

CLAUDIO JOSÉ BORDIN JÚNIOR

**MÉTODOS ESTATÍSTICOS PARA EQUALIZAÇÃO DE CANAIS DE
COMUNICAÇÃO**

Tese apresentada à Escola
Politécnica da Universidade de
São Paulo para obtenção do
título de Doutor em Engenharia.

São Paulo
2006

CLAUDIO JOSÉ BORDIN JÚNIOR

**MÉTODOS ESTATÍSTICOS PARA EQUALIZAÇÃO DE CANAIS DE
COMUNICAÇÃO**

Tese apresentada à Escola
Politécnica da Universidade de
São Paulo para obtenção do
título de Doutor em Engenharia.

Área de Concentração:
Sistemas Eletrônicos

Orientador:
Prof. Dr. Luiz Antonio Baccalá

São Paulo
2006

Bordin Jr., Claudio José

Métodos Estatísticos para Equalização de Canais de Comunicação.
São Paulo, 2006. 132p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Telecomunicações e Controle.

1. Equalização Cega 2. Estimação Bayesiana 3. Filtros de Partículas
I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Telecomunicações e Controle II. t

Agradecimentos

Ao meu orientador, Prof. Dr. Luiz Antonio Baccalá, pela constante ajuda e estímulo, indispensáveis para a realização deste trabalho.

À minha família, pelo apoio e compreensão.

Aos Professores Oswaldo Luiz do Valle Costa e Marcelo Gomes da Silva Bruno pela participação na banca examinadora de qualificação.

À FAPESP, pelo apoio financeiro¹.

A todos os demais que, direta ou indiretamente, contribuíram para a realização deste trabalho.

¹Este trabalho contou com o apoio financeiro da FAPESP sob a forma de bolsa de doutoramento (processo 02/11457-7).

Sumário

1	Introdução	1
1.1	Formulação do Problema	2
1.2	Equalização Bayesiana	4
1.3	Organização do texto	5
1.4	Principais Contribuições	6
1.5	Publicações	7
2	Filtros de Partículas	8
2.1	Introdução	8
2.2	Degenerescência e Reamostragem	11
2.3	Escolha da função de importância	14
2.4	Algoritmo SIR	16
2.5	Filtros de Partículas para Modelos com Parâmetros Desconhecidos . .	19
2.6	Exemplos de aplicação de filtros de partículas	23
3	Filtro de Partículas para Equalização Cega	29
3.1	Modelo de sinal, hipóteses e objetivos	29
3.2	Filtro de Partículas Rao-Blackwellizado	30
3.3	Métodos para a resolução do problema da ambigüidade de fase	33
3.4	Métodos de Evolução Artificial de Parâmetros	36
3.5	Simulações	38
3.6	Comentários	50
4	Equalização e Decodificação Conjuntas empregando Filtros de Partículas	53
4.1	Equalização e Decodificação Conjuntas: Fundamentos	54
4.2	Equalização e Decodificação Conjuntas: Códigos de Bloco	55
4.3	Equalização e Decodificação conjunta: Códigos Convolucionais	61
4.4	Simulações	63
4.5	Comentários	73
5	Conclusões	76
5.1	Sugestões para trabalhos futuros	78

A	Equalização de Canais de Ordem Desconhecida e Variantes no Tempo Empregando Filtros de Partículas	86
A.1	Equalização Cega de Canais de Ordem Desconhecida	86
A.2	Equalização de Canais Variantes no Tempo	94
B	Equalização Cega empregando o Algoritmo EM	99
B.1	O algoritmo EM	99
B.2	Equalização Cega empregando o algoritmo EM	101
B.3	Algoritmo Melhorado	105
B.4	Comentários	106
C	Estimação através de Métodos MCMC	109
C.1	Métodos MCMC e Cadeias de Markov - Definições Básicas	109
C.2	Algoritmo de Metropolis-Hastings	111
C.3	Amostrador de Gibbs	112
C.4	Equalização Cega empregando o Amostrador de Gibbs	113
C.5	Comentários	119
D	Densidades empregadas pelos Filtros de Partículas Rao-Blackwellizados	121
E	Amostragem Sequencial - Fundamentos	125
E.1	Aproximação de uma densidade de probabilidade por medidas discretas	125
E.2	Amostragem por importância	126
E.3	Amostragem Sequencial	127
F	O Algoritmo BCJR	129

Lista de Figuras

2.1	Gráfico de uma realização de $X_{0:n}$ gerado pelo modelo dado em (2.42) e de suas estimativas através de um filtro de Kalman (KF) e de um filtro de partículas (PF).	26
2.2	Gráfico de uma realização de $X_{0:n}$ gerado pelo modelo dado em (2.54) e de suas estimativas através de um filtro de Kalman estendido (EKF) e de um filtro de partículas (PF).	28
3.1	Desempenho médio ao longo de 250 realizações independentes dos algoritmos para equalização cega baseados em filtros de partículas descritos neste capítulo, num sistema BPSK (empregando $N = 100$ partículas) e do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR). Os algoritmos estocásticos e baseados em evolução artificial (EA) empregam codificação diferencial.	42
3.2	Desempenho médio ao longo de 250 realizações independentes dos algoritmos para equalização cega baseados em filtros de partículas descritos neste capítulo, num sistema BPSK (empregando $N = 300$ partículas) e do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR).	42
3.3	Desempenho médio ao longo de 250 realizações independentes dos algoritmos para equalização cega baseados em filtros de partículas descritos neste capítulo, num sistema BPSK (empregando $N = 500$ partículas) e do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR).	43
3.4	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e dos algoritmos para equalização cega baseados em filtros de partículas descritos neste capítulo, num sistema BPSK, empregando $N = 300$ partículas e suavização (com atraso de 5 amostras), em função da relação sinal-ruído (SNR).	43
3.5	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e dos algoritmos para equalização cega baseados em filtros de partículas descritos neste capítulo, num sistema BPSK, empregando $N = 300$ partículas e suavização (com atraso de 10 amostras), em função da relação sinal-ruído (SNR).	44

3.6	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR), do algoritmo determinístico (utilizando $N = 100$ partículas) e dos algoritmos para equalização cega baseados em evolução artificial ($P = 1$), num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR) e do número de partículas N	44
3.7	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR), do algoritmo determinístico (utilizando $N = 100$ partículas) e dos algoritmos para equalização cega baseados em evolução artificial ($P = 5$), num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR) e do número de partículas N	45
3.8	Efeito da escolha da variância δ^2 do núcleo de transição $p(h_n h_{n-1})$ sobre o desempenho médio ao longo de 250 realizações independentes dos algoritmos para equalização cega baseados em evolução artificial ($P = 1$), utilizando $N = 1000$ partículas e suavização (com atraso de 10 amostras) em função da relação sinal-ruído (SNR), comparado com o desempenho do equalizador ótimo MAP (BCJR).	45
3.9	Efeito da escolha da variância δ^2 do núcleo de transição $p(h_n h_{n-1})$ sobre o desempenho médio ao longo de 250 realizações independentes dos algoritmos para equalização cega baseados em evolução artificial ($P = 5$), utilizando $N = 1000$ partículas e suavização (com atraso de 10 amostras) em função da relação sinal-ruído (SNR), comparado com o desempenho do equalizador ótimo MAP (BCJR).	46
3.10	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado no filtro de partículas estocástico empregando a função de importância ótima, num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR), do limiar de reamostragem e do número de partículas empregado.	46
3.11	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado no filtros de partículas estocástico empregando a função de importância <i>a priori</i> , num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR), do limiar de reamostragem e do número de partículas empregado.	47
3.12	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e de algoritmos para equalização cega baseados em filtros de partículas estocásticos (utilizando $N = 300$ partículas e suavização (10 amostras)), empregando a função de importância ótima e <i>a priori</i> , num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR) e do limiar de reamostragem.	47

3.13	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado em evolução artificial, num sistema BPSK com codificação diferencial sob ruído de envoltória Weibull (real, com $\alpha = 1, 1$), em função da relação sinal-ruído (SNR). As linhas cheias mostram os resultados para $N = 500$ partículas, enquanto as linhas tracejadas, resultados para $N = 1000$ partículas.	49
3.14	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado em evolução artificial, num sistema BPSK com codificação diferencial sob ruído de envoltória Weibull (real, com $\alpha = 1, 1$), em função da relação sinal-ruído (SNR), empregando suavização (10 amostras). As linhas cheias mostram os resultados para $N = 500$ partículas, enquanto as linhas tracejadas, resultados para $N = 1000$ partículas.	50
3.15	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado em evolução artificial, num sistema QPSK com codificação diferencial (Gray) sob ruído de envoltória Weibull (complexo, com $\alpha = 1, 1$), em função da relação sinal-ruído (SNR). As linhas cheias mostram os resultados para $N = 500$ partículas, enquanto as linhas tracejadas, resultados para $N = 1000$ partículas.	51
3.16	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado em evolução artificial, num sistema QPSK com codificação diferencial (Gray) sob ruído de envoltória Weibull (complexo, com $\alpha = 1, 1$), em função da relação sinal-ruído (SNR), empregando suavização (10 amostras). As linhas cheias mostram os resultados para $N = 500$ partículas, enquanto as linhas tracejadas, resultados para $N = 1000$ partículas.	52
4.1	Modelo de sinal adotado neste capítulo.	54
4.2	Desempenho médio ao longo de 250 realizações independentes de algoritmos para equalização e decodificação conjunta empregando o filtro de partículas estocástico com função de importância ótima (STO) e o filtro determinístico (DET), e de um esquema treinado (MAP) descrito no texto, em função da SNR e do atraso de suavização d , para um sistema empregando o código de blocos B_1 descrito na Tab. 4.2. . . .	59
4.3	Desempenho médio ao longo de 250 realizações independentes de algoritmos para equalização e decodificação conjunta empregando o filtro de partículas estocástico com função de importância ótima (STO) e o filtro determinístico (DET), e de um esquema treinado (MAP) descrito no texto, em função da SNR e do atraso de suavização d , para um sistema empregando o código de blocos B_2 descrito na Tab. 4.2. . . .	60

4.4	Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (5, 7, 2). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.	65
4.5	Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (17, 12, 04). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.	66
4.6	Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (63, 44, 17). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.	67
4.7	Sistema de comunicação empregando codificação diferencial em cascata com um codificador convolucional.	68
4.8	Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (5, 7). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.	69

4.9	Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (5, 7, 7). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.	70
4.10	Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (53, 75). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.	71
4.11	Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (47, 53, 75). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.	72
4.12	Desempenho dos métodos para equalização e decodificação conjunta baseados em filtros de partículas determinístico (DET) e estocástico (STO) em função da SNR, empregando os métodos de reamostragem multinomial, residual e determinístico (DET-DET), e do método alternativo MAP descrito no texto. Os filtros de partículas estocásticos empregam a função de importância ótima.	74
4.13	Desempenho dos métodos para equalização e decodificação conjunta baseados em filtros de partículas determinístico (DET) e estocástico (STO) em função da SNR, empregando os métodos de reamostragem multinomial, residual e determinístico (DET-DET), empregando suavização (10 amostras) e do método alternativo MAP descrito no texto. Os filtros de partículas estocásticos empregam a função de importância ótima.	75
4.14	Desempenho do algoritmo para equalização e detecção conjunta determinístico (empregando reamostragem determinística e suavização de 10 amostras) e do método alternativo MAP descrito no texto em função do número de partículas empregado e da SNR.	75

5.1	Comparação do desempenho atingido por diversos métodos para equalização cega e pelo algoritmo de Viterbi em função da relação sinal-ruído para o canal $h = [0,41 \ 0,82 \ 0,41]^T$. A ordem do canal e a variância do ruído aditivo (gaussiano e real) foram assumidas conhecidas; os algoritmos EM e Gibbs (amostrador conjunto) operam sobre blocos de 250 amostras do sinal recebido, descartando as 25 de cada extremidade para fins de cálculo de taxa de erro. O algoritmo baseado no filtro de partículas (estocástico - com função de importância ótima) também opera sobre 250 amostras, descartando as 100 primeiras, e emprega suavização (5 amostras). O sinal transmitido é binário e emprega codificação diferencial.	79
A.1	Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega em canal de ordem desconhecida de Miguez et al. [1], para ordens possíveis $1 \leq \hat{L} \leq 5$ e $2 \leq \hat{L} \leq 4$, comparado com o desempenho do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado em filtros de partículas estocásticos com a função de importância ótima (utilizando $N = 300$ partículas e suavização (10 amostras)), num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR).	89
A.2	Histograma de 250 amostras do estimador MAP da ordem do canal provida pelo algoritmo de Miguez et al. com $1 \leq \hat{L} \leq 5$ em função da relação sinal-ruído (SNR). O canal empregado tem ordem $L = 3$	90
A.3	Histograma de 250 amostras do estimador MAP da ordem do canal provida pelo algoritmo de Miguez et al. com $2 \leq \hat{L} \leq 4$ em função da relação sinal-ruído (SNR). O canal empregado tem ordem $L = 3$	90
A.4	Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega baseado no princípio da evolução artificial (com $P = 1$, $N = 300$ partículas e suavização de 5 amostras), comparado com o desempenho do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR) e da ordem assumida para o canal.	91
A.5	Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega baseado no princípio da evolução artificial (com $P = 5$, $N = 300$ partículas e suavização de 5 amostras), comparado com o desempenho do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR) e da ordem assumida para o canal.	91
A.6	Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega baseado no filtro de partículas determinístico ($N = 300$ partículas e suavização de 5 amostras), comparado com o desempenho do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR) e da ordem assumida para o canal.	92

A.7	Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega baseado no filtro de partículas estocástico (utilizando a função de importância ótima, $N = 300$ partículas e suavização de 5 amostras), comparado com o desempenho do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR) e da ordem assumida para o canal.	92
A.8	Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega baseado no filtro de partículas estocástico (utilizando a função de importância <i>a priori</i> , $N = 300$ partículas e suavização de 5 amostras), comparado com o desempenho do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR) e da ordem assumida para o canal.	93
A.9	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (linha cheia) e do algoritmos para equalização cega baseado no filtros de partículas determinístico para canais invariantes do tempo (linha tracejada)(utilizando $N = 300$ partículas e suavização (5 amostras)), em função da relação sinal-ruído (SNR) e do parâmetro δ (indicado com múltiplos do valor $\delta = \sqrt{0,0125}$).	96
A.10	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (linha cheia) e do algoritmos para equalização cega baseado no filtros de partículas determinístico (linha tracejada)(utilizando $N = 300$ partículas e suavização (5 amostras)), em função da relação sinal-ruído (SNR) e do parâmetro δ (indicado com múltiplos do valor $\delta = \sqrt{0,0125}$).	97
A.11	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (linha cheia) e do algoritmos para equalização cega baseado no filtros de partículas estocástico (linha tracejada)(utilizando a função de importância ótima, $N = 300$ partículas e suavização (5 amostras)), em função da relação sinal-ruído (SNR) e do parâmetro δ (indicado com múltiplos do valor $\delta = \sqrt{0,0125}$).	97
A.12	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (linha cheia) e do algoritmos para equalização cega baseado no princípio da evolução artificial (linha tracejada) (com $P = 5$, $N = 300$ partículas e suavização (5 amostras)), em função da relação sinal-ruído (SNR) e do parâmetro δ (indicado com múltiplos do valor $\delta = \sqrt{0,0125}$).	98
A.13	Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (linha cheia) e do algoritmos para equalização cega baseado no princípio da evolução artificial (linha tracejada)(com $P = 5$, $N = 1000$ partículas e suavização (5 amostras)), em função da relação sinal-ruído (SNR) e do parâmetro δ (indicado com múltiplos do valor $\delta = \sqrt{0,0125}$).	98
B.1	Desempenho médio ao longo de 100 realizações independentes do algoritmo para equalização cega descrito na Sec. B.2 em função da relação sinal-ruído(SNR) e o número de iterações.	105

B.2	Desempenho médio ao longo de 100 realizações independentes da versão modificada do algoritmo para equalização cega descrito na Sec. B.2.1 ($\delta(i) = 10^{0,1i}$) em função da relação sinal-ruído(SNR) e o número de iterações.	107
B.3	Desempenho médio ao longo de 100 realizações independentes da versão modificada do algoritmo para equalização cega descrito na Sec. B.2.1 ($\delta(i) = 10^{0,05i}$) em função da relação sinal-ruído(SNR) e o número de iterações.	108
C.1	Desempenho médio ao longo de 100 realizações independentes do algoritmo para equalização cega baseado no amostrador de Gibbs conjunto descrito na Sec. C.4.3 em função da relação sinal-ruído (SNR) e do número de iterações da Cadeia de Markov.	119

Lista de Tabelas

2.1	Descrição de um Filtro <i>Bootstrap</i>	14
2.2	Algoritmo SIR	16
2.3	Algoritmo SIR Determinístico.	17
3.1	Equalizador cego baseado no algoritmo SIR estocástico, empregando a função de importância ótima e codificação diferencial.	34
3.2	Equalizador cego baseado no algoritmo SIR estocástico, empregando a função de importância ótima e pivotamento.	36
3.3	Equalizador cego baseado no algoritmo SIR estocástico, empregando a função de importância ótima e codificação diferencial implícita.	37
3.4	Algoritmo para equalização cega sob ruído com distribuição genérica baseado no método de evolução artificial utilizando função de importância <i>a priori</i>	38
3.5	Algoritmo para equalização cega sob ruído com distribuição genérica baseado no método de evolução artificial utilizando função de importância modificada.	39
3.6	Desempenho dos algoritmos testados ($N = 300$), quantificado pela perda de desempenho (em dB) em relação ao equalizador ótimo (BCJR) sob relações sinal-ruído de 6 e 20 dB, e complexidade computacional dos métodos.	52
4.1	Algoritmo para equalização e decodificação conjunta em sistemas empregando códigos de bloco sistemáticos baseado no algoritmo SIR estocástico, empregando a função de importância ótima.	56
4.2	Códigos de bloco empregados nas simulações da Seção 4.2.1: cada matriz coleciona em suas linhas as palavras de códigos válidas.	59
4.3	Algoritmo para equalização e decodificação conjunta em sistemas empregando códigos convolucionais baseado no algoritmo SIR estocástico, empregando a função de importância ótima.	63
5.1	Complexidade computacional de diversos algoritmos para equalização cega, em função da ordem do canal a equalizar L , do número de amostras de sinal tratado A , no número de partículas ou iterações N , sendo “ o ” uma constante tal que $1 \leq o \leq 10$	78
B.1	Algoritmo EM	100
B.2	Algoritmo Para Equalização Cega baseado no Algoritmo EM	104

C.1	Algoritmo de Metropolis-Hastings	111
C.2	Amostrador de Gibbs	112

Lista de Símbolos

A^H	Conjugado transposto da matriz (ou vetor) A
A^T	Transposto da matriz (ou vetor) A
L	Ordem do canal de transmissão
N	Número total de partículas empregado (em filtros de partículas)
T	Período de símbolo
b_n	Bit transmitido (caso o sinal transmitido seja codificado)
x_n	Símbolo transmitido
y_n	Sinal observado
$q_{a:b}$	Seqüência $\{q_a, \dots, q_b\}$
$\{q_i\}_{i=0}^N$	Conjunto formado pelos elementos q_0, \dots, q_N
$x \sim \pi(x)$	Indica que x é uma amostra da distribuição ² $\pi(x)$
$\mathcal{N}(x \mu, \sigma^2)$	Densidade de probabilidade gaussiana real na variável x , com média μ e variância σ^2 .
$\mathcal{N}_C(x \mu, \sigma^2)$	Densidade de probabilidade gaussiana complexa circular na variável x , com média μ e variância σ^2 .
$\mathcal{B}_L(A)$	Vetor com L variáveis Bernoulli independentes, que podem assumir qualquer um dos valores do conjunto A equiprovavelmente.
$\mathcal{I}\{A\}$	Função indicador de ocorrência do evento A (1 se A ocorre, e 0 caso contrário).
$\delta(x)$	Função delta de Dirac.
$\mathbb{P}(A)$	Probabilidade do evento A .
$p(x)$	Densidade de probabilidade da variável x , caso esta seja contínua, ou probabilidade da variável x assumir um determinado valor, caso seja discreta.
$p_y(x)$	Densidade de probabilidade da variável y avaliada no ponto x .
$ A $	módulo do número A ou cardinalidade do conjunto A .
$\ A\ $	Norma 2 da matriz ou vetor A .
E_π	Esperança em relação à medida π .

²Empregamos o mesmo símbolo caso a função $\pi(x)$ não seja normalizada.

Lista de Abreviaturas³

AR(n)	Sinal ou modelo autoregressivo de ordem n
BER	Taxa (Média) de Erro de Bit
BPSK	<i>Binary Phase Shift Keying</i>
EA	Evolução Artificial
EM	<i>Expectation-Maximization</i>
IID	Independentes e Identicamente Distribuídos
ISI	Interferência Intersimbólica
LMMSE	<i>Linear Minimum Mean Square Error Estimator</i>
LS	Mínimos Quadrados
MAP	Máximo <i>a Posteriori</i>
MCMC	<i>Monte Carlo Markov Chain</i>
ML	Máxima Verossimilhança
MMSE	<i>Minimum Mean Square Error Estimator</i>
QPSK	<i>Quadrature Phase Shift Keying</i>
SNR	Relação Sinal-Ruído
v.a.	Variável ou vetor aleatório

³Procuramos sempre que possível empregar as abreviações mais freqüentemente encontradas na literatura especializada, que, por esta razão, foram mantidas na língua inglesa.

Resumo

Nesta tese analisamos e propomos métodos para a equalização não-treinada (cega) de canais de comunicação lineares FIR baseados em filtros de partículas, que são técnicas recursivas para a solução Bayesiana de problemas de filtragem estocástica. Iniciamos propondo novos métodos para equalização sob ruído gaussiano que prescindem do uso de codificação diferencial, ao contrário dos métodos existentes. Empregando técnicas de evolução artificial de parâmetros, estendemos estes resultados para o caso de ruído aditivo com distribuição não-gaussiana. Em seguida, desenvolvemos novos métodos baseados nos mesmos princípios para equalizar e decodificar conjuntamente sistemas de comunicação que empregam códigos convolucionais ou em bloco. Através de simulações numéricas, observamos que os algoritmos propostos apresentam desempenhos, medidos em termos de taxa média de erro de bit e velocidade de convergência, marcadamente superiores aos de métodos tradicionais, freqüentemente aproximando o desempenho dos algoritmos ótimos (MAP) treinados. Além disso, observamos que os métodos baseados em filtros de partículas determinísticos exibem desempenhos consistentemente superiores aos dos demais métodos, sendo portanto a melhor escolha caso o modelo de sinal empregado permita a marginalização analítica dos parâmetros desconhecidos do canal.

Abstract

In this thesis, we propose and analyze blind equalization methods suitable for linear FIR communications channels, focusing on the development of algorithms based on particle filters - recursive methods for approximating Bayesian solutions to stochastic filtering problems. Initially, we propose new equalization methods for signal models with gaussian additive noise that dispense with the need for differentially encoding the transmitted signals, as opposed to the previously existing methods. Next, we extend these algorithms to deal with non-gaussian additive noise by deploying artificial parameter evolution techniques. We next develop new joint blind equalization and decoding algorithms, suitable for convolutionally or block-coded communications systems. Via numerical simulations we show that the proposed algorithms outperform traditional approaches both in terms of mean bit error rate and convergence speed, and closely approach the performance of the optimal (MAP) trained equalizer. Furthermore, we observed that the methods based on deterministic particle filters consistently outperform those based on stochastic approaches, making them preferable when the adopted signal model allows for the analytic marginalization of the unknown channel parameters.

Capítulo 1

Introdução

A equalização cega de canais de comunicação tornou-se um dos tópicos mais freqüentes na literatura especializada em processamento digital de sinais nas últimas décadas, tanto pela dificuldade como pelos potenciais ganhos associados à sua solução.

Esta técnica - que consiste de estimar uma mensagem digital transmitida por um canal seletivo em freqüência (de parâmetros desconhecidos) a partir da observação exclusiva das saídas deste canal - permite contrabalançar os efeitos deletérios da propagação de sinais por canais dispersivos sem incorrer em desperdício de banda, seja pela necessidade de transmitir informações redundantes (OFDM) ou conhecidas (equalização treinada) para obter os mesmos efeitos, encontrando aplicação em todos os sistemas modernos de comunicação digital (telefones móveis, dispositivos GPS, receptores de rádio e TV digitais).

Desde a publicação do trabalho pioneiro de Sato [2] desenvolveu-se uma extensa literatura sobre algoritmos cegos para a equalização/identificação de canais de comunicação [3]. Dada a impossibilidade prática¹ de se calcular a solução ótima deste problema, a maioria dos métodos inicialmente desenvolvidos (CMA, predição linear, etc. [4]) envolve a determinação de um filtro inverso, tendo portanto o seu desempenho limitado por aquele do equalizador linear ótimo (calculado através da equação de Wiener-Hopf [5]), que pode ser bastante insatisfatório caso o canal de transmissão equivalente em tempo discreto possua zeros sobre a circunferência unitária.

Além levar a um desempenho inferior ao que seria obtido caso se empregassem estimativas Bayesianas, a maioria destes métodos se fundamenta, explicitamente ou não, na estimação de momentos dos sinais recebidos, o que introduz limitações intrínsecas: por um lado, métodos baseados em momentos de ordem superior exibem, em

¹Verifica-se facilmente que a determinação exata da densidade *a posteriori* dos símbolos transmitidos (necessária para a determinação da solução ótima MAP) requer a realização de uma busca exaustiva sobre todas mensagens possíveis de serem transmitidas, cujo número cresce exponencialmente com o tamanho do bloco de sinal recebido.

geral, velocidades de convergência insatisfatórias, dado o elevado número de amostras necessários à sua convergência [3]; por outro lado, métodos baseados em momentos de segunda ordem são incapazes de estimar certas classes de canais, o que leva a severos problemas de robustez [6]. Tais fatos, associados a outras dificuldades de caráter algorítmico, fizeram com que, apesar de todo esforço de pesquisa dedicado nas últimas décadas, o problema da equalização cega carecesse de soluções aplicáveis em sistemas práticos de comunicações.

Em paralelo a estes desenvolvimentos, a disponibilidade de computadores velozes a custos acessíveis estimulou o estudo de métodos computacionalmente intensivos para a aproximação de soluções Bayesianas, como algoritmo EM [7], os métodos MCMC [8] e os filtros de partículas [9]. Estes métodos, à custa de uma complexidade computacional elevada, permitem resolver uma ampla gama de problemas de filtragem estocástica (dos quais a equalização cega é um caso particular) de forma (aproximadamente) ótima e adaptativa, evitando os problemas de métodos tradicionais baseados em momentos.

Neste trabalho, propomos e avaliamos o desempenho de algoritmos que aproximam soluções ótimas para o problema da equalização cega de canais de comunicação. Em particular, enfatizamos o uso e a análise de métodos baseados em filtros de partículas dada a generalidade destas técnicas, que permite o desenvolvimento de algoritmos adequados a modelos de sinal intratáveis por outros métodos.

Apesar de não ser uma novidade [10], a aplicação de métodos computacionalmente intensivos em problemas de telecomunicações não ganhou visibilidade até recentemente [11] [12] [13]. Estendemos aqui muitos dos resultados obtidos nestas últimas três referências para modelos de sinal distintos dos considerados originalmente, desenvolvendo métodos inéditos para i) equalização cega e decodificação conjuntas em canais seletivos em frequência e ii) equalização cega sob ruídos de distribuições não-gaussianas - problemas importantes na área de comunicações para os quais ainda não haviam sido propostas soluções.

Nas seções a seguir, apresentamos o problema da equalização cega e formalizamos os objetivos deste trabalho, dando na seqüência uma prévia dos próximos capítulos.

1.1 Formulação do Problema

Este trabalho tem por foco a detecção de símbolos transmitidos por sistemas de comunicação digitais lineares através de canais seletivos em frequência FIR. Denotando por $x_{0:n}$ a seqüência de símbolos transmitidos até o instante n , assumiremos que as amostras y_n do sinal recebido em banda-base obtidas à taxa de símbolo T (i.e., $y_n = y(t - Tn)$)

podem ser descritas² como

$$y_n = \sum_{i=0}^{L-1} x_{n-i} h_{i,n} + v_n, \quad (1.1)$$

em que:

- $h_{i,n}$ denota o i -ésimo termo da resposta ao impulso do canal no instante n ;
- L denota a ordem (i.e., o número de coeficientes não-nulos da resposta ao impulso) do canal;
- v_n representa a contribuição do ruído aditivo.

Nos próximos capítulos, descrevemos métodos para a estimação dos símbolos transmitidos adotando diferentes hipóteses sobre cada um dos componentes do modelo descrito em (1.1), sem jamais abandoná-lo, no entanto.

Nos termos da Eq. (1.1), a forma mais básica do problema da equalização cega consiste de estimar a seqüência de símbolos transmitida $x_{0:n}$ baseando-se na observação de $y_{0:m}$, $m \geq n$, tendo conhecimento das estatísticas *a priori* dos parâmetros do canal, do ruído aditivo e da seqüência transmitida.

Este problema, que permaneceu sem solução até 1.975, é de fato bastante desafiador do ponto de vista da teoria de estimação, podendo ser entendido como um caso particular de uma classe mais ampla de problemas de filtragem estocástica [14], uma vez que (1.1) pode ser reescrita de modo que as variáveis a estimar (símbolos transmitidos) constituam o estado desconhecido de um sistema dinâmico

$$\begin{cases} X_{n+1} &= F X_n + f x_n \\ y_n &= h_n^H X_n + v_n \end{cases}, \quad (1.2)$$

em que $X_n = [x_n \cdots x_{n-L+1}]^T \in \mathbb{C}^{(L \times 1)}$, $h_n = [h_{0,n} \cdots h_{L-1,n}]^T \in \mathbb{C}^{(L \times 1)}$, e $F \in \mathbb{R}^{(L \times L)}$ e $f \in \mathbb{R}^{(L \times 1)}$ são dados por

$$F = \begin{bmatrix} 0 & 0 & & 0 & 0 \\ 1 & 0 & & 0 & 0 \\ 0 & 1 & & 0 & 0 \\ & & \ddots & 0 & 0 \\ 0 & & & 1 & 0 \end{bmatrix} \quad f = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Ao se reformular o problema da equalização cega como um problema de filtragem estocástica, diversas conclusões decorrem de forma imediata. Primeiramente, o

²Este modelo de sinal representa fielmente o sinal recebido em banda base por um sistema de comunicações linear caso o receptor esteja perfeitamente sincronizado como o receptor.

modelo em espaço de estados de (1.2) conduz à conclusão de que caso o vetor de parâmetros h_n fosse conhecido e os processos x_n (ruído de excitação) e v_n (ruído de observação) tivessem distribuições gaussianas, o vetor de estado X_n poderia ser estimado de forma ótima³ através de um filtro de Kalman [5].

Num sistema de comunicação digital - no qual X_n tem por definição uma distribuição discreta - a aplicação de filtros de Kalman ao modelo de (1.1) produz estimativas ótimas de acordo com o critério mais “fraco” LMMSE⁴ [15] e, mesmo assim, isto ocorre somente se os parâmetros do canal h_n forem conhecidos. Do contrário, a aplicação de um filtro de Kalman clássico deixa de ser possível, uma vez que para isto é necessário que a função que relaciona o vetor de estados às observações seja conhecida (e linear).

Como uma tentativa extrema de aplicar métodos clássicos de estimação na solução do problema da equalização cega, Lee et al. [16] explorou a seguinte representação equivalente de (1.1) no espaço de estados

$$\left\{ \begin{array}{l} \begin{bmatrix} X_{n+1} \\ h_n \end{bmatrix} = \begin{bmatrix} F & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} X_k \\ h_n \end{bmatrix} + \begin{bmatrix} x_n \\ 0 \end{bmatrix} \\ y_n = g(X_n, h) + v_n \end{array} \right. \quad (1.3)$$

com $g(X_n, h_n) = h_n^H X_n$, o que permite que se descreva o sinal observado como uma função não-linear de um vetor de estado estendido (pela inclusão dos parâmetros do canal). Esta representação permite o uso de versões linearizadas do filtro de Kalman, como por exemplo o EKF (*Extended Kalman Filter*). Infelizmente, o algoritmo obtido em [16] apresenta sérios problemas de estabilidade numérica, comuns à maioria dos métodos baseados no EKF.

1.2 Equalização Bayesiana

Lembrando que o objetivo primordial de um sistema de comunicação é, dada uma seqüência de observações $y_{0:n}$, determinar os símbolos transmitidos x_l , $0 \leq l \leq n$ incorrendo na mínima taxa média de erros possível, resultados clássicos da teoria de estimação [17] garantem que este objetivo é atingido pela estimativa MAP (Maximo *a*

³Neste caso, ótima nos sentidos de MMSE, ML e MAP simultaneamente, pelo fato de X_n de um vetor aleatória gaussiano.

⁴Recebe a denominação LMMSE (*Linear Minimum Mean Square Estimator*) o estimador linear que minimiza o erro médio quadrático.

posteriori), i.e.

$$\hat{x}_l = \arg \max_x \mathbb{P}(x_l = x | y_{0:n}). \quad (1.4)$$

Até recentemente, o resultado de (1.4) não era mais que uma curiosidade teórica, uma vez que a distribuição $\mathbb{P}(x_l = x | y_{0:n})$ não pode ser calculada de forma exata em nenhum caso de interesse prático⁵. Nas últimas décadas, no entanto, foram desenvolvidos diversos métodos que permitem aproximar a densidade *a posteriori* (1.4), dentre os quais destacam-se o algoritmo EM, os métodos MCMC e os Filtros de Partículas, estes últimos o foco desta tese.

Os filtros de partículas são métodos recursivos para a solução de problemas de filtragem estocástica, que pela sua generalidade e flexibilidade podem ser aplicados na solução de uma ampla gama de problemas de estimação em comunicações [12]. No Cap. 2 descrevemos detalhadamente alguns algoritmos desta classe, aplicando-os na solução de problemas de comunicações nos capítulos seguintes.

Nos apêndices, descrevemos também o uso de uma versão modificada do algoritmo EM e de métodos MCMC para o mesmo fim. O algoritmo EM difere das demais técnicas pelo seu caráter determinístico, obtendo soluções iterativamente através de um sistema de equações não-lineares. Este método tem sua aplicabilidade limitada a modelos de sinal relativamente simples⁶ e, conforme mostramos, é inadequado à solução direta⁷ do problema da equalização cega. Os métodos MCMC, por outro lado, permitem estimar diretamente as densidades *a posteriori* dos símbolos transmitidos, conduzindo à equalização bem como à identificação dos parâmetros do canal, valendo-se para isto de técnicas de simulação de variáveis aleatórias de aplicabilidade bastante geral, que, no entanto, são apenas adequadas ao processamento em blocos.

1.3 Organização do texto

O conteúdo a seguir está organizado da seguinte forma:

- O Cap. 2 apresenta a teoria e descreve detalhadamente os filtros de partículas, ilustrando sua aplicação em problemas de filtragem estocástica para modelos lineares e não-lineares através de exemplos.
- O Cap. 3 aplica os filtros de partículas na equalização cega de canais invariantes

⁵É fácil demonstrar que a distribuição em questão é uma razão de somas de gaussianas, sendo o número de termos das somas uma função exponencial do tamanho do bloco n [18].

⁶Esta limitação se deve primordialmente à necessidade de se calcular e maximizar analiticamente determinadas esperanças condicionais [19].

⁷O método determina aproximações das estimativa ML/MAP dos parâmetros do canal, que podem ser empregadas para estimar (de forma subótima) os símbolos transmitidos.

no tempo, apresentando novas técnicas para a solução do problema da ambigüidade de fase. Este capítulo introduz também o uso de técnicas de evolução artificial de parâmetros (EA) em equalização, utilizando-as no desenvolvimento de algoritmos para modelos de sinal com ruído aditivo não-gaussiano.

- O Cap. 4 apresenta métodos inéditos para a equalização cega e decodificação conjuntas baseados em filtros de partículas. A Sec. 4.2 descreve um método que estima diretamente o sinal codificado, especialmente adequado a sistemas de comunicação que empregam códigos de bloco sistemáticos, enquanto a Sec. 4.3 apresenta uma segunda família de algoritmos, adequada a sistemas que empregam codificação convolucional.

As conclusões, bem como sugestões de tópicos para investigações futuras são deixadas para o Cap. 5.

No Apêndice A, avaliamos o desempenho dos algoritmos descritos no Cap. 3 em situações nas quais a ordem exata do canal é desconhecida, estendendo-os também para a equalização de canais variantes no tempo segundo um modelo AR(1). Nos Apêndices B e C, por sua vez, empregamos métodos alternativos de estimação Bayesiana para a solução do problema da equalização cega: no Apêndice B, apresentamos a versão Bayesiana do algoritmo EM, utilizando-a no desenvolvimento de um algoritmo para identificação cega e avaliando o seu desempenho através de simulações numéricas; no Apêndice C, por sua vez, apresentamos uma breve introdução aos métodos MCMC, mostrando em seguida dois métodos para equalização cega baseados no Amostrador de Gibbs.

1.4 Principais Contribuições

Neste trabalho, analisamos e estendemos diversos métodos para equalização propostos principalmente em [1],[11],[12] e [13]. Dentre os resultados originais obtidos, destacam-se como mais importantes:

- Desenvolvimento novos métodos para a solução do problema da ambigüidade de fase, através de novos algoritmos para equalização (Cap. 3) e da utilização de códigos com invariâncias rotacionais (Cap. 4).
- Desenvolvimento de algoritmos para equalização cega sob ruído aditivo não-gaussiano [20] (Cap. 3).
- Desenvolvimento de algoritmos para equalização e decodificação conjuntas em canais seletivos em frequência [21][22][23] (Cap. 4).

- Proposta do uso de métodos de *annealing* para a melhoria das propriedades de convergência do algoritmo EM para equalização cega [24] (Apêndice B).

1.5 Publicações

O conteúdo desta tese deu origem às seguintes publicações (anexadas ao final deste documento):

O artigo “*Particle Filters for Blind FIR Channel Equalization in Non-Gaussian Noise*” [20], aceito para publicação nos anais do ICASSP 2006, trata do conteúdo da Sec. 4.3, descrevendo um algoritmo para equalização cega em modelos de sinal com ruídos de envoltória Weibull. Este trabalho inova ao propor o uso de técnicas de evolução artificial de parâmetros para equalização cega, o que permite que se relaxe a hipótese usual de gaussianidade do ruído aditivo.

O artigo “*Joint Blind Equalization and Decoding using Particle Filters*” [21], publicado nos anais do IEEE International Symposium on Signal Processing and Information Technology, aborda o conteúdo da Sec. 4.2 deste texto, propondo um novo algoritmo para a equalização e decodificação conjuntas em sistemas empregando códigos de bloco.

O artigo “*Particle filter Algorithms for Joint Blind Equalization/Decoding of Convolutionally Coded Signals*” [22], publicado nos anais do ICASSP 2005, expõe basicamente o conteúdo da Sec. 4.3, descrevendo um algoritmo para equalização/decodificação conjuntas adequado a sinais codificados convolucionalmente. Em especial, é tratado o caso em que o código empregado é “não-coerentemente não-catastrófico”, propriedade que permite contornar o problema da ambigüidade de fase sem recorrer aos artifícios mais comumente empregados na literatura.

O artigo “*Deterministic Particle Filters for Joint Blind Equalization and Decoding in Frequency Selective Channels*” [23], publicação nos anais do IEEE Statistical Signal Processing Workshop - SSP 05, aborda o resultado da Sec. 4.3, enfocando porém o uso de códigos convolucionais recursivos e de filtros de partículas combinando estruturas estocásticas e determinísticas.

O artigo “*Equalização Cega de Canais Lineares FIR empregando o Algoritmo EM Bayesiano*” [24], publicado nos anais do XXII Simpósio Brasileiro de Telecomunicações, que consiste do conteúdo do Apêndice B, no qual aplicamos uma versão Bayesiana do algoritmo EM para a equalização cega de canais invariantes no tempo, empregando métodos de *simulated annealing* para a melhoria das características de convergência deste algoritmo.

Capítulo 2

Filtros de Partículas

Neste capítulo apresentamos os fundamentos da teoria dos filtros de partículas - métodos recursivos para a aproximação de soluções Bayesianas para problemas de filtragem estocástica. Após deduzir na Sec. 2.1 o equacionamento básico destes métodos, demonstrando a partir de princípios básicos de teoria de probabilidade o uso de medidas empíricas para a aproximação recursiva de densidades *a posteriori*, discutimos nas Sec. 2.2 e 2.3 aspectos relacionados à implementação dos mesmos, como a escolha da chamada *função de importância* e do método de reamostragem. Na Sec. 2.4, por sua vez, descrevemos as versões estocástica e determinística do algoritmo SIR (*Sequential Importance Resampling*), generalizações do filtro *Bootstrap* que empregamos ao longo deste trabalho na solução de problemas de comunicação.

Na Sec. 2.5, discutimos como aplicar filtros de partículas em modelos de sinal com parâmetros desconhecidos, classe de problemas cuja equalização cega é um exemplo. Conforme mostramos, problemas desta categoria podem ser resolvidos através da marginalização analítica (Rao-Blackwellização) ou numérica (Evolução Artificial) dos parâmetros desconhecidos. Por fim, a Sec. 2.6 contém dois exemplos de uso de filtros de partículas em problemas de estimação.

2.1 Introdução

Em processamento estatístico de sinais, freqüentemente nos deparamos com problemas em que, dada uma seqüência de observações $y_{0:n}$, deseja-se estimar um sinal oculto $x_{0:n}$, que se relaciona ao primeiro segundo um modelo probabilístico conhecido. São exemplos desta categoria de problemas: rastreamento de objetos em imagens ruidosas, redução de ruído em sistemas sonoros, equalização de canais de comunicação, entre outros.

Uma forma bem estabelecida de se abordar problemas deste tipo consiste em mo-

delar $y_{0:n}$ como a saída de um sistema dinâmico, cuja variável a estimar seja igual à (ou uma função simples da) variável de estado X_n :

$$\begin{cases} X_{n+1} &= f_n(X_{0:n}, u_n) \\ y_n &= g_n(X_{0:n}, v_n) \end{cases} \quad (2.1)$$

em que f_n e g_n são funções determinísticas conhecidas, possivelmente não-lineares e variantes no tempo, e u_n e v_n são processos estocásticos mutuamente independentes, respectivamente chamados de ruídos de excitação e de observação.

Como é bem conhecido, estimativas ótimas¹ de $X_{0:n}$ podem ser obtidas a partir da densidade *a posteriori* $p(X_{0:n}|y_{0:n})$. Infelizmente, esta densidade pode ser determinada de forma exata apenas em dois casos (bastante restritivos): i) se f_n e g_n forem funções lineares e u_n e v_n processos gaussianos (filtro de Kalman), ou ii) se X_n puder assumir um número finito de valores e $f_n(X_{0:n}, u_n) = f_n(X_n, u_n)$. Em todos os demais casos, $p(X_{0:n}|y_{0:n})$ pode ser determinada somente de forma aproximada.

Os filtros de partículas, cujos princípios de funcionamento detalhamos a seguir, não são nada mais do que um método para obter aproximações de $p(X_{0:n}|y_{0:n})$. Diferentemente dos métodos usuais (como o filtro de Kalman estendido) que empregam aproximações gaussianas para a densidade *a posteriori*, os filtros de partículas aproximam $p(X_{0:n}|y_{0:n})$ por uma soma ponderada de medidas de Dirac, atualizando estas aproximações recursivamente através do método de *amostragem por importância*, como detalhado a seguir.

2.1.1 Aproximação empírica de uma densidade de probabilidade

O princípio da amostragem por importância [19] (vide Apêndice E) é uma técnica usualmente utilizada para reduzir a variabilidade de estimadores Monte Carlo. Explorando este princípio, densidades de probabilidade condicionais arbitrárias $p(X_{0:n}|y_{0:n})$ podem ser aproximadas² de forma empírica através da relação

$$p_N(X_{0:n}|y_{0:n}) = \frac{\sum_{i=1}^N w_n^{(i)} \delta(X_{0:n} - X_{0:n}^{(i)})}{\sum_{i=1}^N w_n^{(i)}}, \quad (2.2)$$

em que

- $X_{0:n}^{(i)}$ são amostras independentes (partículas) tomadas de uma distribuição $\pi(X_{0:n}^{(i)}|y_{0:n})$ (função de importância).

¹Segundo os critérios MAP e MMSE, entre outros.

²Empregando-se conceitos avançados de teoria de probabilidades, pode-se verificar que, $p_N(X_{0:n}|y_{0:n})$ converge em distribuição para $p(X_{0:n}|y_{0:n})$ quando $M \rightarrow \infty$ ([9], Cap. 2).

- $w_n^{(i)} \triangleq p(X_{0:n}^{(i)}|y_{0:n})/\pi(X_{0:n}^{(i)}|y_{0:n})$ são pesos associados a cada partícula.

O objetivo básico de um filtro de partículas é determinar (2.2) de forma recursiva. Para isto, são necessários dois requisitos fundamentais: deve-se ser capaz de (i) gerar amostras da função de importância $\pi(X_{0:n}^{(i)}|y_{0:n})$ seqüencialmente, e (ii) atualizar os pesos $w_n^{(i)}$ de forma recursiva.

Como mostrado no Apêndice E, amostras de uma distribuição condicional da forma $\pi(X_{0:n}^{(i)}|y_{0:n})$ não podem ser obtidas seqüencialmente para uma distribuição π genérica. Esta limitação, no entanto, pode ser contornada [25] definindo-se uma função de importância $\pi(X_{0:n}|y_{0:n})$ a partir das suas marginais, como³:

$$\pi(X_{0:n}|y_{0:n}) \triangleq \pi(X_0|y_0) \prod_{k=1}^n \pi(X_k|X_{0:k-1}, y_{0:k}) . \quad (2.3)$$

Conforme mostramos a seguir, ao se definir $\pi(X_{0:n}|y_{0:n})$ como em (2.3), o requisito (ii) é automaticamente atendido, sendo possível atualizar os pesos $w_n^{(i)}$ recursivamente.

2.1.2 Cálculo recursivo dos pesos

Por definição, os pesos w_n são calculados como:

$$w_n \triangleq \frac{p(X_{0:n}|y_{0:n})}{\pi(X_{0:n}|y_{0:n})} . \quad (2.4)$$

Note agora que, como consequência da lei de Bayes, podemos escrever:

$$p(X_{0:n}|y_{0:n}) = p(X_{0:n-1}|y_{0:n-1}) \frac{p(X_n, y_n|X_{0:n-1}, y_{0:n-1})}{p(y_n|y_{0:n-1})} . \quad (2.5)$$

Ainda pela sua definição, a função de importância se fatora como:

$$\pi(X_{0:n}|y_{0:n}) = \pi(X_{0:n-1}|y_{0:n-1})\pi(X_n|X_{0:n-1}, y_{0:n}) , \quad (2.6)$$

conduzindo a

$$w_n = w_{n-1} \frac{p(X_n, y_n|X_{0:n-1}, y_{0:n-1})}{p(y_n|y_{0:n-1})\pi(X_n|x_{0:n-1}, y_{0:n})} . \quad (2.7)$$

O termo $p(y_n|y_{0:n-1})$ presente no denominador de (2.7), em geral, é de cálculo bastante complexo. Como este termo não é função de $X_{0:n}$, pode-se ignorá-lo. Logo,

³Note que, por definição, $\pi(X_{0:n}|y_{0:n})$ fatora-se em densidades condicionadas somente em elementos passados das séries temporais envolvidas; assim, amostras de $\pi(X_{0:n}|y_{0:n})$ podem ser obtidas seqüencialmente como $X_k \sim \pi(X_k|X_{0:k-1}, y_{0:k})$, $0 \leq k \leq n$.

obtemos finalmente:

$$w_n \propto w_{n-1} \frac{p(X_n, y_n | X_{0:n-1}, y_{0:n-1})}{\pi(X_n | X_{0:n-1}, y_{0:n})}. \quad (2.8)$$

Observação 1 *O equacionamento apresentado até este ponto é absolutamente geral, não necessitando de nenhuma hipótese restritiva além das usuais condições de regularidade sobre o modelo (vide [9], Cap. 2). Note, no entanto, que as densidades necessárias para obter amostras e atualizar os pesos são determinadas pelo modelo de sinal, não tendo em muitos casos uma forma analítica tratável.*

2.2 Degenerescência e Reamostragem

O desempenho dos filtros de partículas, quantificado pela qualidade da aproximação da densidade *a posteriori* de (2.2), é fortemente influenciado pela distribuição dos pesos $w_n^{(i)}$. De fato, pode-se mostrar [26] que o erro da aproximação

$$E|p_N(X_{0:n}|y_{0:n}) - p(X_{0:n}|y_{0:n})|^2$$

é assintoticamente proporcional à variância dos pesos (não-normalizados) $\bar{w}_n^{(i)}$, a qual verifica-se ser uma função não-decrescente do tempo, fruto da seqüência $\bar{w}_{0:n}^{(i)}$ ser um martingal.

A crescente variabilidade dos pesos não-normalizados $\bar{w}_n^{(i)}$ traduz-se numa distribuição fortemente não-uniforme para os pesos normalizados $w_n^{(i)}$. Verifica-se que os pesos associados à maioria das partículas tendem a se tornar desprezíveis após um número finito de iterações. Este fenômeno, conhecido como “degenerescência”, prejudica o funcionamento dos filtros de partículas, sendo em geral combatido por técnicas de reamostragem.

Um indicador bastante utilizado para se avaliar o grau de degenerescência de um conjunto de partículas é o tamanho efetivo da amostra T_n , estimado [26] como:

$$T_n = N \left(N + \sum_{i=1}^N (Nw_n^{(i)} - 1)^2 \right)^{-1}. \quad (2.9)$$

Note que se os pesos distribuírem-se uniformemente $w_n^{(i)} = 1/N$, $1 \leq i \leq N$ (situação ideal, em que $X_n^{(i)}$ são amostras da *posteriori* desejada), T_n assumirá seu valor máximo (igual a 1). Em geral, o processo de reamostragem é acionado sempre que T_n cair abaixo de um determinado limiar. A idéia básica deste processo é descartar as partículas associadas a pesos desprezíveis, substituindo-as por replicações (i.e., cópias)

das partículas com pesos maiores, idealmente sem introduzir perturbações adicionais nas estimativas de $p(X_{0:n}|y_{0:n})$. Assim, dada a estimativa atual $p_N(X_{0:n}|y_{0:n})$, definida pelo conjunto de partículas e pesos $\{X_{0:n}^{(i)}, w_n^{(i)}\}$, deseja-se encontrar uma nova estimativa $\tilde{p}_N(X_{0:n}|y_{0:n})$, definida pelo conjunto $\{X_{0:n}^{k^{(i)}}, \tilde{w}_n^{(i)} = 1/N\}$ (i.e., com pesos uniformes), que seja próxima a $p_N(X_{0:n}|y_{0:n})$ segundo algum critério.

Nas seções a seguir descrevemos alguns dos diversos algoritmos para reamostragem empregados na literatura.

2.2.1 Reamostragem Multinomial

Talvez devido à sua simplicidade, o algoritmo de reamostragem multinomial [9, Cap. 1] seja o mais empregado na literatura. Por este método, os índices $k^{(i)}$ das partículas replicadas são obtidos amostrando-se independentemente N vezes da distribuição discreta

$$\mathbb{P}(k^{(i)} = j) = w_n^{(j)}, \quad 1 \leq j \leq N. \quad (2.10)$$

Pode-se verificar que o número de vezes $(k^{(1)}, \dots, k^{(N)})$ que cada partícula é replicada (i.e., incluída no conjunto reamostrado) tem distribuição multinomial, resultando que $E[k^{(i)}] = Nw_n^{(i)}$. Disto resulta que $E[\tilde{p}_N(X_{0:n}|y_{0:n})] = p_N(X_{0:n}|y_{0:n})$, ou seja, que o processo de reamostragem multinomial não introduz viés.

Observação 2 *Diversos métodos podem ser utilizados para a geração de amostras de uma distribuição discreta como (2.10). Um destes métodos [19] consiste em gerar uma amostra de uma distribuição uniforme $u \sim \mathcal{U}[0; 1]$, determinando j como o índice que satisfaz*

$$\sum_{i=1}^{j-1} w_n^{(i)} \leq u < \sum_{i=1}^j w_n^{(i)}, \quad (2.11)$$

com $w_n^{(0)} \triangleq 0$. Amostras de distribuições uniformes, por sua vez, podem ser obtidas de acordo com inúmeros métodos derivados da teoria de números, como os descritos em [19].

2.2.2 Reamostragem Residual

Apesar de não ser viesado, o processo de reamostragem multinomial introduz variabilidades desnecessárias: note que toda partícula $X^{(i)}$ associada a um peso não-nulo $w^{(i)}$ pode ser replicada $0 \leq k^{(i)} \leq N$ vezes.

Pelo método de reamostragem residual, cada partícula é replicada $k^{(i)} = \lfloor Nw^{(i)} \rfloor + r^{(i)}$ vezes, sendo o número de replicações residual $r^{(i)}$, $1 \leq i \leq N$ obtido como

no método multinomial, considerando agora a densidade discreta determinada pelas probabilidades

$$\left\{ w_n^{(1)} - \frac{\lfloor Nw^{(1)} \rfloor}{N}, \dots, w_n^{(N)} - \frac{\lfloor Nw^{(N)} \rfloor}{N} \right\}.$$

Note que, pelo método de reamostragem residual, o número de replicações de cada partícula está restrito ao intervalo $\lfloor Nw^{(i)} \rfloor \leq n^{(i)} \leq N$, diminuindo a probabilidade de que uma partícula com peso não-nulo seja descartada. Note ainda que o método não introduz viés, uma vez que:

$$E[k^{(i)}] = \lfloor Nw^{(i)} \rfloor + N \left(w^{(i)} - \frac{\lfloor Nw^{(i)} \rfloor}{N} \right) = Nw^{(i)}. \quad (2.12)$$

2.2.3 Reamostragem sistemática

Este método de reamostragem obtém uma redução de variabilidade recorrendo ao princípio da amostragem sistemática [27], bastante utilizado na teoria da integração estocástica. Por este método, o número de replicações $k^{(i)}$ é obtido da seguinte forma [28]:

- Obtenha uma seqüência ordenada de variáveis aleatórias $u^{(i)}$, $1 \leq i \leq N$ como:

$$u^{(i)} = \frac{(i-1) + \bar{u}_n}{N}, \quad \bar{u}_n \sim \mathcal{U}[0; 1] \quad (2.13)$$

- Determine $n^{(i)}$ como:

$$k^{(i)} \equiv \text{numero de vezes que } u^{(i)} \in \left[\sum_{j=1}^{i-1} w^{(j)} ; \sum_{j=1}^i w^{(j)} \right] \quad (2.14)$$

Além de poder ser implementado com complexidade $\mathcal{O}(N)$ pode-se verificar ([9], Cap. 2) que o método de reamostragem sistemática é ótimo no sentido de minimizar $E|\tilde{p}_N(X_{0:n}|y_{0:n}) - p_N(X_{0:n}|y_{0:n})|^2$.

2.2.4 Filtro *Bootstrap*

Combinando o método de amostragem por importância seqüencial de Gordon (Eq. 2.3) com a aplicação de um passo de reamostragem multinomial a cada iteração (independentemente do tamanho efetivo da amostra), chega-se ao algoritmo conhecido como Filtro *Bootstrap* [9]. Na Tabela 2.1, mostramos a formulação mais comum deste método

[9, Cap. 1], adequada a modelos HMM⁴ (*Hidden Markov Model*).

Como se pode observar na Tab. 2.1, o método utiliza como função de importância a densidade de transição de estado $p(X_n|X_{n-1})$. Cada valor proposto $X_n^{(i)}$ recebe então um peso proporcional à sua verossimilhança, sendo replicado (pelo processo de reamostragem multinomial) um número de vezes proporcional a este valor.

<p>(Inicialização) Para $i = 1 : N$, $X_0^{(i)} \sim p(X_0)$. Fim (Algoritmo) Para $n > 0$, 1) Para $i = 1 : N$, A) Amostre $X_n^{(i)} \sim p(X_n X_{n-1}^{(i)})$. B) Calcule os pesos $w_n^{(i)} \propto p(y_n X_n^{(i)})$. Fim 2) Normalize os pesos e reamostre o conjunto de partículas (pelo método multinomial). 3) Estime $p(X_{0:n} y_{0:n}) \approx \sum_{i=1}^N \delta(X_{0:n} - X_{0:n}^{(i)})$. Fim</p>

Tabela 2.1: Descrição de um Filtro *Bootstrap*.

2.3 Escolha da função de importância

Embora a função de importância $\pi(X_n|X_{0:n-1}, y_{0:n})$ possa ser escolhida de forma essencialmente arbitrária⁵, duas escolhas são particularmente frequentes na literatura sobre filtros de partículas: a função de importância ótima e a função de importância *a priori*.

2.3.1 Função de Importância Ótima

Pode-se demonstrar que a função de importância $\pi(X_n|X_{0:n-1}, y_{0:n}) = p(X_n|X_{0:n-1}, y_{0:n})$ é “ótima” no sentido de anular a variância condicional dos pesos (não-normaliza-

⁴Por HMM, entendemos um modelo em que $p(X_n|X_{0:n-1}, y_{0:n-1}) = p(X_n|X_{n-1})$ e que $p(y_n|X_{0:n}, y_{0:n-1}) = p(y_n|X_n)$, o que equivale a assumir que $X_{n+1} = f(X_n, u_n)$ e $y_n = g(X_n, v_n)$, com u_n e v_n independentes. Como veremos a seguir (Sec. 2.2), os filtros *bootstrap* não são mais que um caso particular do algoritmo SIR, podendo ser aplicados a modelos mais gerais do que este.

⁵Bastando que a função escolhida satisfaça (2.3).

dos) dados $(X_{0:n-1}, y_{0:n})$, já que

$$\begin{aligned}
& \text{VAR}[w_n | X_{0:n-1}, y_{0:n}] \\
& \triangleq \int dX_n \bar{w}_n^2 p(X_n | X_{0:n-1}, y_{0:n}) - \left[\int dX_n \bar{w}_n p(X_n | X_{0:n-1}, y_{0:n}) \right]^2 \\
& \propto \int dX_n w_{n-1}^2 \frac{p(X_n, y_n | X_{0:n-1}, y_{0:n-1})^2}{p(X_n | X_{0:n-1}, y_{0:n})} - \left[\int dX_n w_{n-1} p(X_n, y_n | X_{0:n-1}, y_{0:n-1}) \right]^2 \\
& = w_{n-1}^2 p(y_n | X_{0:n-1}, y_{0:n-1}) \int dX_n p(X_n, y_n | X_{0:n-1}, y_{0:n-1}) \\
& \quad - w_{n-1}^2 \left[\int dX_n p(X_n, y_n | X_{0:n-1}, y_{0:n-1}) \right]^2 \\
& = w_{n-1}^2 p(y_n | X_{0:n-1}, y_{0:n-1})^2 - w_{n-1}^2 p(y_n | X_{0:n-1}, y_{0:n-1})^2 \\
& = 0,
\end{aligned} \tag{2.15}$$

fato que beneficia a qualidade da estimativa de $p(X_{0:n} | y_{0:n})$ [18]. Ao se empregar tal função de importância, os pesos podem ser atualizados como

$$w_n \propto w_{n-1} \frac{p(X_n, y_n | X_{0:n-1}, y_{0:n-1})}{p(X_n | X_{0:n-1}, y_{0:n})}, \tag{2.16}$$

A função de importância ótima, por sua vez, pode ser obtida através da relação

$$p(X_n | X_{0:n-1}, y_{0:n}) = \frac{p(X_n, y_n | X_{0:n-1}, y_{0:n-1})}{p(y_n | X_{0:n-1}, y_{0:n-1})} = \frac{p(X_n, y_n | X_{0:n-1}, y_{0:n-1})}{\int dX_n p(X_n, y_n | X_{0:n-1}, y_{0:n-1})}. \tag{2.17}$$

2.3.2 Função de Importância *a priori*

Neste caso, as partículas são amostradas da densidade⁶ $\pi(X_n | X_{0:n-1}, y_{0:n-1})$, sendo os pesos atualizados como

$$w_n \propto w_{n-1} \frac{p(X_n, y_n | X_{0:n-1}, y_{0:n-1})}{p(X_n | X_{0:n-1}, y_{0:n-1})} = p(y_n | X_{0:n}, y_{0:n-1}). \tag{2.18}$$

A escolha da função de importância *a priori* leva, em geral, a resultados piores que os que seriam obtidos empregando a função de importância ótima, o que pode ser justificado intuitivamente pelo fato de que a observação atual y_n é ignorada ao se amostrar $X_n \sim p(X_n | X_{0:n-1}, y_{0:n-1})$. Note, no entanto, que a determinação da função de importância ótima (2.17) exige o cálculo de uma integral, o que em geral não é factível, especialmente para modelos de sinal não-lineares e/ou não-gaussianos.

⁶Note que a definição que empregamos nesta tese para a função de importância *a priori* é mais geral que a comumente encontrada na literatura, que é particularizada para HMMs.

2.4 Algoritmo SIR

Na Tabela 2.2 descrevemos o algoritmo SIR (*Sequential Importance Resampling*), nome cunhado por Doucet [9] para descrever versões generalizadas⁷ do filtro *Bootstrap*. O algoritmo SIR difere do filtro *Bootstrap* pela capacidade de empregar funções de importância genéricas e de propagar os pesos das partículas ao longo das iterações, o que permite reduzir a frequência de reamostragem.

Vale mencionar que embora estas generalizações possam levar a uma substancial melhora de desempenho em muitos casos, a literatura ainda carece de resultados [10] relacionando a escolha de um limiar T_n ótimo para reamostragem ao modelo de sinal e à função de importância adotados.

<p>(Inicialização) Para $i = 1 : N$, $X_0^{(i)} \sim p(X_0)$. $w_0^{(i)} = 1/N$. Fim (Algoritmo) Para $n > 0$, 1) Para $i = 1 : N$, A) Amostre $X_n^{(i)} \sim \pi(X_n X_{0:n-1}^{(i)}, y_{0:n})$. B) Calcule e normalize os pesos $w_n^{(i)} \propto w_{n-1}^{(i)} p(X_n^{(i)}, y_n X_{0:n-1}^{(i)}, y_{0:n-1}) / \pi(X_n^{(i)} X_{0:n-1}^{(i)}, y_{0:n})$. Fim 2) Calcule o tamanho efetivo da amostra $T_n = N(N + \sum_{i=1}^N (Nw_n^{(i)} - 1)^2)^{-1}$. 3) Caso $T_n \leq T_{MIN}$, reamostre o conjunto de partículas e faça $w_n^{(i)} = 1/N$. 4) Estime $p(X_{0:n} y_{0:n}) \approx \sum_{i=1}^M w_n^{(i)} \delta(X_{0:n} - X_{0:n}^{(i)})$. Fim</p>

Tabela 2.2: Algoritmo SIR

2.4.1 Algoritmo SIR Determinístico

O algoritmo SIR determinístico foi proposto independentemente por diversos autores ([12][29]) a partir da observação de que, para modelos de sinal com estados X_n discretos, a função de importância ótima pode ser calculada através da relação

$$p(X_n | X_{0:n-1}, y_{0:n}) = \frac{p(y_n | X_{0:n}, y_{0:n-1}) p(X_n | X_{0:n-1}, y_{0:n-1})}{\sum_{X_n} p(y_n | X_{0:n}, y_{0:n-1}) p(X_n | X_{0:n-1}, y_{0:n-1})}. \quad (2.19)$$

⁷Observar que o algoritmo SIR reduz-se ao filtro *bootstrap* caso se adote $\pi(X_n | X_{0:n-1}, y_{0:n}) = p(X_n | X_{0:n-1}, y_{0:n-1})$ e $T_{MIN} = 1$.

Isto implica que, num instante n , são calculadas as verossimilhanças de todos os estados X_n possíveis. Dai, surgiu a idéia de, ao invés de selecionar independentemente um novo vetor de estados $X_n^{(i)}$ para cada partícula dado $X_{n-1}^{(i)}$ (como no algoritmo SIR tradicional), selecionar N partículas dentre o conjunto de ND vetores X_n possíveis⁸ (dados $X_{n-1}^{(i)}, 0 < i \leq N$). Na Tab. 2.3 descrevemos o algoritmo resultante.

<p>(Inicialização)</p> <p>Para $i = 0 : N$,</p> <p style="padding-left: 2em;">$X_0^{(i)} \sim p(X_0)$.</p> <p style="padding-left: 2em;">$w_0^{(i)} = 1/N$.</p> <p>Fim</p> <p>(Algoritmo)</p> <p>Para $n > 0$,</p> <p>1) Para $i = 1 : N$,</p> <p style="padding-left: 2em;">Para $j = 1 : D$,</p> <p style="padding-left: 4em;">Calcule e normalize os pesos</p> <p style="padding-left: 4em;">$w_n^{(i,j)} \propto w_{n-1}^{(i)} p(X_n^{(i,j)}, y_n X_{0:n-1}^{(i)}, y_{0:n-1})$</p> <p style="padding-left: 2em;">Fim</p> <p style="padding-left: 2em;">Fim</p> <p>2) Utilizando um método de reamostragem estocástico ou determinístico, selecione N entre os ND pares (i, j) baseado nos pesos $w_n^{(i,j)}$, formando o conjunto $\{I_i, J_i\}_{i=1}^N$.</p> <p>3) Para $i = 1 : N$,</p> <p style="padding-left: 2em;">Faça $X_n^{(i)} = X_n^{(I_i, J_i)}$ e $w_n^{(i)} = w_n^{(I_i, J_i)}$.</p> <p style="padding-left: 2em;">Fim</p> <p>4) Estime $p(X_{0:n} y_{0:n}) \approx \sum_{i=1}^N w_n^{(i)} \delta(X_{0:n} - X_{0:n}^{(i)})$.</p> <p>Fim</p>
--

Tabela 2.3: Algoritmo SIR Determinístico.

Uma justificativa teórica para o funcionamento do algoritmo SIR determinístico pode ser dada pela observação de que o conjunto $\{w_n^{(i,j)}, X_n^{(i,j)}\}$ (em que $X_n^{(i,j)}$ denota cada um dos D estados atingíveis dado $X_{n-1}^{(i)}$, indexados em qualquer ordem) é uma amostra da densidade *a posteriori* desejada $p(X_{0:n} | y_{0:n})$. Para demonstrar isto, note que o procedimento descrito na Tab. 2.4 pode ser interpretado como uma instância do algoritmo SIR convencional, em que $X_n^{(i,j)}$ é obtido como

$$X_n^{(i,j)} \sim \mathcal{I} \left\{ \left\{ X_n^{(i,j)} \text{ é atingível dado } X_{n-1}^{(i)} \right\} \cap \left\{ X_n^{(i,j)} \neq X_n^{(i,k)}, \forall k \neq j \right\} \right\}, \quad (2.20)$$

⁸Estamos adotando por definição que cada vetor X_n só pode assumir D valores distintos dado X_{n-1} . Conforme veremos nos próximos capítulos, no caso de um sistema de comunicação digital, D será equivalente à dimensão da constelação empregada.

e os pesos atualizados como

$$w_n^{(i,j)} \propto w_{n-1}^{(i)} p(X_n^{(i,j)}, y_n | X_{0:n-1}^{(i)}, y_{0:n-1}), \quad (2.21)$$

o que obedece (2.8) uma vez que a função de importância empregada é sempre unitária para os valores de $X_n^{(i,j)}$ possíveis.

Pelo fato do conjunto $\{w_n^{(i,j)}, X_n^{(i,j)}\}$ conter ND amostras, é obrigatória a realização de um passo de reamostragem a cada iteração para que o número de partículas utilizado seja mantido constante. Isto pode ser feito através dos métodos estocásticos descritos na Sec. 2.2 ou através de procedimentos determinísticos. Em particular, o procedimento determinístico⁹ apresentado em [30], que consiste simplesmente em selecionar os N elementos de $X_n^{(i,j)}$ com os maiores pesos, freqüentemente leva (por razões ainda mal compreendidas) a resultados melhores do que os que se obteriam utilizando métodos de reamostragem estocástica [12].

2.4.2 Suavização (*Smoothing*)

O algoritmo SIR permite que se obtenham soluções suavizadas [5] de forma trivial, a partir da observação de que num instante $m > n$, o conjunto de partículas e pesos $\{w_m^{(i)}; X_{0:m}^{(i)}\}_{i=1}^N$ é uma amostra da densidade $p(X_{0:m}|y_{0:m})$. Com a notação $X_{n,m}^{(i)}$ para o elemento $X_n^{(i)}$ da seqüência $X_{0:m}^{(i)}$ no instante m , segue que

$$p(X_n|y_{0:m}) \approx \frac{1}{N} \sum_{i=1}^N w_m^{(i)} \delta(X_{n,m}^{(i)} - X_n). \quad (2.22)$$

Este método geralmente leva a bons resultados se $m - n \ll N$, o que pode ser explicado pelo fato de que a amostra de X_n tende a empobrecer (i.e., perder diversidade) na medida em que mais passos de reamostragem são aplicados. Além deste, descreveram-se em [11] outros dois métodos para suavização em filtros de partículas, empregando busca exaustiva e densidades preditivas. Destes, somente o primeiro método leva a resultados (ligeiramente) melhores que os produzidos pelo descrito nesta seção, porém às custas de um aumento na complexidade computacional exponencial em função de $|m - n|$.

⁹Salvo menção em contrário, utilizaremos este método de reamostragem sempre que empregarmos o algoritmo SIR determinístico.

2.5 Filtros de Partículas para Modelos com Parâmetros Desconhecidos

Em muitos problemas de estimação de interesse prático (dentre os quais os problemas de equalização cega), as funções de observação g_n ou de transição de estados f_n (2.1) são desconhecidas. Este fato, que em princípio inviabilizaria a solução do problema de estimação, é usualmente contornado ao assumir-se que f_n e g_n pertencem a uma determinada família de funções, sendo suas descrições completadas por um conjunto de parâmetros θ_n , resultando num modelo como

$$\begin{cases} X_{n+1} &= f_n(\theta_n, X_{0:n}, u_n) \\ \theta_{n+1} &= \phi_n(\theta_{0:n}, \epsilon_n) \\ y_n &= g_n(\theta_n, X_{0:n}, v_n) \end{cases} \quad (2.23)$$

em que ϵ_n , u_n e v_n são processos estocásticos (mutuamente independentes) e ϕ_n , f_n e g_n funções determinísticas e conhecidas. Problemas de estimação com modelos como os de (2.23) são bastante desafiadores, existindo até o presente questões teóricas em aberto quanto a existência ou convergência de estimadores capazes de resolvê-los, especialmente no caso em que os parâmetros θ_n são fixos [31].

Neste trabalho, temos interesse numa classe mais restrita de modelos de sinal, nos quais apenas a função de observação g_n possui parâmetros desconhecidos, i.e.,

$$\begin{cases} X_{n+1} &= f_n(X_{0:n}) + u_n \\ \theta_{n+1} &= \phi_n(\theta_n) + \epsilon_n \\ y_n &= g_n(\theta_n, X_n) + v_n \end{cases} \quad (2.24)$$

A seguir, descrevemos técnicas para a solução desta classe de problemas de estimação através de filtros de partículas. Conforme mostramos, a primeira destas técnicas (Rao-Blackwellização) leva a estimativas mais eficientes, sendo porém aplicável apenas a uma classe restrita de modelos de sinal, para os quais a densidade *a posteriori* conjunta dos parâmetros e do estado $p(X_{0:n}, \theta_{0:n} | y_{0:n})$ possa ser marginalizada analiticamente. Todos os demais casos podem ser resolvidos pelo método de Evolução Artificial (EA), através do qual os parâmetros desconhecidos são marginalizados numericamente.

2.5.1 Rao-Blackwellização

Considerando o modelo de (2.24), poder-se-ia em princípio projetar um filtro de partículas para a aproximação da densidade conjunta $p(X_{0:n}, \theta_{0:n} | y_{0:n})$ utilizando os métodos já descritos. Caso não haja interesse na estimação dos parâmetros $\theta_{0:n}$, no entanto,

a aproximação desta densidade conjunta de dimensão maior não só é desnecessária, como em geral leva a uma maior variabilidade das estimativas das grandezas de interesse. Para casos como este, sempre que possível busca-se reduzir a dimensão da variável estimada via integração analítica, procedimento conhecido como Rao-Blackwellização¹⁰ (*Rao-Blackwellization* [8]). Para os modelos de sinal em que temos interesse (2.24), a aproximação da densidade $p(X_{0:n}|y_{0:n})$ através de filtros de partículas Rao-Blackwellizados pode ser feita apenas sob condições bastante especiais, como descrevemos a seguir.

Deixando de lado a princípio o cálculo¹¹ de $\pi(X_n|X_{0:n-1}, y_{0:n})$, o cálculo da densidade $p(X_n, y_n|X_{0:n-1}, y_{0:n-1})$ permite aplicar qualquer um dos algoritmos de filtragem de partículas descritos neste capítulo na aproximação da *posteriori* $p(X_{0:n}|y_{0:n})$. Para este cálculo, utilizamos inicialmente o fato (geral) de que

$$p(X_n, y_n|X_{0:n-1}, y_{0:n-1}) = p(y_n|X_{0:n}, y_{0:n-1})p(X_n|X_{0:n-1}, y_{0:n-1}) . \quad (2.25)$$

Explorando propriedades de independência condicional induzidas pelo modelo (2.24), temos que

$$p(X_n|X_{0:n-1}, y_{0:n-1}) = p(X_n|X_{0:n-1}) . \quad (2.26)$$

Para calcular o primeiro termo do lado direito de (2.25), note inicialmente que

$$p(y_n|X_{0:n}, y_{0:n-1}) = \int p(y_n, \theta_n|X_{0:n}, y_{0:n-1}) d\theta_n , \quad (2.27)$$

e que

$$p(y_n, \theta_n|X_{0:n}, y_{0:n-1}) = p(y_n|\theta_n, X_{0:n}, y_{0:n-1})p(\theta_n|X_{0:n}, y_{0:n-1}) . \quad (2.28)$$

Propriedades do modelo, novamente, implicam que

$$p(y_n|\theta_n, X_{0:n}, y_{0:n-1}) = p(y_n|\theta_n, X_n) = p_{v_n}(y_n - g_n(\theta_n, X_n)) , \quad (2.29)$$

e que

$$p(\theta_n|X_{0:n}, y_{0:n-1}) = p(\theta_n|X_{0:n-1}, y_{0:n-1}) . \quad (2.30)$$

Assim, para projetar um filtro de partículas Rao-Blackwellizado para o modelo de

¹⁰Este nome é inspirado no teorema de Rao-Blackwell [5], que estabelece que $\text{VAR}[\hat{\theta}|T(\theta)] \leq \text{VAR}[\hat{\theta}]$, em que $\hat{\theta}$ e $T(\theta)$ são respectivamente um estimador não-viesado e uma estatística suficiente da variável θ .

¹¹Note que a função de importância ótima é completamente determinada por $p(X_n, y_n|X_{0:n-1}, y_{0:n-1})$ (que calculamos nesta seção), e que a densidade *a priori* não depende de θ_n para o modelo considerado.

interesse, é necessário:

- Determinar $p(X_n|X_{0:n-1})$ (2.26);
- Determinar $p_{v_n}(y_n - g_n(\theta_n, X_n))$ (2.29);
- Calcular a integral de (2.27);
- Calcular $p(\theta_n|X_{0:n-1}, y_{0:n-1})$ recursivamente.

Esta última exigência é, em geral, a mais restritiva, uma vez que a densidade $p(\theta_n|X_{0:n-1}, y_{0:n-1})$ é a solução de um segundo problema de filtragem estocástica:

$$\begin{cases} \theta_{n+1} &= \phi_n(\theta_n) + \epsilon_n \\ y_n &= g_n(\theta_n, X_n) + v_n \end{cases} \quad (2.31)$$

Como é bem conhecido [14], o problema de (2.31) só admite solução fechada nos casos em que ϕ_n e g_n são lineares e ϵ_n e v_n são processos gaussianos. Neste caso $p(\theta_n|X_{0:n-1}, y_{0:n-1})$ é gaussiana e pode ser calculada recursivamente através de um filtro de Kalman, podendo também a integral de (2.27) ser calculada de forma trivial¹².

2.5.2 Evolução Artificial

Os métodos de evolução artificial [9] são talvez a ferramenta mais geral para a solução de problemas de estimação com parâmetros desconhecidos. Estas técnicas consistem basicamente em incluir os parâmetros desconhecidos no conjunto de variáveis de estado a ser estimado, passando o objetivo do filtro de partículas a ser aproximar $p(X_{0:n}, \theta_{0:n}|y_{0:n})$. Num paralelo ao feito na Sec. 2.1, pode-se deduzir o equacionamento básico de um filtro de partículas com este objetivo a partir da observação de que, pela lei de Bayes

$$p(X_{0:n}, \theta_{0:n}|y_{0:n}) = \frac{p(X_{0:n-1}, \theta_{0:n-1}|y_{0:n-1})p(X_n, y_n, \theta_n|X_{0:n-1}, \theta_{0:n-1}, y_{0:n-1})}{p(y_n|y_{0:n-1})}, \quad (2.32)$$

e, definindo a função de importância como

$$\pi(X_{0:n}, \theta_{0:n}|y_{0:n}) = \pi(X_{0:n-1}, \theta_{0:n-1}|y_{0:n-1})\pi(X_n, \theta_n|X_{0:n-1}, y_{0:n-1}, \theta_{0:n-1}), \quad (2.33)$$

chega-se à conclusão de que, ao amostrar

$$\{X_n, \theta_n\} \sim \pi(X_n, \theta_n|X_{0:n-1}, y_{0:n-1}, \theta_{0:n-1}), \quad (2.34)$$

¹²No Apêndice D demonstramos este cálculo para um modelo de sinal particular, empregado na equalização cega.

os pesos correspondentes podem ser atualizados como

$$w_{n+1} \propto w_n \frac{p(X_n, y_n, \theta_n | X_{0:n-1}, \theta_{0:n-1}, y_{0:n-1})}{\pi(X_n, \theta_n | X_{0:n-1}, y_{0:n-1}, \theta_{0:n-1})} . \quad (2.35)$$

A densidade conjunta necessária à atualização dos pesos em (2.35) pode (em geral) ser reescrita como

$$\begin{aligned} p(X_n, y_n, \theta_n | X_{0:n-1}, \theta_{0:n-1}, y_{0:n-1}) &= p(y_n | X_{0:n}, \theta_{0:n}, y_{0:n-1}) \\ & p(X_n | X_{0:n-1}, \theta_{0:n}, y_{0:n-1}) p(\theta_n | X_{0:n-1}, \theta_{0:n-1}, y_{0:n-1}) . \end{aligned} \quad (2.36)$$

Explorando propriedades de independência condicional¹³ induzidas pelo modelo de (2.24), obtemos então que

$$p(X_n, y_n, \theta_n | X_{0:n-1}, \theta_{0:n-1}, y_{0:n-1}) = p(y_n | X_n, \theta_n) p(X_n | X_{0:n-1}) p(\theta_n | \theta_{0:n-1}) . \quad (2.37)$$

Algoritmo empregando a função de importância *a priori*

O mais simples dos algoritmos baseados em evolução artificial pode ser então deduzido ao se empregar a função de importância *a priori*

$$\pi(X_n, \theta_n | X_{0:n-1}, y_{0:n-1}, \theta_{0:n-1}) = p(X_n | X_{0:n-1}) p(\theta_n | \theta_{0:n-1}) , \quad (2.38)$$

caso em que os pesos podem ser atualizados de acordo com a verossimilhança dos pares (X_n, θ_n) :

$$w_{n+1} = w_n p(y_n | X_n, \theta_n) . \quad (2.39)$$

Observação 3 *Observe que o princípio de evolução artificial não pode ser empregado em conjunto com o algoritmo SIR determinístico se a variável de estado estimada contiver elementos contínuos. Outro aspecto importante é que, se o modelo tratado*

¹³Pelo modelo (2.24), verifica-se facilmente que

$$p(y_n | X_{0:n}, \theta_{0:n}, y_{0:n-1}) = p(y_n | \theta_n, X_n) .$$

uma vez que y_n é independente de das demais variáveis dados θ_n e X_n . Por outro lado, observe que X_n independe de θ_n se y_n não for dado, implicando que:

$$p(X_n | X_{0:n-1}, \theta_{0:n}, y_{0:n-1}) = p(X_n | X_{0:n-1}, \theta_{0:n-1}, y_{0:n-1}) = p(X_n | X_{0:n-1}) .$$

De mesma forma, como θ_n independe das demais variáveis dado θ_{n-1} , segue finalmente que

$$p(\theta_n | X_{0:n-1}, \theta_{0:n-1}, y_{0:n-1}) = p(\theta_n | \theta_{n-1}) .$$

for invariante no tempo

$$p(\theta_n|\theta_{0:n-1}) = \mathcal{I}\{\theta_n = \theta_0\}p(\theta_0) , \quad (2.40)$$

resulta, como consequência de (2.37), que qualquer partícula para que $\theta_n \neq \theta_0$ terá seu peso anulado, e, conseqüentemente, será descartada no próximo passo de reamostragem. Este fato, que impede a adaptação do algoritmo, é usualmente contornado pela adoção de densidades de transição $p(\theta_n|\theta_{0:n-1})$ “suavizadas” para os parâmetros, usualmente densidades gaussianas com variância fixa

$$p(\theta_n|\theta_{0:n-1}) = \mathcal{N}(\theta_n|\theta_{n-1}; R) , \quad (2.41)$$

para $R > 0$, o que corresponde a alterar as premissas iniciais, adotando um modelo variante no tempo para os parâmetros estimados.

2.6 Exemplos de aplicação de filtros de partículas

2.6.1 Exemplo 1 - Modelo linear Autoregressivo Gaussiano

Para ilustrar a aplicação de filtros de partículas, vamos considerar o seguinte problema: deseja-se estimar X_n , um processo gaussiano autoregressivo observado através de y_n , definido como:

$$\begin{cases} X_{n+1} &= \alpha X_n + u_n \\ y_n &= X_n + v_n \end{cases} \quad (2.42)$$

em que $0 < \alpha < 1$, e u_n e v_n são dois processos gaussianos brancos e de média nula, com variância σ_u^2 e σ_v^2 respectivamente.

Neste exemplo, utilizamos a função de importância ótima $p(X_n|X_{0:n-1}, y_{0:n})$. Assim, para projetar um filtro de partículas para o problema em questão, precisamos obter as expressões:

1. da função de importância $p(X_n|X_{0:n-1}, y_{0:n})$.
2. da razão $p(X_n, y_n|X_{0:n-1}, y_{0:n-1})/p(X_n|X_{0:n-1}, y_{0:n})$ (necessária para atualizar os pesos (2.8)).

Cálculo da função de importância

Inicialmente, observe que como consequência da lei de Bayes podemos escrever:

$$p(X_n, y_n|X_{0:n-1}, y_{0:n-1}) = p(y_n|X_{0:n}, y_{0:n-1})p(X_n|X_{0:n-1}, y_{0:n-1}) . \quad (2.43)$$

Analisando (2.42), podemos concluir que

$$p(y_n|X_{0:n}, y_{0:n-1}) = p(y_n|X_n) = \mathcal{N}(y_n|X_n; \sigma_v^2),$$

uma vez que $y_n \triangleq X_n + v_n$. Da mesma forma, obtemos que

$$p(X_n|X_{0:n-1}, y_{0:n-1}) = p(X_n|X_{n-1}) = \mathcal{N}(X_n|\alpha X_{n-1}; \sigma_u^2),$$

do que concluimos que:

$$p(X_n, y_n|X_{0:n-1}, y_{0:n-1}) = \mathcal{N}(X_n|\alpha X_{n-1}; \sigma_u^2) \mathcal{N}(y_n|X_n; \sigma_v^2). \quad (2.44)$$

Por outro lado, observe agora que:

$$p(X_n|X_{0:n-1}, y_{0:n}) = \frac{p(X_n, y_n|X_{0:n-1}, y_{0:n-1})}{p(y_n|X_{0:n-1}, y_{0:n-1})} = \frac{p(X_n, y_n|X_{0:n-1}, y_{0:n-1})}{\int dX_n p(X_n, y_n|X_{0:n-1}, y_{0:n-1})}. \quad (2.45)$$

Combinando (2.44) e (2.45) resulta:

$$p(X_n|X_{0:n-1}, y_{0:n}) = \frac{\mathcal{N}(X_n|\alpha X_{n-1}; \sigma_u^2) \mathcal{N}(y_n|X_n; \sigma_v^2)}{\int dX_n \mathcal{N}(X_n|\alpha X_{n-1}; \sigma_u^2) \mathcal{N}(y_n|X_n; \sigma_v^2)}. \quad (2.46)$$

Após cálculos tediosos (vide Apêndice D), pode-se mostrar que:

$$\begin{aligned} \mathcal{N}(X_n|\alpha X_{n-1}; \sigma_u^2) \mathcal{N}(y_n|X_n; \sigma_v^2) = \\ \mathcal{N}\left(X_n \left| \frac{y_n \sigma_u^2 + \alpha X_{n-1} \sigma_v^2}{\sigma_u^2 + \sigma_v^2}; \frac{\sigma_u^2 \sigma_v^2}{\sigma_u^2 + \sigma_v^2}\right.\right) \mathcal{N}(y_n|\alpha X_{n-1}; \sigma_u^2 + \sigma_v^2). \end{aligned} \quad (2.47)$$

Utilizando (2.47), pode-se verificar que a integral do denominador de (2.46) é dada por $\mathcal{N}(y_n|\alpha X_{n-1}; \sigma_u^2 + \sigma_v^2)$ (já que o termo dependente de X_n tem integral unitária), resultando que:

$$p(X_n|X_{0:n-1}, y_{0:n}) = \mathcal{N}\left(X_n \left| \frac{y_n \sigma_u^2 + \alpha X_{n-1} \sigma_v^2}{\sigma_u^2 + \sigma_v^2}; \frac{\sigma_u^2 \sigma_v^2}{\sigma_u^2 + \sigma_v^2}\right.\right) \quad (2.48)$$

Cálculo da função de atualização dos pesos

Inicialmente, note que a razão de densidades presente em (2.8) pode ser reescrita como

$$\frac{p(X_n, y_n|X_{0:n-1}, y_{0:n-1})}{p(X_n|X_{0:n-1}, y_{0:n})} = p(y_n|X_{0:n-1}, y_{0:n-1}), \quad (2.49)$$

cuja expressão já foi obtida anteriormente, valendo

$$p(y_n | X_{0:n-1}, y_{0:n-1}) = \mathcal{N}(y_n | \alpha X_{n-1}; \sigma_u^2 + \sigma_v^2). \quad (2.50)$$

Algoritmo

A adaptação do filtro de partículas descrito na Tabela 2.2 ao modelo de (2.42) resulta então no seguinte algoritmo:

- Inicialize o algoritmo como descrito na Tabela 2.2.
- Amostre cada partícula fazendo

$$X_n^{(i)} \sim \mathcal{N}\left(X_n \mid \frac{y_n \sigma_u^2 + \alpha X_{n-1}^{(i)} \sigma_v^2}{\sigma_u^2 + \sigma_v^2}; \frac{\sigma_u^2 \sigma_v^2}{\sigma_u^2 + \sigma_v^2}\right). \quad (2.51)$$

- Atualize os pesos por

$$w_n^{(i)} = w_{n-1}^{(i)} \mathcal{N}(y_n | \alpha X_{n-1}^{(i)}; \sigma_u^2 + \sigma_v^2). \quad (2.52)$$

- Normalize os pesos, e estime $p(X_{0:n} | y_{0:n})$ conforme a Tabela 2.2.

Observação 4 *Pode-se estimar $X_{0:n}$ para o modelo descrito em (2.42) de forma ótima através de um filtro de Kalman. De fato, pode-se verificar que X_n tem uma distribuição gaussiana da forma $X_n \sim \mathcal{N}(X_n | \hat{X}_n; P_n)$, em que \hat{X}_n e P_n podem ser calculados através das recursões:*

$$\begin{aligned} \hat{X}_n &= \alpha \hat{X}_{n-1} + \frac{P_{n-1}}{P_{n-1} + \sigma_v^2} (y_n - \alpha \hat{X}_{n-1}) \\ P_n &= \alpha^2 \frac{P_{n-1} \sigma_v^2}{P_{n-1} + \sigma_v^2} + \sigma_u^2 \end{aligned} \quad (2.53)$$

iniciadas com $\hat{X}_0 = E[X_n] = 0$ e $P_0 = \sigma_u^2$.

Simulação

Geramos uma seqüência de 100 amostras de X_n e y_n de acordo com o modelo (2.42), utilizando $\alpha = 0,999$, $\sigma_u^2 = 0,1$ e $\sigma_v^2 = 0,5$. Em seguida, estimamos (a média de) $x_{0:100}$ a partir de $y_{0:100}$ utilizando um filtro de Kalman e um filtro de partículas (empregando $M = 100$ partículas, sem reamostragem). Na Figura 2.1 mostramos os resultados obtidos. Como se pode observar, ambas as estimativas seguem o sinal original com precisão razoável. Observou-se, no entanto, que variância do erro de estimação do filtro de partículas neste experimento (não mostrada no gráfico) foi igual

a 0,16, maior que o observado para o filtro de Kalman (0,14), o que é justificável pelo fato do filtro de Kalman ser ótimo para este modelo de sinal.

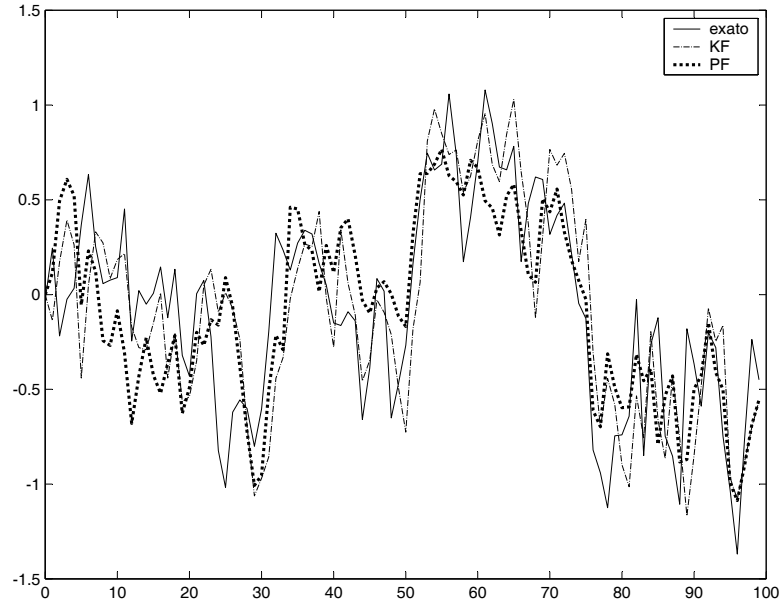


Figura 2.1: Gráfico de uma realização de $X_{0:n}$ gerado pelo modelo dado em (2.42) e de suas estimativas através de um filtro de Kalman (KF) e de um filtro de partículas (PF).

2.6.2 Exemplo 2 - Modelo autoregressivo com função de observação não-linear

Vamos considerar agora um modelo com evolução de estado semelhante à do exemplo anterior, mas observado através de uma função não-linear:

$$\begin{cases} X_{n+1} = \alpha X_n + u_n \\ y_n = (X_n)^{\frac{1}{3}} + v_n \end{cases} \quad (2.54)$$

em que $0 < \alpha < 1$, e u_n e v_n são dois processos gaussianos brancos e de média nula, com variância σ_u^2 e σ_v^2 respectivamente.

Observando-se o desenvolvimento do exemplo anterior, pode-se verificar que a função de observação não-linear faz com que não seja possível determinar analiticamente a função de importância ótima (2.47). Pode-se, no entanto, empregar a função de importância *a priori*, resultando no seguinte algoritmo:

- Inicialize o algoritmo como descrito na Tabela 2.2.

- Amostre cada partícula fazendo

$$X_n^{(i)} \sim \mathcal{N}\left(X_n | \alpha X_{n-1}^{(i)}; \sigma_u^2\right). \quad (2.55)$$

- Atualize os pesos por

$$w_n^{(i)} = w_{n-1}^{(i)} \mathcal{N}\left(y_n | (X_n^{(i)})^{(1/3)}; \sigma_v^2\right). \quad (2.56)$$

- Normalize os pesos, e estime $p(X_{0:n} | y_{0:n})$ conforme a Tabela 2.2.

Observação 5 Dada a não-linearidade do modelo descrito em (2.54), a estimação da densidade a posteriori de $X_{0:n}$ não pode ser calculada através de um filtro de Kalman como no exemplo anterior. Pode-se, no entanto, estimar $X_{0:n}$ de forma subótima empregando uma das diversas versões linearizadas do filtro de Kalman descritas na literatura (filtro de Kalman estendido (EKF) [15], filtro de Kalman baseado na “unscented transform” [32]). Empregando o EKF, resulta que estimativas da média \hat{X}_n e da variância P_n da variável de estado podem ser obtidas através das recursões:

$$\begin{aligned} H_n &= \frac{1}{3} \hat{X}_{n-1}^{-\frac{2}{3}} \\ P_{n|n-1} &= \alpha^2 P_{n-1} + \sigma_u^2 \\ \gamma_n &= H_n^H P_{n|n-1} H_n + \sigma_v^2 \\ \hat{X}_n &= \alpha \hat{X}_{n-1} + \gamma_n^{-1} [y_n - (\alpha \hat{X}_{n-1})^{\frac{1}{3}}] P_{n|n-1} H_n \\ P_n &= P_{n|n-1} - \gamma_n^{-1} P_{n|n-1} H_n H_n^H P_{n|n-1} \end{aligned} \quad (2.57)$$

iniciadas com $\hat{X}_0 = E[X_n] = 0$ e $P_0 = \sigma_u^2$.

2.6.3 Simulação

Geramos uma seqüência de 100 amostras de X_n e y_n de acordo com o modelo (2.54), com $\alpha = 0,999$, $\sigma_u^2 = 0,1$ e $\sigma_v^2 = 1$. Em seguida, estimamos (a média de) $x_{0:100}$ a partir de $y_{0:100}$ utilizando um filtro de Kalman estendido (EKF) e o filtro de partículas descrito na Sec. 2.6.2 (empregando $M = 150$ partículas, com reamostragem multinomial acionada sempre que o tamanho efetivo da amostra cai abaixo de 50%). A Figura 2.1 ilustra os resultados obtidos. Como se pode observar, ambas as estimativas seguem o sinal original com precisão razoável. Neste caso, ao contrário do exemplo anterior, as estimativas produzidas pelo filtro de partículas foram mais precisas: a variância média do erro de estimação (ao longo de 100 realizações) foi igual a 0,4806, contra um valor

médio de 0,6759 para o filtro de Kalman estendido.

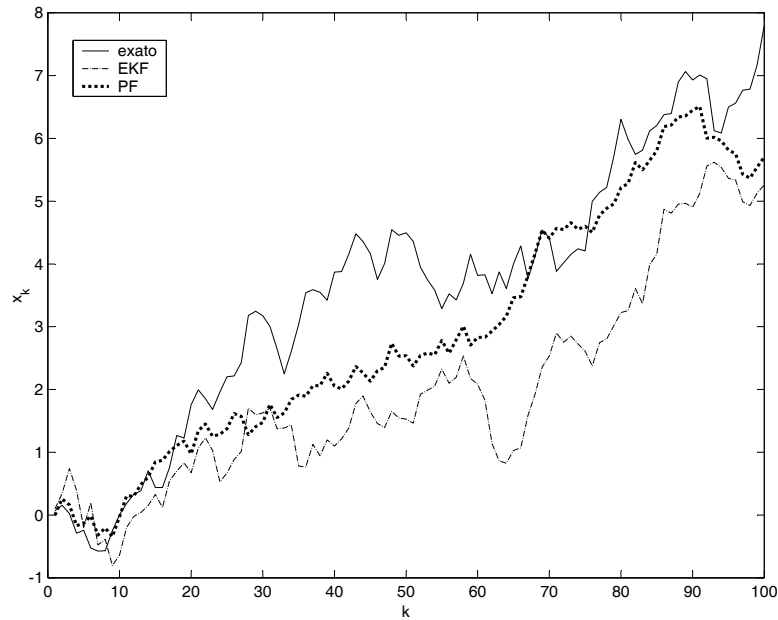


Figura 2.2: Gráfico de uma realização de $X_{0:n}$ gerado pelo modelo dado em (2.54) e de suas estimativas através de um filtro de Kalman estendido (EKF) e de um filtro de partículas (PF).

Capítulo 3

Filtro de Partículas para Equalização Cega

Neste capítulo, empregamos filtros de partículas na solução do problema de equalização cega de canais lineares FIR invariantes no tempo. Este problema, que talvez constitua uma das formulações mais imediatas de filtragem estocástica com parâmetros desconhecidos, é afetado intrinsecamente por ambigüidades de fase, que impossibilitam a operação dos filtros de partículas caso providências para estabelecer coerência de fase (entre as partículas) não sejam tomadas.

Desta forma, após descrever o modelo de sinal adotado (Sec. 3.1) e deduzir o equacionamento básico dos filtros de partículas Rao-Blackwellizados (Sec. 3.2), propomos na Sec. 3.3 três diferentes métodos para contornar os efeitos da ambigüidade de fase. A Sec. 3.4, por sua vez, descreve métodos para equalização cega baseados em evolução artificial, os quais, diferentemente dos algoritmos empregando Rao-Blackwellização, são capazes de operar sob ruídos aditivos não-gaussianos. Finalizando este capítulo, a Sec. 3.5 reúne o resultado de simulações numéricas nas quais o desempenho dos métodos propostos é avaliado.

3.1 Modelo de sinal, hipóteses e objetivos

Neste capítulo, desenvolvemos filtros de partículas para a equalização de canais de ordem conhecida e invariantes no tempo. Retomando a notação apresentada na Sec. 1.1, adotamos que as amostras y_n do sinal recebido em banda base podem ser descritas como

$$y_n = \sum_{i=0}^{L-1} h_i x_{n-i} + v_n , \quad (3.1)$$

em que h_i é a resposta ao impulso do canal de transmissão (de ordem L) e v_n o ruído aditivo (assumido gaussiano, branco, de média nula e variância σ_v^2 conhecida, salvo menção em contrário). Nosso objetivo é aproximar a densidade *a posteriori* $p(x_{0:n}|y_{0:n})$ através de filtros de partículas.

Como argumentamos no Cap. 2, a aplicação de filtros de partículas na solução de problemas de estimação requer que se seja capaz de expressar o sinal observado y_n como a saída de um sistema dinâmico, cuja variável de estado X_n seja preferencialmente uma função simples das variáveis que se desejam estimar. Definindo

$$X_n \triangleq [x_n \dots x_{n-L+1}]^T \in \mathbb{R}^{(L \times 1)},$$

obtemos que o sinal observado y_n pode ser descrito como

$$\begin{cases} X_{n+1} &= F X_n + f x_{n+1} \\ y_n &= h^H X_n + v_n \end{cases}, \quad (3.2)$$

em que $h \triangleq [h_0 \dots h_{L-1}]^T \in \mathbb{R}^{(L \times 1)}$ são os termos da resposta ao impulso do canal, assumidos invariantes no tempo e gaussianos *a priori*, tal que $h \sim \mathcal{N}(h|h_0; \Sigma_0)$ e F e f são descritos na Sec. 1.1.

Observação 1 *Conforme mostramos no Cap. 2, um filtro de partículas gera aproximações para $p(X_{0:n}|y_{0:n})$, a densidade da seqüência de estados dado o sinal observado. Observe no entanto que, para o modelo dado em (3.2),*

$$p(X_{0:n}|y_{0:n}) = p(x_{-L+1:-1})p(x_{0:n}|y_{0:n}) \prod_{m=1}^n \mathcal{I}\{X_{m+1} = F X_m + f x_{m+1}\}. \quad (3.3)$$

ou seja, caso $X_{0:n}$ relacione-se a $x_{-L+1:n}$ de acordo com o modelo dado em (3.2) e $p(x_{-L+1:-1})$ seja uniforme, temos que

$$p(X_{0:n}|y_{0:n}) = p(x_{0:n}|y_{0:n}).$$

3.2 Filtro de Partículas Rao-Blackwellizado

Nesta seção, deduzimos as densidades necessárias ao funcionamento de filtros de partículas Rao-Blackwellizados (Sec. 2.5.1) para o modelo de sinal de (3.2) com canais h de ordem L conhecida. Como mostrado na Sec. 2.4, para projetarmos filtros de partículas para a aproximação de $p(X_{0:n}|y_{0:n})$, devemos¹ obter a expressão de

$p(X_n, y_n | X_{0:n-1}, y_{0:n-1})$, que deduzimos a seguir.

Para isto, note inicialmente que por definição podemos escrever:

$$p(X_n, y_n | X_{0:n-1}, y_{0:n-1}) = p(y_n | X_{0:n}, y_{0:n-1}) p(X_n | X_{0:n-1}, y_{0:n-1}) . \quad (3.4)$$

Explorando o modelo de sinal e utilizando argumentos semelhantes aos de (3.3) obtemos que o segundo termo do lado direito de (3.4) é dado por

$$p(X_n | X_{0:n-1}, y_{0:n-1}) = p(X_n | X_{0:n-1}) = p(x_n | x_{0:n-1}) . \quad (3.5)$$

O cálculo do primeiro termo do lado direito de (3.4) envolve manipulações algébricas bastante extensas, mostradas em detalhes no Apêndice D². Apresentamos aqui uma versão resumida deste cálculo, que se inicia pela observação de que

$$p(y_n | X_{0:n}, y_{0:n-1}) = \int dh p(y_n, h | X_{0:n}, y_{0:n-1}) . \quad (3.6)$$

O integrando de (3.6) pode, por sua vez, ser reescrito como

$$p(y_n, h | X_{0:n}, y_{0:n-1}) = p(y_n | h, X_{0:n}, y_{0:n-1}) p(h | X_{0:n}, y_{0:n-1}) . \quad (3.7)$$

O primeiro termo do lado direito de (3.7) não é nada mais do que a verossimilhança das observações (Sec. 2.5.1), sendo possível mostrar³ que:

$$p(y_n | h, X_{0:n}, y_{0:n-1}) = p(y_n | h, X_n) = \mathcal{N}(y_n | h^H X_n; \sigma_v^2) . \quad (3.8)$$

A obtenção de uma expressão para o segundo termo do lado direito de (3.7) é bastante mais complexa. Para isto exploramos inicialmente o fato de que

$$p(h | X_{0:n}, y_{0:n-1}) = p(h | X_{0:n-1}, y_{0:n-1}) . \quad (3.9)$$

¹A partir desta densidade pode-se obter a função de importância ótima

$$p(X_n | X_{0:n-1}, y_{0:n}) \propto p(X_n, y_n | X_{0:n-1}, y_{0:n-1})$$

e a densidade

$$p(y_n | X_{0:n-1}, y_{0:n-1}) = \int_{X_n} dX_n p(X_n, y_n | X_{0:n-1}, y_{0:n-1}) ,$$

necessária para a atualização dos pesos.

²A notação empregada no Apêndice D difere ligeiramente da deste capítulo, pois lá deduzimos as expressões para o caso mais geral em que os parâmetros do canal são variantes no tempo e o sinal transmitido (bem como o ruído aditivo) complexos.

³Apesar dos sinais tratados neste capítulo serem (por hipótese) reais, as expressões deduzidas a seguir também são adequadas para o caso em que estes sinais sejam complexos.

Como $y_n \triangleq h^H X_n + v_n$, conclui-se que, caso h tenha uma densidade *a priori* gaussiana da forma $h \sim \mathcal{N}(h|\tilde{h}_0; \tilde{\Sigma}_0)$, a densidade *a posteriori* $p(h|X_{0:n-1}, y_{0:n-1})$ também será gaussiana, podendo ser calculada através de um filtro de Kalman, projetado para o seguinte modelo:

$$\begin{cases} h & = h \\ y_n & = h^H X_n + v_n \end{cases} . \quad (3.10)$$

Projetando o filtro de Kalman para o modelo dado em (3.10), obtemos que

$$p(h|X_{0:n-1}, y_{0:n-1}) = \mathcal{N}(h|\tilde{h}_{n-1}; \tilde{\Sigma}_{n-1}),$$

em que:

$$\tilde{h}_n = \tilde{h}_{n-1} + \frac{y_n - \tilde{h}_{n-1}^H X_n}{X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2} \tilde{\Sigma}_{n-1} X_n \quad (3.11)$$

$$\tilde{\Sigma}_n = \tilde{\Sigma}_{n-1} - \frac{\tilde{\Sigma}_{n-1} X_n X_n^H \tilde{\Sigma}_{n-1}}{X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2} \quad (3.12)$$

Substituindo os resultados obtidos em (3.7), obtemos:

$$p(y_n|X_{0:n}, y_{0:n-1}) = \int dh \mathcal{N}(h|\tilde{h}_{n-1}; \tilde{\Sigma}_{n-1}) \mathcal{N}(y_n|h^H X_n; \sigma_v^2) . \quad (3.13)$$

Observe que, para poder calcular a integral em (3.13), devemos manipular as densidades gaussianas de forma a isolar os termos dependentes de h . Após manipulações algébricas bastante longas, resulta que:

$$\begin{aligned} \mathcal{N}(h|\tilde{h}_{n-1}; \tilde{\Sigma}_{n-1}) \mathcal{N}(y_n|h^H X_n; \sigma_v^2) = \\ \mathcal{N}(h|\tilde{h}_n; \tilde{\Sigma}_n) \mathcal{N}(y_n|\tilde{h}_{n-1}^H X_n; \sigma_v^2 + X_n^H \tilde{\Sigma}_{n-1} X_n) , \end{aligned} \quad (3.14)$$

o que, substituído em (3.13), resulta em:

$$p(y_n|X_{0:n}, y_{0:n-1}) = \mathcal{N}(y_n|\tilde{h}_{n-1}^H X_n; \sigma_v^2 + X_n^H \tilde{\Sigma}_{n-1} X_n) . \quad (3.15)$$

Substituindo (3.15) em (3.4), obtemos finalmente que:

$$p(X_n, y_n|X_{0:n-1}, y_{0:n-1}) = \mathcal{N}(y_n|\tilde{h}_{n-1}^H X_n; \sigma_v^2 + X_n^H \tilde{\Sigma}_{n-1} X_n) p(x_n|x_{0:n-1}) . \quad (3.16)$$

Apesar de, em princípio, o resultado de (3.16) permitir o uso de qualquer uma das versões do algoritmo SIR (Sec. 2.4) para a estimação de $x_{0:n}$, a ambigüidade⁴ de fase inerente ao problema da equalização cega destrói a aproximação $p_N(x_{0:n}|y_{0:n})$ ao fazer com que cada partícula seja afetada por uma rotação de fase arbitrária $\alpha^{(i)}$.

Observação 2 *Na prática, devido à replicação de partículas decorrente da reamostragem, o conjunto de valores assumidos pelas rotações de fase $\alpha^{(i)}$ não é uniformemente distribuído (como seriam caso as partículas evoluíssem independentemente). Isto termina permitindo a obtenção de aproximações (potencialmente viesadas) de $p(x_{0:n}|y_{0:n})$, mesmo ao se desconsiderar os efeitos da ambigüidade de fase.*

3.3 Métodos para a resolução do problema da ambigüidade de fase

3.3.1 Resolução da ambigüidade de fase através de codificação diferencial

Um dos métodos mais empregados para contornar o problema da ambigüidade de fase em sistemas de comunicação incoerentes é o uso de codificação diferencial. Seja $b_n = \pm 1$ a mensagem binária⁵ que se deseja transmitir. O sinal efetivamente transmitido num sistema de codificação diferencial é definido como:

$$x_n \triangleq x_{n-1}b_n, \quad n \geq 0, \quad (3.17)$$

com $x_{-1} \triangleq 1$. A mensagem pode então ser recuperada no receptor através da relação

$$b_n = x_n x_{n-1}. \quad (3.18)$$

Pelas relações anteriores, a seqüência transmitida $x_{-1:n}$ define unicamente a seqüência $b_{0:n}$, o que, pelos mesmos argumentos de (3.3), implica que

$$p(b_{0:n}^{(i)}|y_{0:n}) = p(x_{0:n}^{(i)}|y_{0:n}),$$

⁴Por ambigüidade de fase, entende-se o fato de que se o sinal transmitido x_n e os parâmetros do canal h forem tais que exista uma constante de módulo unitário $\alpha \neq 1$ que satisfaça $p(\alpha X_n) = p(X_n)$ e $p(\alpha h) = p(h)$, $\forall x_n, h$, então $p(X_n|\alpha h, y_{0:n}) = p(\alpha X_n|h, y_{0:n})$, ou seja, não se pode determinar unicamente a fase de X_n caso h seja desconhecido.

⁵O uso de técnicas de codificação diferencial não é restrito à sistemas de comunicações binários, uma vez que avanços recentes na teoria da informação permitiram o desenvolvimentos de códigos diferenciais para sistemas de comunicação M -ários [33].

podendo a densidade $p(b_{0:n}^{(i)}|y_{0:n})$ ser estimada através do algoritmo da Tabela⁶ 3.1.

<p>(Inicialização) Para $i = 1 : N$ $X_0^{(i)} \sim \mathcal{B}_L(\pm 1)$ $h_0^{(i)} \sim \mathcal{N}(h h_0; \Sigma_0)$ Fim (Algoritmo) Para $n > 0$ 1) Para $i = 1 : N$ A) Amostre $X_n^{(i)} \sim p(X_n X_{0:n-1}^{(i)}, y_{0:n})$ B) Calcule e normalize os pesos $w_n^{(i)} = w_{n-1}^{(i)}p(y_n X_{0:n-1}^{(i)}, y_{0:n-1})$ C) Calcule $b_n^{(i)} = x_{n-1}^{(i)}x_n^{(i)}$ Fim 2) Reamostre o conjunto de partículas, conforme o critério adotado. 3) Estime a densidade <i>a posteriori</i> dos símbolos transmitidos como $p(b_n = B y_{0:n}) \approx N^{-1} \sum_{i=1}^N \mathcal{I}\{b_n^{(i)} = B\}$. Fim</p>
--

Tabela 3.1: Equalizador cego baseado no algoritmo SIR estocástico, empregando a função de importância ótima e codificação diferencial.

No algoritmo da Tab. 3.1, o passo (A) é certamente o que encerra a maior dificuldade. Assumindo que os símbolos binários transmitidos b_k sejam IID e equiprováveis, verifica-se inicialmente que $\mathbb{P}(x_n = \pm 1|x_{0:n-1}) = 1/2$. Assim, a cada instante n , a variável $X_n^{(i)}$ pode assumir dois valores distintos, com probabilidade proporcional a

$$p(X_n^{(i)}, y_n|X_{0:n-1}^{(i)}, y_{0:n-1}) \propto \mathcal{N}(y_n|\tilde{h}_{n-1}^{(i)H}X_n^{(i)}; \sigma_v^2 + X_n^{(i)H}\tilde{\Sigma}_{n-1}^{(i)}X_n^{(i)}),$$

sendo os parâmetros $\tilde{h}_{n-1}^{(i)}$ e $\tilde{\Sigma}_{n-1}^{(i)}$ determinados iterativamente através do filtro de Kalman de (3.12). Reunindo estes fatos, o passo (A) pode ser descrito mais detalhadamente como:

- Dado $X_{n-1}^{(i)}$, determine os dois valores possíveis para X_n .
- Calcule $\mathbb{P}(X_n) \propto p(X_n|X_{0:n-1})\mathcal{N}(y_n|\tilde{h}_{n-1}^{(i)H}X_n; \sigma_v^2 + X_n^H\tilde{\Sigma}_{n-1}^{(i)}X_n)$ para cada X_n possível e amostre da densidade discreta⁷ resultante, obtendo $X_n^{(i)}$.

⁶Embora o algoritmo descrito na Tabela 3.1 empregue o método SIR estocástico, o mesmo (bem como todos os demais algoritmos descritos nesta seção) poderia utilizar o método SIR determinístico.

⁷Note que a soma dos valores de $\mathbb{P}(X_n)$, que deve necessariamente ser calculada para a normalização da densidade discreta, é igual ao fator de atualização dos pesos (Passo B), como pode ser demonstrado através de (2.19).

- Uma vez escolhido $X_n^{(i)}$, atualize os parâmetros $\tilde{h}_{n-1}^{(i)}$ e $\tilde{\Sigma}_{n-1}^{(i)}$ através de (3.12).

3.3.2 Resolução da ambigüidade de fase por pivotamento⁸

Em vez de recorrer ao uso de codificação diferencial, uma segunda maneira de contornar o problema da ambigüidade de fase é observar que uma de suas causas - o fato dos parâmetros do canal terem uma distribuição *a priori* com simetrias em torno da origem - pode ser suprimida ao se impor que um dos elementos h_l do vetor h tenha fase num intervalo

$$-\vartheta/2 \leq \arg h_l < \vartheta/2, \quad (3.19)$$

em que ϑ é o mínimo ângulo em que a relação $p(X_n) = p(e^{j\vartheta} X_n), \forall X_n$ é satisfeita.

Sob a hipótese adotada nesta seção de que x_n é um sinal binário, (3.19) se reduz a impor que $h_l > 0$, para algum $0 \leq l < L$, escolhido heurísticamente. Explorando o fato de que os filtros de partículas Rao-Blackwellizados determinam os valores dos parâmetros do canal $h^{(i)}$ implicitamente a partir de $X_{0:n}^{(i)}$, propomos implementar a restrição de (3.19) de acordo com o procedimento descrito na Tab. 3.2.

3.3.3 Resolução da ambigüidade de fase através de codificação diferencial implícita

Caso o sinal transmitido seja binário, outra forma de contornar o problema da ambigüidade de fase vem da observação de que a rotação arbitrária que afeta cada partícula $\alpha^{(i)} \in \{\pm 1\}$ mantém-se aproximadamente constante⁹ caso a partícula em questão esteja próxima da moda da distribuição *a posteriori*.

Explorando este fato, verificamos que ao decodificar diferencialmente cada trajetória $x_{0:n}^{(i)}$ gerando as seqüências $b_{0:n}^{(i)}$, a ambigüidade de fase (entre as partículas) é resolvida, uma vez que:

$$b_n^{(i)} = (x_n^{(i)} \alpha^{(i)})(x_{n-1}^{(i)} \alpha^{(i)}) = x_n^{(i)} x_{n-1}^{(i)}. \quad (3.20)$$

A densidade *a posteriori* $p(b_{0:n}|y_{0:n})$ pode ser então aproximada a partir das amostras $b_{0:n}^{(i)}$ determinadas através de (3.20), levando a uma estimativa $\hat{b}_{0:n}$, a partir da qual

⁸Utilizamos o termo "pivotamento" (*pivoting*) para nos referir ao algoritmo descrito nesta seção devido à semelhança entre este método e determinados algoritmos empregados em cálculo numérico, que escolhem heurísticamente um elemento de um vetor ou matriz como referência para a normalização dos demais (em módulo, e não em fase como neste caso).

⁹Pelo fato de que a estimativa dos parâmetros do canal $h^{(i)}$ em geral não varia abruptamente caso seja próxima do valor verdadeiro h . De qualquer forma, partículas distantes das modas das distribuição *a posteriori* normalmente estão associadas a pesos baixos, não tendo grande impacto sobre as estimativas.

<p>(Inicialização)</p> <p>Para $i = 1 : N$</p> <p>$X_0^{(i)} \sim \mathcal{B}_L(\pm 1)$</p> <p>$h_0^{(i)} \sim \mathcal{N}(h h_0; \Sigma_0)$</p> <p>Fim</p> <p>(Algoritmo)</p> <p>Para $n > 0$</p> <p>1) Para $i = 1 : N$</p> <p>A) Amostre $X_n^{(i)} \sim p(X_n X_{0:n-1}^{(i)}, y_{0:n})$</p> <p>B) Calcule e normalize os pesos $w_n^{(i)} = w_{n-1}^{(i)} p(y_n X_{0:n-1}^{(i)}, y_{0:n-1})$</p> <p>C) Se $h_l^{(i)} < 0$, para algum $0 \leq l < L$ pré-determinado, faça $h_l^{(i)} = -h_l^{(i)}$ e $X_{0:n}^{(i)} = -X_{0:n}^{(i)}$</p> <p>Fim</p> <p>2) Reamostre o conjunto de partículas, conforme o critério adotado.</p> <p>3) Estime a densidade <i>a posteriori</i> dos símbolos transmitidos como</p> <p>$p(x_n = X y_{0:n}) \approx N^{-1} \sum_{i=1}^N \mathcal{I}\{x_n^{(i)} = X\}$.</p> <p>Fim</p>

Tabela 3.2: Equalizador cego baseado no algoritmo SIR estocástico, empregando a função de importância ótima e pivotamento.

uma estimativa da sequência transmitida $\hat{x}_{0:n}$ (e não de sua densidade) pode ser obtida codificando diferencialmente $\hat{b}_{0:n}$ através de (3.17), como ilustramos na Tab. 3.3.

3.4 Métodos de Evolução Artificial de Parâmetros

Apesar do modelo de sinal da Sec. 3.1 permitir o desenvolvimento de filtros de partículas Rao-Blackwellizados (que, em princípio, levariam a um desempenho superior ao de outros métodos que estimam diretamente os parâmetros do modelo), nesta seção¹⁰ desenvolvemos algoritmos para equalização cega baseados no princípio da evolução artificial de parâmetros, descrito na Sec. 2.5.2.

Aplicando diretamente os resultados de (2.38)-(2.39) ao modelo de sinal da Sec. 3.1, e adotando como núcleo de transição para os parâmetros desconhecidos h_n a densidade

$$p(h_n|h_{0:n-1}) = \mathcal{N}(h_n|h_{n-1}; I\rho^2), \quad (3.21)$$

obtemos o algoritmo descrito na Tab. 3.4. Diferentemente dos métodos Rao-Black-

¹⁰Embora os algoritmos aqui descritos possam operar de acordo com qualquer um métodos para a resolução do problema da ambigüidade de fase descritos na Sec. 3.3, limitamo-nos à descrição de algoritmos operando com sinais codificados diferencialmente.

<p>(Inicialização)</p> <p>Para $i = 1 : N$</p> <p>$X_0^{(i)} \sim \mathcal{B}_L(\pm 1)$</p> <p>$h_0^{(i)} \sim \mathcal{N}(h h_0; \Sigma_0)$</p> <p>Fim</p> <p>(Algoritmo)</p> <p>Para $n > 0$</p> <p>1) Para $i = 1 : N$</p> <p>A) Amostre $X_n^{(i)} \sim p(X_n X_{0:n-1}^{(i)}, y_{0:n})$</p> <p>B) Calcule e normalize os pesos $w_n^{(i)} = w_{n-1}^{(i)} p(y_n X_{0:n-1}^{(i)}, y_{0:n-1})$</p> <p>C) Calcule $b_n^{(i)} = x_{n-1}^{(i)} x_n^{(i)}$</p> <p>Fim</p> <p>2) Reamostre o conjunto de partículas, conforme o critério adotado.</p> <p>3) Estime a densidade <i>a posteriori</i></p> <p>$p(b_{0:n} y_{0:n}) \approx \sum_{i=1}^N w_n^{(i)} \delta(b_{0:n} - b_{0:n}^{(i)})$.</p> <p>4) Determine a estimativa MAP $\hat{b}_{0:n}$.</p> <p>5) Codifique diferencialmente $\hat{b}_{0:n}$, obtendo a estimativa $\hat{x}_{0:n}$.</p> <p>Fim</p>
--

Tabela 3.3: Equalizador cego baseado no algoritmo SIR estocástico, empregando a função de importância ótima e codificação diferencial implícita.

wellizados, este algoritmo é, em tese, capaz de operar sob ruídos com distribuições genéricas p_{v_n} , exibindo uma complexidade computacional por partícula bastante inferior à dos métodos descritos anteriormente.

Nas simulações numéricas realizadas na Sec. 3.5, verificamos que o algoritmo da Tab. 3.4 não produz resultados satisfatórios, não importando o número de partículas empregado. Após uma extensa análise de simulações numéricas, nas quais consideramos o uso de diversas funções de importância e de estruturas de filtragem de partículas diferentes do algoritmo SIR (algoritmo *Resample-Move*, *Auxiliary Particle Filter* [9]), concluímos que o melhor desempenho entre os algoritmos baseados em evolução artificial é obtido pelo uso do algoritmo SIR em conjunto com a função de importância *discreta*

$$\pi(X_n, h_n | X_{0:n-1}^{(i)}, h_{0:n-1}^{(i)}) \propto p_{v_n}(y_n - h_n^H X_n) p(X_n | X_{n-1}^{(i)}) \mathcal{I}\{h_n = h_n^{(i,r)}\}, \quad (3.22)$$

em que $h_n^{(i,j)}$ é um conjunto de P amostras independentes de $\mathcal{N}(h_n | h_{n-1}^{(i)}; I\rho^2/2)$, uma distribuição gaussiana com metade da variância do núcleo de transição dos parâmetros.

<p>(Inicialização)</p> <p>Para $i = 1 : N$</p> <p style="padding-left: 2em;">$X_0^{(i)} \sim \mathcal{B}_L(\pm 1).$</p> <p style="padding-left: 2em;">$h_0^{(i)} \sim \mathcal{N}(h_0 \bar{h}_0; \Sigma_0).$</p> <p>Fim</p> <p>(Algoritmo)</p> <p>Para $n > 0$</p> <p>1) Para $i = 1 : N$</p> <p style="padding-left: 2em;">A) Amostre $h_n^{(i)} \sim \mathcal{N}(h_n h_{n-1}^{(i)}; I\rho^2).$</p> <p style="padding-left: 2em;">B) Amostre $X_n^{(i)} \sim p(X_n X_{n-1}^{(i)}).$</p> <p style="padding-left: 2em;">C) Obtenha $b_n^{(i)}$ decodificando $x_{0:n}^{(i)}$</p> <p style="padding-left: 2em;">D) Calcule e normalize os pesos</p> <p style="padding-left: 4em;">$w_n^{(i)} \propto p_{v_n}(y_n - h_n^{(i)H} X_n^{(i)}).$</p> <p style="padding-left: 2em;">Fim</p> <p>2) Reamostre o conjunto de partículas de acordo com os pesos $w_n^{(i)}$.</p> <p>3) Estime $p(b_n)$ por</p> <p style="padding-left: 4em;">$p(b_n = B y_{0:n}) \approx N^{-1} \sum_{i=1}^N \mathcal{I}\{b_n^{(i)} = B\}.$</p> <p>Fim</p>

Tabela 3.4: Algoritmo para equalização cega sob ruído com distribuição genérica baseado no método de evolução artificial utilizando função de importância *a priori*.

Verifica-se facilmente que, neste caso, a função de atualização dos pesos é dada por

$$w_n \propto w_{n-1} \sum_{r,s} p_{v_n}(y_n - h_n^{(r)H} X_n^{(s)}) p(X_n^{(s)} | X_{n-1}^{(i)}) \mathcal{N}(h_n^{(r)} | h_{n-1}^{(i)}; I\rho^2/2). \quad (3.23)$$

sendo o algoritmo resultante mostrado na Tab. 3.5.

3.5 Simulações

Para avaliar o desempenho dos algoritmos descritos neste capítulo realizamos simulações numéricas, medindo as taxas médias de erro de bit (BER) obtidas ao longo de 250 realizações independentes. O sistema de comunicação analisado transmite blocos de 400 símbolos BPSK (± 1) (independentes e equiprováveis, codificados diferencialmente ou não dependendo do algoritmo empregado). Em todas as simulações, os 100 primeiros símbolos são descartados para fins de medição das taxas de erro de modo a permitir a convergência dos algoritmos analisados. Nas simulações a seguir, empregamos o canal $h = [0, 41 \ -0, 82 \ 0, 41]^T$ e ruído aditivo gaussiano branco (real) de média nula e variância σ_v^2 . Os filtros de partículas são inicializados de forma aleatória,

(Inicialização)
 Para $i = 1 : N$
 $h_0^{(i)} \sim \mathcal{N}(h_0 | \bar{h}_0; \Sigma_0)$.
 $X_0^{(i)} \sim p(X_0)$.
 Fim
 (Algoritmo)
 Para $n > 0$
 1) Para $i = 1 : N$,
 A) Para $r = 1 : P$, amostre
 $h_n^{(i,r)} \sim \mathcal{N}(h_n | h_{n-1}^{(i)}; I(\rho^2/2))$.
 Fim
 B) Amostre X_n e h_n conjuntamente da densidade discreta
 $\pi(X_n, h_n) \propto p_{v_n}(y_n - h_n^H X_n) p(X_n | X_{n-1}^{(i)}) \mathcal{I}\{h_n = h_n^{(i,r)}\}$,
 C) Calcule e normalize os pesos
 $w_n^{(i)} \propto \sum_{r,s} p_{v_n}(y_n - h_n^{(r)H} X_n^{(s)}) p(X_n^{(s)} | X_{n-1}^{(i)}) \mathcal{N}(h_n^{(r)} | h_{n-1}^{(i)}; I\rho^2/2)$.
 D) Obtenha $b_n^{(i)}$ decodificando $x_{0:n}^{(i)}$.
 Fim
 2) Reamostre o conjunto de partículas de acordo com os pesos $w_n^{(i)}$.
 3) Estime $p(b_n)$ por
 $p(b_n = B | y_{0:n}) \approx \frac{1}{N} \sum_{i=0}^{N-1} \mathcal{I}\{b_n^{(i)} = B\}$.
 Fim

Tabela 3.5: Algoritmo para equalização cega sob ruído com distribuição genérica baseado no método de evolução artificial utilizando função de importância modificada.

fazendo $h_0^{(i)} \sim \mathcal{N}(h|0; I)$, e a relação sinal-ruído (SNR) foi definida como

$$SNR \triangleq \frac{\|h\|^2}{\sigma_v^2}.$$

Na Figura 3.1 mostramos o desempenho médio dos algoritmos para equalização descritos nesta seção em função da relação sinal-ruído (SNR), empregando 100 partículas e limiar de reamostragem¹¹ $T_{MIN} = 1$ (i.e., forçando reamostragem a cada iteração). Para efeito de comparação, mostramos também as taxas médias de erro obtidas através do equalizador ótimo MAP (algoritmo BCJR¹²).

Como se pode observar na Figura 3.1, o algoritmo determinístico teve um desempenho bastante superior ao dos demais. Quanto à forma de combate aos efeitos da

¹¹Aplicável somente aos filtros estocásticos.

¹²O algoritmo BCJR, descrito no Apêndice F, determina a probabilidade *a posteriori* $p(x_{0:n} | y_{0:n})$ dado o bloco $y_{0:n}$ e os parâmetros do canal. Para permitir uma comparação justa, a BER média produzida pelo algoritmo BCJR foi estimada tomando os 300 elementos centrais de um bloco de 400 amostras do sinal decodificado, sendo a estimativa MAP de $x_{0:n}$ calculada e os bits transmitidos $b_{0:n}$ estimados decodificando-se diferencialmente esta seqüência.

ambigüidade de fase, podemos observar também que os métodos (determinísticos)¹³ empregando codificação diferencial e pivotamento exibiram praticamente o mesmo desempenho, enquanto o uso de codificação diferencial implícita levou a um desempenho bastante inferior.

Ao se aumentar o número de partículas empregado para $N = 300$ (Fig. 3.2) e $N = 500$ (Fig. 3.3), observa-se uma melhoria no desempenho dos métodos estocásticos e baseados em evolução artificial. Curiosamente, porém, o desempenho do método determinístico começa a se degradar quando o número de partículas ultrapassa $N = 300$, o que talvez possa ser explicado pela tendência do método de seleção de partículas determinístico de introduzir um viés crescente em função do número de partículas utilizado. Note ainda que a diferença de desempenho de aproximadamente 6 dB entre o método ótimo (BCJR) e o melhor dos algoritmos cegos persiste, mesmo com $N = 500$.

Nas Figuras 3.4 e 3.5, repetimos o mesmo experimento, empregando agora a técnica de suavização descrita na Sec. 2.4.2, com atraso de 5 e 10 amostras respectivamente e $N = 300$ partículas. Observando estas figuras, pode-se notar que o desempenho de todos os métodos é grandemente melhorado pelo uso da técnica de suavização (5 amostras), o que faz com que o desempenho do método determinístico praticamente coincida com o ótimo para SNR intermediárias (persistindo, porém, uma perda de desempenho de aproximadamente 2 dB sob baixos níveis de ruído). Note, porém, que ao se aumentar o atraso para 10 amostras, o desempenho de todos os algoritmos (exceto os de evolução artificial) sofre degradações, especialmente sob baixos níveis de ruído, o que é de certa forma esperado, uma vez que esta combinação de fatores exacerba o problema do empobrecimento amostral. Vale mencionar que testamos o uso de técnicas alternativas de filtragem de partículas (algoritmo *Resample-Move* [9]) para combater o empobrecimento amostral nestas situações, sem obter resultados satisfatórios.

Na Figura 3.6 mostramos o resultado de um experimento distinto, no qual avaliamos o desempenho dos algoritmos baseados em evolução artificial (EA) (com $P = 1$) em função do número de partículas empregado e da SNR. Nestas simulações (e nas simulações a seguir), adotamos o valor $\delta^2 = 0,0125$ para a variância do núcleo de transição (Eq. 3.21), por experimentalmente ter levado aos melhores resultados. Como se pode observar, o desempenho das técnicas EA (sem suavização) é bastante inferior ao do algoritmo determinístico (Rao-Blackwellizado), tendendo a este somente quando o número de partículas empregado torna-se muito superior, situação que se repete na Fig. 3.7, em que mostramos os resultados obtidos para o caso em que $P = 5$.

¹³Estes resultados também foram verificados para os métodos empregando o algoritmo SIR estocástico e para o método baseado em evolução artificial, e não foram incluídos nos gráficos para facilitar as suas leituras.

Nas Figuras 3.8 e 3.9, avaliamos os efeitos da variação do parâmetro δ^2 (variância do núcleo de transição) sobre o desempenho dos algoritmos baseados em evolução artificial (EA), para $P = 1$ e $P = 5$, respectivamente, empregando $N = 1000$ partículas e um atraso de suavização de 10 amostras. Como se pode observar, o valor de δ^2 escolhido estabelece um bom compromisso, uma vez que valores inferiores a este tendem a degradar o desempenho observado sob baixos níveis de ruído, enquanto valores superiores produzem uma degradação em toda a faixa de SNR analisada.

Finalmente, nas Figuras (3.10)-(3.12) analisamos os efeitos conjuntos da variação do número de partículas e do limiar de reamostragem sobre o desempenho dos algoritmos baseados em filtros de partículas estocásticos. Na Figura 3.10 mostramos o desempenho obtido pelo método estocástico empregando a função de importância ótima, enquanto na Fig. 3.11, o resultado para a função de importância *a priori*. Como se pode observar, o efeito da diminuição do limiar de reamostragem é pequeno, sendo percebido com mais intensidade quando o número de partículas é menor.

Na Fig. 3.12, repetimos o mesmo experimento com $N = 300$ partículas e suavização (10 amostras). Neste caso, a diminuição do limiar de reamostragem produziu resultados benéficos, uma vez que algoritmos empregando suavização são mais sensíveis ao empobrecimento amostral causado pela reamostragem freqüente do conjunto de partículas.

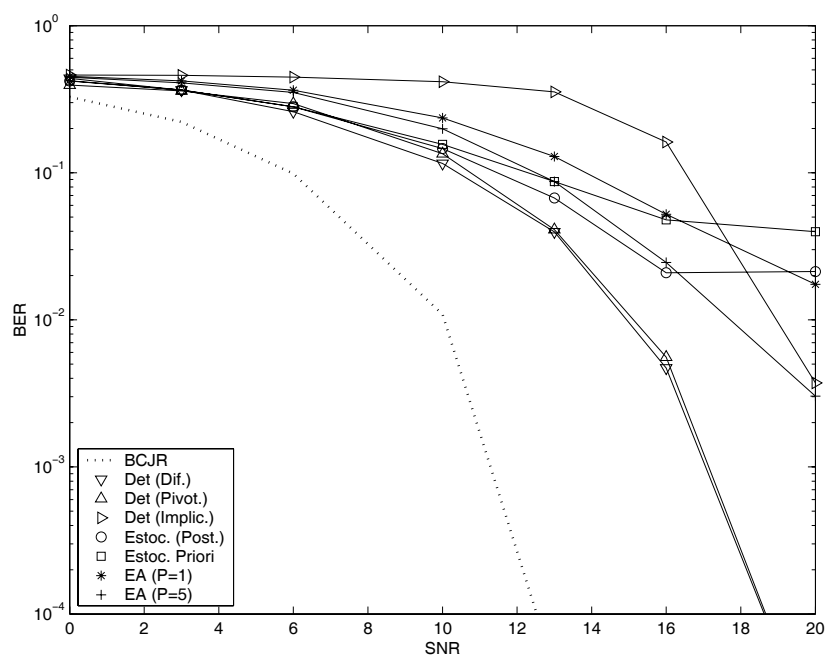


Figura 3.1: Desempenho médio ao longo de 250 realizações independentes dos algoritmos para equalização cega baseados em filtros de partículas descritos neste capítulo, num sistema BPSK (empregando $N = 100$ partículas) e do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR). Os algoritmos estocásticos e baseados em evolução artificial (EA) empregam codificação diferencial.

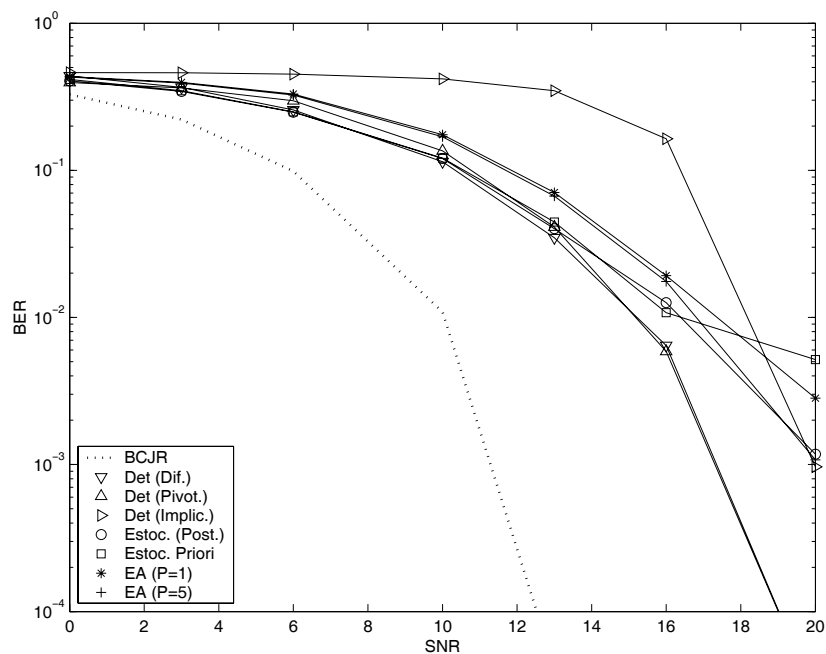


Figura 3.2: Desempenho médio ao longo de 250 realizações independentes dos algoritmos para equalização cega baseados em filtros de partículas descritos neste capítulo, num sistema BPSK (empregando $N = 300$ partículas) e do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR).

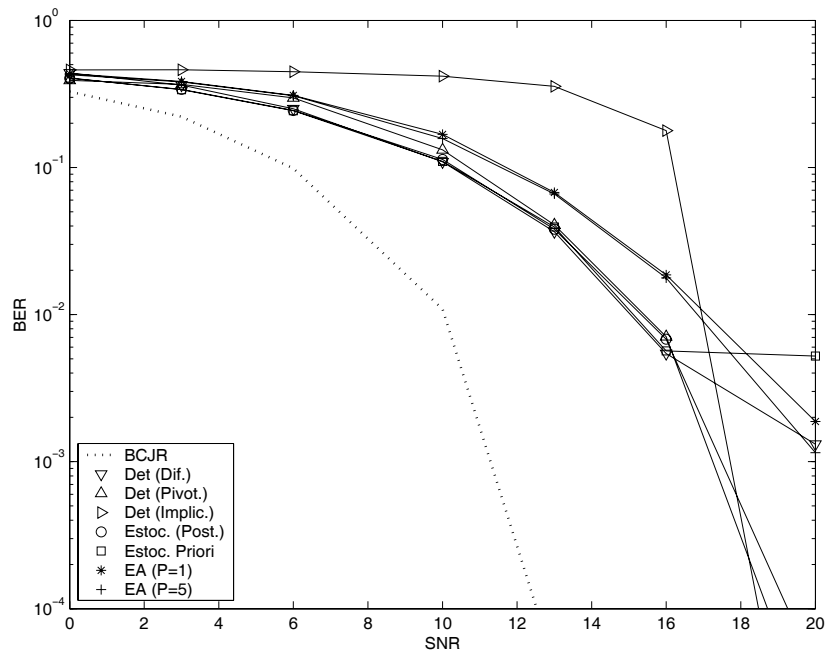


Figura 3.3: Desempenho médio ao longo de 250 realizações independentes dos algoritmos para equalização cega baseados em filtros de partículas descritos neste capítulo, num sistema BPSK (empregando $N = 500$ partículas) e do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR).

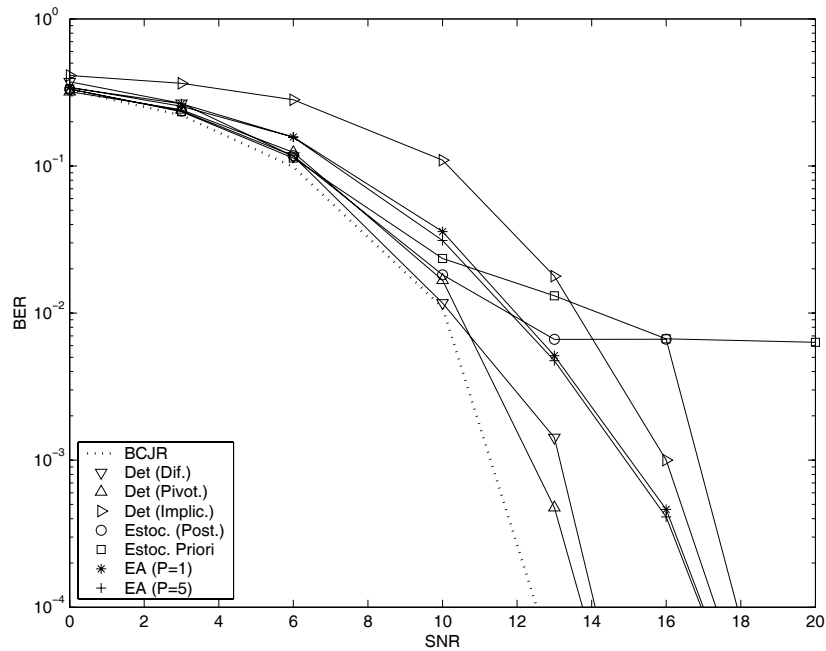


Figura 3.4: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e dos algoritmos para equalização cega baseados em filtros de partículas descritos neste capítulo, num sistema BPSK, empregando $N = 300$ partículas e suavização (com atraso de 5 amostras), em função da relação sinal-ruído (SNR).

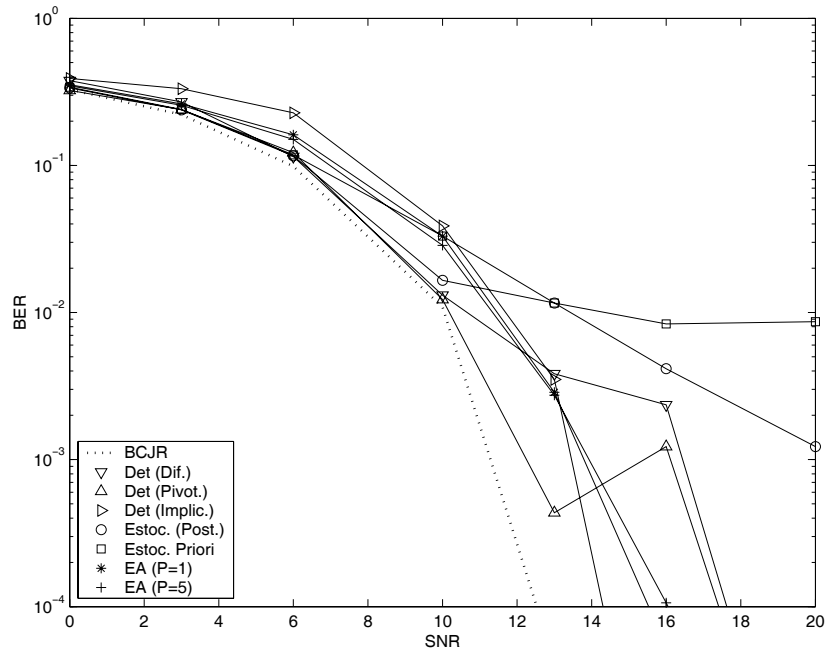


Figura 3.5: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e dos algoritmos para equalização cega baseados em filtros de partículas descritos neste capítulo, num sistema BPSK, empregando $N = 300$ partículas e suavização (com atraso de 10 amostras), em função da relação sinal-ruído (SNR).

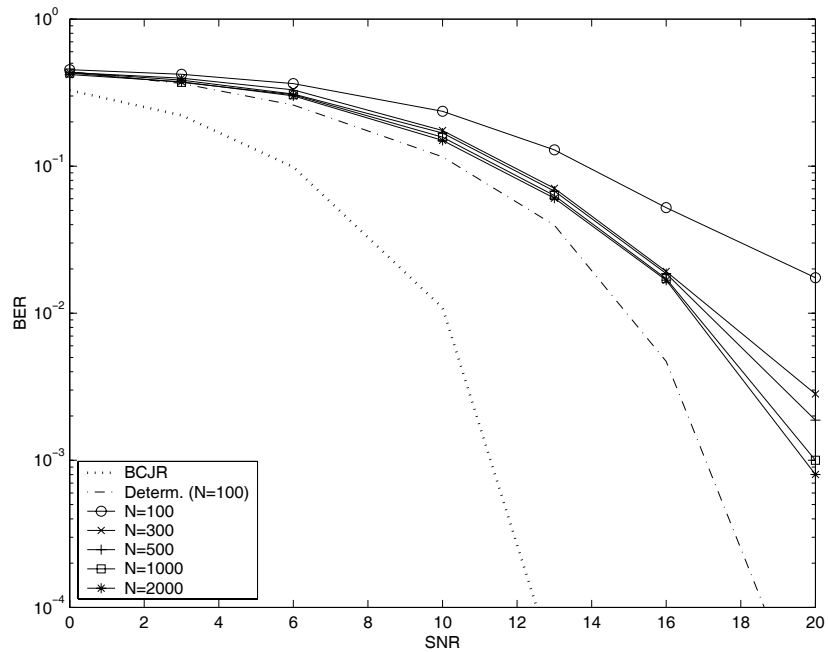


Figura 3.6: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR), do algoritmo determinístico (utilizando $N = 100$ partículas) e dos algoritmos para equalização cega baseados em evolução artificial ($P = 1$), num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR) e do número de partículas N .

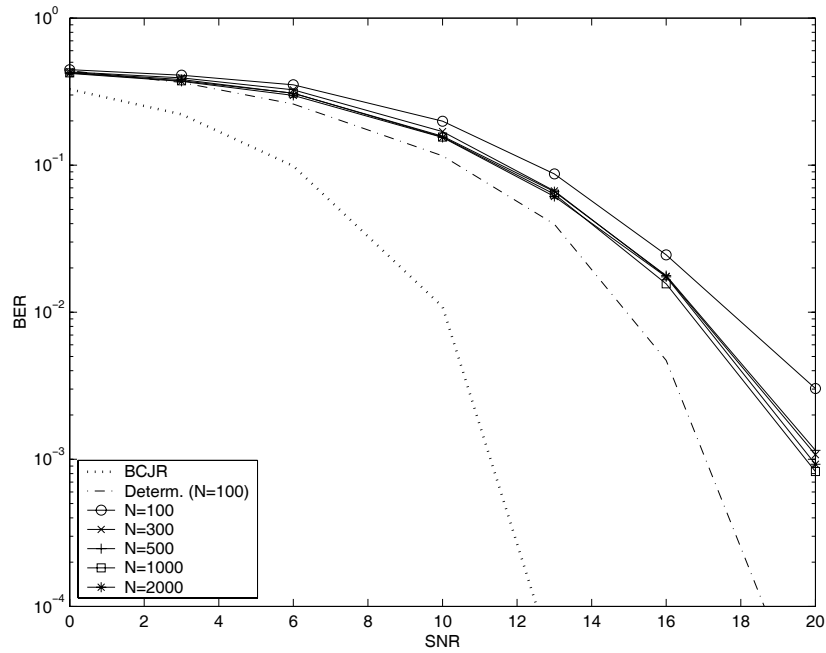


Figura 3.7: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR), do algoritmo determinístico (utilizando $N = 100$ partículas) e dos algoritmos para equalização cega baseados em evolução artificial ($P = 5$), num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR) e do número de partículas N .

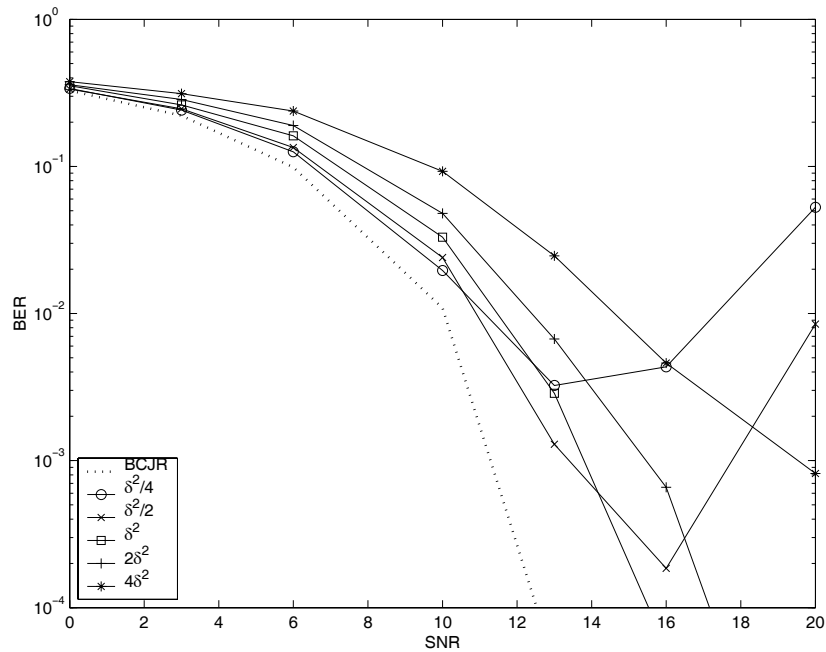


Figura 3.8: Efeito da escolha da variância δ^2 do núcleo de transição $p(h_n|h_{n-1})$ sobre o desempenho médio ao longo de 250 realizações independentes dos algoritmos para equalização cega baseados em evolução artificial ($P = 1$), utilizando $N = 1000$ partículas e suavização (com atraso de 10 amostras) em função da relação sinal-ruído (SNR), comparado com o desempenho do equalizador ótimo MAP (BCJR).

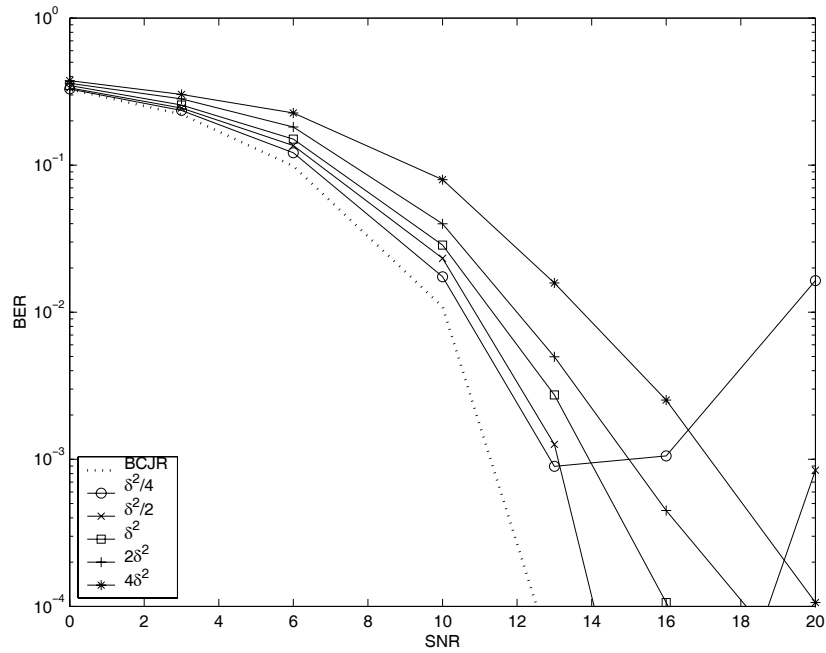


Figura 3.9: Efeito da escolha da variância δ^2 do núcleo de transição $p(h_n|h_{n-1})$ sobre o desempenho médio ao longo de 250 realizações independentes dos algoritmos para equalização cega baseados em evolução artificial ($P = 5$), utilizando $N = 1000$ partículas e suavização (com atraso de 10 amostras) em função da relação sinal-ruído (SNR), comparado com o desempenho do equalizador ótimo MAP (BCJR).

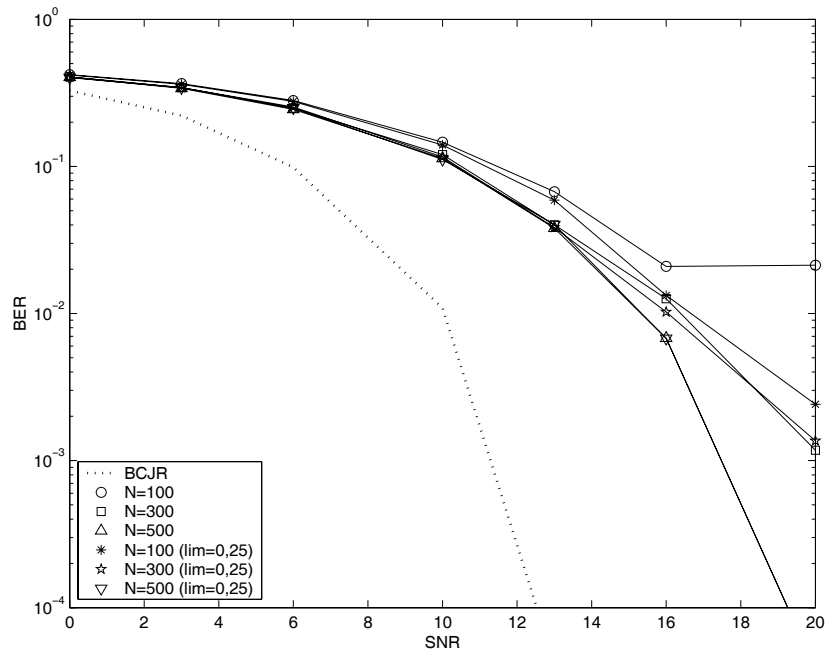


Figura 3.10: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado no filtro de partículas estocástico empregando a função de importância ótima, num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR), do limiar de reamostragem e do número de partículas empregado.

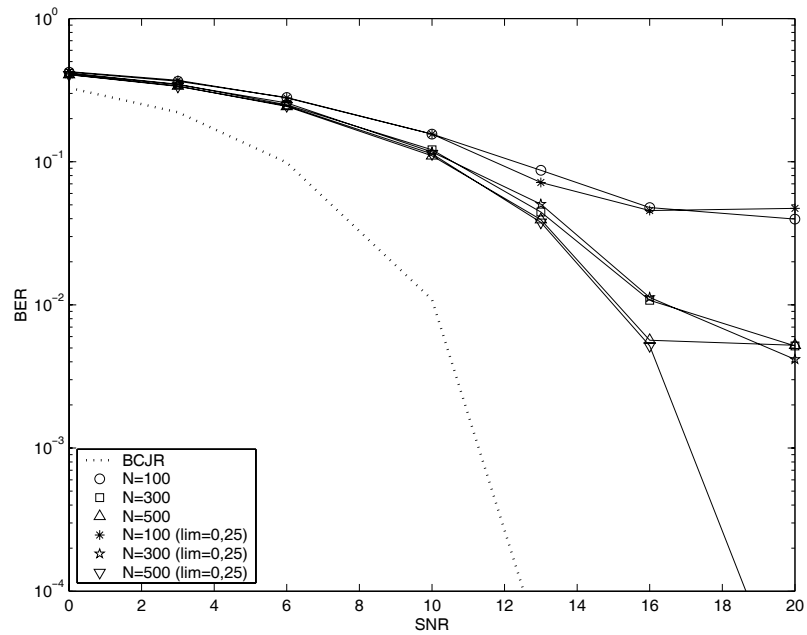


Figura 3.11: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado no filtros de partículas estocástico empregando a função de importância *a priori*, num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR), do limiar de reamostragem e do número de partículas empregado.

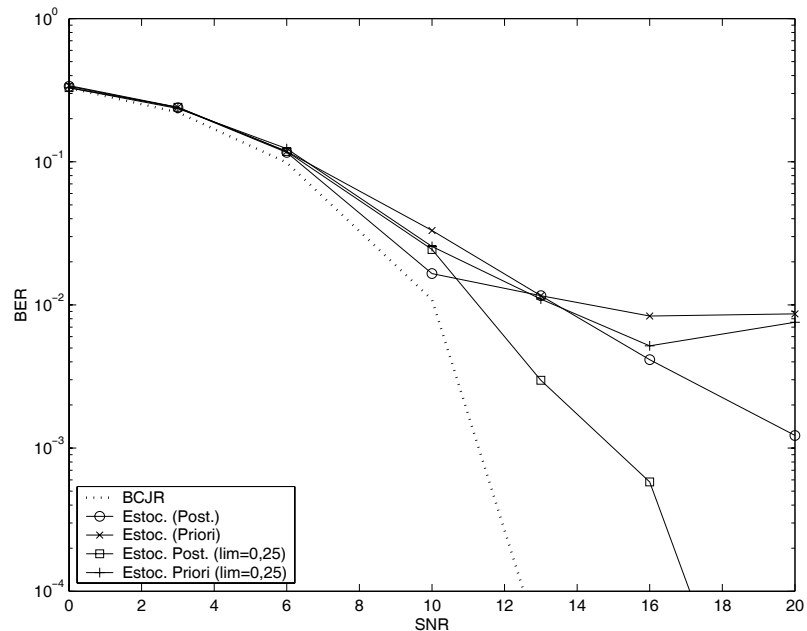


Figura 3.12: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e de algoritmos para equalização cega baseados em filtros de partículas estocásticos (utilizando $N = 300$ partículas e suavização (10 amostras)), empregando a função de importância ótima e *a priori*, num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR) e do limiar de reamostragem.

3.5.1 Equalização sob Ruído Não-Gaussiano

Apesar de oferecerem um desempenho inferior ao dos filtros de partículas Rao-Blackwellizados, os métodos baseados no princípio da evolução artificial (EA) podem ser aplicados a uma ampla gama de modelos de sinal que não permitem a integração analítica dos seus parâmetros. Nesta seção, aplicamos os algoritmos baseados em técnicas de EA descritos na Sec. 3.4 na equalização de um canal com ruído aditivo de envoltória Weibull.

Embora a maioria dos trabalhos na literatura de telecomunicações tenda a recorrer a modelos Gaussianos [34] para o modelamento de perturbações de “cauda longa”, processos com envoltória Weibull são freqüentemente utilizados para o mesmo fim em problemas de detecção com radares [35], por exemplo. Uma variável aleatória complexa $v_n = v_n^R + jv_n^I$ com envoltória Weibull é definida tal que

$$p(|v_n|) = \frac{a}{2\sigma^2} |v_n|^{a-1} \exp\left(-\frac{1}{2\sigma^2} |v_n|^a\right), \quad (3.24)$$

para $a > 1$. Note que ao se adotar o valor $a = 2$, (3.24) se reduz a uma distribuição Rayleigh. Para valores $1 < a \leq 2$, v_n é supergaussiana (i.e., tem curtose positiva), característica enfatizada quando $a \rightarrow 1$.

Observação 3 *Uma variável aleatória que satisfaça (3.24) pode ser obtida [36] a partir de duas variáveis x, y Gaussianas IID, reais, de média nula e variância σ^2 através da transformação*

$$\begin{aligned} v_n^R &= x(x^2 + y^2)^{1/a-1/2} \\ v_n^I &= y(x^2 + y^2)^{1/a-1/2} \end{aligned}$$

□

Pode-se verificar [36] que a densidade de probabilidade conjunta de (v_n^R, v_n^I) é dada por

$$p(v_n^R, v_n^I) = \frac{a}{4\pi\sigma^2} ((v_n^R)^2 + (v_n^I)^2)^{a/2-1} \exp\left(-\frac{1}{2\sigma^2} ((v_n^R)^2 + (v_n^I)^2)^{a/2}\right). \quad (3.25)$$

As densidades marginais $p_{v_n^R}$ e $p_{v_n^I}$, no entanto, não podem ser determinadas analiticamente. Porém, pode-se verificar através de métodos numéricos que

$$p_{v_n^R}(v) = p_{v_n^I}(v) \propto |v|^{\frac{a}{4}-\frac{1}{2}} \exp(|v|^a/\sigma^2). \quad (3.26)$$

Nas Figuras 3.13 e 3.14, mostramos o resultado de simulações numéricas em que avaliamos o desempenho obtido pelo algoritmo da Tab. 3.5, para $P = 1$ e $P = 5$, em função da relação sinal-ruído. Todos os parâmetros desta simulação são iguais aos adotados na seção anterior, à exceção da distribuição do ruído aditivo, assumida como a parte real de uma v.a. com envoltória Weibull de parâmetro¹⁴ $a = 1, 1$, sendo a sua distribuição dada por (3.26).

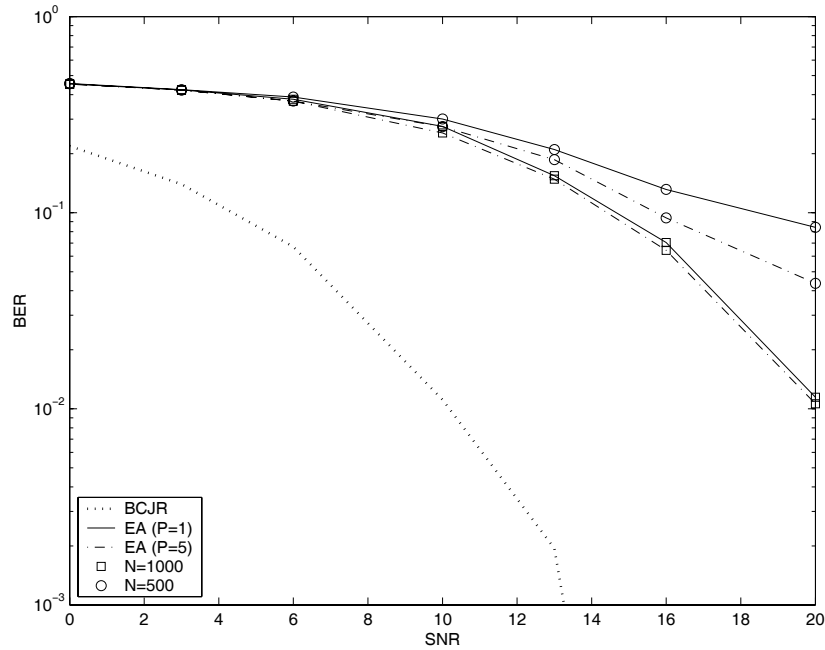


Figura 3.13: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado em evolução artificial, num sistema BPSK com codificação diferencial sob ruído de envoltória Weibull (real, com $\alpha = 1, 1$), em função da relação sinal-ruído (SNR). As linhas cheias mostram os resultados para $N = 500$ partículas, enquanto as linhas tracejadas, resultados para $N = 1000$ partículas.

Como se pode observar na Figura 3.13, o desempenho obtido pelas técnicas de EA é bastante ruim ao não se utilizar suavização. Utilizando suavização (10 amostras - Fig. 3.14), o desempenho obtido se aproxima por cerca de 3 dB do ótimo para o caso em que $P = 5$. Note que, diferentemente do caso Gaussiano, o desempenho dos métodos EA sob ruído Weibull melhora significativamente ao se aumentar o valor de P .

Nas Figuras 3.13 e 3.14, mostramos os resultados obtidos numa simulação distinta, empregando modulação QPSK diferencial (com codificação Gray [37]) e ruído (de

¹⁴Para $a = 1, 1$, a distribuição de v_n^R é marcadamente não-gaussiana. Como indicador disto, podemos citar o fato de que $\mathbb{P}(v_n^R > 3\sigma_{v_n^R}) = 2,5 \cdot 10^{-3}$, enquanto, para uma variável gaussiana, esta probabilidade é de $3,1 \cdot 10^{-5}$.

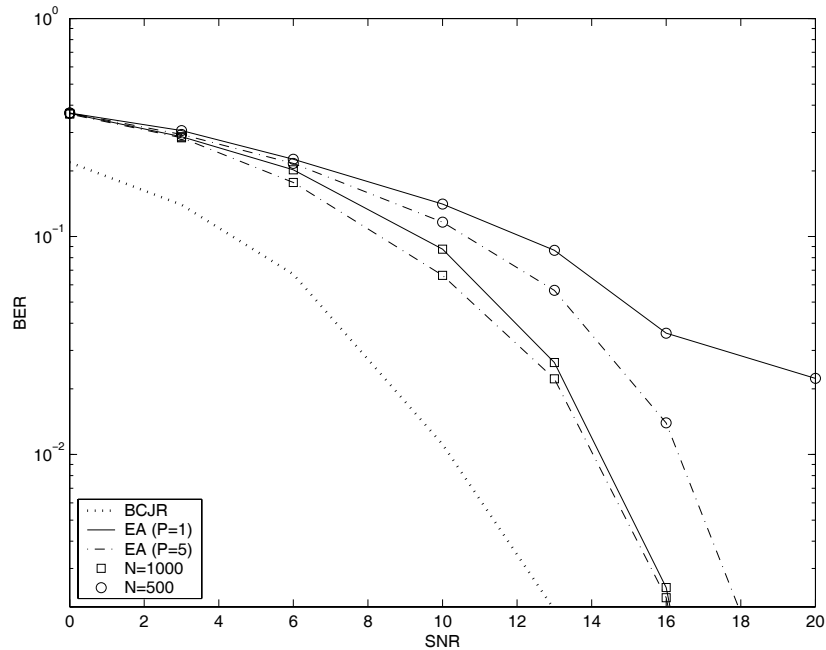


Figura 3.14: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado em evolução artificial, num sistema BPSK com codificação diferencial sob ruído de envoltória Weibull (real, com $\alpha = 1, 1$), em função da relação sinal-ruído (SNR), empregando suavização (10 amostras). As linhas cheias mostram os resultados para $N = 500$ partículas, enquanto as linhas tracejadas, resultados para $N = 1000$ partículas.

envoltória) Weibull complexo, com $a = 1, 1$. O canal empregado agora foi

$$h = [0, 45 \quad -0, 39 - j0, 66 \quad -0, 22 + j0, 39]^T .$$

Como se pode notar, os resultados obtidos são qualitativamente equivalentes ao caso utilizando modulação BPSK. No entanto, pode-se observar que o algoritmo utilizando $P = 1$ tem um desempenho bastante inferior neste caso, e que ao se aumentar o número de partículas para $N = 1000$, consegue-se uma melhora de desempenho não observada no caso anterior.

3.6 Comentários

Neste capítulo, aplicamos diversas técnicas de filtragem de partículas para a solução do problema da equalização cega de um canal linear FIR. Conforme vimos, os algoritmos baseados no filtro de partículas determinístico exibem um desempenho bastante superior ao dos demais, aproximando o desempenho ótimo (MAP) treinado - um fato jamais observado para as técnicas de equalização cega tradicionais baseadas em mo-

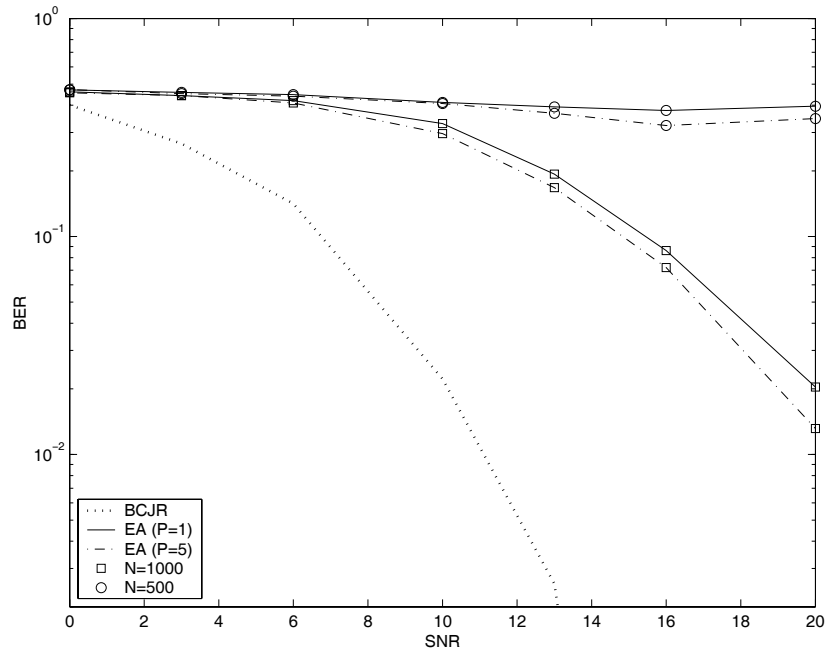


Figura 3.15: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado em evolução artificial, num sistema QPSK com codificação diferencial (Gray) sob ruído de envoltória Weibull (complexo, com $\alpha = 1, 1$), em função da relação sinal-ruído (SNR). As linhas cheias mostram os resultados para $N = 500$ partículas, enquanto as linhas tracejadas, resultados para $N = 1000$ partículas.

mentos.

De uma forma geral, observamos que o desempenho dos algoritmos cresce como função do número de partículas até um determinado ponto, praticamente deixando de ser influenciado por esta grandeza daí para frente, exceto no caso dos algoritmos determinísticos, para os quais um acréscimo no número de partículas além um determinado valor torna-se prejudicial. Pudemos observar também que a convergência dos algoritmos é prejudicada sob baixos níveis de ruído. Este fenômeno ainda não foi quantificado analiticamente, mas pode ser explicado pela observação experimental de que o aumento na variância dos pesos (causado pela “compressão” da função de verossimilhança das observações) influencia negativamente a diversidade amostral, causando uma degradação de desempenho, que é exacerbada ao se empregar a técnica de suavização descrita na Sec. 2.4.2.

Na Tabela 3.6, resumimos os resultados obtidos neste capítulo, reunindo os resultados referentes ao desempenho dos algoritmos para equalização sob ruído gaussiano (com $N = 300$). Os números mostrados nesta tabela são resultados médios obtidos por cada classe de algoritmos testada (utilizando os parâmetros que levam ao melhor desempenho), comparados ao desempenho do equalizador ótimo (BCJR). Como já ha-

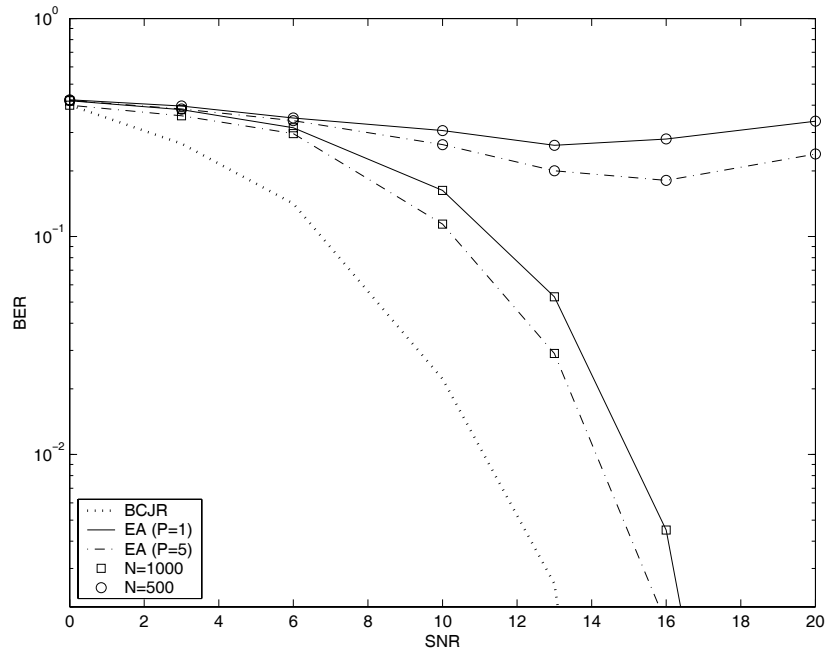


Figura 3.16: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado em evolução artificial, num sistema QPSK com codificação diferencial (Gray) sob ruído de envoltória Weibull (complexo, com $\alpha = 1, 1$), em função da relação sinal-ruído (SNR), empregando suavização (10 amostras). As linhas cheias mostram os resultados para $N = 500$ partículas, enquanto as linhas tracejadas, resultados para $N = 1000$ partículas.

víamos observado na Sec. 3.5, o algoritmo Rao-Blackwellizado determinístico tem um desempenho superior aos demais.

		Sem Suavização		Com Suavização		Complexidade
SNR(dB)		6	20	6	20	
Rao-Black.	Determinístico	5	7	0,5	2	$\mathcal{O}(300L^2)$
	Estocástico	5	9	1	6	$\mathcal{O}(300L^2)$
Evolução Artificial		6	9	1	3	$\mathcal{O}(300L)$

Tabela 3.6: Desempenho dos algoritmos testados ($N = 300$), quantificado pela perda de desempenho (em dB) em relação ao equalizador ótimo (BCJR) sob relações sinal-ruído de 6 e 20 dB, e complexidade computacional dos métodos.

Capítulo 4

Equalização e Decodificação Conjuntas empregando Filtros de Partículas

Atualmente, a quase totalidade dos sistemas de comunicação práticos emprega alguma forma de codificação para detecção ou correção de erros. Conseqüentemente, caso o canal de comunicação considerado seja seletivo em frequência, o desenvolvimento de um receptor ótimo implica no uso de um esquema conjunto de equalização e decodificação, uma vez que a recepção ótima requer a determinação da densidade *a posteriori* dos dados transmitidos dado o sinal observado.

Diversos trabalhos (vide [38]) propuseram métodos para este fim, na sua maior parte baseados na técnica iterativa conhecida como *turbo-equalização*, uma vez que a solução ótima para este problema tem, em geral, complexidade proibitiva. Além de terem aplicabilidade limitada ao caso em que o código empregado possa ser *soft decoded* [37], as técnicas de *turbo-equalização* exibem uma complexidade computacional bastante elevada, além de limitações de desempenho [39].

Neste capítulo descrevemos algoritmos para equalização cega e decodificação conjuntas de canais de comunicação seletivos em frequência baseados em filtros de partículas. Os algoritmos que apresentamos aqui são inéditos¹ até onde temos conhecimento, e levam a desempenhos bastante superiores aos exibidos por algoritmos alternativos que realizam a equalização do canal e a decodificação separadamente.

O conteúdo a seguir está organizado da seguinte forma: na Sec. 4.1, apresentamos

¹Em [12] desenvolveram-se algoritmos para equalização e decodificação conjuntas baseados em filtros de partículas para o caso não-seletivo em frequência. Apesar de mencionar que o caso seletivo poderia ser tratado como uma extensão trivial do seu trabalho, a formulação desenvolvida pela autora de [12], que considerava o estado do codificador como parte da variável de estado a ser amostrada, tornaria este desenvolvimento bastante difícil.

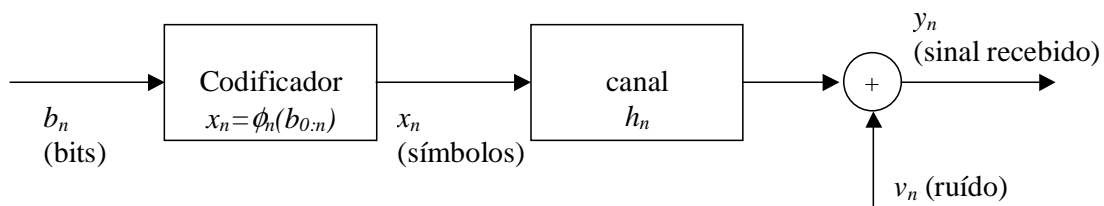


Figura 4.1: Modelo de sinal adotado neste capítulo.

os fundamentos da técnica desenvolvida, que é aplicada na Sec. 4.2 a um sistema de comunicação com codificação em bloco, e nas Sec. 4.3 e seguintes a sistemas empregando codificação convolucional.

4.1 Equalização e Decodificação Conjuntas: Fundamentos

Neste seção, descrevemos os fundamentos teóricos que permitem a aplicação (com algumas modificações) dos algoritmos descritos no Cap. 3 na equalização e decodificação conjuntas de sinais gerados de acordo com o modelo da Fig. 4.1. Assumimos que os codificadores empregados sejam não-catastróficos [37], o que implica que o mapeamento ϕ seja inversível, i.e.

$$x_{0:R(n)} = \phi(b_{0:n}) \Leftrightarrow \phi(b_{0:n}) = \phi^{-1}(x_{0:R(n)}), \quad (4.1)$$

em que a função $R(n)$ mapeia convenientemente os índices dos bits naqueles dos símbolos gerados. Da condição imposta em (4.1) segue que

$$\begin{aligned} \mathbb{P}(b_{0:n}|y_{0:R(n)}) &= \sum_{x_{0:R(n)}=\phi(b_{0:n})} \mathbb{P}(x_{0:R(n)}|y_{0:R(n)}) \\ &= \mathbb{P}(\phi^{-1}(x_{0:R(n)})|y_{0:R(n)}), \end{aligned} \quad (4.2)$$

ou seja, como a transmissão de uma seqüência de bits $b_{0:n}$ define unicamente uma seqüência correspondente de símbolos $x_{0:R(n)}$, as probabilidades *a posteriori* da seqüência de bits $b_{0:n}$ podem ser calculadas a partir das probabilidades *a posteriori* de $x_{0:R(n)}$ pela simples inversão do mapeamento ϕ imposto pelo código. É importante observar neste ponto que as seqüências $x_{0:R(n)}$ fora da imagem de ϕ (i.e., impossíveis de serem geradas pelo código) têm probabilidades *a priori* nulas, o que deve ser levado em conta pelo algoritmo de equalização para permitir o cálculo de (4.2).

Nas seções a seguir, descrevemos métodos originais para a equalização e decodificação conjunta de canais seletivos em frequência (i.e., métodos para a aproximação

de $p(b_{0:n}|y_{0:R(n)})$ baseados no resultado de (4.2). Estes métodos, diferentemente dos desenvolvidos em [12] não necessitam que o estado do codificador seja incluído explicitamente na variável de estados a estimar, o que permite sua aplicação a modelos de sinal mais complexos.

4.2 Equalização e Decodificação Conjuntas: Códigos de Bloco

Considere um sistema de codificação em blocos binário² (M, K) , no qual cada seqüência de K bits da mensagem transmitida $\mathbf{b}_n \triangleq b_{(n-1)K+1:nK}$ é unicamente mapeada num bloco de M elementos $\mathbf{x}_n \triangleq x_{(n-1)M+1:nM}$ da seqüência de símbolos transmitida por $\mathbf{x}_n = \phi(\mathbf{b}_n)$, em que ϕ é uma função determinística definida pelo esquema de codificação (Fig. 4.1).

Neste capítulo, assumimos que os símbolos são transmitidos por um sistema de comunicação linear com canal de ordem finita e conhecida (Eq. 3.1) e invariante no tempo. Nosso objetivo nesta seção é, então, desenvolver um algoritmo para a estimação da densidade $p(b_{0:nK}|y_{0:nM})$, a partir da qual podem-se obter estimativas ótimas para a seqüência de bits transmitidos.

Observação 1 *Os métodos mais comuns (subótimos) para este fim obtêm estimativas preliminares $\hat{\mathbf{x}}_n$ da seqüência de símbolos transmitida (ignorando a codificação) e tentam recuperar os bits transmitidos \mathbf{b}_n através de algum método tradicional, geralmente baseado em hard decisions [37]. É interessante observar, no entanto, que códigos de bloco podem ser decodificados de forma ótima (com complexidade computacional exponencial em K) explorando o fato de que as 2^{M-K} seqüências \mathbf{x}_n não pertencentes à imagem de ϕ não podem ser transmitidas, i.e., suas probabilidades a priori são nulas, do que segue que*

$$p(\mathbf{b}_n|y_{0:n}) = p(\mathbf{x}_n = \phi(\mathbf{b}_n)|y_{0:n}) . \quad (4.3)$$

As probabilidades a posteriori individuais de cada bit transmitido podem então ser determinadas marginalizando-se $p(\mathbf{b}_n|y_{0:n})$.

O algoritmo que propomos baseia seu funcionamento em (4.2): devido ao mapeamento unitário entre \mathbf{b}_n e \mathbf{x}_n , a equalização e decodificação conjuntas do sinal recebido pode ser realizada através dos seguintes passos: 1) projeta-se um filtro de partículas

²Em que M denota o tamanho da palavra e $M - K$ o número de dígitos redundantes.

para a equalização cega do sinal recebido (com densidades *a priori* para x_n propriamente ajustadas de acordo com o código empregado) da mesma forma que no Cap. 3, e 2) inverte-se o mapeamento ϕ .

O primeiro passo pode ser implementado conforme descrito na Sec. 3.3.1, bastando que se determine a densidade $p(x_n|x_{0:n-1})$, necessária para o cálculo de (3.16). Note que esta densidade é unicamente determinada pelo código empregado, podendo ser calculada por busca exaustiva, como no exemplo a seguir. Na Tab. 4.1 descrevemos o algoritmo proposto (empregando o filtro de partículas estocástico).

<p>(Inicialização)</p> <p>Para $i = 1 : N$</p> <p>$X_0^{(i)} \sim \mathcal{B}_L(\pm 1)$</p> <p>$h_0^{(i)} \sim \mathcal{N}(h h_0; \Sigma_0)$</p> <p>Fim</p> <p>(Algoritmo)</p> <p>Para $n > 0$</p> <p>1) Para $i = 1 : N$</p> <p>A) Amostre $X_n^{(i)} \sim p(y_n X_{0:n}^{(i)}, y_{0:n-1})p(x_n x_{0:n-1})$</p> <p>B) Calcule e normalize os pesos</p> <p>$w_n^{(i)} \propto w_{n-1}^{(i)} \sum_{X_n^{(i)}} p(y_n X_{0:n}^{(i)}, y_{0:n-1})p(x_n x_{0:n-1})$</p> <p>Fim</p> <p>2) Reamostre o conjunto de partículas, conforme o critério adotado.</p> <p>3) Caso $(\lfloor n \rfloor - 1)K + 1 \leq n \leq \lfloor n \rfloor K$, estime a densidade <i>a posteriori</i> dos bits transmitidos como</p> <p>$P(b_n = B y_{0:n}) \approx N^{-1} \sum_{i=1}^N \mathcal{I}\{x_n^{(i)} = B\}$.</p> <p>Fim</p>

Tabela 4.1: Algoritmo para equalização e decodificação conjunta em sistemas empregando códigos de bloco sistemáticos baseado no algoritmo SIR estocástico, empregando a função de importância ótima.

Exemplo 1 Considere o código (3,1) cujas palavras associadas aos bits 0 e 1 são respectivamente [010] e [101]. Assumindo o mapeamento $\{0; 1\} \rightarrow \{-1; 1\}$ entre as palavras de código de o sinal transmitido e que os bits transmitidos são equiprováveis e IID, resulta que $x_{3n:3n+2} = \{-1 \ 1 \ -1\}$ ou $x_{3n:3n+2} = \{1 \ -1 \ 1\}$ com probabilidade

0, 5. Temos assim que:

$$\begin{aligned}
p(x_{3n}|x_{0:3n-1}) &= p(x_{3n}) = 0,5, \quad x_{3n} = \pm 1 \\
p(x_{3n+1}|x_{0:3n}) &= p(x_{3n+1}|x_{3n}) = \begin{cases} 1, & \text{se } x_{3n+1} = -1 \quad \text{e } x_{3n} = 1 \\ 1, & \text{se } x_{3n+1} = 1 \quad \text{e } x_{3n} = -1 \\ 0, & \text{caso contrario} \end{cases} \\
p(x_{3n+2}|x_{0:3n+1}) &= p(x_{3n+2}|x_{3n+1}) = \begin{cases} 1, & \text{se } x_{3n+2} = -1 \quad \text{e } x_{3n+1} = 1 \\ 1, & \text{se } x_{3n+2} = 1 \quad \text{e } x_{3n+1} = -1 \\ 0, & \text{caso contrario} \end{cases}
\end{aligned} \tag{4.4}$$

□

O cálculo de $p(x_n|x_{0:n-1})$ é especialmente simplificado no caso em que os bits transmitidos são equiprováveis e IID e o código empregado é sistemático [37], i.e., cada bloco de M bits transmitido é construído como

$$\begin{aligned}
(x_{(n-1)M+1}, \dots, x_{nM}) &= \\
& (b_{(n-1)K+1}, \dots, b_{nK}, p_{nK+1}, \dots, p_{nM})
\end{aligned} \tag{4.5}$$

em que $b_{(n-1)K+1:nK}$ é a mensagem transmitida e $p_{nK+1:nM}$ é um bloco de “paridade”, o qual é uma função dos elementos anteriores. Sob estas condições, $p(x_n = a|x_{0:n-1})$ é dada por

$$\begin{cases} 1/2, & \text{se } (n-1)K+1 \leq n \leq nK \quad \text{e } a = \pm 1 \\ 1, & \text{se } nK+1 \leq n \leq nM \quad \text{e } a = p_n \\ 0, & \text{se } nK+1 \leq n \leq nM \quad \text{e } a \neq p_n \end{cases} \tag{4.6}$$

Assim, usando (3.16) e (4.6) pode-se calcular $p(X_n, y_n|X_{0:n-1}, y_{0:n-1})$ e estimar $p(x_{0:n}|y_{0:n})$ através dos algoritmos descritos anteriormente. Para se estimar $p(b_{0:l}|y_{0:n})$ ($l = M \lfloor n/M \rfloor$), basta notar que cada bloco de comprimento M de $x_{0:n}^{(i)}$ é restrito, por construção, a pertencer à imagem de ϕ . Desta forma, amostras da densidade $p(b_{0:l}|y_{0:n})$ podem ser obtidas pela relação³ $b_{0:l}^{(i)} = \phi^{-1}(x_{0:n}^{(i)})$.

Finalmente, deve-se mencionar que a estimação de $p(b_{0:l}|y_{0:n})$ só é possível se alguma medida for tomada para contrabalançar os efeitos da ambigüidade de fase. Além dos métodos descritos no Cap. 3 para este fim, a codificação em bloco torna possível contornar este problema através do uso de códigos “rotacionalmente variantes”, para os quais $\phi(\mathbf{b}^{(i)}) \neq -\phi(\mathbf{b}^{(j)})$, $\forall i, j$. Na Sec. 4.2.1, analisamos através de simulações

³No caso dos códigos sistemáticos, basta tomar os K primeiros elementos de cada bloco de M amostras.

numéricas o desempenho do algoritmo aqui proposto.

4.2.1 Simulações

Para avaliar o desempenho dos algoritmos descritos neste capítulo realizamos simulações numéricas, medindo as taxas médias de erro de bit (BER) obtidas ao longo de 250 realizações independentes. O sistema de comunicação analisado transmite blocos de 50 palavras de código (cada uma escolhida de forma independente e equiprovável), sendo as 25 primeiras descartadas para fins de medição das taxas de erro de modo a permitir a convergência dos algoritmos analisados.

Nas simulações a seguir, empregamos o canal $h = [0,41 \quad -0,82 \quad 0,41]^T$. Os filtros empregam 250 partículas e são inicializados de forma aleatória, fazendo $h_0^{(i)} \sim \mathcal{N}(h|0; I)$. Na Figura 4.2 mostramos o desempenho médio dos algoritmos para equalização e decodificação conjunta em função da SNR e do atraso de suavização d , empregando o código sistemático B_1 , descrito na Tabela 4.2. Este código (6, 3) é formado [40] tomando-se as 8 palavras do código de Hamming (7, 3) correspondentes ao bit mais significativo "0", permitindo a resolução da ambigüidade de fase devido à sua assimetria rotacional⁴.

Para efeito de comparação, avaliamos também a taxa média de erro obtida através de um esquema alternativo treinado (MAP), utilizando o equalizador ótimo MAP (com canal conhecido), seguido de um decodificador empregando *hard decisions*. Como se pode observar na Fig. 4.2, os algoritmos baseados no filtro de partículas determinístico tiveram um desempenho superior até mesmo ao do esquema treinado⁵. Na Fig. 4.3 repetimos o mesmo experimento, utilizando o código B_2 descrito na Tab. 4.2, formado de forma semelhante ao B_1 , sem excluir, porém, a coluna de zeros comum à todas as palavras de código. Como se pode observar, os resultados obtidos são semelhantes.

⁴Note que, da forma como o código é definido, nenhuma palavra de código válida corresponde à negação de outra palavra de código válida.

⁵Observe que o esquema treinado usado na comparação é subótimo, e que a decodificação ótima MAP de códigos em bloco demanda a realização de buscas exaustivas, o que só é factível para códigos com M muito baixo.

$$B_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Tabela 4.2: Códigos de bloco empregados nas simulações da Seção 4.2.1: cada matriz coleciona em suas linhas as palavras de códigos válidas.

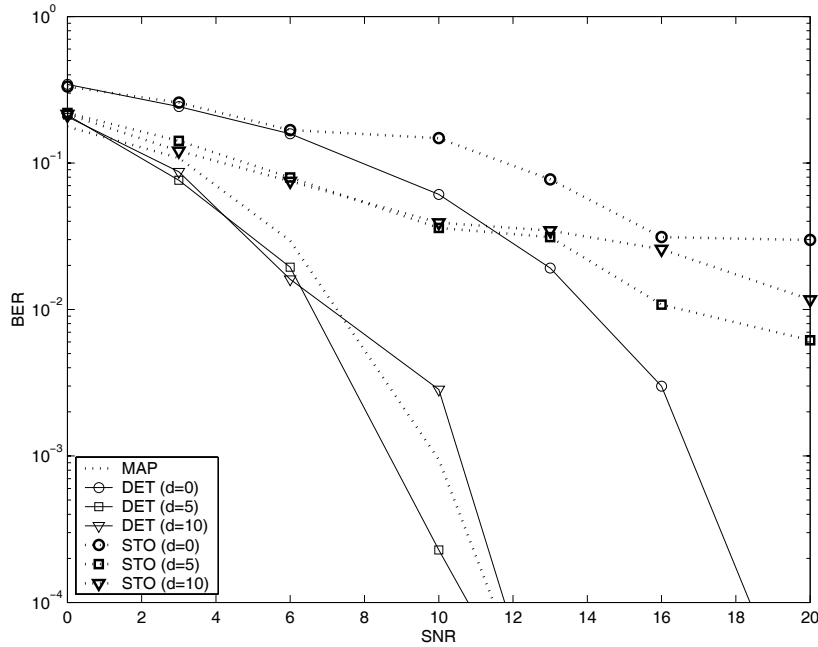


Figura 4.2: Desempenho médio ao longo de 250 realizações independentes de algoritmos para equalização e decodificação conjunta empregando o filtro de partículas estocástico com função de importância ótima (STO) e o filtro determinístico (DET), e de um esquema treinado (MAP) descrito no texto, em função da SNR e do atraso de suavização d , para um sistema empregando o código de blocos B_1 descrito na Tab. 4.2.

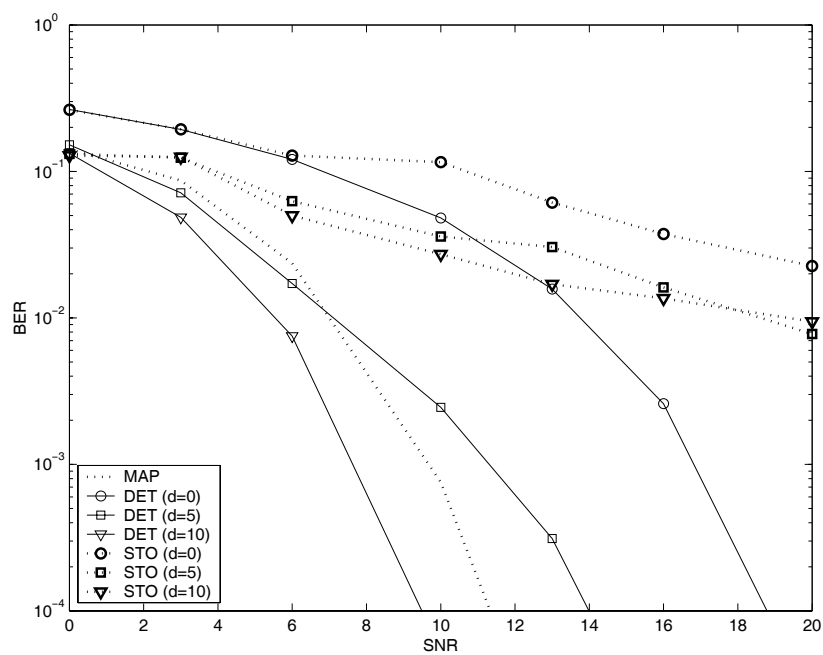


Figura 4.3: Desempenho médio ao longo de 250 realizações independentes de algoritmos para equalização e decodificação conjunta empregando o filtro de partículas estocástico com função de importância ótima (STO) e o filtro determinístico (DET), e de um esquema treinado (MAP) descrito no texto, em função da SNR e do atraso de suavização d , para um sistema empregando o código de blocos B_2 descrito na Tab. 4.2.

4.3 Equalização e Decodificação conjunta: Códigos Convolucionais

No algoritmo desenvolvido na Sec. 4.2, a utilização de códigos em bloco sistemáticos permite que se determine de forma simples a densidade *a priori* $p(x_n|x_{0:n-1})$ e que se decodifiquem as seqüências de símbolos $x_{0:n}$ de forma trivial, condições que possibilitam a determinação da distribuição *a posteriori* dos bits transmitidos, de acordo com os fundamentos descritos na Sec. 4.1. Nesta seção, desenvolvemos métodos que se baseiam no mesmo princípio exposto na Sec. 4.1, mas que, ao invés de gerarem amostras dos símbolos para posterior decodificação, simulam diretamente da densidade *a posteriori* dos bits transmitidos b_n . Estes métodos permitem que se empregue uma classe mais ampla de códigos, requerendo unicamente que a operação de codificação seja simples.

4.3.1 Fundamentos

O algoritmo que descrevemos a seguir emprega filtros de partículas para estimar a densidade $p(b_{0:n}|\underline{y}_{0:n}) \triangleq p(b_{0:n}|y_{0:(n+1)R-1})$, e se aplica a sistemas de comunicação utilizando codificação convolucional não-catastrófica de taxa $1/R$ [37]. Denotando a seqüência de bits transmitidos por $b_n (\in \{0; 1\})$, a seqüência (binária) $x_n (\in \pm 1)$ de símbolos transmitidos por um sistema deste tipo é dada por $x_{nR+m} = 2 c_n^m - 1$, $0 \leq m \leq R$, sendo a seqüência c_n^m obtida como

$$c_n^m = \left(\sum_{i=0}^K b_{n-i} d_i^m + \sum_{i=1}^K c_{n-i}^m r_i^m \right) \text{ mod } 2, \quad (4.7)$$

em que K é o *constraint length* do código e $d_i^m, r_i^m (\in \{0; 1\})$ são os coeficientes do gerador.

Para desenvolver o algoritmo que propomos, devemos inicialmente obter a expressão da densidade $p(b_n, \underline{y}_n | b_{0:n-1}, \underline{y}_{0:n-1})$. Sob a hipótese de que b_n seja uma seqüência IID, segue que

$$\begin{aligned} p(b_n, \underline{y}_n | b_{0:n-1}, \underline{y}_{0:n-1}) &= p(\underline{y}_n | b_{0:n}, \underline{y}_{0:n-1}) p(b_n | b_{0:n-1}, \underline{y}_{0:n-1}) \\ &= p(\underline{y}_n | b_{0:n}, \underline{y}_{0:n-1}) p(b_n). \end{aligned} \quad (4.8)$$

Além disso, assumindo que a seqüência de bits b_n defina unicamente a seqüência

transmitida x_n , (4.8) implica pelos resultados da Sec. 4.1 que

$$p(\underline{y}_n | b_{0:n}, \underline{y}_{0:n-1}) = p(\underline{y}_n | X_{0:(n+1)R-1}, \underline{y}_{0:n-1}), \quad (4.9)$$

em que $X_n \triangleq [x_n \dots x_{n-L+1}]^T$, e $x_{0:(n+1)R-1}$ é a seqüência de símbolos correspondente à seqüência de bits $b_{0:n}$.

A densidade do lado direito de (4.9), por sua vez, pode ser decomposta como

$$p(\underline{y}_n | X_{0:(n+1)R-1}, \underline{y}_{0:n-1}) = \prod_{j=Rn}^{R(n+1)-1} p(y_j | X_{0:j}, y_{0:j-1}), \quad (4.10)$$

uma vez que

$$\begin{aligned} p(\underline{y}_n | X_{0:(n+1)R-1}, \underline{y}_{0:n-1}) &\triangleq p(y_{nR:(n+1)R-1} | X_{0:(n+1)R-1}, y_{0:nR-1}) \\ &= p(y_{(n+1)R-1} | X_{0:(n+1)R-1}, y_{0:(n+1)R-2}) \\ &\quad \times p(y_{(n+1)R-2} | X_{0:(n+1)R-1}, y_{0:(n+1)R-3}) \\ &\quad \vdots \\ &\quad \times p(y_{nR+1} | X_{0:(n+1)R-1}, y_{0:nR}) \\ &\quad \times p(y_{nR} | X_{0:(n+1)R-1}, y_{0:nR-1}) \end{aligned} \quad (4.11)$$

em que⁶, para $j > i$,

$$p(y_i | X_{0:j}, y_{0:i-1}) = p(y_i | X_{0:i}, y_{0:i-1}). \quad (4.13)$$

densidade esta última já calculada em (3.16).

Pelo resultado de (4.10), podemos verificar que a determinação de (4.8) para cada valor possível de b_k requer o cálculo de R passos de um filtro de Kalman (necessários para determinar 4.13). Assim, se N partículas forem empregadas, o algoritmo proposto requer o cálculo de $2NR$ passos de um filtro de Kalman para cada bit decodificado, o dobro⁷ da complexidade de um equalizador cego baseado em filtros de partículas (operando à taxa de símbolos). Na Tab. 4.3 descrevemos o algoritmo proposto (empregando o algoritmo SIR estocástico).

⁶Demonstração de (4.13):

$$\frac{p(y_i | y_{0:i-1}, X_{0:j})}{p(y_i | y_{0:i-1}, X_{0:i})} = \frac{p(X_{i+1:j} | X_{0:i}, y_{0:i})}{p(X_{i+1:j} | X_{0:i}, y_{0:i-1})} = \frac{p(X_{i+1:j} | X_{0:i})}{p(X_{i+1:j} | X_{0:i})} = 1. \quad (4.12)$$

⁷Note que os algoritmos para equalização cega (Sec. 3) Rao-Blackwellizados exigem o cálculo de um passo de um filtro de Kalman por iteração independentemente do tamanho da constelação empregada, uma vez que apenas o símbolo escolhido para cada partícula influencia o valor futuro dos parâmetros.

```

(Inicialização)
Para  $i = 1 : N$ 
   $X_0^{(i)} \sim \mathcal{B}_L(\pm 1)$ 
   $h_0^{(i)} \sim \mathcal{N}(h|h_0; \Sigma_0)$ 
Fim
(Algoritmo)
Para  $n > 0$ 
1) Para  $i = 1 : N$ 
  A) Amostre  $b_n^{(i)} \sim p(b_n|b_{0:n-1}^{(i)}, \underline{y}_{0:n})$  (densidade proporcional à da Eq. 4.8)
  B) Calcule e normalize os pesos  $w_n^{(i)} = w_{n-1}^{(i)} \sum_{b_n} p(b_n|b_{0:n-1}^{(i)}, \underline{y}_{0:n})$ 
Fim
2) Reamostre o conjunto de partículas, conforme o critério adotado.
3) Estime a densidade a posteriori dos bits transmitidos como
   $P(b_n = B|\underline{y}_{0:n}) \approx N^{-1} \sum_{i=1}^N \mathcal{I}\{b_n^{(i)} = B\}$  .
Fim

```

Tabela 4.3: Algoritmo para equalização e decodificação conjunta em sistemas empregando códigos convolucionais baseado no algoritmo SIR estocástico, empregando a função de importância ótima.

4.4 Simulações

4.4.1 Códigos Convolucionais Não-Recursivos

Nesta seção, avaliamos através de simulações numéricas o desempenho de algoritmos para equalização e decodificação conjunta em sistemas empregando códigos convolucionais não-recursivos (i.e., códigos para os quais $r_i^{(m)} = 0, m > 0$). Para avaliar o desempenho destes algoritmos medimos as taxas médias de erro de bit (BER) obtidas ao longo de 250 realizações independentes. O sistema de comunicação analisado transmite blocos de 200 bits IID, sendo os 50 primeiros descartados para fins de medição das taxas de erro, de modo a permitir a convergência dos algoritmos analisados.

Nas simulações a seguir, empregamos o mesmo canal dos experimentos anteriores ($h = [0, 41 \ -0, 82 \ 0, 41]^T$). Os filtros de partículas empregam 250 partículas, suavização (5 amostras) e são inicializados de forma aleatória, fazendo $h_0^{(i)} \sim \mathcal{N}_h(0; I)$. Para efeitos de comparação, mostramos também as taxas médias de erro obtidas através de dois esquemas alternativos: num primeiro (Pf+Vit), empregamos um equalizador cego baseado no algoritmo SIR estocástico, decodificando o código convolucional através do algoritmo de Viterbi⁸. Num segundo esquema (Viterbi), utilizamos um equalizador

⁸A versão do algoritmo de Viterbi que utilizamos efetua um *traceback* de 20 bits.

ótimo MLSE (com canal conhecido), seguido pelo mesmo decodificador de Viterbi do esquema anterior.

Na Figura 4.4 mostramos o desempenho médio dos algoritmos para equalização e decodificação conjunta, empregando o código binário (gerado pelo polinômio dado em representação octal) (5, 7, 2). Este código ($R = 1/3$) possui propriedades de invariância rotacional, exibindo propriedades ótimas [41] quando detectado incoerentemente, o que permite contornar o problema de ambigüidade de fase descrito nos capítulos anteriores. Como se pode observar na Fig. 4.4, algoritmo baseado no filtro de partículas determinístico teve um desempenho superior até mesmo ao do esquema treinado (Viterbi) em todas as faixas de relação sinal-ruído, ultrapassando em muito o do esquema com equalização e decodificação separadas (Pf+Vit). Os algoritmos baseados em filtros de partículas estocásticos exibiram um desempenho significativamente inferior, à exceção do algoritmo empregando reamostragem multinomial, cujo desempenho excedeu o dos demais sob baixos níveis de ruído. Vale ainda notar que o desempenho máximo dos algoritmos estocásticos se deu para um limiar de reamostragem $T_{MIN} = 0, 2$, o que ilustra uma maior sensibilidade destes métodos ao problema do empobrecimento amostral se comparados aos métodos do Cap. 3.

Nas Fig. 4.5 e 4.6 repetimos o mesmo experimento, utilizando os códigos (17, 12, 04) e (63, 44, 17) (que possuem as mesmas propriedades de variância rotacional do código (7, 5, 2) [41]). Como se pode observar, os resultados obtidos são semelhantes no caso do código (17, 12, 04). Para o código (63, 44, 17), no entanto, o desempenho dos filtros de partículas estocásticos é muito pior, fato que não se altera (não mostrado no gráfico) mesmo ao se empregar um número de partículas bastante superior, resultado que pode ser creditado à memória mais longa deste código, que introduz uma dependência mais forte em relação a decisões passadas.

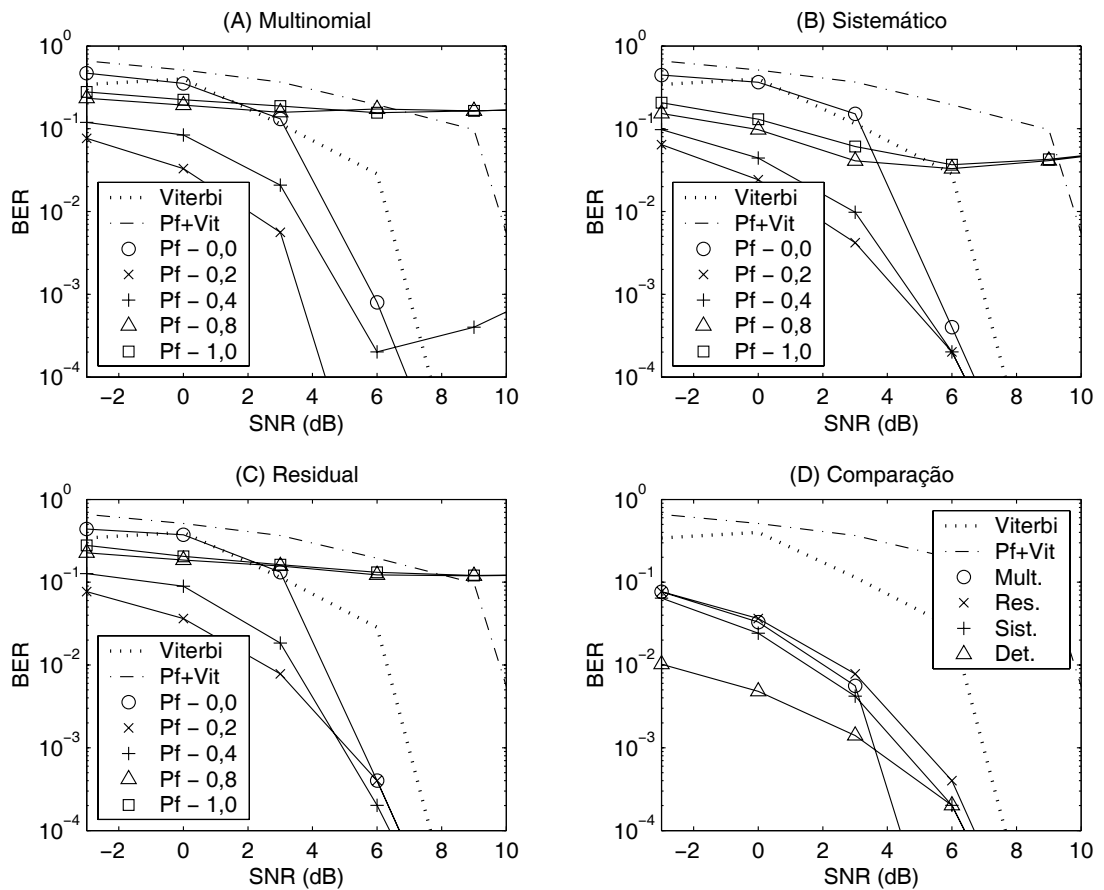


Figura 4.4: Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (5, 7, 2). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.

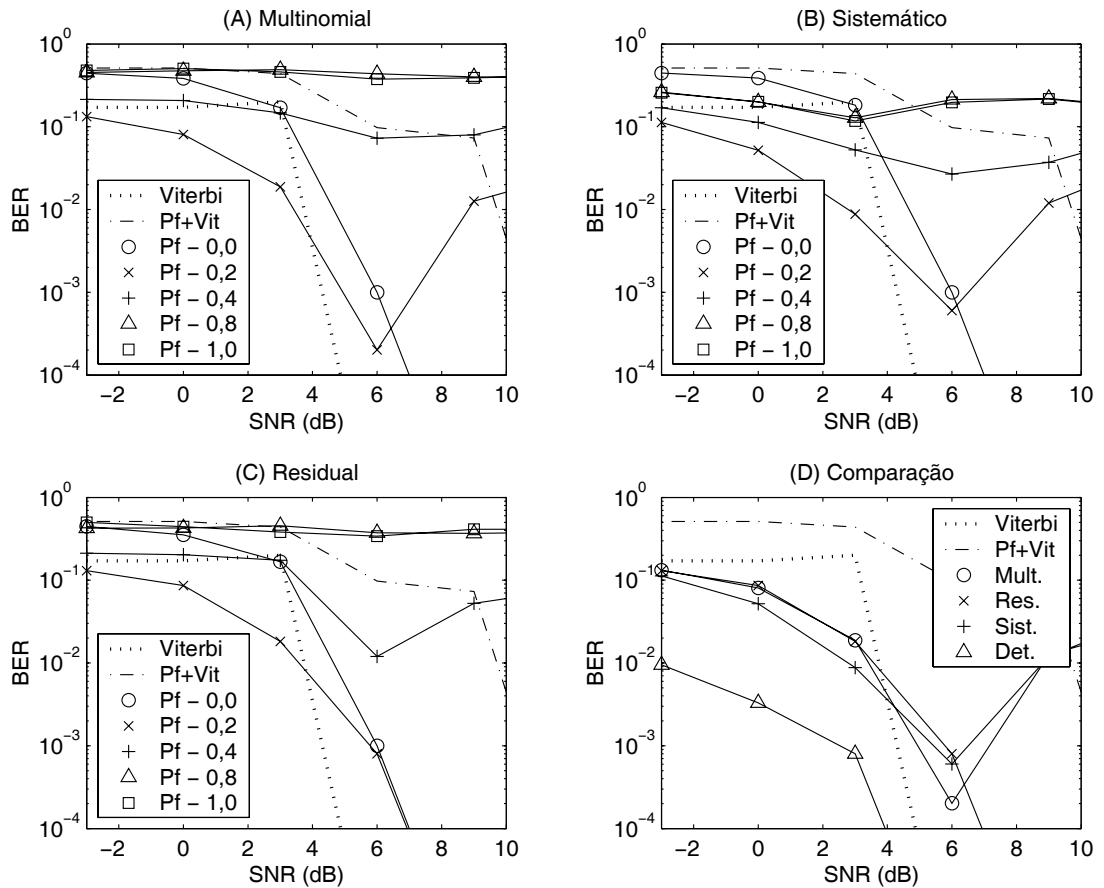


Figura 4.5: Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (17, 12, 04). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.

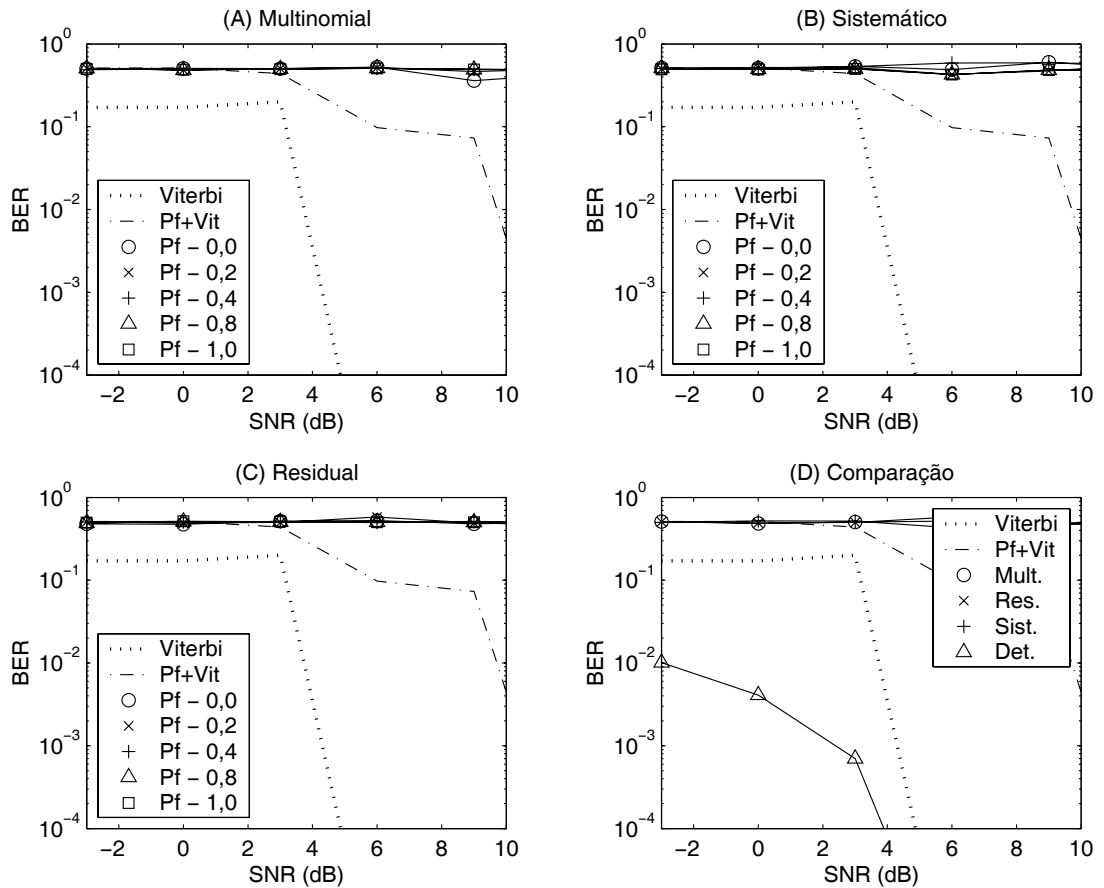


Figura 4.6: Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (63, 44, 17). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.

4.4.2 Sistema empregando codificação diferencial

Nesta seção, analisamos o desempenho dos algoritmos da Sec. 4.3 quando empregados em sistemas de comunicação utilizando codificação diferencial em cascata com o codificador convolucional (Fig. 4.7), esquema que permite o emprego de códigos convolucionais com possíveis invariâncias rotacionais, uma vez que a ambigüidade de fase é resolvida através da codificação diferencial.

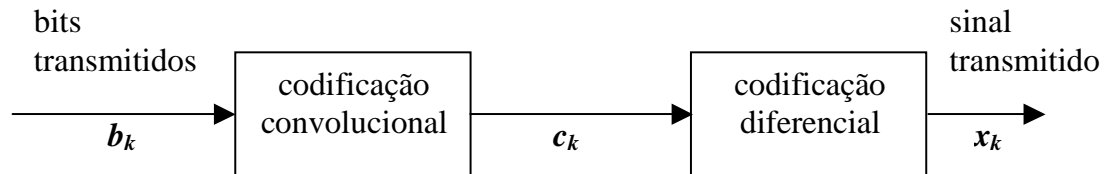


Figura 4.7: Sistema de comunicação empregando codificação diferencial em cascata com um codificador convolucional.

Os resultados a seguir se referem a códigos com propriedades de distância ótimas, com taxas $R = 1/2$ e $R = 1/3$, utilizando as mesmas premissas da Sec. 4.4. Como se pode observar, os resultados obtidos são bastante semelhantes aos descritos na Sec. 4.4.1 para os códigos com $K = 2$, diferindo no entanto para os códigos com $K = 5$: enquanto os algoritmos estocásticos tiveram um comportamento parecido aos dos casos anteriores para o código $(53, 75)$, as taxas de erro tiveram um comportamento não-monotônico para o código $(47, 53, 75)$, mesmo ao se empregar um número mais elevado de partículas (até $N = 1000$, não mostrado na figura).

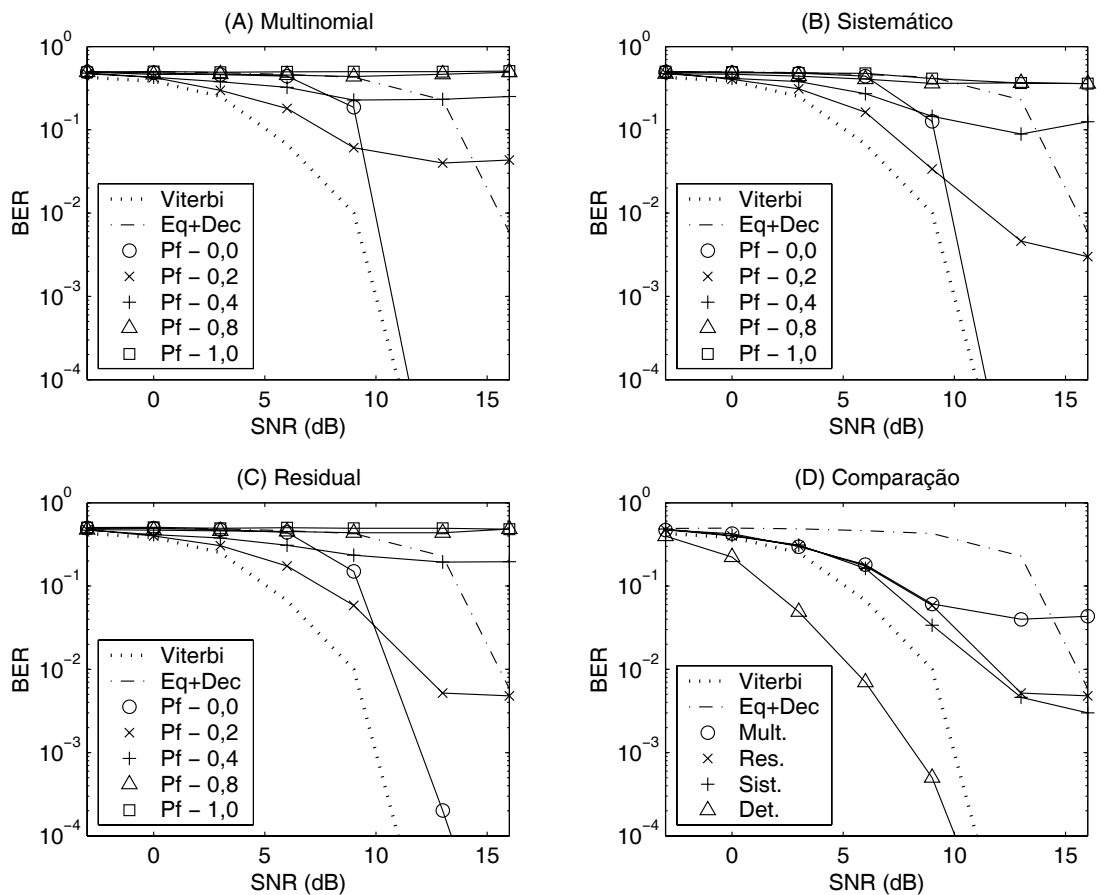


Figura 4.8: Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (5, 7). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.

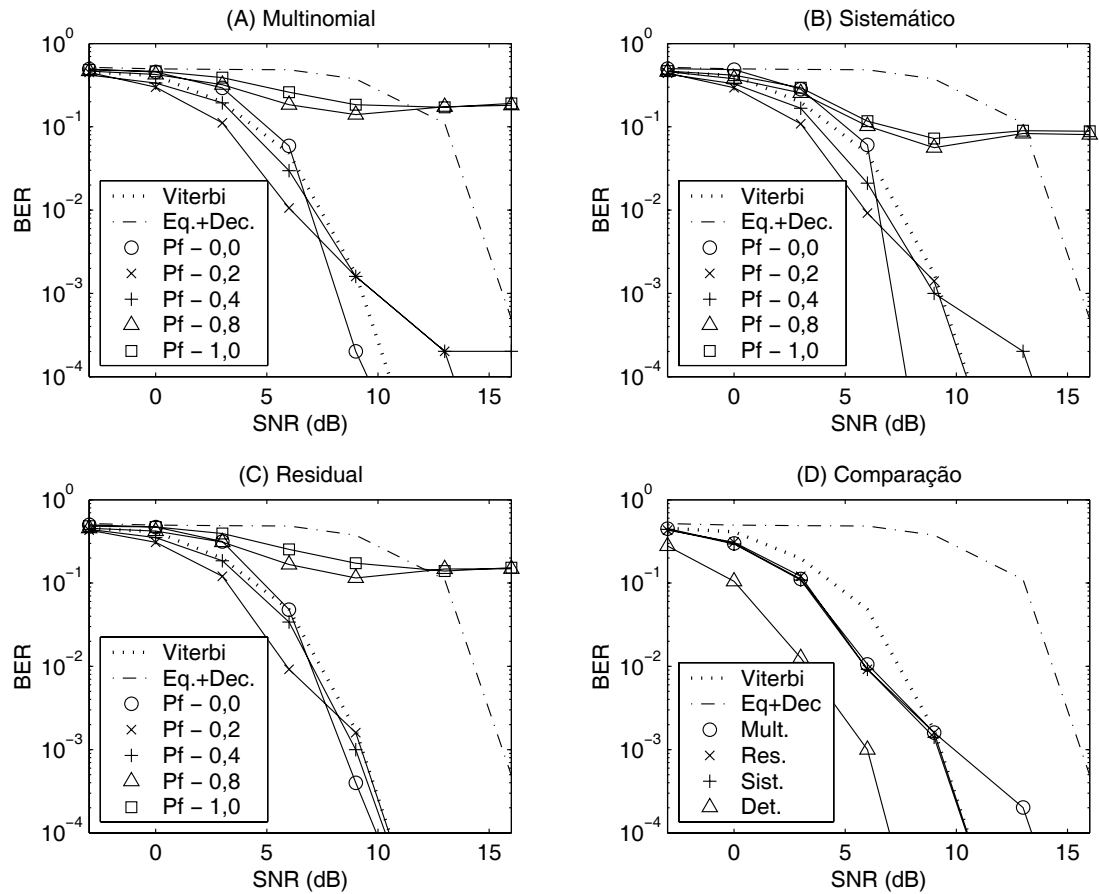


Figura 4.9: Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (5, 7, 7). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.

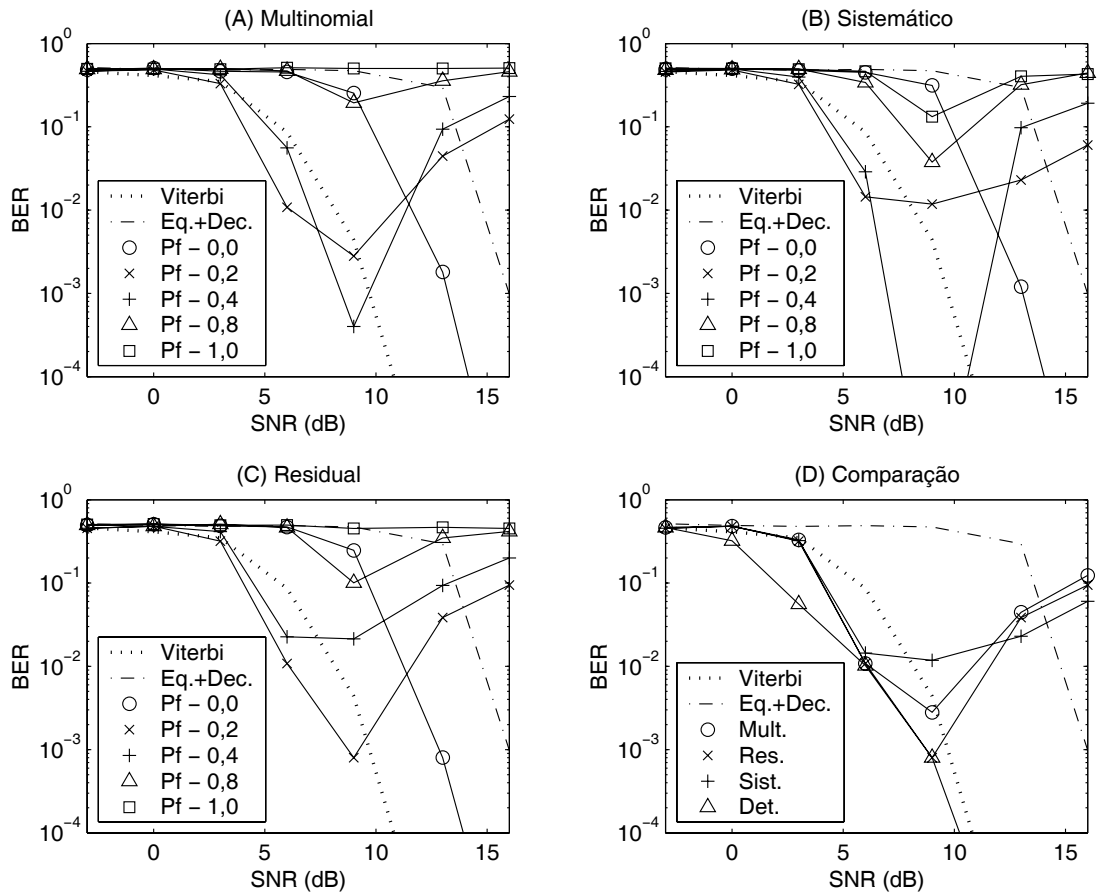


Figura 4.10: Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (53, 75). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.

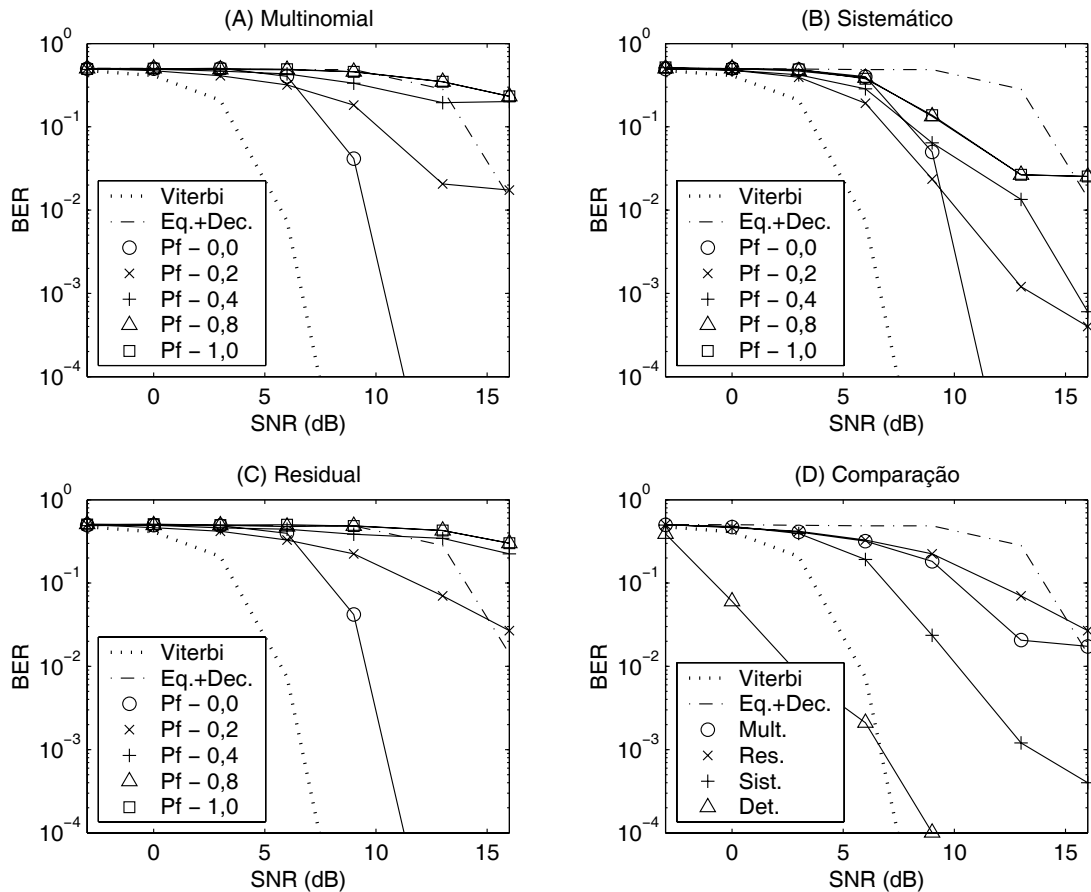


Figura 4.11: Desempenho médio ao longo de 250 realizações independentes de diversos algoritmos para equalização e decodificação conjunta em função da relação sinal-ruído (SNR) e do limiar para reamostragem para um sistema empregando o código convolucional (47, 53, 75). (A) Desempenho dos algoritmos empregando reamostragem multinomial, comparado com os esquemas alternativos descritos no texto. (B) O mesmo para reamostragem residual. (C) O mesmo para reamostragem sistemática. (D) Resultados dos últimos 3 experimentos com limiar igual à 0,2 e do filtro determinístico.

4.4.3 Códigos Convolucionais Recursivos Sistemáticos

Sob as mesmas condições dos experimentos anteriores, realizamos simulações numéricas para avaliar o desempenho dos métodos propostos na Sec. 4.3 quando empregados em sistemas de comunicação que utilizam códigos recursivos sistemáticos. Em particular, selecionamos o código $R = 1/2$ descrito em notação polinomial como

$$\left[1, \frac{1 + D + D^2}{1 + D^2} \right]. \quad (4.14)$$

Para permitir comparações, mostramos também o desempenho de um método alternativo (MAP), consistindo da cascata do equalizador ótimo MAP (utilizando os parâmetros corretos do canal) com o decodificador ótimo MAP (sem realimentação).

Na Fig. 4.12, mostramos a taxa média de erro obtida pelos métodos propostos. Nesta simulação, todos os filtros de partículas empregam $N = 250$ partículas. Diferentemente das simulações anteriores, consideramos aqui o uso de métodos de reamostragem estocástica em conjunto com o algoritmo SIR determinístico. Como se pode verificar, os métodos baseados no filtro de partículas determinístico (DET) tiveram mais uma vez um desempenho superior, sendo o melhor desempenho atingido pelo método utilizando reamostragem determinística. Observe, no entanto, que o desempenho dos métodos propostos deixa de melhorar para SNR superiores a 10 dB, efeito que, conforme observado, persiste mesmo ao se aumentar o número de partículas utilizado.

Na Fig. 4.13 mostramos o desempenho atingido sob as mesmas condições, empregando agora suavização (10 amostras). Neste caso, o desempenho praticamente não se alterou exceto para o algoritmo determinístico (DET-DET), cujo desempenho foi bastante melhorado, aproximando-se daquele do método treinado (MAP).

Finalmente, na Fig. 4.14, avaliamos o desempenho do algoritmo determinístico (DET-DET) em função do número de partículas N . Como se pode observar, o desempenho do método proposto melhora significativamente ao se aumentar o número de partículas para $N = 100$, permanecendo no entanto praticamente estável quando mais partículas são acrescentadas.

4.5 Comentários

Neste capítulo, propusemos duas famílias de algoritmos para equalização e decodificação conjuntas de canais de comunicação seletivos em frequência. Na primeira família, adequada a sistemas utilizando códigos em bloco sistemáticos, os algoritmos simulam a densidade *a posteriori* do sinal codificado x_n , enquanto na segunda família, obtêm-se estimativas da densidade *a posteriori* desejada diretamente, às custas de uma

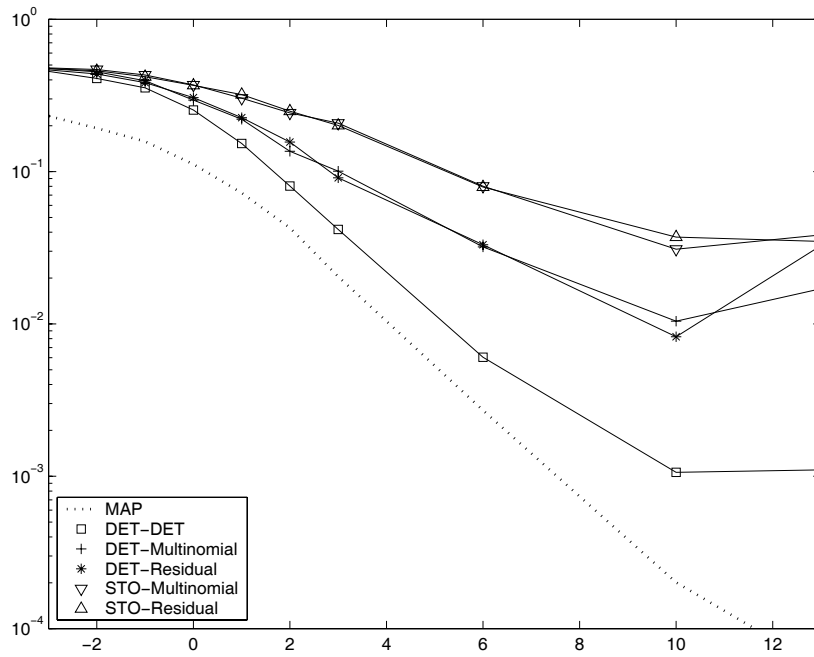


Figura 4.12: Desempenho dos métodos para equalização e decodificação conjunta baseados em filtros de partículas determinístico (DET) e estocástico (STO) em função da SNR, empregando os métodos de reamostragem multinomial, residual e determinístico (DET-DET), e do método alternativo MAP descrito no texto. Os filtros de partículas estocásticos empregam a função de importância ótima.

complexidade computacional mais elevada.

De uma maneira geral, observamos que os métodos propostos (especialmente aqueles empregando filtros de partículas determinísticos) exibem desempenhos surpreendentes, comparáveis aos de métodos treinados. Devido à falta de tempo, uma comparação destes métodos com esquemas iterativos (*turbo equalization*) ficou faltando, apesar das evidências apresentadas em [39] de que estes últimos não levam a bons resultados sob as condições testadas.

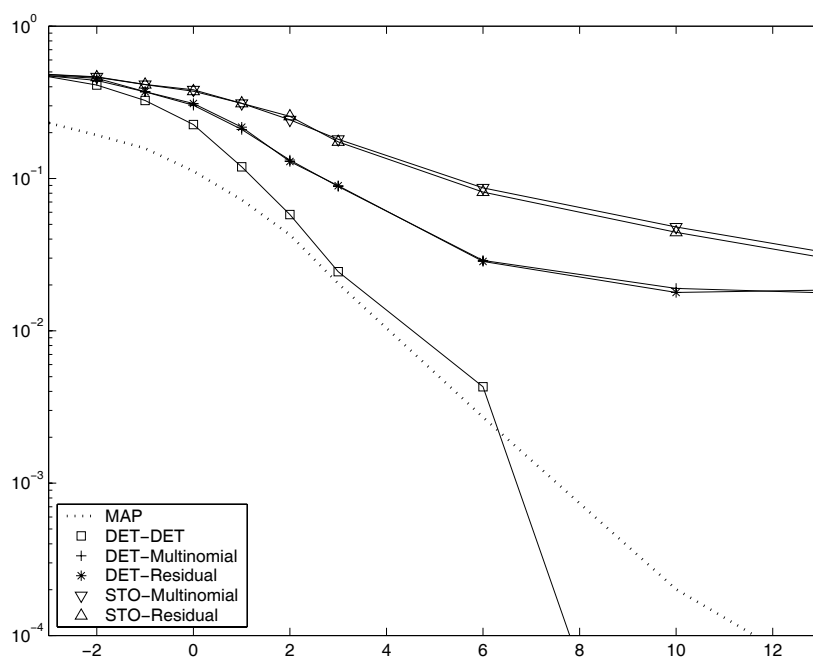


Figura 4.13: Desempenho dos métodos para equalização e decodificação conjunta baseados em filtros de partículas determinístico (DET) e estocástico (STO) em função da SNR, empregando os métodos de reamostragem multinomial, residual e determinístico (DET-DET), empregando suavização (10 amostras) e do método alternativo MAP descrito no texto. Os filtros de partículas estocásticos empregam a função de importância ótima.

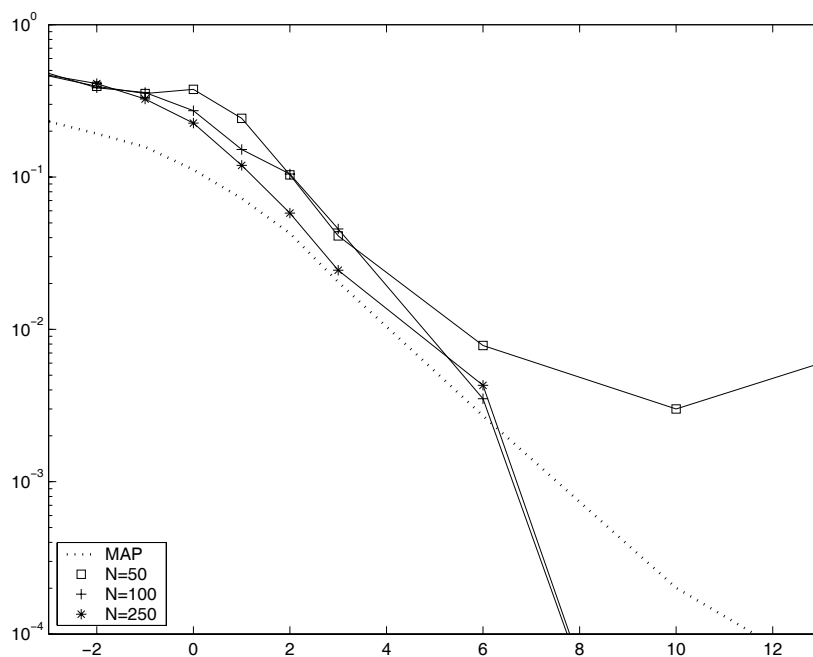


Figura 4.14: Desempenho do algoritmo para equalização e detecção conjunta determinístico (empregando reamostragem determinística e suavização de 10 amostras) e do método alternativo MAP descrito no texto em função do número de partículas empregado e da SNR.

Capítulo 5

Conclusões

Nesta tese, propusemos e analisamos o desempenho de algoritmos para equalização cega baseados em métodos numéricos de estimação Bayesiana. No Cap. 2 descrevemos o algoritmo SIR, estrutura particular de filtragem de partículas empregada no Cap. 3 para a equalização de canais e, no Cap. 4, para a solução conjunta dos problemas de equalização e decodificação. De forma complementar, no Apêndice A avaliamos o desempenho dos algoritmos descritos no Cap. 3 para canais de ordem desconhecida e variantes no tempo. Nos Apêndices B e C, finalmente, analisamos algoritmos para equalização baseados em outras técnicas de estimação Bayesiana - o algoritmo EM e o Amostrador de Gibbs.

Embora o conteúdo de todos os capítulos contenha resultados originais (ao menos no tocante à forma de análise de métodos já existentes), as principais contribuições deste trabalho concentram-se nos Cap. 3 e 4, consistindo principalmente do seguinte:

1. Desenvolvimento de novos métodos para a solução do problema da ambigüidade de fase, através de novos algoritmos de equalização (Cap. 3) e da seleção de códigos com variâncias rotacionais (Cap. 4).
2. Desenvolvimento de algoritmos para equalização cega sob ruído aditivo não-gaussiano (Cap. 3).
3. Desenvolvimento de algoritmos para equalização e decodificação conjuntas em canais seletivos em frequência (Cap. 4).

De uma maneira geral, observamos que os algoritmos para equalização (e decodificação conjuntas) baseados no algoritmo SIR determinístico exibem o melhor desempenho entre os métodos testados, freqüentemente equiparando-se ao desempenho ótimo (MAP), sendo portanto a melhor escolha quando o modelo de sinal empregado permitir a Rao-Blackwellização dos parâmetros do canal.

Como observamos no Cap. 3, no entanto, o algoritmo SIR determinístico tem um comportamento marcadamente distinto dos demais algoritmos. Primeiramente, observamos em muitas situações a existência de um valor ótimo para o número de partículas empregado, a partir do qual o desempenho do método passa a se degradar. Além disso, o algoritmo determinístico exibiu uma piora de desempenho mais acentuada que as demais técnicas em situações nas quais o modelo de sinal informado ao algoritmo divergia do modelo real, o que traz dúvidas sobre a aplicabilidade destes métodos em situações práticas. Apesar destes resultados muitas vezes serem compreensíveis sob a luz da teoria já existente sobre convergência de filtros de partículas [9], falta ainda uma análise de desempenho do algoritmo SIR determinístico pela via analítica. Esta tarefa, infelizmente, demanda o desenvolvimento de um novo arcabouço de ferramentas matemáticas, uma vez que os principais métodos de análise disponíveis ([9, Cap. 2][31]) não podem ser facilmente aplicados neste caso.

Quanto aos resultados referentes à equalização de canais (Cap. 3), surpreendeu-nos a obtenção de taxas médias de erro de bit equivalentes ao se empregar ou não codificação diferencial, bem como o desempenho razoável das técnicas de filtragem de partículas baseadas em evolução artificial para modelos com parâmetros fixos. Estas técnicas, aliás, voltaram a nos surpreender no Apêndice A, onde revelaram ter características transitórias marcadamente insatisfatórias. Naquele mesmo capítulo, observamos também que, apesar dos algoritmos baseados em filtragem de partículas não dependerem do conhecimento exato da ordem do canal para o seu funcionamento, a ambigüidade de atraso inerente ao problema da equalização cega faz com que o desempenho desses algoritmos se degrade fortemente caso a ordem do canal seja sobre-estimada.

No Cap. 4, propusemos duas famílias de algoritmos para a equalização e decodificação conjuntas de canais de comunicação seletivos em frequência, uma delas adequada a sistemas utilizando códigos em bloco e outra, aplicável a uma classe mais ampla de métodos de codificação, às custas, porém, de uma complexidade computacional mais elevada. Observamos que os métodos propostos (especialmente aqueles empregando filtros de partículas determinísticos) exibem desempenhos que muitas vezes excedem os de métodos treinados.

Na Tab. 5.1, mostramos uma comparação da complexidade computacional dos filtros de partículas com outros métodos Bayesianos para equalização cega. Para os parâmetros adotados nas nossas simulações ($L = 3$), os filtros de partículas (Rao-Blackwellizados) exibem uma complexidade notoriamente superior à do algoritmo EM, mas inferior à do Amostrador de Gibbs. Vale observar, porém, que os filtros de partículas levam a um desempenho equivalente ao das demais técnicas (Fig. 5.1) operando com um atraso mínimo (dependente do atraso de suavização escolhido), diferentemente das

demais técnicas, que operam em batelada. Além disso, cabe ressaltar que os filtros de partículas são, dentre estes três métodos, os mais adequados à implementação em computadores paralelos.

Algoritmo	Complexidade	No. de Partículas/ Iterações
EM	$N(oAL^2 + L^3 + o2^L)$	$N \approx 20$
Filtro de Partículas (Determinístico)	$N(oAL^2)$	$N \gg 2^L (N \approx 200)$
Gibbs (Amostrador Conjunto)	$N(oAL^2 + oL^3)$	$N \gg 2^L (N \approx 50)$

Tabela 5.1: Complexidade computacional de diversos algoritmos para equalização cega, em função da ordem do canal a equalizar L , do número de amostras de sinal tratado A , no número de partículas ou iterações N , sendo “ o ” uma constante tal que $1 \leq o \leq 10$.

Como conclusão geral, podemos afirmar que pelo seu desempenho e generalidade, os algoritmos baseados em filtros de partículas constituem uma solução quase que ideal para o problema da equalização cega, reunindo as características de robustez e velocidade de convergência não possuídas por nenhum método baseado exclusivamente em momentos. Isto se verifica, porém, às custas de uma complexidade computacional ainda inadequada para aplicações práticas, desvantagem esta que, felizmente, pode ser revertida num futuro próximo.

5.1 Sugestões para trabalhos futuros

Nesta tese analisamos um tema bastante recente, sobre o qual diversas questões de natureza prática e teórica ainda persistem. Baseados nas constatações que fizemos ao longo da preparação deste trabalho, sugerimos os seguintes tópicos para análise futura:

1. Desenvolvimento de novos algoritmos para equalização de canais com ordem desconhecida.

Neste trabalho, verificamos que algoritmos recém-desenvolvidos para a equalização de canais com ordem desconhecida são afetados por problemas estatísticos de caráter inerente, que levam à sobre-estimação das ordens inferidas. De fato, um problema recorrente [42] que se observa ao se estimar as ordens de modelos por critérios Bayesianos é o de que a verossimilhança de um modelo tende a aumentar com a ordem adotada (mesmo que esta esteja sendo sobre-estimada), fato que aparentemente não foi percebido pelos autores de [1].

Assim, um tópico pertinente para análises futuras seria a investigação dos efeitos i) da escolha de diferentes densidades *a priori* e ii) de funções de importância

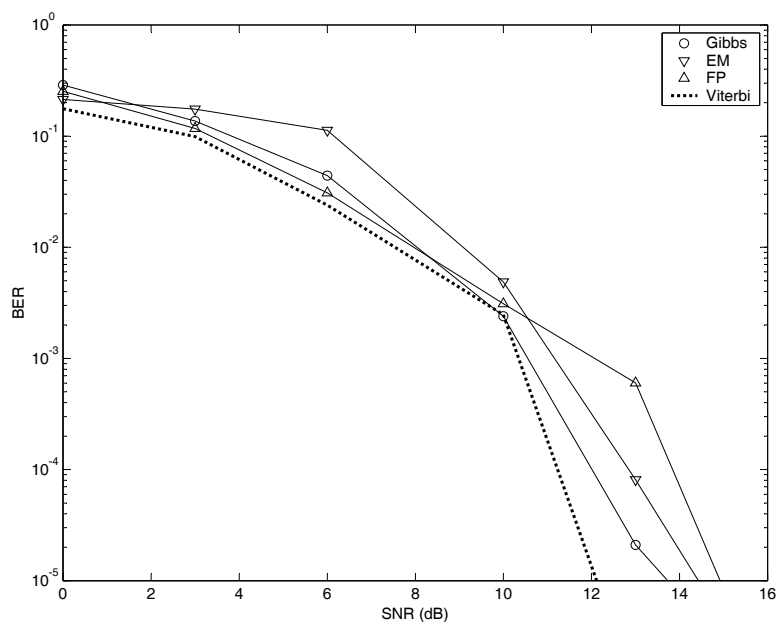


Figura 5.1: Comparação do desempenho atingido por diversos métodos para equalização cega e pelo algoritmo de Viterbi em função da relação sinal-ruído para o canal $h = [0,41 \ 0,82 \ 0,41]^T$. A ordem do canal e a variância do ruído aditivo (gaussiano e real) foram assumidas conhecidas; os algoritmos EM e Gibbs (amostrador conjunto) operam sobre blocos de 250 amostras do sinal recebido, descartando as 25 de cada extremidade para fins de cálculo de taxa de erro. O algoritmo baseado no filtro de partículas (estocástico - com função de importância ótima) também opera sobre 250 amostras, descartando as 100 primeiras, e emprega suavização (5 amostras). O sinal transmitido é binário e emprega codificação diferencial.

sobre o desempenho de algoritmos para equalização de canais de ordem desconhecida, como os propostos em [1].

2. Análise da convergência dos algoritmos para equalização cega.

Dada a novidade do interesse no uso de filtros de partículas para equalização cega, muitos dos métodos propostos não tiveram suas propriedades transitórias devidamente analisadas. Assim, análises numéricas quanto i) a velocidade de convergência e ii) probabilidade de divergência destes algoritmos, em função do número de amostras tratadas, do número de partículas empregadas, e da função de importância empregada produziram resultados inéditos.

Apesar do fato de que simulações numéricas não permitem que se tirem conclusões gerais sobre a convergência de algoritmos da classe dos estudados, os resultados obtidos nestas simulações serviriam de contraponto à falta de resultados teóricos que garantam (ou descartem) a convergência uniforme das soluções geradas por filtros de partículas quando utilizados em modelos como os tratados [31] (com parâmetros fixos e desconhecidos).

3. Desenvolvimento de algoritmos para equalização empregando momentos dos sinais recebidos.

Apesar das suspeitas introduzidas em [31] sobre a possibilidade de não-convergência uniforme de algoritmos baseados em filtros de partículas quando empregados em modelos com parâmetros fixos, um fato bastante conhecido é o de que os momentos do sinal recebido podem ser estimados com precisão crescente em função do tamanho do bloco de sinal observado (e que estas estimativas convergem com probabilidade 1 para seus valores corretos). Como os parâmetros do canal de transmissão são unicamente determinados pelos momentos de 3ª ordem ou superior do sinal recebido, outro tópico de interesse seria o desenvolvimento de algoritmos que empregassem momentos como meio de auxílio à convergência das estimativas.

4. Desenvolvimento de novos algoritmos para equalização de canais baseados na técnica de evolução artificial de parâmetros.

Trata-se de um tópico relativamente pouco explorado, já que a maioria dos trabalhos publicados na área empregam filtros de partículas Rao-Blackwellizados. Neste trabalho, verificamos que métodos baseados em evolução artificial podem fornecer resultados semelhantes aos dos algoritmos Rao-Blackwellizados [10] desde que o número de partículas empregado seja bem mais elevado, o que não é necessariamente desvantajoso, uma vez que a complexidade computacional por partícula destes últimos algoritmos é, em geral, bastante superior.

Desta forma, um tópico de interesse para análises futuras seria o desenvolvimento de algoritmos baseados em evolução artificial que empreguem núcleos de transição diferentes dos comumente utilizados (gaussianos com variância fixa), bem como uma investigação dos efeitos do uso de núcleos com outras distribuições, ou que modificassem adaptativamente a sua variância (utilizando técnicas de *annealing*, por exemplo).

Referências Bibliográficas

- [1] J. Miguez and P. M. Djuric, “Blind equalization of frequency-selective channels by sequential importance sampling,” *IEEE Transactions on Signal Processing*, vol. 52, Outubro 2004.
- [2] Y. Sato, “A Method for Self-Recovering Equalization for Multilevel Amplitude Modulation System,” *IEEE Transactions on Communications*, vol. 23, pp. 679–682, 1975.
- [3] S. Haykin, ed., *Blind Deconvolution*. Englewood Cliffs, NJ: Prentice Hall, 1994.
- [4] Z. Ding and Y. Li, eds., *Blind Equalization and Identification*. New York: Marcel Dekker, 2001.
- [5] L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley Pub. Co., 1991.
- [6] C. J. Bordin Jr., “*Comparação entre algoritmos de Identificação Não-Assistida de Canais de Comunicação Baseados em Estatísticas de Segunda Ordem.*” Dissertação (mestrado). Escola Politécnica da Universidade de São Paulo, 2002.
- [7] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.
- [8] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Berlin: Springer-Verlag, 1999.
- [9] A. Doucet, J. de Freitas, and N. Gordon, eds., *Sequential Monte-Carlo Methods in Practice*. Berlin: Springer-Verlag, 2000.
- [10] J. S. Liu and R. Chen, “Blind Deconvolution via Sequential Imputations,” *Journal of the American Statistical Association*, vol. 90, pp. 567–576, Junho 1995.

- [11] T. Clapp, *Statistical Methods for the Processing of Communication Data*. PhD thesis, University of Cambridge, 2000.
- [12] E. Punskeya, *Sequential Monte Carlo Methods for Digital Communications*. PhD thesis, University of Cambridge, 2003.
- [13] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, “Particle Filtering,” *IEEE Signal Processing Magazine*, vol. 20, pp. 19–38, Setembro 2003.
- [14] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking,” *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, Fevereiro 2002.
- [15] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Upper Saddle River, NJ: Prentice Hall, 1 ed., 2000.
- [16] G. Lee, S. B. Gelfand, and M. P. Fitz, “Bayesian Techniques for Blind Deconvolution,” *IEEE Transactions on Communications*, vol. 44, pp. 826–835, Julho 1996.
- [17] H. L. van Trees, *Detection, Estimation, and Modulation Theory, Part I*. Wiley, 1968.
- [18] A. Doucet, “On Sequential Simulation-Based Methods for Bayesian Filtering,” Tech. Rep. CUED/F-INFENG/TR.310, University of Cambridge, 1998.
- [19] M. A. Tanner, *Tools for Statistical Inference*. Springer Verlag, 3 ed., 1996.
- [20] C. J. Bordin Jr. and L. A. Baccalá, “Particle Filters for Blind FIR Channel Equalization in Non-Gaussian Noise,” in *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2006*, (Toulouse - França), 2006. Trabalho aceito para publicação.
- [21] C. J. Bordin Jr. and L. A. Baccalá, “Joint Blind Equalization and Decoding using Particle Filters,” in *Proceedings of the 4th IEEE International Symposium on Signal Processing and Information Technology - ISSPIT 2004*, (Roma - Itália), pp. 1–4, 2004.
- [22] C. J. Bordin Jr. and L. A. Baccalá, “Particle filter algorithms for joint blind equalization/decoding of convolutionally coded signals,” in *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2005*, vol. 3, (Philadelphia, PA - USA), pp. 497–500, 2005.

- [23] C. J. Bordin Jr. and L. A. Baccalá, “Deterministic Particle Filters for Joint Blind Equalization and Decoding on Frequency Selective Channels,” in *IEEE Statistical Signal Processing Workshop - IEEE SSP 05*, (Bordeaux - França), 2005. Trabalho aceito para publicação.
- [24] C. J. Bordin Jr. and L. A. Baccalá, “Equalização Cega de Canais Lineares FIR empregando o Algoritmo EM Bayesiano,” in *Anais do XXII Simpósio Brasileiro de Telecomunicações*, (Campinas - SP), 2005.
- [25] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to non-linear non-gaussian bayesian state estimation,” *IEE Proceedings-F Image & Signal Processing*, vol. 140, pp. 107–113, Abril 1993.
- [26] A. Kong, J. S. Liu, and W. H. Wong, “Sequential Imputations and Bayesian Missing Data Problems,” *Journal of the American Statistical Association*, vol. 89, pp. 278–288, Março 1994.
- [27] W. G. Cochran, *Sampling Techniques*. New York: John Wiley and Sons, 3 ed., 1977.
- [28] J. D. Hol, “Resampling in particle filters,” Tech. Rep. LiTH-ISY-EX-ET-0283-2004, Institutionen for systemteknik - Linkopings Universitet, 2004.
- [29] R. Chen, J. S. Liu, and X. Wang, “Convergence Analyses and Comparisons of Markov Chain Monte Carlo Algorithms in Digital Communications,” *IEEE Transactions on Signal Processing*, vol. 50, pp. 255–270, Fevereiro 2002.
- [30] J. K. Tugnait and A. H. Haddad, “Detection-Estimation Scheme for State Estimation in Switching Environments.,” *Automatica*, vol. 15, no. 4, pp. 477–481, 1979.
- [31] D. Crisan and A. Doucet, “A Survey of Convergence Results on Particle Filtering Methods for Practicioners,” *IEEE Transactions on Signal Processing*, vol. 50, pp. 736–746, Março 2002.
- [32] S. J. Julier and J. K. Uhlmann, “A New Extension of the Kalman Filter to Nonlinear Systems,” in *In Proc. of AeroSense: The 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls*, (Orlando, FL), SPIE, 1997.
- [33] F. Sun and H. Leib, “Multiple-Phase Codes for Detection Without Carrier Phase Reference,” *IEEE Transactions on Information Theory*, vol. 44, pp. 1477–1491, Julho 1998.

- [34] E. Punskeya, C. Andrieu, A. Doucet, and W. J. Fitzgerald, "Particle Filtering for Demodulation in Fading Channels with non-Gaussian Additive Noise," *IEEE Trans. on Comm.*, vol. 49, pp. 579–582, Abril 2001.
- [35] A. Farina, A. Russo, F. Scannapieco, and S. Barbarossa, "Theory of Radar Detection in Coherent Weibull Clutter," *IEE Proc. F*, vol. 134, no. 2, pp. 174–190, 1987.
- [36] L. Gang and K. Yu, "Modelling and Simulation of Coherent Weibull Clutter," *IEE Proc. F*, vol. 136, pp. 1–12, Fevereiro 1989.
- [37] J. G. Proakis and M. Salehi, *Communication Systems Engineering*. Englewood Cliffs: Prentice Hall, NJ, 1994.
- [38] R. Koetter, A. C. Singer, and M. Tuchler, "Turbo Equalization," *IEEE Signal Processing Magazine*, vol. 21, pp. 67–80, Janeiro 2004.
- [39] A. Roumy, I. Fijalkow, and D. Pirez, "Joint equalization and decoding: why choose the iterative solution?," in *IEEE VTS 50th Vehicular Technology Conference - VTC 1999 - Fall.*, vol. 5, pp. 2989–2993, 1999.
- [40] D. Warrier, U. Madhow, and R. Koetter, "Coding for Noncoherent Communication," in *Proceedings of the International Symposium on Information Theory*, (Sorrento - Italy), p. 76, 2000.
- [41] Y. Kofman, E. Zehavi, and S. Shamai, "nd-Convolutional Codes - Part II: Structural Analysis," *IEEE Transactions on Information Theory*, vol. 43, pp. 576–589, Março 1997.
- [42] P. Guttorp, *Stochastic Modeling of Scientific Data*. CRC Press, 1999.
- [43] H. Lütkepohl, *Introduction to multiple time series analysis*. Berlin: Springer-Verlag, 1993.
- [44] K. Dogancay and V. Krishnamurthy, "Blind Identification of Fractionally Spaced Communication Channels with Markov Inputs," *Signal Processing*, vol. 77, pp. 275–288, Setembro 1999.
- [45] M. Nissilä and S. Pasupathy, "Adaptive Bayesian and EM-Based Detectors for Frequency-Selective Fading Channels," *IEEE Transactions on Communications*, vol. 51, pp. 1325–1336, Agosto 2003.

- [46] A. Doucet, C. Andrieu, and R. Urien, “Bayesian Blind and Semi-Blind Equalization of Channels with Markov Inputs,” *IEE Proceedings-F Image & Signal Processing*, vol. 148, no. 4, pp. 269–274, 2001.
- [47] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate,” *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, Março 1974.
- [48] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of State Calculations by Fast Computing Machines,” *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [49] P. Brémaud, *Markov Chains*. Berlin: Springer-Verlag, 1999.
- [50] W. K. Hastings, “Monte Carlo sampling methods using Markov chains and their application,” *Biometrika*, vol. 57, pp. 97–109, 1970.
- [51] G. D. Forney, “The Viterbi Algorithm,” *Proceedings of IEEE*, vol. 61, pp. 268–278, Março 1973.

Apêndice A

Equalização de Canais de Ordem Desconhecida e Variantes no Tempo Empregando Filtros de Partículas

Neste apêndice descrevemos algoritmos para equalização cega aplicáveis a modelos de sinal mais complexos que o tratado nos capítulos anteriores. Na Sec. A.1, avaliamos através de simulações numéricas os efeitos do desconhecimento *a priori* da ordem do canal sobre os algoritmos descritos na Cap. 3, descrevendo em seguida um método recém-proposto para a equalização de canais de ordem desconhecida. Na Sec. A.2, por sua vez, descrevemos versões dos algoritmos do Cap. 3 adequadas a canais variantes no tempo segundo um modelo AR(1), avaliando os seus desempenhos através de simulações numéricas.

A.1 Equalização Cega de Canais de Ordem Desconhecida

Em princípio, os algoritmos para equalização cega descritos no Cap. 3 poderiam ser utilizados sem modificações na equalização de canais com ordem desconhecida, desde que a ordem assumida pelo algoritmo \hat{L} seja maior ou igual à ordem verdadeira do canal L . No entanto, a ambigüidade de atraso, inerente ao problema da equalização cega¹, causa uma severa degradação no desempenho destes métodos, como verificamos

¹Por ambigüidade de atraso, entende-se o fato de que, se $\hat{L} > L$, existem $\hat{L} - L$ vetores \tilde{X}_n e \tilde{h}_n (versões deslocadas (e completadas com zeros) de X_n e h_n , respectivamente) de ordem \hat{L} tais que $\tilde{h}_n^H \tilde{X}_n = h_n^H X_n$. Assim, mesmo que cada partícula $X_{0:n}^{(i)}$ assumo o valor correto da variável estimada, a possibilidade de que cada partícula convirja para um atraso distinto impede a estimativa da densidade *a posteriori* pelos métodos tradicionais.

a seguir através de simulações numéricas.

Estes resultados, também observados em [11], motivaram o desenvolvimento de técnicas alternativas de filtragem de partículas para a estimação conjunta da ordem e do sinal recebido baseadas em métodos MCMC (*reversible jumps*), verificando-se que tais métodos só produziam resultados satisfatórios caso alguns dos símbolos do sinal transmitido fossem assumidos conhecidos (equalização semi-cega). Mais recentemente, os autores de [1] propuseram uma técnica para a solução deste problema inteiramente baseada em filtros de partículas, a qual descrevemos na Sec. A.1.1.

A.1.1 O Algoritmo de Miguez et al.

Em [1] foi proposta uma técnica original para a equalização de canais com ordem desconhecida, baseada na observação de que a variável \hat{L} que representa a ordem do canal pode ser integrada (Rao-Blackwellizada), uma vez que

$$\begin{aligned} p(X_{0:n}|y_{0:n}) &\propto \sum_{\hat{L} \in \mathcal{L}} p(y_n|X_{0:n}, y_{0:n-1}, \hat{L}) p(X_{0:n-1}, \hat{L}|y_{0:n-1}) \\ &= p(X_{0:n-1}|y_{0:n-1}) \sum_{\hat{L} \in \mathcal{L}} p(y_n|X_{0:n}, y_{0:n-1}, \hat{L}) \\ &\quad \times p(\hat{L}|X_{0:n-1}, y_{0:n-1}), \end{aligned} \quad (\text{A.1})$$

em que \mathcal{L} é um conjunto de valores possíveis para a variável \hat{L} . O primeiro termo da soma em (A.1) tem a mesma expressão de $p(y_n|X_{0:n}, y_{0:n-1})$ (Eq. 3.15), e o segundo termo pode ser calculado de forma recursiva como

$$p(L|X_{0:n-1}, y_{0:n-1}) \propto p(y_{n-1}|X_{0:n-1}, y_{0:n-2}, \hat{L}) p(L|X_{0:n-2}, y_{0:n-2}). \quad (\text{A.2})$$

A função de importância ótima pode ser calculada como

$$\pi(X_n|X_{0:n-1}, y_{0:n}) \propto \sum_{\hat{L} \in \mathcal{L}} p(y_n|X_{0:n}, y_{0:n-1}, \hat{L}) p(\hat{L}|X_{0:n-1}, y_{0:n-1}), \quad (\text{A.3})$$

sendo os pesos atualizados como

$$w_n \propto w_{n-1} \sum_{X_n} \sum_{\hat{L} \in \mathcal{L}} p(y_n|X_{0:n}, y_{0:n-1}, \hat{L}) p(\hat{L}|X_{0:n-1}, y_{0:n-1}). \quad (\text{A.4})$$

O algoritmo proposto por Miguez et al. pode ser interpretado, portanto, com um conjunto de $|\mathcal{L}|$ filtros de partículas operando em paralelo, sendo as estimativas produzidas combinadas de acordo com a verossimilhança $p(y_n|X_{0:n}, y_{0:n-1}, \hat{L})$.

A.1.2 Simulações

Realizamos simulações numéricas para avaliar o desempenho dos algoritmos descritos no Cap. 3 e do algoritmo de Míguez et al. na equalização de canais de ordem desconhecida, medindo as taxas médias de erro de bit (BER) obtidas ao longo de 250 realizações independentes. Da mesma forma que no Cap. 3, o sistema de comunicação analisado transmite blocos de 400 símbolos BPSK (± 1) (independentes e equiprováveis) codificados diferencialmente. Em todas as simulações, os 100 primeiros símbolos são descartados para fins de medição das taxas de erro de modo a permitir a convergência dos algoritmos analisados. Nas simulações a seguir, empregamos o canal $h = [0, 41 \ -0, 82 \ 0, 41]^T$ ($L = 3$) e ruído aditivo gaussiano branco (real) de média nula.

Na Figura A.1, mostramos o desempenho médio do algoritmo de Míguez et al. considerando $\mathcal{L} = \{1, 2, 3, 4, 5\}$ e $\mathcal{L} = \{2, 3, 4\}$, comparando-o com o desempenho oferecido pelo algoritmo da Sec. 3.3.1 (SIS estocástico, com função de importância ótima) e pelo algoritmo BCJR. Como se pode observar, o desempenho do algoritmo de Míguez et al. é significativamente pior que o do algoritmo operando com o conhecimento da ordem exata do canal para $\mathcal{L} = \{2, 3, 4\}$, degradando-se ainda mais ao se assumir $\mathcal{L} = \{1, 2, 3, 4, 5\}$.

Uma vez que o algoritmo de Míguez et al. aproxima a densidade *a posteriori* da ordem do canal $p(\hat{L}|X_{0:n}, y_{0:n})$, pode-se estimar esta grandeza de maneira ótima como

$$\hat{L}_n^{MAP} \triangleq \arg \max_{\hat{L}} p(\hat{L}|X_{0:n}, y_{0:n}) .$$

Nas Fig. A.2 e A.3, mostramos o comportamento médio da estimativa \hat{L}_n^{MAP} ao longo de 250 realizações independentes. Em ambos os casos, as estimativas exibem uma grande variabilidade, podendo-se no entanto notar que para níveis altos de ruído há uma tendência maior de se subestimar a ordem do canal, enquanto para níveis baixos de ruído a probabilidade de se sobre-estimar a ordem do canal aumenta, sendo que, para a SNR de 20 dB, o estimador detecta a ordem correta do canal em pouco mais de 40% dos casos.

Uma análise mais detalhada dos resultados das Fig. (A.2)-(A.3) permite concluir que o método de Míguez et al. esbarra numa limitação fundamental de problemas de estimação de ordens de modelos, que é o fato de que a verossimilhança de um modelo tende a crescer como função da ordem adotada, mesmo que esta seja superior à ordem verdadeira. Vale notar que este problema, que é contornado por critérios de estimação de ordem (como o AIC, BIC [43]) através da introdução de uma penalização (que diminui artificialmente a verossimilhança dos modelos na medida em que a ordem

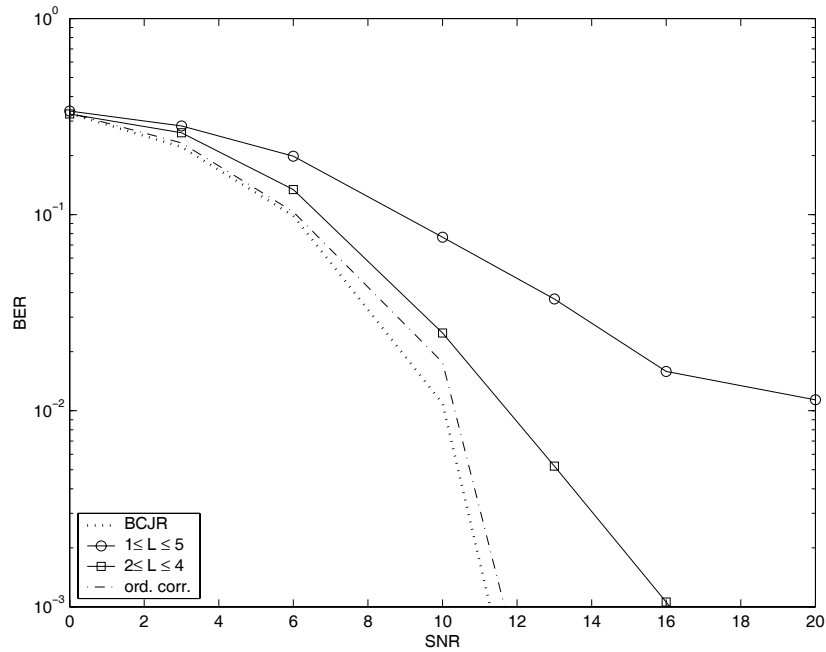


Figura A.1: Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega em canal de ordem desconhecida de Míguez et al. [1], para ordens possíveis $1 \leq \hat{L} \leq 5$ e $2 \leq \hat{L} \leq 4$, comparado com o desempenho do equalizador ótimo MAP (BCJR) e do algoritmo para equalização cega baseado em filtros de partículas estocásticas com a função de importância ótima (utilizando $N = 300$ partículas e suavização (10 amostras)), num sistema BPSK com codificação diferencial, em função da relação sinal-ruído (SNR).

crece), poderia em princípio ser combatido com a utilização de distribuições *a priori* adequadas para a ordem.

Finalizando, nas Fig. A.4 a A.8, mostramos o desempenho médio obtido pelos algoritmos do Cap. 3, empregando $N = 300$ partículas e suavização (5 amostras), em função da ordem adotada para o canal e da relação sinal-ruído. Em todos estes experimentos, observamos que o desempenho obtido é bastante ruim quando a ordem é subestimada. A rapidez com que o desempenho se degrada na medida em que a ordem é aumentada além do seu valor correto, no entanto, varia entre os algoritmos, sendo o algoritmo baseado no filtro de partículas determinístico o menos robusto entre os analisados.

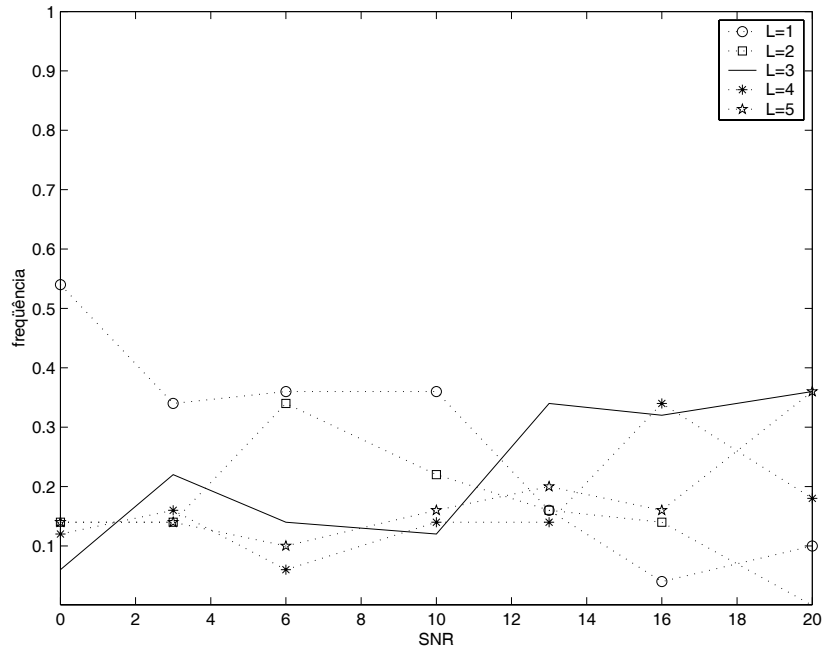


Figura A.2: Histograma de 250 amostras do estimador MAP da ordem do canal provida pelo algoritmo de Miguez et al. com $1 \leq \hat{L} \leq 5$ em função da relação sinal-ruído (SNR). O canal empregado tem ordem $L = 3$.

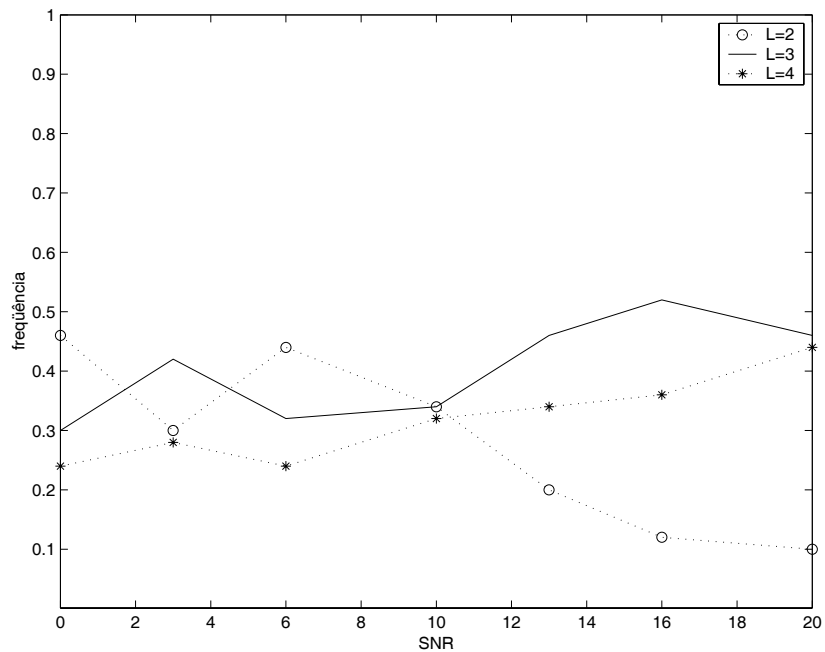


Figura A.3: Histograma de 250 amostras do estimador MAP da ordem do canal provida pelo algoritmo de Miguez et al. com $2 \leq \hat{L} \leq 4$ em função da relação sinal-ruído (SNR). O canal empregado tem ordem $L = 3$.

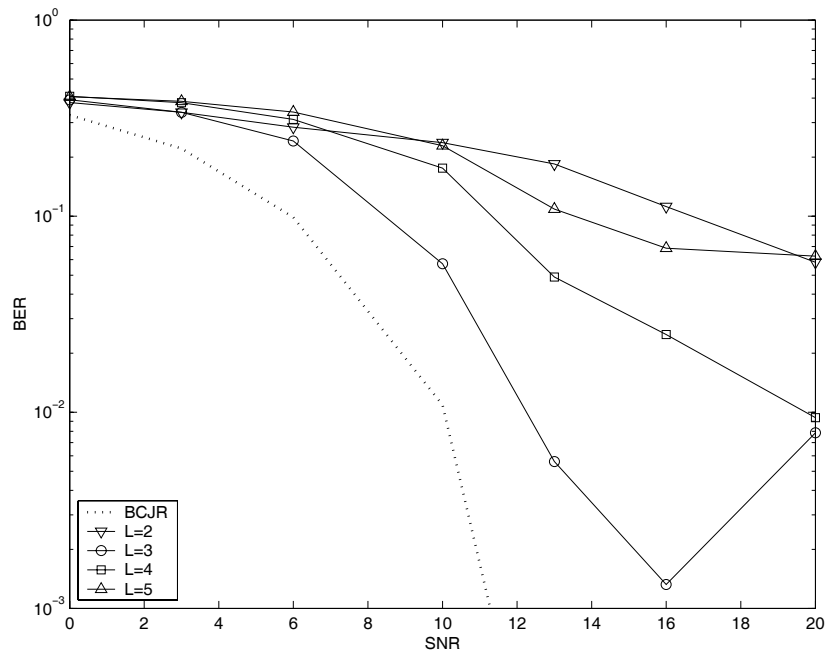


Figura A.4: Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega baseado no princípio da evolução artificial (com $P = 1$, $N = 300$ partículas e suavização de 5 amostras), comparado com o desempenho do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR) e da ordem assumida para o canal.

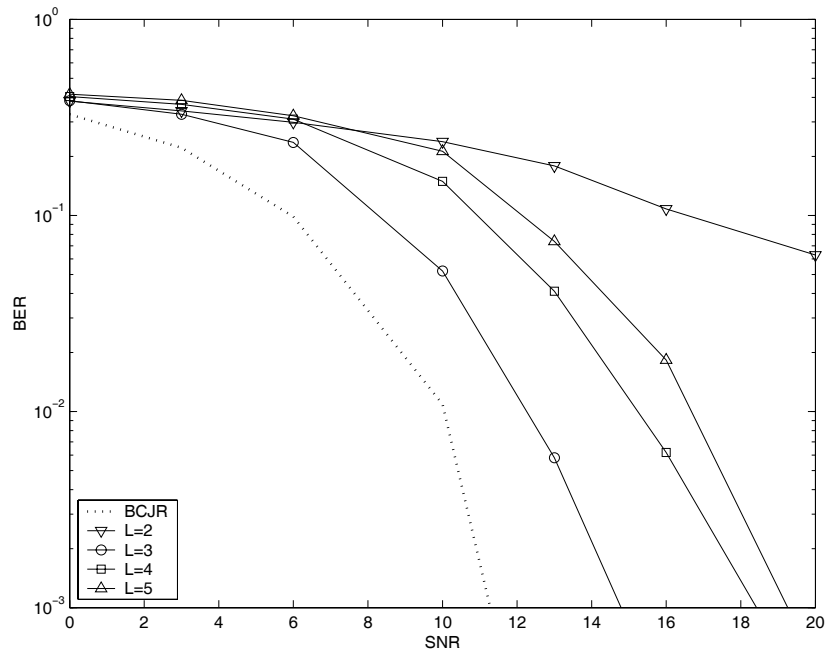


Figura A.5: Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega baseado no princípio da evolução artificial (com $P = 5$, $N = 300$ partículas e suavização de 5 amostras), comparado com o desempenho do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR) e da ordem assumida para o canal.

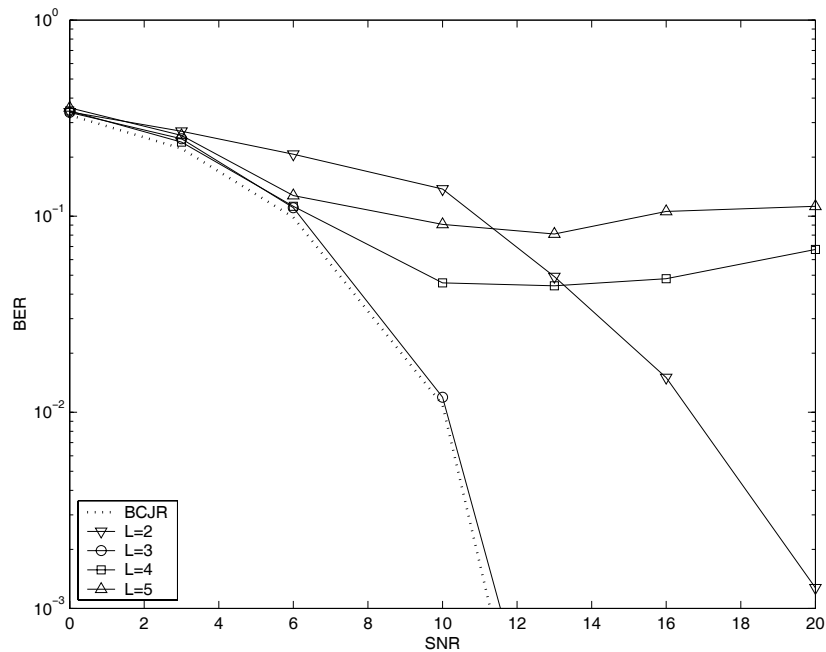


Figura A.6: Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega baseado no filtro de partículas determinístico ($N = 300$ partículas e suavização de 5 amostras), comparado com o desempenho do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR) e da ordem assumida para o canal.

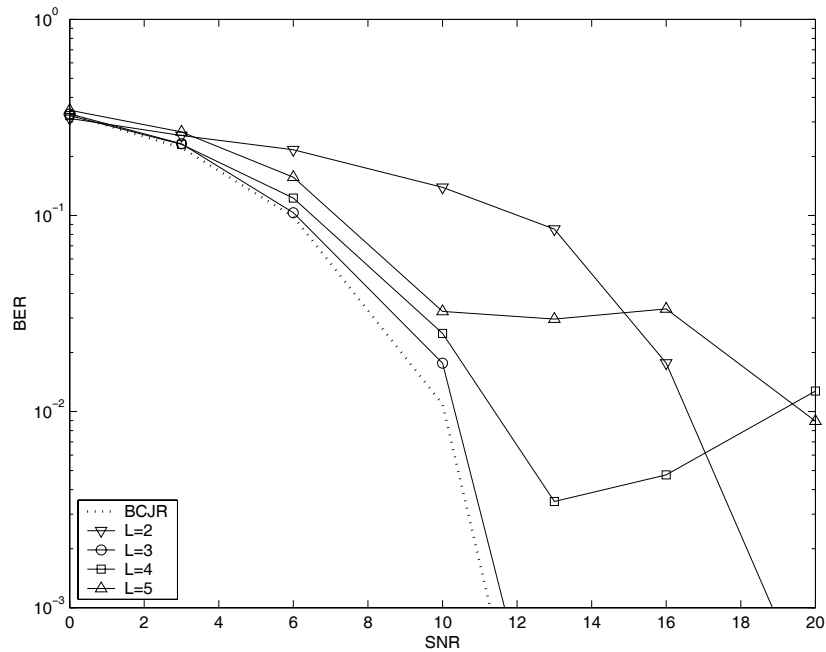


Figura A.7: Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega baseado no filtro de partículas estocástico (utilizando a função de importância ótima, $N = 300$ partículas e suavização de 5 amostras), comparado com o desempenho do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR) e da ordem assumida para o canal.

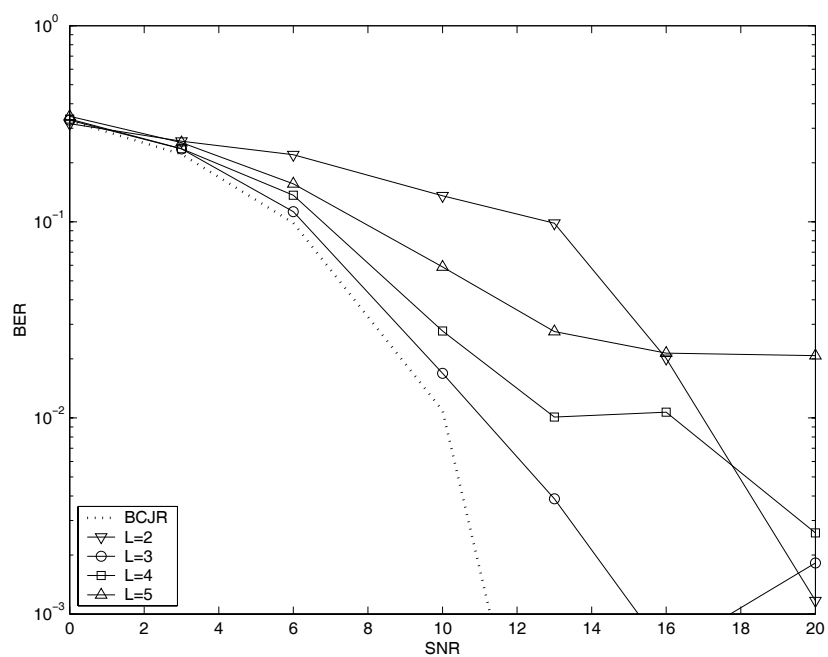


Figura A.8: Desempenho médio ao longo de 250 realizações independentes do algoritmo para equalização cega baseado no filtro de partículas estocástico (utilizando a função de importância *a priori*, $N = 300$ partículas e suavização de 5 amostras), comparado com o desempenho do equalizador ótimo MAP (BCJR), em função da relação sinal-ruído (SNR) e da ordem assumida para o canal.

A.2 Equalização de Canais Variantes no Tempo

Nesta seção, descrevemos como os algoritmos para equalização cega do Cap. 3 podem ser reprojatados para a equalização de canais variantes no tempo. Para tanto, considere inicialmente o modelo de sinal

$$\begin{cases} X_n &= F X_{n-1} + f x_n \\ h_n &= \rho h_{n-1} + \delta u_n \\ y_n &= h_n^H X_n + v_n \end{cases} \quad (\text{A.5})$$

em que F e f são descritos na Sec. 1.1, $u_n \sim \mathcal{N}(u_n|0; I)$ e $v_n \sim \mathcal{N}(v_n|0; \sigma_v^2)$ são processos aleatórios gaussianos brancos e mutuamente independentes, e ρ e δ são constantes reais. Este modelo de sinal permite a aplicação tanto de filtros de partículas Rao-Blackwellizados (Sec. 2.5.1) como de filtros baseados em evolução artificial, conforme mostramos a seguir.

Algoritmos Rao-Blackwellizados

No Apêndice D, deduzimos as densidades necessárias para o projeto de filtros de partículas Rao-Blackwellizados para o modelo de (A.5). Como resultado principal, temos que

$$p(y_n | X_{0:n}, y_{0:n-1}) = \mathcal{N}(y_n | \rho \bar{h}_{n-1}^H X_n; X_n^H (\rho^2 \bar{\Sigma}_{n-1} + I \delta^2) X_n + \sigma_v^2), \quad (\text{A.6})$$

podendo os parâmetros \bar{h}_n e $\bar{\Sigma}_n$ serem calculados através das recursões

$$\begin{cases} \bar{h}_n &= \rho h_{n-1} + \gamma_n^{-1} \bar{\Sigma}_n X_n (y_n - \rho h_{n-1} X_n)^* \\ \bar{\Sigma}_n &= \rho^2 (\bar{\Sigma}_{n-1} - \gamma_n^{-1} \bar{\Sigma}_{n-1} X_n X_n^H \bar{\Sigma}_{n-1}) + \delta^2 I \end{cases} \quad (\text{A.7})$$

sendo $\gamma_n = \sigma_v^2 + X_n^H \bar{\Sigma}_{n-1} X_n$.

A partir de (A.6), podem-se obter todas as densidades necessárias para o projeto de algoritmos baseados nos algoritmos SIR estocásticos e determinísticos. Quanto aos primeiros, note que a função de importância ótima pode ser obtida a partir da relação

$$p(X_n | X_{0:n-1}, y_{0:n}) \propto p(y_n | X_{0:n}, y_{0:n-1}), \quad (\text{A.8})$$

e os pesos atualizados como

$$w_n \propto w_{n-1} \sum_{X_n} p(y_n | X_n, X_{0:n-1}, y_{0:n-1}). \quad (\text{A.9})$$

Filtros determinísticos podem ser projetados observando que

$$p(X_n, y_n | X_{0:n-1}, y_{0:n-1}) = p(y_n | X_{0:n}, y_{0:n-1})p(X_n | X_{0:n-1}) . \quad (\text{A.10})$$

Evolução Artificial

Diferentemente do caso envolvendo canais com parâmetros fixos, pode-se projetar para o modelo de sinal de (A.5) algoritmos baseados em evolução artificial empregando a densidade de transição verdadeira dos parâmetros, que neste caso é dada por

$$p(h_n | h_{n-1}) = \mathcal{N}(h_n | \rho h_{n-1}; I\delta^2) . \quad (\text{A.11})$$

A.2.1 Simulações

Realizamos simulações numéricas para avaliar o desempenho dos algoritmos para equalização cega descritos no Cap. 3 adaptados para o modelo de sinal de (A.5), medindo as taxas médias de erro de bit (BER) obtidas ao longo de 250 realizações independentes. O sistema de comunicação analisado transmite blocos de 400 símbolos BPSK (± 1) (independentes e equiprováveis) codificados diferencialmente. Em todas as simulações, os 100 primeiros símbolos são descartados para fins de medição das taxas de erro de modo a permitir a convergência dos algoritmos analisados, e os filtros de partículas são inicializados de forma aleatória, com $h_0^{(i)} \sim \mathcal{N}(h_0 | 0; I)$ e $X_0 \sim \mathcal{B}(\pm 1)$.

Nas simulações a seguir, o canal h_n é iniciado com o valor $h_0 = [0, 41 \quad -0, 82 \quad 0, 41]^T$ ($L = 3$), e é (sem perda de generalidade) normalizado a cada iteração, de modo que a relação sinal-ruído se mantenha estável. Os parâmetros ρ e δ foram escolhidos de forma que $\delta^2 = 0, 0125$ e $\rho^2 = 1 - \delta^2$, o que faz com que o processo $h_{0:n}$ tenha uma banda passante (Doppler, de 3 dB) igual a $4 \times 10^{-3}(1/T)$.

Na Fig. A.9, mostramos o resultado de uma simulação que ilustra os efeitos da variabilidade do canal sobre o algoritmo para equalização baseado no filtro de partículas determinístico projetado para canais invariantes no tempo, sendo o parâmetro δ do modelo de sinal variado para múltiplos do valor original ($\delta^2 = 0, 0125$). Como se pode observar, o algoritmo projetado em desacordo com o modelo de sinal verdadeiro falha completamente. Na Fig. A.10, por sua vez, mostramos os resultados obtidos para o mesmo algoritmo, empregando agora o modelo de sinal correto, comparados com os resultados do equalizador ótimo MAP: como esperado, o desempenho do equalizador piora na medida em que δ aumenta, distanciando-se do valor ótimo, mesmo para SNR elevadas.

Na Fig. A.11, repetimos o mesmo experimento para o algoritmo baseado no filtro de partículas estocástico (com função de importância ótima), obtendo resultados pi-

ores. Curiosamente, o desempenho deste algoritmo tende a se degradar para valores baixos do parâmetro δ .

Nas Fig. A.12 e A.13, finalmente, mostramos os resultados obtidos pelo algoritmo baseado em evolução de parâmetros. De maneira totalmente surpreendente, estes algoritmos falham, mesmo ao se elevar o número de partículas para $N = 1000$, o que demonstra que, ao contrário da intuição, os métodos de evolução artificial têm um comportamento transitório lento. Vale salientar que testamos o mesmo método empregando a função de importância *a priori*, obtendo resultados até piores.

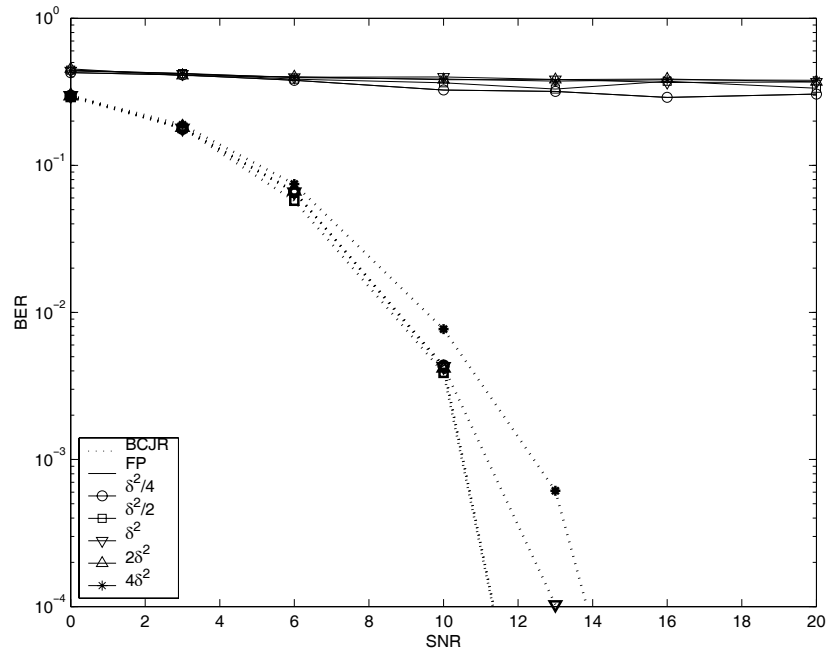


Figura A.9: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (linha cheia) e dos algoritmos para equalização cega baseado nos filtros de partículas determinísticos para canais invariantes no tempo (linha tracejada) (utilizando $N = 300$ partículas e suavização (5 amostras)), em função da relação sinal-ruído (SNR) e do parâmetro δ (indicado com múltiplos do valor $\delta = \sqrt{0,0125}$).

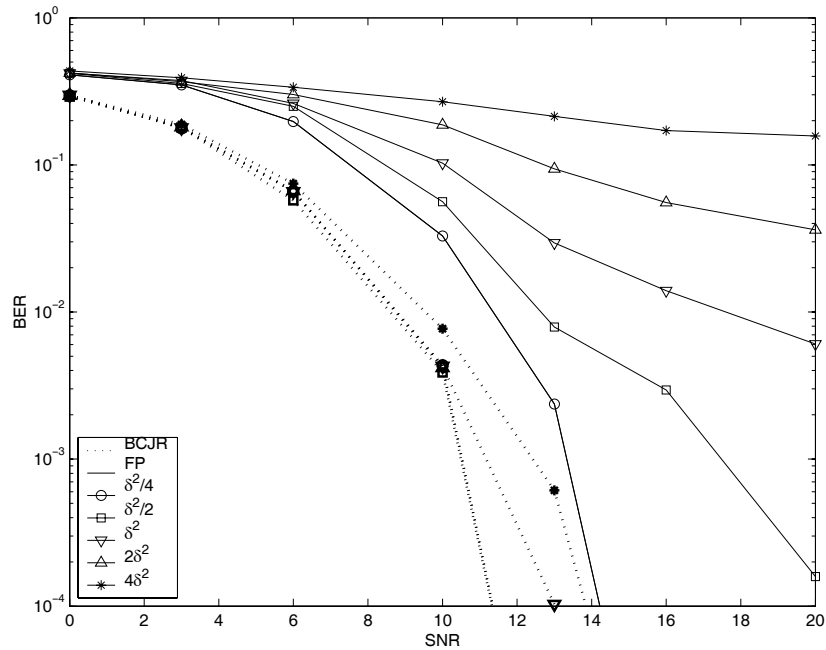


Figura A.10: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (linha cheia) e do algoritmos para equalização cega baseado no filtros de partículas determinístico (linha tracejada)(utilizando $N = 300$ partículas e suavização (5 amostras)), em função da relação sinal-ruído (SNR) e do parâmetro δ (indicado com múltiplos do valor $\delta = \sqrt{0,0125}$).

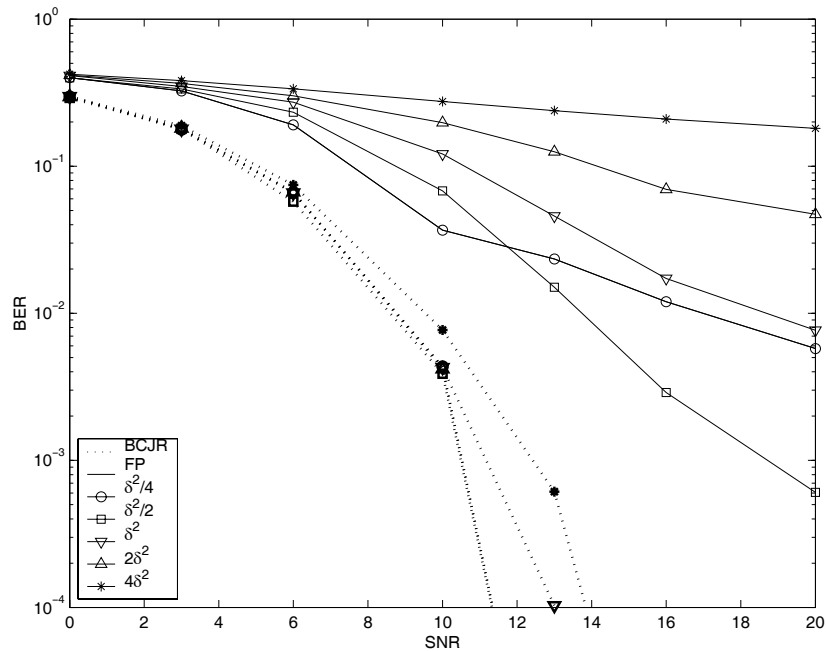


Figura A.11: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (linha cheia) e do algoritmos para equalização cega baseado no filtros de partículas estocástico (linha tracejada)(utilizando a função de importância ótima, $N = 300$ partículas e suavização (5 amostras)), em função da relação sinal-ruído (SNR) e do parâmetro δ (indicado com múltiplos do valor $\delta = \sqrt{0,0125}$).

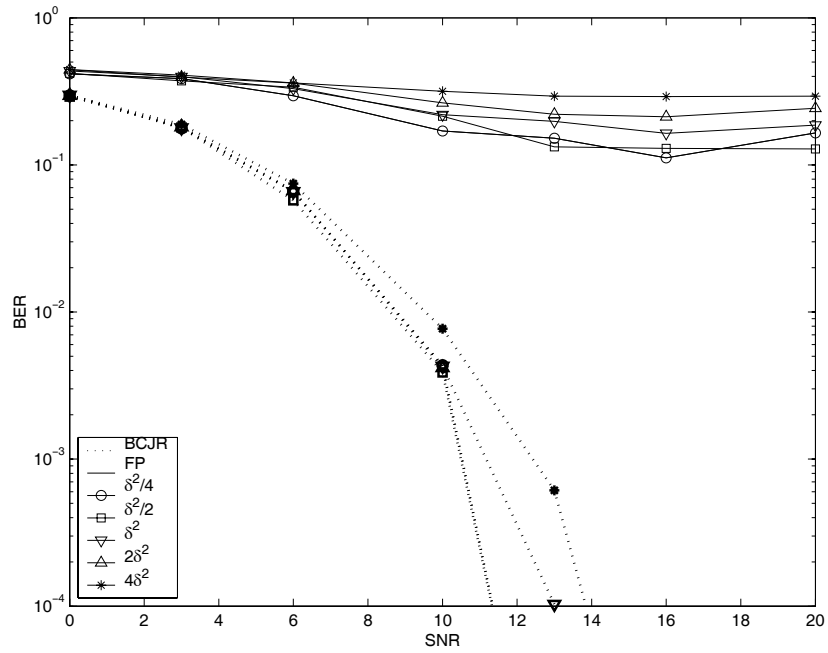


Figura A.12: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (linha cheia) e dos algoritmos para equalização cega baseado no princípio da evolução artificial (linha tracejada) (com $P = 5$, $N = 300$ partículas e suavização (5 amostras)), em função da relação sinal-ruído (SNR) e do parâmetro δ (indicado com múltiplos do valor $\delta = \sqrt{0,0125}$).

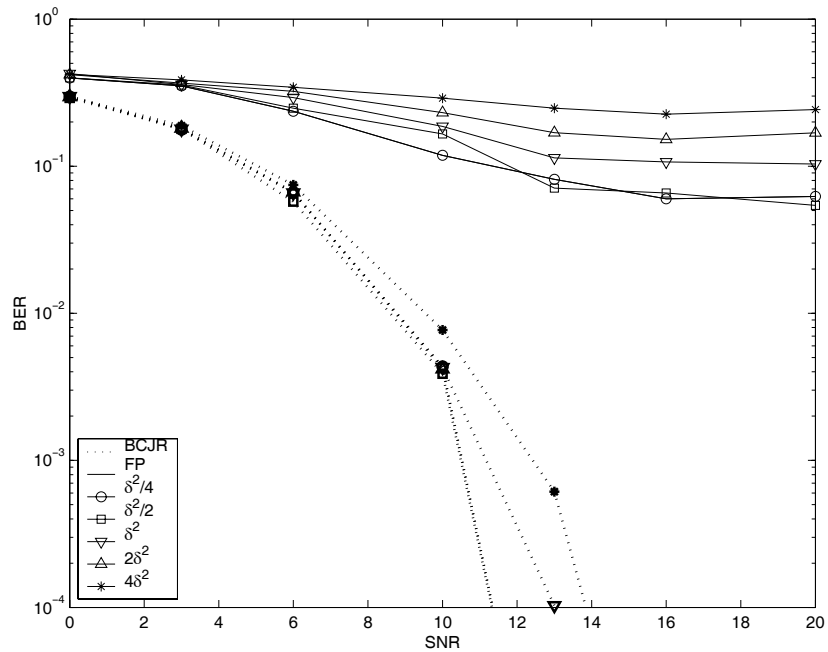


Figura A.13: Desempenho médio ao longo de 250 realizações independentes do equalizador ótimo MAP (linha cheia) e dos algoritmos para equalização cega baseado no princípio da evolução artificial (linha tracejada) (com $P = 5$, $N = 1000$ partículas e suavização (5 amostras)), em função da relação sinal-ruído (SNR) e do parâmetro δ (indicado com múltiplos do valor $\delta = \sqrt{0,0125}$).

Apêndice B

Equalização Cega empregando o Algoritmo EM

Introduzido em 1977 por Dempster *et al.* [7], o algoritmo EM é um método iterativo que produz soluções ótimas (nos sentidos ML ou MAP [5]) para problemas de estimação com dados latentes, ou seja, problemas nos quais a função de verossimilhança (a densidade *a posteriori*) somente é tratável se condicionada a um conjunto de dados não-observados.

Apesar de terem-se desenvolvido diversos métodos de obtenção de soluções ótimas no sentido da máxima verossimilhança (ML) para o problema da equalização cega utilizando o algoritmo EM ([44][45]), apresentamos nesta seção um novo método, desta vez de cunho Bayesiano. Nele obtêm-se aproximações para a estimativa MAP dos parâmetros do canal admitindo-se para isto uma densidade *a priori* gaussiana. Em consequência à análise dos resultados obtidos com esta abordagem, propomos um segundo método modificado para equalização cega empregando uma técnica de *annealing*, que corrige deficiências de desempenho observadas quando a relação sinal-ruído disponível é alta.

O conteúdo a seguir está organizado da seguinte forma: na Sec. B.1 apresentamos o algoritmo EM na sua forma geral. Na Sec. B.2, descrevemos a aplicação do algoritmo EM à solução do problema da equalização cega, avaliando seu desempenho através de simulações numéricas. Na Sec. B.3, finalmente, apresentamos o algoritmo modificado e seu desempenho.

B.1 O algoritmo EM

Dado que se deseje inferir um vetor aleatório θ a partir de um conjunto de observações A e considerando uma variável aleatória B latente (i.e. não-observada), de sorte

que se possa descrever θ por meio de uma $p(\theta|A, B)$ analiticamente simples, o algoritmo EM prescreve como obter uma seqüência de estimativas $\theta^{(i)}$, $i \geq 0$, tais que $p(\theta^{(i+1)}|A) \geq p(\theta^{(i)}|A)$. Em outras palavras, o algoritmo EM permite obter valores $\theta^{(i)}$ que maximizam localmente a densidade *a posteriori* da variável inferida.

Dividido basicamente em dois passos, o algoritmo EM consiste primeiramente (passo E) em determinar a esperança da densidade *a posteriori* aumentada $p(\theta|A, B)$ dadas as observações A e uma estimativa atual $\theta^{(i)}$. A isto segue-se (passo M) o cálculo da nova estimativa de $\theta^{(i+1)}$ obtida ao maximizar-se a esperança determinada no passo E em função de θ , conforme descrito na Tabela B.1.

1. Passo E (*expectation*):

$$Q(\theta, \theta^{(i)}) = \int_B \log p(\theta|A, B)p(B|A, \theta^{(i)})dB \quad (\text{B.1})$$

2. Passo M (*maximization*):

$$\theta^{(i+1)} = \arg \max_{\theta} Q(\theta, \theta^{(i)}) \quad (\text{B.2})$$

3. Itera-se até a convergência, ou seja, até que $\|\theta^{(i+1)} - \theta^{(i)}\|$ seja suficientemente pequeno.

Tabela B.1: Algoritmo EM

Como se pode observar na Tabela B.1, o algoritmo EM é um procedimento bastante geral. Como tal, tem aplicabilidade a diversos problemas de estimação, sendo sua praticidade determinada pela capacidade de se calcular e maximizar $Q(\theta, \theta^{(i)})$ de modo fácil. Cabe salientar que as estimativas obtidas não são necessariamente boas aproximações da estimativa MAP do vetor aleatório θ , já que a convergência do algoritmo só é garantida até máximos locais da densidade *a posteriori* [19], propriedade que demonstramos na proposição a seguir.

Proposição 1 *Iterando-se o algoritmo da Tabela B.1 tem-se que $p(\theta^{(i+1)}|A) \geq p(\theta^{(i)}|A)$.*

Prova: Inicialmente, note que como uma consequência lei de Bayes

$$p(\theta|A, B) = \frac{p(\theta|A)p(B|\theta, A)}{p(B|A)}, \quad (\text{B.3})$$

o que implica que

$$\log[p(\theta|A)] = \log[p(\theta|A, B)] + \log[p(B|A)] - \log[p(B|\theta, A)]. \quad (\text{B.4})$$

Integrando-se cada uma destas densidades com respeito a $p(B|\theta^{(i)}, A)$:

$$\begin{aligned}
\log[p(\theta|A)] &= \int_B \log[p(\theta|A, B)] p(B|\theta^{(i)}, A) dB \\
&+ \int_B \log[p(B|A)] p(B|\theta^{(i)}, A) dB \\
&- \int_B \log[p(B|\theta, A)] p(B|\theta^{(i)}, A) dB \\
&\triangleq Q(\theta, \theta^{(i)}) + K(\theta, \theta^{(i)}) - H(\theta, \theta^{(i)}).
\end{aligned} \tag{B.5}$$

Logo

$$\begin{aligned}
\log[p(\theta^{(i+1)}|A)] - \log[p(\theta^{(i)}|A)] &= \\
&[Q(\theta^{(i+1)}, \theta^{(i)}) - Q(\theta^{(i)}, \theta^{(i)})] \\
&+ [H(\theta^{(i+1)}, \theta^{(i)}) - H(\theta^{(i)}, \theta^{(i)})] \\
&- [K(\theta^{(i+1)}, \theta^{(i)}) - K(\theta^{(i)}, \theta^{(i)})].
\end{aligned} \tag{B.6}$$

Como, por definição, $\theta^{(i+1)}$ é escolhido de modo a maximizar $Q(\theta^{(i+1)}, \theta^{(i)})$, o primeiro termo em (B.6) é positivo (ou nulo se $\theta^{(i+1)} = \theta^{(i)}$). Pode-se verificar também que o segundo termo é nulo, já que $H(\theta^{(i+1)}, \theta^{(i)})$ não é função de $\theta^{(i+1)}$. Quanto ao terceiro termo, note que

$$-[K(\theta^{(i+1)}, \theta^{(i)}) - K(\theta^{(i)}, \theta^{(i)})] \triangleq \int_B \log \left[\frac{p(B|\theta^{(i)}, A)}{p(B|\theta^{(i+1)}, A)} \right] p(B|\theta^{(i)}, A) dB \tag{B.7}$$

ou seja, a diferença equivale à distância de Kullback-Leibler entre as densidades $p(B|\theta^{(i)}, A)$ e $p(B|\theta^{(i+1)}, A)$, sendo portanto positiva (ou nula se as duas densidades forem iguais “almost everywhere”). Assim, conclui-se que $\log[p(\theta^{(i+1)}|A)] - \log[p(\theta^{(i)}|A)] \geq 0$ concluindo a demonstração. □

B.2 Equalização Cega empregando o algoritmo EM

Nesta seção temos como objetivo desenvolver um algoritmo para a obtenção de estimativas MAP dos parâmetros do canal de transmissão h utilizando o modelo de sinal descrito em (1.1). Em outras palavras, desejamos determinar o conjunto de parâmetros h que maximiza a densidade *a posteriori* $p(h|y_{0:n}, \sigma^2)$, em que $y_{0:n} \triangleq \{y_0, \dots, y_n\}$.

Observação 1 A aplicação direta do algoritmo EM à estimação do sinal transmitido (i.e., à estimativa da moda de $p(x_{0:n}|y_{0:n})$) não é possível, uma vez que a grandeza

$$Q(x_{0:n}, x_{0:n}^{(i)}) = \int_h \log p(x_{0:n}|h, y_{0:n}) p(h|x_{0:n}^{(i)}, y_{0:n}) dh$$

não pode ser calculada de forma analítica.

Pode-se mostrar que a determinação exata da densidade $p(h|y_{0:n}, \sigma^2)$ é impraticável [46], uma vez que exige um número de operações que cresce exponencialmente em função do tamanho do bloco de dados n . No entanto, conforme mostramos nesta seção, o modelo de sinal descrito por (1.1) permite que se obtenham diretamente estimativas MAP dos parâmetros h através do algoritmo EM.

Utilizando como variável latente a seqüência de vetores $X_{0:n}$, a densidade *a posteriori* aumentada $p(h|X_{0:n}, y_{0:n}, \sigma^2)$ pode ser expressa¹ como

$$p(h|X_{0:n}, y_{0:n}) = \frac{p(h, X_{0:n}, y_{0:n})}{\int_h p(h, X_{0:n}, y_{0:n}) dh} \quad (\text{B.8})$$

$$\propto p(h, X_{0:n}, y_{0:n}) \quad (\text{B.9})$$

$$= p(y_{0:n}|h, X_{0:n})p(h)p(X_{0:n}) \quad (\text{B.10})$$

$$= p(h) \prod_{j=0}^n p(y_j|h, X_j)p(x_j) . \quad (\text{B.11})$$

Aplicando-se o algoritmo EM para conseguir estimativas de h , obtemos a seguinte expressão para o “passo E”:

$$Q(h, h^{(i)}) = \sum_{X_{0:n}} \log p(h|X_{0:n}, y_{0:n})p(X_{0:n}|h^{(i)}, y_{0:n}) \quad (\text{B.12})$$

$$= E [\log p(h|X_{0:n}, y_{0:n})|h^{(i)}, y_{0:n}] . \quad (\text{B.13})$$

Assumindo que a densidade *a priori* dos parâmetros do canal é Gaussiana complexa, ou seja,

$$\begin{aligned} p(h) &= \pi^{-N} |\Sigma_0|^{-1} \exp \left(-\|h - h_0\|_{\Sigma_0^{-1}}^2 \right) \\ &\triangleq \mathcal{N}_C(h|h_0; \Sigma_0) , \end{aligned} \quad (\text{B.14})$$

e que os símbolos transmitidos são i.i.d equiprováveis, segue que a verossimilhança das observações é dada por

$$p(y_j|h, X_j) = \mathcal{N}_C(y_j|h^H X_j; \sigma^2) . \quad (\text{B.15})$$

¹Omitiremos a partir deste ponto a dependência das densidades condicionais em relação ao parâmetro σ^2 (variância do ruído aditivo), uma vez que este é assumido conhecido.

O logaritmo da densidade *a posteriori* aumentada dada em (B.11) pode ser escrito então como

$$\log p(h|X_{0:n}, y_{0:n}) = \alpha - \|h - h_0\|_{\Sigma_0^{-1}}^2 - \frac{1}{\sigma^2} \sum_{j=0}^n \|y_j - h^H X_j\|^2. \quad (\text{B.16})$$

sendo α uma constante de normalização independente de h . Calculando-se então a esperança condicional de (B.16), obtemos

$$\begin{aligned} Q(h, h^{(i)}) &= -\frac{1}{\sigma^2} \sum_{j=0}^n (E [\|y_j - h^H X_j\|^2 | h^{(i)}, y_{0:n}]) \\ &\quad - \|h - h_0\|_{\Sigma_0^{-1}}^2 + \alpha' \end{aligned} \quad (\text{B.17})$$

$$\begin{aligned} &= -\frac{1}{\sigma^2} \sum_{j=0}^n (-2\Re \{h^H E [X_j | h^{(i)}, y_{0:n}] y_j^*\} + \\ &\quad + h^H E [X_j X_j^H | h^{(i)}, y_{0:n}] h) \\ &\quad - (-2\Re \{h^H \Sigma_0^{-1}\} + h^H \Sigma_0^{-1} h) + \alpha''. \end{aligned} \quad (\text{B.18})$$

em que α' e α'' são constantes que não dependem de h .

O segundo passo do algoritmo EM (passo “M”) consiste de determinar o valor de h que maximiza (B.18). Para isto, note inicialmente que

$$\begin{aligned} \frac{\partial}{\partial h} Q(h, h^{(i)}) &= \\ &\quad - \left(\Sigma_0^{-1} + \frac{1}{\sigma^2} \sum_{j=0}^n E [X_j X_j^H | h^{(i)}, y_{0:n}] \right) h \\ &\quad + \left(\Sigma_0^{-1} h_0 + \frac{1}{\sigma^2} \sum_{j=0}^n E [X_j | h^{(i)}, y_{0:n}] y_j^* \right). \end{aligned} \quad (\text{B.19})$$

e que a matriz Hessiana $\partial^2 Q(h, h^{(i)}) / \partial h^2$ é dada por

$$-\left(\Sigma_0^{-1} + \frac{1}{\sigma^2} \sum_{j=0}^n E [X_j X_j^H | h^{(i)}, y_{0:n}] \right) < 0. \quad (\text{B.20})$$

Igualando-se (B.19) a zero, segue que o máximo de $Q(h, h^{(i)})$ é dado por

$$h^{(i+1)} = \left(\Sigma_0^{-1} + \frac{1}{\sigma^2} \sum_{j=0}^n E [X_j X_j^H | h^{(i)}, y_{0:n}] \right)^{-1} \left(\Sigma_0^{-1} h_0 + \frac{1}{\sigma^2} \sum_{j=0}^n E [X_j | h^{(i)}, y_{0:n}] y_j^* \right). \quad (\text{B.21})$$

As esperanças em (B.21) podem ser calculadas numericamente através do algoritmo BCJR [47] (vide Apêndice), uma vez que

$$E [X_j | h^{(i)}, y_{0:n}] = \sum_{i=1}^{N_s} \mathcal{X}_i \mathbb{P}(X_j = \mathcal{X}_i | h^{(i)}, y_{0:n}) \quad (\text{B.22})$$

$$E [X_j X_j^H | h^{(i)}, y_{0:n}] = \sum_{i=1}^{N_s} \mathcal{X}_i \mathcal{X}_i^H \mathbb{P}(X_j = \mathcal{X}_i | h^{(i)}, y_{0:n}) \quad (\text{B.23})$$

sendo N_s o número de vetores X_j distintos.

Na Tabela B.2 resumimos o conteúdo desta seção, descrevendo o algoritmo proposto. Note que aproximações das estimativas² MAP/ML dos símbolos transmitidos podem ser obtidas a partir de $\mathbb{P}(X_j | h^{(i)}, y_{0:n})$, uma vez que a densidade *a posteriori* dos símbolos transmitidos é dada por

$$\mathbb{P}(x_j = a | h^{(i)}, y_{0:n}) = \sum_{\forall b | x_j = a} \mathbb{P}(X_j = b | h^{(i)}, y_{0:n}). \quad (\text{B.24})$$

(Inicialização) Obtenha $h^{(0)} \sim \mathcal{N}_C(h|h_0; \Sigma_0)$

Para $i \geq 0$, itere:

- Calcule $\mathbb{P}(X_j | h^{(i)}, y_{0:n})$, $0 \leq j \leq n$ através do algoritmo BCJR.
- Determine $h^{(i+1)}$ através de (B.21)-(B.23).

Tabela B.2: Algoritmo Para Equalização Cega baseado no Algoritmo EM

²Observe que as estimativas obtidas através deste método são subótimas, uma vez que $h^{(i)}$ é uma estimativa dos parâmetros do canal.

B.2.1 Simulações

Através de simulações numéricas avaliamos o desempenho do algoritmo segundo a taxa média de erro de bit (BER) obtida ao longo de 100 realizações independentes num sistema de comunicação que transmite blocos de 250 símbolos binários i.i.d. através do canal $h = [0,41 \ -0,82 \ 0,41]^T$ com ordem e potência de ruído conhecidas pelo receptor. A densidade *a priori* do canal foi admitida igual a $\mathcal{N}_C(h|0; I)$.

Na Fig. B.1 mostramos o desempenho obtido em função da relação sinal-ruído e do número de iterações empregado pelo algoritmo EM. Na mesma figura, para efeito de comparação, mostramos o desempenho obtido pelo algoritmo de Viterbi, construído a partir do canal conhecido e implementado considerando-se desconhecidos os bits das extremidades do bloco. Uma característica que salta aos olhos é o desempenho ruim do algoritmo da Sec. B.2 que piora a medida que a relação sinal-ruído ultrapassa 6dB.

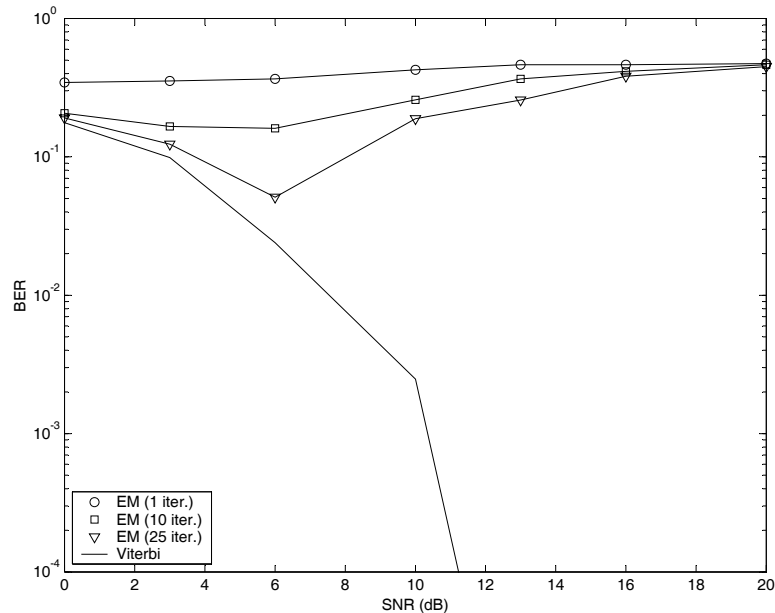


Figura B.1: Desempenho médio ao longo de 100 realizações independentes do algoritmo para equalização cega descrito na Sec. B.2 em função da relação sinal-ruído (SNR) e o número de iterações.

B.3 Algoritmo Melhorado

Analisando mais detidamente o funcionamento do algoritmo da Sec. B.2 observamos que, sob baixos níveis de ruído, as densidades $p(X_k|h^{(i)}, y_{0:n})$ estimadas (através do algoritmo BCJR) tem valores muito próximos de 0 ou 1, num comportamento que pa-

rece contribuir para o aparecimento de algo semelhante a um ciclo-limite³ que impede a convergência do algoritmo para uma solução fixa.

Para contornar este problema, fizemos uma alteração de caráter heurístico no algoritmo da Sec. B.2 inspirada em técnicas de *simulated annealing*. Estas técnicas [48], amplamente empregadas na solução de problemas de otimização não-linear, baseiam-se na observação de fenômenos naturais ligados à cristalização de metais, que sugerem que a convergência para o mínimo global num problema de minimização iterativa pode ser obtida pela introdução de perturbações lentamente decrescentes (nas suas soluções intermediárias).

Para evitar o aparecimento de oscilações cíclicas como as observadas para o algoritmo anteriormente descrito, substituímos a densidade $p(X_k|h^{(i)}, y_{0:n}, \sigma^2)$ (calculada através do algoritmo BCJR), necessária para a determinação de (B.23), pela densidade $\bar{p}_i(X_k|h^{(i)}, y_{0:n}, \sigma^2) = p(X_k|h^{(i)}, y_{0:n}, \sigma^2 + \delta(i))$, com $\delta(i) > 0$ sendo uma função decrescente de i .

Note que, dado o maior nível de ruído informado ao algoritmo BCJR, a densidade $\bar{p}_i(X_k|h^{(i)}, y_{0:n}, \sigma^2)$ exibe uma menor tendência de assumir valores extremos (0 ou 1) do que a densidade *a posteriori* verdadeira. Isto parece contribuir para a eliminação de oscilações cíclicas. Em nossas simulações, testamos diversas formas para a função $\delta(i)$, obtendo melhores resultados para funções $\delta(i)$ da forma exponencial.

B.3.1 Simulações

Nas Figs. B.2 e B.3 mostramos o desempenho obtido pela versão modificada do algoritmo da Sec. B.2, utilizando $\delta(i) = 10^{-0,1i}$ e $\delta(i) = 10^{-0,05i}$, respectivamente, empregando as mesmas hipóteses adotadas na Sec. B.2.1.

Como se pode observar, ao fazer $\delta(i) = 10^{-0,1i}$ nota-se ainda uma degradação de desempenho para SNR superiores a 12 dB (Fig. B.2). Diminuindo-se o nível de ruído adicional mais lentamente ($\delta(i) = 10^{-0,05i}$), no entanto, o desempenho do algoritmo modificado aproxima-se bastante daquele do detector ótimo MLSE (Viterbi) para 25 iterações, como se pode observar na Fig. B.3.

B.4 Comentários

Apresentamos neste capítulo algoritmos para a equalização de canais lineares FIR baseados na versão Bayesiana do algoritmo EM. O algoritmo desenvolvido gera estimativas dos parâmetros do canal de transmissão que maximizam localmente a densidade

³Note que o algoritmo EM opera de forma determinística, e que os valores de $h^{(i)}$ e $y_{0:n}$ definem completamente o comportamento do algoritmo nas iterações posteriores.

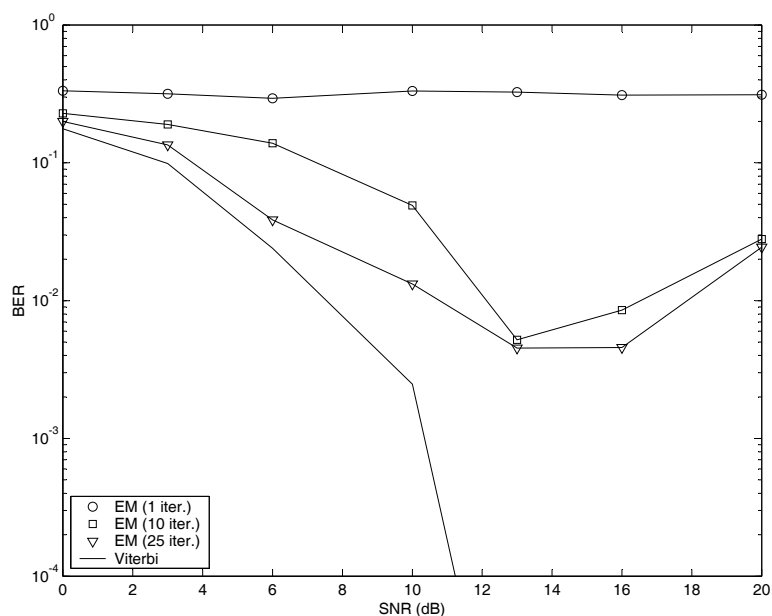


Figura B.2: Desempenho médio ao longo de 100 realizações independentes da versão modificada do algoritmo para equalização cega descrito na Sec. B.2.1 ($\delta(i) = 10^{0,1i}$) em função da relação sinal-ruído(SNR) e o número de iterações.

a posteriori destes parâmetros (estimativa local MAP), determinando estimativas dos símbolos transmitidos indiretamente (e de forma subótima) ao re-estimar os parâmetros do canal.

Conforme verificamos por meio de simulações numéricas, o algoritmo da Sec. B.2 tem a sua convergência prejudicada sob baixos níveis de ruído, o que pôde ser contornado, pelo menos sob o cenário adotado, pela utilização de técnicas de *annealing* (Sec. B.3), sem incorrer num aumento de complexidade computacional.

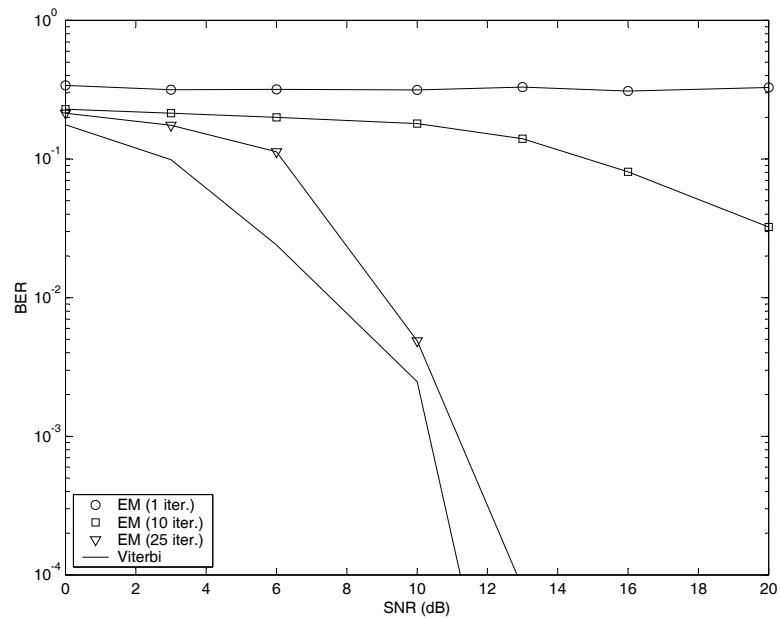


Figura B.3: Desempenho médio ao longo de 100 realizações independentes da versão modificada do algoritmo para equalização cega descrito na Sec. B.2.1 ($\delta(i) = 10^{0,05i}$) em função da relação sinal-ruído(SNR) e o número de iterações.

Apêndice C

Estimação através de Métodos MCMC

Os métodos MCMC (*Markov Chain Monte Carlo*) são ferramentas numéricas para a geração de amostras de variáveis aleatórias baseadas na simulação de cadeias de Markov [49]. A utilização destes métodos, que permaneceu limitada até os anos 90 devido a restrições na área computacional, promoveu grandes avanços na área da estatística aplicada por permitir a obtenção de soluções Bayesianas para inúmeros problemas de estimação praticamente intratáveis por métodos tradicionais.

Neste apêndice apresentamos uma breve introdução às cadeias de Markov (Sec. C.1), descrevendo em seguida os dois principais métodos MCMC: o algoritmo de Metropolis-Hastings (Sec. C.2) e o Amostrador de Gibbs (Sec. C.3). Em seguida, descrevemos dois algoritmos para equalização cega baseados no Amostrador de Gibbs desenvolvidos em [11] e [29] (Sec. C.4), avaliando seu desempenho através de simulações numéricas.

C.1 Métodos MCMC e Cadeias de Markov - Definições Básicas

Nesta seção apresentamos uma síntese de resultados sobre cadeias Markov, enfatizando aqueles ligados ao funcionamento das técnicas MCMC. Referências completas sobre o assunto podem ser encontradas em [8], [49] e [42], entre outros.

Entendemos por cadeias de Markov uma seqüência $x_{0:n}$ de variáveis aleatórias tal que

$$\mathbb{P}(x_{k+1} \in A | x_{0:k}) = \mathbb{P}(x_{k+1} \in A | x_k) . \quad (\text{C.1})$$

Como consequência de (C.1), uma cadeia de Markov é completamente definida

pelo seu núcleo de transição (*transition kernel*) [42], uma função K definida¹ em $\mathcal{X} \times \mathcal{B}(\mathcal{X})$ tal que

- (i) $\forall x \in \mathcal{X}$, $K(x, \cdot)$ é uma medida de probabilidade;
- (ii) $\forall A \in \mathcal{B}(\mathcal{X})$, $K(\cdot, A)$ é mensurável.
- (iii) $\mathbb{P}(x_{k+1} \in A | x_k) = \int_A K(x_k, dx_{k+1})$.

Numa formulação geral, cada uma das variáveis aleatórias x_k , $k \geq 0$ é distribuída distintamente, i.e.² $x_k \sim \pi_k$, valendo que

$$\pi_{k+1}(B) = \int_{\mathcal{X}_k} K(x_k, B) \pi(dx_k) . \quad (\text{C.2})$$

O funcionamento dos métodos MCMC baseia-se na simulação de uma cadeia de Markov com núcleo de transição K tal que π_k convirja para a distribuição p da qual se deseja gerar amostras na medida em que $k \rightarrow \infty$. Uma condição suficiente para que isto ocorra é que a medida invariante π associada ao núcleo K , definida como a solução de

$$\pi(B) = \int_{\mathcal{X}} K(x, B) \pi(dx), \quad \forall B \in \mathcal{B}(\mathcal{X}) \quad (\text{C.3})$$

exista e seja igual à distribuição da qual se deseja amostrar. Assim, desde que a π_k convirja para p a uma taxa suficientemente³ alta, pode-se assumir para fins práticos que $x_k \sim p(x)$, $k > k_0$, sendo as amostras x_0, \dots, x_{k_0} descartadas (período de *burn in*).

Um segundo resultado mais restritivo que garante a existência de uma medida invariante é a satisfação da condição de *balanço detalhado*, definida como

$$K(y, x) \pi(y) = K(x, y) \pi(x) \quad \forall x, y . \quad (\text{C.4})$$

Caso (C.4) for satisfeita, a cadeia de Markov em questão é reversível e possui medida invariante π .

¹Indicamos pela notação $\mathcal{B}(\mathcal{X})$ o σ -campo de Borel gerado pelos elementos de \mathcal{X} .

²Pela notação $x \sim p(x)$ entende-se que a variável aleatória x tem sua densidade dada por $p(x)$.

³Infelizmente, é difícil estabelecer condições gerais de convergência para as medidas invariantes dos núcleos de transição empregados pelos métodos MCMC. É possível demonstrar que, sob condições de regularidade expostas em ([8], Cap. 4 e 6), o algoritmo de Metropolis-Hastings e o Amostrador de Gibbs geram cadeias tais que $\|K^n(x, \cdot) - \pi\|_h \leq Mr^{-n}$, onde M e r são constantes e $\|\cdot\|_h$ denota [8]

$$\|\mu\|_h = \sup_{|g| \leq h} \left| \int g(x) \mu(dx) \right| .$$

C.2 Algoritmo de Metropolis-Hastings

O Algoritmo de Metropolis-Hastings (M-H) [50] é um método para a geração de amostras de densidades uni- ou multivariadas baseado na simulação de cadeias de Markov, aplicável caso a densidade da qual se deseja amostrar tenha forma conhecida⁴.

A operação deste algoritmo é bastante similar⁵ à das técnicas de simulação de variáveis aleatórias por aceitação/rejeição [19], obtendo novas amostras para a seqüência gerada a partir de uma *densidade proposta* $q(\cdot)$. Sendo p a densidade da qual se deseja amostrar e x_k o elemento atual da seqüência gerada, o elemento x_{k+1} é obtido através do procedimento descrito na Tab. C.1.

Dado x_k

1. Gere $y_k \sim q(y|x_k)$.
2. Faça

$$x_{k+1} = \begin{cases} y_k & \text{com probabilidade } \rho(x_k, y_k), \\ x_k & \text{com probabilidade } 1 - \rho(x_k, y_k), \end{cases}$$

em que

$$\rho(x, y) = \min \left\{ \frac{p(y) q(x|y)}{p(x) q(y|x)}, 1 \right\}.$$

Tabela C.1: Algoritmo de Metropolis-Hastings

Sob a condição de que as densidades $p(\cdot)$ e $q(\cdot)$ sejam positivas em seus suportes, pode-se demonstrar que a cadeia de Markov gerada pelo procedimento descrito na Tab. C.1 satisfaz a condição de balanço detalhado (Eq. C.4), tendo $p(\cdot)$ como medida invariante.

O algoritmo M-H encontra aplicação em diversas situações nas quais se necessita gerar amostras de densidades com forma conhecida, mas cujas características tornam esta operação difícil, podendo também ser empregado em conjunto com técnicas de estimação seqüenciais (Filtros de partículas, num esquema conhecido como Algoritmo *Resample-Move* [9]).

⁴A menos de uma constante de normalização.

⁵Uma diferença marcante em relação a estas técnicas é que a densidade proposta $q(x)$ empregada pelo algoritmo M-H pode ser explicitamente condicionada em elementos previamente aceitos para a amostra.

C.3 Amostrador de Gibbs

O Amostrador de Gibbs (AG) é uma técnica baseada na simulação de cadeias de Markov adequada à geração de amostras de distribuições de probabilidade multivariadas. Suponha que, para algum $m > 1$, a variável aleatória multidimensional θ possa ser escrita como $\theta = [\theta^{(1)}, \dots, \theta^{(m)}]^T$, sendo $\theta^{(i)}$ partições uni- ou multidimensionais do vetor θ . Suponha ainda que se possam gerar amostras das densidades “condicionais completas” (p_1, \dots, p_m) , definidas como:

$$p_i = p(\theta^{(i)} | \theta^{(1)}, \theta^{(2)}, \dots, \theta^{(i-1)}, \theta^{(i+1)}, \dots, \theta^{(m)}). \quad (\text{C.5})$$

O núcleo de transição empregado pelo amostrador de Gibbs para a geração de θ_{k+1} a partir de θ_k é definido pelo procedimento descrito na Tab. C.2.

Dado $\theta_k = \{\theta_k^{(1)}, \dots, \theta_k^{(m)}\}$, gere θ_{k+1} como:

1. $\theta_{k+1}^{(1)} \sim p_1(\theta^{(1)} | \theta_k^{(2)}, \dots, \theta_k^{(m)})$
2. $\theta_{k+1}^{(2)} \sim p_2(\theta^{(2)} | \theta_{k+1}^{(1)}, \theta_k^{(3)}, \dots, \theta_k^{(m)})$
- ⋮
- m. $\theta_{k+1}^{(m)} \sim p_m(\theta^{(m)} | \theta_{k+1}^{(1)}, \dots, \theta_{k+1}^{(m-1)})$

Tabela C.2: Amostrador de Gibbs

Partindo de θ_0 arbitrário, mostra-se que, sob condições bastante gerais de regularidade, a seqüência $\theta_0, \theta_1, \dots, \theta_n$ gerada pelo procedimento descrito na Tab. C.2 forma uma cadeia de Markov homogênea e irredutível, cuja distribuição estacionária é dada por $p(\theta)$ ([8], p. 293).

Observação 1 *Como se pode observar por sua própria definição, o Amostrador de Gibbs é útil em situações em que $p(\theta)$ é uma distribuição multivariada que não possui forma analítica conhecida ou é muito difícil de se simular. Por outro, o emprego deste algoritmo exige que se conheçam todas as distribuições condicionais completas $p_i(\theta^{(i)} | \theta^{(1)}, \dots, \theta^{(i-1)}, \theta^{(i+1)}, \dots, \theta^{(m)})$. Conforme veremos a seguir, ambas as condições são verificadas pelas formulações usuais dos problemas de equalização/identificação cega de canais de comunicação.*

Observação 2 *Os métodos MCMC podem ser empregados para a solução de problemas de estimação de parâmetros através de métodos empíricos, empregando as*

amostras obtidas após o período de burn in das cadeias simuladas⁶. Uma vez que as amostras obtidas desta forma são distribuídas de acordo com $p(\theta)$ (a densidade conjunta a posteriori dos parâmetros), pode-se então estimar as densidades marginais de qualquer um dos parâmetros contidos em θ , através do histograma de sua densidade a posteriori ou da relação:

$$\begin{aligned} p(\theta^{(i)}) &= \int p(\theta^{(i)}|\theta^{([-i])})p(d\theta^{([-i])}) \\ &\approx \frac{1}{N} \sum_{n=1}^N p(\theta^{(i)}|\theta^{([-i])})\theta_n^{([-i])}. \end{aligned} \quad (\text{C.6})$$

em que $\theta^{([-i])}$ denota o vetor θ excluído de seu i -ésimo elemento e $\theta_n^{([-i])}$ sua n -ésima amostra.

Uma vez obtidas as distribuições dos parâmetros, podem-se obter estimativas de seus valores segundo qualquer um dos critérios discutidos anteriormente. Por exemplo, estimativas ótimas no sentido MMSE podem ser obtidas tomando a média das distribuições (ou diretamente das amostras) geradas [8], e no sentido MAP tomando a moda desta distribuição.

C.4 Equalização Cega empregando o Amostrador de Gibbs

Soluções conjuntas para os problemas de equalização e de identificação cega de um canal de comunicação podem ser determinadas através de um Amostrador de Gibbs convenientemente projetado para gerar amostras da densidade $p(x_{0:n}, h, \sigma^2|y_{0:n})$. Nas seções a seguir descrevemos dois algoritmos para este fim apresentados em [11] e [29].

C.4.1 Modelo de Sinal - Hipóteses adicionais

Embora a possibilidade de se utilizar conhecimentos prévios a respeito das variáveis inferidas (descritos através das densidades *a priori*) seja uma das principais vantagens técnicas dos estimadores Bayesianos, dificuldades de caráter prático e teórico limitam a escolha destas densidades *a priori*.

⁶Note que os elementos de uma amostra obtida desta forma são dependentes, já que a seqüência $\theta_{n+1}, \dots, \theta_{n+K}$ é uma cadeia de Markov. Um esquema alternativo (e ineficiente) de amostragem que garante a independência das amostras geradas consiste em tomar como amostra da distribuição de interesse apenas o último elemento θ_i gerado, repetindo o mesmo procedimento K vezes, empregando θ_0 independentes.

Nos algoritmos descritos em [11] e [29], adotou-se a hipótese (prescindível) de que os símbolos transmitidos x_j são IID. e equiprováveis, tais que

$$p(x_j) = \frac{1}{|\mathcal{X}|}, \quad (\text{C.7})$$

em que $|\mathcal{X}|$ denota o número de valores que a variável x_j pode assumir.

De forma a garantir que as densidades condicionais completas (Sec. C.3) tenham forma analítica tratável, adotaram-se em [11] e [29] formas conjugadas⁷ [8] para as densidades *a priori* dos parâmetros do canal e da variância do ruído aditivo. Para o modelo de sinal dado em Eq. 1.1, isto equivale a assumir que os parâmetros do canal de transmissão tenham distribuição Gaussiana complexa, i.e.

$$p(h) = \mathcal{N}_C(h|h_0; \Sigma_0), \quad (\text{C.8})$$

e que a variância do ruído aditivo tenha a densidade *a priori* gama inversa, i.e.

$$p(\sigma^2) = \mathcal{IG}(\sigma^2|\alpha; \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} (\sigma^2)^{-(\alpha+1)} e^{-\frac{\beta}{\sigma^2}}. \quad (\text{C.9})$$

Observação 3 *Adotando-se valores adequados para os hiperparâmetros [8] (h_0 , Σ_0 , α e β) é possível fazer com que a escolha das densidades *a priori* tenha um impacto mínimo sobre as variáveis estimadas. De fato, pode-se verificar que ao fazer $\|\Sigma_0\| \rightarrow \infty$ em (C.8) e $(\alpha, \beta) \rightarrow 0$ em (C.9) as densidades em questão tornam-se totalmente não-informativas [8].*

C.4.2 Amostrador Seqüencial

Diversos algoritmos para a geração de amostras da densidade $p(x_{0:n}, \sigma^2, h|y_{0:n})$ baseados no Amostrador de Gibbs podem ser obtidos ao se considerar diferentes partições para o conjunto de variáveis a inferir. Ao se particionar o conjunto de variáveis de forma a considerar separadamente cada símbolo transmitido, obtém-se o seguinte algoritmo para equalização/identificação cega [29]:

1. Tome as amostras iniciais $x_{0:n}^{(0)} \sim p(x_{0:n})$, $h^{(0)} \sim p(h)$ e $\sigma^2 \sim p(\sigma^2)$.

⁷Diz-se que uma densidades *a priori* tem forma conjugada se a densidade *a posteriori* da variável inferida pertencer à mesma família (e.g., gaussiana) da densidade *a priori*.

2. Itere

$$\left\{ \begin{array}{l} x_0^{(i+1)} \sim p(x_0|x_1^{(i)}, \dots, x_n^{(i)}, h^{(i)}, (\sigma^2)^{(i)}, y_{0:n}) \\ x_1^{(i+1)} \sim p(x_1|x_0^{(i+1)}, x_2^{(i)}, \dots, x_n^{(i)}, h^{(i)}, (\sigma^2)^{(i)}, y_{0:n}) \\ \vdots \\ x_{n-1}^{(i+1)} \sim p(x_{n-1}|x_0^{(i+1)}, \dots, x_{n-2}^{(i+1)}, x_n^{(i)}, h^{(i)}, (\sigma^2)^{(i)}, y_{0:n}) \\ x_n^{(i+1)} \sim p(x_n|x_0^{(i+1)}, \dots, x_{n-1}^{(i+1)}, h^{(i)}, (\sigma^2)^{(i)}, y_{0:n}) \\ h^{(i+1)} \sim p(h|x_{0:n}^{(i+1)}, (\sigma^2)^{(i)}, y_{0:n}) \\ (\sigma^2)^{(i+1)} \sim p(\sigma^2|x_{0:n}^{(i+1)}, h^{(i+1)}, y_{0:n}) \end{array} \right. \quad (\text{C.10})$$

por um número i arbitrário de vezes.

3. Descarte um número arbitrário j de amostras iniciais (correspondentes ao *burn in* da cadeia), e estime

$$\hat{p}(x_{0:n}|y_{0:n}) = \sum_{k=j+1}^i \delta(x_{0:n} - x_{0:n}^{(i)}) \quad (\text{C.11})$$

$$\hat{p}(h|y_{0:n}) = \sum_{k=j+1}^i \delta(h - h^{(i)}) \quad (\text{C.12})$$

$$\hat{p}(\sigma^2|y_{0:n}) = \sum_{k=j+1}^i \delta(\sigma^2 - (\sigma^2)^{(i)}) \quad (\text{C.13})$$

Observação 4 Caso $p(x_i)$ seja tal que $p(x_i) = p(kx_i)$, para algum $k \in \mathbb{C}$, não é possível estimar a densidade a posteriori dos símbolos transmitidos através de (C.11) (ambigüidade de fase). Uma solução para este problema (caso os símbolos transmitidos sejam binários) é assumir que os sinal transmitido é codificado diferencialmente, ou seja, $x_n = s_n x_{n-1}$, em que $s_n \in \{\pm 1\}$ é a mensagem transmitida. Neste caso, amostras da mensagem podem ser recuperadas sem ambigüidade de fase (à exceção do primeiro elemento $s_0^{(i)}$) pela relação

$$s_n^{(i)} = x_n^{(i)} x_{n-1}^{(i)},$$

sendo a densidade a posteriori da mensagem estimada como

$$\hat{p}(s_{1:n}|y_{1:n}) = \sum_{k=j+1}^i \delta(s_{1:n} - s_{1:n}^{(i)}) . \quad (\text{C.14})$$

Determinaremos agora as densidades condicionais completas da Eq. (C.10). Definindo inicialmente a seqüência $x_{[-j]} = (x_0, \dots, x_{j-1}, x_{j+1}, \dots, x_n)$, temos que

$$p(x_n = \mathcal{X}_j | x_{[-n]}, h, \sigma^2, y_{0:n}) = \frac{p(x_n = \mathcal{X}_j, x_{[-n]}, y_{0:n}, h, \sigma^2)}{\sum_j p(x_n = \mathcal{X}_j, x_{[-n]}, y_{0:n}, h, \sigma^2)} \quad (\text{C.15})$$

$$= \frac{p(y_{0:n} | x_n = \mathcal{X}_j, x_{[-n]}, h, \sigma^2) p(x_n = \mathcal{X}_j)}{\sum_j p(y_{0:n} | x_n = \mathcal{X}_j, x_{[-n]}, h, \sigma^2) p(x_n = \mathcal{X}_j)} \quad (\text{C.16})$$

$$= \frac{p(x_n = \mathcal{X}_j) \prod_{l=n-(L-1)}^n p(y_l | X_l(j), h, \sigma^2)}{\sum_j p(x_n = \mathcal{X}_j) \prod_{l=n-(L-1)}^n p(y_l | X_l(j), h, \sigma^2)} \quad (\text{C.17})$$

em que $p(y_l | X_l(j), h, \sigma^2) = \mathcal{N}_C(y_l | h^H X_l(j), \sigma^2)$, $X_l(j) = [x_l \dots x_{l-L+1}]^T$ com $x_k = \mathcal{X}_j$. Na passagem de (C.15) para (C.16) exploramos o fato de que

$$p(x_{0:n}, y_{0:n}, h, \sigma^2) = p(y_{0:n} | x_{0:n}, h, \sigma^2) p(h) p(\sigma^2) \prod_{i=0}^n p(x_i), \quad (\text{C.18})$$

e na passagem de (C.16) para (C.17), por sua vez, exploramos o fato de que

$$p(y_{0:n} | x_{0:n}, h, \sigma^2) = \prod_{i=0}^n p(y_i | X_i, h, \sigma^2), \quad (\text{C.19})$$

em que $X_i = [x_i, \dots, x_{i-L+1}]^T \in \mathbb{C}^{L \times 1}$.

A densidade condicional completa do vetor de parâmetros do canal pode ser determinada utilizando argumentos similares, já que

$$p(h | x_{0:n}, \sigma^2, y_{0:n}) = \frac{p(y_{0:n} | x_{0:n}, h, \sigma^2) p(x_{0:n}) p(h) p(\sigma^2)}{\int_h p(y_{0:n} | x_{0:n}, h, \sigma^2) p(x_{0:n}) p(h) p(\sigma^2) dh} \quad (\text{C.20})$$

$$\propto p(y_{0:n} | x_{0:n}, h, \sigma^2) p(h) \quad (\text{C.21})$$

$$= p(h) \prod_{i=0}^n p(y_i | X_i, h, \sigma^2) \quad (\text{C.22})$$

$$= \mathcal{N}_C(h | h_0; \Sigma_0) \prod_{i=0}^n \mathcal{N}_C(y_i | h^H X_i; \sigma^2) \quad (\text{C.23})$$

$$= \mathcal{N}_C(h | h_n; \Sigma_n), \quad (\text{C.24})$$

em que

$$h_n = \Sigma_n \left(\frac{1}{\sigma^2} \sum_{i=0}^n X_i y_i^* + \Sigma_0^{-1} h_0 \right), \quad (\text{C.25})$$

$$\Sigma_n^{-1} = \frac{1}{\sigma^2} \sum_{i=0}^n X_i X_i^H + \Sigma_0^{-1}. \quad (\text{C.26})$$

Caso a variância do ruído aditivo seja assumida desconhecida, sua densidade condicional completa pode ser determinada como

$$p(\sigma^2|x_{0:n}, h, y_{0:n}) = \frac{p(y_{0:n}|x_{0:n}, h, \sigma^2)p(x_{0:n})p(h)p(\sigma^2)}{\int p(y_{0:n}|x_{0:n}, h, \sigma^2)p(x_{0:n})p(h)p(\sigma^2)d\sigma^2} \quad (\text{C.27})$$

$$\propto p(y_{0:n}|x_{0:n}, h, \sigma^2)p(\sigma^2) \quad (\text{C.28})$$

$$= p(\sigma^2) \prod_{i=0}^n p(y_i|X_i, h, \sigma^2) \quad (\text{C.29})$$

$$= \mathcal{IG}(\sigma^2|\alpha; \beta) \prod_{i=0}^n \mathcal{N}_C(y_i|h^H X_i; \sigma^2) \quad (\text{C.30})$$

$$\propto \mathcal{IG}\left(\sigma^2|\alpha + \frac{N}{2}; \beta + \sum_{i=0}^n \|y_i - h^H X_i\|^2\right). \quad (\text{C.31})$$

Observação 5 Amostras de uma variável aleatória com distribuição gama inversa podem ser obtidas facilmente notando-se que se x é uma v.a. com distribuição gama, $y = x^{-1}$ tem por definição distribuição gama inversa.

No pacote numérico MATLAB, uma amostra y com distribuição $\mathcal{IG}(y|\alpha; \beta)$ pode ser obtida através do comando $y=1/\text{gamrnd}(\alpha, 1/\beta)$.

C.4.3 Amostrador Conjunto

Ao invés de se amostrar separadamente cada símbolo transmitido x_i , pode-se projetar uma amostrador de Gibbs em que o bloco $x_{0:n}$ é amostrado conjuntamente da densidade condicional completa $p(x_{0:n}|h^{(i)}, (\sigma^2)^{(i)}, y_{0:n})$, resultando no algoritmo [11][29]:

$$\begin{cases} x_{0:n}^{(i+1)} & \sim p(x_{0:n}|h^{(i)}, (\sigma^2)^{(i)}, y_{0:n}) \\ h^{(i+1)} & \sim p(h|x_{0:n}^{(i+1)}, (\sigma^2)^{(i)}, y_{0:n}) \\ (\sigma^2)^{(i+1)} & \sim p(\sigma^2|x_{0:n}^{(i+1)}, h^{(i+1)}, y_{0:n}) \end{cases} \quad (\text{C.32})$$

Para se obter amostras da densidade $p(x_{0:n}|h^{(i)}, (\sigma^2)^{(i)}, y_{0:n})$, deve-se observar primeiro que

$$p(x_{0:n}|y_{0:n}, h, \sigma^2) = p(x_n|y_{0:n}, h, \sigma^2) \prod_{i=0}^{n-1} p(x_i|x_{i+1:n}, y_{0:n}, h, \sigma^2). \quad (\text{C.33})$$

Cada uma das densidades em (C.33) podem ser determinadas através de um procedimento conhecido como *forward-backward sampling*. O primeiro passo (*forward*)

consiste em fazer

$$p(x_{i+1}|y_{0:i+1}, h, \sigma^2) = \frac{p(y_{i+1}|x_{i+1}, y_{0:i}, h, \sigma^2)p(x_{i+1}|y_{0:i}, h, \sigma^2)}{p(y_{i+1}|y_{0:i}, h, \sigma^2)} \quad (\text{C.34})$$

$$\propto p(y_{i+1}|x_{i+1}, y_{0:i}, h, \sigma^2)p(x_{i+1}|y_{0:i}, h, \sigma^2) \quad (\text{C.35})$$

$$\propto p(y_{i+1}|x_{i+1}, h, \sigma^2)p(x_{i+1}|y_{0:i}, h, \sigma^2). \quad (\text{C.36})$$

No segundo passo (*backward*), por sua vez, é calculada a densidade

$$p(x_{i+1}|y_{0:i}, h, \sigma^2) = \sum_{x_i} p(x_{i+1}|x_i, y_{0:i}, h, \sigma^2)p(x_i|y_{0:i}, h, \sigma^2) \quad (\text{C.37})$$

$$= \sum_{x_i} p(x_{i+1}|x_i)p(x_i|y_{0:i}, h, \sigma^2). \quad (\text{C.38})$$

Utilizando os valores calculados em ambos os passos, podem-se determinar os fatores de (C.33) através da relação

$$p(x_i|x_{i+1:n}, y_{0:n}, h, \sigma^2) = p(x_i|x_{i+1}, y_{0:i}, h, \sigma^2) \quad (\text{C.39})$$

$$= \frac{p(x_{i+1}|x_i, y_{0:i}, h, \sigma^2)p(x_i|y_{0:i}, h, \sigma^2)}{p(x_{i+1}|y_{0:i}, h, \sigma^2)} \quad (\text{C.40})$$

$$= \frac{p(x_{i+1}|x_i)p(x_i|y_{0:i}, h, \sigma^2)}{p(x_{i+1}|y_{0:i}, h, \sigma^2)}. \quad (\text{C.41})$$

Finalmente, segue que fazendo $x_i \sim p(x_i|x_{i+1:n}, \sigma^2, h)$, $0 \leq i \leq n$, $x_{0:n} \sim p(x_{0:n}|h, \sigma^2, y_{0:n})$.

C.4.4 Simulações

Realizamos simulações numéricas nas quais medimos as taxas médias de erro de bit (BER) obtidas pelos algoritmos descritos nas Sec. C.4.2 e C.4.3 ao longo de 100 experimentos independentes. O sistema de comunicação analisado transmite blocos de 100 símbolos binários IID (empregando codificação diferencial), descartando os 5 últimos bits de cada extremidade para a determinação das taxas de erro. Empregamos o canal $h = [0, 41 - 0, 820, 41]^T$, sendo sua ordem e a potência do ruído assumidas conhecidas pelo receptor.

Confirmando os resultados apresentados em [11], o algoritmo baseado em amostragem seqüencial (Sec. C.4.2) teve um desempenho bastante ruim: as taxas de erro convergem para um patamar na faixa dos 20%, não decrescendo mesmo ao se iterar a cadeia de Markov resultante por mais de 1.000 iterações.

Na Fig. C.1 mostramos o desempenho obtido pelo algoritmo baseado no amostrador conjunto (Sec. C.4.3), em função da SNR e do número de iterações da cadeia de

Markov (100, 200 e 300 iterações). Para estimar o sinal transmitido, empregamos em todos os casos as 50% últimas amostras simuladas. Como se pode observar, o algoritmo da Sec. C.4.3 teve um desempenho bastante próximo ao do detector ótimo MLSE treinado (Viterbi).

Como também se pode observar, o desempenho deste algoritmo sofre uma degradação sob relações sinal-ruído elevadas na medida em que se aumenta o número de amostras empregadas, muito provavelmente devido à ocorrência de ciclos-limite (nos moldes do observado para o algoritmo EM), uma vez que o Amostrador de Gibbs comporta-se quase que deterministicamente nestas condições.

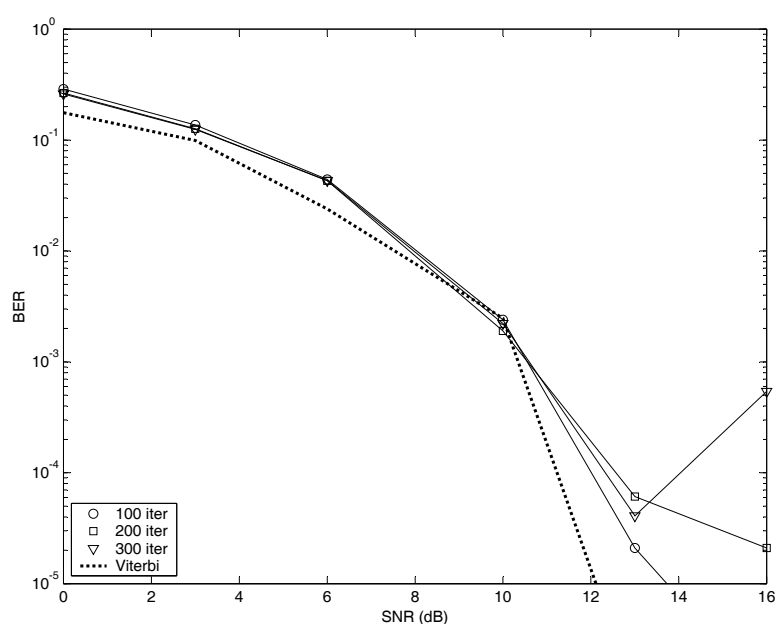


Figura C.1: Desempenho médio ao longo de 100 realizações independentes do algoritmo para equalização cega baseado no amostrador de Gibbs conjunto descrito na Sec. C.4.3 em função da relação sinal-ruído (SNR) e do número de iterações da Cadeia de Markov.

C.5 Comentários

Neste capítulo, apresentamos o algoritmo de Metropolis-Hastings e o Amostrador de Gibbs, descrevendo ainda dois métodos para equalização cega baseados no último algoritmo. Nas simulações que realizamos, verificamos que o amostrador seqüencial (Sec. C.4.2) tem um desempenho bastante ruim, em total oposição ao verificado para o amostrador conjunto (Sec. C.4.2), cujo desempenho praticamente coincidiu com o exibido pelo algoritmo de Viterbi (utilizando os parâmetros verdadeiros do canal).

Os algoritmos para equalização cega baseados em métodos MCMC são afetados

pela ambigüidade de fase inerente ao problema da equalização cega, uma vez que a estimativa das densidades *a posteriori* só pode ser feita se as variáveis mantiverem coerência de fase durante as diversas iterações do algoritmo. Nas simulações que realizamos, este problema foi contornado através do uso codificação diferencial.

Apêndice D

Densidades empregadas pelos Filtros de Partículas Rao-Blackwellizados

Neste apêndice apresentamos a dedução da expressão da densidade $p(y_n|X_{0:n}, y_{0:n-1})$ empregada pelos filtros de partículas Rao-Blackwellizados (Sec. 3.2) para equalização cega sob ruído gaussiano (real ou complexo), com parâmetros variantes no tempo, de acordo com o modelo

$$\begin{cases} X_n &= F X_{n-1} + f x_n \\ h_n &= \rho h_{n-1} + \delta u_n \\ y_n &= h_n^H X_n + v_n \end{cases} \quad (\text{D.1})$$

em que $\rho, \delta \in \mathcal{R}$, o ruído de excitação $u_n \sim \mathcal{N}(u_n|0; I)$, independente do ruído de observação $v_n \sim \mathcal{N}(v_n|0; \sigma_v^2)$, e F e f são descritos na Sec. 1.1.

Inicialmente, note que a densidade de interesse pode ser obtida como

$$\begin{aligned} p(y_n|X_{0:n}, y_{0:n-1}) &= \int dh_n p(y_n, h_n|X_{0:n}, y_{0:n-1}) \\ &= \int dh_n p(y_n|h_n, X_{0:n}, y_{0:n-1})p(h_n|X_{0:n}, y_{0:n-1}) \end{aligned} \quad (\text{D.2})$$

Como consequência do modelo de sinal, primeiramente obtemos que

$$p(y_n|h_n, X_{0:n}, y_{0:n-1}) = p(y_n|h_n, X_n).$$

Mas

$$\frac{p(h_n|X_{0:n}, y_{0:n-1})}{p(h_n|X_{0:n-1}, y_{0:n-1})} = \frac{p(X_n|, h_n, X_{0:n-1}, y_{0:n-1})}{p(X_n|X_{0:n-1}, y_{0:n-1})} = \frac{p(X_n|X_{0:n-1})}{p(X_n|X_{0:n-1})} = 1,$$

o que implica que

$$p(h_n|X_{0:n}, y_{0:n-1}) = p(h_n|X_{0:n-1}, y_{0:n-1}).$$

Assim

$$p(y_n | X_{0:n}, y_{0:n-1}) = \int dh_n \underbrace{p(y_n | h_n, X_n)}_{(\square)} \underbrace{p(h_n | X_{0:n-1}, y_{0:n-1})}_{(\triangle)} \quad (\text{D.3})$$

Como consequência do modelo de sinal

$$(\square) = \mathcal{N}(y_n | h_n^H X_n; \sigma_v^2) . \quad (\text{D.4})$$

É fácil verificar também que h_n é condicionalmente gaussiana dadas as observações e os estados do sistema (Eq. D.1), resultando que

$$(\triangle) = \mathcal{N}(h_n | \tilde{h}_{n-1}; \tilde{\Sigma}_{n-1}) . \quad (\text{D.5})$$

Conseqüentemente

$$(\square)(\triangle) = \left((\epsilon\pi)^{\frac{\epsilon+1}{\epsilon}} \sigma_v |\tilde{\Sigma}_{n-1}|^{-1/\epsilon} \right)^{-1} \exp \left(-\frac{1}{\epsilon} \underbrace{\left[\sigma_v^{-2} |y_n - h_n^H X_n|^2 - \|h_n - \tilde{h}_{n-1}\|_{\tilde{\Sigma}_{n-1}^{-1}}^2 \right]}_{(\#)} \right) , \quad (\text{D.6})$$

em que $\epsilon = 2$ caso u_n e v_n sejam v.a. reais, e $\epsilon = 1$ caso sejam complexas. Para podermos calcular analiticamente a integral de (D.3), devemos verificar se a forma quadrática (#) pode ser reescrita como

$$(\#) = \|h_n - \hat{h}_n\|_{\Sigma_n^{-1}} + \text{RES} \quad (\text{D.7})$$

em que RES deve conter somente termos não dependentes de h_n .

Desenvolvendo (#) e fazendo a identidade com a forma quadrática, obtemos inicialmente que

$$\Sigma_n^{-1} = \sigma_v^{-2} X_n X_n^H + \tilde{\Sigma}_{n-1} \quad (\text{D.8})$$

$$\Sigma_n^{-1} \hat{h}_n = \sigma_v^{-2} X_n y_n^* + \tilde{\Sigma}_{n-1} \tilde{h}_{n-1} \quad (\text{D.9})$$

Pelo lema de inversão de matrizes, obtemos ainda que

$$\Sigma_n = \tilde{\Sigma}_{n-1} - \frac{\tilde{\Sigma}_{n-1} X_n X_n^H \tilde{\Sigma}_{n-1}}{X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2} , \quad (\text{D.10})$$

e que

$$\hat{h}_n = \tilde{h}_{n-1} - \frac{\tilde{\Sigma}_{n-1} X_n \left(y_n - \tilde{h}_{n-1}^H X_n \right)^*}{X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2}. \quad (\text{D.11})$$

Assim

$$\text{RES} = \sigma_v^{-2} y_n y_n^* + \tilde{h}_{n-1}^H \tilde{\Sigma}_{n-1}^{-1} \tilde{h}_{n-1} - \hat{h}_n^H \Sigma_n^{-1} \hat{h}_n, \quad (\text{D.12})$$

que, de fato, não depende de h_n .

Desenvolvendo as expressões acima, resulta que

$$\text{RES} = \frac{|y_n - \tilde{h}_{n-1}^H X_n|^2}{X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2}, \quad (\text{D.13})$$

e finalmente que

$$\begin{aligned} (\square)(\Delta) &= \frac{1}{(\epsilon\pi)^{\frac{O+1}{\epsilon}} \sigma_v |\tilde{\Sigma}_{n-1}|^{-1/\epsilon}} \\ &\exp \left(-\frac{1}{\epsilon} \left[\frac{|y_n - \tilde{h}_{n-1}^H X_n|^2}{X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2} + \|h_n - \hat{h}_n\|_{\Sigma_n^{-1}}^2 \right] \right), \end{aligned} \quad (\text{D.14})$$

e, portanto,

$$(\square)(\Delta) = \kappa_n \mathcal{N}(h_n | \hat{h}_n; \Sigma_n) \mathcal{N}(y_n | \tilde{h}_{n-1}^H X_n; X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2), \quad (\text{D.15})$$

em que

$$\kappa_n = \sqrt{\frac{|\tilde{\Sigma}_{n-1}| \left(X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2 \right)}{|\Sigma_n| \sigma_v^2}}. \quad (\text{D.16})$$

É fácil demonstrar¹ que $\kappa_n \equiv 1$. Substituindo o resultado de (D.15) em (D.3) e notando que a segunda densidade normal de (D.15) não depende de h_n , obtemos o

¹Inicialmente, observe que

$$|\tilde{\Sigma}_{n-1}| |\Sigma_n|^{-1} = |\Sigma_n^{-1} \tilde{\Sigma}_{n-1}^{-1}|$$

e que

$$\Sigma_n^{-1} \tilde{\Sigma}_{n-1}^{-1} = I - \frac{\tilde{\Sigma}_{n-1} X_n X_n^H}{X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2}.$$

É fácil verificar que $u_1 = \tilde{\Sigma}_{n-1}^{-1} X_n$ é um autovetor de $\Sigma_n^{-1} \tilde{\Sigma}_{n-1}^{-1}$, com autovalor associado $\lambda_1 = \sigma_v^2 \left(X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2 \right)^{-1}$. Note que $\lambda_1 < 1$, o que implica que a matriz tem posto completo, e que os demais autovetores $u_l = X_n^\perp$ são associados aos autovalores $\lambda_l = 1$. Dado espectro $(\lambda_1, \dots, \lambda_O)$ de $\Sigma_n^{-1} \tilde{\Sigma}_{n-1}^{-1}$, temos que

$$|\tilde{\Sigma}_{n-1}| |\Sigma_n|^{-1} = \frac{\sigma_v^2}{X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2}$$

e portanto $\kappa_n \equiv 1$.

resultado desejado:

$$p(y_n|X_{0:n}, y_{0:n-1}) = \mathcal{N}(y_n|\tilde{h}_{n-1}^H X_n; X_n^H \tilde{\Sigma}_{n-1} X_n + \sigma_v^2). \quad (\text{D.17})$$

Resta agora obter expressões para \tilde{h}_{n-1} e $\tilde{\Sigma}_{n-1}$. Definindo

$$p(h_{n-1}|X_{0:n-1}, y_{0:n-1}) = \mathcal{N}(h_{n-1}|\bar{h}_{n-1}; \bar{\Sigma}_{n-1})$$

e explorando o fato de que

$$p(h_t|X_{0:n-1}, y_{0:n-1}) = p(h_t|h_{n-1})$$

obtemos

$$\tilde{h}_{n-1} = \rho \bar{h}_{n-1} \quad (\text{D.18})$$

$$\tilde{\Sigma}_{n-1} = \rho^2 \bar{\Sigma}_{n-1} + I\delta^2 \quad (\text{D.19})$$

Combinando todos os resultados, obtemos finalmente que:

$$p(y_n|X_{0:n}, y_{0:n-1}) = \mathcal{N}(y_n|\rho \bar{h}_{n-1}^H X_n; X_n^H (\rho^2 \bar{\Sigma}_{n-1} + I\delta^2) X_n + \sigma_v^2) \quad (\text{D.20})$$

em que (filtro de Kalman)

$$\begin{cases} \bar{h}_n &= \rho h_{n-1} + \gamma_n^{-1} \bar{\Sigma}_n X_n (y_n - \rho h_{n-1} X_n)^* \\ \bar{\Sigma}_n &= \rho^2 (\bar{\Sigma}_{n-1} - \gamma_n^{-1} \bar{\Sigma}_{n-1} X_n X_n^H \bar{\Sigma}_{n-1}) + \delta^2 I \end{cases} \quad (\text{D.21})$$

sendo $\gamma_n = \sigma_v^2 + X_n^H \bar{\Sigma}_{n-1} X_n$.

Note que se $\rho = 1$ e $\delta = 0$ (canal invariante no tempo):

$$p(y_n|X_{0:n}, y_{0:n-1}) = \mathcal{N}(y_n|\bar{h}_{n-1}^H X_n; X_n^H \bar{\Sigma}_{n-1} X_n + \sigma_v^2) \quad (\text{D.22})$$

onde

$$\begin{cases} \bar{h}_n &= \bar{h}_{n-1} + \gamma_n^{-1} \bar{\Sigma}_n X_n (y_n - \bar{h}_{n-1} X_n)^* \\ \bar{\Sigma}_n &= \bar{\Sigma}_{n-1} - \gamma_n^{-1} \bar{\Sigma}_{n-1} X_n X_n^H \bar{\Sigma}_{n-1} \end{cases} \quad (\text{D.23})$$

Apêndice E

Amostragem Seqüencial - Fundamentos

Neste apêndice descrevemos o conceito de amostragem seqüencial por importância (no qual se baseia o funcionamento dos filtros de partículas) a partir de princípios básicos da teoria de probabilidade. O conteúdo deste apêndice está organizado da seguinte forma: na Seção E.1, descrevemos como uma função densidade de probabilidade genérica $p(x)$ pode ser aproximada por uma soma ponderada de medidas discretas. Na Seção E.2, apresentamos o princípio da amostragem por importância, detalhando sua aplicação à aproximação de funções densidade de probabilidade. Na Seção E.3, mudando um pouco de linha, mostramos que a amostragem de densidades multivariadas pode ser realizada de forma seqüencial, mas que isto não se generaliza para densidades multivariadas condicionais.

E.1 Aproximação de uma densidade de probabilidade por medidas discretas

Nesta seção, mostramos como o método de Monte Carlo, normalmente empregado para a estimativa de estatísticas de uma variável aleatória, pode ser empregado na aproximação de uma função densidade de probabilidade. Para isto, considere x , uma variável aleatória com função densidade de probabilidade $p(x)$, e $f(x)$ uma função integrável desta variável. A esperança de $f(x)$, definida como

$$E[f(x)] \triangleq \int_{-\infty}^{\infty} f(x)p(x)dx . \quad (\text{E.1})$$

pode ser aproximada (método de Monte Carlo) por

$$\hat{E}[f(x)] = \frac{1}{N} \sum_{i=1}^N f(x^{(i)}), \quad (\text{E.2})$$

em que $x^{(i)}$, $1 \leq i \leq N$, é um conjunto de amostras independentes da distribuição $p(x)$. De fato, pode-se verificar que $\hat{E}[f(x)]$ é uma estimativa eficiente e não-viesada de $E[f(x)]$.

Suponha agora que $f(x) = \delta(x - X)$ (função de Dirac centrada em X). É fácil verificar que $E[\delta(x - X)] = p(X)$. Baseado neste raciocínio, pode-se propor a seguinte aproximação para a função densidade de probabilidade $p(x)$:

$$\hat{p}_N(x) = \frac{1}{N} \sum_{i=1}^N \delta(x - x^{(i)}). \quad (\text{E.3})$$

Fazendo uso intensivo de conceitos avançados de probabilidade ([9], Cap. 2), pode-se verificar que $\hat{p}_N(x)$ como definido em (E.3) converge em distribuição para a densidade correta quando $N \rightarrow \infty$.

E.2 Amostragem por importância

A aproximação de $p(x)$ dada em (E.3) depende da habilidade de se gerar amostras de $p(x)$, sendo portanto de pouca utilidade caso $p(x)$ seja desconhecido ou difícil de se amostrar. Nesta seção, mostramos que a aplicação do princípio da amostragem por importância permite que se contorne a última dificuldade, aproximando $p(x)$ a partir de amostras de uma distribuição arbitrária¹ $\pi(x)$.

Para isto, note que (E.1) pode ser reescrito² como

$$E[f(x)] = \frac{\int_{-\infty}^{\infty} f(x) \frac{p(x)}{\pi(x)} \pi(x) dx}{\int_{-\infty}^{\infty} \frac{p(x)}{\pi(x)} \pi(x) dx}. \quad (\text{E.4})$$

Definindo $w(x) \triangleq \frac{p(x)}{\pi(x)}$, temos

$$E[f(x)] = \frac{\int_{-\infty}^{\infty} f(x) w(x) \pi(x) dx}{\int_{-\infty}^{\infty} w(x) \pi(x) dx}, \quad (\text{E.5})$$

¹A única restrição que se impõem sobre $\pi(x)$ é que a mesma seja não-nula no suporte de $p(x)$, por razões óbvias.

²Note que o termo do denominador de (E.5) é igual por definição a 1; sua introdução é necessária para que a estimativa (E.6) não seja viesada.

que, pelo mesmo princípio empregado em (E.2), pode ser aproximado por

$$\hat{E}[f(x)] = \frac{\sum_{i=1}^N f(x^{(i)}) w(x^{(i)})}{\sum_{i=1}^N w(x^{(i)})}, \quad (\text{E.6})$$

em que $x^{(i)} \sim \pi(x)$, ou seja, $x^{(i)}$ são amostras independentes da distribuição $\pi(x)$.

Fazendo $f(x) = \delta_X(x)$, obtemos nos mesmos termos de (E.3) a seguinte aproximação para $p(x)$:

$$\hat{p}_N(x) = \frac{\sum_{i=1}^N \delta(x - x^{(i)}) w(x^{(i)})}{\sum_{i=1}^N w(x^{(i)})}. \quad (\text{E.7})$$

E.3 Amostragem Seqüencial

Em muitas situações, como por exemplo, na análise de séries temporais, é desejável gerar ou processar amostras numa determinada ordem. Seja, por exemplo, x_0, \dots, x_n uma série temporal com função densidade de probabilidade conjunta $p(x_0, \dots, x_n)$. A pergunta que se põe então é: como amostrar x_k , dados x_0, \dots, x_{k-1} , $0 < k \leq n$, tal que a seqüência obtida x_0, \dots, x_n tenha a distribuição desejada?

Para responder esta pergunta, imagine inicialmente que cada elemento da série temporal seja amostrado independentemente de uma densidade distinta, de acordo com:

$$\begin{aligned} x_0 &\sim p_0(x_0) \\ x_1 &\sim p_1(x_1) \\ &\vdots \\ x_n &\sim p_n(x_n) \end{aligned} \quad (\text{E.8})$$

A densidade conjunta das amostras obtidas será dada³ então por $\prod_{i=0}^n p_i(x_i)$. Fazendo $p_i(x_i) = p(x_i|x_0, \dots, x_{i-1})$, temos que $\prod_{i=0}^n p_i(x_i) = p(x_0, \dots, x_n)$, respondendo a pergunta inicial, já que:

$$p(x_0)p(x_1|x_0)p(x_2|x_0, x_1) \dots p(x_n|x_0, \dots, x_{n-1}) = p(x_0, \dots, x_n). \quad (\text{E.9})$$

Este resultado, infelizmente, não se aplica a densidades condicionais da forma $p(x_{0:n}|y_{0:n})$. A principal razão para isto é o fato de que uma densidade condicional genérica não pode ser fatorada como (E.9) (explicitamente, em densidades condicionadas somente em elementos passados da série temporal). Para verificar isto, note que

³Note que isto é verdadeiro mesmo se as densidades $p_i(x_i)$ dependerem dos valores das demais amostras, já que a amostragem é independente.

a fatoração (E.9) se generaliza no caso de uma densidade condicional para

$$p(x_{0:n}|y_{0:n}) = p(x_0|y_{0:n}) \prod_{i=1}^n p(x_i|x_{0:i-1}, y_{0:n}) . \quad (\text{E.10})$$

Apêndice F

O Algoritmo BCJR

Seja x_n uma variável aleatória discreta e y_n o sinal observado de acordo com o seguinte modelo:

$$\begin{cases} X_n = f(X_{n-1}, x_n) \\ y_n = g(X_n) + u_n \end{cases} \quad (\text{F.1})$$

em que f e g são funções possivelmente não-lineares e u_n é uma v.a. de distribuição conhecida. O algoritmo BCJR [47] permite que, dada uma seqüência de observações $y_{0:k}$, sejam calculadas recursivamente as densidades *a posteriori* $p(x_i|y_{0:k})$, $0 \leq i \leq k$.

Para demonstrar o funcionamento do algoritmo BCJR, note inicialmente que a densidade conjunta $p(X_n = a, X_{n+1} = b, y_{0:k})$ pode ser fatorada da seguinte forma:

$$p(X_n = a, X_{n+1} = b, y_{0:k}) = p(X_n = a, X_{n+1} = b, y_{0:n-1}, y_n, y_{n+1:k}) \quad (\text{F.2})$$

$$= p(y_{n+1:k}|X_n = a, X_{n+1} = b, y_{0:n-1}, y_n) \\ p(X_n = a, X_{n+1} = b, y_{0:n-1}, y_n) \quad (\text{F.3})$$

$$= p(y_{n+1:k}|X_n = a, X_{n+1} = b, y_{0:n-1}, y_n) \\ p(X_{n+1} = b, y_n|X_n = a, y_{0:n-1})p(X_n = a, y_{0:n-1}) . \quad (\text{F.4})$$

Explorando propriedades Markovianas do modelo dado em (F.1) e reordenando termos obtemos que

$$p(X_n = a, X_{n+1} = b|y_{0:k}) = p(X_n = a, y_{0:n-1})p(X_{n+1} = b, y_n|X_n = a) \\ p(y_{n+1:k}|X_{n+1} = b)/p(y_{0:k}) \quad (\text{F.5})$$

$$\triangleq \alpha_n(a) \gamma_n(a, b) \beta_{n+1}(b) / p(y_{0:k}) . \quad (\text{F.6})$$

O termo $\gamma_n(a, b)$ pode ser calculado diretamente como

$$\gamma_n(a, b) = p(X_{n+1} = b, y_n | X_n = a) \quad (\text{F.7})$$

$$= p(y_n | X_n = a, X_{n+1} = b) p(X_{n+1} = b | X_n = a) \quad (\text{F.8})$$

$$= p(y_n | X_n = a) p(X_{n+1} = b | X_n = a) . \quad (\text{F.9})$$

Assumindo que o ruído u_n tem uma distribuição gaussiana branca com média nula e variância σ^2 , o primeiro fator em (F.9) é dado por

$$p(y_n | X_n = a, X_{n+1} = b) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} |y_n - g(X_n)|^2\right) . \quad (\text{F.10})$$

O segundo fator, por sua vez, pode ser calculado observando-se que

$$p(X_{n+1} = a | X_n = b) = p(X_n = b, x_n = s^{(a,b)}) / p(X_n = b) \quad (\text{F.11})$$

$$= p(X_n = b | x_n = s^{(a,b)}) p(x_n = s^{(a,b)}) / p(X_n = b) \quad (\text{F.12})$$

$$= p(X_n = b) p(x_n = s^{(a,b)}) / p(X_n = b) \quad (\text{F.13})$$

$$= p(x_n = s^{(a,b)}) , \quad (\text{F.14})$$

em que $x_n = s^{(a,b)}$ é o valor assumido pela variável x_n que provoca a transição de $X_n = b$ para $X_{n-1} = a$. Combinando (F.10) e (F.14) obtemos

$$\gamma_n(a, b) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} |y_n - g(X_n)|^2\right) p(x_n = s^{(a,b)}) . \quad (\text{F.15})$$

Os termos α_n em (F.6), podem ser determinados através da recursão

$$\alpha_{n+1}(b) = p(X_{n+1} = b, y_{0:n}) \quad (\text{F.16})$$

$$= \sum_a p(X_{n+1} = b, X_n = a, y_{0:n-1}, y_n) \quad (\text{F.17})$$

$$= \sum_a p(X_{n+1} = b, y_n | X_n = a, y_{0:n-1}) p(X_n = a, y_{0:n-1}) \quad (\text{F.18})$$

$$= \sum_a p(X_{n+1} = b, y_n | X_n = a) p(X_n = a, y_{0:n-1}) \quad (\text{F.19})$$

$$= \sum_a \gamma_n(b, a) \alpha_n(a) , \quad (\text{F.20})$$

iniciada com $\alpha_0(a) = p(X_0 = a)$.

Observação 1 Quando aplicado ao modelo descrito em (F.1), o algoritmo de Viterbi [51] determina a seqüência de estados $X_{0:n}$ que maximiza a verossimilhança

$p(X_{0:n}|y_{0:n})$ utilizando a recursão

$$\tilde{\alpha}_{n+1}(b) = \max_a \{ \tilde{\alpha}_n(a) \gamma_n(a, b) \} .$$

Vale observar que esta recursão produz resultados muito semelhantes a (F.20) caso o ruído aditivo u_n tenha uma variância baixa.

Os termos β_n , por sua vez, são determinados através da recursão

$$\beta_n(a) = p(y_{n:k}|X_n = a) \quad (\text{F.21})$$

$$= \sum_b p(X_{n+1} = b, y_n, y_{n+1:k}|X_n = a) \quad (\text{F.22})$$

$$= \sum_b p(y_{n+1:k}|X_{n+1} = b, X_n = a, y_n) p(X_{n+1} = b, y_n|X_n = a) \quad (\text{F.23})$$

$$= \sum_b p(y_{n+1:k}|X_{n+1} = b) p(X_{n+1} = b, y_n|X_n = a) \quad (\text{F.24})$$

$$= \sum_b \beta_{n+1}(b) \gamma_n(b, a) , \quad (\text{F.25})$$

iniciada com $\beta_k(a) = p(y_k|X_k = a)$.

Para determinar o termo normalizante em (F.6) observe inicialmente que

$$\sum_a \sum_b p(X_n = a, X_{n+1} = b|y_{0:k}) = 1 , \quad (\text{F.26})$$

resultando que

$$p(y_{0:k}) = \sum_a \sum_b \alpha_n(a) \gamma_n(a, b) \beta_{n+1}(b) . \quad (\text{F.27})$$

Introduzindo a notação vetorial

$$\alpha_n = [\alpha_n(1) \dots \alpha_n(|\mathcal{X}|)]^T \in \mathbb{R}^{|\mathcal{X}| \times 1} \quad (\text{F.28})$$

$$\beta_n = [\beta_n(1) \dots \beta_n(|\mathcal{X}|)]^T \in \mathbb{R}^{|\mathcal{X}| \times 1} , \quad (\text{F.29})$$

e a matriz

$$[\mathbf{\Gamma}_n]_{ij} = \gamma_n(i, j) , \quad (\text{F.30})$$

em que $|\mathcal{X}|$ denota o número de estados assumido pela v.a. X_n , obtemos que as recursões (F.20) e (F.25) podem ser reescritas como

$$\alpha_{n+1} = \mathbf{\Gamma}_n \alpha_n \quad (\text{F.31})$$

$$\beta_n = \mathbf{\Gamma}_n^T \beta_{n+1} , \quad (\text{F.32})$$

e o termo normalizante determinado como

$$p(y_{0:k}) = \alpha_n^T \beta_n, \quad 0 \leq n \leq k. \quad (\text{F.33})$$

A partir das probabilidades de transição de estado $p(X_n = a, X_{n+1} = b | y_{0:k})$ pode-se determinar as probabilidades *a posteriori* individuais de cada símbolo transmitido observa-se a relação

$$p(x_n | y_{0:k}) = \sum_{s^{(a,b)}=x_n} p(X_n = a, X_{n+1} = b | y_{0:k}) \quad (\text{F.34})$$

$$= \frac{1}{p(y_{0:k})} \sum_{s^{(a,b)}=x_n} \alpha_n(a) \gamma_n(a, b) \beta_{n+1}(b). \quad (\text{F.35})$$

Para se calcular a distribuição *a posteriori* da variável de estado X_n (Cap. 1) basta se observar a relação

$$p(X_{n+1} = b | y_{0:k}) = \sum_a p(X_n = a, X_{n+1} = b | y_{0:k}) \quad (\text{F.36})$$

$$= \frac{1}{p(y_{0:k})} \sum_a \alpha_n(a) \gamma_n(a, b) \beta_{n+1}(b). \quad (\text{F.37})$$

PARTICLE FILTERS FOR BLIND FIR CHANNEL EQUALIZATION IN NON-GAUSSIAN NOISE

Claudio J. Bordin Jr. and Luiz A. Baccalá

Escola Politécnica, Universidade de São Paulo, Brazil.
e-mail: {bordin, baccala}@lcs.poli.usp.br

ABSTRACT

In this work, we propose new particle filter based blind equalization algorithms for FIR channels subject to additive noise of arbitrary distributions. These algorithms employ artificial evolution methods to jointly generate samples from the missing data and from the unknown channel parameters, which are assumed time-invariant. To achieve these results we introduce a new importance function, which leads to greatly improved performance compared to more obvious alternatives as verified via numerical simulations using Weibull envelope noise processes, in which the performance of the trained MLSE equalizer is approached to a narrow margin.

1. INTRODUCTION

Particle filters have extensively been applied to solving blind equalization [1] and allied communications problems [2]. Most of the literature on this subject, however, assumes that the additive noise distribution is gaussian (or at most a sum of gaussians) [3], both of whose observed signal likelihood functions admit sufficient statistics, so that the unknown channel parameters can be integrated out (Rao-Blackwellized) thereby leading to improved performance. Though gaussian noise models can be adjusted to reflect the impulsive nature of disturbances observed in many realistic communications systems, this work examines new particle filtering algorithms to directly solve the blind equalization problem of FIR channels under non-gaussian distributed additive noise.

Despite the long time recognition of the potential of particle filters to solve non-gaussian estimation problems, their application to blind equalization seems to be missing arguably due to practical and theoretical reasons like (i) the noise found in many practical communications systems may be well approximated by gaussian models and (ii) non-gaussian models lead to estimation problems in which (static) parameters are part of the state. As pointed out in [4], particle filters cannot be shown to uniformly converge to the exact solution of this class of estimation problems, and, in fact, have been observed to diverge on some occasions.

The algorithms developed in this work are based mainly on artificial parameter evolution SIS methods [5, Ch. 10], which possess smaller computational complexity (per particle) and superior performance, at least under the present models. To address the problem, we introduce a new importance function that approximates the optimal one, and, via numerical simulations, quantify the performance of the resulting algorithms under Weibull envelope noise, for differential QPSK modulation.

The remainder of this article is organized as follows: in Sec. 2, we introduce the adopted signal models and estimation objectives. In Sec. 3 we give a brief overview of parameter artificial evolution particle filtering methods applied to blind equalization and, in Sec. 4, describe the proposed methods. Finally Sec. 5 contains some simulation results, followed by conclusions in Sec. 6.

2. SIGNAL MODEL AND PROBLEM STATEMENT

Consider a communication system that differentially encodes a sequence of equiprobable i.i.d QPSK symbols $b_{0:k} \triangleq \{b_0, \dots, b_k\}$, resulting in the (also i.i.d. and equiprobable) sequence $x_{0:k}$, which is transmitted over a FIR additive noise frequency selective channel with known order. The baud-rate received signal sample y_k at instant k is then assumed to obey the relation

$$y_k = \sum_{i=0}^{L-1} x_{k-i} h_{k,i} + v_k = h_k^H X_k + v_k \quad (1)$$

where $h_k = [h_{k,0} \ h_{k,1} \ \dots \ h_{k,L-1}]^T$, $h_{k,j}$, $0 \leq j < L$ are the channel impulse response terms at instant k , $X_k = [x_k \ \dots \ x_{k-L+1}]^T$, L is the channel order, and v_k is an i.i.d noise process with known distribution $g(\cdot)$.

We further assume that the channel parameter vector is time-invariant, statistically independent from the other variables, and that it has a Gaussian prior distribution. Our aim is hence to approximate the posterior density $p(b_k | y_{0:k})$, wherefrom MAP estimates of the transmitted bits can be obtained.

When the additive noise v_k is Gaussian, particle filtering methods can directly approximate the density $p(b_{0:k} | y_{0:k})$ [1],

giving rise to the desired density via marginalization. However, if the usual Gaussian assumption is relaxed and the noise distribution no longer admits sufficient statistics [6], Rao-Blackwellized particle filters (i.e., algorithms which integrate “nuisance” parameters) can no longer be developed. As a consequence, one must resort to methods that estimate jointly parameters and state as described in Sec. 3.

3. PARTICLE FILTERS FOR JOINT PARAMETER AND STATE ESTIMATION

In this section we provide a brief introduction to the use of particle filters in estimation problems with unknown fixed parameters. Please refer to [5, Ch. 2] and [4] for further information and discussion of open theoretical problems regarding the convergence of these algorithms.

Suppose that one desires to sequentially approximate the density $p(X_{0:k}, h_{0:k} | y_{0:k})$ by means of a set of weighted samples (particles). Upon observing y_k , one must come up with samples of the variables X_k and h_k such that the joint posterior distribution of $X_{0:k}$ and $h_{0:k}$ is approximated. According to the classical Gordon method, this is achieved easily by sampling from an arbitrary importance density $\pi(h_{0:k}, X_{0:k} | y_{0:k})$ defined as the product of its marginals, i.e.,

$$\pi(h_{0:k}, X_{0:k} | y_{0:k}) = \pi(h_{0:k-1}, X_{0:k-1} | y_{0:k-1}) \pi(X_k, h_k | h_{0:k-1}, X_{0:k-1}, y_{0:k}),$$

and by attributing the weight

$$w_k^{(i)} = p(X_{0:k}^{(i)}, h_{0:k}^{(i)} | y_{0:k}) / \pi(X_{0:k}^{(i)}, h_{0:k}^{(i)} | y_{0:k}),$$

to each sample $\{h_k^{(i)}, X_k^{(i)}\}$. These weights can be sequentially evaluated as

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(X_k^{(i)}, h_k^{(i)}, y_k | h_{0:k-1}^{(i)}, X_{0:k-1}^{(i)}, y_{0:k-1})}{\pi(h_k^{(i)}, X_k^{(i)} | h_{0:k-1}^{(i)}, X_{0:k-1}^{(i)}, y_{0:k})}. \quad (2)$$

Therefore, to approximate $p(X_{0:k}, h_{0:k} | y_{0:k})$, one must (i) evaluate and sample from the importance density and (ii) evaluate the weighting term on the numerator of the r.h.s of (2).

3.1. Artificial Evolution

To compute the numerator of (2), note that the prior independence of the bits and of the channel parameters leads to

$$p(X_k, h_k, y_k | h_{0:k-1}, X_{0:k-1}, y_{0:k-1}) = p(y_k | h_k, X_k) p(X_k | X_{0:k-1}) p(h_k | h_{0:k-1}). \quad (3)$$

Under the assumption that the parameter vector is time-invariant, one gets that $p(h_k | h_{0:k-1}) = \delta(h_k - h_0)$ (where δ denotes the Dirac impulse function). This causes the weight of any particle for which $h_k^{(i)} \neq h_0^{(i)}$ to be zeroed, which prevents the adaption of the parameters no matter the importance

function chosen. Artificial parameter evolution techniques [5, Ch. 10] provide a way to circumvent this problem by regarding the parameter vector is slowly time-varying. In general, this is done via an AR(1) Gaussian model, such that

$$p(h_k | h_{0:k-1}) = \mathcal{N}(h_k | h_{k-1}; I\rho^2), \quad (4)$$

with $\rho^2 \ll \|h_0\|^2$.

3.2. Basic Blind Equalization Algorithm

For the signal model of Sec. 2, the remaining terms of (3) can be easily determined. While the likelihood term is given by

$$p(y_k | h_k, X_k) = g(y_k - h_k^H X_k), \quad (5)$$

one also obtains that

$$p(X_k | X_{0:k-1}) = p(x_k | x_{k-1}) \prod_{j=k-L+1}^k \prod_{i=1}^{L+k-j-1} \mathcal{I}\{[X_j]_i = [X_{j-1}]_{i+1}\}, \quad (6)$$

where $\mathcal{I}\{\cdot\}$ denotes the event indicator and $[A]_i$ indicates the i -th element of a vector. Notice that the term that multiplies $p(x_k | x_{k-1})$ on the r.h.s. of (6) always equals 1 if the sequence $X_{0:k}$ is consistent with its definition given in Sec. 2. Therefore

$$p(X_k | X_{0:k-1}) = p(x_k), \quad (7)$$

as the coded symbols x_k are assumed i.i.d.

Observe also that (6) implies that $p(X_{0:k}) = p(x_{0:k})$, and, similarly, exploiting the fact that $b_{0:k}$ uniquely defines $x_{0:k}$, we obtain that $p(X_{0:k}, h_{0:k} | y_{0:k}) = p(b_{0:k}, h_{0:k} | y_{0:k})$. By combining (4)-(7), one can evaluate (3) and, as a result, the weight update function (2). Adopting the prior importance function,

$$\pi(h_k^{(i)}, X_k^{(i)} | h_{0:k-1}^{(i)}, X_{0:k-1}^{(i)}, y_{0:k}) = p(x_k) \mathcal{N}(h_k | h_{k-1}^{(i)}; I\rho^2), \quad (8)$$

further simplifies (2), allowing one to estimate $b_{0:k}$ according to the algorithm of Table 1. Unfortunately, this algorithm performs rather poorly (Sec. 5) regardless of the number of particles P employed, a fact that must have discouraged further research on this topic.

A few comments about the algorithm of Table 1: note that (i) the computational complexity (per particle) of this method is much smaller than that of Rao-Blackwellized [1] algorithms, for which a Kalman filter step must be evaluated for each particle at every iteration, and is dominated by the complexity of the resampling step, (ii) its memory requirements are also very small, since one must only store the variables $h_k^{(i)}$ and $X_k^{(i)}$, a total of $2LP$ numbers and finally (iii) the phase ambiguity inherent to the blind equalization problem not only makes the use of a differential coding scheme mandatory, but also prevents the direct estimation of the channel parameters, since the samples $h_k^{(i)}$ are affected by random phase rotations.

(Initialization)

-Draw $h_0^{(i)} \sim \mathcal{N}_C(h_0|\bar{h}_0; \Sigma_0)$.

-Draw $X_0^{(i)} \sim p(X_0)$.

•For $k > 0$

• For $i = 0, \dots, P - 1$,

-Draw $h_k^{(i)} \sim \mathcal{N}(h_k|h_{k-1}^{(i)}; I\rho^2)$.

-Draw $X_k^{(i)} \sim p(X_k^{(i)}|X_{k-1}^{(i)})$.

-Obtain $b_{0:k}^{(i)}$ by decoding $x_{0:k}^{(i)}$

-Calculate and normalize the weights

$$w_k^{(i)} \propto g(y_k - h_k^{(i)H} X_k^{(i)}).$$

• End

-Resample [5] the particle set with probabilities

given by the weights $w_k^{(i)}$.

-Estimate $P(b_k)$ as

$$P(b_k = B|y_{0:k}) \approx \frac{1}{N} \sum_{i=0}^{N-1} \mathcal{I}\{b_k^{(i)} = B\}.$$

•End

Table 1. Basic Blind Equalization Algorithm (AE I)

3.3. Smoothing

A method described in [7] allows one to approximate the smoothing density $p(b_k|y_{0:k+d})$, $d > 0$, by a trivial modification of the algorithm of Table 1. Suppose that at instant k the resampling step produces a set of indexes $j_k(i)$, $0 \leq i < P$. Now, introducing the notation $b_{k,k}^{(i)} \triangleq b_k^{(i)}$ and $b_{l,k+1}^{(i)} \triangleq b_{l,k}^{j_k(i)}$, $l \leq k$, one can show that for $d \geq 0$,

$$P(b_{k-d} = B|y_{0:k}) \approx \frac{1}{N} \sum_{i=0}^{N-1} \mathcal{I}\{b_{k-d,k}^{(i)} = B\}, \quad (9)$$

i.e., smoothed estimates can be obtained by resampling $b_k^{(i)}$, $0 \leq i < P$ according to the weights calculated in successive iterations. As verified in (Sec. 5), this leads to great performance improvements at the cost of a small complexity and memory increase.

4. IMPROVED EQUALIZATION ALGORITHM

The bad performance of the algorithm of Table 1 induced us to look for alternative particle filtering structures to solve the target estimation problem. As the optimal importance function cannot be evaluated for the adopted signal model, we propose an alternative heuristic function that mimics the optimal, approximating the analytically unsolvable integral by a summation (Table 2).

The proposed importance function consists of two steps: in the first, M samples of $h_k^{(i)}$ are drawn from a reduced variance artificial prior, forming the set $h_k^{(i,r)}$. Then, the likelihood of each of the possible pair $\{h_k^{(i,r)}, X_k^{(i)}\}$ is evaluated

and normalized, and a sample is drawn from the resulting discrete joint density. The weights are then evaluated according to (2).

(Initialization)

-Draw $h_0^{(i)} \sim \mathcal{N}_C(h_0|\bar{h}_0; \Sigma_0)$.

-Draw $X_0^{(i)} \sim p(X_0)$.

•For $k > 0$

• For $i = 0, \dots, P - 1$,

-For $r = 0, \dots, M - 1$, draw

$$h_k^{(i,r)} \sim \mathcal{N}_C(h_k|h_{k-1}^{(i)}; I(\rho^2/2)).$$

-Draw X_k and h_k jointly from the **discrete density**

$$\pi(X_k, h_k) \propto g(y_k - h_k^H X_k) p(X_k|X_{k-1}^{(i)}) \mathcal{I}\{h_k = h_k^{(i,r)}\},$$

-Calculate and normalize the weights $w_k^{(i)} \propto$

$$\sum_{r,s} g(y_k - h_k^{(r)H} X_k^{(s)}) p(X_k^{(s)}|X_{k-1}^{(i)}) \mathcal{N}(h_k^{(r)}|h_{k-1}^{(i)}; I\frac{\rho^2}{2}).$$

•End

-Obtain $b_{0:k}^{(i)}$ by decoding $x_{0:k}^{(i)}$, $i = 0, \dots, P - 1$.

-Resample the particle set with probabilities

given by the weights $w_k^{(i)}$.

-Estimate $P(b_{k-L})$ as

$$P(b_{k-L} = B) \approx \frac{1}{N} \sum_{i=0}^{N-1} \mathcal{I}\{b_{k-d,k}^{(i)} = B\}.$$

•End

Table 2. Modified Algorithm (AE II)

5. SIMULATION RESULTS

To quantify the performance of the proposed blind equalization methods, we carried out numerical simulations, evaluating the mean BER (bit error rates) along 250 independent realizations, each containing a block of 400 independently transmitted QPSK symbols. BER evaluation was made after discarding the first 100 symbols to allow for algorithm convergence. The proposed algorithms were initialized as described in Tables 1 and 2, with $\bar{h}_0 = 0$ and $\bar{\Sigma}_0 = I$.

We chose a white Weibull envelope noise model (see the Appendix), with parameter $a = 1.1$ (strongly non-gaussian), and employed order $L = 3$ normalized complex random channels, drawn from the complex Gaussian prior $\mathcal{N}(h|0; I)$. To establish a comparison basis, we also evaluated the performance of the optimal MLSE equalizer (Viterbi) with exact channel knowledge and using the Weibull envelope noise density to compute path metrics.

In all simulations, we adopted $\rho^2 = 0.05$ and employed the residual resampling algorithm [5, Ch. 1]. In Fig. 1 we show the mean BER obtained by the algorithms with a smoothing lag of $d = 10$. As one can readily verify, the algorithm of Table 1 (AE I) performed very poorly (the same was verified for $d = 0$, not shown in the graphic). While the algorithm of Table 2 (AE II) also performed poorly for $M = 1$, its performance was much improved for $M = 5$, approaching that of

the trained MLSE equalizer by a margin of 2-3 dB. As one might expect, increasing the number of particles lead to improved results.

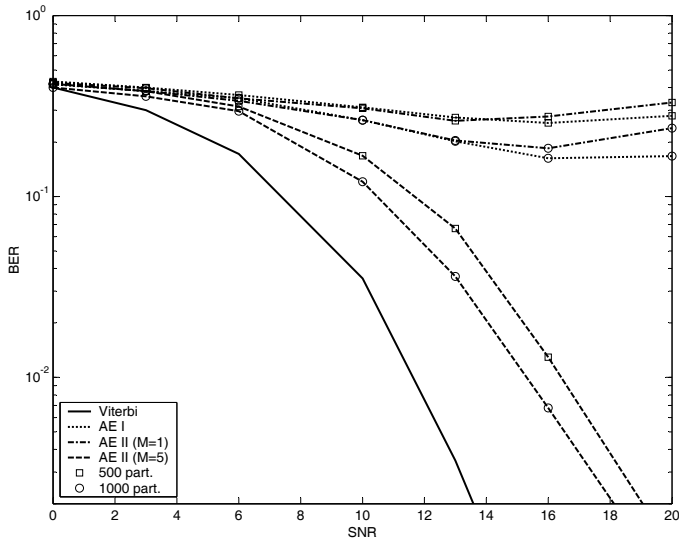


Fig. 1. Performance of blind equalization algorithms (smoothing lag $d = 10$) under Weibull envelope white noise as a function of the signal-to-noise ratio (SNR) and of the number of particles employed, compared to the performance of the optimal MLSE detector (Viterbi).

6. CONCLUSIONS

In this work we proposed and evaluated the performance of blind equalization algorithms for known-order FIR channels subject to additive non-gaussian noise. We introduced a new importance function that led to greatly improved performance under the simulation scenarios considered, in which QPSK differentially modulated symbols are transmitted under Weibull envelope noise.

Due to computational restraints, we did not evaluate the performance of the proposed methods in steady state. However, we did not notice convergence issues when the proposed algorithms are run for up to 10.000 iterations.

7. REFERENCES

- [1] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle Filtering," *IEEE Sig. Proc. Mag.*, vol. 20, no. 5, pp. 19–38, September 2003.
- [2] C. J. Bordin and L. A. Baccalá, "Particle Filter Algorithms for Joint Blind Equalization/Decoding of Convolutionally Coded Signals," in *Proc. of ICASSP 2005*, Philadelphia, PA - USA, 2005, vol. 3, pp. 497–500.
- [3] E. Puskaya, C. Andrieu, A. Doucet, and W. J. Fitzgerald, "Particle Filtering for Demodulation in Fading Channels with non-Gaussian Additive Noise," *IEEE Trans. on Comm.*, vol. 49, no. 4, pp. 579–582, April 2001.
- [4] D. Crisan and A. Doucet, "A Survey of Convergence Results on Particle Filtering Methods for Practitioners," *IEEE Trans. on Sig. Proc.*, vol. 50, no. 3, pp. 736–746, March 2002.
- [5] A. Doucet, J. de Freitas, and N. Gordon, Eds., *Sequential Monte-Carlo Methods in Practice*, Springer-Verlag, Berlin, Germany, 2000.
- [6] N. Storvik, "Particle Filters for State Space Models with the Presence of Unknown Static Parameters," *IEEE Trans. on Sig. Proc.*, vol. 50, no. 2, pp. 281–289, February 2002.
- [7] J. S. Liu and R. Chen, "Blind Deconvolution via Sequential Imputations," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 567–576, Junho 1995.
- [8] L. Gang and K. Yu, "Modelling and Simulation of Coherent Weibull Clutter," *IEE Proc. F*, vol. 136, no. 1, pp. 1–12, February 1989.
- [9] A. Farina, A. Russo, F. Scannapieco, and S. Barbarossa, "Theory of Radar Detection in Coherent Weibull Clutter," *IEE Proc. F*, vol. 134, no. 2, pp. 174–190, 1987.

Appendix: Weibull envelope noise

Consider the complex random variable $v = v_R + jv_I$ obtained [8] from two independent zero-mean real Gaussian variables x, y of variance σ^2 via the transform

$$\begin{aligned} v_R &= x(x^2 + y^2)^{1/a-1/2} \\ v_I &= y(x^2 + y^2)^{1/a-1/2} \end{aligned}$$

for $a > 1$. The joint probability function of (v_R, v_I) is given by

$$p(v_R, v_I) = \frac{a}{4\pi\sigma^2} (v_R^2 + v_I^2)^{a/2-1} \exp\left(-\frac{1}{2\sigma^2} (v_R^2 + v_I^2)^{a/2}\right), \quad (10)$$

and the envelope $|v|$ is Weibull distributed

$$p(v) = \frac{a}{2\sigma^2} |v|^{a-1} \exp\left(-\frac{1}{2\sigma^2} |v|^a\right). \quad (11)$$

The kurtosis of v (as well as its variance) is a function of the parameter a . It can be verified that for $1 < |a| \leq 2$, v is supergaussian, characteristic that is emphasized when $a \rightarrow 1$. Although Weibull envelope processes have long been used to model heavy-tailed disturbances found, for instance, in radar detection problems [9], most authors in the communications literature tend to resort to gaussian sum models for the same purpose.

PARTICLE FILTER ALGORITHMS FOR JOINT BLIND EQUALIZATION/DECODING OF CONVOLUTIONALLY CODED SIGNALS

Claudio J. Bordin Jr. and Luiz A. Baccalá

Escola Politécnica, Universidade de São Paulo, 05508-900, São Paulo - SP, Brazil.

E-mail: `bordin,baccala@lcs.poli.usp.br`

ABSTRACT

This work introduces the use of particle filters for joint blind equalization/decoding of convolutionally coded signals transmitted over frequency selective channels. As in the equalization-only case, we show how to evaluate the optimal importance function recursively via a bank of Kalman filters. Numerical simulation investigations using both stochastic and deterministic particle selection strategies show the outstanding superiority of the deterministic joint equalization/decoding method over approaches that perform blind equalization using particle filters prior to optimal decoding.

1. INTRODUCTION

Either because of slow convergence or excessive complexity, blind equalization still remains a practical challenge despite the many approaches developed in the last two decades (see [1] for a review). In this context, particle filters [2] - numerical techniques for the solution of Bayesian estimation problems - can play a major role as a compromise solution, that balances robustness and convergence speed with complexity of implementation.

Though not exactly new in an equalization-only setting [3], particle filters failed to attract much attention until recently, possibly due to the scarcity of theoretical results coupled with the need for sufficiently powerful computers that are only now becoming more widely available. The importance of particle filters is derived from their generality which allows them to handle elaborate signal models that are otherwise practically intractable (CDMA signal reception [4] for example).

As opposed to previous works [3] addressing equalization-only scenarios, we here propose the idea of using particle filters to jointly blind equalize and decode convolutionally coded signals transmitted over frequency selective channels, in an extension of Punsakaya's work [5]. A further innovative aspect is our use of noncoherently noncatastrophic codes [6] to successfully deal with the phase ambiguities inherent to the blind equalization problem.

After stating the estimation problem (Sec. 2) and over-viewing particle filters (Sec. 3), we show how to obtain the densities applicable to solving the joint blind equalization/decoding problem (Sec. 4). This is followed in Section 5 by numerical illustrations leaving our conclusions to Section 6.

2. SIGNAL MODEL AND PROBLEM STATEMENT

Consider a convolutionally coded digital communication system that transmits BPSK symbols over an additive noise frequency selective channel. Denoting the transmitted bit sequence by x_k and the code rate by $1/R$, the transmitted (binary) symbol sequence c_k is obtained as

$$c_{Rm+n} = \left(\sum_{l=0}^K x_{m-l} p_l^n \right) \bmod 2, \quad 0 \leq n < R, \quad (1)$$

where K is the convolutional code constraint length and p_l^n the code generator coefficients. The standard transmitted BPSK signal is obtained by $s_k = 2c_k - 1$.

We assume a linear and time-invariant FIR transmission channel and perfect receiver synchronization, so that baud rate samples y_k of the received signal are expressed by the base-band equivalent model

$$y_k = \sum_{l=0}^{L-1} h_l s_{k-l} + v_k, \quad (2)$$

where h_l is the channel impulse response, L its duration in symbol intervals and v_k the additive noise assumed to be zero-mean white circular gaussian of variance σ_v^2 .

In this work, we aim at obtaining MAP estimates \hat{x}_k of the transmitted bits given the observed data, i.e.,

$$\hat{x}_k = \arg \max_{x_k} p(x_k | \underline{y}_{0:k+d}), \quad (3)$$

where $d \geq 0$ and $\underline{y}_{0:k} \triangleq (y_0, \dots, y_{(k+1)R-1})$. In next sections we describe how particle filters can be employed to obtain estimates of the posterior densities $p(x_k | \underline{y}_{0:k+d})$ for the considered signal model.

Claudio J Bordin Jr. was supported by the FAPESP grant 02/11457-7.

3. PARTICLE FILTERS

Many estimation problems in discrete-time signal processing, including blind equalization, can be stated as that of tracking the state x_k of a dynamic system

$$\begin{cases} x_{k+1} &= f_k(x_0, \dots, x_k, u_k) \\ y_k &= g_k(x_0, \dots, x_k, v_k) \end{cases} \quad (4)$$

from the observation of its output $y_{0:k}$ sequence, where f_k and g_k , in full generality, may be nonlinear, time-varying functions and u_k and v_k stand respectively for system driving and observation noise.

The Bayesian solution to (4) involves finding the posterior density $p(x_k | \underline{y}_{0:k})$, which can be accomplished in closed-form only in a few restrictive cases. Particle filters provide iterative approximations to $p(x_{0:n} | \underline{y}_{0:n})$ (from which $p(x_n | \underline{y}_{0:n})$ can be obtained by marginalization) via a weighted sum of Dirac measures which approximates densities of arbitrary shape [2].

3.1. Principles

Most particle filtering algorithms rest on the importance sampling principle, whereby the probability density $p(x_{0:k} | \underline{y}_{0:k})$ can be consistently approximated [2] as the weighted sum of Dirac measures

$$\hat{p}(x_{0:k} | \underline{y}_{0:k}) = \frac{\sum_{i=1}^M w_k^{(i)} \delta_{x_{0:k}}(x_{0:k}^{(i)})}{\sum_{i=1}^M w_k^{(i)}}, \quad (5)$$

where $x_{0:k}^{(i)}$, $1 \leq i \leq M$, are independent samples (particles) from an arbitrary density $\pi(x_{0:k} | \underline{y}_{0:k}) > 0$ (importance function), with $\delta_{x_{0:k}}(x_{0:k}^{(i)})$ denoting Dirac measures in the variable $x_{0:k}$ centered in $x_{0:k}^{(i)}$, and where $w_k^{(i)} \triangleq \pi(x_{0:k}^{(i)} | \underline{y}_{0:k}) / p(x_{0:k}^{(i)} | \underline{y}_{0:k})$.

The key to the renewed interest in particle filtering comes from the seminal work of Gordon *et al.* [7] showing that if the importance function $\pi(x_{0:k} | \underline{y}_{0:k})$ can be factored as

$$\pi(x_{0:k} | \underline{y}_{0:k}) = \pi(x_k | x_{0:k-1}, \underline{y}_{0:k}) \pi(x_{0:k-1} | \underline{y}_{0:k-1}), \quad (6)$$

each element of the sequence $x_{0:k}^{(i)}$ can be sampled sequentially from the densities $\pi(x_m | x_{0:m-1}, \underline{y}_{0:m})$, $0 \leq m \leq k$. Likewise the weights $w_k^{(i)}$ can be sequentially updated, since from Bayes's law it follows that

$$p(x_{0:k} | \underline{y}_{0:k}) = p(x_{0:k-1} | \underline{y}_{0:k-1}) \frac{p(x_k, \underline{y}_k | x_{0:k-1}, \underline{y}_{0:k-1})}{p(\underline{y}_k | \underline{y}_{0:k-1})}, \quad (7)$$

which in combination with (6) produces

$$w_n^{(i)} \propto w_{k-1}^{(i)} \frac{p(x_k^{(i)}, \underline{y}_k | x_{0:k-1}, \underline{y}_{0:k-1})}{\pi(x_k^{(i)} | x_{0:k-1}, \underline{y}_{0:k})}. \quad (8)$$

The choice of the importance function, though essentially arbitrary, impacts the algorithm performance: one can show that $\pi(x_k | x_{0:k-1}, \underline{y}_{0:k}) = p(x_k | x_{0:k-1}, \underline{y}_{0:k})$ is optimal in minimizing the conditional variance of the (unnormalized) weights $w_k^{(i)}$ hence improving algorithm performance. After some iterations, due to a phenomenon known

Table 1. Stochastic Particle Filtering Algorithm

For $k = 0, \dots, n$,
For $i = 1, \dots, M$
-Draw $x_k^{(i)}$ from $\pi(x_k x_{0:k-1}, \underline{y}_{0:k})$.
-Update the weights as in (8).
-Normalize the weights.
-Estimate $p(x_{0:k} \underline{y}_{0:k})$ as in (5).

as *degeneracy*, particle filters can lead to particles $x_{0:k}^{(i)}$ that have negligible weights $w_k^{(i)}$ what severely compromise (5)'s estimation. To overcome this, the use of a *selection* scheme is mandatory when sample quality falls below a predefined threshold.

Most particle filter algorithms in the literature employ stochastic selection (namely, multinomial or residual resampling [2]). Recently, [5] proposed algorithms that employ deterministic selection, hence the name “deterministic particle filters”. Despite the lack of rigorous convergence results, their chief interest is their excellent performance in certain situations. Motivation for them comes from observing that the optimal importance function can be expressed as

$$p(x_k | x_{0:k-1}, \underline{y}_{0:k}) = \frac{p(x_k, \underline{y}_k | x_{0:k-1}, \underline{y}_{0:k-1})}{\sum_{x_k} p(x_k, \underline{y}_k | x_{0:k-1}, \underline{y}_{0:k-1})}, \quad (9)$$

i.e., to determine $p(x_k | x_{0:k-1}, \underline{y}_{0:k})$, one must evaluate the term $p(x_k, \underline{y}_k | x_{0:k-1}, \underline{y}_{0:k-1})$ for each possible value of x_k . Hence, supposing that u_k is a discrete vector variable with D different possible values, at each iteration, the particle filter discards $M(D-1)$ “candidates” $x_k^{(i)}$ ($D-1$ for each particle). Thus the idea of performing particle selection at each iteration [8] as described in Table 2 comes to the fore.

As a final remark, it is worth mentioning the particle filtering algorithms described so far can be easily extended to provide fixed-lag smoothed estimates, since for $d > 0$ [9]:

$$p(x_{0:k} | \underline{y}_{0:k+d}) \approx \frac{\sum_{i=1}^M w_{k+d}^{(i)} \delta_{x_{0:k}}(x_{0:k}^{(i)})}{\sum_{i=1}^M w_{k+d}^{(i)}}. \quad (10)$$

4. BLIND EQUALIZATION AND DECODING

All densities needed to obtain estimates of $p(x_{0:k} | \underline{y}_{0:k})$ via both the stochastic particle filter (employing an optimal im-

Table 2. Deterministic Particle filter Algorithm

For $k = 0, \dots, n$
 For $i = 1, \dots, M$
 For $j = 1, \dots, D$
 -Calculate $w^{(i,j)} = w_k^{(i)} p(x_k = x^{(j)}, y_k | x_{0:k-1}, y_{0:k-1})$.
 -Determine $(I, J)_t$, the sequence of M pairs (i, j) corresponding to the M largest $w^{(i,j)}$.
 For $t = 1, \dots, M$
 -Make $x_{0:k-1}^{(t)} = x_{0:k-1}^{(I_t)}$ and $x_k^{(t)} = x^{(J_t)}$.
 -Make $w_k^{(t)} = w^{(I, J)_t}$.
 -Normalize the weights so that $\sum_{i=1}^M w_k^{(i)} = 1$.
 -Estimate $p(x_{0:k} | y_{0:k})$ as in (5).

portance function) and the deterministic particle filter described in Sec. 3.1 can be obtained from $p(x_k, \underline{y}_k | x_{0:k-1}, \underline{y}_{0:k-1})$ using (9). Assuming that x_k is an IID sequence one can show that

$$p(x_k, \underline{y}_k | x_{0:k-1}, \underline{y}_{0:k-1}) = p(\underline{y}_k | x_{0:k}, \underline{y}_{0:k-1}) p(x_k). \quad (11)$$

Moreover, as the bit sequence x_k uniquely defines the transmitted symbol sequence s_k , (11) implies (assuming that $x_k = 0$ for $k < 0$) that

$$p(\underline{y}_k | x_{0:k}, \underline{y}_{0:k-1}) = p(\underline{y}_k | S_{0:(k+1)R-1}, \underline{y}_{0:k-1}), \quad (12)$$

where $S_k \triangleq [s_k \dots s_{k-L+1}]^T$, and $(s_0, \dots, s_{(k+1)R-1})$ is the symbol sequence corresponding to the bit sequence $x_{0:k}$. The density on the r.h.s. of (12) further decomposes as

$$p(\underline{y}_k | S_{0:(k+1)R-1}, \underline{y}_{0:k-1}) = \prod_{j=Rk}^{R(k+1)-1} p(y_j | S_{0:j}, y_{0:j-1}). \quad (13)$$

In (13) we exploited the fact that $p(y_j | S_{0:j}, y_{0:j-1}) = p(y_j | S_{0:k}, y_{0:j-1})$, $k > j$. In order to determine $p(y_j | S_{0:j}, y_{0:j-1})$ note that using definitions above, (2) can be rewritten as

$$\begin{cases} S_{j+1} &= AS_j + e_1 s_{j+1} \\ y_j &= h^H S_j + v_j \end{cases} \quad (14)$$

where A is a $(L \times L)$ shift matrix (all entries zero, except the first subdiagonal, whose entries are ones), $e_1 = [1 \ 0 \ \dots \ 0]^T$ and $h = [h_0 \ \dots \ h_{L-1}]^T$.

From (14) one can see that y_j is conditionally gaussian given S_j and h . Assuming that the parameter h has gaussian prior distribution, and exploiting the fact it is conditionally gaussian given $S_{0:j}$ and $y_{0:j-1}$, it can be integrated out [4], resulting in

$$p(y_j | S_{0:j}, y_{0:j-1}) = \mathcal{N}_C \left(y_j \mid \hat{h}_{j-1}^H S_j ; S_j^H \Sigma_{j-1} S_j + \sigma_v^2 \right), \quad (15)$$

where \hat{h}_j and Σ_j , respectively the conditional mean and variance of h are obtained by means of conventional Kalman filter iterations:

$$\hat{h}_j = \hat{h}_{j-1} + \frac{y_j - S_j^H \hat{h}_{j-1}}{S_j^H \Sigma_{j-1} S_j + \sigma_v^2} \Sigma_{j-1} S_j. \quad (16)$$

$$\Sigma_j = \Sigma_{j-1} - \frac{\Sigma_{j-1} S_j S_j^H \Sigma_{j-1}}{S_j^H \Sigma_{j-1} S_j + \sigma_v^2}. \quad (17)$$

From (11)-(17), one can see that evaluating (11) for each particle requires the evaluation of $2R$ Kalman filter steps. Thus, if M particles are employed, the proposed algorithms require the evaluation of $2MR$ Kalman filter steps per bit, twice the complexity of an equalization-only algorithm (operating at the symbol rate).

5. SIMULATIONS

We implemented blind equalization and decoding algorithms employing particle filters for the signal model in Section 2, and evaluated their performance using Monte Carlo simulations that measured the bit error rates (BER) over 200 independent realizations. In our illustrative simulations, we used the channel $h = [0.41 \ -0.82 \ 0.41]^T$ and adopted two $R = 3$ noncoherently noncatastrophic convolutional codes [6] with constraint length $K = 3$ and $K = 4$ (with coefficients given in octal notation by (7,5,2) and (17,12,4), respectively) in order to avoid phase ambiguities.

To compute the mean BER the algorithms processed 150 message bits in each realization, and employed a fixed smoothing lag of 25 samples. The particles' initial states $S_{-1}^{(i)}$ were drawn from IID equiprobable ± 1 r.v., and we assumed that $\Sigma_{-1}^{(i)} = I$ and $h_{-1}^{(i)} \sim \mathcal{N}_C(h|0; I)$. The BER obtained by the proposed algorithms¹ as a function of the SNR (signal-to-noise ratio) using $M = 100$ particles and the (17,12,4) code is shown in Fig. 1. For comparison, we also show the performance obtained i) by the optimal MLSE equalizer (Viterbi) followed by a hard-decision decoder and ii) by a particle filter based blind equalization-only algorithm [4] (employing differential encoding) followed by an optimal soft decoder (based on the BCJR algorithm [10]).

As one can readily verify, the deterministic joint algorithm outperformed all the others. The stochastic algorithm, in turn, performed on average equivalently to the concatenated scheme, "failing" (i.e., obtaining BER of about 50%) in some realizations even at high SNR.

Figure 2 depicts the bit error performances of the proposed (deterministic) joint equalization and decoding algorithm employing $M = 50$ and $M = 100$ particles and the

¹For the stochastic algorithm, we adopted that a multinomial resampling step that is carried out whenever the "effective sample size" [2] falls below 20%.

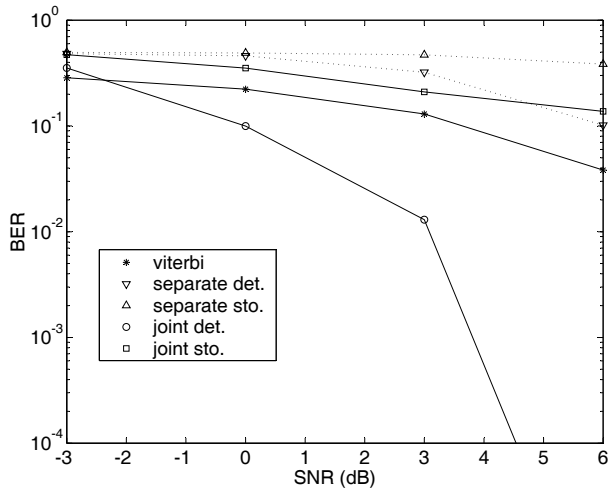


Fig. 1. Performance of the proposed joint equalization and decoding algorithms and of an alternative separate scheme using deterministic and stochastic particle filters as a function of the SNR.

(17,12,4) (square) and (7,5,2) (circle) codes. As one can verify, the performances obtained with $M = 100$ particles are very similar for both codes. With $M = 50$, however, the performance of the (17,12,4) code is worse, arguably because the number of possible states that need to be sampled is larger in this case.

6. CONCLUSIONS

After introducing the novel use of particle filtering algorithms for joint blind equalization and message decoding over frequency selective channels that employ convolutional codes, this work showed that deterministic particle filtering is the best alternative, greatly outperforming both joint stochastic particle filters and methods that equalize and decode messages separately.

7. REFERENCES

- [1] Z. Ding and Y. Li, Eds., *Blind Equalization and Identification*, Marcel Dekker, New York, 2001.
- [2] A. Doucet, J. de Freitas, and N. Gordon, Eds., *Sequential Monte-Carlo Methods in Practice*, Springer-Verlag, Berlin, Germany, 2000.
- [3] J. S. Liu and R. Chen, “Blind deconvolution via sequential imputations,” *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 567–576, june 1995.
- [4] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, “Particle

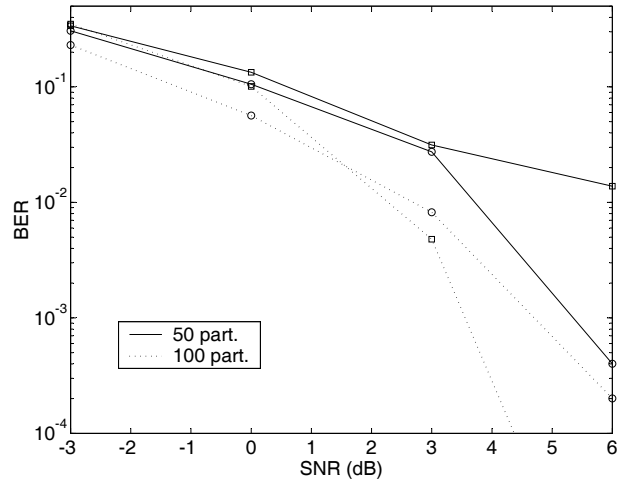


Fig. 2. Performance of the proposed (deterministic) joint equalization algorithm employing $M = 50$ and $M = 100$ particle for the codes (17,12,4) (square) and (7,5,2) (circle) as a function of the SNR.

filtering,” *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, september 2003.

- [5] E. Punskeya, *Sequential Monte Carlo Methods for Digital Communications*, Ph.D. thesis, University of Cambridge, 2003.
- [6] Y. Kofman, E. Zehavi, and S. Shamai, “nd-convolutional codes - part ii: Structural analysis,” *IEEE Transactions on Information Theory*, vol. 43, no. 2, pp. 576–589, march 1997.
- [7] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel-approach to nonlinear non-gaussian bayesian state estimation,” *IEE Proceedings-F Image & Signal Processing*, vol. 140, no. 2, pp. 107–113, april 1993.
- [8] B. Dong, X. Wang, and A. Doucet, “A new class of soft mimo demodulation algorithms,” *IEEE Transactions on Signal Processing*, vol. 51, no. 11, pp. 2752–2763, november 2003.
- [9] T. Clapp, *Statistical Methods for the Processing of Communication Data*, Ph.D. thesis, University of Cambridge, 2000.
- [10] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, march 1974.

DETERMINISTIC PARTICLE FILTERS FOR JOINT BLIND EQUALIZATION AND DECODING ON FREQUENCY SELECTIVE CHANNELS

Claudio J. Bordin Jr. and Luiz A. Baccalá

Escola Politécnica, Universidade de São Paulo, 05508-900, São Paulo, Brazil.
E-mail: {bordin,baccala}@lcs.poli.usp.br.

ABSTRACT

This work proposes deterministic particle filtering structures for joint blindly equalizing/decoding convolutionally coded signals transmitted over frequency selective channels. After describing the proposed structures, we show how to evaluate the weight update functions corresponding to the adopted signal model. Numerical simulations show that the algorithm employing deterministic particle selection greatly outperforms alternative stochastic strategies, even when the latter employ the optimal importance function.

1. INTRODUCTION

Particle filtering techniques have attracted much interest due to their ability to provide approximate solutions to otherwise intractable Bayesian filtering problems. These methods approximate target posterior densities by a weighted sum of Dirac measures centered on the so-called particles - samples of the inferred random variable - which are most frequently obtained by stochastic procedures based on the importance sampling principle.

Recently, several publications [1] [2] pointed out that the raw Monte Carlo procedure used to extend, i.e., draw new elements to the particles at each iteration can be replaced by deterministic procedures when the variable being estimated has a low dimensional discrete distribution. The resulting algorithms, henceforth called deterministic particle filters, generally exhibit greatly improved performance over traditional algorithms.

As described next, deterministic particle filters must perform particle selection (i.e., resample) at each iteration, either via stochastic or deterministic methods [2]. In this work, we provide a new interpretation of deterministic particle filtering algorithms and evaluate their performance in the solution of the joint blind equalization and decoding problem [3], considering now the use of recursive convolutional codes. For the sake of comparison, simulation results of both stochastic and deterministic resampling procedures are provided.

This work is organized as follows: we introduce particle filtering techniques in Sec. 2, followed by a description (Sec. 3) of the joint equalization and decoding problem and the deduction of the allied densities needed to solve it via particle filtering. In Sec. 4 numerical simulation results illustrate the relative performance of the proposed methods, wherefrom conclusions are summed up in Sec. 5.

2. PARTICLE FILTERS

Let y_n denote the observed output at instant n of a possibly non-linear and time-varying stochastically driven system whose state variable x_n we want to estimate. In a Bayesian filtering framework, one wishes to determine the posterior density $p(x_{0:n}|y_{0:n})$, $x_{0:n} := \{x_0, \dots, x_n\}$, which collects all statistical information about $x_{0:n}$ embodied by $y_{0:n}$. Particle filters approximate the desired posterior density by

$$\hat{p}_n(x_{0:n}|y_{0:n}) = \sum_{i=0}^{P-1} w_n^{(i)} \delta(x_{0:n} - x_{0:n}^{(i)}) \quad (1)$$

where $x_{0:n}^{(i)}$ are the so-called particles (totalling P of them), $w_n^{(i)}$ their respective weights and $\delta(\cdot)$ denotes a Dirac unit mass.

Particle filters differ from other Monte Carlo estimation methods in that the estimate $\hat{p}_n(x_{0:n}|y_{0:n})$ is determined recursively in time, and is accomplished sequentially by determining new particle elements $\{x_n^{(i)}\}_{i=0}^{P-1}$ and updating their respective weights $\{w_n^{(i)}\}_{i=0}^{P-1}$ recursively.

2.1. Stochastic Particle Filters (SPF)

In this work, stochastic particle filters refer to the general class of algorithms that perform particle extension, i.e., obtain the elements $\{x_n^{(i)}\}_{i=0}^{P-1}$ by means of random draws. Though extensively described in the literature (see [4] for a review), these algorithms are presented here in a slightly modified form so as to set the stage for an easier introduction of deterministic algorithms, so that both approaches can be appreciated from an unified standpoint.

Mr. Bordin work was funded by FAPESP under the grant 02/11457-7.

Briefly consider Table 1 which describes the SPF algorithm known as ‘‘Bootstrap Filter’’ whose operation is aimed at recursively generating samples of the joint posterior density $p(x_{0:n}|y_{0:n})$. By factorizing this density as

$$p(x_{0:n}|y_{0:n}) \propto p(x_{0:n-1}|y_{0:n-1}) p(x_n|x_{0:n-1}, y_{0:n-1}) p(y_n|x_{0:n}, y_{0:n-1}), \quad (2)$$

one can easily show that, starting from the set

$$\{x_{0:n-1}^{(i)}; w_{n-1}^{(i)}\}_{i=0}^{P-1}$$

that approximates $p(x_{0:n-1}|y_{0:n-1})$ via (1), the set

$$\{x_{0:n}^{(i)}; w_{n-1}^{(i)}\}_{i=0}^{P-1}$$

obtained after step 1 (Table 1) approximates the predictive density $p(x_{0:n}|y_{0:n-1})$, accounting for the factor $p(x_n|x_{0:n-1}, y_{0:n-1})$ in (2). Likewise step 2 accounts for the factor $p(y_n|x_{0:n}, y_{0:n-1})$ in (2), generating a weighted sample of $p(x_{0:n}|y_{0:n})$, which may be replaced by an unweighted sample if step 3 (resampling) is carried out.

1)	Draw $x_n^{(i)} \sim p(x_n x_{0:n-1}, y_{0:n-1})$.
2)	Evaluate and normalize the weights $w_n^{(i)} \propto w_{n-1}^{(i)} p(y_n x_{0:n}, y_{0:n-1})$.
3)	(Optional) Resample particles from the discrete density $(w_n^{(0)}, \dots, w_n^{(P-1)})$ and set $w_n^{(i)} = 1/P$.

Table 1. Stochastic Particle Filter employing Prior Importance Function (Bootstrap Filter).

2.2. Deterministic Particle Filters (DPF)

Stochastically drawing new particle elements as done in step 1 (Table 1) is not mandatory. In fact, this procedure introduces unwanted variabilities ([4], Chap. 2), which can be mitigated in a general framework by the use of semi-deterministic or deterministic methods, especially when the inferred variable x_n has a discrete distribution. In this special case, a set $\{x_{0:n}^{(i,j)}; \hat{w}_n^{(i,j)}\}_{i=0, j=0}^{P-1, D-1}$ with DP samples of the predictive distribution $p(x_{0:n}|y_{0:n-1})$ can be obtained deterministically as follows:

- 1a) Extend the trajectories $x_{0:n-1}^{(i)}$ deterministically, obtaining $x_{0:n}^{(i,j)} := \{x_{0:n-1}^{(i)}, x_n^{(j)}\}$, where $x_n^{(j)}$, $0 \leq j < D - 1$ are all possible values for x_n .
- 1b) Evaluate and normalize the weights $\hat{w}_n^{(i,j)} \propto w_{n-1}^{(i)} p(x_n^{(j)}|x_{0:n-1}^{(i)}, y_{0:n-1})$.

A set $\{x_{0:n}^{(i)}; w_n^{(i)}\}_{i=0}^{P-1}$ with P samples of the updated posterior density can then be obtained according to the following procedure:

- 2) Update and normalize the weights as $w_n^{(i,j)} \propto \hat{w}_n^{(i,j)} p(y_n|x_{0:n}^{(i,j)}, y_{0:n-1})$.
- 3) Apply a (deterministic or stochastic) particle selection algorithm to obtain P samples from the discrete density $\{x_{0:n}^{(i,j)}; w_n^{(i,j)}\}_{i=0, j=0}^{P-1, D-1}$.

Note that differently from stochastic algorithms, deterministic particle filters must perform particle selection (step 3) at each iteration to keep the number of particles constant. In the present comparisons, we consider classical stochastic (multinomial and residual [4]) particle selection methods (SPS) and the deterministic method (DPS) proposed in [2] which has very low computational complexity, selecting a new particle set by simply discarding the $(D - 1)P$ least weighted particles of the original set and re-normalizing the weights of the remaining ones. As opposed to stochastic methods, the deterministic methods produce bias because, on average, each particle is not selected a number of times proportional to its own weight, a fact, however, that does not affect algorithm convergence properties ([4], Cap. 2).

3. JOINT BLIND EQUALIZATION AND DECODING

To state the joint blind equalization/decoding problem and to deduce the densities needed to apply particle filters to its solution, consider initially a $(1/R)$ rate convolutionally coded digital communication system that transmits BPSK symbols over a frequency selective channel subject to additive gaussian noise. Denoting the transmitted bit sequence by b_k , the coded sequence $c_k^{(m)}$, $0 \leq m < R$, is obtained as

$$c_k^{(m)} = \left(\sum_{i=0}^K b_{k-i} d_i^{(m)} + \sum_{i=1}^K c_{k-i}^{(m)} r_i^{(m)} \right) \text{ mod } 2, \quad (3)$$

where K is the convolutional code constraint length and $d_i^{(m)}$ and $r_i^{(m)}$ are the code generating coefficients associated with the direct and the recursive polynomials respectively. The transmitted BPSK signal is then obtained as $s_{Rn+m} = 2c_n^{(m)} - 1$.

We assume a linear and time-invariant FIR transmission channel under perfect receiver synchronization, so that baud rate samples y_k of the received signal are expressed by the base-band equivalent model

$$y_k = \sum_{l=0}^{L-1} h_l s_{k-l} + v_k, \quad (4)$$

where h_l is the channel impulse response, L its duration in symbol intervals and v_k additive zero-mean white circular gaussian noise of variance σ_v^2 . Our objective is to obtain

MAP estimates \hat{b}_k of the transmitted bits given the observed data, i.e.,

$$\hat{b}_k = \arg \max_{b_k} p(b_k | \underline{y}_{0:k}), \quad (5)$$

where $\underline{y}_{0:k} := (y_0, \dots, y_{(k+1)R-1})$.

To obtain estimates of the posterior density via both stochastic and deterministic particle filters, we need to evaluate the densities $p(b_k | b_{0:k-1}, \underline{y}_{0:k-1})$ and $p(\underline{y}_k | b_{0:k}, \underline{y}_{0:k-1})$. Under the assumption that the transmitted bits are equiprobable and IID, the first density reduces to $p(b_k)$. Moreover, as the bit sequence b_k uniquely defines the transmitted symbol sequence s_k , the second density can be determined as

$$p(\underline{y}_k | b_{0:k}, \underline{y}_{0:k-1}) = p(\underline{y}_k | S_{0:(k+1)R-1}, \underline{y}_{0:k-1}), \quad (6)$$

where $S_k \triangleq [s_k \dots s_{k-L+1}]^T$, and $(s_0, \dots, s_{(k+1)R-1})$ is the symbol sequence corresponding to the bit sequence $b_{0:k}$.

Exploiting the fact that $p(y_j | S_{0:j}, y_{0:j-1}) = p(y_j | S_{0:k}, y_{0:j-1})$, $k > j$, the density on the r.h.s. of (6) further decomposes as

$$p(\underline{y}_k | S_{0:(k+1)R-1}, \underline{y}_{0:k-1}) = \prod_{j=Rk}^{R(k+1)-1} p(y_j | S_{0:j}, y_{0:j-1}). \quad (7)$$

To determine $p(y_j | S_{0:j}, y_{0:j-1})$ note that the above definitions allow rewriting (4) as

$$\begin{cases} S_{j+1} &= F S_j + e_1 s_{j+1} \\ y_j &= h^H S_j + v_j \end{cases} \quad (8)$$

where F is an $(L \times L)$ shift matrix (all entries zero, except for the first subdiagonal, whose entries are ones), $e_1 = [1 \ 0 \ \dots \ 0]^T$ and $h = [h_0 \ \dots \ h_{L-1}]^T$.

From (8) one can see that y_j is conditionally gaussian given S_j and h . Under the assumption that the parameters h have a joint gaussian prior distribution, one obtains [5]

$$p(y_j | S_{0:j}, y_{0:j-1}) = \mathcal{N}_C \left(y_j \mid \hat{h}_{j-1}^H S_j; S_j^H \Sigma_{j-1} S_j + \sigma_v^2 \right), \quad (9)$$

where \hat{h}_j and Σ_j , respectively the conditional mean and variance of h are obtained by means of conventional Kalman filter iterations:

$$\hat{h}_j = \hat{h}_{j-1} + \frac{y_j - S_j^H \hat{h}_{j-1}}{S_j^H \Sigma_{j-1} S_j + \sigma_v^2} \Sigma_{j-1} S_j. \quad (10)$$

$$\Sigma_j = \Sigma_{j-1} - \frac{\Sigma_{j-1} S_j S_j^H \Sigma_{j-1}}{S_j^H \Sigma_{j-1} S_j + \sigma_v^2}. \quad (11)$$

As a final observation, it is worth mentioning that the algorithm described so far, in addition to providing filtered estimates, can be easily extended to determine fixed-lag smoothed estimates of the transmitted bits, since for $d > 0$ [6]:

$$p(b_{0:k} | \underline{y}_{0:k+d}) \approx \frac{\sum_{i=1}^M w_{k+d}^{(i)} \delta(b_{0:k}^{(i)} - b_{0:k})}{\sum_{i=1}^M w_{k+d}^{(i)}}. \quad (12)$$

4. SIMULATION RESULTS

To compare these methods we carried out Monte Carlo simulations in which we measured mean bit error rates (BER) over 500 independent realizations produced by a communication system that transmits BPSK (± 1) symbols over a frequency selective channel with 3 random coefficients drawn independently in each realization from the multivariate complex gaussian distribution $\mathcal{N}_c(h|0; I)$. We assumed that the additive noise is a zero-mean complex circular gaussian process with variance σ_v^2 , so that the signal-to-noise ratio (SNR) is defined as

$$SNR := \|h\|^2 / 2\sigma_v^2.$$

The transmitted BPSK symbols are obtained as the non-interleaved output of the binary $1/2$ -rate recursive systematic convolutional code given in polynomial notation by

$$\left[1, \frac{1+D+D^2}{1+D^2} \right].$$

To compute the mean BER, the algorithms processed 200 message bits in each realization, discarding the first 100 to allow for algorithm convergence. The initial particle states $S_{-1}^{(i)}$ were drawn from IID equiprobable ± 1 r.v., and we assumed that $\Sigma_{-1}^{(i)} = I$ and $h_{-1}^{(i)} \sim \mathcal{N}_C(h|0; I)$. For comparison, we also show the performance obtained by an alternative scheme, consisting of a MAP decoder fed with the symbol posterior probabilities obtained by a MAP equalizer employing the correct channel parameters.

The mean BER obtained by the proposed methods as a function of the SNR is shown in Fig. 1. In this simulation, all particle filter based algorithms employed 250 particles and a zero smoothing-lag ($d = 0$). As one can verify, the methods based on DPF outperformed their stochastic counterparts, and the algorithm employing deterministic particle selection strategy (DPF-DPS) outperformed all other blind algorithms by a large margin. Notice, however, that the performance of the proposed methods ceases to improve for SNR's larger than 10 dB. This effect was observed to persist even if a much larger number of particles is employed.

In Fig. 2 we show the performance obtained under the same conditions, adopting now a fixed smoothing lag of $d = 10$. As one may notice, the performance of the SPF and of the DPF-SPS algorithms remained largely unchanged. The performance of deterministic filter employing deterministic selection (DPF-DPS), on the other hand, was much improved, coming close to that of the trained scheme.

Finally, in Fig. 3 we evaluate the performance of the deterministic particle filter employing deterministic selection (DPF-DPS) as a function of the number of particles P . As one may notice, the performance of the proposed method is greatly improved when the number of particles P is raised

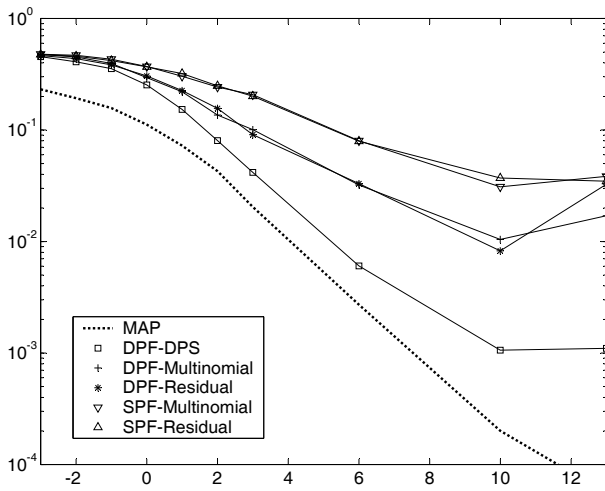


Fig. 1. Performance of the deterministic (DPF) and stochastic (SPF) joint equalization and decoding methods as a function of the SNR, employing multinomial, residual and deterministic (DPS) resampling strategies ($d = 0$), and of the alternative MAP method described in the text. The SPF algorithms employ the optimal importance function and are implemented as described in [3].

to 100, remaining practically unchanged as P is further increased.

As a final remark, it is worth mentioning that the computational complexity of the proposed DPF-DPS scheme is comparable to that of turbo joint blind equalization and decoding techniques [7]. While the proposed DPF-DPS technique requires $\mathcal{O}(PL^2)$ operations per bit, turbo techniques performing MMSE channel re-estimation require $\mathcal{O}(I(L^2 + 2^K))$, where I is the number of iterations performed (usually $I \approx 10$) and K is the code constraint length.

5. CONCLUSION

In this work we evaluated the performance of deterministic particle filtering structures for joint blindly equalizing/decoding convolutionally coded signals transmitted over frequency selective channels. Numerical simulations show that the algorithms based on deterministic particle filters outperform the stochastic alternatives, and that the performance advantage of the deterministic particle filtering method is widened if a deterministic particle selection strategy is employed.

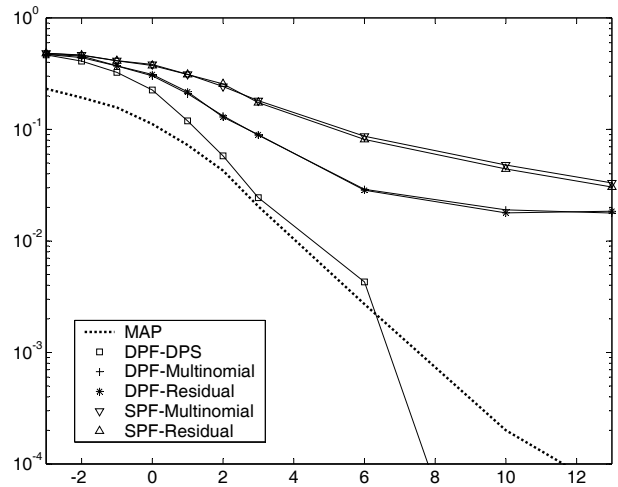


Fig. 2. Performance of the deterministic (DPF) and stochastic (SPF) joint equalization and decoding methods as a function of the SNR, employing multinomial, residual and deterministic (DPS) resampling strategies ($d = 10$), and of the alternative MAP method described in the text. The SPF algorithms employ the optimal importance function and are implemented as described in [3].

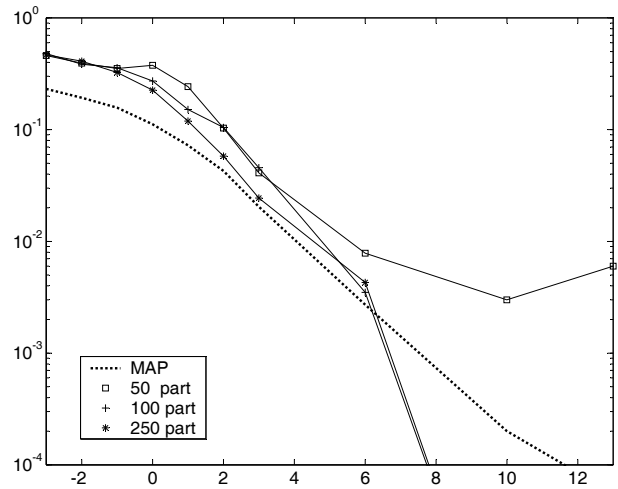


Fig. 3. Performance of the deterministic joint equalization and decoding algorithm (DPF-DPS) and of the alternative MAP method described in the text as a function of the SNR and of the number of particles ($d = 10$).

6. REFERENCES

- [1] E. Punsakaya, C. Andrieu, A. Doucet, and W. Fitzgerald, "Particle filtering for multiuser detection in fading cdma channels," in *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing, 2001.*, Singapore, august 2001.
- [2] B. Dong, X. Wang, and A. Doucet, "A new class of soft mimo demodulation algorithms," *IEEE Transactions on Signal Processing*, vol. 51, no. 11, pp. 2752–2763, november 2003.
- [3] C. J. Bordin and L. A. Baccalá, "Particle filter algorithms for joint blind equalization/decoding of convolutionally coded signals," in *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2005*, Philadelphia, PA - USA, 2005, vol. 3, pp. 497–500.
- [4] A. Doucet, J. de Freitas, and N. Gordon, Eds., *Sequential Monte-Carlo Methods in Practice*, Springer-Verlag, Berlin, Germany, 2000.
- [5] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, september 2003.
- [6] T. Clapp, *Statistical Methods for the Processing of Communication Data*, Ph.D. thesis, University of Cambridge, 2000.
- [7] R. Koetter, A. C. Singer, and M. Tuchler, "Turbo Equalization," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 67–80, January 2004.

Joint Blind Equalization and Decoding using Particle Filters

Claudio J. Bordin Jr. and Luiz A. Baccalá

Escola Politécnica, Universidade de São Paulo, 05508-900, São Paulo - SP, Brazil.
e-mail: {bordin,baccala}@lcs.poli.usp.br.

Abstract—This work examines how particle filtering algorithms - numerical techniques for solving Bayesian estimation problems - can be used for joint blind equalization and decoding of block-coded signals transmitted over frequency selective channels. Via simulations, we verify that the performance of the proposed method is much better than that of an equalization-only algorithm, outperforming the optimal linear equalizer at moderate noise levels at almost no additional computational cost.

Keywords—particle filters, blind equalization, joint equalization and decoding.

I. INTRODUCTION

Blind equalization, i.e. estimating the symbol sequence transmitted in a digital communication system exclusively from the observation of distorted received signals, has challenged researchers despite the many approaches developed since Sato's pioneering work (1975) (see [1] for a review) either because of limited robustness or slow convergence that hampers their practical application.

Even when compensation for channel distortion via an equalizer is unneeded, the effect of additive noise calls for bit error control methods through adequate coding in all practical digital communication systems. This paper shows that message decoding and equalization can be done jointly by monitoring signal probability densities and it is in this context that particle filters [2], which are numerical techniques for the solution of Bayesian estimation problems, can play a role. Though not new in equalization [3], particle filters did not attract much attention until recently, possibly due to the scarcity of theoretical results coupled with the need for sufficiently powerful computers that are only now becoming more widely available. As a matter of fact, [4] shows that, despite their large computational complexity, particle filters are almost optimal while using fairly reduced amounts of data. Another important characteristic of particle filters, because of their generality, is that they allow handling elaborate signal models, as for instance that in synchronous CDMA reception [5] which makes them a rather powerful tool.

After a brief overview of particle filtering (Sec. II), we show how to apply it for blind equalization over a frequency selective channel and in conjunction to decoding in systems that perform error control with help of block codes (Section III). In Section IV we illustrate the particle filtering algorithms via numerical simulations and leave conclusions to Section V.

Mr. Bordin work was funded by FAPESP under the grant 02/11457-7.

II. PARTICLE FILTERS

Many estimation problems in discrete-time signal processing, including blind equalization (see Sec III), can be stated as that of tracking the “state” x_n of the dynamic system

$$\begin{cases} x_{n+1} &= f_n(x_0, \dots, x_n, u_n) \\ y_n &= h_n(x_n, v_n) \end{cases} \quad (1)$$

from the observation of its output (y_0, \dots, y_n) sequence, where f_n and h_n , in full generality, may be nonlinear, time-varying functions and u_n and v_n stand respectively for the system driving and observation noise.

The Bayesian solution to (1) involves finding the posterior density $p(x_n|y_0, \dots, y_n)$, which sums up all available statistical information about x_n [6] given the observations, so that optimal estimates for it may be obtained whatever optimality criterion is chosen.

Closed form solutions to $p(x_n|y_0, \dots, y_n)$ are only known under two rather restrictive cases: i) when both f_n and h_n are linear, and u_n and v_n are gaussian (Kalman filter) and ii) when $f_n(x_0, \dots, x_n, u_n) = f_n(x_n, u_n)$ and x_n have a finite number of state values (HMM filter), leading to the need for approximate solution methods in all other cases. There are many forms of approximate solution; the best known are the linearized versions of the Kalman filter, which approximate all densities by gaussians [6] and do not yield Bayesian solutions in more generality. Particle filters, on the other hand, provide iterative solutions to $p(x_n|y_0, \dots, y_n)$ via a weighted sum of Dirac measures which approximate densities of arbitrary shape [2].

A. Principles

Most particle filter algorithms rest on the importance sampling principle, whereby the probability density $p(x_{0:n}|y_{0:n})$, $x_{0:n} \triangleq (x_0, \dots, x_n)$, can be consistently approximated [2] as the weighted sum of Dirac measures

$$\hat{p}(x_{0:n}|y_{0:n}) = \frac{\sum_{i=1}^M w_n^{(i)} \delta_{x_{0:n}}(x_{0:n}^{(i)})}{\sum_{i=1}^M w_n^{(i)}}, \quad (2)$$

where $x_{0:n}^{(i)}$, $1 \leq i \leq M$, are independent samples (particles) from an arbitrary density $\pi(x_{0:n}|y_{0:n}) > 0$ (importance function), with $\delta_{x_{0:n}}(x_{0:n}^{(i)})$ denoting Dirac measures in the variable $x_{0:n}$ centered in $x_{0:n}^{(i)}$, and where $w_n^{(i)} \triangleq \pi(x_{0:n}^{(i)}|y_{0:n})/p(x_{0:n}^{(i)}|y_{0:n})$.

The key to the renewed interest in particle filtering comes from the seminal work of Gordon *et al.* [7] showing that if the importance function $\pi(x_{0:n}|y_{0:n})$ can be factored as

$$\pi(x_{0:n}^{(i)}|y_{0:n}) = \pi(x_n^{(i)}|x_{0:n-1}^{(i)}, y_{0:n})\pi(x_{0:n-1}^{(i)}|y_{0:n-1}), \quad (3)$$

each element of the sequence $x_{0:n}^{(i)}$ can be sampled sequentially from the densities $\pi(x_m|x_{0:m-1}, y_{0:m})$, $0 \leq m \leq n$. Likewise the weights $w_n^{(i)}$ can be sequentially updated: from Bayes's law, we have

$$p(x_{0:n}|y_{0:n}) = p(x_{0:n-1}|y_{0:n-1}) \frac{p(x_n, y_n|x_{0:n-1}, y_{0:n-1})}{p(y_n|y_{0:n-1})}, \quad (4)$$

which in combination with (3) produces

$$w_n^{(i)} \propto w_{n-1}^{(i)} \frac{p(x_n, y_n|x_{0:n-1}, y_{0:n-1})}{\pi(x_n|x_{0:n-1}^{(i)}, y_{0:n})}. \quad (5)$$

A schematic summary of these procedures can be found in [2], Ch. 1. The choice of the importance function, though essentially arbitrary, impacts algorithm performance: one can show that $\pi(x_n|x_{0:n-1}, y_{0:n}) = p(x_n|x_{0:n-1}, y_{0:n})$ is optimal in minimizing the conditional variance of the (unnormalized) weights $w_n^{(i)}$ whence follows improved algorithm performance.

After some iterations, the described particle filtering algorithm can produce particles $x_{0:m}^{(i)}$ that have negligible weights $w_m^{(i)}$ due to a phenomenon known as *degeneracy*, severely compromising (2)'s estimation. To overcome this effect a procedure known as *resampling* is usually applied when sample quality falls below a predefined threshold as inferred by the effective sample size:

$$I_m = M \left(1 + \sum_{i=1}^M \left(M w_m^{(i)} - 1 \right)^2 \right)^{-1}, \quad (6)$$

whose maximum is attained for $w_m^{(i)} = 1/M$, $1 \leq i \leq M$ (ideal situation). For a given current approximation $\hat{p}(x_{0:m}|y_{0:m})$ (2), resampling consists of finding a new approximation $\tilde{p}(x_{0:m}|y_{0:m})$ (close to $\hat{p}(x_{0:m}|y_{0:m})$ in some sense) whose associated weights $\tilde{w}_m^{(i)}$ are more evenly spread. Here, in our illustrations, we adopt the multinomial resampling [2] scheme, whereby $\tilde{x}_{0:m}^{(i)} \triangleq x_{0:m}^{k(i)}$, and the indexes $k(i)$ are obtained by sampling independently M times from a multinomial distribution with class probabilities $(w_m^{(1)}, \dots, w_m^{(M)})$.

In the particle filtering procedures described so far, both particle generation and resampling are probabilistic procedures. Recently, [8] introduced algorithms that perform these operations deterministically, leading to the name ‘‘deterministic particle filters’’. The interest in these algorithms lies chiefly in their excellent performance under certain situations, even though rigorous results demonstrating their convergence remain amiss.

Motivation for them comes from observing that the optimal importance function can be expressed as

$$p(x_n|x_{0:n-1}, y_{0:n}) = \frac{p(x_n, y_n|x_{0:n-1}, y_{0:n-1})}{\sum_{x_n} p(x_n, y_n|x_{0:n-1}, y_{0:n-1})}, \quad (7)$$

i.e., to determine $p(x_n|x_{0:n-1}, y_{0:n})$, one must evaluate the term $p(x_n, y_n|x_{0:n-1}, y_{0:n-1})$ for each possible x_n value. Hence, supposing that u_n is a discrete vector variable with D different possible values, at each iteration, the particle filter discards $M(D-1)$ ‘‘candidates’’ $x_n^{(i)}$ ($D-1$ for each particle). This brings to the fore the idea of performing resampling at each iteration [8] by deterministically selecting those M candidates with highest weights, as we summarize in Table I of the Appendix.

III. BAYESIAN BLIND EQUALIZATION AND DECODING

A. Signal model and estimation objectives

Let s_k denote the binary sequence transmitted in a linear communication system. We will assume that $y(t)$, the received base-band signal is related to s_k by

$$y(t) = \sum_{n=-\infty}^{+\infty} s_k h(t - kT) + v(t), \quad (8)$$

where $h(t)$ is the channel impulse response (assumed causal and finitely supported) and $v(t)$ is a zero-mean gaussian additive noise process.

By taking discrete time samples of $y(t)$ at the symbol rate we get

$$y_n \triangleq y(t - Tn) = \sum_{i=0}^{L-1} h_i s_{n-i} + v_n, \quad (9)$$

where $h_i \triangleq h(Ti)$ and L is the channel length (in symbol intervals), and v_n is assumed to be white gaussian and of (known) variance σ_v^2 .

A blind equalizer's ultimate goal is to obtain estimates $\hat{s}_{0:n}$ of the transmitted symbols that minimize the expected error rate, given the received signal sequence $y_{0:n}$. Classical detection theory [9] asserts that expected error rate minimization is achieved by the Bayesian (maximum *a posteriori*) estimates

$$\hat{s}_m = \arg \max_{s_m} p(s_m|y_{0:n}), \quad 0 \leq m \leq n. \quad (10)$$

which ensures that the event $s_k \neq \hat{s}_k$ has minimum probability.

B. Optimal Decoding

Taking into account that practical digital communication systems include some form of error control, usually through coding, consider a (N, K) binary block coded system ($N > K$) [10], in which each K -bit long transmitted message block $\mathbf{b}_m \triangleq b_{(m-1)K+1:mK}$ is uniquely mapped onto a N -element block $\mathbf{s}_m \triangleq s_{(m-1)N+1:mN}$ of the transmitted bits via $\mathbf{s}_m = \phi(\mathbf{b}_m)$, where ϕ is a deterministic function defined by the coding scheme.

To estimate the message, the most common (suboptimal) methods start from raw estimates \hat{s}_m of the transmitted sequence (obtained ignoring the coding scheme) and attempt to recover the \mathbf{b}_m sequence via traditional decoding using hard decisions [10].

Optimal message decoding is nonetheless achievable by noting that the 2^{N-K} sequences \mathbf{s}_m not belonging to ϕ 's image should ever occur, i.e., both their prior and posterior probabilities are zero. Thus,

$$p(\mathbf{b}_m|y_{0:n}) = p(\mathbf{s}_m = \phi(\mathbf{b}_m)|y_{0:n}). \quad (11)$$

so that individual message bit probabilities can then be determined by marginalizing $p(\mathbf{b}_m|y_{0:n})$.

C. Particle filters for blind equalization and decoding

Particle filters are useful because both blind equalization and message decoding involve monitoring the message b_l posteriors $p(b_l|y_{0:n})$, $0 \leq l \leq K \lfloor n/N \rfloor$, based on the received signal sequence $y_{0:n}$.

Therefore, from Section III-B, joint optimal blind equalization and decoding can be performed in two steps: 1) optimal equalization (with prior densities for s_k properly set according the coding function ϕ), followed by 2) the inversion of the ϕ mapping. To evaluate $p(s_{0:n}|y_{0:n})$, we recast (9) in state-space as

$$\begin{cases} x_{n+1} &= F x_n + g s_{n+1} \\ y_n &= h^T x_n + v_n \end{cases}, \quad (12)$$

with $F \in \mathbb{R}^{L \times L}$ being a shift matrix (all entries equal zero, except for the first subdiagonal, where all entries equal one), $g = [1 \ 0 \ \dots \ 0]^T \in \mathbb{R}^{L \times 1}$, and $h = [h_0 \ \dots \ h_{L-1}]^T \in \mathbb{R}^{L \times 1}$. Note that $x_{0:n}$ uniquely defines $s_{0:n}$, implying that $p(x_{0:n}|y_{0:n}) = p(s_{0:n}|y_{0:n})$.

Applying the algorithms of Sec. II to obtain $p(x_{0:n}|y_{0:n})$ requires the density¹ $p(x_n, y_n|x_{0:n-1}, y_{0:n-1})$ which, from the conditional independences implicit in (12) can be expressed as:

$$p(x_n, y_n|x_{0:n-1}, y_{0:n-1}) = p(y_n|x_{0:n}, y_{0:n-1})p(x_n|x_{0:n-1}), \quad (13)$$

since $p(x_n|x_{0:n-1}) = p(x_n|x_{0:n-1}, y_{0:n-1})$.

Assuming h has a gaussian prior, i.e., $h \sim \mathcal{N}_h(\hat{h}_0, \Sigma_0)$, the first term $p(y_n|y_{0:n-1}, x_{0:n})$ in (13) can be determined as [3]:

$$p(y_n|y_{0:n-1}, x_{0:n}) = \mathcal{N}_{y_n} \left(\hat{h}_n^T x_n, x_n^T \Sigma_n x_n + \sigma_v^2 \right) \quad (14)$$

where \hat{h}_n^T and Σ_n are respectively the mean and covariance of $p(h|x_{0:n-1}, y_{0:n-1})$, which can be computed via the Kalman filter [3] recursions:

$$\hat{h}_n = \hat{h}_{n-1} + \frac{y_n - x_n^T \hat{h}_{n-1}}{x_n^T \Sigma_{n-1} x_n + \sigma_v^2} \Sigma_{n-1} x_n \quad (15)$$

$$\Sigma_n = \Sigma_{n-1} - \frac{\Sigma_{n-1} x_n x_n^T \Sigma_{n-1}}{x_n^T \Sigma_{n-1} x_n + \sigma_v^2} \quad (16)$$

¹All other relevant densities can be derived from this density using (7).

Evaluation of second factor in (13) is feasible noting that $p(x_n|x_{0:n-1})$ equals $p(s_n|s_{0:n-1})$ which can be determined using the coding function ϕ and the prior distribution of the transmitted message. Thus, let i) the transmitted message bits b_l be equiprobable i.i.d. with values ± 1 and ii) the (N, K) block code be systematic [10], i.e., each N -bit long transmitted block is constructed as

$$(s_{(m-1)N+1}, \dots, s_{mN}) = (b_{(m-1)K+1}, \dots, b_{mK}, p_{mK+1}, \dots, p_{mN}) \quad (17)$$

where $b_{(m-1)K+1:mK}$ is the transmitted message and $p_{mK+1:mN}$ the ‘‘parity’’ block, which is a function of the preceding elements. Under these assumptions, $p(s_n = a|s_{0:n-1})$ is given by

$$\begin{cases} 1/2, & \text{if } (m-1)K+1 \leq n \leq mK & \text{and } a = \pm 1 \\ 1, & \text{if } mK+1 \leq n \leq mN & \text{and } a = p_n \\ 0, & \text{if } mK+1 \leq n \leq mN & \text{and } a \neq p_n \end{cases}. \quad (18)$$

Hence, from (14) and (18) one can find $p(x_n, y_n|x_{0:n-1}, y_{0:n-1})$, and estimate $p(s_{0:n}|y_{0:n})$ using the algorithms described herein. To estimate $p(b_{0:l}|y_{0:n})$, it suffices to note that each N -block of particle $s_{0:n}^{(i)}$ is constrained, by design, to belong in the image of ϕ . Thus, inverting ϕ on each N -block of $s_{0:n}^{(i)}$, leads to desired sequence $b_{0:l}^{(i)}$, which constitutes a sample from $p(b_{0:l}|y_{0:n})$.

Finally, one must mention that estimating $p(b_{0:l}|y_{0:n})$ is only possible if some measure is taken to deal with the phase ambiguity² that is inherent to the blind equalization problem. We avoid phase ambiguity by using a ‘‘rotationally variant’’ coding function ϕ , in which $\phi(\mathbf{b}^{(i)}) \neq -\phi(\mathbf{b}^{(j)})$, $\forall i, j$, as is the case of the code we employ in our illustrative simulations.

IV. SIMULATIONS

We implemented blind equalization and decoding algorithms employing the particle filters described in Sec. II for the signal model described in Sec. III-C, and evaluated their performance using Monte Carlo simulations that measured the bit error rates (BER) over 200 independent realizations.

In our illustrative simulations, we used the channel $h = [0.5 \ -1.0 \ 0.5]^T$ and adopted a $(7, 3)$ binary systematic code, formed³ by taking the 8 codewords of the $(7, 4)$ Hamming code for which the first message bit is ‘‘1’’ to ensure ϕ 's lack of rotational invariance.

In order to compute the mean BER the algorithms processed in each realization 100 message bits, discarding results for the first 25 bits to allow for algorithm convergence. The BER obtained by the proposed algorithms as a function of

²Phase ambiguities arise in blind equalization problems when the prior densities are symmetrical $p(s_{0:n}) = p(-s_{0:n})$ and $p(h) = p(-h)$ as $p(y_n|x_n, h) = p(y_n|-x_n, -h)$, so that that each particle $x_{0:n}^{(i)}$ becomes affected by a random rotation, preventing $p(x_{0:n}|y_{0:n})$ estimation via (2).

³In [11] it is argued that such code exhibits good properties for noncoherent detection purposes. Note that the $(7, 4)$ Hamming code codewords need to be permuted to satisfy (17).

the SNR (signal-to-noise ratio) using $M = 100$ and $M = 250$ particles⁴ is shown in Fig. 1. For comparison, we also show the performance of the ideal linear MMSE equalizer of order 7 (and delay of 4 taps) followed by a hard decision decoder [10]. As one can readily verify, the deterministic algorithm outperforms the MMSE equalizer at SNR higher than 7 dB. The stochastic algorithm, on the other hand, performs poorly on average: in spite of performing well in most cases, the algorithm “fails” miserably (obtains BER of about 50%) in some realizations, even at high SNR.

Figure 2 depicts the raw bit error performances of the proposed (deterministic) joint equalization-decoding algorithm and of a (deterministic) particle filter blind equalization only algorithm (similar to [3]), constructed by assuming that $p(s_n = \pm 1 | s_{0:n-1}) = 1/2, \forall n$. The new algorithm has markedly superior performance at almost no additional computational cost.

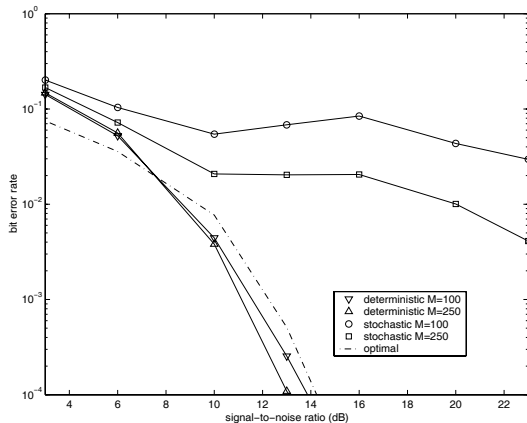


Fig. 1. Performance of the proposed algorithms as a function of the SNR.

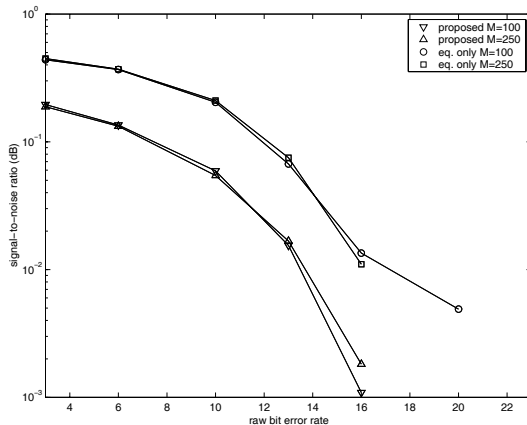


Fig. 2. Raw bit error performances of the proposed (deterministic) algorithm and of a (deterministic) blind equalization only algorithm as a function of the SNR.

⁴For the stochastic algorithm, we adopted that a resampling step is carried out whenever $I_m < 0.8M$.

V. CONCLUSIONS

This work proposed and examined the use of particle filtering algorithms for joint blind equalization and message decoding over frequency selective channels when systematic block codes are used for message error control. Numerical simulations showed that the algorithm based on deterministic particle filtering is the best alternative, outperforming the optimal linear equalizer at moderate SNRs, and that the proposed approach greatly outperforms a similar equalization-only algorithm.

REFERENCES

- [1] Z. Ding and Y. Li, Eds., *Blind Equalization and Identification*. New York: Marcel Dekker, 2001.
- [2] A. Doucet, J. de Freitas, and N. Gordon, Eds., *Sequential Monte-Carlo Methods in Practice*. Berlin, Germany: Springer-Verlag, 2000.
- [3] J. S. Liu and R. Chen, “Blind deconvolution via sequential imputations,” *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 567–576, june 1995.
- [4] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, “Particle filtering,” *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, september 2003.
- [5] E. Punskeya, C. Andrieu, A. Doucet, and W. Fitzgerald, “Particle filtering for multiuser detection in fading cdma channels,” in *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing, 2001.*, Singapore, august 2001.
- [6] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 174–188, february 2002.
- [7] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel-approach to nonlinear non-gaussian bayesian state estimation,” *IEE Proceedings-F Image & Signal Processing*, vol. 140, no. 2, pp. 107–113, april 1993.
- [8] B. Dong, X. Wang, and A. Doucet, “A new class of soft mimo demodulation algorithms,” *IEEE Transactions on Signal Processing*, vol. 51, no. 11, pp. 2752–2763, november 2003.
- [9] H. L. van Trees, *Detection, Estimation, and Modulation Theory, Part I*. Wiley, 1968.
- [10] J. G. Proakis and M. Salehi, *Communication Systems Engineering*. Englewood Cliffs: Prentice Hall, NJ, 1994.
- [11] D. Warrier, U. Madhow, and R. Koetter, “Coding for noncoherent communication,” in *Proceedings of the International Symposium on Information Theory, Sorrento - Italy, 2000*.

APPENDIX

TABLE I
DETERMINISTIC PARTICLE FILTER ALGORITHM

```

Set  $m = 0$  (initialization)
For  $i = 1, \dots, M$ 
  - Draw  $x_0^{(i)}$  from  $\pi(x_0)$ .
  - Set the weight  $w_0^{(i)} = 1/M$ .
For  $m = 1, \dots, n$ 
  For  $i = 1, \dots, M$ 
    For  $j = 1, \dots, K$ 
      - Calculate
         $w^{(i,j)} = w_m^{(i)} p(x_m = x^{(j)}, y_m | x_{0:m-1}, y_{0:m-1})$ 
    - Determine  $(I, J)_t$ , the sequence of  $M$  pairs  $(i, j)$ 
      corresponding to the  $M$  largest  $w^{(i,j)}$ .
    For  $t = 1, \dots, M$ 
      - Make  $x_{0:m-1}^{(t)} = x_{0:m-1}^{(I_t)}$  and  $x_m^{(t)} = x^{(J_t)}$ .
      - Make  $w_m^{(t)} = w^{(I, J)_t}$ .
    - Normalize the weights.
  - Estimate  $p(x_{0:m} | y_{0:m})$  as in (2).

```

Equalização Cega de Canais Lineares FIR empregando o Algoritmo EM Bayesiano

Claudio J. Bordin Jr. e Luiz A. Baccalá

Resumo—Neste trabalho descrevemos métodos para equalizar cegamente canais de comunicação lineares FIR através da versão Bayesiana do consagrado algoritmo EM (*Expectation-Maximization*). Os métodos desenvolvidos fornecem estimativas que maximizam localmente a densidade de probabilidade *a posteriori* dos parâmetros do canal de comunicação, estimando os símbolos transmitidos através do algoritmo BCJR. Para contornar dificuldades de convergência observadas sob baixos níveis de ruído, modificamos a proposta original ao considerar técnicas de *annealing*, que proporcionam um desempenho próximo ao ótimo (Viterbi) sem incorrer em aumento de complexidade computacional, conforme atestam simulações numéricas.

Palavras-Chave—Equalização cega, Algoritmo *Expectation-Maximization* Bayesiano, *Annealing*.

Abstract—This work describes blind equalization methods for linear FIR channels using a Bayesian version of the well-known EM (*Expectation-Maximization*) algorithm. The proposed methods produce estimates that locally maximize the posterior density of the channel parameters, leading to estimates of the transmitted bits via the BCJR algorithm. Without increasing the overall computational complexity, we next propose an enhanced scheme based on annealing techniques to circumvent bad convergence properties observed in the original algorithm when the noise level is low, leading to performances close to optimal (Viterbi for known channel parameters) as confirmed by numerical simulations.

Keywords—Blind Equalization, Bayesian *Expectation-Maximization* Algorithm, *Annealing*.

I. INTRODUÇÃO

A despeito dos avanços observados nas últimas décadas, a equalização cega [1] [2] de canais de comunicação ainda carece de soluções simultaneamente satisfatórias nos quesitos robustez, velocidade de convergência e complexidade computacional. Neste contexto, o algoritmo EM [3] oferece uma solução de compromisso para o problema da equalização cega ao prover estimativas aproximadamente ótimas para os parâmetros do canal de transmissão, embora a uma complexidade computacional relativamente elevada.

Introduzido em 1977 por Dempster *et al.* [4], o algoritmo EM é um método iterativo que aproxima soluções ótimas (nos sentidos ML ou MAP [5]) para problemas de estimação com dados latentes, ou seja, problemas nos quais a função de verossimilhança (a densidade *a posteriori*) somente é tratável se condicionada a um conjunto de dados não-observados.

Apesar de terem-se desenvolvido diversos métodos de obtenção de soluções ótimas no sentido da máxima verossimilhança (ML) para o problema da equalização cega utilizando o algoritmo EM ([6][7]), neste trabalho propomos

um novo método, desta vez de cunho Bayesiano. Nele obtêm-se aproximações para a estimativa MAP dos parâmetros do canal admitindo-se para este uma densidade *a priori* gaussiana. Em conseqüência à análise dos resultados obtidos com esta abordagem, propomos um segundo método para equalização cega empregando uma técnica de *annealing* que corrige deficiências de desempenho observadas quando a relação sinal-ruído disponível é alta.

O conteúdo deste trabalho está organizado da seguinte forma: na Sec. II apresentamos o algoritmo EM, descrevendo em seguida o modelo de sinal empregado (Sec. III). Na Sec. IV, descrevemos a aplicação do algoritmo EM à solução do problema da equalização cega, avaliando seu desempenho através de simulações numéricas. Na Sec. V apresentamos o algoritmo modificado e seu desempenho, conduzindo às conclusões formalizadas na Sec. VI.

II. O ALGORITMO EM

Dado que se deseje inferir um vetor aleatório θ a partir de um conjunto de observações A e considerando uma variável aleatória B latente (i.e. não-observada), de sorte que se possa descrever θ por meio de uma $p(\theta|A, B)$ analiticamente simples, o algoritmo EM prescreve como obter uma seqüência de estimativas $\theta^{(i)}$, $i \geq 0$, tais que $p(\theta^{(i+1)}|A) \geq p(\theta^{(i)}|A)$. Em outras palavras, o algoritmo EM permite obter valores $\theta^{(i)}$ que maximizam localmente a densidade *a posteriori* da variável inferida (veja o Apêndice para detalhes).

Dividido basicamente em dois passos, o algoritmo EM consiste primeiramente (passo E) em determinar a esperança da densidade *a posteriori* aumentada $p(\theta|A, B)$ dadas as observações A e uma estimativa atual $\theta^{(i)}$. A isto segue-se (passo M) o cálculo da nova estimativa de $\theta^{(i+1)}$, obtida ao maximizar-se a esperança determinada no passo E em função de θ , conforme descrito na Tabela I.

Como se pode observar na Tabela I, o algoritmo EM é um procedimento bastante geral. Como tal, tem aplicabilidade a diversos problemas de estimação, sendo sua praticidade determinada pela capacidade de se calcular e maximizar $Q(\theta, \theta^{(i)})$ de modo fácil. Cabe salientar que as estimativas obtidas não são necessariamente boas aproximações da estimativa MAP do vetor aleatório θ , já que a convergência do algoritmo só é garantida até máximos locais da densidade *a posteriori* [8], propriedade que demonstramos no Apêndice.

III. MODELO DE SINAL

Neste trabalho temos como principal interesse a equalização de canais de comunicação lineares com resposta ao impulso de duração finita. Assim, admitimos que a amostra y_k do sinal

Escola Politécnica, Universidade de São Paulo, São Paulo, Brasil, E-mails: {bordin,baccala}@lcs.poli.usp.br.

1) Passo E (*expectation*):

$$Q(\theta, \theta^{(i)}) = \int_B \log p(\theta|A, B) p(B|A, \theta^{(i)}) dB \quad (1)$$

2) Passo M (*maximization*):

$$\theta^{(i+1)} = \arg \max_{\theta} Q(\theta, \theta^{(i)}) \quad (2)$$

3) Itera-se até a convergência, ou seja, até que $\|\theta^{(i+1)} - \theta^{(i)}\|$ seja suficientemente pequeno.

TABELA I
ALGORITMO EM

recebido no instante k (tomada à taxa de símbolos) pode ser expressa como:

$$y_k = \sum_{i=0}^{L-1} h_i x_{k-i} + v_k, \quad (3)$$

em que L denota a duração da resposta ao impulso do canal (em intervalos de símbolo), x_k denota a seqüência de símbolos transmitida, h_k os termos da resposta impulsiva do canal e v_k o ruído aditivo, por hipótese gaussiano complexo circular com média nula e variância σ^2 conhecida.

Observe ainda que o modelo de sinal dado em (3) pode ser representado alternativamente como

$$y_k = h^H X_k + v_k, \quad (4)$$

ao fazer com que $X_k = [x_k \cdots x_{k-L+1}]^T \in \mathbb{C}^{L \times 1}$ seja um vetor que coleciona os valores dos últimos L símbolos transmitidos e $h = [h_0 \cdots h_{L-1}]^T \in \mathbb{C}^{L \times 1}$ seja o vetor que coleciona os coeficientes do canal.

IV. EQUALIZAÇÃO CEGA EMPREGANDO O ALGORITMO EM

Neste trabalho temos como objetivo obter estimativas MAP dos parâmetros do canal de transmissão h , ou seja, determinar o conjunto de parâmetros h que maximiza a densidade *a posteriori* $p(h|y_{0:n}, \sigma^2)$, $y_{0:n} \triangleq \{y_0, \dots, y_n\}$.

Pode-se mostrar que a determinação exata da densidade $p(h|y_{0:n}, \sigma^2)$ é impraticável [9], uma vez que exige um número de operações que cresce exponencialmente em função do tamanho do bloco de dados n . No entanto, conforme mostramos nesta seção, o modelo de sinal descrito por (4) permite que se obtenham diretamente estimativas MAP dos parâmetros h através do algoritmo EM.

Utilizando como variável latente a seqüência de vetores $X_{0:n}$, assumida independente *a priori* do vetor de parâmetros h , obtemos que a densidade *a posteriori* aumentada

$p(h|X_{0:n}, y_{0:n}, \sigma^2)$ pode ser expressa¹ como

$$p(h|X_{0:n}, y_{0:n}) = \frac{p(h, X_{0:n}, y_{0:n})}{\int_h p(h, X_{0:n}, y_{0:n}) dh} \quad (5)$$

$$\propto p(h, X_{0:n}, y_{0:n}) \quad (6)$$

$$= p(y_{0:n}|h, X_{0:n}) p(h) p(X_{0:n}) \quad (7)$$

$$= p(h) \prod_{j=0}^n p(y_j|h, X_j) p(x_j). \quad (8)$$

Aplicando-se o algoritmo EM para conseguir estimativas de h , obtemos a seguinte expressão para o “passo E”:

$$Q(h, h^{(i)}) = \sum_{X_{0:n}} \log p(h|X_{0:n}, y_{0:n}) p(X_{0:n}|h^{(i)}, y_{0:n}) \quad (9)$$

$$= E \left[\log p(h|X_{0:n}, y_{0:n}) | h^{(i)}, y_{0:n} \right]. \quad (10)$$

Assumindo que a densidade *a priori* dos parâmetros do canal é Gaussiana complexa, ou seja,

$$p(h) = (\pi)^{-N} |\Sigma_0|^{-1} \exp \left(-\|h - h_0\|_{\Sigma_0^{-1}}^2 \right) \triangleq \mathcal{N}_C(h|h_0; \Sigma_0), \quad (11)$$

e que os símbolos transmitidos são i.i.d equiprováveis, segue que a verossimilhança das observações é dada por

$$p(y_j|h, X_j) = \mathcal{N}_C(y_j|h^H X_j; \sigma^2). \quad (12)$$

O logaritmo da densidade *a posteriori* aumentada dada em (8) pode ser escrito então como

$$\log p(h|X_{0:n}, y_{0:n}) = \alpha - \|h - h_0\|_{\Sigma_0^{-1}}^2 - \frac{1}{\sigma^2} \sum_{j=0}^n \|y_j - h^H X_j\|^2. \quad (13)$$

sendo α uma constante de normalização independente de h . Calculando-se então a esperança condicional de (13), obtemos

$$Q(h, h^{(i)}) = -\frac{1}{\sigma^2} \sum_{j=0}^n \left(E \left[\|y_j - h^H X_j\|^2 | h^{(i)}, y_{0:n} \right] - \|h - h_0\|_{\Sigma_0^{-1}}^2 + \alpha' \right) \quad (14)$$

$$= -\frac{1}{\sigma^2} \sum_{j=0}^n \left(-2\Re \left\{ h^H E \left[X_j | h^{(i)}, y_{0:n} \right] y_j^* \right\} + h^H E \left[X_j X_j^H | h^{(i)}, y_{0:n} \right] h \right) - \left(-2\Re \left\{ h^H \Sigma_0^{-1} h_0 \right\} + h^H \Sigma_0^{-1} h \right) + \alpha''. \quad (15)$$

em que α' e α'' são constantes que não dependem de h .

O segundo passo do algoritmo EM (passo “M”) consiste de determinar o valor de h que maximiza (15). Para isto, note inicialmente que

$$\frac{\partial}{\partial h} Q(h, h^{(i)}) = - \left(\Sigma_0^{-1} + \frac{1}{\sigma^2} \sum_{j=0}^n E \left[X_j X_j^H | h^{(i)}, y_{0:n} \right] \right) h + \left(\Sigma_0^{-1} h_0 + \frac{1}{\sigma^2} \sum_{j=0}^n E \left[X_j | h^{(i)}, y_{0:n} \right] y_j^* \right). \quad (16)$$

¹Omitiremos a partir deste ponto a dependência das densidades condicionais em relação ao parâmetro σ^2 (variância do ruído aditivo), uma vez que este é assumido conhecido.

e que a matriz Hessiana $\partial^2 Q(h, h^{(i)})/\partial h^2$ é dada por

$$-\left(\Sigma_0^{-1} + \frac{1}{\sigma^2} \sum_{j=0}^n E \left[X_j X_j^H | h^{(i)}, y_{0:n} \right]\right) < 0. \quad (17)$$

Igualando-se (16) a zero, segue que o máximo de $Q(h, h^{(i)})$ é dado por

$$h^{(i+1)} = \left(\Sigma_0^{-1} + \frac{1}{\sigma^2} \sum_{j=0}^n E \left[X_j X_j^H | h^{(i)}, y_{0:n} \right] \right)^{-1} \left(\Sigma_0^{-1} h_0 + \frac{1}{\sigma^2} \sum_{j=0}^n E \left[X_j | h^{(i)}, y_{0:n} \right] y_j^* \right) \quad (18)$$

As esperanças em (18) podem ser calculadas numericamente através do algoritmo BCJR [10] (vide Apêndice), uma vez que

$$E \left[X_j | h^{(i)}, y_{0:n} \right] = \sum_{i=1}^{N_s} \mathcal{X}_i \mathbb{P}(X_j = \mathcal{X}_i | h^{(i)}, y_{0:n}) \quad (19)$$

$$E \left[X_j X_j^H | h^{(i)}, y_{0:n} \right] = \sum_{i=1}^{N_s} \mathcal{X}_i \mathcal{X}_i^H \mathbb{P}(X_j = \mathcal{X}_i | h^{(i)}, y_{0:n}) \quad (20)$$

sendo N_s o número de vetores X_j distintos.

Na Tabela II resumimos o conteúdo desta seção, descrevendo o algoritmo proposto. Note que aproximações das estimativas² MAP/ML dos símbolos transmitidos podem ser obtidas a partir de $\mathbb{P}(X_j | h^{(i)}, y_{0:n})$, uma vez que a densidade *a posteriori* dos símbolos transmitidos é dada por

$$\mathbb{P}(x_j = a | h^{(i)}, y_{0:n}) = \sum_{\forall b | x_j = a} \mathbb{P}(X_j = b | h^{(i)}, y_{0:n}). \quad (21)$$

(Inicialização) Obtenha $h^{(0)} \sim \mathcal{N}_C(h|0; \Sigma_0)$
Para $i \geq 0$, itere:

- Calcule $\mathbb{P}(X_j | h^{(i)}, y_{0:n})$, $0 \leq j \leq n$ através do algoritmo BCJR.
- Determine $h^{(i+1)}$ através de (18)-(20).

TABELA II

ALGORITMO PARA EQUALIZAÇÃO CEGA BASEADO NO ALGORITMO EM

A. Simulações

Através de simulações numéricas avaliamos o desempenho do algoritmo segundo a taxa média de erro de bit (BER) obtida ao longo de 100 realizações independentes num sistema de comunicação que transmite blocos de 250 símbolos binários i.i.d. através do canal $h = [0, 41 - 0, 82 \ 0, 41]^T$ com ordem e potência de ruído conhecidas pelo receptor. A densidade *a priori* do canal foi admitida igual a $\mathcal{N}_C(h|0; I)$.

Na Fig. 1 mostramos o desempenho obtido em função da relação sinal-ruído e do número de iterações empregado pelo

²Observe que as estimativas obtidas através deste método são subótimas, uma vez que $h^{(i)}$ é uma estimativa dos parâmetros do canal.

algoritmo EM. Na mesma figura, para efeito de comparação, mostramos o desempenho obtido pelo algoritmo de Viterbi, construído a partir do canal conhecido e implementado considerando-se desconhecidos os bits das extremidades do bloco. Uma característica que salta aos olhos é o desempenho ruim do algoritmo da Sec. IV que piora a medida que a relação sinal ruído ultrapassa 6dB.

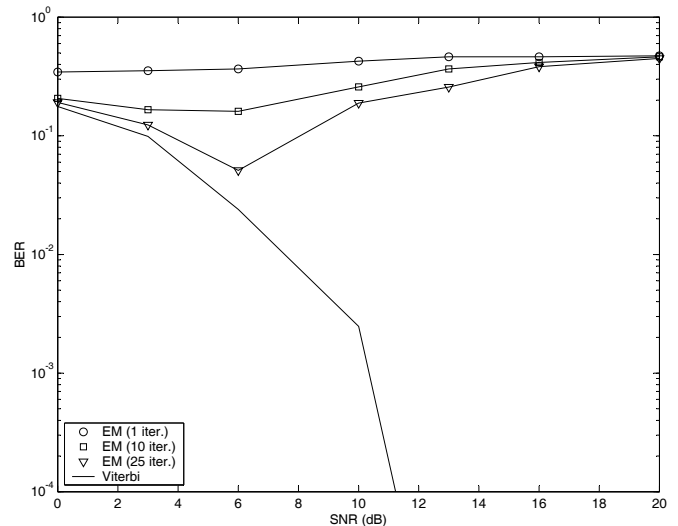


Fig. 1. Desempenho médio ao longo de 100 realizações independentes do algoritmo para equalização cega descrito na Sec. IV em função da relação sinal-ruído(SNR) e o número de iterações.

V. ALGORITMO MELHORADO

Analisando mais detidamente o funcionamento do algoritmo da Sec. IV observamos que, sob baixos níveis de ruído, as densidades $p(X_k | h^{(i)}, y_{0:n})$ estimadas (através do algoritmo BCJR) assumem valores muito próximos de 0 ou 1, num comportamento que parece contribuir para o aparecimento de algo semelhante a um ciclo-limite³, que impede a convergência do algoritmo para uma solução fixa.

Para contornar este problema, fizemos uma alteração de caráter heurístico no algoritmo da Sec. IV inspirada em técnicas de *simulated annealing*. Estas técnicas [11], amplamente empregadas na solução de problemas de otimização não-linear, baseiam-se na observação de fenômenos naturais ligados à cristalização de metais, que sugerem que a convergência para o mínimo global num problema de minimização iterativa pode ser obtida pela introdução de perturbações lentamente decrescentes (nas suas soluções intermediárias).

Para evitar o aparecimento de oscilações cíclicas como as observadas para o algoritmo anteriormente descrito, substituímos a densidade $p(X_k | h^{(i)}, y_{0:n}, \sigma^2)$ (calculada através do algoritmo BCJR), necessária para a determinação de (20), pela densidade $\bar{p}_i(X_k | h^{(i)}, y_{0:n}, \sigma^2) = p(X_k | h^{(i)}, y_{0:n}, \sigma^2 + \delta(i))$, com $\delta(i) > 0$ é uma função decrescente de i .

Note que, dado o maior nível de ruído informado ao algoritmo BCJR, a densidade $\bar{p}_i(X_k | h^{(i)}, y_{0:n}, \sigma^2)$ exibe uma

³Note que o algoritmo EM opera de forma determinística, e que os valores de $h^{(i)}$ e $y_{0:n}$ numa iteração definem completamente o comportamento do algoritmo nas iterações posteriores.

menor tendência de assumir valores extremos (0 ou 1) do que a densidade *a posteriori* verdadeira. Isto parece contribuir para a eliminação de oscilações cíclicas. Em nossas simulações, testamos diversas formas para a função $\delta(i)$, obtendo melhores resultados para funções $\delta(i)$ da forma exponencial.

A. Simulações

Nas Figs. 2 e 3 mostramos o desempenho obtido pela versão modificada do algoritmo da Sec. IV, utilizando $\delta(i) = 10^{-0,1i}$ e $\delta(i) = 10^{-0,05i}$, respectivamente, empregando as mesmas hipóteses adotadas na Sec. IV-A.

Como se pode observar, ao fazer $\delta(i) = 10^{-0,1i}$ nota-se ainda uma degradação de desempenho para SNR superiores a 12 dB (Fig. 2). Diminuindo-se o nível de ruído adicional mais lentamente ($\delta(i) = 10^{-0,05i}$), no entanto, o desempenho do algoritmo modificado aproxima-se bastante daquele do detector ótimo MLSE (Viterbi) para 25 iterações, como se pode observar na Fig. 3.

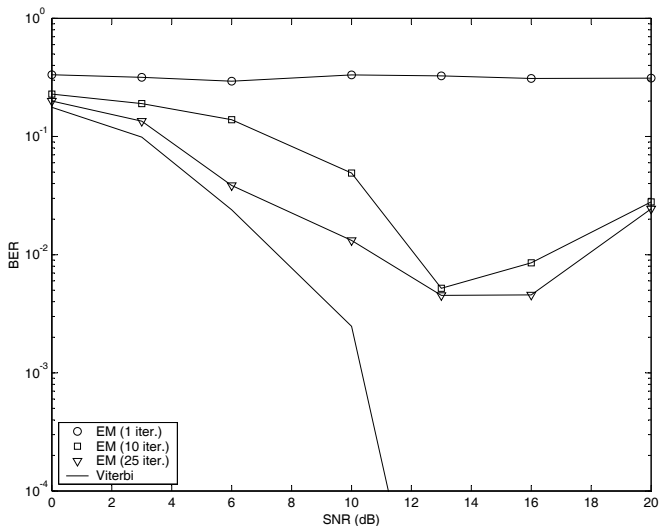


Fig. 2. Desempenho médio ao longo de 100 realizações independentes da versão modificada do algoritmo para equalização cega descrito na Sec. V ($\delta(i) = 10^{0,1i}$) em função da relação sinal-ruído(SNR) e o número de iterações.

VI. CONCLUSÕES

Neste trabalho descrevemos métodos para a equalização cega de canais de comunicação lineares FIR baseados na versão Bayesiana do algoritmo EM (*Expectation-Maximization*). Verificamos que o método básico, obtido através da aplicação direta do algoritmo EM ao problema tratado, apresenta um desempenho insatisfatório sob baixos níveis de ruído. Este resultado nos levou a propor um segundo esquema, empregando uma modificação heurística inspirada nas técnicas de *simulated annealing*, o qual, sem incorrer num aumento da complexidade computacional, proporciona um desempenho próximo ao ótimo, conforme verificado através simulações numéricas.

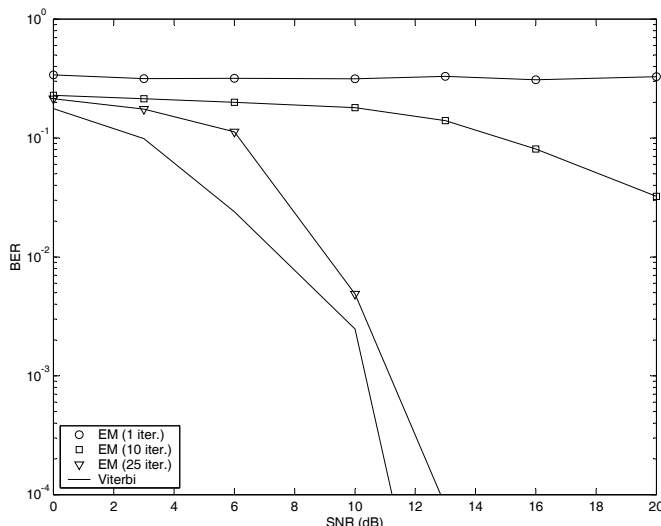


Fig. 3. Desempenho médio ao longo de 100 realizações independentes da versão modificada do algoritmo para equalização cega descrito na Sec. V ($\delta(i) = 10^{0,05i}$) em função da relação sinal-ruído(SNR) e o número de iterações.

REFERÊNCIAS

- [1] S. Haykin, ed., *Blind Deconvolution*. Englewood Cliffs, NJ: Prentice Hall, 1994.
- [2] Z. Ding and Y. Li, eds., *Blind Equalization and Identification*. New York: Marcel Dekker, 2001.
- [3] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer Verlag, 1999.
- [4] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.
- [5] L. Scharf, *Statistical signal processing: detection, estimation, and time series analysis*. Addison-Wesley Pub. Co., 1991.
- [6] K. Dogancay and V. Krishnamurthy, "Blind identification of fractionally spaced communication channels with markov inputs," *Signal Processing*, vol. 77, pp. 275–288, Setembro 1999.
- [7] M. Nissilä and S. Pasupathy, "Adaptive bayesian and em-based detectors for frequency-selective fading channels," *IEEE Transactions on Communications*, vol. 51, pp. 1325–1336, Agosto 2003.
- [8] M. A. Tanner, *Tools for Statistical Inference*. Springer Verlag, 3 ed., 1996.
- [9] A. Doucet, C. Andrieu, and R. Urien, "Bayesian blind and semi-blind equalization of channels with markov inputs," *IEE Proceedings-F Image & Signal Processing*, vol. 148, no. 4, pp. 269–274, 2001.
- [10] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, Março 1974.
- [11] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [12] G. D. Forney, "The {V}iterbi algorithm," *Proceedings of IEEE*, vol. 61, no. 3, pp. 268–278, Março 1973.

APÊNDICE

A. Convergência do Algoritmo EM

Proposição: Iterando-se o algoritmo da Tabela I tem-se que $p(\theta^{(i+1)}|A) \geq p(\theta^{(i)}|A)$.

Prova: Inicialmente, note que como uma consequência lei de Bayes

$$p(\theta|A, B) = \frac{p(\theta|A)p(B|\theta, A)}{p(B|A)}, \quad (22)$$

o que implica que

$$\log[p(\theta|A)] = \log[p(\theta|A, B)] + \log[p(B|A)] - \log[p(B|\theta, A)]. \quad (23)$$

Integrando-se cada uma destas densidades com respeito a $p(B|\theta^-, A)$:

$$\begin{aligned} \log[p(\theta|A)] &= \int_B \log[p(\theta|A, B)] p(B|\theta^-, A) dB \\ &+ \int_B \log[p(B|A)] p(B|\theta^-, A) dB \\ &- \int_B \log[p(B|\theta, A)] p(B|\theta^-, A) dB \\ &\triangleq Q(\theta, \theta^-) + K(\theta, \theta^-) - H(\theta, \theta^-) \end{aligned} \quad (24)$$

Fazendo-se $\theta^- = \theta^{(i)}$ obtemos

$$\begin{aligned} \log[p(\theta^{(i+1)}|A)] - \log[p(\theta^{(i)}|A)] &= \\ &[Q(\theta^{(i+1)}, \theta^{(i)}) - Q(\theta^{(i)}, \theta^{(i)})] \\ &+ [H(\theta^{(i+1)}, \theta^{(i)}) - H(\theta^{(i)}, \theta^{(i)})] \\ &- [K(\theta^{(i+1)}, \theta^{(i)}) - K(\theta^{(i)}, \theta^{(i)})] \end{aligned} \quad (25)$$

Como, por definição, $\theta^{(i+1)}$ é escolhido de modo a maximizar $Q(\theta^{(i+1)}, \theta^{(i)})$, o primeiro termo em (25) é positivo (ou nulo se $\theta^{(i+1)} = \theta^{(i)}$). Pode-se verificar também que o segundo termo é nulo, já que $H(\theta^{(i+1)}, \theta^{(i)})$ não é função de $\theta^{(i+1)}$. Quanto ao terceiro termo, note que

$$\begin{aligned} -[K(\theta^{(i+1)}, \theta^{(i)}) - K(\theta^{(i)}, \theta^{(i)})] &\triangleq \\ &\int_B \log \left[\frac{p(B|\theta^{(i)}, A)}{p(B|\theta^{(i+1)}, A)} \right] p(B|\theta^{(i)}, A) dB \end{aligned} \quad (26)$$

ou seja, a diferença equivale à distância de Kullback-Leibler entre as densidades $p(B|\theta^{(i)}, A)$ e $p(B|\theta^{(i+1)}, A)$, sendo portanto positiva (ou nula de as duas densidades forem iguais “almost everywhere”). Assim, conclui-se que $\log[p(\theta^{(i+1)}|A)] - \log[p(\theta^{(i)}|A)] \geq 0$ concluindo a demonstração. \square

B. O Algoritmo BCJR

O Algoritmo BCJR [10], desenvolvido por Bahl *et al.* em 1974 descreve um método numérico para o cálculo das probabilidades de ocupação dos estados de um HMM (*Hidden Markov Model*). O método é implementado através de duas recursões (para trás e para frente), sendo a primeira delas bastante similar à descrita pelo Algoritmo de Viterbi [12].

Para se calcular a distribuição *a posteriori* da variável de estado X_k (Eq. 20), o algoritmo BCJR explora o fato de que $y_{0:n}$ são observações de um HMM, obtendo que

$$\begin{aligned} p(X_{k+1} = b | y_{0:n}) &= \sum_a p(X_k = a, X_{k+1} = b | y_{0:n}, h) \\ &\propto \sum_a \alpha_k(a) \gamma_k(a, b) \beta_{k+1}(b), \end{aligned} \quad (27)$$

onde

$$\alpha_k(a) = p(X_k = a, y_{0:k-1}) \quad (28)$$

$$\beta_{k+1}(b) = p(y_{k+1:n} | X_{k+1} = b) \quad (29)$$

$$\gamma_k(a, b) = p(X_{k+1} = b, y_k | X_k = a). \quad (30)$$

Sob as hipóteses adotadas para o modelo de sinal (Eq. 3), os termos $\gamma_k(a, b)$ podem ser calculados diretamente, obtendo-se que

$$\gamma_k(a, b) = \frac{1}{\pi\sigma^2} \exp\left(-\frac{1}{\sigma^2}|y_k - h^H X_k|^2\right) p(x_k = s^{(a,b)}), \quad (31)$$

onde $s^{(a,b)}$ denota o valor assumido pela v.a. x_k que provoca a transição do estado $X_k = a$ para o estado $X_{k+1} = b$.

Os demais termos em (30) somente podem ser determinados recursivamente. Para isto, introduzimos a seguinte notação vetorial

$$\alpha_k = [\alpha_k(1) \dots \alpha_k(N_s)]^T \in \mathbb{R}^{|N_s| \times 1} \quad (32)$$

$$\beta_k = [\beta_k(1) \dots \beta_k(N_s)]^T \in \mathbb{R}^{|N_s| \times 1}, \quad (33)$$

e a matriz

$$[\mathbf{\Gamma}_k]_{ij} = \gamma_k(i, j), \quad (34)$$

onde N_s é o número de valores distintos que o vetor aleatório X_k pode assumir.

Utilizando estas definições, obtém-se que os demais termos podem ser calculados através das recursões [10]

$$\alpha_{k+1} = \mathbf{\Gamma}_k^T \alpha_k \quad (35)$$

$$\beta_k = \mathbf{\Gamma}_k \beta_{k+1}, \quad (36)$$

iniciadas com $\beta_k(a) = p(y_k | X_k = a)$ e $\alpha_0(a) = p(X_0 = a)$.

Numa formulação geral, a aplicação do algoritmo BCJR a um bloco de n amostras exige um total de operações da ordem de nN_s^2 . Ao aplicar-se o algoritmo BCJR num sistema de comunicações digitais binário, no entanto, verifica-se que a matriz $\mathbf{\Gamma}_k$ somente possui $2N_s$ elementos não-nulos, onde $N_s = 2^L$, sendo a complexidade total uma função linear de N_s .