

Implementação do Servocontrole Auto-Sintonizado em Tempo-Real Utilizando Rede Perceptron Multicamadas

Zenilda Pedrosa Vieira¹
Adolfo Bauchspiess¹

GRACO – Grupo de Automação e Controle, <http://www.graco.unb.br>
Departamento de Engenharia Elétrica – Universidade de Brasília – UnB
70910-900 – Brasília / DF – BRASIL
E-mails: hzvieira@linkexpress.com.br, adolfo@ene.unb.br

Abstract

This paper shows first results of the identification and compensation, using artificial neural networks, of one of the main non-linearities that occurs in servo positioning systems: the operation point dependency of the system gain. In the data acquisition a DC current motor with velocity and position feedback was used. The dead-zone and the saturation are treated using standard non-linear techniques. A multilayer perceptron network (MLP) trained with the classical back-propagation algorithm was used to identify these non-linearities. The control-loop was closed using a standard discrete PID. The results shows that the network accomplish this task in a satisfactory way, linearizing the system in the dynamic operation range, off course inside the saturation limits.

1. Introdução

Os sistemas físicos, em geral, são lineares apenas em pequenos intervalos de operação. Algumas das não-linearidades encontradas em sistemas físicos são: *zona morta*, pequeno intervalo no qual o sistema mesmo

recebendo excitação não responde, *ganho dependente da faixa de operação*, ou seja, o ganho do sistema não é representado por uma constante e *saturação*, na qual a saída do sistema satura para grandes sinais da entrada. Estas não-linearidades limitam o desempenho de servoposicionadores, fazendo com que o mesmo não opere adequadamente.

Visando tornar o sistema mais robusto às variações paramétricas, propõe-se neste artigo, um método para compensar os efeitos não-lineares do ganho dependente da faixa de operação inerentes a sistemas posicionadores. A parte experimental utiliza o *kit MS-150*, da marca *Feedback®*.

Para que o sistema servoposicionador funcione adequadamente é necessário que haja um controle em malha fechada. A teoria clássica utiliza controladores lineares que em seu projeto despreza as não-linearidades presentes, aproximando a planta por um modelo linear. Desta maneira, o controlador pode apresentar um comportamento inadequado frente aos efeitos das não-linearidades próprias do servomecanismo posicionador.

Utilizando redes neurais artificiais para identificar parâmetros do ganho não-linear, foi possível linearizar o sistema e, assim, controlá-lo utilizando a teoria clássica, como por exemplo, com um controlador PID.

Deste modo, durante o funcionamento da planta real, uma rede é utilizada para identificar e adaptar o sistema a diversas situações de ganho, propiciando um funcionamento muito melhor do servoposicionador em face a um ajuste individualizado do controlador.

2. Aparato Experimental

Neste trabalho utilizou-se o servoposicionador do kit MS150 da marca *Feedback*®. As partes utilizadas desse kit foram: o pré-amplificador, o servoamplificador, a fonte de alimentação, o motor com um tacômetro e um potenciômetro de saída, conforme Fig. 1.

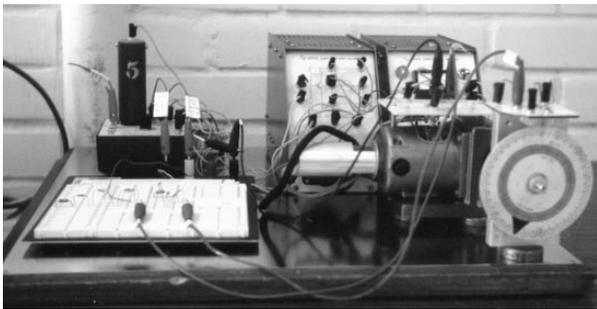


Figura 1 – Servoposicionador MS150, *Feedback*®

Este sistema tem a seguinte função de transferência linearizada, considerando um certo ponto de operação em malha aberta:

$$G(s)H(s) = \frac{K}{s(\tau_1 s + 1)(\tau_2 s + 1)} \quad (1)$$

Na eq. (1) tem-se o ganho K , a constante de tempo mecânica, $\tau_1 = (J_L + J_m)/B$, que considera o momento de inércia da carga J_L , do motor J_m , e o atrito viscoso B . A constante de tempo elétrica do sistema, $\tau_2 = L_A/R_A$, com L_A indutância de armadura e R_A resistência de armadura. O pólo na origem indica controle de posição. Sem este pólo, obtém-se um sistema de controle de velocidade.

Com o tempo sistemas servoposicionadores sofrem alterações devido ao envelhecimento de seus componentes e suas não-linearidades inerentes. A identificação do ganho não-linear com redes neurais artificiais contorna este fato, e o seu uso no controle em malha fechada ameniza os efeitos dessa não-linearidade.

O processo de auto-sintonia consiste em identificar os parâmetros não lineares do processo com sinais de teste adequados. Utilizou-se uma forma de onda quadrada com degraus de amplitude crescente. Uma vez identificada a zona-morta esta é compensada

através da característica inversa correspondente (atrito coulombiano). Para a saturação utiliza-se uma rede adicional que evita o *Wind-Up* do canal integrador [7]. A compensação do ganho dependente do ponto de operação é feita utilizando-se uma rede neural do tipo perceptron multicamadas.

3. A Rede Perceptron Multicamadas (MLP)

Como mencionado, os sistemas posicionadores em geral, apresentam um ganho dependente da faixa de operação. Desta forma, o ganho do sistema não pode ser representado por uma constante e sim por uma função que depende da excitação. Este é um caso típico de mapeamento de funções que pode ser implementado adequadamente por redes neurais artificiais. Para identificar e compensar o ganho não-linear foi escolhida a rede neural *MLP com treinamento Backpropagation* por apresentar uma operação rápida e ser excelente em reconhecimento de padrões e mapeamento. Se o mapeamento for unívoco, a rede não terá restrições em ser treinada para tal.

Em uma rede neural *MLP* cada neurônio computa uma soma ponderada de suas entradas, e passa esta soma em uma função não-linear limitada [2].

Sua arquitetura básica em nível de microestrutura apresenta uma função de transferência limitada, que no caso do ganho não-linear, utilizou-se a função logística (sigmóide). Em nível de mesoestrutura, tem-se duas ou mais camadas com conexão *feedforward*.

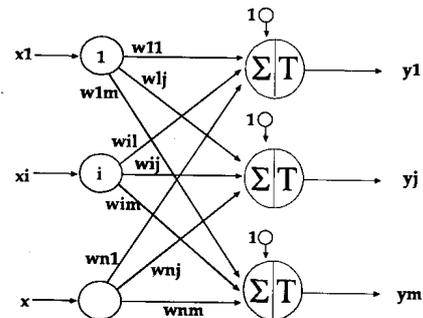


Figura 2 – Topologia da Rede MLP.

O treinamento da rede ocorre da seguinte forma: Apresenta-se o padrão de entrada à camada de entrada da MLP com treinamento *backpropagation* com n unidades. Na próxima camada, calcula-se a entrada como sendo a soma ponderada de todas as ativações da primeira camada (esta soma ponderada é obtida pelo produto do vetor de entrada da primeira camada e a matriz de pesos). Adiciona-se do valor obtido um valor limiar pré-fixado. O resultado é aplicado à função de ativação para obter a saída do neurônio. Portanto, a

ativação do j-ésimo neurônio na segunda camada é dada por:

$$s_j = \sum_i w_{ij} \cdot x_{ij} + w_{0j} \quad (2)$$

$$y_j = f(s_j) = \frac{1}{1 + e^{-s_j/Q}} \quad (3)$$

onde y_j é a saída do j-ésimo neurônio;

f é a função de ativação;

x_{ij} é a i-ésima entrada do neurônio j;

w_{ij} é o peso na conexão da entrada i para o j-ésimo neurônio; j=0 define o valor de limiar;

Q é o espalhamento da sigmóide (também denominado temperatura).

Se mais de uma camada é usada, este processo é repetido para cada camada, que serve de entrada para a seguinte (conexão *feedforward*).

Para treinar a rede, um conjunto de dados de treinamento deve ser obtido através de padrões de treinamento. O conjunto de treinamento deve conter amostras representativas de cada região de operação que se deseja que a rede “aprenda”. Trata-se portanto de um treinamento supervisionado.

Ao iniciar o treinamento de uma rede, determinam-se valores randômicos iniciais para os pesos. Estes valores são limitados; geralmente entre -1 e +1.

3.1. Algoritmo de Treinamento:

No treinamento *backpropagation* cada neurônio recebe várias entradas e gera uma saída. O processamento é feito em seqüência a partir da camada de entrada, passando de camada em camada. Já os neurônios de uma mesma camada atuam em paralelo. O algoritmo *backpropagation* só exige que a função de transferência seja diferenciável em qualquer valor do domínio. Pode-se citar duas funções que satisfazem esta exigência: as funções sigmoidais tangente hiperbólica e logística.

Adotando a seguinte convenção:

$x_i^{(k)}$ – saída da função de ativação sigmoidal do elemento i da camada k.

$s_j^{(k)}$ – soma ponderada dos pesos pelas entradas na saída do módulo somador e entrada ao módulo de ativação.

$w_{ij}^{(k)}$ – pesos das conexões sinápticas na entrada do elemento j da camada k, onde i é o índice de conexão. Para j = 0, é o peso da entrada viés (que sempre recebe o valor fixo de entrada +1).

N_k – número de elementos de processamento da camada k.

Os componentes do vetor X de entrada são $x_i = x_i^{(0)}$. Os componentes do vetor Y de saída são $y_i = x_i^{(m)}$. $D = [d_i]$ é o vetor de saída desejada.

Assim, as equações do treinamento são:

$$s_j^{(k)} = w_{0j}^{(k)} + \sum_{i=1}^{N_k} w_{ij}^{(k)} \cdot x_{ij}^{(k-1)} \quad (4)$$

$$x_j^{(k)} = f(s_j^{(k)}) \quad (5)$$

onde f é função contínua diferenciável.

4. O Ganho Não-Linear

Utilizando-se o aparato experimental descrito em 3 foi obtida a seguinte resposta de velocidade.

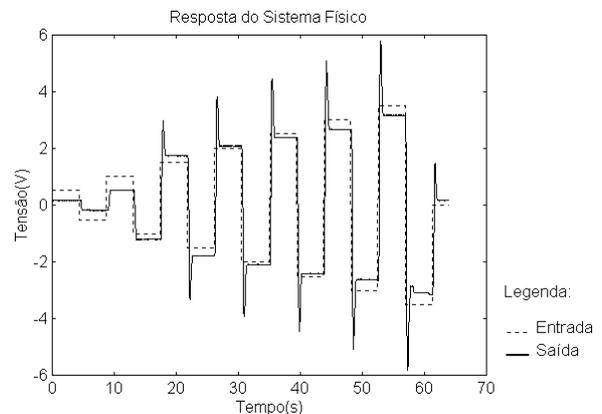


Figura 3 – Resposta Real do Sistema Físico

Nesta resposta vários efeitos não-lineares são visíveis. Nos degraus de pequena amplitude sobressai o efeito da zona morta. O sinal de controle é muito pequeno e passa a maior parte do tempo dentro da zona morta, causando o não acompanhamento do sistema à estes degraus. No último degrau negativo desta figura nota-se também um fenômeno causado pelo atrito, denominado usualmente de *slip-stick*: Quando o sinal de controle é alto o suficiente para vencer o atrito estático, devido ao acúmulo de canal integrador (intrínseco ao sistema) ocorrem saltos do sinal de saída, como mostra a Fig. 3.

Para destacar o efeito do ganho não-linear, foi feita uma ampliação da resposta real do sistema na Fig. 4. Para cada degrau, o sistema responde com um ganho diferente.

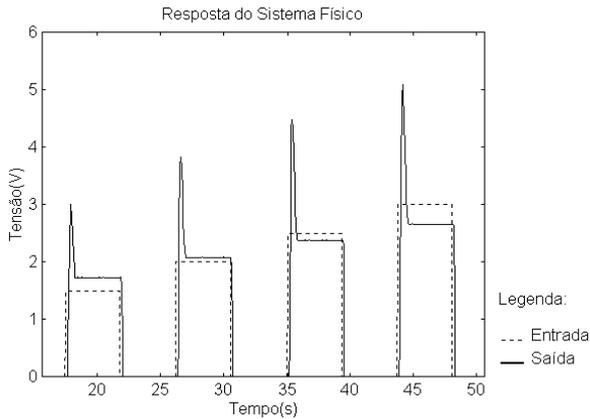


Figura 4 – Caracterização do Ganho Não-Linear

O sistema tenta acompanhar os degraus, mas devido à variação do ganho, ocorrem diferentes erros em regime permanente.

5. Identificação do Ganho Não-Linear

Para compensar o ganho não-linear propõe-se identificar o sistema através de uma rede neural e utilizá-la para obter-se uma resposta linear. Treinando a rede como um modelo inverso é possível saber qual a entrada deverá excitar o sistema para se obter a saída linearizada. Desta forma, apresenta-se como entrada para a rede a saída do sistema que se deseja obter e a rede retorna qual entrada deve ser usada para se obter tal resposta, conforme ilustrado na Fig. 5.

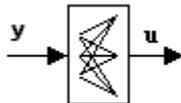


Figura 5 – Modelo Inverso do Sistema

Portanto, para treinar a rede mede-se a saída do motor em malha aberta para várias amplitudes de entrada. As saídas obtidas são apresentadas para a rede como entrada e os valores de excitação são apresentados como saídas desejadas. Por isso, após o treinamento, a rede será denominada de modelo inverso do sistema.

Com essa metodologia, a compensação do sistema pode ser feita mesmo com variações paramétricas devido ao envelhecimento de seus componentes e devido à presença de não-linearidades inerentes a servoposicionadores.

6. Compensação do Ganho Não-Linear

Para utilizar o controlador PID clássico no controle de posição em malha fechada, é preciso linearizar o sistema primeiro. Esta linearização é feita através da introdução na malha da rede utilizada para identificar o ganho do sistema real.

A compensação do ganho não-linear é feita da seguinte forma: para o sinal de controle u deseja-se y linear. A rede deve dizer qual u' deverá excitar o motor para gerar tal y linear (ou y'). Como mostra a Fig. 6, o sinal de controle u é multiplicado pelo ganho médio real do sistema (que pode ser obtido na fase de identificação tirando-se a média dos valores máximos de saída divididos pelas amplitudes dos degraus de entrada respectivos).

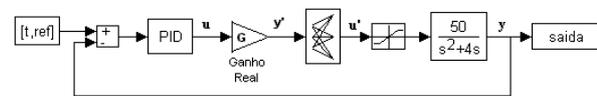


Figura 6 – Compensação do Ganho Não-Linear

Este valor (y') representa o sinal de saída linear desejado (y) e é apresentado como entrada para a rede. A rede, então, retorna o sinal de controle (u') que deve excitar o sistema para obter tal saída linear.

7. Implementação em Tempo-Real

Na parte experimental, foi implementado em laboratório um programa em C++ Builder para aquisição de dados em malha aberta e controle de posição em malha fechada. O algoritmo de treinamento da rede MLP com treinamento *backpropagation* em C++ foi adaptado de [1].

Como plataforma de hardware foi utilizado um PC-Pentium rodando Windows 95[®]. A placa PC30D da Zcom foi utilizada para realizar aquisição de dados e gerar o sinal de controle a uma frequência de amostragem de 16 Hz.

Para permitir a operação em tempo-real foi realizada a sincronização do processo através de DMA disparado por um *timer* residente na placa PC30. O software em *foreground* monitora o processo de aquisição, aguardando a chegada de uma nova amostra. Então é calculado um novo sinal de controle, que é escrito imediatamente no D/A.

Este esquema não garante “*hard real-time*”, uma vez que se forem disparados outros processos em *foreground* pelo Windows 95[®], poderão ocorrer atrasos no processamento de certas amostras. Porém, como em geral utiliza-se uma sobreamostragem de 4 a 8 vezes a frequência de Nyquist, não há prejuízo significativo

para o controle deste sistema. O sistema mecânico funciona como um filtro passa-baixas e estas pequenas variações são imperceptíveis na saída do servoposicionador.

Para a obtenção dos resultados aqui apresentados não foram disparados outros programas pelo Windows 95®, enquanto era feita a aquisição dos dados.

Outros sistemas operacionais, abertos e de larga difusão no meio acadêmico, como o Linux, deverá ser utilizado em implementações futuras. Neste caso existe padronização para as extensões que implementam o gerenciamento de tarefas em tempo real.

8. Resultados Obtidos

Para treinar a rede, foi adquirida a resposta do sistema em malha aberta, conforme Fig. 7.

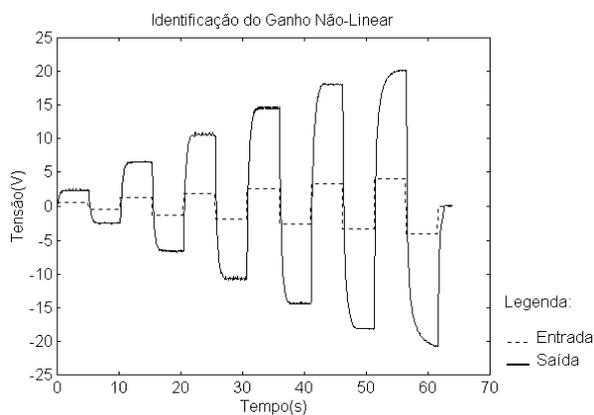


Figura 7 – Dados para Identificação do Ganho Não-Linear

De cada degrau, foi calculada a média dos dez últimos pontos de saída, ou seja, o valor máximo de resposta à cada degrau. Após calcular o ganho médio dividindo este valor pela amplitude do sinal de entrada, todos os valores obtidos foram normalizados de forma que ficassem entre 0 e 1. Esta normalização se deve ao fato de que a função logística está definida entre estes limites de normalização. A faixa de saída do sistema posicionador utilizado é de -25 a 25 volts. A de entrada é de -5 a 5 volts. Assim, para obter a entrada da rede dividiu-se a saída obtida do sistema por 50 e somou-se 0,5. Para a saída da rede, dividiu-se a entrada do sistema por 10 e somou-se 0,5.

Como parâmetros de projeto da rede neural artificial foram adotados [1]:

- Um neurônio na camada de entrada;
- Um neurônio na camada de saída;
- Quatorze neurônios na camada escondida;
- *Learning rate* de 0,5;
- Pior erro objetivo de 0,06;

- Função de ativação logística (sigmóide) em todas as camadas;
- Temperatura da sigmóide de 0,9;
- Alpha de 0,1 (espalhamento da sigmóide).

A rede em C++ apresentou um treinamento rápido. Ela alcançou o pior erro de 0,06 em aproximadamente 13 segundos, convergindo em 370 épocas.

Utilizando esta rede para compensar os efeitos não-lineares em malha fechada, cada sinal de controle calculado é multiplicado pelo ganho real e apresentado à rede, que gera o sinal de controle para obter o sinal linear desejado.

Na Fig. 8 é apresentada a resposta do sistema com a compensação do ganho não-linear. Nota-se que o sistema responde em quase todos os degraus com o mesmo ganho, diminuindo o erro em regime.

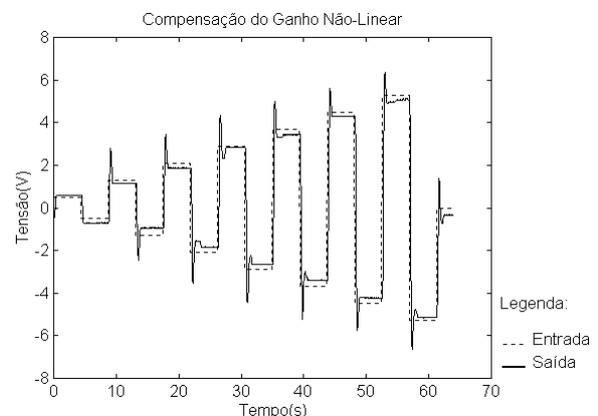


Figura 8 – Controle de Posição com Compensação do Ganho Não-Linear

Desta forma, a compensação do ganho não-linear pela rede neural foi satisfatória, amenizando os efeitos desta não-linearidade no controle de posição.

9. Conclusões

Neste artigo foram apresentados resultados iniciais de uma proposta de identificação e compensação do ganho dependente da faixa de operação, não-linearidade inerente a sistemas posicionadores. Uma rede MLP com treinamento *backpropagation* foi utilizada para identificar o ganho não-linear, e após treinada desta forma, foi adicionada à malha fechada para o controle de posição. À entrada da rede foi apresentado o sinal de controle multiplicado pelo ganho médio do sistema, ou seja, a saída linear desejada. A rede retorna como saída, o sinal de controle que deve excitar o sistema para obter tal saída linear. Os resultados obtidos comprovam experimentalmente a validade do método proposto.

Utilizando redes neurais para identificação de não-linearidades de sistemas posicionadores, tais como o ganho não-linear, é possível compensar seus efeitos indesejados no controle de posição. A rede neural aprende o comportamento dinâmico do sistema a ser controlado e se ajusta automaticamente às variações paramétricas que porventura possam surgir ao longo do tempo, desde que treinada novamente. Desta maneira, a rede proporciona uma auto-sintonia, permitindo sua utilização em qualquer sistema servoposicionador.

Agradecimentos: *Os autores agradecem ao CNPq pela concessão de uma bolsa PIBIC, durante parte da execução deste projeto. Também agradecem ao Departamento de Engenharia Elétrica e ao GRACO pela disponibilização dos recursos materiais necessários ao desenvolvimento da pesquisa.*

Referências

- [1] A. S. Pandya, *Pattern Recognition With Neural Networks In C++*. Florida Atlantic University, Boca Raton, Florida, 1996 By Crc Press Llc.
- [2] M. A. Tafner, M. De Xerez E I. W. R. Filho, *Redes Neurais Artificiais – Introdução E Princípios De Neurocomputação*. Editora Eko, Blumenau, SC, 1995.
- [3] C. Loesch E S. T. Sari, *Redes Neurais Artificiais – Fundamentos E Modelos*. Editora Da Furb, Blumenau, SC, 1995.
- [4] Feedback Inc – Kit MS150.
- [5] Módulo Consultoria E Informática, *Linguagem C – Programação E Aplicações*. Livros Técnicos E Científicos Editora S.A., Rio De Janeiro, RJ, 1987.
- [6] P. Aitken E B. Jones, *C – Guia Do Programador*. Berkeley Brasil Editora, Rio De Janeiro, RJ, 1994.
- [7] Gene F. Franklin, J. David Powell, Abbas Emami-Naeini, *Feedback Control Of Dynamic Systems*, Editora Addison-Wesley Publishing Company, 3.^a Edição, 1994.
- [8] Coelho, M. A. F. E., *Simulação Do Kit Ms150 – Feedback*, Editora Universidade De Brasília, DF, 1992.
- [9] A. Bauchspiess, A. C. Drummond E A. R. S. Romariz, *Redes Neurais De Base Radial Para O Servocontrole Auto-Ajustável De Sistemas Posicionadores*. Iii Congresso Brasileiro de Redes Neurais, Pags. 472 A 477, Florianópolis, SC, 1997.