

Controle de Nível Usando Rede Neural

Pedro Berretta de Lucena
UFRN – CT – DCA
pedro@dca.ufrn.br

Fábio Meneghetti Ugulino de
Araújo
UFRN – CT – DCA
meneghet@dca.ufrn.br

Andrés Ortiz Salazar
UFRN – CT – DCA
andres@dca.ufrn.br

Resumo

Este trabalho trata da implementação de um controlador baseado no jacobiano da planta, estimado a partir de uma rede neural utilizada para identificar a planta em tempo real. Dessa forma, o controlador funciona para sistemas não-lineares e se adapta a mudanças da planta. O objetivo é controlar uma planta de nível de segunda ordem simulada e apresentar os resultados.

1. Introdução

Atualmente, a inteligência artificial (IA) está sendo uma área bastante promissora para contribuir com a área de controle. Um bom motivo para isso, é que há uma certa dificuldade em se controlar sistemas não-lineares ou variantes no tempo com as técnicas convencionais de controle, e a inteligência artificial pode suprir essa falta. As redes neurais artificiais (RNA), são um bom exemplo de ferramenta da inteligência artificial que faz esse importante papel de complementar ou substituir o controle convencional. Isso se dá devido à sua adaptabilidade, generalização do aprendizado, e não-linearidade.

O primeiro sistema de controle neural foi desenvolvido por Widrow e Smith (1963) [1]. Eles usaram um Adaline para estabilizar e controlar o balanceamento de pólo. Outras demonstrações de controle neural foram feitas posteriormente por outros autores como, por exemplo, Waltz e Fu (1965) [2], Michie e Cahmbers (1968) [3] e Barto *et al.* (1983) [4].

O interesse em usar redes neurais para controle somente começou em torno de 1987. A partir da primeira conferência da IEEE, tem-se um apurado aumento de conferências e artigos de jornais publicados sobre controle. Esses artigos têm demonstrado que as redes neurais podem ser aplicadas com sucesso para controle de sistemas não-lineares

desconhecidos [5]. Várias novas estruturas de controle neural foram propostas. Por exemplo, “*feedback error learning*” por Kawato *et al.* [6] em 1987, “*neural internal model control*” por Hunt e Sbarbaro [7] em 1991, “*neural predictive control*” por Willis *et al.* [8] em 1992, “*forward and inverse modeling*” por Jordan e Jacobs [9] em 1990, “*neurofuzzy*” por Harris *et al.* [10] em 1993, “*generalized and specialized learnings*” por Psaltis *et al.* [11] em 1988.

A partir de 1992, podemos ver um aumento no uso de redes neurais com outros controladores. Esses tipos de estruturas híbridas tentam tirar vantagem dos pontos fortes de diferentes controladores e evitar as falhas dos esquemas de controle neural puros [5].

Na opinião de Narendra [12], o controle adaptativo de sistemas não-lineares usando redes neurais irá evoluir, em termos gerais, na mesma direção que o controle adaptativo de sistemas lineares seguiu nas últimas duas décadas.

O esquema de controle apresentado neste trabalho foi inicialmente proposto por Adetona *et al.* [13] em 2001 e posteriormente utilizado por Gabriel Filho [14] em 2003 com algumas modificações. Trata-se de um sistema de controle híbrido indireto com uma rede neural com treinamento em tempo de execução para identificar a planta. Assim, o sistema de controle pode se auto-ajustar para atender a novos parâmetros de projeto ou à variação da planta. A rede neural é utilizada como identificador da planta, a estrutura de modelo do identificador neural implementado foi a NNARX (*Neural Network AutoRegressive with eXogenous input*) [15]. O algoritmo de treinamento da rede neural utilizado foi o de retropropagação [16], como proposto em [14].

Uma breve introdução às redes neurais é apresentada a seção 2. O modelo de identificador neural NNARX é discutido na seção 3. A seção 4 trata do controlador utilizado, baseado no jacobiano estimado da planta [13][14], e do sistema de controle

por completo. Neste trabalho, o sistema de controle foi avaliado através de simulação computacional. Alguns resultados são mostrados e comentados na seção 5. A seção 6 traz a conclusão deste trabalho.

2. Redes Neurais

Uma RNA consiste na utilização de neurônios artificiais com suas entradas conectadas a nós de fonte (valores externos) ou computacionais (saídas de outros neurônios). Existem várias arquiteturas de redes neurais, porém aqui só será abordada a arquitetura dos *perceptrons* de múltiplas camadas. Um *perceptron* de múltiplas camadas consiste em uma RNA organizada em camadas onde só há conexão entre neurônios de camadas vizinhas. A camada de entrada é uma cama de nós de fonte, ou seja, não possui neurônios, já a camada de saída e as camadas ocultas (intermediárias), caso existam, são constituídas de neurônios (nós computacionais). Dizemos que um *perceptron* de múltiplas camadas é totalmente conectado quando todos os nós de todas as camadas estão conectados com todos os nós das camadas vizinhas. A Figura 1 ilustra um exemplo de RNA *perceptron* de múltiplas camadas.

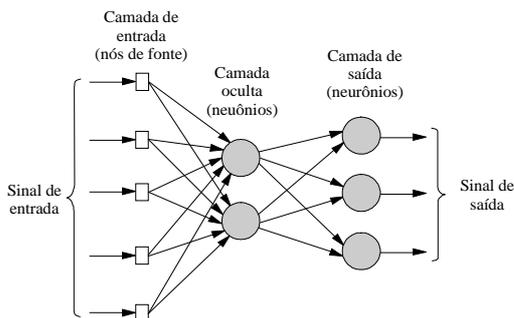


Figura 1 - Exemplo de um *perceptron* de múltiplas camadas totalmente conectado.

Para se treinar um *perceptron* de múltiplas camadas, uma das alternativas é utilizar o algoritmo de retropropagação (*back-propagation*). O objetivo do processo de aprendizagem é ajustar os parâmetros livres da rede (pesos e *bias*) para minimizar a energia média do erro quadrado ξ_{med} . O algoritmo de retropropagação faz isso aplicando uma modificação em cada parâmetro livre baseando-se no gradiente de ξ_{med} [16].

3. Identificador Neural

Para identificar a planta, foi utilizado o modelo NNARX (*Neural Network AutoRegressive with*

exogenous inputs – rede neural auto-regressiva com entradas exógenas) descrito por NØrgaard *et al.* [15]. A expressão matemática do modelo não-linear estendido NNARX é descrita como

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

onde \hat{y} é a saída estimada da planta, k é o instante atual, $d = n_y - n_u$ é o atraso da planta, n_y é a ordem da saída da planta, n_u é a ordem da entrada da planta, $n = n_y$ é a ordem da planta, $f(\cdot)$ é uma função não linear mapeada pela RNA, y é a saída da planta e u é a entrada da planta. A Figura 2 ilustra o modelo de estrutura NNARX.

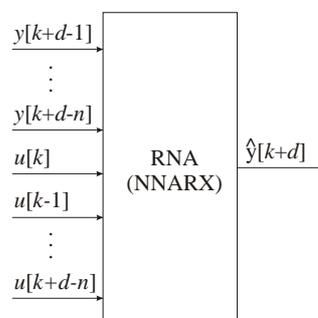


Figura 2 - Modelo de estrutura NNARX.

A RNA utilizada para mapear a função $f(\cdot)$ é um *perceptron* de múltiplas camadas que possui três camadas: a camada de entrada com nós de fonte; a camada oculta, com neurônios com função de ativação sigmóide (unipolar ou bipolar); e a camada de saída com um neurônio com função de ativação linear.

O modelo NNARX é BIBO (*Bounded Input, Bounded Output* – entrada limitada, saída limitada) estável por não possuir realimentação da saída estimada [13]. Por não possuir problemas relativos à estabilidade, esse modelo é a estrutura preferida quando o sistema a ser modelado é determinístico ou o nível de ruído é insignificante [15].

4. Sistema de Controle

O esquema de controle híbrido indireto utilizado baseia-se na proposta de Adetona *et al.* [13] modificada por Gabriel Filho [14], que trocou a rede neural com função de base radial do identificador por uma rede neural *perceptron* de múltiplas camadas usando o algoritmo de retropropagação [16] em seu

treinamento. O termo indireto refere-se aos parâmetros do controlador que são ajustados com base num modelo estimado da planta e não no seu modelo nominal.

No esquema de controle, foram utilizados um identificador neural (IN) – modelo NNARX – e um controlador baseado no jacobiano estimado da planta conforme a Figura 3:

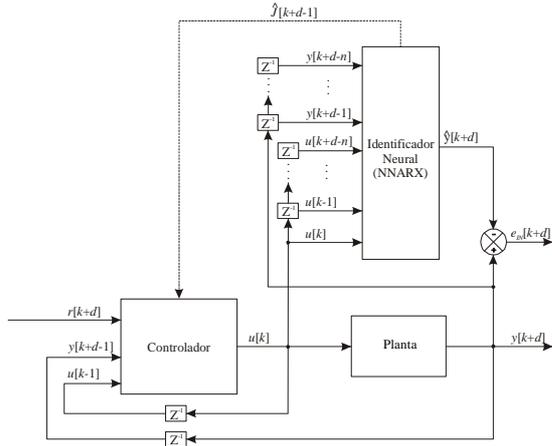


Figura 3 - Esquema de controle híbrido indireto.

onde Z^{-1} é o operador de atraso unitário no tempo discreto e $e_{IN}[k+d]$ é o erro usado para o treinamento do IN em tempo real.

Para calcular o sinal de controle u no instante k , foi utilizada a seguinte lei de controle:

$$u[k] = u[k-1] + \frac{r[k+d] - \hat{y}[k+d-1]}{\hat{J}[k+d-1]} \quad (2)$$

em que r é o valor de referência e \hat{J} é o jacobiano estimado da planta que é dado por

$$\begin{aligned} \hat{J}[k+d-1] &= \frac{\partial \hat{y}[k+d-1]}{\partial u[k-1]} \\ &= \sum_{j=1}^p \left[\frac{\partial \varphi^{(o)}}{\partial v_1^{(o)}} \cdot \frac{\partial \varphi^{(h)}}{\partial v_j^{(h)}} \cdot w_{j,(2n-d+1)}^{(h)} \cdot w_{1,j}^{(o)} \right] \end{aligned} \quad (3)$$

onde p é o número de neurônios da camada oculta, $\frac{\partial \varphi^{(o)}}{\partial v_1^{(o)}}$ é a derivada da função de ativação do neurônio da camada de saída com relação ao seu campo local

induzido $\partial v_1^{(o)}$ (a letra o vem de *out* – saída), $\frac{\partial \varphi^{(h)}}{\partial v_j^{(h)}}$

são as derivadas da função de ativação dos neurônios da camada oculta com relação aos seus respectivos campos locais induzidos $\partial v_j^{(h)}$, $w_{j,(2n-d+1)}^{(h)}$ são os pesos que correspondem à conexão da entrada $u[k-1]$, de posição $2n-d+1$ contada de cima para baixo, com os neurônios j da camada oculta.

5. Resultados

Os resultados foram obtidos utilizando-se uma rede neural (*perceptron* de múltiplas camadas) de três camadas para identificar a planta de um sistema de tanques simulada no computador. A estrutura de modelo de identificador neural utilizado foi a NNARX, com os neurônios da camada oculta possuindo função de ativação sigmóide bipolar e o neurônio da camada de saída possuindo função de ativação linear. O modelo do sistema de tanques foi retirado do manual do kit *Coupled Water Tank Experiments* [17], da Quanser, (configuração #2). Os testes foram simulados por computador, a programação foi feita através do *Borland C++ Builder 5*. A Figura 4 mostra o esquema da planta, onde o sinal de controle é a tensão (em volts) aplicada à bomba e a saída da planta é o nível L_2 (em centímetros) do tanque 2.

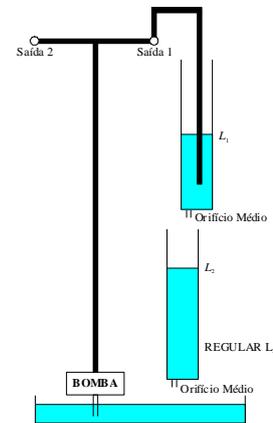


Figura 4 - Esquema do sistema de tanques (planta a ser controlada).

Para a simulação da planta, seu modelo não-linear [17] – Sistema de Equações (4) – foi discretizado aproximando-se as derivadas através do método de *Euler*. O Sistema de Equações (5) define o modelo não-linear discretizado da planta.

$$\begin{cases} \dot{L}_1 = -\frac{a_1}{A_1}\sqrt{2g L_1} + \frac{K_m}{A_1}V_p \\ \dot{L}_2 = \frac{a_1}{A_2}\sqrt{2g L_1} - \frac{a_2}{A_2}\sqrt{2g L_2} \end{cases} \quad (4)$$

$$\begin{cases} L_1[k+1] = \left(-\frac{a_1}{A_1}\sqrt{2g L_1[k]} + \frac{K_m}{A_1}V_p[k]\right)T_s + L_1[k] \\ L_2[k+1] = \left(\frac{a_1}{A_2}\sqrt{2g L_1[k]} - \frac{a_2}{A_2}\sqrt{2g L_2[k]}\right)T_s + L_2[k] \end{cases} \quad (5)$$

onde L_1 é o nível da água no tanque 1 em centímetros; L_2 é o nível da água no tanque 2 em centímetros; \dot{L}_1 é a derivada primeira de L_1 com relação ao tempo em segundos; \dot{L}_2 é a derivada primeira de L_2 com relação ao tempo em segundos; a_1 é a área do orifício de saída do tanque 1 em cm^2 ; a_2 é a área do orifício de saída do tanque 2 em cm^2 ; A_1 é a área interna da seção transversal do tanque 1 em cm^2 ; A_2 é a área interna da seção transversal do tanque 2 em cm^2 ; g é a aceleração da gravidade em cm/s^2 ; K_m é a constante da bomba em $\text{cm}^3/(\text{s}\cdot\text{V})$; V_p é a tensão aplicada à bomba em Volts; T_s é o período de amostragem em segundos; k é o instante no tempo discreto.

Os valores das constantes utilizados foram os seguintes:

$$\begin{aligned} T_s &= 50 \cdot 10^{-3} \text{ s} \\ a_1 &= 0,17813919765 \text{ cm}^2 \\ a_2 &= 0,17813919765 \text{ cm}^2 \\ A_1 &= 15,5179 \text{ cm}^2 \\ A_2 &= 15,5179 \text{ cm}^2 \\ K_m &= 4,6 \frac{\text{cm}^3}{\text{s} \cdot \text{V}} \\ g &= 981 \frac{\text{cm}}{\text{s}^2} \end{aligned}$$

Para o identificador neural, foram utilizados os seguintes parâmetros: $n = 2$, $d = 1$ e $p = 10$. No treinamento, foi utilizada uma tolerância de 10^{-3} . A função de ativação utilizada nos neurônios da camada oculta foi a sigmóide bipolar (tangente hiperbólica).

O número de exemplos utilizado para treinar a rede neural foi três. Assim, a cada nova amostra, surge um novo exemplo para ser usado no treinamento e o exemplo mais antigo é eliminado do conjunto de treinamento para manter apenas três exemplos. Assim,

a planta é identificada em apenas um pequeno intervalo de operação. Como o ponto de operação varia no tempo, o identificador se adapta ao novo intervalo.

Inicialmente, quando ainda não existem os três primeiros exemplos de treinamento, o sinal de controle usado é um valor aleatório de 0 a 22V.

A Figura 5 ilustra um gráfico com o valor do sinal de saída da planta (L_2) controlada, em vermelho, e o valor de referência (r), em azul, representados no eixo das ordenadas; já o eixo das abscissas, representa o tempo em segundos. O valor de referência foi alterado em intervalos de tempo inconstantes.



Figura 5 - Gráfico da referência r variando em intervalos de tempo inconstantes e da saída da planta L_2 controlada.

Um gráfico que ilustra a saída da planta controlada L_2 e a referência r como sendo um degrau que parte do zero até 15 cm é mostrado na Figura 6.

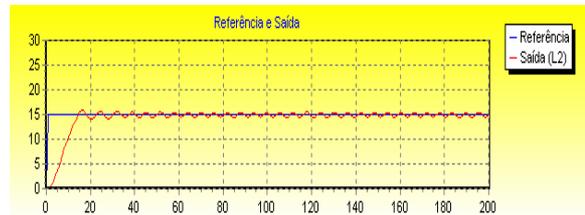


Figura 6 - Gráfico da referência r sendo um degrau de 0 a 15 cm e saída da planta L_2 controlada.

Um gráfico com valores da saída da planta controlada L_2 e da referência r como sendo um degrau que parte do zero até 5 cm pode ser visto na Figura 7 e como sendo um degrau que parte do zero até 25 cm pode ser visto na Figura 8.



Figura 7 - Gráfico da referência r sendo um degrau de 0 a 5 cm e saída da planta L2 controlada.

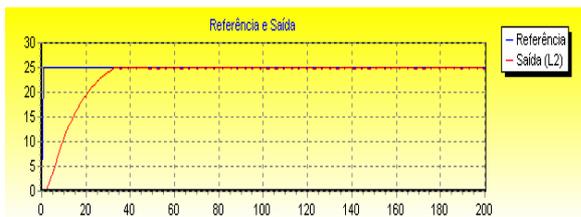


Figura 8 - Gráfico da referência r sendo um degrau de 0 a 25 cm e saída da planta L2 controlada.

6. Conclusão

Como se pode observar na Figura 5, Figura 6, Figura 7 e Figura 8, o sistema de controle se mostrou eficaz em sua função, conseguindo controlar a saída da planta levando-a para perto da referência. Uma pequena oscilação da saída da planta está ocorrendo em torno da referência. Pelas observações feitas, suspeita-se que isso deva ocorrer devido ao valor do jacobiano estimado ficar próximo do zero.

7. Referências

- [1] WIDROW, B.; SMITH, F. W. **Pattern-recognizing control systems**. In 'Computer and Information Sciences Symposium Proceedings'. Spartan, Washington, DC, 1963.
- [2] WALTZ, M. D.; FU, K. S. **A heuristic approach to reinforcement learning control systems**. IEEE Transactions on Automatic Control AC-10(4), pp. 390-398. 1965.
- [3] MICHIE, D.; CHAMBERS, R. A. **Boxes: An experiment in adaptive control**. In J. T. Tou and R. H. Wilcox (Eds.). 'Machine Intelligence'. Chapter 2, pp. 137-152. Edingurgh: Oliver and Boyd, 1968.
- [4] BARTO, A. G.; SUTTON, R. S.; ANDERSON, C. W. **Neuronlike adaptive elements that can solve difficult learning control problems**. IEEE Transactions on Systems Man and Cybernetics 13(5), pp. 834-846. 1983.
- [5] NG, G. W. **Application of neural networks to adaptive control of nonlinear systems**. London, England: Research Studies Press Ltd., 1997.
- [6] KAWATO, M.; FURUKAWA, K.; SUZUKI, R. **A hierarchical neural network model for control and learning of voluntary movement**. Biological Cybernetics 57, pp. 169-185. 1987.
- [7] HUNT, K. J.; SBARBARO, D. **Neural networks for non-linear internal model control**. IEE Proceedings Control Theory and Applications 138, pp. 431-438. 1991.
- [8] WILLIS, M. J.; MONTAGUE, G. A.; DIMASSIMO, C.; THAM, M. T.; MORRIS, A. J. **Artificial neural networks in process estimation and control**. Automatica 28(6), pp. 1181-1187. 1992.
- [9] JORDAN, M. I.; JACOBS, R. A. **Learning to control an unstable system with forward modeling**. In R. P. Lippmann, S. E. Moody and D. S. Touretzky (Eds.). 'Advances in Neural Information Processing Systems'. San Mateo: Morgan Kaufmann, 1990.
- [10] HARRIS, C. J.; MOORE, C. G.; BROWN, M. **Intelligent control: aspects of fuzzy logic and neural nets**. World Scientific. 1993.
- [11] PSALTIS, D.; SIDERIS, A.; YAMAMURA, A. **A multilayered neural network controller**. IEEE Control Systems Magazine 8(2), pp. 17-21. 1988.
- [12] NARENDRA, K. S. **Adaptive control of dynamical systems using neural networks**. In D. White and D. Sofge (Eds.). 'Handbook of Intelligent Control', pp. 141-183. New York: Van Nostrand Reinhold, 1992.
- [13] ADETONA, O.; SATHANANTHAN, S.; KEEL, L. H. **Robust Nonlinear Adaptive Control Using Neural Networks**. Proc of the American Control Conference, pp. 3884-3889. Arlington, VA. USA, 2001.
- [14] GABRIEL FILHO, Oscar. **Controlador híbrido indireto baseado em redes neurais – parte I: desenvolvimento e implementação**. VI Simpósio Brasileiro de Automação Inteligente, pp. 183-188. Bauru, 2003.
- [15] NØRGAARD, M.; RVN, O.; POULSE, N. K.; HANSEN, L. K. **Neural networks for modelling and control of dynamic systems**. London, England: Springer-Verlag London Limited, 2001.
- [16] HAYKIN, Simon. **Neural networks: a comprehensive foundation**. 2. ed. New Jersey: Prentice Hall, 1999.
- [17] **Coupled Water Tank Experiments**. Quanser Consulting Inc.