

# ESTRUTURAS NÃO-LINEARES APLICADAS À RECUPERAÇÃO DE SINAIS BINÁRIOS

Magno T. Madeira da Silva<sup>1\*</sup>, Max Gerken<sup>2\*</sup>, Maria D. Miranda<sup>3</sup>

(1,2) Departamento de Engenharia de Telecomunicações e Controle  
Escola Politécnica da Universidade de São Paulo

<<http://www.lcs.poli.usp.br>>

{magno, mgk}@lcs.poli.usp.br

(3) Universidade Presbiteriana Mackenzie, São Paulo, SP  
Curso de Pós-Graduação em Engenharia Elétrica

## RESUMO

Neste artigo é feita uma comparação do desempenho de quatro estruturas para recuperação de sinais binários: um filtro linear transversal e três diferentes tipos de redes neurais. São consideradas implementações com e sem realimentação de decisões já tomadas. Em geral, a introdução da realimentação de decisões resulta numa melhora de desempenho ou numa diminuição de complexidade para uma mesma taxa de erro. É apresentado um exemplo que explica este efeito, baseado na interpretação do problema de equalização como um problema de classificação.

## 1. INTRODUÇÃO

A equalização adaptativa linear é uma técnica largamente empregada em comunicações digitais com o objetivo de minorar os efeitos indesejáveis da interferência intersimbólica e do ruído no canal de transmissão. Entretanto, em muitas situações realistas, por exemplo quando os canais não são de mínima fase e/ou apresentam não-linearidades, o equalizador linear nem sempre é uma solução ótima. Este é o caso da comunicação por satélite [1], [5]. Diante disso, estruturas não-lineares vêm sendo amplamente estudadas [1]-[5] com o objetivo de serem empregadas em equalização. Exemplos são as configurações baseadas no filtro transversal com realimentação de decisões (*Decision Feedback Equalizer* - DFE), as baseadas em séries de Volterra e as que usam redes neurais.

As redes neurais, apesar de serem computacionalmente complexas, apresentam vantagens sobre outras estruturas não-lineares como, por exemplo, as baseadas em séries de Volterra. Isso se deve ao alto grau de paralelismo das redes neurais que as tornam atraentes para implementação em circuitos integrados [1]. Configurações utilizando realimentação de decisões têm sido recentemente combinadas com essas redes com o objetivo de reduzir a interferência intersimbólica, procurando melhorar o desempenho do equalizador.

Uma comparação teórica entre diferentes estruturas não-lineares não é muito evidente devido às dificuldades inerentes ao problema. Desta forma, foram feitas simulações visando comparar o desempenho de algumas dessas estruturas. As configurações escolhidas foram o filtro transversal, a rede de perceptrons multicamada (*Multilayer Perceptrons* - MLP), a rede neural utilizando funções

de base radial (*Radial Basis Functions* - RBF) e uma rede neural recorrente (*Recurrent Neural Network* - RNN). Todas estas estruturas, exceto a última, foram também testadas com realimentação de decisões (*Decision Feedback* - DF).

No texto a seguir são descritas as configurações utilizadas e as respectivas complexidades computacionais. São também apresentadas curvas de taxa de erro de bits em função da relação sinal-ruído sendo feita uma comparação de desempenhos. No final, é apresentada uma explicação para a melhora de desempenho obtida com o uso de realimentação de decisões.

## 2. ESTRUTURAS UTILIZADAS

As simulações das estruturas seguem o esquema básico de equalização da Fig. 1, onde  $H(z)$  é a transformada-Z da resposta ao pulso unitário finita (FIR) do modelo do canal,  $n(k)$  representa o ruído branco gaussiano adicionado na saída do canal e  $\hat{y}(k)$  é a seqüência de observação na entrada do receptor. Os inteiros  $M$  e  $\tau_d$  representam o número de entradas e o atraso do equalizador respectivamente. O problema a ser considerado é como usar a seqüência  $\hat{y}(k)$ ,  $\hat{y}(k-1)$ , ...,  $\hat{y}(k-M+1)$  para estimar no instante  $k$  o símbolo transmitido  $x(k-\tau_d)$ . Na fase de treinamento o sinal de entrada atrasado  $x(k-\tau_d)$  é usado como sinal de referência. Obtém-se dessa forma o sinal de erro  $e(k)$  que é utilizado pelo algoritmo de treinamento para atualização dos parâmetros do equalizador. Após o treinamento, a própria estimativa  $\hat{x}(k-\tau_d)$  é utilizada para o cálculo do erro [6].

As estruturas com realimentação de decisões seguem o esquema

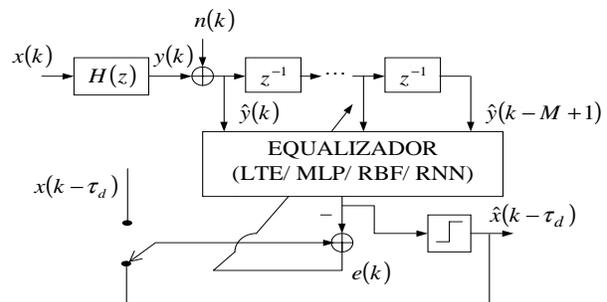


Fig. 1. Esquema básico empregado para equalização adaptativa.

\* Este trabalho foi financiado pela FAPESP<sup>1</sup> (proc. 99/00188-0) e CNPq<sup>2</sup> (proc. 300521/92-8).

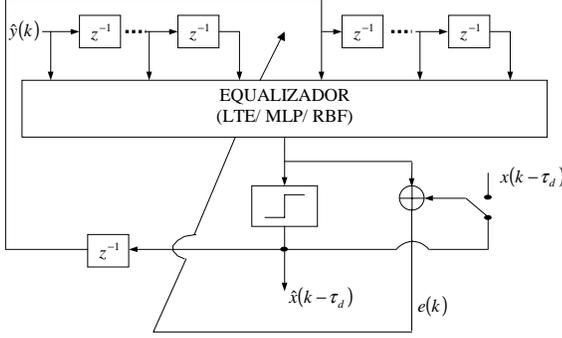


Fig. 2. Esquema empregado para equalização adaptativa com DF.

da Fig. 2. Neste caso, a entrada do equalizador é constituída de  $M_f$  amostras da seqüência  $\hat{y}(k)$  e  $M_b$  amostras das decisões já tomadas  $\hat{x}(k)$ .

### 2.1. O Equalizador Linear Transversal (LTE)

No caso do Equalizador Linear Transversal (*Linear Transversal Equalizer* - LTE), o problema de equalização consiste em aplicar um algoritmo adaptativo para estimar a cada instante  $k$  o símbolo transmitido  $x(k - \tau_d)$  [6] como

$$\hat{x}(k - \tau_d) = \text{sgn} \left( \sum_{j=0}^{M-1} w_j \hat{y}(k - j) \right),$$

sendo  $w_j$  os pesos a serem adaptados pelo algoritmo e  $\text{sgn}(x) = -1$  para  $x \leq 0$  e  $\text{sgn}(x) = 1$  para  $x > 0$ .

É comum utilizar um algoritmo adaptativo do tipo gradiente estocástico, ou dos mínimos quadrados recursivo (*Recursive Least Squares* - RLS), ou ainda variações destes. Neste trabalho, o algoritmo usado na implementação do LTE foi o da família LSL (*Least Squares Lattice*) *a priori* com realimentação de erro [7] (EF-LSL). Este algoritmo foi usado porque apresenta um bom compromisso entre estabilidade e complexidade numéricas [7]. A implementação deste algoritmo exige a execução de  $13M$  multiplicações,  $2M$  divisões e  $9M$  somas, sendo  $M$  o número de parâmetros ajustáveis do filtro adaptativo. Mais detalhes podem ser encontrados em [6] e [7].

### 2.2. A rede MLP (*Multilayer perceptrons*)

Tipicamente uma rede MLP consiste de um conjunto de nós (neurônios) que estão dispostos em uma camada de entrada, em uma ou mais camadas ocultas e uma camada de saída [8]. A configuração de uma rede com  $L$  camadas é indicada por  $(M_1, M_2, \dots, M_L)$ , sendo  $M_l$  o número de nós da camada  $l$ .

O sinal de entrada se propaga pela rede progressivamente, camada por camada. Em cada nó  $j$  da camada  $l$ , é calculada a soma ponderada das entradas  $z_i^{(l-1)}$  adicionada a um nível de ajuste  $\theta_j$  (*bias*), obtendo-se o nível de atividade interna  $v_j^{(l)} = \sum_{i=1}^{M_{l-1}} (w_{ji}^{(l)} z_i^{(l-1)} - \theta_j)$ . Nesta expressão o peso  $w_{ji}^{(l)}$  representa o ganho da conexão entre os neurônios  $i$  da camada  $(l-1)$  e  $j$  da camada  $(l)$ . Este resultado é aplicado a uma função de ativação  $\varphi(\cdot)$  que usualmente é uma função sigmoideal definida como

$$\varphi(v) = (1 - e^{-cv}) / (1 + e^{-cv}), \quad (1)$$

sendo  $c$  uma constante inteira positiva. Neste trabalho, utilizou-se a função sigmoideal com  $c = 2$ . Obtém-se assim a saída  $z_j^{(l)} =$

$\varphi(v_j^{(l)})$  do nó  $j$  da camada  $l$ . Este cálculo é repetido para todos os nós, camada por camada.

A rede MLP é treinada com o algoritmo *Backpropagation* (BP) que é baseado no algoritmo do gradiente estocástico [8]. A complexidade computacional deste algoritmo na fase de treinamento é mostrada na Tabela I. Nesta Tabela,  $M_0$  representa o número de entradas da rede. A operação NL indica o número de vezes em que se utiliza a função não-linear incluindo também a utilização de sua derivada que é usada no algoritmo de treinamento.

Operações	Nº de operações por iteração
×	$M_A + 5M_B$
NL	$2M_B$
+	$M_A + M_B + 2M_L$

Tabela I

COMPLEXIDADE COMPUTACIONAL DO ALGORITMO BP NA FASE DE TREINAMENTO DE UMA REDE MLP  $(M_1, M_2, \dots, M_L)$ , SENDO  $M_A = 3M_0M_1 + 4(M_1M_2 + M_2M_3 + \dots + M_{L-1}M_L)$ ,  $M_B = M_1 + M_2 + \dots + M_L$ .

### 2.3. A rede RBF (*Radial Basis Function*)

A rede RBF consiste usualmente de três camadas de neurônios sendo que a primeira alimenta a segunda diretamente (pesos iguais a 1) e a camada de saída é apenas um combinador linear [8], [4]. A camada oculta implementa um mapeamento não-linear que é calculado em dois passos. No primeiro passo, o sinal de entrada  $\hat{\mathbf{y}}(k) = [\hat{y}(k) \hat{y}(k-1) \dots \hat{y}(k-M+1)]^T$  é comparado com um conjunto de vetores de referência  $\mathbf{r}_i(k)$  para  $i = 0, 1, \dots, M-1$ , sendo  $M$  o número de nós ocultos. Estes vetores são chamados de centros. A comparação entre o sinal de entrada e os centros pode ser feita através da norma Euclidiana gerando um conjunto de distâncias  $s_i(k) = \|\hat{\mathbf{y}}(k) - \mathbf{r}_i(k)\|$ ,  $i = 0, 1, \dots, M-1$ . Estas distâncias são então aplicadas a uma função não-linear com simetria radial que normalmente é uma gaussiana:  $f(s_i(k)) = \exp(-s_i^2(k)/\sigma_i^2(k))$ . O parâmetro  $\sigma_i(k)$  controla o espalhamento da função e consequentemente o seu raio de influência [4]. O sinal de saída é calculado como  $F(\hat{\mathbf{y}}(k)) = \varphi[\mathbf{w}^T(k)f(\mathbf{s}(k))]$ , sendo  $\varphi(\cdot)$  a função de ativação, usualmente sigmoideal. A atualização dos pesos foi feita usando um algoritmo do tipo gradiente estocástico (LMS) [8]. A complexidade computacional deste algoritmo no treinamento de uma rede RBF com uma saída é mostrada na Tabela II.

Operações	Nº de operações por iteração
×	$M_0M_1 + 3M_1 + 4$
NL	$M_1 + 1$
+	$2M_0M_1 + 2M_1 + 3$

Tabela II

COMPLEXIDADE COMPUTACIONAL NA FASE DE TREINAMENTO DE UMA REDE RBF COM UMA SAÍDA, CONSIDERANDO A ATUALIZAÇÃO DOS PESOS SEGUNDO O LMS, SENDO  $M_0$  O NÚMERO DE ENTRADAS DA REDE E  $M_1$  O NÚMERO DE NEURÔNIOS DA CAMADA OCULTA.

### 2.4. A rede RNN (*Recurrent Neural Network*)

Uma RNN de  $M$  entradas externas tem em geral  $N$  unidades completamente interconectadas [3]. Um exemplo com  $M = 1$  e  $N = 3$  pode ser visto na Fig. 3. No caso de uma RNN, a saída de

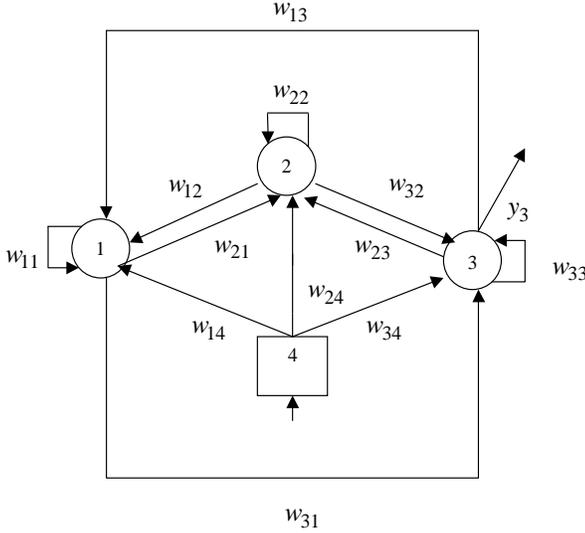


Fig. 3. RNN com  $N = 3$ ,  $M = 1$ , entrada  $x$  e saída  $y_3$ .

uma unidade no tempo  $k + 1$  não somente depende das entradas externas do instante de tempo anterior ( $x_j(k)$ ,  $j = 1, \dots, M$ ) mas também das saídas anteriores de todas as unidades ( $y_j(k)$ ,  $j = 1, \dots, N$ ). A dinâmica de uma RNN é descrita pelas equações

$$v_j(k+1) = \sum_{i=1}^N w_{ji}(k)y_i(k) + \sum_{i=1}^M w_{j,i+N}(k)x_i(k), \quad (2)$$

e

$$y_j(k+1) = \varphi(v_j(k+1)). \quad (3)$$

Nestas expressões  $w_{ji}(k)$  é o peso da conexão entre as unidades  $i$  e  $j$  no tempo  $k$  e  $\varphi(\cdot)$  usualmente é a função de ativação sigmoideal da equação (1).

Entre os algoritmos propostos para o treinamento da RNN, o mais conhecido é o RTRL (*Real-Time Recurrent Learning*) [3], [8] que pode ser usado para atualizar os pesos em tempo real. Nesse caso o sinal de erro é calculado pela equação

$$e_j(k) = \begin{cases} d_j(k) - y_j(k), & j \in \mathcal{N}_s \\ 0, & j \notin \mathcal{N}_s, \end{cases} \quad (4)$$

sendo  $d_j$  o sinal desejado e  $\mathcal{N}_s$  o conjunto das unidades visíveis cujas saídas são saídas da rede. No exemplo da Fig. 3, o conjunto  $\mathcal{N}_s$  é formado apenas pelo neurônio 3. Os pesos são atualizados de acordo com a equação

$$w_{\lambda l}(k+1) = w_{\lambda l}(k) + \eta \Delta w_{\lambda l}(k), \quad (5)$$

sendo  $\eta$  o passo de aprendizagem e

$$\Delta w_{\lambda l}(k) = \sum_{j \in \mathcal{N}_s} e_j(k) \Pi_{\lambda l}^j(k). \quad (6)$$

O termo  $\Pi_{\lambda l}^j(k)$  é calculado como:

$$\Pi_{\lambda l}^j(k) = \varphi'(v_j(k-1)) \left[ \sum_{i \in \mathcal{N}_o} w_{ji}(k-1) \Pi_{\lambda l}^i(k-1) + \right.$$

$$\left. + \delta_{\lambda j} u_l(k-1) \right], \quad \Pi_{\lambda l}^j(0) = 0, \quad (7)$$

sendo

- $\varphi'(\cdot)$  a derivada da função  $\varphi(\cdot)$ ;
- $u_l(k-1) = \begin{cases} x_l(k-1) & i \in \mathcal{N}_e \\ y_l(k-1) & i \in \mathcal{N}_o \end{cases}$ ;
- $\delta_{\lambda j} = \begin{cases} 1 & \lambda = j \\ 0 & \lambda \neq j \end{cases}$ ;
- $\mathcal{N}_o$  o conjunto formado por todas as unidades menos as de entrada da rede; e
- $\mathcal{N}_e$  o conjunto formado apenas pelas unidades de entrada.

A complexidade computacional do algoritmo RTRL para uma rede com  $N$  unidades completamente interconectadas e  $M$  entradas externas é mostrada na Tabela III.

Operações	$N^2$ de operações por iteração
$\times$	$M[N^2(N+1)+2]+$ $+(N+1)(N^3+1)+2N^2$
NL	$2N$
$+$	$MN(N^2-N-3)+$ $(1-N)(1-N^3)+3N^2$

Tabela III

COMPLEXIDADE COMPUTACIONAL DO ALGORITMO RTRL, NO TREINAMENTO DE UMA REDE RNN COM  $M$  ENTRADAS E  $N$  UNIDADES.

### 3. RESULTADOS EXPERIMENTAIS

As funções de transferência dos canais usados nas simulações estão na Tabela IV. O canal H2 segue um modelo não-linear apresentado em [5], em que se separa a parte linear da não-linear. A saída deste canal é  $y_1(k)$ , sendo  $y(k)$  a saída da parte linear.

<b>H1</b> (Linear com fase não-mínima [2]) $H_1(z) = 0,3482 + 0,8704z^{-1} + 0,3482z^{-2}$ Zeros: $\{-0,5001; -1,9996\}$
<b>H2</b> (Não-Linear [5]) Parte não-linear: $y_1(k) = y(k) + 0,2y^2(k) - 0,1y^3(k)$ Parte linear: $H_{2l} = H_1(z)$ Zeros da parte linear: $\{-0,5001; -1,9996\}$
<b>H3</b> (Linear com zeros no círculo unitário [3]) $H_3(z) = 1 - 2z^{-1} + z^{-2}$ Zeros: $\{1; 1\}$

Tabela IV

MODELOS DE CANAIS DE COMUNICAÇÃO USADOS NAS SIMULAÇÕES.

Com o objetivo de comparar o desempenho dos diversos tipos de equalizadores, foram obtidas experimentalmente curvas de taxa de erro de bits para os canais da Tabela IV. Considerou-se os pesos dos equalizadores fixos, depois dos algoritmos de treinamento dos mesmos terem convergido. Nesta situação, utilizou-se uma seqüência de  $10^6$  ou  $10^7$  bits para a medida da taxa de erros. Com isso, o número mínimo de erros para uma dada relação sinal-ruído resultou sempre maior que 30.

As configurações de cada equalizador usado estão mostradas na Tabela V. Todos os equalizadores utilizados são de 4ª ordem ( $M = 5$ ) e o atraso é de 3 amostras ( $\tau_d = 3$ ), o que é suficiente para definir a estrutura do LTE. Para a rede MLP utilizou-se a configuração proposta por Gibson *et. al.* em [2]. Para a rede RNN, por sua vez, segue-se a configuração de [3] tomando-se o cuidado de aumentar o número de entradas externas.

No caso da RBF, o número de neurônios utilizados é bem maior que nas outras estruturas. Considerando uma situação sem ruído, para  $M = 5$  e um canal de 2ª ordem, o número total de diferentes entradas possíveis (estados de entrada do equalizador) é 128 [4]. Dessa forma, se tais estados fossem conhecidos poder-se-ia usar 128 neurônios na camada oculta da RBF com centros nestes estados para implementar um equalizador quase ideal. No entanto, a rede foi treinada com 256 unidades ocultas, cujas funções gaussianas ficaram centradas nos vetores iniciais de entrada e a variância foi feita igual à do sinal. Ou seja, o fato de não se conhecer os estados de entrada foi compensado com o aumento do número de neurônios. Não foi utilizada nenhuma técnica para adaptação dos centros e variâncias.

Op.	LTE	MLP (5,9,3,1)	RBF ( $M_1 = 256$ )	RNN ( $N = 3$ )
×	65	465	2052	320
÷	10	—	—	—
NL	—	36	257	6
+	45	395	3075	124

Tabela V

CONFIGURAÇÕES DOS EQUALIZADORES UTILIZADOS E NÚMERO DE OPERAÇÕES DE SEUS ALGORITMOS DE TREINAMENTO PARA  $M = 5$ .

Na Fig. 4-a, estão representadas as curvas de taxa de erro de bits para o canal H1. Neste caso, o desempenho do equalizador MLP para o intervalo de relação sinal-ruído (SNR) considerado é muito próximo ao do obtido com o equalizador linear. No entanto, os equalizadores RBF e RNN possuem desempenhos superiores ao do equalizador MLP e consequentemente ao do linear. É importante notar que a rede recorrente consegue resultados melhores que os da RBF, possuindo uma complexidade computacional aproximadamente 6 vezes menor no treinamento (considerando o número de multiplicações dos algoritmos RTRL e LMS para atualização dos pesos e a configuração da Tabela V).

No caso do canal H2, obtém-se as curvas de taxa de erro de bits da Fig. 4-b. Nota-se que neste caso, o equalizador MLP apresenta um desempenho superior ao do linear, diferente do que acontece nos canais lineares. Além disso, observa-se que o equalizador não-linear recorrente (RNN) apresenta novamente o melhor desempenho.

Considerando agora o canal H3, obteve-se as curvas de taxa de erro de bits da Fig. 4-c. Para este canal, o desempenho da rede recorrente (RNN) é muito superior ao desempenho dos outros equalizadores.

A partir dessas simulações, observou-se que a rede recorrente treinada com o algoritmo RTRL possui um desempenho superior a todos os tipos de equalizadores apresentados. Além disso, como pode ser observado na Tabela V, seu algoritmo de treinamento apresenta a menor complexidade computacional quando comparado aos algoritmos de treinamento das outras redes consideradas. Os bons resultados da rede neural recorrente sugerem a introdução de algum tipo de recorrência nas redes MLP e RBF visando a melhora

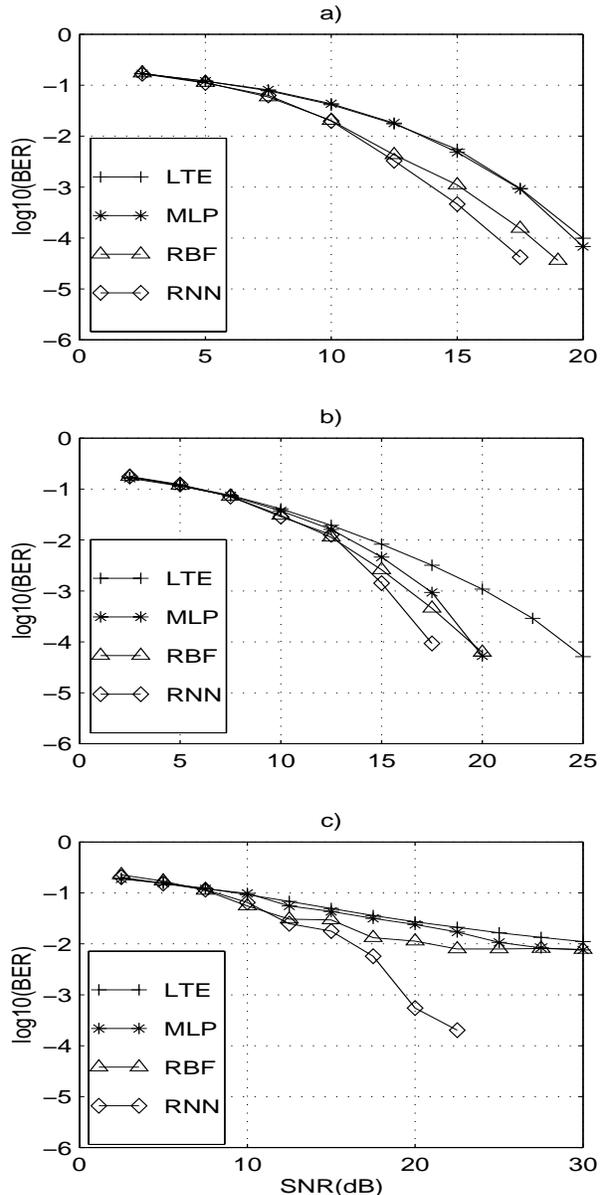


Fig. 4. Curvas de taxa de erro de bits para  $M = 5$ ,  $\tau_d = 3$ , 2-PAM, configurações dos equalizadores da Tabela V e canal a) H1; b) H2; c) H3.

de desempenho. Isso pode ser feito de diversas formas, mas a que é mais usual quando se trata de equalização, é a realimentação de decisões (DF). Fez-se então simulações para os canais da Tabela IV utilizando esta técnica (Fig. 2).

No caso do uso de DF utiliza-se  $M_f$  entradas correspondentes à seqüência de saída do canal afetada de ruído e  $M_b$  entradas relacionadas à realimentação de decisões. Para efeito de comparação, as estruturas com DF foram simuladas com o mesmo número de entradas das estruturas sem DF, isto é,  $M = M_f + M_b$ . Verificou-se experimentalmente que se deve usar  $M_f > M_b$  para obter melhores resultados.

Comparando as Figuras 4 e 5, conclui-se que a realimentação das decisões já tomadas garante um melhor desempenho na equalização. Para o canal H3, por exemplo, a realimentação de

decisões possibilitou uma melhora do desempenho do LTE e das redes MLP e RBF, sendo que tais redes apresentam resultados superiores aos da rede recorrente considerada.

Para o canal H2, houve melhoras no desempenho das redes MLP e RBF, enquanto que o LTE-DF apresentou maior taxa de erros que a estrutura implementada sem DF. Isso decorre da realimentação de decisões errôneas, o que aumenta a probabilidade de erro desse equalizador. Porém, esse efeito não foi observado quando se utilizou as estruturas não-lineares na equalização, fato que merece ser estudado mais a fundo. Um outro aspecto que merece maior atenção está relacionado aos valores de  $M_f$  e  $M_b$ . Por exemplo, independentemente do canal simulado, a rede RBF com  $M_f = 3$  e  $M_b = 2$  tem um desempenho muito pior do que com  $M_f = 4$  e  $M_b = 1$ . Isso pode estar relacionado com o número de neurônios da rede e/ou com o tipo de canal utilizado. Esse problema também deve ser investigado mais a fundo com o objetivo de se escolher esses números de uma forma mais sistemática.

Quando se usa a realimentação de decisões, torna-se possível a redução do número de neurônios das redes para uma mesma probabilidade de erro. No caso do canal H2, isso pode ser comprovado observando-se a Fig. 6 e a Tabela VI. Nota-se que com uma configuração mais simples ( $MLP_2$  e  $RBF_2$ ) e com o uso da realimentação de decisões, o desempenho dessas redes ainda é comparável ao das consideradas anteriormente. Na Fig. 6 é mostrada também a taxa de erro de bits da rede RNN que, para este canal, ficou bem próxima das demais obtidas com realimentação de decisões. Comparando agora as Tabelas V e VI, observa-se que, mesmo com o uso de recorrência, as redes MLP e RBF apresentam um treinamento com maior complexidade que a rede RNN. Comparando-se a complexidade computacional dos algoritmos de treinamento da rede MLP da Tabela VI com a rede RNN da Tabela V, observa-se que a primeira necessita de um menor número de multiplicações e de um maior número de somas e de cálculos de não-linearidades (NL). Desta forma, a rede RNN parece ser uma boa alternativa para a equalização não-linear, considerando a complexidade computacional de seu algoritmo de treinamento e o desempenho obtido.

Op.	$MLP_2$ (5,5,2,1)	$RBF_2$ ( $M_1 = 150$ )
×	288	1204
÷	—	—
NL	26	151
+	238	1803

Tabela VI

CONFIGURAÇÕES DOS EQUALIZADORES  $MLP_2$  E  $RBF_2$  UTILIZADOS PARA OBTENÇÃO DA FIGURA 6 E OS NÚMERO DE OPERAÇÕES NO TREINAMENTO.

#### 4. A EQUALIZAÇÃO COM REALIMENTAÇÃO DE DECISÕES VISTA COMO CLASSIFICAÇÃO

A interpretação de um equalizador como um classificador permite entender porque ocorrem melhoras com o uso da realimentação de decisões. Para exemplificar, será considerado um canal com função de transferência  $H(z) = 0,5 + z^{-1}$  e um equalizador com duas entradas ( $M = 2$ ). Supondo transmissão binária sem ruído, as possíveis entradas do equalizador  $y(k)$  e  $y(k-1)$ , chamadas aqui de estados, são calculadas através das combinações de  $x(k)$ ,  $x(k-1)$  e  $x(k-2)$ . Na Tabela VII estão listadas todas as possíveis entradas do equalizador na situação de ruído nulo.

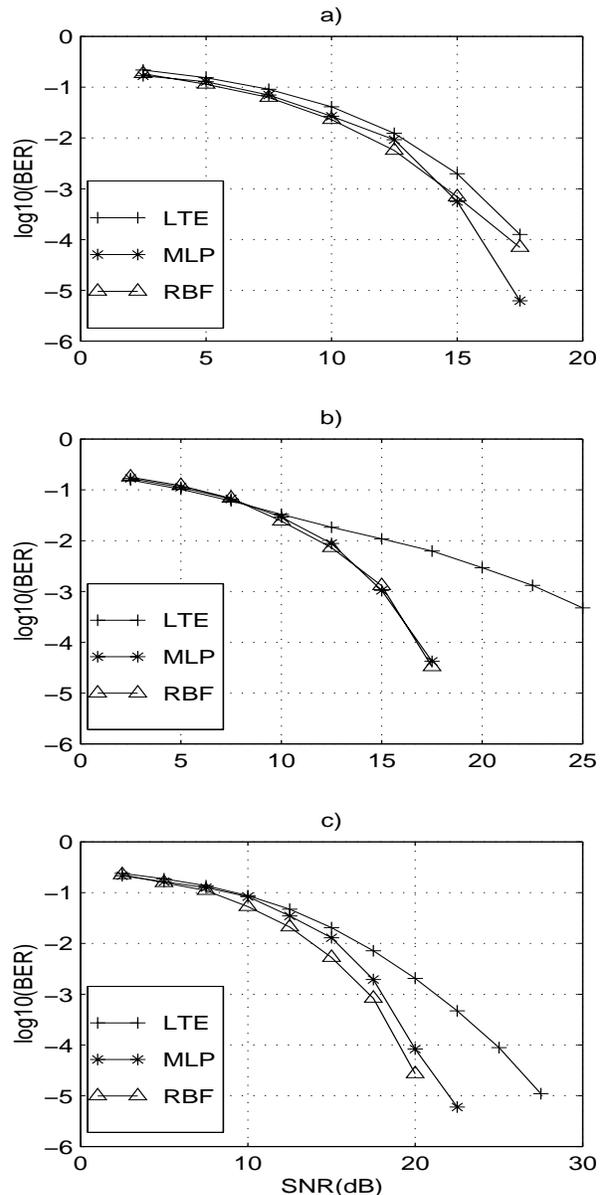


Fig. 5. Curvas de taxa de erro de bits para  $M_f = 3$ ,  $M_b = 2$  (MLP e LTE) e  $M_f = 4$ ,  $M_b = 1$  (RBF),  $\tau_d = 3$ , 2-PAM, configurações dos equalizadores da Tabela V, utilização de realimentação de decisões e canal a) H1; b) H2; c) H3.

Considerando um equalizador de 1<sup>o</sup> ordem ( $M = 2$ ,  $\tau_d = 0$ ) e o canal da Tabela VII obtém-se a curva de separação do equalizador ótimo segundo Bayes [4] mostrada na Fig.7-a. Nesta Figura, nota-se que a curva ótima de separação segundo Bayes é não-linear, o que faz com que um equalizador linear não atinja um bom desempenho.

Supondo agora que a decisão  $\hat{x}(k-1)$  esteja correta, pode-se utilizá-la juntamente com  $y(k)$ ,  $y(k-1)$  para estimar  $x(k)$ . Assim, obtém-se as regiões de separação das Figuras 7-b e 7-c que também se referem ao canal  $H(z) = 0,5 + z^{-1}$  com atraso nulo.

A partir dessas figuras, conclui-se que neste caso a realimentação de decisões corretas garante um melhor desempenho na classificação devido à separação dos estados em dois

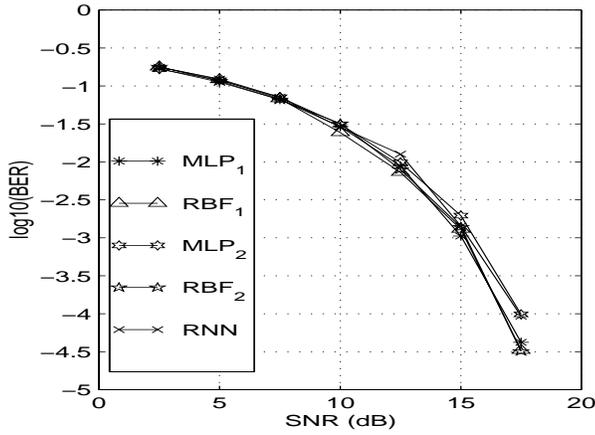


Fig. 6. Curvas de taxa de erro de bits para  $M = 5$  (RNN),  $M_f = 3$ ,  $M_b = 2$  (MLP) e  $M_f = 4$ ,  $M_b = 1$  (RBF),  $\tau_d = 3$ , 2-PAM, configurações dos equalizadores MLP<sub>1</sub>, RBF<sub>1</sub> e RNN segundo a Tabela V, e MLP<sub>2</sub>, RBF<sub>2</sub> segundo a Tabela VI; canal H2.

planos distintos. No caso em que se considerou atraso nulo sem a realimentação de decisões, um equalizador linear transversal não consegue proporcionar uma separação coerente das regiões de decisão. Por outro lado, fazendo uso da decisão passada  $\hat{x}(k-1)$ , ainda no caso de atraso nulo, o equalizador linear passa a ser ótimo segundo Bayes para uma relação sinal-ruído de 10 dB.

## 5. REFERÊNCIAS

- [1] BOUCHIRED, S.; ROVIRAS, D.; CASTANI, F. Equalisation of satellite mobile channels with neural network techniques. *Space Communications*, v. 15, p. 209-220, 1998/1999.
- [2] GIBSON, G. J.; SIU, S.; COWAN, C. F. N. The application of nonlinear structures to the reconstruction of binary signals. *IEEE Transactions on Signal Processing*, v. 39, p.1877-1884, Aug.1991.
- [3] KECHRIOTIS, G.; ZERVAS, E.; MONOLAKOS, E. S. Using recurrent neural networks for adaptive communication channel equalization. *IEEE Transactions on Neural Networks*, v.5, p.267-278, Mar. 1994.
- [4] MULGREW, B. Applying radial basis function. *IEEE Signal Processing Magazine*, v.13, p.50-65, Mar. 1996.
- [5] DESTRO FILHO, J. B. *Egalisation aveugle de canaux de communication à l'aide d'algorithmes de Bussgang de réseaux neuronaux*. Nice, 1998. 261p. Thèse de Docteur en Sciences - Ecole Doctorale "Sciences Pour l'Ingénieur", Université de Nice-Sofhia Antipolis.
- [6] HAYKIN, S. *Adaptive Filter Theory*. 3.ed. New Jersey, Prentice Hall, 1996.
- [7] MIRANDA, M. D.; GERKEN, M.; SILVA, M. T. M. Efficient implementation of error-feedback LSL algorithm. [Letter]. *Electronics Letters*, v.35, n.16, p.1308-1309, Aug. 1999.
- [8] HAYKIN, S. *Neural Networks*. 2.ed. New Jersey, Prentice Hall, 1999.

$x(k)$	$x(k-1)$	$x(k-2)$	$y(k)$	$y(k-1)$
1	1	1	1,5	1,5
1	1	-1	1,5	-0,5
-1	1	1	0,5	1,5
-1	1	-1	0,5	-0,5
1	-1	1	-0,5	0,5
1	-1	-1	-0,5	-1,5
-1	-1	1	-1,5	0,5
-1	-1	-1	-1,5	-1,5

Tabela VII

ENTRADA E ESTADOS DESEJADOS DO CANAL  $H(z) = 0,5 + z^{-1}$ ,  $M = 2$ .

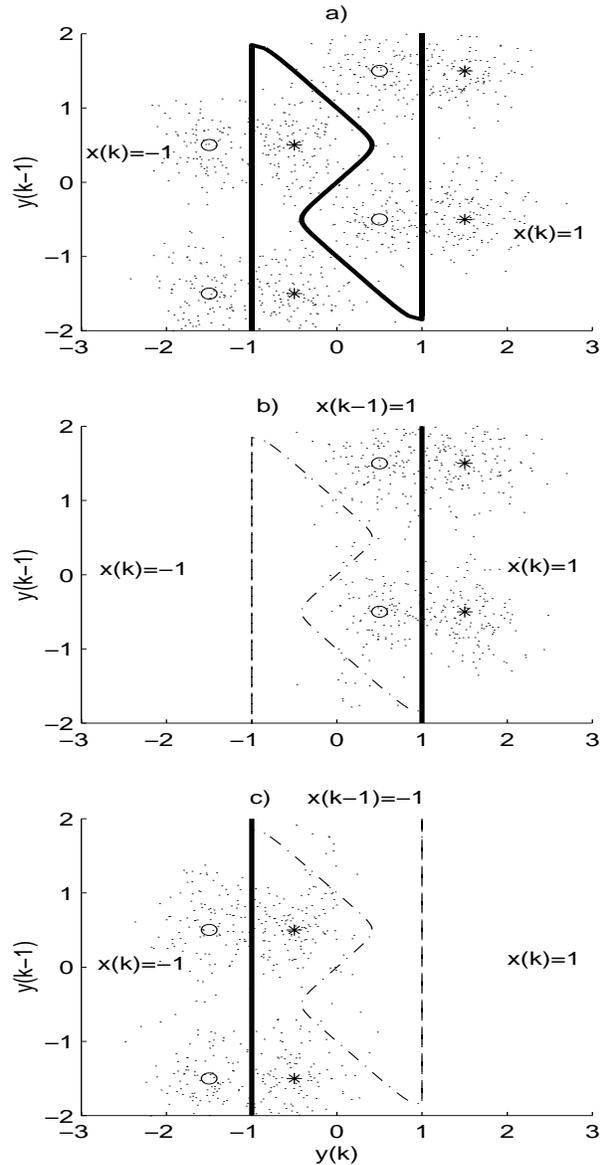


Fig. 7. Estados desejados; - curva ótima de separação das regiões de decisão, 2-PAM, SNR=10dB,  $\tau_d = 0$ ,  $H(z) = 0,5 + z^{-1}$ , a) Sem o uso do DF,  $M = 2$ ; b) Com o uso do DF,  $M_f = 2$ ,  $M_b = 1$ ,  $\hat{x}(k-1) = 1$ ; c) Com o uso do DF,  $M_f = 2$ ,  $M_b = 1$ ,  $\hat{x}(k-1) = -1$ ;