

REDES NEURAI E GMDH APLICADOS NA MONITORAÇÃO DE SENSORES DO REATOR IEA-R1

Elaine Inácio Bueno¹, Iraci Martinez Pereira², Antonio Teixeira e Silva²

¹Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Campus Bragança Paulista
Avenida Francisco Samuel Lucchesi Filho, 770 – 12910-600 – Bragança Paulista/SP
ebueno@cefetsp.br

²Instituto de Pesquisas Energéticas e Nucleares (IPEN/CNEN-SP)
Av. Professor Lineu Prestes, 2242 – 05508-000 – São Paulo/SP
martinez@ipen.br, teixeira@ipen.br

Resumo –Este trabalho tem como objetivo apresentar um estudo comparativo entre as metodologias de Redes Neurais Artificiais (RNAs) e GMDH(Group Method of Data Handling) aplicados na Monitoração de sensores de temperatura de um reator experimental. Foram desenvolvidos dois modelos de Monitoração: um utilizando Redes Neurais e o outro o algoritmo GMDH. Para tanto foi utilizada uma base de dados gerada por um modelo teórico do reator. Os resultados obtidos utilizando o algoritmo GMDH foram melhores do que os obtidos com as Redes Neurais.

Palavras-chave – Sistema de Monitoração, Redes Neurais, GMDH, Reator Experimental

1. Introdução

Os estudos na área de Monitoração de sensores têm sido estimulados em virtude da demanda crescente em qualidade, confiabilidade e segurança nos processos de produção. Nos processos industriais, a interrupção da produção por alguma anomalia pode colocar em risco a segurança do operador, além de provocar perdas econômicas, aumentando assim os custos com a reparação de algum equipamento danificado. Esses dois fatores estimulam o desenvolvimento de sistemas de controle confiáveis e tolerantes a falhas [4],[14].

Nas centrais nucleares existe um grande número de variáveis a serem observadas continuamente. Durante uma falha, os operadores recebem um grande volume de informações através da leitura dos instrumentos, indicando um padrão típico da mesma. Devido à grande quantidade de informação em um período de tempo muito pequeno, os operadores são obrigados a tomar algumas decisões em condições estressantes, dificultando em alguns casos a identificação da falha. Com o propósito de auxiliar os operadores das centrais nucleares têm-se utilizado muitas técnicas de Inteligência Artificial na Monitoração, incluindo a Lógica Nebulosa [8], Redes Neurais Artificiais (RNAs) [15],[16], GMDH (Group Method of Data Handling) [14] e Algoritmos Genéticos (AGs), [15]. A utilização desses métodos é justificada por permitir a modelagem do processo sem a utilização de equações algébricas que descrevem matematicamente o fenômeno [12]. Esta modelagem é realizada utilizando-se uma base de dados contendo o histórico temporal de operação da planta.

O uso de RNAs na Monitoração de sensores é muito interessante, tanto pela forma como estes problemas são representados internamente pela rede, como também pelos resultados gerados. Em RNAs, o procedimento usual na solução de problemas passa inicialmente por uma fase de *aprendizagem*, em que um conjunto de exemplos é apresentado para a rede, que extrai automaticamente dos mesmos as características necessárias para representar a informação fornecida. Essas características são utilizadas posteriormente para gerar respostas a problemas com características similares às dos exemplos. Outro método que pode ser utilizado na Monitoração de sensores é o GMDH (Group Method of Data Handling). O GMDH é um algoritmo auto-organizável de propagação indutiva que permite a obtenção de um modelo matemático do sistema a partir de observações de amostras de dados [5],[11]. Além disso, o GMDH

apresenta algumas vantagens, como: possui uma arquitetura que evolui durante o processo de modelagem; uma auto-organização indutiva, ou seja, o número de camadas e nós é estimado pelo mínimo de um critério externo (escolha objetiva); a convergência é obtida através de um modelo de complexidade ótima; otimizada simultaneamente a estrutura e dependências no modelo, não sendo uma técnica consumidora de tempo; e as variáveis de entrada não apropriadas para o modelo são excluídas automaticamente [6].

O presente trabalho tem como objetivo apresentar um estudo comparativo entre as metodologias GMDH e Redes Neurais na Monitoração de sensores. As duas técnicas supracitadas já foram utilizadas no desenvolvimento de sistemas de Monitoração e Diagnóstico de Falhas em sensores [2], [6], onde os resultados obtidos demonstraram a viabilidade de aplicação desses métodos para tal tarefa. Este trabalho possibilitará identificar qual o melhor método para a aplicação proposta, além de estimular o desenvolvimento de trabalhos futuros, usando a metodologia GMDH e Redes Neurais de modo combinado.

2. DESCRIÇÃO DO REATOR IEA-R1

O Reator IEA-R1 está situado no IPEN (Instituto de Pesquisas Energéticas e Nucleares) localizado na Cidade Universitária em São Paulo. Foi projetado e construído pela “Babcock & Wilcox”, em 1956. É um reator de pesquisa do tipo piscina, refrigerado e moderado a água leve, que utiliza berílio como refletor. A Figura 1 mostra um diagrama esquemático do reator IEA-R1 [10].

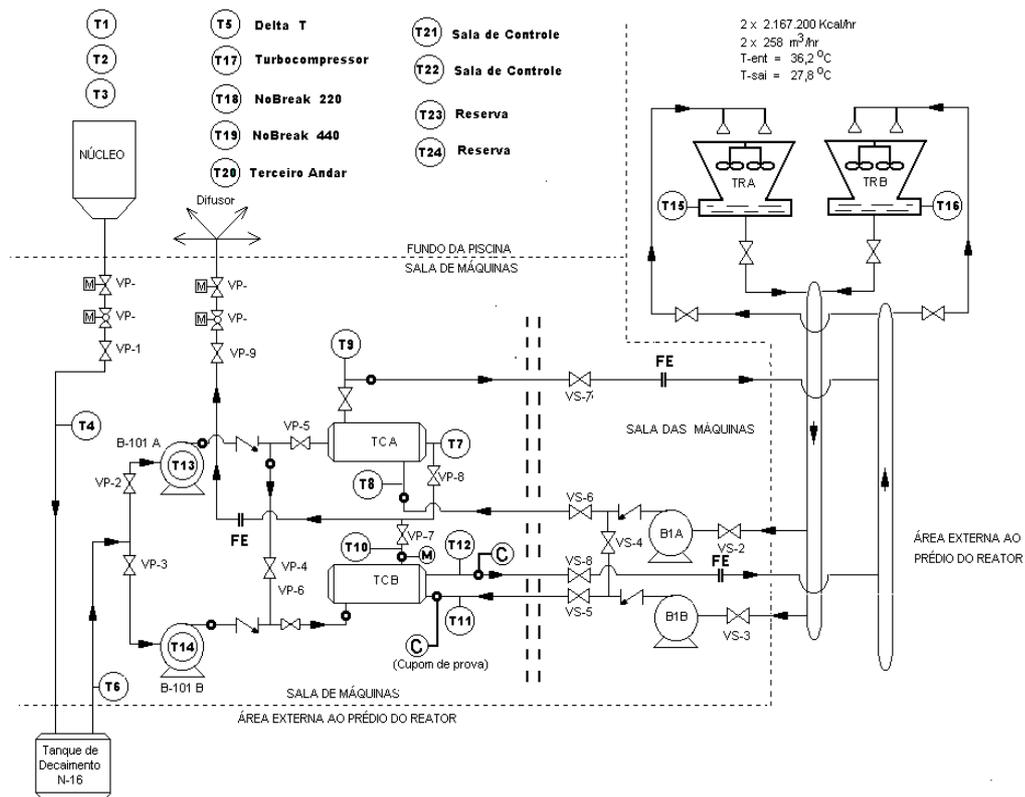


Figura 1. Diagrama esquemático do reator IEA-R1

3. MODELO TEÓRICO DO REATOR IEA-R1

O modelo teórico do reator [7] foi desenvolvido com o uso da ferramenta GUIDE do Matlab [3], que permite gerar dados em diversas condições de operação, possibilitando simular situações que não são possíveis de se obter experimentalmente devido às restrições de operação do reator nuclear. As equações que regem o comportamento das variáveis de processo foram baseadas no balanço de massa e energia do reator IEA-R1, levando-se em consideração todos os aspectos físicos e operacionais, tais como, comprimento e diâmetro das tubulações, relações entre os valores de vazão, temperaturas e perda de

carga. No modelo teórico, foram consideradas as seguintes variáveis: T1 (Temperatura na superfície da piscina), T2 (Temperatura a meia altura da piscina), T3 (Temperatura sobre o núcleo do reator), T4 (Temperatura na entrada do tanque de decaimento), T6 (Temperatura na saída do tanque de decaimento), T7 (Temperatura na saída do primário), T8 (Temperatura na entrada do secundário), T9 (Temperatura na saída do secundário), F1M3 (Vazão do circuito primário), F2M3 (Vazão do circuito secundário) e Pot (Potência).

A interface do programa foi criada de forma a lembrar o fluxograma de processo do reator IEA-R1, como pode ser visto na Figura 2. Nessa interface é possível visualizar os valores da potência térmica (em MW), calculados no núcleo e no lado primário do trocador de calor. Neste modelo é possível: adicionar falhas às variáveis; gerar dados na condição normal de operação do reator; atribuir níveis de ruídos nas variáveis de entrada através de uma janela, sendo o seu valor determinado em %; e determinar o número total de pontos que serão gerados para as condições estabelecidas.

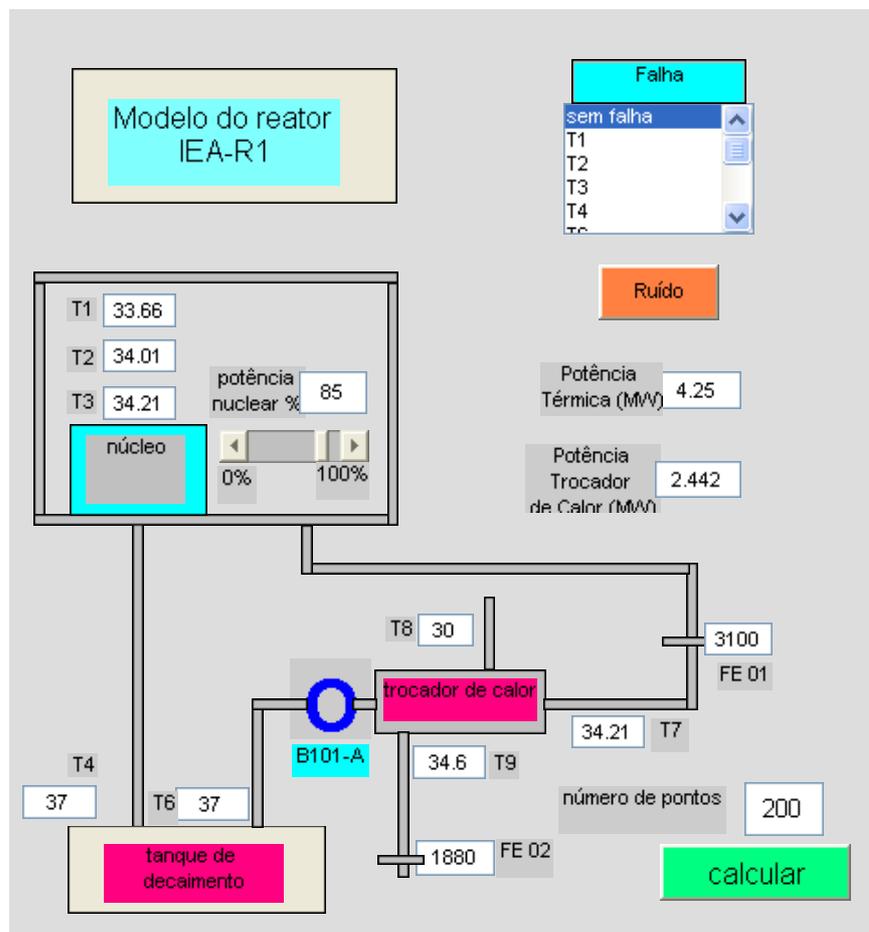


Figura 2. Interface do programa desenvolvido

4. REDES NEURAIS ARTIFICIAIS

As Redes Neurais Artificiais podem ser definidas como sistemas paralelos compostos por unidades de processamento simples, dispostas em camadas e altamente interligadas, inspiradas no cérebro humano [9]. O aprendizado de uma RNA é realizado a partir de um algoritmo, conhecido como algoritmo de aprendizagem, onde os pesos sinápticos são atualizados com o propósito de alcançar o objetivo desejado. As formas em que os neurônios das RNAs são organizados estão intimamente associadas ao algoritmo de aprendizado utilizado no treinamento das redes. Geralmente, podemos identificar três tipos de arquiteturas de RNAs [9]: redes com propagação para frente de uma camada: compostas por uma camada de entrada e outra de saída; redes com propagação para frente de múltiplas camadas: compostas

por uma camada de entrada, camadas ocultas e uma camada de saída; e redes recorrentes: contêm realimentação das saídas para as entradas.

O principal objetivo do treinamento de uma RNA é fazer com que a aplicação de um conjunto de entradas produza um conjunto de saídas desejadas ou no mínimo um conjunto de saídas consistentes. Cada conjunto de entrada ou saída é denominado *vetor*. O treinamento é realizado pela aplicação seqüencial dos vetores de entrada (e em alguns casos também os de saída), enquanto os pesos da rede são ajustados de acordo com um procedimento de treinamento pré-determinado. Durante o treinamento, os pesos da rede gradualmente convergem para determinados valores, de tal maneira que a aplicação dos vetores de entrada produza as saídas necessárias. Os procedimentos de treinamento das RNAs podem ser classificados em duas classes: supervisionado e não supervisionado.

O treinamento *supervisionado* necessita de um vetor de entrada e um vetor de saída, conhecido como vetor alvo. Esses dois vetores são então utilizados para o treinamento da RNA. O procedimento de treinamento funciona da seguinte maneira: o vetor de entrada é aplicado, a saída da rede é calculada e comparada com o correspondente vetor alvo. O erro encontrado é então realimentado através da rede e os pesos são atualizados de acordo com um algoritmo com o propósito de minimizar este erro. Este processo de treinamento é repetido até que o erro para os vetores de treinamento alcance valores pré-determinados. O treinamento *não-supervisionado*, por sua vez, não requer vetor alvo para as saídas. O conjunto de treinamento modifica os pesos da rede, de forma a produzir saídas que sejam consistentes. O processo de treinamento extrai as propriedades estatísticas do conjunto de treinamento e agrupa os vetores similares em classes, onde a aplicação de um vetor de uma determinada classe à entrada da rede produzirá um vetor de saída específico. Neste trabalho, a arquitetura escolhida foi a MLP (Redes de Camadas Múltiplas), treinada pelo algoritmo de retropropagação (Backpropagation).

5. GMDH

O método GMDH (Group Method of Data Handling) é composto por um algoritmo proposto por Ivakhnenko. Consiste em um método algébrico para estimar estados de um sistema, saídas de controladores e funções de atuadores [11]. A metodologia pode ser considerada como um algoritmo auto-organizável de propagação indutiva para a solução de problemas práticos complexos. Além disso, é possível obter um modelo matemático do processo a partir de observações de amostras de dados, que será utilizado na identificação e no reconhecimento de padrões, ou até mesmo para descrever o próprio processo. Este método soluciona o problema multidimensional de otimização do modelo, pelo procedimento de escolha e seleção, a partir de um conjunto de modelos candidatos de acordo com um critério fornecido. A maioria dos algoritmos GMDH utiliza funções de referência polinomiais. Uma conexão genérica entre variáveis de entrada e de saída pode ser expressa pela série funcional de Volterra, discreta análoga do polinômio de Kolmogorov-Gabor:

$$y = a + \sum_{i=1}^m b_i x_i + \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_i x_j + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m d_{ijk} x_i x_j x_k + \dots \quad (1)$$

Onde:

$\{x_1, x_2, x_3 \dots\}$: variáveis de entrada

$\{a, b, c \dots\}$: coeficientes do polinômio.

As componentes da matriz de entrada podem ser variáveis independentes, formas funcionais ou termos de diferenças finitas, além disso, podem-se utilizar outras funções de referência não lineares. O método permite, ainda, encontrar simultaneamente a estrutura do modelo e a dependência da saída do sistema modelado em função dos valores de entrada mais significativos do sistema.

Aksenova e Yurachkovsky (1988) [1] obtiveram as bases teóricas da efetividade do GMDH como método adequado e robusto de construção de modelos para estimativa. A modelagem auto-organizável é baseada em redes de aprendizado estatístico, que capturam relações complexas não lineares em uma forma de execução rápida e compacta. Essas redes subdividem o problema em partes menores de fácil manuseio e aplicam técnicas de regressão avançadas para resolver cada um desses problemas.

5.1. Descrição da metodologia GMDH

Em geral, os dados disponíveis são as variáveis de entrada e de saída de um sistema, representadas pela matriz de variáveis de entrada $X = (x_{ij})$, e pelo vetor da variável de saída $Y = (y_i)$, onde: $i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$; n é o número total de observações e m é o número total de variáveis.

O algoritmo GMDH é executado de acordo com os seguintes passos:

1º passo: Consiste em construir um novo conjunto de variáveis Z . Para tanto, inicialmente deve-se dividir as observações em dois subconjuntos distintos. O primeiro subconjunto será utilizado para ajuste ($i=1$ a nt) e o segundo para verificação ($i=nt+1$ até n).

A seguir tomam-se as variáveis independentes do subconjunto de ajuste duas de cada vez para todas as combinações possíveis e calcula-se os coeficientes da seguinte regressão polinomial, onde p e q são as colunas da matriz X :

$$y_i = a_{pq} + b_{pq} x_{ip} + c_{pq} x_{iq} + d_{pq} x_{ip}^2 + e_{pq} x_{iq}^2 + f_{pq} x_{ip} x_{iq} \quad \begin{cases} p = 1, 2 \dots m, p \neq q, \\ q = 1, 2 \dots m, q > p, \\ i = 1, 2 \dots nt \end{cases} \quad (2)$$

Para cada combinação das variáveis independentes (colunas de X) há um conjunto de coeficientes da regressão, resultando na matriz de coeficientes C . Como se tem $m(m-1)/2$ diferentes combinações das variáveis de entrada, resultam $m(m-1)/2$ linhas da matriz de coeficientes C .

$$C = (a_{pq}, b_{pq}, c_{pq}, d_{pq}, e_{pq}, f_{pq}) \quad \begin{cases} p = 1, 2 \dots m, p \neq q, \\ q = 1, 2 \dots m, p > q \end{cases} \quad (3)$$

Para cada conjunto de coeficientes C_{pq} , avalia-se o polinômio para todos os n pontos de dados para calcular uma nova estimativa de y , chamada z_{pq} :

$$z_{i,pq} = a_{pq} + b_{pq} x_{ip} + c_{pq} x_{iq} + d_{pq} x_{ip}^2 + e_{pq} x_{iq}^2 + f_{pq} x_{ip} x_{iq} \quad (4)$$

Esses novos valores são armazenados na primeira coluna de uma matriz Z .

Repete-se esta avaliação para todos os conjuntos de coeficientes e para todos os n dados, gerando uma nova matriz Z chamada nova geração de variáveis.

$$Z = (z_{ij}) \quad \begin{cases} i = 1, 2 \dots n, \\ j = 1, 2 \dots m(m-1)/2 \end{cases} \quad (4)$$

onde cada j corresponde a um par p, q

$$Z = \begin{bmatrix} z_{11} & z_{12} & \dots & \dots & z_{1,m(m-1)/2} \\ z_{21} & z_{22} & \dots & \dots & \dots \\ \dots & \dots & \dots & z_{ij} & \dots \\ \dots & \dots & \dots & \dots & \dots \\ z_{n1} & z_{n2} & \dots & \dots & z_{n,m(m-1)/2} \end{bmatrix} \quad (5)$$

Pode-se interpretar estas variáveis (matriz Z) como novas variáveis que têm melhor poder de estimativa do que aquelas da geração original (matriz X).

2º passo. Compara-se cada coluna de Z com a variável dependente Y . Isto é, para cada coluna j de Z calcula-se o desvio médio quadrático r_j para i variando de $nt+1$ até n , ou seja, utilizando o subconjunto de verificação:

$$r_j^2 = \frac{\sum_{i=nt+1}^n (y_i - z_{ij})^2}{\sum_{i=nt+1}^n y_i^2} \quad j=1, 2, \dots, m(m-1)/2 \quad (6)$$

Ordena-se as colunas de Z em ordem crescente de r_j e então seleciona-se aquelas colunas de Z que satisfaçam um critério selecionado anteriormente (por exemplo $r_j < R$, onde R é um valor residual mínimo). Pode-se notar que o número de variáveis salvas, digamos m_1 , pode ser igual, menor ou maior do que o número original m .

3º passo. Esta nova matriz Z irá substituir a matriz original X . Estas novas variáveis são combinadas exatamente como se fez anteriormente. Ou seja, calculam-se todas as equações de regressão quadráticas de y em função destas novas variáveis (duas a duas). Isso irá fornecer um novo conjunto de $m_1(m_1-1)/2$ equações de regressão para estimar y a partir das novas variáveis. Agora são selecionadas as melhores das novas estimativas, gerando novas variáveis independentes a partir das equações selecionadas para substituir as anteriores, e combinar todos os pares destas novas variáveis.

4º passo. Do passo anterior encontra-se o menor r_j e chama-se de RMIN. Se o valor de RMIN é menor que o valor de RMIN da geração prévia, repete-se os passos 1, 2 e 3. Se o valor de RMIN é maior que o valor precedente, assume-se que a "curva RMIN" atingiu seu mínimo, o processo é interrompido e usam-se os resultados da geração anterior.

A primeira coluna da matriz Z conterá os valores de y_i do polinômio de Ivakhnenko avaliados para os n pontos de dados originais. Em outras palavras, a primeira coluna de Z é a estimativa da variável dependente. Para encontrar os coeficientes a , b_i , c_{ij} , d_{ijk}, \dots no polinômio de Ivakhnenko, deve-se salvar todos os conjuntos de coeficientes de todas as gerações que foram computados a cada iteração e sistematicamente avaliar esta árvore de quadráticos até chegar em um polinômio de alta ordem nas variáveis originais x_1, x_2, \dots, x_m . De um ponto de vista computacional, desde que se tenham todos os polinômios de regressão quadráticos armazenados no computador, é possível computar a estimativa da saída y a partir destes quadráticos.

De forma resumida, os valores estimados são propagados sucessivamente para gerações superiores do algoritmo, melhorando a cada estágio sucessivo. A cada estágio de aproximação z_{ipq} é formado a partir de pares dos sinais de entrada (desta geração), e novos valores da variável estimada são propagados para a próxima geração. A iteração continua até que o desvio médio quadrático entre os valores estimados e medidos para determinada variável de saída seja menor do que um valor desejado, ou quando um número máximo de gerações previamente definido seja atingido.

O que se tem é uma estimativa de y como uma forma quadrática de duas variáveis, as quais são por sua vez uma combinação quadrática de duas mais variáveis, as quais são também combinações de duas outras variáveis, das variáveis originais. Em outras palavras, se forem feitas as substituições algébricas necessárias, o polinômio resultante teria a forma mostrada na equação a seguir, conhecido como o polinômio de Ivakhnenko [11].

$$y = a + \sum_{i=1}^m b_i x_i + \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_i x_j + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m d_{ijk} x_i x_j x_k + \dots (7)$$

6. RESULTADOS OBTIDOS

6.1. Monitoração usando RNAs

Para estabelecer o modelo de monitoração usando Redes Neurais Artificiais foram utilizadas as variáveis do modelo teórico: T1, T2, T3, T4, T6, T7, T8, T9, F1M3, F2M3 e Pot, descritas anteriormente; porém serão apresentados apenas os resultados obtidos na monitoração dos sensores de temperatura. Todas as simulações computacionais foram realizadas utilizando a ferramenta de Redes Neurais do Matlab, versão R2008a.

No treinamento das RNAs foi utilizado o algoritmo de retropropagação, com a seguinte arquitetura: uma camada de entrada, uma camada oculta e uma camada de saída. Na camada de entrada foi utilizada a função linear, na camada oculta, a tangente hiperbólica e na camada de saída, a função linear. Para escolher o melhor modelo, variou-se o número de neurônios na camada escondida e foram calculados os resíduos em cada caso. A escolha do melhor modelo foi feita através do menor resíduo obtido, calculado através da equação (2):

$$res = \left(\left| \frac{y - y_d}{y} \right| \right) \times 100 \quad (2)$$

Onde:

y = saída atual

y_d = saída desejada

Para cada variável de temperatura monitorada foram testadas dez arquiteturas de RNAs, variando-se o número de neurônios na camada escondida de 1 a 10. A Figura 3 mostra o resultado da monitoração de T3 (variável de saída), onde as variáveis de entrada foram: T1, T2, T4, T6, T7, T8, T9, F1M3 e F2M3. Para as demais variáveis de temperatura o procedimento foi semelhante.

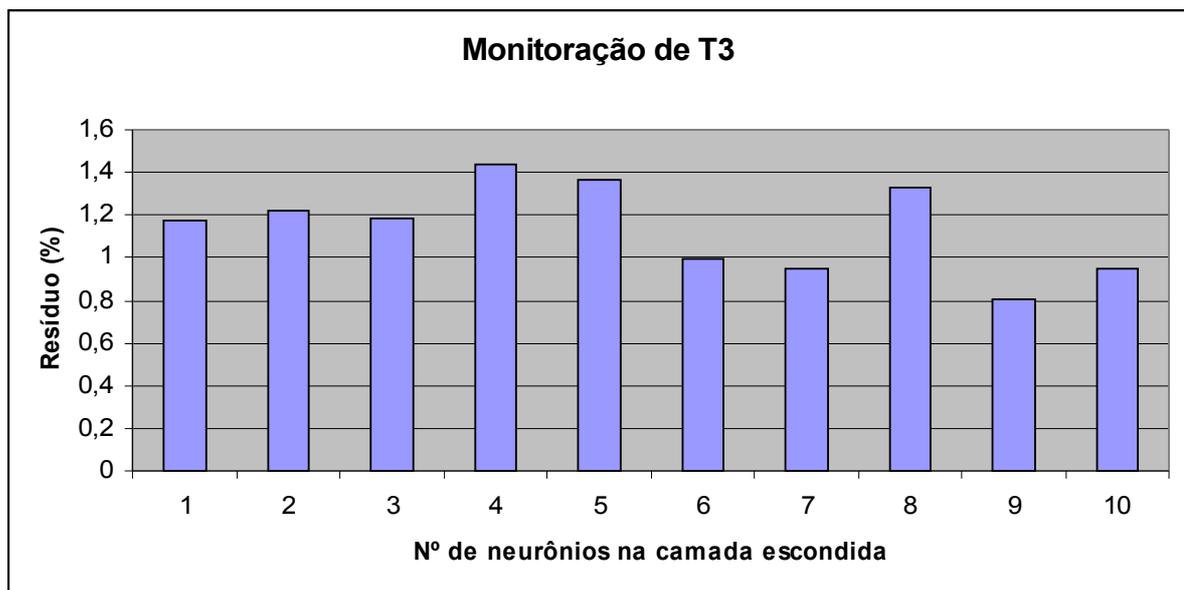


Figura 3. Monitoração de T3 usando Redes Neurais

6.2. Monitoração usando GMDH

O GMDH foi usado para encontrar o melhor modelo de monitoração para os sensores de temperatura. Por ser um algoritmo auto-organizável, a saída do sistema é encontrada em função das entradas mais significativas para o mesmo, não havendo necessidade de utilização do conhecimento especialista para realizar esta tarefa. Na Figura 4 são apresentados os resultados obtidos na monitoração de cada um dos termopares. A escolha do melhor modelo é feita através da equação (2).

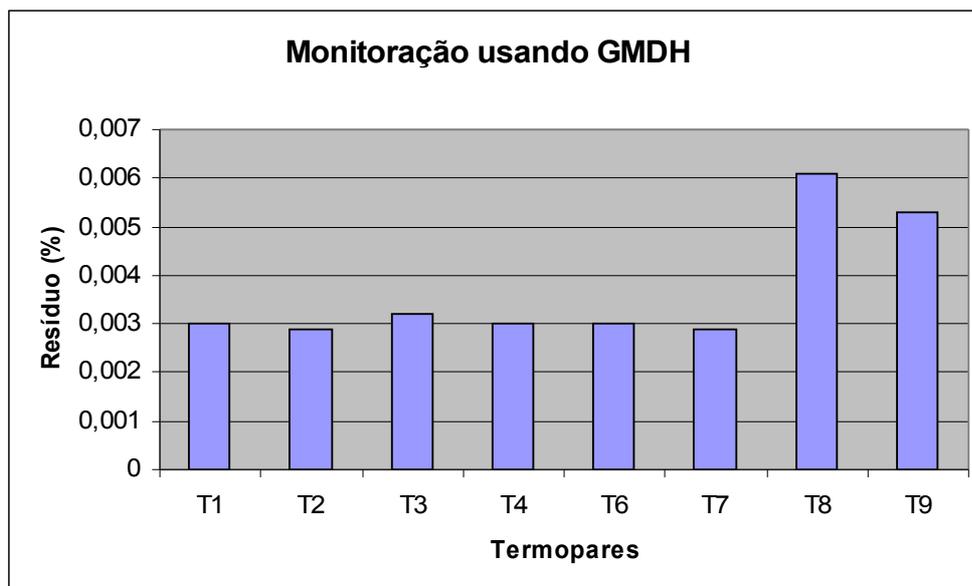


Figura 4. Monitoração de termopares usando GMDH

6.3. Análise dos resultados obtidos

Na Tabela 1 é apresentado um resumo dos resultados obtidos na monitoração das variáveis de temperatura usando GMDH e Redes Neurais.

Tabela 1. Resumo dos resultados obtidos na monitoração de termopares

	Resíduo (%)	
	RNAs	GMDH
T1	0,6059	0,0030
T2	1,1927	0,0029
T3	0,8081	0,0032
T4	0,6588	0,0030
T6	0,6521	0,0030
T7	0,6576	0,0029
T8	0,7162	0,0061
T9	0,7306	0,0053

Através da análise da Tabela 1, verifica-se que os resíduos obtidos na monitoração de termopares usando a metodologia GMDH foram menores (da ordem de 10^{-3}) do que os obtidos usando RNAs (da ordem de 10^{-1}).

7. CONCLUSÕES

Neste trabalho foi apresentado um estudo comparativo entre os métodos GMDH e Redes Neurais, aplicados na monitoração de sensores de um reator experimental. Para realizar tal estudo, utilizou-se uma base de dados gerada por um modelo teórico do reator de pesquisas IEA-R1.

Na monitoração usando RNAs, diversas redes foram treinadas com o propósito de estabelecer o melhor modelo para a monitoração de uma determinada variável. Pela arquitetura de rede e algoritmo de treinamento usado neste trabalho, é necessário conhecer as variáveis de maior relevância para a monitoração de um determinado sensor.

Já na monitoração usando GMDH, determinando-se a variável de saída, o modelo é obtido através da eliminação das variáveis de entrada irrelevantes.

Para a escolha do melhor modelo foram calculados os resíduos obtidos na monitoração usando cada uma das metodologias supracitadas.

Os resultados obtidos neste trabalho demonstram que os modelos obtidos usando GMDH apresentaram resíduos menores do que os obtidos usando RNAs. Esses resultados estimulam a continuidade nos estudos nesta área, o que possibilitará o desenvolvimento de sistemas de controle confiáveis e tolerantes a falhas.

8. REFERÊNCIAS BIBLIOGRÁFICAS

1. AKSENOVA, T. I.; YURACHKOVSKY, Yu. P. **A Characterisation at Unbiased Structure and Conditions of Their J-Optimality.** *Sov. J. of Automation and Information Sciences*, 21 (4): p. 36-42, 1988.
2. BUENO, E. I. **Utilização de Redes Neurais Artificiais na Monitoração e Detecção de Falhas em Sensores do reator IEA-R1.** Dissertação (Mestrado), Universidade de São Paulo – IPEN, 98 p., São Paulo, 2006.
3. **Creating Graphical User Interfaces**, Version 7.6, the MATH WORKS Inc, USA, 2008.
4. ECHENDU, J. E. A., ZHU, H. **Detecting changes in the condition of process instruments.** In Conf. Rec. of IMCT, p. 515-518, Irvine CA, USA, 1993.
5. FERREIRA, P. B. **Incipient Fault Detection and Isolation of Sensors and Field Devices.** Tese (Doutorado) - University of Tennessee, August 1999.
6. GONÇALVES, I. M. P. **Monitoração e diagnóstico para detecção de falhas de sensores utilizando a metodologia GMDH.** Tese (Doutorado), IPEN, São Paulo, 2006.
7. GONÇALVES, I. M. P., TING, D. K. S. **A theoretical model for the IPEN research reactor IEA-R1.** INAC 2005 – International Nuclear Atlantic Conference Proceedings (Cdroom), Agosto 2005.
8. GOODE, P. V. **Using a Neural/Fuzzy System to extract heuristic knowledge of incipient faults in induction motors: Part I – Methodology.** IEEE Transactions on Industrial Electronics, 42(2), 1995.
9. HAYKIN, S. **Neural Networks - A Comprehensive Foundation.** Prentice Hall, 1999.
10. INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES (IPEN). Relatório de Análise de Segurança do reator de pesquisas do IPEN IEA-R1 (RAS), 1999.
11. IVAKHNENKO, A. G. **Self-teaching Systems of Recognition and Automatic Control.** Moscou: *Tekhnika*, 392, 1969.
12. KALOGIROU, S. A. **Application of artificial neural networks for energy systems.** Applied Energy, 67 (1): p. 17-35, Setembro, 2000.
13. PATTON, R. J.; CHEN, J.; NIELSEN, S. B. **Model-Based Methods for Fault Diagnosis: Some Guidelines.** IEEE Transactions of the Institute of Measurement and Control, 17(2), 1995.
14. PUIG, V., et. al. **A GMDH neural network-based approach to passive robust fault detection using a constraint satisfaction backward test.** Engineering Applications of Artificial Intelligence, 20, p. 886-897, 2007.
15. ROVITHAKIS, G. A.; MANIADAKIS, M; ZERVAKIS, M. **A Hybrid Neural Network/Genetic Algorithm approach to optimizing feature extraction for signal validation.** IEEE Transactions on Systems, Man, And Cybernetics – Part B: CYBERNETICS, 34 (1), 2004.
16. SAMANTA, B. **Gear fault detection using Artificial Neural Networks and support vector machines with genetic algorithms.** Mechanical Systems and Signal Processing, 18 (3): p. 625-644, 2004.
17. SYDENHAM, P. H., THORN, R. **Strategies for sensor performance assessment.** In Conf. Rec. of IMCT, p. 353 – 358, Irvine CA, USA, 1993.