos o quantizador da frequência como os anteriores, porém percebemos que este possuía três problemas:

1. Não considerava a relação entre os harmônicos do modelo de sinal elétrico;

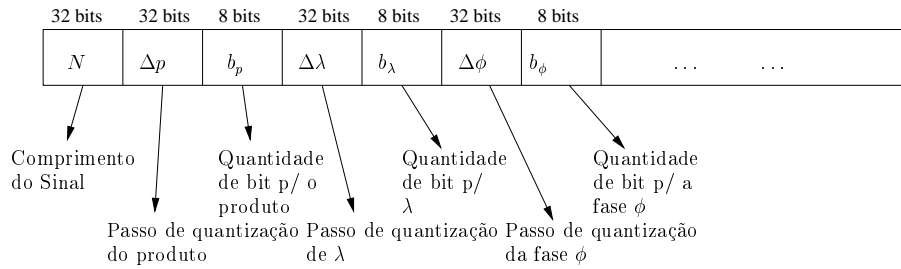


Figura 7.18: Cabeçalho da compressão de um sinal informando o seu comprimento, e os passos de quantização de: produto interno, λ e ϕ e o número de bits utilizados na quantização de cada um deles.

2. Necessitava um grande número de bits para obter a precisão da relação da frequência de amostragem (f_a) com a frequência da fundamental (f_f):

$$r_f = \frac{f_a}{f_f}; \quad (7.17)$$

3. Não levava em consideração que ao contrário dos outros parâmetros a frequência ξ está limitada em $[0, 2\pi]$.

Fazemos então o cálculo do passo de quantização de ξ a partir de r_f . Podemos então calcular o passo de quantização da frequência:

$$\Delta_{\xi_q} = \frac{2\pi}{r_f}. \quad (7.18)$$

Cabe uma observação, em alguns casos, esta escolha do passo de quantização não utiliza de forma ótima o número de bits, por exemplo: caso $r_f = 60$ necessitaremos 7 bits para codificar a frequência, desta forma poderíamos utilizar $r_f = 120$ que ainda assim utilizaríamos o mesmo número de bits e o passo seria dividido por dois, permitindo utilizar os mesmos valores para a frequência quantizada. Levamos em consideração esta observação na hora de calcular o número de níveis ou valores permitidos para a frequência maximizando assim os bits utilizados e mantendo a relação de Δ_{ξ_q} e r_f que será dada agora por:

$$\Delta_{\xi_q} = \frac{2\pi}{Pr_f}, \quad (7.19)$$

onde P é um número inteiro positivo. Podemos então obter o índice da frequência a armazenar/transmitir através:

$$i_\xi = \left\lfloor \frac{\xi_i + \frac{\Delta_{\xi_q}}{2}}{\Delta_{\xi_q}} \right\rfloor, \quad (7.20)$$

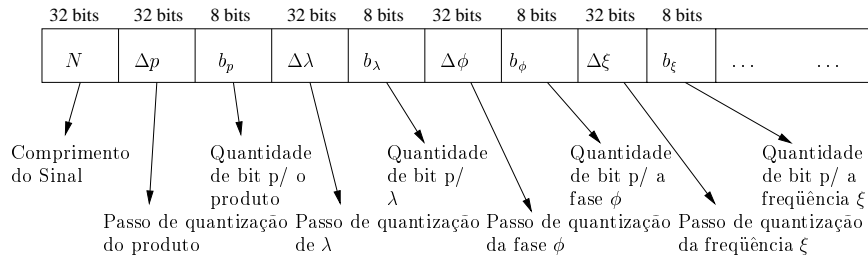


Figura 7.19: Cabeçalho da compressão de um sinal.

e a freqüência quantizada é obtida através de:

$$\xi_q = i_\xi \times \Delta_{\xi_q}. \quad (7.21)$$

Para os parâmetros u , t_o e t_f usaremos L bits tal que L seja o menor número inteiro maior ou igual a $\log_2 N$. Logo esta informação já é transmitida ao informarmos N . Com isto teremos que o cabeçalho com a informação para a decodificação será dado pelo mostrado na figura 7.19. Teremos então um gasto extra de bits dado por:

$$b_{extra} = 5 \times 32 + 4 \times 8 = 192 \text{ bits}. \quad (7.22)$$

Podemos agora armazenar/transmitir as estruturas utilizando uma quantidade de b_p bits para o produto, b_λ bits para λ , b_ϕ para a fase ϕ , b_ξ para a freqüência ξ e L para u , t_o e t_f . Teremos por estrutura um gasto total de b_e bits dado por:

$$b_e = b_p + b_\lambda + b_\phi + b_\xi + 3L. \quad (7.23)$$

Teremos então que o número total de bits gastos na compressão de um sinal será dado por:

$$b_{comp} = b_{extra} + M \times b_e, \quad (7.24)$$

onde M é o número de estruturas utilizadas.

Se o sinal original possui N amostras e está quantizado com b_o bits por amostra, seu tamanho em bits será:

$$b_{orig} = N \times b_o. \quad (7.25)$$

com isto pode-se calcular a taxa de compressão através de:

$$r_{comp} = \frac{\text{Quant. bits sinal original}}{\text{Quant. bits do sinal comprimido}} = \frac{b_{orig}}{b_{comp}}. \quad (7.26)$$

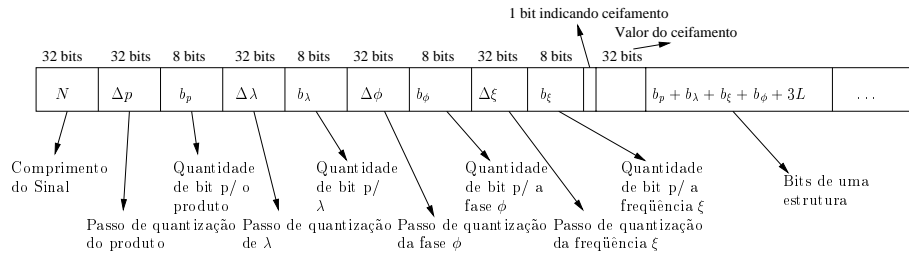


Figura 7.20: Cabeçalho completo da compressão de um sinal com uma estrutura.

Falta-nos informar se há ceifamento ou não no sinal. Isto pode ser feito utilizando um bit para indicar se há ou não ceifamento e 32 bits para indicar o nível deste caso exista. Este esquema pode ser visto na figura 7.20, onde mostramos também como serão enviadas as estruturas, repare que os 32 bits do ceifamento só necessitam ser gastos caso este ocorra. Neste caso teremos para o cabeçalho:

$$b_{extra} = \begin{cases} 5 \times 32 + 4 \times 8 + 1 = 193 \text{ bits, caso não haja ceifamento;} \\ 5 \times 32 + 4 \times 8 + 1 + 32 = 225 \text{ bits, caso haja ceifamento;} \end{cases} \quad (7.27)$$

7.3.1 Resultados

Vejamos agora os resultados obtidos para os sinais até aqui estudados quando realizamos a quantização das estruturas. Estes resultados são obtidos realizando a decomposição dos sinais através dos *Matching Pursuits* e realizando a procura do átomo exponencial apresentada na seção 7.2, até atingirmos o limiar de coerência da seção 7.2.1. Após isto as estruturas são quantizadas. Repare que esta quantização irá mapear diversos produtos internos pequenos em zero eliminando diversas estruturas de baixa energia. Os resultados deste processo podem ser vistos na tabela 7.5 para alguns casos dos sinais x001 e x002, onde apresentamos o SNR para a taxa de compressão obtida e o número de estruturas presentes após a quantização (Num. str.). Podemos observar que apesar do sinal ter comprimento pequeno obtemos taxas de compressão razoáveis. Repare que a quantização do produto interno elimina estruturas da representação. Perceba que a influência do número de bits utilizados na quantização de um parâmetro está atrelada também aos valores máximos e mínimos que este parâmetro pode assumir, que por vezes pertencem a estruturas eliminadas devido à quantização do produto interno, isto indica que podemos vir a pensar em um quantizador que leve em consideração os produtos internos nas estruturas

no momento de projetar os quantizadores de cada parâmetro, e não só o valor do parâmetro. Veja ainda que a quantização do produto interno com 3 bits fornece o número correto de estruturas presentes no sinal x001, porém no caso do sinal x002 ocorre um erro eliminando uma das estruturas. Isto pode ser explicado através dos valores dos produtos internos apresentados na tabela 7.1, já que a quarta estrutura possui um produto interno pequeno que quando quantizado torna-se zero. Já com o sinal x001 com 4 bits obtemos uma estrutura a mais que o número de estruturas presentes no sinal onde o argumento para que isto ocorra é semelhante ao anterior, porém com 3 bits o número de estruturas já é o correto.

Quant. bits			Sinal x001			Sinal x002		
b_p	b_λ	b_ϕ	Num. str.	SNR dB	r_{comp}	Num. str.	SNR dB	r_{comp}
4	4	4	5	35.528235	5.291990	4	47.484682	5.885057
4	4	3	5	34.769586	5.361257	4	45.977025	5.953488
4	3	3	5	34.769586	5.432361	4	45.591825	6.023530
4	3	2	5	31.976769	5.505376	4	34.464519	6.095238
4	2	2	5	31.976769	5.580381	4	34.464519	6.168674
3	2	2	4	34.637216	6.243902	3	35.896185	6.965986
3	1	1	4	35.459774	6.400000	3	38.493357	7.111111
2	2	2	4	27.783441	6.320988	3	29.815688	7.037801

Tabela 7.5: SNR e Taxa de compressão obtidas com os sinais x001 e x002.

Na figura 7.21 podemos ver o desempenho na reconstrução do sinal “x005” após a quantização de suas estruturas coerentes, decomposição conforme a seção 7.2.1. Fizemos a quantização das estruturas permitindo um máximo de 7 bits e um mínimo de 1 bit para cada um dos parâmetros: produto interno, ϕ e λ ; enquanto para os outros parâmetros a escolha do número de bits é automática conforme apresentado. Nesta figura apresentamos o SNR em função da taxa de compressão ao variarmos os números de bits para o produto interno, ϕ e λ como mencionado. Além disso mostramos os setores que resultam da utilização de um dado número de bits para o produto. Vemos nesta figura que a maior influência na taxa de compressão é do número de bits utilizados para o produto interno ou amplitude, já que este irá influenciar no número de estruturas a serem utilizadas na representação do

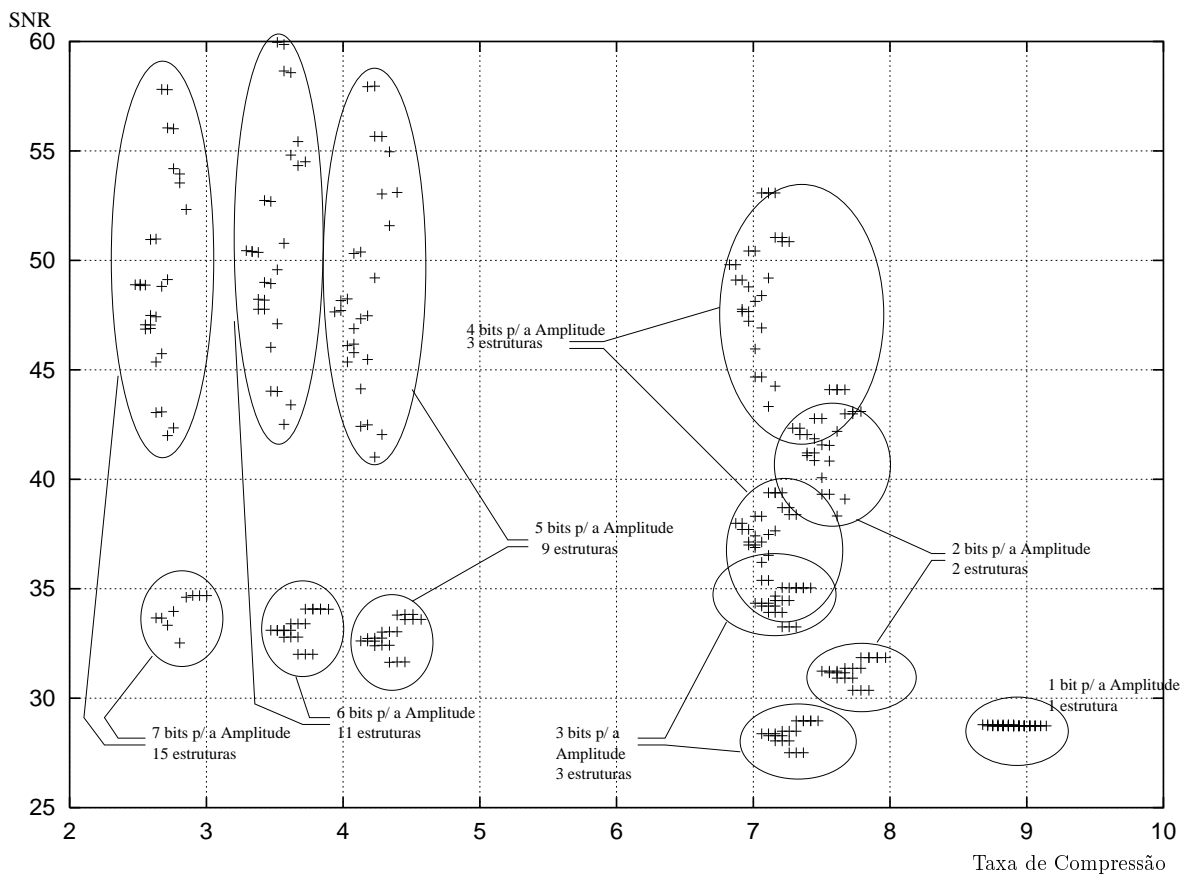


Figura 7.21: SNR \times Taxa de compressão para a quantização do sinal x005.

sinal. Pode-se observar ainda que quanto maior o número de bits utilizados para a amplitude, maior na média, o SNR. Mas ainda assim, para uma quantidade específica de bits para a amplitude, a quantidade de bits utilizada para a fase, ϕ , e o decaimento da exponencial, λ , possui relevância significativa. Resta saber quem possui maior influência, a quantidade de bits utilizadas para λ ou a usada para ϕ ? Quando escolhemos alguns pontos deste gráfico vemos que utilizando 2 bits para a amplitude e os casos: com 3 bits para λ e 4 para ϕ obtemos um SNR de 31,37 dB e com 4 bits para λ e 3 para ϕ 39,09 dB, desta forma mantemos a mesma taxa de compressão e podemos estudar a diferença na relação sinal ruído. No caso deste sinal, este é o comportamento na média: mais bits para λ fornecem um desempenho melhor que mais bits para ϕ . Porém deve-se observar que este é um comportamento extremamente dependente do sinal. Sinais onde a composição dos mesmos é de exponenciais reais moduladas necessitam mais bits para λ , sinais onde a principal característica é a variação de fase necessitam mais bits para ϕ .

Na figura 7.22 podemos ver os melhores quantizadores encontrados para a

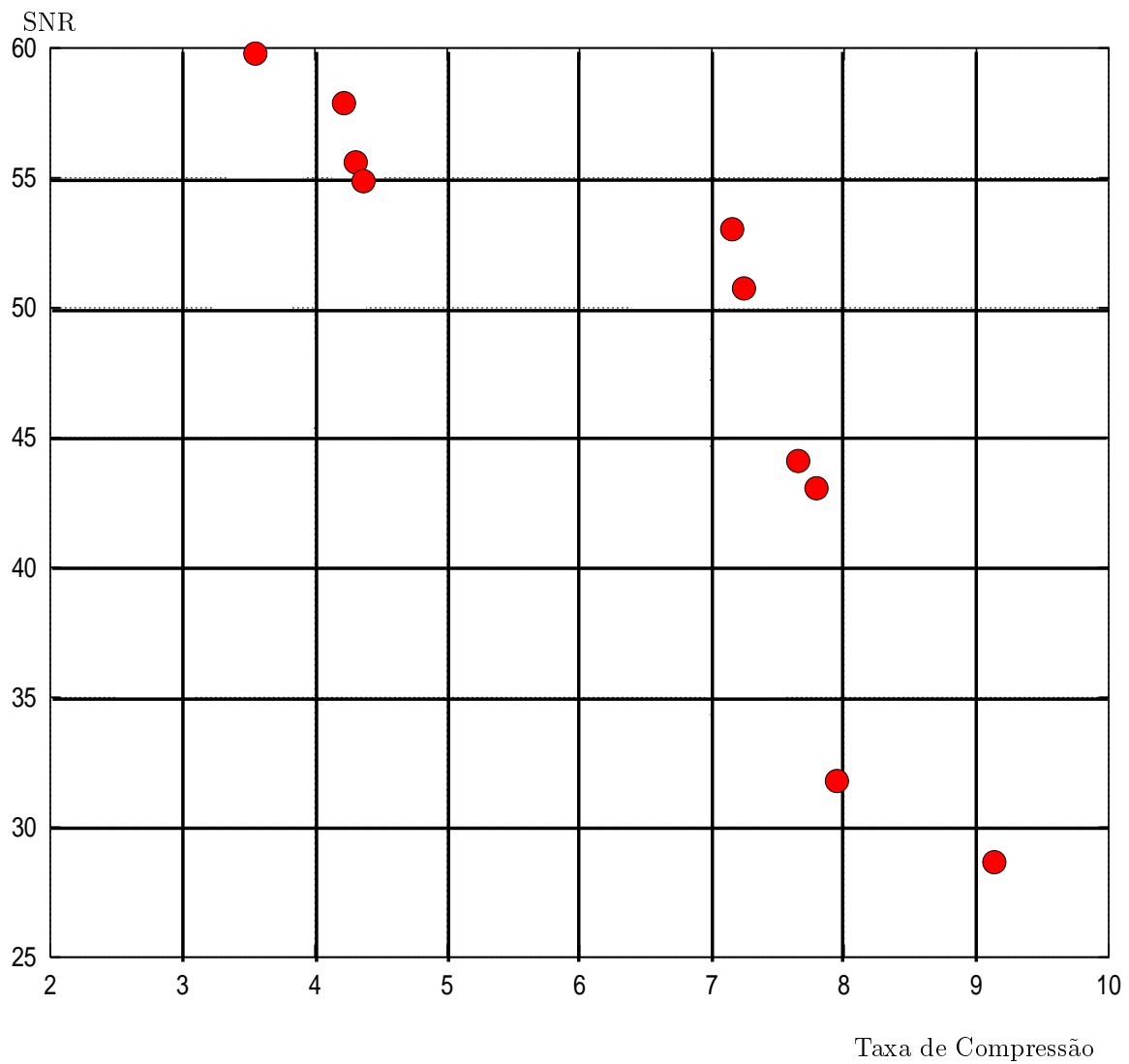


Figura 7.22: Melhores quantizadores para as estruturas do sinal x005.

quantização das estruturas do sinal “x005”. São aqueles com maior relação sinal ruído, SNR, e maior taxa de compressão.

7.4 Aplicação a Sinais Reais

Aplicamos o algoritmo de decomposição através de átomos exponenciais apresentado até aqui junto à quantização das estruturas aos sinais reais apresentados na seção A.2. Na tabela 7.6 podemos ver o resultado da compressão destes sinais, inclusive o número de bits por amostra utilizados na codificação do sinal (bps). Fizemos uma pequena modificação no algoritmo: além do critério de parada baseado na taxa de aproximação média interrompemos o processo quando identificamos uma

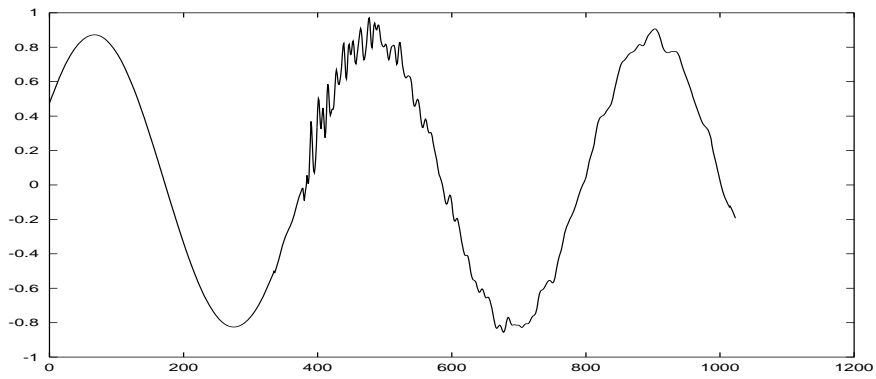


Figura 7.23: Sinal R001 reconstruído.

estutura cuja norma é menor que $\frac{1}{N}$ da norma do sinal, sendo N o comprimento do sinal. Este opera simplesmente como um critério de energia e não de coerência, de forma a parar a codificação de sinais pouco ruidosos. Os sinais reconstruídos podem ser vistos nas figuras 7.23 para o sinal R001, para o sinal R002 na figura 7.24, na figura 7.25 para o sinal R003, e na figura 7.26 para o sinal R004.

Sinal	Quant. bits				N. Estr.	SNR dB	r_{comp}	bps
	b_p	b_λ	b_ϕ	b_ξ				
R001	6	6	6	9	10	54,064907	21,222797	0,754
R002	5	6	6	12	9	55,824042	22,382513	0,715
R003	5	5	5	13	12	49,491845	19,117853	0,837
R004	5	12	5	12	10	54,196683	19,458433	0,822

Tabela 7.6: SNR e Taxa de compressão obtidas para os sinais reais, quantidade de bits utilizados por parâmetro e figura do sinal reconstruído.

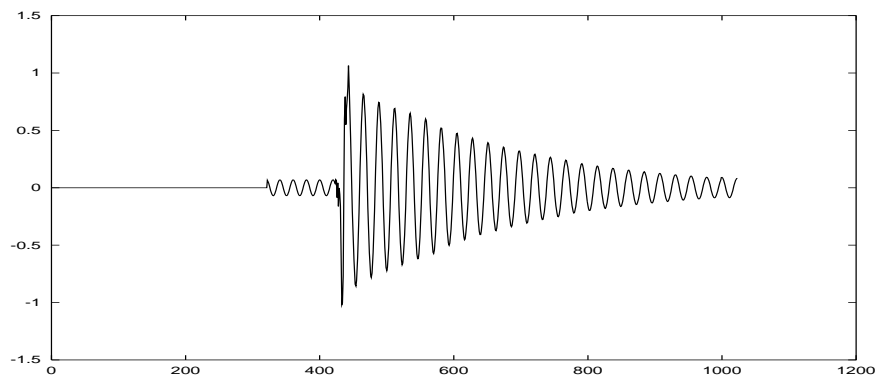


Figura 7.24: Sinal R002 reconstruído.

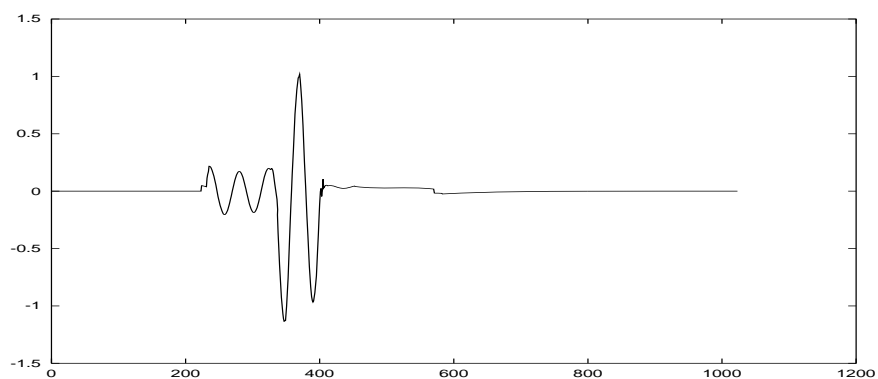


Figura 7.25: Sinal R003 reconstruído.

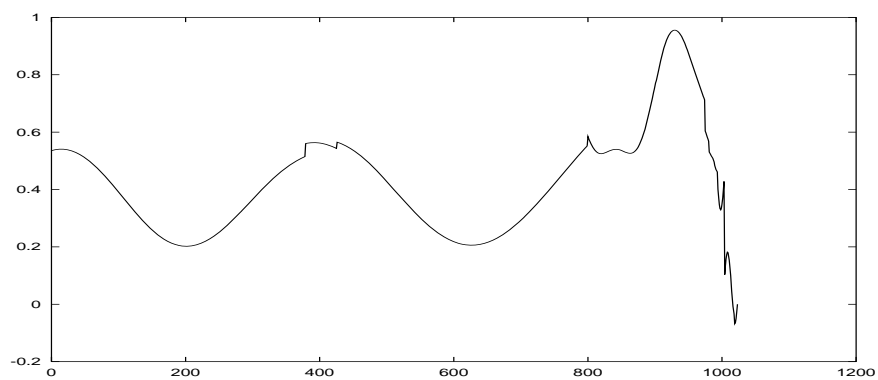


Figura 7.26: Sinal R004 reconstruído.

Capítulo 8

Conclusões

Neste trabalho vimos a decomposição coerente de sinais, em especial de sinais elétricos, utilizando exponenciais moduladas. Vimos então que, a partir de um *frame*, podemos obter uma representação de um sinal e que realizando uma aproximação através de passos, sucessiva, com o algoritmo de *Matching Pursuits*, podemos obter não só uma representação coerente, mas também compacta. A vantagem destas representações em relação a outros métodos é que, além de nos permitir representar o sinal, a mesma permite analisar as características do mesmo fenomenologicamente. Não só reconstruímos o sinal mas além disso temos a possibilidade entender os fenômenos presentes no mesmo. Porém, mais do que elucidar, este trabalho levantou muitas questões. Neste capítulo veremos então três aspectos:

- As conclusões do trabalho desenvolvido;
- As questões levantadas por este trabalho;
- Possíveis trabalhos a serem realizados sobre o que aqui foi estudado.

Tomamos a liberdade de tratar estes três aspectos conjuntamente utilizando como elo de ligação o assunto no qual eles se encaixam e não os aspectos por separados. Ao final colocamos por separado o que acreditamos serem as contribuições deste trabalho.

8.1 Representações Redundantes e Adaptativas

Neste trabalho vimos não só a viabilidade teórica da utilização do *Matching Pursuits* na representação de sinais, mas também diversos de seus aspectos práticos. Vimos que este algoritmo funciona, porém sua falha está no alto custo computacional. Algoritmos mais rápidos existem, porém os mesmos possuem o problema do alto custo de memória. Vimos também que o projeto das funções ou átomos a utilizar nestes procedimentos possui grande influência sobre o resultado da representação.

Por outro lado observou-se que a natureza voraz dos *Matching Pursuits* não é adequada para identificar padrões e analisar sinais. Isto se deve à confusão que a maximização do produto interno faz na representação agrupando fenômenos distintos em uma mesma estrutura. Um procedimento que utilizamos neste trabalho para a identificação de padrões, foi a minimização do erro relativo, página 97. Talvez a minimização do erro relativo pudesse ser utilizada para realizar os passos do *Matching Pursuits*. Isto pode fornecer-nos uma nova representação onde mudamos completamente o critério e este novo critério favorece o casamento de estruturas na região. Devemos ressaltar que neste novo critério não poderíamos permitir impulsos no dicionário, pois estas funções minimizariam o erro, já que um impulso com a amplitude da amostra do sinal forneceria um erro relativo nulo. Vimos que este critério aliado ao critério do produto interno, como um segundo passo, forneceu bons resultados na identificação das estruturas de sinais elétricos. Isto se deve ao fato que este critério na verdade é a minimização da norma l_2 , base dos métodos dos *frames*.

Quanto aos objetivos desejados para uma representação adaptativa, apresentados no capítulo 3, não conseguimos atingir o de velocidade. Esta é uma direção a seguir tentando acelerar o método proposto.

8.2 Dicionário de Exponenciais

O dicionário de exponenciais fornece um bom *frame* para a representação de sinais elétricos. Acreditamos também que este poderá vir a ser usado futuramente na representação de sinais de voz e áudio [7]. Em [7] há a indicação de uma metodologia para acelerar o casamento com exponenciais baseada em filtros IIR [6, 31]. Porém, além da questão de como acelerar o processo de *Matching Pursuits* com este

dicionário, temos que observar a questão de como amostrá-lo de forma a garantir que geramos um *frame*. Garantimos que o dicionário de exponenciais é um *frame* já que ele comporta além da base de exponenciais complexas, base de *Fourier*, uma base de impulsos, e ainda uma série de exponencias. Porém não encontramos uma maneira de amostrar o parâmetro λ das exponenciais reais de forma a garantir um dicionário amostrado que forma um *frame* como o dicionário de *Gabor*. O que temos é um dicionário de átomos gerados a partir de parâmetros contínuos. Esta é uma questão que resta ser estudada. Já que não há uma maneira óbvia de garantir uma divisão do plano tempo-freqüência a partir de exponenciais moduladas.

Porém, de certa forma, a partir da modelagem de um sinal elétrico por uma série de exponenciais, obtemos uma boa representação do mesmo, sem realizar cálculos complicados para a obtenção dos átomos ideais [32]. Outro método visto para a utilização de *frames* específicos foi o treinamento [33, 34], que também não necessitamos realizar.

Vimos então que o dicionário de átomos exponenciais fornece um bom modelo para a representação de sinais elétricos.

8.3 Quanto ao Critério de Coerência

O critério de coerência utilizado neste trabalho realmente foi capaz de finalizar o processo de aproximação quando o sinal se assemelha a um processo de ruído branco Gaussiano. Isto pode ser especialmente notado na subseção 7.2.2. Além disso, utilizando este critério, temos como retirar ruído do sinal.

Devemos reparar que esta medida de coerência se torna tanto mais útil quanto mais ruidoso é o sinal. Para um sinal com ruído muito pequeno o algoritmo necessita um número muito grande de iterações para alcançar o limiar de coerência, enquanto em sinais com um ruído de maior energia este é alcançado mais rapidamente.

Este critério pode ser usado também na troca de dicionários quando são realizados *Matching Pursuits* com diferentes *frames* [33]. Desta forma temos como identificar automaticamente em que ponto da aproximação trocar de *frame* e continuar os *Matching Pursuits* com um novo *frame*.

Vimos que uma boa forma de tornar o processo de identificação das estru-

turas mais robusto e automático é através da adição de um ruído Gaussiano de pequena energia ao sinal. Desta forma a coerência é atingida mais rapidamente e não necessitamos representar estruturas de pequena energia resultantes de pequenas diferenças entre os parâmetros das estruturas identificadas e as estruturas do sinal.

8.4 Quanto às Estruturas

O método é capaz de identificar as exponencias que compõem o sinal, sendo portanto capaz também de identificar senóides. Desta forma as próprias estruturas do dicionário possuem relação com as componentes de sinais elétricos. Sendo assim as estruturas que utilizamos para representar o sinal são condizentes com aquelas presentes em sinais elétricos.

Entretanto o método falha quando se depara com variações de fase que podem ser observadas nos sinais x006 e R004. Nestes casos acreditamos ser possível identificar esta variação de fase após a identificação de um átomo observando a taxa de variação do produto interno (ou seja através da derivada do produto interno que deverá mudar devido à variação do sinal a representar), ou utilizando um processo de identificação de fase similar ao do PSK¹, ou ainda quantizando a frequência, futuramente pretendemos estudar este assunto.

Outro tópico é a melhora da representação de sinais ceifados. Identificando o ceifamento e se no passo em questão o sinal reconstruído será ceifado, podemos aumentar o valor absoluto do produto interno e desta forma minimizar o SNR em dado passo, obtendo uma aproximação mais rápida e mais coerente.

8.5 Compressão de Sinais Elétricos

Ao quantizarmos as estruturas do dicionário de exponenciais obtivemos bons resultados tanto na medida objetiva de distorção o SNR, quanto visualmente nos sinais reconstruídos a partir das estruturas quantizadas.

Para a frequência, e os parâmetros temporais, conseguimos desenvolver processos automáticos de escolha da quantidade de bits a utilizar na quantização. Porém

¹Phase Shift Keying.

resta-nos desenvolver processos semelhantes para os outros parâmetros dos átomos. Acreditamos que isto seja factível através da análise da variância de cada um dos parâmetros.

Em termos da compressão restam-nos então melhorar os quantizadores dos parâmetros produto interno, λ e ϕ , apresentados na seção 7.3, assim como chegar a um critério de alocação de bits através de uma análise subjetiva. Outra questão a estudar na quantização das estruturas é a possível influência da quantização do produto interno no projeto dos quantizadores dos outros parâmetros. Para isto quantizaríamos primeiro o produto interno. A partir das estruturas que permanecem após esta quantização projetamos os quantizadores dos outros parâmetros.

8.6 Nossas Contribuições

Resumindo podemos colocar que as principais contribuições deste trabalho são:

- Estudo de um critério de coerência de forma a tornar uma decomposição de um sinal num dicionário coerente com o mesmo.

Desta forma levantamos valores para taxas de aproximação que definem quando a decomposição de um sinal deixa de ser coerente com o dicionário. Ou seja quando o resíduo da decomposição passa a ser interpretado pelo algoritmo de *Matching Pursuits* como um ruído Gaussiano. Este estudo foi realizado para diversos dicionários. Além disso este critério pode ser utilizado para realizar *denoising* a partir de estruturas coerentes.

- Eliminação de ruído de pré-eco e pós-eco na representação de sinais através do *Matching Pursuits* com o dicionário de *Gabor*.

Vimos como eliminar o ruído de pré-eco e pós-eco que surgem na decomposição através do *Matching Pursuits* com o dicionário de *Gabor*. Esta eliminação foi feita considerando, além do produto interno, o resíduo do passo. Utilizamos este critério em uma decomposição com exponenciais na qual obtivemos bons resultados, eliminando os mesmos problemas.

- Identificação de exponenciais em sinais.

Vimos que podemos identificar exponenciais em um sinal, identificando portanto o comportamento do mesmo.

- Decomposição de sinais através de exponenciais.

A identificação acima foi utilizada para decompor um sinal através de exponenciais e vimos que os resultados deste processo são muito bons.

- Investigação da possibilidade de quantização dos elementos geradores de estruturas em dicionários de parâmetros “contínuos”.

Estudamos também a possibilidade de quantizar os parâmetros das exponenciais identificadas. Apesar de termos utilizado uma abordagem bastante simples vimos que é possível quantizar os mesmos e obter bons resultados

- Compressão coerente de sinais elétricos reais com altas taxas de compressão.

Por fim reunimos os itens acima num método eficiente de compressão de sinais elétricos com altas taxas de compressão. Este ainda pode ser melhorado utilizando de maneira adequada o ceifamento, estudando melhor como identificar a variação de fase e sua representação, e melhorando a quantização dos parâmetros dos átomos exponenciais.

Apêndice A

Sinais Estudados

Neste capítulo apresentamos os diferentes sinais utilizados nesta tese. Foram colocados por separado de forma a poder realizar uma descrição objetiva dos mesmos. Além da forma de onda dos sinais veremos também as suas características.

A.1 Sinais Sintéticos

Os primeiros sinais estudados neste trabalho foram retirados de *Rodrigues* [19]. São oito sinais com características de sinais de redes de transmissão de energia elétrica. Estes sinais foram utilizados inicialmente para verificar a validade do método implementado e poder compará-lo com o trabalho original de [19].

Estes sinais são construídos sinteticamente através de um modelo de [19], equação (7.1), onde os sinais são gerados através de exponenciais moduladas por senóides com diferentes regiões de suporte. Os sinais assim obtidos podem ser vistos a seguir nas figuras A.1 a A.8. Todas as curvas possuem 128 amostras e foram obtidas a uma taxa de 1200 amostras por segundo. Os valores dos parâmetros utilizados para gerar os sinais podem ser vistos na tabela A.1, n_{0_q} e n_{f_q} são respectivamente as amostras de início e fim das componentes, e são obtidas a partir do tempo e da taxa de amostragem. Na última coluna temos a indicação se o sinal está ceifado e o nível onde o sinal está ceifado. Este é um fenômeno importante de observar, já que ocorre com frequência no registro de valores em sistemas elétricos.

Sinal	$F(\text{Hz})$	A_q	k_q	$\phi_q(\text{graus})$	λ_q	t_{0_q}	t_{f_q}	n_{0_q}	n_{f_q}	Ceifado
x001	60	1.000	1	85	0	0	0.0350	0	42	não
		0.450	1	135	0	0.0350	0.0830	42	100	
		0.250	2	270	0.005	0.0350	0.0750	42	90	
		0.250	8	39	0.001	0.0355	0.0830	43	100	
x002	60	1.000	1	90	0	0	0.0333	0	40	não
		0.500	1	180	0	0.0333	0.0017	40	110	
		0.050	3	23	0	0.0417	0.0833	50	100	
		0.500	5	0	0.100	0.0500	0.1059	60	127	
x003	60	1.000	1	0	0	0	0.0167	0	20	não
		1.000	1	0	0.050	0.0167	0.0500	20	60	
		0.125	1	0	-0.050	0.0500	0.0833	60	100	
		1.000	1	0	0.100	0.0833	0.1059	0	127	
x004	60	1.000	1	0	0	0	0.1059	0	127	em 1
		1.000	0	90	0.050	0	0.1059	0	127	
x005	30	1.000	1	0	0	0	0.1059	0	127	em 1
		1.000	8	0	0.050	0	0.1059	50	127	
x006	60	1.000	1	0	0	0	0.0312	0	37	não
		1.000	1	225	0	0.0312	0.0625	37	75	
		1.000	1	270	0	0.0625	0.01059	75	127	
x007	60	1.000	8	0	0.080	0.0312	0.1059	38	127	não
x008	30	1.000	1	0	-0.025	0	0.1059	0	127	em 1

Tabela A.1: Parâmetros utilizados na geração do sinais de testes através do modelo da equação 7.1.

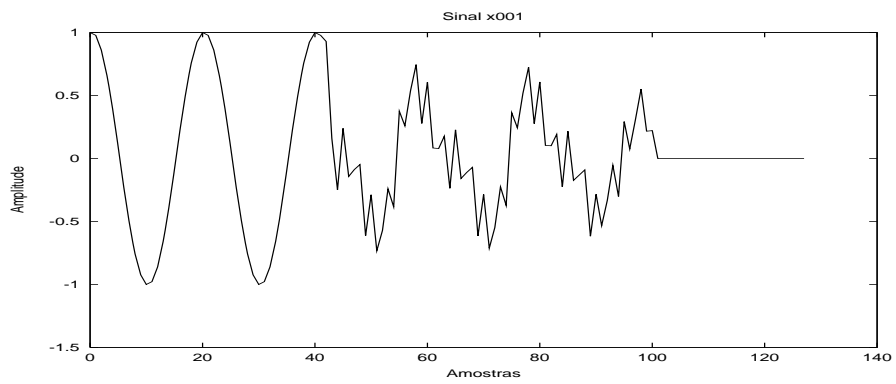


Figura A.1: Sinal x001

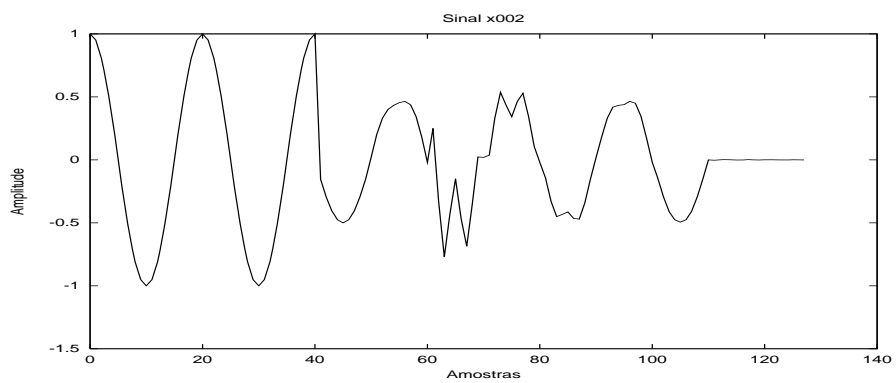


Figura A.2: Sinal x002

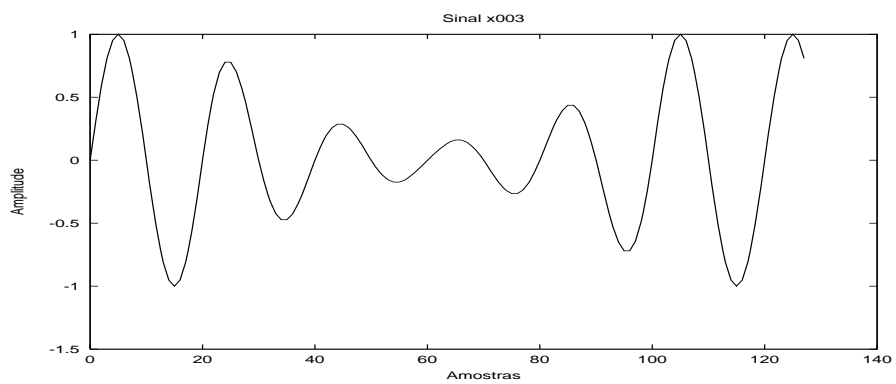


Figura A.3: Sinal x003

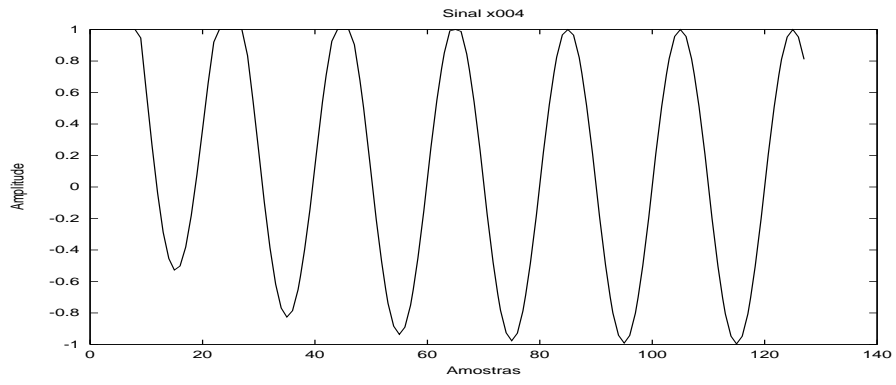


Figura A.4: Sinal x004

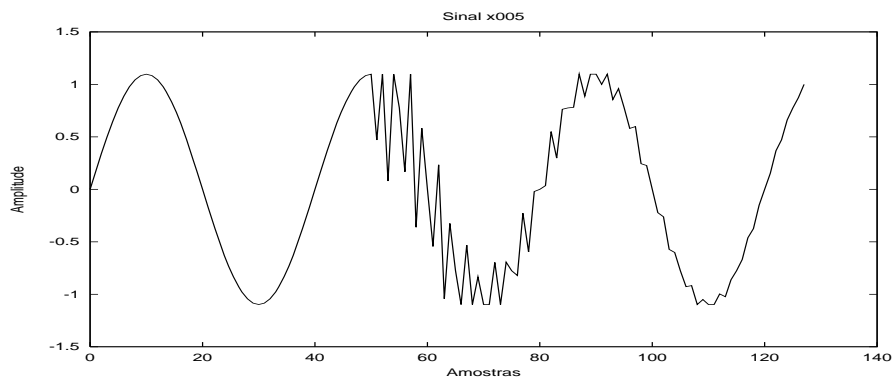


Figura A.5: Sinal x005

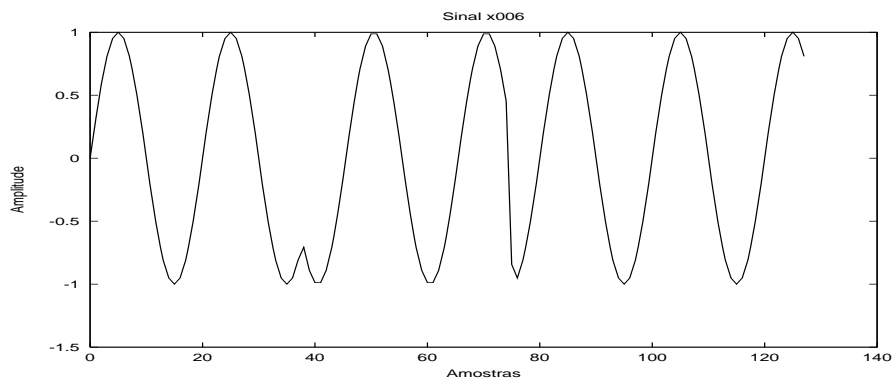


Figura A.6: Sinal x006

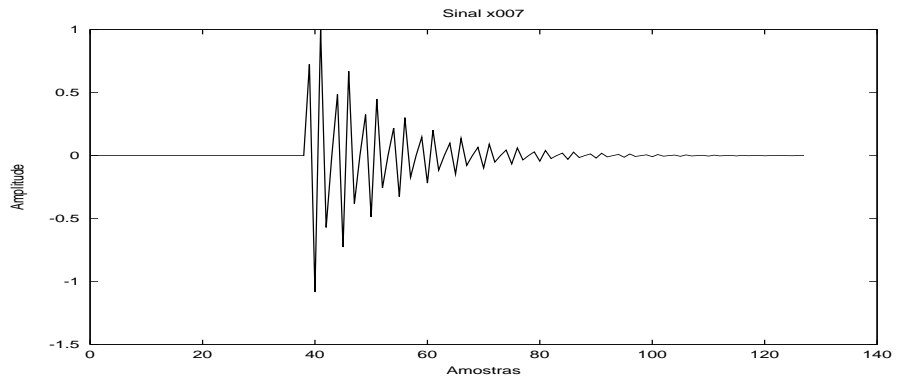


Figura A.7: Sinal x007

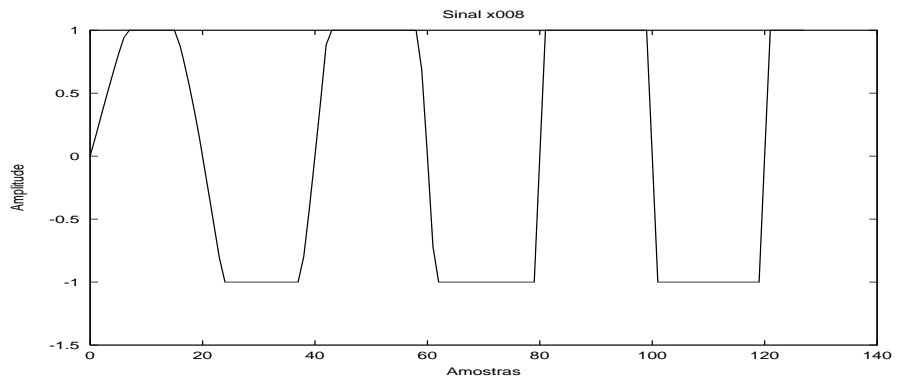


Figura A.8: Sinal x008

A.2 Sinais Reais

Além dos sinais sintéticos, apresentados na seção anterior, devemos utilizar sinais reais para avaliar o desempenho do algoritmo proposto na representação de sinais elétricos. Neste contexto utilizou-se os sinais apresentados nesta seção. Estes sinais podem ser divididos em dois grupos:

- Colhidos no campo;
- Simulados no ATP, um simulador de linhas de transmissão que leva em consideração as equações diferenciais das linhas.

Além disso devemos observar as peculiaridades de cada um dos sinais, não só em relação aos transientes presentes em cada, mas também em relação à sua taxa de amostragem e bits utilizados na conversão A/D.

Os sinais utilizados são:

- **R001** - Oscilograma simulado no ATP que representa a corrente somada a uma exponencial real, armazenado com 16 bits, e uma relação entre frequência de amostragem e a frequência da fundamental de aproximadamente 416. Este sinal pode ser visto na figura A.9.
- **R002** - Oscilograma colhido no campo com 16 bits, com desligamento e religamento de linha. Este sinal é apresentado na figura A.10.
- **R003** - Curva real de sobrecorrente com desligamento amostrada com 16 bits. Este sinal é apresentado na figura A.11.
- **R004** - Oscilograma simulado no ATP armazenado com 16 bits. Este sinal pode ser visto na figura A.12.

A.3 Ruídos Gaussianos

Para realizar o estudo de coerência no *Matching Pursuit*, seção 3.3, com o dicionário de *Gabor*, capítulo 5, foi necessário a geração de sinais de ruído branco Gaussiano. Isto foi realizado neste trabalho gerando ruídos de média zero e variância



Figura A.9: Sinal R001.

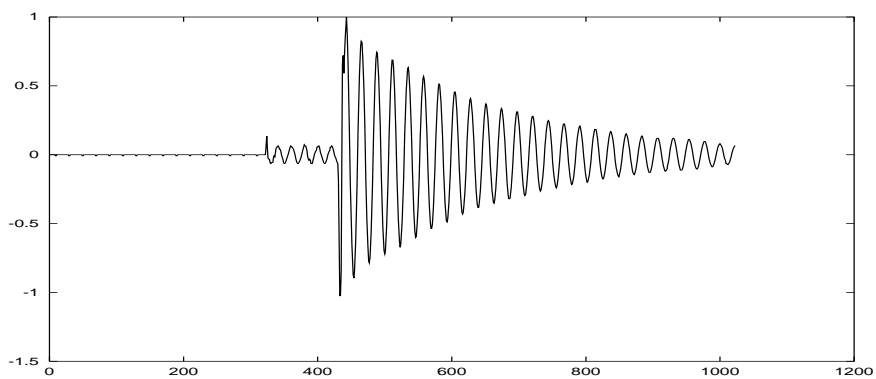


Figura A.10: Sinal R002.

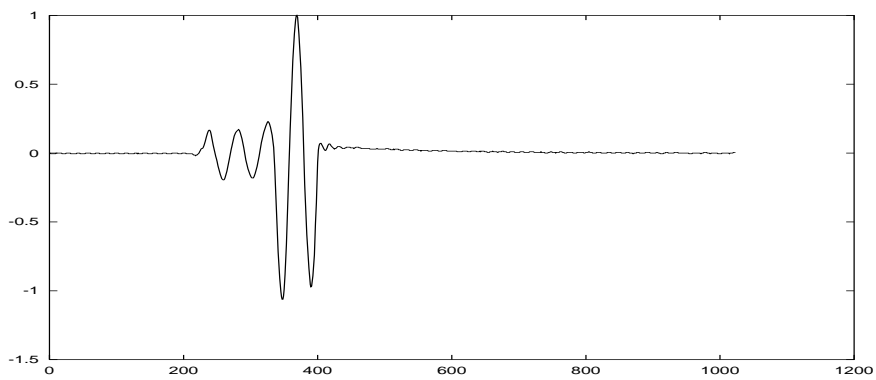


Figura A.11: Sinal R003.

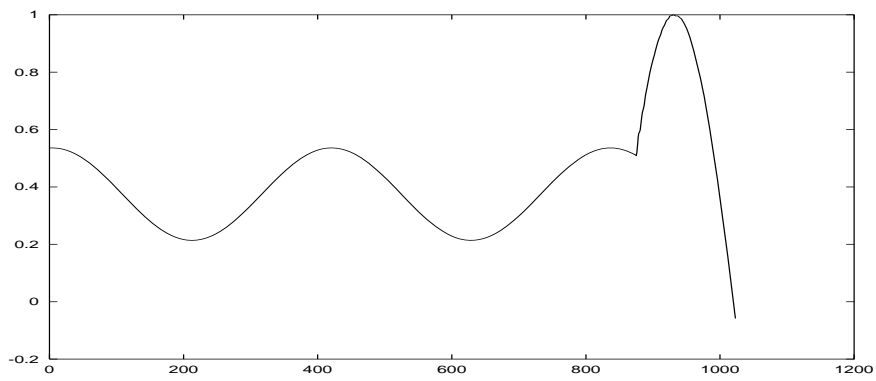


Figura A.12: Sinal R004.

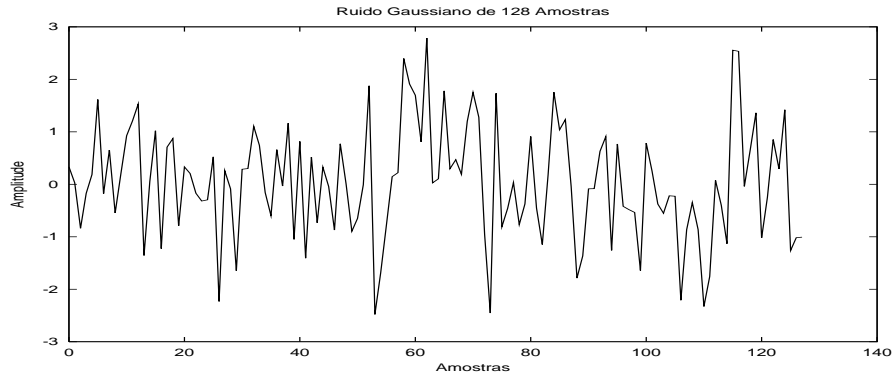


Figura A.13: Ruído Gaussiano de 128 Amostras

unitária. Geramos sinais de ruídos Gaussianos com 128, 256, 512 e 1024 amostras para poder realizar este estudo. Estes sinais podem ser vistos nesta seção.

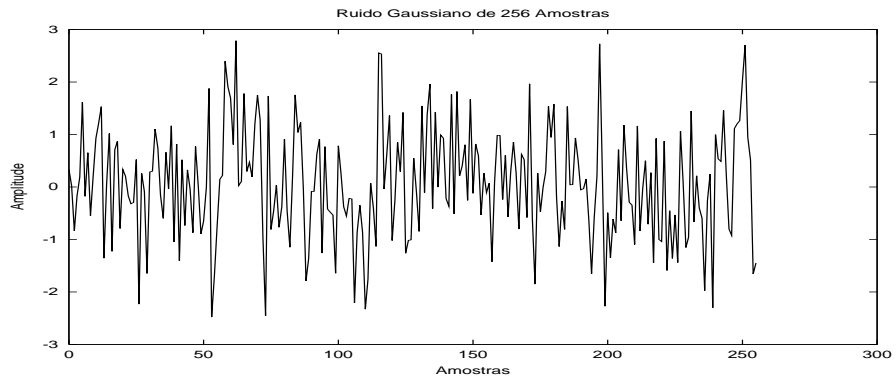


Figura A.14: Ruído Gaussiano de 256 Amostrs

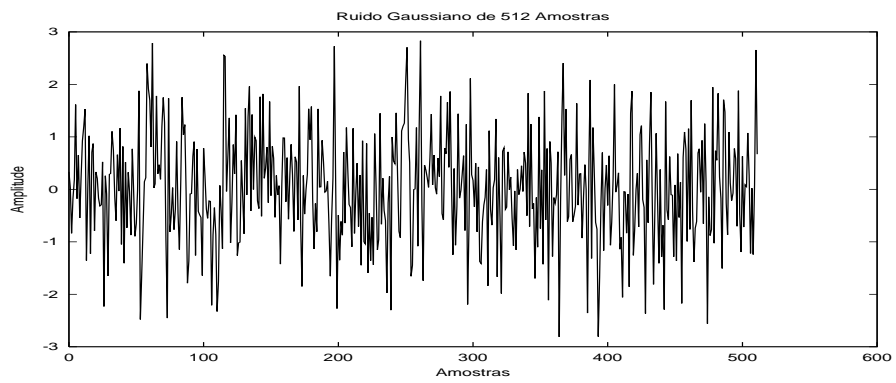


Figura A.15: Ruído Gaussiano de 512 Amostrs

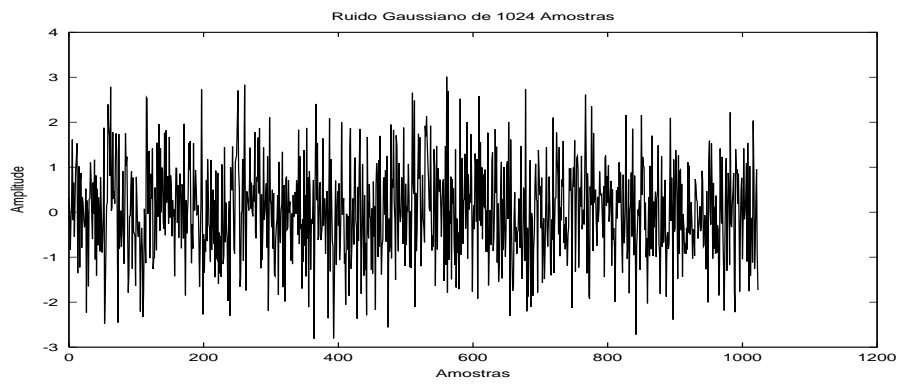


Figura A.16: Ruído Gaussiano de 1024 Amostrs

Referências Bibliográficas

- [1] MALLAT, S., ZANG, Z., “Matching Pursuits With Time-Frequency Dictionaries”, *IEEE Transactions on Signal Processing*, v. 41, n. 12, pp. 3397–3415, Dezembro 1993.
- [2] MALLAT, S., *A Wavelet Tour of Signal Processing*. Academic Press, 1998.
- [3] SANTOSO, S., POWERS, E. J., GRADY, W. M., “Power Quality Disturbance Data Compression using Wavelet Transform Methods”. *IEEE/PES Summer Meeting*, IEEE, Denver, Colorado, July/August 1996.
- [4] RUDIN, A., *Real and Complex Analysis*. McGraw-Hill, 1973.
- [5] SPIEGEL, M. R., *Cálculo Avançado*. McGraw-Hill, 1968.
- [6] OPPENHEIM, A. V., SCHAFER, R. W., *Digital Signal Processing*. Prentice-Hall, 1975.
- [7] GOODWIN, M. E., VETTERLI, M., “Matching Pursuits and Atomic Signal Models Based on Recursive Filters Banks”, *IEEE Transactions on Signal Processing*, v. 47, n. 7, pp. 1890–1902, Julho 1999.
- [8] DAUBECHIES, I., *Ten Lectures on Wavelets*. Philadelphia, Pensilvania, USA, Society for Industrial and Applied Mathematics, 1991.
- [9] GOYAL, V. K., VETTERLI, M., THAO, N. T., “Quantized Overcomplete Expansions in \mathbb{R}^N : Analysis, Synthesis, and Algorithms”, *IEEE Transactions on Information Theory*, v. 44, n. 1, pp. 16–31, January 1998.
- [10] GOYAL, V. K., *Quantized Overcomplete Expansions in \mathbb{R}^N : Analysis, Synthesis, and Algorithms*. In: Memorandum UCB/ERL M95/97, Electronics Re-

search Laboratory, College of Engineering, University of California, Berkeley, Berkley, CA 94720, July 1995.

- [11] BURRUS, C. S., GOPINATH, R. A., GUO, H., *Introduction To Wavelets and Wavelets Transforms A Primer*. 1 ed. Upper Saddle River, New Jersey 07458, USA, Prentice Hall, 1998.
- [12] COIFMAN, R. R., WICKERHAUSER, M. V., “Entropy-based algorithms for best-basis selection”, *IEEE Transactions on Information Theory*, v. 38, pp. 713–718, 1992.
- [13] CHEN, S., DONOHO, D., *Basis Pursuit*. In: Report, Department of Statistics, Stanford University, <http://www-stat.stanford.edu/~donoho/Reports/1994/asilomar.ps.Z>, 1994.
- [14] DAVIS, G., MALLAT, S., ZHANG, Z., *Adaptive Time-Frequency Approximations with Matching Pursuits*. In: Relatório de Pesquisa, Courant Institute of Mathematical Sciences, Computer Science Department, New York University, 251 Mercer Street, New York, NY 10012, USA, 1999.
- [15] CONWAY, J., SLOANE, N., *Sphere Packings, Lattices and Groups*, A Series of Comprehensive Studies in Mathematics. 2 ed. New York, New York 10010, USA, Springer Verlag, 1993.
- [16] GAREY, M. R., JOHNSON, D. S., *Computers and Intractability: A guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [17] CHEN, S., DONOHO, D., SAUNDERS, M., *Basis Pursuit*. In: Report, Department of Statistics, Stanford University, <http://www-stat.stanford.edu/~donoho/Reports/1995/30401.ps.Z>, 1995.
- [18] MALLAT, S., ZANG, Z., *Matching Pursuits With Time-Frequency Dictionaries*. In: Relatório de Pesquisa, Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, USA.
- [19] RODRIGUES, M. A. M., *Decomposição Eficientes para Codificação de Sinais*. Tese de doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Março 1999. Nota: tese de doutorado.

- [20] DAVIS, G., *Adaptive Nonlinear Approximations*. Tese de doutorado, Courant Institute of Mathematical Sciences, Computer Science Department, New York University, 251 Mercer Street, New York, NY 10012, USA, Setembro 1994.
- [21] COHEN, L., “Time-Frequency Distributions - A Review”, *Proceeding of the IEEE*, v. 77, n. 7, pp. 941–981, July 1989.
- [22] LOVISOLO, L., da Silva, E. A. B., “Uniform Distributions of Points on an Hyper-Sphere with Applications to Vector Bit-Plane Encoding”, *IEE Proceeding on Vision, Image and Signal Processing*, , 2001. A ser publicado.
- [23] HLAWATSCH, F., BOUDREAUX-BARTELS, G. F., “Linear and Quadratic Time-Frequency Signal Representations”, *IEEE Signal Processing Magazine*, , Abril 1992.
- [24] BOSE, N. K., “Multidimensional Digital Signal Processing: Problems, Progress, and Future Scopes”, *Proceedings of the IEEE*, v. 4, n. 78, pp. 590–597, April 1990.
- [25] VETTERLI, M., HERLEY, C., “Wavelets and Filter Banks: Theory and Design”, *IEEE Transaction on Signal Processing*, v. 40, pp. 2207–2232, September 1992.
- [26] GALLI, A. W., HEYDT, G. T., RIBEIRO, P. F., “Exploring the Power of Wavelet Analysis”, *IEEE Computer Applications in Power*, pp. 37–41, October 1996.
- [27] FERRANDO, S. E., KOLASA, L. A., KOVACEVIC, N., *A Flexible Implementation of Matching Pursuit for Gabor Dictionaries*. In: Report, Department of Mathematics, Physics and Computer Science, Ryerson Polytechnic University, <http://www.scs.ryerson.ca/ferrando/Research/mpGaborInterval.ps>.
- [28] DONOHO, D., JOHNSTONE, I. M., *Ideal Denoising in an orthonormal basis chosen from a library of bases*. In: Report, Department of Statistics, Stanford University, <http://www-stat.stanford.edu/~donoho/Reports/1994/idealbasis.ps.Z>, 1994.

- [29] GERSHO, A., GRAY, R. M., *Vector Quantization And Signal Compression*, Engineering and Computer Science. 1 ed. Boston/Dordrecht/London, Kluwer Academic Publishers, 1992.
- [30] LIM, J. S., *Two Dimensional Signal and Image Processing*, Prentice Hall Signal Processing Series. 1 ed. Englewood Cliffs, New Jersey 07632, Prentice Hall PTR, 1990.
- [31] DINIZ, P. S. R., da Silva, E. A. B., NETTO, S. L., *Digital Signal Processing: System Analysis and Design*. Cambridge Press, 2001. Nota: A ser publicado.
- [32] DONOHO, D., HUO, X., *Uncertainty Principles and Ideal Atomic Decomposition*. In: Report, Department of Statistics, Stanford University, <http://www-stat.stanford.edu/~donoho/Reports/1999/IADUP.ps.Z>, 1999.
- [33] ENGAN, K., AASE, S. O., HUSOY, J. H., “Multi-Frame Compression: Theory and Design”, *Elsevier Signal Processing*, v. 80, pp. 2121–2140, 2000.
- [34] ENGAN, K., AASE, S. O., HUSOY, J. H., “Designing frames for matching pursuits algorithms”. *ICASSP'98*, IEEE, Seattle, USA, May 1998.