UFMG

INTRODUÇÃO À INTELIGÊNCIA COMPUTACIONAL

SISTEMAS NEBULOSOS ADAPTATIVOS:

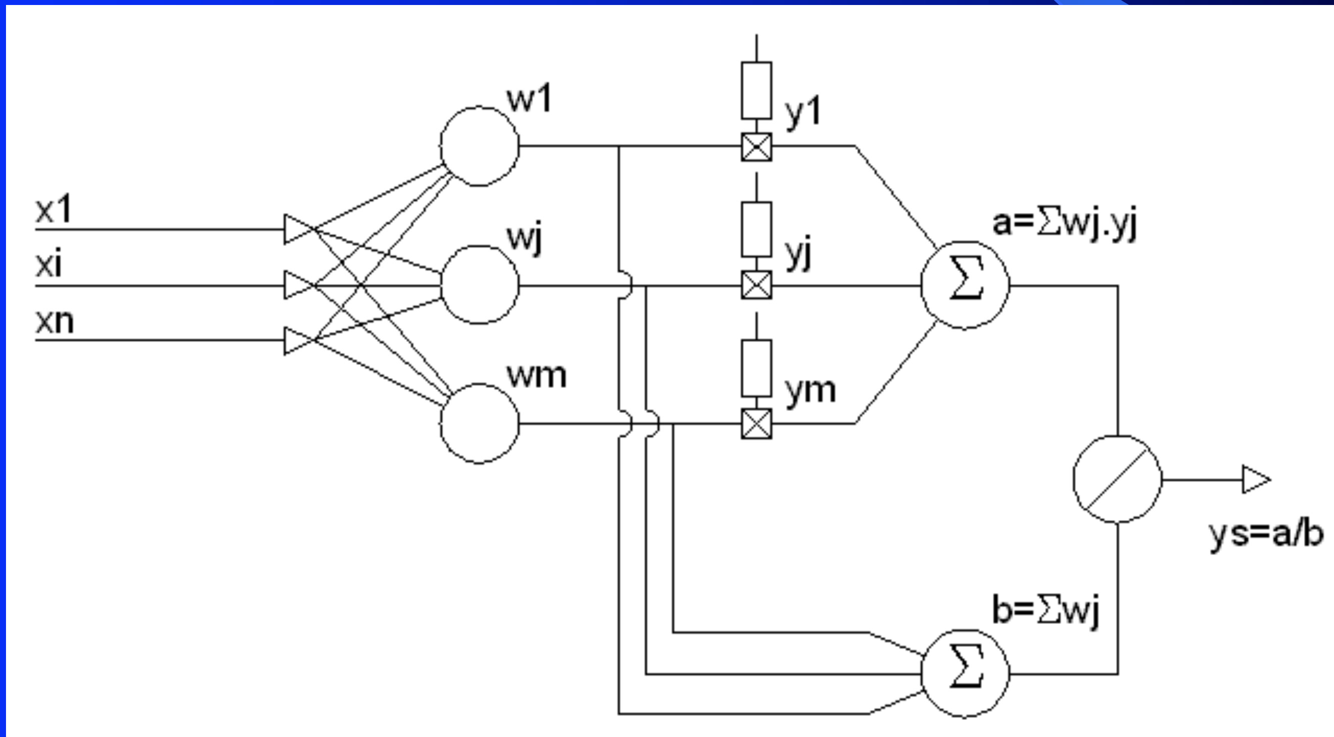
NEO-FUZZY-NEURON

Prof. Walmir Matos Caminhas - DELT/UFMG

PROPOSTA

- YAMAKAWA: NEO-FUZZY-NEURON;
- CAMINHAS: TREINAMENTO ON-LINE.

Estrutura Neurofuzzy



Regras implementadas

Se (x_1 é A_{1j}) e (x_2 é A_{2j})..e (x_i é A_{ij})..e (x_n é A_{nj})

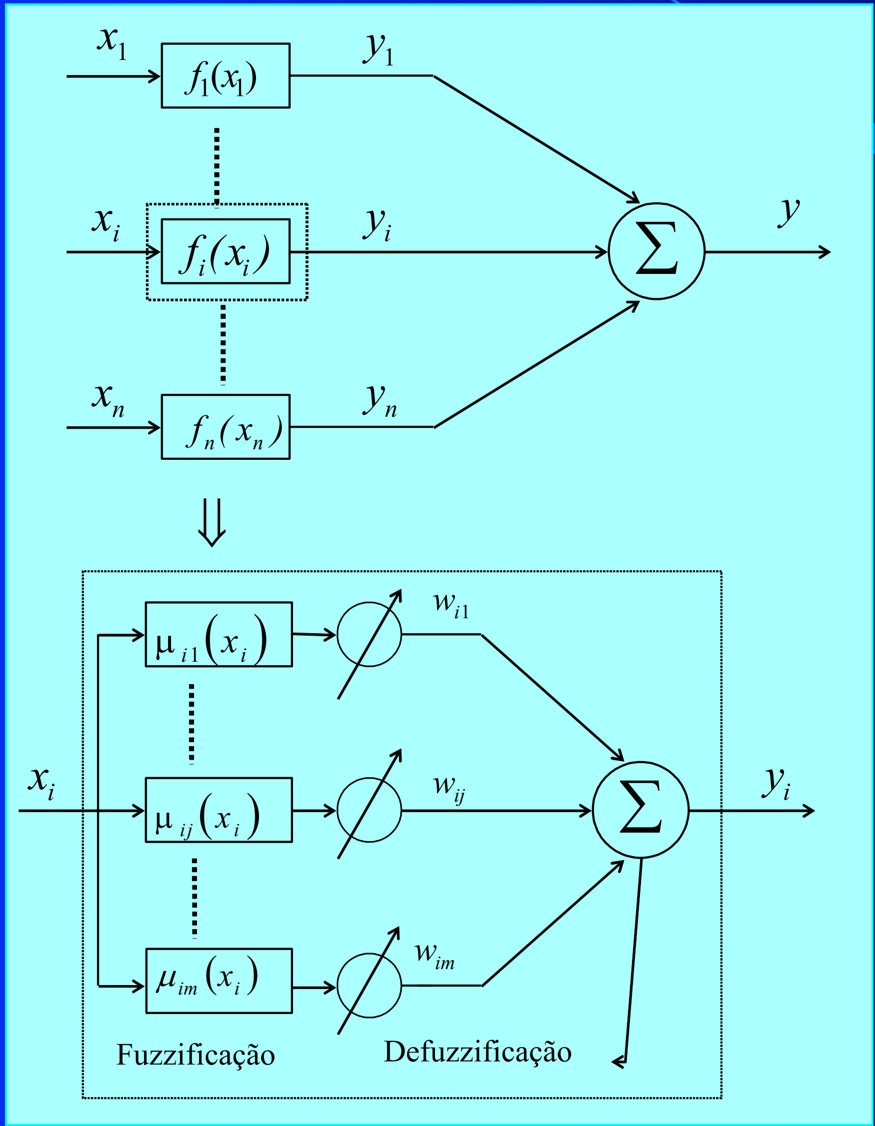
Então $y = P_{1j}x_1 + P_{2j}x_1 + \ldots + + P_{ij}x_1 + P_{nj}x_n + q_j$

$$y_j = \sum_{i=1}^{n} P_{ij} x_i + q_j$$

$$w_j = \prod_{i=1}^{n} \mu_{Aij}(x_i)$$

$$w_j = \mu_{A1j}(x_1).\ \mu_{A2j}(x_2).\ \ldots\ .\mu_{Aij}(x_j).\ \mu_{Anj}(x_n)$$

ESTRUTURA DO NFN



ESTRURURA DO NFN

Se x_i é A_{i1} Então y_i é w_{i1}

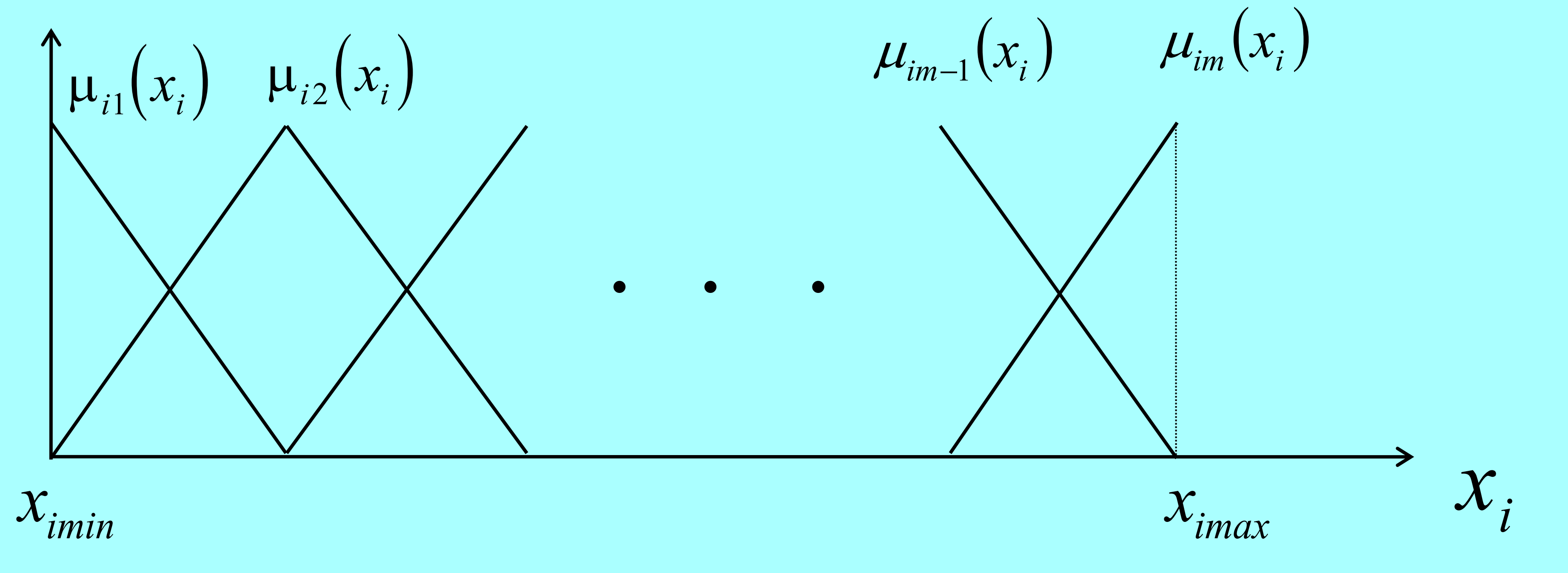
$\vdots$

Se x_i é A_{im} Então y_i é w_{im}

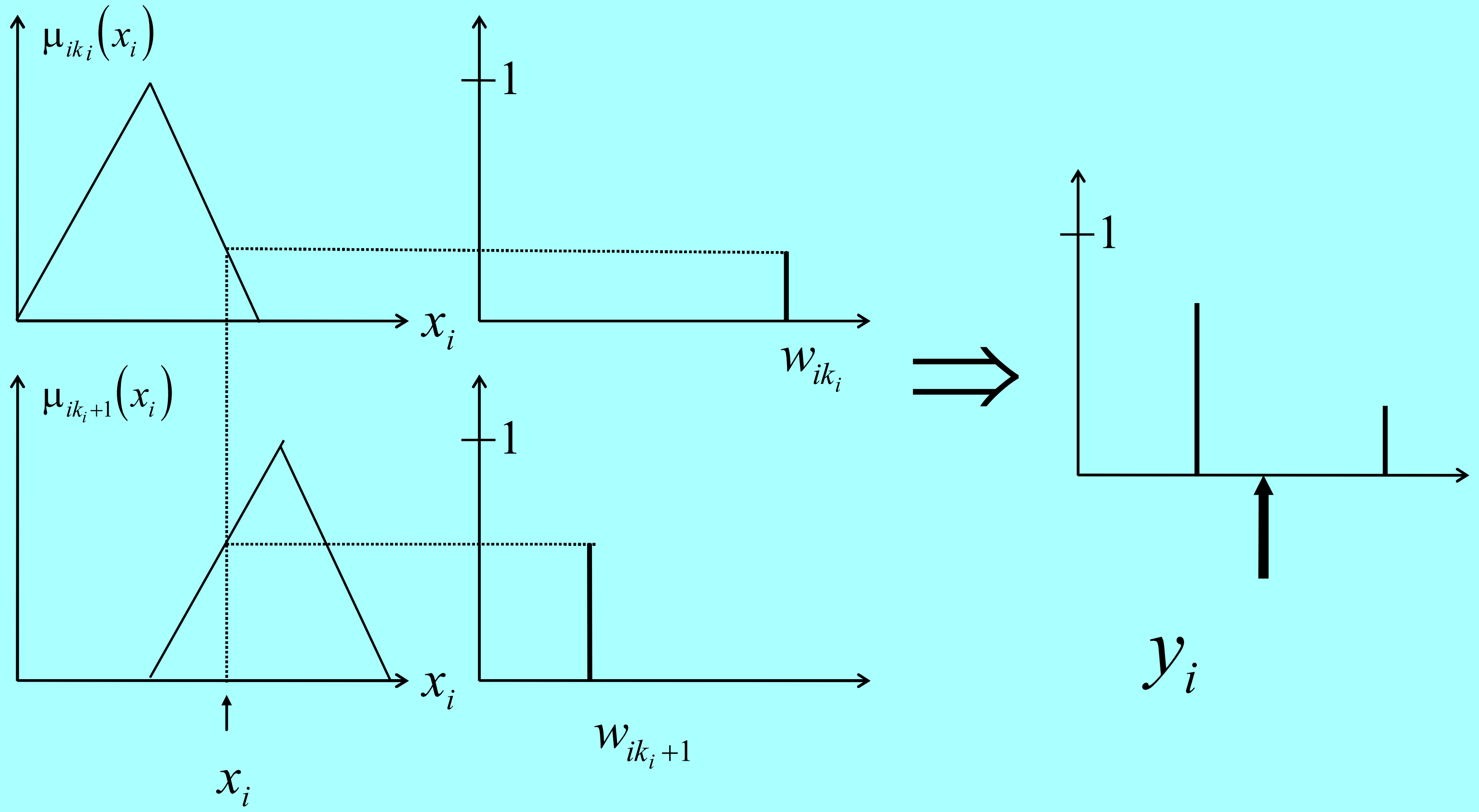
ESTRUTURA DO NFN

$$y = \sum_{i=1}^{n} y_i(x_i) = \sum_{i=1}^{n} f_i(x_i)$$

ESTRUTURA



MECANISMO DE INFERÊNCIA



COMPUTAÇÃO DA SAÍDA

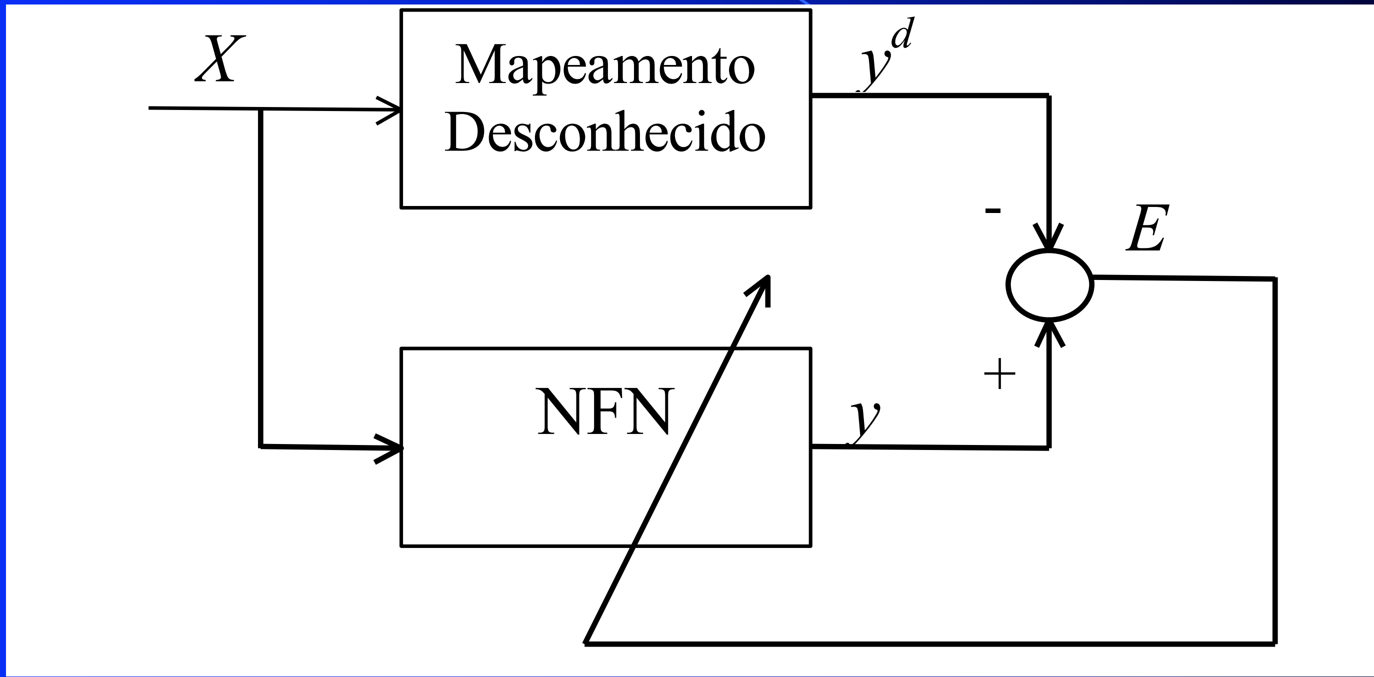
$$\mathrm{f_i(x_i)} = \frac{\sum_{j=1}^{\mathrm{m}} \mu_{ij}(\mathrm{x_i}).\mathrm{w_{ij}}}{\sum_{j=1}^{m} \mu_{ij}(\mathrm{x_i})}$$

$$= \frac{\mu_{\mathrm{ik_i}}(\mathrm{x_i}).\mathrm{w_{ik_i}} + \mu_{\mathrm{ik_i+1}}(\mathrm{x_i}).\mathrm{w_{ik_i+1}}}{\mu_{i\mathrm{k_i}}(\mathrm{x_i}) + \mu_{\mathrm{ik_i+1}}(\mathrm{x_i})}$$

COMPUTAÇÃO DA SAÍDA

$$f_i(x_i) = \mu_{ik_i}(x_i).w_{ik_i} + \mu_{ik_i+1}(x_i).w_{ik_i+1}$$

APRENDIZADO DO NFN



$$\mathrm{E}_{\mathrm{t}} = \frac{1}{2}\left(y_t - y_t^d\right)^2 = E_t\left(w_{ij}\right)$$

APRENDIZADO DO NFN

$$\min E_t = E_t(w_{ij})$$

$$w_{ij} \in \Re$$

$$\forall i = 1,\ldots,n; j = 1,\ldots,m$$

APRENDIZADO DO NFN

$$\min \qquad E_t = E_t(w_{ij}) = E_t(\tilde{w})$$

$$w_{ij} \in \Re$$

$$\forall i = 1,\ldots,n; j = k_i,(k_i+1)$$

$$\tilde{w} = \left\lfloor w_{1k_1}\ w_{1(k_1+1)} \ldots w_{ik_i}\ w_{i(k_i+1)} \ldots w_{nk_n}\ w_{n(k_n+1)} \right\rfloor$$

APRENDIZADO DO NFN

$$\nabla E_t(\widetilde{w}) = \left(y_t - y_t^d\right)\begin{bmatrix} \mu_{1k_1} \\ \mu_{1k_1+1} \\ \vdots \\ \mu_{ik_i} \\ \vdots \\ \mu_{nk_n+1} \end{bmatrix}$$

APRENDIZADO DO NFN

$$H=\begin{bmatrix} \mu_{1k_1}^2 & \mu_{1k_1}.\mu_{1k_1+1} & \cdots & \mu_{1k_1}.\mu_{ik_i} & \cdots & \mu_{1k_1}.\mu_{nk_n+1} \\ \mu_{1k_1}.\mu_{1k_1+1} & \mu_{1k_1+1}^2 & & \cdots & & \mu_{1k_1+1}.\mu_{nk_n+1} \\ \vdots & & \ddots & & & \vdots \\ \mu_{1k_1}.\mu_{ik_i} & & \cdots & \mu_{ik_i}^2 & \cdots & \mu_{ik_i}.\mu_{nk_n+1} \\ \vdots & & \cdots & & \ddots & \\ \mu_{1k_1}.\mu_{nk_n+1} & \mu_{1k_1+1}.\mu_{nk_n+1} & \cdots & \mu_{ik_i}.\mu_{nk_n+1} & \cdots & \mu_{nk_n+1}^2 \end{bmatrix}$$

$$V^T.H.V=\left(v_1.\mu_{1k_1}+v_2.\mu_{1k+1}+\cdots+v_{2n}.\mu_{nk_n+1}\right)^2\geq 0$$

APRENDIZADO DO NFN

H semi-definida positivo e E(w) é convexa, portanto, mínimo global.

APRENDIZADO DO NFN

$$\tilde{h}^{j} = -\nabla E_{t}\left(\tilde{w}^{j}\right)$$

$$\mathrm{w}_{\mathrm{ik_i}}^{j+1} = \mathrm{w}_{\mathrm{ik_i}}^{j} - \alpha^{j} \cdot \frac{\partial \mathrm{E_t}\left(\mathrm{w}_{\mathrm{ik_i}}^{j}\right)}{\partial \mathrm{w}_{\mathrm{ik_i}}^{j}}$$

$$w_{ik_i}^{j+1} = w_{ik_i}^{j} - \alpha^{j} \cdot \left(y_t^{j} - y_t^{d}\right) \cdot \mu_{ik_i}\left(x_{ti}\right)$$

Taxa de Apredizado Ótima

Teorema:

Dado um padrão e o valor desejado para o mapeamento é possível determinar uma expressão fechada para a taxa de aprendizado α tal que utilizando o método do gradiente, os pesos do NFN que proporcionam erro de aproximação nulo podem ser determinados em um passo.

Prova do Teorema

$$\mathrm{E}_{\mathrm{t}}^{1} = \frac{1}{2}\left(y_t^1 - y_t^d\right)^2 = \frac{1}{2}.\left(e_t^1\right)^2 = 0$$

$$e_t^1 = \left(\left(\sum_{i=1}^{n} w_{ik_i}^1 .\mu_{k_i}(x_i) + w_{ik_i}^1 .\mu_{k_i+1}(x_{ti})\right) - y_t^d\right)$$

$$w_{\mathrm{ik_i}}^1 = w_{\mathrm{ik_i}}^0 - \alpha.e_t^0 .\mu_{\mathrm{ik_i}}\left(\mathrm{x}_{ti}\right)$$

$$w_{\mathrm{ik_i+1}}^1 = w_{\mathrm{ik_i+1}}^0 - \alpha.e_t^0 .\mu_{\mathrm{ik_i+1}}\left(\mathrm{x}_{ti}\right)$$

APRENDIZADO DO NFN

$$\alpha = \frac{1}{\sum_{i=1}^{n} \mu_{ik_i}\left(x_{ti}\right)^2 + \mu_{ik_i+1}\left(x_{ti}\right)^2}$$

Taxa de aprendizado ótima (Caminhas, 1997)

Identificação de Sistemas Dinâmicos

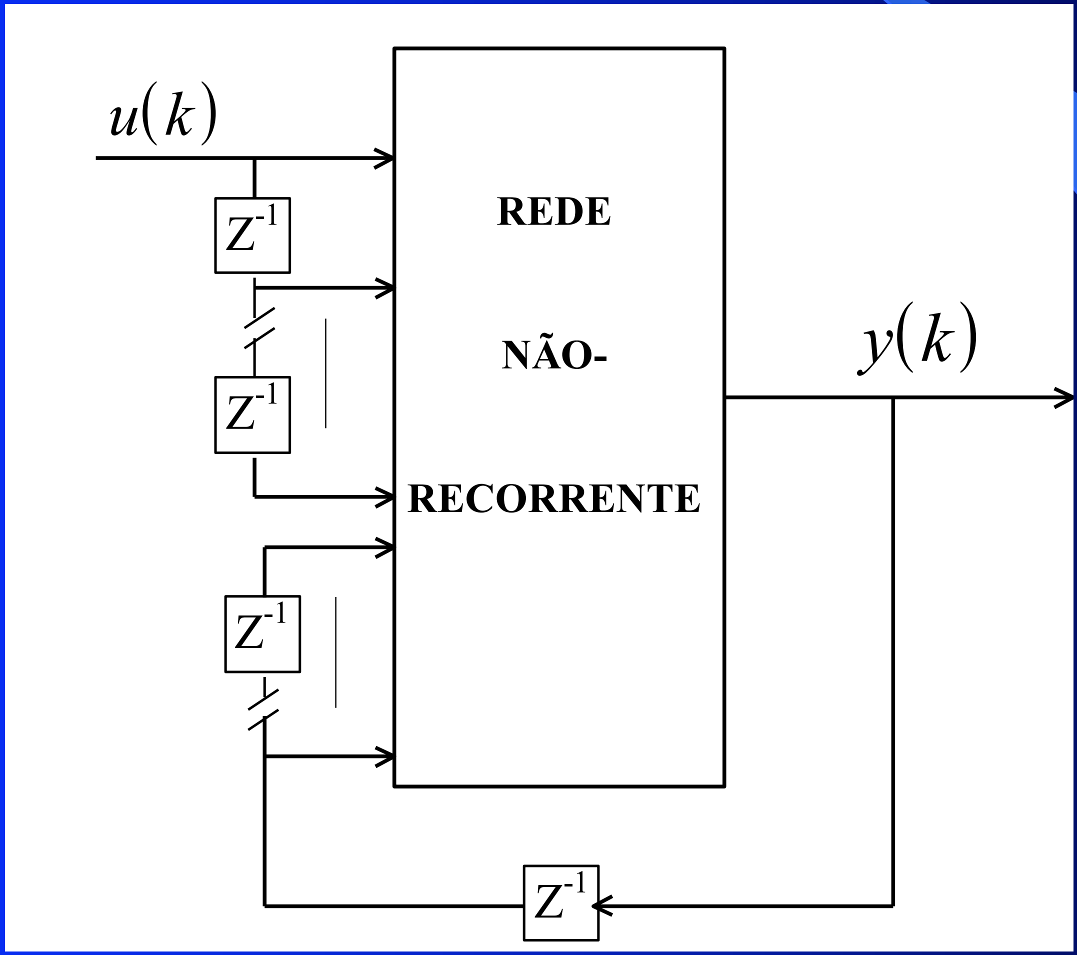
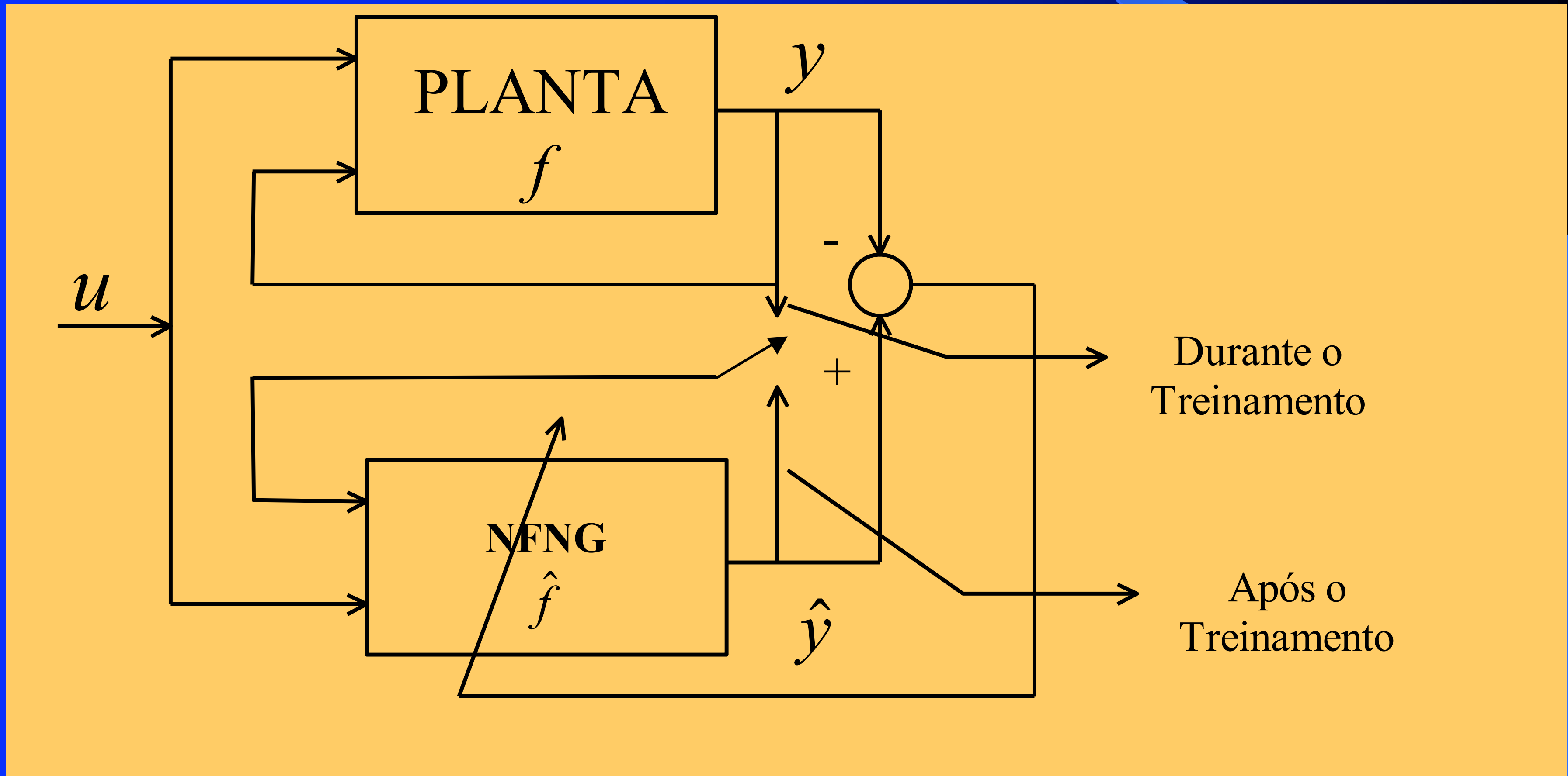


Diagrama da Estratégia de treinamento do NFNG



Exemplo 1

$$y(k+1) = 0.3.y(k) + 0.5.y(k-1) + g[u(k)]$$

$$g(u) = 0.6\,\mathrm{sen}(\pi u) + 0.3\,\mathrm{sen}(3\pi u) + 0.1\,\mathrm{sen}(5\pi u)$$

$$\hat{y}(k+1) = 0.3.\hat{y}(k) + 0.5.\hat{y}(k-1) + \hat{g}[u(k)]$$

Estrutura do NFN

- Número de entradas: 1
- Número de saídas: 1
- Número de partições nebulosas: 20

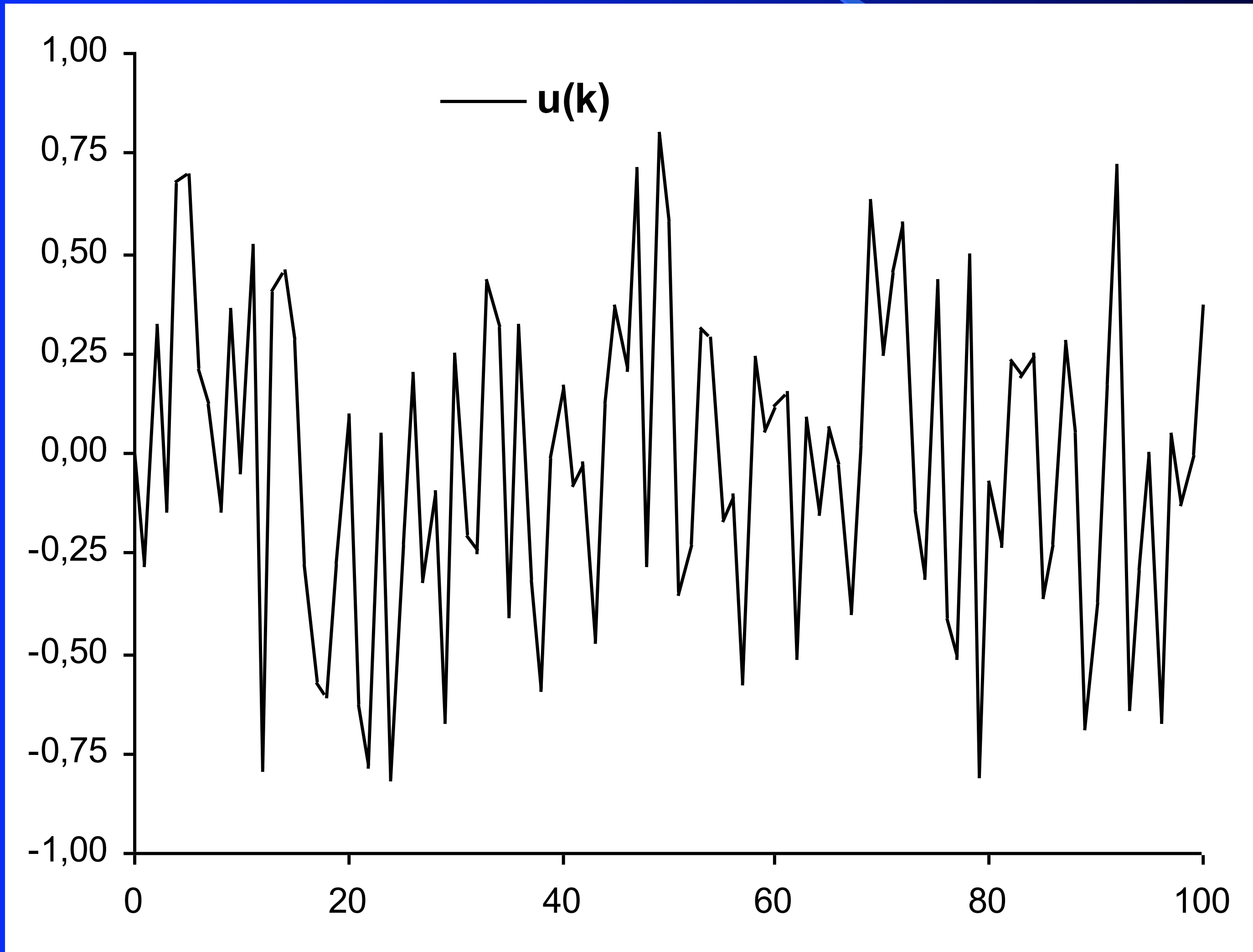
Metodologia de treinamento

- padrões de treinamento: 10.000 com u aleatório no intervalo [-1, +1]
- padrões de validação:

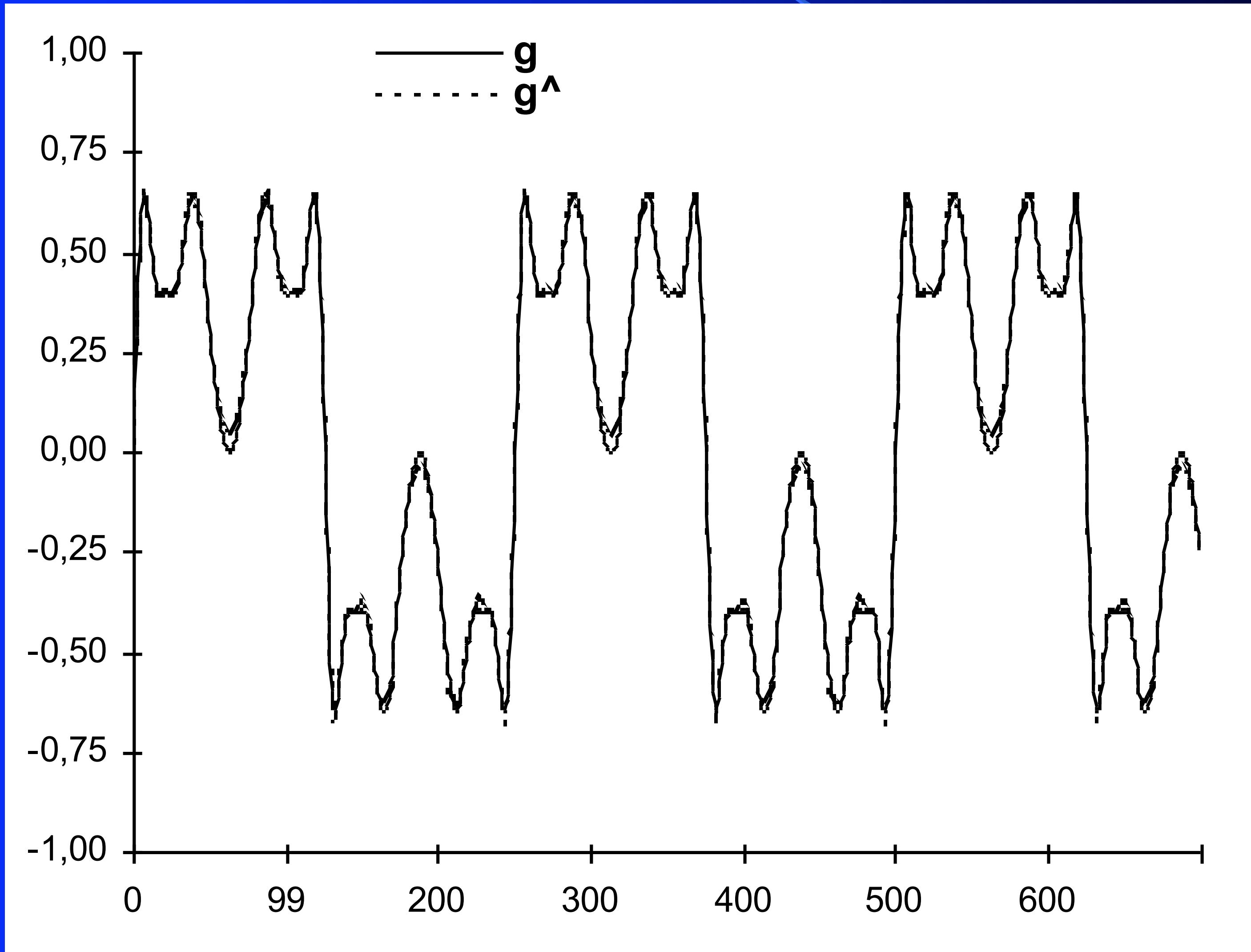
$$u(k) = \mathrm{sen}(2.\pi.k/250)$$

$$u(k) = \begin{cases} \mathrm{sen}(2.\pi.k/250), & 251 \leq k \leq 500 \\ 0.5\,\mathrm{sen}(2.\pi.k/25) + 0.5\,\mathrm{sen}(2.\pi.k/250), & \text{caso contrário} \end{cases}$$

Sinal de u utilizado no treinamento

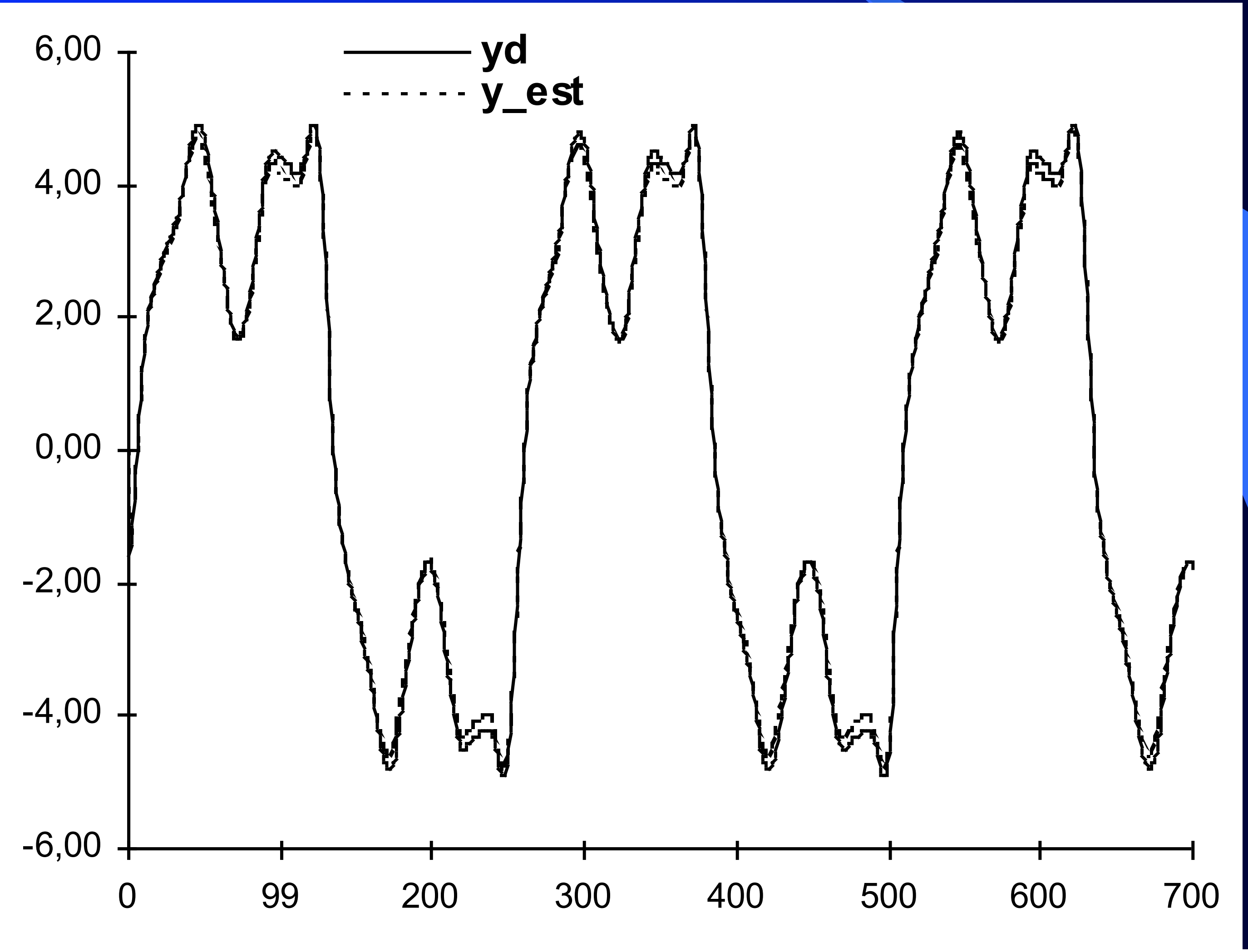


Resultado da identificação



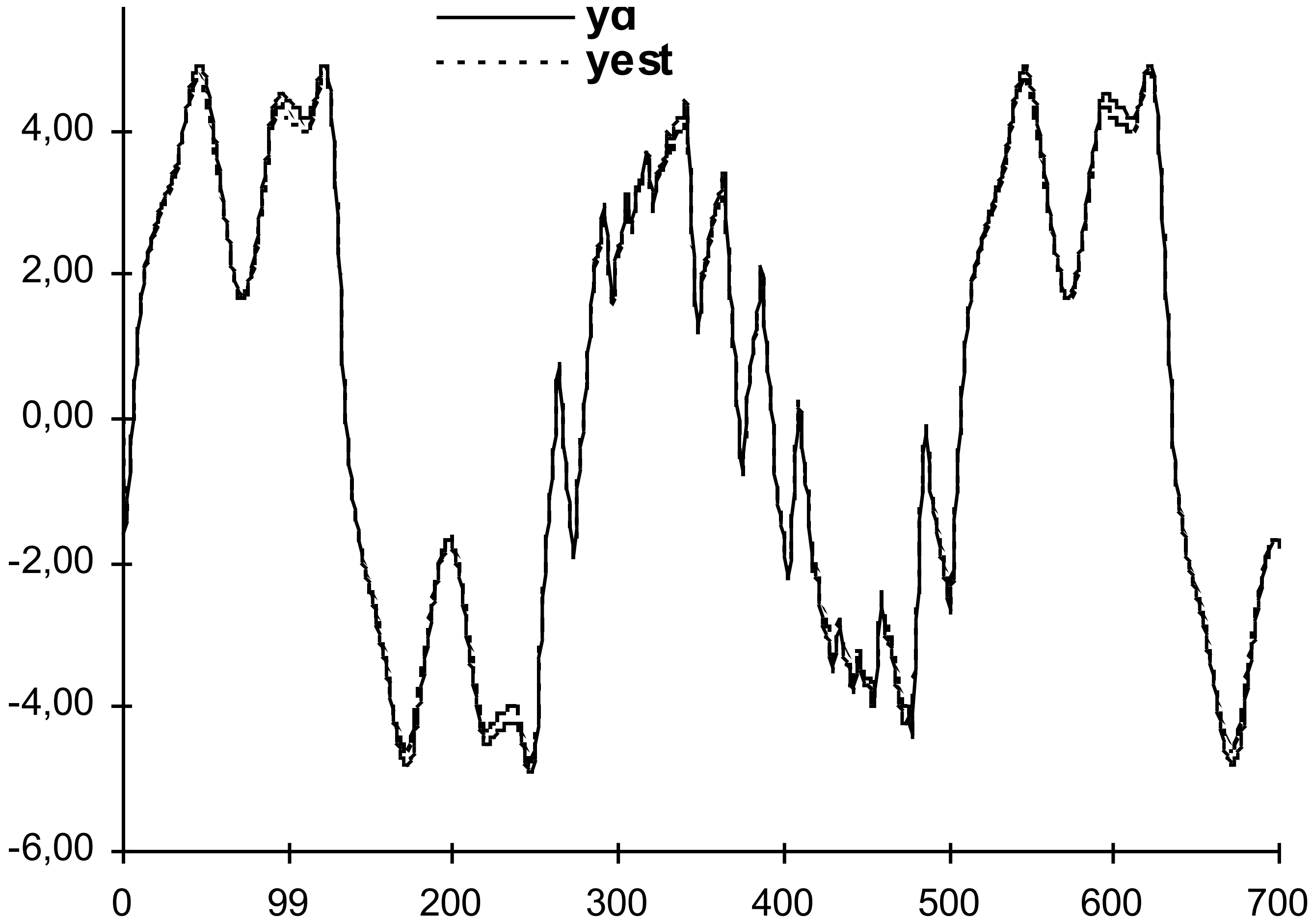
Resultado da identificação

$$u(k) = \mathrm{sen}(2.\pi.k / 250)$$



Resultado da identificação

$$u(k)=\begin{cases} \text{sen}(2.\pi.k/250), & 251 \le k \le 500 \\ 0.5\,\text{sen}(2.\pi.k/25)+0.5\,\text{sen}(2.\pi.k/250), & \text{caso contrário} \end{cases}$$



RESULTADOS DE IDENTIFICAÇÃO

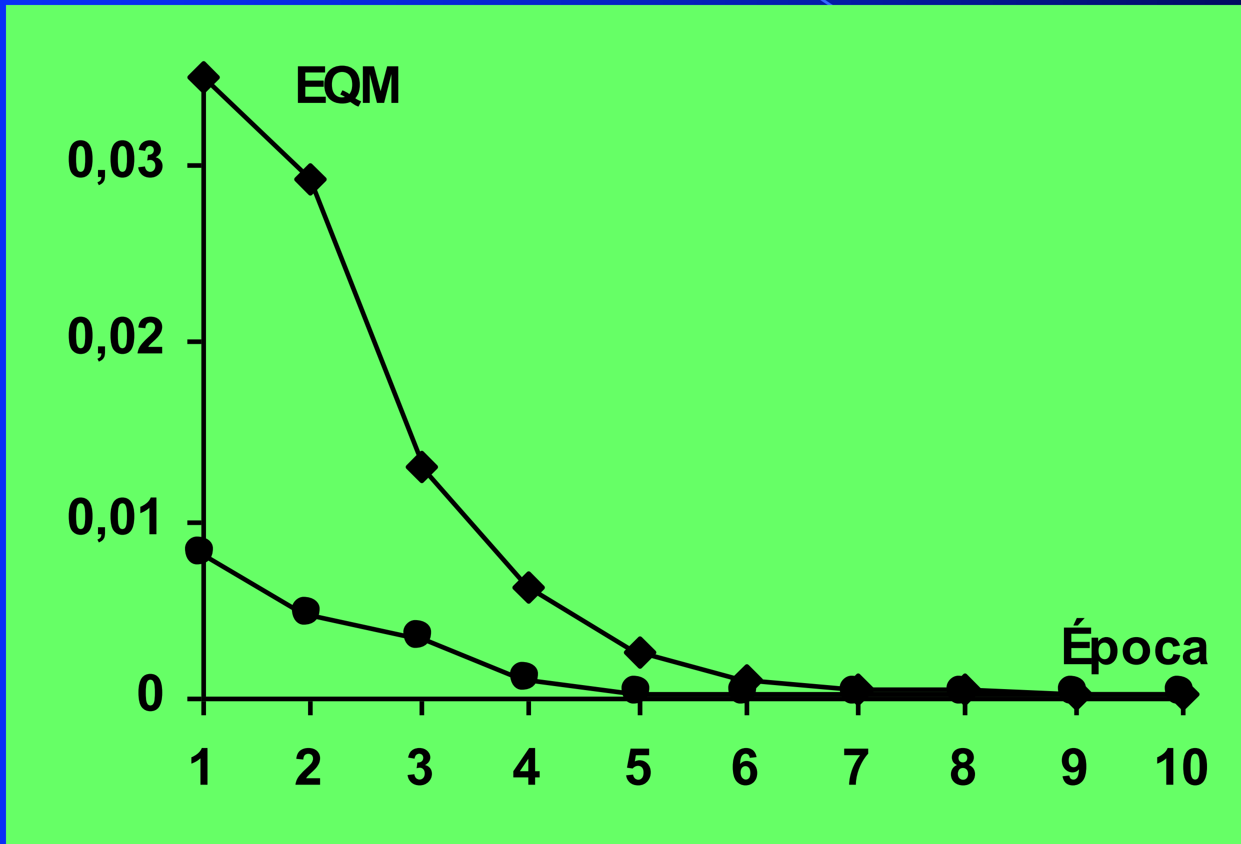
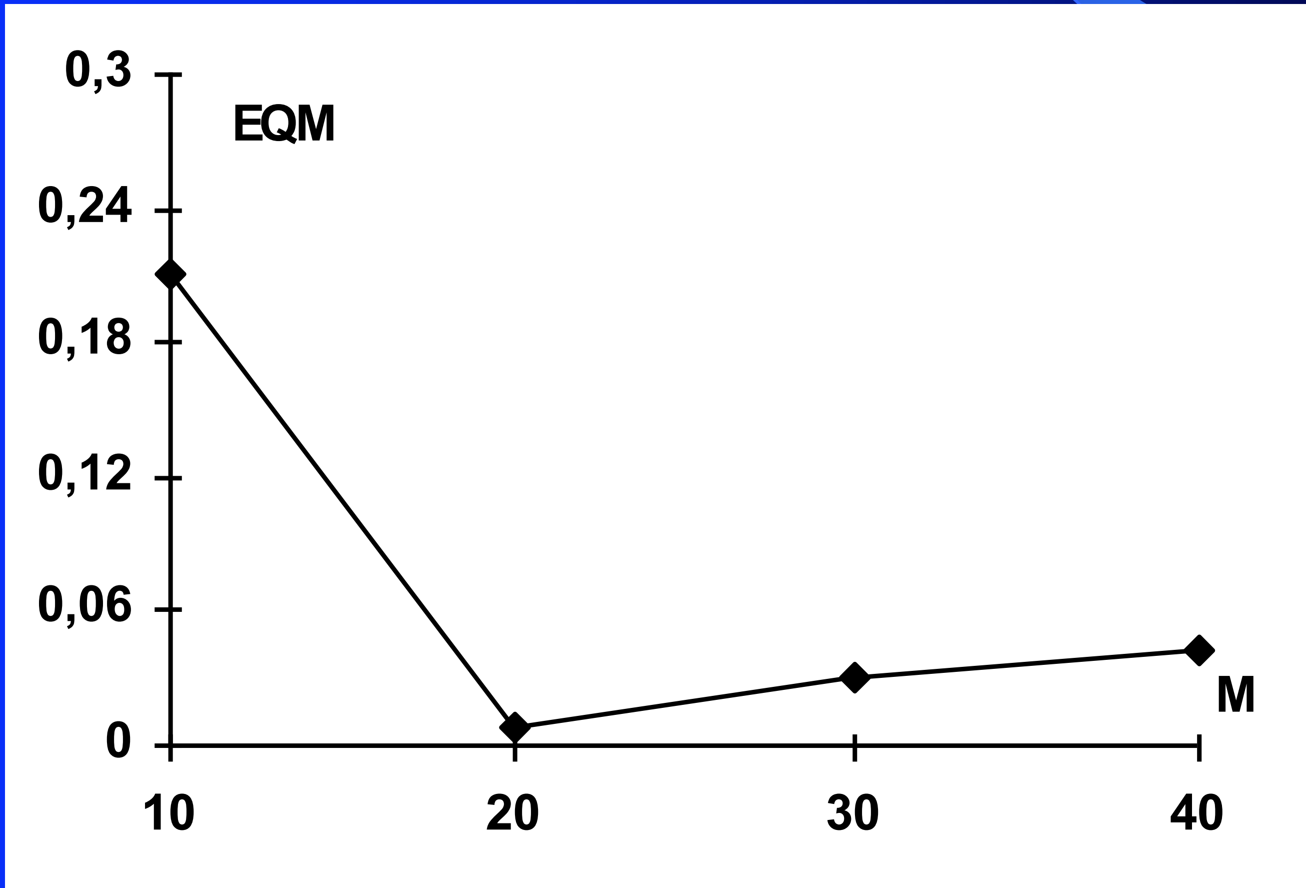


Figura 4 - Erro quadrático médio em função da época de treinamento para identificação usando NFN (•) e RNA (♦).

Influência do número de partições



Exemplo 2

$$y(k+1) = g\left[y(k), y(k-1)\right] + u(k)$$

$$g\left[y(k), y(k-1)\right] = \frac{y(k)\,y(k-1)\left[y(k)+2.5\right]}{1+y^2(k)+y^2(k-1)}$$

$$u(k) = \mathrm{sen}(2\pi k/25)$$

$$\hat{y}(k+1) = \hat{g}\left[\hat{y}(k), \hat{y}(k-1)\right] + u(k)$$

Estrutura do NFN

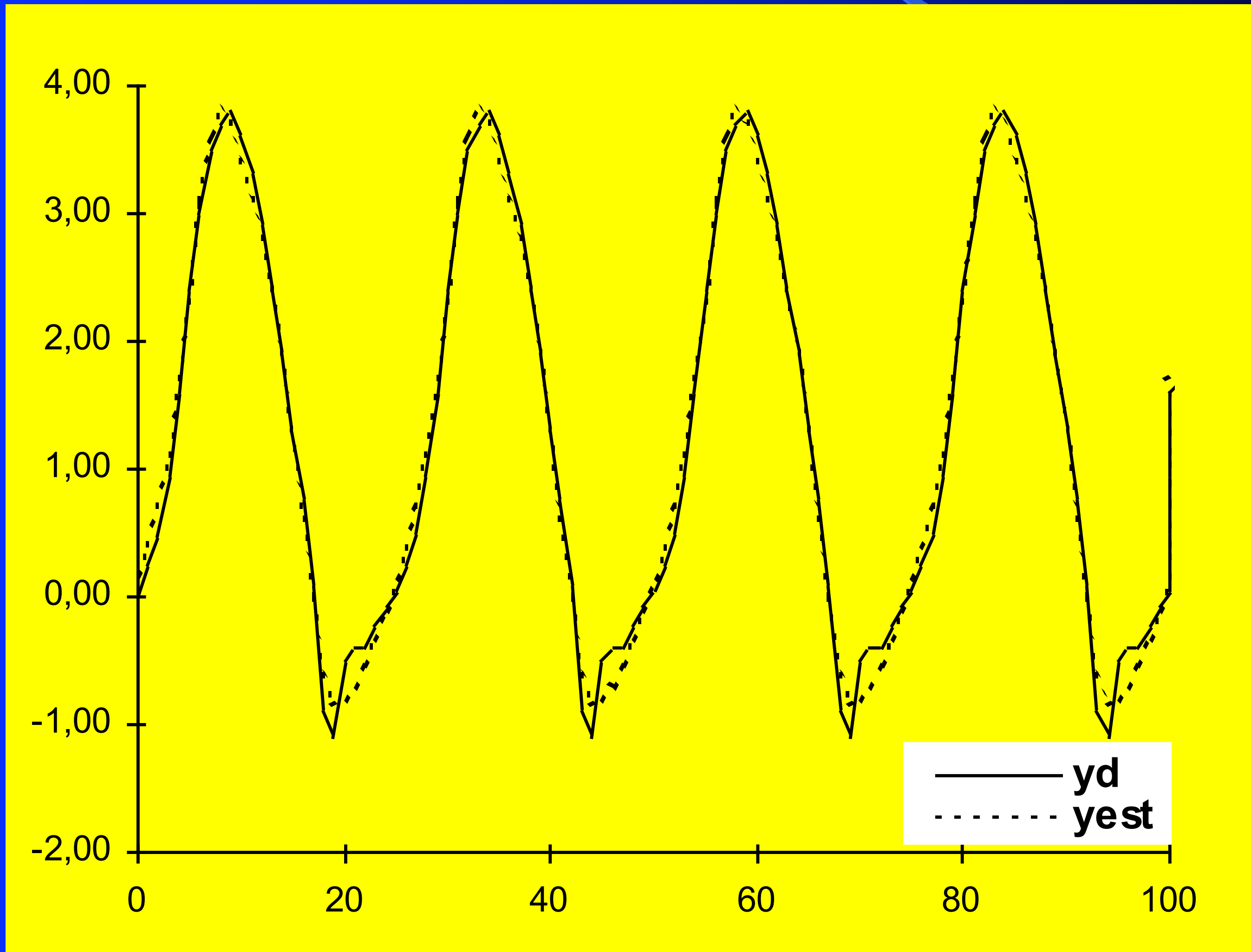
- Número de entradas: 2 (*y(k) e y(k-1)*);
- Número de saídas: 1
- Número de partições nebulosas: 5 por entrada

Metodologia de treinamento

- padrões de treinamento: 10.000 com u aleatório no intervalo [-1, +1]
- padrões de validação:

$$u(k) = \text{sen}\left(2.\pi.k / 250\right)$$

Resultado da identificação



Exemplo 3

$$y(k+1) = g\left[y(k), y(k-1), y(k-2), u(k), u(k-1)\right]$$

$$g(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 x_2 x_3 x_5 (x_3 - 1) + x_4}{1 + x_3^2 + x_4^2}$$

$$u(k) = \begin{cases} \text{sen}(2\pi k / 250), \forall \text{ k} \leq 500 \\ 0.8\,\text{sen}(2\pi k / 250) + 0.2\,\text{sen}(2\pi k / 25), \forall k > 500 \end{cases}$$

$$\hat{y}(k+1) = \hat{g}\left[\hat{y}(k), \hat{y}(k-1), \hat{y}(k-2), u(k), u(k-1)\right]$$

Estrutura do NFN

- Número de entradas: $y(k)$, $y(k\text{-}1)$, $y(k\text{-}2)$,
  $u(k)$ e $u(k\text{-}1)$

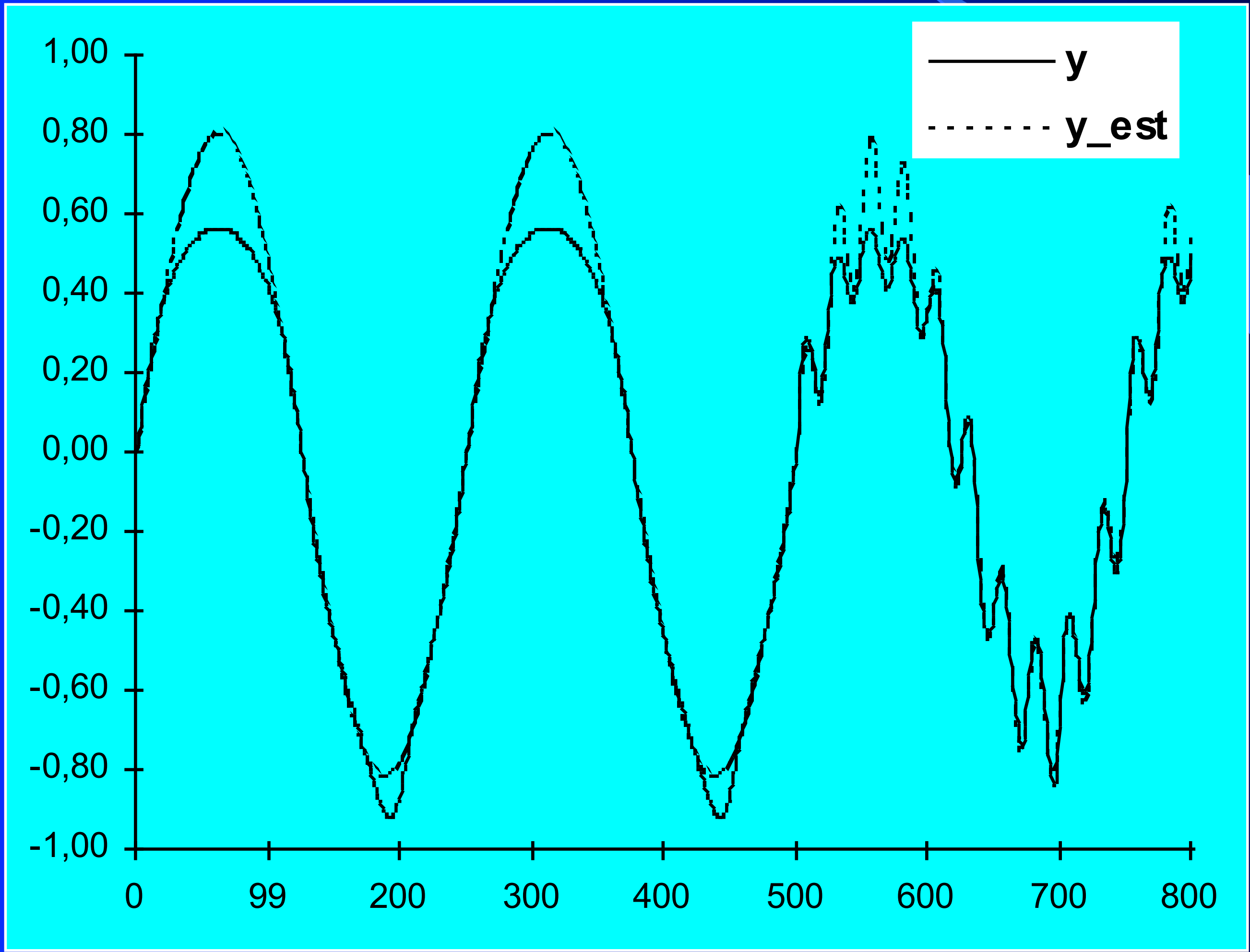
- Número de saídas: 1 ($y(k+1)$)

- Número de partições nebulosas: 3 para cada entrada

Metodologia de treinamento 1

- Foram utilizados cinco mil padrões com $u(k)$ gerado aleatoriamente no intervalo $[-1,1]$;

Resultado para Metodologia 1



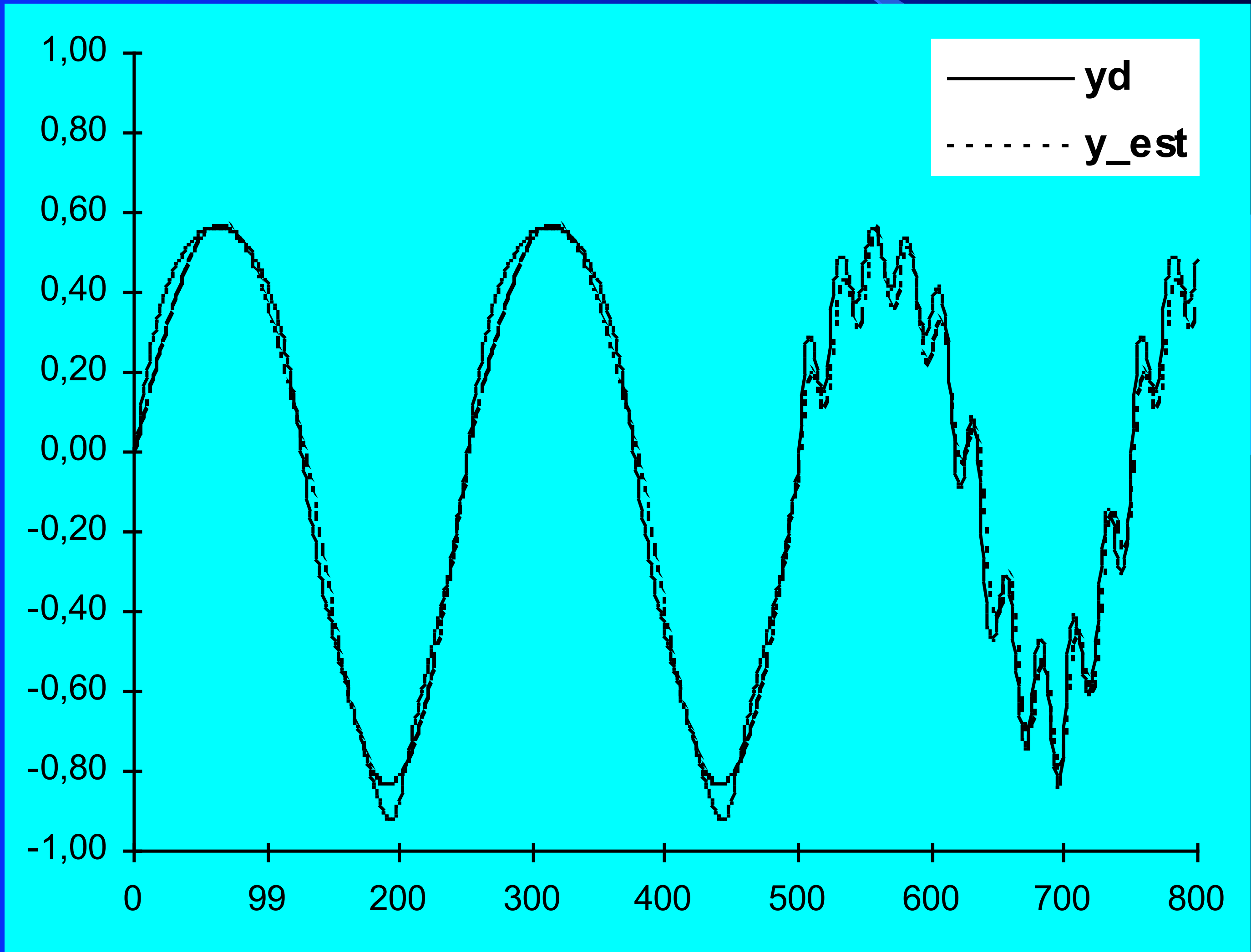
Metodologia de treinamento 2

- foram gerados duzentos e cinqüenta padrões com $u(k)$, determinado da seguinte forma:

$$u(k) = \mathrm{sen}\left(2\pi k / 250\right) para k = 1,2,\ldots,250$$

- Estes duzentos e cinqüenta padrões foram repetidos 20 vezes.

Resultado para Metodologia 2



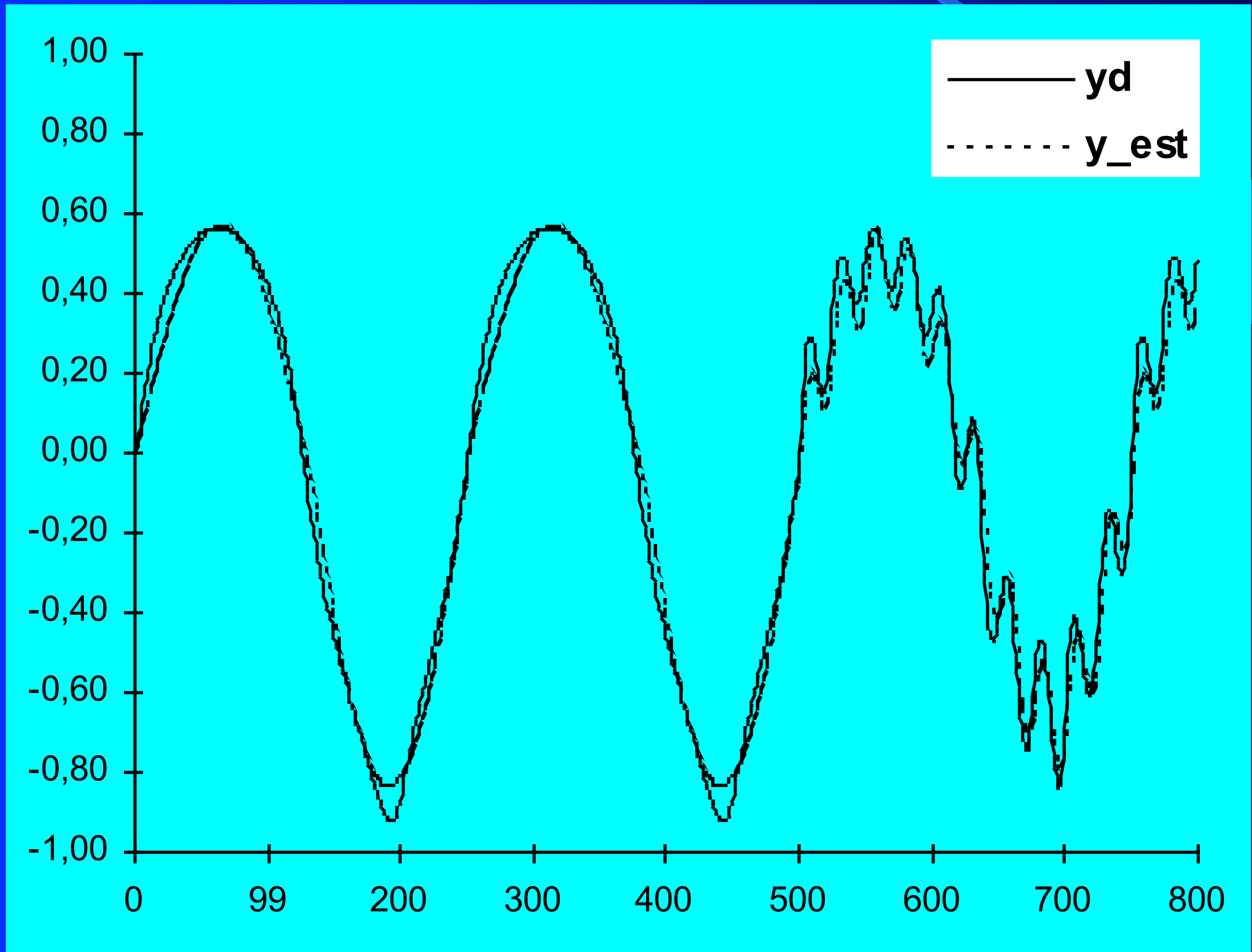
Metodologia de treinamento 2

- foram gerados cinco mil padrões com a entrada, $u(k)$, determinada da seguinte forma:

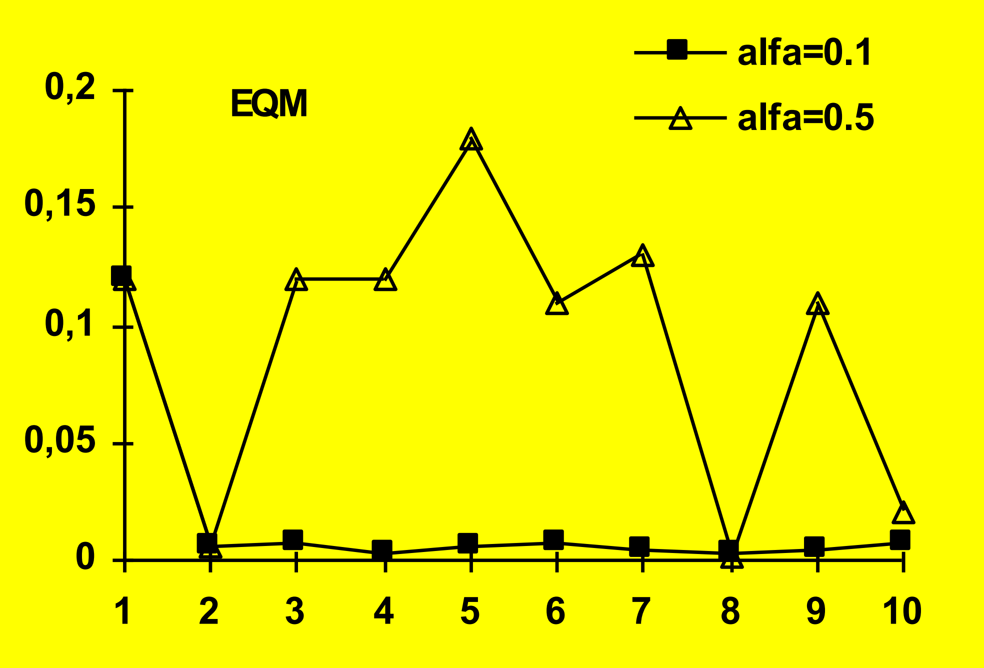
$$u(k) = a_1 \operatorname{sen}(2\pi k / b_1) + (1 - a_1), para k = 1,2,\ldots,5.000$$

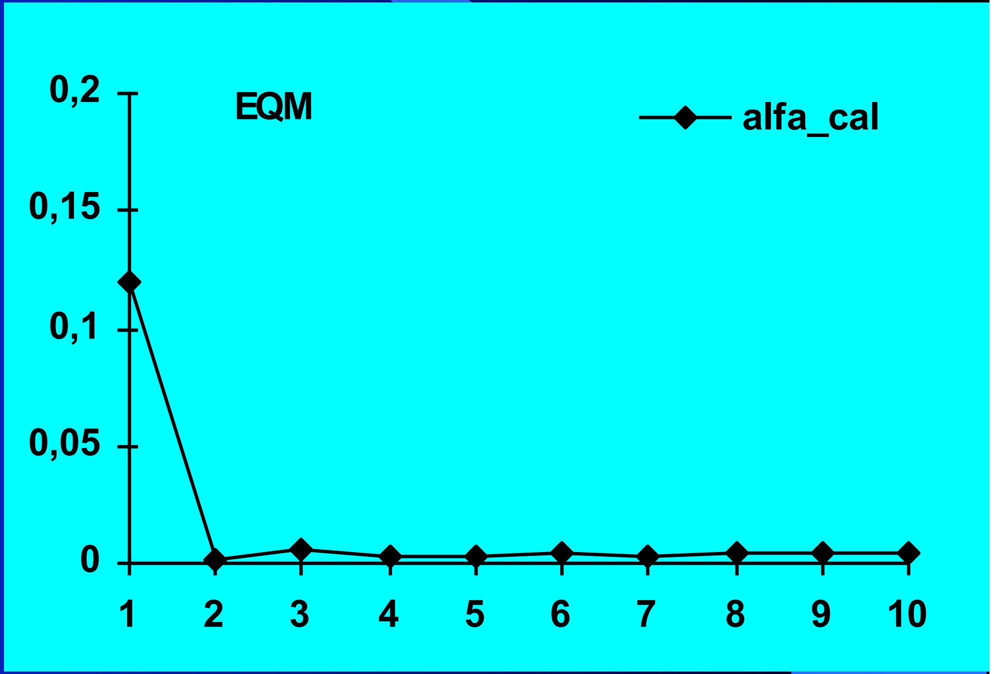
- sendo a_1, e b_1 números aleatórios com densidade de distribuição uniforme nos intervalos [0, 1] e [25, 250], respectiva-mente, alterados a cada 500 iterações.

Resultado para Metodologia 3



Influência da Taxa de Aprendizado





Taxa de aprendizado fixa

Taxa de aprendizado ótima

COMPLEXIDADE COMPUTACIONAL

$$y_l = \sum_{i=1}^{n} \mu_{ik_i}(x_i).w_{ik_i l} + \mu_{ik_i+1}(x_i).w_{i(k_i+1)l}$$

NFN

$$Y = \Gamma\left[W^3 \Gamma\left[W^2 \Gamma\left[W^1 X\right]\right]\right]$$

RNA

COMPLEXIDADE COMPUTACIONAL

Tabela 1 – Comparação entre o NFN e a RNA

Estrutura	Operações de X	Operações de + e/ou -	cálculos de funções
RNA [*n,p,q,m*]	*n.p + p.q + q.m*	*n.p + p.q + q.m*	*p+q+m (função sigmóide)*
NFN [*n,p,m*]	*2.n.m*	*n.m +n*	*n (função linear)*

Comparação entre o NFN e a RNA

Estrutura	Operações X	Operações +/-	Cálculo de Funções
RNA [2,10,20,1]	250	250	31 sigmóides
NFN [2,5,1]	10	20	2 funções lineares

Conclusões

- NFN é uma opção atraente em aplicações em tempo real e em treinamento *on-line;*
- **Alerta:** é possível utilizar estruturas simples, comparadas com RNA multicamadas, para identificar sistemas dinâmicos.

Algoritmo Neo-Fuzzy-Neuron (Generalização)

$$f_{li}(x_i)=\mu_{ik_i}(x_i).w_{ik_i l}+\mu_{i(k_i+1)}(x_i).w_{i(k_i+1)l}$$

$$y_l=\sum_{i=1}^{n}y_{li}(x_i)=\sum_{i=1}^{n}f_{li}(x_i)=f_{l1}(x_1)+f_{l2}(x_2)+\ldots+f_{ln}(x_n)$$

$$\varepsilon_{lt}=\frac{1}{2}\left(y_{lt}-y_{lt}^{d}\right)^{2}=\varepsilon_t\left(W_{ijl}\right), l=1,\ldots,m$$