

Equalizadores Neurais com Treinamento Baseado em Filtros de Kalman

Gabriel R. Fernandes, Antonio C. de C. Lima
Universidade Federal da Bahia
E-mails: gabrielf@ufba.br, acdcl@ufba.br

Abstract

This work presents performance and convergence comparisons among different kinds of neural equalizers and standard decision feedback equalizers. The channels investigated here are considered linear, nonlinear, fixed and time-varying added with white Gaussian noise. The digital communication system employed in all simulations is BPSK and the learning algorithms applied for training are Kalman and RTRL.

1. Introdução

Hoje em dia, a grande maioria dos sistemas de comunicação, empregados em diversas aplicações, é do tipo digital. Assim como, na comunicação analógica, os sinais contendo informações codificadas digitalmente são degradados pelo canal de comunicação, chegando ao receptor de forma distorcida e com certo nível de ruído. Esta deterioração do sinal causa um aumento na taxa de bits errados (BER) que chega ao receptor. Valores elevados da BER podem, a depender do sistema de decodificação utilizado, levar a uma redução da qualidade da informação (som, voz e imagem) ou forçar a redução da taxa de transmissão. Para minimizar os efeitos danosos, introduzidos pelos canais de comunicação, são utilizados equalizadores e sistemas de codificação robustos [1]. Os equalizadores, em particular, são utilizados para compensar as distorções, através de um processo de identificação do canal e filtragem inversa. Há várias décadas equalizadores vêm sendo utilizados em sistemas de comunicação. As arquiteturas mais conhecidas são aquelas que utilizam filtros de resposta ao impulso finita (FIR) [2]. Técnicas mais eficientes foram introduzidas em seguida, como por exemplo, os chamados equalizadores com realimentação do sinal detectado (DFE) [1][2] e os equalizadores neurais (NE) [3][4][5]. As primeiras arquiteturas de NE utilizadas foram as do tipo perceptron de multicamadas (MLP), treinadas com algoritmo de retro-propagação (BP) [3]. Em seguida vieram os NE baseados em redes completamente recorrentes (FRN), com treinamento através de aprendizado recorrente de tempo real (RTRL) [4][5]. Outros artigos têm sido reportados na literatura, incluindo aqueles que discutem sobre equalização de canais variantes no tempo [6]. Entretanto, até a presente data, não foram encontrados artigos que apresentem aplicações de MLP recorrentes (RMLP) em problemas de equalização, par-

ticularmente aquelas treinadas com algoritmos do tipo Kalman, como: filtro de Kalman estendido (EKF) e filtro de Kalman com transformação unscented (UKF) [7].

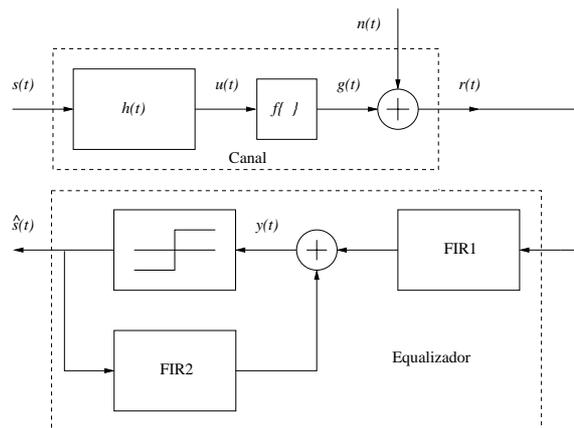


Figura 1: Sistema de comunicação com um equalizador do tipo DFE.

A Fig. 1 mostra o diagrama em blocos de um sistema de comunicação que utiliza um equalizador com realimentação do sinal detectado. A variável $s(t)$ representa um sinal binário do tipo BPSK ou PAM (-1 ou +1) que é transmitido por um canal de comunicação cuja transformada Z da função resposta ao impulso $h(t)$ é igual a

$$H(z) = h_0 + h_1z^{-1} + h_2z^{-2} + \dots + h_nz^{-n} \quad (1)$$

O sinal que chega ao receptor é fornecido por

$$r(t) = h(t) * s(t) + n(t) \quad (2)$$

No caso de canais não-lineares, o sinal que chega ao receptor é dado por

$$r(t) = f\{h(t) * s(t)\} + n(t) \quad (3)$$

sendo $f\{\cdot\}$ uma função que introduz a não-linearidade no sistema e $n(t)$ um ruído branco com distribuição gaussiana de média zero e variância igual σ^2 . O símbolo $*$ representa a operação de convolução.

Como foi dito anteriormente, os equalizadores tradicionais, com realimentação do sinal detectado, utilizam filtros do tipo FIR ou transversos e são caracterizados pela a equação de diferenças

$$y(t) = \sum_{k=0}^{M-1} a(k)r(t-k) + \sum_{k=0}^{N-1} b(k)\hat{s}(t-k) \quad (4)$$

ou na forma vetorial

$$y(t) = \mathbf{w}_a \mathbf{r}(t) + \mathbf{w}_b \hat{\mathbf{s}}(t) \quad (5)$$

sendo $\mathbf{r}(t) = [r(t) \ r(t-1) \ r(t-2) \ \dots \ r(t-M)]^T$ e $\hat{\mathbf{s}}(t) = [\hat{s}(t) \ \hat{s}(t-1) \ \hat{s}(t-2) \ \dots \ \hat{s}(t-M)]^T$, enquanto que $\mathbf{w}_a = [a_0 \ a_1 \ a_2 \ \dots \ a_{M-1}]$ e $\mathbf{w}_b = [b_0 \ b_1 \ b_2 \ \dots \ b_{N-1}]$. A estimativa do sinal transmitido é então obtida classificando-se os valores $y(t)$ na saída do equalizador através da função

$$\hat{s}(t) = \begin{cases} +1, & \text{para } y(t) > 0 \\ -1, & \text{para } y(t) < 0 \end{cases} \quad (6)$$

Os coeficientes dos filtros que compõe o equalizador são obtidos tradicionalmente transmitindo-se uma seqüência de treinamento e empregando-se um dos algoritmos do tipo gradiente, como por exemplo, LMS e RLS [2]. A função custo utilizada neste caso é o erro quadrático, dado por

$$J = e^2(t) = [d(t) - \hat{s}(t)]^2 \quad (7)$$

onde $d(t)$ representa os símbolos conhecidos, utilizados durante o processo de estimação dos coeficientes do filtro.

2. Neuroequalizadores

Os neuroequalizadores analisados aqui podem ser associados a três tipos de redes neurais artificiais: redes completamente recorrentes (FRN), redes do tipo Elman e redes recorrente do tipo multicamada (RMLP) com duas camadas escondidas. A Fig. 2 mostra cada uma das arquiteturas destes equalizadores. A descrição matemática para cada modelo é apresentada a seguir.

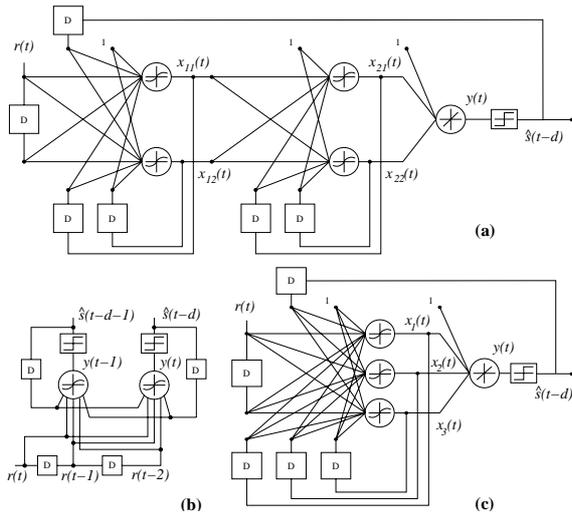


Figura 2: Neuroequalizadores do tipo:(a) RMLP, (b) FRN e (c) Elman.

- Equalizadores baseado em rede do tipo FRN:

$$\mathbf{x}(t) = \varphi\{\mathbf{W}_1[\mathbf{r}(t)^T \ \mathbf{x}(t-1)^T \ \hat{\mathbf{s}}(t-1)^T \ 1]^T\} \quad (8)$$

sendo a saída dada por

$$y(t) = x_0(t) \quad (9)$$

- Equalizadores baseado rede do tipo Elman:

$$\mathbf{x}(t) = \varphi\{\mathbf{W}_1[\mathbf{r}(t)^T \ \mathbf{x}(t-1)^T \ \hat{\mathbf{s}}(t-1)^T \ 1]^T\} \quad (10)$$

sendo a saída fornecida por

$$y(t) = \mathbf{W}_2[\mathbf{x}(t)^T \ 1]^T \quad (11)$$

- Equalizadores baseado em rede do tipo RMLP:

$$\mathbf{x}_1(t) = \varphi\{\mathbf{W}_1[\mathbf{r}(t)^T \ \mathbf{x}_1(t-1)^T \ \hat{\mathbf{s}}(t-1)^T \ 1]^T\} \quad (12)$$

e

$$\mathbf{x}_2(t) = \varphi\{\mathbf{W}_2[\mathbf{x}_1(t)^T \ \mathbf{x}_2(t-1)^T \ 1]^T\} \quad (13)$$

sendo a saída

$$y(t) = \mathbf{W}_3[\mathbf{x}_2(t)^T \ 1]^T \quad (14)$$

Em todas as arquiteturas, as entradas de valor unitário estão associadas a polarização (*bias*) dos neurônios, enquanto \mathbf{W}_1 , \mathbf{W}_2 e \mathbf{W}_3 são as matrizes dos pesos sinápticos. Os vetores $\mathbf{x}_i(t) = [x_{i0}(t) \ x_{i1}(t) \ x_{i2}(t) \ \dots]^T$ representam as saídas dos i -ésimas camadas da redes neurais, $y(t)$ a saída de cada rede e $\varphi\{\}$ as funções de ativação que, no caso deste artigo, são todas tangentes hiperbólicas.

Em geral, os equalizadores baseados em redes do tipo FRN são treinados através do algoritmo RTRL, sua descrição é mostrada em [8].

3. Filtros de Kalman

O treinamento dos neuroequalizadores, baseados em redes do tipo Elman e RMLP, podem ser realizados através de algoritmos para estimação de parâmetros que utilizam o conceito de filtragem de Kalman. É de conhecimento geral que estes algoritmos são baseados nas equações de processo e medição. No caso de algoritmos aplicados à estimação de parâmetros, como por exemplo na determinação dos pesos sinápticos de uma rede neural, essas equações são, respectivamente, escritas como [8]

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \omega(t) \quad (15)$$

$$y(t) = \mathbf{h}\{\mathbf{w}(t), \mathbf{r}(t), \mathbf{x}(t-1)\} + \nu(t) \quad (16)$$

onde $\omega(t)$ é denominado de erro de processo e $\nu(t)$ de erro de medição. O vetor $\mathbf{w}(t)$ contém os elementos das matrizes apresentadas nas equações (10) e (11) para o caso de redes Elman ou (12), (13) e (14) para caso de redes RMLP. Enquanto o vetor \mathbf{h} representa as saídas do modelo neural antes da adição do ruído $\nu(t)$.

3.1. Filtro de Kalman Estendido (EKF)

Como os neuroequalizadores são modelos não-lineares, então, para se aplicar o filtro de Kalman torna-se necessário a linearização de \mathbf{h} utilizando-se um aproximação de Taylor de primeira ordem, o que dá origem à jacobiana

$$\mathbf{H}(t) = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{w}} \right|_{\mathbf{w}=\hat{\mathbf{w}}(t)} \quad (17)$$

Aplicando-se esta jacobiana no algoritmo filtro de Kalman tradicional obtém-se a versão estendida (EKF) para modelos não-lineares, ou seja,

$$\mathbf{A}(t) = [\mathbf{R}(t) + \mathbf{H}^T(t)\mathbf{P}(t)\mathbf{H}(t)]^{-1} \quad (18)$$

$$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}(t)\mathbf{A}(t) \quad (19)$$

$$\hat{\mathbf{w}}(t+1) = \hat{\mathbf{w}}(t) + \mathbf{K}(t)\mathbf{e}(t) \quad (20)$$

$$\mathbf{P}(t+1) = \mathbf{P}(t) - \mathbf{K}(t)\mathbf{H}^T(t)\mathbf{P}(t) + \mathbf{Q}(t) \quad (21)$$

sendo $\mathbf{e}(t) = \mathbf{d}(t) - \mathbf{y}(t)$ o vetor erro, $\mathbf{R}(t) = E[\omega(t)\omega(t)^T]$ a matriz covariância do ruído de medição, $\mathbf{Q}(t) = E[\nu(t)\nu(t)^T]$ a matriz covariância do ruído de processo e $\mathbf{P}(t) = E[\mathbf{w}(t)\mathbf{w}(t)^T]$ a matriz covariância dos parâmetros. Essas matrizes são inicializadas como segue:

$$\mathbf{R}(0) = \eta^{-1}\mathbf{I} \quad (22)$$

$$\mathbf{Q}(0) = q\mathbf{I} \quad (23)$$

$$\mathbf{P}(0) = \epsilon^{-1}\mathbf{I} \quad (24)$$

onde, neste artigo, \mathbf{I} é a matriz identidade, $\epsilon = q = 0, 01$ e $\eta = 0, 1$.

3.2. Filtro de Kalman Unscented (UKF)

O cálculo da jacobiana \mathbf{H} pode ser suprimido do algoritmo de Kalman e ainda se obter uma aproximação de pelo menos primeira ordem, através do uso da transformação *unscented* (UT). Com esta abordagem é possível se obter as estatísticas das variáveis aleatórias de saída de um processo não-linear.

Considerando-se que vetor pesos sinápticos \mathbf{w} ($L \times 1$) constitui as variáveis aleatórias que se quer aplicar a transformação *unscented* e que a saída do neuroequalizador é representada por $y = f\{\mathbf{w}\}$, então, utilizando-se a média $\bar{\mathbf{w}}$ e a matriz covariância $\mathbf{P}_{\mathbf{w}}$, pode-se obter as estatísticas de y através de $2L + 1$ valores representativos de \mathbf{w} . Este valores, conhecidos como pontos sigma, podem ser agrupados numa matriz χ , tal que

$$\chi = [\chi_0 \ \chi_i^+ \ \chi_i^-] \quad (25)$$

sendo

$$\chi_0 = \bar{\mathbf{w}} \quad (26)$$

$$\chi_i^+ = \bar{\mathbf{w}} + (\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{w}}})_i, \quad i = 1, \dots, L \quad (27)$$

$$\chi_i^- = \bar{\mathbf{w}} - (\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{w}}})_{i-L}, \quad i = L+1, \dots, 2L \quad (28)$$

onde $\lambda = \alpha^2(L+\kappa) - L$ é um fator de ajuste e α uma constante que determina o espalhamento dos pontos sigma em torno da média $\bar{\mathbf{w}}$. Em geral o valor atribuído a α é pequeno e positivo, na faixa $0, 001 < \alpha < 1$. A constante κ é um segundo fator de ajuste cujo valor é igual a $3 - L$.

Quando os pontos sigma são introduzidos como entrada num sistema não-linear, que no nosso caso são os neuroequalizadores, os valores de saída que se obtém são representados por

$$\mathcal{Y}_i = f(\chi_i) \quad (29)$$

A média e a covariância da variável de saída podem ser obtidas, respectivamente, através das médias ponderadas [14]

$$\bar{y} \approx \sum_{i=0}^{2L} W_i^m \mathcal{Y}_i \quad (30)$$

$$\mathbf{P}_y \approx \sum_{i=0}^{2L} W_i^c (\mathcal{Y}_i - \bar{y})(\mathcal{Y}_i - \bar{y})^T \quad (31)$$

onde os pesos são fornecidos por

$$W_0^m = \frac{\lambda}{L + \lambda}$$

$$W_0^c = \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta) \quad (32)$$

$$W_i^m = W_i^c = \frac{1}{2(L + \lambda)}, \quad i = 1, 2, \dots, 2M$$

Os índices m and c estão relacionados com a média e covariância, respectivamente. A constante β é utilizada levando-se em consideração conhecimentos a priori da distribuição de \mathbf{w} . No caso de distribuições gaussianas o valor ótimo de β é 2. Maiores detalhes sobre a transformação *unscented* pode ser obtida na referência [7].

O algoritmo UKF para a estimação de parâmetros é descrito a seguir:

- Inicializa-se os pesos sinápticos e a matriz covariância

$$\hat{\mathbf{w}}(0) = E[\mathbf{w}] \quad (33)$$

$$\mathbf{P}(0) = E[(\mathbf{w} - \hat{\mathbf{w}}(0))(\mathbf{w} - \hat{\mathbf{w}}(0))^T]. \quad (34)$$

- Calcula-se, para $t = 1, 2, \dots$, os pontos sigma

$$\mathbf{\Gamma}(t) = (L + \lambda)(\mathbf{P}(t) + \mathbf{Q}(t)) \quad (35)$$

$$\mathcal{W}(t) = \left[\hat{\mathbf{w}}(t) \ \hat{\mathbf{w}}(t) + \sqrt{\mathbf{\Gamma}(t)} \ \hat{\mathbf{w}}(t) - \sqrt{\mathbf{\Gamma}(t)} \right] \quad (36)$$

$$\mathcal{D}(t) = \mathbf{h} \{ \mathcal{W}(t), \mathbf{r}(t), \hat{\mathbf{s}}(t), \mathbf{x}(t-1) \} \quad (37)$$

$$\mathbf{y}(t) = \mathbf{h} \{ \hat{\mathbf{w}}(t), \mathbf{r}(t), \hat{\mathbf{s}}(t), \mathbf{x}(t-1) \} \quad \text{ou} \quad (38)$$

$$\sum_{i=0}^{2L} W_i^m \mathcal{D}_i(t)$$

- Atualiza-se as matrizes covariância e os pesos sinápticos

$$\mathbf{P}_{yy}(t) = \sum_{i=0}^{2L} W_i^c [\mathcal{D}_i(t) - \hat{y}(t)][\mathcal{D}_i(t) - \hat{y}(t)]^T$$

$$\begin{aligned}
& +\mathbf{R}(t) \quad (39) \\
\mathbf{P}_{\mathbf{w}_y}(t) &= \sum_{i=0}^{2L} W_i^c [\mathcal{W}_i(t) - \hat{\mathbf{w}}(t)] [\mathcal{W}_i(t) - \hat{\mathbf{w}}(t)]^T \quad (40)
\end{aligned}$$

$$\mathbf{\Upsilon}(t) = \mathbf{P}_{\mathbf{w}_y}(t)\mathbf{P}_{\mathbf{y}\mathbf{y}}^{-1}(t) \quad (41)$$

$$\hat{\mathbf{w}}(t+1) = \hat{\mathbf{w}}(t) + \mathbf{\Upsilon}(t)\mathbf{e}(t) \quad (42)$$

$$\mathbf{P}(t+1) = \mathbf{P}(t) - \mathbf{\Upsilon}(t)\mathbf{P}_{\mathbf{y}\mathbf{y}}(t)\mathbf{\Upsilon}^T(t) \quad (43)$$

Lembrando-se que $\mathbf{R}(t)$ e $\mathbf{Q}(t)$ são inicializadas de acordo com (22) e (23), respectivamente.

4. Simulações e Resultados

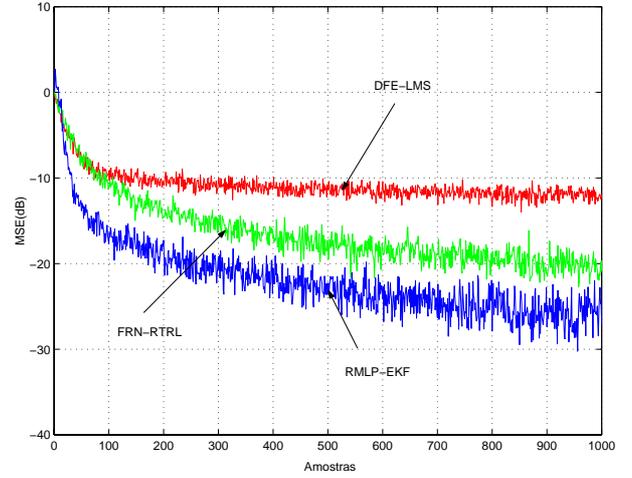
Os resultados apresentados nesta seção estão divididos em grupos de acordo com o tipo de canal de comunicação. Todos os equalizadores analisados aqui utilizam, de uma forma ou de outra, realimentação do sinal detectado e seus coeficientes foram obtidos a partir de treinamento on-line. As discussões sobre resultados são baseadas nas curvas de convergência e de taxa de bits errados (BER). Primeiro serão mostradas comparações entre os neuroequalizadores e os DFEs tradicionais, aplicados a canais lineares. O segundo grupo de resultados envolve a aplicação de equalizadores em canais não-lineares e, por fim, no último grupo são comparados os desempenhos dos equalizadores em canais variantes no tempo.

4.1. Canais lineares

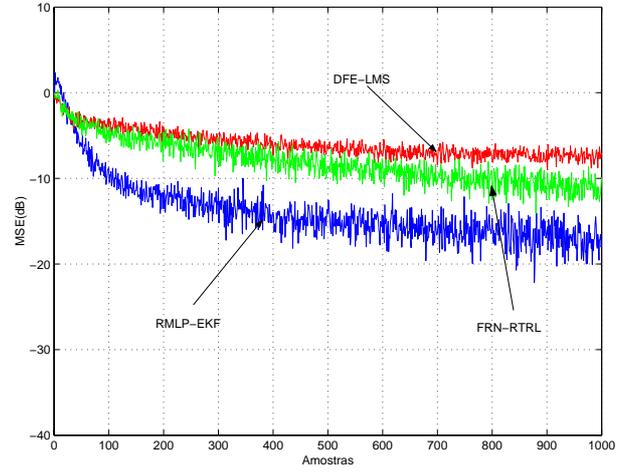
Dois tipos de canais lineares foram considerados nas simulações. Suas funções de transferências são $H_1(z) = 0,407 + 0,815z^{-1} + 0,407z^{-2}$ e $H_2(z) = 0,227 + 0,46z^{-1} + 0,688z^{-2} + 0,46z^{-3} + 0,227z^{-4}$. Ambas representam canais telefônicos típicos, sendo que uma análise do espectro associado a segunda revela um nulo no meio da banda de passagem, isto faz com que o canal 2 seja de difícil equalização.

A Fig. 3a mostra uma comparação de convergência para três equalizadores, quando aplicados num sistema de comunicação que utiliza o canal 1. Um deles é o DFE tradicional com treinamento baseado no algoritmo LMS e os outros dois são os neuroequalizadores FRN utilizando RTRL e Elman com treinamento baseado no EKF. Enquanto que a Fig. 3b mostra a mesma comparação para o canal 2. Pode-se observar em ambas as figuras que a arquitetura Elman tem o melhor desempenho em termos de convergência. Na Fig. 3b, fica claro também que o canal 2 é de difícil equalização. Nestas simulações foram consideradas uma relação sinal-ruído de 16dB. Todas as curvas representam a média de 100 experimentos independentes. O equalizador DFE considerado aqui tem três entradas ($M = 3$) e quatro realimentações ($N = 4$), o neuroequalizador FRN tem três entradas e 4 neurônios, enquanto que o Elman tem três entradas, quatro realimentações e uma camada escondida com 3 neurônios recorrentes. O instante de decisão dos equalizadores tem atraso de duas amostras. Em

todas as simulações foram utilizados, durante o processo de aprendizagem on-line, os seguintes parâmetros: taxa de aprendizagem de 0,1 para os algoritmos LMS e RTRL; $\eta = 0,1$ e $\delta = 0,01$ para o algoritmo EKF.



(a)



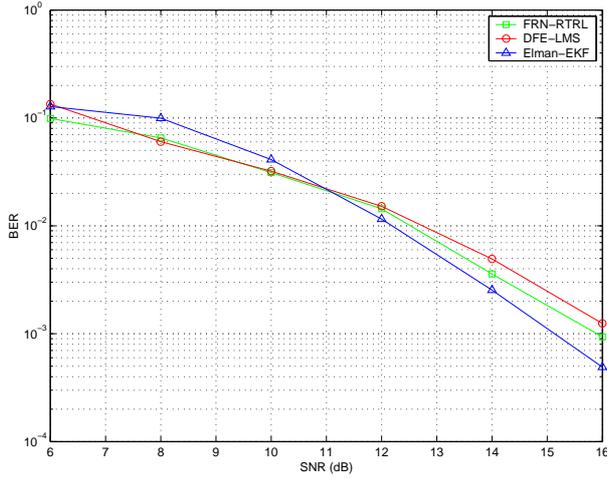
(b)

Figura 3: Comparação de convergência para diferentes arquiteturas: (a) canal 1 e (b) canal 2.

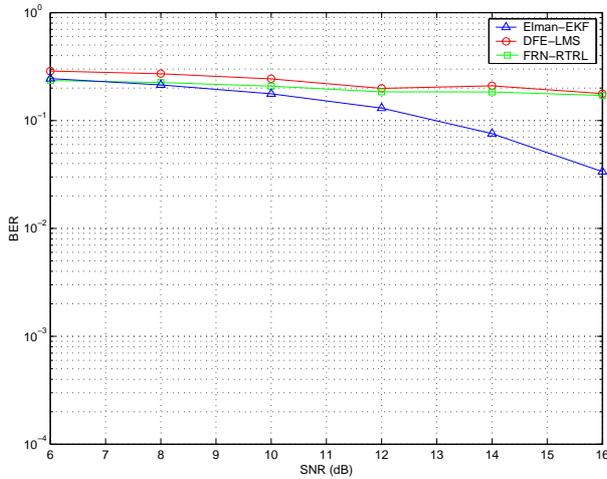
As Fig. 4a e Fig. 4b mostram os desempenhos dos equalizadores com relação a BER. Observa-se mais uma vez a performance superior do neuroequalizador baseado na estrutura Elman. Para uma relação sinal-ruído de 16dB a diferença na taxa de erro entre a arquitetura Elman e FRN é expressiva no caso do canal 2. Nota-se também que o desempenho do neuroequalizador FRN em frente aos DFE tradicionais é praticamente o mesmo. Todas as curvas são resultado da média de 20 experimentos independentes. O treinamento on-line foi efetuado com uma sequência de 100 símbolos para o canal 1 e 200 símbolos para o canal 2. O número de símbolos utilizados na medição das BERs foi de 10000.

4.2. Canais não-lineares

Para analisar a performance dos neuroequalizadores na presença de canais de comunicações não-lineares foi



(a)



(b)

Figura 4: Comparação em função da BER para diferentes arquiteturas: (a) canal 1 e (b) canal 2.

escolhido o mesmo canal adotado em [5], onde o sinal recebido é fornecido por $r(t) = u(t) + 0,2u^2(t)$, sendo $u(t) = h_3(t) \star s(t)$ e $H_3(z) = 0,3482 + 0,8704z^{-1} + 0,3482z^{-2}$.

A Fig. 5 mostra uma comparação de desempenho entre os neuroequalizadores FRN e Elman treinados com os algoritmos RTRL (no caso do FRN), EKF e UKF. Nota-se que os equalizadores baseados na arquitetura Elman tem desempenho superior ao FRN, sendo que aquele onde os coeficientes foram obtidos com UKF tem a melhor performance.

O neuroequalizador FRN utilizado neste experimento tem as mesmas características daquele utilizado no experimento anterior. A estrutura Elman neste caso tem três entradas, duas realimentações e dois neurônios recorrentes na camada escondida. Todas as curvas apresentadas na Fig. 5 são resultado da média de 20 experimentos independentes. O treinamento on-line foi efetuado com uma seqüência de 100 símbolos e 10000 foram empregados na determinação das taxas de bits errados.

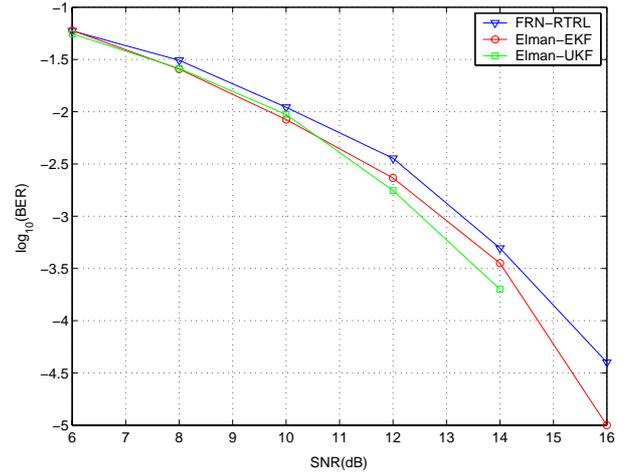


Figura 5: Desempenho em função da BER para diferentes arquiteturas.

4.3. Canais variantes no tempo

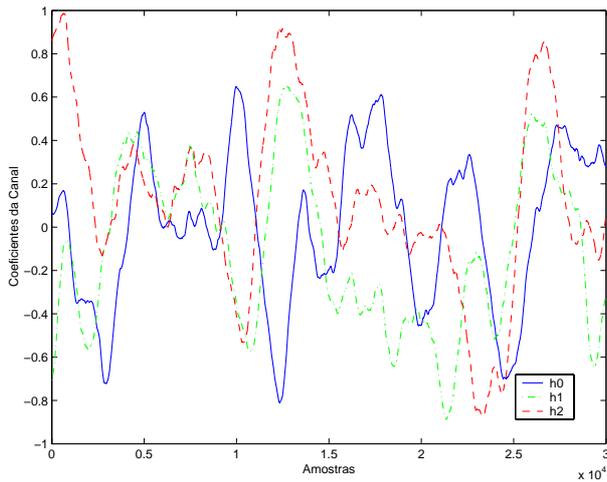
As simulações para canais variantes no tempo foram executadas utilizando um modelo de canal baseado num processo markoviano de 2ª ordem. Este modelo foi obtido excitando-se um filtro Butterworth de segunda ordem com ruído branco gaussiano de média zero e variância 1. O resultado é uma função resposta ao impulso com coeficientes que variam aleatoriamente no tempo, conforme a função de transferência mostrada a seguir:

$$H(z) = h_0(t) + h_1(t)z^{-1} + h_2(t)z^{-2} \quad (44)$$

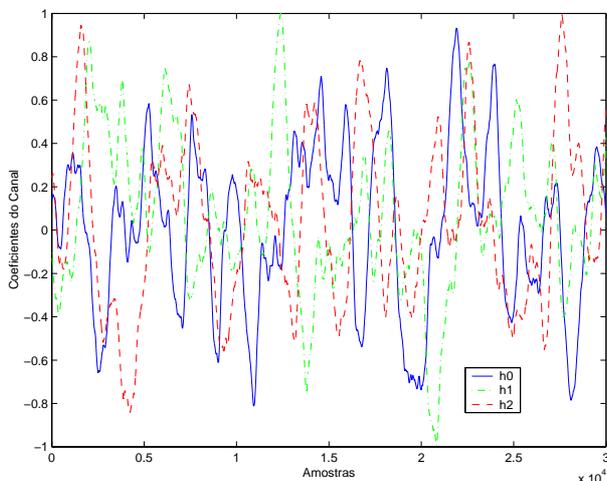
A Fig.6 mostra a variação dos coeficientes da função de transferência, eq. (44), para a condição de *fading* rápido ($f_c = 0,5\text{Hz}$) e muito rápido ($f_c = 1\text{Hz}$), onde f_c é a frequência de corte do filtro Butterworth. Já a Fig. 7 apresenta o comportamento dos neuroequalizadores para diferentes taxas de *fading*. Observa-se que o equalizador baseado em redes completamente recorrentes (FRN) tem um péssimo desempenho em ambas as situações, enquanto que o equalizador do tipo Elman, treinados com EKF e UKF, atinge valores razoáveis de BER, particularmente quando treinado com UKF. O neuroequalizadores utilizados neste experimento tem as mesmas características daqueles utilizados no experimento anterior. Todos as curvas apresentadas na Fig. 7 são resultado de 100 experimentos independentes, onde 100 símbolos foram utilizados como seqüência de treinamento e 1000 símbolos para medição das taxas de erros.

5. Conclusões

Pelos resultados apresentados neste trabalho, pode-se afirmar que os equalizadores baseados em redes neurais do tipo Elman, com coeficientes obtidos a partir de treinamento on-line utilizando-se algoritmos do tipo



(a)



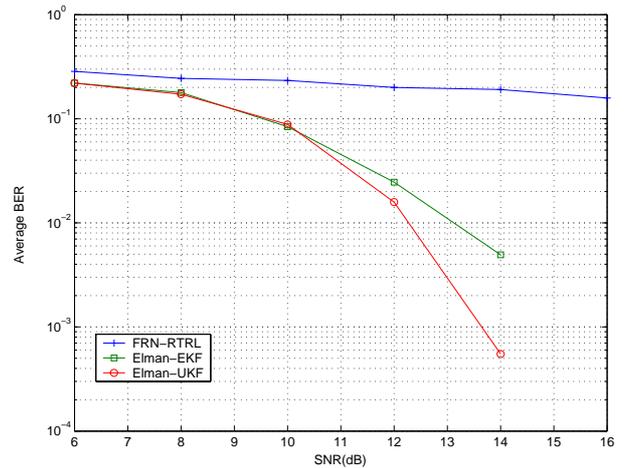
(b)

Figura 6: Variação dos coeficientes do canal de comunicação em função do tempo para *fading* (a) rápido (0,5Hz). e (b) muito rápido (1Hz)

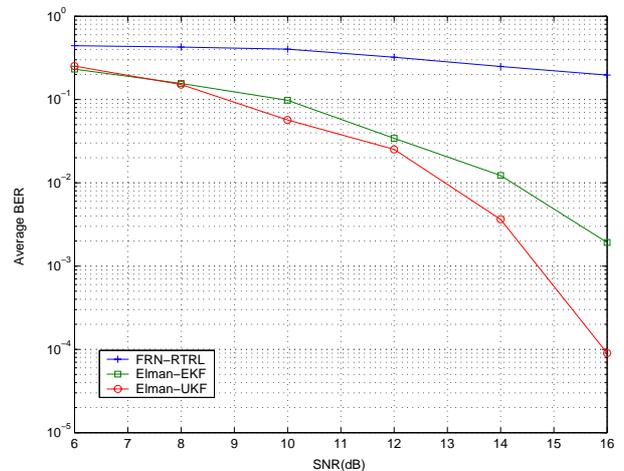
EKF e UKF, apresentam um melhor desempenho em convergência e na taxa de bits errados em todos os tipos de canais estudados aqui. A convergência rápida, introduzida pelos filtros de Kalman, possibilita a realização de treinamento ou estimação dos coeficientes dos neuroequalizadores através de seqüências de comprimentos curtos. Esta característica é extremamente importante em sistemas de comunicação sem fio, onde as condições do canal podem variar rapidamente ao longo do tempo. O desempenho dos neuroequalizadores treinados com o algoritmo UKF é superior a todos os outros, por estimar de maneira mais eficiente as estatísticas dos sinais de saída destes equalizadores. A desvantagem deste algoritmo está na sua complexidade e no longo tempo de processamento quando comparados com as outras duas técnicas apresentadas.

Referências

[1] J. G. Proakis. *Digital Communications*. McGraw-Hill, New York, 2001.



(a)



(b)

Figura 7: Desempenho em função da BER para diferentes arquiteturas e taxas de . (a) *fading* rápido ($f_c = 0,5\text{Hz}$) e (b) *fading* muito rápido ($f_c = 1\text{Hz}$).

- [2] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, New Jersey, 2001.
- [3] G. J. Gibson, S. Siu, and C. F. N. Cowan. Application of multilayer perceptron as adaptive channel equalizer. In *Proceedings of of IEEE ICASSP, Glasgow, U.K.*, pages 1183–1186, 1989.
- [4] S. Ong, C. You, and D. Hong. A decision feedback recurrent neural equalizer as an infinite impulse response filter. *IEEE Trans. on Signal Processing*, 45(11):2851–2858, 1997.
- [5] R. Parisi, E. D. D. Claudio, G. Orlandi, and B. D. Rao. Fast adaptive digital equalization by recurrent neural networks. *IEEE Trans. on Signal Processing*, 45(11):2731–2739, 1997.
- [6] A. Zerguine, A. Shafi, and M. Bettayeb. Multilayer perceptron-based dfe with lattice structure. *IEEE Trans. on Neural Networks*, 12:532–545, 2001.
- [7] S. Haykin. *Kalman Filtering and Neural Network*. John Wiley & Sons, New York, 2001.
- [8] S. Haykin. *Neural Network: A Comprehensive Foundation*. Prentice-Hall, New Jersey, 1998.