

ALGORITMOS BASEADOS EM MODELOS BAYESIANOS PARA  
RESTAURAÇÃO DE SINAIS DE ÁUDIO

Flávio Rainho Ávila

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

Aprovada por:

---

Prof. Luiz Wagner Pereira Biscainho, D.Sc.

---

Dr. Paulo Antonio Andrade Esquef, D.Sc.

---

Prof. Eduardo Antonio Barros da Silva, Ph.D.

---

Prof. Dani Gamerman, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2008

ÁVILA, FLÁVIO RAINHO

Algoritmos baseados em modelos bayesianos para restauração de sinais de áudio [Rio de Janeiro] 2008

XII, 99 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Elétrica, 2008)

Dissertação - Universidade Federal do Rio de Janeiro, COPPE

1.Processamento de Sinais

2.Processamento de Áudio

3.Inferência Estatística

I.COPPE/UFRJ    II.Título (série)

## Agradecimentos

É um prazer agradecer às pessoas que contribuíram para a realização deste trabalho.

Em primeiro lugar, agradeço a meu orientador e amigo, Luiz Wagner, cuja excelência técnica só é comparável a sua bondade, honestidade e senso de humor. Este trabalho não seria possível sem seu apoio. Obrigado, Luiz.

Agradeço ao professor Diniz, que me ofereceu trabalhos que possibilitaram meu sustento durante boa parte do mestrado. Essa ajuda foi fundamental para que pudesse concluí-lo.

Meu colega Rafael de Jesus, fazendo jus ao nome, demonstrou imensa bondade ao me emprestar seu computador para que pudesse terminar de escrever a dissertação. Não tenho palavras para lhe agradecer.

Agradeço ao colega de trabalho Paulo Esquef, que me emprestou algumas rotinas que foram úteis neste trabalho, e contribuiu com sugestões durante sua execução.

Minha psicóloga Geny Nunes me ajudou a superar grandes dificuldades, muitas vezes criadas por mim mesmo. Agradeço imensamente seu apoio.

Minha família, embora distante, esteve sempre presente quando precisei. Meu pai, meus irmãos, minha tia Marta e a memória de minha mãe são fontes constantes de inspiração e estímulos. A todos expresso minha gratidão.

Por fim, agradeço aos colegas do Laboratório de Processamento de Sinais por ajudarem a criar um ambiente de trabalho tranquilo, amigável e divertido.

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

## ALGORITMOS BASEADOS EM MODELOS BAYESIANOS PARA RESTAURAÇÃO DE SINAIS DE ÁUDIO

Flávio Rainho Ávila

Março/2008

Orientador: Luiz Wagner Pereira Biscainho

Programa: Engenharia Elétrica

Uma aplicação freqüente de técnicas de restauração de áudio é na remoção de defeitos audíveis em gravações de interesse artístico e histórico, tipicamente de mídias analógicas. As técnicas de restauração digital variam de métodos heurísticos computacionalmente eficientes a métodos estatísticos sofisticados. Estes últimos permitem modelar o sinal de áudio e distúrbios de forma bastante realista, ao custo de alta complexidade computacional.

Esta dissertação lida com o problema de remoção de dois tipos de ruído localizado: *clicks* e pulsos longos. Foram empregados métodos bayesianos para a modelagem estatística dos distúrbios, e foram usadas técnicas baseadas em MCMC para a estimação numérica das variáveis envolvidas. No caso dos *clicks*, propõe-se um modelo que os descreve individualmente, com o objetivo de reduzir a complexidade computacional em relação aos métodos bayesianos encontrados na literatura. Para os pulsos longos, é proposta uma modificação no modelo anterior dos *clicks*, permitindo a detecção e remoção conjuntas desse tipo de defeito. Os dois algoritmos têm estrutura similar e se baseiam no algoritmo de Metropolis-Hastings com saltos reversíveis associado ao amostrador de Gibbs. A qualidade da restauração obtida foi aferida através de simulações com sinais sintéticos e reais.

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

BAYESIAN MODEL-BASED ALGORITHMS FOR DIGITAL AUDIO  
RESTORATION

Flávio Rainho Ávila

March/2008

Advisor: Luiz Wagner Pereira Biscainho

Department: Electrical Engineering

A common application of audio restoration techniques is the removal of audible defects from sound recordings of artistic and historical interest, typically found on analog media. Digital restoration techniques span from low-complexity heuristic methods to sophisticated statistical ones. The latter allows for modelling of audio signals and defects in a realistic fashion, at the cost of higher computational burden.

This dissertation deals with the removal of two kinds of localized disturbances: clicks and long pulses. Bayesian methods have been employed for statistical modeling of those defects, and MCMC-based techniques have been used for numerical estimation of the involved variables. In the case of clicks, a new model which describes them individually yields reduced computational complexity compared to Bayesian methods found in literature. For long pulses, the former model is modified to allow the joint detection and removal of this kind of defect. Both algorithms exhibit a similar structure, mixing the Reversible-Jump Metropolis-Hastings algorithm and the Gibbs Sampler. The attained performance has been assessed through simulations on synthetic as well as real signals.

# Sumário

Glossário	xii
<b>1 Introdução</b>	<b>1</b>
<b>2 Métodos Bayesianos de Inferência Estatística</b>	<b>4</b>
2.1 Algumas interpretações de probabilidade . . . . .	4
2.2 Problema geral de inferência e algumas soluções . . . . .	6
2.3 Inferência bayesiana . . . . .	8
2.3.1 Teorema de Bayes . . . . .	8
2.4 Modelo bayesiano hierárquico . . . . .	10
2.5 Distribuição <i>a priori</i> . . . . .	11
2.5.1 Prioris conjugadas . . . . .	11
2.5.2 Priori não-informativa . . . . .	12
2.6 Eliminação de parâmetros . . . . .	13
2.7 Teoria da decisão bayesiana . . . . .	13
2.8 Comentários conclusivos . . . . .	14
<b>3 Técnicas Numéricas de Estimação</b>	<b>15</b>
3.1 Técnicas de Monte Carlo . . . . .	16
3.2 Método da rejeição . . . . .	16
3.3 Cadeias de Markov . . . . .	17
3.3.1 Conceituação . . . . .	17
3.3.2 Distribuição invariante . . . . .	19
3.3.3 Ergodicidade . . . . .	19
3.3.4 Cadeias de Markov para espaço de estados contínuo . . . . .	20

3.4	Amostrador de Gibbs . . . . .	21
3.5	Algoritmo de Metropolis-Hastings . . . . .	22
3.6	MCMC com saltos reversíveis . . . . .	23
3.7	Análise de convergência . . . . .	25
3.8	Conclusões . . . . .	26
<b>4</b>	<b>Alguns Métodos de Correção de <i>Clicks</i></b>	<b>28</b>
4.1	Modelagem do sinal corrompido . . . . .	29
4.2	Modelagem do sinal de áudio . . . . .	30
4.3	Remoção de <i>clicks</i> por filtragem inversa . . . . .	32
4.4	Correção de <i>clicks</i> pelo método bayesiano . . . . .	34
4.4.1	Estimação em blocos . . . . .	34
4.4.2	Estimação seqüencial . . . . .	36
4.4.3	Estimação iterativa através do algoritmo EM e do amostrador de Gibbs . . . . .	36
4.5	Conclusões . . . . .	38
<b>5</b>	<b>Métodos alternativos para a Correção de <i>clicks</i></b>	<b>39</b>
5.1	Modelo proposto . . . . .	40
5.2	Restauração usando modelo multiplicativo para o ruído . . . . .	42
5.2.1	Distribuição <i>a priori</i> . . . . .	43
5.2.1.1	Localização ( $n_0$ ) e duração ( $M$ ) . . . . .	43
5.2.1.2	Amplitude ( $V$ ) . . . . .	44
5.2.1.3	Taxa de decaimento ( $\lambda$ ) . . . . .	44
5.2.1.4	Sinal original ( $\mathbf{x}$ ) . . . . .	44
5.2.2	Obtenção da distribuição <i>a posteriori</i> . . . . .	45
5.3	Restauração pelo método de Fibonacci . . . . .	46
5.3.1	Estimação de $n_0$ . . . . .	48
5.3.2	Estimação de $V$ . . . . .	49
5.3.3	Estimação de $\lambda$ . . . . .	50
5.3.4	Resultados . . . . .	50
5.3.4.1	Situação A: <i>click</i> artificial . . . . .	51

5.3.4.2	Situação B: <i>click</i> real . . . . .	51
5.4	Restauração através de algoritmo MH e amostrador de Gibbs . . . . .	52
5.4.1	Inicialização . . . . .	54
5.4.2	Distribuições condicionais e escolha das propostas . . . . .	55
5.4.2.1	Amostragem do sinal original $\mathbf{x}$ . . . . .	55
5.4.2.2	Amostragem da localização $\mathbf{n}_0$ e da duração $\mathbf{M}$ . . . . .	56
5.4.2.3	Amostragem da amplitude $\mathbf{V}$ . . . . .	57
5.4.2.4	Amostragem da taxa de decaimento $\lambda$ . . . . .	58
5.4.3	Resultados . . . . .	59
5.5	Restauração através de algoritmo de Metropolis-Hastings com saltos reversíveis . . . . .	61
5.5.1	Resultados . . . . .	69
5.5.2	Análise da complexidade computacional . . . . .	70
5.6	Conclusão . . . . .	78
<b>6</b>	<b>Correção de Pulsos Longos</b>	<b>80</b>
6.1	Técnicas existentes . . . . .	80
6.2	Método proposto . . . . .	82
6.3	Cálculo das distribuições condicionais e escolha das propostas . . . . .	83
6.4	Resultados . . . . .	87
6.5	Considerações sobre a complexidade computacional . . . . .	90
6.6	Conclusões . . . . .	90
<b>7</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>94</b>
	<b>Referências Bibliográficas</b>	<b>96</b>

# Lista de Figuras

2.1	Exemplo de modelo bayesiano hierárquico . . . . .	10
3.1	Evolução de um parâmetro de uma cadeia de Markov a partir de diferentes valores iniciais. . . . .	26
5.1	Sinal contaminado por <i>clicks</i> . . . . .	40
5.2	<i>Clicks</i> retirados do sinal contaminado. . . . .	41
5.3	Detalhe de um distúrbio impulsivo. . . . .	41
5.4	Distribuição <i>a posteriori</i> de $n_0$ com $V$ e $\lambda$ fixos. . . . .	48
5.5	Distribuição <i>a posteriori</i> de $V$ com $n_0$ e $\lambda$ fixos. . . . .	49
5.6	Distribuição <i>a posteriori</i> de $\lambda$ com $V$ e $n_0$ fixos. . . . .	50
5.7	Sinal corrompido com <i>click</i> artificial. . . . .	52
5.8	Sinal original (linha contínua), sinal estimado pelo método seqüencial (linha sólido-pontilhada) e sinal estimado pelo método de Fibonacci (linha tracejada). . . . .	52
5.9	Sinal corrompido com <i>click</i> real. . . . .	53
5.10	Sinal original (linha contínua), sinal estimado pelo método seqüencial (linha sólido-pontilhada) e sinal estimado pelo método de Fibonacci (linha tracejada). . . . .	53
5.11	Sinal com <i>clicks</i> presentes a partir das amostras 100 e 300. . . . .	60
5.12	Distribuição conjunta de $V$ e $\lambda$ . . . . .	61
5.13	Histograma de $V$ e $\lambda$ sendo as demais variáveis fixas. . . . .	61
5.14	Distribuição e histograma de $V$ . . . . .	62
5.15	Distribuição e histograma de $\lambda$ . . . . .	62
5.16	Distribuição e histograma de $n_0$ . . . . .	63

5.17	Distribuição e histograma de $M$ . . . . .	63
5.18	Amostras da condicional total de $V$ . . . . .	64
5.19	Amostras da condicional total de $\lambda$ . . . . .	64
5.20	Amostras da condicional total de $M$ . . . . .	65
5.21	Amostras da condicional total de $n_0$ . . . . .	65
5.22	Sinais original (linha contínua) e restaurado pelo método baseado no algoritmo de Metropolis-Hastings (linha sólido-pontilhada). . . . .	66
5.23	Sinal contaminado por <i>clicks</i> superpostos. . . . .	66
5.24	Localização dos <i>clicks</i> superpostos. . . . .	67
5.25	Gráficos relativos à situação A. (a) Sinal gerado artificialmente usando modelo AR de ordem 30 com 3 <i>clicks</i> artificiais inseridos. (b) Detalhe do sinal corrompido (linha contínua) e sinal restaurado (linha sólido-pontilhada). (c) Diferença entre sinais original e restaurado. . . . .	71
5.26	Gráficos relativos à situação A. (a) Evolução de $\mathbf{n}_0$ . (b) Evolução de $\mathbf{M}$ . (c) Evolução de $\mathbf{V}$ . (d) Evolução de $\boldsymbol{\lambda}$ . . . . .	72
5.27	Gráficos relativos à situação B. (a) Sinal gerado artificialmente usando modelo AR de ordem 30 com 3 <i>clicks</i> reais inseridos. (b) Sinal corrompido (linha contínua) e sinal restaurado (linha sólido-pontilhada). (c) Diferença entre sinais original e restaurado. . . . .	73
5.28	Gráficos relativos à situação B. (a) Evolução de $\mathbf{n}_0$ . (b) Evolução de $\mathbf{M}$ . (c) Evolução de $\mathbf{V}$ . (d) Evolução de $\boldsymbol{\lambda}$ . . . . .	74
5.29	Gráficos relativos à situação C. (a) Sinal real contaminado por <i>click</i> . (b) Detalhe do sinal corrompido (linha contínua) e sinal restaurado (linha sólido-pontilhada). . . . .	75
5.30	Gráficos relativos à situação C. (a) Evolução de $\mathbf{n}_0$ . (b) Evolução de $\mathbf{M}$ . (c) Evolução de $\mathbf{V}$ . (d) Evolução de $\boldsymbol{\lambda}$ . . . . .	76
6.1	Trecho de sinal contaminado por pulso longo. . . . .	81
6.2	Amostras da variável $V_c$ . . . . .	88

6.3	Gráficos relativos à situação A. (a) Sinal gerado artificialmente usando modelo AR de ordem 40. (b) Pulso longo artificial. (c) Sinal artificial corrompido. (d) Sinal restaurado. (e) Pulso longo inserido (linha tracejada) e pulso longo extraído (linha contínua). (f) Espectro dos sinais original (linha contínua), corrompido (linha tracejada) e restaurado (linha sólido-pontilhada). . . . .	89
6.4	Gráficos relativos à situação B. (a) Sinal real sem defeitos. (b) Pulso longo real. (c) Sinal artificial corrompido por pulso longo real. (d) Sinal restaurado. (e) Pulso longo inserido (linha tracejada) e pulso longo extraído (linha contínua). (f) Espectro dos sinais original (linha contínua), corrompido (linha tracejada) e restaurado (linha sólido-pontilhada). . . . .	91
6.5	Gráficos relativos à situação C. (a) Sinal real corrompido por pulso longo. (b) Sinal restaurado. (c) Sinal corrompido e pulso extraído (mais suave). (d) Espectro dos sinais corrompido (linha tracejada) e restaurado (linha contínua). . . . .	92

# Glossário

$\mathbf{x}$  – dados observados

$\boldsymbol{\theta}$  – parâmetros

MAP – *Maximum a posteriori*

MH – algoritmo de Metropolis-Hastings

RJ-MH – algoritmo de Metropolis-Hastings com saltos reversíveis

MCMC – *Markov Chain Monte Carlo*

$x(n)$  – sinal de áudio não corrompido

$y(n)$  – sinal de áudio corrompido

$i(n)$  – sequência indicativa de amostras corrompidas

AR – autorregressivo

$\mathbf{a}$  – vetor de parâmetros do modelo AR

$\mathbf{A}$  – matriz associada ao modelo AR

$\sigma_e^2$  – variância de excitação do modelo AR

$n_0$  – amostra inicial de um defeito (*click* ou pulso longo)

$M$  – duração de um *click* ou duração da descontinuidade inicial de um pulso longo

$\lambda$  – taxa de decaimento de um *click*

$V$  – amplitude de um *click*

IG – distribuição gama inversa

$\sigma_{V_a}$  – variância da descontinuidade inicial de um pulso longo

$V_c$  – fator de amplitude da cauda de um pulso longo

$\tau_e$  – constante de tempo associada ao decaimento da envoltória de um pulso longo

$\tau_f$  – constante de tempo associada ao decaimento da frequência de um pulso longo

$\phi$  – fase inicial do pulso

$f_{min}/f_{max}$  – frequência de oscilação mínima/máxima da cauda de um pulso longo

# Capítulo 1

## Introdução

Sistemas de restauração de áudio têm como objetivo reduzir ou eliminar defeitos audíveis presentes em sinais de áudio, introduzidos pelos mecanismos de gravação e reprodução, ou resultantes de deterioração ou desgaste dos meios de gravação [1].

Os defeitos mais comuns presentes em sinais de áudio costumam ser designados por onomatopéias. *Click, crackle, hiss, wow* são alguns exemplos. De forma simplificada, as degradações são classificadas em dois grupos: localizados (como *click* ou *crackle*) e globais (como *hiss* ou *wow*). Degradações do primeiro tipo contaminam apenas algumas amostras do sinal e tendem a ocorrer em rajadas, enquanto que defeitos do segundo tipo afetam todas as amostras do sinal.

As primeiras técnicas nessa área consistiam de processamentos analógicos simples, que variavam desde filtragem passa-baixas, para remoção de ruído de fundo, até retirada manual das áreas defeituosas do sinal. O procedimento em geral demandava muito tempo e gerava resultados insatisfatórios [1].

Com o aumento do poder computacional dos processadores digitais nos anos 1980, técnicas mais sofisticadas de processamento de sinais puderam ser implementadas para restauração de áudio, tornando o procedimento mais rápido (em alguns casos em tempo real) e garantindo melhor resultado. No caso de defeitos localizados, a abordagem em geral era realizada no domínio do tempo e consistia em localizar a região defeituosa e substituí-la por uma estimativa do sinal original. Os defeitos globais eram usualmente tratados por filtragem no domínio da frequência. Recentemente têm sido propostas técnicas de restauração por ressíntese [1].

Essas técnicas heurísticas computacionalmente eficientes formaram a primeira fase de trabalhos na área de restauração digital de áudio. Em particular, a técnica baseada em filtragem inversa para localização e remoção de *clicks* [2] e o método de subtração espectral para redução de ruído de fundo (*hiss*) [3] tornaram-se populares devido a sua simplicidade de implementação e por produzir bons resultados perceptivos.

Usando técnicas sofisticadas de inferência estatística, Godsill [4, 5] deu início a uma segunda fase na área. Métodos bayesianos foram utilizados para os problemas de correção da variação de *pitch* (*wow*) e remoção de *clicks*. Para essa última tarefa, a aplicação dos algoritmos EM (*Expectation-Maximization*) e de técnicas numéricas baseadas em MCMC (*Markov-Chain Monte Carlo*), em particular o amostrador de Gibbs, permitiram o uso de modelos hierárquicos bastantes sofisticados, garantindo resultados de alta qualidade. O grande inconveniente dessas técnicas é a alta complexidade computacional exigida.

Esta dissertação lida principalmente com o problema de restauração de *clicks*. O objetivo é propor novos métodos de estimação que tornem o procedimento de restauração computacionalmente mais eficiente, mantendo a sofisticação permitida pelas técnicas baseadas em MCMC. Será tratado ainda o problema de remoção de pulsos longos, usando uma abordagem similar.

Para isso, serão propostos novos modelos com características que explorem as propriedades dos *clicks* e permitem reduzir a complexidade computacional. Para seleção de modelos e estimação do sinal de áudio original, serão adotadas técnicas de Monte Carlo via Cadeias de Markov (MCMC), em particular os algoritmos de Metropolis-Hastings (MH) e de Metropolis-Hastings com saltos reversíveis (RJ-MH).

Este texto está organizado da seguinte forma: Além desta introdução, apresentamos no Capítulo 2 os princípios básicos de inferência bayesiana, que incluem os conceitos de distribuição *a priori* e distribuição *a posteriori*, priori não-informativa e modelo hierárquico. No Capítulo 3, apresentamos alguns métodos numéricos que permitem lidar com problemas de estimação complicados, que surgem naturalmente da aplicação do método bayesiano; será feita uma introdução a cadeias de Markov e alguns algoritmos baseados em MCMC serão descritos com algum detalhe. O texto

prossegue com a descrição do problema de restauração no Capítulo 4; serão apresentados alguns modelos possíveis para a descrição de *clicks* e alguns métodos de restauração existentes na literatura que serão usados como referência para comparação. No Capítulo 5, é apresentado um novo método de restauração de *clicks* que procura resolver alguns inconvenientes dos métodos expostos no Capítulo 4. Nesse capítulo são reportados os resultados obtidos com o método proposto e a comparação com os métodos existentes. O Capítulo 6 propõe um algoritmo para remoção de pulsos longos usando uma abordagem similar ao desenvolvido para remoção *clicks*. As conclusões finais estão no Capítulo 7.

# Capítulo 2

## Métodos Bayesianos de Inferência Estatística

Este capítulo se destina à apresentação de conceitos básicos de inferência estatística, com enfoque nos conceitos e métodos que serão diretamente usados nos algoritmos desenvolvidos nesta tese. Para uma apresentação mais completa sobre o tema, recomendamos os livros de Bernardo e Smith [6], W. Bolstad [7] e Gamerman [8].

O capítulo começa com uma discussão sobre duas importantes interpretações para o conceito de probabilidade: clássica, freqüentista e bayesiana (ou subjetiva). Em seguida, apresenta-se o problema geral de otimização com alguns possíveis métodos freqüentistas de solução. O paradigma bayesiano de inferência estatística é apresentado na Seção 2.3 e as diferenças em relação à abordagem freqüentista são salientadas. Apresentam-se ainda o teorema de Bayes, base para inferência bayesiana, e os conceitos de probabilidade *a priori*, priori não-informativa e prioris conjugadas. Para finalizar o capítulo, descreve-se sucintamente a teoria de decisão bayesiana, um método útil para localização de amostras corrompidas num sinal de áudio.

### 2.1 Algumas interpretações de probabilidade

O conceito de probabilidade tem provocado inúmeras controvérsias históricas. Nesta seção, vamos apresentar três interpretações desse conceito: clássica, freqüen-

tista e bayesiana (ou subjetiva). Para uma exposição mais completa, veja [9].

Pela definição *clássica* de probabilidade, é preciso estabelecer um conjunto bem definido de eventos elementares, todos supostos igualmente prováveis. A probabilidade de um evento qualquer (formado por uma união de eventos elementares) é então calculada como a razão entre o número de eventos elementares favoráveis àquele evento e o número total de eventos possíveis. Essa definição levanta algumas questões, por exemplo: Como estabelecer *a priori* que certos eventos são igualmente prováveis?

Outro conceito de probabilidade bastante popular é o *freqüentista*, cujo nome se deve à noção de freqüência relativa na qual ele se baseia. Em vez de estabelecer *a priori* um conjunto de eventos possíveis, como na definição clássica, pela visão freqüentista deve-se levar em conta o número de eventos efetivamente obtidos em experimentos, supostamente em condições similares (idealmente idênticas). Se o número de experimentos é suficientemente grande, a probabilidade é a razão entre o número de resultados favoráveis e o número total de experimentos. Pela lei dos grandes números, essa razão deve se aproximar cada vez mais de um certo valor quanto maior for o número de experimentos.

Como exemplo, seja o experimento de lançamento de um dado. Existem 6 possibilidades de resultado para cada lançamento. Pela simetria da situação, um adepto da visão clássica diria que todos os resultados são igualmente prováveis. A probabilidade de uma dada face ser obtida é, portanto, de  $1/6$ . A probabilidade de o resultado ser, digamos, um número par é igual à razão entre a quantidade de números pares entre 1 e 6 e a quantidade total resultados possíveis. Para um freqüentista, é impossível determinar *a priori* as probabilidades envolvidas; a probabilidade só pode ser obtida depois de um grande número de experimentos. A probabilidade de sair um número par é obtida contando-se o número de vezes em que este resultado realmente foi obtido e dividindo-se pelo número total de experimentos. Se o dado não for viciado, espera-se que essa razão seja próxima a  $1/2$ , o mesmo valor obtido pela definição clássica. No entanto, só se pode afirmar que o dado não é viciado após os experimentos serem realizados.

É comum, entretanto, que se fale de probabilidades associadas a eventos que

não podem ser repetidos e para os quais nem a noção clássica nem a freqüentista fazem sentido. Por exemplo, “a probabilidade de o Flamengo ser rebaixado este ano é de 90%” ou “a probabilidade de a rede do LPS não funcionar hoje é de 99,9%”. Para permitir o tratamento matemático de sentenças como essas, algumas escolas estatísticas aceitam a noção de probabilidade vista como “grau de confiança” (subjéitiva) sobre a veracidade de uma afirmação. Essa visão ficou conhecida como bayesiana ou subjéitiva.

Essas duas principais interpretações de probabilidade são o ponto de partida para as duas escolas de estatísticas antagônicas, a freqüentista e a bayesiana. Embora ambas sejam bem sucedidas no tratamento de diversos problemas, é comum que os estatísticos adotem uma visão e rejeitem a outra, reforçando a longa controvérsia entre essas duas escolas [10].

## 2.2 Problema geral de inferência e algumas soluções

Inferência estatística é empregada quando se deseja obter informações a partir de dados que podem ser descritos por modelos probabilísticos. Por exemplo, o receptor de um sistema de comunicação precisa decidir com base nos dados recebidos quais foram os prováveis dados transmitidos. Na aplicação desta tese, desejamos estimar o sinal de áudio original com base num conjunto de amostras corrompidas.

Para formalizar o conceito, seja  $\mathbf{x}$  uma variável aleatória (possivelmente multidimensional) descrita pela distribuição<sup>1</sup>  $p(\mathbf{x}; \boldsymbol{\theta})$ , em que  $\boldsymbol{\theta}$  é um vetor contendo parâmetros (desconhecidos) que definem as características da distribuição (tais como média ou variância, por exemplo). Dispondo de um conjunto de amostras de  $\mathbf{x}$ , deseja-se estimar o vetor de parâmetros  $\boldsymbol{\theta}$ .

---

<sup>1</sup>O nome “distribuição” de uma variável aleatória  $X$  se referirá tanto à densidade de probabilidade  $p(x)$  (no caso de V.A. contínua) quanto à probabilidade  $P(x)$  (no caso de V.A. discreta); quando for importante será feita a distinção clara disto. Mas haverá alguma liberdade de terminologia, por exemplo, ao dizer que a maximização da densidade de probabilidade maximiza a probabilidade.

De acordo com a abordagem freqüentista,  $\boldsymbol{\theta}$  é um valor fixo porém desconhecido. Partindo-se dessa premissa, diversos critérios de otimização podem ser usados para estimar o valor dos parâmetros. Os mais usuais são a máxima verossimilhança (*Maximum Likelihood*, ML), o erro médio quadrático (*Mean Square Error*, MSE) e a mínima variância (*Minimum Variance*, MV) [11,12]. Devido a sua importância para inferência bayesiana, o critério de máxima verossimilhança será brevemente descrito aqui.

A função de verossimilhança,  $l(\mathbf{x}; \boldsymbol{\theta})$ , é definida como:

$$l(\mathbf{x}; \boldsymbol{\theta}) = p(\mathbf{x}|\boldsymbol{\theta}), \quad (2.1)$$

e é interpretada como uma quantificação da probabilidade de os dados  $\mathbf{x}$  serem observados quando o vetor de parâmetros é  $\boldsymbol{\theta}$ . O critério ML estabelece que o valor estimado para  $\boldsymbol{\theta}$  é aquele que maximiza a função de verossimilhança, ou seja, é o valor de  $\boldsymbol{\theta}$  que faz os dados observados serem os mais prováveis. Matematicamente, pelo critério ML:

$$\boldsymbol{\theta}^{\text{ML}} = \operatorname{argmax}_{\boldsymbol{\theta} \in \Theta} l(\mathbf{x}; \boldsymbol{\theta}), \quad (2.2)$$

em que  $\Theta$  é o conjunto de valores possíveis para  $\boldsymbol{\theta}$ .

Como exemplo de aplicação do critério ML, vamos considerar o modelo linear, que aparece freqüentemente em diversos problemas de processamento de sinais. Nesse modelo, o vetor de dados observados  $\mathbf{x}$  é obtido pela soma de um produto matricial e um ruído  $\mathbf{v}$ :

$$\mathbf{x} = \mathbf{G}\boldsymbol{\theta} + \mathbf{v}. \quad (2.3)$$

Esse modelo pode descrever, por exemplo, um sinal formado por uma mistura de senóides com amplitudes desconhecidas imersas em ruído.

Freqüentemente  $\mathbf{v}$  é modelado como ruído branco gaussiano de média zero e variância  $\sigma_v^2$ , caso em que a verossimilhança é dada por [4]:

$$p(\mathbf{x}|\boldsymbol{\theta}) = p_v(\mathbf{x} - \mathbf{G}\boldsymbol{\theta}) = \frac{1}{(2\pi\sigma_v^2)^{N/2}} e^{-\frac{1}{2\sigma_v^2}(\mathbf{x} - \mathbf{G}\boldsymbol{\theta})^T(\mathbf{x} - \mathbf{G}\boldsymbol{\theta})}, \quad (2.4)$$

e a estimação pelo critério ML fornece [4]:

$$\boldsymbol{\theta}^{\text{ML}} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{x}. \quad (2.5)$$

Esse resultado coincide com o conhecido método de mínimos quadrados [13], cujo critério de otimização não é estocástico. Veremos na próxima seção como esse mesmo problema é encarado pela abordagem bayesiana.

## 2.3 Inferência bayesiana

Inferência bayesiana é uma técnica que parte da noção subjetiva de probabilidade, vista como “grau de confiança”, e trata os parâmetros desconhecidos como variáveis aleatórias com uma densidade de probabilidade associada. A distribuição associada aos parâmetros antes de os dados estarem disponíveis recebe o nome de priori ou distribuição *a priori*; essa função deve refletir o “grau de confiança” do projetista sobre os possíveis valores dos parâmetros. Após os dados serem levados em consideração, a nova distribuição dos parâmetros é chamada de posteriori ou distribuição *a posteriori*. Na Seção 2.5 o conceito de probabilidade *a priori* será melhor analisado.

Para este trabalho, algumas características do paradigma bayesiano se mostraram importantes. A principal delas é a possibilidade de se incorporar qualquer conhecimento do usuário, muitas vezes subjetivo, ao modelo probabilístico, algo proibido pela ótica freqüentista. Num sistema de restauração, o conhecimento do projetista sobre o comportamento típico de um defeito (por exemplo, a taxa de amostras corrompidas) é de grande valia para o desempenho do sistema. Além disso, a abordagem bayesiana permite que parâmetros indesejados possam ser marginalizados (integrados) e retirados da análise, tornando desnecessário um pré-processamento para estimação desses parâmetros (ver Seção 2.6).

### 2.3.1 Teorema de Bayes

Através do teorema (ou regra) de Bayes, podemos utilizar os dados disponíveis para atualizar o conhecimento sobre os prováveis valores dos parâmetros. Seja  $p(\theta|H)$  a distribuição *a priori*, em que  $H$  (de história) representa todas as informações à disposição do usuário sobre os parâmetros de interesse *antes* de os dados estarem disponíveis. Pelo teorema de Bayes, podemos obter a distribuição *a poste-*

riori

$$p(\boldsymbol{\theta}|\mathbf{x}, H) = \frac{p(\mathbf{x}|\boldsymbol{\theta}, H)p(\boldsymbol{\theta}|H)}{p(\mathbf{x})}, \quad (2.6)$$

em que  $p(\mathbf{x})$  é a densidade de probabilidade associada ao vetor de dados  $\mathbf{x}$  e na equação acima é um fator de proporcionalidade que garante que a integral do quociente seja igual a 1.

$$p(\mathbf{x}) = \int_{\boldsymbol{\theta}} p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}. \quad (2.7)$$

Como os dados são supostos conhecidos,  $p(\mathbf{x})$  é uma constante e não afeta a forma da função  $p(\boldsymbol{\theta}|\mathbf{x}, H)$ . Por essa razão é usual escrever o teorema de Bayes por uma relação de proporcionalidade (simbolizada por  $\propto$ ), deixando de fora  $p(\mathbf{x})$ :

$$p(\boldsymbol{\theta}|\mathbf{x}, H) \propto p(\mathbf{x}|\boldsymbol{\theta}, H)p(\boldsymbol{\theta}|H). \quad (2.8)$$

O valor de  $\boldsymbol{\theta}$  pode ser então estimado estabelecendo-se algum critério de otimização, sendo os mais comuns o BMSE (*Bayesian* MSE) e o MAP (*Maximum a Posteriori*) [12].

Pelo critério MAP, o valor estimado para os parâmetros é aquele que maximiza a distribuição *a posteriori*:

$$\boldsymbol{\theta}^{\text{MAP}} = \operatorname{argmax}_{\boldsymbol{\theta} \in \Theta} p(\boldsymbol{\theta}|\mathbf{x}, H). \quad (2.9)$$

Voltando ao exemplo do modelo linear, se pudermos considerar  $\boldsymbol{\theta}$  uma variável aleatória, o problema pode ser tratado pela abordagem bayesiana. Considerando uma priori gaussiana com média  $\mathbf{m}_{\boldsymbol{\theta}}$  e matriz de covariância  $\mathbf{C}_{\boldsymbol{\theta}}$ , o valor estimado pelo critério MAP é dado por [4]:

$$\mathbf{x}^{\text{MAP}} = (\mathbf{G}^T \mathbf{G} + \sigma_v^2 \mathbf{C}_{\boldsymbol{\theta}}^{-1})^{-1} (\mathbf{G}^T \mathbf{x} + \sigma_v^2 \mathbf{C}_{\boldsymbol{\theta}}^{-1} \mathbf{m}_{\boldsymbol{\theta}}). \quad (2.10)$$

Podemos ver na equação acima que se os elementos de  $\mathbf{C}_{\boldsymbol{\theta}}$  forem altos, as parcelas introduzidas pela priori se tornam irrelevantes, e então a solução MAP se aproxima da ML. Isso é razoável, uma vez que valores altos para  $\mathbf{C}_{\boldsymbol{\theta}}$  indicam que a priori é bastante “espalhada” e todos os valores são quase igualmente improváveis.

## 2.4 Modelo bayesiano hierárquico

A regra de Bayes é a ferramenta matemática que permite atualizar o conhecimento sobre os parâmetros, permitindo que os dados sejam levados em consideração. A aplicação desse procedimento seguidas vezes dá origem a um modelo hierárquico, em que a probabilidade *a posteriori* de um certo nível é a probabilidade *a priori* de um nível superior, como ilustra o gráfico da Figura 2.1.

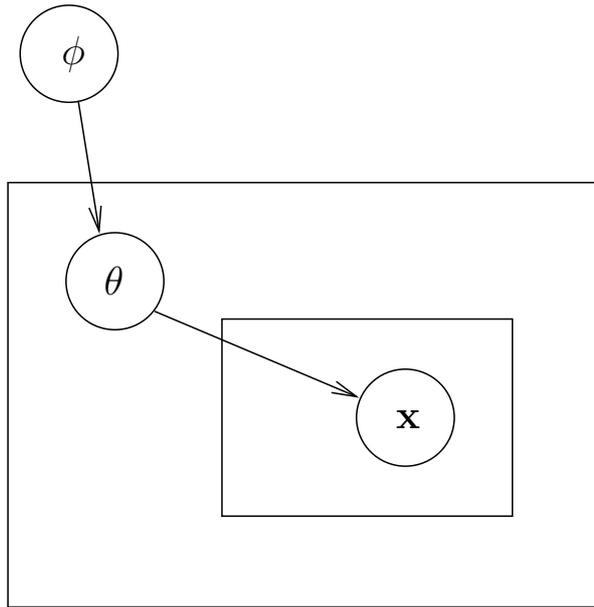


Figura 2.1: Exemplo de modelo bayesiano hierárquico

Considere a distribuição *a posteriori*  $p(\boldsymbol{\theta}|\mathbf{x})$ . A distribuição *a priori* de  $\boldsymbol{\theta}$  pode depender de outros parâmetros, digamos  $\boldsymbol{\phi}$ , que são também desconhecidos e podem ter uma probabilidade *a priori* associada. Aplicando a regra de Bayes duas vezes, temos:

$$p(\boldsymbol{\theta}, \boldsymbol{\phi}|\mathbf{x}) \propto p(\mathbf{x}|\boldsymbol{\theta}, \boldsymbol{\phi})p(\boldsymbol{\theta}|\boldsymbol{\phi})p(\boldsymbol{\phi}). \quad (2.11)$$

Podemos repetir o procedimento quantas vezes desejarmos, até obtermos uma priori que não depende de nenhum parâmetro desconhecido, usualmente uma priori não informativa (ver Seção 2.5.2).

## 2.5 Distribuição *a priori*

Um dos aspectos fundamentais da teoria de inferência bayesiana reside no conceito de distribuição *a priori*, ausente na abordagem freqüentista. Como afirmado, a distribuição *a priori* (simbolizada por  $p(\boldsymbol{\theta})$ ) é uma função que reflete o conhecimento do projetista sobre os possíveis valores dos parâmetros sem levar em consideração os dados.

Embora seja possível utilizar critérios freqüentistas para a escolha de  $p(\boldsymbol{\theta})$ , usualmente o conhecimento proveniente da experiência subjetiva do projetista é o fator preponderante para a escolha da probabilidade *a priori*. Diversas técnicas servem para guiar o projetista nessa tarefa, dentre as quais destacamos o método do histograma, o método da função de distribuição e o método de verossimilhança relativa [8].

Apesar de úteis, essas técnicas costumam gerar distribuições complicadas e de difícil tratamento analítico, o que prejudica o procedimento posterior de otimização.

### 2.5.1 Prioris conjugadas

Para garantir que a distribuição *a posteriori* seja analiticamente tratável, é comum o uso de prioris conjugadas, que são definidas como distribuições com estrutura algébrica similar à da verossimilhança. Isso garante que a posteriori seja do mesmo tipo da priori, mas com parâmetros modificados. Por exemplo, se a verossimilhança é gaussiana, a escolha de uma gaussiana como priori gera uma posteriori também gaussiana, uma distribuição com características que facilitam a otimização dos parâmetros.

Nesta tese, usamos prioris conjugadas definidas pelas distribuições gama e gama-inversa [14], ambas definidas por suas médias e variâncias e, além disso, conjugadas em relação à gaussiana. Mais detalhes sobre essas distribuições serão dados no Capítulo 4.

## 2.5.2 Priori não-informativa

Priori não-informativa, como o nome sugere, é uma distribuição que idealmente não contém qualquer informação sobre os possíveis valores dos parâmetros. A escolha aparentemente mais óbvia é uma distribuição uniforme, que associa igual probabilidade para todos os valores possíveis dos parâmetros, fazendo com que a posteriori coincida com a função de verossimilhança. No caso discreto, essa escolha não apresenta problemas; para variáveis contínuas, entretanto, uma distribuição uniforme é inadequada porque não é invariante em relação a uma transformação de variáveis um-para-um, isto é, se a distribuição de  $\boldsymbol{\theta}$  é uniforme, a distribuição de  $\boldsymbol{\phi} = f(\boldsymbol{\theta})$  será não-uniforme [8]. Esse resultado não é razoável, pois se não temos conhecimento sobre  $\boldsymbol{\theta}$ , o mesmo deve ocorrer com  $\boldsymbol{\phi}$ .

Para eliminar esse inconveniente, Jeffreys introduziu uma classe de distribuições invariantes a transformações um-para-um [8]. A priori de Jeffreys associada à variável  $\boldsymbol{\theta}$  é dada por:

$$p(\boldsymbol{\theta}) \propto |\mathbf{I}(\boldsymbol{\theta})|^{1/2}, \quad (2.12)$$

onde  $\mathbf{I}(\boldsymbol{\theta})$  é a informação de Fisher de  $\boldsymbol{\theta}$ , definida por:

$$I(\boldsymbol{\theta}) = E_{\mathbf{X}|\boldsymbol{\theta}} \left[ -\frac{\partial^2 \log(p(\mathbf{X}|\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}^2} \right], \quad (2.13)$$

que mostra que a priori depende apenas da verossimilhança.

No caso de modelos invariantes com relação à escala, obtém-se a seguinte prior de Jeffreys:

$$p(\boldsymbol{\theta}) \propto \frac{1}{\boldsymbol{\theta}}. \quad (2.14)$$

Há outras alternativas para a escolha de prioris não-informativas. Jaynes [15] propôs que o critério de máxima entropia fosse usado na especificação da distribuição *a priori*. Bernardo [16] também defende que a escolha da priori não-informativa deva ser feita através de critérios baseados na teoria da informação.

Na maioria dos casos, a distribuição gerada é imprópria, isto é, sua integral é diferente de 1. Esse fato não chega a ser um problema, visto que estamos interessados, no mais das vezes, na forma da distribuição, e não nos seus valores exatos.

## 2.6 Eliminação de parâmetros

É comum que o vetor de parâmetros contenha elementos que não são de interesse para estimação. Nesse caso, podemos integrar esses parâmetros indesejados (*nuisance parameters*) e então trabalhar apenas com os parâmetros de interesse. Se  $\boldsymbol{\theta}$  for particionado entre os parâmetros indesejados e os de interesse, de tal forma que  $\boldsymbol{\theta} = (\boldsymbol{\phi}, \boldsymbol{\psi})$ , a posteriori para os parâmetros de interesse é:

$$p(\boldsymbol{\phi}) = \int_{\boldsymbol{\psi}} p(\boldsymbol{\phi}, \boldsymbol{\psi} | \mathbf{x}). \quad (2.15)$$

Essa operação nem sempre pode ser feita analiticamente, especialmente nos casos em que a distribuição é multivariável e não possui uma forma conhecida. Nesses casos, os algoritmos que serão vistos no próximo capítulo permitem realizar a integral acima numericamente.

## 2.7 Teoria da decisão bayesiana

A teoria da decisão bayesiana [11] lida com o problema de classificação ou reconhecimento de padrões. Pressupõe-se a existência de um conjunto finito (ou pelo menos contável) de estados ocultos  $S$  e um conjunto de dados observados  $\mathbf{x}$ . O objetivo é selecionar, dentre todos os estados de  $S$ , aquele que “melhor” (num sentido a ser especificado) se adapta aos dados observados.

Pela abordagem bayesiana do problema, cada estado  $s_i$  tem associada a si uma probabilidade *a priori*  $P(s_i)$ . Tanto a função de verossimilhança  $l(\mathbf{x}; s_i) = p(\mathbf{x} | s_i)$  quanto a probabilidade *a priori* são supostas conhecidas. O teorema de Bayes pode ser então aplicado, gerando, para cada estado, a probabilidade *a posteriori*:

$$P(s_i | \mathbf{x}) = \frac{p(\mathbf{x} | s_i) P(s_i)}{p(\mathbf{x})}, i = 1, \dots, N, \quad (2.16)$$

em que  $N$  é o número de estados e a probabilidade total é dada por:

$$p(\mathbf{x}) = \sum_{i=1}^N p(\mathbf{x} | s_i) P(s_i). \quad (2.17)$$

Diversos critérios podem ser adotados para a obtenção do “melhor”  $s_i$ . O mais simples é o MAP, que define o estado escolhido como aquele com maior probabilidade

*a posteriori.*

$$s^{\text{MAP}} = \operatorname{argmax}_{s \in S} P(s_i | \mathbf{x}), \quad (2.18)$$

em que  $S = \{s_1, \dots, s_N\}$ .

Voltaremos a falar da teoria da decisão bayesiana no Capítulo 4, quando a aplicaremos ao problema de seleção das amostras corrompidas em um sinal de áudio.

## 2.8 Comentários conclusivos

Neste capítulo vimos os conceitos mais importantes da teoria de inferência bayesiana. Embora seja possível resolver problemas práticos apenas com esses conceitos, em boa parte das aplicações as distribuições envolvidas são multivariáveis e multi-modais e os problemas de otimização que surgem naturalmente são difíceis de ser resolvidos pelas técnicas clássicas de otimização. No capítulo seguinte, veremos alguns métodos e algoritmos indicados para essa tarefa.

## Capítulo 3

# Técnicas Numéricas de Estimação

A modelagem realística de sistemas físicos muitas vezes exige a utilização de modelos hierárquicos sofisticados, com distribuições bastante complexas e de difícil tratamento analítico. Para lidar com esses difíceis problemas de estimação, foram propostas diversas técnicas numéricas, dentre as quais destacam-se o algoritmo *Expectation-Maximization* (EM), o amostrador de Gibbs e os algoritmos Metropolis-Hastings (MH) e *Reversible-Jump* Metropolis-Hastings (RJ-MH). Com exceção do primeiro, esses algoritmos se baseiam em técnicas de Monte Carlo via cadeias de Markov (MCMC), um procedimento bastante poderoso que consiste em obter amostras de uma certa distribuição (em geral multivariável) através de uma cadeia de Markov e realizar a otimização usando técnicas de Monte Carlo a partir das amostras geradas. Este capítulo dedica-se à descrição de alguns desses métodos.

Começamos apresentando a idéia geral por trás de técnicas de Monte Carlo. Em seguida, descrevemos o método da rejeição, uma técnica simples de Monte Carlo usada para obtenção de amostras de uma distribuição, e que permitirá uma compreensão intuitiva dos algoritmos mais sofisticados que serão descritos posteriormente. Prosseguimos com a apresentação da teoria de cadeias de Markov (MC), base para os algoritmos que serão descritos em seguida. Nas Seções 3.4 e 3.5 apresentamos os algoritmos de amostragem de Gibbs e Metropolis-Hastings, dois dos mais populares algoritmos MCMC. Será descrita ainda uma modificação do algoritmo MH, proposta por Green [17], que permite lidar com modelos de dimensão variável. O capítulo termina com a apresentação de técnicas empíricas para diagnóstico de convergência

dos algoritmos.

Optamos por descrever os algoritmos de forma sucinta, nos moldes do trabalho de Andrieu et. al [18] e de Neal [19]. Como referência mais completa sobre o assunto, recomendamos os livros de Gamerman e Lopes [20] e Robert e Casella [21].

### 3.1 Técnicas de Monte Carlo

Técnicas de Monte Carlo consistem em gerar amostras i.i.d de uma certa distribuição e em seguida usá-las para obter uma aproximação de alguma característica da distribuição difícil (ou impossível) de obter analiticamente. Mais precisamente, de posse de um conjunto de amostras  $X = \{x^{(1)}, \dots, x^{(N)}\}$  de uma certa distribuição  $p(x)$ , podemos aproximar uma integral do tipo:

$$I(f) = \int_x f(x)p(x)dx \quad (3.1)$$

através de um somatório:

$$I_N(f) = \frac{1}{N} \sum_{i=1}^N f(x^{(i)}). \quad (3.2)$$

É possível mostrar que o estimador acima é não polarizado e, pela lei dos grandes números, converge para a integral da Equação (3.1) *quase certamente* quando  $N$  tende a infinito.

Obter amostras de uma distribuição diretamente nem sempre é trivial. As técnicas descritas nas próximas seções são formas de realizar essa tarefa indiretamente.

### 3.2 Método da rejeição

O método da rejeição é usado quando se deseja obter amostras de uma distribuição complexa usando uma distribuição auxiliar supostamente fácil de se amostrar.

Sejam  $\pi(x)$  a distribuição de interesse e  $q(x)$  uma distribuição que, para algum  $A$ , satisfaz  $Aq(x) \geq \pi(x), \forall x$ . O método da rejeição consiste em se gerar uma amostra de  $q(x)$ , digamos  $x^*$ , e aceitá-la com probabilidade  $\pi(x^*)/(Aq(x^*))$ .

Na prática, obtém-se uma amostra  $u$  de uma distribuição uniforme entre 0 e 1 e aceita-se  $x^*$  se  $u \leq \pi(x^*)/(Aq(x^*))$ .

A eficiência do método depende do valor de  $A$ , uma medida da “proximidade” entre  $q(x)$  e  $\pi(x)$ . Quanto maior o valor de  $A$ , maior é o percentual de amostras rejeitadas e conseqüentemente maior o número de iterações necessárias para se obter o conjunto de amostras exigidas para a estimação de Monte Carlo.

A principal desvantagem desse método é a dificuldade, em certos casos, de escolher uma distribuição auxiliar que leve a uma probabilidade de aceitação não muito baixa. Para distribuições multivariáveis essa tarefa é particularmente difícil, e nesses casos os algoritmos baseados em cadeias de Markov são a escolha mais indicada.

### 3.3 Cadeias de Markov

A teoria de cadeias de Markov é a base para os algoritmos MCMC que serão descritos mais adiante. Nesta seção, apresentaremos uma descrição sucinta da teoria e enunciaremos algumas propriedades importantes para o desenvolvimento e análise de algoritmos MCMC. Começamos considerando cadeias de Markov com espaço de estados discreto e em seguida estenderemos os resultados para o caso contínuo.

#### 3.3.1 Conceituação

Uma cadeia de Markov é um processo aleatório discreto no tempo que apresenta a *propriedade de Markov*, que estabelece que o estado atual da cadeia depende apenas do estado imediatamente anterior. Mais formalmente, seja  $X^{(n)}$  a variável aleatória que representa o estado da cadeia no instante  $n$  e seja  $S$  o espaço de estados para as variáveis  $X^{(n)}$ . Então:

$$P(X^{(n)} \in A^{(n)} | X^{(n-1)} \in A^{(n-1)}, \dots, X^{(0)} \in A^{(0)}) = P(X^{(n)} \in A^{(n)} | X^{(n-1)} \in A^{(n-1)}), \quad (3.3)$$

para quaisquer  $A^{(0)}, \dots, A^{(n)} \in S$ .

No caso em que  $S$  é um conjunto finito (ou ao menos contável), a cadeia de Markov é dita discreta. Seja  $S = \{s_1, \dots, s_N\}$ . As probabilidades condicionais

definem a *matriz de transição*,  $\mathbf{T}_n$ , cujo elemento da linha  $x$  e da coluna  $y$  é dado por:

$$T_n(x|y) = P(X^{(n)} = s_x | X^{(n-1)} = s_y). \quad (3.4)$$

É fácil ver que a soma dos elementos em cada linha de  $\mathbf{T}_n$  é igual a 1. Matrizes com essa propriedade são chamadas *matrizes estocásticas* e possuem uma série de propriedades importantes para a análise de algoritmos baseados em MCMC. Uma delas é a existência de pelo menos um autovalor igual a 1. No caso em que todos os elementos de  $\mathbf{T}_n$  são positivos, podemos garantir que todos os demais autovalores são distintos e menores que 1.

A probabilidade de  $X^{(n)}$ , denotada por  $P_n(x)$ , pode ser obtida a partir da probabilidade de  $X^{(n-1)}$  através de:

$$P_n(x) = \sum_{y \in S} T_n(x|y) P_{n-1}(y), \quad (3.5)$$

que pode ser escrita vetorialmente como:

$$\mathbf{P}_n = \mathbf{T}_n \mathbf{P}_{n-1}, \quad (3.6)$$

em que  $\mathbf{P}_n = [P_n(s_1) \dots P_n(s_n)]^T$ .

Se  $\mathbf{T}_n$  é independente de  $n$ , dizemos que a cadeia de Markov é *homogênea* e a matriz de transição passa a ser denotada simplesmente por  $\mathbf{T}$ . Nesse caso, a aplicação da Equação (3.6)  $n$  vezes fornece:

$$\mathbf{P}_n = \mathbf{T}^n \mathbf{P}_0, \quad (3.7)$$

em que  $\mathbf{P}_0$  é a distribuição do estado inicial da cadeia.

Para o desenvolvimento de algoritmos MCMC, a cadeia de Markov deve possuir as duas propriedades abaixo:

- Irredutibilidade: Partindo de qualquer estado, existir uma probabilidade não-nula de a cadeia mover-se para qualquer outro estado em um número finito de passos. Equivalentemente,  $T^n(x|y) > 0$ , para algum  $n$ .
- Aperiodicidade: A cadeia não ficar presa em ciclos.

### 3.3.2 Distribuição invariante

Uma distribuição é dita invariante (ou estacionária) se permanece fixa sob a aplicação da matriz de transição. No desenvolvimento de algoritmos MCMC, estamos interessados em construir cadeias de Markov que façam com que uma certa distribuição seja invariante. Se estivermos trabalhando com métodos bayesianos, a distribuição de interesse é a distribuição *a posteriori*.

Denotando a distribuição invariante por  $\Pi(x)$ , devemos ter:

$$\Pi(x) = \sum_{y \in S} T(x|y)\Pi(y). \quad (3.8)$$

Vetorialmente:

$$\mathbf{\Pi} = \mathbf{T}\mathbf{\Pi}. \quad (3.9)$$

Pela equação acima, vemos que  $\mathbf{\Pi}$  é um autovetor associado ao autovalor  $\lambda = 1$ . Para se determinar  $\mathbf{\Pi}$  unicamente, impõe-se a condição de que a soma de seus elementos seja igual a 1.

Para algoritmos MCMC, é usual impor a condição *detailed balance* para assegurar que uma dada distribuição é invariante. Essa condição estabelece que a probabilidade de a cadeia passar de um estado  $x$  no instante  $(n - 1)$  para o estado  $y$  no instante  $(n)$  é igual à probabilidade de a transição inversa ocorrer, isto é:

$$\Pi(y)T(x|y) = \Pi(x)T(y|x). \quad (3.10)$$

Para ver que  $\Pi(x)$  é distribuição invariante, basta aplicar o somatório em relação a  $y$  a ambos os membros da igualdade acima e perceber que o resultado é a Equação (3.8). Embora mais restritiva que a Equação (3.8), essa condição é mais simples de se impor para algoritmos MCMC.

### 3.3.3 Ergodicidade

Além de garantir que  $\Pi(x)$  seja a distribuição invariante desejada, devemos assegurar que  $P_n(x)$  convirja para  $\Pi(x)$  quando  $n$  tender a infinito, qualquer que seja a distribuição inicial  $P_0(x)$ . Nesse caso, dizemos que  $\Pi(x)$  é a distribuição-limite da cadeia.

Essa propriedade é conhecida como *ergodicidade*. Para garantir que a cadeia de Markov seja ergódica, é necessário que a cadeia seja aperiódica e irredutível. No caso discreto, isso equivale a ter os autovalores de  $\mathbf{T}$  todos distintos. Para ver isso, basta escrever a distribuição inicial usando os autovetores de  $\mathbf{T}$  como base e em seguida calcular a expressão  $\mathbf{T}^n \mathbf{P}_0$  [19].

$$\mathbf{P}_0 = \mathbf{\Pi} + c_2 \mathbf{v}_2 + \dots + c_N \mathbf{v}_N; \quad (3.11)$$

$$\mathbf{P}_n = \mathbf{T}^n \mathbf{P}_0 = \mathbf{\Pi} + c_2 \lambda_2^n \mathbf{v}_2 + \dots + c_N \lambda_N^n \mathbf{v}_N. \quad (3.12)$$

Na primeira equação usamos o fato de que  $\mathbf{\Pi}$  é o autovetor associado ao autovalor 1. Como todos os demais autovalores são menores que 1, concluímos que:

$$\lim_{n \rightarrow \infty} \mathbf{P}_n = \lim_{n \rightarrow \infty} \mathbf{T}^n \mathbf{P}_0 = \mathbf{\Pi}. \quad (3.13)$$

### 3.3.4 Cadeias de Markov para espaço de estados contínuo

Para o caso em que o espaço de estados  $S$  é contínuo, as propriedades descritas nas seções anteriores são expressas através de densidades de probabilidade. A propriedade de Markov é definida por:

$$p(x^{(n)} | x^{(n-1)}, \dots, x^{(0)}) = p(x^{(n)} | x^{(n-1)}). \quad (3.14)$$

O *núcleo de transição*  $K_n(x|y)$  no instante  $n$  é definido por:

$$K_n(x|y) = p_{X^{(n)}}(x | X^{(n-1)} = y), \quad (3.15)$$

em que  $p_{X^{(n)}}(x)$  denota a densidade de probabilidade da variável aleatória  $X^{(n)}$ .

Assim, a distribuição do estado da cadeia no instante  $n$  é dada por:

$$p_n(x) = \int_{y \in S} K_n(x|y) p_{n-1}(y) dy. \quad (3.16)$$

No caso de a cadeia ser homogênea,  $K_n$  independe de  $n$  e a condição *detailed balance* se torna:

$$\int_A \int_B K(x|y) \pi(y) dy dx = \int_B \int_A K(y|x) \pi(x) dx dy, \quad (3.17)$$

para quaisquer conjuntos  $A$  e  $B$  pertencentes a  $S$ .

Como no caso discreto, a condição *detailed balance* é suficiente para que  $\pi(x)$  seja uma distribuição invariante da cadeia de Markov definida por  $K(x|y)$ . Para garantir que a distribuição invariante seja também a distribuição-limite, a cadeia deve ser aperiódica e irredutível.

### 3.4 Amostrador de Gibbs

O amostrador de Gibbs (*Gibbs Sampler*) [22] é um caso especial de MCMC (*Markov Chain Monte Carlo*), indicado para os casos em que a distribuição conjunta é mais difícil de amostrar do que as condicionais. A técnica consiste em particionar a variável conjunta em diversos componentes (possivelmente multivariáveis) e obter amostras das distribuições condicionais de cada componente, considerando os demais fixos. O processo é repetido usando os últimos valores amostrados de cada componente como condicionante da distribuição dos demais componentes.

Seja  $\pi(\boldsymbol{\theta})$  a distribuição conjunta da qual se deseja obter amostras. A variável  $\boldsymbol{\theta}$  é então particionada em  $k$  componentes, de tal forma que  $\boldsymbol{\theta} = \{\theta_1, \dots, \theta_k\}$ . A  $i$ -ésima iteração do amostrador de Gibbs pode ser expressa como:

$$\begin{aligned}\theta_1^{(i)} &\sim \pi(\theta_1 | \theta_2^{(i-1)}, \dots, \theta_k^{(i-1)}) \\ \theta_2^{(i)} &\sim \pi(\theta_2 | \theta_1^{(i)}, \dots, \theta_k^{(i-1)}) \\ &\vdots \\ \theta_k^{(i)} &\sim \pi(\theta_k | \theta_1^{(i)}, \dots, \theta_{k-1}^{(i)}),\end{aligned}\tag{3.18}$$

em que o símbolo  $\sim$  indica que a variável da esquerda é uma amostra da distribuição à direita.

Para ver que  $\pi(\boldsymbol{\theta})$  é uma distribuição invariante em cada operação acima, vamos calcular a distribuição de  $\boldsymbol{\theta}$  após a primeira operação. Supondo  $p(\boldsymbol{\theta}^{(i-1)}) = \pi(\boldsymbol{\theta}^{(i-1)})$ :

$$\begin{aligned}p(\theta_1^{(i)}, \theta_2^{(i-1)}, \dots, \theta_k^{(i-1)}) &= \pi(\theta_1^{(i)} | \theta_2^{(i-1)}, \dots, \theta_k^{(i-1)}) \pi(\theta_2^{(i-1)}, \dots, \theta_k^{(i-1)}) \\ &= \pi(\theta_1^{(i)}, \theta_2^{(i-1)}, \dots, \theta_k^{(i-1)}).\end{aligned}\tag{3.19}$$

O mesmo raciocínio aplicado às operações seguintes indica que, a cada amostragem, a distribuição resultante permanece igual a  $\pi(\boldsymbol{\theta})$ . Sob condições fáceis de se obter na prática, mostra-se que a cadeia é ergódica, isto é, após a convergência, as amostras geradas em cada iteração correspondem a amostras da distribuição conjunta  $\pi(\boldsymbol{\theta})$ .

### 3.5 Algoritmo de Metropolis-Hastings

Nem sempre as distribuições condicionais necessárias para o amostrador de Gibbs são fáceis de se obter. Nesses casos, o algoritmo de Metropolis-Hastings (MH) é mais indicado. A idéia do algoritmo é semelhante à do método da rejeição, descrito na Seção 3.2. Obtêm-se amostras de uma distribuição auxiliar (supostamente mais simples que a distribuição de interesse) e decide-se aceitar ou rejeitar essa amostra dependendo de algum critério.

Pelo algoritmo MH, uma amostra  $x^*$  é obtida a partir de uma distribuição proposta, designada por  $q(x^*|x^{(i)})$ , em que  $x^{(i)}$  é o estado atual da cadeia de Markov. Essa amostra é aceita com uma probabilidade  $\alpha$  dada por:

$$\alpha(x^{(i)}, x^*) = \min \left( 1, \frac{\pi(x^*)q(x^{(i)}|x^*)}{\pi(x^{(i)})q(x^*|x^{(i)})} \right). \quad (3.20)$$

Se a amostra gerada for aceita, o novo estado da cadeia é  $x^{i+1} = x^*$ ; em caso contrário, a cadeia permanece no seu estado atual, isto é,  $x^{i+1} = x^{(i)}$ . O núcleo de transição é dado por:

$$K(x^{(i+1)}|x^{(i)}) = q(x^{(i+1)}|x^{(i)})\alpha(x^{(i)}, x^{(i+1)}) + \delta_{x^{(i)}}(x^{(i+1)})r(x^{(i)}), \quad (3.21)$$

em que

$$r(x^{(i)}) = \int_{x^* \in S} q(x^*|x^{(i)}) (1 - \alpha(x^{(i)}, x^*)) dx^* \quad (3.22)$$

é a probabilidade de a cadeia permanecer no estado atual. A expressão  $\delta_{x^{(i)}}(x^{(i+1)})$  é a função impulso unitário (delta de Dirac) localizada em  $x^{(i)}$  e aplicada em  $x^{(i+1)}$ , que indica uma distribuição “concentrada” em  $x^{(i)}$ .

Podemos verificar que esse núcleo de transição satisfaz a condição *detailed balance*, e portanto  $\pi(x)$  é uma distribuição invariante da cadeia. Para garantir

que  $\pi(x)$  é também a distribuição-limite, temos que verificar a irreduzibilidade e aperiodicidade da cadeia. Como o algoritmo sempre permite a rejeição, segue que a cadeia é aperiódica. Para assegurar irreduzibilidade, é preciso que o suporte de  $q(\cdot)$  inclua o suporte de  $\pi(\cdot)$  [18].

A eficiência do algoritmo MH depende fundamentalmente da escolha da proposta,  $q(\cdot)$ . É usual escolher como proposta uma gaussiana centrada no estado atual, isto é,  $q(x^*|x^{(i)}) = N(x^*|x^{(i)}, \sigma_q^2 I)$ . A escolha da variância  $\sigma_q^2$  é crucial. Se  $q(\cdot)$  é muito estreita, apenas estados próximos ao máximo de  $\pi(x)$  são visitados. Por outro lado, se  $q(\cdot)$  é muito ampla, o percentual de amostras rejeitadas é muito alto e, conseqüentemente, a cadeia será altamente correlacionada, invalidando a hipótese de independência das amostras, necessária para a estimação de Monte Carlo. Para resolver esse problema, recomenda-se descartar uma fração das amostras, reduzindo assim a correlação entre elas, mas, por outro lado, aumentando o tempo para convergência.

### 3.6 MCMC com saltos reversíveis

Para tratar o problema de seleção de modelos de diferentes dimensões, Green [17] propôs uma modificação no algoritmo MH que ficou conhecida como MCMC com saltos reversíveis (RJ-MCMC), que chamaremos neste texto de RJ-MH. Será apresentada aqui a derivação alternativa mais simples do algoritmo RJ-MH proposta por Godsill [23].

O problema de seleção de modelos consiste em determinar, dentre um conjunto de modelos escolhido inicialmente, o mais adequado para a descrição dos dados  $x$ . Vamos considerar um conjunto de  $K$  modelos  $\{M_1, \dots, M_K\}$ , cada um especificado respectivamente pelos vetores de parâmetros  $\theta_1, \dots, \theta_K$ . O objetivo é obter amostras da probabilidade condicional do índice  $k \in \mathcal{K} = \{1, \dots, K\}$  de cada modelo,  $P(k|x)$ , e usar técnicas de Monte Carlo para selecionar o índice de maior probabilidade. Se as dimensões dos componentes  $\theta_K$  forem iguais, os algoritmos vistos até agora podem ser usados para obtenção de amostras de  $P(k|x)$ , e o valor de  $k$  pode ser obtido de acordo com algum critério de otimização. No caso mais

geral em que  $\dim(\boldsymbol{\theta}_i) \neq \dim(\boldsymbol{\theta}_j)$  para  $i \neq j$ , o amostrador de Gibbs e o algoritmo de Metropolis-Hastings não podem ser usados, pois estes exigem dimensionalidade constante.

Godsill começa definindo o espaço composto de parâmetros:

$$U = \mathcal{K} \times \Theta_1 \times \Theta_2 \times \dots \times \Theta_K, \quad (3.23)$$

sendo  $\Theta_k$  o domínio de  $\boldsymbol{\theta}_k$ , para  $k \in \mathcal{K}$ . O vetor  $(k, \boldsymbol{\theta}) \in U$  contém o índice do modelo  $k$  e  $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K\}$ , que é composto pelos parâmetros de todos os modelos, incluindo os que não são usados pelo modelo  $k$ . Neste espaço, a dimensionalidade é fixa e o algoritmo MH padrão pode ser aplicado.

Assumindo que  $p(\boldsymbol{\theta}_k|k, \boldsymbol{\theta}_{-k}) = p(\boldsymbol{\theta}_k|k)$ , em que  $\boldsymbol{\theta}_{-k}$  são os parâmetros não usados pelo modelo  $k$ , e  $p(x|k, \boldsymbol{\theta}) = p(x|k, \boldsymbol{\theta}_k)$ , a distribuição *a posteriori* é dada por [23]:

$$p(k, \boldsymbol{\theta}|x) = \frac{p(x|k, \boldsymbol{\theta}_k)p(\boldsymbol{\theta}_k|k) \left( \prod_{i \neq k} p(\boldsymbol{\theta}_i|k) \right) p(k)}{p(x)}. \quad (3.24)$$

A aplicação do algoritmo MH padrão ao modelo acima usando uma proposta do tipo

$$q(k^*, \boldsymbol{\theta}^*|k, \boldsymbol{\theta}) = q_1(k^*|k)q_2(\boldsymbol{\theta}_{k^*}^*|\boldsymbol{\theta}_k)p(\boldsymbol{\theta}_{-k^*}^*|\boldsymbol{\theta}_{k^*}^*, k^*) \quad (3.25)$$

leva a uma probabilidade de aceitação

$$\alpha(k^{(i)}, \boldsymbol{\theta}_{k^{(i)}}^{(i)}; k^*, \boldsymbol{\theta}_{k^*}^*) = \min \left( 1, \frac{p(k^*, \boldsymbol{\theta}_{k^*}^*|x)q_1(k^{(i)}|k^*)q_2(\boldsymbol{\theta}_{k^{(i)}}^{(i)}|\boldsymbol{\theta}_{k^*}^*)}{p(k^{(i)}, \boldsymbol{\theta}_{k^{(i)}}^{(i)}|x)q_1(k^*|k^{(i)})q_2(\boldsymbol{\theta}_{k^*}^*|\boldsymbol{\theta}_{k^{(i)}}^{(i)})} \right), \quad (3.26)$$

que só depende dos parâmetros usados pelo modelo  $k$ . Essa é a mesma equação obtida por Green usando um procedimento mais sofisticado.

Considerando o caso em que alguns dos parâmetros de diferentes modelos têm o mesmo significado, é possível incorporar informações úteis dos parâmetros atuais para a gerações dos novos parâmetros. Green propõe que variáveis auxiliares  $\mathbf{u}^*$  e  $\mathbf{u}$  sejam concatenadas aos parâmetros  $\boldsymbol{\theta}_{k^{(i)}}^{(i)}$  e  $\boldsymbol{\theta}_{k^*}^*$  de tal forma que os novos vetores  $(\boldsymbol{\theta}_{k^{(i)}}^{(i)}, \mathbf{u}^*)$  e  $(\boldsymbol{\theta}_{k^*}^*, \mathbf{u})$  tenham a mesma dimensão. É preciso ainda estabelecer uma bijeção determinística  $(\boldsymbol{\theta}_{k^*}^*, \mathbf{u}) = g(\boldsymbol{\theta}_{k^{(i)}}^{(i)}, \mathbf{u}^*)$  para assegurar a reversibilidade da cadeia. O vetor  $\mathbf{u}^*$  é gerado por amostragem a partir de uma distribuição  $q_2(\mathbf{u}^*)$

e os novos parâmetros são obtidos através da transformação acima. Nesse caso, a probabilidade de aceitação se torna [17]:

$$\alpha(k^{(i)}, \boldsymbol{\theta}_{k^{(i)}}^{(i)}; k^*, \boldsymbol{\theta}_{k^*}^*) = \min \left( 1, \frac{p(k^*, \boldsymbol{\theta}_{k^*}^* | x) q_1(k^{(i)} | k^*) q_2(\mathbf{u}^{(i)})}{p(k^{(i)}, \boldsymbol{\theta}_{k^{(i)}}^{(i)} | x) q_1(k^* | k^{(i)}) q_2(\mathbf{u}^*)} \left| \frac{\partial g(\boldsymbol{\theta}_{k^*}^*, \mathbf{u})}{\partial (\boldsymbol{\theta}_{k^{(i)}}^{(i)}, \mathbf{u}^*)} \right| \right), \quad (3.27)$$

em que o jacobiano da transformação leva em conta a mudança de medida decorrente da transformação.

A escolha da distribuição  $q_2(\mathbf{u})$  e da transformação determinística é crucial para a velocidade de convergência do algoritmo. Essa tarefa deve considerar as características particulares das distribuições usadas.

### 3.7 Análise de convergência

Os algoritmos apresentados até agora garantem que os estados da cadeia de Markov sejam amostras da distribuição-limite quando o número de iterações tende a infinito. Como na prática precisamos usar um número finito de iterações, é importante saber quando a cadeia está suficientemente próxima da distribuição limite para que suas amostras possam ser usadas para estimação de Monte Carlo.

Muito trabalho tem sido feito em torno desse problema. Métodos teóricos para diagnóstico de convergência têm sido propostos, mas os resultados tiveram até o momento pouco impacto prático [20]. Na prática, usam-se mais frequentemente métodos informais de convergência, que consistem de análise estatística dos dados gerados pela cadeia. Embora simples, essas técnicas não permitem garantir a convergência de forma geral.

Dentre os métodos informais, três abordagens podem ser usadas. A primeira baseia-se na realização  $n$  cadeias paralelas. Calcula-se o histograma de  $n$  amostras na  $m$ -ésima iteração e se o compara com o histograma obtido  $k$  iterações adiante. Se os histogramas forem bastante similares, considera-se que a convergência foi atingida. O valor de  $k$  deve ser grande o bastante para evitar que a correlação entre os estados sucessivos da cadeia causem uma falsa impressão de similaridade entre os histogramas.

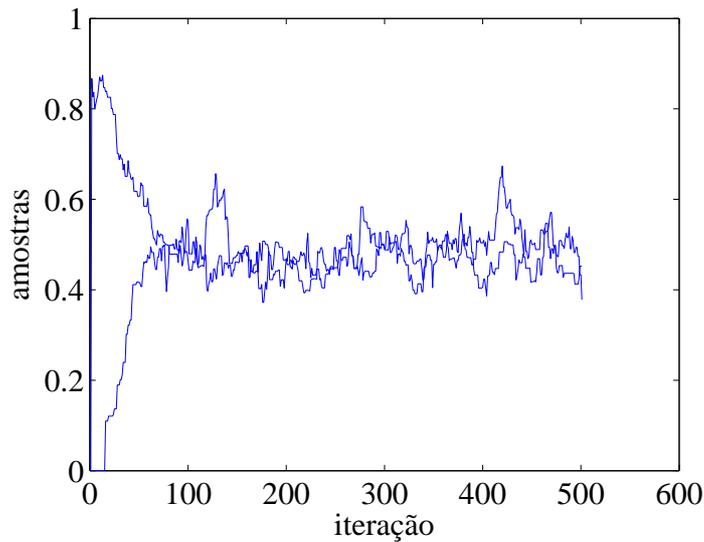


Figura 3.1: Evolução de um parâmetro de uma cadeia de Markov a partir de diferentes valores iniciais.

Outro método baseia-se na construção de uma única cadeia e no cálculo da média ergódica das amostras obtidas. Espera-se que após a convergência, a média se aproxime de um certo valor constante. Portanto, através da análise visual do gráfico das médias, é possível diagnosticar a convergência.

A terceira técnica consiste na análise visual da evolução dos parâmetros da cadeia. Após a convergência, esses parâmetros devem exibir um padrão constante, que pode ser usado para diagnosticá-la. Analisando a Figura 3.1, que mostra a evolução de um parâmetro de uma cadeia de Markov partindo de pontos iniciais diferentes, podemos concluir que a convergência ocorre em torno da iteração 50.

### 3.8 Conclusões

Esse capítulo fez uma breve exposição teórica de técnicas numéricas indicadas para exploração de distribuições multidimensionais de formato complicado (por exemplo, multi-modal). Em particular, o amostrador de Gibbs e os algoritmos de Metropolis-Hastings e Metropolis-Hastings com saltos reversíveis foram descritos com algum detalhe. No Capítulo 5 veremos como essas técnicas podem ser usadas

para a remoção de *clicks* em sinais de áudio.

# Capítulo 4

## Alguns Métodos de Correção de *Clicks*

Os *clicks* são um dos defeitos mais comuns em gravações de áudio antigas. São principalmente causados por partículas de poeira ou por pequenos arranhões presentes na superfície do disco. Caracterizam-se por ter tanto localização quanto duração aleatórias, amplitude em geral menor ou comparável à do sinal e curta duração (menor que 1 ms). É comum que *clicks* estejam presentes com grande frequência ao longo do sinal: uma gravação antiga pode ter até 20% de suas amostras corrompidas por *clicks*, podendo 2% ser já um percentual que causa bastante incômodo. Dependendo de sua frequência de ocorrência e de sua amplitude, sua impressão pode não ser individualizada pelo ouvido humano.

Uma das primeiras técnicas propostas para restauração digital de *clicks* descreve o sinal por um modelo autorregressivo (AR) [4], [24] tradicionalmente usado para sinais de áudio, e trata o ruído (*click*) como uma perturbação no sinal que não pode ser descrita por esse modelo. Essa técnica, denominada filtragem inversa, permite determinar as amostras corrompidas, e a restauração do sinal pode ser feita por interpolação. Embora computacionalmente eficiente, a técnica não explora as propriedades estatísticas desse tipo particular de ruído, que poderiam ser usadas para gerar um resultado de melhor qualidade.

Em [4] e [25] é descrito um método de restauração em que tanto o sinal original (não-corrompido) quanto o ruído são modelados estatisticamente, e o problema é

tratado de acordo com o paradigma bayesiano. As amostras corrompidas são descritas pela soma do sinal original e o ruído, cuja amplitude é modelada como gaussiana de média zero e variância constante. Essa técnica exige que se determine para cada amostra se o ruído está presente ou ausente, o que nos casos práticos gera um custo computacional elevado.

Nas próximas seções, apresentamos a modelagem matemática do sinal e do ruído e alguns métodos bayesianos para correção de *clicks* que serão usados como padrão de comparação em relação às técnicas propostas nesta dissertação.

## 4.1 Modelagem do sinal corrompido

Trabalhos anteriores nesta área utilizam um modelo para o sinal em que o *click* é considerado como ruído aditivo presente em apenas algumas amostras do sinal. Considerando  $x(n)$  o sinal original (que foi gerado acusticamente) e  $v(n)$  o ruído, o sinal corrompido é escrito como:

$$y(n) = x(n) + i(n)v(n), \quad (4.1)$$

em que  $i(n)$  determina se a  $n$ -ésima amostra está corrompida ou correta: se  $i(n) = 0$ , a amostra está correta; se  $i(n) = 1$ , a amostra está corrompida.

O objetivo do sistema de restauração é estimar o sinal  $x(n)$  sendo conhecido o sinal  $y(n)$ . Usualmente, isso é feito em duas etapas: detecção e interpolação. Na detecção, o objetivo é decidir quais amostras estão corrompidas, ou seja, estimar o sinal  $i(n)$ . A tarefa da interpolação é estimar o sinal  $x(n)$ , sendo conhecidos os sinais  $y(n)$ , diretamente observado, e  $i(n)$ , obtido pelo procedimento de detecção.

O sinal original  $x(n)$  é usualmente descrito por um modelo autorregressivo (AR). Em alguns trabalhos, atribui-se um modelo estatístico para o ruído,  $v(n)$ . Outros usam apenas o fato de que o ruído é uma perturbação que não pode ser descrita pelo modelo especificado para o sinal. Nesse caso, a parte do sinal considerada corrompida é tratada como perdida (*missing data*) e a correlação existente no sinal de áudio é explorada para se determinar essas porções do sinal [4]. Na próxima seção, veremos detalhes sobre a modelagem do sinal de áudio e em seguida serão relatados os trabalhos anteriores relevantes para esta tese.

## 4.2 Modelagem do sinal de áudio

Sinais de áudio são tradicionalmente descritos por um modelo autorregressivo, que interpreta o sinal  $x(n)$  como a saída de um filtro só-polos excitado por ruído branco  $e(n)$ :

$$x(n) = \sum_{i=1}^P a_i x(n-i) + e(n). \quad (4.2)$$

O vetor  $\mathbf{a} = [a_1, a_2, \dots, a_P]^T$  contém os coeficientes do filtro só-polos de ordem  $P$ . Assumindo que a distribuição de  $e(n)$  é conhecida, a distribuição de  $x(n)$  condicionada às  $P$  amostras anteriores de  $x(n)$ , é a distribuição de  $e(n)$  com média deslocada de

$$\sum_{i=1}^P a_i x(n-i). \quad (4.3)$$

Sendo assim, a distribuição de  $x(n)$  é dada por:

$$p(x(n)|x(n-1), x(n-2), \dots, x(n-P)) = p_e \left( x(n) - \sum_{i=1}^P a_i x(n-i) \right). \quad (4.4)$$

Como a excitação  $e(n)$ , por hipótese, é composta de amostras estatisticamente independentes, podemos obter a distribuição conjunta das amostras  $x(P+1), x(P+2), \dots, x(N)$ , sendo  $N$  o tamanho de um bloco do sinal, condicionada às amostras  $x(1), x(2), \dots, x(P)$ :

$$\begin{aligned} p(x(P+1), x(P+2), \dots, x(N)|x(1), x(2), \dots, x(P)) &= \\ &= \prod_{n=P+1}^N p_e \left( x(n) - \sum_{i=1}^P a_i x(n-i) \right). \end{aligned} \quad (4.5)$$

Definindo os vetores

$$\mathbf{x} = [x(1), x(2), \dots, x(N)]^T, \quad (4.6)$$

$$\mathbf{x}_0 = [x(1), x(2), \dots, x(P)]^T, \quad (4.7)$$

$$\mathbf{x}_1 = [x(P+1), x(P+2), \dots, x(N)]^T \text{ e} \quad (4.8)$$

$$\mathbf{a} = [a_1, a_2, \dots, a_P]^T, \quad (4.9)$$

podemos reescrever a Equação (4.2) de forma vetorial:

$$\mathbf{x}_1 = \mathbf{X}\mathbf{a} + \mathbf{e}, \quad (4.10)$$

em que a matrix  $\mathbf{X}$  é dada por:

$$\mathbf{X} = \begin{bmatrix} x(P) & x(P-1) & \dots & x(2) & x(1) \\ x(P+1) & x(P) & \dots & x(3) & x(2) \\ \vdots & & \ddots & & \vdots \\ x(N-2) & x(N-3) & \dots & x(N-P) & x(N-P-1) \\ x(N-1) & x(N-2) & \dots & x(N-P+1) & x(N-P) \end{bmatrix}. \quad (4.11)$$

A Equação (4.5) pode então ser escrita de forma vetorial:

$$p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{a}) = p_e(\mathbf{x}_1 - \mathbf{X}\mathbf{a}). \quad (4.12)$$

Se assumirmos excitação gaussiana de média zero e variância  $\sigma_e^2$ , teremos:

$$p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{a}) = \frac{1}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}} e^{-\frac{1}{2\sigma_e^2}(\mathbf{x}_1 - \mathbf{X}\mathbf{a})^T(\mathbf{x}_1 - \mathbf{X}\mathbf{a})}. \quad (4.13)$$

Se o vetor  $\mathbf{x}$  for conhecido, a maximização da expressão acima fornece o estimador ML para os coeficientes  $\mathbf{a}$ :

$$\mathbf{a}^{\text{ML}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{x}_1. \quad (4.14)$$

A Equação (4.12) pode ser escrita num formato equivalente, que será útil na derivação dos algoritmos de remoção de *clicks*. Definindo a matriz  $\mathbf{A}$  de  $(N-P)$  linhas e  $P$  colunas por:

$$\mathbf{A} = \begin{bmatrix} -a_P & \dots & a_1 & 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & -a_P & \dots & -a_1 & 1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \dots & 0 & 0 & -a_P & \dots & -a_1 & 1 & 0 & 0 \\ 0 & \dots & 0 & 0 & -a_P & \dots & -a_1 & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 & -a_P & \dots & -a_1 & 1 \end{bmatrix}, \quad (4.15)$$

podemos escrever:

$$\mathbf{e} = \mathbf{A}\mathbf{x}, \quad (4.16)$$

e a distribuição condicional de  $\mathbf{x}_1$  se torna:

$$p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{a}) = \frac{1}{(2\pi\sigma_e^2)^{\frac{(N-P)}{2}}} e^{-\frac{1}{2\sigma_e^2} \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}}. \quad (4.17)$$

Pode-se mostrar [4] que a distribuição conjunta de  $\mathbf{x}_1$  e  $\mathbf{x}_0$  (isto é,  $\mathbf{x}$ ) é dada por:

$$p(\mathbf{x}|\mathbf{a}) = \frac{1}{(2\pi\sigma_e^2)^{\frac{N}{2}} |\mathbf{M}_{x_0}|^{1/2}} e^{-\frac{1}{2\sigma_e^2} \mathbf{x}^T \mathbf{M}_x \mathbf{x}}, \quad (4.18)$$

em que

$$\mathbf{M}_x^{-1} = \mathbf{A}^T \mathbf{A} + \begin{bmatrix} \mathbf{M}_{x_0}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad (4.19)$$

sendo  $\mathbf{M}_{x_0}$  a matriz de covariância para  $P$  amostras de dados obtidos de um processo AR com vetor de coeficientes  $\mathbf{a}$  e excitação com variância unitária.

Num processamento feito bloco a bloco, podem-se considerar conhecidas as  $P$  primeiras amostras e usar a Equação (4.18) para obtenção da verossilhança. Se isso não puder ser feito, usa-se a aproximação:

$$p(\mathbf{x}|\mathbf{a}) \approx p(\mathbf{x}_1|\mathbf{x}_0, \mathbf{a}), \quad (4.20)$$

que será tão mais precisa quanto maior for a diferença entre  $N$  e  $P$ .

### 4.3 Remoção de *clicks* por filtragem inversa

Alguns dos primeiros trabalhos de remoção digital de *clicks* são baseados em detecção por filtragem inversa. Nessa técnica, é atribuído ao sinal de áudio um modelo autorregressivo cujos parâmetros são obtidos diretamente a partir do sinal corrompido. Considerando que no sinal original o erro de predição é identicamente distribuído (isto é, possui média e variância independentes do tempo), pode-se, através do erro, estimar as amostras corrompidas. Um erro de predição alto indica que a amostra correspondente não pôde ser bem descrita pelo modelo do sinal, e portanto deve estar corrompida.

Na prática, divide-se o sinal em blocos e determina-se o erro de predição para cada bloco através da filtragem inversa do bloco de sinal ( $\mathbf{e} = \mathbf{A}\mathbf{y}$ ) e consideram-se corrompidas as amostras com erro de predição associado acima de um certo limiar, que depende da potência do bloco de sinal analisado.

Para a descrição matemática do método, é conveniente escrever os sinais em vetores contendo amostras nas posições correspondentes ao sinal corrompido e nas posições em que o sinal está correto. Seguindo a notação de [4], usamos o sub-índice  $-i$  para indicar as amostras intactas e o sub-índice  $i$  para as amostras corrompidas.

Por exemplo, se  $\mathbf{x}$  denota o vetor com as amostras do sinal original,  $\mathbf{x}_{-i}$  e  $\mathbf{x}_i$  são vetores que contêm as amostras corretas e corrompidas do sinal, respectivamente. Sendo assim, podemos escrever:

$$\mathbf{x} = \mathbf{U}\mathbf{x}_i + \mathbf{K}\mathbf{x}_{-i}, \quad (4.21)$$

em que as matrizes  $\mathbf{U}$  e  $\mathbf{K}$  definem as posições dos elementos de  $\mathbf{x}_i$  e  $\mathbf{x}_{-i}$  no sinal original.

Para um sinal de 5 amostras, estando as amostras 1 e 3 corrompidas e as demais corretas, teremos  $\mathbf{x}_i = [x(1) \ x(3)]^T$  e  $\mathbf{x}_{-i} = [x(0) \ x(2) \ x(4)]^T$ . Nesse caso:

$$\mathbf{U} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.22)$$

e

$$\mathbf{K} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}. \quad (4.23)$$

Uma vez determinado o sinal  $i(n)$ , uma estimativa para as amostras corrompidas pode ser obtida calculando-se inicialmente a distribuição condicional:

$$p(\mathbf{x}_i|\mathbf{x}_{-i}) = \frac{p(\mathbf{x})}{p(\mathbf{x}_{-i})} = \frac{p(\mathbf{U}\mathbf{x}_i + \mathbf{K}\mathbf{x}_{-i})}{p(\mathbf{x}_{-i})}. \quad (4.24)$$

A distribuição de  $p(\mathbf{x})$  é dada por:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{N/2}|\mathbf{R}_{\mathbf{x}}|^{1/2}} e^{(-\frac{1}{2}\mathbf{x}^T\mathbf{R}_{\mathbf{x}}^{-1}\mathbf{x})}, \quad (4.25)$$

em que  $\mathbf{R}_x = E[\mathbf{x}\mathbf{x}^T]$ .

Como a distribuição é gaussiana, os estimadores MAP e MMSE levam ao mesmo resultado, que é obtido igualando-se a zero o gradiente de:

$$\mathbf{x}^T \mathbf{R}_x^{-1} \mathbf{x} = (\mathbf{U}\mathbf{x}_i + \mathbf{K}\mathbf{x}_{-i})^T \mathbf{R}_x^{-1} (\mathbf{U}\mathbf{x}_i + \mathbf{K}\mathbf{x}_{-i}). \quad (4.26)$$

Com isso, obtemos

$$\mathbf{x}_i^{\text{MAP}} = -\mathbf{M}_{ii}^{-1} \mathbf{M}_{i-i} \mathbf{x}_{-i}, \quad (4.27)$$

em que

$$\mathbf{M}_{ii} = \mathbf{U}^T \mathbf{R}_x^{-1} \mathbf{U} \quad (4.28)$$

e

$$\mathbf{M}_{i-i} = \mathbf{U}^T \mathbf{R}_x^{-1} \mathbf{K}. \quad (4.29)$$

Esse resultado não leva em conta as amostras corrompidas  $\mathbf{y}_i$ . Veremos a seguir um método que atribui um modelo ao ruído e permite levar em conta as amostras corrompidas.

## 4.4 Correção de *clicks* pelo método bayesiano

Através do paradigma bayesiano, é possível incorporar um modelo estatístico ao processo de geração de *clicks*, e assim tratar o problema de sua detecção pela teoria de decisão bayesiana, e usar a teoria de estimação bayesiana para restauração do sinal original. Nas próximas seções, daremos detalhes dessa abordagem, de acordo com o apesendo em [4].

### 4.4.1 Estimação em blocos

Um bloco de sinal corrompido por *clicks* pode ser descrito vetorialmente da seguinte forma:

$$\mathbf{y} = \mathbf{x} + \mathbf{iv}. \quad (4.30)$$

Apenas o vetor  $\mathbf{y}$  é conhecido. Inicialmente queremos detectar as amostras corrompidas, isto é, estimar o vetor  $\mathbf{i}$ . Pelo paradigma bayesiano, formamos a distribuição *a posteriori*:

$$P(\mathbf{i}|\mathbf{y}) = \frac{P(\mathbf{y}|\mathbf{i})P(\mathbf{i})}{p(\mathbf{y})}, \quad (4.31)$$

em que  $P(\mathbf{i})$  é a probabilidade *a priori* de  $\mathbf{i}$ , que será discutida mais adiante.

Como o sinal  $y(n)$  é conhecido, o denominador da expressão acima é independente de  $\mathbf{i}$  e pode ser desconsiderado no processo de otimização.

Considerando o sinal  $\mathbf{x}$  descrito por um modelo AR e o ruído gaussiano com matriz de covariância  $\mathbf{R}_v$ , mostra-se [4] que a verossimilhança é dada pelo formulário abaixo (sendo  $l = \mathbf{i}^T \mathbf{i}$  o número de amostras corrompidas):

$$p(\mathbf{y}|\mathbf{i}) = \frac{\sigma_e^l e^{\left(-\frac{1}{2\sigma_e^2} E_{\text{MIN}}\right)}}{(2\pi\sigma_e^2)^{\frac{N-P}{2}} |\mathbf{R}_v|^{1/2} |\mathbf{\Phi}|^{1/2}}, \quad (4.32)$$

em que

$$E_{\text{MIN}} = E_0 - \mathbf{\Theta}^T \mathbf{x}_i^{\text{MAP}}, \quad (4.33)$$

$$\mathbf{x}_i^{\text{MAP}} = \mathbf{\Phi}^{-1} \mathbf{\Theta}, \quad (4.34)$$

$$\mathbf{\Phi} = \mathbf{A}_i^T \mathbf{A}_i + \sigma_e^2 \mathbf{R}_v^{-1}, \quad (4.35)$$

$$\mathbf{\Theta} = -(\mathbf{A}_i^T \mathbf{A}_{-i} \mathbf{y}_i - \sigma_e^2 \mathbf{R}_v^{-1} \mathbf{y}_i), \quad (4.36)$$

$$E_0 = \mathbf{y}_{-i}^T \mathbf{A}_{-i}^T \mathbf{A}_{-i} \mathbf{y}_{-i} + \sigma_e^2 \mathbf{y}_i^T \mathbf{R}_v^{-1} \mathbf{y}_i. \quad (4.37)$$

Considerando o ruído i.i.d. gaussiano de média zero e variância  $\sigma_v^2$ , teremos  $\mathbf{R}_v = \sigma_v^2 \mathbf{I}_l$ . Essa hipótese torna simples a implementação do algoritmo, mas nem sempre é adequada para casos práticos, haja vista a ocorrência comum de *clicks* de amplitudes variadas ao longo do sinal. O valor de  $\sigma_v^2$  precisa ser estimado inicialmente através de algum pré-processamento.

Para obtenção da localização das amostras corrompidas pela teoria de detecção bayesiana, precisamos calcular as probabilidades *a posteriori* para todos os possíveis valores de  $\mathbf{i}$  e selecionar o maior. O sinal restaurado é obtido diretamente da Equação (4.34), considerando o vetor  $\mathbf{i}$  selecionado na etapa de detecção.

Como cada elemento de  $\mathbf{i}$  pode ser 0 ou 1, para um bloco de tamanho  $N$ , existem  $2^N$  candidatos a considerar. Para um bloco de 1024 amostras (tamanho típico), o número de candidatos tornaria o método impraticável.

#### 4.4.2 Estimação seqüencial

Uma maneira de se tornar a detecção mais eficiente é fazê-la de forma seqüencial, o que ainda lhe atribui a vantagem de poder ser realizada em tempo real, à medida que novas amostras são geradas.

O procedimento consiste em se estimar a probabilidade *a posteriori* para os  $n$  primeiros elementos de  $\mathbf{i}$ , a partir da probabilidade *a posteriori* dos  $(n-1)$  elementos de  $\mathbf{i}$ . Imaginemos que, no instante  $n$ , tenhamos um conjunto de  $C$  candidatos para o vetor de detecção  $\mathbf{i}$ . Para cada um dos candidatos, teremos, no instante  $(n+1)$ , dois novos candidatos: um com o novo elemento igual a 1, e outro como novo elemento igual a 0. Isso faz com que a cada iteração o número de candidatos dobre, o que geraria um crescimento exponencial indesejável. Para se reduzir a complexidade, adotam-se técnicas de eliminação de candidatos menos prováveis. Os testes que realizamos mostram que é possível obter bons resultados, usando-se um conjunto de apenas 10 candidatos. Mais detalhes sobre essas técnicas e os critérios que podem ser usados são apresentadas em [4].

#### 4.4.3 Estimação iterativa através do algoritmo EM e do amostrador de Gibbs

Um problema dos métodos das seções anteriores é a necessidade de se estimar os parâmetros dos modelos do sinal e do ruído a partir do sinal corrompido, o que pode prejudicar a qualidade da restauração. É possível contornar esse problema através do algoritmo EM [26] e métodos baseados em MCMC, como o amostrador de Gibbs, por exemplo. Esses métodos permitem que os parâmetros do modelo sejam numericamente marginalizados.

O algoritmo EM permite fazer a estimação do sinal original, considerando o vetor de detecção,  $\mathbf{i}$ , conhecido. A função a ser maximizada é a distribuição  $a$

*posteriori*:

$$\mathbf{x}_i^{\text{MAP}} = \operatorname{argmax}_{\mathbf{x}_i} \{p(\mathbf{x}_i|\mathbf{x}_{-i})\} = \operatorname{argmax}_{\mathbf{x}_i} \left\{ \int_{\boldsymbol{\theta}} p(\mathbf{x}_i, \boldsymbol{\theta}|x_{-i}) d\boldsymbol{\theta} \right\}, \quad (4.38)$$

em que  $\boldsymbol{\theta}$  é um vetor que reúne as variáveis relacionadas ao ruído e ao sinal de áudio, a saber:  $\sigma_e^2$ ,  $\mathbf{a}$  e  $\sigma_v^2$ .

O algoritmo EM realiza a otimização acima através de operações sucessivas de expectância (cálculo de um valor esperado) e maximização, até que se atinja a convergência de acordo com algum critério adequado. A desvantagem desse método é exigir algum pré-processamento para estimação de  $\mathbf{i}$ , o que o torna sub-ótimo.

O amostrador de Gibbs permite obter amostras da distribuição conjunta  $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$  através da amostragem das distribuições condicionais (possivelmente multivariáveis) de cada variável envolvida, com todas as demais fixas em seus últimos valores amostrados.

Além de permitir o uso de modelos hierárquicos mais sofisticados, o amostrador de Gibbs tem a vantagem de permitir que se considere a variância do ruído variável no tempo, tornando-o adequado para remoção de *clicks* de amplitudes variadas.

O algoritmo apresentado em [4] e [27] consiste basicamente na amostragem das variáveis  $\mathbf{x}$ ,  $\mathbf{a}$ ,  $\sigma_e^2$ ,  $\sigma_{v_t}^2$ , com  $t = \{1, \dots, N\}$ , e  $\mathbf{i}$ . Opcionalmente, os hiperparâmetros das distribuições *a priori* podem ser tratados como variáveis e amostrados juntamente com as demais variáveis.

A parte mais crítica do algoritmo é a amostragem do vetor  $\mathbf{i}$ , que exigiria o cálculo de  $P(\mathbf{i}|\mathbf{y})$  para todos os  $2^N$  possíveis valores de  $\mathbf{i}$ . Para tornar o método prático, Godsill propõe que essa variável seja amostrada em sub-blocos de  $q$  amostras, o que reduz o número de operações para  $M2^q$ , em que  $M = N/q$  é o número de sub-blocos. O autor afirma que mesmo para  $q = 1$ , o método permite obter resultados satisfatórios. A consequência adversa desse procedimento é aumentar o número de iterações necessárias para a convergência.

## 4.5 Conclusões

Neste capítulo vimos algumas técnicas existentes para remoção de *clicks*, que variam em complexidade computacional e qualidade fornecida. À medida que se consideram modelos mais sofisticados para o ruído, a complexidade do procedimento de restauração tende a aumentar. No próximo capítulo, veremos alguns métodos alternativos para a remoção de *clicks* que procuram reduzir a complexidade computacional, mantendo a sofisticação das técnicas baseadas em MCMC.

## Capítulo 5

# Métodos alternativos para a Correção de *clicks*

A teoria bayesiana de detecção para localização de *clicks* exige que os parâmetros do ruído sejam conhecidos ou tenham que ser estimados através de algum pré-processamento. Técnicas de estimação mais sofisticadas como o algoritmo EM e o amostrador de Gibbs permitem que esses parâmetros sejam marginalizados, evitando assim que estimativas de baixa acuidade prejudiquem o resultado. A desvantagem dessas técnicas avançadas, em particular o amostrador de Gibbs, é a alta complexidade computacional exigida.

Numa tentativa de reduzir o custo computacional dessas técnicas, pensou-se em modelos diferentes para a perturbação (*clicks*), em que se desse uma descrição individualizada para cada *click*, reduzindo assim o número de variáveis a estimar.

A aplicação desse modelo ao problema de restauração de *clicks* gerou uma distribuição bastante complexa que só pôde ser tratada através de técnicas numéricas. As soluções propostas seguiram uma ordem crescente de generalidade: na primeira (Fibonacci), considerou-se a ocorrência de um único *click*; na segunda (Metropolis-Hastings), admitiu-se a ocorrência de um número  $K$  conhecido de *clicks*; por fim (Metropolis-Hastings com saltos reversíveis), permitiu-se estimar também o valor de  $K$ .

## 5.1 Modelo proposto

A Figura 5.1 mostra um sinal de áudio real contaminado por *clicks*. Este sinal passa por um restaurador (baseado em filtro inverso) e a diferença entre o sinal corrompido e o restaurado é calculada, obtendo-se uma representação dos *clicks* presentes no sinal (Figura 5.2). Observando um *click* individualmente (Figura 5.3), percebemos que seu comportamento segue um formato mais ou menos padrão: há uma descontinuidade abrupta e em seguida um comportamento oscilatório de amplitude decrescente. De acordo com [28], esse comportamento é causado pela resposta do aparelho de reprodução a um impulso. Essa resposta em geral é invariante no tempo e não-linear; se a amplitude do impulso for muito alta, a resposta pode apresentar um longo transitório de baixa frequência, caso em que, no presente trabalho, o distúrbio é classificado como pulso longo (ver Capítulo 6).

Essa observação nos levou a propor um modelo em que cada *click* é descrito por ruído branco gaussiano modulado por um pulso exponencial com a taxa de decaimento, a amplitude, localização e duração desconhecidas.

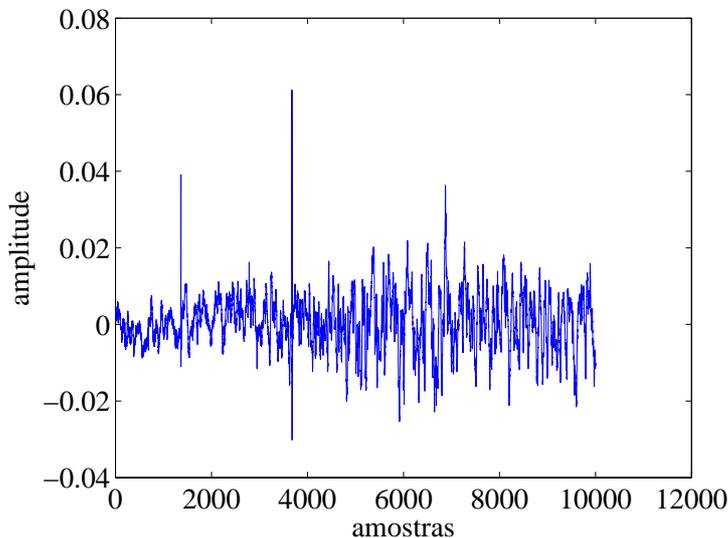


Figura 5.1: Sinal contaminado por *clicks*.

Supondo que um bloco de sinal contenha apenas um *click*, por esse modelo -

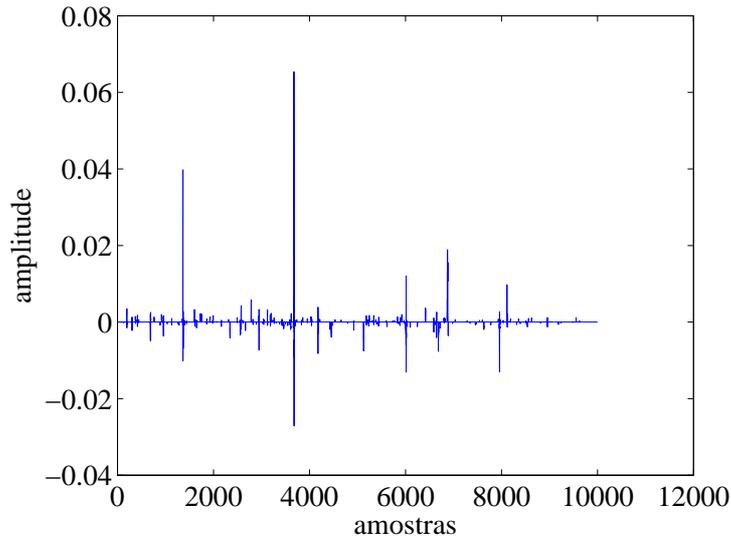


Figura 5.2: *Clicks* retirados do sinal contaminado.

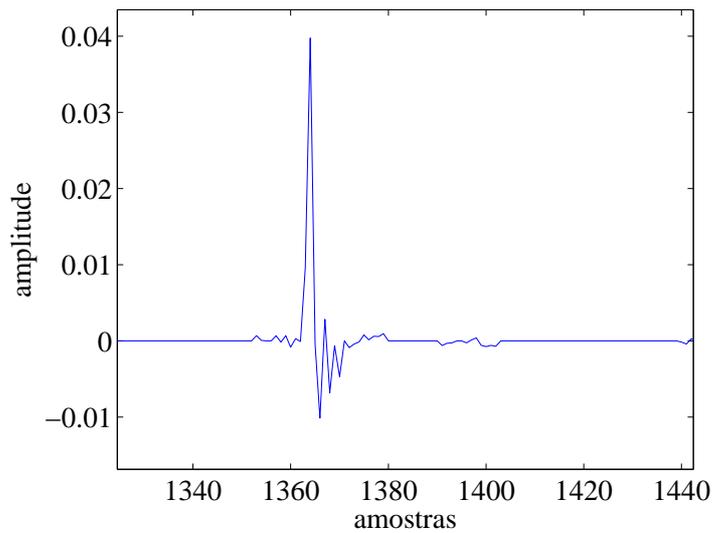


Figura 5.3: Detalhe de um distúrbio impulsivo.

que chamaremos de *modelo multiplicativo* - o ruído é matematicamente descrito por:

$$v(n) = r(n)V\lambda^{0,5(n-n_0)} [u(n - n_0) - u(n - n_0 - M)], \quad (5.1)$$

em que  $r(n)$  é ruído branco gaussiano de média zero e variância unitária; e  $V$ ,  $n_0$ ,  $M$  e  $\lambda$  representam a amplitude, localização, duração e taxa de decaimento do pulso exponencial. Para garantir que o pulso tenha comportamento decrescente,

devemos forçar  $-1 < \lambda < 1$ . Na prática, podemos impor sem perda de generalidade  $0 \leq \lambda < 1$ .

Embora esse modelo não leve em conta a correlação existente entre amostras consecutivas do *click*, tem a vantagem de ser simples de tratar matematicamente (como veremos nas próximas seções).

No caso mais geral de  $K$  *clicks*, o modelo do ruído se torna:

$$v(n) = r(n) \sum_{k=1}^K V_k \lambda_k^{0,5(n-n_{0_k})} [u(n - n_{0_k}) - u(n - n_{0_k} - M_k)]. \quad (5.2)$$

A razão para o fator 0,5 nas expressões acima é simplificar os cálculos subsequentes.

Uma forma conveniente de se caracterizar os *clicks* é através dos vetores

$$\mathbf{V} = [V_1 \quad \dots \quad V_K]^T, \quad (5.3)$$

$$\boldsymbol{\lambda} = [\lambda_1 \quad \dots \quad \lambda_K]^T, \quad (5.4)$$

$$\mathbf{n}_0 = [n_{0_1} \quad \dots \quad n_{0_K}]^T \text{ e} \quad (5.5)$$

$$\mathbf{M} = [M_1 \quad \dots \quad M_K]^T. \quad (5.6)$$

Deve-se notar que o modelo acima permite a superposição de *clicks*.

## 5.2 Restauração usando modelo multiplicativo para o ruído

As quatro variáveis do modelo ( $\boldsymbol{\lambda}$ ,  $\mathbf{V}$  e  $\mathbf{n}_0$  e  $\mathbf{M}$ ) são supostas desconhecidas, inicialmente. Para obtenção do sinal restaurado, duas abordagens podem ser seguidas:

1. Estimar as variáveis do ruído inicialmente (com sinal original marginalizado) e em seguida estimar o sinal original  $x(n)$  considerando conhecidas as variáveis do ruído;
2. Estimar as variáveis do ruído e o sinal original conjuntamente.

Antes de apresentarmos os detalhes das duas abordagens de restauração, vamos discutir a escolha das distribuições *a priori* do sinal original e das variáveis do ruído.

## 5.2.1 Distribuição *a priori*

Como mencionado na Seção 2.5, a distribuição *a priori* reflete o conhecimento subjetivo do projetista sobre os parâmetros que se deseja estimar. É importante ainda que se use distribuições bem conhecidas para as quais métodos eficientes de amostragem estão disponíveis. A escolha das prioris nesse trabalho seguiu as diretrizes de [4] e [20]. Veremos a seguir a atribuição de prioris para cada variável do ruído e para o sinal original.

### 5.2.1.1 Localização ( $n_0$ ) e duração ( $M$ )

A localização e a duração de um *click* podem ser consideradas discretas ou contínuas (o que faz sentido se pensarmos no sinal original analógico, antes da digitalização). Como ficará claro mais adiante, tratar essas variáveis como contínuas é vantajoso para a derivação dos algoritmos e ajuda a reduzir a complexidade computacional.

Se nenhum conhecimento *a priori* sobre a localização do *click* estiver disponível, é razoável assumir que todos os possíveis valores para  $n_0$  são equiprováveis. Sendo assim, a distribuição *a priori* para cada elemento  $n_{0_k}$  pode ser uma variável uniforme com valores entre 0 e  $N - 1$ , em que  $N$  é o número de amostras de cada bloco.

$$p(n_{0_k}) = \frac{1}{N - 1} [u(n_{0_k}) - u(n_{0_k} - (N - 1))], k = 1, \dots, K. \quad (5.7)$$

Se algum pré-processamento de baixa complexidade puder ser feito para se ter uma estimativa inicial da localização, esse conhecimento pode ser incluído na distribuição *a priori*. Nesse caso, uma gaussiana centrada no valor estimado pode ser uma boa escolha.

A duração deve ser sempre uma variável maior que 0, embora valores da ordem de algumas dezenas sejam típicos, considerando uma taxa de amostragem de

44, /!1 kHz. Uma priori conveniente para  $M$  é a distribuição gama inversa [14], que é definida apenas para valores positivos e cujo formato é determinado pelos parâmetros  $\alpha$  e  $\beta$ , podendo ser mais ou menos vaga conforme o conhecimento disponível.

$$p(M_k) = \text{IG}(M_k|\alpha_M, \beta_M) = \frac{\beta_M^{\alpha_M}}{\tau(\alpha_M)} M_k^{-(\alpha_M+1)} e^{(-\beta_M/M_k)}, k = 1, \dots, K. \quad (5.8)$$

### 5.2.1.2 Amplitude ( $V$ )

Veremos mais adiante que é mais simples trabalhar com  $V^2$  do que com  $V$ . Como  $V^2$  é sempre positiva, uma priori adequada é a distribuição gama inversa (definida acima), ou seja:

$$p(V_k^2) = \text{IG}(V_k^2|\alpha_V, \beta_V), k = 1, \dots, K. \quad (5.9)$$

Essa distribuição é conjugada em relação à gaussiana. Como veremos, essa propriedade será útil para a obtenção de amostras da distribuição *a posteriori* de  $V^2$ . Os parâmetros da distribuição podem ser estimados inicialmente ou tratados como variáveis aleatórias e incorporadas ao modelo bayesiano hierárquico. Neste trabalho, vamos tratar esses parâmetros como constantes conhecidas.

### 5.2.1.3 Taxa de decaimento ( $\lambda$ )

A variável  $\lambda$  representa a taxa de decaimento do pulso exponencial, tendo, portanto, que pertencer ao intervalo  $[0, 1]$ . Uma distribuição uniforme entre esses extremos é uma escolha adequada para a distribuição *a priori* de  $\lambda$ :

$$p(\lambda) = \text{U}(\lambda|0, 1). \quad (5.10)$$

### 5.2.1.4 Sinal original ( $\mathbf{x}$ )

Como vimos no capítulo anterior, um bloco de curta duração sinal de áudio é usualmente representado por um modelo AR. Os parâmetros do modelo podem ser estimados inicialmente (a partir do próprio sinal corrompido) e usados como constantes nos procedimentos subsequentes, ou tratados como variáveis aleatórias e incorporadas ao modelo bayesiano hierárquico. Vamos considerar inicialmente que

os parâmetros são conhecidos. A distribuição *a priori* do sinal original é, portanto (Equação 4.4),

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{N/2} |\mathbf{R}_x|^{1/2}} e^{-\frac{1}{2} \mathbf{x}^T \mathbf{R}_x^{-1} \mathbf{x}}, \quad (5.11)$$

em que  $\mathbf{R}_x = E[\mathbf{x}\mathbf{x}^T] = \frac{1}{\sigma_e^2} \mathbf{A}^T \mathbf{A}$ .

A seguir veremos o cálculo da distribuição *a posteriori* para as variáveis do ruído, considerando o Procedimento 1: realizar inicialmente a estimação dos parâmetros do ruído, com o sinal original marginalizado.

## 5.2.2 Obtenção da distribuição *a posteriori*

Inicialmente, os parâmetros do modelo AR,  $\mathbf{a}$  e  $\sigma_E^2$ , serão tratados como constantes conhecidas. Definimos o vetor  $\boldsymbol{\theta}$  contendo os parâmetros do modelo do ruído:

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{n}_0 \\ \mathbf{M} \\ \mathbf{V} \\ \lambda \end{bmatrix}. \quad (5.12)$$

A verossimilhança pode ser obtida através da Equação (4.32):

$$p(\mathbf{y}|\boldsymbol{\theta}) = p(\mathbf{y}|\mathbf{i}) = \frac{\sigma_e^l e^{-\frac{1}{2\sigma_e^2} E_{\text{MIN}}}}{(2\pi\sigma_e^2)^{\frac{N-P}{2}} |\mathbf{R}_v|^{1/2} |\boldsymbol{\Phi}|^{1/2}}, \quad (5.13)$$

em que o vetor  $\mathbf{i}$  é obtido a partir de  $\boldsymbol{\theta}$  a partir das expressões abaixo:

$$\mathbf{i}_k = \begin{bmatrix} \mathbf{0}_{1 \times (n_{0_k} - 1)} & \mathbf{1}_{1 \times M_k} & \mathbf{0}_{1 \times (N - n_{0_k} - M_k)} \end{bmatrix}^T, \quad (5.14)$$

$$i(n) = \max\{\mathbf{i}_{1_n}, \mathbf{i}_{2_n}, \dots, \mathbf{i}_{K_n}\}, n = 1, 2, \dots, N, \quad (5.15)$$

sendo  $\mathbf{0}_{1 \times n}$  e  $\mathbf{1}_{1 \times n}$  vetores-linha contendo  $n$  elementos iguais a 0 e 1, respectivamente.

Considerando o modelo da Equação (5.2), a matriz de covariância do ruído é determinada por:

$$R_{v_{ij}} = \begin{cases} \sum_{k=1}^K V_k^2 \lambda_k^{i-n_{0_k}} [u(i-n_{0_k}) - u(i-n_{0_k} + M_k - 1)], & i = j \\ 0, & i \neq j, \end{cases} \quad (5.16)$$

para  $i = 1, 2, \dots, N$

Pela regra de Bayes, obtemos a probabilidade *a posteriori*:

$$p(\boldsymbol{\theta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}), \quad (5.17)$$

em que  $p(\boldsymbol{\theta})$  é a probabilidade *a priori* do vetor de parâmetros  $\boldsymbol{\theta}$ , cujos componentes são supostos independentes *a priori*, isto é:

$$p(\boldsymbol{\theta}) = p(\mathbf{n}_0)p(\mathbf{M})p(\mathbf{V}^2)p(\boldsymbol{\lambda}). \quad (5.18)$$

Substituindo as Equações (5.7), (5.8), (5.9), (5.10) e em (5.18), obtemos a distribuição *a priori* de  $\boldsymbol{\theta}$ . Os resultados das Equações (5.18) e (5.13) aplicados em (5.17) permitem obter a distribuição *a posteriori*:

$$p(\boldsymbol{\theta}|\mathbf{y}) \propto \frac{\sigma_e^l e^{-\frac{1}{2\sigma_e^2}E_{\text{MIN}}}}{(2\pi\sigma_e^2)^{\frac{N-P}{2}}|\mathbf{R}_v|^{1/2}|\boldsymbol{\Phi}|^{1/2}} \times \prod_{k=0}^K U(\lambda_k|0, 1)\text{IG}(V_k^2|\alpha_V, \beta_V)\text{IG}(M_k^2|\alpha_M, \beta_M) \frac{1}{N-1} [u(\mathbf{n}_{0_k}) - u(\mathbf{n}_{0_k} - (N-1))]. \quad (5.19)$$

Até aqui foi descrita a obtenção da distribuição *a posteriori* associada ao modelo proposto. Nas próximas seções, serão apresentadas algumas técnicas numéricas que permitem estimar os parâmetros de interesse a partir daquela distribuição. Pelo critério MAP, os parâmetros estimados são aqueles que maximizam a distribuição *a posteriori*; o critério BMSE estabelece que os parâmetros estimados são a média da distribuição *a posteriori*. A obtenção dessas estimativas se mostrou difícil de ser feita de forma analítica. Na próxima seção, será proposto um método de se obter a estimativa MAP dos parâmetros do ruído para o caso particular de apenas um *click* presente no sinal. Nas seções subseqüentes, algoritmos baseados em MCMC serão propostos para o tratamento casos mais geral. Nesse último caso, o critério de otimização será o BMSE.

### 5.3 Restauração pelo método de Fibonacci

Métodos numéricos de otimização multivariável (como o método do gradiente, o método de Newton, etc.) requerem o cálculo das derivadas em relação a cada uma

das variáveis, o que é impossível de ser feito analiticamente para as variáveis  $n_0$  e  $M$ , e difícil para as demais.

Optamos pelo uso de um método de otimização univariável (o método de Fibonacci [29]), empregado de forma iterativa, partindo de um intervalo inicial para os parâmetros de interesse, e maximizando cada variável por vez, considerando as demais fixas. Mais especificamente, parte-se de uma estimativa inicial para duas variáveis, digamos,  $n_0$  e  $\lambda$ . Em seguida, obtém-se, pelo algoritmo de Fibonacci, uma estimativa para  $V^2$ . Então, o valor de  $V^2$  obtido e o valor de  $\lambda$  pré-estabelecido podem ser usados para se obter uma nova estimativa de  $n_0$ . Esse procedimento continua até que se alcance a convergência (de acordo com algum critério). Os valores iniciais para os parâmetros são obtidos a partir da amostragem das respectivas distribuições *a priori*. Nessa abordagem preliminar, a duração  $M$  do *click* foi tratada como constante. Os detalhes estão no algoritmo a seguir:

1. Inicialização:

(a)  $n_0^{(0)} \sim p(n_0)$ ,

(b)  $(V^2)^{(0)} \sim p(V^2)$ ,

(c)  $\lambda^{(0)} \sim p(\lambda)$ ,

(d)  $j = 0$ .

2. Repita

(a)  $n_0^{(j)} = \text{fib} \left( \pi \left( n_0, (V^2)^{(j-1)}, \lambda^{(j-1)} | \mathbf{y} \right), I_{n_0}, N \right)$ ,

(b)  $(V^2)^{(j)} = \text{fib} \left( \pi \left( (V^2), n_0^{(j-1)}, (V^2)^{(j-1)}, \lambda^{(j-1)} | \mathbf{y} \right), I_{V^2}, N \right)$ ,

(c)  $\lambda^{(j)} = \text{fib} \left( \pi \left( \lambda, (V^2)^{(j-1)}, n_0^{(j-1)} | \mathbf{y} \right), I_\lambda, N \right)$ ,

(d)  $j = j + 1$ ,

3. até que  $\max\{n_0^{(j)}/n_0^{(j-1)}, \lambda^{(j)}/\lambda^{(j-1)}, (V^2)^{(j)}/(V^2)^{(j-1)}\} < \delta$ .

A função  $\text{fib}(f(x, \cdot), I_x, N)$  retorna uma aproximação para o máximo de  $f(x)$  através do método de Fibonacci, sendo dados o intervalo inicial,  $I_x$ , e o número de iterações,  $N$ , do algoritmo. Esse intervalo pode ser obtido inicialmente através

de algum pré-processamento. Os valores de  $N$  e  $\delta$  estão relacionados à acuidade do resultado e podem ser escolhidos pelo usuário conforme o erro tolerado e as restrições de complexidade.

Para o método funcionar corretamente, é preciso que a função a ser otimizada (no caso, a distribuição *a posteriori*) seja unimodal. Embora não tenhamos desenvolvido uma prova formal, verificamos essa propriedade empiricamente. Além disso, é preciso garantir que dentro da faixa inicial de valores para  $n_0$  esteja presente apenas um *click*. Essa limitação será superada pelos algoritmos apresentados nas seções subseqüentes.

### 5.3.1 Estimação de $n_0$

O valor de  $n_0$  determina quais amostras do sinal estão corrompidas, e portanto permite obter o vetor  $\mathbf{y}_i$ , que aparece no argumento da exponencial das Equações (4.36) e (4.37). Como consequência, essa equação não tem expressão analítica em função de  $n_0$ , o que implica o uso de métodos numéricos para sua otimização. A Figura 5.4 mostra, para um caso particular, que a distribuição *a posteriori* de  $n_0$  é unimodal. Portanto, o método de Fibonacci pode ser aplicado.

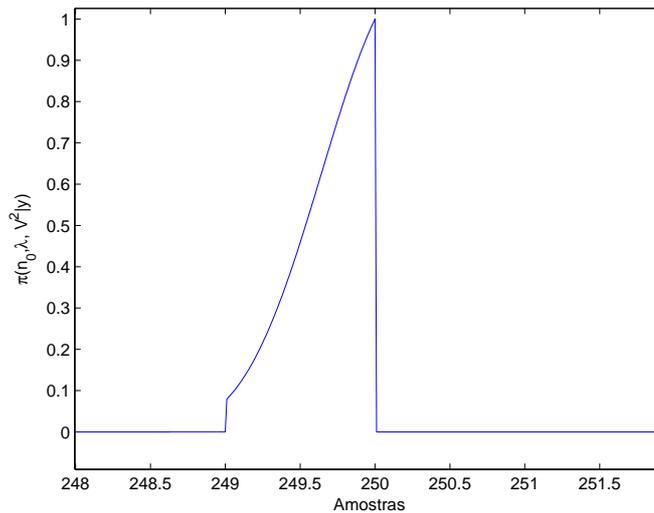


Figura 5.4: Distribuição *a posteriori* de  $n_0$  com  $V$  e  $\lambda$  fixos.

O intervalo inicial pode ser estabelecido através de algum pré-processamento

simples (filtro inverso, por exemplo) que estabeleça uma região menor para os valores possíveis de  $n_0$ . Caso não se disponha de uma estimativa inicial, todo o intervalo correspondente a um bloco deve ser considerado, desde que se garanta que haja apenas um *click* nesse bloco.

A distribuição  $p(n_0|\mathbf{y})$  pode ser obtida, a menos de uma constante, por inspeção da função  $p(\boldsymbol{\theta}|\mathbf{y})$  (Equação (5.17)), considerando-se apenas os termos dependentes de  $n_0$ . O resultado é:

$$p(n_0|\mathbf{y}) = \frac{e^{(-E_{\text{MIN}})}}{|\mathbf{R}_v|^{1/2}|\boldsymbol{\Phi}|^{1/2}}. \quad (5.20)$$

### 5.3.2 Estimação de $V$

Usando este mesmo recurso, a distribuição *a posteriori* de  $V^2$  pode ser facilmente obtida:

$$p(V^2|\mathbf{y}) = \frac{e^{(-E_{\text{MIN}})}}{|\mathbf{R}_v|^{1/2}|\boldsymbol{\Phi}|^{1/2}} \text{IG}(V^2|\alpha_V, \beta_V). \quad (5.21)$$

A Figura 5.5 mostra o gráfico da distribuição *a posteriori* de  $V^2$ . Vemos que a função é unimodal, e portanto o método de Fibonacci pode ser aplicado. O intervalo inicial para o método de Fibonacci precisa incluir o máximo da função. Observando alguns sinais típicos, podemos escolher um intervalo adequado.

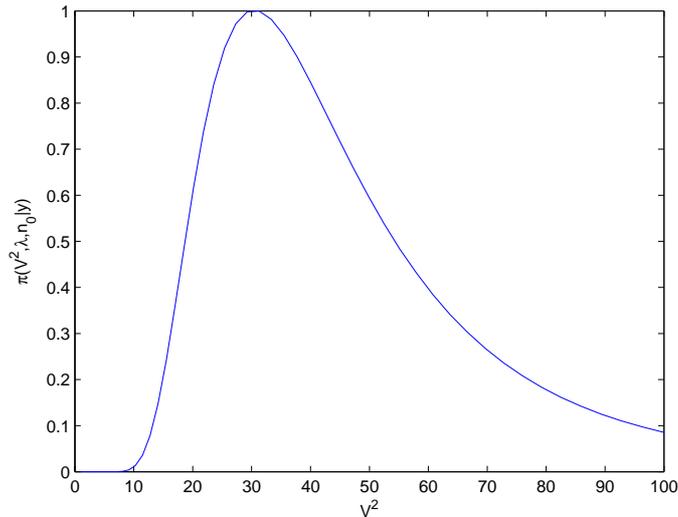


Figura 5.5: Distribuição *a posteriori* de  $V$  com  $n_0$  e  $\lambda$  fixos.

### 5.3.3 Estimação de $\lambda$

A distribuição *a posteriori* de  $\lambda$  pode ser obtida a partir da Equação (5.17) escrevendo-se somente os termos que dependem de  $\lambda$ :

$$p(V^2|\mathbf{y}) = \frac{e^{(-E_{\text{MIN}})}}{|\mathbf{R}_v|^{1/2}|\Phi|^{1/2}}U(\lambda|0, 1). \quad (5.22)$$

A distribuição *a posteriori* de  $\lambda$  é mostrada na Figura 5.6, a partir da qual podemos verificar a convexidade da função.

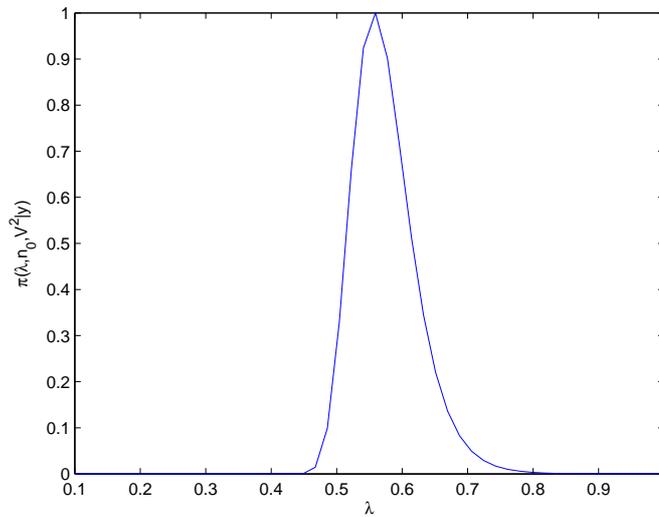


Figura 5.6: Distribuição *a posteriori* de  $\lambda$  com  $V$  e  $n_0$  fixos.

### 5.3.4 Resultados

Esta seção se dedica à apresentação dos resultados obtidos com o modelo proposto, e à sua comparação com os obtidos pelo método bayesiano seqüencial (Seção 4.4.2). Foram feitos testes com sinais de áudio reais, em duas situações distintas:

- (A) *click* artificial, gerado a partir do modelo multiplicativo (equação (5.1)), e
- (B) *click* real, retirado de um outro sinal corrompido, e inserido num sinal não corrompido.

Em ambos os casos, analisou-se um bloco de sinal com apenas um *click* presente. O sinal escolhido, nas duas situações, foi uma nota de violão com os *clicks* facilmente audíveis. Foi selecionado um bloco de 500 amostras de um sinal de áudio amostrado a 44,1 kHz, em que cada amostra é representada por 16 bits. O *click* está localizado próximo à amostra 250, como mostram as Figuras 5.7 e 5.9.

Para o teste com o método seqüencial, foi usado um modelo AR de ordem 10, cujos coeficientes foram determinados a partir do sinal corrompido pela minimização do erro médio quadrático. A variância do ruído para o método seqüencial no caso de *clicks* artificiais foi calculada diretamente a partir do ruído que foi inserido no sinal.

Usou-se o mesmo modelo AR para o sinal na implementação do método proposto. Nessas simulações, usamos distribuições *a priori* do tipo gaussiana para  $V^2$  (com média zero e variância  $\sigma_V^2 = 1$ ) e gama inversa para  $\lambda$  (com parâmetros  $\alpha_\lambda = 5$  e  $\beta_\lambda = 0,8$ ).

#### 5.3.4.1 Situação A: *click* artificial

O *click* foi gerado artificialmente, através do modelo da Equação (5.1), com os parâmetros:  $n_0 = 250$ ,  $\lambda = 0,2$  e  $V = 0,3$ .

As estimativas obtidas para esses parâmetros foram:  $\hat{n}_0 = 250$ ,  $\hat{\lambda} = 0,1493$  e  $\hat{V} = 0,2975$ . Os sinais original e estimados pelos métodos seqüencial e proposto estão mostrados na Figura 5.8. Vemos que o método proposto fornece uma estimativa mais suave e mais parecida com o sinal original do que o método seqüencial. Além disso, numa implementação em Matlab, observamos que nosso algoritmo é significativamente mais rápido.

#### 5.3.4.2 Situação B: *click* real

A Figura 5.10 mostra o sinal original e os sinais estimados pelo método seqüencial e pelo método baseado no modelo proposto. Vemos que a curva obtida pelo método proposto é mais suave, e representa melhor o sinal original. Ouvindo os sinais estimados, percebemos um pequeno *click* no sinal restaurado pelo método seqüencial, mas nenhuma degradação no sinal obtido pelo método proposto.

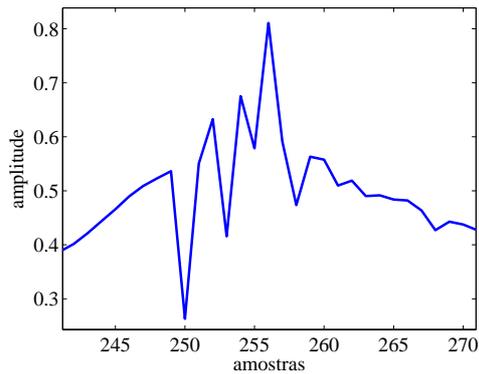


Figura 5.7: Sinal corrompido com *click* artificial.

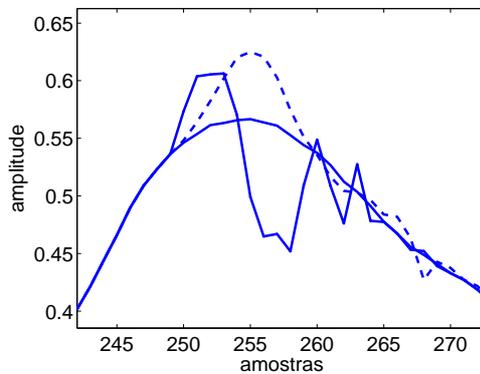


Figura 5.8: Sinal original (linha contínua), sinal estimado pelo método seqüencial (linha sólido-pontilhada) e sinal estimado pelo método de Fibonacci (linha tracejada).

Esses testes preliminares indicaram que o modelo proposto permite obter bons resultados, quando comparado aos da literatura. Entretanto, para torná-lo operacional em casos práticos, é preciso tratar o caso geral, em que o número de *clicks* num bloco é desconhecido.

## 5.4 Restauração através de algoritmo MH e amostrador de Gibbs

O método proposto na seção anterior é limitado pelo fato de assumir a presença de apenas um *click* em cada bloco do sinal. Nesta seção, apresentaremos um

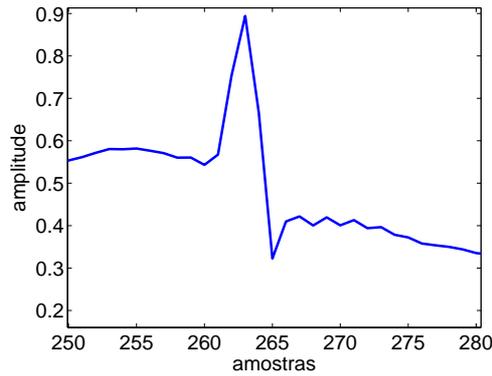


Figura 5.9: Sinal corrompido com *click* real.

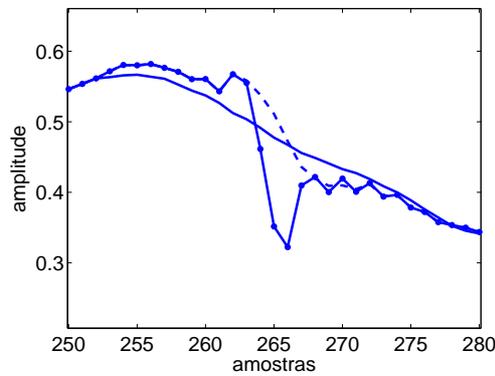


Figura 5.10: Sinal original (linha contínua), sinal estimado pelo método seqüencial (linha sólido-pontilhada) e sinal estimado pelo método de Fibonacci (linha tracejada).

método mais sofisticado, baseado no algoritmo de Metropolis-Hastings e no amostrador de Gibbs, que permite realizar a restauração para o caso de um número  $K$  qualquer, mas conhecido, de *clicks* por bloco. A determinação do número de *clicks* deve ser feita através de algum pré-processamento. Neste trabalho, usamos um algoritmo baseado em filtragem inversa, desenvolvido em [30] e publicado em [31].

O algoritmo desenvolvido possui uma estrutura similar ao amostrador de Gibbs, em que as distribuições condicionais são amostradas iterativamente, e o algoritmo de Metropolis-Hastings é usado quando essa amostragem não pode ser feita de forma simples. A estrutura geral do algoritmo é descrita a seguir.

1. Inicialização: gerar os parâmetros  $K$  e  $\theta^{(0)}$ .

2. Para  $j$  de 1 até  $N_{it}$

- (a) Para  $k$  de 1 até  $K$ , amostrar  $n_{0_k}^{(j+1)}$  e  $M_k^{(j+1)}$  de  $\pi(n_{0_k}, M_k | \mathbf{y}, \boldsymbol{\theta}_{-[n_{0_k}, M_k]}^{(j)})$ ;
- (b) Amostrar  $\mathbf{x}^{(j+1)}$  de  $\pi(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta}^{(j)})$ ;
- (c) Para  $k$  de 1 até  $K$ , amostrar  $V_k^2$  de  $\pi(V_k^2 | \mathbf{x}^{(j+1)}, \mathbf{y}, \boldsymbol{\theta}_{-V_k^2}^{(j)})$ ;
- (d) Para  $k$  de 1 até  $K$ , amostrar  $\lambda_k$  de  $\pi(\lambda_k | \mathbf{x}^{(j+1)}, \mathbf{y}, \boldsymbol{\theta}_{-\lambda_k}^{(j)})$ ;

Após cada amostragem, os parâmetros amostrados são inseridos no vetor  $\boldsymbol{\theta}$  nas posições correspondentes.

A seguir detalhamos as etapas de inicialização e os cálculos das distribuições condicionais e as escolhas das distribuições propostas (quando for o caso) para cada variável.

### 5.4.1 Inicialização

A inicialização aleatória dos parâmetros tornaria a convergência do algoritmo inaceitavelmente lenta. Por essa razão é conveniente usar algum procedimento de baixa complexidade para se obter uma estimativa inicial dos parâmetros, especialmente da localização  $n_0$ , cuja convergência é mais crítica. O papel do algoritmo MH é fazer um refinamento dessa estimativa inicial.

Para inicialização dos componentes de  $\mathbf{n}_0$  usamos como ponto de partida o sinal de erro de predição ( $e(n)$ ) gerado pelo processo de detecção desenvolvido em [30], obtido usando o modelo AR. Espera-se que *clicks* estejam presentes nas regiões onde esse erro é alto. O procedimento de inicialização consiste em dividir o bloco de sinal (que está sendo processado) em sub-blocos de pequeno comprimento (usamos 10 amostras), calcular a energia do erro dentro de cada sub-bloco e determinar os sub-blocos para os quais a energia do erro exceda um certo limiar (escolhido inicialmente). O valor de  $K$  é o número de sub-blocos selecionados e os valores de  $n_{0_k}$ ,  $k = 1, 2, \dots, K$ , são escolhidos aleatoriamente dentro dos sub-blocos selecionados.

A inicialização dos parâmetros  $\mathbf{M}$ ,  $\mathbf{V}^2$  e  $\boldsymbol{\lambda}$  não é crítica e pode ser feita de forma simples por amostragem de suas respectivas distribuições *a priori*. O sinal  $\mathbf{x}$  não precisa ser inicializado, pois não é necessário para a amostragem de  $\mathbf{n}_0$ .

## 5.4.2 Distribuições condicionais e escolha das propostas

Os parâmetros do modelo AR são supostos conhecidos e constantes ao longo de cada bloco do sinal (na próxima seção, tratá-los-emos como variáveis aleatórias). As distribuições condicionais para cada variável desconhecida  $\mathbf{x}$  e  $\boldsymbol{\theta}$  podem ser obtidas a partir da distribuição conjunta  $p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})$ , uma vez que

$$p(\theta_j|\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}_{-j}) = \frac{p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y})}{p(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}_{-j})} \quad (5.23)$$

é independente de  $\boldsymbol{\theta}_j$ .

A distribuição conjunta pode ser obtida através da aplicação sucessiva do teorema de Bayes:

$$p(\mathbf{x}, \boldsymbol{\theta}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \quad (5.24)$$

$$= p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}) \quad (5.25)$$

$$= p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})p(\mathbf{n}_0)p(\mathbf{M})p(\mathbf{V})p(\boldsymbol{\lambda}). \quad (5.26)$$

A verossimilhança  $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$  tem a mesma forma da distribuição do ruído, com média deslocada de  $\mathbf{x}$ . Nas amostras em que o ruído está ausente (isto é,  $i(n) = 0$ ), o sinal observado é igual ao sinal original. Portanto:

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \prod_{\{i(n)=0\}} \delta(y(n) - x(n))N(\mathbf{y}_i|\mathbf{x}_i, \mathbf{R}_{v_i}), \quad (5.27)$$

em que  $\delta(x)$  é a função impulso unitário e

$$\mathbf{R}_{v_{ij}} = \begin{cases} \sum_{k=1}^K V_k \lambda_k^{i-n_{0k}} [u(i - n_{0k}) - u(i - n_{0k} + M_k - 1)], & i = j \\ 0 & i \neq j \end{cases}. \quad (5.28)$$

A seguir apresentamos os cálculos das distribuições condicionais para cada variável desconhecida e os métodos adotados para obter suas amostras.

### 5.4.2.1 Amostragem do sinal original $\mathbf{x}$

A distribuição condicional completa de  $\mathbf{x}$  pode ser obtida a partir da distribuição conjunta, retirando-se os termos que não dependem de  $\mathbf{x}$  e analisando-se a expressão resultante.

$$p(\mathbf{x}|\boldsymbol{\theta}, \mathbf{y}) = \prod_{i(n)=0} \delta(y(n) - x(n))N(\mathbf{y}_i|\mathbf{x}_i, \mathbf{R}_{v_i})N(\mathbf{x}|\mathbf{0}, \sigma_e^2(\mathbf{A}^T \mathbf{A})^{-1}). \quad (5.29)$$

Pela expressão acima, reconhecemos que:

$$p(\mathbf{x}_{-i}|\boldsymbol{\theta}, \mathbf{y}) = \delta_{N-l}(\mathbf{y}_{-i} - \mathbf{x}_{-i}) \quad (5.30)$$

e

$$p(\mathbf{x}_i|\boldsymbol{\theta}, \mathbf{y}) = N(\mathbf{y}_i|\mathbf{x}_i, \mathbf{R}_{v_i})N(\mathbf{x}|\mathbf{0}, \sigma_e^2(\mathbf{A}^T\mathbf{A})^{-1}) \quad (5.31)$$

$$= N(\mathbf{x}_i|\mathbf{x}_i^{\text{MAP}}, \sigma_e^2\boldsymbol{\Phi}^{-1}), \quad (5.32)$$

em que

$$\mathbf{x}_i^{\text{MAP}} = -\boldsymbol{\Phi}(\mathbf{A}_i^T\mathbf{A}_{-i}\mathbf{y}_{-i} - \sigma_e^2\mathbf{R}_{v_i}\mathbf{y}_i), \quad (5.33)$$

$$\boldsymbol{\Phi} = \mathbf{A}_i^T\mathbf{A}_i + \sigma_e^2\mathbf{R}_{v_i}^{-1} \text{ e} \quad (5.34)$$

$\delta_m$  é a função impulso unitário  $m$ -dimensional.

Essas expressões foram obtidas a partir do fato de que o produto de duas gaussianas  $p_1(\mathbf{x}) = N(\mathbf{x}|\boldsymbol{\mu}_1, \mathbf{C}_1)$  e  $p_2(\mathbf{x}) = N(\mathbf{x}|\boldsymbol{\mu}_2, \mathbf{C}_2)$  é também uma gaussiana de média

$$\boldsymbol{\mu}_p^T = (\boldsymbol{\mu}_1^T\mathbf{C}_1^{-1} + \boldsymbol{\mu}_2^T\mathbf{C}_2^{-1}) (\mathbf{C}_1^{-1} + \mathbf{C}_2^{-1})^{-1} \quad (5.35)$$

e matriz de covariância

$$\mathbf{C}_p = (\mathbf{C}_1^{-1} + \mathbf{C}_2^{-1})^{-1}. \quad (5.36)$$

#### 5.4.2.2 Amostragem da localização $\mathbf{n}_0$ e da duração $\mathbf{M}$

Pelo método da composição [4], podem-se obter amostras da distribuição conjunta de  $\mathbf{x}_i$ ,  $\mathbf{n}_0$  e  $\mathbf{M}$ , amostrando-se seqüencialmente as distribuições

$$p(\mathbf{x}_i|\boldsymbol{\theta}, \mathbf{y}) \quad (5.37)$$

e

$$p(\mathbf{n}_0, \mathbf{M}|\boldsymbol{\theta}_{-[n_0, M]}, \mathbf{y}). \quad (5.38)$$

Para a obtenção da distribuição acima, inicialmente integra-se  $p(\mathbf{x}, \mathbf{y}|\boldsymbol{\theta})$  em relação a  $\mathbf{x}$  e em seguida calcula-se a posteriori, através da regra de Bayes:

$$p(\mathbf{n}_0, \mathbf{M}|\boldsymbol{\theta}_{-[n_0, M]}, \mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\mathbf{n}_0)p(\mathbf{M}), \quad (5.39)$$

em que a expressão para  $p(\mathbf{y}|\boldsymbol{\theta})$  está descrita na Seção 5.3.1. Supôs-se que  $\mathbf{n}_0$  e  $\mathbf{M}$  fossem independentes *a priori*, donde pudemos fazer  $p(\mathbf{n}_0, \mathbf{M}) = p(\mathbf{n}_0)p(\mathbf{M})$ .

A geração de amostras da distribuição  $p(\mathbf{n}_0, \mathbf{M} | \boldsymbol{\theta}_{-[n_0, M]}, \mathbf{y})$  pode ser feita pelo algoritmo de Metropolis-Hastings, aplicado seqüencialmente à localização e duração de cada *click*. Mais precisamente, para  $k$  de 1 até  $K$ , propõem-se as variáveis  $n_{0_k}^*$  e  $M_k^*$  a partir de uma distribuição proposta adequada (por exemplo, uma gaussiana bi-dimensional de média  $(n_{0_k}^{(j)}, M_k^{(j)})$  e matriz de covariância diagonal) e se as aceitam com probabilidade  $\alpha$  obtida a partir da Equação (3.20). Para redução da complexidade computacional, podemos artificialmente ignorar a presença dos outros *clicks*, o que equivale a fazer suas amplitudes iguais a zero e substituir as amostras de  $\mathbf{y}$  nas posições corrompidas pelas respectivas amostras do sinal  $\mathbf{x}$  amostrado na iteração anterior. Isso faz com que as operações envolvam matrizes de ordem igual à duração do  $k$ -ésimo *click*, em vez da soma das durações de todos os *clicks*.

#### 5.4.2.3 Amostragem da amplitude $V$

Inicialmente, calculamos a distribuição condicional total para cada componente de  $\mathbf{V}$ :

$$P(V_k | \boldsymbol{\theta}_{-V_k}, \mathbf{x}, \mathbf{y}) \propto N(\mathbf{y} | \mathbf{x}_j, \mathbf{R}_v) \text{IG}(V_k | \alpha_V, \beta_V), \quad (5.40)$$

$$P(V_k | \boldsymbol{\theta}_{-V_k}, \mathbf{x}, \mathbf{y}) \propto \frac{1}{|\mathbf{R}_{v_k}|} e^{(-\frac{1}{2} \mathbf{v}_k^T \mathbf{R}_{v_k}^{-1} \mathbf{v}_k)} V_k^{-\alpha+1} e^{(-\frac{\beta}{V_k})}, \quad (5.41)$$

em que

$$\mathbf{v}_k = \left[ y(\lceil n_{0_k} \rceil) - x(\lceil n_{0_k} \rceil) \quad \dots \quad y(\lfloor n_{0_k} + M_k \rfloor - 1) - x(\lfloor n_{0_k} + M_k \rfloor - 1) \right]^T \quad (5.42)$$

e

$$\mathbf{R}_{v_{ij}} = \begin{cases} \sum_{k=1}^K (V_k)^2 \lambda_k^{i-n_{0_k}} [u(i-n_{0_k}) - u(i-n_{0_k} + M_k - 1)], & i = j \\ 0 & i \neq j \end{cases}. \quad (5.43)$$

para  $i = \lceil n_{0_k} \rceil, \lceil n_{0_k} \rceil + 1, \dots, \lfloor n_{0_k} + M_k \rfloor - 1$ .

(Nas expressões acima, a notação  $\lceil x \rceil$  indica o menor inteiro maior que  $x$  ou igual; e a notação  $\lfloor x \rfloor$  indica o maior inteiro menor que  $x$ .)

Vemos que essa distribuição pode depender de outras amplitudes ( $V_i$ ) nos casos em que existe superposição de *clicks*. Nos casos em que não existe superposição no  $k$ -ésimo *click*, a matriz de covariância do ruído torna-se mais simples:

$$\mathbf{R}_{v_k} = V_k^2 \lambda_k^{\lceil n_{0_k} \rceil - n_{0_k}} \text{diag} \left\{ 1, \lambda_k^{-1}, \dots, \lambda_k^{-M_k' + 1} \right\}, \quad (5.44)$$

em que  $M' = \lfloor n_{0_k} + M_k \rfloor - \lfloor n_{0_k} \rfloor$ .

Nesse caso, reconhecemos que a distribuição de  $V_k$  é uma gama inversa, com parâmetros:

$$\alpha'_v = \alpha_v + \frac{M'}{2} \quad (5.45)$$

e

$$\beta'_v = \beta_v + \sum_{j=0}^{M'-1} v_j (\lfloor n_{0_j} \rfloor + i)^2 \lambda_k^{-\lfloor n_{0_j} \rfloor - n_{0_j} + i}, \quad (5.46)$$

a partir da qual podem-se obter amostras de forma simples.

No caso em que há superposições, a distribuição resultante é mais complicada, e o algoritmo de Metropolis-Hastings é usado para sua amostragem. Para isso, usamos uma proposta obtida a partir da Equação (5.41), retirando-se os termos correspondentes a *clicks* superpostos. O cálculo da probabilidade de aceitação é simplificado pelo fato de a proposta e a posteriori terem estruturas similares.

#### 5.4.2.4 Amostragem da taxa de decaimento $\lambda$

Seguindo o mesmo procedimento da seção anterior, calculamos a distribuição condicional total para cada componente de  $\lambda$ :

$$p(\lambda_k | \mathbf{x}, \boldsymbol{\theta}_{-\lambda_k}, \mathbf{y}) \propto p(\mathbf{x}, \boldsymbol{\theta} | \mathbf{y}). \quad (5.47)$$

Isolando os termos dependentes de  $\lambda_i$  na distribuição conjunta de  $\mathbf{x}$  e  $\boldsymbol{\theta}$ , obtemos, após simplificações, para a situação em que o  $k$ -ésimo *click* não está superposto a algum outro:

$$\pi(\lambda_k) = p(\lambda_k | \mathbf{x}, \boldsymbol{\theta}_{-\lambda_k}, \mathbf{y}) \propto \frac{1}{\lambda^{M'(M'-1)/2}} e^{-\frac{v(\lfloor n_{0_k} \rfloor)}{2\lambda_k} - \frac{v(\lfloor n_{0_k} \rfloor + 1)}{2\lambda_k} - \dots - \frac{v(\lfloor n_{0_k} \rfloor + M' - 1)}{2\lambda_k}} p(\lambda_k). \quad (5.48)$$

Essa distribuição não é de uma forma bem conhecida. Por isso, usaremos o algoritmo de Metropolis-Hastings para obtenção de suas amostras de forma indireta. Como proposta, usaremos uma gaussiana centrada no valor atual da cadeia,  $\lambda_k^{(j)}$ , e variância  $\sigma_q^2 = 0,01$ . Esse valor foi escolhido por tentativa e erro, observando a correlação obtida nas amostras obtidas. Para reduzir a complexidade do cálculo

da probabilidade de aceitação, usaremos a expressão simplificada para a razão das distribuições condicionais totais:

$$\begin{aligned} \text{Razão das condicionais} &= \frac{\pi(\lambda_k^*)}{\pi(\lambda_k^{(j)})} = \\ &= \left( \frac{\lambda_k^{(j)}}{\lambda_k^*} \right)^{M'(M'-1)/2} e^{-\frac{1}{2}(\sum_{i=0}^{M'-1} v(\lceil n_{0_k} \rceil + i) ((\lambda_k^*)^{-i} - (\lambda_k^{(j)})^{-i}))} \end{aligned} \quad (5.49)$$

No caso em que o  $k$ -ésimo *click* está superposto a algum outro, a simplificação acima não pode ser feita e a complexidade computacional do cálculo de  $\alpha$  será maior. Não sabemos quão freqüente essa superposição de *clicks* ocorre na prática. De todo modo, a complexidade da operação de amostragem dessa variável é bem menor que outras etapas do algoritmo, e pode ser negligenciada no cálculo da complexidade total.

### 5.4.3 Resultados

Nesta seção apresentamos os resultados obtidos com o algoritmo Metropolis-Hastings, considerando um número de *clicks* conhecido. Usamos para teste um sinal sintético, gerado a partir de um modelo AR de ordem 30 (cujos coeficientes foram obtidos pelo método Least Squares [13] a partir de um bloco de sinal de áudio real com taxa de amostragem de 44.1 kHz e cada amostra codificada com 16 bits). Inserimos dois *clicks* gerados artificialmente nas amostras 100 e 300, como mostrado na Figura 5.11. Os vetores de parâmetros dos *clicks* foram  $\mathbf{V} = [5 \ 15]^T$ ,  $\boldsymbol{\lambda} = [0,2 \ 0,4]^T$  e  $M = [10 \ 10]^T$ .

Para permitir a comparação dos resultados experimentais obtidos pelo método proposto com os esperados teoricamente, apresentamos gráficos das distribuições dos parâmetros do ruído. Para simplificar, apenas o primeiro *click* é considerado.

A Figura 5.12 mostra a distribuição conjunta teórica dos parâmetros  $\boldsymbol{\lambda}$  e  $\mathbf{V}$ , considerando as demais variáveis fixas em seus valores corretos. Podemos perceber que a superfície possui um único máximo global, que ocorre em torno de  $\lambda = 0,2$  e  $V = 10$ . A Figura 5.13 mostra o histograma obtido a partir de 10000 iterações do algoritmo, com apenas as variáveis  $V$  e  $\lambda$  sendo atualizadas.

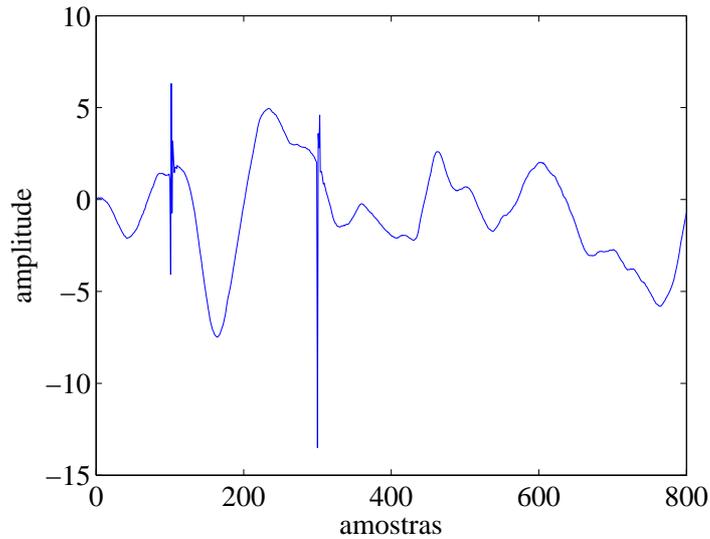


Figura 5.11: Sinal com *clicks* presentes a partir das amostras 100 e 300.

As Figuras 5.14, 5.15, 5.16 e 5.17 mostram os histogramas de  $V$ ,  $\lambda$ ,  $n_0$  e  $M$  e suas respectivas distribuições. Em todos os quatro casos o algoritmo realizou a atualização de apenas uma variável, permitindo que fosse feita a comparação entre os curvas teóricas e simuladas. A evolução dos parâmetros com o “tempo” é mostrada nas Figuras, 5.18, 5.19, 5.20 e 5.21. A semelhança das curvas nos permite verificar a correção do algoritmo, isto é, o algoritmo permite obter amostras da distribuição desejada.

O sinal restaurado é obtido pela média aritmética das amostras de  $\mathbf{x}$ , descartadas as 100 primeiras (quando supomos que a convergência foi atingida). A Figura 5.22 apresenta os sinais original e restaurado. A análise subjetiva do sinal revela que o *click* tornou-se auditivamente imperceptível após a restauração.

Também foi testado o caso em que *clicks* estão superpostos, como mostrado na Figura 5.23. Dois *clicks* foram inseridos num sinal artificial a partir das amostras 100 e 105. A Figura 5.24 mostra a evolução do parâmetro  $n_0$  com o tempo, ilustrando a convergência do algoritmo para os valores corretos.

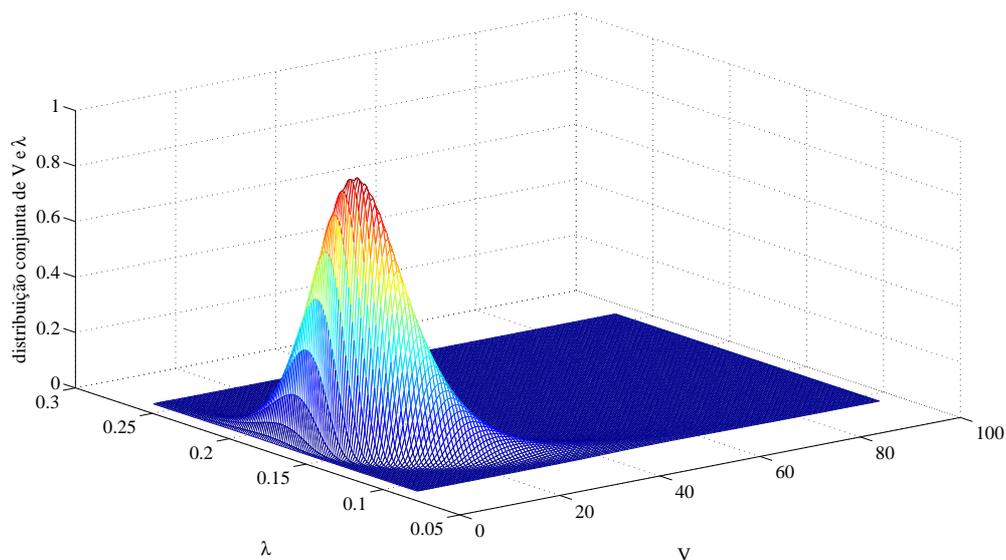


Figura 5.12: Distribuição conjunta de  $V$  e  $\lambda$ .

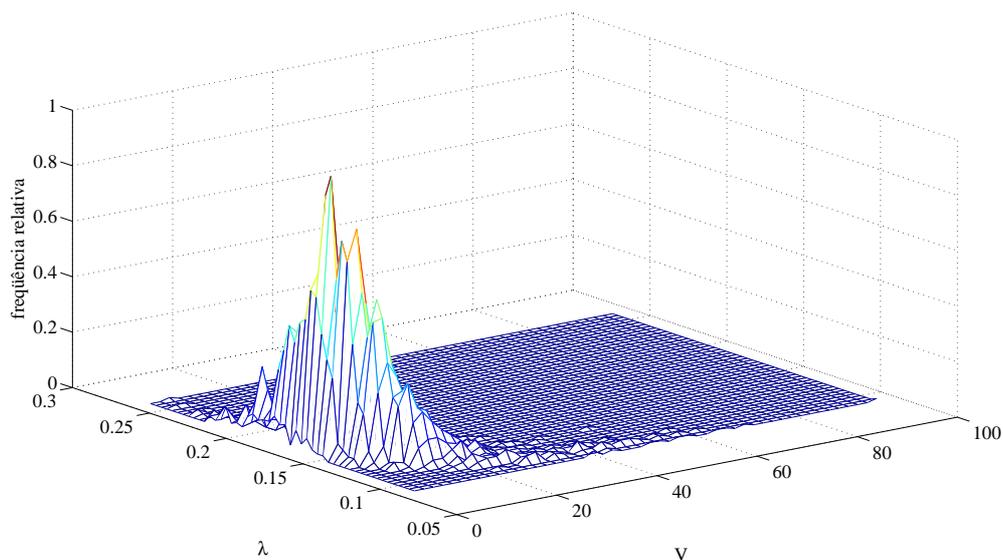


Figura 5.13: Histograma de  $V$  e  $\lambda$  sendo as demais variáveis fixas.

## 5.5 Restauração através de algoritmo de Metropolis-Hastings com saltos reversíveis

O algoritmo da seção anterior requer que o número de *clicks* presentes num bloco seja inicialmente conhecido. Nesta seção, apresentaremos um método baseado

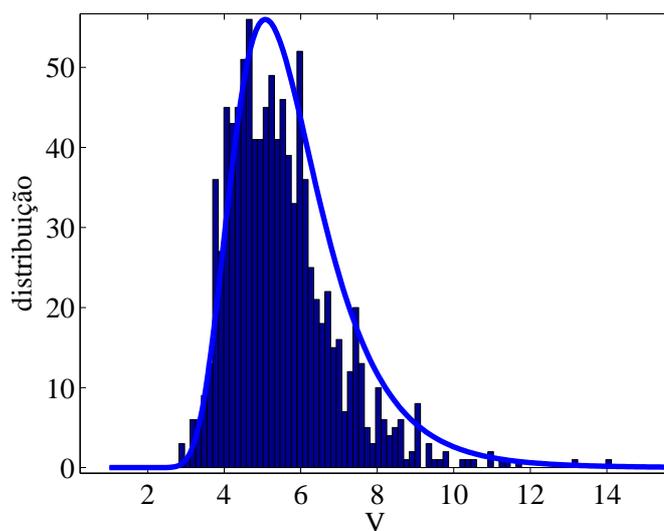


Figura 5.14: Distribuição e histograma de  $V$ .

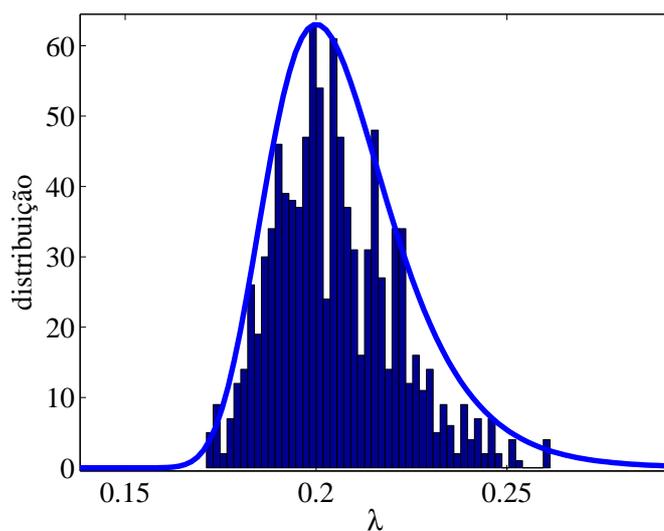


Figura 5.15: Distribuição e histograma de  $\lambda$ .

no algoritmo de Metropolis-Hastings com saltos reversíveis (Seção 3.6) que permite determinar o número de *clicks* de forma iterativa (além, é claro, das demais variáveis de interesse).

O método proposto foi inspirado no trabalho de Andrieu et al [32], no qual é apresentado um método para estimação de senóides presentes num sinal formado

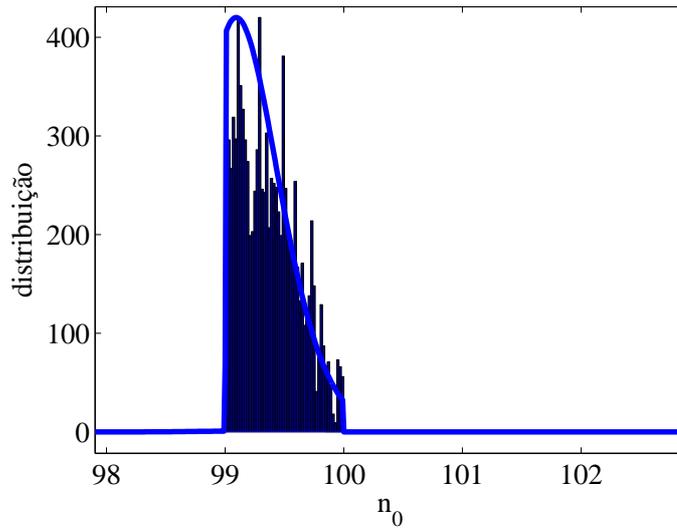


Figura 5.16: Distribuição e histograma de  $n_0$ .

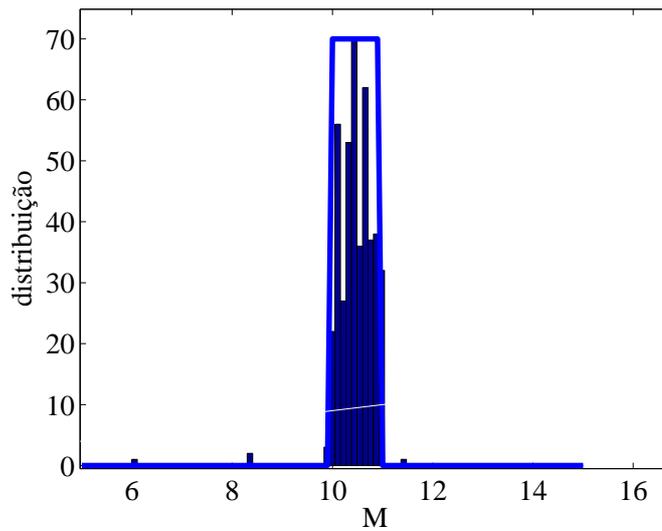


Figura 5.17: Distribuição e histograma de  $M$ .

por uma combinação linear de senóides, contaminado com ruído aditivo gaussiano. Naquele trabalho, o número de senóides é tratada como desconhecido, e o algoritmo MH com saltos reversíveis é aplicado. Podemos traçar um paralelo entre as frequências das senóides e a localização dos *clicks* do nosso problema.

Como vimos na Seção 3.6, o algoritmo RJ-MH permite fazer transições en-

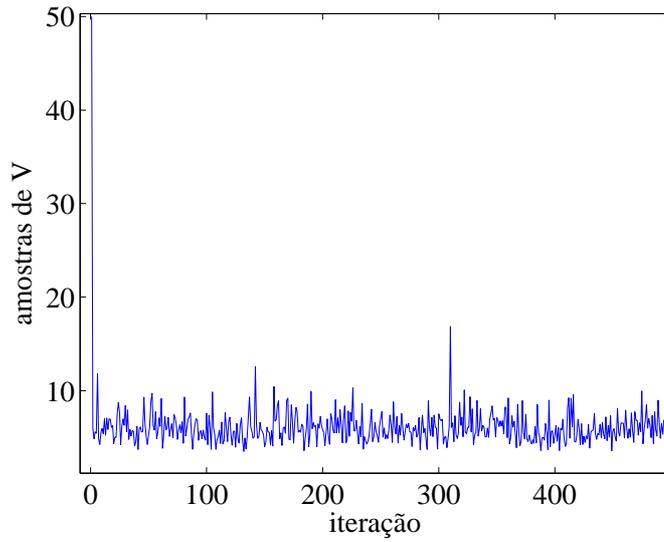


Figura 5.18: Amostras da condicional total de  $V$ .

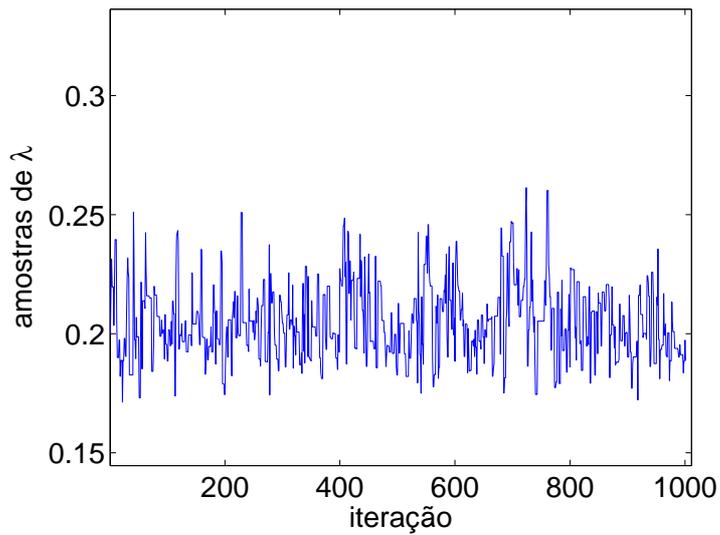


Figura 5.19: Amostras da condicional total de  $\lambda$

tre modelos de dimensões diferentes. Chamaremos essas transições de *movimentos*. Embora diversos tipos de movimento sejam permitidos, no presente trabalho usamos apenas três tipos simples: nascimento, morte e atualização. Nascimento (simbolizada por  $b$  de *birth*) significa a adição de um novo *click* ao conjunto de *clicks* da iteração anterior. Morte (simbolizada por  $d$  de *death*) significa retirada de um *click*

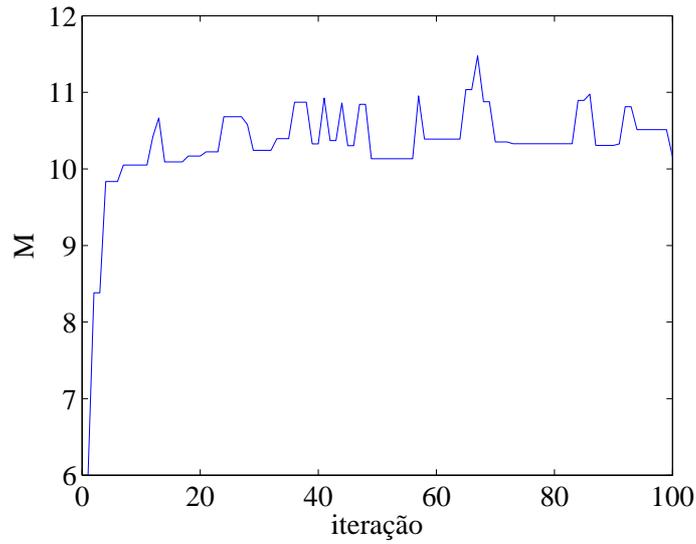


Figura 5.20: Amostras da condicional total de  $M$ .

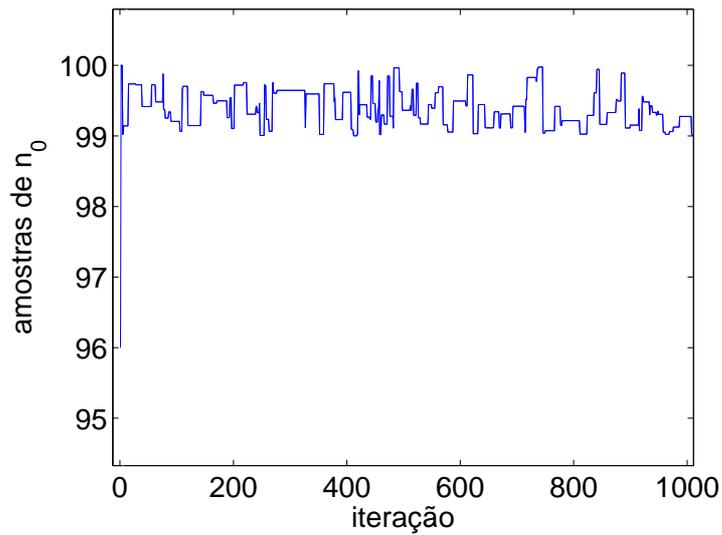


Figura 5.21: Amostras da condicional total de  $n_0$ .

do conjunto de *clicks* da iteração anterior. E atualização (simbolizada por  $u$  de *update*) é a amostragem das variáveis do sinal e do ruído com o número de *clicks* fixo (como apresentado na seção anterior). Seguindo a recomendação de [17], as probabilidades para aceitação dos movimentos de nascimento e morte são, respectivamente:

$$b_k = c \min \left\{ 1, \frac{P(k+1)}{P(k)} \right\} \text{ e} \quad (5.50)$$

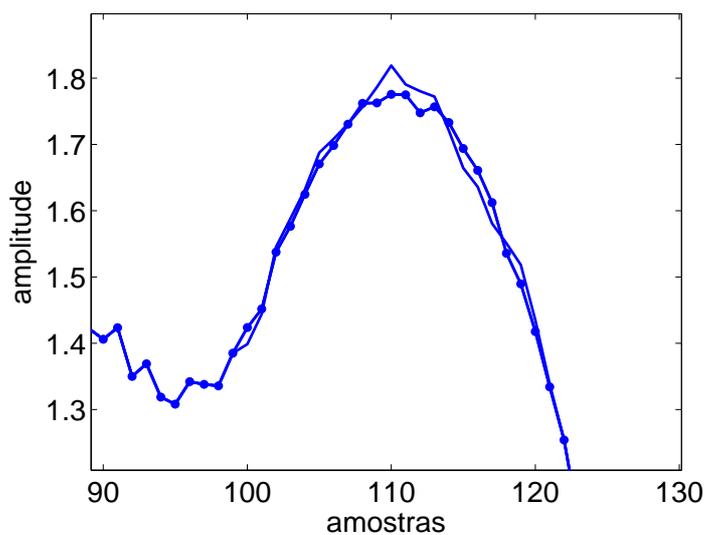


Figura 5.22: Sinais original (linha contínua) e restaurado pelo método baseado no algoritmo de Metropolis-Hastings (linha sólido-pontilhada).

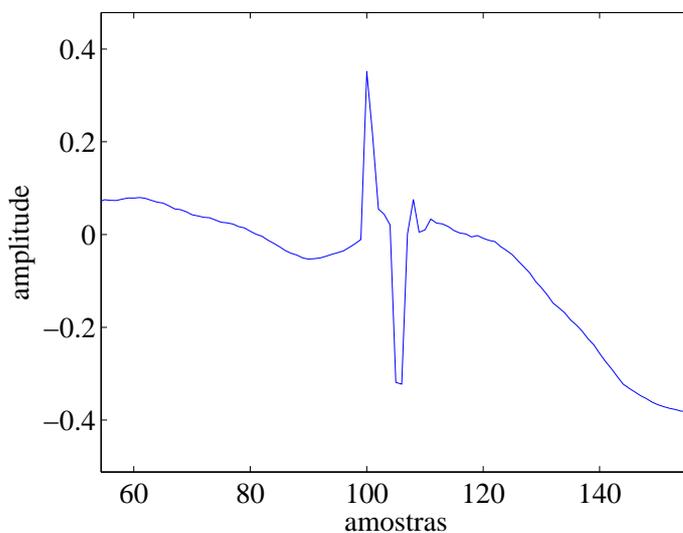


Figura 5.23: Sinal contaminado por *clicks* superpostos.

$$d_{k+1} = c \min \left\{ 1, \frac{P(k)}{P(k+1)} \right\}, \quad (5.51)$$

em que  $P(k)$  é a probabilidade *a priori* de  $k$  (número de *clicks*). A probabilidade de atualização,  $u_k$ , é tal que  $b_k + d_k + u_k = 1$ . O valor de  $c$  determina a proporção entre as propostas de atualização e mudança de dimensionalidade. Green [17] sugere que

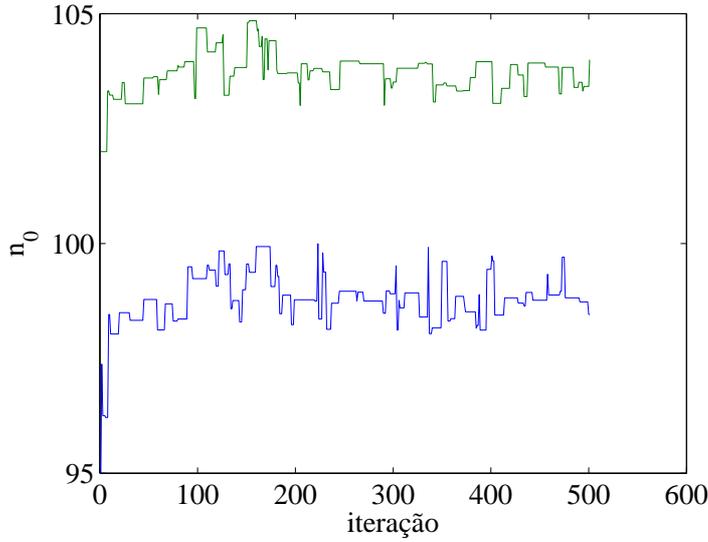


Figura 5.24: Localização dos *clicks* superpostos.

$c = 0,5$  seja usado, fazendo com que  $b_k + d_k \in [0,5, 1]$ .

A estrutura geral do algoritmo é descrita a seguir:

1. Inicialização: gerar os parâmetros  $k^{(0)}$ ,  $\theta_k^{(0)}$ ,  $\mathbf{a}^{(0)}$  e  $\sigma_e^{2(0)}$ .
2. Para  $j$  de 1 até  $N_{it}$ 
  - (a) Amostrar  $u \sim U(u|0, 1)$ ;
  - (b) Se  $(u \leq b_k)$ , então: determinar o “nascimento” de um *click*;
  - (c) Senão, se  $(u \leq b_k + d_k)$ , então: determinar a “morte” de um *click*;
  - (d) Senão, atualizar os parâmetros.

Descreve-se a seguir cada uma das operações.

### 1. Inicialização:

A inicialização pode ser feita a partir das distribuições *a priori* de cada variável ou através de algum pré-processamento. A geração de  $K^{(0)}$  e  $n_0^{(0)}$  pode ser feita usando-se o sinal do erro de predição pelo modelo AR,  $e(n)$ . As variáveis  $\mathbf{a}^{(0)}$  e  $\sigma_e^{2(0)}$  podem ser estimadas pelo critério *Least Squares* usando o próprio sinal corrompido. As demais variáveis podem ser inicializadas trivialmente a partir de suas distribuições *a priori*.

## 2. Nascimento:

Na operação de nascimento, um novo *click* é gerado e incorporado ao conjunto de *clicks* existentes na iteração anterior, e o valor de  $K$  é incrementado. As variáveis associadas aos *clicks* existentes são mantidas e as variáveis associadas ao novo *click* são propostas, formando um novo conjunto proposto:

$$\{k^*, \boldsymbol{\theta}_{k^*}^*\} = \left\{ k^{(j)} + 1, \left[ \boldsymbol{\theta}_{k^{(j)}}^{(j)T} \quad n_0^* \quad M^* \quad V^{2*} \quad \lambda^* \right]^T \right\}. \quad (5.52)$$

A variável  $n_0^*$  é gerada por amostragem da distribuição  $q_{n_0}^{(j)}(n) \propto e(n)$ , normalizada para que a soma resulte em 1. Se a proposta for aceita, reduzimos a função  $q_{n_0}^{(j)}(n)$  de um fator (2, por exemplo) no valor de  $n_0^*$  amostrado, gerando uma nova distribuição  $q_{n_0}^{(j+1)}(n)$ , que deve preservar a soma em 1. O objetivo é evitar que o algoritmo proponha muitas vezes os mesmos valores, o que tornaria a convergência mais lenta.

As variáveis  $M$ ,  $V^{2*}$  e  $\lambda^*$  são geradas a partir de suas respectivas prioris. De acordo com a notação da Seção 3.6, teremos:

$$q_1(k^*|k) = q_1(k^{(j)} + 1|k^{(j)}) = b_{k^{(j)}}, \quad (5.53)$$

$$q_1(k^{(j)}|k^*) = q_1(k^{(j)}|k^{(j)} + 1) = d_{k^*}, \quad (5.54)$$

$$\begin{aligned} q_2(\boldsymbol{\theta}_{k^*}^*|\boldsymbol{\theta}_{k^{(j)}}^{(j)}) &= q_2(\boldsymbol{\theta}_{k^{(j)}+1}^{(j+1)}|\boldsymbol{\theta}_{k^{(j)}}^{(j)}) = \\ &= \delta_{k^{(j)}} \left( (\boldsymbol{\theta}_{k^*}^*)^{1:k^{(j)}} - \boldsymbol{\theta}_{k^{(j)}}^{(j)} \right) q_{n_0}^{(j)}(n_0^*) p(M^*) p(V^{2*}) p(\lambda^*), \end{aligned} \quad (5.55)$$

$$q_2(\boldsymbol{\theta}_{k^{(j)}}^{(j)}|\boldsymbol{\theta}_{k^*}^*) = q_2(\boldsymbol{\theta}_{k^{(j)}}^{(j)}|\boldsymbol{\theta}_{k^{(j)}+1}^{(j+1)}) = \delta_{k^{(j)}} \left( (\boldsymbol{\theta}_{k^*}^*)^{1:k^{(j)}} - \boldsymbol{\theta}_{k^{(j)}}^{(j)} \right), \quad (5.56)$$

em que  $\delta_k$  é a função impulso unitário  $k$  dimensional e a notação  $(\boldsymbol{\theta})^{1:k}$  indica a partição de  $\boldsymbol{\theta}$  relacionada aos  $k$  primeiros *clicks*. Sendo assim, de acordo com a Equação (3.26), com probabilidade

$$\alpha_b = \min \left\{ 1, \frac{p(k^*, \boldsymbol{\theta}_{k^*}^*|\mathbf{y}, \boldsymbol{\omega})}{p(k^{(j)}, \boldsymbol{\theta}_{k^{(j)}}^{(j)}|\mathbf{y}, \boldsymbol{\omega})} \frac{1}{q_{n_0}^{(j)}(n_0^*) p(M^*) p(V^{2*}) p(\lambda^*)} \frac{d_{k^*}}{b_{k^{(j)}}} \right\} \quad (5.57)$$

a proposta é aceita, fazemos  $\{k^{(j+1)}, \boldsymbol{\theta}_{k^{(j+1)}}^{(j+1)}\} = \{k^*, \boldsymbol{\theta}_{k^*}^*\}$  e o algoritmo inicia uma nova iteração.

### 3. Morte:

A operação de morte remove um *click* aleatoriamente (com igual probabilidade para todos), isto é, um dos  $k^{(j)}$  *clicks*, digamos o  $m$ -ésimo, é removido. A variável  $k$  é decrementada e as variáveis relacionadas ao  $m$ -ésimo *click* são retiradas do vetor  $\boldsymbol{\theta}_{k^{(j)}}$ , formando o vetor  $\boldsymbol{\theta}_{k^*}$ . A probabilidade de aceitação da operação de morte é obtida seguindo os mesmos passos apresentados no item anterior, e o resultado é:

$$\alpha_d = \min \left\{ 1, \frac{p(k^*, \boldsymbol{\theta}_{k^*} | \mathbf{y}, \boldsymbol{\omega})}{p(k^{(j)}, \boldsymbol{\theta}_{k^{(j)}} | \mathbf{y}, \boldsymbol{\omega})} \frac{q_{n_0}^{(j_m)}(n_0^{(m)}) p(M^{(m)}) p(V^{2(m)}) p(\lambda^{(m)}) b_{k^*}}{1 d_{k^{(j)}}} \right\}, \quad (5.58)$$

em que  $j_m$  é a iteração na qual o  $m$ -ésimo *click* foi proposto.

### 4. Atualização:

A operação de atualização envolve os mesmos passos descritos na Seção 5.4 acrescidos do passo de amostragem dos parâmetros do modelo AR,  $\mathbf{a}$  e  $\sigma_E^2$ . Considerando uma priori conjunta  $p(\mathbf{a}, \sigma_e^2) = \text{IG}(\sigma_e^2 | \alpha_e, \beta_e)$  (independente de  $\mathbf{a}$ ), a amostragem desses parâmetros na  $j$ -ésima iteração é dada por [4]:

$$\mathbf{a}^{(j)} \sim \text{N}(\mathbf{a} | (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{x}_1, \sigma_e^{2(j-1)} (\mathbf{X}^T \mathbf{X})^{-1}), \quad (5.59)$$

$$\sigma_e^{2(j)} \sim \text{IG}(\sigma_e^2 | \alpha_e + (N - P)/2, \beta_e + E(\mathbf{x}^{(j)}, \mathbf{a}^{(j)})/2), \quad (5.60)$$

em que  $\mathbf{X}$  é obtido do sinal  $\mathbf{x}^{(j)}$  de acordo com a Equação (4.11) (o índice foi omitido por clareza) e  $E(\mathbf{x}^{(j)}, \mathbf{a}^{(j)})/2 = \mathbf{e}^T \mathbf{e}$ , sendo  $\mathbf{e} = \mathbf{A}^{(j)} \mathbf{x}^{(j)}$  o erro de predição calculado usando-se os últimos dados amostrados de  $\mathbf{a}$  e  $\mathbf{x}$ .

Os valores de  $\alpha_e$  e  $\beta_e$  podem ser escolhidos bem baixos, o que implica uma priori bastante vaga. Usamos, como em [4],  $\alpha_e = \beta_e = 10^{-10}$ .

## 5.5.1 Resultados

Para análise do algoritmo foram feitas simulações em três situações distintas: (A) Sinal artificial gerado a partir de um modelo AR com *clicks* artificiais inseridos (todos com  $V = 0.3$ ,  $M = 10$  e  $\lambda = 0.5$ ); (B) Sinal artificial com *clicks* reais inseridos; e (C) Sinal real corrompido por *clicks*. Na situação A os parâmetros do

modelo AR são tratados como conhecidos e iguais aos que foram usados para gerar o sinal. O modelo AR tem ordem 30, e na situação C os parâmetros do modelo são inicializados pelo método *Least Squares* a partir do próprio sinal corrompido e são atualizados durante a execução do algoritmo de acordo com o apresentado na seção anterior.

Nas três situações, o sinal tem comprimento de 800 amostras. Nas situações A e B, três *clicks* foram inseridos nas posições 100, 300 e 500, e na situação C pode-se identificar claramente um *click* próximo à amostra 380. Em todas as situações, o algoritmo é inicializado com  $K^{(0)} = 0$  (isto é, nenhum *click*) e roda por 50 iterações e a média das 20 últimas amostras de  $x$  é calculada para se obter o sinal restaurado.

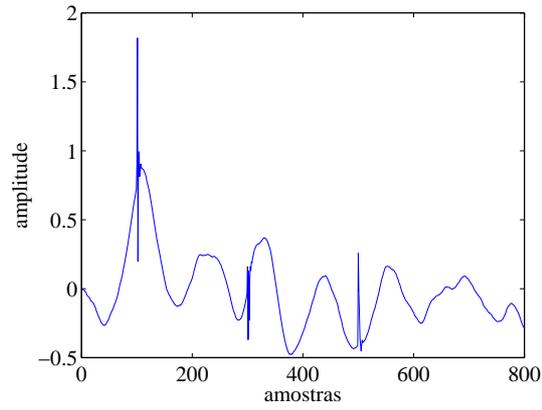
Os resultados referentes à situação A estão mostrados nas Figuras 5.25 e 5.26, nas quais aparecem os gráficos dos sinais original, corrompido e restaurado, bem como a evolução das variáveis  $\mathbf{n}_0$ ,  $\mathbf{M}$ ,  $\mathbf{V}^2$  e  $\lambda$ . A análise visual indica que a variável  $\mathbf{n}_0$  aproxima-se de seu valor correto antes da décima iteração, o que não ocorre com as demais. Isso deve-se ao formato da distribuição de  $\mathbf{n}_0$ , que é acentuadamente concentrada em um certo ponto. Embora não se tenha usado técnicas sofisticadas para diagnóstico de convergência, obtiveram-se resultados úteis em torno da vigésima iteração.

As Figuras 5.27 e 5.28 mostram essas mesmas variáveis relativas à situação B. Observamos um comportamento das variáveis similar ao da situação A, um indicativo de que o modelo proposto lida bem com *clicks* reais.

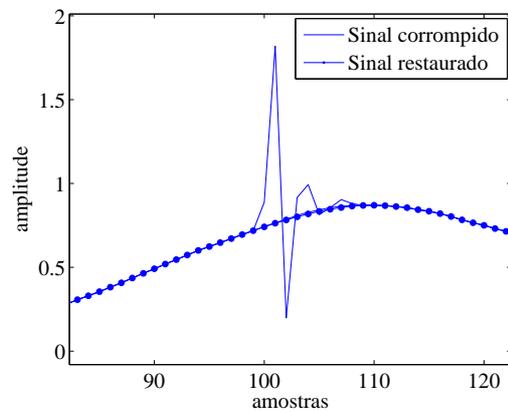
Os gráficos relativos à situação C estão mostrados nas Figuras 5.29 e 5.30. Nessa simulação, os parâmetros do modelo AR foram tratadas como variáveis aleatórias e atualizados durante a execução do algoritmo. Talvez por esse motivo tenha-se observado nessa situação um maior “tempo” para convergência, que parece ocorrer em torno da iteração 30. As razões para esse comportamento deverão ser investigadas em trabalhos futuros.

## 5.5.2 Análise da complexidade computacional

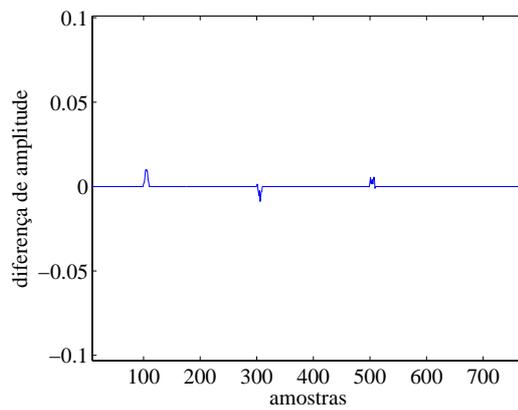
Nesta seção será analisada a complexidade das principais etapas de cada iteração do algoritmo.



(a)

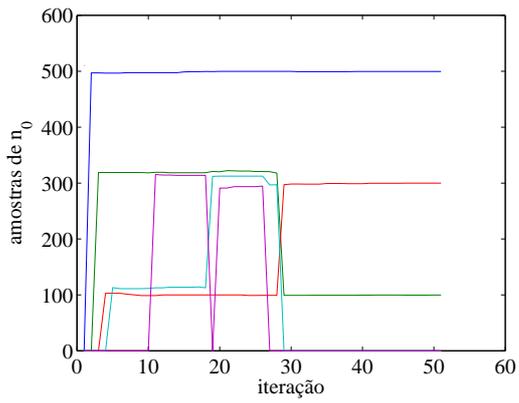


(b)

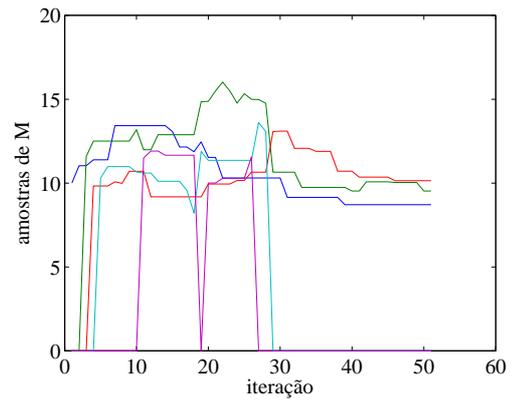


(c)

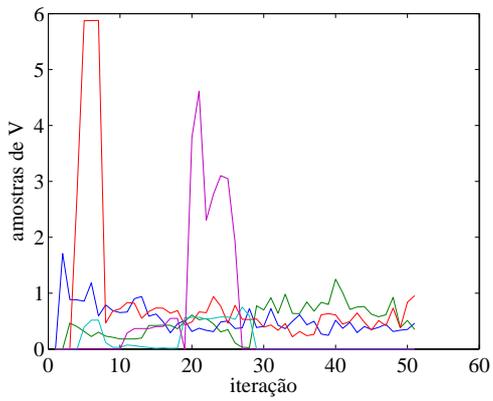
Figura 5.25: Gráficos relativos à situação A. (a) Sinal gerado artificialmente usando modelo AR de ordem 30 com 3 *clicks* artificiais inseridos. (b) Detalhe do sinal corrompido (linha contínua) e sinal restaurado (linha sólido-pontilhada). (c) Diferença entre sinais original e restaurado.



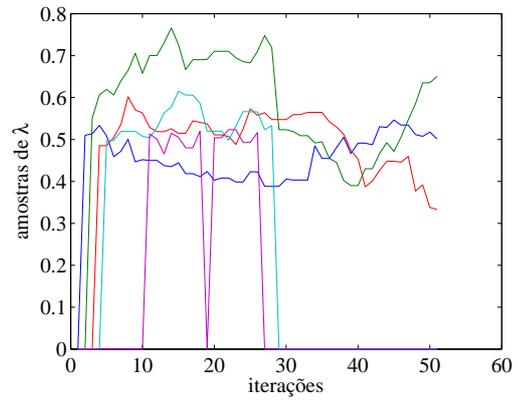
(a)



(b)

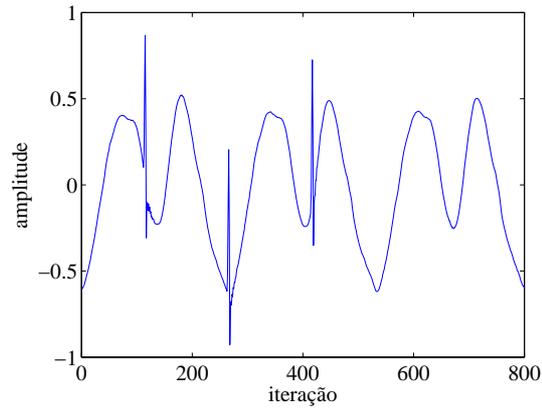


(c)

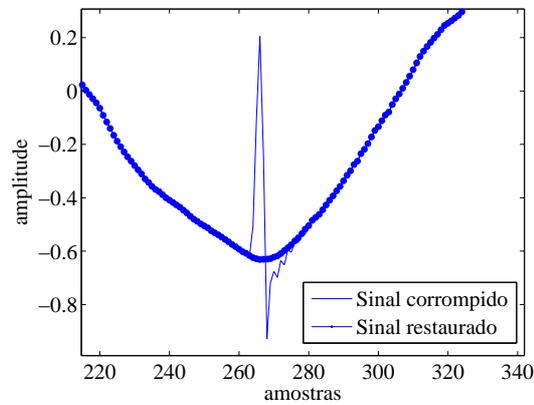


(d)

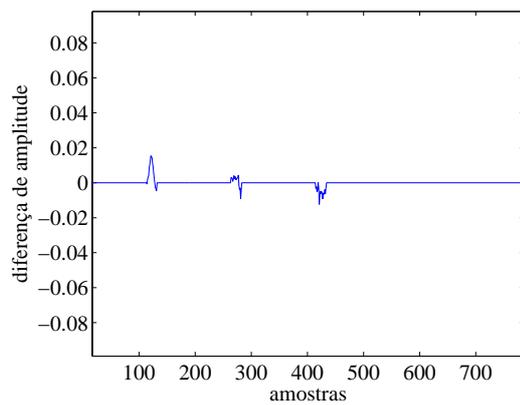
Figura 5.26: Gráficos relativos à situação A. (a) Evolução de  $\mathbf{n}_0$ . (b) Evolução de  $\mathbf{M}$ . (c) Evolução de  $\mathbf{V}$ . (d) Evolução de  $\lambda$ .



(a)

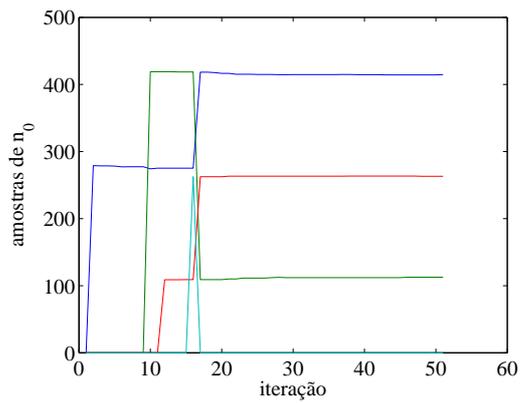


(b)

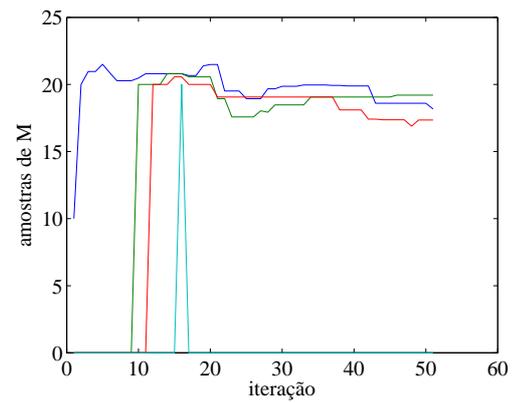


(c)

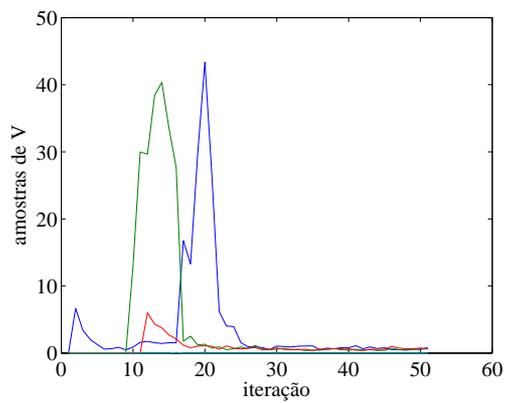
Figura 5.27: Gráficos relativos à situação B. (a) Sinal gerado artificialmente usando modelo AR de ordem 30 com 3 *clicks* reais inseridos. (b) Sinal corrompido (linha contínua) e sinal restaurado (linha sólido-pontilhada). (c) Diferença entre sinais original e restaurado.



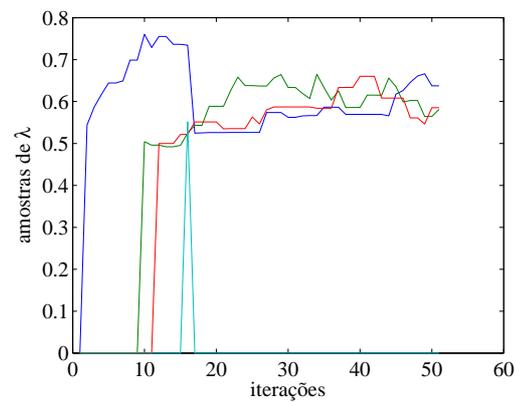
(a)



(b)

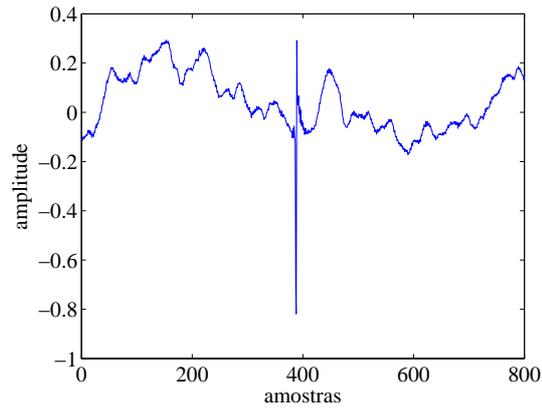


(c)

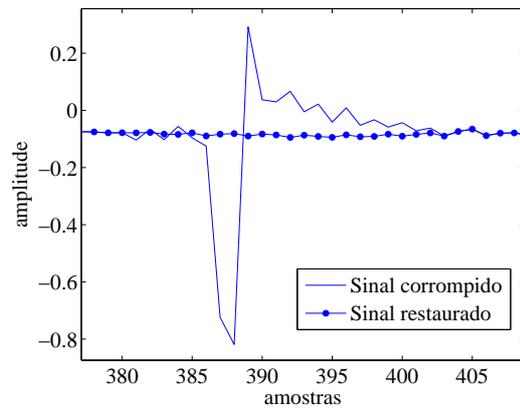


(d)

Figura 5.28: Gráficos relativos à situação B. (a) Evolução de  $\mathbf{n}_0$ . (b) Evolução de  $\mathbf{M}$ . (c) Evolução de  $\mathbf{V}$ . (d) Evolução de  $\lambda$ .

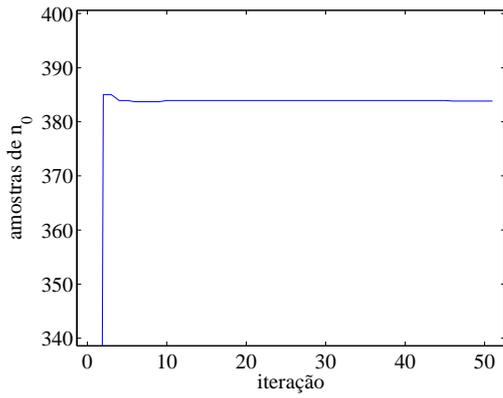


(a)

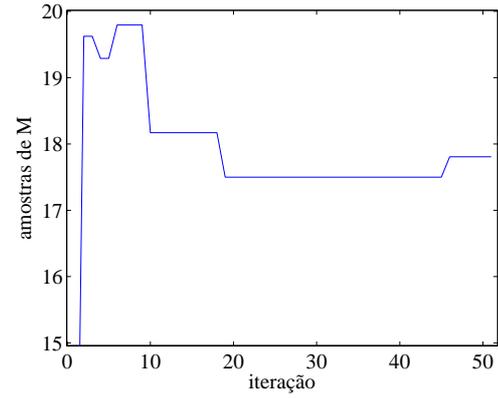


(b)

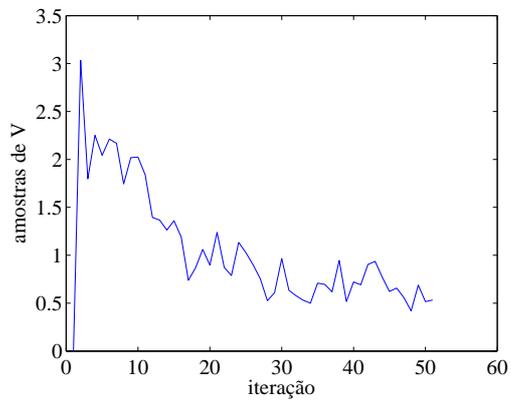
Figura 5.29: Gráficos relativos à situação C. (a) Sinal real contaminado por *click*. (b) Detalhe do sinal corrompido (linha contínua) e sinal restaurado (linha sólido-pontilhada).



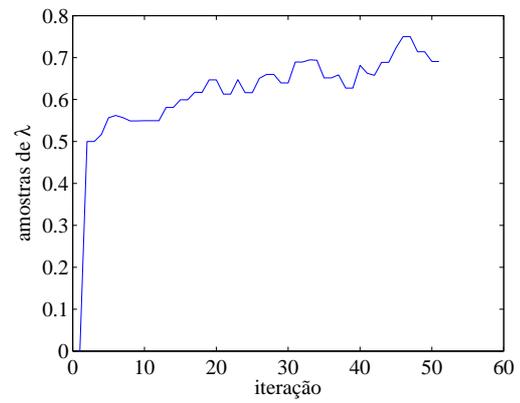
(a)



(b)



(c)



(d)

Figura 5.30: Gráficos relativos à situação C. (a) Evolução de  $\mathbf{n}_0$ . (b) Evolução de  $\mathbf{M}$ . (c) Evolução de  $\mathbf{V}$ . (d) Evolução de  $\lambda$ .

- Inicialização. Em princípio, a inicialização poderia ser feita associando-se quaisquer valores às variáveis  $\mathbf{n}_0^{(0)}$ ,  $\mathbf{M}^{(0)}$ ,  $\boldsymbol{\lambda}$ ,  $\mathbf{V}^{2(0)}$ ,  $\sigma_e^{2(0)}$  e  $\mathbf{a}^{(0)}$ . Para acelerar a convergência, é adotado um processamento que fornece uma estimativa inicial para essas variáveis, descrito na Seção 5.5. Essa etapa envolve operações com matrizes da ordem do número de amostras corrompidas no bloco de sinal. A complexidade das operações é muito menor do que a das etapas subsequentes e pode ser desprezada.
- Amostragem de  $\mathbf{n}_0$  e  $\mathbf{M}$ . A operação computacionalmente mais complexa é a obtenção da matriz  $\mathbf{x}^{\text{MAP}}$ , que é dominada pela inversão de uma matriz de ordem  $M_k$ , que pode ser feita em  $O(M_k^3)$  operações. Esse procedimento deve ser feito para cada um dos  $K$  *clicks* propostos na iteração em questão. Como cada amostragem exige o cálculo da condicional para os valores propostos e atuais, segue que a complexidade total dessa etapa é de:

$$O\left(\sum_{k=1}^K 2M_k^3\right). \quad (5.61)$$

- Amostragem de  $\mathbf{x}$ . A amostragem dessa variável, a partir de uma distribuição gaussiana, pode ser simplificada usando-se o valor de  $\mathbf{x}^{\text{MAP}}$  obtido na etapa anterior. Métodos eficientes para amostragem de uma distribuições conjuntas, baseados na decomposição de Cholesky, permitem realizar essa operação em  $O(M_k^3)$  operações.
- Amostragem de  $\mathbf{V}^2$ . Essa operação consiste na amostragem de distribuição gama inversa, que pode ser feita de forma eficaz, com complexidade que pode ser desprezada comparando-se com as demais operações.
- Amostragem de  $\boldsymbol{\lambda}$ . Envolve basicamente o cálculo de expressões simples (Equação (5.49)), cuja complexidade também pode ser desprezada.
- Amostragem de  $\sigma_e^2$ . É feita a partir de uma distribuição gama-inversa de forma simples.
- Amostragem de  $\mathbf{a}$ . Consiste na amostragem de uma distribuição gaussiana de dimensão  $P$ , que, como vimos pode ser feita em  $O(P^3)$  operações.

A complexidade total por iteração é a soma das complexidades das etapas anteriores, que é dominada pela amostragem de  $\mathbf{n}_0$  e  $\mathbf{a}$ . Para reduzir a complexidade, as variáveis do modelo AR poderiam não ser amostradas a cada iteração, como sugerido em [4].

O custo total do algoritmo está associado ao número de iterações necessárias para atingir o que se poderia considerar como convergência. Na prática, se usarmos pré-processamento de inicialização, valores a partir de 10 iterações já parecem suficientes para se obter um resultado aceitável. Todavia, a análise informal da evolução das variáveis indica que a convergência só é atingida com várias dezenas de iterações (tipicamente, entre 50 e 100 iterações).

## 5.6 Conclusão

Este capítulo apresentou um novo modelo para a descrição de *clicks*, que os trata de forma individual através de variáveis de localização, duração, amplitude e taxa de decaimento. Métodos bayesianos foram aplicados ao modelo, e técnicas numéricas foram adotadas para obtenção do sinal restaurado. A aplicabilidade do modelo a casos práticos foi avaliada através de um algoritmo baseado no método de Fibonacci, que, todavia, é limitado por exigir que o bloco de sinal possua apenas um *click*. Uma solução baseada no algoritmo MH e no amostrador de Gibbs permitiu a generalização para o tratamento de blocos de sinais com mais de 1 *click*, mas ainda exigia que o número de *clicks* fosse conhecido. Por fim, a incorporação do algoritmo RJ-MH gerou um algoritmo que permitia tratar o caso mais geral em que o número de *clicks* é desconhecido.

Numa implementação não-otimizada em Matlab, sendo executada num computador pessoal moderno, o tempo total de processamento, tipicamente, pode ser até 3 ordens de grandeza maior que a duração do sinal processado.

A diferença do algoritmo proposto em relação ao algoritmo baseado no amostrador de Gibbs de Godsill (Seção 4.4.3) está na amostragem das variáveis do ruído. Segundo o autor, a menor complexidade por iteração que se pode obter na detecção (estimação do vetor  $\mathbf{i}$ ) ocorre quando se consideram sub-blocos de apenas uma amos-

tra (todavia, o autor adverte que nessa situação a convergência é mais lenta). Nesse caso, a complexidade é de  $O(NP)$ , sendo que  $N$  é o tamanho do bloco e  $P$  é a ordem do modelo AR. Em nosso algoritmo, a detecção se dá pela amostragem da variável  $\mathbf{n}_0$ , cuja complexidade é expressa na Equação 5.49, que depende da quantidade e duração dos *clicks* presentes em cada bloco do sinal. Como consequência, o algoritmo proposto poderá ter menor ou maior complexidade que o de Godsill, dependendo do número médio de *clicks* por bloco. Como tipicamente o número de *clicks* por blocos é de algumas unidades (blocos com mais de 4 *clicks* são raros mesmo em sinais muito corrompidos), devemos esperar que, em média, nosso método possua complexidade menor.

# Capítulo 6

## Correção de Pulsos Longos

Um dos tipos mais desagradáveis de distúrbio encontrados em gravações analógicas de áudio são os pulsos de longa duração e com conteúdo significativo de baixa frequência. São caracterizados por uma descontinuidade de alta amplitude (similar a um *click*) seguida de um longo transitório de frequência decrescente, que chamaremos de cauda. A Figura 6.1 mostra um pulso longo típico. As linhas verticais indicam a região correspondentes à descontinuidade inicial.

Em geral, os pulsos longos são causados por defeitos graves na mídia, como por exemplo arranhões ou mesmo quebra, casos em que se observam pulsos aproximadamente periódicos. Os transitórios de baixa frequência resultam da resposta do equipamento de reprodução à descontinuidade inicial, o que lhes confere um formato mais ou menos bem definido que pode ser explorado para sua detecção/correção.

### 6.1 Técnicas existentes

Uma das primeiras técnicas para correção digital de pulsos longos apareceu em [2] e posteriormente em [24]. O método se baseia na similaridade tipicamente presente entre os pulsos. Assume-se que os pulsos longos presentes num dado sinal diferem de um pulso-padrão apenas em amplitude. A localização dos pulsos é feita deslocando-se o pulso-padrão ao longo do sinal, calculando-se a correlação entre o pulso-padrão e o sinal, e selecionando a amostra que fornecer a maior correlação. Para a obtenção de uma estimativa do sinal original, obtém-se a diferença entre o si-

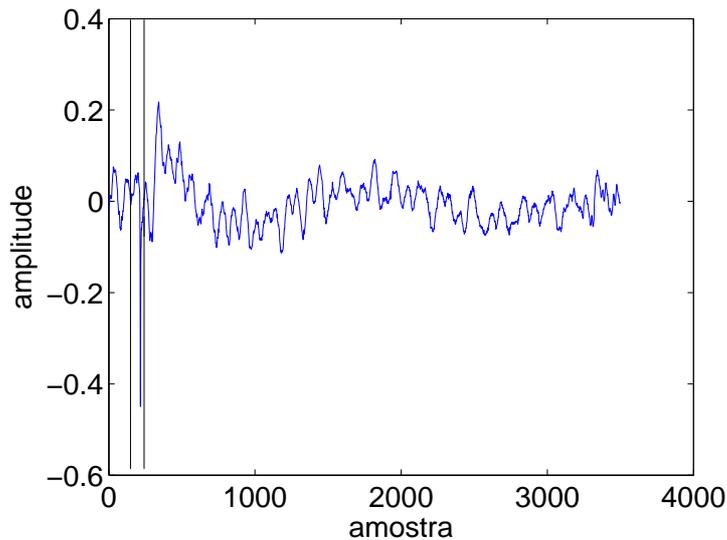


Figura 6.1: Trecho de sinal contaminado por pulso longo.

nal degradado e o pulso-padrão multiplicado por um fator (que pode ser trivialmente obtido assumindo não-correlação entre o pulso-padrão e o sinal original).

O método apresenta bons resultados quando a hipótese de similaridade é verdadeira, o que ocorre por exemplo se as falhas são causadas por quebra do disco. Em casos práticos, entretanto, tal hipótese nem sempre é válida, sendo comum a presença de pulsos de formatos bastante variados e até mesmo a superposição entre pulsos muito próximos.

Em [4] e [5], é apresentado um método que permite tratar casos mais gerais de pulsos longos, ao custo de uma alta complexidade computacional. O método associa modelos AR diferentes para o sinal e para o pulso, sendo o sinal degradado modelado pela soma de ambos. O sinal original é então estimado separando-se os dois processos. O método requer que a localização do pulso e os parâmetros do modelo sejam conhecidos inicialmente. O autor sugere técnicas simples para obtenção dessas grandezas; todavia, a detecção do pulso enfrenta o problema de confundir a descontinuidade existente no início do pulso longo com um *click* comum.

Um método para correção de pulsos longos de menor complexidade com resultados similares é apresentado em [33]. Nesse trabalho, uma filtragem não-linear chamada *two-pass split window* (TPSW) é usada para obtenção de uma estimativa

do formato do pulso, que é então suavizada através de ajuste polinomial por partes.

## 6.2 Método proposto

A tentativa de aplicar diretamente o algoritmo para remoção de *clicks* apresentado no capítulo anterior ao problema de pulsos longos mostrou-se sem sucesso, o que não chega a surpreender, dada a diferença dos tipos de degradação, que não permite compartilhar um mesmo modelo.

Propusemos uma modificação do modelo do ruído, que permitisse caracterizar melhor os pulsos longos e mantivesse a estrutura do algoritmo. O modelo proposto do ruído descreve o pulso como uma descontinuidade inicial seguida de oscilações amortecidas e de frequência decrescente. A descontinuidade é modelada por ruído branco de  $M$  amostras com variância  $\sigma_{V_d}^2$  fixa (ao contrário do usado em nosso modelo para correção de *clicks*, em que a variância decrescia com o tempo):

$$v_d(n) = r(n) [u(n - n_0) - u(n - n_0 - M)], \quad (6.1)$$

em que  $r(n) = N(0, \sigma_{V_d}^2)$  e tanto  $n_0$  e  $M$  são supostos desconhecidos inicialmente.

O modelo para o transitório de baixa frequência foi inspirado em [33] e é matematicamente descrito por:

$$v_c(n) = V_c e^{-n/f_s \tau_e} \sin \left( 2\pi n \frac{f_n}{f_s} + \phi \right) [u(n - n_0 - M - 1)], \quad (6.2)$$

sendo

$$f_n = (f_{\max} - f_{\min}) e^{-n/f_s \tau_f} + f_{\min}, \quad (6.3)$$

em que [33]:

- $f_s$  é a taxa de amostragem do sinal (usualmente 44,1 kHz);
- $\tau_e$  é a constante de tempo em segundos associada ao decaimento do envoltória do pulso;
- $f_{\max}$  e  $f_{\min}$  são, respectivamente, as frequências de oscilação máxima e mínima em Hz admitidas na cauda;

- $\tau_f$  é a constante de tempo em segundos associada ao decaimento da frequência do pulso;
- $\phi$  é a fase inicial do pulso.

(As variáveis  $n_0$  e  $M$  das equações 6.1 e 6.2 são as mesmas.)

Esse modelo pode ser considerado determinístico no sentido de que, se suas variáveis forem conhecidas, o formato do pulso estará unicamente determinado.

Para simplificar, vamos dividir o vetor de variáveis do ruído em duas partes,  $\boldsymbol{\theta}_d$  e  $\boldsymbol{\theta}_c$ , de tal forma que  $\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_d^T & \boldsymbol{\theta}_c^T \end{bmatrix}^T$ , sendo

$$\boldsymbol{\theta}_d = \begin{bmatrix} n_0 & M & \sigma_{V_d}^2 \end{bmatrix}^T \text{ e} \quad (6.4)$$

$$\boldsymbol{\theta}_c = \begin{bmatrix} V_c & \tau_e & \tau_f & \phi & f_{\min} & f_{\max} \end{bmatrix}^T. \quad (6.5)$$

### 6.3 Cálculo das distribuições condicionais e escolha das propostas

O algoritmo para correção de pulsos longos usando o modelo proposto tem estrutura similar ao usado no capítulo anterior. Em linhas gerais, é preciso calcular a condicional total para cada uma das variáveis envolvidas e usar algum procedimento para obter amostras delas. Quando a distribuição resultante tem um formato conhecido, as amostras podem ser obtidas através de métodos simples e rápidos; no caso em que a distribuição não é conhecida o algoritmo de Metropolis-Hastings é usado, sendo necessária a escolha de uma distribuição proposta adequada.

Como feito no capítulo anterior, começamos definindo o sinal  $i(n)$ , que agora pode assumir os valores: 0, 1 e 2, que indicam, respectivamente, se a amostra pertence a uma região sem ruído, à descontinuidade inicial do pulso longo ou ao transitório de baixa frequência (cauda). Designamos por  $\mathbf{i}_{(0)}$ ,  $\mathbf{i}_{(1)}$  e  $\mathbf{i}_{(2)}$  as partições do vetor  $\mathbf{i}$  correspondentes às amostras 0, 1 e 2. De forma similar definimos os vetores  $\mathbf{x}_{(i_0)}$ ,  $\mathbf{x}_{(i_1)}$ ,  $\mathbf{x}_{(i_2)}$ ,  $\mathbf{y}_{(i_0)}$ ,  $\mathbf{y}_{(i_1)}$ ,  $\mathbf{y}_{(i_2)}$ . As seguintes relações são satisfeitas:

$$\mathbf{y}_{(i_0)} = \mathbf{x}_{(i_0)}, \quad (6.6)$$

$$\mathbf{y}_{(i_1)} = \mathbf{x}_{(i_1)} + \mathbf{v}_d, \quad (6.7)$$

$$\mathbf{y}_{(i_2)} = \mathbf{x}_{(i_2)} + \mathbf{v}_c. \quad (6.8)$$

A distribuição *a posteriori* dos parâmetros e do sinal é dada por:

$$p(\boldsymbol{\theta}, \mathbf{x} | \mathbf{y}) \propto p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}), \quad (6.9)$$

sendo:

$$p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}) = \delta(\mathbf{y}_{(i_0)} - \mathbf{x}_{(i_0)}) N(\mathbf{y}_{(i_1)} - \mathbf{x}_{(i_1)} | \mathbf{0}, \mathbf{R}_{v_d}) \delta(\mathbf{y}_{(i_2)} - \mathbf{x}_{(i_2)} - \mathbf{v}_c), \quad (6.10)$$

$$p(\mathbf{x} | \boldsymbol{\theta}) = N(\mathbf{x} | \mathbf{0}, \mathbf{R}_x), \quad (6.11)$$

$$\mathbf{R}_{v_d} = \sigma_{V_d}^2 \mathbf{I}_M, \quad (6.12)$$

$$\mathbf{R}_x = \frac{1}{\sigma_E^2} \mathbf{A}^T \mathbf{A}, \quad (6.13)$$

e, assumindo independência *a priori* entre os parâmetros do ruído:

$$p(\boldsymbol{\theta}) = \prod_i \theta_i. \quad (6.14)$$

Vamos agora considerar separadamente a amostragem de cada parâmetro desconhecido ( $\boldsymbol{\theta}$  e  $\mathbf{x}$ ).

A variância de  $\mathbf{v}_d$  é amostrada de forma análoga à variável  $V^2$  do capítulo anterior. Assumindo para  $\sigma_{V_d}^2$  uma priori do tipo gama inversa, com parâmetros  $\alpha_{v_d}$  e  $\beta_{v_d}$ , essa a variável é amostrada a partir da distribuição  $\text{IG}(\sigma_{V_d}^2 | \alpha'_v, \beta'_v)$  com:

$$\alpha'_{v_d} = \alpha_{v_d} + \frac{M}{2} \quad (6.15)$$

e

$$\beta'_v = \beta_v + \sum_{i=0}^{M-1} v_d(n_0 + i)^2. \quad (6.16)$$

A amostragem das demais variáveis é feita pelo método da composição, descrito na Seção 5.4. Amostra-se inicialmente  $\boldsymbol{\theta}_{-\sigma_{v_d}^2}$  (isto é, todos os componentes de  $\boldsymbol{\theta}$  com exceção de  $\sigma_{v_d}^2$ ) a partir de  $p(\boldsymbol{\theta}_v | \mathbf{y}, \sigma_{v_d}^2)$  (isto é,  $\mathbf{x}$  é marginalizado) e em seguida amostra-se  $\mathbf{x}$  (na extensão inteira da cauda) a partir de sua condicional total. Para calcular  $p(\boldsymbol{\theta} | \mathbf{y})$ , inicialmente integra-se a verossimilhança  $p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta})$  em relação a  $\mathbf{x}$ :

$$p(\mathbf{y} | \boldsymbol{\theta}) = \int_{\mathbf{x}} \delta(\mathbf{y}_{(i_0)} - \mathbf{x}_{(i_0)}) p_{v_d}(\mathbf{y}_{(i_1)} - \mathbf{x}_{(i_1)}) \delta(\mathbf{y}_{(i_2)} - \mathbf{x}_{(i_2)}) p_x(\mathbf{x}) d\mathbf{x} \quad (6.17)$$

$$= \int_{\mathbf{x}_{(i_1)}} p_{v_d}(\mathbf{y}_{(i_1)} - \mathbf{x}_{(i_1)}) p_x(\mathbf{x}_{(i_0)}, \mathbf{x}_{(i_1)}, \mathbf{x}_{(i_2)}) d\mathbf{x}_{(i_1)}. \quad (6.18)$$

Seguindo os passos sugeridos em [4], obtemos:

$$p(\mathbf{y}|\boldsymbol{\theta}) = \frac{\sigma_e^M e^{\left(-\frac{1}{2\sigma_e^2} E_{\text{MIN}}\right)}}{(2\pi\sigma_e^2)^{\frac{N-P}{2}} |\mathbf{R}_{v_d}|^{1/2} |\boldsymbol{\Phi}|^{1/2}}, \quad (6.19)$$

em que

$$E_{\text{MIN}} = E_0 - \boldsymbol{\Theta}^T \mathbf{x}_{(i_1)}^{\text{MAP}}, \quad (6.20)$$

$$\mathbf{x}_{(i_1)}^{\text{MAP}} = \boldsymbol{\Phi}^{-1} \boldsymbol{\Theta}, \quad (6.21)$$

$$\boldsymbol{\Phi} = \mathbf{A}_{(i_1)}^T \mathbf{A}_{(i_1)} + \sigma_e^2 \mathbf{R}_{v_d}^{-1}, \quad (6.22)$$

$$\boldsymbol{\Theta} = - \left( \mathbf{A}_{(i_1)}^T \begin{bmatrix} \mathbf{A}_{(i_0)} & \mathbf{A}_{(i_2)} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{(i_0)} \\ \mathbf{x}_{(i_2)} \end{bmatrix} - \sigma_e^2 \mathbf{R}_{v_d}^{-1} \mathbf{y}_{(i_1)} \right), \quad (6.23)$$

$$E_0 = \begin{bmatrix} \mathbf{x}_{(i_0)}^T & \mathbf{x}_{(i_2)}^T \end{bmatrix} \begin{bmatrix} \mathbf{A}_{(i_0)}^T \\ \mathbf{A}_{(i_2)}^T \end{bmatrix} \begin{bmatrix} \mathbf{A}_{(i_0)} & \mathbf{A}_{(i_2)} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{(i_0)} \\ \mathbf{x}_{(i_2)} \end{bmatrix} + \sigma_e^2 \mathbf{y}_{(i_1)}^T \mathbf{R}_{v_d}^{-1} \mathbf{y}_{(i_1)}. \quad (6.24)$$

A distribuição *a posteriori* pode ser então calculada pela regra de Bayes.

A amostragem das variáveis relacionadas à descontinuidade inicial,  $n_0$  e  $M$  pode ser feita de forma análoga à apresentada na Seção 5.4.2.2. Para ambas as variáveis usa-se o algoritmo de Metropolis-Hastings, com proposta gaussiana centrada em  $n_0^{(j)}$  para  $n_0$  e proposta discreta para  $M$ , definida por:

$$q(M^*|M^{(j)}) = \begin{cases} 1/2, & M^* = M^{(j)} \pm 1 \\ 0, & \text{em caso contrário,} \end{cases} \quad (6.25)$$

isto é, propõem-se apenas valores uma unidade acima ou uma unidade abaixo do valor atual com igual probabilidade. A geração das amostras é feita de forma similar ao apresentado na Seção 5.4.2.2.

Para amostragem dos parâmetros da cauda,  $\boldsymbol{\theta}_c$ , vamos inicialmente deduzir uma expressão simplificada para sua verossimilhança, a partir da Equação (6.19).

$$p(\mathbf{y}|\boldsymbol{\theta}_c) \propto \exp\left(-\frac{1}{2} \mathbf{z}^T \mathbf{R} \mathbf{z}\right), \quad (6.26)$$

em que

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}_{i_0} \\ \mathbf{x}_{i_2} \end{bmatrix} \quad (6.27)$$

e

$$\mathbf{R} = \frac{1}{\sigma_e^2} \mathbf{A}_{i_1}^T \begin{bmatrix} \mathbf{A}_{i_0} & \mathbf{A}_{i_2} \end{bmatrix} \Phi^{-1} \mathbf{A}_{i_1}^T \begin{bmatrix} \mathbf{A}_{i_0} & \mathbf{A}_{i_2} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{i_0}^T \\ \mathbf{A}_{i_2}^T \end{bmatrix} \begin{bmatrix} \mathbf{A}_{i_0} & \mathbf{A}_{i_2} \end{bmatrix}. \quad (6.28)$$

Pela definição de  $\mathbf{v}_d$ , concluímos que:

$$\mathbf{z} = \begin{bmatrix} \mathbf{x}_{i_0} & \mathbf{x}_{i_2} \end{bmatrix}^T - \begin{bmatrix} \mathbf{v}_{d_{i_0}} & \mathbf{v}_{d_{i_2}} \end{bmatrix}^T. \quad (6.29)$$

Definindo o vetor  $\mathbf{p}$  tal que  $\begin{bmatrix} \mathbf{v}_{d_{i_0}} & \mathbf{v}_{d_{i_2}} \end{bmatrix}^T = V_d \mathbf{p}$ , podemos concluir que a condicional total de  $V_d$  é uma gaussiana de média:

$$\mu_{V_d} = \frac{\mathbf{z}^T \mathbf{R} \mathbf{p}}{\mathbf{p}^T \mathbf{R} \mathbf{p}} \quad (6.30)$$

e variância:

$$\sigma_{V_d}^2 = \frac{1}{\mathbf{p}^T \mathbf{R} \mathbf{p}}. \quad (6.31)$$

Essa dedução é válida para os casos em que não há superposição de pulsos. Se essa condição não for válida, a distribuição de  $V_d$  não é gaussiana e sua amostragem é feita pelo algoritmo de Metropolis-Hastings.

Os demais componentes de  $\boldsymbol{\theta}_c$ , mesmo quando não há superposição, não têm condicionais totais de forma bem conhecida, o que nos leva ao uso do algoritmo de Metropolis-Hastings para obtenção de suas amostras. Para todas as variáveis usamos uma proposta gaussiana centrada no valor inicial e com variância selecionada de acordo com a ordem de grandeza típica de cada parâmetro (ver próxima seção). A razão entre as verossimilhanças é dada por:

$$\frac{p(\mathbf{y}|\boldsymbol{\theta}_c^*)}{p(\mathbf{y}|\boldsymbol{\theta}_c^{(j)})} = -\frac{1}{2} (\mathbf{z}^* - \mathbf{z}^{(j)})^T \mathbf{R} (\mathbf{z}^* - \mathbf{z}^{(j)}), \quad (6.32)$$

em que  $\mathbf{z}^* = \mathbf{z}(\boldsymbol{\theta}^*)$  e  $\mathbf{z}^{(j)} = \mathbf{z}(\boldsymbol{\theta}^{(j)})$ .

A escolha das priors para as variáveis da cauda não é crítica porque, devido ao grande número de dados, a verossimilhança deve prevalecer sobre a priori. Escolhemos priors gaussianas de variância elevada para as variáveis  $V_c$  e  $\phi$ , que podem assumir valores positivos e negativos. As demais variáveis, que só podem assumir valores positivos, são convenientemente descritas por priors do tipo gama inversa.

## 6.4 Resultados

Nesta seção apresentamos os resultados obtidos usando o algoritmo apresentado na seção anterior. Foram feitas simulações em três situações: (A) pulso longo gerado artificialmente (usando o modelo das Equações (6.1) e (6.2)) e inserido num sinal artificial; (B) pulso real retirado de um sinal corrompido e inserido num sinal sem defeitos e (C) sinal originalmente corrompido por pulsos longos. A situação A permite testar o algoritmo propriamente dito (porque permite desconsiderar o erro de modelagem), enquanto que a situação B testa se o modelo proposto é adequado para tratar pulsos reais. A situação C permite verificar se o método proposto é aplicável em casos práticos.

Começamos considerando a situação A. Um pulso longo artificial é gerado com os seguintes parâmetros:  $n_0 = 100$ ,  $M = 10$ ,  $\sigma_{V_d}^2 = 0,1$ ,  $V_t = 0,1$ ,  $\tau_e = 0,07$  s,  $\tau_f = 0,013$  s,  $f_{\min} = 20$  Hz,  $f_{\max} = 60$  Hz,  $\phi = 0$ . Esse pulso é inserido num bloco de sinal de 5000 amostras gerado artificialmente a partir de um modelo AR de ordem 40, cujos coeficientes foram obtidos de um sinal de áudio real usando o método *Least Squares*. O algoritmo roda por 500 iterações, sendo as 100 primeiras descartadas e as 400 restantes usadas para estimação dos parâmetros.

Embora não tenhamos usado técnicas sofisticadas para diagnóstico de convergência, pudemos verificar que algumas variáveis tornam-se bem próximas de seus valores corretos com algumas dezenas de iterações. A Figura 6.2 mostra um exemplo da evolução da variável  $V_c$  com o “tempo”, no qual se percebe que já em 50 iterações o valor de  $V_c$  se aproxima de seu valor correto, para o qual esperávamos que convergisse. Isso ocorre porque a variância da distribuição de  $V_d$  é muito pequena, o mesmo ocorrendo com as variáveis  $n_0$  e  $M$ . Essas variáveis podem ser usadas para diagnóstico de convergência de maneira informal.

A Figura 6.3 mostra o sinal original, o pulso longo gerado artificialmente, o sinal corrompido, o sinal restaurado (obtido pela média das 400 últimas iterações) usando o método proposto, o pulso extraído e os espectros do sinal original, corrompido e restaurado. O gráfico (e) indica que o algoritmo foi capaz de estimar satisfatoriamente o pulso inserido. O gráfico (f) mostra que os componentes de baixa

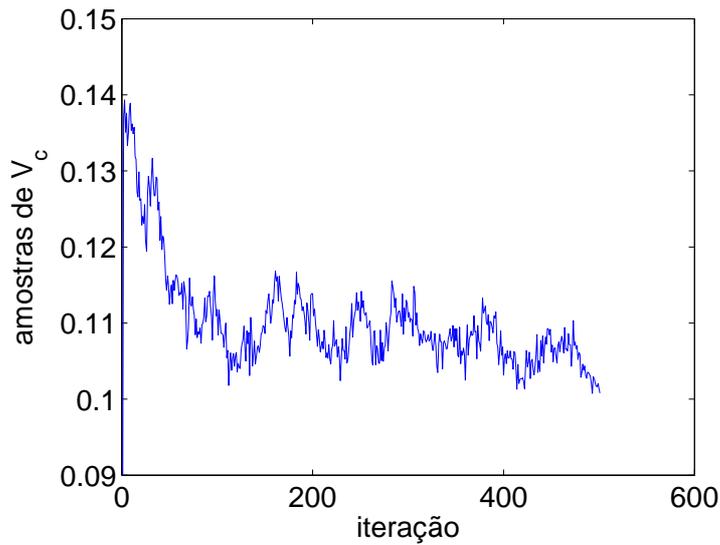


Figura 6.2: Amostras da variável  $V_c$ .

freqüência (relacionados ao pulso longo) foram quase completamente suprimidos do sinal corrompido.

As Figuras 6.4 e 6.5 mostram gráficos relativos às situações B e C, respectivamente. Como esperado, o erro de estimação na situação B é maior que na A, o que pode ser verificado pelo gráfico do espectro do sinal restaurado, que mostra um componente significativo na freqüência de 60 Hz. Isso ocorre porque um pulso real (neste caso e em geral) não tem um formato que possa ser exatamente descrito pelo modelo proposto para o pulso. Quanto mais próximo for o pulso real do formato permitido pelo modelo, menor deverá ser o erro (naturalmente, um erro menor não significa um resultado auditivamente mais agradável).

Na situação C, o modelo AR do sinal foi estimado a partir de um trecho de sinal de 800 amostras imediatamente anterior ao pulso. (Poderíamos estimá-lo iterativamente, da mesma forma que foi feita na restauração de *click*; entretanto, a complexidade tornar-se-ia muito elevada devido ao grande número de amostras do bloco de sinal, que chega a alguns milhares de amostras.) Como consequência, a modelagem não é tão boa quanto nos casos anteriores e, embora não tenhamos um sinal original para comparação, podemos ver na Figura 6.5, que o sinal restaurado, tanto no tempo quanto na freqüência, apresenta resquícios do distúrbio presente no

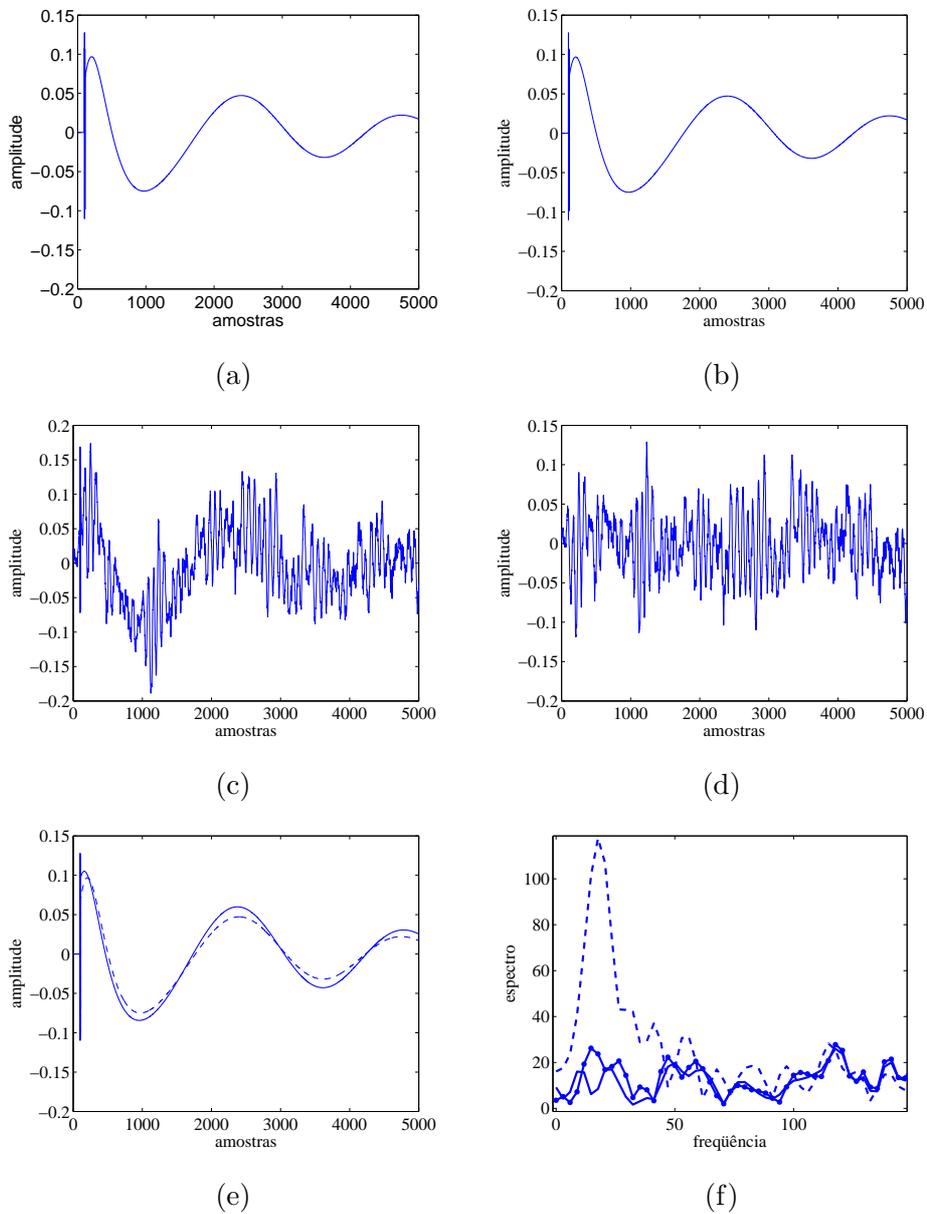


Figura 6.3: Gráficos relativos à situação A. (a) Sinal gerado artificialmente usando modelo AR de ordem 40. (b) Pulso longo artificial. (c) Sinal artificial corrompido. (d) Sinal restaurado. (e) Pulso longo inserido (linha tracejada) e pulso longo extraído (linha contínua). (f) Espectro dos sinais original (linha contínua), corrompido (linha tracejada) e restaurado (linha sólido-pontilhada).

sinal corrompido. (Embora o sinal original pudesse apresentar componentes de baixa frequência, no caso analisado é provável que estes tenham vindo do ruído, porque nas regiões vizinhas ao defeito, tais componentes são baixos).

## 6.5 Considerações sobre a complexidade computacional

O algoritmo proposto como apresentado nas seções anteriores é capaz de conjuntamente realizar tanto a localização quanto a restauração do sinal, sendo por isso mais geral que os algoritmos existentes, que fazem a localização e restauração separadamente. Além disso, os parâmetros do modelo AR do sinal podem, em princípio, ser tratados como *nuisance variables* e marginalizadas numericamente, evitando o uso de estimativas pouco confiáveis. Esse procedimento, todavia, levaria a um algoritmo de complexidade inaceitavelmente alta para a aplicação prática, considerando o poder computacional disponível atualmente. Uma possível solução para redução da complexidade é usar um número menor de amostras de sinal para amostragem dos coeficientes do modelo do sinal.

Caso a localização e os parâmetros do modelo AR possam ser tratados como conhecidos (isto é, fazendo-se  $n_0$ ,  $M$  e  $\mathbf{a}$  constantes no algoritmo), a complexidade é drasticamente reduzida. Nesse caso, a matriz  $\mathbf{R}$  é fixa durante a execução do algoritmo e só precisa ser calculada uma vez. A amostragem de cada variável envolve apenas multiplicações de matrizes (nenhuma inversão de matrizes precisa ser feita). A complexidade torna-se então comparável à do método de separação de processos AR, que não é iterativo mas envolve inversão de matrizes de alta dimensão. Ainda assim, sua complexidade é bem maior que o do método do pulso-padrão e o baseado em filtragem TPSW.

## 6.6 Conclusões

Esse capítulo apresentou uma técnica para remoção de pulsos longos de sinais de áudio usando um método baseado no algoritmo de Metropolis-Hastings combi-

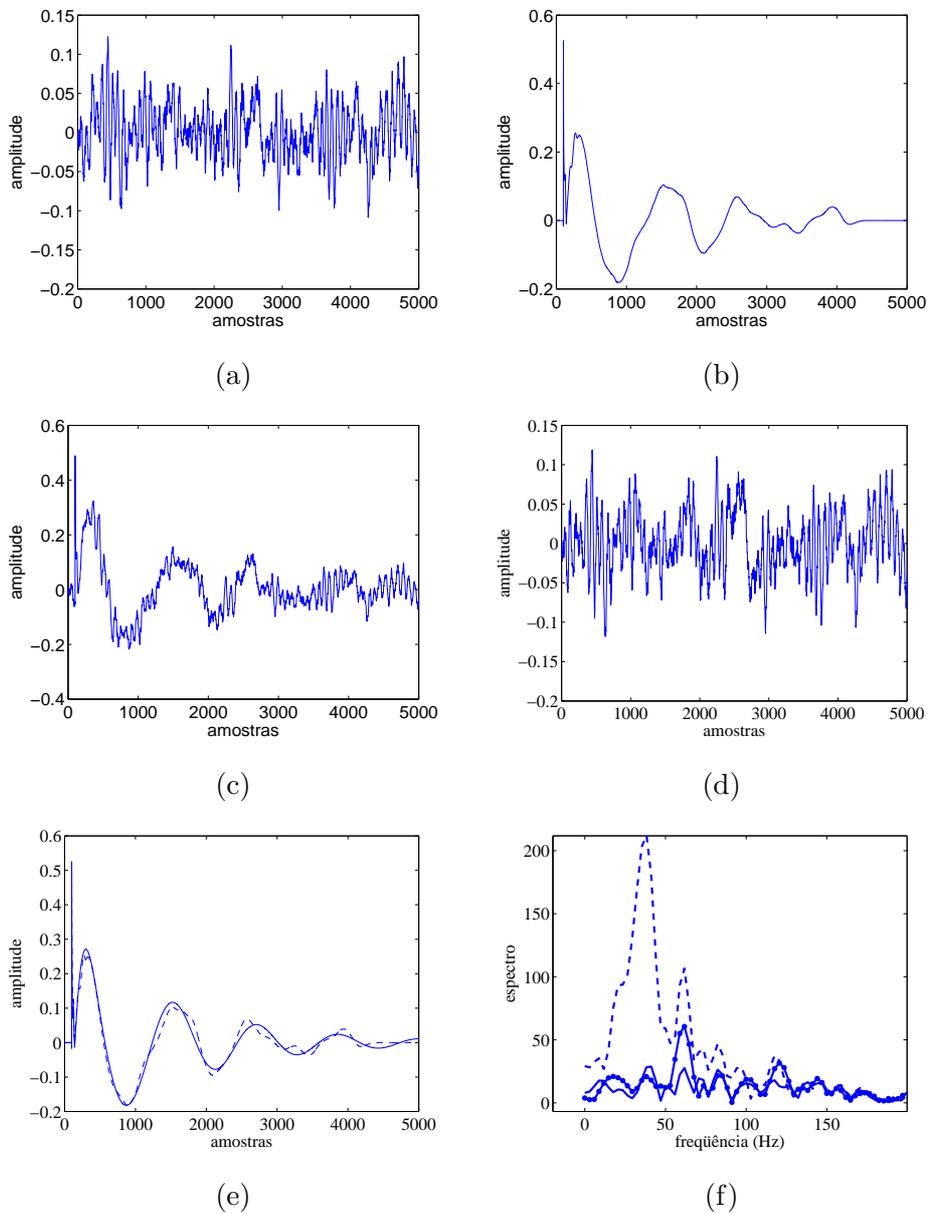


Figura 6.4: Gráficos relativos à situação B. (a) Sinal real sem defeitos. (b) Pulso longo real. (c) Sinal artificial corrompido por pulso longo real. (d) Sinal restaurado. (e) Pulso longo inserido (linha tracejada) e pulso longo extraído (linha contínua). (f) Espectro dos sinais original (linha contínua), corrompido (linha tracejada) e restaurado (linha sólido-pontilhada).

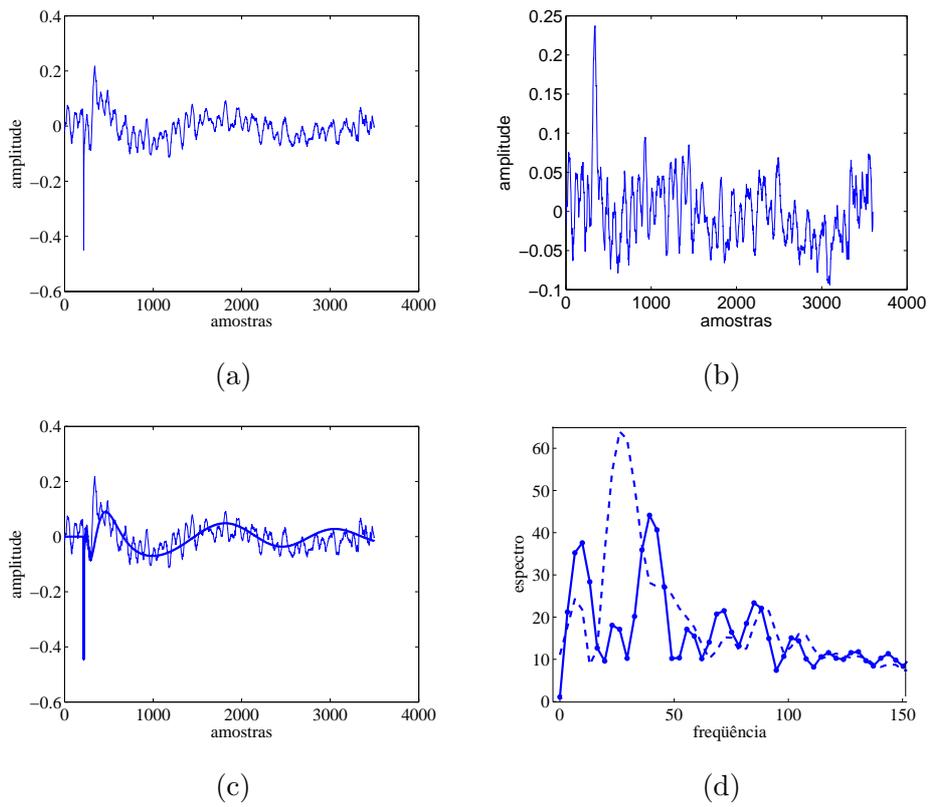


Figura 6.5: Gráficos relativos à situação C. (a) Sinal real corrompido por pulso longo. (b) Sinal restaurado. (c) Sinal corrompido e pulso extraído (mais suave). (d) Espectro dos sinais corrompido (linha tracejada) e restaurado (linha contínua).

nado com amostragem de Gibbs. É associado ao distúrbio um modelo que consiste na concatenação de ruído branco gaussiano e uma cauda com formato de senóide amortecida com frequência variável, amortecida. Cada variável desse modelo, assim como o sinal restaurado, foi amostrada usando o algoritmo de Metropolis-Hastings ou amostragem de Gibbs, dependendo do caso.

Os resultados, ainda preliminares, indicam que o método é eficaz para remoção de pulsos longos quando esses possuem formato razoavelmente parecido com o modelo proposto. Isso torna o método proposto menos geral que o baseado em separação de processos AR e o método baseado em filtragem TPSW. Como trabalho futuro, pretendemos elaborar modelos mais realísticos para pulsos longos, mantendo a estrutura do algoritmo.

# Capítulo 7

## Conclusões e Trabalhos Futuros

Esta dissertação tratou do problema de remoção de dois dos tipos de defeitos mais comuns presentes em gravações de áudio antigas: *clicks* e pulsos longos. Para ambos os problemas, foram empregados métodos bayesianos que permitiram levar em conta o conhecimento *a priori* sobre esses tipos de defeitos. Foram empregadas técnicas numéricas baseadas em Monte Carlo via Cadeias de Markov (MCMC) para se estimar as variáveis de interesse.

Para o problema de remoção de *clicks*, propusemos um novo modelo para descrição do ruído, que trata cada *click* individualmente. Por esse modelo, um *click* é modelado como um pulso que segue uma distribuição cuja variância decresce exponencialmente. Sua descrição estatística é parametrizada pelas variáveis de localização, duração, amplitude e taxa de decaimento. A aplicação dos métodos bayesianos para o problema de restauração usando esse novo modelo nos conduziu à elaboração de um algoritmo para remoção de *clicks* formado por uma mistura do algoritmo de Metropolis-Hastings com saltos reversíveis com o amostrador de Gibbs. O algoritmo proposto apresenta uma complexidade por iteração que pode ser menor do que o algoritmo de Godsill baseado no amostrador de Gibbs [4, 27], dependendo do número de *clicks* existentes em um bloco.

A remoção de pulsos longos foi abordada de forma similar. Propusemos um modelo para pulsos longos formado pela concatenação de uma descontinuidade inicial com um transitório de baixa frequência (cauda). A descontinuidade foi modelada por um pulso (ruído branco) com localização, duração e variância (constante) des-

conhecidas; a cauda foi descrita por uma senóide exponencialmente amortecida e de frequência decrescente. Em relação aos métodos bayesianos encontrados na literatura, o método proposto tem a vantagem de realizar conjuntamente de forma ótima a detecção e remoção do defeito. Uma limitação do método é a baixa qualidade (aferida pela análise visual do sinal) do resultado quando os pulsos têm um formato muito diferente do formato permitido pelo modelo.

As continuações naturais deste trabalho incluem os seguintes tópicos:

- Simplificação das expressões da probabilidade condicional da localização buscando reduzir a complexidade computacional.
- Implementação do algoritmo de Godsill [4, 27] e comparação mais sistemática do número de iterações necessárias para a convergência.
- Avaliação da qualidade das estimativas realizadas pelo algoritmo através do cálculo de regiões de confiança.
- Estudo de técnicas de aceleração de convergência para algoritmos MCMC e sua aplicação aos algoritmos propostos nesta dissertação.
- Verificação do comportamento do algoritmo de remoção de pulsos longos para o caso de superposição de pulsos.
- Elaboração de modelos mais realistas para pulsos longos que permitam o tratamento de defeitos de formatos mais variados, mantendo-se a estrutura do algoritmo—em particular, associando um modelo AR de baixa ordem à cauda.
- Aplicação de técnicas bayesianas para outros problemas de restauração de áudio ainda não solucionados na literatura, em particular os de talhas de muita longa duração.
- Testes de audição para avaliar os resultados.

# Referências Bibliográficas

- [1] ESQUEF, P. A. A., BISCAINHO, L. W., “DSP techniques for sound enhancement of old recordings”. In: *Advances in Audio and Speech Signal Processing*, Hector Perez-Meana, pp. 93–128, 2007.
- [2] VASEGHI, S. V., *Algorithms for Restoration of Archived Gramophone Recordings*. PhD thesis, Department of Engineering, University of Cambridge, Cambridge, UK, 1988.
- [3] BOLL, S. F., “Suppression of acoustic noise in speech using spectral subtraction”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, v. 27, n. 2, pp. 113–120, 1979.
- [4] GODSILL, S. J., RAYNER, P. J. W., *Digital Audio Restoration - A Statistical Model Based Approach*. Cambridge, UK, Cambridge University Press, 1998.
- [5] GODSILL, S. J., *The Restoration of Degraded Audio Signals*. PhD thesis, Department of Engineering, University of Cambridge, Cambridge, UK, 1993.
- [6] BERNARDO, J. M., SMITH, A. F. M., *Bayesian Theory*. New Jersey, USA, John Wiley & Sons, 2007.
- [7] BOLSTAD, W. M., *Introduction to Bayesian Statistics*. New Jersey, USA, John Wiley & Sons, 2004.
- [8] MIGON, H. S., GAMERMAN, D., *Statistical Inference: An Integrated Approach*. Londres, UK, Hodder Arnold, 1999.
- [9] HAJEK, A., “Interpretations of Probability”, 2007, <http://plato.stanford.edu/entries/probability-interpret>, 03/2008.

- [10] JAYNES, E., *What's Wrong With Bayesian Methods*, Technical report, St. John's College and Cavendish Laboratory, Cambridge, UK, 1984.
- [11] R. O. DUDA, P. E. HART, D. G. S., *Pattern Classification*. New Jersey, USA, Jhon Wiley & Sons, 2001.
- [12] KAY, S. M., *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. New Jersey, USA, Prentice Hall, 1993.
- [13] BOYD, S., *Convex Optimization*. Cambridge, UK, Cambridge University Press, 2004.
- [14] PAPOULIS, A., *Probability, Random Variables, and Stochastic Process*. New York, USA, McGraw-Hill, 1984.
- [15] JAYNES, E., "Prior probabilities", *IEEE Transactions on Systems Science and Cybernetics*, v. 4, n. 3, pp. 227–241, 1978.
- [16] BERNARDO, J. M., "Reference posterior distributions for Bayesian inference", *Journal of the Royal Statistical Society*, v. 41, n. 2, pp. 113–147, 1979.
- [17] GREEN, P., "Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination", *Biometrika*, v. 82, n. 4, pp. 711–732, 1995.
- [18] ANDRIEU, C., FREITAS, N. D., DOUCET, A., JORDAN, M., "An introduction to MCMC for machine learning". In: *Machine Learning*, Kluwer Academic Publishers, pp. 5–43, 2003.
- [19] NEAL, R., *Probabilistic Inference Using Markov Chain Monte Carlo Methods*, Technical Report CRG-TR-93-1, University of Toronto, Department of Computer Science, Toronto, Canadá, 1993.
- [20] GAMERMAN, D., LOPES, H., *Markov Chain Monte Carlo - Stochastic Simulation for Bayesian Inference*. Boca Raton, USA, Chapman & Hall/CRC, 2006.
- [21] ROBERT, C., CASELLA, G., *Monte Carlo Statistical Methods*. New York, USA, Springer, 2004.

- [22] GEMAN, S., GEMAN, D., “Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 6, n. 6, pp. 721–741, 1984.
- [23] GODSILL, S., “On the relationship between Markov Chain Monte Carlo methods for model uncertainty”, *Journal of Computational and Graphical Statistics*, v. 10, n. 2, pp. 1–19, 2001.
- [24] VASEGHI, S. V., FRAYLING-CORK, R., “Restoration of old gramophone recordings”, *Journal of the Audio Engineering Society*, v. 40, n. 10, pp. 791–801, 1992.
- [25] GODSILL, S. J., RAYNER, P. J. W., “A Bayesian approach to the detection and correction of bursts of errors in audio signals”. In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 261–264, San Francisco, CA, USA, March 1992.
- [26] A. P. DEMPSTER, N. M. L., RUBIN, D. B., “Maximum likelihood from incomplete data via the EM algorithm”, *Journal of the Royal Statistical Society, Series B*, v. 39, n. 1, pp. 1–38, 1977.
- [27] GODSILL, S. J., RAYNER, P. J., “Robust reconstruction and analysis of autoregressive signals in impulsive noise using the Gibbs sampler”, *IEEE Transactions on Speech and Audio Processing*, v. 6, n. 5, pp. 352–372, 1998.
- [28] VASEGHI, S. V., *Advanced Digital Signal Processing and Noise Reduction*. New York, USA, John Wiley & Sons, 2000.
- [29] NERO, A. D., *Otimização: Aspectos Teóricos e Métodos Numéricos*. Rio de Janeiro, Brasil, Fornecido pelo autor, 2005.
- [30] ESQUEF, P. A. A., *Restauração de Sinais de Áudio Degradados por Ruído Impulsivo*. Tese de mestrado, COPPE/UFRJ Laboratório de Processamento de Sinais, Rio de Janeiro, RJ, Brasil, 1999.

- [31] ESQUEF, P. A. A., BISCAINHO, L. W. P., DINIZ, P. S. R., FREELAND, F. P., “A double-threshold-based approach to impulsive noise detection in audio signals”. In: *Proceedings of the X European Signal Processing Conference (EUSIPCO)*, pp. 2061–2064, Tampere, Finland, September 2000.
- [32] ANDRIEU, C., DOUCET, A., “Joint Bayesian model selection and estimation of noisy sinusoids via reversible jump MCMC”, *IEEE Transactions on Signal Processing*, v. 47, n. 10, pp. 2667–2676, 1999.
- [33] ESQUEF, P. A. A., BISCAINHO, L. W., VÄLIMÄKI, V., “An efficient algorithm for the restoration of audio signals corrupted with low-frequency pulses”, *Journal of the Audio Engineering Society*, v. 51, n. 6, pp. 502–517, 2003.