

Aplicação de Máquinas de Aprendizado Extremo ao Problema de Identificação de Sistemas Não-Lineares

Renan D. B. Brotto Levy Boccato João M. T. Romano

Resumo— Este trabalho propõe a aplicação de máquinas de aprendizado extremo ao problema de identificação de sistemas não-lineares. Tal estrutura se credencia como uma opção promissora pelo fato de apresentar capacidade de aproximação universal e simplicidade no processo de adaptação dos parâmetros. A estratégia proposta é analisada em um cenário relacionado ao sistema de acionamento de uma válvula pneumática, sendo, então, comparada com dois modelos que exploram redes do tipo perceptron de múltiplas camadas. Os resultados obtidos mostram que a ELM é bastante adequada para a realização da tarefa.

Palavras-Chave— Identificação de sistemas não-lineares, Redes neurais artificiais, Máquinas de aprendizado extremo.

Abstract— This work proposes the application of extreme learning machines to the problem of nonlinear system identification. Such structure constitutes a promising option because it presents the capability of approximating nonlinear mappings and simplicity in the process of parameter adaptation. The proposed strategy is analyzed in a scenario related to the control system of a pneumatic valve, being compared with two other models based on multilayer perceptron networks. The obtained results show that the ELM is very suitable for the task.

Keywords— Nonlinear system identification, Artificial neural networks, Extreme learning machines.

I. INTRODUÇÃO

O problema de identificação de sistemas consiste na obtenção de um modelo que apresente a mesma dinâmica que o sistema desconhecido, a partir de medidas da entrada e saída do mesmo [1], [2]. Esquemáticamente, este problema é ilustrado na Figura 1, onde $x[n]$ é o sinal de entrada, $y[n]$ representa a saída fornecida pelo modelo, $\eta[n]$ representa um possível ruído de medida e $d[n]$ é o sinal efetivamente observado na saída do sistema desconhecido.

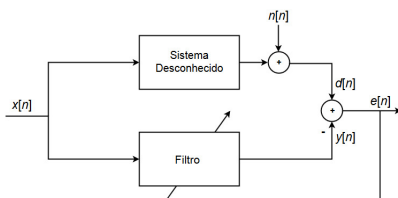


Fig. 1. Ilustração do problema de identificação de sistemas.

Com base no comparativo entre a resposta obtida, $y[n]$, e a resposta desejada, $d[n]$, um sinal de erro é produzido ($e[n]$), o

Renan D. B. Brotto, Departamento de Engenharia de Comunicações, Levy Boccato, Departamento de Engenharia de Computação e Automação Industrial, João M. T. Romano, Departamento de Engenharia de Comunicações. Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas-SP, Brasil, E-mails: rbrotto@decom.fee.unicamp.br, lboccato@dca.fee.unicamp.br, romano@dmo.fee.unicamp.br.

qual, juntamente com o sinal de entrada, fará parte do processo de adaptação do modelo, caracterizando, assim, um cenário de aprendizado supervisionado [3].

Em sua vertente não-linear, o problema lida com sistemas desconhecidos que apresentem ao menos uma componente não-linear [4], como é o caso dos sistemas de Wiener¹, ilustrado na Figura 2:

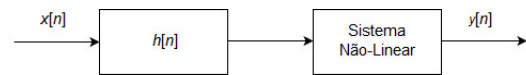


Fig. 2. Diagrama esquemático de um sistema de Wiener.

Para tratar o caso não-linear, uma possível abordagem pressupõe haver conhecimento *a priori* acerca da não-linearidade do sistema, o que, dentro de contextos específicos, permite a obtenção da solução ótima global (segundo o critério de quadrados mínimos), como é apresentado em [5] e [6]. Por outro lado, em situações em que informações sobre a não-linearidade não estão disponíveis, faz-se necessário o uso de estruturas mais genéricas, tais como as redes neurais artificiais [7]. Contudo, neste caso, a solução obtida está sujeita a ótimos locais. Portanto, existe um compromisso entre a complexidade paramétrica do modelo usado na identificação e a garantia de obtenção da solução ótima global.

Este trabalho tem como objetivo realizar a identificação de sistemas de Wiener através de máquinas de aprendizado extremo (ELMs, do inglês, *Extreme Learning Machines*), uma estrutura cujo processo de adaptação se destaca por sua simplicidade, mas que ao mesmo tempo apresenta capacidade de aproximação universal [8], [9], sendo, portanto, uma maneira de obter uma solução genérica para a identificação de sistemas e com capacidade para evitar ótimos locais, conforme será especificado na seção II. A seção III detalhará a metodologia empregada e a seção IV realizará o comparativo entre as máquinas de aprendizado extremo com regularização e outras duas arquiteturas de redes neurais: 1) uma rede perceptron de múltiplas camadas (MLP, do inglês *Multilayer Perceptron*), com uma camada intermediária ajustada via algoritmo SCGM (do inglês, *Scaled Conjugate Gradient Modified*) [7] e 2) MLP com duas camadas intermediárias (uma linear e outra não-linear), ajustada via algoritmo de evolução diferencial modificado [10]. Por fim, a seção V sintetiza as observações referentes ao desempenho das ELMs no problema de identificação de sistemas e aponta algumas perspectivas de trabalhos futuros.

¹Usualmente, os sistemas de Wiener apresentam uma não-linearidade sem memória.

II. MÁQUINAS DE APRENDIZADO EXTREMO (ELM)

As máquinas de aprendizado extremo utilizadas neste trabalho possuem uma arquitetura semelhante àquela associada às redes MLP, com somente uma camada intermediária, como mostrado na Figura 3.

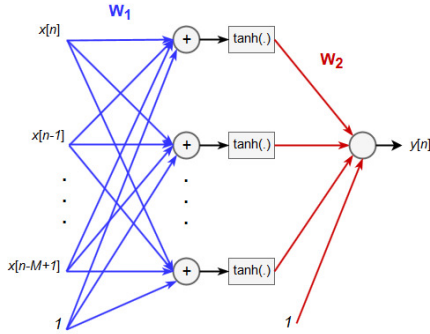


Fig. 3. Arquitetura básica de uma ELM.

O diferencial das ELMs é que apenas os pesos sinápticos da camada de saída (representados pela matriz \mathbf{W}_2 na figura acima) são ajustados tendo em vista o sinal de erro entre a saída da rede e um sinal de referência. Os parâmetros da camada intermediária podem ser escolhidos de maneira aleatória, permanecendo, então, fixos durante o treinamento. É importante destacar que a ausência de ajuste de parâmetros não compromete a capacidade de aproximação do modelo neural, como mostrado por [8] e [11].

Uma vez que a camada de saída de uma ELM essencialmente consiste de um combinador linear, o processo de treinamento como um todo se reduz a um problema de regressão linear, cuja solução ótima pode ser obtida de forma fechada ou com o auxílio de algoritmos iterativos, como o LMS (do inglês, *Least Mean Squares*) e o RLS (do inglês, *Recursive Least Squares*) [3].

A matriz de entrada da rede é definida por:

$$\mathbf{X} = \begin{bmatrix} x[0] & x[1] & \dots & x[T-1] \\ x[-1] & x[0] & \dots & x[T-2] \\ \vdots & \vdots & \ddots & \vdots \\ x[-M+1] & x[-M+2] & \dots & x[T-M] \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{bmatrix},$$

a qual pode ser reescrita numa notação mais compacta:

$$\mathbf{X} = [\mathbf{x}_0 \quad \mathbf{x}_1 \quad \dots \quad \mathbf{x}_{T-1}],$$

onde M denota o número de atrasos no sinal de entrada e T o número de amostras disponíveis para treinamento e assume-se $x[i] = 0$ para $i < 0$. A linha unitária da matriz \mathbf{X} tem por finalidade introduzir o efeito das polarizações (*biases*) sobre os neurônios da camada intermediária.

Os pesos da camada de entrada são aleatoriamente inicializados e podem ser representados pela matriz \mathbf{W}_1 mostrada a seguir:

$$\mathbf{W}_1 = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \dots \quad \mathbf{w}_N],$$

com $\mathbf{w}_i \in \mathbb{R}^{(M+1) \times 1}$, para $i = 1, 2, \dots, N$, onde N é o número de neurônios na camada intermediária.

Portanto, é possível computar a saída de cada neurônio da camada intermediária para cada um dos padrões de entrada, gerando, assim, a matriz:

$$\mathbf{H} = \begin{bmatrix} f(\mathbf{w}_1^T \mathbf{x}_0) & f(\mathbf{w}_2^T \mathbf{x}_0) & \dots & f(\mathbf{w}_N^T \mathbf{x}_0) \\ f(\mathbf{w}_1^T \mathbf{x}_1) & f(\mathbf{w}_2^T \mathbf{x}_1) & \dots & f(\mathbf{w}_N^T \mathbf{x}_1) \\ \vdots & \vdots & \ddots & \vdots \\ f(\mathbf{w}_1^T \mathbf{x}_{T-1}) & f(\mathbf{w}_2^T \mathbf{x}_{T-1}) & \dots & f(\mathbf{w}_N^T \mathbf{x}_{T-1}) \end{bmatrix},$$

sendo $f(\cdot)$ a função de ativação do neurônio, tipicamente uma função sigmoidal.

Logo, a saída da rede neural é dada por:

$$\mathbf{y} = \mathbf{H}\mathbf{w}_2, \quad (1)$$

com $\mathbf{w}_2 \in \mathbb{R}^N \times 1$, $\mathbf{y} \in \mathbb{R}^T \times 1$ e $\mathbf{H} \in \mathbb{R}^T \times N$.

Seja

$$\mathbf{d} = [d[0] \quad d[1] \quad \dots \quad d[T-1]]^T$$

o vetor contendo as amostras do sinal de saída do sistema desconhecido, disponíveis para a etapa de adaptação dos pesos sinápticos da rede. A obtenção dos coeficientes \mathbf{w}_2 passa pela solução do seguinte problema de otimização, de acordo com o critério de quadrados mínimos:

$$\begin{aligned} \mathbf{w}_2^* &= \arg \min_{\mathbf{w}_2} J(\mathbf{w}_2) \\ &= \arg \min_{\mathbf{w}_2} \sum_{i=0}^{T-1} (d[i] - y[i])^2. \end{aligned} \quad (2)$$

Devido à equação (1), o problema de otimização dado por (2) se torna uma função estritamente convexa com respeito ao parâmetro \mathbf{w}_2 quando as colunas da matriz \mathbf{H} são linearmente independentes, permitindo, deste modo, obter a solução global, dada por [8]:

$$\mathbf{w}_2 = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{d}. \quad (3)$$

A fim de favorecer a capacidade de generalização da rede, é interessante introduzir mecanismos de regularização no processo de treinamento. Uma possibilidade simples consiste em impor uma penalização proporcional à norma do vetor de coeficientes, dando origem ao seguinte problema de otimização [12]:

$$\begin{aligned} \mathbf{w}_2^* &= \arg \min_{\mathbf{w}_2} J(\mathbf{w}_2) \\ &= \arg \min_{\mathbf{w}_2} \sum_{i=0}^{T-1} (d[i] - y[i])^2 + C \|\mathbf{w}_2\|_2^2, \end{aligned} \quad (4)$$

onde C é o fator que controla o peso da regularização no treinamento, cujo valor precisa ser adequadamente escolhido. A solução ótima para (4) é dada por:

$$\mathbf{w}_2 = (\mathbf{H}^T \mathbf{H} + C\mathbf{I})^{-1} \mathbf{H}^T \mathbf{d}. \quad (5)$$

III. METODOLOGIA

A máquina de aprendizado extremo com regularização, descrita anteriormente, será comparada com outras duas redes neurais no contexto do problema de identificação de sistemas não-lineares: 1) uma MLP com duas camadas intermediárias, uma linear e outra não-linear, com parâmetros ajustados via evolução diferencial modificada; 2) uma MLP com uma única camada intermediária, treinada via SCGM. É importante mencionar que a primeira arquitetura, formada por duas camadas intermediárias, tem como motivação imitar a própria estrutura do sistema não-linear a ser identificado (que é do tipo Wiener). Além disso, o método de otimização empregado em seu treinamento corresponde a uma modificação do algoritmo de evolução diferencial, que será descrito brevemente na próxima seção.

A. Algoritmo de Evolução Diferencial

No algoritmo de evolução diferencial [10], inicialmente uma população de P soluções candidatas é aleatoriamente criada segundo uma distribuição uniforme, sendo que cada solução candidata corresponde a um vetor de L posições. Cada indivíduo sofre um processo de mutação, segundo a equação (6):

$$\mathbf{q}_{i,G+1} = \mathbf{p}_{j_1,G} + F(\mathbf{p}_{j_2,G} - \mathbf{p}_{j_3,G}), \quad (6)$$

onde $\mathbf{p}_{i,G}$ denota o i -ésimo indivíduo da população na geração G , j_1 , j_2 e j_3 são índices mutuamente distintos (também diferentes de i) e F é uma constante real ($F \in [0, 2]$) que determina o tamanho do passo na direção do vetor diferença ($\mathbf{p}_{j_2,G} - \mathbf{p}_{j_3,G}$).

O passo seguinte é a aplicação de um mecanismo de *crossover*, o qual dará origem a novos indivíduos, denominados *trial vectors*, $\mathbf{r}_{i,G+1}$, como especificado abaixo:

$$r_{i,G+1}[k] = \begin{cases} p_{i,G}[k], & \text{se } \alpha \leq cr \text{ ou } k = l; \\ q_{i,G+1}[k], & \text{se } \alpha > cr, \end{cases} \quad (7)$$

com $k = 1, 2, \dots, L$, cr é a taxa de *crossover* ($cr \in [0, 1]$), α é uma variável aleatória com distribuição uniforme no intervalo $[0, 1]$ e com uma realização para cada valor de k e l é um índice previamente determinado.

Finalmente, é realizada a seleção sobre a população, que mantém nesta os melhores indivíduos dentre os originais ($\mathbf{p}_{i,G}$) e aqueles gerados via *crossover* ($\mathbf{r}_{i,G+1}$):

$$\mathbf{p}_{i,G+1} = \begin{cases} \mathbf{p}_{i,G}, & \text{se } fit(\mathbf{p}_{i,G}) > fit(\mathbf{r}_{i,G+1}); \\ \mathbf{r}_{i,G+1}, & \text{caso contrário,} \end{cases} \quad (8)$$

para $i = 1, 2, \dots, P$ e $fit(\cdot)$ representa a função de avaliação (*fitness*) das soluções candidatas.

Neste trabalho, empregamos uma versão modificada do algoritmo de evolução diferencial em que, após a etapa de seleção, um percentual R dos piores indivíduos da população é substituído por novos indivíduos aleatoriamente criados. Esta modificação visa evitar uma perda rápida de diversidade na população de soluções candidatas, o que poderia comprometer a exploração do espaço de busca.

As etapas anteriormente descritas são repetidas por um número máximo de gerações e, ao final, o melhor indivíduo é adotado como solução do problema.

B. Validação Cruzada por K -pastas

Devido à presença de um ruído de medida, existe a necessidade de realizar a validação cruzada sobre os dados disponíveis. Para tanto, foi adotado o método de validação cruzada de K -pastas (em inglês, *K-fold cross-validation*), descrito a seguir [7]:

- 1) Inicialmente, os dados disponíveis (amostras de $x[n]$ e $d[n]$) são divididos em K conjuntos, de tamanhos (aproximadamente) iguais;
- 2) Um dos conjuntos é usado para a validação da rede neural, enquanto que o restante é empregado em seu treinamento;
- 3) O passo 2) é repetido K vezes, de modo que cada um dos conjuntos formados é utilizado uma vez na validação e $(K - 1)$ vezes no treinamento da rede;
- 4) Por fim, os erros de validação de cada uma das K soluções obtidas são comparados e a solução de menor erro é adotada como conjunto de pesos sinápticos para a rede neural.

O objetivo da validação cruzada é evitar o sobreajuste da rede neural, cenário no qual o ruído sobre os dados também é aprendido pela rede. Deste modo, busca-se ampliar as chances de obter uma estrutura com boa capacidade de generalização.

No caso da ELM, é justamente na etapa de validação que ocorre a determinação do parâmetro C apresentado em (4): para cada valor de C no intervalo $\{2^{-24}, 2^{-23}, \dots, 2^{24}, 2^{25}\}$, determina-se \mathbf{w}_2 ; o vetor de pesos \mathbf{w}_2 associado ao valor de C que conduz ao menor erro frente ao conjunto de validação é adotado para \mathbf{w}_2^* , conforme [13]. Como não é possível garantir que um valor para o parâmetro C fora do intervalo em questão produza resultados significativamente melhores, o uso de tal abordagem pode comprometer a garantia da obtenção da solução ótima global. O mesmo procedimento de validação por K -pastas é usado no treinamento das redes MLPs, sendo que a solução final adotada é o conjunto de pesos dentre as K execuções que atinge o menor erro de validação.

Todos os experimentos realizados neste trabalho foram feitos em ambiente MATLAB, executando em um Intel Core i7-2600, 3.40GHz, com 8GB de memória RAM. Os resultados obtidos são apresentados na próxima seção.

IV. RESULTADOS

Esta seção explora o problema de identificação de sistemas não-lineares em um cenário real, o qual envolve o acionamento de uma válvula pneumática. A relação entre o sinal de controle pneumático, $x[n]$, e o fluxo através da válvula no instante n , $f[n]$, pode ser modelada por um sistema de Wiener, com as seguintes componentes [14]:

$$s[n] = 1.4138s[n-1] - 0.6065s[n-2] + 0.1044x[n-1] + 0.0833x[n-2], \quad (9)$$

$$f(s[n]) = \frac{0.3163s[n]}{\sqrt{0.1 + 0.9s^2[n]}}, \quad (10)$$

onde $s[n]$ denota a posição da válvula no instante n .

O cenário em questão é bastante desafiador, uma vez que a relação não-linear do sistema é de difícil aproximação paramétrica, além do fato de que este tipo de válvula apresenta características fortemente dependentes das condições de operação, motivando, portanto, uma abordagem adaptativa do problema [15].

Para determinar o número de neurônios na camada intermediária das redes utilizadas, foi monitorada a evolução da energia do erro² frente a um conjunto de teste³ da solução adotada em função do número de neurônios, como é mostrado na Figura 4.

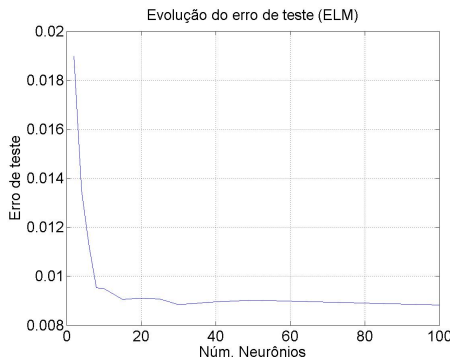


Fig. 4. Evolução da energia do erro de teste em função do número de neurônios.

É possível observar na Figura 4 que o erro cometido pela rede tem uma redução muito pequena após um número superior a 20 neurônios, sendo, portanto, este o valor adotado para o número de neurônios da ELM. Um procedimento análogo foi realizado para as outras duas redes, obtendo 20 neurônios para a MLP ajustada via SCGM. Para a MLP com parâmetros ajustados via evolução diferencial modificada, foram utilizados 5 neurônios em cada uma das camadas.

Para a validação, foram empregadas $K = 5$ pastas e um total de $T = 2000$ amostras para o sinal de entrada e ruído de medida. Em todos os modelos neurais estudados, foram considerados $M = 3$ atrasos para o sinal de entrada⁴ e adotada a função tangente hiperbólica como função de ativação para os neurônios.

Com respeito à ELM, os pesos sinápticos da camada intermediária foram aleatoriamente gerados segundo uma distribuição uniforme no intervalo $[-1, 1]$, enquanto que os *biases* também seguiram uma distribuição uniforme no intervalo $[0, 1]$.

A Tabela I apresenta os parâmetros utilizados no algoritmo de evolução diferencial modificado:

²A energia do erro corresponde à média da diferença ao quadrado da saída produzida pelo sistema, $d[n]$, e a saída da rede, $y[n]$, $\mathbb{E}[(d[n] - y[n])^2]$, com o operador de esperança sendo estimado via média amostral.

³O conjunto de teste é formado por 2000 amostras da resposta do sistema desconhecido frente a uma realização de um ruído branco diferente da usada na etapa de treinamento.

⁴O número de atrasos foi escolhido analisando-se o desempenho da rede neural em função de tal parâmetro. Valores superiores a 3 atrasos pouco aumentavam o desempenho da rede.

TABELA I

PARÂMETROS DO ALGORITMO DE EVOLUÇÃO DIFERENCIAL MODIFICADO.

$P = 20$ indivíduos.
$R = 30\%$.
$F = 0.01$.
$CR = 0.7$.
Máximo de gerações: 300.

A função de *fitness* adotada é mostrada a seguir:

$$fitness = \frac{1}{\mathbb{E}[e_t^2[n]] + 1}, \quad (11)$$

onde $e_t[n]$ denota o erro frente ao conjunto de treinamento para a n -ésima amostra e o operador de esperança, $\mathbb{E}[\cdot]$, é substituído pela média amostral.

Para o sinal de entrada, $x[n]$, foi utilizado um ruído branco, de média nula e variância unitária e para o ruído de medida, $\eta[n]$, um ruído branco de média nula e $\sigma^2 = 0.001$.

A Tabela II, mostrada a seguir, realiza uma comparação das três abordagens propostas. Para cada uma das redes, foram realizadas 20 execuções com condições iniciais distintas e, para cada uma dessas execuções, o desempenho das três redes foi comparado com a resposta esperada para as entradas ruído branco (com uma realização diferente da etapa de treinamento), senóide ($u[n] = \sin[10n]$) e degrau unitário. Para esta comparação, o critério adotado foi, também, a energia do erro.

TABELA II

ENERGIA DO ERRO PARA DIFERENTES ENTRADAS DE CADA UMA DAS REDES ANALISADAS.

	Ruído Branco	Senóide	Degrau
ED Mod.	0.0092	0.0080	0.0083
SCGM	0.0090	0.0086	0.0082
ELM	0.0092	0.0077	0.0082

Da Tabela II, pode ser observado que o desempenho para a ELM regularizada é muito próximo, e, em alguns casos, superior, ao desempenho das redes ajustadas via SCGM ou evolução diferencial modificada. Contudo, o processo de treinamento de uma ELM requer um custo computacional muito inferior ao das demais abordagens, além de não estar sujeito a mínimos locais. Isto pode ser verificado quando comparamos o tempo (médio) de treinamento e validação associado a cada rede neural considerada, conforme mostrado na Tabela III.

TABELA III

TEMPO MÉDIO DE TREINAMENTO E VALIDAÇÃO (20 EXECUÇÕES)

Rede	Média	Desvio Padrão
ED Mod.	30.6704 min	0.4546 min
SCGM	23.7494 s	5.8315 s
ELM	0.1275 s	0.0133 s

As figuras 5, 6 e 7 apresentam o comparativo da resposta produzida pela ELM (curvas em vermelho) e a resposta produzida pelo sistema desconhecido (curvas em azul), para as três entradas da Tabela II.

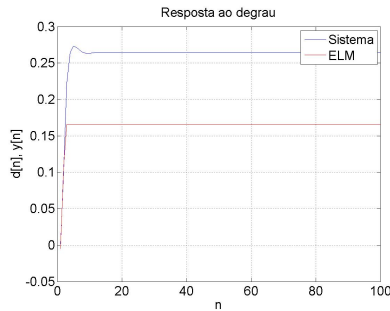


Fig. 5. Comparativo entre a resposta ao degrau obtida pela ELM (curva vermelha) e a produzida pelo sistema desconhecido (curva azul) em função do tempo.

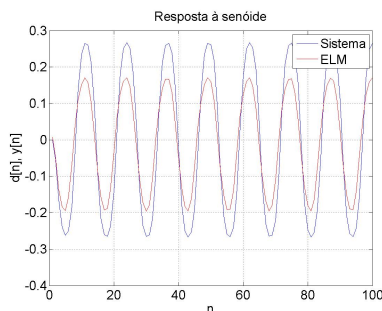


Fig. 6. Comparativo entre a resposta à senóide $\text{sen}[10n]$ obtida pela ELM (curva vermelha) e a produzida pelo sistema desconhecido (curva azul) em função do tempo.

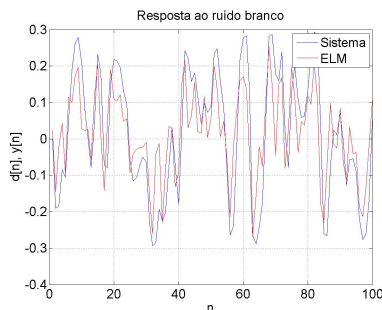


Fig. 7. Comparativo entre a resposta a um ruído branco obtida pela ELM (curva vermelha) e a produzida pelo sistema desconhecido (curva azul) em função do tempo.

É possível observar nas Figuras 5 a 7 que a ELM é capaz de aproximar relativamente bem a saída do sistema mesmo para sinais de entrada distintos daquele utilizado na etapa de treinamento.

As máquinas de aprendizado extremo, portanto, apresentam um bom compromisso entre versatilidade e capacidade de obtenção da solução ótima global, o que, unido à sua simplicidade de treinamento e capacidade para realizar mapeamentos não-lineares, faz com que tal estrutura desponte como uma solução adequada para o problema de identificação de sistemas não-lineares.

V. CONCLUSÃO

Este trabalho abordou o problema de identificação de sistemas não-lineares, tendo como proposta o uso de ELMs como

modelo de aproximação do sistema desconhecido. As ELMs apresentam um bom compromisso entre a generalidade do modelo adotado e a capacidade do mesmo em obter a solução ótima global. Aliado a esses dois fatos, tem-se a simplicidade do processo de ajuste dos parâmetros desta estrutura.

Os resultados obtidos na identificação de uma válvula pneumática mostram que a ELM consegue aproximar de maneira adequada o mapeamento intrínseco ao sistema, com qualidade semelhante àquela obtida por redes MLPs, e com um custo computacional bastante reduzido.

Tais observações, portanto, reforçam a visão de que as ELMs podem ser opções interessantes para o problema de identificação de sistemas não-lineares, inclusive dentro de uma abordagem adaptativa (*online*). Ademais, é possível aplicá-las na identificação de outros tipos de sistemas não-lineares, como os sistemas de Hammerstein [16] e de Wiener-Hammerstein, sendo esta uma perspectiva para a continuidade deste trabalho. Outro aspecto interessante a ser analisado mais detalhadamente é o desempenho das ELMs frente a diferentes níveis de ruído presente nas observações.

REFERÊNCIAS

- [1] L. Zadeh, *From Circuit Theory to System Theory*, Proceedings of the IRE, 1962.
- [2] L. A. Aguirre, *Introdução à Identificação de Sistemas: Técnicas Lineares e Não Lineares aplicadas a Sistemas Reais*, Editora UFMG, 1ª ed., 2000.
- [3] S. S. Haykin, *Adaptive Filter Theory*, Pearson, 5th ed, 2013.
- [4] L. Ljung, *Approaches to Identification of Nonlinear Systems*, Proceedings of the 29th Chinese Control Conference, 1-5, 29-31 July 2010.
- [5] E.-W. Bai, *An Optimal Two-Stage Identification Algorithm for Hammerstein-Wiener Nonlinear Systems*, Automatica, Vol. 34, No. 3, pp. 333-338, 1998.
- [6] E.-W. Bai, Z. Cai, S. Dudley-Javorosk, R. K. Shields, *Identification of a modified Wiener-Hammerstein system and its application in elletrically stimulated paralyzed skeletal muscle modeling*, Automatica 45 (2009) 736-743.
- [7] S. S. Haykin, *Neural Networks and Learning Machines*, Pearson Prentice Hall, 3rd ed., 2009.
- [8] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, *Extreme learning machine: theory and applications*. Neurocomputing, 70, 489-501, 2006.
- [9] G.-B. Huang., D.H. Wang, Y. Lan, *Extreme Learning Machines: a survey*, Int. J. Mach. Learn. & Cyber. (2011). doi:10.1007/s13042-011-0019-y
- [10] K. Price, R. M. Storn, J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Natural Computing Series, Springer, 2005.
- [11] G.-B. Huang, L. Chen, C.-K.Siew, *Universal approximation using incremental constructive feedforward networks with random hidden nodes*, IEEE Transactions on Neural Networks, 17(4), 879-892, 2006.
- [12] J. M. Martínez-Martínez, P. Escandell-Montero, E. Soria-Olivas, J. D. Martín-Guerrero, R. Magdalena-Benedito, J. Gómez-Sanchis, *Regularized extreme learning machine for regression problems*, Neurocomputing, Volume 74, Issue 17, p. 3716-3721, October 2011.
- [13] A. C. P. Kulaiif, F. J. Von Zuben, *Improved regularization in extreme learning machines*, XI Congresso Brasileiro de Inteligência Computacional (CBIC) 2013.
- [14] A. Janczak, *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models: A Block-Oriented Approach*, Lecture Notes in Control and Information Sciences, Vol. 310, Chapter 2: Neural Network Wiener Models, Springer-Verlag Berlin Heidelberg, 2005.
- [15] T. Wigren, *Recursive Prediction Error Identification Using the Nonlinear Wiener Model*, Automatica, Vol. 29, No. 4, pp. 1011-1025, 1993.
- [16] Y. Tang, Z. Li, X. Guan, *Identification of nonlinear system using extreme learning machine based Hammerstein model*, Communications in Nonlinear Science and Numerical Simulation, Vol. 19, pp. 3171-3183, 2014.