

IDENTIFICAÇÃO DE UM SISTEMA TÉRMICO USANDO REDE NEURAL DE B-SPLINES

Leandro dos Santos Coelho

Laboratório de Automação e Sistemas, LAS
Programa de Graduação em Engenharia de Produção e Sistemas, PPGEPS
Pontifícia Universidade Católica do Paraná – PUCPR
Rua Imaculada Conceição, 1155, CEP 80215-901, Curitiba, PR, Brasil
leandro.coelho@pucpr.br

Viviana Cocco Mariani

Programa de Pós-Graduação em Engenharia Mecânica
Pontifícia Universidade Católica do Paraná – PUCPR
Rua Imaculada Conceição, 1155, CEP 80215-901, Curitiba, PR, Brasil
viviana.mariani@pucpr.br

Resumo. As redes neurais de B-splines empregam funções de base para aproximar uma função contínua, possivelmente multidimensional. As redes de B-splines podem ser comparadas com as redes neurais de função de base radial, que utilizam funções Gaussianas como funções elementares de base. As redes B-splines são geralmente treinadas por algoritmos baseados em informação do gradiente, o que pode levar a um ponto de mínimo local durante o procedimento de treinamento. Para superar os problemas encontrados pelos métodos convencionais de treinamento, o método de otimização com evolução diferencial utiliza uma busca estocástica para otimização global das redes B-splines. Uma abordagem de evolução diferencial para ajustar os pontos de controle (knots) de uma rede neural de B-splines é proposto neste artigo. Os resultados de simulação para identificação um passo à frente de um sistema térmico SISO (Single-Input Single Output) demonstram a eficiência da abordagem de rede neural de B-splines proposta.

Palavras chave: sistema térmico, rede neural de B-splines, identificação não-linear, evolução diferencial

1. Introdução

As redes neurais são metodologias da inteligência computacional inspiradas em estudos do cérebro e do sistema nervoso humano. As redes neurais podem ser consideradas como sistemas dinâmicos não lineares consistindo de elementos de processamento (neurônios), que são interconectados entre si, com ponderações ajustáveis (Haykin, 1996). Entre as características relevantes das redes neurais têm-se: processamento paralelo, aprendizado, memória associativa e distribuída. Estas características são inspiradas nas redes neurais biológicas. As redes neurais providenciam, usualmente, conhecimento quantitativo não-paramétrico sendo adequadas para identificação de sistemas, aprendizado e adaptação em sistemas com comportamento dinâmico complexo (Coelho e Villa, 2005).

Em particular, as redes neurais de B-spline (RN-BS) são uma alternativa de modelo não-linear para problemas de regressão e classificação de padrões. Parte do crescente interesse pelas RNs-BS é proveniente do fato delas possuírem fundamentação estatística e serem consideradas aproximadores universais de funções com dada precisão, da mesma forma que as redes neurais função de base radial (Guerra e Coelho, 2005), realizando um mapeamento das entradas de rede de dimensão vetorial m , em uma saída de dimensão vetorial n de qualquer função vetorial contínua. Neste contexto, a literatura sobre as RNs-BS tem apresentado vários estudos em identificação e controle de processos não-lineares (An *et al.*, 1994; Wang *et al.*, 1995; Lightbody *et al.*, 1997; Yiu *et al.*, 2001; Chen *et al.*, 2004).

As RNs-BS apresentam potencialidades de generalização local, onde a interferência no aprendizado é minimizada e o aprendizado é relativamente rápido, devido ao número mínimo de pesos ajustados a cada vetor de treinamento apresentado à rede neural. Na literatura, várias abordagens de treinamento de RN-BS têm sido apresentadas, entre as quais usando algoritmo ANOVA (An *et al.*, 1994; de Vries *et al.*, 1998), algoritmos genéticos (Leu, 2004), entre outros (Wang *et al.*, 1995).

A contribuição deste trabalho é avaliar as potencialidades de usar uma abordagem de treinamento da RN-BS baseada em otimização por evolução diferencial (ED). A ED é um paradigma da computação evolutiva (ou evolucionária) desenvolvido por Rainer Storn e Kenneth Price (Storn e Price, 1995; Storn, 1997) para problemas de otimização não-linear contínua. Basicamente, a ED realiza mutações nos vetores pela adição ponderada de diferenças aleatórias entre eles. A ED, neste contexto, determina os nós de cada uma das funções de base B-spline da RN-BS em uma aplicação de identificação não-linear de um sistema térmico.

O artigo é organizado da seguinte forma. A descrição da RN-BS e o procedimento de treinamento usando ED são detalhados na seção 2. Uma breve descrição do sistema térmico e a análise dos resultados obtidos na identificação são detalhadas na seção 3. Finalizando, a conclusão e comentários sobre futura pesquisa são apresentados na seção 4.

2. Rede neural de B-spline

Uma RN-BS consiste de uma rede neural da classe de redes *feedforward*, sendo que a RN-BS é composta de funções B-splines. Na RN-BS, cada função de base é composta de q segmentos polinomiais. Aqui existe uma relação simples e recursiva estável para avaliação da pertinência de uma função de base B-spline de ordem q , tal que

$$N_q^j(x) = \left(\frac{x - \lambda_{j-q}}{\lambda_{j-1} - \lambda_{j-q}} \right) N_{q-1}^{j-1}(x) + \left(\frac{\lambda_j - x}{\lambda_j - \lambda_{j-q+1}} \right) N_{q-1}^j(x) \quad (1)$$

$$N_1^j(x) = \begin{cases} 1 & \text{se } x \in I_j \\ 0 & \text{outros casos} \end{cases} \quad (2)$$

onde $N_q^j(\cdot)$ é definido como a j -ésima função de base univariada de ordem q ; λ_j é o j -ésimo nó (*knot*) e I_j é o j -ésimo intervalo. A saída da RN-BS é dada por

$$\hat{y}_k = f(x_k) = \sum_{j=1}^p w_j N_q^j(x_k) \quad (3)$$

sendo p o número de *knots*, x_k e \hat{y}_k são as entradas e saída da rede, respectivamente; w_j é a ponderação atribuída para a j -ésima função de base e $N_q^j(\cdot)$ é dada pela equação recursiva (2). O índice j é associado com a região de suporte local $\lambda_{(j-q)} \leq x \leq \lambda_{(j)}$, enquanto o índice q indica a ordem das funções de base (Harris *et al.*, 1993).

A qualidade da aproximação depende da alocação de nós das funções B-spline. O objetivo da otimização da RN-BS usando ED é a determinação dos nós de cada uma das funções de base B-spline. Entretanto, neste trabalho o número de funções de base é escolha do projetista.

2.1. Otimização da RN-BS usando evolução diferencial (ED)

Os algoritmos evolucionários (ou evolutivos) são ferramentas promissoras para busca, otimização, aprendizado de máquina e para resolução de problemas de projeto. Estes algoritmos utilizam a evolução simulada para busca de soluções em problemas complexos (Bäck *et al.*, 1997; Whitley, 2001). Os algoritmos evolucionários são baseados em uma população de indivíduos, onde cada um representa um ponto de busca no espaço de soluções potenciais de um dado problema. Os algoritmos evolucionários possuem alguns procedimentos de seleção baseados na aptidão (*fitness*) dos indivíduos mais aptos, operadores de cruzamento e mutação.

Os algoritmos evolucionários incluem algoritmos genéticos, programação evolucionária, estratégias evolutivas, programação genética, entre outras variantes. Segundo Bäck *et al.* (1997), existem diversas variantes de implementações de algoritmos evolucionários, no entanto, as diferenças principais repousam na representação dos indivíduos, no projeto dos operadores de cruzamento e/ou mutação e nos mecanismos de seleção.

O operador de seleção emprega o princípio de sobrevivência dos indivíduos mais aptos, através de uma metáfora aos procedimentos de reprodução assexuada e seleção natural, de acordo com o grau de adaptação do indivíduo ao ambiente. O objetivo básico do operador de seleção é enfatizar as melhores soluções que constituem uma população. O operador não cria nenhuma nova solução. Este operador seleciona as soluções relativamente aptas de uma população e remove as soluções remanescentes.

O operador de cruzamento é responsável pela troca de material genético entre os indivíduos, com maior probabilidade de reproduzirem os indivíduos mais aptos em relação ao ambiente. O operador de mutação modifica o valor dos genes do indivíduo e visa restaurar o material genético perdido ou não explorado em uma população. Este operador, quando projetado de forma apropriada, pode prevenir a convergência prematura de um algoritmo evolutivo para soluções sub-ótimas e manter a diversidade da população.

Dentre estes algoritmos evolucionários destaca-se o algoritmo de evolução diferencial. A ED foi desenvolvida por Price e Storn (Storn e Price, 1995; Storn, 1997) visando a busca por melhores resultados com uma abordagem um

pouco diferente da utilizada nos algoritmos genéticos e em estratégias evolutivas. Algumas das potencialidades da evolução diferencial têm-se a rapidez de convergência da otimização, a facilidade de implementação e validação.

Basicamente, a ED realiza mutações nos vetores pela adição ponderada de diferenças aleatórias entre eles. A escolha da ED para problemas de otimização é baseada nas características de que a ED:

- é um algoritmo de busca estocástica que é motivado pelos mecanismos de seleção natural;
- é menos susceptível a mínimos (ou máximos) locais, pois busca a solução ótima global pela manipulação de uma população de soluções candidatas, ou seja, busca um número de diferentes áreas simultaneamente no espaço de busca;
- não requer informação de derivadas para o cálculo da função de aptidão;
- manipula diretamente os números de ponto flutuante (fenótipo), diferente dos algoritmos genéticos canônicos ou binários que manipulam cadeias de bits (genótipo), o que diminui o custo computacional necessário para transformação de uma representação genotípica em fenotípica;
- geralmente não necessita de populações grandes para funcionar eficientemente.

Na ED clássica, cada variável (indivíduo) é representada por um valor real (ponto flutuante). A variante implementada neste trabalho foi a ED/rand/1/bin, que é regida pelas seguintes etapas:

Etapa 1: *Iniciar os parâmetros de controle da evolução diferencial*: O projetista deve escolher os parâmetros de controle da ED, tais como tamanho da população (M), limites (máximos e mínimos) das variáveis de otimização, taxa de mutação ($f_m(t)$), taxa de cruzamento (CR) e o critério de parada do procedimento de otimização.

Etapa 2: *Iniciar o contador de gerações*: Atribuir geração inicial, $t=1$.

Etapa 3: *Iniciar a população inicial de indivíduos (soluções)*: Gerar uma população inicial aleatória, com distribuição uniforme, de soluções factíveis à resolução do problema em questão, onde as regras de “reparo” garantem que os valores atribuídos às variáveis estão internas às fronteiras delimitadas pelo projetista.

Etapa 4: *Avaliar os indivíduos da população*: Avaliar a função objetivo (custo) de cada um dos indivíduos da população.

Etapa 5: *Aplicar a operação de mutação (ou operação diferença)*: A mutação é uma operação que adiciona um vetor diferencial para o vetor dos indivíduos da população, de acordo com a equação:

$$z_i(t+1) = x_{i,r_1}(t) + f_m(t)[x_{i,r_2}(t) - x_{i,r_3}(t)] \quad (4)$$

sendo $i=1,2,\dots,M$ o índice do indivíduo da população; $j=1,2,\dots,n$ é a posição do indivíduo em um espaço n -dimensional; t é a geração (tempo); $x_i(t) = [x_{i_1}(t), x_{i_2}(t), \dots, x_{i_n}(t)]^T$ consiste da posição do i -ésimo indivíduo de uma população de M vetores n -dimensionais; $z_i(t) = [z_{i_1}(t), z_{i_2}(t), \dots, z_{i_n}(t)]^T$ é responsável pela posição do i -ésimo indivíduo de um vetor que sofrerá mutação; r_1, r_2 e r_3 são valores inteiros mutuamente diferentes, selecionados aleatoriamente com distribuição uniforme do conjunto $\{1, 2, \dots, i-1, i+1, \dots, N\}$; $f_m(t) > 0$ é um parâmetro real denominado de taxa de mutação, que controla a amplificação da diferença entre os dois indivíduos de índices r_2 e r_3 para evitar a estagnação da busca e é usualmente projetado com valores no intervalo $[0,4; 1,0]$.

Etapa 6: *Aplicar a operação de cruzamento*: Após a operação de mutação, o cruzamento (ou recombinação) é aplicado a população. O cruzamento é empregado para gerar um novo vetor tentativa ou vetor doador (*trial vector*) pela substituição de certos parâmetros do vetor destino (*target vector*) pelos seus parâmetros correspondentes ao vetor doador, estes gerados aleatoriamente.

Nesta operação, para cada vetor, $z_i(t+1)$, um índice $rnbr(i) \in \{1, 2, \dots, n\}$ é escolhido aleatoriamente usando uma função densidade de probabilidade uniforme, e um vetor denominado de vetor tentativa, $u_i(t+1) = [u_{i_1}(t+1), u_{i_2}(t+1), \dots, u_{i_n}(t+1)]^T$. Neste caso é gerado um novo vetor tal que,

$$u_{i_j}(t+1) = \begin{cases} z_{i_j}(t+1), & \text{se } (randb(j) \leq CR) \text{ ou } (j = rnbr(i)), \\ x_{i_j}(t), & \text{se } (randb(j) > CR) \text{ ou } (j \neq rnbr(i)) \end{cases} \quad (5)$$

sendo $randb(j)$ a j -ésima avaliação da geração de um número aleatório com distribuição uniforme no intervalo $[0, 1]$; e CR é a taxa de cruzamento (ou recombinação) no intervalo $[0, 1]$. Geralmente, o desempenho do algoritmo de ED depende do projeto de três variáveis: o tamanho da população, M , a taxa de mutação, $f_m(t)$, e a taxa de cruzamento, CR .

Etapa 7: *Aplicar a operação de seleção*: A seleção é um procedimento em que os “melhores” descendentes (indivíduos filhos) são produzidos. Para decidir se o vetor $u_i(t+1)$ será (ou não) um membro da população na próxima geração, ele é comparado com o vetor $x_i(t)$. Assim considerando que F denota a função objetivo sob minimização, então:

$$x_i(t+1) = \begin{cases} u_i(t+1), & \text{se } F(u_i(t+1)) < F(x_i(t)), \\ x_i(t), & \text{outros} \end{cases} \quad (6)$$

Neste caso, o custo de cada vetor tentativa $u_i(t+1)$ é comparado com seu vetor destino $x_i(t)$. Se o custo, F , do vetor destino $x_i(t)$ possui valor menor que o custo do vetor tentativa, é permitido ao vetor destino continuar na próxima geração. Caso contrário, o vetor destino é substituído pelo vetor tentativa na próxima geração.

Etapa 8: *Verificar se o critério de parada foi atendido*: Atribuir à geração $t = t + 1$. Retornar para a Etapa 4 até que o critério de parada seja atendido, usualmente o número de gerações, t_{max} .

Na figura 1 é apresentado um resumo do fluxo de dados da ED adotada neste trabalho.

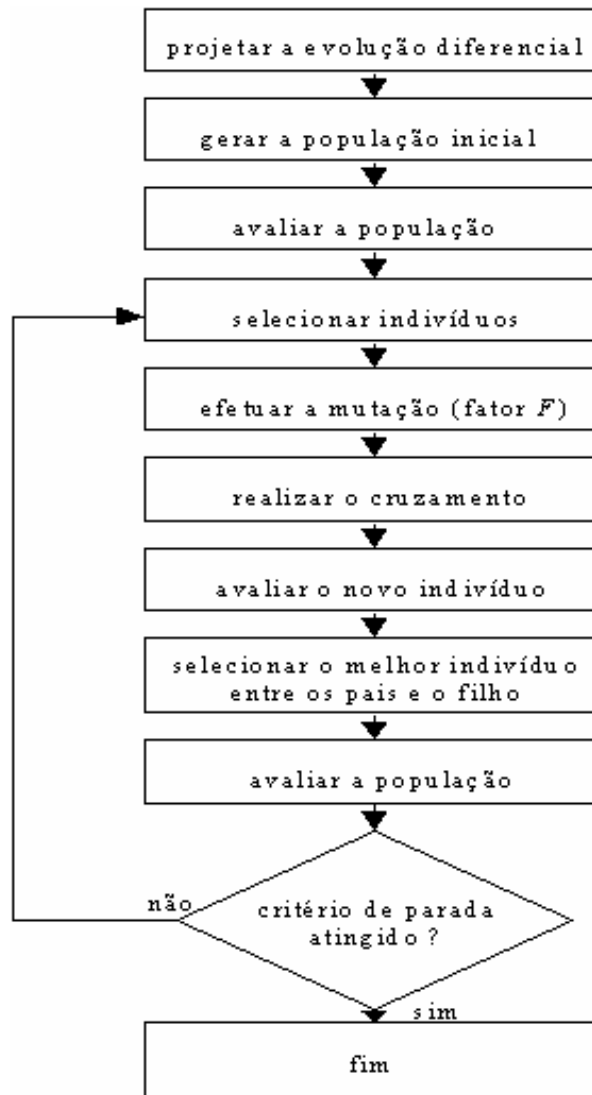


Figura 1 - Fluxograma da ED para otimização de projeto.

3. Identificação de um sistema térmico

A identificação de sistemas é uma área do conhecimento que envolve o estudo de formas de modelar e analisar matematicamente fenômenos a partir de dados.

A idéia da identificação de sistemas é permitir a elaboração do modelo matemático de um sistema dinâmico baseado em medidas coletadas pelo ajuste de parâmetros e/ou do modelo matemático, até que a saída do sistema aproxime, de forma adequada, os valores de saída desejada (Johansson, 1993). O procedimento para a identificação de um sistema consiste de quatro etapas básicas, conforme mostra a figura 2.

Um estudo de caso de identificação não-linear de um sistema térmico foi abordado neste trabalho, este disponível na base de dados *Database for the Identification of Systems* (Daisy, 2006).

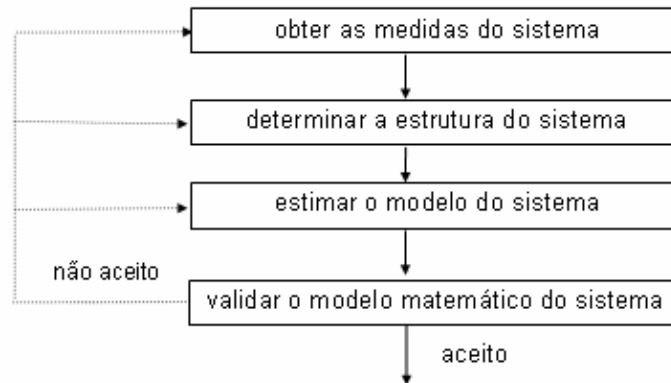


Figura 2 - Procedimento utilizado na identificação de um sistema.

O sistema térmico analisado consiste de uma lâmpada de halógena de 330 Watts suspensa sobre uma placa fina de aço. A saída do sistema consiste da medida de um termopar localizado abaixo da placa (Daisy, 2006). O período de amostragem dos dados é de 2 segundos e o procedimento de identificação, adotado neste trabalho, utiliza como entrada a tensão aplicada à lâmpada (em Volts) e como variável de saída é adotada a temperatura da placa, esta medida em graus Celsius (Dullerud & Smith, 1996).

O experimento foi configurado para utilizar as amostras 1 a 650 (2 a 1300 segundos) para a fase de estimação (otimização da RN-BS) e as amostras 651 a 800 (1302 a 1600 segundos), para a fase de validação (teste) da RN-BS. Os dados de entrada e a saída do sistema térmico avaliado são apresentados na figura 3.

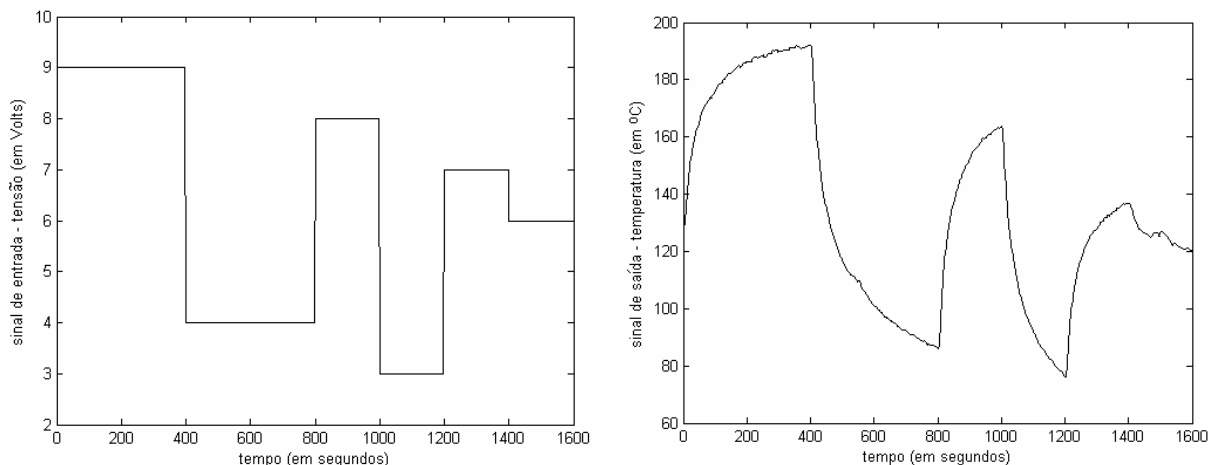


Figura 3 - Dados de entrada e saída do sistema térmico.

O procedimento de análise da identificação através de RN-BS usando ED é dividido nas seguintes etapas: (i) escolha da estrutura da RN-BS para representar o processo, (ii) determinação da otimização da RN-BS através da ED; (iii) estimação dos parâmetros do modelo matemático (*fase de estimação*), e (iv) a validação do modelo matemático (*fase de validação*).

Para identificação foi escolhido um modelo matemático para a representação do processo. A estrutura de modelo testada é o modelo multivariável *NARX* (*Nonlinear AutoRegressive with eXogenous inputs*) em concepção série-paralela com 3 entradas e 1 saída, isto é, as entradas $[u_1(t-1); y(t-2); y(t-1)]$ e a saída $\hat{y}(t)$.

A identificação do sistema térmico é apropriada se um critério de erro definido previamente pelo projetista está entre valores admissíveis às necessidades do projeto. O critério escolhido neste estudo (função custo a ser maximizada) foi o coeficiente de correlação múltipla regido pela equação

$$R^2 = 1 - \frac{\sum_{t=1}^{Na} (y(t) - \hat{y}(t))^2}{\sum_{t=1}^{Na} (y(t) - \bar{y})^2}, \quad (7)$$

onde Na é o número de amostras avaliado (1300 amostras na fase de estimação e 300 amostras na fase de validação), $y(t)$ é a saída real do processo, $\hat{y}(t)$ é a saída estimada pela RN-BS, \bar{y} é a média das medidas do sistema térmico.

Quando o valor de R^2 é igual a 1,0 indica uma aproximação exata do modelo aos dados medidos do processo. O valor de R^2 entre 0,9 e 1,0 é considerado suficiente para aplicações práticas, principalmente em projetos de identificação e sistemas de controle baseados em modelo.

Adota-se, neste artigo, uma ED com tamanho de população de 10 indivíduos (vetores solução) com critério de parada de 100 gerações. Foram testadas de 3 a 5 B-splines na camada intermediária para cada entrada da RN-BS. Entretanto, os resultados de identificação usando RN-BS com ED foram mais precisos com 3 funções B-splines na RN-BS onde o R^2 obtido foi 0,9998801 (na fase de estimação) e 0,9938608 (na fase de validação), ou seja, estes resultados significam um erro médio quadrático de 0,1768661 (estimação) e 0,1589295 (validação). Nas figuras 4 e 5 são apresentados, respectivamente, os sinais da saída real e estimada do processo e também o e o sinal de erro de predição obtidos usando a RN-BS com 3 funções B-splines.

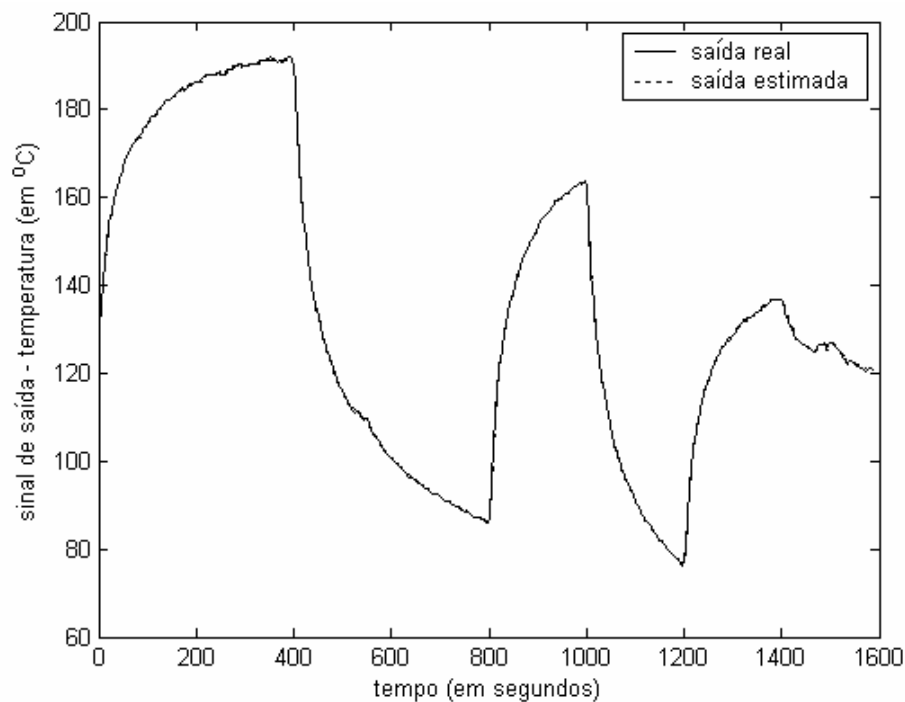


Figura 4 – Saída real e estimada usando 3 funções B-splines para cada entrada na RN-BS.

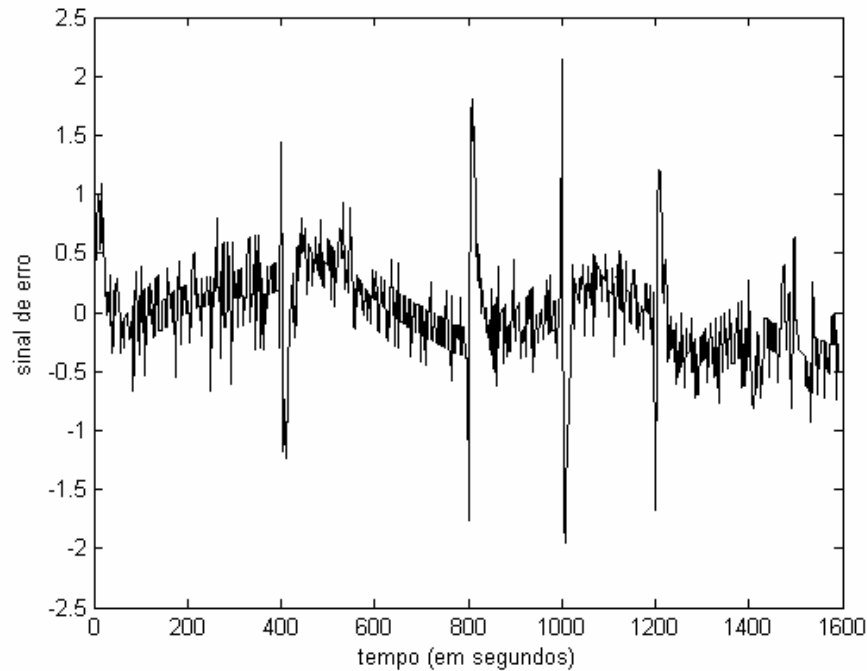


Figura 5 - Erro de predição (dos sinais apresentados na figura 4) usando 3 funções B-splines na RN-BS.

4. Conclusão e futura pesquisa

Neste artigo foi apresentada uma metodologia de otimização baseada em ED para ajustar os pontos de controle (*knots*) de uma RN-BS.

Os resultados de simulação para identificação um passo à frente, de um sistema térmico, demonstraram a eficiência da abordagem de otimização para uma RN-BS. Entretanto, a análise de um melhor compromisso entre precisão, complexidade computacional e ordens dos vetores de entrada deve ser abordada em futura pesquisa.

Em pesquisa futura, deseja-se utilizar procedimentos de otimização com múltiplos objetivos junto ao projeto do modelo de RN-BS. Neste contexto deseja-se determinar um conjunto de soluções de identificação dentro do qual se verifica o compromisso entre os diversos objetivos. O conjunto destas soluções se dá o nome de Pareto-ótimo. Neste contexto, deseja-se projetar uma RN-BS aliado a um procedimento de otimização visando a incorporação de informação da precisão do modelo (usando R^2), variância presente no residuo decorrente da diferença entre a saída do sistema e a saída estimada, e o número de parâmetros estimados (graus de liberdade do sistema ou complexidade do modelo).

5. Referências

- An, P. E., Brown, M., Harris, C. J., Lawrence, A. J., Moore, C. G., 1994, "Associative Memory Neural Networks: Adaptive Modeling Theory, Software Implementations and Graphical User Interface", Engineering Applications of Artificial Intelligence, Vol. 7, No. 1, p. 1-21.
- Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.), 1997, "Handbook of Evolutionary Computation", Bristol, Philadelphia: Institute of Physics Publishing, NY, USA, Oxford: Oxford University Press.
- Chen, Y. Q., Moore, K. L., Bahl, V., 2004, "Learning Feedforward Control using a Dilated B-spline Network: Frequency Domain Analysis and Design", IEEE Transactions on Neural Networks, Vol. 15, No. 2, pp. 355-366.
- Coelho, L. S., Villa, L. F., 2005, "Identificação de Processo Não-Linear Usando Rede Neural com Método de Treinamento Híbrido baseado em Filtro de Kalman e Máxima Descida", Anais do 7º Congresso Brasileiro de Redes Neurais, Natal, RN.
- Daisy, Database for the identification of systems, <http://www.esat.kuleuven.ac.be/sista/daisy/> [Acesso 16/02/2006], 2006.
- De Vries, T. J. A., Idema, L. J., Velthuis, W. J. R., 1998, "Parsimonious Learning Feed-Forward Control", Proceedings of European Symposium on Artificial Neural Networks, Bruges, Belgium, pp. 85-90.
- Dullerud, G., Smith, R., 1996, "Sampled Data Model Validation: an Algorithm and Experimental Application", International Journal of Robust and Nonlinear Control, Vol. 6, No. 9/10, pp. 1065-1078.

- Harris, C. J., Moore, C. G., Brown, M., 1993, "Intelligent Control: Aspects of Fuzzy Logic and Neural Nets", Word Scientific.
- Haykin, S., 1996, "Neural Networks", 2nd edition, Prentice-Hall, Upper Saddle River, NJ, USA.
- Johansson, R., 1993, "System Modeling and Identification", Prentice-Hall, Englewood Cliffs, USA.
- Leu, Y. -G., 2004, "Nonlinear System Modeling Using GA-based B-spline Membership Fuzzy-Neural Networks", Proceedings of 2nd International Conference on Autonomous Robots and Agents, Palmerston North, New Zeland, pp. 229-233.
- Lightbody, G., O'Reilly, P., Irwin, G. W., Elly, K., McCormick, J., 1997, "Neural Modeling of Chemical Plant Using MLP and B-Spline Networks", Control Engineering Practice, Vol. 5, No. 11, pp. 1501-1515.
- Storn, R., Price, K., 1995, "Differential Evolution: a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces", Technical Report TR-95-012, International Computer Science Institute, Berkeley, USA.
- Storn, R., 1997, "Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, Vol. 11, No. 4, pp. 341-359.
- Wang, C. H., Wang, W. Y., Lee, T. T., Tseng, P. S., 1995, "Fuzzy B-Spline Membership Function (BMF) and its Applications in Fuzzy-Neural Control", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 25, No. 2, pp. 841-851.
- Whitley, D., 2001, "An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls", Information and Software Technology, Vol. 43, pp. 817-831.
- Yiu, K. F. C., Wang, S., Teo, K. L., Tsoi, A. C., 2001, "Nonlinear System Modeling via Knot-Optimizing B-splines Networks", IEEE Transactions on Neural Networks, Vol. 12, No. 5, pp. 1013-1022.

HEATING SYSTEM IDENTIFICATION USING B-SPLINES NEURAL NETWORK

Leandro dos Santos Coelho
Laboratório de Automação e Sistemas, LAS
Programa de Graduação em Engenharia de Produção e Sistemas, PPGEPS
Pontifícia Universidade Católica do Paraná – PUCPR
Rua Imaculada Conceição, 1155, CEP 80215-901, Curitiba, PR, Brasil
leandro.coelho@pucpr.br

Viviana Cocco Mariani
Programa de Pós-Graduação em Engenharia Mecânica
Pontifícia Universidade Católica do Paraná – PUCPR
Rua Imaculada Conceição, 1155, CEP 80215-901, Curitiba, PR, Brasil
viviana.mariani@pucpr.br

Abstract

B-spline neural networks employ basis functions to approximate a continuous, possibly multidimensional, function. B-spline networks can be compared to, e.g., radial basis function networks, which use Gaussian functions as their elemental basis function. B-spline networks are usually trained by gradient-based methods, which may fall into local minimum during the learning procedure. To overcome the problems encountered by the conventional learning methods, the differential evolution provides a stochastic search for global optimization of B-spline networks. A differential evolution approach to adjust the control points (knots) of a B-spline neural network is proposed in this paper. Simulation results for one step ahead identification of a heating system in SISO (Single-Input Single-Output) conception demonstrate the effectiveness of the proposed B-spline neural network.

Keywords: heating system, B-spline neural network, nonlinear identification, differential evolution