

Laércio Alves Nogueira

# Aplicação de *Wavelets* em Sistemas de Espalhamento Espectral

Tese apresentada à Escola Politécnica  
da Universidade de São Paulo para  
obtenção do Título de Doutor em Enge-  
nharia.

Laércio Alves Nogueira

# Aplicação de *Wavelets* em Sistemas de Espalhamento Espectral

Tese apresentada à Escola Politécnica  
da Universidade de São Paulo para  
obtenção do Título de Doutor em Enge-  
nharia.

Área de concentração:  
Sistemas Eletrônicos

Orientador:  
Prof. Dr. Silvio Ernesto Barbin

# Agradecimentos

A Deus, por tudo que tem feito em minha vida. (*“Thus far has the Lord helped us.”* I Sm 7.12)

Ao meu prezado orientador Prof. Dr. Silvio Ernesto Barbin pelo conhecimento transmitido, apoio e dedicação e pela conduta decisiva para a conclusão deste trabalho.

Ao Prof. Dr. Lajos Hanzo por ter me recebido em 2008 na Universidade de Southampton, na Inglaterra, viabilizando pesquisas iniciais deste trabalho.

À FDTE-Qualcomm pela bolsa recebida em um período do Doutorado.

À Tânia e ao meu filho Samuel pelo apoio, amizade e paciência.

A todos os amigos que me ajudaram em diversas fases da realização deste trabalho; aos colegas, pesquisadores e professores do LCS, em especial ao Prof. Dr. Amazonas, por suas contribuições.

Ao Prof. Marcos Stefanelli pela amizade, pela grande ajuda e pelo exemplo.

A minha saudosa irmã e amiga Nina, pelo incentivo e por acreditar em mim.

Às amigas Flávia Gerjoi e Maria Calzada pelo incentivo constante.

A todos os meus amigos da Faculdade de Engenharia do Centro Universitário Fundação Santo André, que me apoiaram desde o início do meu programa de doutoramento.

# Resumo

Este trabalho apresenta uma nova aplicação de *wavelets* em sistemas de espalhamento espectral. A aplicação apresentada foi realizada no contexto de detecção multiusuário e pode ser empregada em sistemas de acesso múltiplo por divisão de código. A detecção e a estimação são realizadas através de um algoritmo de inferência bayesiana, empregando o Amostrador de Gibbs e utilizando as *wavelets* para melhorar a relação sinal ruído na recepção. Na transmissão, um código específico, baseado na *wavelet* de Haar, modula o sinal contendo a informação preparando-o contra alterações que possam ocorrer devido a características intrínsecas do canal de ruído branco, gaussiano e aditivo. Esta técnica foi aqui denominada de modulação *wavelet*. Na recepção, uma transformada *wavelet* de Haar é aplicada ao sinal, gerando coeficientes *wavelets*. A ortogonalidade da *wavelet* de Haar torna possível executar um procedimento de cancelamento de ruído, em inglês *denoising*, que melhora a qualidade da recepção. Para validar a abordagem apresentada neste trabalho, foi executada uma série de simulações. Os resultados mostraram um melhor desempenho sistêmico com o emprego de *wavelets*. Nas simulações foram observados vários parâmetros de sistema, dentre eles a amplitude do sinal, a taxa de erro de bits (BER, *bit error rate*) e a variância do sinal. Os resultados foram bastante satisfatórios, indicando uma redução do ruído da ordem de 3dB em relação ao valor sem cancelamento de ruído. Este trabalho abre novas oportunidades de pesquisa para o uso de *wavelets* em sistemas de espalhamento espectral e telecomunicações em geral.



# Abstract

This work presents a new application of wavelets to spread spectrum systems. The presented application was made in the context of multiuser detection and can be used in code-division multiple access systems. Detection and estimation are performed through a bayesian inference algorithm and, employing the Gibbs sampler and utilizing wavelets to improve the signal-to-noise ratio at the reception. At the transmission, a specific code based on the Haar wavelet modulates the signal containing the information, preparing it to alterations that may occur due to the intrinsic characteristics of the additive Gaussian noise channel. This technique was here denominated wavelet modulation. At the reception, a Haar wavelet transform is applied to the signal, generating wavelet coefficients. The orthogonality of the Haar wavelet makes it possible to perform a noise cancelling process, called denoising, which enhances the receiving quality. To validate the approach presented in this work, a number of simulations were performed. The results showed a better system performance with the use of wavelets. Several parameters were monitored in the simulations, among them the signal amplitude, the BER (bit error rate), and the signal variance. The results were very satisfactory, indicating a noise reduction of approximately 3dB referred to the value without noise cancelling. This work opens new research opportunities for the use of wavelets in spread spectrum systems and Telecommunications in general.

# Sumário

## Lista de Figuras

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	2
1.2	Motivação . . . . .	2
1.3	Principais Contribuições do Trabalho . . . . .	4
1.4	Estrutura do Trabalho . . . . .	5
<b>2</b>	<b>Detecção Multiusuário e Inferência Bayesiana</b>	<b>6</b>
2.1	Introdução . . . . .	7
2.2	Comunicação Multiusuário . . . . .	7
2.2.1	Introdução às técnicas de acesso múltiplo . . . . .	7
2.2.2	Capacidade dos métodos de acesso múltiplo [10, 11, 75] . .	9
2.2.3	Acesso múltiplo por divisão de código (CDMA) . . . . .	13
2.3	Inferência Bayesiana . . . . .	27
2.3.1	Procedimentos bayesianos . . . . .	28
2.3.2	Distribuições <i>a priori</i> . . . . .	29
<b>3</b>	<b><i>Wavelets</i></b>	<b>31</b>
3.1	Introdução . . . . .	32
3.2	Transformada janelada de Fourier . . . . .	33
3.2.1	Definição da WFT e Definição . . . . .	33
3.2.2	Localização no tempo e na frequência . . . . .	39
3.2.3	A Fórmula de Reconstrução . . . . .	41
3.2.4	Processamento de sinais no domínio tempo-frequência . . .	42

3.3	Transformadas <i>wavelets</i> contínuas . . . . .	42
3.3.1	Motivação e definição da transformada <i>wavelet</i> . . . . .	42
3.3.2	Expressão de Reconstrução . . . . .	45
3.3.3	Localização em frequência e sinais analíticos . . . . .	50
3.3.4	Análise de multirresolução e transformada <i>wavelet</i> discreta . . . . .	52
<b>4</b>	<b>Detecção Multiusuário com aplicação de <i>wavelets</i></b>	<b>63</b>
4.1	Detecção multiusuário adaptativa bayesiana para sistemas de espalhamento espectral . . . . .	64
4.2	Detector multiusuário bayesiano adaptativo . . . . .	66
4.2.1	Determinação das probabilidades do modelo . . . . .	69
4.2.2	Detector Multiusuário Iterativo . . . . .	70
4.3	Processamento digital com modulação por <i>wavelets</i> . . . . .	72
4.3.1	Análise de <i>wavelets: denoising</i> . . . . .	75
4.3.2	Demonstração do algoritmo da transformada <i>wavelet</i> utilizado no detector multiusuário bayesiano . . . . .	76
<b>5</b>	<b>Comparação entre os algoritmos Amostrador de Gibbs e Metropolis-Hastings</b>	<b>78</b>
5.1	Introdução . . . . .	79
5.2	Integração Monte Carlo clássica . . . . .	79
5.3	Métodos de Monte Carlo via cadeias de Markov . . . . .	80
5.3.1	Processos de Markov [56] . . . . .	81
5.4	Algoritmos Amostrador de Gibbs e Metropolis Hastings . . . . .	87
5.4.1	Os algoritmos Metropolis . . . . .	87
5.4.2	O algoritmo Amostrador de Gibbs em inferência bayesiana . . . . .	91
5.4.3	Análise de Convergência . . . . .	93
5.4.4	Motivos da escolha do Amostrador de Gibbs . . . . .	94
5.4.5	Análise de Complexidade ( <i>time bounds</i> ) . . . . .	95
5.5	Testes dos algoritmos e resultados . . . . .	96

5.5.1	Densidades de Rayleigh e de Pareto . . . . .	102
5.6	Covariância nas cadeias de Markov . . . . .	104
5.7	Análise comparativa das amostras geradas pelos dois algoritmos .	105
5.8	Métricas para Análise de Desempenho dos algoritmos do sistema .	109
5.8.1	Amostras, Estatísticas e Estimadores . . . . .	109
5.9	Considerações finais . . . . .	116
<b>6</b>	<b>Detecção multiusuário bayesiana com aplicação de <i>wavelets</i></b>	<b>117</b>
6.1	Detecção multiusuário bayesiana com aplicações de <i>wavelets</i> . . .	118
6.2	Sistema utilizado - Caso síncrono . . . . .	120
6.2.1	Resultados das estimações das amplitudes e variância do ruído e da detecção multiusuário . . . . .	121
6.2.2	Robustez ao efeito <i>near-far</i> . . . . .	133
6.2.3	Caso assíncrono . . . . .	135
6.3	Considerações finais . . . . .	141
<b>7</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>142</b>
	<b>Referências</b>	<b>144</b>
	<b>ANEXO 1</b>	<b>149</b>
<b>8</b>	<b>Apêndices</b>	<b>163</b>
8.1	Simulação de Distribuições Específicas . . . . .	164
8.1.1	A Transformação Integral de Probabilidade . . . . .	164
8.1.2	Simulação . . . . .	165
8.1.3	Algoritmos com Propósitos Especiais . . . . .	167

# Lista de Figuras

2.1	Capacidade total por Hertz como função de $E_b/N_0$ para o FDMA. (Figura adaptada de [11].) . . . . .	10
2.2	Capacidade normalizada como função de $E_b/N_0$ para o FDMA. (Figura modificada de [11].) . . . . .	12
2.3	Definição das correlações cruzadas ( $i < j$ ). . . . .	14
2.4	Receptor ótimo multiusuário de transmissão síncrona. . . . .	17
2.5	Receptor ótimo síncrono com filtros casados. . . . .	18
2.6	Estrutura do receptor do descorrelacionador. . . . .	22
3.1	Parte superior: o sinal ‘chirp’ $f(u) = \sin(\pi u^2)$ ; parte inferior: a densidade espectral de energia $ \hat{f}(\omega) ^2$ de $f$ . . . . .	36
3.2	A função janela $g(u)$ . . . . .	37
3.3	A função janela $g(u)$ com intervalo reproduzido ao longo do eixo do tempo. . . . .	37
3.4	A função janela $g(u)$ . . . . .	38
3.5	Gráfico superior: a versão localizada $f_7(u)$ do sinal <i>chirp</i> usando a janela $g(u)$ . Gráfico inferior: a densidade de energia de $f_7$ mos- trando boa localização em volta da frequência instantânea $\omega_{inst}(7) =$ 7. . . . .	39
3.6	Curva superior: gráfico de $h = Re[g_{2,4} + g_{4,6}]$ para a janela em $g(u)$ . Curva inferior: Densidade espectral de energia de $h$ mostrando problema de resolução de frequência para $\Delta\omega = 2$ . . . . .	40
3.7	Curva superior: gráfico de $h = Re[g_{2,4} + g_{3,6}]$ para a janela em $g(u)$ . Curva inferior: Densidade espectral de energia de $h$ mostrando problema de resolução de frequência para $\Delta\omega = 1$ . . . . .	40
3.8	Gráficos de $\psi(u) = ue^{-u^2}$ (centro) e suas versões transladadas e dilatadas $\psi_{3,15}(u)$ (direita) e $\psi_{-\frac{1}{2},-10}(u)$ (esquerda). . . . .	44

3.9	Uma ilustração da DWT em 3 níveis (figura obtida com o software R). . . . .	56
3.10	Densidade espectral de potência do sinal $x(k)$ do exemplo. . . . .	56
3.11	Síntese do sinal $x(k)$ (figura adaptada de [45]). . . . .	57
3.12	Resposta em frequência de filtros QMF (gráfico superior); resposta em frequência de filtros do tipo <i>brickwall</i> (gráfico inferior). . . . .	60
3.13	(a) banco de filtros de análise $H^*(f)$ (passa-baixas) e $G^*(f)$ (passa-altas) com decimação ( <i>downsampling</i> ); (b) banco de filtros de reconstrução com interpolação ( <i>upsampling</i> ). São usados os filtros duais $G(f)$ e $F(f)$ . . . . .	62
4.1	Sistema de espalhamento espectral de acesso múltiplo. . . . .	64
4.2	Processamento iterativo do receptor de detecção multiusuário. . .	71
4.3	Transmissor de sistema de espalhamento espectral de acesso múltiplo.	73
4.4	Transmissor modificado: apresenta o código de Haar, que substitui o codificador de canal. . . . .	73
4.5	Receptor modificado, que apresenta o processamento por <i>wavelets</i> .	74
4.6	Receptor com processamento iterativo. . . . .	74
5.1	Resultados das simulações do Metropolis-Hastings por passeio aleatório com diferentes tamanhos de passo ( <i>step sizes</i> ), (a) $\gamma = 0,01$ , (b) $\gamma = 0,05$ , (c) $\gamma = 0,10$ e (d) $\gamma = 1,00$ . O tamanho da amostra em cada figura é 5000. . . . .	90
5.2	Diagnósticos de convergência para a componente $x$ com os valores de $\gamma$ como na Figura 5.1. . . . .	91
5.3	Gráfico da densidade dada pela Eq. 5.46, amostrada pelo Metropolis-Hastings. . . . .	96
5.4	Gráfico da densidade dada pela Eq. 5.46, amostrada pelo Metropolis-Hastings, com detalhes da visualização. . . . .	97
5.5	Gráfico de densidades obtidas da Eq. 5.46; a distribuição empírica foi amostrada com o Amostrador de Gibbs. . . . .	98
5.6	Gráfico da densidade dada pela Eq. 5.47 amostrada pelo Metropolis-Hastings. . . . .	99

5.7	Gráfico da densidade dada pela Eq. 5.48 amostrada pelo Amostrador de Gibbs. . . . .	100
5.8	Gráfico da densidade dada pela Equação 5.49 amostrada pelo Metropolis-Hastings. . . . .	101
5.9	Gráfico da função ACF sobre a variável $x$ da densidade da Eq. 5.49.102	
5.10	Gráfico de uma densidade Rayleigh amostrada com o Amostrador de Gibbs. . . . .	103
5.11	Gráfico da distribuição acumulada de Rayleigh. . . . .	103
5.12	Gráfico das amostras de uma distribuição de Pareto amostrada com o Amostrador de Gibbs. . . . .	104
5.13	Gráfico das v.a. $x$ e $y$ e da função de autocorrelação das amostras obtidas com o Amostrador de Gibbs . . . . .	106
5.14	Gráfico das v.a. $x$ e $y$ e da função de autocorrelação das amostras obtidas com o Metropolis-Hastings, com o parâmetro $\gamma$ igual a 1,0. 106	
5.15	Gráfico das v.a. $x$ e $y$ e da função de autocorrelação das amostras obtidas com o Metropolis-Hastings, com o parâmetro $\gamma$ igual a 0,5. 107	
5.16	Gráfico das v.a. $x$ e $y$ e da função de autocorrelação das amostras obtidas com o Metropolis-Hastings, com o parâmetro $\gamma$ igual a 0,1. 107	
5.17	Gráfico das v.a. $x$ e $y$ e da função de autocorrelação das amostras obtidas com o Metropolis-Hastings, com o parâmetro $\gamma$ igual a 0,05.108	
5.18	Gráfico das v.a. $x$ e $y$ e da função de autocorrelação das amostras obtidas com o Metropolis-Hastings, com o parâmetro $\gamma$ igual a 0,01.108	
5.19	Gráfico das amplitudes de dois usuários e da variância do ruído do canal, obtidas por simulação. . . . .	112
5.20	Gráfico das amplitudes de sinais do usuário 1, que mostra a rápida convergência. . . . .	114
5.21	Gráfico das Amplitudes e variância dos sinais; a convergência é rápida para o algoritmo adotado. . . . .	115
6.1	Sistema de comunicação de acesso múltiplo. . . . .	120
6.2	Diagrama em blocos do receptor Bayes- <i>only</i> . . . . .	121
6.3	Símbolos detectados de um usuário; estimação das amplitudes de dois usuários e variância do ruído. . . . .	123

6.4	Erro médio quadrático <i>versus</i> $M$ (comprimento do quadro). . . .	124
6.5	Valores estimados das amplitudes de cinco usuários e variância do ruído. . . . .	126
6.6	Gráfico BER x SNR. . . . .	127
6.7	Transmissor modificado com a inclusão do código de Haar no lugar do bloco codificador de canal. . . . .	128
6.8	Receptor modificado apresentando o processamento por <i>wavelets</i> . . . . .	128
6.9	Gráfico das amplitudes de dois usuários e da variância do ruído, sem a aplicação do <i>denoising</i> . . . . .	129
6.10	Resultado da ação do <i>denoising</i> sobre um sinal composto recebido do canal sem fio, a energia do ruído caiu pela metade. . . . .	130
6.11	Coeficientes da transformada <i>wavelet</i> aplicada a um sinal de espalhamento espectral. . . . .	131
6.12	Coeficientes da transformada <i>wavelet</i> aplicada a um sinal de espalhamento espectral com código de Haar; os coeficientes das escalas 2, 3 e 4 têm valor zero. . . . .	132
6.13	Coeficientes da transformada <i>wavelet</i> aplicada a um sinal SS com código de Haar e corrompido por ruído. . . . .	132
6.14	BER $\times$ razão <i>near-far</i> $\frac{A_2}{A_1}$ , variância = 0, 1. . . . .	133
6.15	BER $\times$ razão <i>near-far</i> $\frac{A_2}{A_1}$ , variância = 0, 05. . . . .	135
6.16	Modo assíncrono, estimação das amplitudes de cinco usuários e da variância do ruído. . . . .	136
6.17	Modo assíncrono, estimação das amplitudes de cinco usuários e da variância do ruído, com aplicação do <i>denoising</i> . . . . .	137
6.18	Modo assíncrono, estimação das amplitudes de dois usuários e da variância do ruído, comparação com o filtro casado. . . . .	138
6.19	Modo assíncrono, estimação das amplitudes de cinco usuários e da variância do ruído, comparação com o detector convencional. . . .	139
6.20	Resultado do teste com o Gold-512, curva superior do gráfico. . .	140



## LISTA DE ABREVIATURAS

**AWGN** *additive white Gaussian Noise*

**BER** *bit error rate*

**bps** bits por segundo

**BPSK** *Binary Phase Shift Keying*

**CDF** *Cumulative Distribution Function*

**CDMA** *Code-Division Multiple Access*

**CWT** *Continuous Wavelet Transform*

**DEP** *Densidade Espectral de Potência*

**DS-CDMA** *Direct Sequence Code-Division Multiple Access*

**DS-SS** *Direct Sequence Spread Spectrum*

**DWT** *Discrete Wavelet Transform*

**DWPT** *Discrete Wavelet Packet Transform*

**EM** *Expectation-Maximization*

**EMQ** Estimador de Mínimos Quadrados

**EMV** Estimador de Máxima Verossimilhança

**EQM** Erro Quadrático Médio

**FDMA** *Frequency Division Multiple Access*

**FEC** *Forward Error Correction*

**f.d.p.** função densidade de probabilidade

**FFT** *Fast Fourier Transform*

**FIR** *Finite Impulse Response*

**f.p.** função de probabilidade

**ICWT** *Inverse Continuous Wavelet Transform*

**IDWT** *Inverse Discrete Wavelet Transform*

**i.i.d.** *independent, identically distributed*

**IIR** *Infinite Impulse Response*

**ISI** *Intersymbol Interference*

**MAI** *Multiple Access Interference*

**MAP** *Maximum a Posteriori*

**MCMC** *Markov Chain Monte Carlo*

**MEP** *Mean Error Percentual*

**MIMO** *Multiple Input - Multiple Output*

**ML** *Maximum Likelihood*

**MLSD** *Maximum Likelihood Sequence Detection*

**MLSE** *Maximum Likelihood Sequence Estimation*

**MMSE** *Minimum Mean Squared Error*

**MRA** *Multiresolution Analysis*

**MSE** *Mean Square Error*

**MuD** *Multiuser Detection*

**PIC** *Parallel Interference Canceller*

**PDF** *Probability Density Function*

**PDS** *Processamento Digital de Sinais*

**SE** *Square Error*

**SIC** *Successive Interference Cancellation*

**SMC** *Sequential Monte Carlo*

**SS** *Spread Spectrum*

**SNR** *Signal to Noise Ratio*

**TCB** *Time Complexity per Bit*

**TDF** Transformada Discreta de Fourier

**TDMA** *Time Division Multiple Access*

**TFTD** *Transformada de Fourier de Tempo Discreto*

**WFT** *Windowed Fourier Transform*

# 1 Introdução

## Objetivos e Motivação

Este capítulo apresenta os objetivos, a motivação e uma breve exposição das principais contribuições deste trabalho.

## 1.1 Objetivos

O objetivo geral deste trabalho é apresentar uma nova aplicação de *wavelets* que cancela o ruído adquirido em uma comunicação sem fio de espalhamento espectral. Para isso é necessário atingir os seguintes objetivos intermediários.

- Desenvolver um detector bayesiano com cancelamento de ruído; neste desenvolvimento foi adotada uma abordagem usando *wavelets*.
- Desenvolver um código de transmissão específico baseado numa *wavelet*.
- Validar a proposta através de simulações, avaliando o desempenho quanto ao nível de sinal, a variância do sinal e a BER.
- Investigar a robustez da solução quanto ao efeito *near-far*.
- Investigar o modo assíncrono e a influência de diferentes códigos de espalhamento.

## 1.2 Motivação

As comunicações sem fio representam uma área que se expandiu apreciavelmente desde a década de 1990, tanto em pesquisa e desenvolvimento como em crescimento da infraestrutura [1]. A evolução das técnicas de processamento digital de sinais (PDS) contribuiu significativamente para esses progressos. O surgimento de *smartphones* e a oferta de pequenos computadores de uso geral foram viabilizados pela convergência das Telecomunicações com a Internet. Atualmente, é possível processar funções complexas através da Internet com alta confiabilidade. Neste contexto, sistemas de espalhamento espectral representam uma excelente opção técnica para uma comunicação sem fio, pela sua robustez quanto à interferência, simplicidade e facilidade de implementação. A motivação deste trabalho está relacionada à necessidade de melhorar a qualidade de recepção nesse tipo de sistema de multiusuários. Essa motivação reveste-se ainda de maior interesse pela possibilidade inovadora de se aplicar técnicas modernas, como o emprego de *wavelets*, a um detector bayesiano para a detecção multiusuário. Neste contexto os objetivos estabelecidos foram plenamente atingidos, conforme detalhamento apresentado nos próximos capítulos e validação de resultados obtidos por simulação.

Com relação à proposta do detector multiusuário, os principais requisitos es-

tabelecidos são os seguintes:

- detecção multiusuário em sistemas DS-CDMA (acesso múltiplo por divisão de códigos e de sequência direta), que leva em conta a interferência MAI, separada do ruído do canal, na obtenção do sinal desejado [2];
- inferência bayesiana no processamento de sinais, utilizada na estimação e/ou detecção de sinais por meio do método sequencial de Monte Carlo (SMC, *Sequential Monte Carlo*) [3];
- utilização da transformada *wavelet* para cancelar o ruído adicionado pelo canal [4, 5].

Em um sistema mais completo para determinadas aplicações práticas, seriam ainda considerados requisitos adicionais tais como: – códigos corretores de erro (códigos Turbo) e sistemas MIMO (múltiplas entradas-múltiplas saídas), ambos utilizados em canais sem fio [6];

- antenas inteligentes, que melhoram o ganho do sistema adaptando-se no espaço e no tempo ao sinal recebido [2];
- processamento digital adaptativo, utilizado em equalização de sinais [7];
- estimação cega, utilizada em sistemas onde não se utiliza uma sequência de treinamento [8];

Com relação às transformadas *wavelets*, ressalta-se que são ferramentas muito importantes na área de processamento digital de sinais. Com o seu advento, tornou-se possível fazer análises de tempo-frequência de sinais reais, utilizados nos sistemas de comunicação. Neste trabalho elas são aplicadas como filtros, gerando coeficientes que são posteriormente explorados para cancelamento de ruído (em inglês *denoising*), conforme será visto mais adiante. O *denoising* em geral é aplicado sobre os coeficientes *wavelets*: determina-se um certo nível de sinal, que é fixado como “referência”; comparações são então realizadas entre estes coeficientes *wavelets* e a referência: os coeficientes cujos valores são menores ou iguais ao nível de referência pré-estabelecido são zerados. Tem-se dessa forma um procedimento padrão, muito utilizado, para cancelamento de ruído em imagens, por exemplo. Neste trabalho o procedimento realizado de *denoising* é mais elaborado que o procedimento padrão: na transmissão um código específico é aplicado ao sinal digital a ser transmitido, que após ser adicionado de ruído será detectado na recepção empregando-se transformada *wavelet*.

A detecção multiusuário é uma técnica que pode substituir com vantagens a detecção convencional, empregando o filtro casado. O desempenho do filtro

casado, utilizado ainda hoje nos sistemas de comunicação móvel sem fio, degrada sob o efeito “*near-far*”, apesar de ser ótimo em um sistema sem interferências, onde somente o ruído do canal afeta a qualidade da comunicação [9]. Ele não leva em conta na determinação da relação sinal-ruído a interferência de acesso múltiplo (MAI, *multiple access interference*). Sistemas que utilizam o filtro casado têm, entretanto, algumas vantagens: simplicidade de arquitetura; processamento menos complexo, pois despreza as informações de outros usuários; e possibilidade de explorar combinações de diversidade de multipercurso, utilizando o receptor RAKE. O desempenho do detector multiusuário é superior ao filtro casado.

Detecção multiusuário tornou-se tema de pesquisa de grande interesse ao redor do mundo após ser demonstrado que ele realmente tem desempenho superior ao do detector convencional. Porém devido à sua complexidade computacional, que cresce exponencialmente com o número de usuários, o detector multiusuário tem sido considerado inviável na prática. Tal fato fez com que muitos pesquisadores abandonassem a linha do detector ótimo e migrassem para detectores subótimos. Em geral, detectores subótimos apresentam um desempenho não muito inferior ao do ótimo e são implementáveis na prática. Como exemplo de tais detectores podem ser citados o linear, também conhecido como detector de erro médio quadrático médio mínimo (MMSE); o detector descorrelacionador (do Inglês: *decorrelator*) e o detector bayesiano, somente para citar alguns. Neste trabalho, foi implementado o detector adaptativo bayesiano que alia baixa complexidade com bom desempenho. Adicionalmente, incluiu-se um bloco que realiza uma transformada *wavelet* para o cancelamento de ruído.

## 1.3 Principais Contribuições do Trabalho

Com relação aos resultados alcançados neste trabalho, destacam-se o código desenvolvido com um software “R” da Estatística, utilizado nas simulações do detector multiusuário bayesiano, e a modulação por *wavelets*, que cancela o ruído de um sinal recebido de um canal AWGN, melhorando, assim, o processo de detecção e estimação da informação no sistema. Segue-se uma lista com as contribuições originais desta tese:

- Desenvolvimento de um novo algoritmo de cancelamento de ruído.
- Proposta de um sistema que trabalha com inferência bayesiana, transformada *wavelet* e detecção multiusuário.

- Demonstração, por meio de simulações, de que o modelo sugerido pode ser uma opção de sistema comercial.

Acredita-se que texto elaborado possa servir de referência para futuros trabalhos bem como para pesquisadores que pretendem iniciar estudos na área.

## 1.4 Estrutura do Trabalho

A estrutura geral deste trabalho está dividida em capítulos, subdivididos em seções e subseções. O Capítulo 2 apresenta uma introdução à detecção multiusuário e à inferência bayesiana; no Capítulo 3 descrevem-se os fundamentos da teoria das *wavelets*; no Capítulo 4 apresenta-se o detector empregando transformadas *wavelets* e o cancelamento de ruído; nos Capítulos 5 e 6 são apresentados a teoria, testes e resultados da análise dos algoritmos que resolvem o problema multidimensional numérico das equações obtidas para o sistema, e os resultados específicos dos testes realizados com o sistema; no Capítulo 7 apresentam-se as Conclusões e sugestões para futuros trabalhos.

No texto, vetores e matrizes são representados em negrito; por exemplo, dado um vetor “a”, este é representado por **a**; dada uma matriz “B”, sua representação é **B**. A transposta de uma matriz é indicada pela letra *T*, por exemplo, **B<sup>T</sup>**. O acento circunflexo, “^”, sobre alguma variável representa um estimador.



## 2 Detecção Multiusuário e Inferência Bayesiana

### Fundamentos Teóricos

## 2.1 Introdução

Os sistemas de comunicação de acesso múltiplo são em essência de múltiplos usuários. Estes usuários não estão sincronizados, eles utilizam o sistema de forma aleatória, com demandas diferentes, com diversos e variados objetivos e necessidades próprias. Dentre os vários tipos de canais existentes nos sistemas de Telecomunicações, o canal sem fio tem recebido atenção especial por parte de engenheiros e cientistas, devido ao seu valor econômico e tecnológico, e também à comodidade e praticidade que o mesmo proporciona aos usuários. Uma informação gerada em uma localidade por um usuário é enviada através do canal de comunicação a outro usuário que se encontra a uma certa distância do primeiro, parado ou em movimento. O processo de recuperação da informação enviada pode ser resumido na obtenção do sinal no receptor, sendo sua detecção o aspecto mais fundamental. A detecção do sinal de um usuário para ser ótima deve levar em conta a existência dos outros sinais que interferem no sinal desejado, isto é, todos os sinais envolvidos no processo devem ser analisados conjuntamente. A ferramenta básica utilizada é estatística, sendo a inferência bayesiana uma das possíveis abordagens. Na análise estatística do sinal a ser detectado, o modelo matemático pode tornar-se complexo e de difícil solução numérica, de maneira convencional. A utilização de métodos de Monte Carlo é uma alternativa apropriada aos métodos numéricos de resolução de integrais, particularmente em cenários multidimensionais. Assim é natural que estes métodos desempenhem um papel importante na abordagem bayesiana, onde o cálculo por via analítica ou numérica de quantidades *a posteriori*, que serão definidas neste capítulo, muitas delas requerendo integração, é frequentemente complexo.

Este capítulo está dividido em seções e subseções, onde serão abordados os fundamentos da detecção multiusuário e da inferência bayesiana; os métodos de Monte Carlo serão tratados na Seção 5.2.

## 2.2 Comunicação Multiusuário

### 2.2.1 Introdução às técnicas de acesso múltiplo

Atualmente existem diversas formas de acesso múltiplo disponíveis a usuários de sistemas de comunicação. Nesses tipos de comunicação, um grande número de usuários compartilha um mesmo canal de comunicação para transmitir informações. Este canal comum pode ser um enlace de subida de um sistema de satélite, ou um cabo ao qual está conectado um conjunto de terminais que aces-

sam um computador central. Este canal pode ser ainda uma faixa de frequências no espectro de rádio que é compartilhada por múltiplos usuários para se comunicarem com um receptor *wireless*: em um sistema de comunicação móvel celular os usuários utilizam transmissores móveis em uma célula qualquer do sistema e o receptor encontra-se no equipamento de rádio da célula.

Um segundo tipo de sistema multiusuário é uma rede de difusão na qual um único transmissor envia informações a múltiplos receptores. Exemplos desse tipo de sistema incluem o sistema de rádio difusão comercial, o sistema de TV aberta e também enlaces de descida de um sistema por satélite. As redes de difusão são provavelmente os sistemas de comunicações multiusuários mais comuns.

Neste capítulo serão tratados os métodos de acesso múltiplo para comunicações multiusuários. Em geral, existem diversas formas pelas quais um grupo de usuários possa enviar informações através de um canal de comunicação a um receptor. Um método simples é dividir a faixa de frequências do canal disponível em um número  $K$  de subcanais independentes e designar apenas um subcanal a cada usuário sob demanda. Esse método é chamado de acesso múltiplo por divisão de frequência (FDMA, *Frequency Division Multiple Access*) e é normalmente usado em canais com fio (*wireline channels*) para acomodar múltiplos usuários, com o objetivo de se transmitir voz (sistema analógico).

Um outro método para se criar múltiplos subcanais para utilização em acesso múltiplo consiste em subdividir um intervalo de tempo de duração  $T_f$ , chamada de duração de quadro em, por exemplo,  $K$  subintervalos sem intersecções, cada um com duração  $T_f/K$ . Então a cada usuário que deseja transmitir informações é atribuída uma janela de tempo particular dentro de cada quadro. Este tipo de método é chamado de acesso múltiplo por divisão de tempo (TDMA, *Time Division Multiple Access*) e é utilizado em transmissão de voz digital e de dados.

Pode-se observar que tanto no FDMA quanto no TDMA o canal é basicamente dividido em subcanais independentes do tipo *single-user*. Com essa idéia, os métodos de projeto de sistema de comunicação que são descritos para comunicação *single-user* são diretamente aplicáveis e não há problemas novos encontrados em um ambiente de acesso múltiplo, a não ser o da tarefa de designar usuários para os canais disponíveis.

Problemas interessantes começam a surgir quando os dados dos usuários que acessam a rede são do tipo “rajada” (do Inglês *burst*). O que significa que as transmissões das informações de um único usuário são separadas por períodos sem transmissões, onde esses períodos de silêncio podem ser até maiores do que os períodos de transmissão verdadeira. Esse é o caso de usuários de terminais de uma rede de computadores. Sob certo ponto de vista, esse é também o caso

de sistemas de comunicação móvel celular onde trafega voz digitalizada, pois os sinais de voz, tipicamente, contêm longas pausas.

Em um ambiente onde a transmissão de vários usuários é em rajadas e com ciclo baixo de atividade, o FDMA e o TDMA tendem a ser ineficientes porque uma porcentagem das janelas disponíveis de frequência ou de tempo atribuídas aos usuários não carregam informação. Por fim, um sistema de acesso múltiplo planejado de forma ineficiente limita o número de usuários que podem utilizar simultaneamente o canal.

Como uma alternativa ao FDMA e ao TDMA, pode-se permitir que mais que um usuário compartilhe o canal ou o subcanal pela utilização de sinais de espalhamento espectral de sequência direta (*direct-sequence spread spectrum*). Neste método, atribui-se a todo usuário uma sequência-código única (também chamada de *signature sequence*), que possibilita ao usuário espalhar o sinal de informação ao longo da banda de frequências designadas. Portanto, sinais de vários usuários são separados no receptor pela correlação cruzada do sinal recebido com cada uma das possíveis sequências de assinaturas de usuário.

### 2.2.2 Capacidade dos métodos de acesso múltiplo [10, 11, 75]

Torna-se interessante comparar o FDMA, o TDMA e o CDMA (*Code-Division Multiple Access*) em termos de taxa de informação que cada método de acesso múltiplo alcança em um canal ideal AWGN (*additive white Gaussian noise*) com largura de banda  $W$ . Faz-se aqui a comparação da capacidade de  $K$  usuários, onde cada usuário transmite com uma potência média  $P_i = P$ , para todo  $i$  tal que  $1 \leq i \leq K$ . Deve-se lembrar que em um canal AWGN ideal, limitado em banda, de largura de banda  $W$ , sua capacidade para um único usuário é dada por

$$C = W \log_2 \left( 1 + \frac{P}{WN_0} \right) \quad (2.1)$$

onde  $\frac{1}{2}N_0$  é a densidade espectral de potência bilateral do ruído aditivo do canal [10].

No FDMA, uma banda  $\frac{W}{K}$  é atribuída a todo usuário. Assim, a capacidade de cada usuário é

$$C_k = \frac{W}{K} \log_2 \left( 1 + \frac{P}{\frac{W}{K}N_0} \right) \quad (2.2)$$

e a capacidade total para  $K$  usuários é

$$KC_K = W \log_2 \left( 1 + \frac{KP}{WN_0} \right) \quad (2.3)$$

Portanto, a capacidade total é equivalente à aquela de um *single-user* (usuário único, o desejado) com potência média  $P_{av} = KP$ .

É interessante notar que para uma banda  $W$  fixa, a capacidade total vai para o infinito conforme o número de usuários aumenta linearmente com  $K$ . Por outro lado, conforme  $K$  aumenta, a cada usuário é alocada uma largura de banda menor  $(\frac{W}{K})$  e, conseqüentemente, a capacidade por usuário diminui. Essa expressão é dada por

$$\frac{C_K}{W} = \frac{1}{K} \log_2 \left[ 1 + K \frac{C_K}{W} \left( \frac{E_b}{N_0} \right) \right] \quad (2.4)$$

onde  $E_b$  é a energia por bit.

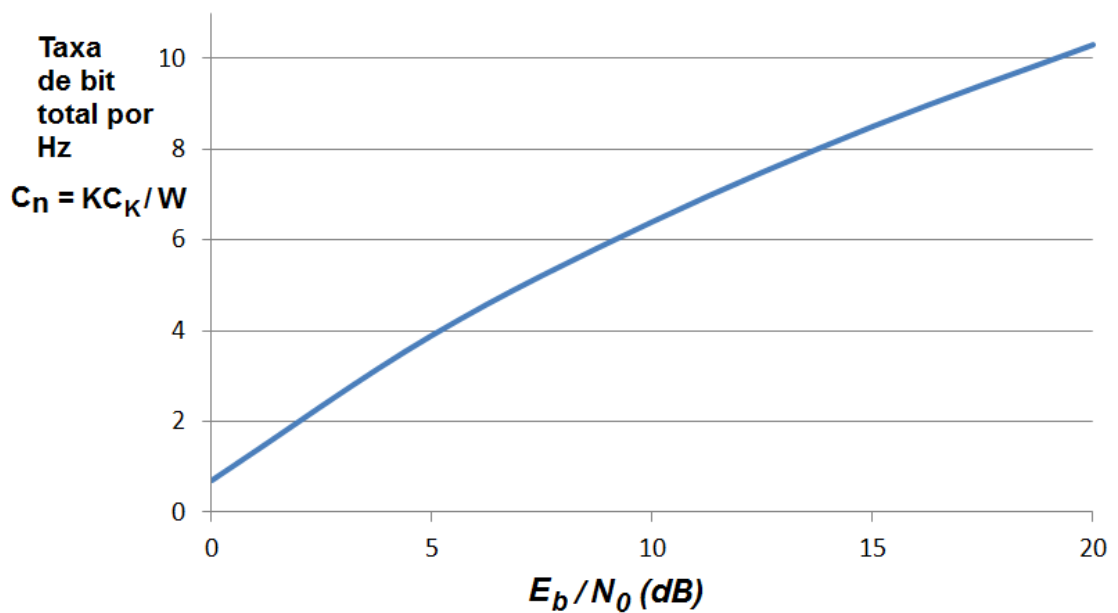
Uma forma mais compacta da Equação 2.4 é obtida definindo-se a capacidade total normalizada  $C_n = KC_K/W$ , que é a taxa total de bit para todos os  $K$  usuários por unidade de banda. Portanto, a Equação 2.4 pode ser expressa por

$$C_n = \log_2 \left( 1 + C_n \frac{E_b}{N_0} \right) \quad (2.5)$$

ou, equivalentemente,

$$\frac{E_b}{N_0} = \frac{2^{C_n} - 1}{C_n} \quad (2.6)$$

O gráfico de  $C_n$  versus  $E_b/N_0$  é mostrado na Figura 2.1. Observa-se que  $C_n$  aumenta com o aumento de  $E_b/N_0$  sobre o valor mínimo de  $\ln 2$ .



**Figura 2.1:** Capacidade total por Hertz como função de  $E_b/N_0$  para o FDMA. (Figura adaptada de [11].)

Em um sistema TDMA, cada usuário transmite por  $1/K$  do tempo no canal de banda  $W$ , com potência média  $KP$ . Portanto, a capacidade por usuário é

$$C_K = \frac{1}{K} W \log_2 \left( 1 + \frac{KP}{WN_0} \right) \quad (2.7)$$

que é idêntica à capacidade de um sistema FDMA. No entanto, do ponto de vista prático, deve-se enfatizar que no TDMA pode não ser possível para os transmissores sustentarem uma potência de transmissão  $KP$  quando  $K$  é muito grande [12, 11]. Portanto, existe um limite prático além do qual não se consegue mais aumentar a potência do transmissor com o aumento de  $K$ .

Em um sistema CDMA cada usuário transmite um sinal pseudo aleatório de banda  $W$  e potência média  $P$ . A capacidade do sistema depende do nível de cooperação entre os  $K$  usuários. Pode-se dizer que em um extremo está o CDMA não cooperativo, no qual o receptor de cada sinal de usuário não conhece os códigos e as formas de onda dos sinais de espalhamento dos outros usuários, ou decide ignorá-los no processo de demodulação. A este tipo de detecção, que caracteriza o sistema, dá-se o nome de detecção *single-user*. Portanto, os sinais dos outros usuários aparecem como interferência no receptor de cada usuário. Neste caso, o receptor multiusuário consiste de um banco de  $K$  filtros casados do tipo *single-user*. Se for assumido que cada código ou sinal de espalhamento de cada usuário é gaussiano, então cada sinal de usuário é afetado por interferência gaussiana de potência  $(K-1)P$  e ruído gaussiano aditivo de potência  $WN_0$ . Portanto, a capacidade por usuário por detecção *single-user* é

$$C_K = W \log_2 \left[ 1 + \frac{P}{WN_0 + (K-1)P} \right] \quad (2.8)$$

ou, equivalentemente,

$$\frac{C_K}{W} = \log_2 \left[ 1 + \frac{C_K}{W} \frac{E_b/N_0}{1 + (K-1)(C_K/W)(E_b/N_0)} \right] \quad (2.9)$$

A Figura 2.2 ilustra o gráfico de  $C_K/W$  versus  $E_b/N_0$ , com  $K$  como parâmetro.

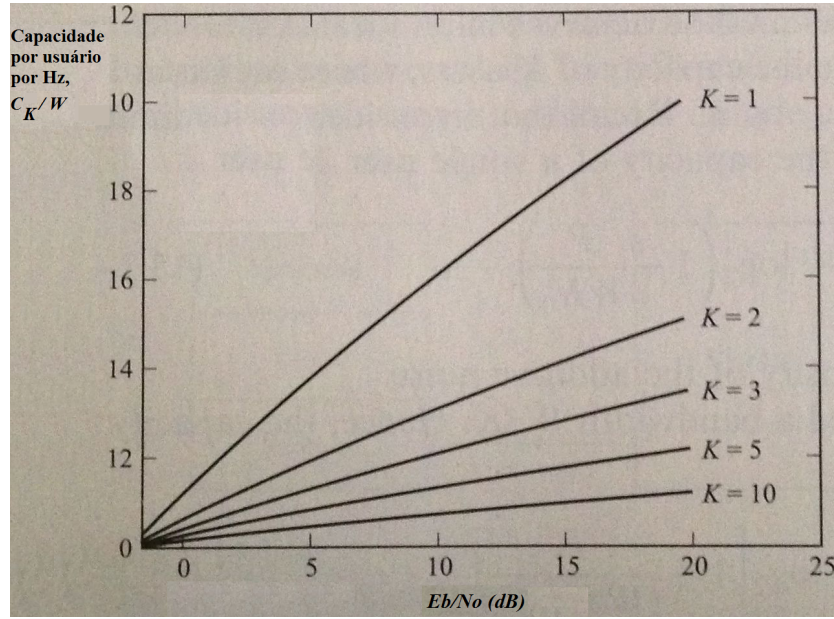
Para um número grande de usuários, pode-se utilizar a aproximação  $\ln(1+x) \leq x$ . Assim,

$$\frac{C_K}{W} \leq \frac{C_K}{W} \frac{E_b/N_0}{1 + K(C_K/W)(E_b/N_0)} \log_2 e \quad (2.10)$$

ou, equivalentemente, a capacidade total normalizada  $C_n = KC_K/W$  é

$$C_n \leq \log_2 e - \frac{1}{E_b/N_0} \leq \frac{1}{\ln 2} - \frac{1}{E_b/N_0} < \frac{1}{\ln 2} \quad (2.11)$$

Neste caso, observa-se que a capacidade total não aumenta com  $K$  como no



**Figura 2.2:** Capacidade normalizada como função de  $E_b/N_0$  para o FDMA. (Figura modificada de [11].)

TDMA, ou no FDMA.

Suponhamos agora que  $K$  usuários cooperam transmitindo seus sinais codificados sincronamente no tempo, e o receptor multiusuário demodula e decodifica conjuntamente todos os sinais dos usuários. A este tipo de detecção dá-se o nome de detecção multiusuário. A cada usuário é estabelecida uma taxa  $R_i$ ,  $1 \leq i \leq K$ , e um grupo de códigos (*code book*) contendo um conjunto de  $2^{nR_i}$  palavras códigos de potência  $P$ . Em cada intervalo de sinal, cada usuário seleciona uma palavra código arbitrária, por exemplo  $\mathbf{X}_i$ , do seu próprio *code book* e todos os usuários transmitem suas palavras códigos simultaneamente. Portanto, o decodificador no receptor observa

$$\mathbf{Y} = \sum_{i=1}^K \mathbf{X}_i + \mathbf{Z} \quad (2.12)$$

onde  $\mathbf{Z}$  é um vetor de ruído aditivo. O decodificador ótimo procura as  $K$  palavras código, uma de cada *code book*, que tem a soma vetorial mais próxima possível do vetor  $\mathbf{Y}$  em distância Euclideana.

A região de taxa  $K$  dimensional alcançável para os  $K$  usuários em um canal AWGN, assumindo potências iguais para todos os usuários, é dada pelas seguintes equações:

$$R_i < W \log_2 \left( 1 + \frac{P}{WN_0} \right), \quad 1 \leq i \leq K \quad (2.13)$$

$$R_i + R_j < W \log_2 \left( 1 + \frac{2P}{WN_0} \right), \quad 1 \leq i, j \leq K \quad (2.14)$$

$$R_{SUM}^{MU} = \sum_{i=1}^K R_i < W \log_2 \left( 1 + \frac{KP}{WN_0} \right) \quad (2.15)$$

onde  $R_{SUM}^{MU}$  é a taxa total alcançada pelos  $K$  usuários empregando detecção multiusuário. No caso especial em que todas as taxas são idênticas, a desigualdade (2.15) torna-se dominante sobre as outras  $(K - 1)$ .

### 2.2.3 Acesso múltiplo por divisão de código (CDMA)

Conforme observa-se, TDMA e FDMA são métodos de acesso múltiplo nos quais o canal é dividido em subcanais *single-user*, independentes, isto é, em janelas de tempo ou em bandas de frequências não sobrepostas, respectivamente. No CDMA, uma sequência distinta de assinatura (ou código de espalhamento) é atribuída a cada usuário, com a qual o usuário modula e espalha o sinal que contém a informação. A sequência de assinatura permite ao receptor demodular a mensagem transmitida por múltiplos usuários do canal, que transmitem simultaneamente e assincronamente.

#### 2.2.3.1 Modelos de sinal CDMA

Seja um canal CDMA que é compartilhado por  $K$  usuários simultaneamente. A cada usuário é atribuída uma forma de onda de assinatura  $s_k(t)$  de duração  $T$ , onde  $T$  é o intervalo de símbolo. A forma de onda da assinatura pode ser expressa por

$$s_k(t) = \sum_{n=0}^{L-1} a_k(n)p(t - nT_c), \quad 0 \leq t \leq T. \quad (2.16)$$

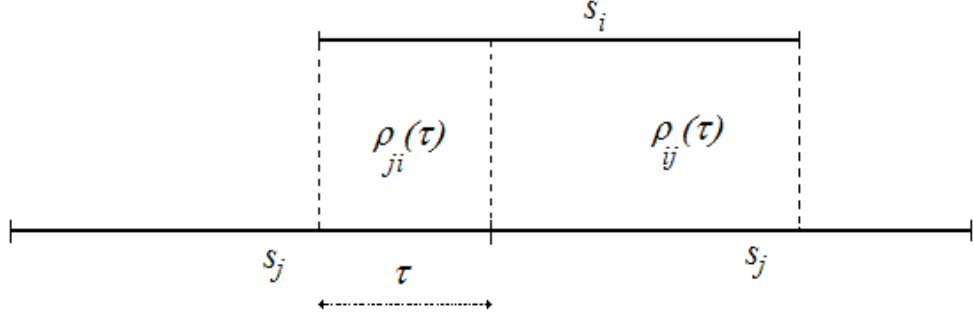
onde  $\{a_k(n), 0 \leq n \leq L - 1\}$  é uma sequência pseudo aleatória consistindo de  $L$  “chips” que tomam valores  $\{\pm 1\}$ ,  $p(t)$  é um pulso de duração  $T_c$ , e  $T_c$  é o intervalo de *chip*. Assim, tem-se  $L$  chips por símbolo e  $T = LT_c$ . Sem perda de generalidade, assume-se que todos os  $K$  sinais de espalhamento têm energia unitária, isto é,

$$\|s_k\|^2 = \int_0^T s_k^2(t)dt = 1 \quad (2.17)$$

As correlações cruzadas entre pares de sinais de espalhamento têm um papel importante nas métricas para o detector de sinal e sobre o seu desempenho. Definem-se as seguintes correlações cruzadas, onde  $0 \leq \tau \leq T$  e  $i < j$ ,

$$\rho_{ij}(\tau) = \int_{\tau_c}^T s_i(t)s_j(t - \tau)dt \quad (2.18)$$





**Figura 2.3:** Definição das correlações cruzadas ( $i < j$ ).

e

$$\rho_{ji}(\tau) = \int_0^\tau s_i(t) s_j(t + T - \tau) dt \quad (2.19)$$

note-se que o primeiro subíndice de  $\rho_{ij}(\tau)$  denota o sinal da esquerda na correlação (veja Figura 2.3).

As correlações cruzadas nas Equações 2.18 e 2.19 aplicam-se a transmissões assíncronas entre  $K$  usuários. Para transmissão síncrona, necessita-se somente de  $\rho_{ij}$ .

Assume-se que sinais binários antipodais são usados para transmitir a informação de cada usuário. Assim, seja a sequência de informação do  $k$ -ésimo usuário, representada por  $b_k(m)$ , onde o valor de cada bit de informação pode ser  $\pm 1$ . É conveniente considerar a transmissão de um bloco de bits de algum comprimento arbitrário, por exemplo  $N$ . Então, o bloco de dados do  $k$ -ésimo usuário é

$$\mathbf{b}_k = [b_k(1), \dots, b_k(N)]^T \quad (2.20)$$

e o sinal do filtro passa-baixas equivalente passa-baixas transmitido pode ser expresso como

$$g_k(t) = \sqrt{(E_b)_k} \sum_{i=1}^N b_k(i) s_k(t - iT) \quad (2.21)$$

onde  $(E_b)_k$  é a energia do sinal por bit. O sinal composto transmitido pelos  $K$  usuários pode ser expresso por

$$g(t) = \sum_{k=1}^K g_k(t - \tau_k) = \sum_{k=1}^K \sqrt{(E_b)_k} \sum_{i=1}^N b_k(i) s_k(t - iT - \tau_k) \quad (2.22)$$

onde  $\{\tau_k\}$  são os atrasos de transmissão, que satisfazem a condição  $0 \leq \tau_k < T$

para  $1 \leq k \leq K$ . Assume-se que o sinal transmitido é corrompido por AWGN. Assim, o sinal recebido pode ser expresso por

$$r(t) = g(t) + n(t) \quad (2.23)$$

onde  $g(t)$  é dado pela Equação 2.22 e  $n(t)$  é o ruído, com densidade espectral de potência  $\frac{1}{2}N_0$ .

### 2.2.3.2 Receptor ótimo

O receptor ótimo é definido como o receptor que seleciona a sequência de bits mais provável de ter sido enviada pelo transmissor,  $\{b_k(n), 1 \leq n \leq N, 1 \leq k \leq K\}$ , dado o sinal recebido  $r(t)$ , observado dentro do intervalo de tempo  $0 \leq t \leq NT + 2T$ . Primeiramente, será visto o caso da transmissão síncrona; e após, será repassada a transmissão assíncrona.

**Transmissão síncrona.** Na transmissão síncrona, cada (usuário) interferente produz exatamente um símbolo que interfere com o símbolo desejado. Considerando o caso do AWGN, é suficiente se for considerado o sinal recebido em um intervalo de símbolo, isto é  $0 \leq t \leq T$ , e determinar o receptor ótimo. Assim,  $r(t)$  pode ser expresso como

$$r(t) = \sum_{k=1}^K \sqrt{(E_b)_k} b_k(1) s_k(t) + n(t), \quad 0 \leq t \leq T \quad (2.24)$$

O receptor de máxima verossimilhança (ML, *maximum likelihood*) ótimo calcula a função de verossimilhança:

$$\Lambda_{sinc}(\mathbf{b}) = \int_0^T \left[ r(t) - \sum_{k=1}^K \sqrt{(E_b)_k} b_k(1) s_k(t) \right]^2 dt \quad (2.25)$$

e seleciona a sequência de informação  $\{b_k(1), 1 \leq k \leq K\}$  que minimiza  $\Lambda_{sinc}(\mathbf{b})$ .

Se for expandida a integral na Equação 2.25, obtém-se:

$$\begin{aligned} \Lambda_{sinc}(\mathbf{b}) = & \int_0^T r^2(t) dt - 2 \sum_{k=1}^K \sqrt{(E_b)_k} b_k(1) \int_0^T r(t) s_k(t) dt + \\ & + \sum_{j=1}^K \sum_{k=1}^K \sqrt{(E_b)_j (E_b)_k} b_j(1) b_k(1) \int_0^T s_k(t) s_j(t) dt \end{aligned} \quad (2.26)$$

Pode-se observar que a integral envolvendo  $r^2(t)$  é comum a todas as sequências possíveis  $\{b_k(1)\}$  e não tem relevância na determinação de qual sequência foi

transmitida. Portanto pode ser desconsiderada. O termo

$$r_k = \int_0^T r(t) s_k(t), \quad 1 \leq k \leq K \quad (2.27)$$

representa a correlação cruzada entre o sinal recebido e cada uma das  $K$  sequências de assinatura. Ao invés de se usar correladores, filtros casados podem ser utilizados. Finalmente, a integral envolvendo  $s_k(t)$  e  $s_j(t)$  é dada por:

$$\rho_{jk} = \int_0^T s_j(t) s_k(t) \quad (2.28)$$

Essas métricas de correlação podem ser expressas também na forma de produto interno vetorial como

$$C(\mathbf{r}_K, \mathbf{b}_K) = 2 \mathbf{b}_K^T \mathbf{r}_K - \mathbf{b}_K^T \mathbf{R}_s \mathbf{b}_K \quad (2.29)$$

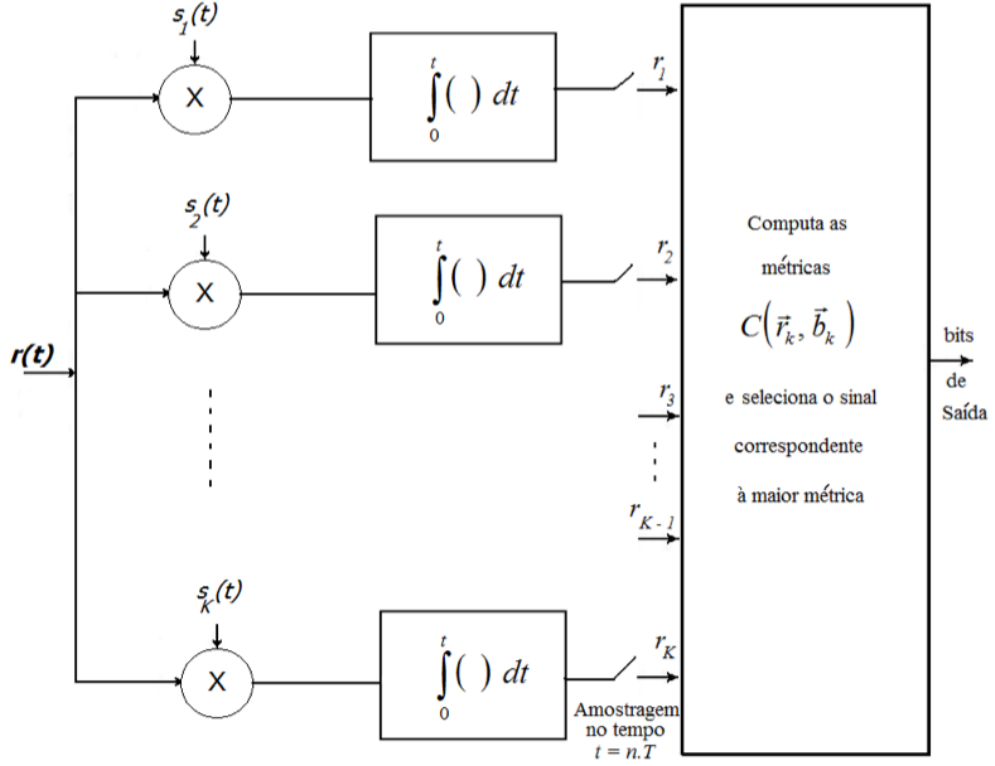
onde:

$$\mathbf{r}_K = [r_1 \ r_2 \ \dots \ r_K];$$

$$\mathbf{b}_K = [\sqrt{(E_b)_1} b_1(1) \ \dots \ \sqrt{(E_b)_K} b_K(1)]^T;$$

e  $\mathbf{R}_s$  é a matriz de correlação, com elementos  $\rho_{jk}(0)$ . Observa-se que o detector ótimo deve necessariamente ter conhecimento das energias do sinal recebido de forma que possa calcular as métricas da correlação. A Figura 2.4 mostra o receptor multiusuário ótimo. De fato, existem  $2^K$  possibilidades de escolha dos bits na sequência de informação dos  $K$  usuários. O detector ótimo calcula as métricas da correlação para cada sequência e seleciona a sequência que origina a maior dentre todas as métricas da correlação. Observa-se que o detector ótimo tem uma complexidade que cresce exponencialmente com o número de usuários,  $K$ . Em resumo, o receptor ótimo para transmissão de símbolos sincronizados consiste em um banco de  $K$  correladores ou filtros casados, seguidos por um detector que computa as  $2^K$  métricas de correlações dadas pela Equação 2.29. Os filtros casados podem ser vistos na Figura 2.5, correspondentes às  $2^K$  possíveis sequências de informação transmitidas. Assim, o detector seleciona a sequência correspondente à maior métrica de correlação.

**Transmissão assíncrona.** No modelo síncrono considerado nos parágrafos anteriores os intervalos de bit estão alinhados no receptor. Para o caso assíncrono, existem dois símbolos consecutivos de cada interferente que se sobrepõem a um símbolo desejado. Assume-se que o receptor conhece as energias do sinal recebido  $\{(E_b)_k\}$  para os  $K$  usuários e os atrasos de transmissão  $\{\tau_k\}$ . Certamente, esses parâmetros devem necessariamente ser medidos no receptor ou fornecidos ao receptor como informação adicional pelos outros usuários por meio de algum



**Figura 2.4:** Receptor ótimo multiusuário de transmissão síncrona.

canal de controle.

O receptor ótimo, assíncrono, de máxima verossimilhança (ML, *maximum likelihood*) computa a função de verossimilhança:

$$\begin{aligned}
 \Lambda_{asinc}(\mathbf{b}) &= \int_0^{NT+2T} \left[ r(t) - \sum_{k=1}^K \sqrt{(E_b)_k} \sum_{i=1}^N b_k(i) s_k(t - iT - \tau_k) \right]^2 dt = \\
 &= \int_0^{NT+2T} r^2(t) dt - 2 \sum_{k=1}^K \sqrt{(E_b)_k} \sum_{i=1}^N b_k(i) \int_0^{NT+2T} r(t) s_k(t - iT - \tau_k) dt + \\
 &+ \sum_{k=1}^K \sum_{l=1}^K \sqrt{(E_b)_k (E_b)_l} \sum_{i=1}^N \sum_{j=1}^N b_k(i) b_l(j) \int_0^{NT+2T} s_k(t - iT - \tau_k) s_l(t - jT - \tau_l) dt
 \end{aligned} \tag{2.30}$$

onde  $\mathbf{b}$  representa as sequências de dados dos  $K$  usuários. A integral envolvendo  $r^2(t)$  pode ser ignorada, uma vez que é um termo comum a todas as sequências de informação possíveis. A integral

$$r_k(i) \equiv \int_{iT+\tau_k}^{(i+1)T+\tau_k} r(t) s_k(t - iT - \tau_k) dt, \quad 1 \leq i \leq N \tag{2.31}$$

representa as saídas do correlator, ou do filtro-casado, para o  $k$ -ésimo usuário

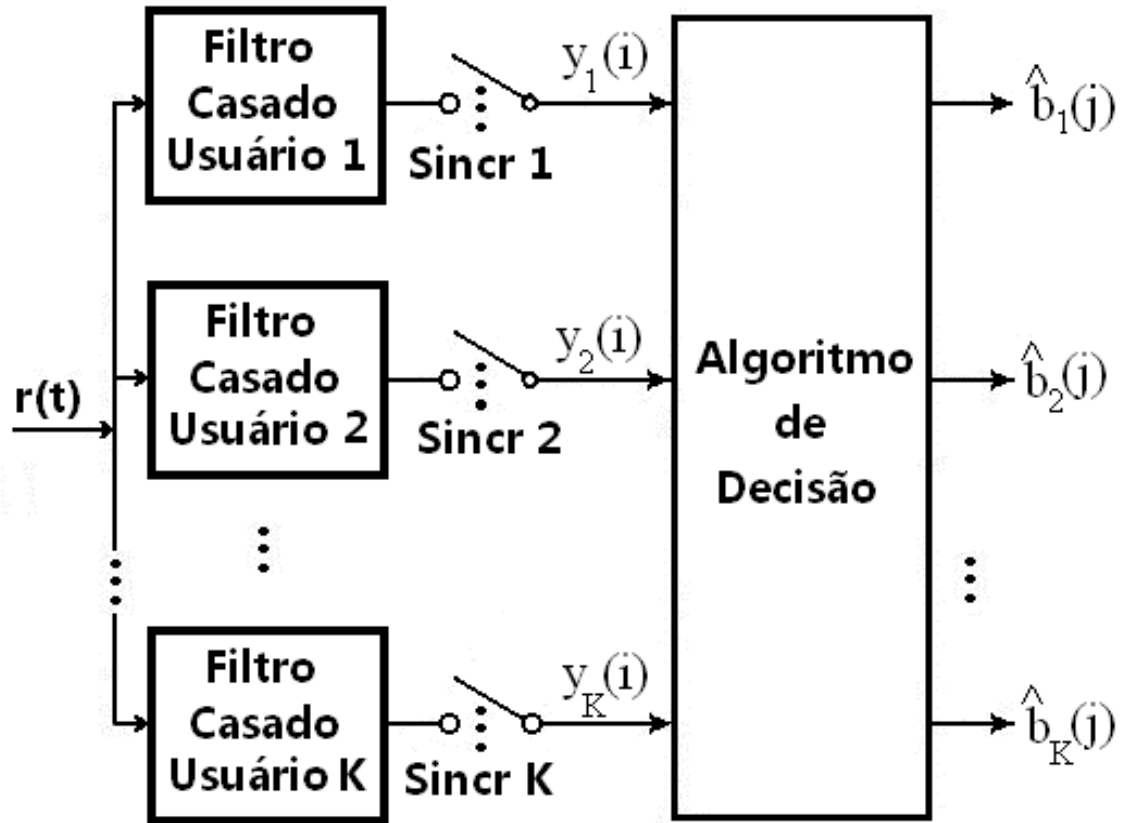


Figura 2.5: Receptor ótimo síncrono com filtros casados.

em todos os intervalos de sinais. Finalmente, a integral

$$\begin{aligned}
 \int_0^{NT+2T} s_k(t - iT - \tau_k) s_l(t - jT - \tau_l) dt &= \\
 &= \int_{-iT - \tau_k}^{NT+2T - iT - \tau_k} s_k(t) s_l(t + iT - jT + \tau_k - \tau_l) dt
 \end{aligned} \quad (2.32)$$

pode ser facilmente decomposta em termos envolvendo a correlação cruzada  $\rho_{kl}(\tau) = \rho_{kl}(\tau_l - \tau_k)$  para  $k \leq 1$  e  $\rho_{ik}(\tau)$  para  $k > 1$ . Portanto, observa-se que a função de verossimilhança pode ser expressa em termos de uma métrica de correlação que envolve as saídas  $\{r_k(i), 1 \leq k \leq K, 1 \leq i \leq N\}$  de  $K$  correlatores ou filtros casados, um para cada sequência de espalhamento. Utilizando-se notação vetorial, pode-se mostrar que as  $NK$  saídas do correlator ou do filtro casado  $\{r_k(i)\}$  pode ser expressa na forma

$$\mathbf{r} = \mathbf{R}_N \mathbf{b} + \mathbf{n} \quad (2.33)$$

onde, por definição

$$\begin{aligned}\mathbf{r} &= [\mathbf{r}^T(1) \ \mathbf{r}^T(2) \ \cdots \ \mathbf{r}^T(N)]^T \\ \mathbf{r}(i) &= [r_1(i) \ r_2(i) \ \cdots \ r_K(i)]^T \\ \mathbf{b} &= [\mathbf{b}^T(1) \ \mathbf{b}^T(2) \ \cdots \ \mathbf{b}^T(N)]^T\end{aligned}\tag{2.34}$$

$$\mathbf{b}(i) = [\sqrt{E_1} b_1(i) \ \sqrt{E_2} b_2(i) \ \cdots \ \sqrt{E_K} b_K(i)]^T\tag{2.35}$$

$$\begin{aligned}\mathbf{n} &= [\mathbf{n}^T(1) \ \mathbf{n}^T(2) \ \cdots \ \mathbf{n}^T(N)]^T \\ \mathbf{n}(i) &= [n_1(i) \ n_2(i) \ \cdots \ n_K(i)]^T\end{aligned}\tag{2.36}$$

$$\mathbf{R}_N = \begin{pmatrix} \mathbf{R}_a(0) & \mathbf{R}_a^T(1) & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{R}_a(1) & \mathbf{R}_a(0) & \mathbf{R}_a^T(1) & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}(1) & \mathbf{R}_a(0) & \mathbf{R}_a^T(1) \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{R}_a(1) & \mathbf{R}_a(0) \end{pmatrix}\tag{2.37}$$

e  $\mathbf{R}_a(m)$  é uma matriz  $K \times K$  com elementos:

$$R_{kl}(m) = \int_{-\infty}^{\infty} s_k(t - \tau_k) s_l(t + mT - \tau_k) dt\tag{2.38}$$

Os vetores gaussianos de ruído  $\mathbf{n}(i)$  tem média zero e matriz de autocorrelação:

$$\mathbb{E}[\mathbf{n}(k) \mathbf{n}^T(j)] = \frac{1}{2} N_0 \mathbf{R}_a(k - j)\tag{2.39}$$

Pode-se notar que o vetor  $\mathbf{r}$  dado pela Equação 2.33 constitui um conjunto de estatísticas suficientes para se estimarem os bits transmitidos  $b_k(i)$ . Se por exemplo for adotado um enfoque de processamento em bloco, o detector de máxima verossimilhança (ML, *maximum likelihood*) ótimo deve computar  $(2NK)$  métricas de correlações e selecionar as  $K$  sequências de comprimento  $N$  que correspondem à maior métrica de correlação. Claramente, tal abordagem é muito complexa para ser efetivada computacionalmente na prática, especialmente quando  $K$  e  $N$  são grandes. Uma abordagem alternativa é a estimação de sequência ML utilizando o algoritmo de Viterbi. Para se construir um detector do tipo sequencial, faz-se uso do fato de que todo símbolo transmitido sobrepõe-se no máximo com  $(2K - 2)$

símbolos. Portanto, uma redução significativa na complexidade computacional é obtida com respeito ao parâmetro  $N$ , de tamanho do bloco, mas a dependência exponencial de  $K$  não pode ser reduzida.

É aparente que o receptor ML ótimo que emprega o algoritmo de Viterbi tem sua própria complexidade computacional muito alta, e assim, seu uso na prática torna-se viável somente quando o número de usuários é extremamente pequeno, em torno de  $K < 10$ . Para valores maiores de  $K$ , deve-se utilizar detectores conhecidos como subótimos, cuja complexidade cresce linearmente com  $K$ .

### 2.2.3.3 Detectores subótimos

Na discussão anterior, observou-se que um detector ótimo para os  $K$  usuários de um sistema de acesso múltiplo por divisão de código tem uma complexidade computacional que cresce com  $K$  exponencialmente. Essa complexidade é medida pelo número de operações aritméticas (adições, multiplicações, divisões) por símbolo modulado. Nesta subseção, serão descritos alguns detectores subótimos, cuja complexidade computacional varia linearmente com o número de usuários  $K$ . O primeiro detector, que é o mais simples, é o detector convencional, também denominado de detector *single-user*.

#### Detector convencional (*single-user*)

Na detecção convencional, o receptor de cada usuário consiste em um demodulador que correlaciona, ou filtra, o sinal recebido com a sequência de espalhamento do usuário, e passa a saída do correlator ao detector, que toma a decisão baseado na saída de um único correlator. Assim, o detector convencional não leva em conta a presença de outros usuários que utilizam o canal simultaneamente, ou, equivalentemente, assume que o ruído mais interferência é ruído branco e gaussiano.

Considerando uma transmissão síncrona, a saída do correlator para o  $k$ -ésimo usuário, para um sinal no intervalo  $0 \leq t \leq T$  é

$$r_k = \int_0^T r(t) s_k(t) dt \quad (2.40)$$

$$= \sqrt{(E_b)_k} b_k(1) + \sum_{\substack{j=1 \\ j \neq k}}^K \sqrt{(E_b)_k} b_j(1) \rho_{jk}(0) + n_k(1) \quad (2.41)$$

onde o componente do ruído  $n_k(1)$  é dado por:

$$n_k(1) = \int_0^T n(t) s_k(t) dt \quad (2.42)$$

sendo  $n(t)$  um ruído branco gaussiano com densidade espectral de potência  $\frac{1}{2}N_0$ . Assim, sua variância é

$$\mathbb{E}[n_k^2(1)] = \frac{1}{2}N_0 \int_0^T s_k^2(t)dt = \frac{1}{2}N_0 \quad (2.43)$$

Se as sequências de espalhamento fossem ortogonais as interferências dos outros usuários, dadas pelo termo central da Equação 2.41, claramente desapareceriam e o detector convencional (*single-user*) seria ótimo. De outra forma, se uma ou mais das outras sequências de espalhamento não forem ortogonais à sequência de espalhamento do usuário desejado, as interferências dos outros usuário podem tornar-se excessivas. Quando o nível de potência do sinal de um ou mais usuários for maior do que o nível de potência do usuário  $k$ , ocorrerá a situação que é conhecida em comunicações multiusuários como problema *near-far*. Neste caso, requer-se um controle de potência eficiente para se utilizar a detecção convencional.

Na transmissão assíncrona, o detector convencional torna-se mais vulnerável à interferências de outros usuários. Isso ocorre devido ao fato de que não é possível projetar sequências de espalhamento para qualquer par de usuários que sejam ortogonais para todos os atrasos (*time-offsets*). Consequentemente, interferência de outros usuários é algo inevitável em transmissão assíncrona com detecção *single-user* convencional. O problema *near-far* resultante das diferenças entre níveis de potências nos sinais transmitidos por vários usuários é sério. A solução prática requer um método de ajuste de potência que seja controlado pelo receptor via canal de comunicação, o qual é monitorado continuamente por todos os usuários. Uma outra alternativa seria empregar um dos detectores multiusuário descritos a seguir.

### Detector de descorrelação

O detector convencional tem uma complexidade que cresce linearmente com o número de usuários, conforme foi observado. Outro tipo de detector que também tem complexidade computacional linear mas que não exibe uma vulnerabilidade às interferências de outros usuários é o detector de descorrelação.

Primeiramente será considerado o caso síncrono. Neste caso, o vetor do sinal recebido  $\mathbf{r}$  que representa a saída dos  $K$  filtros casados é:

$$\mathbf{r}_K = \mathbf{R}_s \mathbf{b}_K + \mathbf{n}_K \quad (2.44)$$

onde  $\mathbf{b}_K = [\sqrt{E_1}b_1(1) \ \sqrt{E_2}b_2(1) \ \cdots \ \sqrt{E_K}b_K(1)]^T$ , e o vetor do ruído com ele-



mentos  $\mathbf{n}_K = [n_1(1) \ n_2(1) \ \dots \ n_K(1)]^T$  tem covariância

$$\mathbb{E}(\mathbf{n}_K \mathbf{n}_K^T) = \frac{N_0}{2} \mathbf{R}_s \quad (2.45)$$

Sendo o ruído gaussiano,  $\mathbf{r}$  é descrito por uma função densidade de probabilidade (PDF) gaussiana  $K$ -dimensional com média  $\mathbf{R}_s \mathbf{b}_K$  e covariância  $\mathbf{R}_s$ , isto é:

$$p(\mathbf{r}_K | \mathbf{b}_K) = \frac{1}{\sqrt{(N_0 \pi)^K \det \mathbf{R}_s}} \exp \left[ -\frac{1}{N_0} (\mathbf{r}_K - \mathbf{R}_s \mathbf{b}_K)^T \mathbf{R}_s^{-1} (\mathbf{r}_K - \mathbf{R}_s \mathbf{b}_K) \right] \quad (2.46)$$

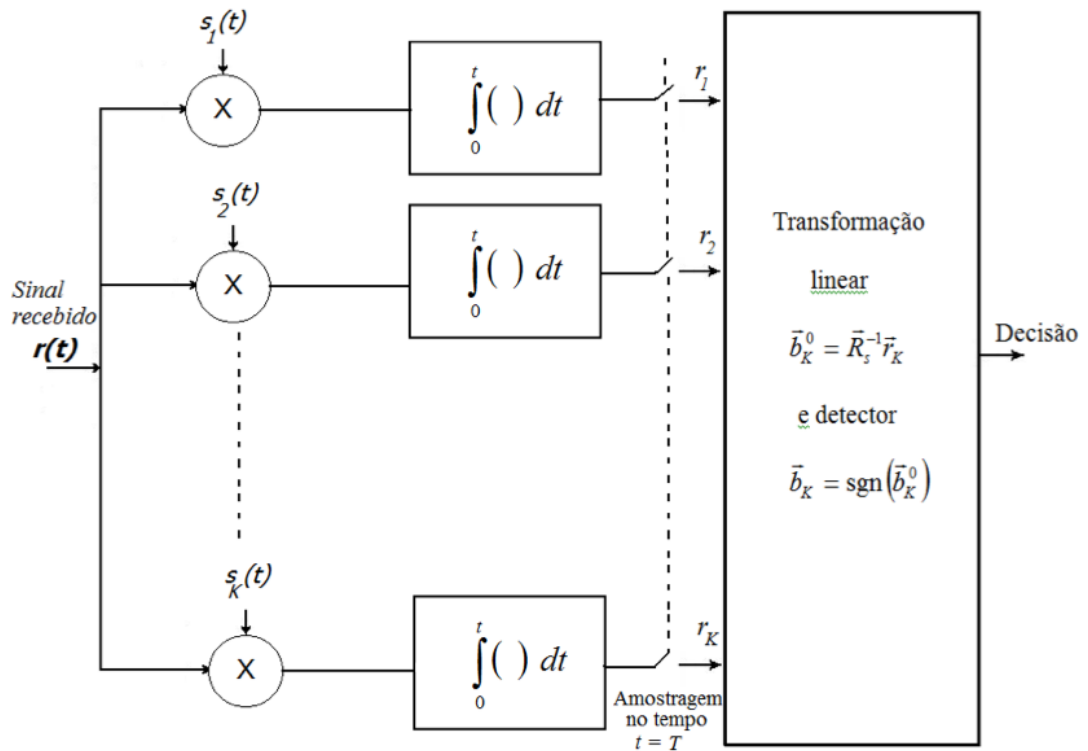
O melhor estimador linear de  $\mathbf{b}_K$  é aquele que minimiza a função de verossimilhança:

$$\Lambda(\mathbf{b}_K) = (\mathbf{r}_K - \mathbf{R}_s \mathbf{b}_K)^T \mathbf{R}_s^{-1} (\mathbf{r}_K - \mathbf{R}_s \mathbf{b}_K) \quad (2.47)$$

O resultado desta minimização é igual a:

$$\mathbf{b}_K^0 = \mathbf{R}_s^{-1} \mathbf{r}_K \quad (2.48)$$

A Figura 2.6 ilustra a estrutura do receptor. Uma vez que a estimação  $\mathbf{b}_K^0$  foi obtida desenvolvendo-se uma transformação linear sobre as saídas do vetor de correladores, a complexidade computacional é linear em  $K$ .



**Figura 2.6:** Estrutura do receptor do descodificador.

Deve-se observar que a melhor (máxima verossimilhança) estimação linear de  $\mathbf{b}_K$  dada pela Equação 2.48 é diferente da do detector de sequência ML não linear e ótimo, que encontra a melhor sequência de valores discretos  $\{\pm 1\}$  e que maximiza a função de verossimilhança. É interessante também notar que a estimação  $b_K^0$  é a melhor estimação linear que maximiza a métrica de correlação dada pela Equação 2.29. A transformação  $\mathbf{R}_s^{-1}$  elimina os componentes de interferência entre os usuários. Consequentemente, o problema *near-far* é eliminado e não há necessidade de controle de potência.

No caso da transmissão assíncrona, o sinal recebido na saída dos correladores é dado pela Equação 2.33. Assim, a função de verossimilhança é dada por

$$\Lambda(\mathbf{b}) = (\mathbf{r} - \mathbf{R}_N \mathbf{b})^T \mathbf{R}_N^{-1} (\mathbf{r} - \mathbf{R}_N \mathbf{b}) \quad (2.49)$$

onde  $\mathbf{R}_N$  é definido pela Equação 2.37 e  $\mathbf{b}$  é dado pela Equação 2.35. O vetor  $\mathbf{b}$  que minimiza  $\Lambda(\mathbf{b})$  é:

$$\mathbf{b}^0 = \mathbf{R}_N^{-1} \mathbf{r} \quad (2.50)$$

Esta é a estimação ML de  $\mathbf{b}$  que é obtida novamente pelo desenvolvimento da transformação linear das saídas do banco de correladores.

O vetor  $\mathbf{r}$  é dado por  $\mathbf{r} = \mathbf{R}_N \mathbf{b} + \mathbf{n}$ , assim, segue da Equação 2.50 que

$$\mathbf{b}^0 = \mathbf{b} + \mathbf{R}_N^{-1} \mathbf{n} \quad (2.51)$$

Portanto,  $\mathbf{b}^0$  é uma estimação não viesada de  $\mathbf{b}$ . Isso significa que a interferência multiusuário foi eliminada, da mesma forma que no caso da transmissão síncrona. Assim, esse detector para transmissão assíncrona é também chamado de descorrelacionador. Existem diversos métodos que podem ser utilizados para inverter a matriz  $\mathbf{R}_N$ . Métodos iterativos para descorrelacionar os sinais têm sido também utilizados.

### Detector de erro mínimo médio quadrático

Na discussão feita anteriormente, foi visto que a estimação linear ML de  $\mathbf{b}$  é obtida minimizando-se a função de verossimilhança quadrática dada na Equação 2.28. Assim, obteve-se o resultado dado pela Equação 2.50, que é uma estimação derivada desenvolvendo-se uma transformação linear sobre as saídas do banco de correladores ou filtros casados.

Outra solução pode ser obtida se for utilizada uma transformação linear do tipo  $\mathbf{b}^0 = \mathbf{M} \mathbf{r}$ , onde a matriz  $\mathbf{M}$  deve ser determinada de forma a minimizar o

erro médio quadrático (MSE, *mean square error*).

$$J(\mathbf{b}) = \mathbb{E}[(\mathbf{b} - \mathbf{b}^0)^T (\mathbf{b} - \mathbf{b}^0)] = \mathbb{E}[(\mathbf{b} - \mathbf{M}\mathbf{r})^T (\mathbf{b} - \mathbf{M}\mathbf{r})] \quad (2.52)$$

onde a esperança é calculada levando-se em conta o vetor de dados  $\mathbf{b}$  e o ruído aditivo  $\mathbf{n}$ . A matriz  $\mathbf{M}$  ótima pode ser encontrada ao fazer-se com que o erro  $(\mathbf{b} - \mathbf{M}\mathbf{r})$  seja ortogonal ao vetor de dados  $\mathbf{r}$ . Assim,

$$\begin{aligned} \mathbb{E}[(\mathbf{b} - \mathbf{M}\mathbf{r})\mathbf{r}^T] &= \mathbf{0} \\ \mathbb{E}(\mathbf{b}\mathbf{r}^T) - \mathbf{M}\mathbb{E}(\mathbf{r}\mathbf{r}^T) &= \mathbf{0} \end{aligned} \quad (2.53)$$

Considerando o caso síncrono tem-se

$$\mathbb{E}(\mathbf{b}_K\mathbf{r}_K^T) = \mathbb{E}(\mathbf{b}_K\mathbf{b}_K^T)\mathbf{R}_s^T = \mathbf{D}\mathbf{R}_s^T \quad (2.54)$$

e

$$\mathbb{E}(\mathbf{r}_K\mathbf{r}_K^T) = \mathbb{E}[(\mathbf{R}_s\mathbf{b}_K + \mathbf{n}_K)(\mathbf{R}_s\mathbf{b}_K + \mathbf{n}_K)^T] = \mathbf{R}_s\mathbf{D}\mathbf{R}_s^T + \frac{N_0}{2}\mathbf{R}_s^T \quad (2.55)$$

onde  $\mathbf{D}$  é uma matriz diagonal com elementos  $\{E_k, 1 \leq k \leq K\}$ . Substituindo-se as Equações 2.54 e 2.55 na Equação 2.53 e resolvendo com relação à  $\mathbf{M}$ , obtém-se:

$$\mathbf{M}^0 = \left(\mathbf{R}_s + \frac{N_0}{2}\mathbf{D}^{-1}\right)^{-1} \quad (2.56)$$

Então,

$$\mathbf{b}_K^0 = \mathbf{M}^0\mathbf{r}_K \quad (2.57)$$

e

$$\hat{\mathbf{b}}_K = \text{sgn}(\mathbf{b}_K^0) \quad (2.58)$$

Similarmente, para a transmissão assíncrona, pode ser demonstrado que a escolha ótima de  $\mathbf{M}$  que minimiza  $J(\mathbf{b})$  é

$$\mathbf{M}^0 = (\mathbf{R}_N + \frac{1}{2}N_0\mathbf{I})^{-1} \quad (2.59)$$

e a seguir,

$$\mathbf{b}^0 = (\mathbf{R}_N + \frac{1}{2}N_0\mathbf{I})^{-1}\mathbf{r} \quad (2.60)$$

A saída do detector é então  $\hat{\mathbf{b}} = \text{sgn}(\mathbf{b}^0)$ . A estimação dada pela Equação 2.57 ou 2.60 é chamada de estimação *MSE mínima* (MMSE) de  $\mathbf{b}$ . Nota-se que quando  $\frac{1}{2}N_0$  é pequeno, comparado com os elementos da diagonal de  $\mathbf{R}_N$ , a solução MMSE aproxima-se da solução ML dada pela Equação 2.50. Por outro

lado, quando o nível de ruído for grande, comparado com o nível de sinal nos elementos da diagonal de  $\mathbf{R}_N$ ,  $\mathbf{M}^0$  aproxima-se da matriz identidade. Neste caso de baixa SNR, o detector basicamente ignora as interferências dos outros usuários, porque o ruído aditivo é o termo dominante. Deve ser também observado que o critério MMSE produz uma estimação viesada de  $\mathbf{b}$ . Assim, existe um resíduo de interferência multiusuário.

Para realizar os cálculos que resultam nos valores de  $\mathbf{b}$ , resolve-se o conjunto de equações lineares

$$\left(\mathbf{R}_N + \frac{1}{2}N_0 \mathbf{I}\right)\mathbf{b} = \mathbf{r} \quad (2.61)$$

Assim, detectar  $NK$  bits requer  $3NK^2$  multiplicações. Portanto, a complexidade é  $3K$  multiplicações por bit, que é independente do comprimento bloco  $N$  e é linear em  $K$ .

Observa-se que ambos os detectores descritos acima, envolvem a realização de transformações lineares sobre um bloco de dados obtido a partir de  $K$  correlatores ou filtros casados. Estes detectores multiusuários para transmissão assíncrona podem ser construídos empregando-se um “filtro de linha de atraso com terminações” (*tapped-delay-line*), com coeficientes ajustáveis para cada usuário, e selecionando-se os coeficientes do filtro para eliminar a interferência entre usuários ou minimizar o MSE para cada sinal de usuário. Portanto, os bits de informação recebidos são sequencialmente estimados com um atraso finito, ao invés de se estimar por bloco.

### Outros tipos de detectores multiusuários

Devido ao grande interesse no desenvolvimento de sistemas de comunicação digital de acesso múltiplo, os desenvolvimentos de algoritmos de detecção multiusuário continua a ser uma área de pesquisa muito ativa, mesmo após quase trinta anos de sua descoberta [13]. O tratamento desenvolvido nesta seção está focado no algoritmo ótimo MLSE (*maximum-likelihood sequence estimation*), nos algoritmos lineares subótimos (detecção MMSE e de descorrelacionamento), e nos algoritmos de cancelamento sucessivo de interferência baseados em decisões “*hard*”, em oposição aos outros algoritmos já considerados, cuja decisão é classificada como “*soft*”.

Adicionalmente, a esses tipos de algoritmos relativamente simples, uma série de algoritmos mais complexos tem sido publicada na literatura [2] os quais são mais apropriados para canais dispersivos que resultam em ISI (*intersymbol interference*, ou interferência intersimbólica). Adicionalmente, pode-se assumir que o conhecimento dos sinais de espalhamento dos outros usuários não está disponível para o receptor de um usuário. Assim, um receptor de usuário tem que lidar

tanto com a ISI quanto com a MAI. Para tal cenário, torna-se necessário criar algoritmos de supressão de interferência, similares aos algoritmos utilizados em equalizadores [11].

Algoritmos adaptativos para suprimir ISI e MAI em sistemas multiusuários estão descritos em artigos [14], [15], [16], [17], e [18, 11], por exemplo. Em alguns casos, os algoritmos adaptativos são preparados para convergir sem utilizar sequências de treinamento. Tais algoritmos são chamados *blind multiuser detection algorithms*. Exemplos de algoritmos cegos são encontrados em artigos [19], [20], [21], [22, 2, 11].

Em geral, os sinais transmitidos por diversos usuários em um sistema de comunicação de acesso múltiplo são codificados utilizando-se codificações comuns, com um único nível de código, ou codificações com códigos concatenados. Um importante representante desses códigos concatenados é o código Turbo, que já foi descrito por [23] [24], [25], [26, 2].

#### 2.2.3.4 Detectores não lineares: Cancelamento Sucessivo de Interferência

Uma outra técnica de detecção multiusuário é chamada de cancelamento sucessivo de interferência (SIC, *successive interference cancellation*). Esta técnica baseia-se em remover os sinais interferentes do sinal recebido, um por vez, ao passo em que são detectados. Uma possível abordagem é a demodulação dos usuários na ordem decrescente das potências recebidas. Assim, o usuário cujo sinal recebido é o mais forte é demodulado primeiramente. Após um sinal ter sido demodulado e detectado, a informação detectada é usada para subtrair o sinal de um usuário em particular do conjunto dos sinais recebidos.

Ao tomar-se a decisão sobre a informação de um usuário  $k$  transmitida, assume-se que as decisões dos usuários  $k + 1, \dots, K$  são corretas dispensando a presença dos usuários  $1, \dots, k - 1$ . Portanto, a decisão para o bit de informação do  $k$ -ésimo usuário, para transmissão síncrona, é:

$$\hat{b}_k = \text{sgn} \left[ r_k - \sum_{j=k+1}^K \sqrt{(E_b)_j} \rho_{jk} \hat{b}_j \right] \quad (2.62)$$

onde  $r_k$  é a saída do correlator ou filtro casado correspondente à sequência de assinatura do  $k$ -ésimo usuário. O enfoque baseado na demodulação dos sinais na ordem decrescente de potência recebida não leva em consideração as correlações cruzadas entre os usuários. Uma abordagem alternativa é a demodulação dos sinais dos usuários de acordo com as potências nas saídas dos correladores cruzados

ou filtros casados, isto é, de acordo com a métrica de correlação:

$$\mathbb{E} \left\{ \left[ \int_0^T s_k(t) r(t) dt \right]^2 \right\} = (E_b)_k + \sum_{j \neq k} (E_b)_j \rho_{jk}^2(0) + \frac{N_0}{2} \quad (2.63)$$

que se aplica ao caso de transmissão síncrona. Algumas observações importantes sobre a SIC são: a SIC exige que se estime as potências dos sinais recebidos dos usuários para cancelar a interferência; erros de estimação resultam em interferência multiusuário residual, que causa uma degradação de desempenho; a interferência de usuários cujos sinais são mais fracos que o sinal do usuário desejado é tratada como interferência aditiva; a complexidade computacional na demodulação do bit de informação de um usuário é linear com o número de usuários; finalmente, o atraso na demodulação do usuário mais fraco aumenta linearmente com o número de usuários. O detector multiusuário SIC, dado na Equação 2.62, é também um detector subótimo, pois os sinais dos usuários mais fracos são tratados como interferência aditiva.

## 2.3 Inferência Bayesiana

Três abordagens da inferência estatística são as mais utilizadas atualmente: a frequentista, a bayesiana e a verossimilhançista.

A frequentista utiliza procedimentos baseados na amostragem repetida, a partir de um modelo particular que define a distribuição de probabilidade dos dados observados, condicionalmente aos parâmetros desconhecidos. Propriedades do procedimento são analisadas neste arcabouço de amostragens repetidas para valores fixos de parâmetros desconhecidos; bons procedimentos têm bom desempenho em uma ampla faixa de valores de parâmetros. A crítica que se faz à teoria frequentista é a existência de replicação de dados, bem como o uso de extrapolação assintótica.

Uma teoria que não faz uso de tais argumentos é a inferência bayesiana, cujos fundamentos foram estabelecidos por Thomas Bayes em 1763. A inferência bayesiana exige a adoção de um modelo de amostragem e, adicionalmente, o estabelecimento de uma distribuição estatística *a priori* para todas as quantidades desconhecidas no modelo (parâmetros e dados inexistentes). A distribuição *a priori* e a verossimilhança são usadas para computar a distribuição condicional dos parâmetros desconhecidos, dadas as informações observadas, (a distribuição "*a posteriori*"), a partir da qual as inferências estatísticas são formuladas. Permitindo-se que os dados observados tenham algum papel na determinação da distribuição *a priori*, produz-se o que se denomina de abordagem empírica de

Bayes. A abordagem bayesiana examina procedimentos para um experimento repetido de amostragem de itens desconhecidos, sorteados de uma distribuição posterior para um dado conjunto de dados. A abordagem empírica de Bayes pode também examinar procedimentos sob amostragens repetidas tanto de dados quanto de itens desconhecidos a partir de sua distribuição conjunta.

Finalmente, a abordagem “verossimilhancista” (*“likelihoodist”* ou “Fisheriana”) desenvolve um modelo de amostragem sem a necessidade de se ter uma distribuição “*a priori*”, como ocorrem nas outras duas abordagens. Inferências estão restritas a procedimentos que usam os dados somente, como relatado pela verossimilhança.

### 2.3.1 Procedimentos bayesianos

Nos procedimento bayesianos distingue-se como objetivo:

— Realizar inferências sobre o parâmetro não observável  $\theta$ . As inferências clássicas são baseadas em probabilidades associadas com as diferentes amostras,  $\mathbf{y}$ , que poderiam ocorrer para algum valor fixo, porém desconhecido, do parâmetro  $\theta$ . É o que ocorre quando se fazem inferências por amostragem de certas estatísticas com base em distribuições de probabilidades.

De fato, uma distribuição por amostragem pondera, com as respectivas probabilidades ou densidades de probabilidade, os valores que a variável ou a estatística pode assumir quando se percorre todo o espaço amostral.

As inferências bayesianas são baseadas em probabilidades subjetivas ou credibilidades *a posteriori* associadas com diferentes valores do parâmetro  $\theta$  e condicionadas ao valor particular observado de  $\mathbf{y}$ . O valor  $\mathbf{y}$  é fixo e corresponde à variação de  $\theta$  considerada.

Na abordagem bayesiana, especifica-se o modelo para os dados observados  $\mathbf{y} = (y_1, \dots, y_n)$ , dado um vetor de parâmetros desconhecidos  $\theta$ , usualmente na forma de uma distribuição de probabilidade  $f(\mathbf{y}|\theta)$ , e supõe-se também que  $\theta$  é uma quantidade aleatória, que tem uma distribuição “*a priori*”  $\pi(\theta|\eta)$ , onde  $\eta$  é um vetor de hiperparâmetros (os parâmetros da distribuição *a priori* são chamados de hiperparâmetros). A inferência sobre  $\theta$  é, então, baseada em sua distribuição *a posteriori*, dada por:

$$p(\theta|\mathbf{y}, \eta) = \frac{p(\mathbf{y}, \theta|\eta)}{p(\mathbf{y}|\eta)} = \frac{p(\mathbf{y}, \theta|\eta)}{\int p(\mathbf{y}, \mathbf{u}|\eta) d\mathbf{u}} = \frac{f(\mathbf{y}|\theta)\pi(\theta|\eta)}{\int f(\mathbf{y}|\mathbf{u})\pi(\mathbf{u}|\eta) d\mathbf{u}} \quad (2.64)$$

A Equação 2.64 representa o conhecido Teorema de Bayes. Note-se a contribuição tanto dos dados experimentais (na forma da verossimilhança  $f$ ) quanto

da visão prévia do experimentador (na forma da distribuição  $\pi$ , *a priori*) para a distribuição *a posteriori* na equação. O resultado da integral no denominador pode ser escrito como  $m(\mathbf{y}|\boldsymbol{\eta})$ , e é denominado de distribuição *marginal* dos dados  $\mathbf{y}$  dado o hiperparâmetro  $\boldsymbol{\eta}$ .

A Equação 2.64 pode ser expressa na forma abreviada (suprime-se a dependência da distribuição *a priori* sobre o valor de  $\boldsymbol{\eta}$  caso este seja uma constante):

$$p(\boldsymbol{\theta}|\mathbf{y}) \propto f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) \quad (2.65)$$

Fica claro por esta equação que a verossimilhança pode ser multiplicada por qualquer constante (ou qualquer função de  $\mathbf{y}$  isoladamente) sem alterar a distribuição *a posteriori*.

### 2.3.2 Distribuições *a priori*

A abordagem bayesiana, como indicado na subseção anterior, depende da adoção de distribuições de probabilidade não somente para as variáveis de dados, como  $\mathbf{y}$ , mas também para os parâmetros, como  $\boldsymbol{\theta}$ . Tal exigência pode ser ou não consistente com a noção comum de frequência, isto é de experimentos repetidos muitas vezes, ou seja de probabilidade. A análise bayesiana é baseada na crença em probabilidade subjetiva, na qual quantificam-se quaisquer sentimentos, mesmo que vagos, que se tenha sobre  $\theta$  antes de se observar os dados  $\mathbf{y}$  na distribuição  $\pi$ . Esta distribuição é então atualizada com o uso dos dados através do Teorema de Bayes da Equação 2.64; assim a distribuição *a posteriori* resultante contempla as informações dos dados e da distribuição *a priori*.

Uma entidade fundamental existente na associação de probabilidades é a *coerência*. Se um indivíduo julga que um evento  $A$  é mais provável que seu complementar em um dado universo, então ele naturalmente deverá, confiando na maior probabilidade de ocorrência de  $A$ , associar a ele uma probabilidade  $p(A)$  maior que  $1/2$ . Por exemplo, se ele julgar que uma proporção de três para um a favor de  $A$  é razoável, então ele deverá sugerir  $p(A) = 3/4$ . O Teorema de Bayes fornece uma maneira coerente de se considerar e atualizar opiniões.

Historicamente, uma das maiores barreiras à difusão do uso do paradigma bayesiano tem sido a determinação apropriada da distribuição *a priori*  $\pi$ , que é em geral uma tarefa muito difícil. Tipicamente, essas distribuições são especificadas com base em informações acumuladas de estudos passados ou a partir de opiniões de especialistas. De forma a dinamizar o processo de seleção da distribuição *a priori*, e também simplificar o trabalho computacional subsequente, os



pesquisadores frequentemente limitam a escolha de  $\pi$  a um grupo de distribuições familiares. Uma alternativa ainda mais simples, em alguns casos, é estabelecer-se a distribuição *a priori* com pouco conteúdo informativo, de forma que os dados do estudo atual serão dominantes na determinação da distribuição *a posteriori*.

### Distribuições *a priori* conjugadas

Ao escolher-se uma distribuição *a priori* pertencente a uma família específica de distribuições  $\pi(\boldsymbol{\theta}|\boldsymbol{\eta})$ , algumas seleções podem ser mais convenientes computacionalmente que outras. Em particular, pode ser possível selecionar-se uma representante da família que seja conjugada com a verossimilhança  $f(\mathbf{y}|\boldsymbol{\theta})$ , isto é, uma distribuição que leve à distribuição *a posteriori*  $p(\boldsymbol{\theta}|\mathbf{y})$ , pertencente à mesma família da distribuição *a priori*. A esta propriedade, da distribuição *a posteriori* seguir a mesma forma paramétrica da distribuição *a priori*, dá-se o nome de *conjugação*: a distribuição *a priori* Gamma,  $\Gamma$ , por exemplo, pertence a uma família conjugada à verossimilhança de Poisson [27, 28, 29].

### Distribuições *a priori* não informativas

Conforme mencionado, geralmente ou não existe informação *a priori* confiável sobre  $\boldsymbol{\theta}$ , ou simplesmente deseja-se realizar uma inferência baseada somente nos dados. À primeira vista pode parecer que a inferência bayesiana não é adequada em tais cenários, mas esta conclusão não é correta. Supondo que se possa encontrar uma distribuição  $\pi(\boldsymbol{\theta})$ , que não contenha informação alguma sobre  $\boldsymbol{\theta}$ , isto é, não favorece nenhum valor de  $\boldsymbol{\theta}$  sobre qualquer outro, deve-se referir a tal distribuição como distribuição *a priori* não informativa para  $\boldsymbol{\theta}$ . Neste caso, todas as informações resultantes da distribuição *a posteriori*  $p(\boldsymbol{\theta}|\mathbf{x})$  terão vindo dos dados, e portanto todas as inferências resultantes serão completamente *objetivas*, ao invés de *subjetivas* [30].

Neste trabalho foi adotada a abordagem bayesiana, como descrito em [9].

## 3 *Wavelets*

### Fundamentos Teóricos

## 3.1 Introdução

Em diversas ocasiões na Engenharia, na Matemática ou nas outras Ciências, profissionais são confrontados com uma versão tipicamente similar do seguinte problema. É dada uma função  $f(t)$ , definida para  $t \in \mathbb{R}$  que descreve um fenômeno da vida real e deseja-se transmitir, armazenar ou analisar esta função utilizando algum dispositivo finito. Uma ilustração deste problema poderia ser uma situação em que  $f$  representa um sinal de voz e deseja-se transmiti-lo pela linha telefônica ou armazená-lo em um CD (*compact disk*). Para tornar o desenvolvimento matematicamente simples assume-se que  $f \in L^2(\mathbb{R})$ , espaço dos sinais reais. A função completa  $f$  é dada pela totalidade de seus valores nos pontos de  $\mathbb{R}$ , isto é, o problema é de domínio contínuo. Supondo que exista uma base ortonormal  $(f_n)_{n \in \mathbb{N}}$  em  $L^2(\mathbb{R})$ , pode-se escrever

$$f = \sum_{n \in \mathbb{N}} a_n f_n \quad (3.1)$$

onde a série converge em  $L^2(\mathbb{R})$  e os coeficientes são univocamente determinados por  $a_n = \langle f, f_n \rangle$ , para  $n \in \mathbb{N}$ . Assim, ao invés de se transmitir a função  $f$  é suficiente transmitir a sequência de coeficientes  $(a_n)_{n \in \mathbb{N}}$  e deixar que o receptor realize a somatória da série. Para tornar o procedimento realmente finito deve-se escolher um conjunto finito  $A \subset \mathbb{N}$  tal que  $\sum_{n \in A} a_n f_n$  seja uma boa aproximação de  $\sum_{n \in \mathbb{N}} a_n f_n$ . Como os processos são estocásticos, não se pode esperar obter transmissão perfeita em geral. Na realidade o receptor está realizando a somatória  $\sum_{n \in A} \tilde{a}_n f_n$ , onde  $\tilde{a}_n$  é uma boa estimativa de  $a_n$  e tem que se esperar que o resultado ainda seja ‘quase igual’ a  $\sum_{n \in \mathbb{N}} a_n f_n$ . Em uma situação real deve-se decidir e basear esta decisão na experiência do pesquisador. Matematicamente falando, precisa-se de uma distância entre funções, que é frequentemente fornecida por uma norma.

Esta é uma situação bem geral e existem muitas formas de se lidar com as várias instâncias especiais dos diferentes aspectos deste tipo de problema. As *wavelets* surgiram como mais uma ferramenta, e é considerada muito eficiente para se lidar com tais problemas. [31, 32, 33, 34, 35, 36]

O termo *wavelet* teve sua origem na área de sismologia, quando foi criado por Ricker, em 1940, para descrever o distúrbio causado por um impulso sísmico ou uma carga explosiva [37, 38]. Em 1982, Morlet e colaboradores [39] mostraram como essas *wavelets* “sísmicas” podiam ser efetivamente modeladas com funções matemáticas. Assim, a utilização das *wavelets* evoluiu a partir da década de 80

do Século XX, influenciada por idéias de matemática pura (análise harmônica, análise funcional, teoria da aproximação, conjuntos fractais, etc.) e da matemática aplicada (processamento de sinal, física matemática, etc.). Quase que instantaneamente o tema tornou-se uma história de sucesso, com milhares de artigos publicados ao longo dos anos e uma ampla gama de aplicações [40, 41, 42].

Este capítulo é dividido em seções onde serão vistos os fundamentos e as técnicas que precederam a teoria das *wavelets*, e, logo após, será apresentado um resumo sobre as próprias *wavelets*. Para se compreender melhor os objetivos e o cenário onde se inserem as *wavelets*, será visto a seguir a teoria da transformada janelada de Fourier (*WFT*, *windowed Fourier transform* ou *short-time Fourier transform*).

## 3.2 Transformada janelada de Fourier

### 3.2.1 Definição da WFT e Definição

Suponha que se queira analisar o trecho de uma música por suas componentes de frequência. O intervalo musical, tal como percebido pelo ouvido, pode ser modelado com exatidão por uma função  $f(t)$  que representa a pressão do ar nos tímpanos como uma função do tempo. Se a ‘música’ consistisse de uma única nota, regular, com frequência angular fundamental  $\omega_1$  (em ciclos por unidade de tempo), então  $f(t)$  seria periódica com período  $T = (2\pi)/\omega_1$  e a descrição natural de seu conteúdo de frequências seria a Série de Fourier, uma vez que os coeficientes de Fourier  $c_n$  dão as amplitudes das várias frequências harmônicas  $\omega_n = n\omega_1$ , que ocorrem em  $f$  (ver [43]). Se a música é uma série de tais notas, ou uma melodia, então em geral ela não é periódica e não se pode utilizar a Série de Fourier diretamente. Uma abordagem neste caso seria computar a transformada integral de Fourier  $\hat{f}(\omega)$  de  $f(t)$ . No entanto, este método é falho sob o ponto de vista prático: para computar  $\hat{f}(\omega)$  deve-se integrar  $f(t)$  ao longo de todo o tempo, e assim  $\hat{f}(\omega)$  contém a amplitude total para a frequência  $\omega$  no trecho completo ao invés da distribuição de harmônicas em cada nota individual! Portanto, se o trecho continuasse por um certo período de tempo, seria necessário aguardar até que ele encerrasse antes de computar  $\hat{f}$ , e então o resultado seria completamente não informativo sob o ponto de vista musical. A mesma análise seria verdadeira, naturalmente, fosse o sinal  $f(t)$  um sinal de voz ou, no caso multidimensional, uma imagem ou um sinal de vídeo.

Uma outra abordagem seria recortar  $f$  aproximadamente em notas simples e analisar cada nota separadamente. Esta análise tem o problema óbvio de ser de

alguma forma arbitrária, pois é impossível dizer exatamente quando uma dada nota termina e a próxima inicia. Diferentes formas de recortar o sinal podem resultar em análises completamente diferentes. Além disso, este tipo de análise deve ser feito sob medida para o sinal particular que se está estudando (decidir como particionar o sinal em notas, por exemplo). Assim isso não é realmente ‘automático’. Para se criar uma abordagem mais natural, pode-se utilizar algum conhecimento da experiência da audição. Nossos ouvidos podem ouvir mudanças contínuas de tons, assim como as abruptas, e eles fazem isso sem uma partição arbitrária do sinal em ‘notas’. Pode-se construir um modelo muito simples para o ouvir, o qual embora fisiologicamente não seja preciso, servirá principalmente como um dispositivo de motivação para a definição da transformada de Fourier por janelas.

Pelo fato do ouvido analisar a distribuição de frequências de um dado sinal  $f$  em tempo real, ele deve dar informações sobre  $f$  simultaneamente nos domínios da frequência e do tempo. Assim, modela-se a saída do ouvido por uma função  $\tilde{f}(\omega, t)$  que depende tanto da frequência  $\omega$  quanto do tempo  $t$ . Para qualquer valor fixo de  $t$ ,  $\tilde{f}(\omega, t)$  representa a distribuição de frequência ‘percebida pelo ouvido’ no tempo  $t$ , e esta distribuição varia naturalmente com  $t$ . Como o ouvido não consegue analisar o que ainda não ocorreu, somente os valores  $f(u)$  para  $u \leq t$  podem ser usados na computação de  $\tilde{f}(\omega, t)$ . É também razoável assumir que o ouvido tem uma ‘memória’ finita. Isto significa que existe um intervalo de tempo  $T > 0$  tal que somente os valores  $f(u)$  para  $u \geq t - T$  podem influenciar a saída no tempo  $t$ . Consequentemente,  $\tilde{f}(\omega, t)$  depende somente de  $f(u)$  para  $t - T \leq u \leq t$ . Finalmente, espera-se que valores  $f(u)$  próximos aos extremos  $u \approx t - T$  e  $u \approx t$  tenham menos influência sobre  $\tilde{f}(\omega, t)$  do que valores do meio do intervalo. Estas declarações podem ser formuladas matematicamente como segue: seja  $g(u)$  uma função que não possui valores fora do intervalo  $-T \leq u \leq 0$ , isto é, tal que  $\text{supp } g \subset [-T, 0]$ .  $g(u)$  será a função peso, ou a *janela*, a qual será utilizada para ‘localizar’ sinais no tempo. Pode-se fazer  $g$  de valor complexo, apesar de que em muitas aplicações ela é real. Para todo  $t \in \mathbb{R}$ , define-se

$$f_t(u) \equiv \overline{g(u - t)} f(u), \quad (3.2)$$

where  $\overline{g(u - t)} \equiv \overline{g(u - t)}$ . Então  $\text{supp } f_t \subset [t - T, t]$ , e entende-se  $f_t$  como uma *versão localizada* de  $f$  que depende apenas dos valores de  $f(u)$  no intervalo  $t - T \leq u \leq t$ . Se  $g$  é contínua, então os valores  $f_t(u)$  com  $u \approx t - T$  e  $u \approx t$  são pequenos. Isto significa que a localização acima é suave ao invés de abrupta, uma qualidade que como será visto é importante. Define-se agora a transformada de

Fourier por janelas (WFT) de  $f$  como a transformada de Fourier de  $f_t$ :

$$\tilde{f}(\omega, t) \equiv \hat{f}_t(\omega) = \int_{-\infty}^{\infty} e^{-2\pi i \omega u} f_t(u) du = \int_{-\infty}^{\infty} e^{-2\pi i \omega u} \bar{g}(u-t) f(u) du. \quad (3.3)$$

Conforme visto anteriormente,  $\tilde{f}(\omega, t)$  depende de  $f(u)$  somente no intervalo  $t - T \leq u \leq t$  e (se  $g$  for contínua) dá pequeno peso aos valores de  $f$  próximos aos extremos.

Notas:

(a) A condição  $\text{supp } g \subset [-T, 0]$  foi imposta somente para dar uma motivação física para a WFT. Para que a WFT faça sentido basta somente assumir que  $g(u)$  seja quadraticamente integrável, isto é,  $g \in L^2(\mathbb{R})$ .

(b) No caso extremo quando  $g(u) \equiv 1$  (quando então  $g \notin L^2(\mathbb{R})$ ), a WFT se reduz à transformada de Fourier comum. Na sequência, simplesmente assume-se que  $g \in L^2(\mathbb{R})$ .

Se for definido

$$g_{\omega, t}(u) \equiv e^{2\pi i \omega u} g(u-t), \quad (3.4)$$

então  $\|g_{\omega, t}\| = \|g\|$ ; portanto  $g_{\omega, t}$  também pertence ao  $L^2(\mathbb{R})$ , e a WFT pode ser expressa como o produto interno de  $f$  com  $g_{\omega, t}$ ,

$$\tilde{f}(\omega, t) = \langle g_{\omega, t}, f \rangle \equiv g_{\omega, t}^* f, \quad (3.5)$$

que faz sentido se ambas as funções são do  $L^2(\mathbb{R})$  (o asterisco simboliza o conjugado do número complexo).

Exemplo 3.2: WFT de um sinal ‘*chirp*’. Um ‘*chirp*’ (no vocabulário das comunicações sem fio) é um sinal com amplitude estável mas com frequência crescente:

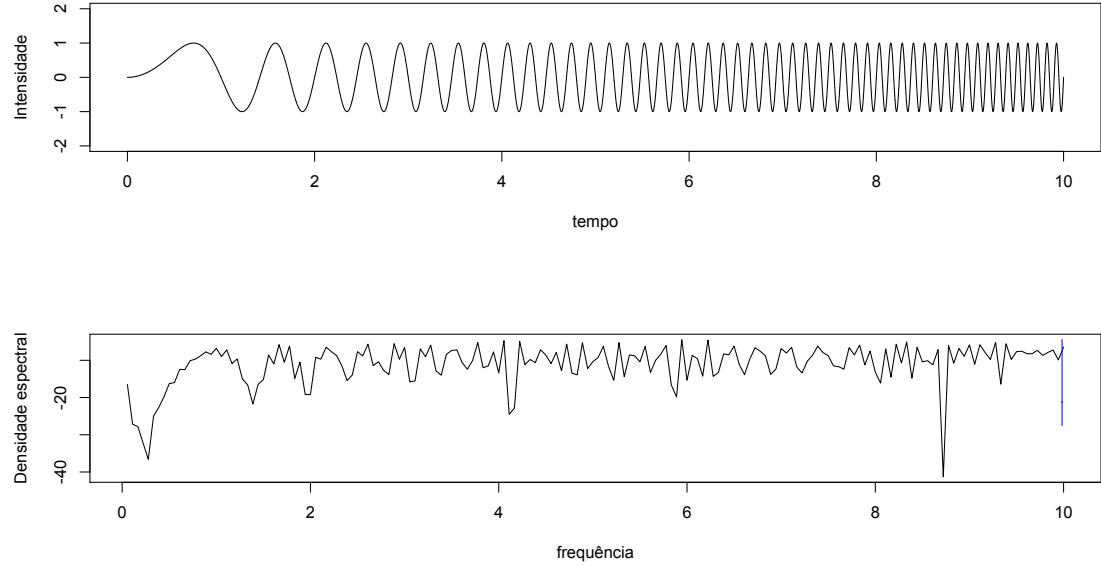
$$f(u) = \sin(\pi u^2). \quad (3.6)$$

De fato, a frequência instantânea  $\omega_{inst}(u)$  de  $f$  pode ser definida como a derivada de sua fase:

$$2\pi\omega_{inst}(u) \equiv \partial_u(\pi u^2) = 2\pi u. \quad (3.7)$$

A análise de Fourier usual esconde o fato que um ‘*chirp*’ tem uma frequência instantânea bem definida pela integração sobre todo o tempo (ou praticamente

sobre um longo período), assim chegando a um espectro de frequência muito extenso. A Figura 3.1 mostra  $f(u)$  para  $0 \leq u \leq 10$  e sua densidade espectral de energia  $|\hat{f}(\omega)|^2$  naquela faixa. O espectro está realmente muito espalhado..

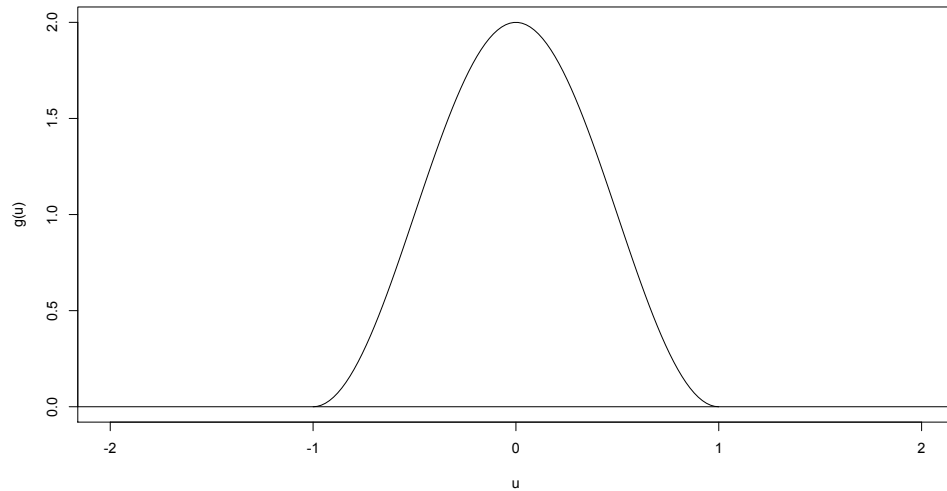


**Figura 3.1:** Parte superior: o sinal ‘chirp’  $f(u) = \sin(\pi u^2)$ ; parte inferior: a densidade espectral de energia  $|\hat{f}(\omega)|^2$  de  $f$ .

Agora, será feita uma análise de  $f$  usando a função janela

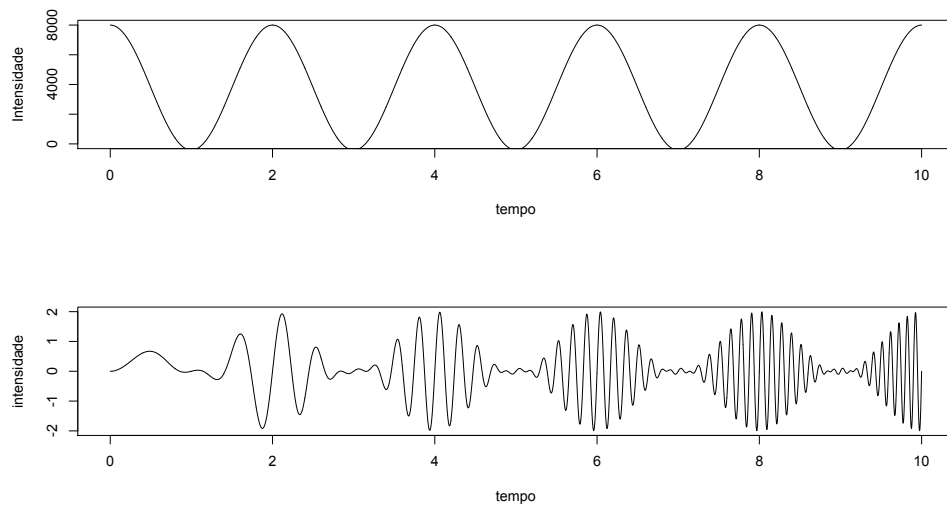
$$g(u) \equiv \begin{cases} 1 + \cos(\pi u) & -1 \leq u \leq 1 \\ 0 & \text{qualquer outro caso,} \end{cases} \quad (3.8)$$

que é apresentada na Figura 3.2.



**Figura 3.2:** A função janela  $g(u)$ .

A Figura 3.3 apresenta a função janela deslocada para valores consecutivos de tempo, apresentados todos juntos.

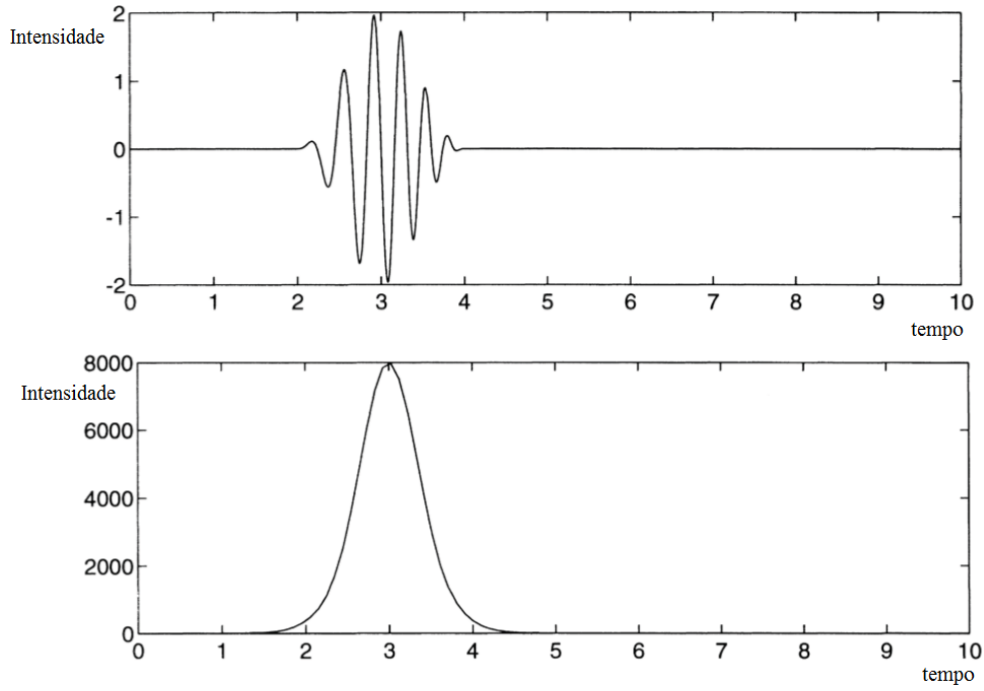


**Figura 3.3:** A função janela  $g(u)$  com intervalo reproduzido ao longo do eixo do tempo.

A Figura 3.4 mostra a versão localizada  $f_3(u)$  de  $f(u)$  e sua densidade de energia  $|\hat{f}_3(\omega)|^2$ . Como esperado, o sinal localizado tem uma frequência bem definida, (embora não exata)  $\omega_{inst}(3) = 3$ . Ela é portanto razoavelmente bem localizada tanto no tempo como no domínio da frequência. A Figura 3.5 repete esta análise para  $t = 7$ .

Agora a energia é localizada próximo a  $\omega_{inst}(7) = 7$ . As Figuras 3.6 e 3.7





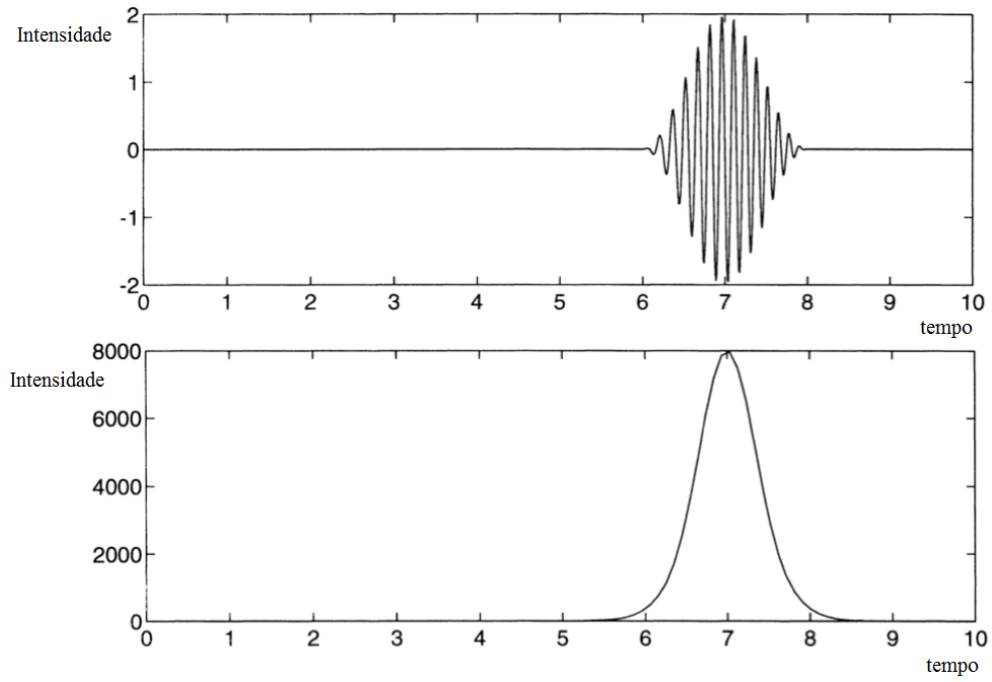
**Figura 3.4:** A função janela  $g(u)$ .

ilustram a resolução de frequência. Na Figura 3.6 a curva superior representa a função

$$h(u) = \text{Re}[g_{2,4}(u) + g_{4,6}(u)] = g(u-4)\cos(4\pi u) + g(u-6)\cos(8\pi u), \quad (3.9)$$

que corresponde à parte real da soma de duas ‘notas’, uma delas centrada em  $t = 4$  com  $\omega = 2$  e outra centrada em  $t = 6$  com  $\omega = 4$ . A parte de baixo da figura mostra a densidade de energia espectral de  $h$ . Os dois picos são essencialmente cópias de  $|\hat{g}(\omega)|^2$  (que está centrada em  $\omega = 0$ ) transladadas para  $\omega = 2$  e  $\omega = 4$ , respectivamente. Isto é repetido na Figura 3.7 para  $h = \text{Re}[g_{2,4} + g_{3,6}]$ . Estas duas figuras mostram que a janela pode resolver intervalos de frequências de até  $\Delta\omega = 2$  mas não intervalos da ordem de  $\Delta\omega = 1$ .

Será visto que os vetores  $g_{\omega,t}$ , parametrizados por todas as frequências  $\omega$  e tempos  $t$ , formam algo análogo a uma base do  $L^2(\mathbb{R})$ . Note que o produto interno  $\langle g_{\omega,t}, f \rangle$  é bem definido para toda escolha de  $\omega$  e  $t$ ; assim os valores  $\tilde{f}(\omega, t)$  de  $\tilde{f}$  são bem definidos. Por contraste, lembrando que o valor de  $f(u)$  de uma ‘função’  $f \in L^2(\mathbb{R})$  em qualquer ponto individual em geral não é bem definido, desde que  $f$  pode ser modificada em um conjunto de medida zero (tal como o conjunto  $u$  de um ponto) sem ser modificado como um elemento de  $L^2(\mathbb{R})$  [44]. O mesmo pode ser dito sobre a transformada de Fourier  $\hat{f}$  como uma função de  $\omega$ . Esta observação já indica que as WFT tais como a  $\tilde{f}(\omega, t)$  comportam-se melhor do



**Figura 3.5:** Gráfico superior: a versão localizada  $f_7(u)$  do sinal *chirp* usando a janela  $g(u)$ . Gráfico inferior: a densidade de energia de  $f_7$  mostrando boa localização em volta da frequência instantânea  $\omega_{inst}(7) = 7$ .

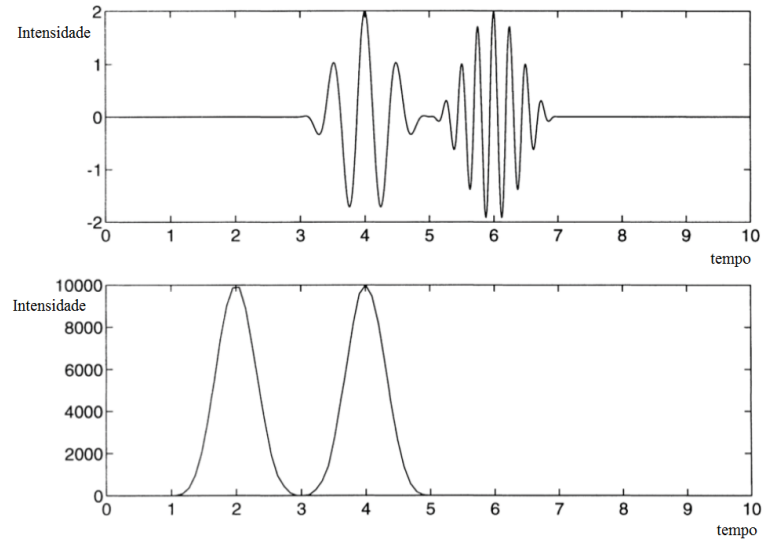
que os sinais correspondentes  $f(t)$ , no domínio do tempo ou suas transformadas de Fourier  $\hat{f}(\omega)$ , no domínio da frequência. Um outro exemplo deste tipo pode ser obtido aplicando-se a desigualdade de Schwarz a (3.5), que dá

$$|\tilde{f}(\omega, t)| = |\langle g_{\omega, t}, f \rangle| \leq \|g_{\omega, t}\| \|f\| = \|g\| \|f\|, \quad (3.10)$$

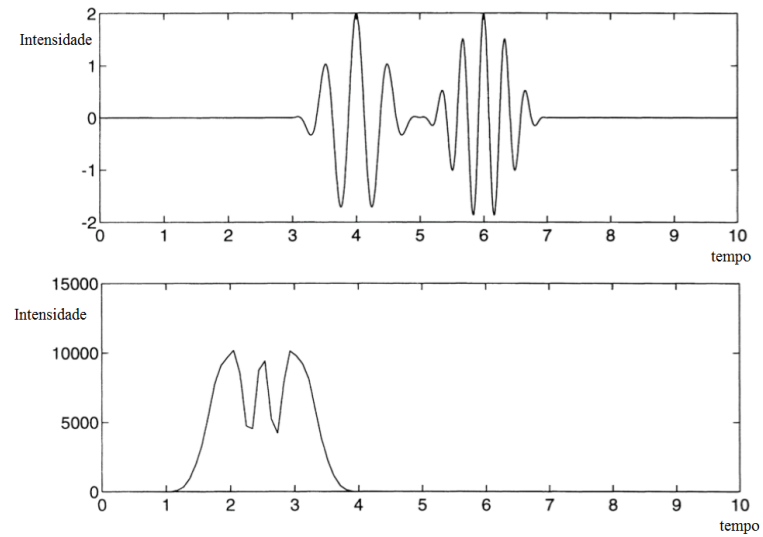
mostrando que  $\tilde{f}(\omega, t)$  é uma função limitada, pois o termo do lado direito é finito e independente de  $\omega$  e de  $t$ . Assim, uma condição necessária para que uma dada função  $h(\omega, t)$  seja a WFT de algum sinal  $f \in L^2(\mathbb{R})$  é que  $h$  seja limitada. No entanto, esta condição mostra-se distante ainda da condição *suficiente*. Isto é, nem toda função limitada  $h(\omega, t)$  é a WFT  $\tilde{f}(\omega, s)$  de algum sinal  $f \in L^2(\mathbb{R})$ .

### 3.2.2 Localização no tempo e na frequência

Um aspecto marcante da transformada de Fourier comum é a dualidade que ela encerra entre o domínio do tempo e o domínio da frequência, isto é, o fato que as relações para  $f \mapsto \hat{f}$  e  $h \mapsto \check{h}$  são idênticas exceto pelo sinal do expoente. Pode parecer que esta simetria é perdida quando se lida com a WFT, porque a frequência e o tempo foram tratados de forma bem diferente na definição de  $\tilde{f}$ . Na verdade, a WFT é também simétrica também com respeito aos dois domínios,



**Figura 3.6:** Curva superior: gráfico de  $h = \text{Re}[g_{2,4} + g_{4,6}]$  para a janela em  $g(u)$ . Curva inferior: Densidade espectral de energia de  $h$  mostrando problema de resolução de frequência para  $\Delta\omega = 2$ .



**Figura 3.7:** Curva superior: gráfico de  $h = \text{Re}[g_{2,4} + g_{3,6}]$  para a janela em  $g(u)$ . Curva inferior: Densidade espectral de energia de  $h$  mostrando problema de resolução de frequência para  $\Delta\omega = 1$ .

como será visto agora. Pela identidade de Parseval:

$$\tilde{f}(\omega, t) = \langle g_{\omega, t}, f \rangle = \langle g_{\omega, t}, \hat{f} \rangle, \quad (3.11)$$

e o lado direito é computado como

$$\tilde{f}(\omega, t) = e^{-2\pi i \omega t} \int_{-\infty}^{\infty} e^{2\pi i \nu t} \bar{\hat{g}}(\nu - \omega) \hat{f}(\nu) d\nu \quad (3.12)$$

onde  $\bar{\hat{g}}(\nu - \omega) \equiv \overline{\hat{g}(\nu - \omega)}$ . A Equação 3.12 é muito parecida com a Equação 3.3 mas com a variável tempo  $u$  trocada pela variável frequência  $\nu$  e a janela de tempo  $g(u - t)$  trocada pela janela de frequência  $\hat{g}(\nu - \omega)$ . (O fator extra  $e^{-2\pi i \omega t}$  na Equação 3.12 está relacionado às ‘relações de comutação de Weyl’ do grupo *Weyl-Heisenberg*, que governa translações no tempo e em frequência.) Assim, do ponto de vista do domínio da frequência, inicia-se com o sinal  $\hat{f}(\nu)$  e ‘localiza-se’ a ele próximo à frequência  $\omega$  utilizando-se a função janela  $\hat{g} : \hat{f}_{\omega}(\nu) \equiv \bar{\hat{g}}(\nu - \omega) \hat{f}(\nu)$ ; toma-se então a transformada de Fourier inversa de  $\hat{f}_{\omega}$  e faz-se a multiplicação dela pelo fator extra  $e^{-2\pi i \omega t}$ , que é uma modulação devido à translação de  $\hat{g}$  por  $\omega$  no domínio da frequência. Se a janela  $g$  é razoavelmente bem localizada em frequência assim como no tempo, isto é, se  $\hat{g}(\nu)$  é pequena fora de uma banda estreita, além de  $g(t)$  ser pequena fora de um intervalo de tempo curto, então a Equação 3.12 mostra que a WFT dá um análise de tempo-frequência local do sinal  $f$  no sentido de que ela provê informação precisa sobre  $f$  simultaneamente no domínio do tempo e no domínio da frequência. No entanto, todas as funções, as *janelas* inclusive, obedecem ao *princípio da incerteza*, que diz que ‘localizações precisas no tempo e em frequência são eventos mutuamente exclusivos’. Informalmente isto significa que se uma função não nula  $g(t)$  é pequena fora de um intervalo de tempo  $T$  e sua transformada de Fourier é pequena fora de uma banda de frequência de largura  $\Omega$ , então uma desigualdade do tipo  $\Omega T \geq c$  deve existir para alguma constante positiva  $c \approx 1$ . O valor exato de  $c$  depende de como as larguras de  $T$  e  $\Omega$  do sinal no tempo e em frequência são medidos.

### 3.2.3 A Fórmula de Reconstrução

A WFT é uma opção, de tempo real, à transformada de Fourier, que dá a distribuição de frequência dinâmica (variável no tempo) de  $f(t)$ . O próximo passo, natural, é encontrar uma substituta para a transformada inversa de Fourier, isto é, para reconstruir  $f$  a partir de  $\tilde{f}$ . Para isso, notando que  $\tilde{f}(\omega, t) = \hat{f}_t(\omega)$ , então pode-se aplicar a transformada inversa de Fourier com relação à variável  $\omega$  para se obter

$$\bar{g}(u-t)f(u) \equiv f_t(u) = \int_{-\infty}^{\infty} e^{2\pi i \omega u} \tilde{f}(\omega, t) d\omega. \quad (3.13)$$

Não se pode recuperar  $f(u)$  dividindo-se por  $\bar{g}(u-t)$ , pois esta função pode anular-se. Ao invés disso, pode-se multiplicar a Equação 3.13 por  $g(u-t)$  e integrar em  $t$ :

$$\int_{-\infty}^{\infty} |g(u-t)|^2 f(u) dt = \int_{-\infty}^{\infty} dt \int_{-\infty}^{\infty} e^{2i\omega u} g(u-t) \tilde{f}(\omega, t). \quad (3.14)$$

Entretanto o lado esquerdo é simplesmente  $\|g\|^2 f(u)$ , assim

$$f(u) = C^{-1} \int_{-\infty}^{\infty} dt \int_{-\infty}^{\infty} e^{2i\omega u} g(u-t) \tilde{f}(\omega, t), \quad (3.15)$$

onde  $C \equiv \|g\|^2$ . Isto faz sentido se  $g$  for um vetor qualquer não nulo no  $L^2(\mathbb{R})$ , pois  $0 < \|g\| < \infty$ . Pela definição de  $g_{\omega,t}$ , a Equação 3.15 pode ser escrita

$$f(u) = C^{-1} \int g_{\omega,t}(u) \tilde{f}(\omega, t) dt d\omega, \quad (3.16)$$

que é a fórmula de reconstrução desejada.

Pode-se, agora, combinar a WFT com sua inversa para se obter a análoga da resolução da unidade.

### 3.2.4 Processamento de sinais no domínio tempo-frequência

A WFT converte a função  $f(u)$  de uma variável para uma função  $\tilde{f}(\omega, t)$  de duas variáveis sem alterar sua energia total.

## 3.3 Transformadas *wavelets* contínuas

[33, 45, 35]

### 3.3.1 Motivação e definição da transformada *wavelet*

A WFT foi concebida como um modelo, apesar de não preciso, para o processo de audição. O ouvido tem realmente um ‘tempo de resposta’ característico, o qual está relacionado, dentre outras coisas, ao seu tamanho físico. Sendo assim,

não é necessariamente ruim introduzir tal intervalo de tempo  $T$  na análise. No entanto, é legítimo imaginar se alguma forma pode ser encontrada para processar sinais localmente sem prejuízo da escala. A análise por *wavelets* é um método *independente de escala*. Começa-se com uma função janela de valor complexo  $\psi(t)$ , chamada de *wavelet* mãe ou *wavelet* básica. Como antes,  $\psi$  apresenta uma escala, sua largura, à análise. Como se deseja evitar compromisso com qualquer escala em particular, utiliza-se não somente  $\psi$  mas *todas as escalas possíveis de*  $\psi$ . Isso pode ser feito da seguinte forma: fixa-se um  $p \geq 0$  arbitrário, e para qualquer número real  $s \neq 0$ , define-se

$$\psi_s(u) \equiv |s|^{-p} \psi\left(\frac{u}{s}\right). \quad (3.17)$$

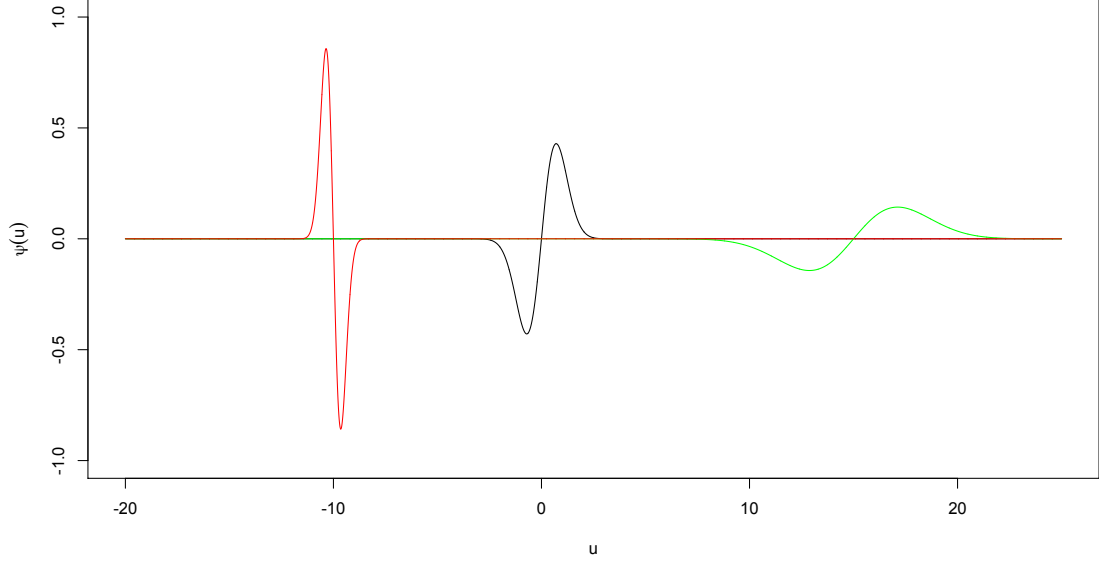
Para visualizar  $\psi_s$ , imagine que se tenha um gráfico de  $\psi(u)$ , com  $u$  medido em minutos e então se decida redesenhar o gráfico de  $\psi$  mas com  $u$  dado em segundos. Claramente, o novo gráfico se parecerá muito com o anterior mas será *expandido* por um fator de 60. De fato, não se trata de nada mais do que do gráfico de  $\psi(u/60)$ . Para  $s > 1$  qualquer,  $\psi_s(u)$  é uma versão de  $\psi$  *esticada por um fator  $s$*  na coordenada horizontal. Similarmente, se  $0 < s < 1$ , então  $\psi_s$  é uma versão de  $\psi$  *comprimida* na direção horizontal. Se  $s = -1$ , então  $\psi_s$  é uma versão refletida de  $\psi$ . Se  $-1 < s < 0$ , então  $\psi_s$  é uma versão refletida e comprimida de  $\psi$ . Finalmente, se  $s < -1$  então  $\psi_s$  é uma versão esticada e refletida de  $\psi$ . ‘ $s$ ’ é denominado de *fator de escala*.

O fator  $|s|^{-p}$  na Eq. (3.17) tem um efeito similar na direção vertical, isto é, na variável dependente. Se  $p > 0$ , então  $\psi$  é comprimida na vertical sempre que for esticada na horizontal e é esticada na vertical sempre que for comprimida na horizontal. (Se  $p < 0$ , então  $\psi$  é simultaneamente esticada ou comprimida em ambas as direções, mas somente convenções com  $p \geq 0$  são utilizadas.) Por exemplo, se  $p = 1$ , então a integral  $\int |\psi(u)| du$  não é afetada pela operação de escala. Se  $p = 1/2$ , então a integral de  $|\psi(u)|$  é afetada mas a norma  $L^2$  não é, isto é,  $\|\psi_s\| = \|\psi\|$ . Na verdade, como pode ser verificado na literatura, o valor de escolha de  $p$  é totalmente irrelevante para a teoria básica. Escolhas diferentes são feitas por diferentes autores. Por exemplo, Chui [46], Daubechies [47] e Meyer [48] utilizam  $p = 1/2$ ; já Mallat e Zhong [49] fazem uso de  $p = 1$ . Ao se trabalhar, por exemplo, com bases ortonormais de *wavelets*, a escolha  $p = 0$  é em alguns casos conveniente.

Obtém-se a localização de sinais no tempo ‘olhando-se’ para eles através de versões transladadas de  $\psi_s(u)$ . Se  $\psi(u)$  tem suporte em um intervalo de comprimento  $T$  próximo a  $u = 0$ , então  $\psi_s(u)$  tem suporte em um intervalo de

comprimento  $|s|T$  próximo a  $u = 0$  e a função

$$\psi_{s,t}(u) \equiv \psi_s(u - t) = |s|^{-p} \psi\left(\frac{u - t}{s}\right) \quad (3.18)$$



**Figura 3.8:** Gráficos de  $\psi(u) = ue^{-u^2}$  (centro) e suas versões transladadas e dilatadas  $\psi_{3,15}(u)$  (direita) e  $\psi_{-\frac{1}{2}, -10}(u)$  (esquerda).

tem suporte em um intervalo de comprimento  $|s|T$  próximo a  $u = t$ . Na Figura 3.8, estão ilustradas a *wavelet*  $\psi(u) = ue^{-u^2}$  e suas versões dilatada e transladada. Assume-se que  $\psi$  pertence ao  $L^2(\mathbb{R})$ . Então também  $\psi_{s,t}$ , pois

$$\|\psi_{s,t}\|^2 = |s|^{-2p} \int_{-\infty}^{\infty} \left| \psi\left(\frac{u-t}{s}\right) \right|^2 du = |s|^{1-2p} \|\psi\|^2. \quad (3.19)$$

As funções  $\psi_{s,t}$  são as *wavelets* geradas por  $\psi$  e desempenharão um papel similar a aquele executado anteriormente pelas ‘notas’  $g_{\omega,t}(u)$ . A transformada *wavelet* contínua (CWT, *Continuous Wavelet Transform*) de um sinal  $f(t)$  é definida como

$$\tilde{f}(s, t) \equiv \int_{-\infty}^{\infty} \overline{\psi_{s,t}}(u) f(u) du = \langle \psi_{s,t}, f \rangle = \psi_{s,t}^* f. \quad (3.20)$$

O produto interno no lado direito existe desde que ambas as funções  $f$  e  $\psi_{s,t}$  pertençam a  $L^2(\mathbb{R})$ .

Pode-se dar à transformada *wavelet*  $\tilde{f}(s, t)$  uma interpretação mais ou menos direta, dependendo da *wavelet* utilizada: como uma função de  $t$  para um valor de  $s$  fixo, ela representa o *detalhe* contido no sinal  $f(t)$ , na escala  $s$ .

O uso de versões transladadas e dilatadas de uma função única foi proposto por Morlet em 1983 [50] para análise de dados sísmicos.

### 3.3.2 Expressão de Reconstrução

O próximo passo é reconstruir  $f(u)$  a partir de sua CWT  $\tilde{f}(s, t)$ . O caminho é aplicar a identidade de Parseval à Equação 3.20,

$$\tilde{f}(s, t) = \langle \psi_{s,t}, f \rangle = \langle \hat{\psi}_{s,t}, \hat{f} \rangle, \quad (3.21)$$

e calcular

$$\hat{\psi}_{s,t}(\omega) = |s|^{1-p} e^{-2\pi i \omega t} \hat{\psi}(s\omega). \quad (3.22)$$

Assim, a Equação 3.21 torna-se

$$\tilde{f}(s, t) = |s|^{1-p} \int_{-\infty}^{\infty} e^{2\pi i \omega t} \overline{\hat{\psi}(s\omega)} \hat{f}(\omega) d\omega = |s|^{1-p} \left( \overline{\hat{\psi}(s\omega)} \hat{f}(\omega) \right)^{\vee}(t). \quad (3.23)$$

Isto é, como uma função de  $t$ ,  $\tilde{f}(s, t)$  é a transformada inversa de Fourier de  $|s|^{1-p} \overline{\hat{\psi}(s\omega)} \hat{f}(\omega)$  com relação à variável  $\omega$ , com  $s$  fazendo o papel de um parâmetro. Portanto, aplicando a transformada de Fourier com relação a  $t$  a ambos os lados resulta em

$$\int_{-\infty}^{\infty} e^{-2\pi i \omega t} \tilde{f}(s, t) dt = |s|^{1-p} \overline{\hat{\psi}(s\omega)} \hat{f}(\omega). \quad (3.24)$$

Se for possível recuperar  $\hat{f}(\omega)$ , então  $f(t)$  pode ser encontrada. Não se pode simplesmente dividir a Equação 3.24 por  $\overline{\hat{\psi}(s\omega)}$ , pois essa função pode desaparecer por cancelamento. Ao invés disso, multiplicam-se ambos os lados da equação por  $\hat{\psi}(s\omega)$  e integra-se com relação parâmetro de escala  $s$ . Porém, será necessário utilizar uma *função peso*  $w(s)$ , que é atualmente desconhecida, mas que será determinada por reconstrução. A Equação 3.24 implica em

$$\int_0^{\infty} w(s) ds \int_{-\infty}^{\infty} e^{-2\pi i \omega t} \hat{\psi}(s\omega) \tilde{f}(s, t) dt = \int_0^{\infty} w(s) s^{1-p} |\hat{\psi}(s\omega)|^2 \hat{f}(\omega) ds. \quad (3.25)$$

Defina

$$Y(\omega) \equiv \int_0^{\infty} w(s) s^{1-p} |\hat{\psi}(s\omega)|^2 ds \quad (3.26)$$

e assuma, no momento, que tanto  $Y(\omega)$  quanto sua recíproca são limitadas (exceto



possivelmente em um conjunto de medida zero), isto é, que

$$A \leq Y(\omega) \leq B \text{ quase certamente} \quad (3.27)$$

para constantes  $A$  e  $B$ , com  $0 < A \leq B < \infty$ . Então, pela Equação 3.25,

$$\begin{aligned} \hat{f}(\omega) &= Y(\omega)^{-1} \int_0^\infty w(s) ds \int_{-\infty}^\infty e^{-2\pi i \omega t} \hat{\psi}(s\omega) \tilde{f}(s, t) \\ &= Y(\omega)^{-1} \int_0^\infty w(s) s^{p-1} ds \int_{-\infty}^\infty \hat{\psi}_{s,t} \tilde{f}(s, t) dt \\ &= \int_0^\infty w(s) s^{p-1} ds \int_{-\infty}^\infty \hat{\psi}^{s,t}(\omega) \tilde{f}(s, t) dt, \end{aligned} \quad (3.28)$$

onde foi definido um novo conjunto de ‘wavelets’  $\psi^{s,t}(u)$  através de suas transformadas de Fourier  $\hat{\psi}^{s,t}(\omega)$  por

$$\hat{\psi}^{s,t}(\omega) \equiv Y(\omega)^{-1} \hat{\psi}_{s,t}(\omega) = s^{1-p} Y(\omega)^{-1} e^{-2\pi i \omega s} \hat{\psi}(s\omega). \quad (3.29)$$

Note que

$$|\hat{\psi}^{s,t}(\omega)| \leq A^{-1} |\hat{\psi}_{s,t}(\omega)| \quad (3.30)$$

por (3.27), e consequentemente por (3.19),  $\hat{\psi}^{s,t} \in L^2(\mathbb{R})$ , com

$$\|\psi^{s,t}\|^2 \leq A^{-2} \|\psi_{s,t}\|^2 = |s|^{1-2p} A^{-2} \|\psi\|^2. \quad (3.31)$$

No domínio do tempo as novas wavelets têm a forma

$$\psi^{s,t}(u) = s^{1-p} \int_{-\infty}^\infty e^{2\pi i \omega(u-t)} Y(\omega)^{-1} \hat{\psi}(s\omega) \equiv \psi^s(u-t), \quad (3.32)$$

onde

$$\psi^s(u) \equiv s^{1-p} \int_{-\infty}^\infty e^{2\pi i \omega u} Y(\omega)^{-1} \hat{\psi}(s\omega) = s^{-p} \int_{-\infty}^\infty e^{2\pi i \omega u/s} Y(\omega/s)^{-1} \hat{\psi}(\omega). \quad (3.33)$$

O sinal original pode ser recuperado agora tomando-se a transformada inversa de Fourier da Equação 3.28:

$$f(u) = \int_0^\infty w(s) s^{p-1} \int_{-\infty}^\infty \psi^{s,t}(u) \tilde{f}(s, t) \quad (3.34)$$

Em notação vetorial (isto é, eliminando-se  $u$  e olhando-se  $f$  e  $\psi^{s,t}$  como vetores

no  $L^2(\mathbb{R})$ ,

$$f = \int_0^\infty w(s) s^{p-1} ds \int_{-\infty}^\infty \psi^{s,t} \tilde{f}(s,t) dt. \quad (3.35)$$

Esta equação apresenta  $f$  como uma superposição de novos vetores  $\psi^{s,t}$ , com  $\tilde{f}(s,t)$  como a função de coeficiente.

A expressão de reconstrução, Equação 3.35, não é satisfatória como se apresenta por exigir que se calcule uma nova família de vetores completa  $\psi^{s,t}$ . Isto não seria muito problemático se a família inteira pudesse ser obtida a partir de uma única ‘*wavelet mãe*’ por translações e dilatações, assim como no caso das *wavelets* originais  $\psi_{s,t}$ . Pois então seria necessário somente calcular a nova *wavelet* mãe e usá-la para gerar a nova família. A Equação 3.32 mostra que as translações não são um problema, então deve-se concentrar nas dilatações. Pela Equação 3.33, uma condição suficiente para que a nova família tenha uma mãe é que

$$Y(\omega/s) = Y(\omega) \text{ para (quase) todas as } \omega \in (R), s > 0, \quad (3.36)$$

desde que  $\psi^s(u) = \psi^1(u/s)$  e, por (3.33),

$$\psi^{s,t}(u) = s^{-p} \psi^1\left(\frac{u-t}{s}\right), \quad (3.37)$$

onde  $\psi^1$  é  $\psi^s$  com  $s = 1$ . Na sequência,  $\psi^1$  gera a nova família da mesma forma que  $\psi$  gerou a anterior. A Equação 3.36 significa que  $Y(\omega)$  deve ser uma constante para todo  $\omega > 0$ , e uma (possivelmente diferente) constante para todo  $\omega < 0$ . (Fixe  $\omega \neq 0$ , e faça  $s$  variar de 0 a  $\infty$ .) Pode-se escolher agora, de forma livre, a função peso  $w(s)$  para fazer com que a Equação 3.36 seja verificada. Por que  $\omega$  entra na definição de  $Y(\omega)$  somente via o produto  $s\omega$ , exige-se que  $w(s)$  seja tal que

$$ds w(s) s^{1-p} = \frac{ds}{s} \quad (3.38)$$

Isto funciona porque, para qualquer  $\omega > 0$ , a mudança de variáveis  $s\omega = \xi$  gera

$$Y(\omega) = \int_0^\infty \frac{ds}{s} |\hat{\psi}(s\omega)|^2 = \int_0^\infty \frac{d\xi}{\xi} |\hat{\psi}(\xi)|^2 \equiv C_+, \quad (3.39)$$

e para qualquer  $\omega < 0$ , a mudança de variáveis  $s\omega = -\xi$  resulta

$$Y(\omega) = \int_0^\infty \frac{ds}{s} |\hat{\psi}(s\omega)|^2 = \int_0^\infty \frac{d\xi}{\xi} |\hat{\psi}(-\xi)|^2 \equiv C_-, \quad (3.40)$$

A Equação 3.27 portanto é verificada para  $\omega \neq 0$  se e somente se

$$0 < C_{\pm} = \int_0^{\infty} \frac{d\xi}{\xi} |\hat{\psi}(\pm\xi)|^2 < \infty. \quad (3.41)$$

Esta é denominada de *condição de admissibilidade* para a *wavelet* mãe  $\psi$ . Como já notado, a escolha de  $w(s)$  ditada pela Equação 3.38, isto é,  $w(s) = s^{p-2}$ , não somente garante a existência de uma *wavelet* mãe para a nova família mas também faz  $Y(\omega)$  constante por partes. Isso simplifica o cálculo da nova *wavelet* mãe, pois agora a Equação 3.33, com  $s = 1$ , torna-se

$$\psi^1(u) = C_-^{-1} \int_{-\infty}^0 e^{2\pi i \omega u} \hat{\psi}(\omega) + C_+^{-1} \int_0^{\infty} e^{2\pi i \omega u} \hat{\psi}(\omega) \equiv C_-^{-1} \psi_-(u) + C_+^{-1} \psi_+(u), \quad (3.42)$$

onde  $\psi_{\pm}$  são as componentes de frequências negativa e positiva de  $\psi$ , obtida tomando-se a transformada inversa de Fourier de  $\hat{\psi}(\omega)$  sobre apenas as frequências positivas e negativas, respectivamente. Tais funções são chamadas de *sinais analíticos*. A função  $\psi^1$  sera chamada de *wavelet* mãe *recíproca*, e  $\psi^{s,t}$ , de família *wavelet recíproca* de  $\psi_{s,t}$ . É fácil mostrar que  $\psi^1$  é também admissível no sentido de (3.41), e que a *wavelet* mãe recíproca de  $\psi^1$  é  $\psi$  [33].

Sumarizando o que foi visto até o momento, são apresentados os seguintes Teoremas.

**Teorema 3.3.1.** Seja  $\psi$  admissível no sentido de (3.41), e seja  $\{\psi^{s,t}\}$  a família *wavelet* recíproca de  $\psi_{s,t}$ . Então qualquer sinal  $f \in L^2(\mathbb{R})$  pode ser reconstruído a partir de sua transformada *wavelet* contínua  $\tilde{f}(s, t) = \psi_{s,t}^* f$  por

$$f(u) = \int_0^{\infty} s^{2p-3} ds \int_{-\infty}^{\infty} \psi^{s,t}(u) \tilde{f}(s, t) \quad (3.43)$$

isto resulta na resolução de unidade no  $L^2(\mathbb{R})$ :

$$\int_0^{\infty} s^{2p-3} ds \int_{-\infty}^{\infty} \psi^{s,t} \psi_{s,t}^* dt = I. \quad (3.44)$$

**Teorema 3.3.2.** Seja  $\psi$  admissível no sentido de que  $0 < C < \infty$ , onde

$$C \equiv \int_{-\infty}^{\infty} \frac{d\xi}{|\xi|} |\hat{\psi}(\xi)|^2 = C_- + C_+. \quad (3.45)$$

Then  $f \in L^2(\mathbb{R})$  pode ser recuperada de  $\tilde{f}$  por

$$f(u) = C^{-1} \int \int_{\mathbb{R}^2} |s|^{2p-3} \psi_{s,t}(u) \tilde{f}(s, t) ds dt, \quad (3.46)$$

e tem-se a correspondente resolução de unidade no  $L^2(\mathbb{R})$ :

$$C^{-1} \int \int_{\mathbb{R}^2} |s|^{2p-3} \psi_{s,t} \psi_{s,t}^* ds dt = I \quad (3.47)$$

A Equação (3.46) é mais geral que (3.43).

Uma consequência imediata de (3.47) é

$$\|f\|^2 = C^{-1} \int \int_{\mathbb{R}^2} |s|^{2p-3} f^* \psi_{s,t} \psi_{s,t}^* f = C^{-1} \int \int_{\mathbb{R}^2} |s|^{2p-3} |\tilde{f}(s,t)|^2. \quad (3.48)$$

A função

$$\rho(s,t) \equiv C^{-1} |s|^{2p-3} |\tilde{f}(s,t)|^2 \quad (3.49)$$

pode portanto ser interpretada como a *densidade de energia do sinal no plano tempo-escala*.

Seja o espaço de todas as transformadas *wavelets* (com relação a uma  $\psi$  mãe fixada) por  $\mathcal{F}$ :

$$\mathcal{F} \equiv \{\tilde{f} : f \in L^2(\mathbb{R})\}. \quad (3.50)$$

Então a Equação 3.48 mostra que  $\mathcal{F}$  é um subspaço do espaço de todas as funções  $h(s,t)$  que são de quadrado integrável com relação à função peso  $|s|^{2p-3}$ , isto é, para as quais a norma

$$\|h\|^2 \equiv C^{-1} \int \int_{\mathbb{R}^2} |s|^{2p-3} |h(s,t)|^2 dt ds < \infty. \quad (3.51)$$

O conjunto de todas essas tais funções forma um espaço de Hilbert (com produto interno obtido pela polarização (3.51)). Seja este espaço de Hilbert denotado por  $\mathcal{L}$ . Então (3.48) mostra que  $\|f\|_{L^2(\mathbb{R})}^2 = \|\tilde{f}\|_{\mathcal{L}}^2$ . Isto mostra que o mapeamento  $f \mapsto \tilde{f}$  define um *operador*  $T : L^2(\mathbb{R}) \rightarrow \mathcal{L}$  que satisfaz  $\|f\|_{L^2(\mathbb{R})}^2 = \|Tf\|_{\mathcal{L}}^2$ , análoga à fórmula de Plancherel da análise de Fourier. A Equação 3.50 afirma que a faixa de  $T$  é  $\mathcal{F}$ . Aplicando a identidade de polarização à Equação 3.48 resulta na identidade análoga à identidade de Parseval:

$$\langle f, g \rangle_{L^2(\mathbb{R})} = \langle Tf, Tg \rangle_{\mathcal{L}} \quad \text{para todo } f, g \in L^2(\mathbb{R}) \quad (3.52)$$

isto é, o operador  $T$  preserva os produtos internos e, portanto, “comprimentos” e “ângulos”. Tal operador é chamado de *isometria*. Assim  $T$  é apenas uma isometria parcial porque sua faixa  $\mathcal{F}$  não é todo o  $\mathcal{L}$ .

Nem toda função  $h \in \mathcal{L}$  pode ser a CWT de algum sinal  $f \in L^2(\mathbb{R})$ . Pois se  $h = \hat{f}$ , então por (3.47)

$$h(s',t') = \psi_{s',t'}^* f = \psi_{s',t'}^* I f = C^{-1} \int \int_{\mathbb{R}^2} |s|^{2p-3} \psi_{s',t'}^* \psi_{s,t} \psi_{s,t}^* f ds dt \quad (3.53)$$

$$= \int \int_{\mathbb{R}^2} |s|^{2p-3} K(s', t'|s, t) h(s, t) ds dt,$$

onde

$$K(s', t'|s, t) \equiv C^{-1} \langle \psi_{s', t'}, \psi_{s, t} \rangle = C^{-1} \langle \hat{\psi}_{s', t'}, \hat{\psi}_{s, t} \rangle \quad (3.54)$$

$$= C^{-1} |s'|^{1-p} \int_{-\infty}^{\infty} e^{2\pi i \omega(t' - t)} \bar{\hat{\psi}}(s' \omega) \hat{\psi}(s \omega) d\omega$$

pela Eq. 3.22.  $K(s', t'|s, t)$  é chamado *kernel de reprodução* associado com  $\psi$ . Como no caso da WFT, pode-se mostrar que a condição (3.53) não é somente necessária mas também suficiente para uma função  $h \in \mathcal{L}$  para pertencer a  $\mathcal{F}$ . Novamente, esta condição está relacionada à dependência linear entre as *wavelets*  $\psi_{s, t}$ . Existe também uma “aproximação de mínimos quadrados”  $f_h$  determinada pelas funções  $h(s, t)$  que não necessariamente pertencem a  $\mathcal{F}$ .

### 3.3.3 Localização em frequência e sinais analíticos

A CWT  $\tilde{f}$  representa uma *análise de tempo-escala* do sinal. Isto é muito próximo da análise de tempo-frequência discutida anteriormente. Será mostrada agora a relação entre estas duas análises. Adicionalmente, será apresentada uma forma efetiva de tratar sinais de *valores reais* considerando-se somente suas componentes de frequência positiva, os chamados sinais analíticos apresentados por D. Gabor (1946) [51]. As condições de admissibilidade dos Teoremas 3.3.1 e 3.3.2 exigem que  $\hat{\psi}(\omega) \rightarrow 0$  quando  $\omega \rightarrow 0$ . Se  $\hat{\psi}(\omega)$  for contínua (que é o caso, por exemplo, se  $\int |\psi(u)| du < \infty$ ), então segue que  $\hat{\psi}(0) = 0$ , isto é,

$$\int_{-\infty}^{\infty} \psi(u) du = 0 \quad (3.55)$$

ou seja, uma *wavelet* deve ser necessariamente uma “onda pequena”! Não pode também ser uma função como a gaussiana, ela tem que passear em volta do eixo dos tempos. Desde que  $\hat{\psi}(\omega)$  é de quadrado integrável, ela deve decair com  $|\omega| \rightarrow \infty$ . Se  $\hat{\psi}(\omega)$  decai rapidamente com  $|\omega| \rightarrow \infty$  e com  $\omega \rightarrow 0$ , então será pequena fora de uma “banda de frequências”  $\alpha \leq |\omega| \leq \beta$  para valores fixados de  $0 < \alpha < \beta$ . Então 3.22 mostra que  $\hat{\psi}_{s, t}(\omega) \approx 0$  fora da banda de frequências  $\alpha/|s| \leq |\omega| \leq \beta/|s|$ , e a Equação 3.23 mostra que  $\tilde{f}(s, t)$  contém informação sobre  $\hat{f}(\omega)$  principalmente nesta banda, com  $\hat{\psi}(s\omega)$  atuando como a janela localizadora em frequência. A WFT usa modulação no domínio do tempo para transladar a janela em frequência, de forma similar a CWT usa a resolução (*scaling*) no domínio do tempo para dimensionar a janela no domínio da frequência. Sob um

ponto de vista físico, a resolução de frequências parece mais natural: na música, ir uma oitava acima envolve dobrar a frequência, ao invés de deslocá-la por um termo aditivo constante. Naturalmente, a reconstrução é altamente redundante pois todas as escalas  $s \neq 0$  são usadas. Idealmente, o espectro completo de frequências deve ser coberto por escalas discretas de  $\psi$ . Isto é de fato o que ocorre na análise de *wavelets* discretas.

Se a *wavelet* mãe  $\psi$  é real, então sua transformada de Fourier tem reflexão simétrica, isto é,  $\overline{\hat{\psi}(\omega)} = \hat{\psi}(-\omega)$ . Portanto a banda de frequências efetiva de  $\psi$  mencionada anteriormente é simétrica com relação à origem. Alguma vezes é mais conveniente trabalhar com *wavelets* mães que têm somente frequências positivas, isto é, para as quais  $\hat{\psi}(\omega) = 0$  para  $\omega < 0$ . Tais *wavelets* são necessariamente de valores complexos (não reais) porque  $\hat{\psi}(\omega)$  não pode ser do tipo de reflexão simétrica.

Segundo Kaiser [33], a CWT e a WFT são casos especiais de um método mais geral de análise e reconstrução de sinais, denominado teoria dos *frames* [47] ou dos arcabouços em maior generalidade. O uso de *frames* na descrição de sinais é uma alternativa ao uso de bases. Enquanto as bases representam um número mínimo de vetores necessários para representar um vetor (sinal) qualquer, os *frames* são conjuntos com mais vetores que o mínimo necessário (ou seja, são redundantes). A família de funções  $\psi_{s,t}$  é linearmente dependente porque cada vetor do *frame* (que é infinito-dimensional) pode ser decomposto como uma superposição linear contínua dos outros vetores.

Define-se o operador síntese  $S$  da transformada *wavelet* inversa contínua como:

$$S = (T^*T)^{-1}T^* \quad (3.56)$$

em que  $T : X \rightarrow Y$  corresponde ao operador de análise definida pela CWT e  $T^* : Y \rightarrow X$  é o operador adjunto de  $T$ , que deve satisfazer a relação

$$\langle y, Tx \rangle = \langle T^*y, x \rangle. \quad (3.57)$$

Se  $(T^*T)^{-1}$  existe, como é o caso da CWT (e também da WFT), garante-se que vale a resolução da identidade

$$ST = I \quad (3.58)$$

Kaiser [33] afirma que se pode passar da descrição em tempo contínuo para

uma descrição em tempo discreto em que  $\tau = k\Delta\tau$ ,  $k \in \mathbb{Z}$ , e  $s = \sigma m \in \mathbb{R}$ , desde que o tempo de amostragem  $\Delta\tau$  seja suficientemente pequeno ( $\Delta\tau \approx 0$ ) e que o fator de escala  $\sigma$  seja escolhido suficientemente próximo da unidade, isto é,  $\sigma \approx 1$  (representação de  $x(t)$  num plano tempo-escala finamente discretizado). Este resultado não surpreende, porque, se *frames* contínuos são infinitamente redundantes, então espera-se que frames discretos, obtidos com  $\Delta\tau \approx 0$  e  $\sigma \approx 1$ , sejam altamente redundantes. Entretanto, Mallat propôs, em meados da década de 1980, um método radicalmente diferente (e surpreendente) de implementação da transformada *wavelet* discreta, em que sinais são representados com  $\Delta\tau$  finito e  $\sigma \neq 1 + \epsilon$ , com  $\epsilon$  arbitrário, denominado análise de multirresolução (MRA, *Multiresolution Analysis*). As propriedades de análise e reconstrução são mantidas, mesmo sem variações de escala e de tempo do tipo infinitesimal. A MRA é completamente recursiva, sendo portanto ideal para implementações computacionais.

Na MRA,  $\Delta\tau = 1$  e  $\sigma = 2$ , o que resulta numa forma de decomposição de  $x(t)$  em que as escalas de tempo são diádicas (potências de 2). A reconstrução de  $x(t)$  é perfeita. As *wavelets* usadas na MRA formam conjuntos de bases ortonormais ao invés de *frames*. Portanto, essas novas *wavelets* não podem ser obtidas via discretização de um *frame* contínuo genérico, pois  $\sigma = 2$ , como dito acima. A teoria da MRA permite a construção dessas novas *wavelets*, as quais devem satisfazer outras restrições além da condição de admissibilidade.

### 3.3.4 Análise de multirresolução e transformada *wavelet* discreta

Basicamente, existem dois tipos de transformadas *wavelets*: a CWT (*continuous wavelet transform*) e a DWT (*discrete wavelet transform*). A DWT pode ser formulada para sinais de tempo discreto (como fazem, por exemplo, Percival e Walden em [35]), sem que haja o estabelecimento de uma conexão explícita com a CWT. Por outro lado, não se deve entender que o termo “discreto” da DWT para sinais de tempo contínuo signifique que esta transformada seja definida sobre um sinal de tempo discreto, mas tão somente que os coeficientes produzidos por esta transformada pertencem a um subconjunto  $D = \{\omega_{j,k} = W_\psi(2^j, 2^j k), j \in \mathbb{Z}, k \in \mathbb{Z}\}$ . De fato, os coeficientes da DWT para sinais de tempo contínuo também podem ser obtidos diretamente, por meio da integral. A CWT de um sinal  $x(t)$  consiste num conjunto  $Q = \{W_\psi(s, \tau), s \in \mathbb{R}^+, \tau \in \mathbb{R}\}$ , em que  $\tau$  é o parâmetro de localização no tempo,  $s$  representa a escala e  $\psi$  denota uma função *wavelet*, de coeficientes *wavelet* no plano tempo-escala (também conhecido como plano

tempo-frequência) contínuo dados por<sup>1</sup>

$$W_\psi(s, \tau) = \left\langle \psi_{0(s, \tau)}, x \right\rangle = \int_{-\infty}^{\infty} \frac{1}{\sqrt{s}} \psi_0^* \left( \frac{\lambda - \tau}{s} \right) x(\lambda) d\lambda, \quad (3.59)$$

em que  $\psi_{0(s, \tau)}(t) = s^{-1/2} \psi_0 \left( \frac{t - \tau}{s} \right)$  denota uma versão dilatada e deslocada da *wavelet* “mãe”  $\psi_0(t)$ . O fator  $1/\sqrt{s}$  em (3.59) é usado para que todas as funções da classe

$$\mathcal{W} = \left\{ \frac{1}{\sqrt{s}} \psi_0 \left( \frac{t - \tau}{s} \right) \in \mathbb{R} \right\} \quad (3.60)$$

tenham a mesma energia (norma).

$$w_{j,k} = \langle \psi_{0(2^j, 2^j k)}, x \rangle = \int_{-\infty}^{\infty} 2^{-j/2} \psi_0^*(2^{-j} \lambda - k) x(\lambda) d\lambda, \quad (3.61)$$

em que os índices  $j$  e  $k$  são chamados de escala e localização, respectivamente, que não envolve um sinal de tempo discreto, mas o sinal de tempo contínuo  $x(t)$ . A Equação 3.61 mostra que a DWT de tempo contínuo corresponde a uma versão criticamente amostrada da CWT definida por (3.59) nas escalas diádicas  $s = 2^j$ ,  $j = \dots, -1, 0, 1, 2, \dots$ , em que os instantes de tempo na escala diádica  $s = 2^j$  estão separados por múltiplos de  $2^j$ . A função  $\psi_0$  de (3.61) deve ser definida a partir de uma análise de multirresolução (MRA) do sinal  $x(t)$  [47] a qual é apresentada na sequência. Observe-se que a teoria da MRA de tempo contínuo é similar à de tempo discreto (veja [35], por exemplo).

Uma MRA é por definição [52, 53, 54, 35] uma sequência de subespaços fechados  $\{V_j\}_{j \in \mathbb{Z}}$  de  $L^2(\mathbb{R})$  com as seguinte propriedades:

1.  $V_j \subset V_{j-1}$  para todo  $j \in \mathbb{Z}$ ;
2.  $x(t)$  é um membro de  $V_0$  se e somente se  $x_{0,k}(t)$  for um membro de  $V_0$  também, onde  $x_{0,k}(t) \equiv x(t - k)$  para  $k \in \mathbb{Z}$ ;
3.  $x(t)$  é um membro de  $V_0$  se e somente se  $x_{j,0}(t)$  for um membro de  $V_j$ , onde  $x_{j,0}(t) = x(t/2^j)/\sqrt{2^j}$  para  $j \in \mathbb{Z}$ ;
4.  $\bigcap_{j \in \mathbb{Z}} V_j = \{\vec{0}\}$ ; (onde aqui  $\vec{0}$  representa a função nula, isto é, uma função que é identicamente igual a zero em todos os pontos do eixo real.)
5.  $\overline{\bigcup_{j \in \mathbb{Z}} V_j} = L^2(\mathbb{R})$ ;
6. Existe uma função  $\phi(t) \in V_0$ , denominada *função de escala*, tal que o conjunto  $\{\phi_{0,k}(t) : k \in \mathbb{Z}\}$  forma uma base ortonormal para  $V_0$ , onde  $\phi_{0,k}(t) \equiv \phi(t - k)$ .

O subespaço  $V_j$  é conhecido como o espaço de aproximação associado à escala  $\lambda_j = 2^j$  (supondo-se que  $V_0$  seja o espaço de aproximação com escala unitária).

<sup>1</sup>Neste trabalho, o produto interno entre as funções  $f(\cdot)$  e  $g(\cdot)$  que pertencem a um espaço de funções definidas num domínio  $\mathcal{D}$  é dado por:  $\langle f, g \rangle = \int_{\mathcal{D}} w(x) f^*(x) g(x) dx$ , em que  $w(x)$  denota uma função não negativa arbitrária.



Se a projeção sobre  $V_j$  de  $x(t)$  é representada pelos coeficientes de escala

$$v_{j,k} = \langle \phi_{j,k}, x \rangle = \int_{-\infty}^{\infty} 2^{-j/2} \phi_0^*(2^{-j}t - k) x(t) dt, \quad (3.62)$$

então as propriedades 1 e 5 garantem que  $\lim_{j \rightarrow -\infty} \sum_k \phi_{j,k}(t) v_{j,k} = x(t)$ ,  $\forall x \in L^2(\mathbb{R})$ . A propriedade 3 implica que o subespaço  $V_j$  é uma versão em escala do subespaço  $V_0$  (multirresolução). A base ortonormal mencionada na propriedade 6 é obtida por translações no tempo da função passa-baixas  $\phi_j$ .

Considere a sequência de aproximações (também conhecidas na literatura como *wavelet smooths* [35], ou suavizações *wavelets*) sucessivas de  $x(t)$

$$\mathcal{S}_j(t) = \sum_k \phi_{j,k}(t) v_{j,k} \quad j = \dots, -1, 0, 1, \dots \quad (3.63)$$

Como  $V_{j+1} \subset V_j$ , tem-se que  $\mathcal{S}_{j+1}(t)$  é uma aproximação mais grosseira de  $x(t)$  do que  $\mathcal{S}_j(t)$ . Este fato ilustra a idéia fundamental da MRA, que consiste em eliminar a perda da informação quando se vai de  $\mathcal{S}_j(t)$  para  $\mathcal{S}_{j+1}(t)$ :

$$\mathcal{S}_j(t) = \mathcal{S}_{j+1}(t) + \Delta x_{j+1}(t), \quad (3.64)$$

em que  $\Delta x_{j+1}(t)$  (dito “detalhe” de  $x_j(t)$ ) pertence ao subespaço  $W_{j+1}$ , denominado espaço do detalhe [35] (também chamado de subespaço *wavelet*), o qual está associado às flutuações (ou variações) do sinal na escala de tempo mais refinada  $s_j = 2^j$  (qualitativamente, os coeficientes *wavelets*  $w_{j,k}$  da escala  $j$  são proporcionais às diferenças entre médias adjacentes do sinal  $x_t$  na escala de tempo  $\tau_j = s_{j-1} = 2^{j-1}$  [35], e que corresponde ao complemento ortogonal de  $V_{j+1}$  em  $V_j$ . A MRA mostra que os sinais de detalhe  $\Delta x_{j+1}(t) = \mathcal{D}_{j+1}(t)$  podem ser obtidos diretamente a partir de projeções sucessivas do sinal original  $x(t)$  sobre subespaços *wavelet*  $W_j$ . Além disso, a teoria da MRA demonstra que existe uma função  $\psi_0(t)$ , denominada “*wavelet mãe*”, que é obtida a partir de  $\phi_0(t)$ , tal que  $\psi_{j,k}(t) = 2^{-j/2} \phi_0(2^{-j}t - k)$   $k \in \mathbb{Z}$  é uma base ortonormal de  $W_j$ . O detalhe  $\mathcal{D}_{j+1}(t)$  é obtido pela equação

$$\mathcal{D}_{j+1}(t) = \sum_k \psi_{j+1,k}(t) \langle \psi_{j+1,k}(t), x(t) \rangle, \quad (3.65)$$

em que o produto interno  $\langle \psi_{j+1,k}(t), x(t) \rangle = w_{j+1,k}$  denota o coeficiente *wavelet* associado à escala  $j + 1$  e tempo discreto  $k$  e  $\{\psi_{j+1,k}(t)\}$  é uma família de funções *wavelets* que gera o subespaço  $W_{j+1}$ , ortogonal ao subespaço  $V_{j+1}$  ( $W_{j+1} \perp V_{j+1}$ )

isto é,

$$\langle \psi_{j+1,n}, \phi_{j+1,p} \rangle = 0, \quad \forall n, p. \quad (3.66)$$

Portanto, o sinal de detalhe  $\mathcal{D}_{j+1}(t)$  pertence ao subespaço complementar  $W_{j+1}$  de  $V_j$ , pois

$$V_j = V_{j+1} \oplus W_{j+1}, \quad (3.67)$$

ou seja,  $V_j$  é dado pela soma direta de  $V_{j+1}$  e  $W_{j+1}$ , e isto quer dizer que qualquer elemento em  $V_j$  pode ser determinado a partir da soma de dois elementos ortogonais pertencentes a  $V_{j+1}$  e  $W_{j+1}$ . Iterando-se (3.67), tem-se que

$$V_j = W_{j+1} \oplus W_{j+2} \oplus \dots \quad (3.68)$$

A Equação 3.68 diz que a aproximação  $\mathcal{S}_j(t)$  é dada por

$$\mathcal{S}_j(t) = \sum_{i=j+1}^{\infty} \sum_k w_{i,k} \psi_{i,k}(t). \quad (3.69)$$

A MRA de um sinal de tempo contínuo  $x(t)$  é iniciada com a determinação dos coeficientes  $v_0(k) = \langle \phi_{0,k}(t), x(t) \rangle$ , em que  $k = 0, 1, \dots, N - 1$ , que estão associados à projeção de  $x(t)$  no subespaço de aproximação  $V_0$ . Em seguida, a sequência  $v_0(k)$  é decomposta via filtragem e subamostragem por um fator de 2 (*downsampling*) em duas sequências:  $\{v_1(k)\}$  e  $\{w_1(k)\}$ , cada uma contendo  $N/2$  pontos. Este processo de filtragem e subamostragem é repetido várias vezes, obtendo-se as sequências

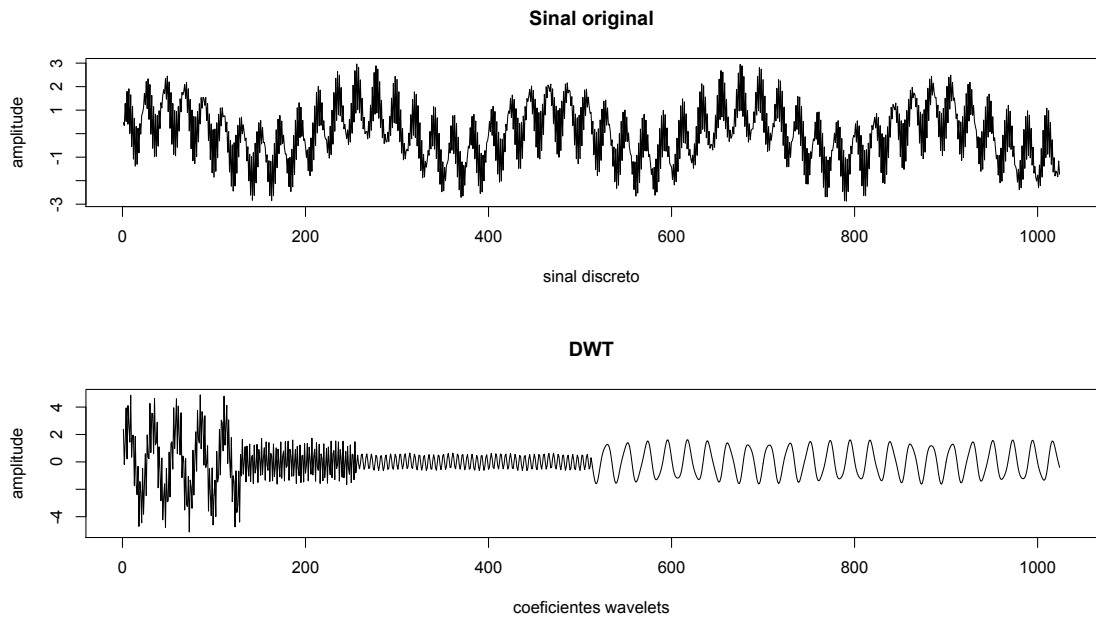
$$\{\{v_0(k)\}_N, \{v_1(k)\}_{\frac{N}{2}}, \{v_2(k)\}_{\frac{N}{4}}, \dots, \{v_j(k)\}_{\frac{N}{2^j}}, \dots, \{v_J(k)\}_{\frac{N}{2^J}}\} \quad (3.70)$$

e

$$\{\{w_1(k)\}_{\frac{N}{2}}, \{w_2(k)\}_{\frac{N}{4}}, \dots, \{w_j(k)\}_{\frac{N}{2^j}}, \dots, \{w_J(k)\}_{\frac{N}{2^J}}\}. \quad (3.71)$$

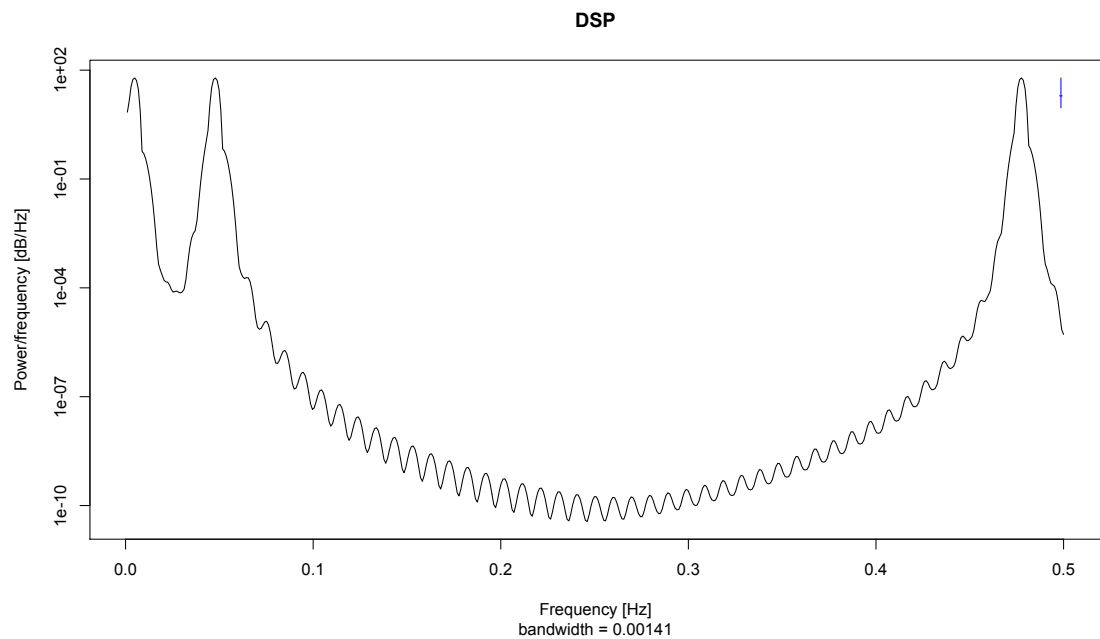
Este conjunto de coeficientes forma a DWT (*Discrete Wavelet Transform*) de um sinal  $x(t)$ .

A Figura 3.9 ilustra a DWT de 3 níveis (decomposição nas escalas  $j = 1, 2, 3$ ) associada a 1024 amostras do sinal de tempo discreto  $x(k) = \text{sen}(3k) + \text{sen}(0,3k) + \text{sen}(0,03k)$ , que corresponde à superposição de 3 senóides nas frequências  $f_1 \approx 0,004775$ ,  $f_2 \approx 0,04775$  e  $f_3 \approx 0,4775$ .



**Figura 3.9:** Uma ilustração da DWT em 3 níveis (figura obtida com o software R).

A Figura 3.10 mostra a densidade espectral de potência (DEP) deste sinal.



**Figura 3.10:** Densidade espectral de potência do sinal  $x(k)$  do exemplo.

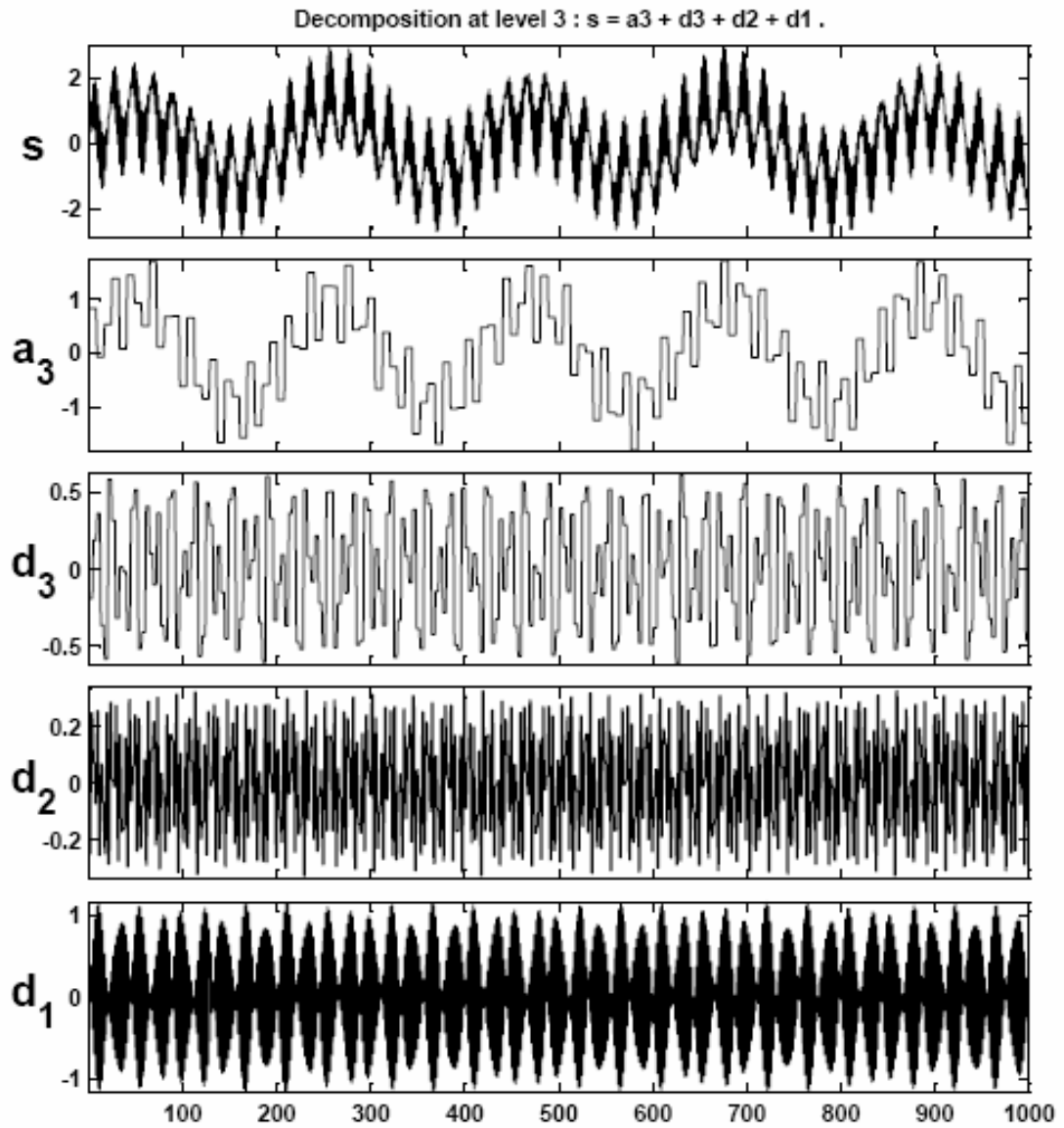
A reconstrução de  $x(t)$  é implementada via filtragem e sobreamostragem por um fator de 2 (*upsampling*) das sequências (3.70) e (3.71), obtendo-se uma apro-

ximação de  $x(t)$  no subespaço  $V_0$

$$\mathcal{S}_0(t) = \mathcal{S}_J(t) + \mathcal{D}_1(t) + \mathcal{D}_2(t) + \dots + (\mathcal{D})_J(t) \quad (3.72)$$

ou

$$x(t) \approx \sum_k v(J, k) \phi_{J,k}(t) + \sum_{j=1}^J \sum_k w_{j,k} \psi_{j,k}(t). \quad (3.73)$$



**Figura 3.11:** Síntese do sinal  $x(k)$  (figura adaptada de [45]).

A Equação 3.73 define a transformada discreta *wavelet* inversa - *Inverse Discrete Wavelet Transform* (IDWT). A Figura 3.11 ilustra a síntese do sinal  $x(k) = \text{sen}(3k) + \text{sen}(0,3k) + \text{sen}(0,03k)$  conforme a Equação 3.72 (utilizou-se a *wavelet* de Haar).

Diz-se que a função  $\phi_0(t) = \phi(t)$  determina uma MRA de  $x(t)$  de acordo com (3.72), se a mesma obedece às seguintes condições:

1. Ortonormalidade intra-escala

$$\langle \phi(t - m), \phi(t - n) \rangle = \delta_{m,n} \quad (3.74)$$

onde  $\delta_{m,n}$  é o delta de Kronecker ( $\delta_{m,n} = 1$  se  $m = n$ ,  $\delta_{m,n} = 0$  para  $m \neq n$ ). A Equação 3.74 impõe uma condição de ortonormalidade na escala  $j = 0$ .

2. Média unitária

$$\int_{-\infty}^{\infty} \phi(t) dt = 1. \quad (3.75)$$

3.

$$\frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) = \sum_n g_n \phi(t - n), \quad (3.76)$$

pois cabem várias  $\phi(t - k)$  em  $\phi\left(\frac{t}{2}\right)$  (é uma consequência da Propriedade (1) da MRA).

A Equação 3.76 pode ser reescrita na forma

$$\phi(t) = \sum_n \sqrt{2} g_n \phi(2t - n), \quad (3.77)$$

sendo conhecida como Equação de Dilação. As Equações 3.76 e 3.77 podem ser escritas, respectivamente, no domínio das frequências como

$$\sqrt{2} \Phi(2\nu) = G(\nu) \Phi(\nu), \quad (3.78)$$

e

$$\Phi(\nu) = \frac{1}{\sqrt{2}} G\left(\frac{\nu}{2}\right) \Phi\left(\frac{\nu}{2}\right), \quad (3.79)$$

em que  $\Phi(\nu)$  é a transformada de Fourier de  $\phi(t)$  e  $G(\nu) = \sum_n g_n e^{-j2\pi\nu n}$ , conhecido como **filtro de escala** (passa-baixas), representa um filtro periódico em  $\nu$ .

Como o subespaço  $W_{j+1}$  é ortogonal a  $V_{j+1}$  e está contido em  $V_j$ , tem-se que

$$\frac{1}{\sqrt{2}} \psi\left(\frac{t}{2}\right) = \sum_n h_n \phi(t - n), \quad (3.80)$$

ou

$$\psi(t) = \sum_n \sqrt{2} h_n \psi(2t - n), \quad (3.81)$$

que é a Equação da *Wavelet*. Aplicando-se a transformada de Fourier em (3.80) e (3.81) obtém-se, respectivamente,

$$\sqrt{(2)}\Psi(2\nu) = H(\nu)\Phi(\nu) \quad (3.82)$$

e

$$\Psi(\nu) = \frac{1}{\sqrt{2}}H\left(\frac{\nu}{2}\right)\Phi\left(\frac{\nu}{2}\right) \quad (3.83)$$

onde  $H(\nu)$  é filtro *wavelet* (passa-altas).

Reescrevendo-se (3.66) em termos do domínio das frequências e usando-se (3.78) e (3.82) resulta na condição de ortogonalidade.

$$\int_{-\infty}^{\infty} G(\nu) H^*(\nu) \|\Phi(\nu)\|^2 d\nu = 0, \quad (3.84)$$

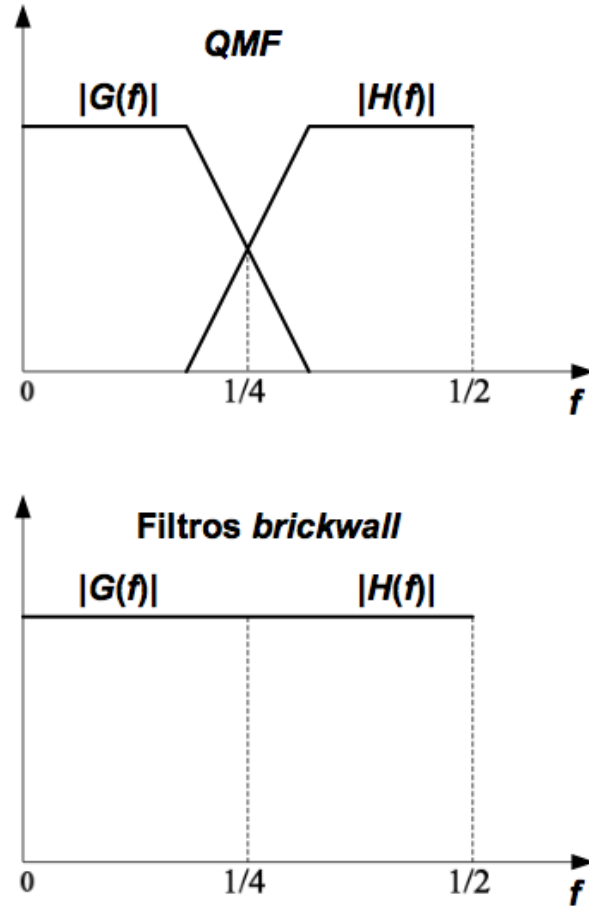
que o filtro  $H$  deve atender para que a família  $\{\psi_{1,k}(t)\}$  seja ortogonal à família  $\{\phi_{1,k}(t)\}$ . Pode-se mostrar, [35], que a condição

$$h_n = (-1)^n g_{L-1-n} \leftrightarrow H(z) = -z^{-L+1} G(-z^{-1}) \quad (3.85)$$

onde  $L$  denota o comprimento de um filtro FIR (*Finite Impulse Response*)  $g_n$ , é suficiente para que (3.84) seja válida. Diz-se que  $g_n$  e  $h_n$  são **filtros espelhados em quadratura** (***Quadrature Mirror Filters (QMF)***) quando estão relacionados por (3.85). A Figura 3.12 mostra os gráficos de resposta em frequência dos filtros QMF e também ilustra a resposta em frequência de filtros do tipo *brickwall*, que não são fisicamente realizáveis.

De acordo com a Equação 3.77, a MRA começa a partir de uma definição (dentre várias possíveis) da função de escala  $\phi(t)$ , que está relacionada ao filtro de escala  $g_n$  pela Equação 3.76. A Equação 3.85 diz que a escolha de um filtro  $\{g_n\}$  do tipo FIR implica um  $\{h_n\}$  que também seja FIR. Finalmente, a função *wavelet* é determinada por (3.80). As funções de escala  $\phi(t)$  e *wavelet*  $\psi(t)$  associadas aos filtros FIR  $\{g_n\}$  e  $\{h_n\}$  possuem suporte compacto, oferecendo, portanto, a funcionalidade de *resolução temporal*.

A função de escala mais simples que satisfaz (3.74) é a função característica



**Figura 3.12:** Resposta em frequência de filtros QMF (gráfico superior); resposta em frequência de filtros do tipo *brickwall* (gráfico inferior).

do intervalo  $I = [0, 1)$ , que corresponde à função de escala de Haar:

$$\phi^{(H)}(t) = \chi_{[0,1)}(t) = \begin{cases} 1, & \text{se } 0 \leq t < 1 \\ 0, & \text{caso contrário.} \end{cases} \quad (3.86)$$

Neste caso (MRA de Haar), o filtro de escala de Haar associado é dado por

$$g_n = \{\dots, 0, g_0 = 1/\sqrt{2}, g_1 = 1/\sqrt{2}, 0, \dots\} \quad (3.87)$$

o filtro *wavelet* de Haar por

$$h_n = \{\dots, 0, h_0 = g_1 = 1/\sqrt{2}, h_1 = -g_0 = -1/\sqrt{2}, 0, \dots\} \quad (3.88)$$

e a função *wavelet* de Haar por

$$\psi^{(H)}(t) = \chi_{[0,1/2)}(t) - \chi_{[1/2,1)}(t). \quad (3.89)$$

Existem certas funções de escala e *wavelets* chamadas de funções de Daubechies [47] com  $N = 2, 3, 4$  momentos ou cumulantes nulos (*vanishing moments*)

$$\int_{-\infty}^{\infty} t^m \psi(t) dt = 0, \quad m = 0, 1, \dots, N - 1. \quad (3.90)$$

Ingrid Daubechies [47] foi a primeira a propor um método para construção de sequências de funções de transferência  $\{G^{(N)}(z)\}_{N=1,2,3,\dots}$  e  $\{H^{(N)}(z)\}_{N=1,2,3,\dots}$ , em que  $G^{(N)}(z)$  está associada ao filtro FIR passa-baixas  $g_n^{(N)}$  e  $H^{(N)}(z)$  ao filtro passa-altas  $h_n^{(N)}$ . As funções de escala e *wavelet* correspondentes têm suporte em  $[0, 2N - 1]$ . O primeiro membro da sequência é o sistema de Haar  $\phi^{(1)} = \phi^{(H)}$ ,  $\psi^{(1)} = \psi^{(H)}$ . Os filtros de Daubechies são generalizações do sistema de Haar para  $N \geq 2$ .

Demonstra-se que [53]

$$v_j(n) = \sum_k g(k - 2n) v_{j-1}(k) \quad (3.91)$$

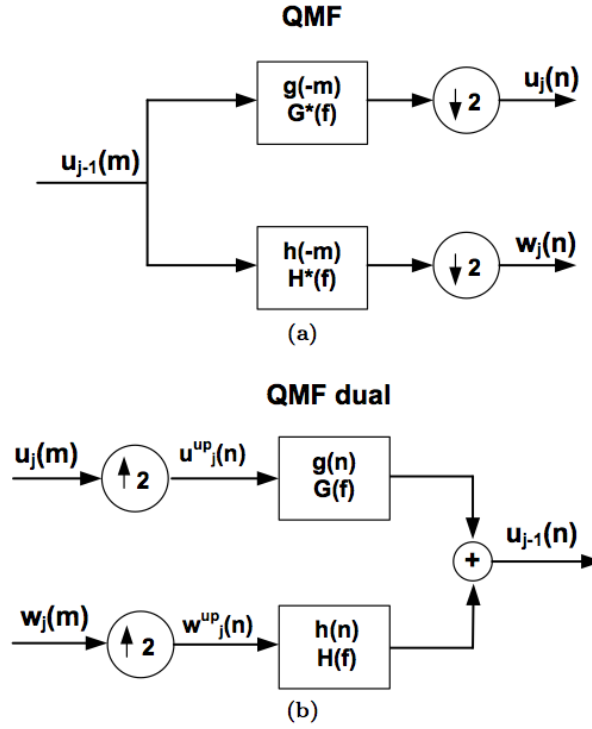
e que

$$w_j(n) = \sum_k h(k - 2n) v_{j-1}(k). \quad (3.92)$$

Segundo as Equações (3.91) e (3.92), os coeficientes  $v_j(n)$  e  $w_j(n)$  podem ser obtidos a partir dos coeficientes de escala  $v_{j-1}(m)$  por meio de uma operação de decimação da sequência  $\{v_{j-1}(m)\}$  por um fator de 2. A decimação consiste no cascadeamento de um filtro passa-baixas  $h(-m)$  (com função de transferência  $\bar{H}(z) = H(1/z)$  e resposta em frequência  $H^*(f)$ ) ou passa-altas  $g(-m)$  (com função de transferência  $\bar{H}(z) = G(1/z)$  e resposta em frequência  $G^*(f)$ ) com um compressor (ou decimador) por um fator de 2, conforme Figura 3.13. Note que decimar um sinal por um fator  $D$  é o mesmo que reduzir sua taxa de amostragem em  $D$  vezes.

A Figura 3.13 item *a*) indica que os coeficientes de uma transformada *wavelet* discreta podem ser determinados a partir de um algoritmo piramidal baseado em convoluções com filtros espelhados em quadratura (conforme proposta original de Mallat [53]) [45]. Sendo assim, a MRA é implementada via bancos de filtros de análise passa-baixas  $H^*(f)$  e passa-altas  $G^*(f)$  adequadamente posicionados para separação das sequências de coeficientes de escala das sequências de coeficientes





**Figura 3.13:** (a) banco de filtros de análise  $H^*(f)$  (passa-baixas) e  $G^*(f)$  (passa-altas) com decimação (*downsampling*); (b) banco de filtros de reconstrução com interpolação (*upsampling*). São usados os filtros duais  $G(f)$  e  $F(f)$ .

*wavelet*. Posteriormente é possível reconstruir o sinal original utilizando-se o banco de filtros duais de reconstrução QMF passa-baixas  $G(f)$  e passa-altas  $H(f)$  [53], conforme mostra a Figura 3.13, item b). De acordo com esta Figura 3.13, item b),  $v_{j-1}(n)$  pode ser reconstruído por meio da inserção de zeros entre cada duas amostras de  $v_j(m)$  e  $w_j(m)$ , gerando-se os sinais  $v_j^{up}(n)$  e  $w_j^{up}(n)$  nas saídas dos interpoladores (a inserção de zeros é conhecida como operação de interpolação, em que há um aumento da taxa de amostragem do sinal *upsampling*), aos quais se seguem convoluções com os filtros  $G(f)$  e  $H(f)$ , respectivamente. Cabe ressaltar aqui a complexidade do algoritmo da pirâmide: assumindo que se queira calcular a DWT de  $N$  amostras a complexidade é igual a  $O(N)$ ; ao passo que o cálculo “direto” da DWT (que envolve multiplicação de matrizes) é  $O(N^2)$  [35].

## 4 Detecção Multiusuário com aplicação de *wavelets*

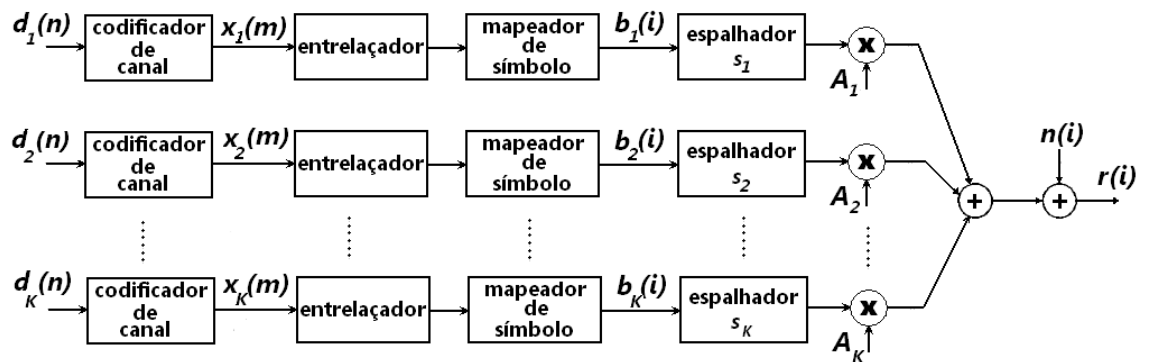
### Descrição Sistêmica

Neste capítulo será apresentado o sistema de detecção multiusuário com aplicação de *wavelets*.

## 4.1 Detecção multiusuário adaptativa bayesiana para sistemas de espalhamento espectral

O objetivo deste sistema é a detecção multiusuário de sinais corrompidos por ruído branco, gaussiano e aditivo, de média zero, em um canal sem fio; a estimação de parâmetros desconhecidos dos sinais recebidos (no caso específico do trabalho desenvolvido, os parâmetros são a variância do ruído, as amplitudes dos sinais de informação), utilizando uma abordagem bayesiana; e a diminuição do ruído adquirido do canal utilizando-se a transformada *wavelet* e um código especial. Este sistema é uma evolução da proposta de Wang e Chen, de um receptor multiusuário adaptativo que utiliza abordagem bayesiana [9, 55], e contempla testes do efeito *near-far* para o sistema proposto; testes com alguns códigos de espalhamento diferentes do da proposta original, como o código Gold, e variação do comprimento desses códigos ou sequências de espalhamento; e simulação do sistema assíncrono. Em resumo a proposta deste trabalho foi simular o sistema originalmente proposto e desenvolver de forma independente uma evolução do sistema, forma a validar o código escrito na linguagem, ou ambiente, R aqui desenvolvido e criar novas possibilidades de análises e testes relativos às técnicas utilizadas.

É considerado um sistema de espalhamento espectral, de acesso múltiplo, síncrono, com  $K$  usuários, que emprega sinais normalizados de modulação  $s_1, s_2, \dots, s_K$ . A transmissão dos sinais gerados por este sistema ocorre em um canal de ruído aditivo, branco e gaussiano (AWGN, *additive white gaussian noise*). O diagrama em blocos do transmissor desse sistema é apresentado na Figura 4.1.



**Figura 4.1:** Sistema de espalhamento espectral de acesso múltiplo.

Nesse sistema, o bloco codificador de canal é um elemento típico de um sistema de comunicação digital. Os dígitos binários de informação  $\{d_k(n)\}$  do usuário  $k$  são codificados utilizando-se um código de canal, como por exemplo um código de bloco, um código convolucional, ou um código entrelaçado, etc. Um entrelaçador de bits (*interleaver*) é utilizado na saída do codificador de canal, para prevenir erros múltiplos adjacentes que possam ocorrer devido ao canal. Os bits de código entrelaçados são mapeados para símbolos BPSK (*Binary Phase Shift Keying*), gerando o conjunto  $\{b_k(i)\}$  de símbolos. Cada símbolo é então espalhado por um sinal, ou código, de espalhamento  $\mathbf{s}_k$  e transmitido através do canal.

O sinal recebido no destino é a superposição dos sinais transmitidos pelos  $K$  usuários mais o ruído adicionado pelo meio, e é dado por:

$$\mathbf{r}(i) = \sum_{k=1}^K A_k b_k(i) \mathbf{s}_k + \mathbf{n}(i), \quad i = 0, \dots, M-1 \quad (4.1)$$

Na Equação 4.1,  $M$  é o número de símbolos de dados, por usuário e por quadro;  $A_k$ ,  $b_k(i)$  e  $\mathbf{s}_k$  denotam, respectivamente, a amplitude do sinal, o  $i$ -ésimo símbolo, e o sinal de espalhamento normalizado do  $k$ -ésimo usuário; e  $\mathbf{n}(i) = [n_0(i) \ n_1(i) \ \dots \ n_{P-1}(i)]^T$  é o vetor de ruído branco, com média zero. O sinal de espalhamento é da forma:

$$\mathbf{s}_k = \frac{1}{\sqrt{P}} [\beta_{k,0} \ \beta_{k,1} \ \dots \ \beta_{k,P-1}]^T, \quad \beta_{k,j} \in \{+1, -1\} \quad (4.2)$$

onde  $P$  é o fator de espalhamento. Assume-se que o receptor conhece o sinal de espalhamento de todos os usuários ativos no sistema. O receptor está em uma estação rádio base, por exemplo.

As probabilidades de símbolo *a priori* são dadas por

$$\rho_k(i) \equiv p[b_k(i) = +1], \quad i = 0 \dots, M-1; \quad k = 1, \dots, K. \quad (4.3)$$

Note-se que quando não há informação prévia disponível, adota-se  $\rho_k(i) = \frac{1}{2}$ , isto é, todos os símbolos são igualmente prováveis e a incerteza será máxima, quanto a qual símbolo foi enviado. Assume-se ainda que o ruído ambiente aditivo  $\{\mathbf{n}(i)\}$  é uma sequência de vetores aleatórios, independentes e identicamente distribuídos; de média zero; e que é independente também da sequência de símbolos  $\{b_k(i)\}_{k=1}^K$ . Além disso, o ruído  $\mathbf{n}(i)$  é por hipótese composto de amostras independentes e identicamente distribuídas  $\{\mathbf{n}_j(i)\}_{j=0}^{P-1}$ . A distribuição correspondente ao ruído gaussiano aditivo  $\mathbf{n}_j(i)$  é:

$$\mathbf{n}_j(i) \sim \mathcal{N}(0, \sigma^2) \quad (4.4)$$

onde  $\sigma^2$  é a variância do ruído.

Por definição, a sequência de sinais recebidos é  $\mathbf{Y} \equiv \{\mathbf{r}(0), \mathbf{r}(1), \dots, \mathbf{r}(M-1)\}$ . O problema da estimação é o da determinação das probabilidades *a posteriori* dos símbolos transmitidos

$$p[b_k(i) = +1 | \mathbf{Y}], \quad i = 0, \dots, M-1; \quad k = 1, \dots, K \quad (4.5)$$

com base nos sinais recebidos  $\mathbf{Y}$  e na informação *a priori*  $\{\rho_k(i)\}_{k=1; i=0}^{K, M-1}$ , sem conhecer as amplitudes  $\{A_k\}_{k=1}^K$  ou o parâmetro do ruído (isto é, a variância  $\sigma^2$ ). Essas probabilidades *a posteriori* são então utilizadas pelo decodificador do receptor para decodificar, ou recuperar, os bits de informação  $\{d_k(n)\}$  originais.

## 4.2 Detector multiusuário bayesiano adaptativo

O problema de se computar as probabilidades *a posteriori* da Equação 4.5 sob a hipótese de que o ruído ambiente é gaussiano é considerado a seguir.

A função distribuição de probabilidade de  $\mathbf{n}(i)$  é dada por:

$$p[\mathbf{n}(i)] = \frac{1}{(2\pi\sigma^2)^{P/2}} \exp\left(-\frac{\|\mathbf{n}(i)\|^2}{2\sigma^2}\right). \quad (4.6)$$

Sejam, por definição, as seguintes variáveis:

$$\mathbf{x}(i) \equiv [b_1(i) \quad b_2(i) \quad \dots \quad b_K(i)]^T, \quad i = 0, 1, \dots, M-1, \quad (4.7)$$

sendo  $\mathbf{x}(i)$  o vetor de símbolos;

$$\mathbf{B}(i) \equiv \text{diag}[b_1(i), b_2(i), \dots, b_K(i)], \quad i = 0, 1, \dots, M-1, \quad (4.8)$$

sendo  $\mathbf{B}(i)$  a matriz diagonal de símbolos;

$$\mathbf{X} \equiv [\mathbf{x}(0) \quad \mathbf{x}(1) \quad \dots \quad \mathbf{x}(M-1)], \quad (4.9)$$

sendo  $\mathbf{X}$  o vetor dos vetores de símbolos;

$$\mathbf{Y} \equiv [\mathbf{r}(0) \quad \mathbf{r}(1) \quad \dots \quad \mathbf{r}(M-1)], \quad (4.10)$$

sendo  $\mathbf{Y}$  o vetor dos sinais recebidos do canal;

$$\mathbf{a} \equiv [A_1 \quad A_2 \quad \dots \quad A_K]^T, \quad (4.11)$$

sendo  $\mathbf{a}$  o vetor das amplitudes;

$$\mathbf{S} \equiv [\mathbf{s}_1 \quad \mathbf{s}_2 \quad \dots \quad \mathbf{s}_K], \quad (4.12)$$

e sendo  $\mathbf{S}$  o vetor dos sinais de espalhamento.

Assim, a Equação 4.1 pode ser escrita como:

$$\mathbf{r}(i) = \mathbf{S}\mathbf{A}\mathbf{x}(i) + \mathbf{n}(i) \quad (4.13)$$

$$\mathbf{r}(i) = \mathbf{S}\mathbf{B}(i)\mathbf{a} + \mathbf{n}(i), \quad i = 0, 1, \dots, M-1 \quad (4.14)$$

O Amostrador de Gibbs é um algoritmo conhecido em Estatística. Ele permite amostrar certas distribuições complexas, e será utilizado para calcular as estimações máximas *a posteriori* (MAP) das quantidades ou variáveis desconhecidas ( $\mathbf{a}$ ,  $\sigma^2$  e  $\mathbf{X}$ ). Mais detalhes sobre o algoritmo Amostrador de Gibbs serão vistos nos parágrafos seguintes e no capítulo dos resultados do trabalho. Um maior aprofundamento sobre o assunto pode ser encontrado nas referências [27, 56].

### Inferência bayesiana e a necessidade de simulações Monte Carlo

Supõe-se que as quantidades desconhecidas  $\mathbf{a}$ ,  $\sigma^2$  e  $\mathbf{X}$  são independentes entre si e têm distribuições *a priori*  $p(\mathbf{a})$ ,  $p(\sigma^2)$  e  $p(\mathbf{X})$ , respectivamente. Sendo  $\{\mathbf{n}(i)\}_{i=0}^{M-1}$  uma sequência de vetores gaussianos independentes, utilizando-se (4.6) e (4.13), a distribuição conjunta *a posteriori* dessas quantidades desconhecidas ( $\mathbf{a}$ ,  $\sigma^2$ ,  $\mathbf{X}$ ), baseada no sinal recebido  $\mathbf{Y}$ , é dada por:

$$p(\mathbf{a}, \sigma^2, \mathbf{X} | \mathbf{Y}) = p(\mathbf{Y} | \mathbf{a}, \sigma^2, \mathbf{X}) p(\mathbf{a}) p(\sigma^2) p(\mathbf{X}) / p(\mathbf{Y}) \quad (4.15)$$

$$p(\mathbf{a}, \sigma^2, \mathbf{X} | \mathbf{Y}) = C \left( \frac{1}{\sigma^2} \right)^{PM/2} \exp \left( - \frac{1}{2\sigma^2} \sum_{i=0}^{M-1} |\mathbf{r}(i) - \mathbf{S}\mathbf{A}\mathbf{x}(i)|^2 \right) \times p(\mathbf{a}) p(\sigma^2) p(\mathbf{X}) \quad (4.16)$$

onde  $C$  é uma constante de normalização independente dos parâmetros desconhecidos ( $\mathbf{a}$ ,  $\sigma^2$ ,  $\mathbf{X}$ ). As probabilidades *a posteriori*, dadas pela Equação 4.5 dos símbolos transmitidos, podem ser então obtidas a partir da distribuição conjunta *a posteriori*, Equação 4.16, de acordo com:

$$p[x_k(i) = +1 | \mathbf{Y}] = \sum_{\mathbf{X}: x_k(i)=+1} p(\mathbf{X} | \mathbf{Y}) = \sum_{\mathbf{X}: x_k(i)=+1} \int p(\mathbf{a}, \sigma^2, \mathbf{X} | \mathbf{Y}) d\mathbf{a} d\sigma^2. \quad (4.17)$$

Pode-se perceber que a solução da Equação 4.17 não é trivial, pois envolve  $2^{KM-1}$  integrais multidimensionais, o que torna inviável qualquer implementação prática. Para se evitar o cálculo direto da estimação e detecção por inferência bayesiana, Equação 4.17, recorre-se ao Amostrador de Gibbs. A idéia básica é gerar amostras aleatórias ergódicas  $\{\mathbf{a}^{(n)}, (\sigma^2)^{(n)}, \mathbf{X}^{(n)} : n = n_0, n_0 + 1, \dots\}$  da distribuição *a posteriori*, Equação 4.16, e então tirar a média  $\{x_k(i)^{(n)} : n = n_0, n_0 + 1, \dots\}$  para se obter uma aproximação das probabilidades *a posteriori*, Equação 4.17.

### Amostrador de Gibbs [27, 30]

O Amostrador de Gibbs é um algoritmo que resolve o problema da amostragem de certas distribuições consideradas complexas. No Capítulo 5, sobre os algoritmos Metropolis-Hastings e Amostrador de Gibbs, são apresentados mais detalhes sobre este algoritmo.

Sendo  $\theta = [\theta_1 \ \theta_2 \ \dots \ \theta_d]^T$  um vetor de parâmetros desconhecidos e  $\mathbf{Y}$  o vetor dos dados observados, suponha que se queira encontrar a distribuição marginal *a posteriori* de algum parâmetro, como por exemplo  $\theta_j$ , condicionado à observação  $\mathbf{Y}$ , isto é,  $p(\theta_j|\mathbf{Y})$ ,  $1 \leq j \leq d$  (onde  $d$  é a dimensão do parâmetro). O cálculo direto envolve integrar sobre os demais parâmetros da densidade conjunta posterior, ou seja,

$$p(\theta_j|\mathbf{Y}) = \int \int \dots \int p(\theta|\mathbf{Y}) d\theta_1 \dots d\theta_{j-1} d\theta_{j+1} \dots d\theta_d. \quad (4.18)$$

Na maioria dos casos, tentar resolver um cálculo laborioso e complexo como esse diretamente não é computacionalmente viável, especialmente quando a dimensão  $d$  do parâmetro é grande. O Amostrador de Gibbs é um procedimento de simulação Monte Carlo (MCMC, *Markov chain Monte Carlo*) para o cálculo numérico de integrais multidimensionais, como por exemplo a Equação 4.18. O método consiste em gerar amostras aleatórias da distribuição conjunta posterior  $p(\theta_j|\mathbf{Y})$ , e então estimar cada uma das distribuições marginais utilizando essas amostras.

O comportamento de convergência do Amostrador de Gibbs é investigado na literatura [57, 58] e condições gerais são dadas para os dois seguintes resultados:

- A distribuição de  $\theta^{(n)}$  converge geometricamente para  $p(\theta|\mathbf{Y})$ , quando  $n \rightarrow \infty$ .
- $\frac{1}{N} \sum_{n=1}^N f(\theta^{(n)}) \rightarrow \int f(\theta)p(\theta|\mathbf{Y})d\theta$ , quando  $N \rightarrow \infty$ , para qualquer função integrável  $f$ . (a convergência aqui é do tipo “quase certamente”!)

O Amostrador de Gibbs requer um período inicial de transição para convergir até o equilíbrio. O período inicial de comprimento  $n_0$  é conhecido como “*burnin*”,

e as primeiras  $n_0$  amostras devem ser sempre descartadas. A detecção da convergência é feita usualmente de forma particular, isto é, não há um procedimento de caráter universal ainda estabelecido.

### 4.2.1 Determinação das probabilidades do modelo

#### Distribuições *a priori*

Com relação ao vetor de amplitudes desconhecidas,  $\mathbf{a}$ , supõe-se uma distribuição *a priori* gaussiana truncada:

$$p(\mathbf{a}) \propto \mathcal{N}(\mathbf{a}_0, \Sigma_0) I_{d\{\mathbf{a}>0\}} \quad (4.19)$$

onde  $I_{d\{\mathbf{a}>0\}}$  é um indicador, para o qual se atribui o valor 1 caso todos os elementos de  $\mathbf{a}$  sejam positivos, e zero, no caso contrário.

Assume-se para a variância do ruído  $\sigma^2$  uma distribuição *a priori* Qui-Quadrado Inversa (distribuição *a priori* conjugada [30]).

$$p(\sigma^2) = \frac{\left(\frac{v_0 \lambda_0}{2}\right)^{v_0/2}}{\Gamma\left(\frac{v_0}{2}\right)} \left(\frac{1}{\sigma^2}\right)^{(v_0/2)+1} \exp\left(-\frac{v_0 \lambda_0}{2\sigma^2}\right) \sim \chi^2(v_0, \lambda_0) \quad (4.20)$$

O valor pequeno de  $v_0$  corresponde a distribuições *a priori* menos informativas. O valor de  $v_0 \lambda_0$  reflete a confiança do valor de  $\sigma^2$ .

Finalmente, supondo-se que os símbolos  $\{x_k(i)\}_{k=1; i=0}^{K; M-1}$  sejam independentes, a distribuição *a priori*  $p(\mathbf{X})$  pode ser expressa em termos das probabilidades *a priori* de símbolos definidas na Equação 4.3 como

$$p(\mathbf{X}) = \prod_{i=0}^{M-1} \prod_{k=1}^K \rho_k(i)^{\delta_{ki}} [1 - \rho_k(i)]^{1-\delta_{ki}} \quad (4.21)$$

onde  $\delta_{ki}$  é o indicador tal que  $\delta_{ki} = 1$  se  $b_k(i) = +1$  e  $\delta_{ki} = 0$  se  $b_k(i) = -1$ .

#### Distribuições condicionais *a posteriori*

As seguintes distribuições condicionais *a posteriori* são exigidas pelo detector multiusuário, que utiliza o Amostrador de Gibbs para resolver o problema multidimensional, e que transmite num canal de ruído branco gaussiano.

1—) A distribuição condicional do vetor amplitude  $\mathbf{a}$ , sendo dados  $\sigma^2$ ,  $\mathbf{X}$  e  $\mathbf{Y}$ , é dada por:

$$p(\mathbf{a}|\sigma^2, \mathbf{X}, \mathbf{Y}) \propto \mathcal{N}(\mathbf{a}_*, \Sigma_*) I_{\{\mathbf{a}>0\}} \quad (4.22)$$



com

$$\Sigma_*^{-1} \equiv \Sigma_0^{-1} + \frac{1}{\sigma^2} \sum_{i=0}^{M-1} \mathbf{B}(i)(\mathbf{S}^T \times \mathbf{S})\mathbf{B}(i), \quad (4.23)$$

e

$$\mathbf{a}_* \equiv \Sigma_* \left( \Sigma_0^{-1} \mathbf{a}_0 + \frac{1}{\sigma^2} \sum_{i=0}^{M-1} \mathbf{B}(i) \mathbf{S}^T \mathbf{r}(i) \right) \quad (4.24)$$

2-) A distribuição condicional da variância do ruído  $\sigma^2$  dados  $\mathbf{a}$ ,  $\mathbf{X}$  e  $\mathbf{Y}$ , é dada por:

$$p(\sigma^2 | \mathbf{a}, \mathbf{X}, \mathbf{Y}) \propto \chi^{-2} \left( v_0 + PM, \frac{v_0 \lambda + s^2}{v_0 + PM} \right) \quad (4.25)$$

ou

$$\frac{v_0 \lambda_0 + s^2}{\sigma^2} \sim \chi^2(v_0 + PM) \quad (4.26)$$

com

$$s^2 \equiv \sum_{i=0}^{M-1} \left\| \mathbf{r}(i) - \mathbf{S} \mathbf{A} \mathbf{x}(i) \right\|^2 \quad (4.27)$$

3-) As probabilidades condicionais de  $b_k(i) = \pm 1$ , dados  $\mathbf{a}$ ,  $\sigma^2$ ,  $\mathbf{X}_{ki}$  e  $\mathbf{Y}$ , são obtidas de:

$$\frac{p[b_k(i) = +1 | \mathbf{a}, \sigma^2, \mathbf{X}_{ki}, \mathbf{Y}]}{p[b_k(i) = -1 | \mathbf{a}, \sigma^2, \mathbf{X}_{ki}, \mathbf{Y}]} = \frac{\rho_k(i)}{1 - \rho_k(i)} \times \exp \left\{ \frac{2A_k}{\sigma^2} s_k^T [\mathbf{r}(i) - \mathbf{S} \mathbf{A} \mathbf{x}_k^0(i)] \right\} \quad (4.28)$$

$$k = 1, \dots, K; \quad i = 0, \dots, M - 1$$

onde

$$\mathbf{x}_k^0(i) \equiv [x_1(i), \dots, x_{k-1}(i), 0, x_{k+1}(i), \dots, x_K(i)]^T.$$

## 4.2.2 Detector Multiusuário Iterativo

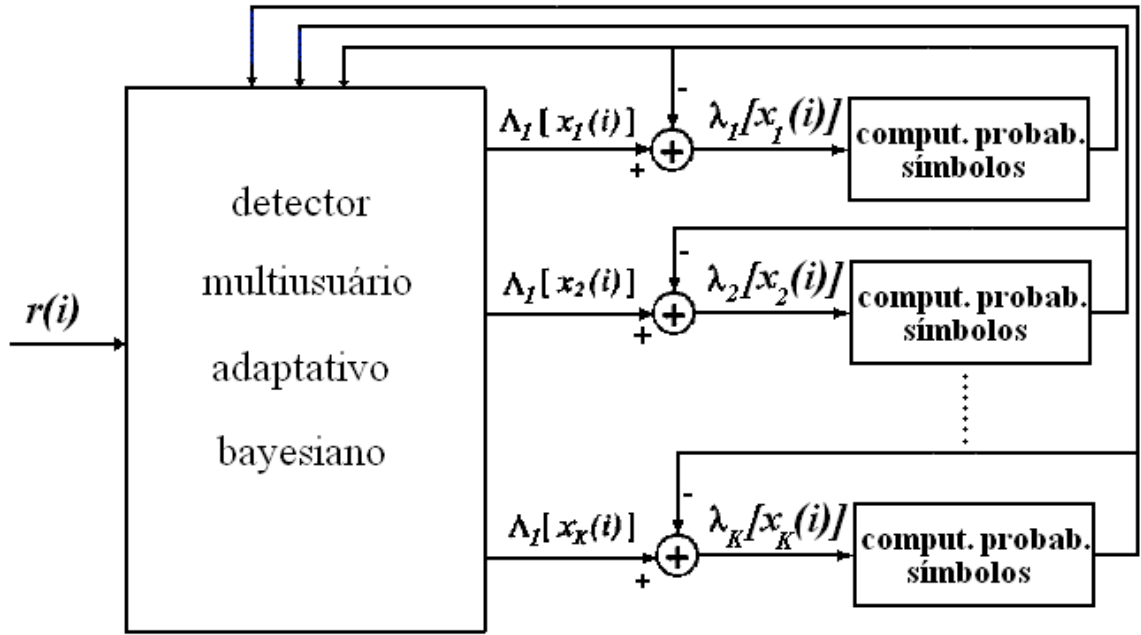
O trabalho foi desenvolvido criando-se simulações, utilizando o software R

$$\text{http} : // \text{cran.R} - \text{project.org}$$

como ferramenta básica, e anotando-se todos resultados para posterior análise. Nestas simulações o sistema foi dividido em blocos, e a cada bloco foi designada uma função. O sistema utilizado é um sistema de comunicação digital, que utiliza a técnica de espalhamento espectral para otimizar a transmissão do sinal, e além disso provê acesso múltiplo. No entanto, não há os blocos de codificação e decodificação de canal, que são típicos nestes sistemas. O porquê de não se utilizar um codificador e um decodificador de canal no sistema foi em razão de se optar por um sistema menos complexo, porém eficiente, que tornasse sua análise

e investigação mais fáceis, e também porque a falta de tais blocos não afeta o resultado final da recepção. E como o objetivo básico do projeto é melhorar a comunicação com o emprego de algoritmos estatísticos, além da utilização das transformadas *wavelets*, esses blocos podem ser posteriormente adicionados ao trabalho, sem prejuízo da análise atual. Pode-se estudar, por exemplo, condições menos favoráveis de propagação do sinal num canal com desvanescimento e multipercuro, e assim os blocos dispensados passariam a ser necessários. Decidiu-se também não envolver a análise de sincronismo no projeto, pois esta não está diretamente relacionada com o tema deste trabalho.

Na configuração adotada, o receptor utiliza probabilidades de símbolos *a priori*, e calculam-se as probabilidades *a posteriori*. O receptor executa um procedimento de retroalimentação, utilizando amostras de informações da saída, para refinar o processamento iterativo, Figura 4.2, dos resultados.



**Figura 4.2:** Processamento iterativo do receptor de detecção multiusuário.

O detector adaptativo bayesiano provê as probabilidades de símbolo *a posteriori*:

$$\left\{ p[b_k(i) = +1 | \mathbf{Y}] \right\}, \quad i = 0, \dots, M-1; \quad k = 1, \dots, K. \quad (4.29)$$

Com base nessas informações, primeiramente calcula-se a razão de verossimilhança *a posteriori* dos símbolos ‘+1’ ou ‘-1’, para se verificar qual tem maior probabilidade de ter sido enviado.

$$\Lambda_1[b_k(i)] \equiv \log \frac{p[b_k(i) = +1 | \mathbf{Y}]}{p[b_k(i) = -1 | \mathbf{Y}]} \quad k = 1, \dots, K; \quad i = 0, \dots, M-1 \quad (4.30)$$

Usando o Teorema de Bayes, Equação 2.64 do Capítulo 2, a Equação 4.30 pode ser escrita como:

$$\Lambda_1[b_k(i)] \equiv \log \frac{p[\mathbf{Y}|b_k(i) = +1]}{p[\mathbf{Y}|b_k(i) = -1]} + \log \frac{p[b_k(i) = +1]}{p[b_k(i) = -1]} \quad (4.31)$$

$$\Lambda_1[b_k(i)] \equiv \lambda_1[b_k(i)] + \lambda_2^p[b_k(i)] \quad (4.32)$$

O segundo termo na Equação 4.32 representa a razão de verossimilhança *a priori* do símbolo  $b_k(i)$ , que para o *loop* atual foi calculado pelo sistema na iteração anterior, e que foi realimentado para o detector bayesiano adaptativo. O primeiro termo,  $\lambda_1$ , da Equação 4.32 representa a informação denominada *extrínseca*, entregue pelo detector multiusuário bayesiano adaptativo, e baseada nos sinais recebidos  $\mathbf{Y}$ ; no sinal multiusuário dado pela Equação 4.1; e na informação *a priori* de todos os outros símbolos. A informação extrínseca é usada para computar as distribuições *a priori* simbólicas  $\{\rho_k(i)\}_{k=1; i=0}^{K; M-1}$ , definidas na Equação 4.5, das correspondentes razões de verossimilhanças, como nas equações a seguir. Reproducindo o segundo termo,  $\lambda_2^p$ , da Equação 4.32, tem-se que:

$$\lambda_2^p[b_k(i)] \equiv \log \frac{p[b_k(i) = +1]}{p[b_k(i) = -1]} \quad (4.33)$$

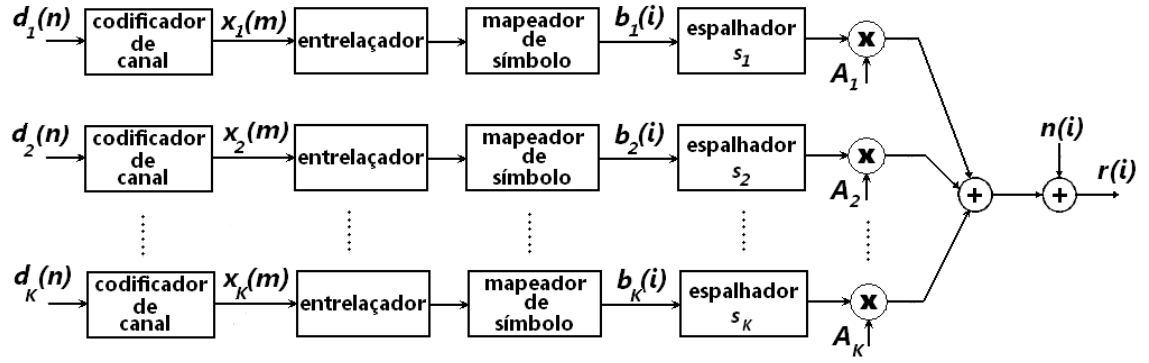
Após algumas manipulações, tem-se

$$\rho_k(i) \equiv p[b_k(i) = +1] = \frac{1}{2} \left[ 1 + \tanh \left( \frac{1}{2} \lambda_2^p[b_k(i)] \right) \right] \quad (4.34)$$

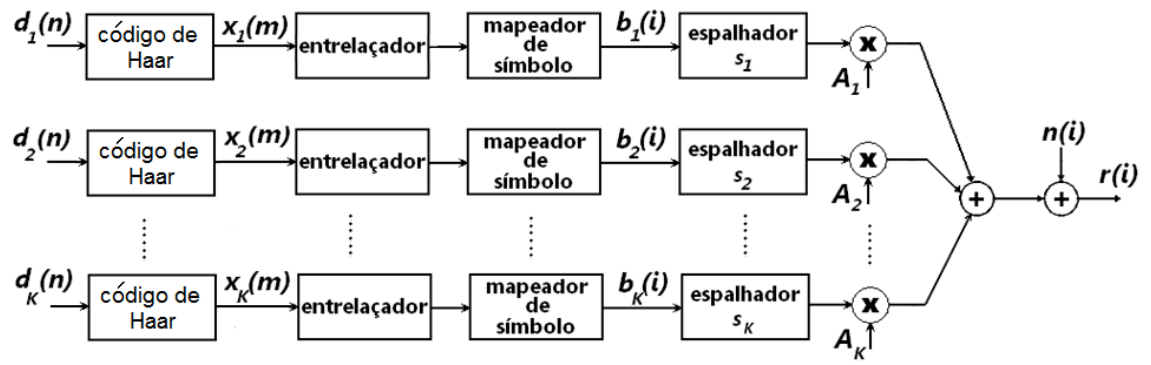
As probabilidades simbólicas  $\{\rho_k(i)\}_{k=1; i=0}^{K; M-1}$  são então realimentadas ao detector multiusuário bayesiano como informação *a priori* para a próxima iteração.

### 4.3 Processamento digital com modulação por *wavelets*

Para que o processamento por *wavelets* fosse efetivado no projeto, primeiramente foi necessário preparar a informação no transmissor através de uma modificação, conforme mostram as Figuras 4.3 e 4.4. Pode-se notar pela diferença entre as figuras que foi adicionado ao transmissor, no local reservado anteriormente para um código de canal, um “código de Haar”. Este código de Haar, que é uma sequência de +1’s e -1’s alternados, é utilizado para modular os bits  $d_k(i)$  que ficam com uma estrutura adequada para o *denoising*, ou cancelamento de ruído, no receptor.



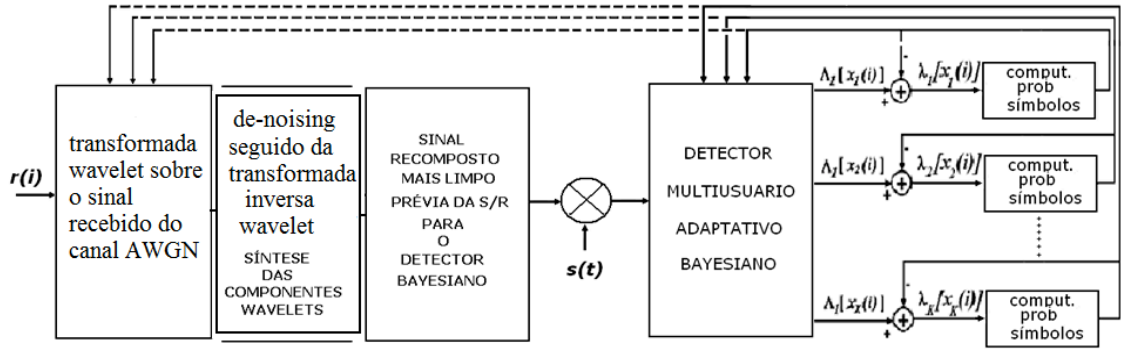
**Figura 4.3:** Transmissor de sistema de espalhamento espectral de acesso múltiplo.



**Figura 4.4:** Transmissor modificado: apresenta o código de Haar, que substitui o codificador de canal.

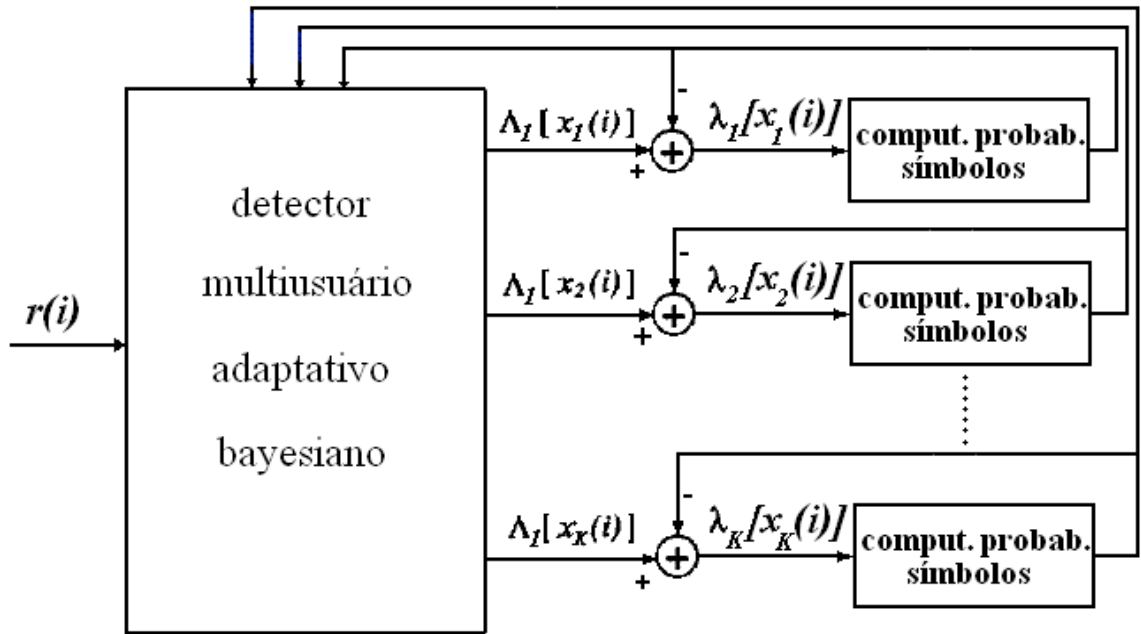
Com a aplicação da transformada *wavelet* sobre tais bits mascarados, que agora são denominados de símbolos, seus coeficientes *wavelets* passam a se concentrar em uma faixa do espectro específica, de nível zero, segundo a notação da teoria das *wavelets*. Existem outros níveis acima do zero, mas não haverá coeficientes *wavelets* da informação transmitida nesses níveis superiores. Esses níveis superiores de coeficientes são chamados de vãos.

Na Figura 4.5, é mostrado o receptor modificado, que apresenta o bloco da transformada *wavelet*, além do bloco de cancelamento de ruído, aqui denominado de *denoising*.



**Figura 4.5:** Receptor modificado, que apresenta o processamento por *wavelets*.

Esta nova versão do receptor pode ser comparada ao receptor com a configuração anterior, de detector multiusuário bayesiano puro, sem *wavelets*, conforme Figura 4.6, copiada aqui para facilitar a visualização.



**Figura 4.6:** Receptor com processamento iterativo.

No ambiente R, o software das simulações, a transformada *wavelet* é uma aplicação já pronta, que pode ser utilizada na forma de comando. Dada uma sequência de dados, que pode ser uma palavra código, a transformada *wavelet* desses símbolos é obtida por meio de um comando, cujo argumento é o conjunto dos símbolos.

Ex.: um certo código *codigoX* possui o seguinte formato:

$$\text{codigoX} = [+1, -1, +1, -1, +1, -1, +1, -1, +1, -1, +1, +1, +1, +1, -1, -1],$$

nas linhas a seguir encontra-se a sequência de passos para se obter os coeficientes *wavelets* deste *codigoX*:

```
> codigoX=c(+1, -1, +1, -1, +1, -1, +1, -1,...
... +1, -1, +1, +1, +1, +1, -1, -1)
> codigoX
[1] 1 -1 1 -1 1 -1 1 -1 1 -1 1 1 1 1 -1 -1
> dwt(codigoX,'haar') # comando da transformada wavelet do "codigoX"
$d1
[1] -1.414214 -1.414214 -1.414214 -1.414214...
... -1.414214 0.000000 0.000000 0.000000
$d2
[1] 0 0 1 -2
$d3
[1] 0.0000000 -0.7071068
$d4
[1] 0.5
$s4
[1] 0.5
attr(,"class")
[1] "dwt"
attr(,"wavelet")
[1] "haar"
attr(,"boundary")
[1] "periodic"
```

Os valores gerados a partir do comando são os coeficientes *wavelets*.

#### 4.3.1 Análise de *wavelets*: *denoising*

Nesta subseção será apresentada a técnica conhecida por várias denominações: limiarização ( "*thresholding*" ), encolhimento (*shrinkage*); ou o termo mais diretamente relacionado à área de engenharia de comunicações: *denoising*, ou cancelamento de ruído [5].

O princípio da *wavelet shrinkage* tem por objetivo a redução (ou completa remoção) do ruído presente em um sinal, diminuindo (ou zerando) a magnitude dos coeficientes *wavelets*. [59]

Considere o modelo

$$y_i = f(t_i) + n_i, \quad i = 1, 2, \dots, M \quad (4.35)$$

onde  $n_i \sim \mathcal{N}(0, \sigma^2)$  e  $t_i = (i-1)/M$ . No caso usual, supõe-se que  $f$  obedeça a um modelo particular, por exemplo, uma função linear de  $p$  parâmetros  $\theta_1, \theta_2, \dots, \theta_p$ . A estimação dos parâmetros é feita usando-se mínimos quadrados. Na situação do modelo da Equação 4.35, supõe-se que  $f$  pertença à determinada classe de funções, satisfazendo certas propriedades de regularidade. O enfoque tradicional usa estimadores baseados em núcleos e séries de Fourier.

Se  $f_i = f(t_i)$ , o objetivo é estimar  $\mathbf{f} = (f_1, f_2, \dots, f_M)^T$ , com o menor erro quadrático médio. O procedimento de *shrinkage* consiste nos três estágios seguintes:

- 1—) Tomar a transformada *wavelet* discreta de  $y_1, y_2, \dots, y_M$ , obtendo-se os  $M$  coeficientes *wavelets*  $w_{j,k}$ , que estão corrompidos por ruído.
- 2—) Usar limiares para reduzir ou anular aqueles coeficientes abaixo de certo valor (técnica usada em cancelamento de ruído de imagens).
- 3—) Tomar a transformada *wavelet* inversa dos coeficientes do item “2—)” para obter as estimativas  $\hat{f}_i$ , ou o sinal mais limpo.

A técnica adotada neste trabalho explora a possibilidade de trabalhar diretamente com os dados na fonte de informação, codificando-os e explorando essa codificação novamente na recepção. O procedimento de *denoising* utilizando o código de Haar e as transformadas *wavelets* foi originado devido à facilidade que se tem de acessar tanto o transmissor quanto o receptor.

### 4.3.2 Demonstração do algoritmo da transformada *wavelet* utilizado no detector multiusuário bayesiano

O objetivo desta subseção é mostrar que um sinal modulado pelo “código de Haar” e após, tendo sido obtidos seus coeficientes *wavelets*, pela aplicação da transformada discreta, a localização destes coeficientes limita-se à primeira escala dos coeficientes *wavelets* (as outras escalas, superiores, ficam vazias).

- 1—) Seja um sinal  $x(t)$ ,  $x(t) \in L^2(\mathbb{R})$ ; este sinal é então modulado pelo código de Haar =  $[+1, -1, +1, -1]$ , gerando um novo sinal:

$$x_0(t) = \psi_0 \otimes x(t); \quad (4.36)$$

Calculando-se os coeficientes *wavelets* de  $x_0(t)$ , obtém-se:

$$v_{0,k} \equiv \langle x_0(t), \phi_{0,k} \rangle \quad (4.37)$$

$$\omega_{j,k}(t) = \langle \psi_{j,k}(\cdot) | v_{0,n}(\cdot) \rangle \equiv \int_{-\infty}^{\infty} \psi_{j,k}(t) v_{0,n}(t) dt = \begin{cases} 1, & \text{se } j = 0 \text{ e } k = n \\ 0, & \text{qualquer outro caso} \end{cases} \quad (4.38)$$

Como as funções *wavelets* constituem uma base ortonormal, os coeficientes *wavelets*  $\omega$  somente serão diferentes de zero para a primeira escala, com  $j = 0$  e  $k = n$ . Isto é, o produto de cada *wavelet* com todas as outras no mesmo sistema diádico (ou seja, aquelas que são versões transladadas e/ou dilatadas uma das outras) é zero.



## 5 Comparação entre os algoritmos Amostrador de Gibbs e Metropolis-Hastings

Análise e resultados

## 5.1 Introdução

Neste capítulo são apresentados os algoritmos Amostrador de Gibbs e Metropolis-Hastings e explorados resultados obtidos por simulação para fim de comparação de desempenho dos mesmos. Estes algoritmos são utilizados para se efetivar a amostragem de densidades de probabilidade complexas. As amostras obtidas podem ser utilizadas na análise inferencial de características de uma população, sobre a qual diversas conclusões podem ser obtidas.

Para melhor compreender a construção destes algoritmos e o modo como eles podem ser utilizados em Inferência Bayesiana, há a necessidade de se rever alguns conceitos básicos das cadeias de Markov. As propriedades da cadeia de Markov serão apresentadas mais adiante no texto.

O desenvolvimento aqui apresentado está na forma de respostas a questões propostas, sendo que o objetivo final é estabelecer comparações entre os dois algoritmos e analisar sua eficiência, rapidez de convergência, complexidade, etc. As principais questões levantadas são:

- Quais são, e como podem ser apresentados, os resultados obtidos da aplicação do Amostrador de Gibbs e/ou Metropolis-Hastings a algumas distribuições de probabilidade?
- Como cada um dos algoritmos é controlado?
- Como poder-se-ia implementar uma análise de convergência?
- Quais são as justificativas convincentes das razões de escolha do Amostrador de Gibbs neste trabalho?
- Como chegar à expressão da complexidade do detector proposto, que utiliza o Amostrador de Gibbs?

## 5.2 Integração Monte Carlo clássica

Sejam  $X$  uma variável aleatória e  $x$  os valores possíveis de serem observados, isto é, observa-se  $X = x$ ; onde a variável aleatória  $X$  gera a observação  $x$ . Seja a função de probabilidade de  $X$  dada por  $f(x) = p\{X = x\}$ , e suponha que se queira calcular

$$\mathbb{E}[h(X)] = \int h(x)f(x)dx \quad (5.1)$$

para alguma função especificada  $h$ . O método Monte Carlo propõe utilizar uma amostra  $(X_1, \dots, X_m)$  gerada a partir da densidade  $f$ , para aproximar a Equação 5.1 pela média empírica

$$\bar{h}_m = \frac{1}{m} \sum_{j=1}^m h(x_j) \quad (5.2)$$

desde que  $\bar{h}_m$  convirja *quase certamente* para  $\mathbb{E}[h(X)]$  pela Lei Forte dos Grandes Números. Além disso, quando  $h^2$  tem uma esperança finita sob  $f$ , a velocidade de convergência de  $\bar{h}_m$  pode ser avaliada desde que a variância

$$\text{var}(\bar{h}_m) = \frac{1}{m} \int (h(x) - \mathbb{E}[h(X)])^2 f(x)dx \quad (5.3)$$

possa também ser estimada pela amostra  $(X_1, \dots, X_m)$  com

$$v_m = \frac{1}{m^2} \sum_{j=1}^m [h(x_j) - \bar{h}_m]^2. \quad (5.4)$$

Para  $m$  grande,

$$\frac{\bar{h}_m - \mathbb{E}[h(X)]}{\sqrt{v_m}} \quad (5.5)$$

é então distribuída aproximadamente como uma variável Normal com média zero e variância um,  $\mathcal{N}(0, 1)$ , isto leva à construção de um teste de convergência e de limites de confiança sobre a aproximação de  $\mathbb{E}[h(X)]$ .

## 5.3 Métodos de Monte Carlo via cadeias de Markov

São abordados nesta seção os métodos de Monte Carlo baseados na simulação de cadeias de Markov, cuja distribuição estacionária é a distribuição *a posteriori* de interesse, a partir da qual não é possível gerar-se diretamente uma amostra para o conjunto de inferências via Monte Carlo.

Na Seção 5.2, foi visto como um método de amostragem, com a propriedade de que a distribuição empírica da amostra que a gera aproxima-se da distribuição *a posteriori* de interesse, pode ser usado em inferência bayesiana. A aplicação

direta deste método não é, contudo, trivial pois simular uma distribuição *a posteriori* não é, em geral, uma tarefa fácil. Realmente, esta distribuição é em geral de natureza multivariada e extremamente complexa. Consequentemente, para uma utilização eficiente do método de Monte Carlo há necessidade de encontrar-se métodos indiretos de simulação de distribuições. Os métodos existentes são os métodos de Monte Carlo, que são muito úteis para simular sistemas complexos.

A idéia básica por trás desses métodos é transformar o problema estático em consideração em um problema de natureza dinâmica, construindo com este intuito um processo estocástico temporal, artificial, que seja fácil de simular, e que convirja, embora fracamente (ou lentamente), para a distribuição original. Tal processo temporal é, em geral, uma cadeia de Markov homogênea, cuja distribuição de equilíbrio é a distribuição que se pretende simular. Para se implementar este método há a necessidade de construir-se cadeias de Markov com distribuições de equilíbrio específicas. Metropolis *et al.* (1953) [60] desenvolveram um algoritmo para essa finalidade, o qual foi mais tarde generalizado por Hastings (1970) [27, 61]. A associação destes algoritmos para a simulação de distribuições com o método de Monte Carlo para a aproximação de integrais conduz aos chamados Métodos de Monte Carlo via cadeias de Markov (MCMC).

### 5.3.1 Processos de Markov [56]

Suponha que se queira gerar uma amostra de uma distribuição  $\pi(x)$  com  $x \in \Theta \subseteq \mathbb{R}^n$ ,  $\Theta$  é o espaço de estados, mas que não seja possível fazê-lo diretamente (ver Apêndice 1: Simulação de Distribuições Específicas, onde a geração de uma amostra diretamente é explicada). Admite-se, contudo, que seja possível construir-se uma cadeia de Markov com espaço de estados  $\Theta$ , cuja distribuição de equilíbrio é precisamente  $\pi(x)$ . Se em um experimento de período de tempo longo, para garantir-se que seja atingido o equilíbrio forem simulados estados da cadeia, estes estados podem vir a ser usados para se tirar conclusões sobre as características da distribuição  $\pi(x)$  de interesse. O problema resume-se então em como se construir uma cadeia de Markov com uma distribuição de equilíbrio específica. Considere  $n = 1$  e seja  $\Theta$  um conjunto finito ou infinito enumerável de estados  $x_1, x_2, \dots$ . Representa-se por  $X_n$  o estado  $s$  em que o sistema se encontra em certo instante. Seja  $X_0, X_1, \dots$  uma sequência de variáveis aleatórias que tomam valores em algum conjunto numerável  $\Theta$ , chamado de espaço de estados. Cada  $X_n$  é uma variável aleatória discreta que toma um dos  $N$  possíveis valores, onde  $N = |\Theta|$ ;

o caso  $N = \infty$  também é possível.

**Definição.** O processo  $X$  é uma cadeia de Markov se o mesmo satisfaz a condição de Markov:

$$p(X_n = s | X_0 = x_0, X_1 = x_1, \dots, X_{n-1} = x_{n-1}) = p(X_n = s | X_{n-1} = x_{n-1}), \quad (5.6)$$

para todo  $n \geq 1$  e todo  $s, x_1, \dots, x_{n-1} \in \Theta$ . Ou seja, o processo estocástico  $X$  é uma *cadeia de Markov* se a distribuição de  $X_n$  for independente de todos os estados prévios em que se encontrou a cadeia, com exceção do estado imediatamente anterior.

Assumiu-se que  $X$  toma valores de algum conjunto  $\Theta$  contável. A razão para isso é essencialmente a mesma que por que se tratam separadamente variáveis contínuas e discretas. Como  $\Theta$  é considerado contável, ele pode ser posto em uma correspondência de um para um com algum subconjunto  $\Theta'$  dos inteiros, e sem perda de generalidade pode-se assumir que  $\Theta$  é este conjunto  $\Theta'$  de inteiros. Se  $X_n = i$ , então diz-se que a cadeia está no “ $i$ -ésimo estado ao  $n$ -ésimo passo”.

A evolução de uma cadeia é descrita por suas “probabilidades de transição”

$$p(X_{n+1} = x_j | X_n = x_i) \quad (5.7)$$

**Definição.** A cadeia  $X$  é chamada *homogênea* se

$$p(X_{n+1} = x_j | X_n = x_i) = p(X_1 = x_j | X_0 = x_i) \quad (5.8)$$

para todo  $n, i$  e  $j$ . A *matriz de transição*  $\mathbf{P} = (p_{ij})$ , com elemento geral  $p_{ij}$ , é a matriz  $|\Theta| \times |\Theta|$  de *probabilidades de transição*

$$p_{ij} = p(X_{n+1} = x_j | X_n = x_i). \quad (5.9)$$

**Teorema 5.3.1.** A matriz de transição  $\mathbf{P}$  é uma matriz estocástica, que significa que:

- (a)  $\mathbf{P}$  tem entradas não negativas, ou  $p_{ij} \geq 0$  para todo  $i, j$ ;
- (b)  $\mathbf{P}$  tem somas de linhas iguais a um, ou  $\sum_j p_{ij} = 1$  para todo  $i$ .

**Definição.** A matriz de transição  $\mathbf{P}(m, m+n) = p_{ij}(m, m+n)$ , de passo  $\mathbf{n}$ , é a matriz de probabilidades de transição  $p_{ij}(m, m+n) = p(X_{m+n} = x_j | X_m = x_i)$  de passo  $\mathbf{n}$ .

Por suposição de homogeneidade,  $\mathbf{P}(m, m+1) = \mathbf{P}$ . A matriz  $\mathbf{P}(m, m+n)$  não

depende de  $m$  como consequência do seguinte fato importante.

**Teorema 5.3.2.** Equações de Chapman-Kolmogorov

$$p_{ij}(m, m+n+r) = \sum_k p_{ik}(m, m+n) p_{kj}(m+n, m+n+r). \quad (5.10)$$

Portanto,  $\mathbf{P}(m, m+n+r) = \mathbf{P}(m, m+n)\mathbf{P}(m+n, m+n+r)$ , e  $\mathbf{P}(m, m+n) = \mathbf{P}^n$ , a  $n$ -ésima potência de  $\mathbf{P}$ .

Pode-se escrever

$$p_{ij}^{(n)} = \mathbf{P}(X_{m+n} = x_j | X_m = x_i), \quad (5.11)$$

para designar as probabilidades de transição a  $n$ -passos, tem-se

$$p_{ij}^{(1)} = p_{ij}, \quad p_{ij}^{(n+1)} = \sum_k p_{ik}^{(n)} p_{kj}. \quad (5.12)$$

Se a matriz cujos elementos são as probabilidades de transição a  $n$  passos  $p_{ij}^{(n)}$  for representada por  $P^{(n)}$ , então pela Equação (5.12) tem-se a relação

$$\mathbf{P}^{(n)} = \mathbf{P}^n. \quad (5.13)$$

A matriz  $\mathbf{P}^{(n)}$  é também uma matriz estocástica. O estado  $x_j$  é acessível a partir do estado  $x_i$  se para algum  $n > 0$ ,  $p_{ij}^{(n)} > 0$ . Diz-se que dois estados  $x_i$  e  $x_j$  que são acessíveis um a partir do outro *comunicam-se* entre si. Uma cadeia de Markov é *irredutível* se todos os estados comunicam-se entre si. A comunicação é uma relação de equivalência e portanto numa cadeia de Markov irredutível todos os estados encontram-se na mesma *classe*.

Diz-se que o estado  $x_i$  tem *período*  $d$  se  $p_{ii}^{(n)} = 0$ , a menos que  $n$  seja divisível por  $d$  e este seja o maior inteiro com essa propriedade. Um estado diz-se *aperiódico* se tiver período  $d = 1$ . Uma cadeia de Markov *aperiódica* é uma cadeia de Markov em que todos os estados são aperiódicos. Dado que a *periodicidade* é uma propriedade de classe, para que uma cadeia de Markov irredutível seja aperiódica é suficiente que um dos seus estados seja aperiódico. [62]

Para quaisquer estados  $x_i$  e  $x_j$ , podem ser definidas as probabilidade de *primeira passagem*  $f_{ij}^{(n)}$  como

$$f_{ij}^{(n)} = p(X_t = x_j, X_{t-1} \neq x_j, \dots, X_{t-n+1} \neq x_j | X_{t-n} = x_i), \quad (5.14)$$

sendo os tempos médios de primeira passagem [63] (tempos médios de recorrência se  $x_i = x_j$ ) definidos por

$$m_{ij} = \sum_{n=1}^{\infty} n f_{ij}^{(n)} \quad (5.15)$$

desde que

$$\sum_{n=1}^{\infty} f_{ij}^{(n)} = 1 \quad (5.16)$$

Se  $\sum_{n=1}^{\infty} f_{ii}^{(n)} = 1$ , designa-se o estado  $x_i$  por *recorrente* que é positivo se então  $m_{ii} < \infty$  e nulo se  $m_{ii} = \infty$ . Um estado que não é recorrente é chamado de *transiente*. Se um estado  $x_j$  for transiente, então

$$\sum_{n=1}^{\infty} p_{ij}^{(n)} < \infty \quad \forall i, \quad (5.17)$$

isto é, partindo de qualquer  $x_i$ , o número esperado de transições para o estado  $x_i$  é finito. Assim, se  $x_i$  é transiente,  $p_{ij}^{(n)} \rightarrow 0$  com  $n \rightarrow \infty$ . A recorrência, tal como a periodicidade, é uma propriedade de classe. Designa-se por *ergódico* um estado recorrente positivo e aperiódico. Se  $x_i$  é um estado ergódico então tem-se que

$$\pi_j = \lim_{n \rightarrow \infty} p_{ij}^{(n)} = m_{jj}^{-1} (> 0) \quad (5.18)$$

Diz-se que, nestas condições,  $\pi_j$  é a distribuição limite, de equilíbrio, da cadeia.

Uma distribuição estacionária é uma coleção de números positivos  $\{p_j, j \geq 1\}$  cuja soma é 1, satisfazendo a equação de estacionariedade

$$p_j = \sum_{i=1}^{\infty} p_i p_{ij} \quad j \geq 1. \quad (5.19)$$

Pode-se mostrar que  $\{\pi_j, j = 1, 2, \dots\}$ , definida como em (5.18), é uma distribuição estacionária, invariante, da cadeia. [62]

Assim, quando se desejar “amostrar” uma distribuição que atribui ao estado  $x_j$  probabilidade  $\pi_j > 0$ , deve-se construir uma cadeia de Markov ergódica com matriz de transição  $\mathbf{P}$  tal que as equações de equilíbrio 5.19 sejam satisfeitas, isto é,

$$\pi^T = \pi^T \mathbf{P}, \quad (5.20)$$

onde  $\pi^T = \{\pi_1, \pi_2, \dots\}$ .

Uma outra propriedade importante é a propriedade de reversibilidade [63].

**Definição.** Seja  $X = \{X_n : 0 \leq n \leq N\}$  uma cadeia de Markov irredutível tal que  $X_n$  tem distribuição estacionária igual a  $\pi$  para todo  $n$ . A cadeia é chamada de **reversível** se as matrizes de transição de  $X$  e de  $Y$ , reversa de  $X$  no tempo, são as mesmas, o que significa

$$\pi_i p_{ij} = \pi_j p_{ji}, \quad \forall i, j, \quad (5.21)$$

As Equações 5.21 são chamadas de equações de equilíbrio e são fundamentais no estudo de cadeias reversíveis. Em geral, diz-se que uma matriz de transição  $\mathbf{P}$  e uma distribuição  $\lambda$  estão em equilíbrio se  $\lambda_i p_{ij} = \lambda_j p_{ji}$  para todo  $i, j \in \Theta$ .

Uma cadeia irredutível  $X$  que possui uma distribuição estacionária  $\pi$  é dita *reversível em equilíbrio* se sua matriz de transição  $\mathbf{P}$  está em equilíbrio com  $\pi$ .

**Teorema 5.3.3.** Seja  $\mathbf{P}$  a matriz de transição de uma cadeia irredutível  $X$  e suponha que existe uma distribuição  $\pi$  tal que  $\pi_i p_{ij} = \pi_j p_{ji}$  para todo  $i, j \in S$ . Então  $\pi$  é uma distribuição estacionária da cadeia. Além disso,  $X$  é reversível em equilíbrio.

**Prova.** Suponha que  $\pi$  satisfaz as condições do teorema. Então

$$\sum_i \pi_i p_{ij} = \sum_i \pi_j p_{ji} = \pi_j \sum_i p_{ji} = \pi_j$$

e então  $\pi = \pi \mathbf{P}$ , e assim  $\pi$  é estacionária. ■

Será apresentado a seguir como gerar uma cadeia de Markov com probabilidades estacionárias arbitrárias que podem ser especificadas somente até uma constante multiplicativa. Sejam  $b_j, j = 1, \dots$  números positivos cuja soma  $B = \sum_{j=1}^{\infty} b_j$  é finita. O algoritmo Metropolis-Hastings pode ser usado para gerar uma cadeia de Markov reversível no tempo, cujas probabilidades estacionárias são



$$\pi_j = b_j/B, \quad j = 1, \dots \quad (5.22)$$

Seja  $\mathbf{Q}$  uma matriz inteira de transição de probabilidades, irredutível de Markov, com  $q_{i,j}$  representando o elemento de  $\mathbf{Q}$  da linha  $i$  e coluna  $j$ . Define-se agora uma cadeia de Markov  $X_n$ ,  $n \geq 0$ , como mostrado a seguir. Quando  $X_n = i$ , gerar uma variável aleatória  $Y$  tal que  $p(Y = j) = q_{i,j}$ , para  $j = 1, 2, \dots$ . Se  $Y = j$ , então fazer  $X_{n+1}$  igual a  $j$  com probabilidade  $\alpha_{i,j}$ , e fazer também igual a  $i$  com probabilidade  $1 - \alpha_{i,j}$ . Sob essas condições, nota-se que a sequência de estados constitui uma cadeia de Markov com probabilidades de transição  $p_{i,j}$  dadas por

$$p_{i,j} = q_{i,j}\alpha_{i,j}, \quad \text{se } j \neq i \quad (5.23)$$

$$p_{i,i} = q_{i,i} + \sum_{k \neq i} q_{i,k}(1 - \alpha_{i,k}) \quad (5.24)$$

Esta cadeia de Markov será reversível no tempo e terá probabilidades estacionárias  $\pi_j$  se

$$\pi_i p_{i,j} = \pi_j p_{j,i} \quad \text{para } j \neq i \quad (5.25)$$

que é equivalente a

$$\pi_i q_{i,j} \alpha_{i,j} = \pi_j q_{j,i} \alpha_{j,i} \quad (5.26)$$

Mas se for tomado  $\pi_j = b_j/B$  e designado

$$\alpha_{i,j} = \min \left( \frac{\pi_j q_{j,i}}{\pi_i q_{i,j}}, 1 \right) \quad (5.27)$$

então a Equação 5.26 é satisfeita, basta fazer a substituição e verificar que a equação continua verdadeira. Pois, se

$$\alpha_{i,j} = \frac{\pi_j q_{j,i}}{\pi_i q_{i,j}} \quad (5.28)$$

então  $\alpha_{j,i} = 1$  e Equação 5.26 é verificada, e se  $\alpha_{i,j} = 1$  então

$$\alpha_{j,i} = \frac{\pi_i q_{i,j}}{\pi_j q_{j,i}} \quad (5.29)$$

e novamente a Equação 5.26 é verificada, portanto mostrando que a cadeia de Markov é reversível no tempo com probabilidades estacionárias  $\pi_j$ . Ainda, desde que  $\pi_j = b_j/B$ , vê-se da Equação 5.27 que

$$\alpha_{i,j} = \min \left( \frac{b_j q_{j,i}}{b_i q_{i,j}}, 1 \right) \quad (5.30)$$

que mostra que o valor de  $B$  não é necessário para definir-se a cadeia de Markov, porque os valores  $b_j$  já são suficientes.

## 5.4 Algoritmos Amostrador de Gibbs e Metropolis Hastings

### 5.4.1 Os algoritmos Metropolis

O objetivo principal que se tem em vista com a utilização dos algoritmos apresentados aqui é o de obter amostras de uma distribuição *a posteriori*  $h(\theta|x)$ . Os algoritmos Amostrador de Gibbs e Metropolis-Hastings permitem atingir este objetivo através da construção de uma cadeia de Markov  $\theta_t$  com espaço de estados geral  $\Theta \subset \mathbb{R}^k$  e distribuição de equilíbrio  $h(\theta|x)$ .

#### Espaço de estados discreto [64, 56]

Suponha que exista um sistema com estados  $x_j \in \Theta$ ,  $\Theta$  *discreto*, e que se deseja obter uma amostra de uma distribuição que atribui a cada estado  $x_j$  uma probabilidade  $\pi = p(X = x_j) > 0$ . Sabe-se que isso é possível somente se for encontrada uma cadeia de Markov ergódica com espaço de estados  $\Theta$  e probabilidade de transição  $p_{ij}$  satisfazendo as condições de reversibilidade

$$\pi_i p_{ij} = \pi_j p_{ji}, \quad \forall i \neq j, \quad (5.31)$$

Sendo  $\mathbf{Q}$  uma matriz de transição da cadeia de probabilidades de transição  $q_{ij}$ , considere que  $p_{ij}$  tem a forma

$$p_{ij} = q_{ij} \alpha_{ij}, \quad i \neq j \quad (5.32)$$

$$p_{ii} = 1 - \sum_{i \neq j} p_{ij}, \quad (5.33)$$

onde

$$\alpha_{ij} = \frac{s_{ij}}{1 + \frac{\pi_i q_{ij}}{\pi_j q_{ji}}}, \quad (5.34)$$

e  $s_{ij}$  é uma função simétrica de  $i$  e  $j$  escolhida de tal modo que  $0 \leq \alpha_{ij} \leq 1$  para todo  $i$  e  $j$ .

Para simular este processo executam-se, em cada instante  $t$ , os dois passos a seguir.

1. Suponha que no instante  $t$  a cadeia encontra-se no estado  $x_i$ , isto é,  $X_t = x_i$ . Selecionando-se um estado  $x_j$  usando a distribuição dada pela  $i$ -ésima linha de  $\mathbf{Q}$ , onde  $\mathbf{Q}$  é a matriz de transição da cadeia cujas probabilidades de transição são  $q_{ij}$ .
2. Esse estado é o da aceitação, com probabilidade  $\alpha_{ij}$ , como o novo estado da cadeia no instante  $t + 1$ , ou é rejeitado com probabilidade  $1 - \alpha_{ij}$ . Nesta situação a cadeia permanece no estado  $x_i$ .

A taxa de rejeição é dada pela proporção de instantes  $t$ , tal que  $X_{t+1} = X_t$ . Deve-se evitar escolher uma matriz de transição  $\mathbf{Q}$  que dê origem a uma elevada taxa de rejeição.

Há diversas propostas para escolher  $s_{ij}$  (ou o  $\alpha_{ij}$ ).

Se por exemplo  $\mathbf{Q}$  for escolhida simétrica e

$$s_{ij} = 1 + \min\left(\frac{\pi_i}{\pi_j}, \frac{\pi_j}{\pi_i}\right) \quad (5.35)$$

então

$$\alpha_{ij} = \min\left(1, \frac{\pi_j}{\pi_i}\right) \quad (5.36)$$

e o algoritmo resultante é o algoritmo proposto por Metropolis e colaboradores em 1953 [60].

A escolha de  $\alpha_{ij}$  na Equação 5.34 foi sugerida por Hastings em 1970 [61], sendo esta a razão do algoritmo chamar-se Metropolis-Hastings.

O algoritmo Amostrador de Gibbs é um caso muito especial do algoritmo Metropolis-Hastings, com probabilidade de rejeição nula. Ele é útil principalmente em situações em que  $\pi(x)$ , da Equação 5.31, é de natureza multivariada.

Os passos a seguir instruem como implementar o algoritmo Metropolis-Hastings.

1. Escolhe-se um valor inicial  $x$ ,  $x \in \mathbb{R}^n$ , e faz-se  $k = 1$ .
2. Sorteia-se  $y$ ,  $y \in \mathbb{R}^n$ , da distribuição de transição proposta  $q(x_k, y)$  e calcula-se a razão de aceitação  $\alpha$

$$\alpha(x_k, y) = \min \left( 1, \frac{\pi(y)q(y, x_k)}{\pi(x_k)q(x_k, y)} \right). \quad (5.37)$$

3. Obtém-se  $t$ ,  $t \in [0, 1]$ , a partir da densidade de probabilidade Uniforme.

4. Se  $\alpha(x_k, y) \geq t$ , faz-se  $x_{k+1} = y$ , caso contrário faz-se  $x_{k+1} = x_k$ . Quando  $k$  for igual ao tamanho amostral desejado,  $k = K$ , deve-se parar; caso contrário incrementa-se o valor de  $k$  em uma unidade,  $k \leftarrow k + 1$ , e volta-se ao passo 2.

**Exemplo de teste:** neste exemplo, testa-se o algoritmo Metropolis-Hastings com uma densidade de pequena dimensão. Considera-se uma densidade  $\eta \in \mathbb{R}^2$  definida como

$$\eta(x, y) \propto \exp(-x^2 - y^2) \quad (5.38)$$

a sequência de passos do Metropolis-Hastings é construída utilizando-se a distribuição proposta de passeio aleatório para explorar esta densidade. Define-se

$$q(x, y) \propto \exp \left( -\frac{1}{2\gamma^2} \|x - y\|^2 \right). \quad (5.39)$$

Em outras palavras, assume-se que o passo aleatório de  $x$  a  $y$  é distribuído como um ruído branco,

$$W = Y - X \quad \sim \mathcal{N}(0, \gamma^2) \quad (5.40)$$

Feita a escolha da distribuição de transição, pode-se utilizar os seguintes passos com sorteios e atualizações, em linguagem algorítmica.

Escolha um valor inicial  $x_1$ . Faça  $x = x_1$ .

*for* ( $k = 2 : K$ ) *do*

    Calcule  $\eta(x)$ .

    Sorteie  $\omega \sim \mathcal{N}(0, \gamma^2 I)$ , faça  $y = x + \omega$ .

    Calcule  $\eta(y)$ .

    Calcule  $\alpha(x, y) = \min \left( 1, \frac{\eta(y)}{\eta(x)} \right)$ .

    Sorteie  $u \sim \mathcal{U}([0, 1])$

*if* ( $u < \alpha(x, y)$ )

        Aceite: faça  $x = y, x_k = x$ ,

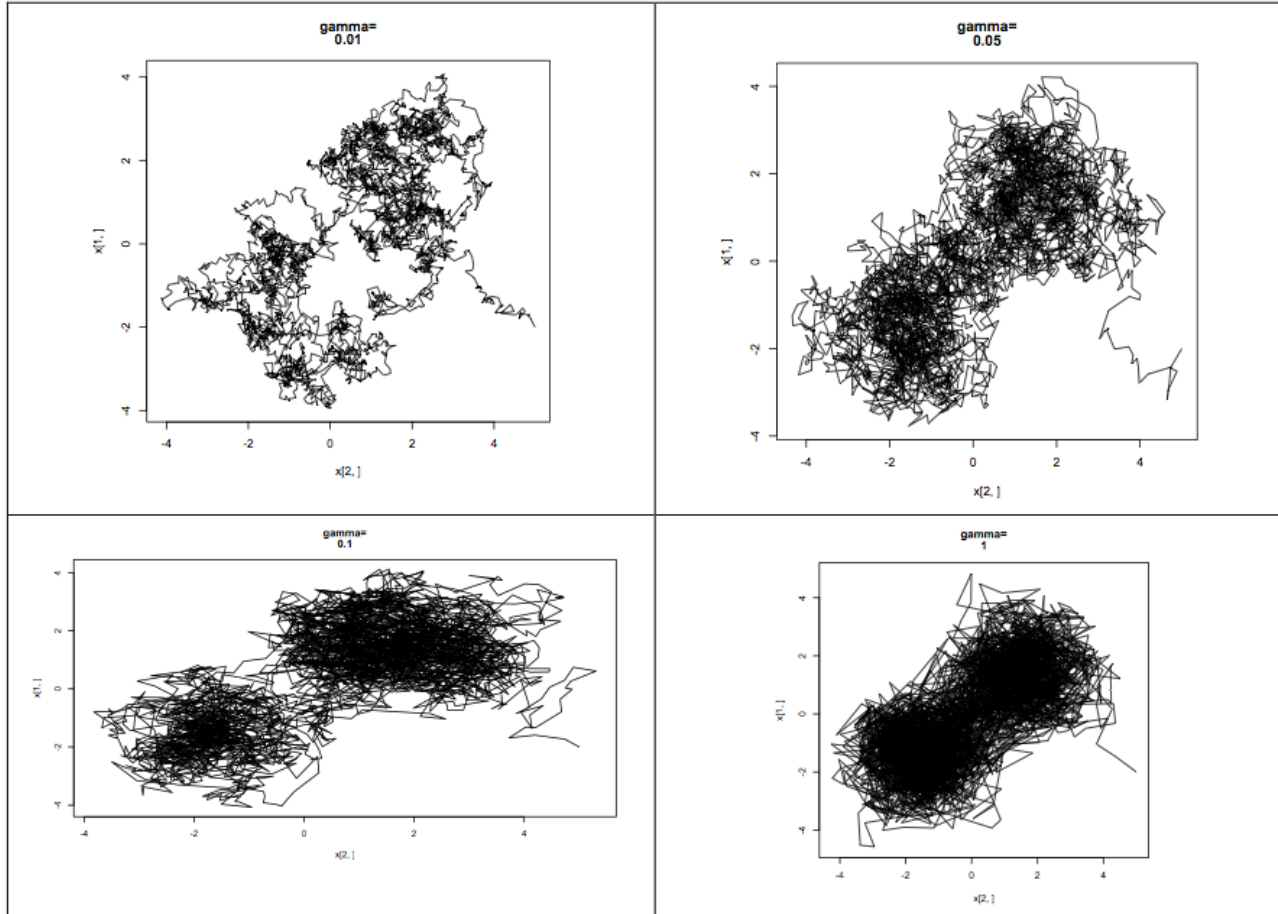
*else*

        Rejeite: faça  $x_k = x$ .

*end*

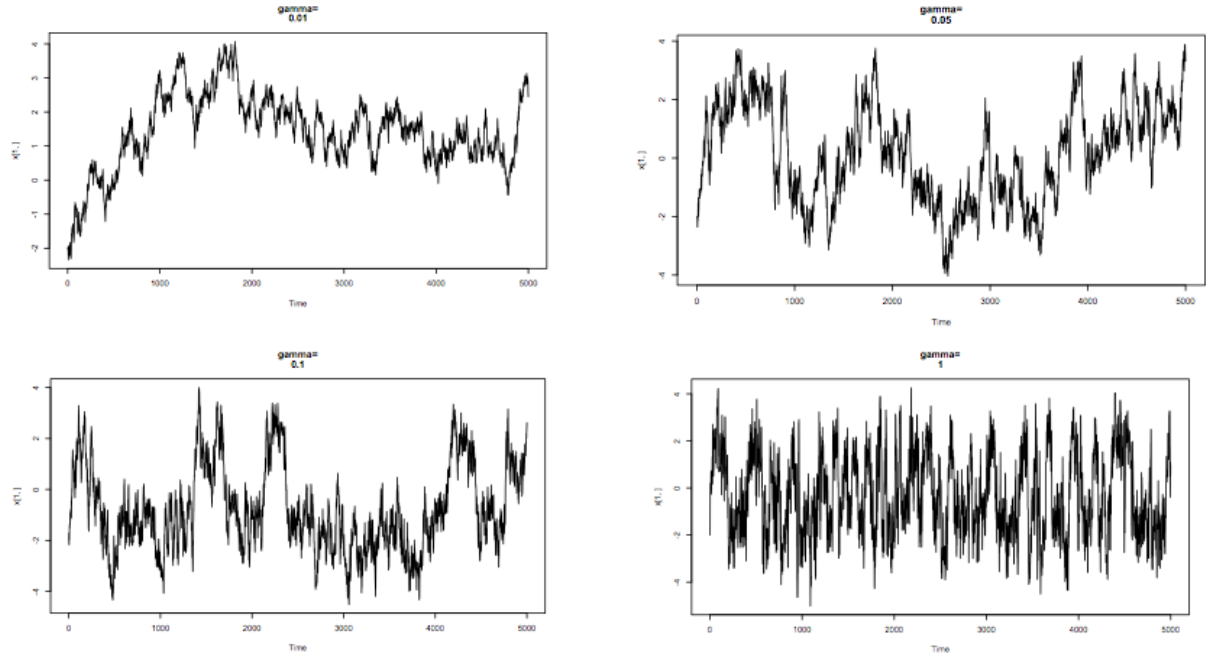
end

O controle do algoritmo Metropolis-Hastings é realizado por meio do parâmetro  $\gamma$ . O  $\gamma$  controla o comprimento do passo aleatório (*random step length*). Na verdade trata-se de um ajuste fino do algoritmo. Na Figura 5.1, observa-se a variação do comportamento das variáveis  $x$  e  $y$  dependentes do valor de  $\gamma$ , com valores 0,01; 0,05; 0,10 e 1,00.



**Figura 5.1:** Resultados das simulações do Metropolis-Hastings por passeio aleatório com diferentes tamanhos de passo (*step sizes*), (a)  $\gamma = 0,01$ , (b)  $\gamma = 0,05$ , (c)  $\gamma = 0,10$  e (d)  $\gamma = 1,00$ . O tamanho da amostra em cada figura é 5000.

A Figura 5.2 mostra os resultados de quatro processamentos do algoritmo com diferentes valores da constante  $\gamma$  controlando o comprimento do passo aleatório. O efeito do ajuste de  $\gamma$  pode também ser observado na Figura 5.1.



**Figura 5.2:** Diagnósticos de convergência para a componente  $x$  com os valores de  $\gamma$  como na Figura 5.1.

No caso do Amostrador de Gibbs, o passo aleatório não existe. Todos os sorteios são aceitos. Mesmo assim há ainda uma possibilidade de se controlar o Amostrador de Gibbs, pois ele é sensível ao desvio-padrão das densidades marginais condicionais que compõem as densidades a serem amostradas.

### 5.4.2 O algoritmo Amostrador de Gibbs em inferência bayesiana

Geman e Geman em 1984 [65] apresentaram o método de amostragem Amostrador de Gibbs como um algoritmo de simulação de distribuições multivariadas complexas e de dimensão elevada, que aparecem em problemas de reconstrução de imagens. Gelfand e Smith em 1990 [58] mostraram como o algoritmo pode ser usado para simular sobre distribuições *a posteriori*, como pode ser usado para resolver problemas em estatística bayesiana. O algoritmo é baseado no fato que se a distribuição conjunta  $h(\theta|x)$  for positiva em  $\Theta_1 \times \cdots \times \Theta_k$ , com  $\Theta_i$  sendo suporte da distribuição de  $\theta_i$  para  $i = 1, \dots, k$ , então ela é unicamente determinada pelas distribuições condicionais completas  $h(\theta_i|x, \theta(-i))$ ,  $i = 1, \dots, k$ , onde  $\theta = (\theta_1, \dots, \theta_k)$  e  $\theta(-i)$  representa o vetor  $\theta$  sem a  $i$ -ésima componente, isto é,  $\theta(-i) = (\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_k)$ . O algoritmo é portanto um esquema markoviano que requer a amostragem destas distribuições condicionais como se descrito a seguir.

### Descrição do algoritmo

Seja  $\theta^0 = (\theta_1^{(0)}, \dots, \theta_k^{(0)})$  um valor arbitrário inicial para o vetor  $\theta$ .

Procede-se iterativamente da seguinte forma:

obtem-se  $\theta_1^{(1)}$  de  $h(\theta_1|x, \theta_2^{(0)}, \dots, \theta_k^{(0)})$ ,  
 obtem-se  $\theta_2^{(1)}$  de  $h(\theta_2|x, \theta_1^{(1)}, \theta_3^{(0)}, \dots, \theta_k^{(0)})$   
 obtem-se  $\theta_3^{(1)}$  de  $h(\theta_3|x, \theta_1^{(1)}, \theta_2^{(1)}, \theta_4^{(0)}, \dots, \theta_k^{(0)})$   
 ...  
 obtem-se  $\theta_k^{(1)}$  de  $p(\theta_k|x, \theta_1^{(1)}, \dots, \theta_{k-1}^{(1)})$

Completa-se assim uma iteração do esquema e uma transição de  $\theta^0$  para  $\theta^1 = (\theta_1^{(1)}, \dots, \theta_k^{(1)})$ .

O esquema anterior é repetido com  $\theta^1$  anteriormente obtido, agora como vetor inicial, para obter um novo vetor  $\theta^2$  e assim haver uma transição de  $\theta^1$  para  $\theta^2 = (\theta_1^{(2)}, \dots, \theta_k^{(2)})$ .

Itera-se  $t$  vezes este ciclo de geração de observações aleatórias de cada uma das distribuições condicionais, produzindo assim  $\theta^0, \dots, \theta^t$ .

Numa linguagem menos técnica, o que se fez aqui foi amostrar a distribuição conjunta complexa, em geral multivariada, amostrando na verdade suas distribuições condicionais, e trabalhando com somente uma dimensão por vez (univariada). O procedimento toma cada uma das distribuições condicionais, uma por vez.

A sucessão  $\theta^0, \dots, \theta^t, \dots$  é uma realização de uma cadeia de Markov com espaço de estados  $\Theta$  e função de transição dada por [66]

$$p(\theta^t, \theta^{t+1}) = \prod_{i=1}^k h(\theta_i^{(t+1)}|x, \theta_j^{(t)}, j > i, \theta_j^{(t+1)}, j < i). \quad (5.41)$$

Os resultados apresentados anteriormente permitem concluir que, quando  $t \rightarrow \infty$ , o vetor  $(\theta_1^{(t)}, \dots, \theta_k^{(t)})$  tende em distribuição para um vetor aleatório cuja função densidade de probabilidade conjunta é  $h(\theta|x)$ . Em particular,  $\theta_i^{(t)}$  converge em distribuição para uma quantidade aleatória cuja densidade é  $h(\theta_i|x)$  (densidade marginal *a posteriori* de  $\theta_i$ ) e

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N h(\theta_i) \rightarrow \mathbb{E}[h(\theta)|x] \quad \text{quase certamente,} \quad (5.42)$$

para qualquer função  $h(\cdot)$ , onde  $\mathbb{E}[h(\theta)|x]$  representa o valor esperado em relação à distribuição *a posteriori*  $h(\theta|x)$ .

**Passos do algoritmo:** Os passos a seguir instruem como implementar o algoritmo Amostrador de Gibbs para uma função de duas variáveis.

Selecione um valor inicial  $x_1$ . Faça  $x = x_1$ .

laço: para  $(j = 2 : N)$  faça {

Calcule  $\Phi_1(t) = \int_{-\infty}^t \eta(x^1, x^2) dx^1$

Sorteie  $u \sim \mathcal{U}([0, 1])$ , faça  $x^1 = \Phi_1^{-1}(u)$

Calcule  $\Phi_2(t) = \int_{-\infty}^t \eta(x^1, x^2) dx^2$

Sorteie  $u \sim \mathcal{U}([0, 1])$ , faça  $x^2 = \Phi_2^{-1}(u)$

Faça  $x_k = x$  }

*fm*

### 5.4.3 Análise de Convergência

A análise de convergência dos algoritmos Metropolis-Hastings e Amostrador de Gibbs ainda é um tema sob pesquisa. A investigação neste campo tem sido bastante intensa, e muitas estratégias já foram propostas para abordar o problema da avaliação de convergência.

O parâmetro  $\gamma$ , como foi visto anteriormente, influencia diretamente a convergência no Metropolis-Hastings, e portanto deve-se tomar cuidado especial quando da escolha de seu valor. No Amostrador de Gibbs este parâmetro não existe.

Os problemas teóricos associados à avaliação da convergência desses algoritmos estão relacionados com a convergência dos Métodos de Monte Carlo via cadeias de Markov (MCMC) e, mais especificamente, com a convergência de cadeias de Markov irredutíveis. [64]

Algumas estratégias para se averiguar a convergência são:

- 1—) Métodos *ad-hoc* baseados em representações gráficas das estimativas das densidades.
- 2—) Monitoração das médias ergódicas.
- 3—) Método de Gelman e Rubin: sugerem uma abordagem para averiguar a convergência do Algoritmo de Gibbs, usando componentes da variância de sequências



múltiplas da cadeia, simuladas a partir de uma variedade de pontos iniciais dispersos.

Nos testes de convergência deve-se olhar para o *trace* da série temporal, para o gráfico das variáveis aleatórias geradas *versus* o número de iterações. Além de colocar em evidência um possível distanciamento entre as sequências ou funções, tais *traces* podem também mostrar o período de *burnin*. Outros gráficos importantes são o da autocorrelação, que pode mostrar em que classe a série temporal analisada se insere ou caracterizar a dependência entre as amostras, e o gráfico dos quartis.

Um procedimento simples que pode ser empregado para se verificar a convergência é o método de verificação do erro entre duas amostras sucessivas. Tomam-se duas amostras consecutivas e calcula-se a diferença entre elas; quando o valor desta diferença permanecer aproximadamente constante dentro de um limite pré-estabelecido, e ainda repetir-se para as amostras posteriores, ter-se-á um bom indicador de que já se saiu do período de *burnin*, e que a convergência foi atingida, com uma boa probabilidade.

#### 5.4.4 Motivos da escolha do Amostrador de Gibbs

O Amostrador de Gibbs é utilizado diretamente na amostragem de densidades condicionais, que é a forma com se apresentam as densidades de probabilidade na Estatística bayesiana (Teorema de Bayes); portanto o Amostrador de Gibbs pode ser aplicado diretamente ao modelo de detecção multiusuário.

Como foi visto na seção do Amostrador de Gibbs, existe um teorema que garante a unicidade entre uma densidade conjunta multivariada e suas densidades condicionais. Existem casos para os quais torna-se muito difícil obter-se tais distribuições condicionais a partir da densidade conjunta original, mas para esses casos pode-se utilizar o algoritmo de Metropolis-Hastings no lugar do Amostrador de Gibbs. Este, no entanto, não se trata de um problema do campo da utilização do Amostrador de Gibbs ou do Metropolis, mas sim de um problema que pertence à própria inferência bayesiana: o problema de se conseguir densidades casadas para facilitar os cálculos e as análises. [64] O Amostrador de Gibbs é apenas um pouco mais lento que o Metropolis-Hastings, entretanto, em geral, ele produz cadeias com as menores correlações possíveis dentre todos os outros algoritmos da classe MCMC [67]. Se fosse utilizado o Metropolis-Hastings, por exemplo, para o sistema sob investigação, talvez houvesse a necessidade de se descartar algumas amostras sucessivas, como forma de melhorar a correlação en-

tre as amostras resultantes. O Amostrador de Gibbs, como se sabe, é derivado do Metropolis-Hastings, e é conceitualmente mais simples do que este. Ele não necessita da determinação de uma probabilidade de transição para funcionar.

#### 5.4.5 Análise de Complexidade (*time bounds*)

A medida de desempenho de um algoritmo que é, em geral, mais amplamente aceita é o tempo que ele gasta para produzir a resposta final. Esse intervalo de tempo pode variar consideravelmente de um computador para outro, devido a diferenças desde a velocidade, o número de núcleos de computação, até o repertório de instruções.

A complexidade computacional de qualquer detector pode ser quantificada por sua complexidade no tempo por bit (TCB, *time complexity per bit*), isto é, o número de operações requeridas pelo detector para demodular a informação transmitida, dividido pelo número total de bits demodulados. Uma TCB da função  $f(K)$  é escrita como  $O(g(k))$ , se existir uma constante  $c > 0$  tal que para valores de  $K$  suficientemente grandes,  $f(K) \leq c g(K)$ . A complexidade do detector multiusuário bayesiano é analisada pelos cálculos executados por iteração. O cálculo da Equação 5.45 insere um fator de complexidade igual a  $O(K^2)$  [68], pois há uma inversão de matrizes. A complexidade total é:

$$O[K^2 + KM] \quad (5.43)$$

isto é, existe um termo quadrático com relação ao número de usuários [devido à inversão da matriz simétrica definida positiva na Equação 5.45] e mais um termo linear com relação ao tamanho do bloco dos símbolos (*frame* ou *quadro*). Portanto tem-se aqui este termo que se torna dominante em relação aos outros na expressão geral do detector multiusuário, e cujo valor é igual a  $K^2$ . Assim, a complexidade é dada por  $O(K^2 + KM)$ . Se também o *burnin* fosse considerado no cálculo geral, acrescentar-se-ia à fórmula o fator  $(n_0 + N)$ .

Para se desenvolver a fórmula que caracteriza a complexidade do detector, são utilizadas expressões conhecidas de complexidades de operações básicas [69, 70]: por exemplo, para se calcular a inversa de uma matriz  $\mathbb{N}$ , resolvendo  $\mathbb{N} \mathbf{x} = \mathbf{e}_i$ , para  $i = 1, 2, \dots, n$ , pode-se fazer isso em  $O(n^3)$  passos usando eliminação gaussiana; multiplicação de matrizes,  $O(n^3)$ ; inversão de matrizes simétricas positivas,  $O(n^2)$ ; operação linear,  $O(n)$ ; etc. Para o detector multiusuário bayesiano o número total de usuários é  $K$ , e cada usuário gera um quadro de  $M$  símbolos; portanto existe um processamento total para  $K \times M$ . O cálculo é executado para

uma única iteração, e em cada iteração é feita uma inversão da matriz simétrica da distribuição Normal: o detector multiusuário que utiliza o Amostrador de Gibbs calcula a densidade posterior condicional para determinar o valor da amplitude:

$$p(\mathbf{a}|\sigma^2, \mathbf{X}, \mathbf{Y}) \propto \mathcal{N}(\mathbf{a}_*, \Sigma_*) \quad (5.44)$$

onde  $\Sigma_*$  é dada por

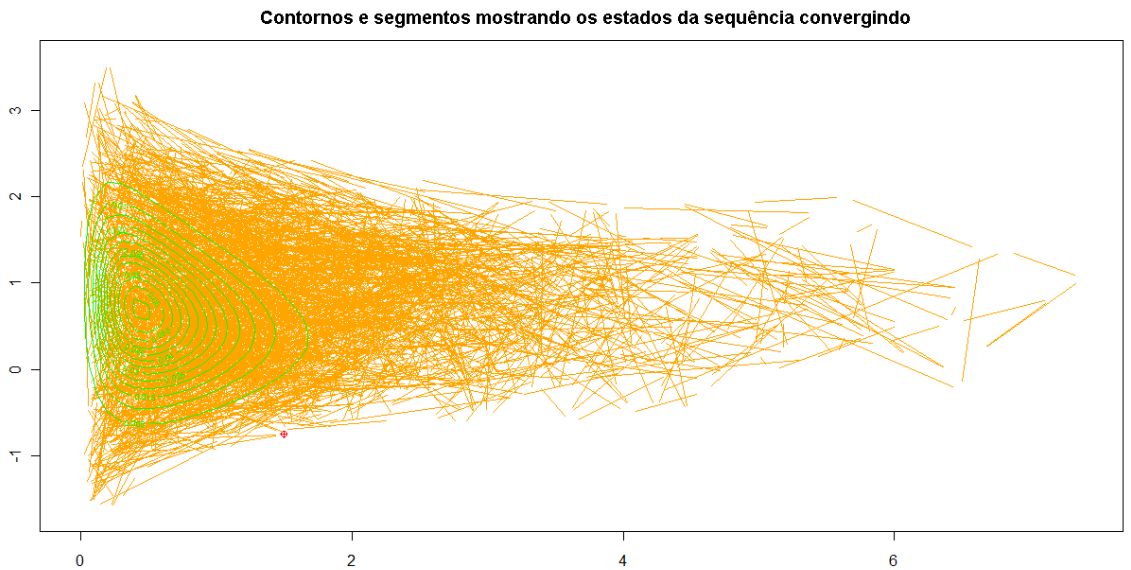
$$\Sigma_*^{-1} \triangleq \Sigma_0^{-1} + \frac{1}{\sigma^2} \sum_{i=0}^{M-1} \mathbf{B}(i) \mathbf{R} \mathbf{B}(i) \quad (5.45)$$

## 5.5 Testes dos algoritmos e resultados

Com o objetivo de se estabelecer uma referência para ser utilizada na verificação das características do Amostrador de Gibbs, o algoritmo Metropolis-Hastings foi testado sobre uma função densidade de probabilidade de dimensão dois. Seja a densidade  $\lambda \in \mathbb{R}^2$  definida por

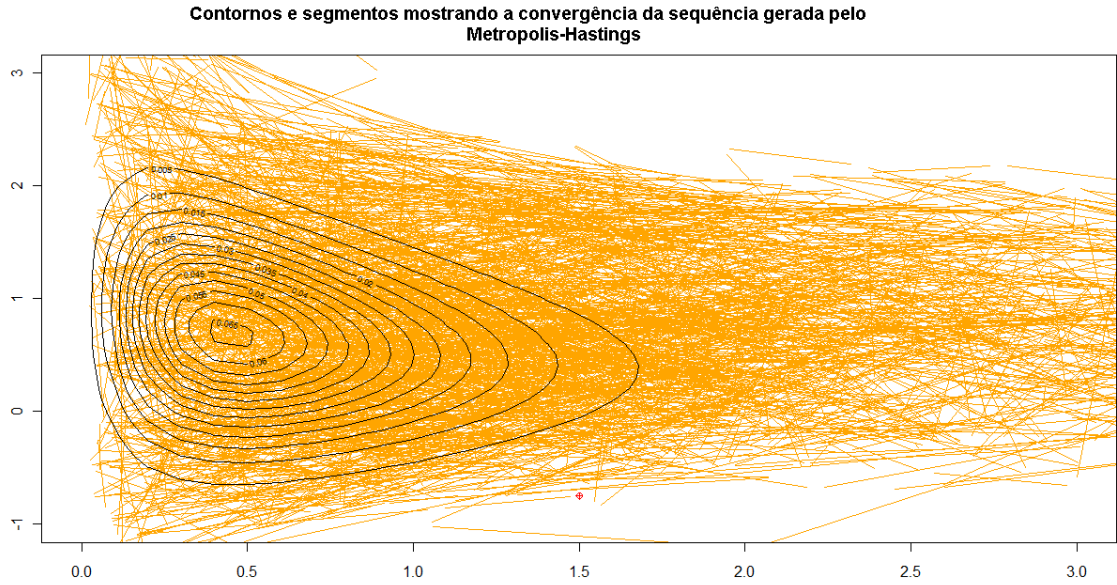
$$\lambda(x, y) \propto x^2 \exp(-xy^2 - y^2 + 2y - 4x) \quad (5.46)$$

o gráfico desta densidade é visto na Figura 5.3. A sequência gerada com o Metropolis-Hastings para explorar esta densidade foi construída usando a distribuição de transição  $q(x, y)$ .



**Figura 5.3:** Gráfico da densidade dada pela Eq. 5.46, amostrada pelo Metropolis-Hastings.

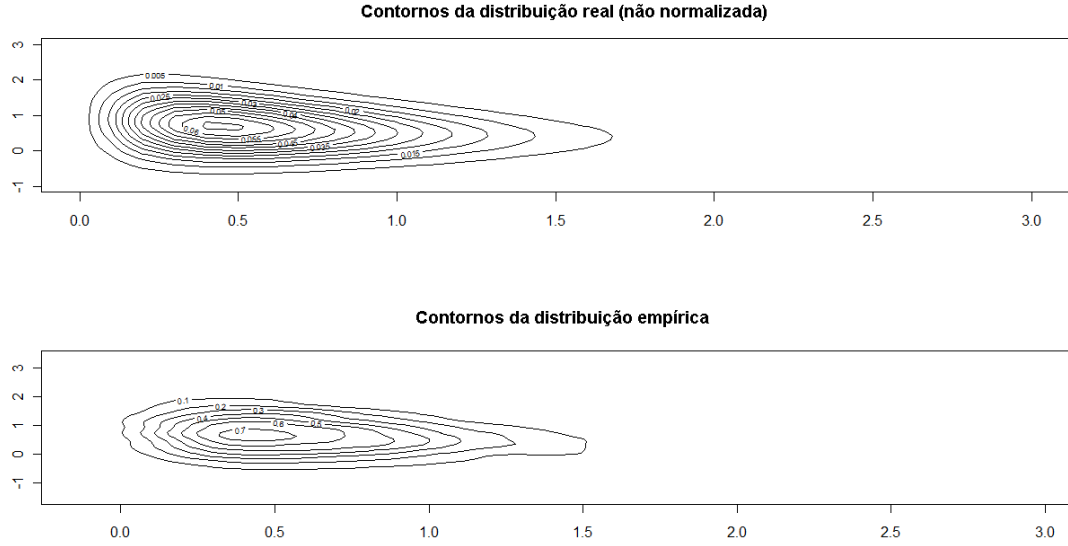
O gráfico apresentado na Figura 5.3 foi reproduzido na Figura 5.4, onde uma ampliação foi implementada para se melhor visualizar os detalhes da imagem da função; da mesma forma, a ampliação facilita a comparação com os gráficos da Figura 5.5, que são apresentados a seguir.



**Figura 5.4:** Gráfico da densidade dada pela Eq. 5.46, amostrada pelo Metropolis-Hastings, com detalhes da visualização.

Utilizando o Amostrador de Gibbs, outras simulações foram feitas e novas amostras foram obtidas a partir da mesma Equação 5.46. Para isso, houve a necessidade de se transformar a densidade conjunta em densidades condicionais. A Equação 5.46 é uma densidade conjunta em  $x$  e  $y$  e pode ser decomposta em duas densidades marginais, uma densidade Normal e uma densidade conhecida por Gama Inversa; estas densidades fazem parte de uma mesma família, e por isso são chamadas de densidades conjugadas. Quando se está trabalhando com uma densidade Normal de variância desconhecida, pesquisadores preferem trabalhar com a inversa da variância como sendo o parâmetro investigado, denominada de precisão. Neste caso, a Gama Inversa referir-se-á à precisão [27, 71, 28]. Assim, para se verificar o comportamento do algoritmo Amostrador de Gibbs, o mesmo foi também aplicado à Equação 5.46, e o resultado obtido pode ser visto na Figura 5.5.

Comparando-se a Figura 5.4 com a Figura 5.5, nota-se visualmente que os gráficos são praticamente iguais, os contornos por exemplo possuem medidas muito próximas, os diâmetros também têm desvios muito pequenos, em média eles são iguais; as densidades foram obtidas corretamente através das simulações



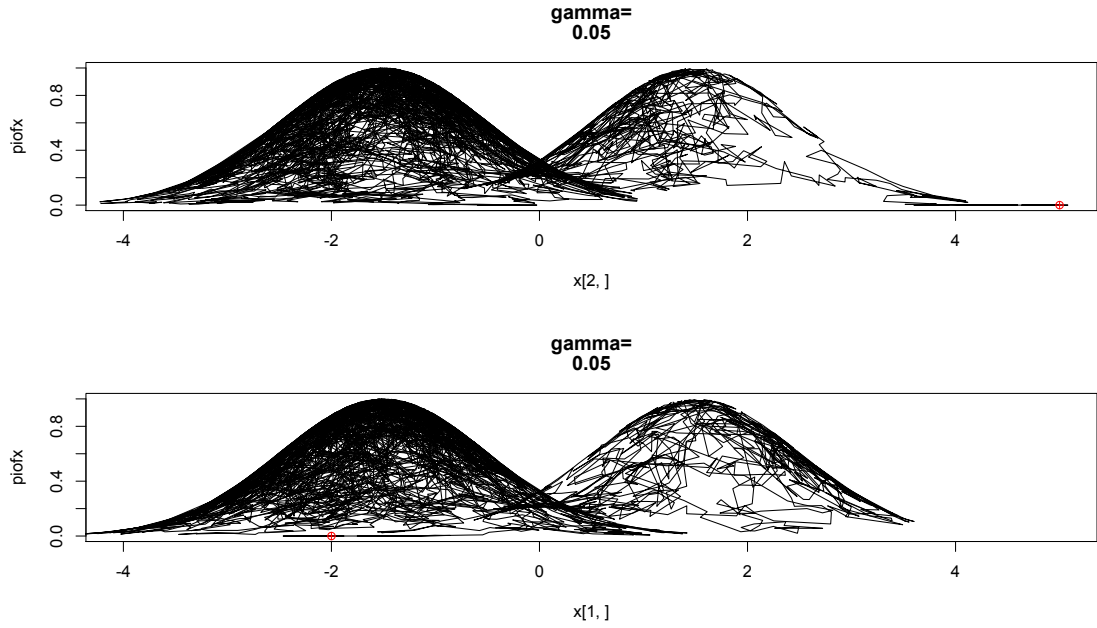
**Figura 5.5:** Gráfico de densidades obtidas da Eq. 5.46; a distribuição empírica foi amostrada com o Amostrador de Gibbs.

o que mostra que o Amostrador de Gibbs foi tão eficaz quanto o Metropolis-Hastings, com relação à amostragem da distribuição.

Prosseguindo com os testes, a próxima função densidade de probabilidade conjunta verificada foi a Normal bimodal de dimensão dois, dada pela Equação 5.47.

$$f(x, y) \propto \exp(-0.5(x + 1.5)^2 - 0.5(y + 1.5)^2) + \exp(-0.5(x - 1.5)^2 - 0.5(y - 1.5)^2) \quad (5.47)$$

Na Figura 5.6, observam-se os perfis da normal bimodal amostrada com o algoritmo Metropolis-Hastings.



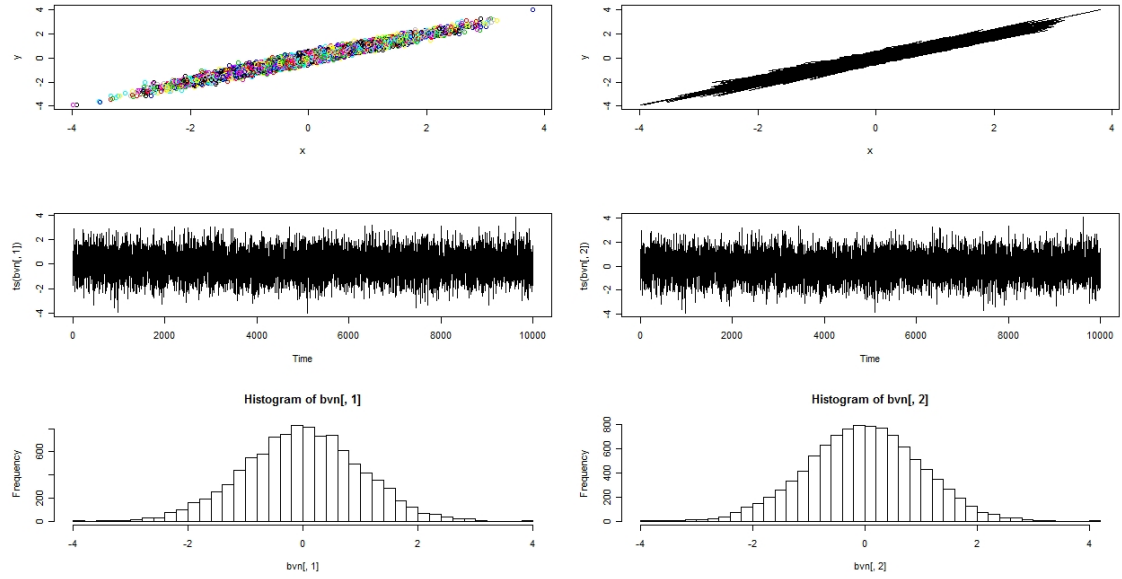
**Figura 5.6:** Gráfico da densidade dada pela Eq. 5.47 amostrada pelo Metropolis-Hastings.

Observando-se o gráfico do perfil da bimodal, nota-se que uma das superfícies, relativa à uma das médias, não foi formada completamente. A explicação para esse fenômeno é que a sequência da cadeia de Markov originada não obteve muitos valores computados para esta média; porém, se o tempo de simulação aumentasse o resultado mudaria.

Outra densidade analisada foi a Normal bivariada, amostrada utilizando-se o Amostrador de Gibbs e dada pela Equação 5.48.

$$f(x, y) \propto \exp(-x^2 - y^2) \quad (5.48)$$

O gráfico que mostra a saída gerada pelo Amostrador de Gibbs pode ser visto na Figura 5.7. Na parte superior do gráfico, vê-se a imagem da função gerada aleatoriamente pelo algoritmo; nota-se que a correlação entre as variáveis é próxima da unidade, e assim o gráfico tem um formato linear positivo. Na mesma figura, na linha central, encontram-se os valores simulados das duas variáveis independentes da função,  $x$  e  $y$ , e que representam as duas densidades marginais com forte correlação entre si. Na última linha do gráfico da Figura 5.7 é visto o histograma, cuja linha superior de contorno indica que as densidades marginais são realmente densidades Normais.

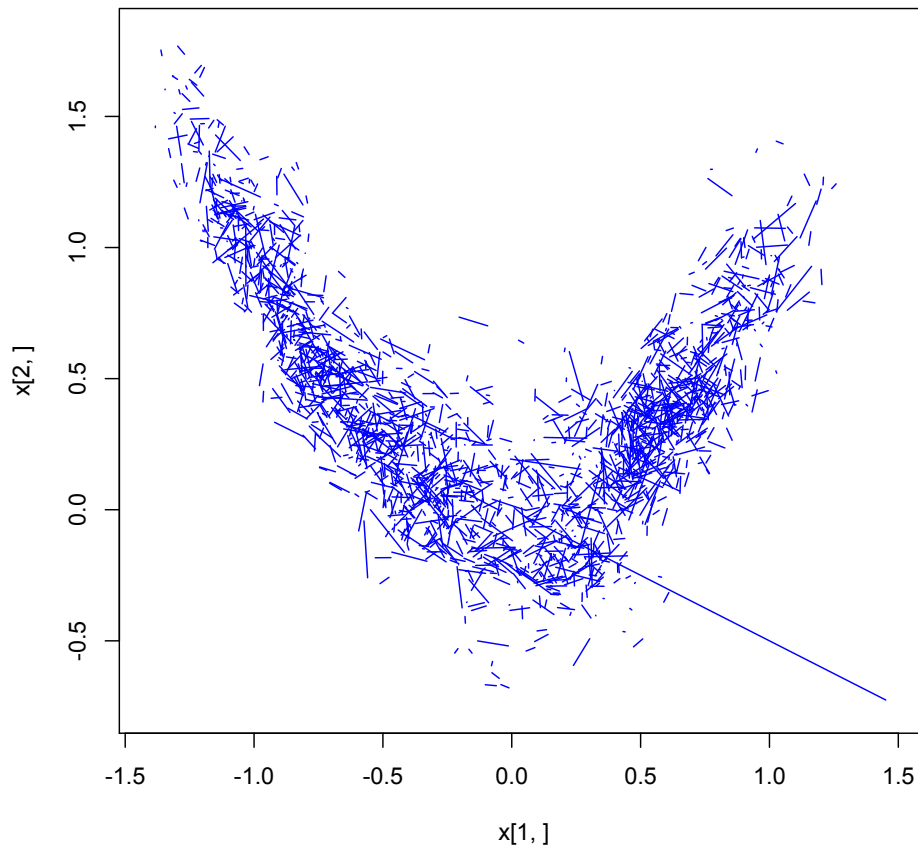


**Figura 5.7:** Gráfico da densidade dada pela Eq. 5.48 amostrada pelo Amostrador de Gibbs.

Na sequência, continuando com a lista de algumas densidades importantes, amostradas com o algoritmo Metropolis-Hastings e com o Amostrador de Gibbs, uma densidade exponencial de quarta potência, nas variáveis  $x$  e  $y$ , é vista na Equação 5.49.

$$h(x, y) \propto \exp[-10(x^2 - y)^2 - (y - 1/4)^4] \quad (5.49)$$

O gráfico que representa esta densidade, obtido com os resultados da simulação do algoritmo Metropolis-Hastings sobre a exponencial, é apresentado na Figura 5.8. Pode-se observar a rápida convergência do algoritmo e a representação por meio de uma quantidade maior de pontos sobre a região de mais alta densidade.

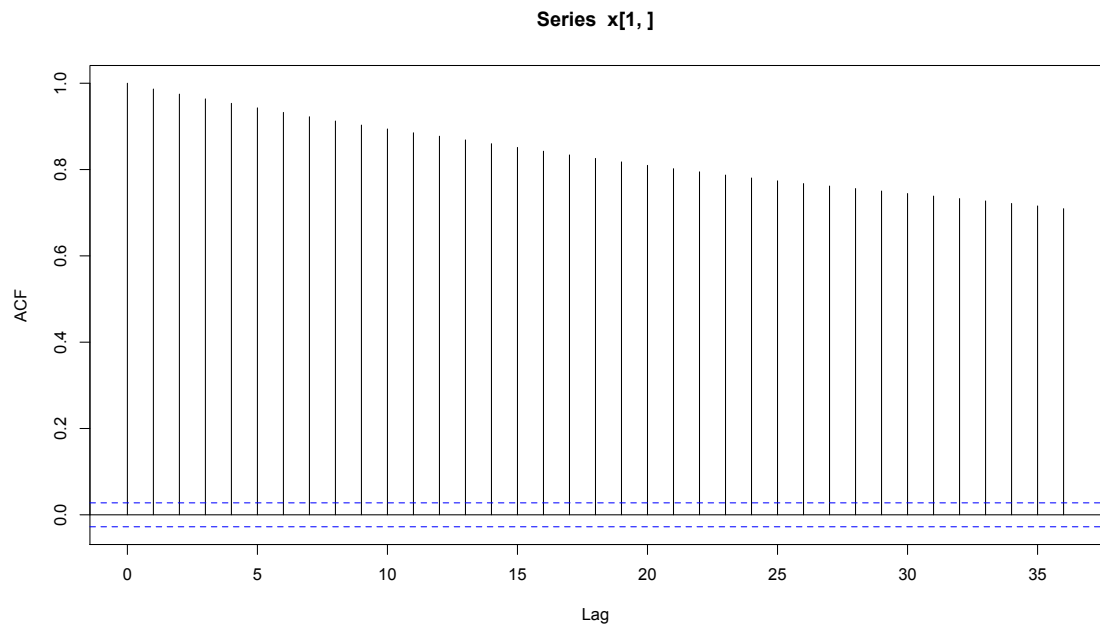


**Figura 5.8:** Gráfico da densidade dada pela Equação 5.49 amostrada pelo Metropolis-Hastings.

Uma função muito importante de análise estatística é a função de correlação. Ela pode ser usada para indicar, ou quantificar, a dependência entre as várias variáveis ou amostras de um certo sistema. A função de autocorrelação é utilizada para se analisar em uma sequência de variáveis por exemplo qual a relação entre uma parte da sequência com a(s) outra(s) parte(s) da mesma sequência.

No gráfico da Figura 5.9 observa-se a dependência entre os vários valores assumidos pela variável aleatória  $X_t$ . Finalizando a lista de densidades abordadas nesta seção, serão apresentadas a seguir as densidades de Rayleigh e de Pareto.



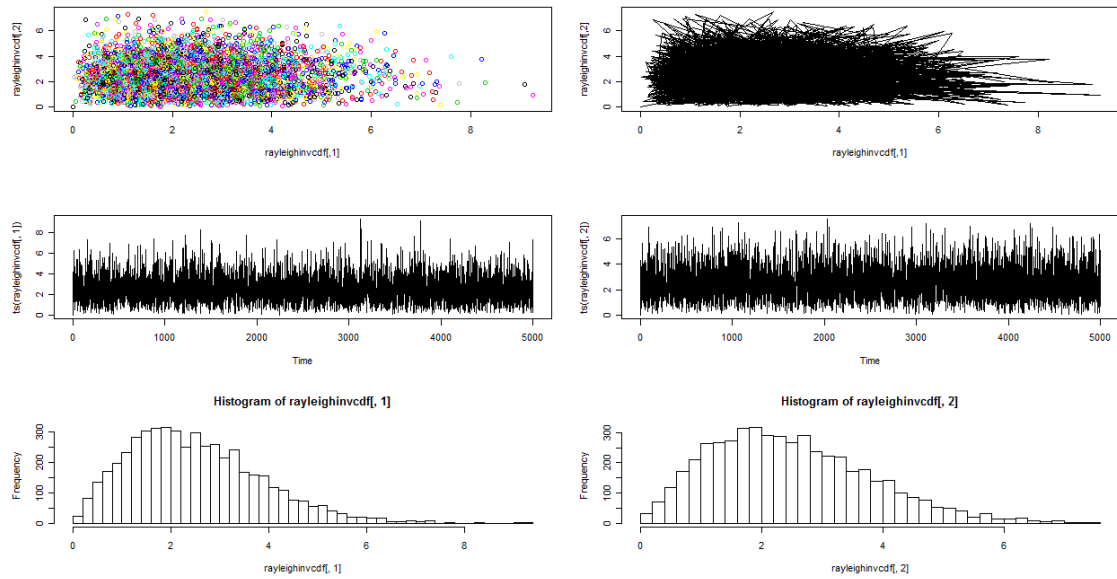


**Figura 5.9:** Gráfico da função ACF sobre a variável  $x$  da densidade da Eq. 5.49.

### 5.5.1 Densidades de Rayleigh e de Pareto

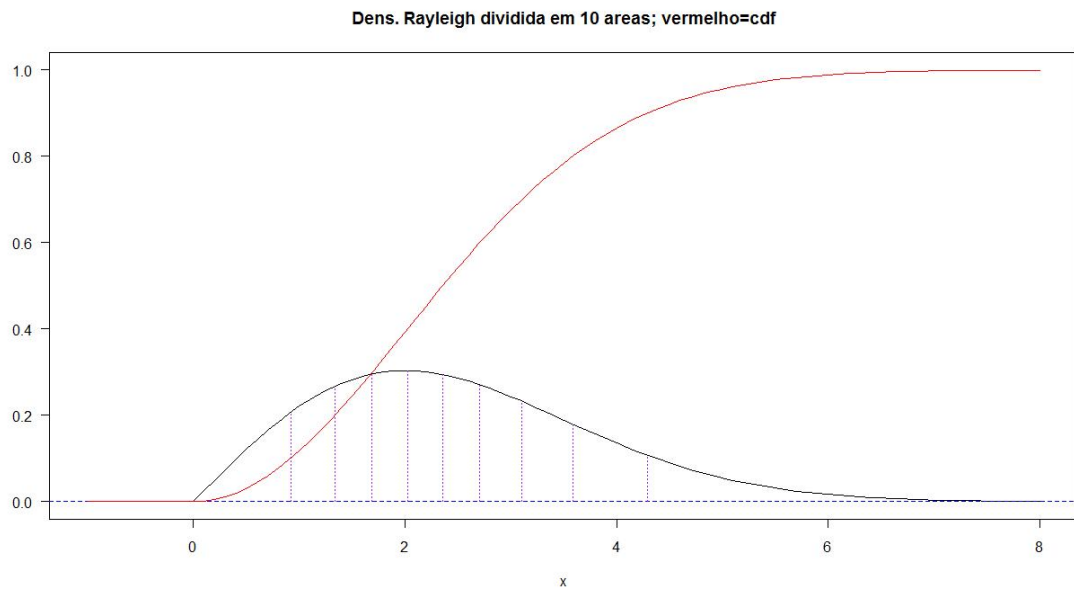
Utilizando o Amostrador de Gibbs, amostras foram obtidas de uma densidade de Rayleigh, dada pela Equação 5.50; o gráfico desta amostragem pode ser visto na Figura 5.10.

$$f(x, \sigma) = \frac{x}{\sigma^2} e^{\frac{-x^2}{2\sigma^2}}, \quad x \geq 0, \text{ para } \sigma > 0 \quad (5.50)$$



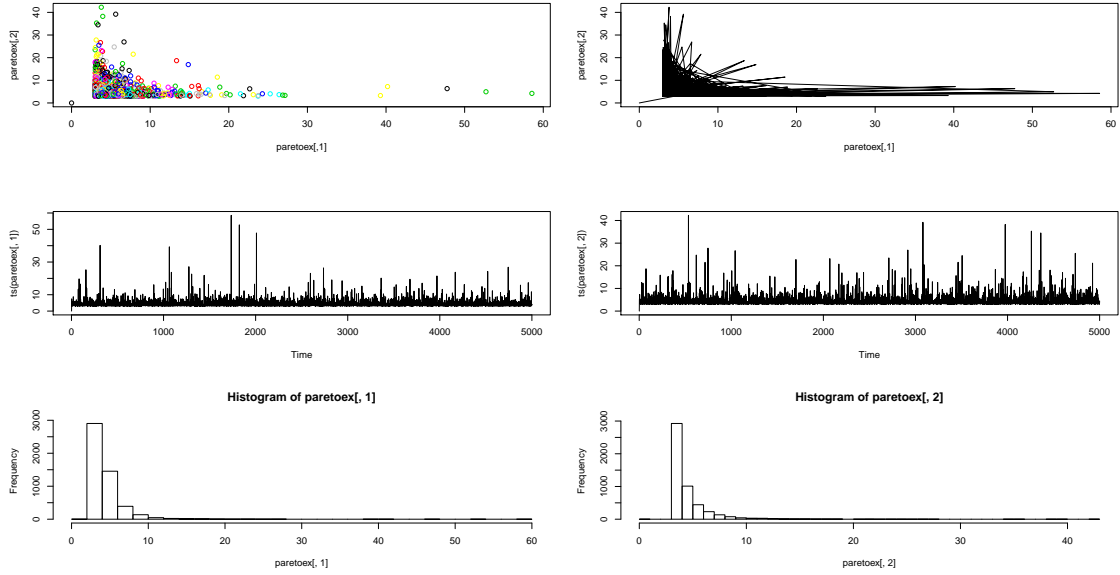
**Figura 5.10:** Gráfico de uma densidade Rayleigh amostrada com o Amostrador de Gibbs.

Na Figura 5.11, pode-se observar a distribuição acumulada de Rayleigh, em uma dimensão.



**Figura 5.11:** Gráfico da distribuição acumulada de Rayleigh.

Fechando a lista das densidades, pode-se observar na Figura 5.12 o resultado da amostragem, através do algoritmo Amostrador de Gibbs, da densidade de Pareto.



**Figura 5.12:** Gráfico das amostras de uma distribuição de Pareto amostrada com o Amostrador de Gibbs.

Uma variável aleatória  $X$  tem uma distribuição de Pareto com parâmetros  $x_0$  e  $\alpha$  ( $x_0 > 0$  e  $\alpha > 0$ ) se  $X$  tiver uma distribuição contínua para a qual a função de densidade de probabilidade  $f(x | x_0, \alpha)$  é dada por:

$$f(x) = \begin{cases} \frac{\alpha x_0^\alpha}{x^{\alpha+1}} & \text{para } x \geq x_0 \\ 0 & \text{para } x < x_0 \end{cases}$$

## 5.6 Covariância nas cadeias de Markov

Uma aplicação da desigualdade de Chebychev mostra que a convergência de uma média de variáveis aleatórias de uma cadeia de Markov pode ser ligada ao comportamento das covariâncias, com uma condição suficiente para convergência em probabilidade seria que as covariâncias fossem para zero. Assume-se que a cadeia de Markov é Harris positiva e aperiódica, e além disso, estacionária. Assume-se também que as variáveis aleatórias da cadeia têm variância finita. Assim, seja  $(X_n)$  uma cadeia de Markov estacionária ergódica com média zero e variância finita. A variância da média das  $X_i$ 's é

$$\text{var}\left(\frac{\sum_{i=0}^n X_i}{n+1}\right) = \frac{1}{n+1}\text{var}(X_0) + \frac{2}{n+1} \sum_{k=1}^n \frac{n-k+1}{n+1} \text{cov}(X_0, X_k), \quad (5.51)$$

assim, o termo da covariância na Eq. 5.51 vai para zero se  $\sum_{k=1}^n \text{cov}(X_0, X_k)/n$  for para zero, e uma condição suficiente para isso é que  $\text{cov}(X_0, X_k)$  convirja para zero.

A formalização do que foi visto está no seguinte teorema.

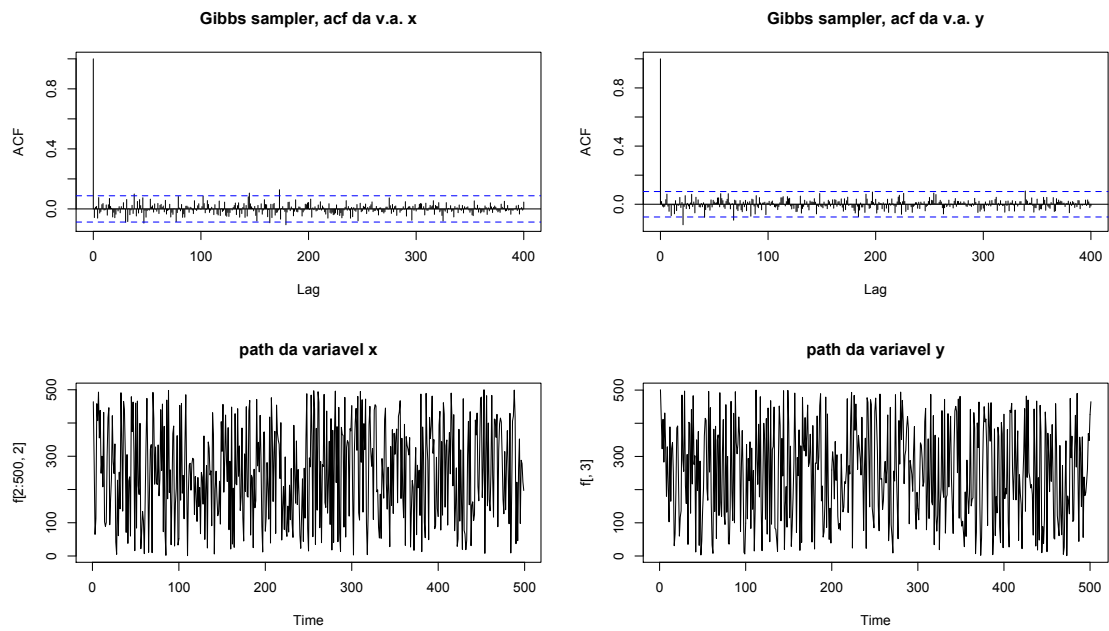
### Teorema

Se a cadeia de Markov  $(X_n)$  for positiva e aperiódica, com  $\text{var}(X_n) < \infty$ , então a  $\text{cov}(X_0, X_k)$  converge para 0.

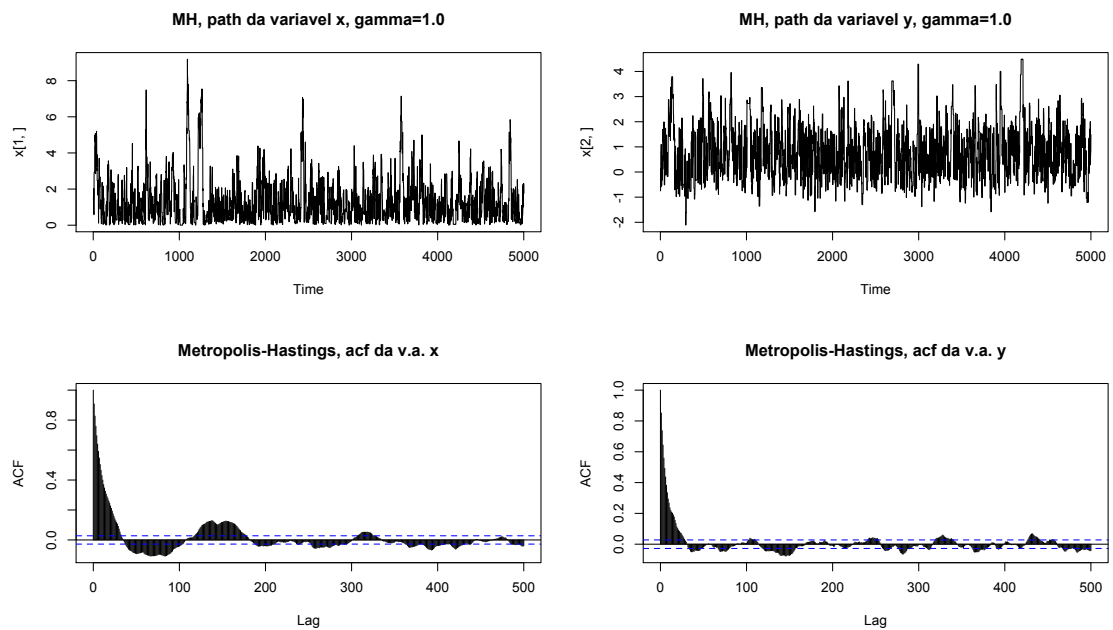
## 5.7 Análise comparativa das amostras geradas pelos dois algoritmos

É necessário verificar que as amostras geradas pelo algoritmo Amostrador de Gibbs realmente têm dependência entre si menor do que as amostras geradas por outros algoritmos da classe MCMC. Para verificar esta condição simulações foram executadas, tanto para o Amostrador de Gibbs quanto para o Metropolis-Hastings, variando-se o parâmetro  $\gamma$  no método de Metrópolis-Hastings e comparando-se o melhor caso obtido com o algoritmo Amostrador de Gibbs. O resultado foi realmente favorável ao Amostrador de Gibbs, conforme pode ser observado na Figura 5.13, referente ao algoritmo Amostrador de Gibbs, e nas Figuras 5.14, 5.15, 5.16, 5.17 e 5.18, referentes ao algoritmo Metropolis-Hastings. Dependendo da escolha de  $\gamma$  o algoritmo Metropolis-Hastings pode deixar de convergir, e a função de autocorrelação indicará a correlação não ideal entre as amostras (ideal seria não haver correlação entre elas), como mostrado principalmente nas Figuras 5.17 e 5.18.

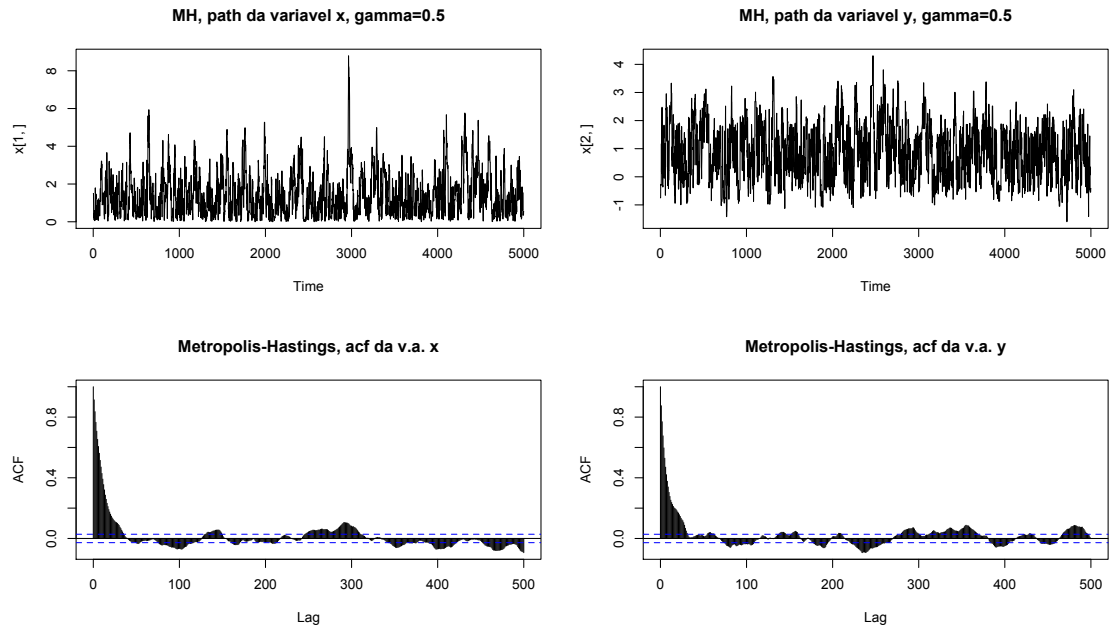
Estes gráficos também foram obtidos por simulação usando o software R.



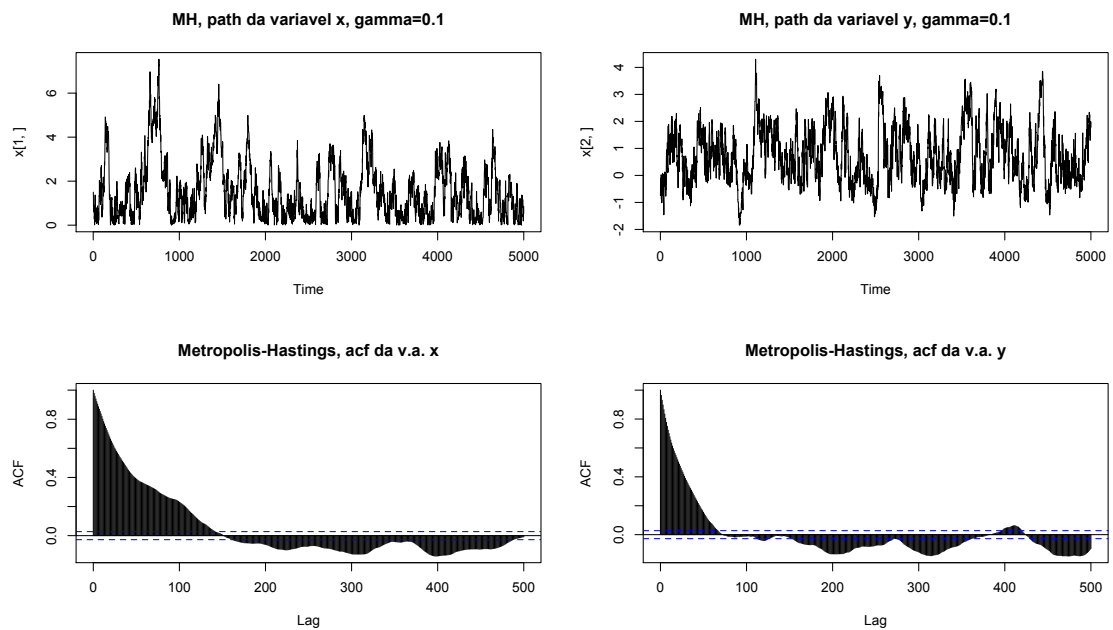
**Figura 5.13:** Gráfico das v.a.  $x$  e  $y$  e da função de autocorrelação das amostras obtidas com o Amostrador de Gibbs



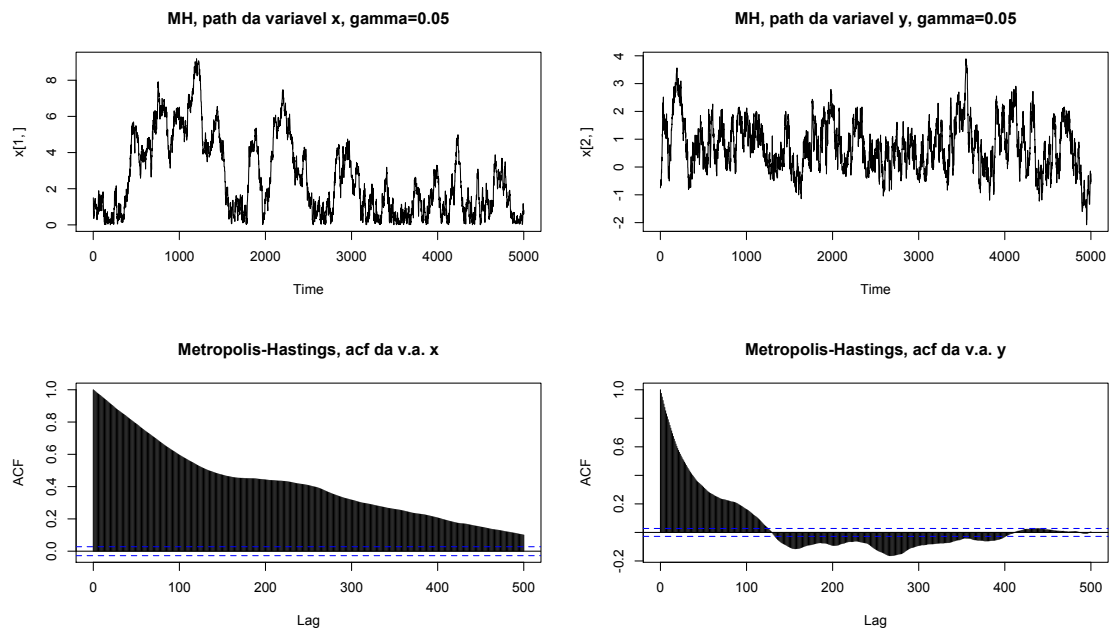
**Figura 5.14:** Gráfico das v.a.  $x$  e  $y$  e da função de autocorrelação das amostras obtidas com o Metropolis-Hastings, com o parâmetro  $\gamma$  igual a 1,0.



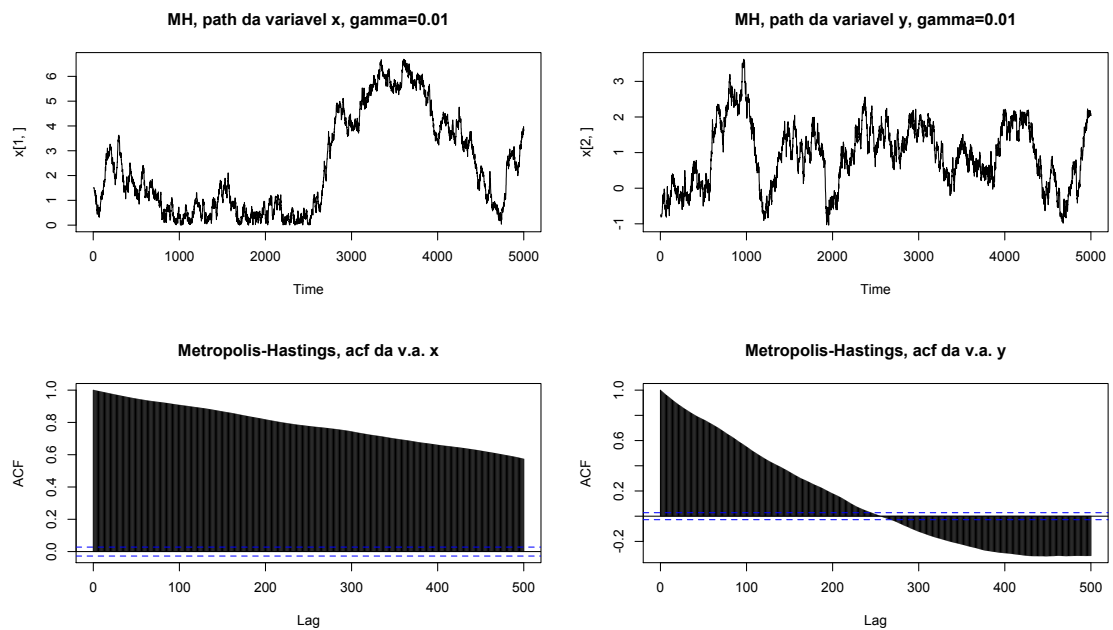
**Figura 5.15:** Gráfico das v.a.  $x$  e  $y$  e da função de autocorrelação das amostras obtidas com o Metropolis-Hastings, com o parâmetro  $\gamma$  igual a 0,5.



**Figura 5.16:** Gráfico das v.a.  $x$  e  $y$  e da função de autocorrelação das amostras obtidas com o Metropolis-Hastings, com o parâmetro  $\gamma$  igual a 0,1.



**Figura 5.17:** Gráfico das v.a.  $x$  e  $y$  e da função de autocorrelação das amostras obtidas com o Metropolis-Hastings, com o parâmetro  $\gamma$  igual a 0,05.



**Figura 5.18:** Gráfico das v.a.  $x$  e  $y$  e da função de autocorrelação das amostras obtidas com o Metropolis-Hastings, com o parâmetro  $\gamma$  igual a 0,01.

## 5.8 Métricas para Análise de Desempenho dos algoritmos do sistema

### 5.8.1 Amostras, Estatísticas e Estimadores

Nesta seção os conceitos de estatística e estimador são introduzidos. Critérios para a comparação de estimadores são também considerados. [72]

#### 5.8.1.1 Definição

O conjunto de valores de uma característica (observável) associada a uma coleção de indivíduos ou objetos de interesse é dito ser uma população.

Qualquer parte (ou subconjunto) de uma população é denominada uma amostra. De maneira mais formal, tem-se

#### 5.8.1.2 Definição

Uma sequência  $X_1, \dots, X_n$  de  $n$  variáveis aleatórias independentes e identicamente distribuídas (*i.i.d.*) com função densidade de probabilidade (f.d.p.) ou, no caso discreto, função de probabilidade (f.p.)  $f(x|\theta)$ , é dita ser uma amostra aleatória de tamanho  $n$  da distribuição de  $X$ . Neste caso, temos,

$$f(x_1, \dots, x_n|\theta) = \prod_{i=1}^n f(x_i|\theta) = f(x_1|\theta) \cdots f(x_n|\theta). \quad (5.52)$$

Conclui-se, a partir da Definição 5.8.1.2, que a amostra  $X_1, X_2, \dots, X_n$  é utilizada para se obter informação sobre o parâmetro  $\theta$ . A função de densidade (ou de probabilidade) conjunta dada em (5.52) é denominada **função de verossimilhança** de  $\theta$ , correspondente à amostra observada  $\mathbf{x} = (x_1, \dots, x_n)'$  e será denotada por

$$L(\theta; \mathbf{x}) = \prod_{i=1}^n f(x_i|\theta). \quad (5.53)$$



### 5.8.1.3 Definição

Qualquer função da amostra que não dependa de parâmetros desconhecidos é denominada uma estatística.

#### Exemplo 5.8.1.1

Sejam  $X_1, \dots, X_n$  uma amostra aleatória da variável aleatória  $X$ , com f.d.p. ou f.p.  $f(x|\theta)$ . Exemplos de estatística são: a média, a variância, a mediana, o valor máximo e o valor mínimo observado.

### 5.8.1.4 Definição

O conjunto  $\Theta$  em que  $\theta$  toma valores é denominado espaço paramétrico.

### 5.8.1.5 Definição

Qualquer estatística que assuma valores em  $\Theta$  é um estimador para  $\theta$ .

Em muitas situações, o interesse é estimar uma função  $g(\theta)$ . Supondo que, por exemplo, dada uma variável aleatória  $X \sim \mathcal{N}(\mu, \sigma^2)$ , o objetivo é estimar somente  $\mu$ , sendo  $\sigma^2$  um parâmetro de perturbação. Nesse caso,  $g(\theta) = \mu$ .

### 5.8.1.6 Definição

Qualquer estatística que assuma valores somente no conjunto dos possíveis valores de  $g(\theta)$  é um estimador para  $g(\theta)$ .

Um dos procedimentos comumente utilizados para se avaliar o desempenho de um estimador é o seu erro médio quadrático.

### 5.8.1.7 Definição

O erro quadrático médio (EQM ou MSE, *mean square error*) de um estimador  $\hat{\theta}$  do parâmetro  $\theta$  é dado por

$$MSE(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \theta)^2] \quad (5.54)$$

O MSE depende da variância de  $\hat{\theta}$  e de uma função denominada de **vício**

(*bias*) do estimador  $\hat{\theta}$ , onde

$$\mathbb{B}(\hat{\theta}) = \mathbb{E}(\hat{\theta}) - \theta$$

que pode ser entendida como o valor esperado de desvio do valor estimado, com relação ao valor verdadeiro do parâmetro.

Diz-se que um estimador  $\hat{\theta}$  é **não viciado** para  $\theta$  se

$$\mathbb{E}(\hat{\theta}) = \theta$$

para todo  $\theta \in \Theta$ , ou seja  $\mathbb{B}(\hat{\theta}) = 0$ , para todo  $\theta \in \Theta$ . Se, para todo  $\theta \in \Theta$ ,  $\lim_{n \rightarrow \infty} \mathbb{B}(\hat{\theta}) = 0$ , dizemos que o estimador  $\hat{\theta}$  é **assintoticamente não viciado** para  $\theta$ .

**Exemplo.** Um exemplo de estimador viciado é a média amostral modificada definida por:

$$\check{A} = \frac{1}{2N} \sum_{n=0}^{N-1} x[n] \quad (5.55)$$

que em média resulta em:

$$\mathbb{E}(\check{A}) = \frac{1}{2}A. \quad (5.56)$$

Um estimador não viciado é o estimador que em média gera o valor verdadeiro dos parâmetros. Matematicamente,

$$\mathbb{E}(\hat{\theta}) = \theta. \quad (5.57)$$

Um estimador da média amostral é um estimador não viciado, definido por:

$$\check{A} = \frac{1}{N} \sum_{n=0}^{N-1} x[n]. \quad (5.58)$$

o qual resulta em média

$$\mathbb{E}(\check{A}) = A. \quad (5.59)$$

Um resultado muito útil é

$$MSE(\hat{\theta}) = var(\hat{\theta}) + [\mathbb{B}(\hat{\theta})]^2, \quad (5.60)$$

### **Prova**

Se o erro for escrito como

$$(\hat{\theta} - \theta) = [\hat{\theta} - \mathbb{E}(\hat{\theta})] + [\mathbb{E}(\hat{\theta}) - \theta];$$

e se o termo à direita da igualdade for elevado ao quadrado, mantendo-se os colchetes intactos, e forem tomadas as respectivas médias, o termo intermediário cruzado será zero, pois  $\mathbb{E}[\hat{\theta} - \mathbb{E}(\hat{\theta})] = 0$ . Os outros dois termos são  $[\mathbb{E}(\hat{\theta})]^2$  e  $var(\hat{\theta})$ . (Se um lado é infinito, o outro também deve ser, e, portanto, a Equação 5.60 está demonstrada.) ■

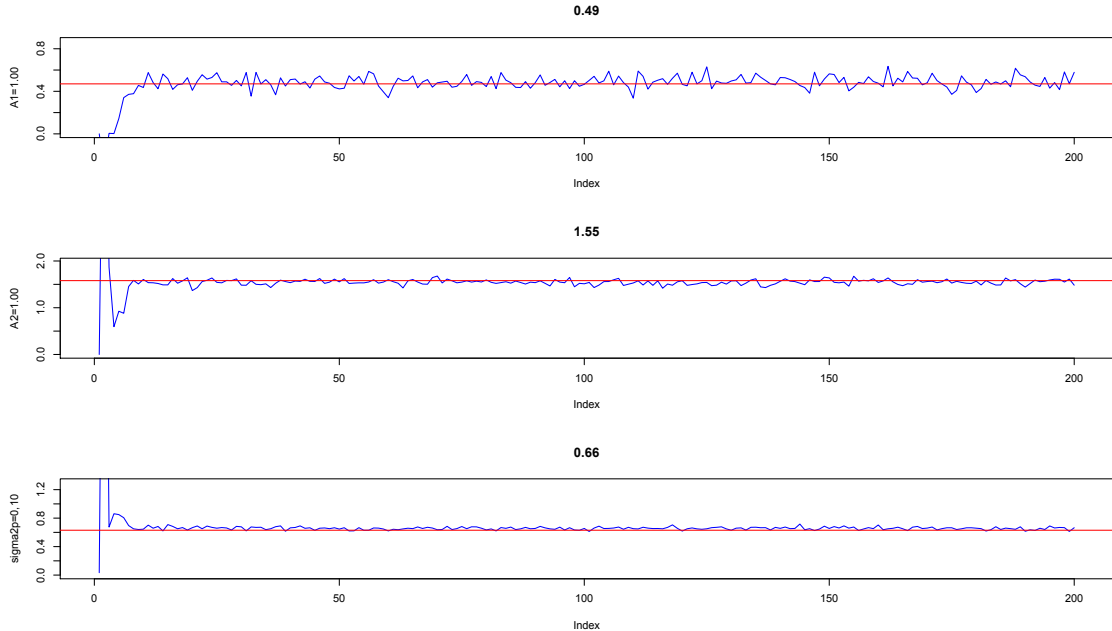
No caso em que  $\hat{\theta}$  é um estimador não viciado para  $\theta$ , tem-se que

$$MSE(\hat{\theta}) = var(\hat{\theta}), \quad (5.61)$$

ou seja, o erro quadrático médio de  $\hat{\theta}$  reduz-se à sua variância.

Para estabelecer a análise de desempenho do detector bayesiano, foram designadas três métricas principais: BER (*bit error rate*); MSE (*mean square error*) e MEP (*mean error percentual*).

Com o seguinte cenário, foram determinados, através de simulações, valores de MSE para os estimadores amplitude,  $\mathbf{a}$ , e variância do ruído do canal,  $\sigma^2$ . O cenário de teste foi composto por dois assinantes,  $K = 2$ ; com tamanho de quadro igual a 256,  $M = 256$ , e um laço (*loop*) de simulação. O gráfico da Figura 5.19 representa um dos resultados de simulações que foram executadas para se determinar os MSE.



**Figura 5.19:** Gráfico das amplitudes de dois usuários e da variância do ruído do canal, obtidas por simulação.

Os valores dos MSE obtidos são vistos na Tabela 5.1. Nesta tabela, MSE1 é

o erro quadrático médio do estimador da amplitude do usuário 1, na recepção do sinal; MSE2 é o erro quadrático médio do estimador da amplitude do usuário 2; e o MSE3 é o erro do estimador da energia do ruído do canal. SE(MSE) é o desvio padrão do erro, que foi calculado como, aproximadamente, a raiz quadrada do MSE.

**Tabela 5.1:** Resultados dos cálculos de MSE para algumas grandezas

MSE, SE e BER	resultados
MSE1, ampl. usuário 1	0.0032
MSE2, ampl. usuário 2	0.0041
MSE3, energia do ruído	0.0011
MEP1, ampl. usuário 1	0.3180
MEP2, ampl. usuário 2	0.4079
MEP3, energia do ruído	0.1071
SE1(MSE1), ampl. usuário 1	0.4906
SE2(MSE2), ampl. usuário 2	1.5544
SE3(MSE3), energia do ruído	0.6611
<i>setup</i> na transmissão:	
amplitudes usuário 1	0.470
amplitude usuário 2	1.580
energia do ruído	0.630
grandezas medidas na recepção:	
amplitude usuário 1	0.482
amplitude usuário 2	1.569
energia do ruído	0.662
BER1	0.371
BER2	0.051

Como pode ser observado, os valores de MSE obtidos são bem pequenos, o que demonstra a eficiência dos estimadores mesmo para o caso de poucas amostras. Como conclusão, entende-se que o Amostrador de Gibbs provê estimações precisas e confiáveis na maioria dos casos, e ele próprio é um estimador não viciado.

### 5.8.1.8 BER

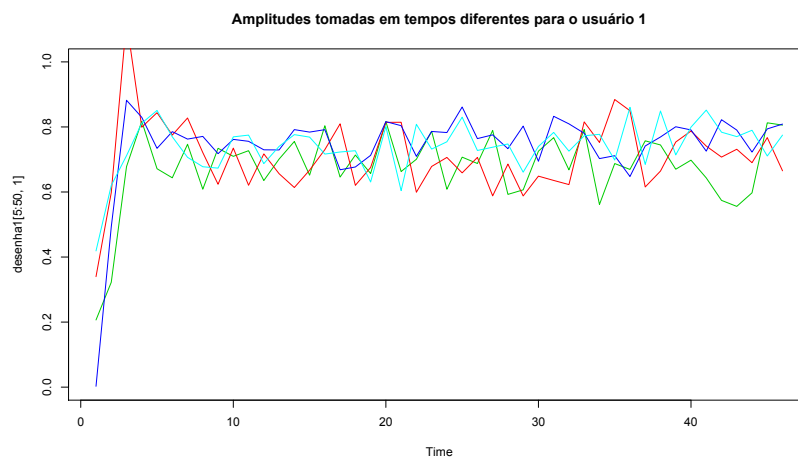
Outra métrica importante na análise de sistemas de comunicação é a chamada taxa de erro de bit, **BER** (*bit error rate*), já vista na Tabela 5.1. Os resultados obtidos por simulações da taxa de erro de bits mantiveram-se dentro de uma faixa de variação considerada boa, comparativamente aos valores obtidos para o detector filtro casado, ou detector convencional. Portanto, os resultados demonstram que o detector MuD analisado provê bons resultados de taxa de erros para bits transmitidos, num cenário de canal de ruído branco. Neste trabalho foram feitos testes com dois, cinco, dez e vinte usuários utilizando o sistema, com diferen-

tes configurações de amplitudes. O resultado das simulações serão analisados no Capítulo 6 deste trabalho.

### 5.8.1.9 Análise do tempo de convergência

Conforme foi visto na seção que descreve o Amostrador de Gibbs, um método cientificamente comprovado de diagnóstico da convergência ainda não foi estabelecido. Porém, na prática alguns procedimentos podem ser adotados no sentido de se verificar se o algoritmo está no caminho certo ou não da convergência para a distribuição desejada. Pode-se, por exemplo, monitorar os valores gerados pelo algoritmo de forma a se verificar visualmente, através de gráficos, por exemplo, se as sequências em formação já atingiram a região de convergência (nesta região os desvios são mais limitados, indicando uma tendência) e, portanto já ultrapassaram a fronteira do *burnin*. De outra forma, pode-se também verificar se a sequência já atingiu a estabilidade, após um período de variações bruscas, e está convergindo para o valor final, por meio da geração de sequências adicionais, cada uma com ponto de início diferente, e da posterior verificação da evolução das mesmas, observando se há convergência para um mesmo ponto ou região. Apesar de que estes métodos não são efetivos em cem por cento dos casos, eles têm sido muito citados em trabalhos passados. [64]

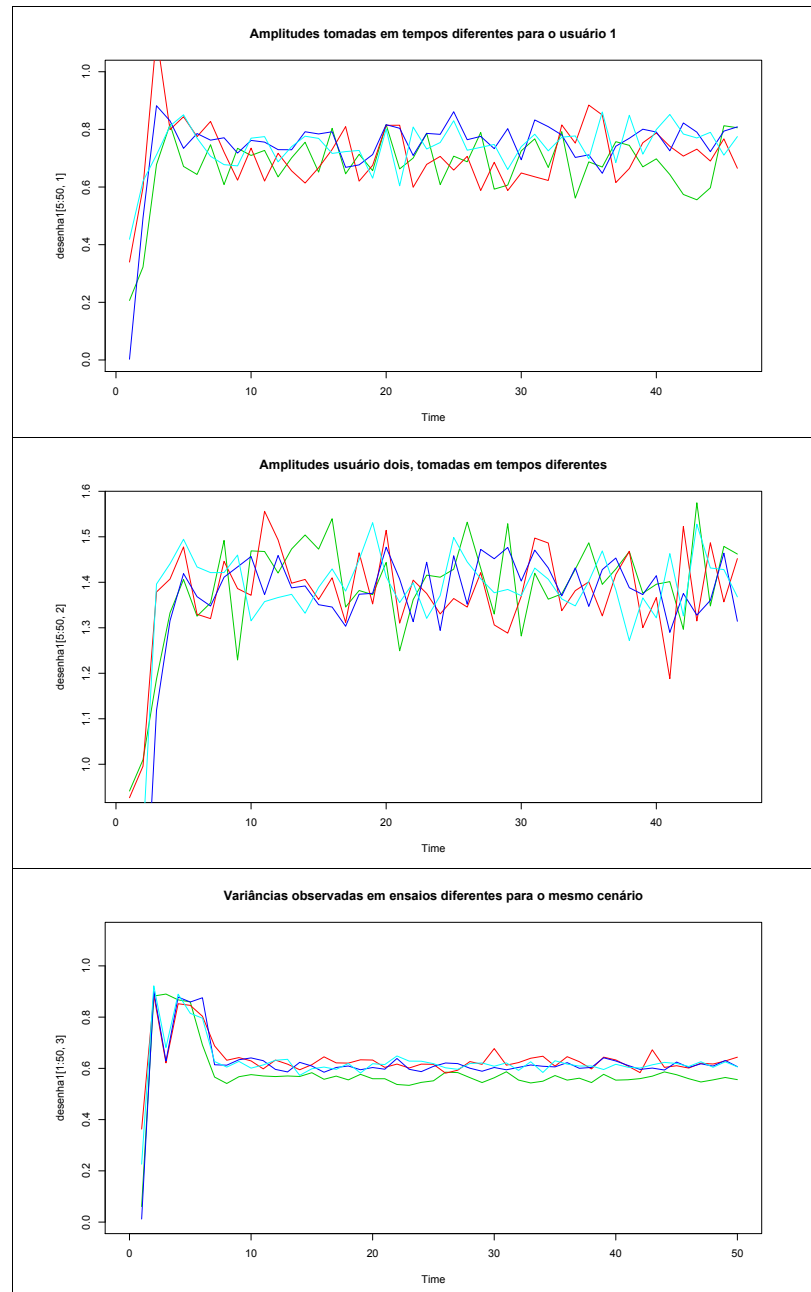
Na Figura 5.20, observa-se como a convergência é rápida para amplitudes de sinais de somente um usuário, num cenário de dois usuários e canal de ruído branco.



**Figura 5.20:** Gráfico das amplitudes de sinais do usuário 1, que mostra a rápida convergência.

Na Figura 5.21 são mostradas as sequências dos valores das amplitudes de

usuários e da variância do ruído branco. Nos gráficos desta figura, observa-se o período entre o início das sequências e aproximadamente após a estabilização das mesmas. A convergência dessas sequências é rápida, conforme pode-se observar.



**Figura 5.21:** Gráfico das Amplitudes e variância dos sinais; a convergência é rápida para o algoritmo adotado.

## 5.9 Considerações finais

Neste capítulo, alguns testes e análises foram realizados para ajudar na decisão se o Amostrador de Gibbs é um algoritmo que pode ser utilizado no projeto de um sistema de espalhamento espectral com detecção multiusuário. Foi verificado, através dos resultados obtidos, que o algoritmo converge para a resposta final rapidamente; por exemplo, no caso de um laço com 200 iterações, antes mesmo da vigésima iteração ocorrer, os valores das variáveis de teste já estavam oscilando ao redor do valor final correto, conforme Figura 5.19. Uma análise de comparação do Amostrador de Gibbs com o Metropolis-Hastings gerou resultados muito próximos para os dois algoritmos, e, assim, não indicaram uma superioridade de um algoritmo em relação ao outro. A conclusão a que se chegou foi que o Amostrador de Gibbs é um bom algoritmo para resolver problemas multidimensionais, pois convergiu rapidamente nos testes realizados; praticamente chegou à resposta correta em todos os casos testados; é um algoritmo muito simples de se implementar e sua complexidade é baixa. Portanto, concluiu-se que o Amostrador de Gibbs poderia ser utilizado sem problemas em sistemas de comunicação por espalhamento espectral com inferência bayesiana e modulação *wavelet*.

## 6 Detecção multiusuário bayesiana com aplicação de *wavelets*

### Análise e Resultados



## 6.1 Detecção multiusuário bayesiana com aplicações de wavelets

Em um sistema de comunicação sem fio, de acesso múltiplo, usuários utilizam os recursos do sistema de forma não coordenada, assincronamente. O canal de comunicação utilizado é um canal aleatório cujas características variam no tempo e são de natureza não linear. A detecção dos sinais de forma ótima, demonstrada em [13, 73], utiliza um detector multiusuário ótimo, que detecta os sinais corrompidos por ruído branco e explora os sinais interferentes de acesso múltiplo, proporcionando, assim, um ganho superior ao alcançado por um sistema que utiliza um detector convencional, ou filtro casado. Uma característica adicional deste tipo de detector (multiusuário) é que ele é robusto ao efeito *near-far*, e portanto, superior ao detector convencional, que não trata esse efeito.

No entanto, como em todos os sistemas reais, o detector ótimo também tem desvantagens. A principal é em relação ao seu tempo de computação. Como o algoritmo que resolve o problema de detecção ótima tem complexidade TCB (*time complexity per bit*) de  $O(\frac{2^K}{K})$  [74, 75], isto é, ele precisa de um tempo infinito para apresentar a solução correta ao problema de detecção, o que o tornou inviável para implementação, novos algoritmos foram desenvolvidos ao longo das duas últimas décadas. Tais algoritmos deram origem aos detectores denominados subótimos, porque apesar de resolverem o problema em um tempo finito, sua eficiência fica abaixo da do detector ótimo. A grande vantagem desses detectores é que eles são implementáveis na prática.

Dentro dessa nova classe de detectores, dentre os quais encontram-se o Descorrelacionador, o MMSE (linear) e o PIC (*parallel interference canceller*) [2], para citar somente alguns, existe também um detector multiusuário que utiliza o método da inferência bayesiana para resolver o problema da detecção. Basicamente o que o detector bayesiano faz é calcular a probabilidade *a posteriori* a partir de uma informação recebida e de distribuições *a priori*. Com este cálculo resolvem-se dois problemas: o da detecção e o da estimação de parâmetros desconhecidos. Esta é basicamente a metodologia da inferência bayesiana.

Com relação ainda ao detector bayesiano, que é empregado neste trabalho, o projeto original foi idealizado em [9], na forma de um detector adaptativo (não utiliza sequência de treinamento) bayesiano, síncrono, que utiliza no processamento de detecção e decodificação casada um código Turbo. O trabalho desenvolvido aqui tem uma contribuição original, de implementação prática e que abriu possibilidades de novas investigações com a associação das transformadas *wavelets* e a detecção multiusuário. Embora sem utilizar código corretor de erro, foram pes-

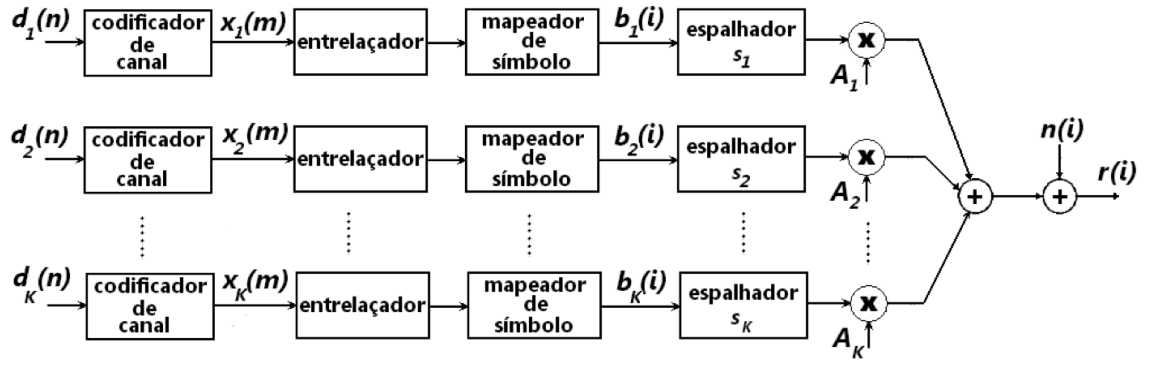
quisadas novas funcionalidades do detector MuD bayesiano e estabelecidos novos testes de verificação de características a serem exploradas. Os objetivos principais foram, resumidamente, verificar o funcionamento do detector original, criar um código para simulações do sistema, adicionar o cancelamento de ruído por transformadas *wavelets*, e investigar o desenvolvimento de novas funcionalidades.

Os itens principais nos quais o trabalho baseou-se foram:

- caracterização do efeito *near-far* para o detector MuD bayesiano;
- verificação do cancelamento das interferências (MAI) pelo detector, quantificando a melhoria da detecção e da estimação;
- comparação das características deste detector com as do detector convencional;
- testes do detector no modo assíncrono;
- condições de testes para um número maior do que dois usuários utilizando o canal simultaneamente (foram testados casos com 2, 5 e 10 usuários);
- contribuição mais importante do trabalho: desenvolvimento de uma função totalmente nova para o detector bayesiano: *denoising* por meio de *wavelets* (a partir do qual o detector passou a ser denominado de “detector multiusuário bayesiano com aplicação de *wavelets*”);
- outros testes de caracterização, como, por exemplo, uso de sequências de Gold.

## 6.2 Sistema utilizado - Caso síncrono

Conforme já foi visto no capítulo específico da descrição do sistema, o tipo de sistema que é utilizado neste trabalho é um sistema de detecção multiusuário, utilizado na detecção de sinais de símbolos enviados através de canal sem fio de ruído branco gaussiano. O algoritmo do detector faz adicionalmente a estimação de parâmetros desconhecidos. O diagrama do transmissor síncrono com  $K$  usuários é visto na Figura 6.1 (figura reproduzida aqui para facilitar a visualização).



**Figura 6.1:** Sistema de comunicação de acesso múltiplo.

Um receptor que utiliza a metodologia da inferência bayesiana com detecção multiusuário, mas que não utiliza código Turbo, foi planejado para compor o sistema de comunicação desejado. Este receptor, denominado aqui de detector Bayes-*only* foi o primeiro protótipo do trabalho. A Figura 6.2 mostra o receptor. Utilizou-se o software “R” (<http://cran.r-project.org>) [76] para desenvolver os códigos do transmissor, do canal e do receptor do sistema. A codificação do software foi realizada e vários testes foram então feitos por simulação. O R é uma linguagem orientada a objetos, de domínio público, que possui muitas funções estatísticas pré-programadas. É um ambiente de programação, similar ao Matlab, mas com colaboração de pesquisadores do mundo inteiro. Foi escolhida para este trabalho pois tem aplicações prontas para inferência bayesiana e permite programação de novas funções, além de se comunicar com outras linguagens: aceita subrotinas da linguagem C, por exemplo.

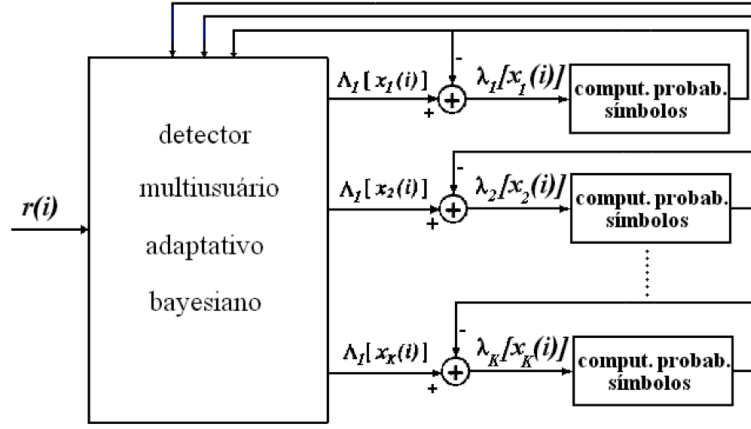


Figura 6.2: Diagrama em blocos do receptor Bayes-only.

### 6.2.1 Resultados das estimações das amplitudes e variância do ruído e da detecção multiusuário

#### Configuração

Foi considerado primeiramente o caso síncrono, com dois usuários ( $K = 2$ ), com ganho de processamento ( $P = 10$ ). A matriz das sequências de espalhamento  $\mathbf{S}$  é dada por:

$$\mathbf{S}^T = \frac{1}{\sqrt{10}} \begin{bmatrix} -1 & -1 & +1 & +1 & -1 & +1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 \end{bmatrix} \quad (6.1)$$

Inicialmente, estes valores foram escolhidos com base nos valores testados no artigo original do detector bayesiano, para validar o código desenvolvido no ambiente R. Nas simulações posteriores outros valores de “sequência de máximo comprimento” foram utilizados. Também outros tipos de códigos de espalhamento, como o Gold, por exemplo, foram testados.

As amplitudes configuradas no transmissor são  $A_1 = 0,63$  e  $A_2 = 1,58$ ; a variância do ruído configurada foi  $\sigma^2 = 0,63$ . O fato importante a ser observado aqui é que com estes valores de amplitude, diferente um do outro, a de maior intensidade pode causar uma interferência forte na outra; outro fator que colabora com a interferência é a correlação entre os sinais, estabelecida pelos códigos de espalhamento. Entretanto, o que foi constatado, pela análise dos resultados dos testes, é que o detector multiusuário tem um processamento que impede que essas diferenças de amplitudes causem problema de recepção de sinais pelos usuários, e a isto denomina-se robustez ao efeito *near-far*.

A matriz de correlação é dada por:

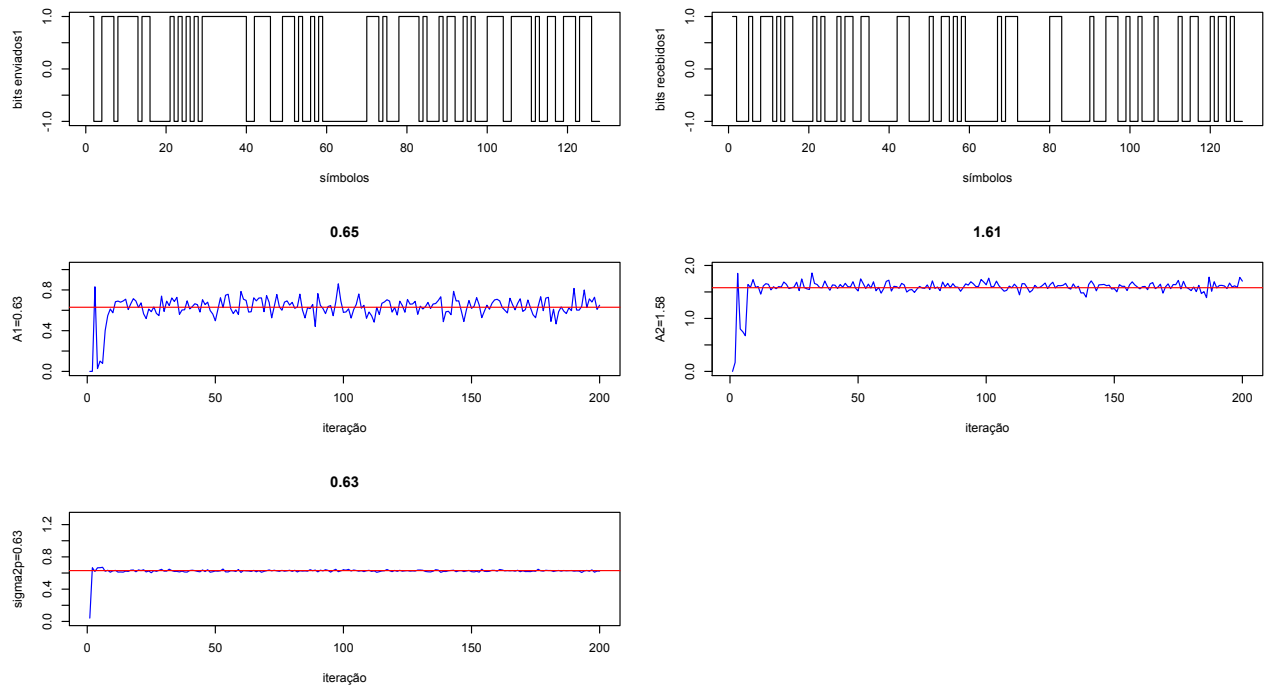
$$\mathbf{R} = \mathbf{S}^T \times \mathbf{S} = \frac{1}{10} = \begin{bmatrix} 10 & -2 \\ -2 & 10 \end{bmatrix} \quad (6.2)$$

Na simulação, as seguintes distribuições *a priori* conjugadas, não informativas, foram utilizadas para o Amostrador de Gibbs:

$$\begin{aligned} p(\mathbf{a}^{(0)}) &\sim \mathcal{N}(\mathbf{a}_0, \mathbf{\Sigma}_0), I_{d_{\{\mathbf{a}^{(0)} > 0\}}} \rightarrow \mathbf{a}_0 = [11]^T \quad \mathbf{\Sigma}_0 = 1000I \\ p(\sigma^{2(0)}) &\sim \chi^2(v_0, \lambda_0) \rightarrow v_0 = 1, \lambda_0 = 0.1 \end{aligned} \quad (6.3)$$

onde as amplitudes seguem uma distribuição Normal, com média  $\mathbf{a}_0$  e variância  $\mathbf{\Sigma}_0$ ;  $\mathbf{a}^{(0)}$  é o vetor dos valores iniciais das amplitudes sorteados a partir dos parâmetros configurados;  $\mathbf{a}_0$  é o vetor das amplitudes com valores iniciais configurados;  $\mathbf{\Sigma}_0$  é a matriz das variâncias das amplitudes;  $I_d$  é o indicador que exige que as amplitudes sorteadas sejam positivas;  $I$  é a matriz identidade. A variância do ruído segue uma distribuição Qui-Quadrado Inversa com parâmetros  $v_0$  e  $\lambda_0$ , estes parâmetros também são configurados com valores iniciais que indicam uma fraca convicção por parte do especialista responsável pela configuração inicial sobre os mesmos, e que permitem à verossimilhança ter um peso maior no cálculo da probabilidade, ou densidade, *a posteriori*.

Na Figura 6.3 são apresentados os gráficos dos resultados da estimação dos parâmetros e detecção dos símbolos para o cenário com dois usuários. Na linha superior da figura, os dois gráficos representam “transmissão”, à esquerda, e “recepção”, à direita, dos símbolos para apenas um dos usuários.



**Figura 6.3:** Símbolos detectados de um usuário; estimação das amplitudes de dois usuários e variância do ruído.

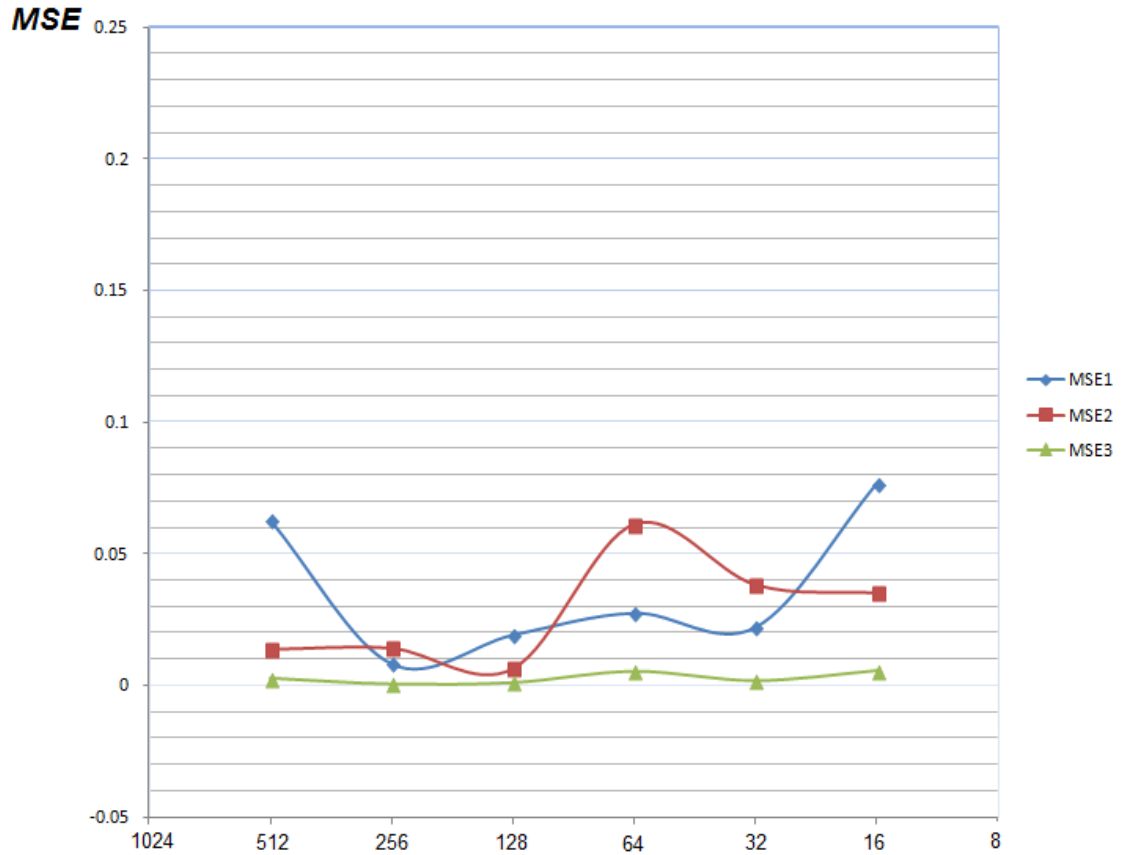
Na segunda linha, há dois gráficos das estimações das amplitudes dos usuários 1 e 2. Nota-se que mesmo sendo a amplitude do usuário 1 de intensidade muito inferior à amplitude do usuário 2, a estimação foi executada sem problemas e os resultados são muito próximos dos valores configurados. Os valores são apresentados também na Tabela 6.1.

**Tabela 6.1:** Estimação das amplitudes e variância para o teste com dois usuários.

parâmetro	valor configurado	valor estimado
amplitude usuário 1	1.58	1.61
amplitude usuário 2	0.63	0.65
variância do ruído	0.63	0.63

### Influência do tamanho do quadro na estimação das amplitudes e do ruído

A seguir são apresentados os resultados da análise da influência do tamanho do quadro, comprimento  $M$ , na estimação das amplitudes dos usuários e da variância do ruído. Conforme mostra o gráfico da Figura 6.4, observa-se que o tamanho do quadro não influencia diretamente, segundo os dados obtidos, o cálculo do erro médio quadrático, para os testes executados. No gráfico, MSE1 é o erro quadrático médio da estimação da amplitude do usuário 1; MSE2 refere-se ao usuário 2 e MSE3 refere-se à variância do ruído estimada.



**Figura 6.4:** Erro médio quadrático *versus*  $M$  (comprimento do quadro).

Com  $M = 128$ ,  $n_0 = 20$  e  $N = 5$  ( $N$  é o *loop* ou laço) fixados, variou-se o parâmetro  $K$ , que é o número de usuários no sistema, e posteriormente o tempo de execução foi anotado.

Para os valores de  $K$ :  $K = 2$ ;  $K = 5$  e  $K = 10$ , os tempos de execução dos testes foram registrados na Tabela 6.2.

**Tabela 6.2:** Resultados dos tempos de execução com variação do número de usuários.

$M, n_0, N$ e $K$	$\Delta t$
M=128; $n_0 = 70$ ; N=5; K=2	0 min e 57 s
M=128; $n_0 = 70$ ; N=5; K=5	2 min e 15 s
M=128; $n_0 = 70$ ; N=5, K=10	5 min e 27s

Com relação à Figura 6.5, a estimação é calculada para um grupo de cinco usuários mais a variância do ruído.

$$\mathbf{S}^T = \frac{1}{\sqrt{10}} \begin{bmatrix} -1 & -1 & +1 & +1 & -1 & +1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & -1 & +1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & -1 & -1 & -1 & +1 & -1 \\ -1 & -1 & +1 & -1 & -1 & -1 & +1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & +1 & -1 & -1 & -1 & -1 \end{bmatrix} \quad (6.4)$$

A Equação 6.4 apresenta as sequências de espalhamento utilizadas e a matriz de correlação é dada por:

$$\mathbf{R} = \mathbf{S}^T \times \mathbf{S} = \frac{1}{\sqrt{10}} \begin{bmatrix} +10 & -2 & -2 & +4 & -2 \\ -2 & +10 & +2 & +0 & +2 \\ -2 & +2 & +10 & -4 & +2 \\ +4 & +0 & -4 & +10 & -4 \\ -2 & +2 & +2 & -4 & +10 \end{bmatrix} \quad (6.5)$$

Novamente, as distribuições *a priori* não informativas são utilizadas na configuração inicial do Amostrador de Gibbs.

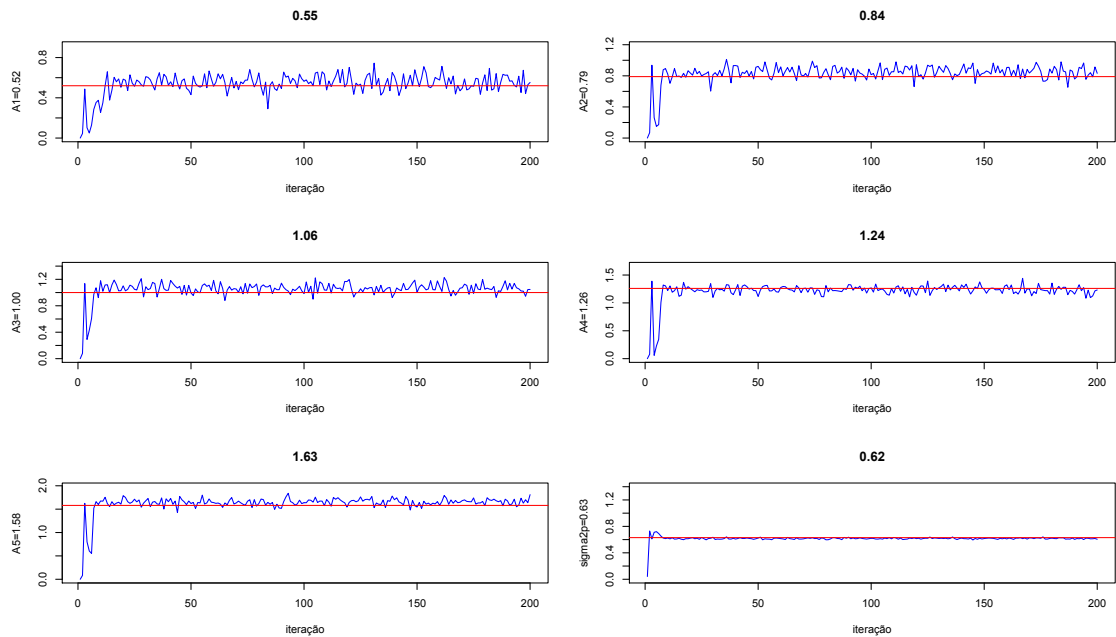
$$p(\mathbf{a}^{(0)}) \sim \mathcal{N}(\mathbf{a}_0, \Sigma_0) I_{d\{\mathbf{a}^{(0)} > 0\}} \rightarrow \mathbf{a}_0 = [11111]^T \quad \Sigma_0 = 1000I \quad (6.6)$$

$$p(\sigma^{2(0)}) \sim \chi^2(v_0, \lambda_0) \rightarrow v_0 = 1, \lambda_0 = 0.1$$

As amplitudes selecionadas para o teste são:  $A_1 = 0,52$ ;  $A_2 = 0,79$ ;  $A_3 = 1,00$ ;  $A_4 = 1,26$ ;  $A_5 = 1,58$ ; e a variância do ruído é igual a:  $\sigma^2 = 0,63$ . Pode-se verificar que o pior caso, o da amplitude do usuário 1, mesmo tendo o valor de 0,52, inferior ao da variância do ruído e também inferior aos valores das amplitudes dos outros usuários, foi estimada com sucesso e com uma boa precisão, apenas 5,8% de diferença.

Na Figura 6.5 podem ser vistos os gráficos das amplitudes recebidas do canal sem fio; estas amplitudes são estimadas pelo Amostrador de Gibbs. As amplitudes

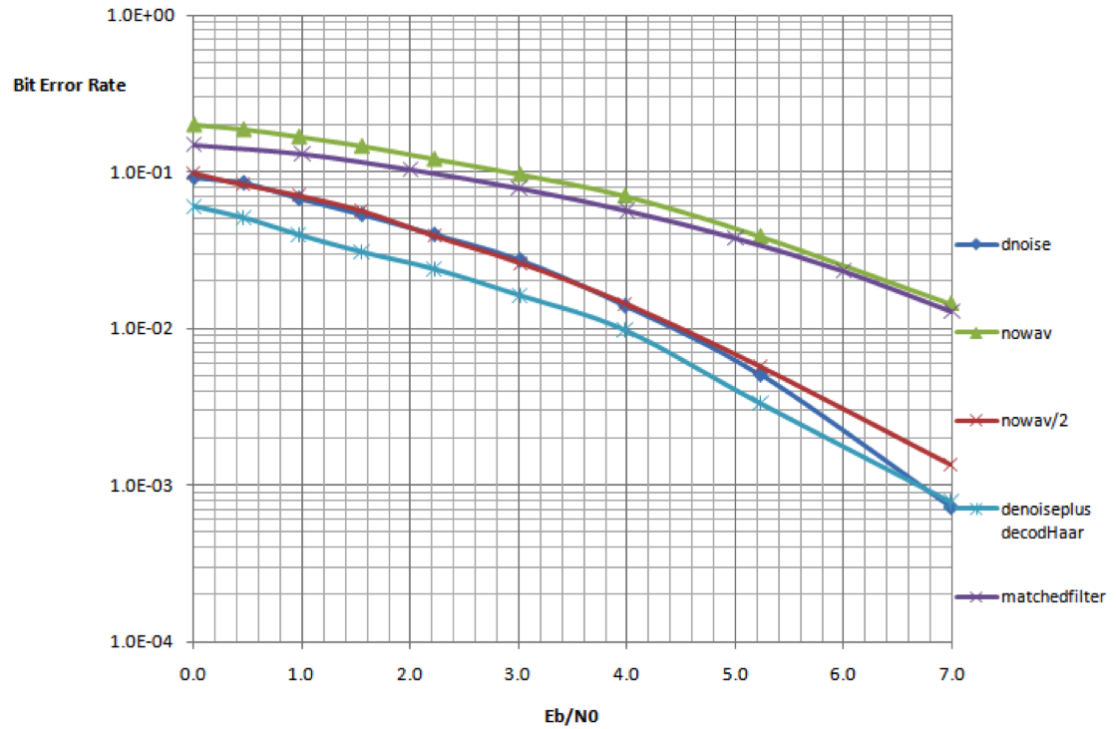




**Figura 6.5:** Valores estimados das amplitudes de cinco usuários e variância do ruído.

configuradas no transmissor estão ao lado dos gráficos; os valores sobre os gráficos são os estimados. Pode-se notar uma rápida convergência e uma boa concordância entre valor configurado e valor estimado. Os valores de amplitudes e variância foram escolhidos aleatoriamente, o cenário é composto por cinco usuários que transmitem sinais de amplitude fixa no intervalo de medição. Na determinação dos valores a serem simulados, somente houve o cuidado de selecionar amplitudes diferentes e algum valor comparável ao da variância do ruído, e também valores maiores do que a variância do ruído. Como pode ser observado, valores “grandes” de amplitude não prejudicaram a estimação dos valores menores, isto é, não ocorreu o efeito prejudicial da interferência com valores mais intensos.

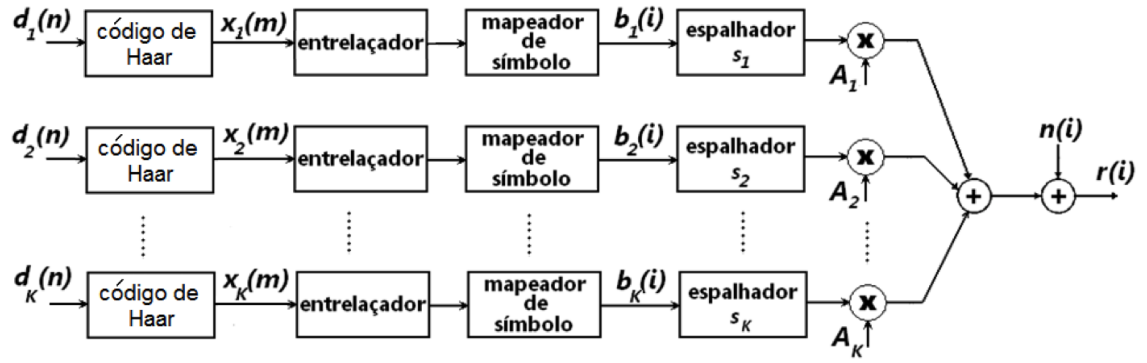
Com relação à detecção dos símbolos enviados, a Figura 6.6 mostra um gráfico de  $BER \times SNR$  (*bit error rate*  $\times$  *signal to noise ratio*) de um cenário com apenas dois usuários. Neste gráfico da Figura 6.6 existem diversos resultados de simulações de cenários diferentes. As curvas que importam para a análise no momento são a resposta do filtro casado, curva na cor roxa (de acordo com a legenda, curva do *matched-filter*), que será usada como referência nesta análise, e a curva de cor verde (as duas curvas encontram-se na parte superior do gráfico). A taxa de erro de bit do MuD bayesiano (cor verde), obtida segundo os dados coletados por simulação e colocados em um gráfico nesta figura, é satisfatória, pois para uma detecção *single-user* o filtro casado é ótimo num ambiente sem interferência. Os resultados mostram que a detecção feita pelo MuD bayesiano é satisfatória, isto é,



**Figura 6.6:** Gráfico BER x SNR.

o MuD bayesiano é um candidato a substituir o detector convencional levando-se em conta esta análise. Mas existe espaço para se melhorar ainda mais, conforme veremos com a adição do bloco das *wavelets*.

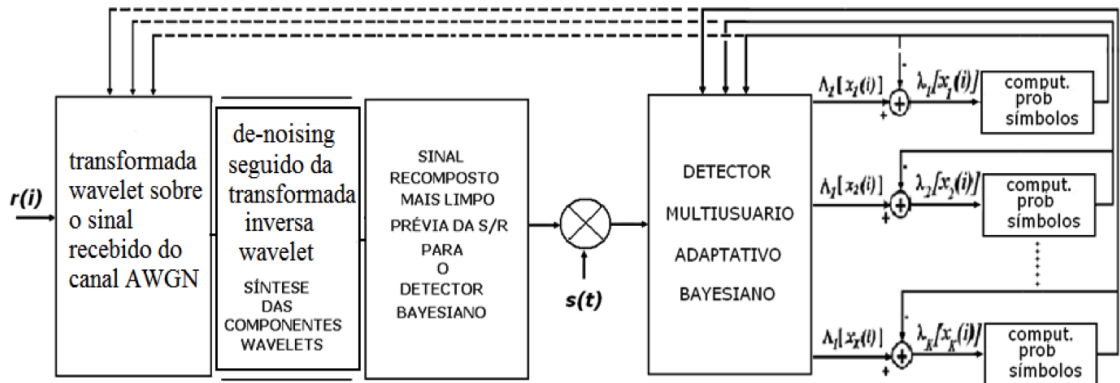
O sistema inicialmente proposto, de um detector multiusuário bayesiano, foi validado, isto é, dados de estimação e detecção foram coletados, como os apresentados nas figuras anteriores, e assim concluiu-se que o sistema poderia ser utilizado como um protótipo de teste para desenvolvimento de novas funções. O passo seguinte foi imaginar como a transformada *wavelet* poderia ser utilizada no cancelamento ou na diminuição do ruído recebido do canal. Assim foi concebida a idéia de uma modulação especial, trabalhada inicialmente no transmissor e que poderia utilizar as DWT (*Discrete Wavelet Transforms*) como ferramenta de visualização de certos padrões no sinal recebido. A idéia se concretizou com a criação do código especial de Haar e a verificação de que um sinal modulado por tal código ficaria localizado no espectro. Sabendo-se a localização do sinal no espectro pode-se cancelar faixas de frequências que possuem somente ruído. O objetivo então é preparar o sinal a ser transmitido, de forma similar a uma codificação FEC (*Forward Error Correction*), e o restante do trabalho é implementado no receptor, durante o processo de detecção. O diagrama em blocos do transmissor modificado passou a ter a seguinte configuração, conforme a Figura 6.7.



**Figura 6.7:** Transmissor modificado com a inclusão do código de Haar no lugar do bloco codificador de canal.

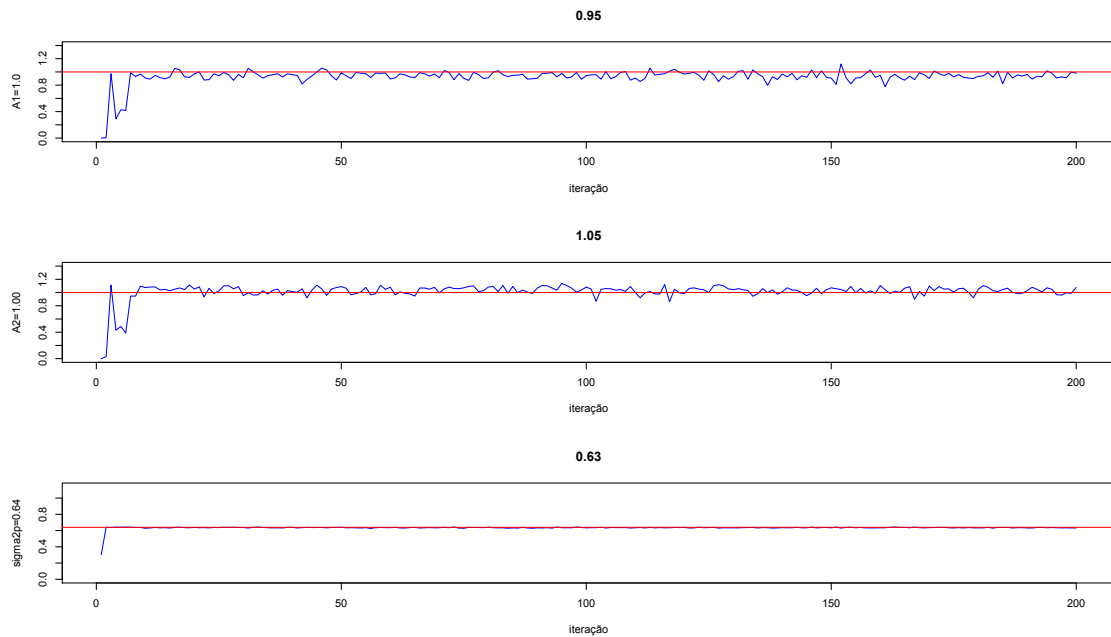
O processo de modulação com o código de Haar pode ser entendido como a adição de uma “informação” redundante, ou máscara, ao sinal que se torna “protegido” contra a ação do ruído do canal.

Para que esta informação redundante, adicionada ao sinal a ser transmitido, pudesse ser explorada no destino, o receptor protótipo inicial precisou ser modificado. Essa modificação ocorreu com a adição do bloco de processamento por *wavelets*. O diagrama em blocos do receptor modificado é dado pela Figura 6.8.



**Figura 6.8:** Receptor modificado apresentando o processamento por *wavelets*.

No receptor, que apresenta agora a nova configuração, ocorre o processamento por transformada *wavelet* e são obtidos, assim, os coeficientes *wavelets* do sinal recebido. Procede-se ao cancelamento do ruído e, após, à transformada *wavelet* inversa para recuperar o sinal. Em virtude dessa operação afetar somente o ruído, o sinal após o *denoising* melhora a sua energia, comparativamente à energia do ruído, que foi atenuada. Um sinal menos ruidoso contribui melhor com os processos de detecção e de estimação. As Figuras 6.9 e 6.10 mostram duas fases



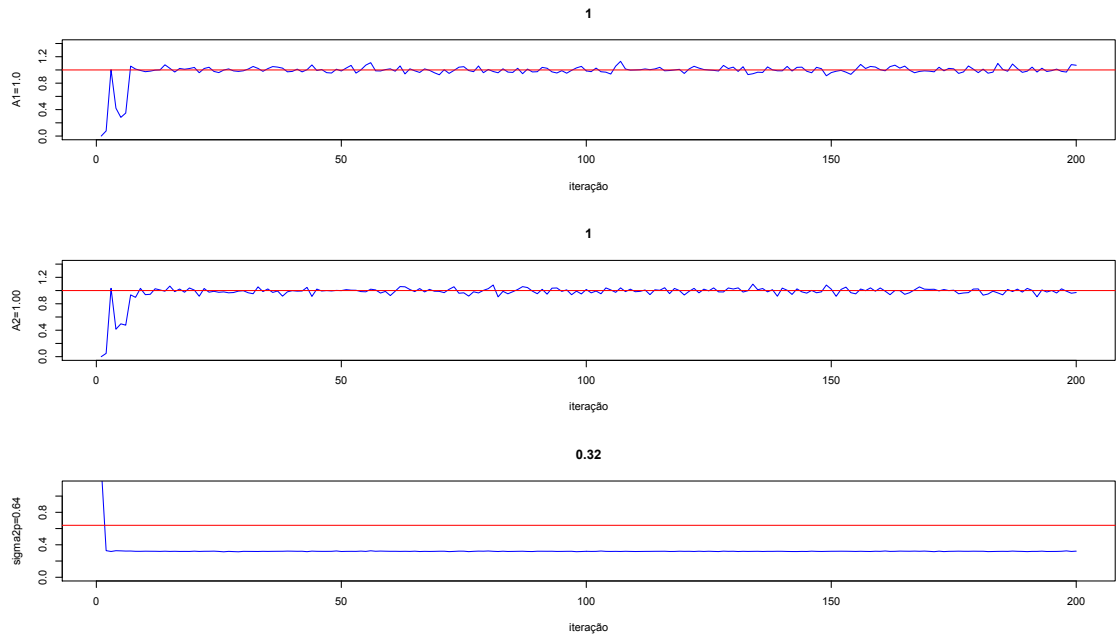
**Figura 6.9:** Gráfico das amplitudes de dois usuários e da variância do ruído, sem a aplicação do *denoising*.

do processamento do *denoising*, antes e depois de ocorrer o cancelamento do ruído. Pode-se perceber claramente a diminuição da variância do ruído, que caiu pela metade após a ação do *denoising*.

Uma outra maneira de se visualizar a ação do *denoising* é olhando-se para os coeficientes *wavelets*, obtidos da aplicação da transformada *wavelet* discreta sobre o sinal recebido. A Figura 6.11 apresenta os coeficientes *wavelets* resultantes dessa aplicação da transformada sobre um sinal de espalhamento espectral. A ocupação de todos os níveis na figura pelos coeficientes *wavelets* indica que o sinal tem uma faixa espectral larga.

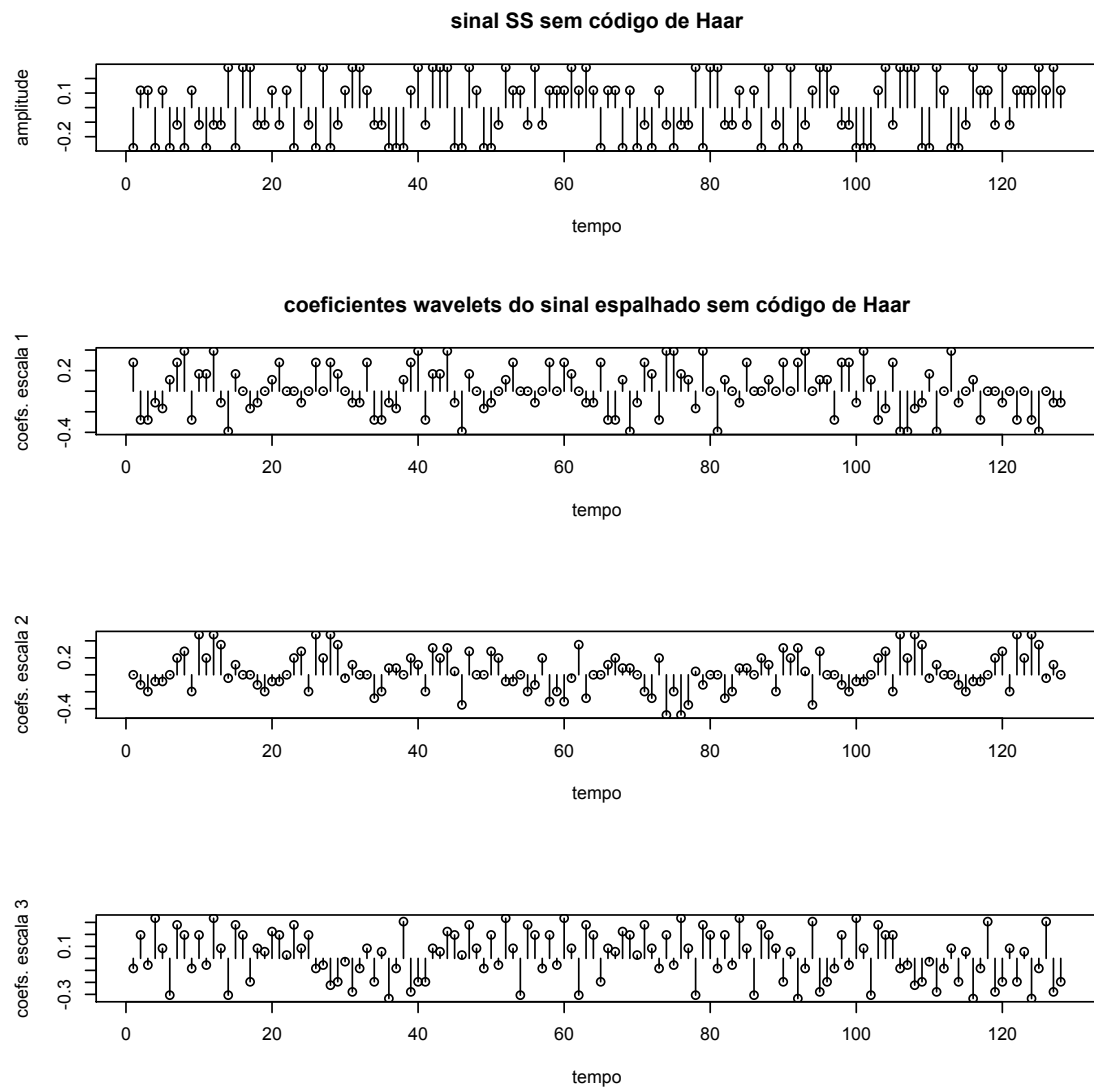
Se o transmissor utilizado para enviar o sinal de espalhamento espectral possui o código de Haar, conforme Figura 6.7, os coeficientes *wavelets* obtidos dos sinais desse transmissor agora possuirão a propriedade de restringirem suas localizações a faixas específicas do espectro, que no caso analisado correspondem à escala de nível  $m = 1$  de decomposição *wavelet*, como mostrado na Figura 6.12. Nota-se que as escalas superiores de coeficientes *wavelets* estão vazias, isto é, os coeficientes *wavelets* das escalas 2, 3 e 4 têm valor zero. O sinal está concentrado nos coeficientes *wavelets* da escala 1.

Continuando com a análise, se agora, ao invés de um sinal de espalhamento espectral, modulado pelo código de Haar mas sem ruído, chegasse ao receptor um sinal corrompido por ruído, e sobre este sinal fosse aplicada uma transformada *wavelet*, as escalas vistas anteriormente na Figura 6.12, que não estavam ocupadas,

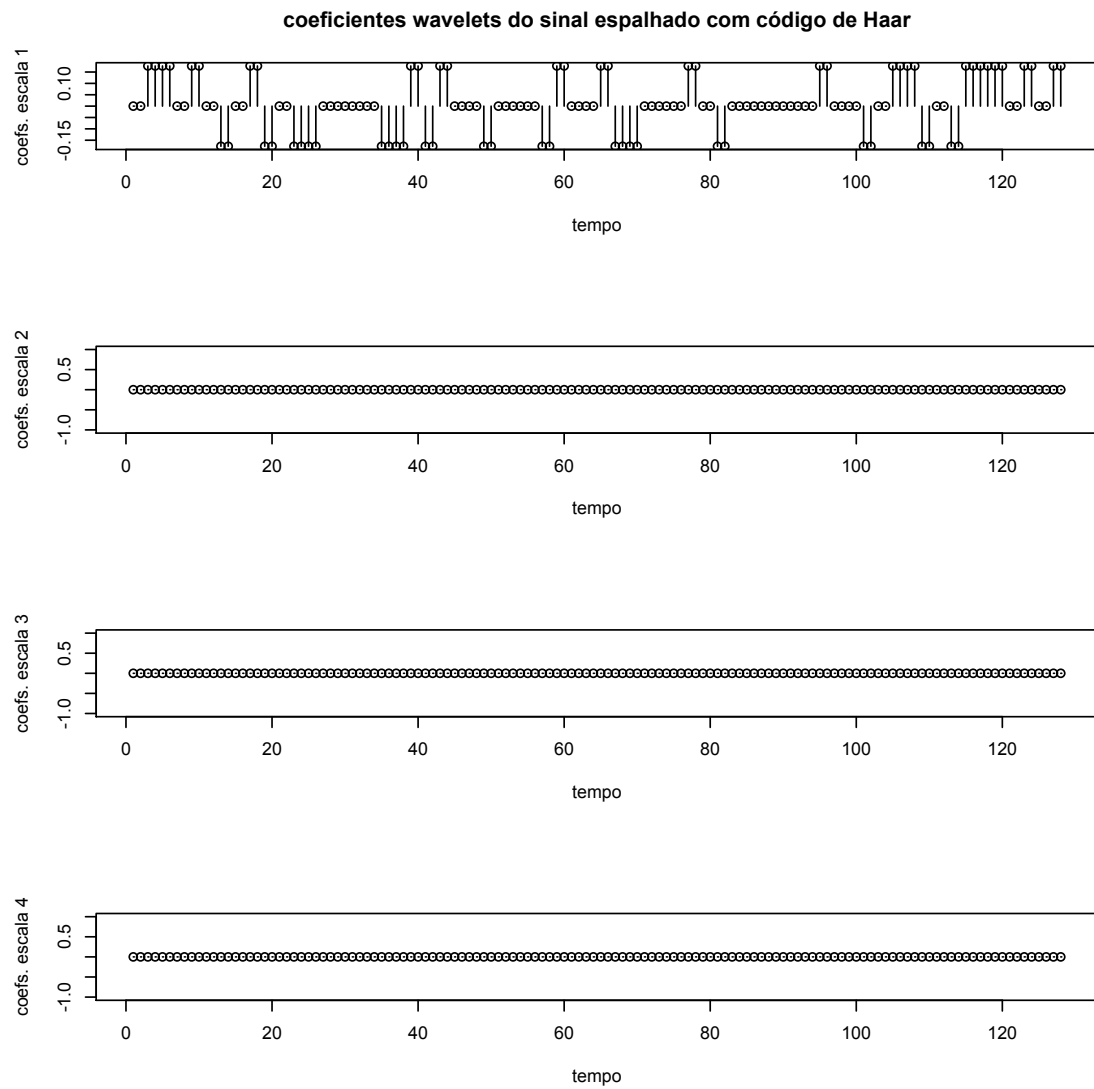


**Figura 6.10:** Resultado da ação do *denoising* sobre um sinal composto recebido do canal sem fio, a energia do ruído caiu pela metade.

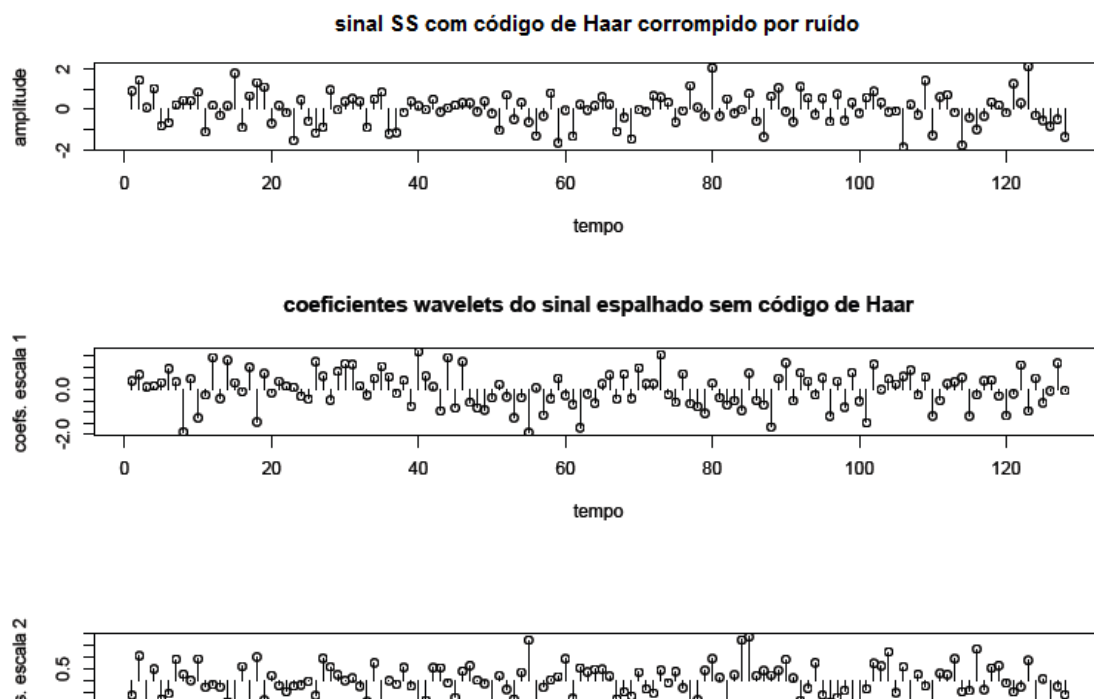
passariam a ser preenchidas por ruído puro do canal, pois o sinal de informação está limitado à primeira escala. E essa condição pode ser facilmente verificada bastando para isso que se aplique a transformada *wavelet* a este sinal composto recebido do canal. Esta nova composição das escalas *wavelets* está exemplificada na Figura 6.13.



**Figura 6.11:** Coeficientes da transformada *wavelet* aplicada a um sinal de espalhamento espectral.



**Figura 6.12:** Coeficientes da transformada *wavelet* aplicada a um sinal de espalhamento espectral com código de Haar; os coeficientes das escalas 2, 3 e 4 têm valor zero.



Portanto, a partir da simples verificação de quais escalas estão “preenchidas” com ruído puro, pode-se proceder ao cancelamento dos coeficientes *wavelets* do ruído, e portanto o receptor passará a ter menos energia de ruído. A seguir, pode-se aplicar a transformada inversa aos coeficientes *wavelets* que não foram cancelados, para recuperar o sinal de informação. Este sinal resultante, que será entregue ao bloco seguinte do detector bayesiano, possuirá uma relação sinal ruído mais adequada ao processamento de tomada de decisão do receptor.

### 6.2.2 Robustez ao efeito *near-far*

O efeito *near-far* basicamente é o cancelamento de um sinal mais fraco por um interferente de maior intensidade. Este efeito é muito prejudicial aos usuários de sistemas que utilizam o detector convencional, que não tem como lidar com tal efeito. Mas os detectores multiusuários são em geral robustos ao tal efeito *near-far*. O objetivo desta subseção é mostrar como o detector MuD bayesiano naturalmente também tem essa robustez. Foram feitos testes de comparação com outros detectores, e os resultados foram anotados e apresentados graficamente, conforme a Figura 6.14. No mesmo gráfico são apresentadas curvas de outros detectores, obtidas a partir das fórmulas dadas em [75]. Os detectores são o ótimo; o descorrelacionador; o filtro casado; e o MMSE (*minimum mean squared error*).

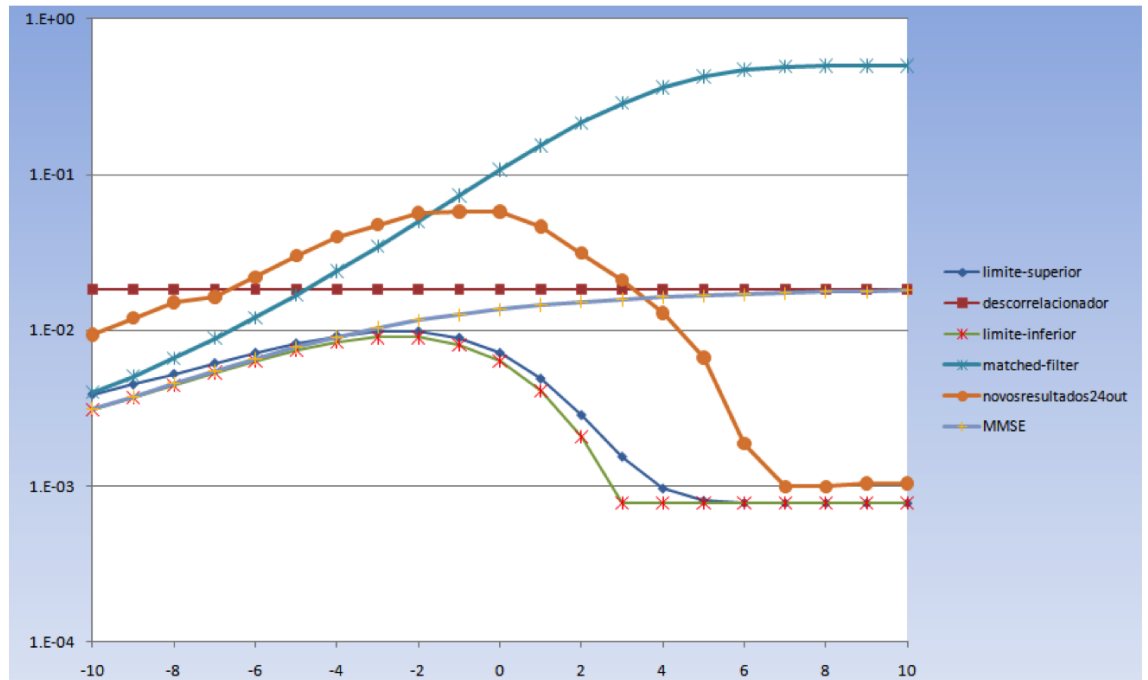
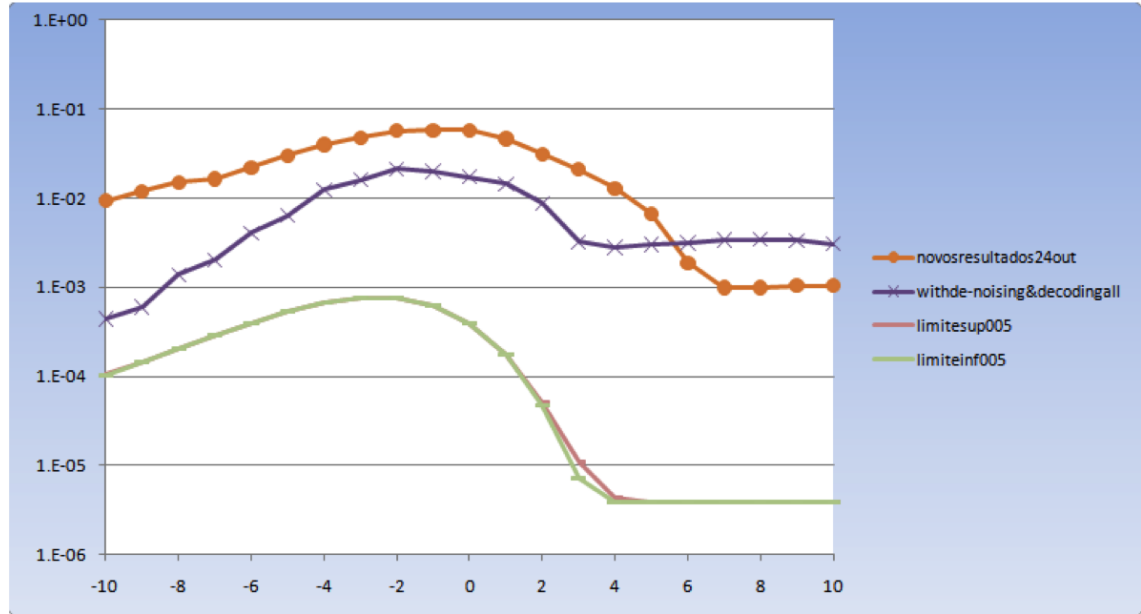


Figura 6.14: BER  $\times$  razão *near-far*  $\frac{A_2}{A_1}$ , variância = 0, 1.



A robustez ao efeito *near-far* é característica também do detector MuD bayesiano, conforme pode-se compreender pela análise da Figura 6.14. Nesta Figura 6.14 estão representados os gráficos de alguns detectores, e os seus desempenhos podem ser comparados ao do ótimo, que foi obtido pela Equação (6.47) de [75]. Neste caso, o cenário é composto por dois usuários com correlação cruzada igual a 0,75. O valor da SNR (*signal to noise ratio*) do usuário desejado é igual a 10dB. A probabilidade de erro do detector MuD bayesiano está mostrada em função da razão de *near-far*  $A_2/A_1$ , e é comparada com a probabilidade do descorrelacionador, e também com a do detector ótimo. Os resultados demonstram que o detector MuD bayesiano tem desempenho superior ao do descorrelacionador na região de baixa interferência, porém pior do que os dos outros detectores. No entanto, para valores de interferência baixos, ele se aproxima do detector ótimo mais do que os outros detectores. Para um sistema onde todos são mantidos com a mesma amplitude o detector bayesiano tem um comportamento inferior aos dos outros detectores analisados.

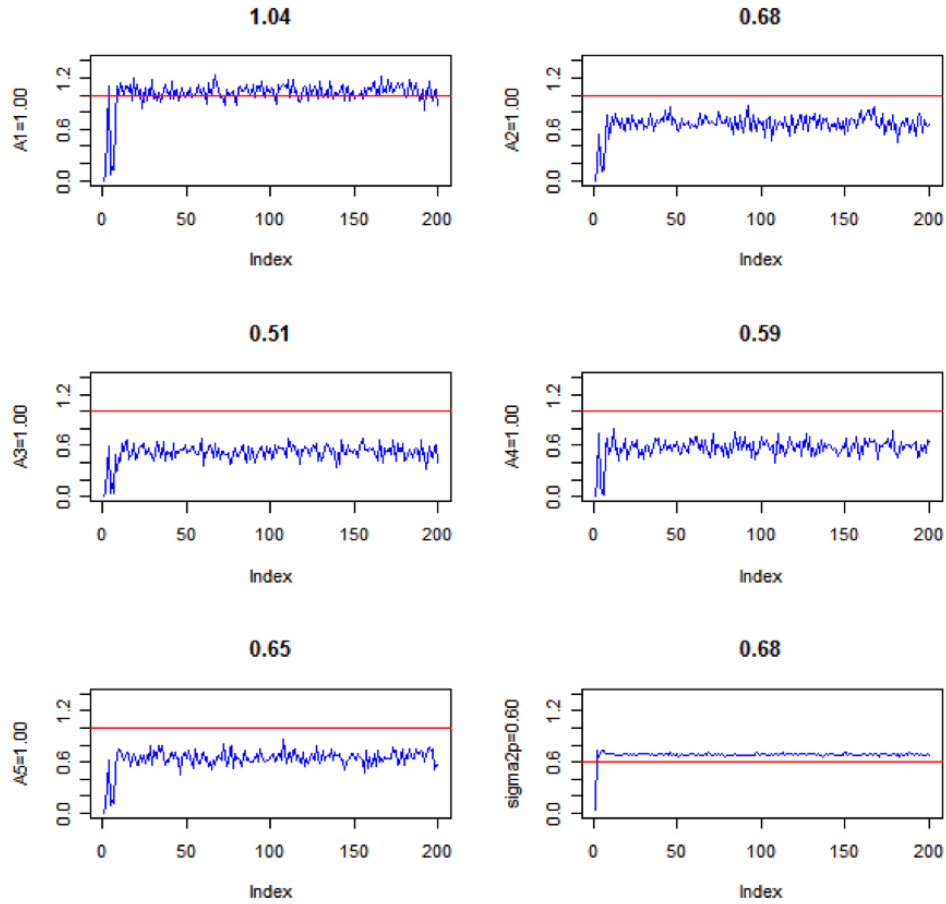
A Figura 6.15 mostra os resultados obtidos por simulação do efeito *near-far*, para um valor de variância menor do que no caso anterior; neste cenário em questão a aplicação do *denoising* está sendo utilizada. Portanto, o valor da variância diminuiu para metade do valor anteriormente analisado. As curvas mantiveram-se com seus formatos típicos, porém ficaram mais distantes das curvas do detector ótimo. A curva roxa, entre as curvas laranja e curvas do detector ótimo, foi obtida considerando-se a aplicação do *denoising* ao sinal e, no processo de “demodulação” dos símbolos  $b_k(i)$ , ou separação do código de Haar do dado original, foi executado um procedimento de “decodificação” dos símbolos, utilizando uma tabela da verdade, já que o código de Haar é na realidade associado ao dado (bit) por um processo de codificação, e naturalmente pode-se obter o dado original invertendo-se o processo, decodificando-se a palavra, e obtendo-se assim um pequeno ganho a mais no processo de recepção como um todo.



**Figura 6.15:** BER  $\times$  razão *near-far*  $\frac{A_2}{A_1}$ , variância = 0,05.

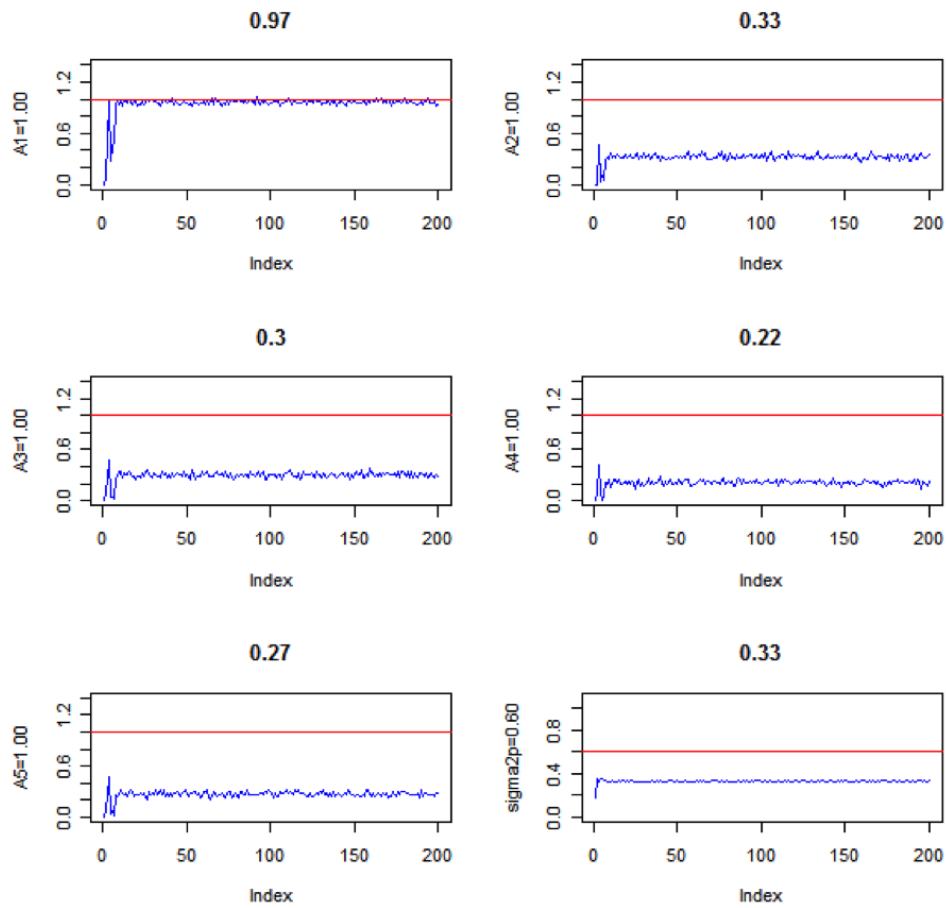
### 6.2.3 Caso assíncrono

O comportamento do detector multiusuário bayesiano também foi analisado para o cenário assíncrono, com cinco usuários, todos independentes entre si. Os resultados coletados dos testes estão apresentados graficamente na Figura 6.16. As amplitudes configuradas para os assinantes nestes testes são todas iguais a 1,0. O que pode ser observado no gráfico é que devido à falta de sincronismo entre os usuários no sistema, a estimação gerou bons resultados somente para o usuário desejado (detecção sem uso das *wavelets*).



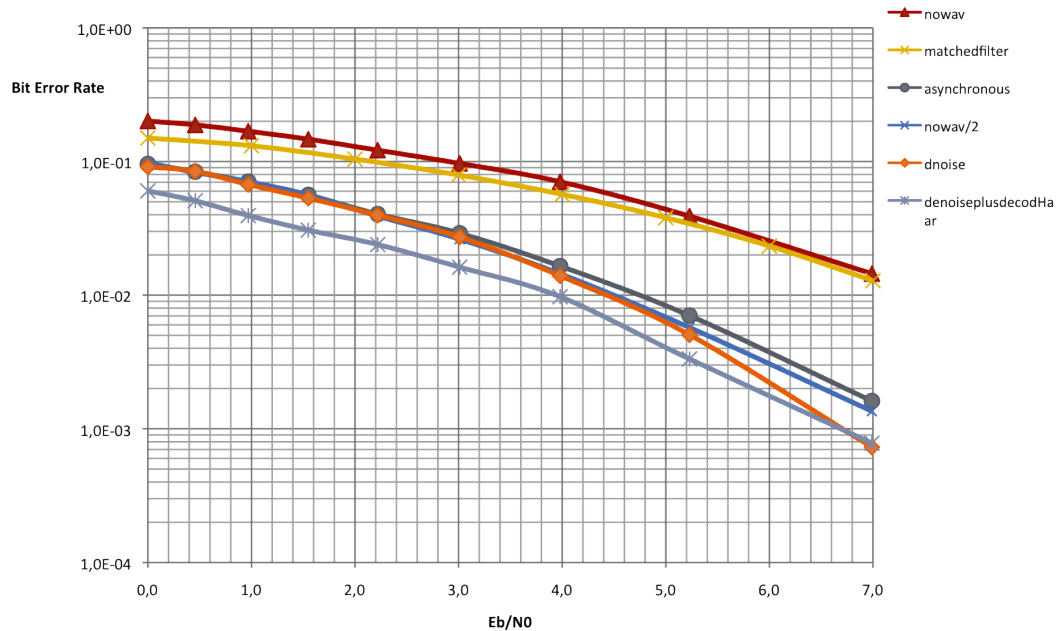
**Figura 6.16:** Modo assíncrono, estimação das amplitudes de cinco usuários e da variância do ruído.

Na Figura 6.17 estão registrados os resultados das simulações com cinco usuários, cenário similar ao anterior, porém utilizou-se neste caso o bloco das transformadas *wavelets* e a aplicação do *denoising*. Pode-se observar uma certa melhoria da estimação da amplitude do usuário desejado, comparativamente ao cenário anterior, cuja causa principal deve ser devido ao ruído que passou a ter menor intensidade.



**Figura 6.17:** Modo assíncrono, estimação das amplitudes de cinco usuários e da variância do ruído, com aplicação do *denoising*.

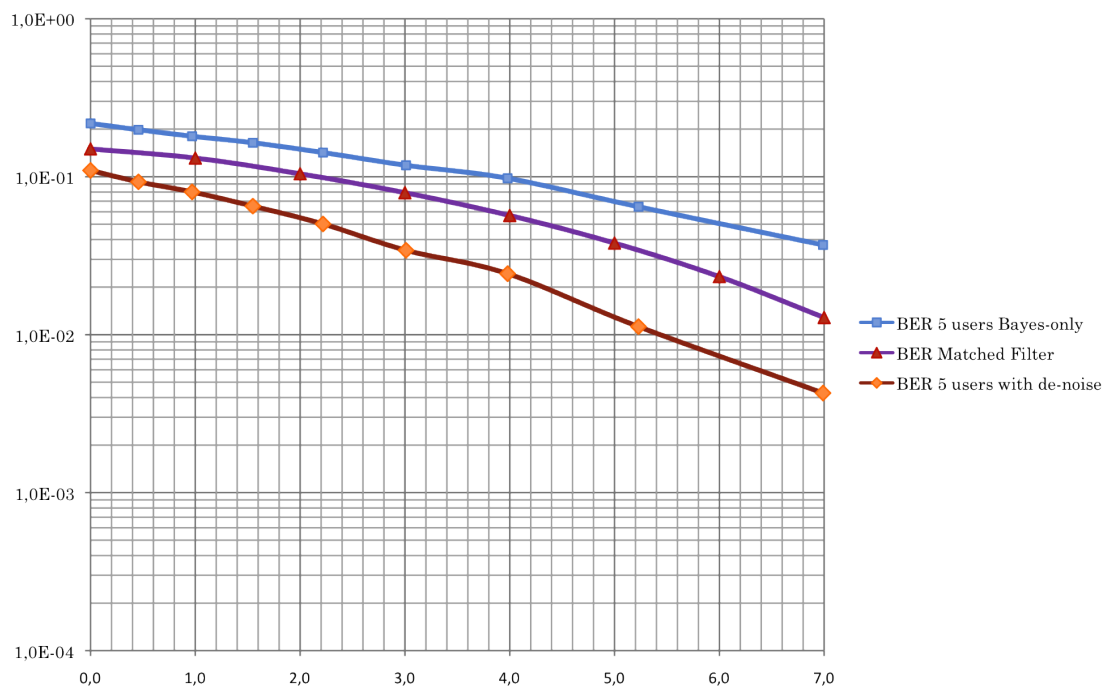
Os gráficos das Figuras 6.18 e 6.19 já foram vistos anteriormente na Figura 6.6. O fato novo que trazem esses gráficos são os resultados coletados de um sistema assíncrono. Na Figura 6.18, que mostra os resultados das simulações para dois usuários, e seguindo a legenda de cores do gráfico, tem-se:



**Figura 6.18:** Modo assíncrono, estimação das amplitudes de dois usuários e da variância do ruído, comparação com o filtro casado.

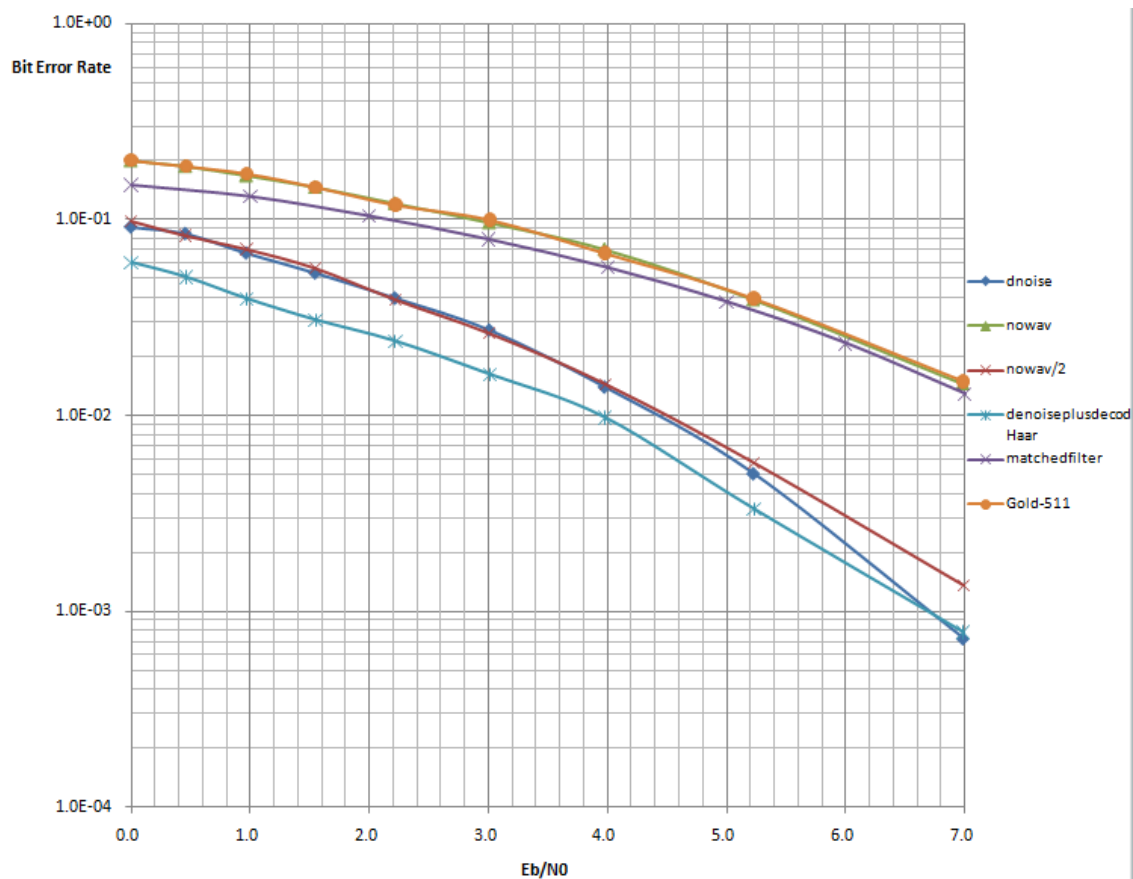
- a curva na cor vermelha representa os resultado de  $BER \times SNR$  para o detector bayesiano, não utilizando o algoritmo de *denoising*;
- a curva na cor laranja representa a resposta do filtro casado;
- a curva na cor azul é a curva o detector bayesiano cujos valores de amplitudes configurados são duas vezes menores do que no caso da cor vermelha;
- a curva na cor abóbora é a curva do detector multiusuário bayesiano com *wavelets* e com processamento de *denoising*;
- a curva na cor cinza claro é o resultado da simulação com *denoising* e adicionalmente a utilização de detecção do símbolo, quando do mesmo for retirado o código de Haar.
- finalmente, a curva na cor cinza escuro é a curva resultado da simulação no modo assíncrono. Pode-se perceber pelos gráficos que a falta de sincronismo não afetou muito a operação do sistema, e, portanto, este resultado indica que o sistema poderia ser uma opção para implementação real. Cabe aqui uma observação, esta análise e conclusão referem-se somente à configuração do sincronismo entre usuários do sistema, no caso de sistemas mais complexos, como é o caso de um sistema sem fio com desvanescimento ou com multipercurso, a análise exigiria

um modelo matemático muito mais elaborado, de solução ainda mais difícil, e diversos outros testes adicionais. Este cenário não foi investigado neste trabalho. Portanto, a curva que representa este novo cenário, do modo assíncrono, está representada pelo gráfico de cor cinza, na Figura 6.18 (o *denoising* foi executado quando da obtenção desta curva). Na Figura 6.19 são apresentados os resultados de simulações para um cenário com cinco usuários, modo assíncrono; o *denoising* foi novamente utilizado aqui neste caso. Pode-se notar que entre os gráficos com dois e com cinco usuários as características dos resultados não se alteraram.



**Figura 6.19:** Modo assíncrono, estimação das amplitudes de cinco usuários e da variância do ruído, comparação com o detector convencional.

Foram realizados testes com diferentes tipos de códigos de espalhamento. O resultado BER x SNR para um código Gold-512 é apresentado na Figura 6.20. Nesta figura a curva referente ao código Gold é a curva na parte mais alta do gráfico. Os resultados dos testes demonstraram que o Gold-512 é um código cujas características são muito boas para uso em sistemas de espalhamento espectral. Ele tem características boas de correlação cruzada e auto-correlação



**Figura 6.20:** Resultado do teste com o Gold-512, curva superior do gráfico.

## 6.3 Considerações finais

Foram observados, neste capítulo, a detecção dos sinais e a estimação dos parâmetros: amplitude do sinal e variância do sinal. A taxa de erro de bit foi medida e fez-se comparações com as taxas do filtro casado. Os resultados foram melhores que os do detector convencional. A taxa de erro de bit verificada para vários cenários de testes também foi satisfatória, pois os erros foram da ordem de aproximadamente 1 em  $10^2$  ou mesmo 1 em  $10^3$ . O tamanho do quadro também é um parâmetro que pode impactar nos resultados finais; nos testes executados, diversos tamanhos de quadro foram analisados, sendo o tamanho máximo considerado igual a 512 símbolos. A Tabela 6.2 apresenta resultados para quadros de tamanho igual a 128. Não ocorreu no entanto nenhum caso de atraso que afetasse o resultado final. A análise do modo assíncrono é importante pois é o cenário mais comum dos sistemas existentes. Conforme foi possível verificar, no modo assíncrono, o sinal de interesse foi recuperado, embora as estimativas dos sinais interferentes tenham sido prejudicadas. O efeito *near-far* foi analisado e concluiu-se que o sistema é resistente ao mesmo. Os resultados de estimação e detecção foram considerados muito satisfatórios.



## 7 Conclusões e Trabalhos Futuros

Este Capítulo resume as principais conclusões do trabalho desenvolvido e apresenta sugestões para pesquisas futuras.

Foi realizada uma aplicação inovadora de *wavelets* em sistemas de espalhamento espectral para uso em sistemas de detecção multiusuário com inferência bayesiana, incluindo um novo modelo de cancelamento de ruído (em inglês *denoising*). Inicialmente, foi feita uma breve exposição dos fundamentos teóricos necessários para o desenvolvimento do trabalho. A seguir, apresentou-se o desenvolvimento do detector multiusuário bayesiano com cancelamento de ruído; o algoritmo escolhido para a detecção bayesiana emprega o Amostrador de Gibbs e o cancelamento de ruído é efetuado empregando-se a transformada *wavelet*. Por fim, foram apresentados resultados de simulação que validaram a aplicação. No modelo de cancelamento de ruído, atuou-se tanto na transmissão quanto na recepção, respeitando as afirmações de Shannon [77], segundo as quais, para se transmitir uma informação com probabilidade de erro mínima em um canal com ruído aditivo, deve-se codificar o sinal na transmissão e recuperá-lo por processo inverso na recepção. A *wavelet* escolhida foi a de Haar por ter um comportamento que permite a geração imediata de códigos bipolares, como convencionalmente utilizado em espalhamento espectral de sinais. Para a transmissão do sinal, códigos baseados nessa *wavelet* foram propostos, neste trabalho. Ao processo de codificação deu-se o nome simplificado de modulação *wavelet* [78]. Na recepção, a decodificação utiliza a transformada *wavelet* de Haar, que simplifica consideravelmente a análise, embora qualquer outra *wavelet* pudesse ser utilizada, porém com maior complexidade. Foi mostrado que devido à ortogonalidade da função empregada, os coeficientes *wavelet* do conjunto de dados modulados são nulos, com exceção dos que estão na escala determinada pelo código escolhido. Isto garante uma melhor relação sinal ruído na entrada do bloco de inferência bayesiana do detector multiusuário, diferentemente do que acontece no processamento digital executado em outras áreas como a de processamento de imagens, onde

não se atua na geração do sinal. Diversas simulações empregando o programa R foram realizadas no desenvolvimento deste trabalho. Resultados obtidos para os algoritmos empregando o Amostrador de Gibbs e o Metropolis-Hastings para detecção e estimação demonstraram serem os mesmos, equivalentes quanto à convergência e à velocidade. A opção pelo Amostrador de Gibbs deu-se em função de sua maior simplicidade. A solução proposta incluindo o cancelamento de ruído foi também validada através de resultados obtidos por simulação. A detecção dos sinais foi efetivada com sucesso, sendo que a variância do ruído e a amplitude do sinal estimadas no processo de detecção estão em excelente conformidade com os valores configurados *a priori*. Os objetivos propostos de investigação do efeito *near-far*, do caso assíncrono e de testes de outras sequências de espalhamento, como o código de Gold, foram alcançados. Os resultados obtidos sobre o efeito *near-far* demonstraram que a robustez e o desempenho do detector desenvolvido são melhores que os do detector convencional do tipo filtro casado. Os testes em modo assíncrono mostraram que a aplicação funciona mesmo em um ambiente de tempos de acesso totalmente aleatórios. Os resultados de testes com outros códigos de espalhamento não demonstraram dependência forte do tipo de código de espalhamento utilizado. Dos resultados apresentados, verifica-se uma redução da potência do ruído de pelo menos 3 dB em relação ao valor sem cancelamento de ruído. Após o processamento do cancelamento do ruído, a detecção pode ser realizada sobre os símbolos com menor probabilidade de erro e maior confiança na recuperação da informação enviada.

Os resultados obtidos nesta pesquisa são bastante promissores no que se refere a aplicação de *wavelets* em sistemas de comunicação. Em decorrência desses resultados novas linhas de pesquisa tornaram-se viáveis para futuros trabalhos. Por exemplo, seria importante investigar o uso de outras transformadas *wavelets* na recepção, visando a melhoria da qualidade de recepção incluindo a redução da interferência de acesso múltiplo. Novos códigos baseados em outras *wavelets* poderiam ser desenvolvidos visando a otimização do processo de cancelamento de ruído. O impacto da técnica proposta poderia ser avaliado em sistemas 3G, 4G e com outras modulações. Também a aplicação de *wavelet packet* poderia ser explorada em sistemas de comunicação sem fio em geral.

# Referências

- [1] RESENDE, J. Desempenho e panorama de servicos. *ANATEL*, v. 15, p. 15–71, 2012.
- [2] HONIG, M. *Advances in Multiuser Detection*. Hoboken, NJ: Wiley, 2009.
- [3] LIU, J.; CHEN, R. Blind deconvolution via sequential imputations. *Journal of American Statistical Association*, v. 90, p. 567–576, 1995.
- [4] LIN, L.; ZHOU, W.; QIU, P. Wavelet multiuser detector for asynchronous ds-cdma systems. *IEEE 6th CAS Symp. Emerging Technologies: Mobile and Wireless Comm.*, v. 46, p. 231–233, 2004.
- [5] DONOHO, D. De-noising by soft thresholding. *IEEE Trans. Inform. Theory*, v. 41, p. 613–627, 1995.
- [6] BLISS, D.; GOVINDASAMY, S. *Adaptive wireless Communications MIMO channels and networks*. New York, NY: Cambridge University Press, 2013.
- [7] HAYKIN, S. *Adaptive Filters*. Upper Saddle River, NJ: Prentice Hall, 2013.
- [8] POOR, H.; WANG, X. Blind equalization and multiuser detection for cdma communications in dispersive channels. *IEEE Transactions on Communications*, v. 46, p. 91–103, 1998.
- [9] WANG, X.; CHENG, R. Adaptive bayesian multiuser detection for synchronous CDMA with gaussian and impulsive noise. *IEEE Transactions on Signal Processing*, v. 47, p. 2013–2028, 2000.
- [10] HAYKIN, S.; MOHER, M. *Modern wireless communications*. Upper Saddle River, NJ: Prentice Hall, 2004.
- [11] PROAKIS, J. *Digital Communications*. New York, NY: McGraw-Hill, 2001.
- [12] GOLDSMITH, A. *Wireless communications*. New York, NY: Cambridge University Press, 2005.
- [13] VERDÚ, S. *Optimal multiuser signal detection*. Tese (Doutorado) — University of Illinois, Urbana, june 1984.
- [14] ABDULRAHMAN, A.; FALCONER, D.; SHEIKH, A. Decision feedback equalization for cdma in indoor wireless communications. *IEEE J. Select. Areas Commun.*, v. 12, p. 698–706, 1994.
- [15] HONIG, M. Adaptive linear interference suppression for packet ds-cdma. *European Transactions on Telecommunications*, v. 9, p. 944–960, 1998.
- [16] MILLER, S. An adaptive direct-sequence code-division multiple access receiver for multiuser interference rejection. *IEEE Transactions on Communications*, v. 43, p. 1746–1755, 1995.

- [17] RAPAJIC, P.; VUCETIC, B. Adaptive receiver structures for asynchronous cdma systems. *IEEE J. Select. Areas Communications*, v. 12, p. 685–697, 1994.
- [18] MITRA, U.; POOR, H. Adaptive receiver algorithm for near-far resistant cdma. *IEEE Transactions on Communications*, v. 43, p. 1713–1724, 1995.
- [19] HONIG, M.; MADHOW, U.; VERDU, S. Blind adaptive multiuser detection. *IEEE Trans. Information Theory*, v. 41, p. 944–960, 1995.
- [20] MADHOW, U. Blind adaptive interference suppression for direct sequence cdma. *Proc. IEEE*, v. 86, p. 2049–2069, 1998.
- [21] WANG, X.; POOR, H. Blind equalization and multiuser detection for cdma communications in dispersive channels. *IEEE Trans. Communications*, v. 46, p. 91–103, 1998.
- [22] BENSLEY, S.; AAZHANG, B. Subspace-based channel estimation for code-division multiple access communication systems. *IEEE Transactions on Communications*, v. 44, p. 1009–1020, 1996.
- [23] REDD, M. et al. Iterative multiuser detection for cdma with fec: near single-user performance. *IEEE Transactions on Communications*, v. 46, p. 1693–1699, 1998.
- [24] MOHER, M. An iterative multiuser decoder for near-capacity communications. *IEEE Transactions on Communications*, v. 46, p. 870–880, 1998.
- [25] ALEXANDER, P. et al. Iterative multiuser interference reduction: Turbo cdma. *IEEE Transactions on Communications*, v. 47, p. 1008–1014, 1999.
- [26] WANG, X.; POOR, H. Iterative (turbo) soft interference cancellation and decoding for coded cdma. *IEEE Transactions on Communications*, v. 47, p. 1046–1061, 1999.
- [27] GELMAN, A.; CARLIN, J.; RUBIN, D. *Bayesian Data Analysis*. Boca Raton, FL: Chapman and Hall, 2004.
- [28] BOX, G. E.; TIAO, G. C. *Bayesian Inference in Statistical Analysis*. Englewood Cliffs, NJ: Addison-Wesley Publishing, 1973.
- [29] DEGROOT, M. *Optimal statistical decisions*. New Jersey, NJ: Wiley, 1970.
- [30] CARLIN, B.; LOUIS, T. *Bayesian Methods for Data Analysis*. Boca Raton, FL: Chapman and Hall CRC, 2008.
- [31] PEARLMAN, W. *Wavelet image compression*. Austin, TX: Morgan and Claypool, 2013.
- [32] OUAHABI, A. *Signal and image multiresolution analysis*. Hoboken, NJ: John Wiley & Sons, Inc., 2012.
- [33] KAISER, G. *A Friendly Guide to Wavelets*. Austin, TX: Birkhauser, 2011.
- [34] WOJTASZCZYK, P. *A mathematical introduction to wavelets*. Cambridge, UK: Cambridge University Press, 1997.

- [35] PERCIVAL, D.; WALDEN, A. *Wavelet Methods for Time Series Analysis*. Cambridge, NY: Cambridge University Press, 2000.
- [36] MORETTIN, P. *Ondas e ondaletas: da análise de Fourier à análise de ondaletas*. São Paulo, SP: Edusp, 1999.
- [37] RICKER, N. The form and nature of seismic waves and the structure of seismograms. *Geophysics*, v. 5, p. 348–366, 1940.
- [38] STOLLNITZ, E.; DeRose, A.; SALESIN, D. *Wavelets for computer graphics: theory and applications*. San Francisco, CA: Morgan Kaufmann, 1996.
- [39] MORLET, J. et al. Wave propagation and sampling theory. *Geophysics*, v. 47, p. 203–236, 1982.
- [40] LAKSHMANAN, M.; NIKOOKAR, H. A review of wavelets for digital wireless communication. *Springer Wireless Personal Communications*, v. 37, p. 387–420, 2006.
- [41] MEYER, Y. Wavelets: their past and their future. *Progress in wavelet analysis and applications*, v. 1, p. 9–18, 1993.
- [42] PERCIVAL, D.; MONDAL, D. Wavelet variance analysis for random fields on a regular lattice. *IEEE Trans. Image Processing*, v. 21, p. 537–549, 2012.
- [43] SPIEGEL, M. *Fourier analysis with boundary values problems*. New York, NY: McGraw Hill, 1974.
- [44] RUDIN, W. *Real and complex analysis*. New York, NY: McGraw Hill, 1966.
- [45] LIMA, A. *Contribuições à modelagem de teletráfego fractal*. Tese (Doutorado) — Escola Politécnica da USP, june 2008.
- [46] CHUI, C. *An introduction to wavelets*. New York, NY: Academic Press, 1992.
- [47] DAUBECHIES, I. *Ten lectures on wavelets*. Philadelphia, PA: SIAM, 1992.
- [48] MEYER, Y. *Wavelets and operators*. New York, NY: Cambridge University Press, 1993.
- [49] MALLAT, S.; ZHONG, S. *Wavelets and their applications*. Boston, MA: Jones and Bartlett, 1992.
- [50] MORLET, J. *Sampling theory and wave propagation, in NATO ASI Series, Vol. I, Issues in Acoustic Signal/Image Processing and Recognition*. Berlin: Springer, 1983.
- [51] GABOR, D. Theory of communication. *J. Inst. Electr. Eng.*, v. 93, p. 429–457, 1946.
- [52] DAUBECHIES, I. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, v. 41, p. 909–996, 1988.
- [53] MALLAT, S. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transaction on pattern analysis and machine intelligence*, v. 11, p. 674–693, 1989.

- [54] JAWERTH, B.; SWELDENS, W. An overview of wavelet based multiresolution analyses. *SIAM Review*, v. 36, p. 377–412, 1994.
- [55] WANG, X.; POOR, H. *Wireless Communication Systems*. Upper Saddle River, NJ: Prentice Hall, 2004.
- [56] ROBERT, C.; CASELLA, G. *Monte Carlo Statistical Methods*. New York, NY: Springer, 2004.
- [57] CHAN, K. Asymptotic behavior of the gibbs sampler. *J. Amer. Stat. Association*, v. 88, p. 320–326, 1993.
- [58] GELFAND, A.; SMITH, A. Sampling based approaches to calculating marginal densities. *J. Amer. Stat. Association*, v. 85, p. 398–409, 1990.
- [59] DONOHO, D.; JOHNSTONE, I. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, v. 81, p. 425–455, 1994.
- [60] METROPOLIS, N. et al. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, v. 21, p. 1087–1092, 1953.
- [61] HASTINGS, W. *Monte Carlo sampling methods using Markov chains and their applications*. *Biometrika*, v. 57, p. 97–109, 1970.
- [62] ROSS, S. *Probability Models*. New York, NY: Wiley, 2000.
- [63] GRIMMETT, G.; STIRZAKER, D. *Probability and Random Processes*. New York, NY: Oxford, 2005.
- [64] PAULINO, C.; TURKMAN, M.; MURTEIRA, B. *Estatística Bayesiana*. Av. de Berna, Lisboa: Calouste Gulbenkian, 2003.
- [65] GEMAN, S.; GEMAN, D. Stochastic relaxation, gibbs distribution and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 6, p. 721–741, 1984.
- [66] SMITH, A.; ROBERTS, G. Monte Carlo sampling methods using Markov chains and their applications. *J. Royal Statist. Soc.*, v. 55, p. 3–23, 1993.
- [67] DRAPER, D. *Bayesian Hierarchical Models*. New York, NY: Wiley, 2000.
- [68] MARTINSSON, P.; ROKHLIN, V.; TYGERT, M. A fast algorithm for the inversion of general Toeplitz matrices. *Elsevier Computers and Mathematics with applications*, v. 50, p. 741–752, 2005.
- [69] CORMEN, T. et al. *Algoritmos, teoria e prática*. Rio de Janeiro, RJ: Elsevier, 2012.
- [70] SHAFFER, C. A. *A practical introduction to data structures and algorithm analysis in Java*. Upper Saddle River, NJ: Prentice Hall, 1998.
- [71] DEGROOT, M.; SCHERVISH, M. *Probability and Statistics*. Berkeley, CA: Pearson, 2011.
- [72] BOLFARINE, H.; SANDOVAL, M. *Introdução à Inferência Estatística*. Rio de Janeiro, RJ: SBM, 2010.

- [73] VERDÚ, S. Minimum probability of error for asynchronous gaussian multiple-access channels. *IEEE Transactions on Information Theory*, v. 32, p. 85–96, 1986.
- [74] VERDÚ, S. Computational complexity of optimum multiuser detection. *Algorithmica*, v. 4, p. 303–312, 1989.
- [75] VERDÚ, S. *Multiuser Detection*. New York, NY: Cambridge University Press, 1998.
- [76] ALBERT, J. *Bayesian Computation with  $\mathbf{R}$* . New York, NY: Springer, 2009.
- [77] SHANNON, C. A mathematical theory of communication. *Bell Syst. Tech. Journal*, v. 27, p. 379–423, 1948.
- [78] NOGUEIRA, L.; AMAZONAS, J. Improvement of the signal to noise ratio by means of wavelet modulation. *Rev. Invest. Univ. Quindío*, v. 23, p. 56–63, 2012.
- [79] BOX, G.; MULLER, M. A note on the generation of random normal deviates. *Annals of Mathematical Statistics*, v. 29, p. 610–611, 1958.
- [80] TANNER, M. *Tools for statistical inference*. New York, NY: Springer, 1997.

# ANEXO 1

Código desenvolvido para o detector multiusuário bayesiano com aplicação de *wavelets*

```
graphics.off()    #ultima execucao 28/outubro/2009    6nov2012
rm(list=ls(all=TRUE)) # Clear workspace
library(geoR)
library(mvtnorm)
library(waveslim)
K <- 2
P <- 64
Pcod <- 4
M <- 128#4#64 #256
loopvezes <- 1#200
J0=log2(P*Pcod*M)
contapositivo1<-matrix(c(0),loopvezes)
contapositivo2<-matrix(c(0),loopvezes)
for (jota in 1:loopvezes){
  cat("  jota no.: ", jota, fill=T)
  rhokai1 <- matrix(c(0.5:0.5),1,2*M)
  rhokai2 <- matrix(c(0.5:0.5),1,2*M)
  LLR1 <- matrix(c(0:0),1,M)
  LLR2 <- matrix(c(0:0),1,M)
  LLR3 <- matrix(c(0:0),1,M)
  LLR4 <- matrix(c(0:0),1,M)
  draws1 <- NULL
  draws2 <- NULL
  draws3 <- NULL
  #drawsx1 <- NULL
  Bbind <- NULL
  Bkbind <- NULL
  Xbind <- NULL
```



```

Xkbind <- NULL
Ybind <- NULL
Yi <- matrix(0,ncol=M,nrow=Pcod*P)
codificado <- matrix(0,ncol=M,nrow=P*Pcod)
Sigmastar <- NULL
Sigma0 <- diag(c(1000,1000))
invSigma0 <- solve(Sigma0)
codigocanal1 = c(+1,-1,+1,-1)
codigocanal2 = c(-1,+1,-1,+1)
cod1 = (1/sqrt(Pcod))*codigocanal1
cod2 = (1/sqrt(Pcod))*codigocanal2
cod =matrix(c(cod1,cod2),4,2)
# s1=(1/sqrt(P))*c(1,1,1,1,1,1,-1,1,1,-1,1,-1,-1,1,-1,-1,-1,-1,1,-1,1,
# 1,-1,-1,1,-1,1,-1,1,-1,-1,1,-1,-1,1,1,1,1,-1,-1,-1,-1,-1,
# 1,1,-1,1,1,1,-1,-1,1,1,-1,-1,-1,1,1,1,-1,1,-1,1) # Gold-64
# s2=(1/sqrt(P))*c(-1,1,1,-1,1,-1,-1,-1,1,-1,-1,-1,1,-1,1,-1,
# 1,-1,-1,1,-1,1,-1,1,-1,-1,1,-1,-1,1,1,1,1,-1,-1,-1,-1,-1,
# 1,1,-1,1,1,1,-1,-1,1,1,-1,-1,-1,1,1,1,-1,1,-1,1,1,1,1,1) # Gold-64
s1=(1/sqrt(P))*c(+1,1,1,1,1,1,-1,1,1,-1,1,-1,-1,1,-1,-1,
-1,-1,-1,1,-1,1,1,-1,-1,1,-1,1,-1,1,-1,-1,
1,-1,-1,1,1,1,-1,-1,-1,-1,-1,1,1,-1,1,
1,1,-1,-1,1,1,-1,-1,-1,1,1,1,-1,1,-1,1) # Gold-64
s2=(1/sqrt(P))*c(-1,1,1,-1,1,-1,-1,-1,1,-1,-1,-1,1,-1,-1,
1,1,-1,-1,1,-1,1,-1,-1,-1,1,-1,-1,1,1,1,
1,-1,-1,-1,-1,-1,1,1,-1,1,1,1,-1,-1,1,1,-1,
-1,-1,1,1,1,-1,1,-1,1,1,1,1,1,1,1) # Gold-64

S <- matrix(c(s1,s2),P,K)
ST <- t(S)
R <- ST%*%S

nuzero <- 1
lambdazero <- 0.1
sigma2p <- rinvcchisq(1,df=nuzero,scale=lambdazero)
A1 <- 0.63
A2 <- 1.58
A <- diag(c(A1,A2))
Ak <- matrix(c(0,0,0,0),2)

```

```

repeat{
amu0 <- t(rmvnorm(1, mean=c(1,1),sigma=Sigma0))
if(all(amu0>0))
break}
amu <- matrix(c(0,0),2)
x  <- matrix(0,ncol=M,nrow=2)
Bi <- array(0,c(2,2,M))
Bki <- array(0,c(2,2,M))
xk <- matrix(0,ncol=M,nrow=2)
for (k in 1:2){
for (m in 1:M){
xk[k,m] <- 2*rbinom(1,1,0.5)-1
}
}
for (k in 1:2){
for (m in 1:M){
x[k,m]  <- 2*rbinom(1,1,0.5)-1
}
}
for (i in 1:M){
Xbind <- rbind(Xbind,x[1,i],x[2,i])
Xkbind <- rbind(Xkbind,xk[1,i],xk[2,i])
}
Xi <- array(Xbind,c(2,1,M))
Xki <- array(Xkbind,c(2,1,M))
for (i in 1:M){
Bi[, ,i] <- diag(c(x[1,i],x[2,i]))
Bki[, ,i] <- diag(c(xk[1,i],xk[2,i]))}
awgn<-matrix(c(0:0),P*Pcod,M)
#for (i in 1:M){
sigma2dB <- 0.501187#-2
sigma2 <- 0.63 # 10^(sigma2dB/10)
#####
# sigma2p = sigma2  #!!!!somente para teste !!! comentar
#####          #
sigma1<-sqrt(sigma2)
for (i in 1:M){
awgn[,i] <- rnorm(Pcod*P,0,sigma1)}

```

```

#Scod = matrix(as.vector(cod1%*%t(as.vector(S))),P*Pcod,K)
#Scod1=as.vector(cod1%*%t(ST[1,]))
#Scod2=as.vector(cod2%*%t(ST[2,]))
Scod1=as.vector(kronecker(cod1,(S[,1])))
Scod2=as.vector(kronecker(cod2,(S[,2])))
Scod=matrix(c(Scod1,Scod2),P*Pcod,K)
STcod=t(Scod)
R=STcod%*%Scod
#-----
for (i in 1:M){
  codificado[,i] <- Scod%*%A%*%Xi[,i]
  Yi[,i] <- Scod%*%A%*%Xi[,i] + awgn[,i]
}
for (i in 1:M){
  Ybind <- rbind(Ybind,Yi[,i])
}
Y <- array(t(Ybind),c(P*Pcod,1,M))
#=====
#
# Wavelets dwt
codificado.haar <- dwt(codificado,'haar',J0)
Yi.haar <- dwt(Yi, 'haar', J0)
rizero=matrix(0,P*Pcod,M)
ondarec2.haar <- dwt(rizero,'haar',J0)
awgn.haar <- dwt (awgn,'haar',J0)
#X11()
#plot.ts(as.vector(Yi), axes=TRUE, ylab="", main="spread & coded signal")
#X11()
#par(mfrow=c(4,1))
#for(i in 1:4)
#plot.ts(up.sample(Yi.haar[[i]],2^i), type="h", axes=TRUE)
##### Denoising
#
  for (i in 2:J0) {Yi.haar[[i]]=ondarec2.haar[[i]]}
#
#####
#X11()
#par(mfrow=c(4,1))

```

```

#for(i in 1:4)
#plot.ts(up.sample(Yi.haar[[i]],2^i), type="h", axes=TRUE)

#===== Transformada inversa
Yi=idwt(Yi.haar)
Yi <- matrix(c(Yi),ncol=M,nrow=Pcod*P)
for (i in 1:M){
Ybind <- rbind(Ybind,Yi[,i])
}
Y <- array(t(Ybind),c(P*Pcod,1,M))
#-----
#=====#
# Function(s): rtnorm(), rtmvnorm()
# Purpose: generates random numbers from univariate and multivariate #
#           truncated Normal distribution
# Arguments: 'n' the number of realizations, 'mu' the mean vector,      #
#           'sigma' the standard deviation, 'Sigma' the covariance      #
#           matrix, 'lo' the lower truncation limit, 'up' the upper     #
#           truncation limit
# Author: Dimitris Rizopoulos                                         #
# Last update: 2005-08-13                                             #
#####
rtnorm <- function(n, mu = 0, sigma = 1, lo = -Inf, up = Inf){
  if(!is.numeric(n) || n < 0)
    stop("invalid value for 'n'")
  l.lo <- length(lo)
  if(!is.numeric(lo) || (l.lo > 1 & l.lo != n))
    stop("'lo' must be a numeric vector of length either 1 or ", n)
  if(l.lo == 1)
    lo <- rep(lo, n)
  l.up <- length(up)
  if(!is.numeric(up) || (l.up > 1 & l.up != n))
    stop("'up' must be a numeric vector of length either 1 or ", n)
  if(l.up == 1)
    up <- rep(up, n)
#problema ocorre na proxima linha para qualquer valor da media < -0.079
#cat("                INVALIDO lo ", lo,fill=T)  #
#cat("  INVALIDO up   ", up, fill=T)              # para monitorar descomentar!

```

```

if(any(lo > up))
  stop("invalid values for lower and upper limits")
if(all(lo == -Inf & up == Inf))
  rnorm(n, mu, sigma)
else
  qnorm(runif(n, min = pnorm(lo, mean = mu, sd = sigma),
    max = pnorm(up, mean = mu, sd = sigma)), mean = mu, sd = sigma) }
rtmvnorm <- function(n, mu, Sigma, lo, up){
  if(!is.numeric(n) || n < 0)
    stop("invalid value for 'n'")
  p <- length(mu)
  if(!all(dim(Sigma) == c(p, p)))
    stop("incompatible arguments")
  l.lo <- length(lo)
  if(!is.numeric(lo) || (l.lo > 1 & l.lo != p))
    stop("'lo' must be a numeric vector of length either 1 or ", p)
  if(l.lo == 1)
    lo <- rep(lo, p)
  l.up <- length(up)
  if(!is.numeric(up) || (l.up > 1 & l.up != p))
    stop("'up' must be a numeric vector of length either 1 or ", p)
  if(l.up == 1)
    up <- rep(up, p)
  if(any(lo > up))
    stop("invalid values for lower and upper limits")
  lo.inf <- lo == -Inf
  up.inf <- up == Inf
  if(!identical(all.equal(Sigma, t(Sigma)), TRUE))
    stop("'Sigma' is not symmetric")
  ed <- eigen(Sigma, symmetric = TRUE)
  ev <- ed$values
  if(!all(ev >= -1e-06 * abs(ev[1])))
    stop("'Sigma' is not positive definite")
  inv.Sigma <- ed$vectors %*% (t(ed$vectors) / ev)
  hi2 <- 1 / diag(inv.Sigma)
  hi <- sqrt(hi2)
  a <- lo - mu
  b <- up - mu

```

```

    index <- row(Sigma) != col(Sigma)
    ci <- -hi2 * inv.Sigma
    diag(ci) <- 0
# cat("  a  ", a,fill=T)
z<-c(0,0) #inicializando z
if((a[1] > 7.5) & (a[2] > 7.5)) {z <- c(a[1],a[2])}
# cat(" z (a) problema ",z,fill=T)
# cat(" a new ",a,fill=T)
}

if((a[1]>7.5) & (a[2] <= 7.5)) {z[1]<- c(a[1])
z[2]<-rtnorm(1,0,1,c(a[2]),b[2])}
if((a[2]>7.5) & (a[1] <= 7.5)) {z[2]<- c(a[2])
z[1] <- rtnorm(p,0,1,a[1],b[1])}
if((a[1] <= 7.5) & (a[2] <= 7.5))
{z <- rtnorm(p, 0,1,a, b)
# cat("z do a",z,fill=T)
}

x <- matrix(0, n + 1, p)
for(i in seq(1, n + 1)){
  cz <- c(ci %*% z)
  a. <- (a - cz) / hi
  a.[lo.inf] <- -Inf
  b. <- (b - cz) / hi
  b.[up.inf] <- Inf
# cat(" a.  ",a.,fill=T)

if((a.[1] > 7.5) & (a.[2] > 7.5)) {z<-cz+hi*c(a.[1],a.[2])}
# cat(" z (a.) problema ",z,fill=T)
# cat (" a. new", a.,fill=T)
}

if((a.[1]> 7.5) & (a.[2]<= 7.5)) {z[1]<- c(a.[1])
z[2]<-rtnorm(1,0,1,c(a.[2]),b.[2])}
if((a.[2]> 7.5) & (a.[1] <= 7.5)) {z[2]<- c(a.[2])
z[1] <- rtnorm(1,0,1,c(a.[1]),b.[1])}

if((a.[1] <= 7.5) & (a.[2] <= 7.5))
{z <- cz + hi * rtnorm(p, 0,1,a. , b.)}
# cat("z do a.",z,fill=T)

```

```

}

      x[i, ] <- mu + z
    }
    x[-1, ] }

drawamu <- function(Bki,STcod,amu0,Y,sigma2p,invSigma0,Sigmastar){
  sumBSTr <- matrix(c(0,0),2)
  for (k in 1:M){
    sumBSTr <- ( Bki[, ,k] %*% STcod %*% Y[, ,k] ) + sumBSTr
    media <- Sigmastar%*%(sumBSTr/sigma2p+invSigma0%*%amu0)
# if (media[1,] < -0.079) {media[1,] <- -0.07}
# if (media[2,] < -0.050) {media[2,] <- -0.05}
# cat("
          MEDIA= ", media,fill=T)
    sky <- (t(rtmvnorm(1, mu = c(media[1,1],media[2,1]),
    Sigma=Sigmastar, lo = c(0,0),up = c(Inf,Inf))))
    cat("sky
          ",sky,fill=T)
    return(matrix(c(sky),2,1))}
drawsigma2p <- function(nuzero,lambdazero,Y,Scod,Ak,Xki,xk){
normSvetor <- 0
sumRiminusSAxi<- matrix(c(seq(0,0,length=2*P*Pcod*M)),P*Pcod,M)
for (i in 1:M){
sumRiminusSAxi[,i] <- Y[, ,i]-Scod%*%Ak%*%Xki[, ,i]
#cat(" sumRiminusSAxi=",sumRiminusSAxi,fill=T)
}
normSvetor <- normSvetor + sum((sumRiminusSAxi)^2)
#cat(" normSvetor=",normSvetor,fill=T)
grausliber <- nuzero+P*M*Pcod
escala<-(nuzero*lambdazero+normSvetor)/(nuzero+P*M*Pcod)
#cat(" scale=",escala,fill=T)
return(rinvchisq(1,df=grausliber,scale=escala))
}

draws1 <- rbind(draws1,c(xk[1,]))
draws2 <- rbind(draws2,c(xk[2,]))
draws3 <- rbind(draws3,c(amu[1,],amu[2,],sigma2p,rhokai1[1]))
##### ITERACOES
#-----+ 1
iteramax <- 100
iteraplota <- 200
Burnin <- 70

```

```

for (iteracao in 1:1){
cat("                iteracao no.:", iteracao,fill=T)
BiRBi <- matrix(c(0,0,0,0),2)
for (i in 1:M){
BiRBi <- Bki[,i]%%R%%Bki[,i]+ BiRBi}
Sigmastar <- solve(invSigma0+BiRBi/sigma2p)
amu <- drawamu(Bki,STcod,amu0,Y,sigma2p,invSigma0,Sigmastar)
Ak      <-      matrix(c(amu[1,],0,0,amu[2,]),2)
sigma2p <-      drawsigma2p(nuzero,lambdazero,Y,Scod,Ak,Xki,xk)
for (m in 1:M){
YM <- Y[,m]
expok1 <- exp(((2*Ak[1,1]/sigma2p)*t(Scod[,1]))%*(YM-Scod%%Ak%%
matrix(c(0,xk[2,m]),2)))
  LLR3[m] <- ((rhokai1[m]/(1-rhokai1[m]))*expok1)
if (LLR3[m] > 1) xk[1,m] <- 1
else xk[1,m] <- -1
expok2 <-exp(((2*Ak[2,2]/sigma2p)*t(Scod[,2]))%*(YM-Scod%%Ak%%
matrix(c(xk[1,m],0),2)))
LLR4[m] <- ((rhokai2[m]/(1-rhokai2[m]))*expok2)
if (LLR4[m] > 1) xk[2,m] <- 1
else xk[2,m] <- -1
}
lambda2p3 <- matrix(c(0:0),M)
lambda2p4 <- matrix(c(0:0),M)
numerador1 <- matrix(c(0:0),M)
denominador1 <- matrix(c(0:0),M)
numerador2 <- matrix(c(0:0),M)
denominador2 <- matrix(c(0:0),M)
## atualizacoes
for (slot in 1:M){
numerador1[slot] <- sum(draws1[,slot][[(draws1[,slot]>0)])/(iteracao+1)
denominador1[slot] <- abs(sum(draws1[,slot][[(draws1[,slot]<0)])) }
for (j in 1:M){
if (numerador1[j] > 0) {
lambda2p3[j] <- numerador1[j]/denominador1[j] # se mudar deixa de convergir!!
rhokai1[j] <- 0.5*(1+tanh(0.5*log(lambda2p3[j]))) }
else { if (numerador1[j] < 1) { rhokai1[j] <- 0.001 }
      else { rhokai1[j] <- 0.999 } } }
}

```



```

for (slot in 1:M){
  numerador2[slot] <- sum(draws2[,slot][draws2[,slot]>0])/(iteracao+1)
  denominador2[slot] <- abs(sum(draws2[,slot][draws2[,slot]<0]))}
for ( j in 1:M) {
  if ( numerador2[j] > 0 ) {
    lambda2p4[j] <- numerador2[j]/denominador2[j]
    rhokai2[j] <- 0.5*(1+tanh(0.5*log(lambda2p4[j])))
  }
  else { if (numerador2[j] < 1) { rhokai2[j] <- 0.001 }
    else { rhokai2[j] <- 0.999 } } }
for ( m in 1:M){
  Xki[,m] <- matrix(c(xk[1,m],xk[2,m]),2)}
for (m in 1:M){
  Bki[,m] <- diag(c(xk[1,m],xk[2,m]))}
draws1 <- rbind(draws1,c(xk[1,]))
draws2 <- rbind(draws2,c(xk[2,]))
draws3 <- rbind(draws3,c(amu[1,],amu[2,],sigma2p,rhokai1[1])) }
#-----++ iteracao 2 ate iteracao 100
for (iteracao in 2:iteramax){
  cat("          iteracao no.:", iteracao,fill=T)
  BiRBi <- matrix(c(0,0,0,0),2)
  for (i in 1:M){
    BiRBi <- Bki[,i]%*%R%*%Bki[,i]+ BiRBi}
  Sigmastar <- solve(invSigma0+BiRBi/sigma2p)
  amu <- drawamu(Bki,STcod,amu0,Y,sigma2p,invSigma0,Sigmastar)
  Ak <- matrix(c(amu[1,],0,0,amu[2,]),2)
  sigma2p <- drawsigma2p(nuzero,lambdazero,Y,Scod,Ak,Xki,xk)
  for (m in 1:M){
    YM <- Y[,m]
    expok1 <- exp(((2*Ak[1,1]/sigma2p)*t(Scod[,1]))%*%(YM-Scod%*%Ak%*%
matrix(c(0,xk[2,m]),2)))
    LLR3[m] <- ((rhokai1[m]/(1-rhokai1[m]))*expok1)
    if (LLR3[m] > 1) xk[1,m] <- 1
    else xk[1,m] <- -1
    expok2 <-exp(((2*Ak[2,2]/sigma2p)*t(Scod[,2]))%*%(YM-Scod%*%Ak%*%
matrix(c(xk[1,m],0),2)))
    if (rhokai2[m]==1) LLR4[m] <- 2
    else LLR4[m] <- ((rhokai2[m]/(1-rhokai2[m]))*expok2)
  }
}

```

```

#          LLR4[m] <- ((rhokai2[m]/(1-rhokai2[m]))*expok2)
if (LLR4[m] > 1) xk[2,m] <- 1
else xk[2,m] <- -1 }
lambda2p3      <- matrix(c(0:0),M)
lambda2p4      <- matrix(c(0:0),M)
numerador1     <- matrix(c(0:0),M)
denominador1   <- matrix(c(0:0),M)
numerador2     <- matrix(c(0:0),M)
denominador2   <- matrix(c(0:0),M)
media1 <- matrix(c(0:0),M)
media2 <- matrix(c(0:0),M)
desviopadrao1 <- matrix(c(0:0),M)
desviopadrao2 <- matrix(c(0:0),M)
alphamaisbeta1 <- matrix(c(0:0),M)
alphamaisbeta2 <- matrix(c(0:0),M)
alpha1 <- matrix(c(0:0),M)
beta1 <- matrix(c(0:0),M)
alfa1 <- matrix(c(0:0),M)
bbeta1 <- matrix(c(0:0),M)
alfa2 <- matrix(c(0:0),M)
bbeta2 <- matrix(c(0:0),M)
alpha2 <- matrix(c(0:0),M)
beta2 <- matrix(c(0:0),M)
Esper1 <- matrix(c(0:0),M)
Esper2 <- matrix(c(0:0),M)
varia1 <- matrix(c(0:0),M)
varia2 <- matrix(c(0:0),M)
## atualizacoes
for (slot in 1:M){
  numerador1[slot] <- sum(draws1[,slot][(draws1[,slot]>0))]/(iteracao+1)
  denominador1[slot] <- abs(sum(draws1[,slot][(draws1[,slot]<0))]) }
for (j in 1:M){
  if (numerador1[j] > 0) {lambda2p3[j] <- numerador1[j]/(1-numerador1[j])
  rhokai1[j] <- 0.5*(1+tanh(0.5*log(lambda2p3[j])))
  }
  else { lambda2p3[j] <- (1-abs(numerador1[j]))/abs(numerador1[j])
  rhokai1[j] <- 0.5*(1+tanh(0.5*log(lambda2p3[j])))
  } }

```

```

for (slot in 1:M){
  numerador2[slot] <- sum(draws2[,slot][(draws2[,slot]>0)]/(iteracao+1)
  denominador2[slot] <- abs(sum(draws2[,slot][(draws2[,slot]<0)]))}
for (j in 1:M){
  if (numerador2[j] > 0) {lambda2p4[j] <- numerador2[j]/(1-numerador2[j])
  rhokai2[j] <- 0.5*(1+tanh(0.5*log(lambda2p4[j])))
  }
  else { lambda2p4[j] <- (1-abs(numerador2[j]))/abs(numerador2[j])
  rhokai2[j] <- 0.5*(1+tanh(0.5*log(lambda2p4[j])))
  } }
for ( m in 1:M){
  Xki[,m] <- matrix(c(xk[1,m],xk[2,m]),2)}
for (m in 1:M){
  Bki[,m] <- diag(c(xk[1,m],xk[2,m]))}
draws1 <- rbind(draws1,c(xk[1,]))
draws2 <- rbind(draws2,c(xk[2,]))
draws3 <- rbind(draws3,c(amu[1,],amu[2,],sigma2p,rhokai1[1])) }
#-----+++
for (iteracao in 101:iteraplota){
  cat("          jota no.: ", jota)
  cat(" iteracao no.:", iteracao,fill=T)
  BiRBi <- matrix(c(0,0,0,0),2)
  for (i in 1:M){
    BiRBi <- Bki[,i]%*%R%*%Bki[,i]+ BiRBi}
  Sigmastar <- solve(invSigma0+BiRBi/sigma2p)
  amu <- drawamu(Bki,STcod,amu0,Y,sigma2p,invSigma0,Sigmastar)
  Ak <- matrix(c(amu[1,],0,0,amu[2,]),2)
  sigma2p <- drawsigma2p(nuzero,lambdazero,Y,Scod,Ak,Xki,xk)
  for (m in 1:M){
    YM <- Y[,m]
    #xk[1,m] <- drawxk1(m,Ak,sigma2p,YM,Scod,xk)
    #xk[2,m] <- drawxk2(m,Ak,sigma2p,YM,Scod,xk) }
    expok1<-exp(((2*Ak[1,1]/sigma2p)*t(Scod[,1]))%*%(YM-Scod%*%Ak%*%
    matrix(c(0,xk[2,m]),2)))
    #if (rhokai1[m]==1) LLR3[m] <- 2
    #else LLR3[m] <- ((rhokai1[m]/(1-rhokai1[m]))*expok1)
    LLR3[m] <- ((rhokai1[m]/(1-rhokai1[m]))*expok1)
    if (LLR3[m] > 1) xk[1,m] <- 1
  }
}

```

```

else xk[1,m] <- -1
expok2 <- exp(((2*Ak[2,2]/sigma2p)*t(Scod[,2]))**%(YM-Scod**Ak**%
matrix(c(xk[1,m],0),2)))
#if (rhokai2[m]==1) LLR4[m] <- 2
#else LLR4[m] <- ((rhokai2[m]/(1-rhokai2[m]))*expok2)
LLR4[m] <- ((rhokai2[m]/(1-rhokai2[m]))*expok2)
if (LLR4[m] > 1) xk[2,m] <- 1
else xk[2,m] <- -1 }
## atualizacoes
medial <- mean( ( draws1[Burnin,] + abs(draws1[Burnin,]) )/2 )
desviopadrao1 <- sd( (draws1[Burnin,]+abs(draws1[Burnin,]))/2 )
Esper1<-medial
varia1<-desviopadrao1*desviopadrao1
alphamaisbeta1 <- Esper1*(1-Esper1)/varia1-1
alpha1 <- alphamaisbeta1*Esper1
beta1 <- alphamaisbeta1*(1-Esper1)
alfa1 <- alpha1 + sum( draws1[iteracao,][[(draws1[iteracao,]>0)] ] )
bbeta1 <- beta1+ M-sum(draws1[iteracao,][[(draws1[iteracao,]>0)]])
for (i in 1:M) rhokai1[j] <- rbeta(1,alfa1,bbeta1)
media2 <- mean( ( draws2[Burnin,] + abs(draws2[Burnin,]) )/2 )
desviopadrao2 <- sd( (draws2[Burnin,]+abs(draws2[Burnin,]))/2 )
Esper2<-media2
varia2<-desviopadrao2*desviopadrao2
alphamaisbeta2 <- Esper2*(1-Esper2)/varia2-1
alpha2 <- alphamaisbeta2*Esper2
beta2 <- alphamaisbeta2*(1-Esper2)
alfa2 <- alpha2+sum(draws2[iteracao,][[(draws2[iteracao,]>0)]])
bbeta2 <- beta2+ M-sum(draws2[iteracao,][[(draws2[iteracao,]>0)]])
for (i in 1:M) rhokai2[j] <- rbeta(1,alfa2,bbeta2)
for ( m in 1:M){
Xki[, ,m] <- matrix(c(xk[1,m],xk[2,m]),2)}
for (m in 1:M){
Bki[, ,m] <- diag(c(xk[1,m],xk[2,m]))}
draws1 <- rbind(draws1,c(xk[1,]))
draws2 <- rbind(draws2,c(xk[2,]))
draws3 <- rbind(draws3,c(amu[1,],amu[2,],sigma2p,rhokai1[1])) }
#-----
##### Posteriores marginais

```

```

rotulo <- c("xk11","xk12","A1=0.63","A2=1.58","sigma2p=0,63","rhokai1[1]")
#X11()
par(mfrow=c(3,2))
# plota xk[1,1] a xk[1,2]
plot.ts(xk[1,])
plot.ts(xk[2,])
# plota A1
#plot(draws3[1:iteraplota,1],ylab=rotulo[3],ylim=c(0,A1+0.4),type="l",col=4)
plot(draws3[1:100,1],ylab=rotulo[3],ylim=c(0,A1+0.4),type="l",col=4)
abline(A1,0,col=2)
title(paste(round(mean(draws3[Burnin:iteramax,1]),2)))
# plota A2
plot(draws3[1:iteraplota,2],ylab=rotulo[4],ylim=c(0,A2+0.4),type="l",col=4)
abline(A2,0,col=2)
title(paste(round(mean(draws3[Burnin:iteramax,2]),2)))
# plota sigma2p
plot(draws3[1:iteraplota,3],ylab=rotulo[5],ylim=c(0,1.3),type="l",col=4)
abline(sigma2,0,col=2)
title(paste(round(mean(draws3[Burnin:iteramax,3]),2)))
# plota rhokai1[1]
plot(draws3[1:iteraplota,4],ylab=rotulo[6],type="l")
title(paste(sign(mean(draws3[Burnin:iteramax,4]))))
contador1<-abs(x[1,]-xk[1,])
contador2<-abs(x[2,]-xk[2,])
contapositivo1[jota,]<-length(contador1[][(contador1[]>0)])
contapositivo2[jota,]<-length(contador2[][(contador2[]>0)])
# para 6dB S/N BER=0.02955729 454/60*256 17/01/2007
} # end do loop: "for (jota in 1:loopvezes){"
cat("BER1=",sum(contapositivo1)/(M*loopvezes),fill=T)
cat("BER2=",sum(contapositivo2)/(M*loopvezes))

```

## 8 Apêndices

Apêndice 1: Simulando distribuições específicas

## 8.1 Simulação de Distribuições Específicas

Quando se necessitar executar simulações estatísticas, deve-se possuir a capacidade de se gerar valores pseudo-aleatórios a partir de diversas distribuições. Nesta seção, são apresentados alguns métodos de simulação de distribuições específicas.

Muitos pacotes de aplicações computacionais (*computer packages*) que possuem ferramentas estatísticas são capazes de gerar números pseudo-aleatórios com distribuição uniforme, no intervalo  $[0, 1]$ . [71]

Assume-se até o final desta seção que se tem disponível uma amostra arbitrariamente grande de variáveis aleatórias, que se acredita serem i.i.d. (números pseudo-aleatórios), com distribuição uniforme no intervalo  $[0, 1]$ . Usualmente, buscam-se variáveis aleatórias com outras distribuições, e o objetivo desta seção é fazer uma revisão de alguns métodos comuns de transformação de variáveis aleatórias uniformes para variáveis aleatórias com outras distribuições.

### 8.1.1 A Transformação Integral de Probabilidade

Supõe-se que uma variável aleatória  $X$  tem uma função de distribuição contínua  $F$ , e seja  $Y = F(X)$ . Esta transformação de  $X$  para  $Y$  é chamada de transformação integral de probabilidade. Mostra-se a seguir que a distribuição de  $Y$  deve ser necessariamente uma distribuição uniforme no intervalo  $[0, 1]$ . Primeiro, uma vez que  $F$  é a função de distribuição de uma variável aleatória, então  $0 \leq F(x) \leq 1$ , para  $-\infty < x < \infty$ . Portanto,  $P(Y < 0) = P(Y > 1) = 0$ . Sendo  $F$  uma função contínua, o conjunto de  $x$  tal que  $F(x) = y$  é um intervalo  $[x_0, x_1]$  limitado, não vazio e fechado, para cada  $y$  no intervalo  $(0, 1)$ . Seja  $F^{-1}(y)$  o ponto extremo  $x_0$  à esquerda deste intervalo, o qual é chamado de  $y$ -quantil de  $F$ . Desta forma  $Y \leq y$  se, e somente se,  $X \leq x_1$ . Seja  $G$  a função de distribuição de  $Y$ . Então

$$G(y) = p(Y \leq y) = p(X \leq x_1) = F(x_1) = y. \quad (1)$$

Portanto,  $G(y) = y$  para  $0 < y < 1$ . Uma vez que esta função é a função de distribuição de uma distribuição uniforme no intervalo  $[0, 1]$ , esta distribuição uniforme é a distribuição de  $Y$ .

Agora supõe-se que  $X$  é uma variável aleatória com uma distribuição contínua  $F$  e que  $G$  é uma distribuição contínua na reta real. Supõe-se ainda que se deseja construir uma variável aleatória  $Z = r(X)$  cuja função de distribuição será a  $G$ .

Para cada valor de  $y$  no intervalo  $0 < y < 1$ , o conjunto de  $z$  tal que  $G(z) = y$  é um intervalo  $[z_0, z_1]$ , limitado e fechado, pois  $G$  é contínua. Seja  $G^{-1}(y)$  o ponto  $z_0$ , limite inferior do intervalo, que é o  $y$ -quantil da distribuição  $G$ . Agora, define-se a variável aleatória  $Z$  como a seguir.

$$Z = G^{-1}[F(X)] \quad (2)$$

Desta forma,  $Z \leq z$  se, e somente se,  $F(X) \leq G(z)$ . Para verificar que a função de distribuição de  $Z$  é realmente  $G$ , nota-se que para todo número  $z$  tal que  $0 < G(z) < 1$ ,

$$P(Z \leq z) = P[F(X) \leq G(z)]. \quad (3)$$

Segue da transformação integral de probabilidade que  $F(X)$  tem uma distribuição uniforme e, portanto, que

$$P[F(X) \leq G(z)] = G(z). \quad (4)$$

Portanto,  $P(Z \leq z) = G(z)$ , que significa que a função de distribuição de  $Z$  é  $G$ , conforme exigido.

## 8.1.2 Simulação

### 8.1.2.1 Números Pseudo-Aleatórios

A maioria dos pacotes computacionais que fazem análises estatísticas também produzem o que se chama de números pseudo-aleatórios. Estes números demonstram ter algumas das propriedades que uma amostra aleatória deve ter, apesar de terem sido gerados por algoritmos determinísticos. O mais fundamental desses programas são os que geram números pseudo-aleatórios que parecem ter uma distribuição uniforme no intervalo  $[0, 1]$ . Serão feitas referências a tais funções como geradoras de números pseudo-aleatórios uniformes. As características importantes que um gerador de número pseudo-aleatório uniforme têm que ter são as seguintes. Os números que ele produz precisam estar espalhados de alguma forma uniformemente sobre o intervalo  $[0, 1]$  e eles precisam parecer que são valores observados de variáveis aleatórias independentes. Esta última característica é muito complicada para colocar em palavras precisamente. Um exemplo de uma sequência que não parece ser observações de variáveis aleatórias independentes



seria uma que fosse perfeitamente e igualmente espaçada. Um outro exemplo seria uma com o seguinte comportamento. Supondo que se olhasse a sequência  $X_1, X_2, \dots$  uma por vez, e em todo o tempo fosse encontrado um  $X_i > 0,5$ , seria anotado o próximo número  $X_{i+1}$ . Se a subsequência de números anotada não está espalhada aproximadamente uniformemente sobre o intervalo  $[0, 1]$ , então a sequência original não se parece com observações de variáveis independentes aleatórias com uma distribuição uniforme no intervalo  $[0, 1]$ . A razão é que a distribuição condicional de  $X_{i+1}$  dada que  $X_i > 0,5$  é supostamente uniforme sobre o intervalo  $[0, 1]$ .

### 8.1.2.2 Geração de Números Pseudo-Aleatórios de uma Distribuição Específica

Um gerador de números pseudo-aleatórios pode ser usado para gerar valores de uma variável aleatória  $Y$  que tem uma função de distribuição contínua específica  $G$ . Se uma variável aleatória  $X$  tem uma distribuição uniforme sobre o intervalo  $[0, 1]$  e se a função de quantil  $G^{-1}$  for definida como anteriormente, então segue da transformação integral de probabilidade que a distribuição da variável aleatória  $Y = G^{-1}(X)$  será  $G$ . Portanto, se um valor de  $X$  é produzido por um gerador de número pseudo-aleatório uniforme, então o valor correspondente de  $Y$  terá a propriedade desejada. Se  $n$  valores independentes  $X_1, \dots, X_n$  são produzidos pelo gerador, então os valores correspondentes  $Y_1, \dots, Y_n$  parecerão ter a forma de uma amostra aleatória de tamanho  $n$  da distribuição com função de distribuição  $G$ .

**Exemplo 8.1** Geração de valores independentes de uma função de distribuição de probabilidade específica.

Supõe-se que um gerador de número pseudo-aleatório uniforme deve ser usado para gerar três valores independentes de uma distribuição para a qual a função de distribuição de probabilidade  $g$  é dada como a seguir.

$$g(y) = \begin{cases} \frac{1}{2}(2 - y) & \text{para } 0 < y < 2 \\ 0 & \text{para outros casos} \end{cases} \quad (5)$$

Para  $0 < y < 2$ , a função de distribuição de  $G$  da distribuição dada é

$$G(y) = y - \frac{y^2}{4} \quad (6)$$

Ainda, para  $0 < x < 1$ , a função inversa  $y = G^{-1}(x)$  pode ser encontrada

resolvendo-se a equação  $x = G(y)$  para  $y$ . O resultado é

$$y = G^{-1}(x) = 2[1 - (1 - x)^{1/2}]. \quad (7)$$

O próximo passo é gerar três números pseudo-aleatórios uniformemente distribuídos  $x_1$ ,  $x_2$  e  $x_3$  utilizando o gerador. Supõe-se que os três valores gerados são  $x_1 = 0,4125$ ,  $x_2 = 0,0894$  e  $x_3 = 0,8302$ .

Quando estes valores de  $x_1$ ,  $x_2$  e  $x_3$  são substituídos sucessivamente na Equação (7), os valores de  $y$  que são obtidos são  $y_1 = 0,47$ ,  $y_2 = 0,09$  e  $y_3 = 1,18$ . Estes valores de  $y$  são então tratados como os valores observados de três variáveis aleatórias independentes com a distribuição cuja função de distribuição de probabilidade é  $g$ .

### 8.1.3 Algoritmos com Propósitos Especiais

Existem casos em que a função de distribuição  $G$  não é fácil de se inverter. Por exemplo, se  $G$  é a Normal padrão, então  $G^{-1}$  deve ser obtida por aproximação numérica. No entanto, existe um método mais elaborado para se transformar duas variáveis aleatórias uniformes e independentes sobre um intervalo  $[0, 1]$  em duas variáveis Normais padronizadas. Este método foi descrito por Box e Muller em 1958 [79].

#### **Exemplo: Gerando duas variáveis independentes da Normal padrão**

Sejam  $X_1$ ,  $X_2$  independentes com distribuição Uniforme no intervalo  $[0, 1]$ . A função densidade de probabilidade conjunta de  $(X_1, X_2)$  é

$$f(x_1, x_2) = 1, \text{ para } 0 < x_1, x_2 < 1. \quad (8)$$

Definindo-se

$$Y_1 = [-2 \log(X_1)]^{1/2} \sin(2\pi X_2), \quad (9a)$$

$$Y_2 = [-2 \log(X_1)]^{1/2} \cos(2\pi X_2). \quad (9b)$$

A inversa dessa transformação é

$$X_1 = \exp[-(Y_1^2 + Y_2^2)/2], \quad (10a)$$

$$X_2 = \frac{1}{2\pi} \arctan(Y_1/Y_2). \quad (10b)$$

Computando-se o Jacobiano, que é o determinante da matriz de derivadas

parciais da função inversa

$$\begin{bmatrix} -y_1 \exp[-(y_1^2 + y_2^2)/2] & -y_2 \exp[-(y_1^2 + y_2^2)/2] \\ \frac{1}{2\pi y_2} \frac{1}{1+(y_1/y_2)^2} & -\frac{y_1}{2\pi y_2^2} \frac{1}{1+(y_1/y_2)^2} \end{bmatrix} \quad (11)$$

O determinante desta matriz é  $J = \exp[-(y_1^2 + y_2^2)/2]/(2\pi)$ . A função densidade de probabilidade conjunta de  $(Y_1, Y_2)$  é então

$$\begin{aligned} g(y_1, y_2) &= f\left(\exp[(y_1^2 + y_2^2)/2], \arctan(y_1/y_2)/(2\pi)\right) |J| \\ &= \exp[-(y_1^2 + y_2^2)/2]/(2\pi). \end{aligned} \quad (12)$$

Esta é a função densidade de probabilidade de duas variáveis da Normal padrão.

Para mais métodos de simulação deve-se consultar as referências [71, 80].