

# CONTROLE NEURAL DE UM PROCESSO DE NÍVEL DE 4ª ORDEM

JULIO C. P. OLIVEIRA, ADOLFO BAUCHSPIESS

*Grupo de Robótica, Automação e Visão Computacional*  
*Departamento de Engenharia Elétrica*  
*Universidade de Brasília - 70910-970 Brasília – DF*  
juliocpo@uol.com.br, adolfobs@unb.br

**Abstract** - This paper presents the NARMA-L2 control of a 4<sup>th</sup> order nonlinear process - a four coupled water reservoirs. This non-linear process at the University of Brasilia can be configured in different manners leading to great differences in the system response, accordingly to the operating point, being so a good test-bed for advanced control algorithms. In this paper the training was done using the following algorithms: Conjugate Gradient Backpropagation with Powell-Beale Restarts, Conjugate Gradient Backpropagation with Fletcher-Reeves Updates and Conjugate Gradient Backpropagation with Polak-Ribiere Updates are presented. The choice of the number of neurons and error goals are discussed. The experimental results for the different training options are presented and evaluated.

**Keywords** - neural controller, level process, nonlinear control, NARMA-L2

**Resumo** - Este artigo apresenta o controle neural NARMA-L2 de um processo de 4ª ordem não-linear – um conjunto acoplado de quatro reservatórios de água. Este processo não linear na Universidade de Brasília pode ser configurado de diferentes formas produzindo sistemas com tempos de resposta muito diferentes, dependendo do ponto de operação, sendo uma plataforma de teste para algoritmos de controle avançados. Neste artigo o treinamento foi realizado com os seguintes algoritmos: *Conjugate Gradient Backpropagation with Powell-Beale Restarts*, *Conjugate Gradient Backpropagation with Fletcher-Reeves Updates* e *Conjugate Gradient Backpropagation with Polak-Ribiere Updates*. A escolha do número de neurônios por camada e os objetivos de erro são discutidos. Os resultados experimentais para as diferentes condições de treinamento são apresentados e avaliadas.

**Palavras-chaves** – controlador neural, processo de nível, controle não linear, NARMA-L2

## 1 Introdução

As Redes Neurais Artificiais são importantes ferramentas para aplicação em controle de plantas não-lineares dadas suas características de mapeadores universais e pela capacidade de aprender por treinamento. Aplicações envolvendo controle adaptativo através da linearização do sinal de controle para sistemas SISO (Chen & Khalil, 1995) e controle por Modelo Interno (Hunt & Sbarbaro, 1991) são algumas das muitas utilizações de redes neurais em controle de processos.

Sistemas dinâmicos não-lineares são de grande interesse para a engenharia de controle. Devido à complexidade de tais sistemas é difícil desenvolver técnicas abrangentes para o projeto de controladores. Algumas técnicas baseiam-se no cancelamento das não-linearidades (Slotine & Weiping, 1990). Apresentam, porém, como principal limitação o fato de necessitarem de um modelo “exato” da planta e a mesma deve possuir dinâmica inversa estável.

Três abordagens principais são usadas para descrever sistemas não lineares: 1) Modelamento, 2) Identificação e 3) Treinamento.

Identificação é um procedimento que, no caso mais simples, procura sintonizar uma estrutura pré-definida com sinais medidos do processo. É um procedimento de otimização que procura os valores dos parâmetros que minimizam o erro entre o processo e o modelo. Os sinais de perturbação do processo também podem ser modelados. Se temos a

estrutura do processo todo, esta é a melhor aproximação experimental de um sistema não linear. Na prática é muito difícil formular corretamente a estrutura interna de um processo não-linear.

As Redes Neural Artificiais (RNAs) são um aproximador universal não-linear. Nem a estrutura e nem os parâmetros são necessários. Se os dados de treinamento são ricos em informação, o número de neurônios é suficiente, e temos um algoritmo de treinamento adequado, podemos esperar uma solução plausível. Uma solução ótima global raramente é necessária para problemas práticos.

Quando uma faixa ampla de operação é esperada é mais difícil que uma única RNA tenha bom desempenho em toda esta faixa. Neste artigo propomos a composição de um conjunto de RNAs treinadas por faixas de operação. Uma transição suave deve ocorrer quando há mudança da faixa de operação.

Este artigo apresenta o controle neural NARMA-L2 de um processo de 4ª ordem não-linear – um conjunto acoplado de quatro reservatórios de água. Os resultados para diferentes algoritmos de treinamento como *Conjugate Gradient Backpropagation With Powell-Beale Restarts*, *Conjugate Gradient Backpropagation With Fletcher-Reeves Updates* e *Conjugate Gradient Backpropagation With Polak-Ribiere Updates*, número de camadas e erros usando este controlador são apresentados.

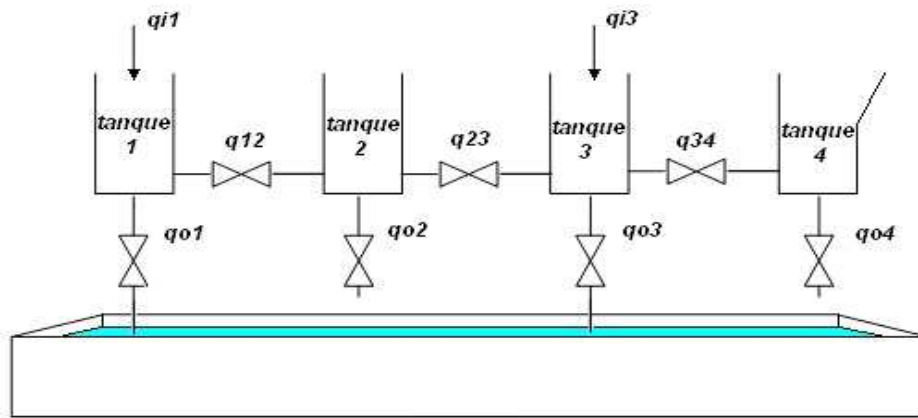


Figura 1. Representação esquemática da planta tanques acoplados

## 2 O Processo não linear de tanques acoplados

O processo investigado neste trabalho é um sistema multivariável de tanques acoplados para avaliação de técnicas de controle do Laboratório LAVSI, figuras 1 e 2. Três desses tanques possuem dimensões de 49,5x10x6 cm e o quarto tanque tem secção variável. As secções transversais dos tanques 1, 2 e 3 de 60 cm<sup>2</sup>. A largura do tanque 4 a partir de 14,7 cm de altura, representa 29% do nível de líquido, passa a aumentar linearmente até 20,2 cm de largura (31,65°) com área da secção transversal passando de 60 cm<sup>2</sup> para 121,2 cm<sup>2</sup>.

O tanque 1, possui entrada de água, que é bombeada do reservatório para este tanque por meio de uma moto-bomba  $u_1$ , e também possui uma saída de água,  $q_{o1}$ , que é bombeada do tanque 1 para o reservatório por meio de uma moto-bomba  $u_2$ . O tanque 2, possui uma saída de água,  $q_{o2}$ , através de um furo de 2mm, situado no fundo desse tanque para o reservatório. O tanque 3, possui entrada de água, que é bombeada do reservatório para este tanque por meio de uma moto-bomba  $u_3$ , e também possui uma saída de água,  $q_{o3}$ , que é bombeada do tanque 3 para o reservatório por meio de uma moto-bomba  $u_4$ . O tanque 4, possui uma saída de água,  $q_{o4}$ , através de um furo de 2mm, situado no fundo desse tanque para o reservatório.

Entre os compartimentos existe uma ranhura de 2mm de largura com altura variável, que determina o parâmetro  $k$ , interconectando os tanques.

As moto-bombas são acionadas por meio de um sistema de potência comandado a partir de um sinal de tensão entre 0 e 10 Vcc. Cada tanque é dotado de um sensor de nível tipo capacitivo para a medição dos 49,5 cm de coluna de líquido.

Através da vazão de entrada no tanque 1 via válvula  $k_{12}$ ,  $k_{23}$  e  $k_{34}$ , podemos controlar a vazão entre os tanques 1, 2, 3 e 4, respectivamente. O problema de controle abordado neste trabalho consiste na regulação da altura do nível do tanque 4 ( $h_4$ ). A vazão de saída dos tanques 2 e 4 acontece por furos de 2 mm no fundo dos respectivos tanques.



Figure 2a. Processo de Nível de Líquidos.



Figure 2b. Detalhe dos 4 reservatórios de água.

O modelo teórico deste processo pode ser obtido utilizando leis de conservação de energia (Aguirre, 2000). A vazão volumétrica de água que entra menos a que sai é igual à variação acumulada no tanque.

As vazões e as alturas foram assim definidas:

$q_{i1}$  e  $q_{i3}$  = entrada nos tanques 1 e 3, [cm<sup>3</sup>/s];

$q_{o1}$  e  $q_{o3}$  = saída nos tanques 1 e 3, [cm<sup>3</sup>/s];

$q_{12}$ ,  $q_{23}$  e  $q_{34}$  = vazões entre os tanques 1-2, 2-3 e 3-4, [cm<sup>3</sup>/s];

$q_{o2}$  e  $q_{o4}$  = vazões de saída nos tanques 2 e 4, através dos furos de 2mm [cm<sup>3</sup>/s];

$h_1$ ,  $h_2$ ,  $h_3$  e  $h_4$  = altura do nível de água nos tanques 1, 2, 3 e 4, [cm].

$A$  é a área da seção transversal dos tanques

$k$  é o parâmetro da válvula entre os tanques  $k_{12}$ ,  $k_{23}$ , e  $k_{34}$  [cm<sup>2.5</sup>/s]

O modelo dinâmico completo do processo de tanques acoplados do LAVSI é representado pelas equações 1 (Melo e Bernardes, 2006):

$$\begin{aligned} A_1 \frac{dh_1}{dt} &= q_{i1} - k_{o1} \sqrt{h_1} - \text{signa}(h_1 - h_2) k_{12} \sqrt{|h_1 - h_2|} \\ A_2 \frac{dh_2}{dt} &= \text{signa}(h_1 - h_2) k_{12} \sqrt{|h_1 - h_2|} - k_{o2} \sqrt{h_2} \\ &\quad - \text{signa}(h_2 - h_3) k_{23} \sqrt{|h_2 - h_3|} \\ A_3 \frac{dh_3}{dt} &= q_{i3} + \text{signa}(h_2 - h_3) k_{23} \sqrt{|h_2 - h_3|} - k_{o3} \sqrt{h_3} \\ &\quad - \text{signa}(h_3 - h_4) k_{34} \sqrt{|h_3 - h_4|} \\ A_4 \frac{dh_4}{dt} &= \text{signa}(h_3 - h_4) k_{34} \sqrt{|h_3 - h_4|} - k_{o4} \sqrt{h_4} \end{aligned} \quad (1)$$

### 3 Ferramenta NARMA-L2®

O modelo aproximado NARMA-L2 representa o sistema proposto por Narendra e Mukhopadhyay. O modelo NARMA-L2 pode ser usado com o modelo da planta previamente identificado, usa duas redes neurais, sendo uma rede controladora e outra rede do modelo da planta. Tipicamente, o número de atrasos incrementa com a ordem da planta. NARMA-L2 – ferramenta para identificação de uma planta, transforma o sistema não linear em um sistema linear através do cancelamento das não linearidades.

Diversos modelos de equações diferenciais não-lineares apropriados para a identificação foram desenvolvidos. Um exame excelente pode ser encontrado nos trabalhos de Billings. O modelo standard que está sendo usado para representar sistemas gerais não lineares no tempo discreto é o modelo NARMA (Nonlinear Autoregressive-Moving Average).

O controlador Narma-L2 é um rearranjo da rede neural através do modelo da planta, que é treinado offline. O controlador transforma a dinâmica não-linear do sistema em dinâmica linear cancelando as não linearidades. Diversas etapas devem ser seguidas

para aplicar o controlador Narma-L2 dentro do sistema. A primeira etapa é identificar uma rede neural para representar a dinâmica do sistema. O modelo NARMA (Nonlinear Autoregressive-Moving Average) é representado pelo sistema:

$$\hat{y}(k+d) = f[y(k), y(k-1), \dots, y(k-n+1), \dots, u(k-m+1)] + g[y(k), y(k-1), \dots, y(k-n+1), u(k-1), \dots, u(k-m+1)]u(k)$$

Onde  $d \leq 2$ , usando o controlador modelo NARMA-L2, obtêm-se a forma:

$$u(k+1) = \frac{y_r(k+d) - f[Y, U]}{g[Y, U]}$$

$$\text{onde: } \begin{cases} Y = [y(k), \dots, y(k-n+1)] \\ U = [u(k), u(k-1), \dots, u(k-n+1)] \end{cases}$$

O desempenho do controlador Narma-L2 depende essencialmente da identificação do sistema com uma rede neural. Um bom modelo de uma rede neural para o sistema fornece bons resultados nos pontos de operação desejados.

O controlador NARMA-L2 é mostrado na figura 3, onde os blocos TDL são linhas de atrasos ligeiros que armazenam valores precedentes do sinal de entrada e as funções de transferência são: sigmoideal e linear para as duas camadas respectivamente.

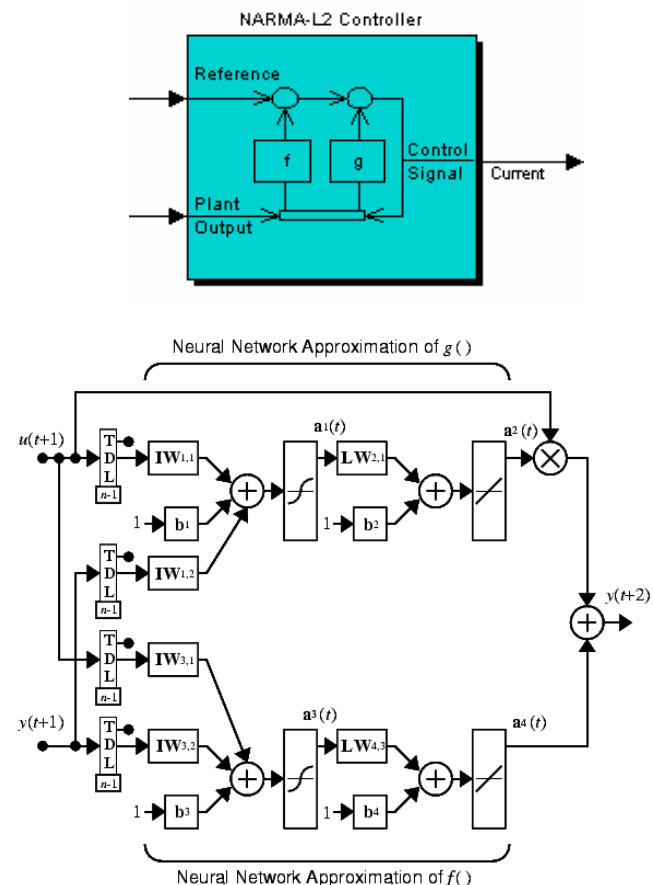


Figura 3. Controlador NARMA-L2

NARMA-L2 é uma ferramenta incluída no software MATLAB® que utiliza a teoria de redes neurais e identificação de parâmetros para realizar a identificação e controle de sistemas, como mostrado na figura 2.

Composta por três campos de interface: arquitetura da rede, dados de treinamento e parâmetros de treinamento. A identificação inicia-se pelos dados de treinamento que podem ser gerados pela própria ferramenta (Generate training data) ou importados de algum arquivo (Import data). Define-se a topologia da rede (Network Architecture). A escolha do número de camadas, *delays* e intervalo de amostragem é feita baseada no sistema não-linear a ser controlado. A etapa de identificação é necessário escolher os parâmetros de treinamento (Training parameters), que também serão definidos baseados na complexidade do sistema estudado e na capacidade computacional. Definidos os três campos, inicia-se o treinamento. O erro médio quadrático destes dados servem como critério limitante para o treinamento. No término do treinamento, são fornecidas de forma gráfica as seguintes respostas para os três tipos de dados: erro médio quadrático, erro quadrático, sinal de entrada, sinal de saída da planta e sinal de saída da rede.

Por fim, a rede neural obtida é aplicada ao bloco de controle NARMA-L2® (figura 5). Nessa ferramenta o controlador será a própria rede neural, ou seja, a própria rede será inserida na malha do sistema. O treinamento obtém o modelo inverso do sistema dinâmico.

#### 4 Metodologia

O procedimento foi iniciado, encontrando-se o tempo de amostragem para a determinação de amostras que representam o modelo desejado. Aplicou-se um degrau na entrada do sistema, caracterizando o tempo de resposta da planta.

A planta foi identificada pelos dados de treinamento que podem ser gerados pela ferramenta Generate Training Data do NARMA-L2. Feito isso, definimos a topologia da rede Network Architecture, escolhendo o número de camadas, *delays* e intervalo de amostragem.

A rede encontrada foi aplicada ao controlador. Para se verificar a resposta do controlador utilizou-se a configuração descrita na figura 6. Esse procedimento foi repetido para várias configurações da rede neural, como tipo de algoritmo, número de camadas e atrasos

Durantes os testes foi verificado se o sinal de saída da rede acompanha adequadamente o sinal de entrada e, reflete satisfatoriamente o comportamento mostrado na saída da planta. Observando, ainda, que o sinal de saída da planta e da rede atingem o regime sempre em valores de amplitudes menores que  $q_{i1} = 0$  a  $66$  [cm<sup>3</sup>/s].

Todas as simulações dos diagramas de blocos descritos acima foram realizadas no SIMULINK® do software MATLAB®.

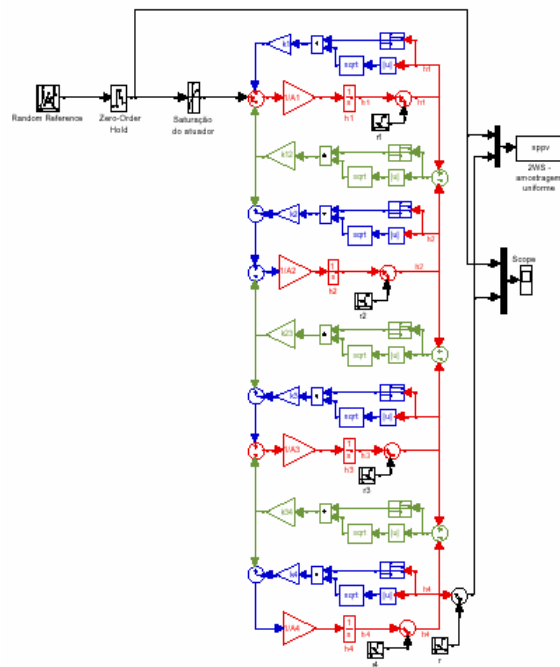


Figura 5. Diagrama de blocos dos tanques

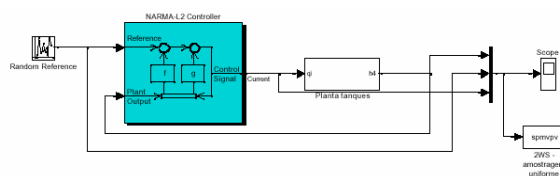


Figura 6 – Modelo do sistema controlado

#### 5 Resultados

Como descrito, foram obtidas respostas para várias configurações dos tanques acoplados de água. Para a apresentação dos resultados gráficos, utilizou-se os seguintes parâmetros:

- $q_{i1} = 0$  a  $66$  [cm<sup>3</sup>/s];
- $q_{i3} = 0$  [cm<sup>3</sup>/s];
- $h_4 = 0$  a  $46$  [cm]
- $k_1=0, k_2=5, k_3=0$  e  $k_5=5$  [cm<sup>2</sup>,5/s];
- $k_{12} = 12, k_{23} = 2.5, e k_{34} = 6$  [cm<sup>2</sup>,5/s]

A planta foi identificada pelos dados de treinamento que podem ser gerados pela ferramenta Generate Training Data do NARMA-L2. Feito isso, definimos a topologia da rede Network Architecture, escolhendo o número de camadas, *delays* e intervalo de amostragem.

Na Tabela 1 temos as respostas para várias configurações dos tanques acoplados de água, para várias configurações considerando o número de camadas, *delay in*, *delay out* e os erros de treinamento e validação. O objetivo desta comparação é obter dados para avaliar a performance de cada algoritmos. Para isto utilizaremos uma rede neural composta: números de camadas e atrasos na entrada e na saída.

Tabela 1 – Respostas para várias configurações dos tanques acoplados de água

Algoritmo train	Número camada	Delay In	Delay out	Error Goal	Erro Treinamento	Erro Validação	Erro Teste
trainlm	9	6	6	0.00018	$-0.3 a + 0.2 \times 10^{-4}$	$-0.2 a + 0.2 \times 10^{-4}$	$-0.1 a + 0.1 \times 10^{-4}$
trainlm	9	8	8	0.00018	$-0.3 a + 0.2 \times 10^{-4}$	$-0.2 a + 0.2 \times 10^{-4}$	$-0.1 a + 0.1 \times 10^{-4}$
trainlm	24	16	16	0.00160	$-0.5 a + 1 \times 10^{-4}$	$-0.6 a + 0.4 \times 10^{-4}$	$-0.4 a + 0.4 \times 10^{-4}$
<b>Traincgb (a)</b>	<b>9</b>	<b>4</b>	<b>4</b>	<b>0.37180</b>	<b><math>-5 a + 15 \times 10^{-4}</math></b>	<b><math>-5 a + 15 \times 10^{-4}</math></b>	<b><math>-2 a + 6 \times 10^{-4}</math></b>
traincgb	5	4	4	0.10040	$-1 a + 4 \times 10^{-4}$	$-4 a + 6 \times 10^{-4}$	$-5 a + 10 \times 10^{-4}$
traincgb	5	3	3	8.82390	$-5 a + 15 \times 10^{-4}$	$-5 a + 15 \times 10^{-4}$	$-5 a + 10 \times 10^{-4}$
<b>Traincgf (b)</b>	<b>5</b>	<b>4</b>	<b>4</b>	<b>1.23600</b>	<b><math>-20 a + 20 \times 10^{-4}</math></b>	<b><math>-5 a + 15 \times 10^{-4}</math></b>	<b><math>-5 a + 10 \times 10^{-4}</math></b>
<b>Traincgp (c)</b>	<b>5</b>	<b>4</b>	<b>4</b>	<b>0.86700</b>	<b><math>-10 a + 15 \times 10^{-4}</math></b>	<b><math>-20 a + 20 \times 10^{-4}</math></b>	<b><math>-5 a + 10 \times 10^{-4}</math></b>
trainrp	5	4	4	5.84090	$-2 a + 6 \times 10^{-4}$	$-2 a + 6 \times 10^{-4}$	$-2 a + 3 \times 10^{-4}$

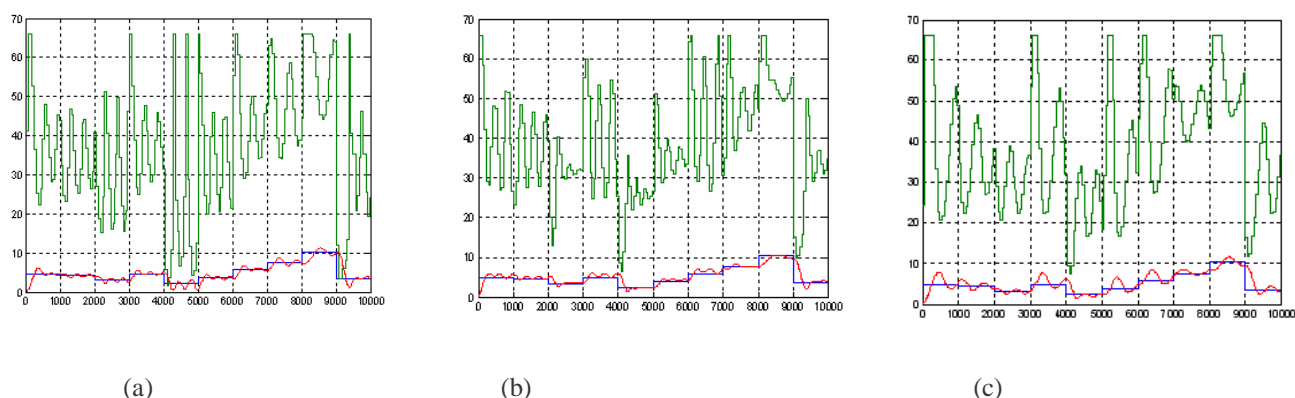


Figura 7 – Sinais na planta: valor desejado, variável de processo e variável manipulada. Algoritmos de treinamento: Conjugate Gradient Backpropagation With Powell-Beale Restarts - traincgb (a), Conjugate Gradient Backpropagation With Fletcher-Reeves Updates - traincgf (b) e Conjugate Gradient Backpropagation With Polak-Ribiere Updates - traincgp (c). Vetores: SP (azul) MV (verde) PV (vermelho).

Representamos a variação do *Erro Goal*, *Erro de Treinamento*, *Erro de Validação* e *Erro Teste* durante os diversos algoritmos usados na rede. Decidimos abortar o treinamento em tantos ciclos pois o erro não apresentou mais redução significativa durante os próximos ciclos, o que significa que a rede ou encontrou o máximo de fidelidade possível no mapeamento da função que desejamos que aprenda ou então que encontrou um mínimo local na paisagem de erro.

Na Tabela nº 2, mostramos o Tempo de Resposta nos experimentos para o Controlador

NARMA-L2. Nessa tabela mostramos: o valor desejado, o ascendente teórico, o tempo de subida da variável de nível em segundos, o tempo de pico da variável de nível em segundos e o sobre sinal da variável de nível em percentual para o algoritmo *Conjugate Gradient Backpropagation With Fletcher-Reeves Updates* – traincgf, que obteve um menor erro e uma melhor excursão da variável manipulada para o processo de nível de líquidos de 4ª ordem.

Tabela 2 – Respostas para várias configurações dos tanques acoplados de água

Controlador NARMA_L2	valor Desejado (cm)	Ascendente Teórico (%)	Tempo Subida (seg)	Tempo Pico (seg)	Sobre Sinal (%)
RNA - CGF	5cm	00cm a 5,0cm	251	416	46,5
RNA - CGF	10cm	00cm a 10,0cm	498	672	10,96
RNA - CGF	14cm	00cm a 14,0cm	1064	1225	2,08

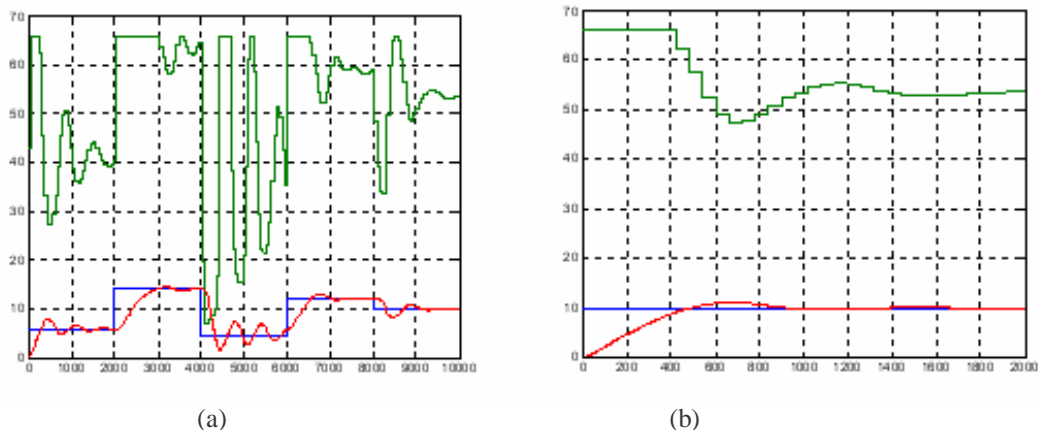


Figura 8 – Sinais na planta: valor desejado, variável de processo  $y(k)$  e variável manipulada  $u(k)$  Algoritmo de treinamento *traincgf*. Vetores: SP (azul) MV (verde) PV (vermelho).

## 6 Conclusões

Apresentamos neste trabalho um controlador NARMA-L2, para um controle de um processo de nível de líquidos de 4ª ordem, que foi modelado e realizado no SIMULINK® do software MATLAB®. Apresentamos uma tabela comparativa de tempo de resposta entre os controladores: *Conjugate Gradient Backpropagation With Powell-Beale Restarts*, *Conjugate Gradient Backpropagation With Fletcher-Reeves Updates* e *Conjugate Gradient Backpropagation With Polak-Ribiere Updates*.

A abordagem de controle utilizada é tal que o controlador NARMA-L2 substitua uma arquitetura projetada de uma singular rede neural artificial e outros controladores do tipo PID, para planta de tanques acoplados de 4ª ordem.

Estudos e experimentos estão em andamento no Laboratório de Automação, Visão e Sistemas Inteligentes – LAVSI, no Departamento de Engenharia Elétrica, da Universidade de Brasília para que o controlador NARMA-L2 controle o processo de nível de líquidos de 4ª ordem através de Controlador Lógico Programável em *ladder logic*, por ser uma abordagem factível para a utilização industrial.

## Referências Bibliográficas

Aguirre, L. A. (2000) Introdução à Identificação de Sistemas: Técnicas lineares e não-lineares aplicadas a sistemas reais. Ed. UFMG.  
 A. Pukrittayakamee, O. Jesus, M. Hagan. (2002); Smoothing the Control Action for NARMA-L2 Controllers; 45th IEEE International Midwest Symposium on Circuits and Systems Oklahoma State University.

Billings, S.A. (1985a). "Introduction to Nonlinear Systems Analysis and Identification", in K. R. Godfrey and R. P. Jones (eds) Lecture Notes in Control Systems, Signal Processing for Control, SERC Vacation School, Springer-Verlag, pp. 262-294.  
 Billings, S.A. (1985b). "An Overview of Nonlinear Systems Identification", in Proc. Of the 7 th IFAC Symposium in System Identification and System Parameter Estimation, pp.  
 Chen, F.C. e Khalil, H.K. (1995). Adaptive Control of a Class of Nonlinear Discrete-Time Systems Using Neural Networks, IEEE Trans. on Automat. Control, 40, pp. 791-801.  
 Cybenko, G. (1989). Approximation by a superpositions of a sigmoidal function, Math Control Signal Systems, Vol.2, pp. 303-314.  
 Hunt, K.J.; Sbarbaro, D. (1991). Neural Networks for nonlinear internal model control, IEE Proc., Part D, 138, pp. 431-438.  
 Narendra, K.S. and S. Mukhopadhyay, "Adaptive Control Using Neural Networks and Approximate Models," IEEE Transactions on Neural Networks, vol. 8, 475-485 (1997).  
 Oliveira, Júlio César P.; Resende, P. e Maia, C. A. (1997). Controle Neural Gain Scheduling de Tanques Interativos, 3º SBAI, Vitória, pp. 224-228.  
 Oliveira, Júlio César P.; Bauchspiess, A. . Neural Control of a Non Linear Process with Gain Scheduling . In: SBRN 2004, 2004, São Luís, 2004.  
 Oliveira, Júlio César P.; Bauchspiess, Controle Neural com pesos selecionados pela faixa de operação aplicados a um processo de nível de líquidos de 4ª ordem.. CBA 2006, Salvador-BA.  
 Reed, Russell D., Marks, Robert J. (1999); Neural Smithing Supervised Learning in Feedforward Artificial Neural Networks, The Massachusetts Institute of Technology Press, London, England.  
 Slotine, J.J.E.; Weiping, L. (1990). Applied Nonlinear Control, Prentice Hall, New Jersey.