Anais do IX Congresso Brasileiro de Redes Neurais / Inteligência Computacional (IX CBRN)
Ouro Preto 25-28 de Outubro de 2009

ANÁLISES EXATA E APROXIMADA DE UM ALGORITMO DE DECONVOLUÇÃO CEGA

Diego Barreto Haddad [†*], Mariane Rembold Petraglia[†], Paulo Bulkool Batalheiro[‡]

[*] *Dept. Telecomunicações*
CEFET-RJ UnEd Nova Iguaçu
CEP 26041-271, Nova Iguaçu, RJ, Brasil

[†] *Programa de Engenharia Elétrica, COPPE - DEL/Poli*
Universidade Federal do Rio de Janeiro
CP 68504, CEP 21945-970, Rio de Janeiro, RJ, Brasil

[‡] *Departamento de Engenharia Eletrônica e Telecomunicações*
Universidade do Estado do Rio de Janeiro
CEP 20559-900, Rio de Janeiro, RJ, Brasil

Emails: diego@pads.ufrj.br, mariane@pads.ufrj.br, bulkool@pads.ufrj.br

Abstract— Adaptive blind processing algorithms present many applications. Their mathematical grounds are in mature stage. Nevertheless, there are few performance and convergence analyses in the literatura of blind source separation/deconvolution of signals. This paper presents two analyses - an approximated and an exact one - of a well know blind deconvolution algorithm.

Keywords— Blind Deconvolution, Adaptive Blind Signal Processing, Independent Component Analysis.

Resumo— Algoritmos de processamento adaptativo autodidata apresentam inúmeras aplicações. Seus fundamentos matemáticos encontram-se maduros. Porém, na literatura existem poucas análises de desempenho e de convergência de algoritmos de separação/deconvolução cega de sinais. Este artigo apresenta duas análises - uma aproximada e outra exata - de um conhecido algoritmo de deconvolução cega.

Keywords— Deconvolução cega, Processamento Adaptativo Cego de Sinais, Análise de Componentes Independentes.

1 Introdução

Embora haja na literatura muitos artigos dedicados à análise de desempenho de algoritmos de processamento supervisionado de sinais, relativamente poucas análises contemplam técnicas de processamento cego. Entre estas, a maioria trata do caso de misturas instantâneas. Este número reduzido de análises reflete a maior dificuldade de tratamento analítico das técnicas cegas. Duas são as características basicamente responsáveis por essa dificuldade: não-linearidade e impossibilidade de decomposição do problema multidimensional em diversos problemas unidimensionais independentes (Minker (2007)).

Recuperar um sinal distorcido por um canal é a tarefa de um algoritmo de equalização. Não raro este algoritmo é supervisionado, por exigir um período de treinamento, no qual um trecho do sinal a recuperar é conhecido. Quando inexistem informações tanto acerca do canal quanto do sinal (apenas conhecidas algumas características estatísticas deste), o algoritmo é denominado *cego*.

Um conhecido algoritmo de deconvolução cega de fontes é apresentado em S. C. Douglas (2005), onde afirma-se ser difícil efetuar uma análise acurada da convergência do algoritmo. Este é o propósito deste artigo.

2 Algoritmo DCGN monocanal

O algoritmo sob questão é denominado aqui pela sigla DCGN (de Deconvolução Cega utilizando Gradiente Natural). O algoritmo DCGN é uma evolução do algoritmo original apresentado em S. Amari (1997), cujo gradiente natural apresentava um viés que reduzia o desempenho final[1].

Seja $h(n)$ o filtro de ordem M correspondente ao canal e $s(n)$ o sinal a recuperar (fonte). Logo, no receptor, o sinal recebido $x(n)$ é dado por $x(n) = (s * h)(n)$. Seja w o filtro de ordem L do equalizador. Definindo:

$$\mathbf{w}(n) = \begin{bmatrix} w_0(n) & w_1(n) & w_2(n) & \dots & w_L(n) \end{bmatrix}, \tag{1}$$

$$\mathbf{x}(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ x(n-2) \\ \vdots \\ x(n-L) \end{bmatrix}, \tag{2}$$

[1] A iteração de atualização dos parâmetros incluía termos oriundos de amostras de sinais de entrada as quais não influenciavam a função custo. Estes termos adicionais foram introduzidos ao se calcular o "gradiente natural"; uma forma diferente de truncamento da seqüência de entrada elimina estes termos prejudiciais (vide S. C. Douglas (2005) para maiores detalhes).

temos que a n-ésima amostra da saída (na n-ésima iteração) do equalizador é dada por:

$$y(n) = \mathbf{w}(n)\mathbf{x}(n) \tag{3}$$

O vetor $\mathbf{w}(n)$ é adaptado de modo *on-line*, de forma que o vetor $\mathbf{w}(n)$ em geral é diferente de $\mathbf{w}(n-1)$.
O método DCGN supõe um conhecimento acerca da distribuição das amostras de $s(n)$ (supostas iid - independentes e identicamente distribuídas). Este é o único parâmetro de *nuisance* (incômodo; vide Cardoso (1998)) do algoritmo. Sendo esta distribuição $p(s)$, podemos definir a função f ideal como:

$$f(y(n)) = -\frac{\mathrm{d}\log p(y(n))}{\mathrm{d}y(n)}, \tag{4}$$

embora uma função f diferente da real possa angariar resultados satisfatórios de deconvolução.
Seja a matriz simétrica $\mathbf{R}(n)$ de autocorrelação dos coeficientes $w_i(n)$ dada por:

$$\mathbf{R}(n) = \begin{bmatrix} r_n(0) & r_n(1) & \dots & r_n(L) \\ r_n(-1) & r_n(0) & \dots & r_n(L-1) \\ \vdots & \ddots & \ddots & \vdots \\ r_n(-L) & r_n(-L+1) & \cdots & r_n(0) \end{bmatrix}, \tag{5}$$

onde

$$r_n(l) = \sum_{p=0}^{L-|l|} w_p(n) w_{p+|l|}(n). \tag{6}$$

Se $\mathbf{z}(n) = \mathbf{R}(n)\mathbf{x}(n)$, a equação de atualização do algoritmo DCGN pode ser exposta como:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\left[\mathbf{w}(n) - f(y(n))\mathbf{z}^T(n)\right] \tag{7}$$

3 Análise de Primeira Ordem (APO) do Algoritmo DCGN

Faremos uma breve descrição da APO, pois já a expusemos em D. B. Haddad (2009). Aqui, embora exemplifiquemos por simplicidade a configuração $M = L = 1$, estenderemos nossa análise para valores genéricos de M e L por meio de um método de refinamento das soluções de equações não-lineares.
A APO parte de algumas hipóteses: (1) o algoritmo converge para uma solução de bom desempenho e (2) a fonte apresenta amostras iid, de média zero e distribuição simétrica.
Após a convergência e admitindo-se μ suficientemente pequeno (sendo portanto válida a suposição $w_i(n+1) \approx w_i(n)$), podemos escrever:

$$\begin{array}{c} 0 \approx w_0 - (w_0^2 + w_1^2)\mathrm{E}\left[f(y(n))x(n)\right] \\ -w_0 w_1 \mathrm{E}\left[f(y(n))x(n-1)\right] \end{array}, \tag{8}$$

$$\begin{array}{c} 0 \approx w_1 - (w_0^2 + w_1^2)\mathrm{E}\left[f(y(n))x(n-1)\right] \\ -w_0 w_1 \mathrm{E}\left[f(y(n))x(n)\right] \end{array}. \tag{9}$$

Importa analisar os termos $\mathrm{E}\left[f(y(n))x(n)\right]$ e $\mathrm{E}\left[f(y(n))x(n-1)\right]$. Supondo que o algoritmo, ao estabilizar-se, obteve uma solução de bom desempenho, podemos empregar a aproximação[2]:

$$f(y(n)) \approx f(c_1 s(n-1)) + \sum_{i=0, i\neq 1}^{2} c_i f'(c_1 s(n-1)) s(n-i), \tag{10}$$

onde c_n são os coeficientes do filtro global, de modo que $c(n) = (h * w)(n)$, onde supusemos que o maior valor em módulo de c_n é igual a c_1 (pode-se demonstrar que essa solução é a ótima). A aproximação acima permite-nos calcular as médias estatísticas. O resultado gera um sistema de equações não-lineares. Particularizando para o caso de distribuições exponenciais (vide D. B. Haddad (2009) e a Seção 5 para maiores detalhes), as equações finais são[3]:

$$\begin{array}{rcl} h_1 w_0^2 + (h_0 w_1 - 1) w_0 + h_1 w_1^2 & \approx & 0 \\ h_0 w_1^2 + (h_1 w_0 - 1) w_1 + h_0 w_0^2 & \approx & 0 \end{array} \tag{11}$$

Assim, temos uma técnica que permite-nos estimar os valores finais do filtro equalizador, a partir das soluções do sistema de equações. Estas estimativas possibilitam uma previsão do desempenho final do algoritmo. Para uma técnica de processamento cego, o ideal seria estimar o desempenho do algoritmo sem referência ao filtro do canal. Porém, isso é impossível, já que o desempenho final do algoritmo depende fortemente do canal. Eis um fato que dificulta a análise das configurações convolutivas. No entanto, uma análise que contemple o desempenho a partir do filtro do canal pode ser útil, pois um conhecimento estatístico acerca dos filtros do canal pode levar a alguma informação acerca da variabilidade do desempenho do algoritmo.

4 Configurações Genéricas

Nesta subseção, contemplamos as configurações onde M e L podem admitir qualquer valor. O cálculo algébrico para estas configurações é, em geral, muito trabalhoso. Por isso, criamos um código que automaticamente gera o sistema de equações associado à convergência do algoritmo para quaisquer valores de M e L e para qualquer atraso resultante superior ou igual a L (atrasos inferiores a L implicam estimativas subótimas).
O código calcula $\mathrm{E}\left[f(y(n))x(n-l)\right]$, simplificando os termos que são nulos, devido à independência entre amostras da fonte. Estes termos simplificados são utilizados para gerar o vetor $\mathrm{E}\left[f(y(n))\mathbf{x}^T(n)\right]$, o qual é empregado no termo $-f(y(n))\mathbf{z}^T(n)$ da equação (7) de atualização. O resultado é apresentado no formato texto, o qual pode ser diretamente

[2] Aqui utilizamos a hipótese de bom desempenho, o que implica que a aproximação linear não deve ser problemática.

[3] Supomos aqui que o canal é invariante no tempo e que $c_1 > 0$; para analisar a possibilidade $c_1 < 0$, basta inverter o sinal dos parâmetros w_0 e w_1.

empregado em linguagens de programação que possuam recursos de matemática simbólica.

O sistema de equações final pode ser resolvido por funções presentes em programas matemáticos (como por exemplo, a instrução *solve* do MATLAB). Para os sistemas de equações obtidos, as respostas destes métodos costumam ser bastante confiáveis. Porém, recorrer à instrução *solve* só é possível quando o número de equações é muito reduzido (da ordem de 3 ou, no máximo, 4 equações). Para um número maior de equações, a resolução do sistema de equações é problemática, pois a instrução *solve* do MATLAB adota o procedimento de primeiro buscar as soluções analíticas do sistema, para depois calculá-las numericamente. Devido às não-linearidades, o primeiro passo deste estratagema, do qual depende o segundo, não obtém sucesso para um número de equações elevado.

No entanto, o sistema de equações apresenta soluções bem próximas das obtidas pelo algoritmo. Porém, estas soluções não são as únicas. Para encontrar as soluções do sistema que mais se aproximam das reais, implementamos um algoritmo que refina continuamente uma inicialização aleatória (chute inicial), rumo a uma solução do sistema. Resolver numericamente sistemas de equações não-lineares é um tema secundário a este artigo, de modo que não exploramo-lo profundamente. O algoritmo implementado, embora simples, serviu completamente aos nossos propósitos, sendo resumido a seguir.

Inspirado nas técnicas de recozimento simulado (do inglês *Simulated Annealing*), o algoritmo calcula o resultado de cada uma das equações do sistema. Idealmente, o lado direito de todas as equações deveria resultar em zero (por exemplo, vide as Equações (11)). Seja $\mathbf{e}$ o vetor linha formado pelo lado direito de todas as equações. Inicialmente, $\mathbf{e}$ é um vetor não-nulo. O algoritmo almeja tornar cada um de seus componentes o mais próximo possível de zero. Para este objetivo, o algoritmo define a função custo:

$$F = \frac{\mathbf{e}\mathbf{e}^T}{L}, \tag{12}$$

basicamente o valor quadrático médio de $\mathbf{e}$.

O algoritmo, a cada iteração, adiciona uma perturbação gaussiana a uma das variáveis $(w_0, w_1, \ldots, w_L)$, verificando se a perturbação reduz a função custo. Se a função custo for reduzida, a perturbação é mantida; caso contrário, a variável mantém seu valor anterior. É interessante que a variância da perturbação gaussiana se reduza gradualmente, à medida em que nos aproximamos da solução do sistema de equações. Esta redução da variância não foi implementada, pois o tempo de convergência não foi particularmente alto.

Uma inicialização interessante do algoritmo de busca da solução é o conjunto de parâmetros obtidos pelo algoritmo DCGN. Em geral, estes parâmetros resultam num valor da função custo muito baixo (da ordem de 10^{-4}), o que indica que a inicialização está bem próxima de uma solução do sistema (segundo esperado). Em nossas simulações, os casos onde isso não acontecem são devidos à ambiguidade de escalamento, bastando alterar o sinal de todas as variáveis de inicialização para garantir uma função custo inicial de valor baixo. O algoritmo refina a estimativa inicial até que a função custo atinja um valor entre 10^{-17} e 10^{-19}, o que significa que estamos muito próximos da solução do sistema.

Uma objeção para esta escolha da inicialização é que ela pressupõe o conhecimento dos valores para os quais o algoritmo DCGN converge. Podemos interpretar a abordagem escolhida como a análise do melhor caso (ou seja, da solução que mais se aproxima dos valores reais para os quais o algoritmo converge). Porém, não é de todo gratuito este proceder: para sistemas com poucas equações, podemos encontrar todas as soluções facilmente. Dentre estas, a que apresenta a menor ISI sempre é a que mais se aproxima da solução para a qual o algoritmo converge. Por isso, uma técnica de resolução de sistema de equações que apresente todas as soluções poderia nos permitir escolher (de forma cega) qual é a estimativa do método proposto. Pretendemos explorar melhor esta tema no futuro.

5 Caso particular: distribuições exponenciais

Um caso particular da função f simplifica o sistema de equações não lineares, sendo portanto analisado a seguir. Esta análise também permite explicitar a metodologia de parte do cálculo das médias estatísticas (para a análise aproximada).

Suponhamos que a distribuição de $s(n)$ seja exponencial, de média zero e parâmetro $\gamma > 0$. Logo, temos:

$$p(s) = \frac{1}{2\gamma} e^{-\frac{|s|}{\gamma}} \tag{13}$$

Escolhendo f de forma que $f(s) = -\frac{\mathrm{d}\log p(s)}{\mathrm{d}s}$, temos:

$$f(s) = -\frac{\mathrm{d}\log\left(\frac{1}{2\gamma}\right)}{\mathrm{d}s} + \frac{\mathrm{d}\,\frac{|s|}{\gamma}}{\mathrm{d}s} = \frac{\mathrm{sign}(s)}{\gamma} \tag{14}$$

Seja $\alpha(c_1) = \mathrm{E}\left[s(n-1)f(c_1 s(n-1))\right]$. Neste caso, podemos calcular $\alpha(c_1)$ analiticamente:

$$\begin{aligned} \alpha(c_1) &= \mathrm{E}\left[s(n-1)f(c_1 s(n-1))\right] \\ &= \int_{-\infty}^{\infty} s f(c_1 s) p(s) ds \\ &= \int_{-\infty}^{\infty} s \frac{\mathrm{sign}(c_1 s)}{\gamma} \frac{1}{2\gamma} e^{-\frac{|s|}{\gamma}} ds \\ &= \frac{1}{2\gamma^2} \int_{-\infty}^{\infty} s\ \mathrm{sign}(c_1 s) e^{-\frac{|s|}{\gamma}} ds \end{aligned} \tag{15}$$

Após o cálculo da integral, chegamos a:

$$\alpha(c_1) = \mathrm{sign}(c_1) \tag{16}$$

Sendo $f(s) = \frac{\text{sign}(s)}{\gamma}$, temos que $f'(s) = 0$ (exceto na descontinuidade presente em $s = 0$; porém esta condição possui probabilidade nula de ocorrência, o que nos permite desconsiderá-la). Isso anula outros termos que contém $f'(s)$, e por isso não são citados aqui (maiores detalhes em D. B. Haddad (2009)).

6 Análise Exata da Convergência e do Desempenho

Sem utilizar a APO (ver Equação (10)), podemos calcular as médias estatísticas em questão de forma analítica, caso conheçamos a função f e a função densidade de probabilidade da fonte $s(n)$. O cálculo é muito trabalhoso, porém gera resultados precisos a respeito da convergência média.
Por exemplo, supondo distribuições exponenciais e função f ótima (vide Seção 5), para calcular $\text{E}[f(y(n))x(n)]$ cumpre desenvolver a expressão[4]:

$$\begin{aligned}\text{E}\left[f(y(n))x(n)\right] &= h_0\text{E}\left[\frac{\text{sign}(y(n))}{\gamma}s(n)\right] \\ &\quad + h_1\text{E}\left[\frac{\text{sign}(y(n))}{\gamma}s(n-1)\right],\end{aligned} \tag{17}$$

onde $\gamma \in \Re^+$ é o parâmetro da distribuição exponencial. Supondo (por simplificação) que $c_0 > 0$ e definindo $s_i = s(n-i)$, chegamos a:

$$\text{E}\left[f(y(n))x(n)\right] =$$

$$\begin{aligned}&\tfrac{h_0}{4\gamma^4}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\frac{c_1}{c_0}s(n-1)-\frac{c_2}{c_0}s(n-2)}^{\infty} g_1(\mathbf{s})ds_0ds_1ds_2 \\ &+\tfrac{h_1}{4\gamma^4}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\frac{c_1}{c_0}s(n-1)-\frac{c_2}{c_0}s(n-2)}^{\infty} g_2(\mathbf{s})ds_0ds_1ds_2,\end{aligned} \tag{18}$$

onde $g_i(\mathbf{s}) = s_i e^{-\frac{|s_0|+|s_1|+|s_2|}{\gamma}}$. O resultado final dos valores esperados apresenta uma grande extensão, o que impossibilita sua apresentação neste artigo.
A análise exata em questão é a análise da média dos parâmetros $w_i(n)$. Ela não fornece limites precisos de μ para os quais o algoritmo converge numa determinada simulação. Neste sentido, uma análise da média quadrática da diferença entre os coeficientes de cada iteração e os finais pode efetuar uma contribuição sólida. Outra vantagem da análise exata é que já não é necessária a restritiva hipótese (empregada na APO) de que o algoritmo convirja para uma solução de baixa ISI (interferência intersimbólica).

7 Simulações

7.1 Experimento 1

Seja o caso $M = L = 2$. Parte do sistema de equações da APO a ele associado pode ser visto em D. B. Haddad (2009), onde supusemos $p(s)$ exponencial com $\lambda = 0,1$, $f(s) = \frac{\text{sign}(s)}{\lambda}$ (escolha ótima). Testamos o algoritmo com 150.000 iterações *on-line*, $\mu = 5 \times 10^{-5}$, e o vetor $\mathbf{w}$ inicializado como [0 1 0].
Sejam as quatro possíveis escolhas para o filtro do canal abaixo apresentadas:

$$\begin{aligned}\mathbf{h}_1 &= [\ \ 0,19687 \quad -1,5556 \quad\ 0,087869\ ] \\ \mathbf{h}_2 &= [\ \ 0,11839 \quad -0,78388 \quad -0,18287\ ] \\ \mathbf{h}_3 &= [\ -0,37657 \quad\ 1,3414 \quad\ 0,073948\ ] \\ \mathbf{h}_4 &= [\ -0,072526 \quad 0,39283 \quad -0,0047669]\end{aligned} \tag{19}$$

A Figura 1 ilustra a aproximação dos parâmetros para os quais o algoritmo converge, revelando que a APO foi bastante razoável neste experimento (em todas as 4 escolhas, a ISI final do algoritmo é bastante reduzida, segundo a hipótese empregada na APO).

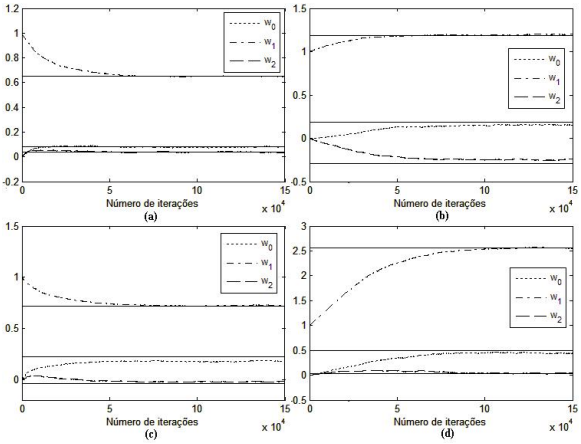

Figura 1: Evolução dos parâmetros w_0, w_1 e w_2 para os quatro casos do experimento 1. As linhas horizontais refletem o valor teórico. (a) filtro $\mathbf{h}_1$; (b) filtro $\mathbf{h}_2$; (c) filtro $\mathbf{h}_3$; e (d) filtro $\mathbf{h}_4$.

7.1.1 Experimento 2

Analisar um número maior de coeficientes dos filtros envolvidos é importante em situações práticas. Neste experimento, adotamos $M = L = 6$. Seja o filtro correspondente ao canal abaixo:

$$\begin{aligned}\mathbf{h} = [\ &-0,1841 \quad -0,0863 \quad\ 0,0870 \\ &\ 1,4511 \quad\ 0,1498 \quad -0,2256\ ]\end{aligned} \tag{20}$$

A ISI final do algoritmo no filtro de canal acima ficou abaixo de -20 dB (com um atraso $D = 6$). As escolhas da função f e da função densidade de probabilidade da fonte são idênticas à do experimento anterior, a menos do número de iterações, que foi de 500.000. A inicialização de $\mathbf{w}$ foi o vetor nulo, à exceção do quarto elemento, que é 1. Como o número de equações associado ao sistema é elevado, o método por nós proposto (e já apresentado)

[4] O cálculo a seguir apresenta uma parte do desenvolvimento necessário, a título de ilustração do procedimento.

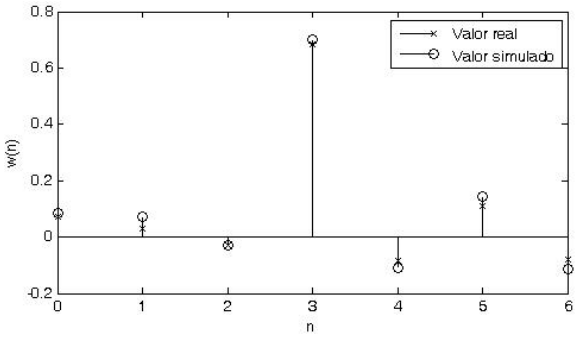

Figura 2: Comparação do filtro $w(n)$ final simulado com o teórico da APO do experimento 2.

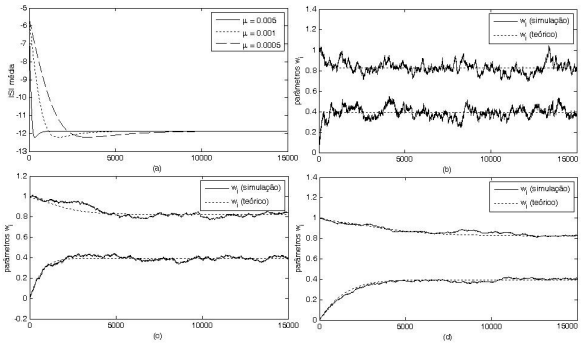

Figura 3: (a) evolução da ISI (em dB), com μ admitindo três valores: 5×10^{-3}, 10^{-3} e $0,5 \times 10^{-3}$; (b) evolução dos parâmetros w_i, com $\mu = 5 \times 10^{-3}$; (c) evolução dos parâmetros w_i, com $\mu = 10^{-3}$ e (d) evolução dos parâmetros w_i, com $\mu = 0,5 \times 10^{-3}$.

para estimar a solução do sistema foi empregado.

A Figura 2 ilustra os resultados reais e simulados. Nesta Figura, verificamos uma razoável capacidade de estimar os parâmetros da APO, a qual não encontramos na literatura pertinente. Infelizmente, os pequenos erros nas estimativas da APO dos coeficientes tendem a degradar a estimativa da ISI à medida que aumentamos M e L, o que a restringe a configurações mais simples (em nossas simulações, para L e M inferiores a 5).

7.2 Experimento 3

Analisar o comportamento médio exato de um algoritmo de deconvolução cega é o propósito deste experimento, para uma configuração $M = L = 1$. Seja $\mathbf{h} = [0,95 \quad -0,5]$ e o vetor $\mathbf{w}$ inicializado como [1 0]. Admitindo distribuições exponenciais (com $\gamma = 1$), função f ótima (vide Seção 5) e 15.000 iterações, efetuamos a análise exata. O resultado da análise teórica exata e das simulações, para três diferentes valores de μ, encontra-se na Figura 3. Esta Figura revela o esperado aumento da taxa de convergência com o aumento de μ. Os gráficos que apresentam a evolução dos coeficientes $w_i(n)$ evidenciam que o decréscimo de μ reduz a variação dos coeficientes, consequentemente incrementando o desempenho do algoritmo após a convergência (numa determinada simulação). No comportamento médio, a ISI final não se altera.

8 Conclusões

Neste artigo, duas técnicas (uma aproximada e outra exata) para análise de desempenho de uma técnica de deconvolução cega monocanal foram apresentadas. Admitindo convergência para uma solução e valores reduzidos de M e L, a abordagem aproximada (APO) revelou-se bastante acurada. Já a estimativa exata, embora mais complexa, não estima somente o desempenho final como a taxa de convergência em diferentes configurações, não precisando admitir um fator de aprendizagem tão reduzido quanto o da APO. A técnica de estimativa exata parece-nos promissora, tanto para ser estendida para as configurações multicanal quanto para estabelecer fundamentos teóricos que relacionam μ com a taxa de convergência, com o desempenho e com a estabilidade do algoritmo. Estendê-la para a convergência quadrática do algoritmo provavelmente trará grandes benefícios ao estudo teórico do algoritmo DCGN. Pretendemos explorar estes temas no futuro.

Referências

Cardoso, J.-F. (1998). Blind signal separation: Statistical principles, *Proceedings of the IEEE* **86**(10): 2009–2025.

D. B. Haddad, M. R. Petraglia, P. B. B. (2009). Análises de um algoritmo de deconvolução cega, *CBRN* . Submetido para publicação.

Minker, J. B. W. (2007). *Time-Domain Beamforming and Blind Source Separation - Speech Input in the Car Environment*, Springer.

S. Amari, S. C. Douglas, A. C. H. Y. (1997). Multichannel blind deconvolution and equalization using the natural gradient, *Proc. 1st IEEE Workshop Signal Processing Advanced Wireless Communications* pp. 101–104.

S. C. Douglas, H. Sawada, S. M. (2005). Natural gradient multichannel blind deconvolution and speech separation using causal fir filters, *IEEE Transactions on Signal Processing* **13**: 92–104.