
UM CONTROLADOR NEURAL *GAIN SCHEDULING* PARA PLANTAS NÃO-LINEARES

Carlos Andrey Maia

Departamento de Engenharia Elétrica, UFMG
Cx.P. 209, Belo Horizonte, MG, 30161-970
E-mail: maia@novell.cce.ufmg.br

e

Peterson Resende

Departamento de Engenharia Eletrônica,
UFMG
Cx.P. 209, Belo Horizonte, MG, 30161-970
E-mail: pr@cpdee.ufmg.br

RESUMO. Este artigo apresenta uma técnica de projeto de controlador neural via *gain scheduling* para plantas não-lineares. Em um primeiro estágio, um modelo neural é identificado para representar a planta não-linear em sua faixa dinâmica de operação. Em seguida, o modelo neural obtido é utilizado para determinar aproximações lineares em torno de vários pontos de operação da planta e os correspondentes controladores lineares são projetados. Finalmente, treina-se o controlador neural *gain scheduling* nestes pontos de operação de forma a representar os diversos controladores lineares. Simulações são apresentadas mostrando que a técnica descrita pode ser aplicada a uma ampla classe de sistemas dinâmicos não-lineares.

Palavras-chave: Redes Neurais, Controle não-linear, Linearização, *Gain Scheduling*.

A GAIN SCHEDULING NEURAL CONTROLLER FOR NONLINEAR PLANTS

ABSTRACT. This paper presents a gain scheduling neural controller design technique for nonlinear plants. In a first stage, a neural network model is identified to cover the dynamic range of the plant. In the following stage, using the neural network model, linear approximations are obtained about several operating points of the plant and corresponding linear controllers are designed. Finally, the gain scheduling neural controller is trained at these operating points in order to represent the linear controllers. Simulation results are presented, showing that the proposed technique can be used in a broad class of nonlinear dynamic systems.

Key words: Neural Networks, Nonlinear Control, Linearization, Gain Scheduling

Artigo submetido em 17/10/96

1a. Revisão em 27/02/97; 2a. Revisão em 20/08/97

3a. Revisão em 09/09/97; 4a. Revisão em 21/01/98

Aceito sob recomendação do Ed. Cons. Prof.Dr. Fernando A. Gomide

1 INTRODUÇÃO

As redes neurais artificiais, são de grande importância dentro da área de controle de sistemas não-lineares, devido suas características de mapeadores universais e da capacidade de aprender por treinamento (Cybenko, 1989; Hunt *et al.* 1992; Fukuda & Shibata, 1992). Nesse sentido, diversos trabalhos têm sido realizados: aplicações envolvendo controle adaptativo através da linearização do sinal de controle para sistemas SISO (Chen & Khalil, 1995), modelamento de sistemas NARMAX (Wang *et al.*, 1994) e controle por modelo interno (Hunt & Sbarbaro, 1991) são alguns dos muitos exemplos. Uma abordagem mais formal do problema de controle e identificação de sistemas dinâmicos não-lineares com vários exemplos interessantes é dada em Narendra & Parthasarathy (1990), onde redes neurais recorrentes com múltiplas camadas são utilizadas em modelos não-lineares SISO e MIMO. Um fato importante é que controladores neurais podem apresentar melhores resultados que controladores adaptativos convencionais lineares e não-lineares quanto aos aspectos de precisão e robustez do sistema em malha fechada (Boskovic & Narendra, 1995). Entretanto, nas técnicas discutidas acima, o controle neural é baseado na dinâmica inversa da planta. Assim, estas técnicas são limitadas a plantas fase-mínima.

Por outro lado, a metodologia de controle por *gain scheduling* tem se mostrado de grande utilidade em muitas aplicações de engenharia (Shamma & Athans, 1990, 1992; Rugh, 1991). Esta técnica consiste na obtenção de modelos linearizados em torno de cada ponto de operação da planta, para os quais são projetados controladores lineares utilizando técnicas convencionais de controle linear. A seguir, um controlador não-linear global é obtido pelo 'escalonamento' ou interpolação dos ganhos obtidos para cada ponto de operação. As principais vantagens desta técnica são a simplicidade e a generalidade: utiliza técnicas convencionais de controle linear e é aplicável a uma grande variedade de plantas não-lineares.

A utilização de redes neurais para a implementação de controladores escalonados foi proposta por Ahmed & Tasadduq (1994) que utilizam redes com neurônios especiais (representados por dois neurônios simples que funcionam em paralelo sendo que sua saída é a diferença entre as saídas dos dois neurônios), e por Maia & Resende (1996), que utilizam redes neurais multicamadas *feedforward* treinadas com o algoritmo de retropropagação (RNAMR) com neurônios convencionais e com pré-processamento. Nesses artigos, para o treinamento do controlador neural global, são projetados controladores lineares a cada período de amostragem.

No presente trabalho, são utilizadas redes neurais para a implementação de controladores *gain scheduling* para plantas não-lineares SISO. São selecionados diversos pontos de operação de modo a cobrir a faixa de dinâmica da planta. Estes pontos de operação são determinados pelos valores de referência. Para cada um destes pontos de operação, são obtidas aproximações linearizadas invariantes no tempo da planta, a partir das quais são projetados os controladores lineares correspondentes. Uma rede neural é então treinada para constituir um controlador global de modo a substituir todos os controladores lineares. Uma efetiva interpolação entre os pontos de operação é também atingida (Maia, 1996).

2 MODELAGEM NEURAL DA PLANTA

Para a obtenção do controlador neural *gain scheduling*, é necessário, primeiramente, determinar um modelo da planta adequado para geração de aproximações lineares. Caso este modelo não esteja disponível, é mostrada, nesta seção, uma maneira de construção do modelo, através de uma identificação neural da planta. Neste sentido, é necessária a criação de um conjunto de dados que descrevam a relação entrada - saída em toda a faixa dinâmica da planta. A seguir, será feita uma breve discussão a este respeito.

O problema de identificação consiste basicamente na seleção de uma estrutura de modelo e no ajuste adequado de seus parâmetros de forma a se otimizar uma determinada função de custo que avalia a proximidade do modelo à planta. Para a identificação neural, a estrutura da rede deve ser escolhida previamente e os parâmetros da mesma são ajustados por um algoritmo de busca que procura minimizar uma função de custo.

Seja uma planta não-linear SISO dada pela seguinte relação:

$$y(t+1) = f\{\phi(t)\} \quad (1)$$

onde $y(t)$ é a saída da planta e $\phi(t)$ é um vetor de dimensão n_ϕ composto da saída e da entrada em diversos instantes discretos, sendo $f(\cdot)$ uma função não-linear que representa a relação dinâmica. Assim, a equação (1) pode ser dada pela representação discreta:

$$y(t+1) = f\{y(t), \dots, y(t-n+1), u(t), \dots, u(t-\ell+1)\} \quad (2)$$

onde n e ℓ são inteiros positivos.

Dessa forma, uma RNAMR pode ser treinada (Rumelhart *et al.*, 1986) de modo a representar a função $f(\cdot)$. A rede neural utilizada pode ter múltiplas camadas, funções de ativação não-lineares e termos de polarização. No entanto, pode-se provar que, sob certas condições, uma RNAMR com uma única camada escondida com número suficientes de neurônios tem capacidade de aproximar qualquer função contínua de

múltiplas variáveis reais em um domínio compacto (Cybenko, 1989). Uma forma de representação dessa rede é a seguinte:

$$\hat{y}(t+1) = NN\{\mathbf{W}, \phi(t)\} \quad (3)$$

na qual \hat{y} é a saída da rede neural e \mathbf{W} contém todos os conjuntos de pesos da rede. Neste caso, $\phi(t)$ fornece o padrão de entrada da rede neural. Esta equação pode ser reescrita como:

$$\hat{y}(t+1) = NN\{\mathbf{W}, y(t), \dots, y(t-n+1), u(t), \dots, u(t-\ell+1)\} \quad (4)$$

Para o ajuste dos pesos da rede define-se o erro entre as saídas da planta e do modelo neural identificado como sendo:

$$e(t+1) = y(t+1) - \hat{y}(t+1) \quad (5)$$

A RNAMR deve ser treinada de forma a se obter uma boa aproximação, dentro do domínio de operação da planta. Assim, $e(t+1)$ deve ser sempre suficientemente pequeno. Essa tarefa exige que os dados de treinamento da rede sejam gerados de forma adequada, ou seja, deve-se excitar a planta com um sinal de controle suficientemente 'rico' para que se possa modelar razoavelmente sua dinâmica.

3 PROJETO DO CONTROLADOR

O procedimento para a obtenção do controlador neural *gain scheduling* proposto é desenvolvido em três etapas:

- Obtenção de aproximações lineares invariantes no tempo para diversos pontos de operação da planta.
- Projeto de controladores lineares baseado nas aproximações lineares.
- Treinamento de um controlador neural global em substituição aos controladores lineares nos pontos de operação correspondentes.

É importante observar que todas as etapas são desenvolvidas *off-line*, gerando o controlador neural *gain scheduling* para a planta não linear. Estas etapas são descritas a seguir.

3.1 Aproximações lineares

De posse do modelo neural da planta não-linear determinado na seção 2, pode-se obter aproximações linearizadas para cada ponto de operação da planta. É assumido que a planta seja estável em malha aberta ou estabilizável em malha fechada e que a função $f(\cdot)$, em (2), seja de classe C^1 (contínua e com primeiras derivadas parciais contínuas). Uma aproximação linearizada da planta pode ser representada por:

$$A(q^{-1})\delta y(t+1) = B(q^{-1})\delta u(t+1) \quad (6)$$

onde q é o operador avanço, sendo $\delta y(t) = y(t) - y^*$ e $\delta u(t) = u(t) - u^*$. Sendo que y^* é o valor de referência e u^* é a entrada da planta correspondente em regime estacionário (O apêndice 1 descreve uma maneira simples de se determinar u^* a partir do modelo neural da planta). Os operadores $A(q^{-1})$ e $B(q^{-1})$ são representados pelos polinômios:

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_n q^{-n} \\ B(q^{-1}) &= b_1 q^{-1} + \dots + b_\ell q^{-\ell} \end{aligned} \quad (7)$$

na qual $a_i \equiv -\frac{\partial \hat{y}(t+1)}{\partial \hat{y}(t-i+1)}$ e $b_j \equiv \frac{\partial \hat{y}(t+1)}{\partial u(t-j+1)}$ para $i=1, \dots, n$; e $j=1, \dots, \ell$.

Os parâmetros a_i e b_j são calculados facilmente a partir da equação (4), que descreve o modelo neural, através da regra da cadeia, da forma descrita no apêndice 2. Estes parâmetros dependem dos pesos, das funções de ativação e das entradas do modelo neural da planta.

Para se garantir, em regime estacionário, erro nulo entre a saída da planta e o valor de referência, é introduzido um integrador na entrada da planta. Assim, $u(t)$ passa a ser obtido pela integração da variável $v(t)$, ou seja:

$$u(t+1) = u(t) + v(t) \quad (8)$$

Desta forma, a equação (6) pode ser manipulada algebricamente, obtendo-se o seguinte modelo incremental:

$$A(q^{-1})\delta y(t+1) = A(q^{-1})\delta y(t) + B(q^{-1})v(t) \quad (9)$$

3.2 Controladores lineares

A partir das aproximações linearizadas obtidas na seção 3.1, pode-se projetar controladores lineares adequados para os respectivos pontos de operação da planta. Para este fim, foi utilizada técnica do regulador linear quadrático (Anderson & Moore, 1990), sendo a função de custo dada por:

$$J = \sum_0^{\infty} \{ Q \delta y(t)^2 + R v(t-1)^2 \} \quad (10)$$

Sendo as ponderações, $R > 0$ e $Q \geq 0$, escalares. Obviamente, com essa função de custo pondera-se, não só o erro entre a saída da planta e o valor de referência, como também as variações do sinal de controle (ponderando-se $v(t)$ de forma adequada pode-se evitar grandes variações indesejadas em $u(t)$).

Um esquema que representa o controle da planta, a partir dos controladores lineares projetados, é mostrado na figura 1, onde o bloco 'controle' inclui também os atrasos requeridos. Neste esquema o controlador linear é selecionado pelo valor de referência y^* .

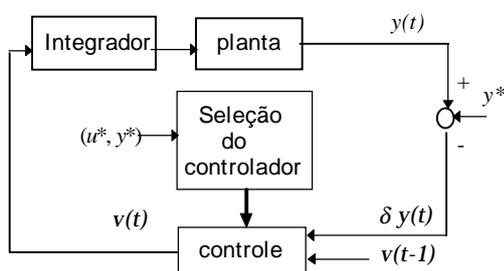


Figura 1. Esquema para o controle linearizado da planta.

3.3 O controlador neural gain scheduling

Nesta seção, é determinado um controlador neural global que representa os diversos controladores lineares projetados na seção 3.2, no sentido de implementar a técnica de *gain scheduling*. Assim, de acordo com o esquema da figura 1, para os diversos pontos de operação e os controladores lineares correspondentes, são gerados dados representativos de suas relações entrada - saída. Estes dados são então utilizados para o

treinamento do controlador neural *gain scheduling*. Cumpre observar que o controlador neural resultante deve também fornecer interpolações adequadas entre os pontos de operação treinados.

O controlador neural *gain scheduling* deve representar uma relação dinâmica da forma:

$$v(t) = g\{\delta y(t), \dots, \delta y(t-n), v(t-1), \dots, v(t-m), y^*\} \quad (11)$$

sendo que m é um inteiro positivo e n é dado na equação (2).

É importante observar que um pré-processamento do padrão de entrada da rede é necessário, pois $v(t)$ deve ser zero quando $\delta y(t), \dots, \delta y(t-n), v(t-1), \dots, v(t-m)$ são nulos (devido à condição de erro nulo em regime estacionário, assegurada pela introdução de integrador na entrada da planta). Isso é conseguido definindo o vetor:

$$z = [z_1^T, z_2^T]^T \quad (12)$$

onde $z_1 = [\delta y(t), \dots, \delta y(t-n), v(t-1), \dots, v(t-m)]^T$ e $z_2 = y^*$. O pré-processamento pode ser da seguinte forma:

$$\tilde{z} = T(z) \quad (13)$$

sendo T uma transformação contínua de $\mathcal{R}^L \rightarrow \mathcal{R}^L$ e $L=n+m+2$, satisfazendo as condições:

a) $T([0^T, z_2^T]^T) = [0^T, 0^T]^T$

b) $T([z_1^T, z_2^T]^T)$ deve ser unívoca sempre que $\|z_1\| \neq 0$

Um mapeamento simples que atende a essas exigências é:

$$T([z_1^T, z_2^T]^T) = [z_1^T, z_2 \alpha(z_1)]^T \quad (14)$$

onde a função $\alpha(\cdot)$ é definida como

$$\alpha(z_1) = z_1^T M z_1 \quad (15)$$

sendo M é uma matriz $(n+m+1) \times (n+m+1)$ definida positiva. Uma vez conseguida a transformação T , os dados de treinamento são pré-processados, obtendo-se assim \tilde{z} (Eq. 13). Feito isso, a rede neural é então treinada para aprender o mapeamento não-linear $\tilde{z} \rightarrow v(t)$.

O esquema do treinamento do controlador neural *gain scheduling* é mostrado na figura 2. Neste esquema, o treinamento da rede é feito da forma usual através do algoritmo de retropropagação do erro (Rumelhart *et al.*, 1986). Desta forma, é escolhida uma RNMR sem termos de polarização e com uma função de ativação tangente hiperbólica.

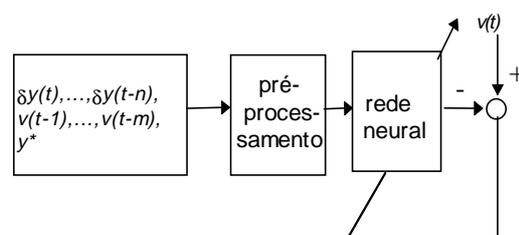


Figura 2. Esquema do treinamento do controlador neural *gain scheduling*

Após o treinamento da rede, o controlador neural *gain scheduling* obtido é então utilizado para controlar a planta, como mostrado na estrutura dada na figura 3 (onde o bloco

'controlador neural *gain scheduling*' inclui atrasos, pré-processamento e a rede neural treinada).

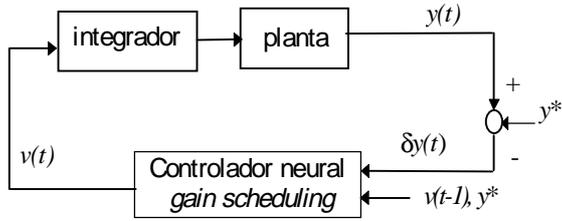


Figura 3. Controle neural *gain scheduling* da planta

4 RESULTADOS DE SIMULAÇÕES

A seguir, a técnica de projeto de controladores neurais, apresentada nos itens anteriores, é aplicada a duas plantas discretas não-lineares. Os resultados obtidos são descritos passo-a-passo.

Planta 1

Considere a planta não-linear dada pela seguinte equação dinâmica discreta:

$$y(t+1) = \frac{0,9y(t) + u(t)}{1 + y(t)^2} \quad (16)$$

Na qual y é a saída da planta e u , a entrada e t , o tempo discreto. É fácil ver que o modelo neural desta planta tem a seguinte estrutura $\hat{y}(t+1) = NN\{W, y(t), u(t)\}$. Para a identificação deste, foi utilizada uma rede neural com seis neurônios na camada escondida e um neurônio na camada de saída (todos neurônios sem termo de polarização e com função de ativação tangente hiperbólica). Os dados de treinamento foram obtidos excitando a planta com um sinal de entrada $u(t)$ variando aleatoriamente entre $-0,6$ e $0,6$.

O projeto dos controladores lineares, para a planta em questão, foi feito escolhendo-se as ponderações da função de custo, equação (10), como sendo $R=30$ e $Q=1$.

A figura 4 mostra a resposta do sistema para os controladores lineares projetados para alguns valores de referência. Estes valores servirão de base para a avaliação do desempenho do controlador neural, já que não serão utilizados no treinamento.

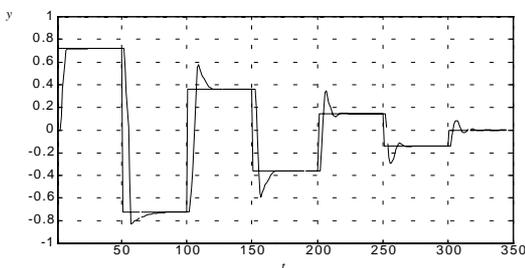


Figura 4. Controle do sistema linearizado em alguns pontos de operação

Para a obtenção de dados de treinamento do controlador neural, foram gerados aleatoriamente dezenas de valores de referência na faixa de $-0,8$ a $0,8$ e os respectivos controladores lineares foram projetados. Estes valores de referência, os correspondentes sinal de controle e saída da planta obtidos são mostrados na figura 5

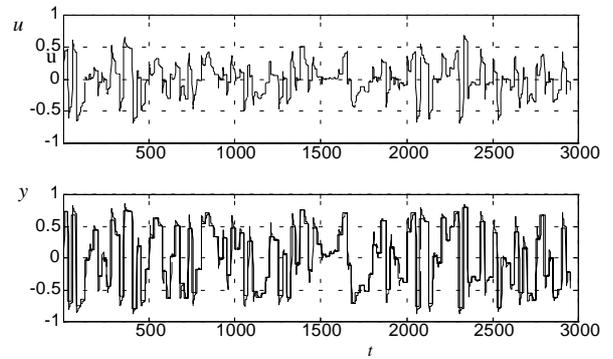


Figura 5 Dados gerados para o treinamento do controlador neural

Com os dados de entrada e saída do sistema em malha fechada (planta controlada) e das respectivas referências, faz-se então o pré-processamento de acordo com o mostrado na figura 2, sendo $m=0$ e $n=1$, utilizando a transformação (14-15) e usando uma matriz de ponderação, M , igual a $=0,II$, sendo I uma matriz identidade 2×2 . O controlador neural então é treinado conforme o diagrama mostrado na mesma figura. Para esse fim, utilizou-se uma RNAMR com duas camadas, 6 neurônios na camada escondida e um na camada de saída. A figura 6 mostra a saída da planta quando o controlador neural é utilizado em substituição aos controladores lineares nos diversos pontos de operação, conforme o esquema descrito na figura 3.

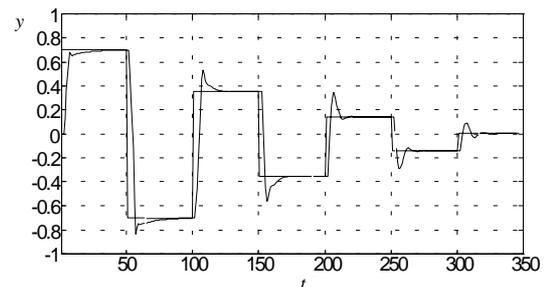


Figura 6 Resposta do sistema utilizando o controlador neural *gain scheduling*

Para observar a capacidade de generalização do controlador neural, treinado com os dados da figura 5, é mostrado na figura 6 a resposta da planta utilizando o controle neural fornecendo resultados bastante próximos aos da figura 4 cujos dados não foram utilizados no treinamento.

Planta 2

A segunda planta não-linear utilizada nas simulações é descrita pela seguinte equação:

$$y(t+1) = 1,4y(t) - 0,6y(t-1) + u^3(t) - 2u(t) + u^3(t-1) - 2u(t-1) \quad (17)$$

Neste exemplo, a não-linearidade está presente somente no sinal de controle. É importante observar que o modelo da planta apresenta fase não-mínima uma vez que possui dinâmica de zeros instável (Slotine & Weiping, 1990) quando restringimos a saída da planta a zero (isto pode ser verificado considerando, na equação (15), uma entrada auxiliar definida como $u_{aux}(t) = u(t)^3 + u(t)$). Por esse motivo, não possui dinâmica inversa estável. É fácil ver que o modelo neural da planta deve ter a seguinte estrutura:

$$\hat{y}(t+1) = NN\{\mathbf{W}, y(t), y(t-1), u(t), u(t-1)\} \quad (18)$$

A identificação desse modelo foi feita utilizando uma rede neural com seis neurônios na camada escondida e um neurônio na camada de saída (todos neurônios sem termo de polarização e com função de ativação tangente hiperbólica). Os dados de treinamento foram obtidos excitando a planta com um sinal $u(t)$ variando aleatoriamente entre -1 e 1.

Para o projeto dos controladores lineares, as ponderações escolhidas para a função de custo foram: $R = 2,5 \times 10^3$ e $Q = 1$.

A figura 7 mostra a resposta do sistema em malha fechada para os controladores lineares projetados para alguns pontos de operação (como já foi dito para a Planta 1, este gráfico será utilizado posteriormente para a avaliação do desempenho do controlador neural).

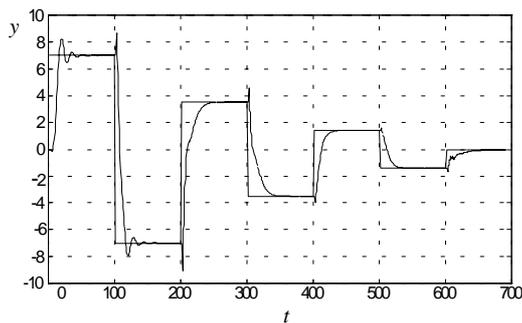


Figura 7. Resposta da planta utilizando controle linear em alguns pontos de operação.

Para a obtenção dos dados de treinamento do controlador neural foram gerados aleatoriamente dezenas de valores de referência na faixa de -7 a 7 e os respectivos controladores lineares foram projetados. Estes valores de referência, os correspondentes sinal de controle e saída da planta obtidos são mostrados na figura 8.

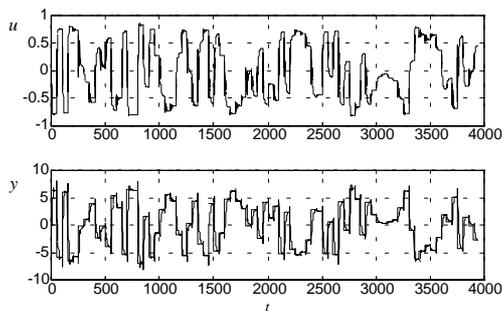


Figura 8. Dados gerados para o treinamento do controlador neural.

Com os dados de entrada do integrador, de saída da planta e dos valores de referência, é feito o pré-processamento através do vetor z definido em (12), sendo $m=1$ e $n=2$, utilizando a transformação (14)-(15) e usando uma matriz de ponderação $M=0,1I$ (sendo I uma matriz identidade 4×4). O controlador neural *gain scheduling* é então treinado conforme foi mostrado na figura 2. Utilizou-se uma RNAMR, de duas camadas com seis neurônios na camada escondida, sem termos de polarização e com função de ativação tangente hiperbólica.

Após o treinamento, o controlador neural *gain scheduling* é utilizado para controlar a planta segundo a estrutura mostrada na figura 3. A resposta do sistema em malha fechada é mostrada na figura 9 para alguns valores de referência.

Comparando estes resultados com a figura 7 (não treinada), pode-se ver que o controlador neural aprendeu a dinâmica global dos controladores lineares.

O controlador proposto tem sido aplicado com sucesso em vários tipos de plantas. Em especial, tem se mostrado efetivo no controle de plantas não-lineares de fase não-mínima, onde técnicas baseadas na dinâmica inversa seguramente não teriam sucesso.

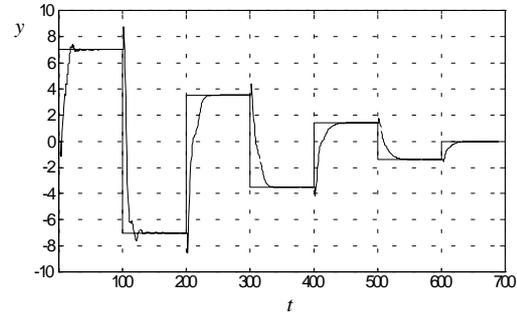


Figura 9. Resposta da planta utilizando controlador neural *gain scheduling*.

5 CONCLUSÃO

Este trabalho apresentou uma técnica de projeto de controlador neural via *gain scheduling* para plantas não-lineares SISO. O esquema apresentado não é adequado para plantas MIMO, neste caso uma abordagem diferente de controle neural via *gain scheduling* é proposta em Maia & Resende (1997). Através de exemplos, os resultados de simulações mostram que o controlador neural proposto pode representar a dinâmica global dos controladores lineares projetados para diversos pontos de operação da planta.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- Ahmed, M.S.; Tasadduq, I.A. (1994). Neural net controller for nonlinear plants: design approach through linearisation, *IEE Proc. Control Theory Appl.*, **141**, 315-322.
- Anderson, B.D.O.; Moore, J.B. (1990). *Optimal Control: Linear Quadratic Methods*, Prentice Hall, New Jersey.
- Boskovic, J.D.; Narendra K. (1995). Comparison of linear, nonlinear and neural network based adaptive controllers for a class of fed-batch fermentation processes, *Automatica*, **31**, 817-840.
- Chen, F.C.; Khalil, H.K. (1995). Adaptive Control of a Class of Nonlinear Discrete-Time Systems Using Neural Networks, *IEEE Trans. Automat. Contr.*, **40**, 791-801.
- Cybenko, G. (1989). Approximation by a superposition of a sigmoidal function, *Math. Control Signal Systems*, **2**, 303-314.
- Fukuda, T.; Shibata, T. (1992). Theory and applications of neural networks for industrial control systems, *IEEE Trans. on Industrial Electronics*, **39**, 472-489.
- Hunt, K.J.; Sbarbaro, D. (1991). Neural Networks for nonlinear internal model control, *IEE Proc., Part D*, **138**, 431-438, .

- Hunt, K.J.; Sbarbaro, D.; Zbikowski, R.; Gawthrop, P.J. (1992). Neural Networks for Control Systems - A Survey, *Automatica*, **28**, 1083-1112.
- Maia, C. A.; Resende, P. (1997). Neural control of MIMO nonlinear plants: a gain scheduling approach, *Proceedings of the 1997 IEEE International Symposium on Intelligent Control*, Istanbul, 193-198.
- Maia, C. A.; Resende, P. (1996). Projeto de controladores neurais para plantas não-lineares via linearização, *Anais do XI Congresso Brasileiro de Automática*, S. Paulo, 1561-1566.
- Maia, C. A. (1996). *Controle neural de plantas não-lineares via linearização*. Dissertação de Mestrado, PPGEE, UFMG
- Narendra, K.; Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks, *IEEE Trans. Neural Networks*, **1**, 4-27.
- Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. (1986). Learning internal representations by error propagation, in *Parallel distributed processing*, MIT Press.
- Rugh, W. J. (1991). Analytical framework for gain scheduling, *IEEE Control Syst. Mag.*, **11**, 79-84.
- Shamma, J. S.; Athans, M. (1990). Analysis of gain scheduled control for nonlinear plants, *IEEE Trans. Automat. Contr.*, **35**, 898-907.
- Shamma, J. S.; Athans, M. (1992). Gain Scheduling: potential hazards and possible remedies, *IEEE Control Syst. Mag.*, **12**, 101-107.
- Slotine, J.J.E.; Weiping, L. (1990). *Applied Nonlinear Control*, Prentice Hall, New Jersey.
- Wang, H.; Brown, H.; Harris, C.J. (1994). Neural network modeling of unknown nonlinear systems subject to immeasurable disturbances, *IEE Proc. Control Theory Appl.*, **141**, 216-222.

APÊNDICE 1. DETERMINAÇÃO DE u^*

Seja o modelo neural da planta dado pela equação (4). O objetivo então é encontrar um vetor u^* tal que $y^* = NN\{\mathbf{W}, \hat{y}(t), \dots, \hat{y}(t-n+1), u(t), \dots, u(t-l+1)\}$ quando $\hat{y}(t-i+1) = y^*$ e $u(t-j+1) = u^*$, para $i = 1, \dots, n$ e $j = 1, \dots, l$. Uma forma simples de obtenção de u^* correspondente a y^* , a partir do modelo neural, é dada a seguir.

Para um dado sinal u_k qualquer, define-se:

$$\tilde{y} = NN\{\mathbf{W}, \hat{y}(t), \dots, \hat{y}(t-n+1), u(t-1), \dots, u(t-l+1)\} \Big| \hat{y}(t-i+1) = y^* \text{ e } u(t-j+1) = u_k \quad (19)$$

Para a busca de u^* , é necessária a definição de uma função de erro entre a saída do modelo dado pela equação (19) e o valor desejado para a mesma, y^* . Uma definição simples é a seguinte:

$$E = \frac{1}{2} \|y^* - \tilde{y}\|^2 \quad (20)$$

Logo, o objetivo se resume em encontrar um sinal u_k tal que o erro quadrático definido na equação (20), seja menor do que ε , a precisão desejada. Esse objetivo pode ser alcançado utilizando um método de busca qualquer. Por exemplo, com o método do gradiente, tem-se a seguinte equação recursiva de busca:

$$u_{k+1} = u_k + \mu \left. \frac{\partial \tilde{y}}{\partial u} \right|_{(y^*, u_k)} \times (y^* - \tilde{y}) \quad (21)$$

Sendo μ uma constante.

APÊNDICE 2. DETERMINAÇÃO DOS COEFICIENTES DAS APROXIMAÇÕES LINEARES

Considerando uma RNAMR com duas camadas, uma escondida e outra de saída e função de ativação tangente hiperbólica, a equação (4) pode ser escrita explicitamente utilizando as seguintes equações:

$$\mathbf{h} = \tanh(\mathbf{W}_h \mathbf{p} + \boldsymbol{\delta}_h) \quad (22)$$

$$\bar{y}(t+1) = \tanh(\mathbf{W}_o + \boldsymbol{\delta}_o)$$

$$\hat{y}(t+1) = k\bar{y}(t)$$

Nas quais $\mathbf{p} = [y(t), \dots, y(t-n+1), u(t), \dots, u(t-l+1)]^T$, \mathbf{W}_h e \mathbf{W}_o são matrizes formadas com os pesos da camada escondida e de saída respectivamente. $\boldsymbol{\delta}_h$ e $\boldsymbol{\delta}_o$ são os termos de polarização, \mathbf{h} é o vetor de entrada da camada de saída e $\bar{y}(t)$ é a saída normalizada. Sendo k um fator de escala tal que $-0,9 < \bar{y}(t) < 0,9$.

Calculando as derivadas parciais dadas pela equação (7), a partir da equação (22), pode ser mostrado que os coeficientes são dados pela seguinte equação (Maia, 1996):

$$\boldsymbol{\theta} = k(1 - \bar{y}^{*2}) \mathbf{W}_h^T \boldsymbol{\psi} \quad (23)$$

Sendo que $\boldsymbol{\theta} = [a_1 \dots a_n \ b_1 \dots b_\ell]^T$, \bar{y}^* é o valor calculado, de acordo com a equação (22), para o ponto de operação da planta, ou seja, quando $y(t-i) = y^*$ e $u(t-j) = u^*$, $i = 1, \dots, n$ e $j = 1, \dots, \ell$. Além disso, $\boldsymbol{\psi}$ é um vetor auxiliar tal que:

$$\boldsymbol{\psi} = \mathbf{W}_o^T - \mathbf{W}_o^T \diamond (\mathbf{h}^* \diamond \mathbf{h}^*) \quad (24)$$

na qual \mathbf{h}^* é o valor de \mathbf{h} calculado para o ponto de operação da planta, segundo a equação (22), e \diamond indica o produto de Schur (produto elemento a elemento).