

Detecção de Falhas em Sistemas Dinâmicos: Abordagem Baseada em Redes Neurofuzzy com Aprendizado On-Line

Walmir Caminhas¹

Hermano Tavares²

Fernando Gomide³

¹ UFMG-DEE-CPDEE

Av. Pres. Antônio Carlos, 6627 - Campus Pampulha
31.270-901 Belo Horizonte MG – Brasil.

² Unicamp-FEEC-DENSIS, Campinas – SP, Brasil

³ Unicamp-FEEC-DCA, Campinas – SP, Brasil

E-mails: caminhas@eee.ufmg.br, tavares@densis.fee.unicamp.br, gomide@dca.fee.unicamp.br

Abstract

Neste trabalho propõe-se uma estratégia de detecção e diagnóstico de falhas em sistemas dinâmicos baseada em redes neurofuzzy com aprendizado on-line. A estratégia foi empregada para detectar e diagnosticar falhas em um sistema de acionamento elétrico empregando máquina de corrente contínua. Os resultados apresentados confirmam a eficiência da estratégia proposta e a credencia como uma abordagem promissora na solução de problemas complexos de DDF.

1. Introdução

O desenvolvimento de sistemas tolerantes a falhas é uma preocupação crescente, principalmente porque além de fornecer uma maior segurança aos operadores ou usuários das plantas, proporcionam economia no que diz respeito a paradas para manutenção, reduzindo custos e propiciando uma maior qualidade do produto final. Um assunto que vem ganhando destaque nas duas últimas décadas é o desenvolvimento de técnicas para detecção e diagnóstico de falhas em sistemas dinâmicos [1].

Tipicamente, a detecção de falhas em sistemas dinâmicos pode ser feita por redundância de *hardware* ou de *software* (ou redundância analítica). O enfoque deste trabalho é a segunda opção. Basicamente as estratégias de detecção de falhas fundamentadas no conceito de redundância analítica, podem ser agrupadas em duas classes: baseadas em modelos quantitativos do processo e em modelos qualitativos [1].

Na prática os métodos baseados em modelos qualitativos são, normalmente, combinados com os métodos baseados em modelos quantitativos [1]. Para a primeira classe, pode-se citar o emprego de: observadores de estados tipo Luenberger dedicados ou o filtro de Kalman [2], [3]; observadores robustos [4], [5]; estimação de parâmetros [6], [7]; técnicas de aprendizado (redes neurais e *neurofuzzy*) [8], [9] e

outras. As técnicas que empregam modelos qualitativos enfatizam o uso da lógica clássica [10] ou *fuzzy* [11].

Neste trabalho propõe-se uma estratégia baseada em treinamento *on-line* de uma rede neurofuzzy para identificação sistemas dinâmicos e um sistema de classificação de padrões.

2. Descrição da Estratégia Proposta

A figura 1 mostra o diagrama de blocos da abordagem empregada para detecção e diagnóstico de falhas em sistemas dinâmicos proposta neste trabalho. A idéia básica é tratar o problema de detecção de falhas como um problema de classificação de padrões. As entradas do sistema de classificação são os vetores: u (entradas da planta), y (saídas da planta) e y^* (saídas de referência) e r (resíduos).

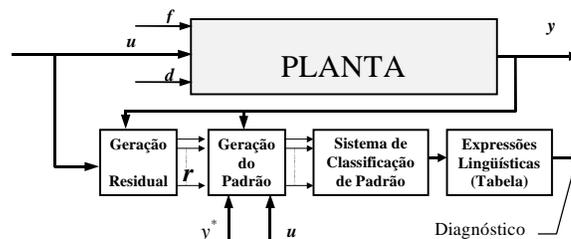


Figura 1: Estrutura do sistema de detecção e diagnóstico de falhas

Os resíduos são gerados a partir da estimação de parâmetros do modelo. É empregada a topologia de rede *neurofuzzy* proposta em [12] para identificação dos parâmetros do modelo da planta, figura 2. Esses parâmetros correspondem aos pesos da rede. É empregado o algoritmo de treinamento *on-line* da rede *neurofuzzy* proposto em [14]. Com esse algoritmo, os pesos são ajustados para obter erro nulo na estimação

das saídas da planta. Os pesos W_0 (vetor) são determinados através de treinamento *off-line* da rede. Nesta etapa, os dados fornecidos para treinamento correspondem ao funcionamento normal do sistema, isto é, operação sem a ocorrência de falhas. Já os pesos W são obtidos via treinamento *on-line*. Quando ocorrerem falhas no sistema, para que o erro da estimação da saída seja nulo, os pesos W são diferentes de W_0 , sendo essa diferença utilizada para gerar os resíduos. Os resíduos são as entradas do sistema de classificação.

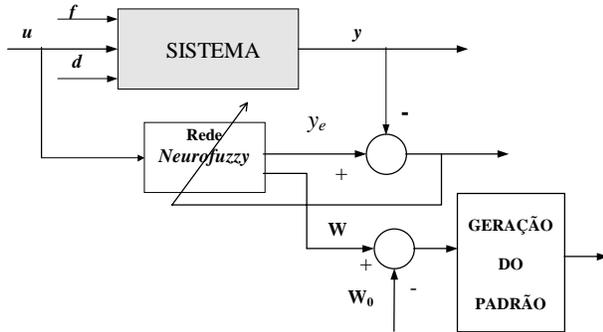


Figura 2: Geração de resíduos

O sistema de classificação de padrões empregado é baseado em um sistema *fuzzy*, implementado por uma rede *neurofuzzy and/or* [13]. Como as saídas do sistema de classificação de padrões são números, é utilizada uma tabela para as correspondentes conversões em expressões linguísticas. Estas expressões linguísticas são do tipo: se a saída do sistema de classificação é “i”, então a falha é no atuador “j”. Na figura 1 o vetor f é usado somente durante a fase de treinamento do sistema de classificação.

3. Descrição e Modelagem de Falhas de um Sistema de Acionamento Elétrico

3.1. Descrição do sistema

Utilizou-se, como exemplo típico de um sistema dinâmico, um acionamento elétrico com máquina de corrente contínua (C.C.). O sistema de acionamento C.C. é composto por duas fontes de alimentação, conversores estáticos controlados, uma máquina de corrente contínua e uma carga mecânica, figura 3. Foram consideradas falhas nos conversores, na máquina e nos sensores de corrente e velocidade.

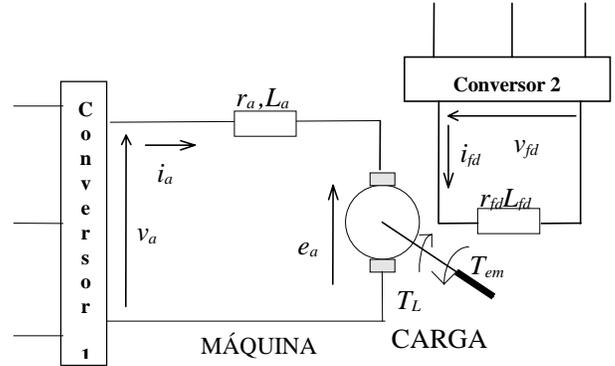


Figura 3: Exemplo de sistema de acionamento C.C.

3.2. Modelagem das falhas

Do ponto de vista de classificação de falhas, o sistema de acionamento C.C. pode ser dividido em três grandes grupos: atuadores (conversores da armadura e do campo); planta ou processo (máquina e carga mecânica) e instrumentação (sensores de corrente e velocidade). As falhas de interesse neste trabalho estão descritas na tabela 1. O modelo geral, equações (1) e (2) permite simular estas falhas, considerando que i_a , i_{fd} e ω_r são as variáveis medidas.

Tabela 1: Falhas no sistema de acionamento C.C.

Tipo de Falha	Parâmetros Indicadores da Falha	Definição dos Parâmetros
abertura da bobina da armadura	k_{aa}	{0, 1}
abertura da bobina de campo	k_{af}	{0, 1}
curto-circuito do conversor da armadura	k_{cca}	{0, 1}
curto-circuito do conversor do campo	k_{ccfd}	{0, 1}
falha no sensor de corrente da armadura	$k_{i_a}^f$	{0, 1}
falha no sensor de corrente da fonte	$k_{i_{fd}}^f$	{0, 1}
falha no sensor de velocidade	$k_{\omega_r}^f$	{0, 1}

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -k_{aa} \cdot \frac{r_a}{L_a} & -k_{aa} \cdot \frac{L_{afd}}{L_a} & 0 \\ 0 & -k_{afd} \cdot \frac{r_{fd}}{L_{fd}} & 0 \\ \frac{L_{afd}}{J_m} \cdot x_2 & 0 & -\frac{B_m}{J_m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & \frac{1}{L_{fd}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} k_{aa} \cdot \bar{k}_{cca} \cdot v_a \\ k_{afd} \cdot \bar{k}_{ccfd} \cdot v_{fd} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \cdot [T_L] \quad (1)$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \bar{k}_{i_a}^f & 0 & 0 \\ 0 & \bar{k}_{i_{fd}}^f & 0 \\ 0 & 0 & \bar{k}_{\omega_r}^f \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \text{ onde } \bar{k}_i = 1 - k_i$$

$$T_L = C_0 + C_1 \cdot \omega_r + C_2 \cdot \omega_r^2 \quad (2)$$

Onde:

- v_a : tensão de alimentação do circuito da armadura;
- v_{fd} : tensão de alimentação do circuito de campo;
- i_a : corrente do circuito de armadura;
- i_{fd} : corrente do circuito de campo;
- ω_r : velocidade mecânica de rotação em rad/s;
- r_a : resistência do circuito da armadura;
- r_{fd} : resistência do circuito de campo;
- L_a : indutância do circuito da armadura;
- L_{fd} : indutância do circuito de campo;
- L_{afd} : indutância mútua armadura/campo;
- T_L : conjugado mecânico exigido pela carga;
- B_m : coeficiente de atrito viscoso;
- J_m : momento de inércia do sistema motor/carga.

4. Rede Neurofuzzy

4.1. Descrição

A estrutura da rede neurofuzzy – RNF - [12] é mostrada na figura 4. As características de cada sinapse são representadas por uma função não-linear $f_i(x_i)$, determinada por inferência *fuzzy* e *defuzzificação*. A recodificação dos sinais sinápticos deixa de ser caracterizada por uma função sigmoideal, como nas redes convencionais, passando a ser obtida simplesmente por uma soma algébrica. Isto faz com que a função erro a ser minimizada seja quadrática e convexa com relação aos pesos. Logo, o mínimo local é mínimo global, conforme demonstrado em [14].

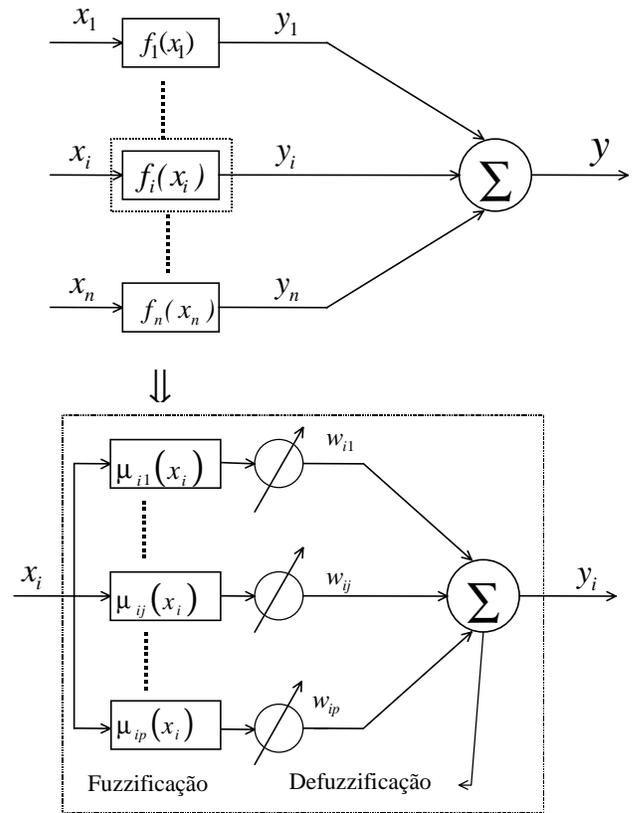


Figura 4: Estrutura da RNF

A saída da RNF é dada por:

$$y = \sum_{i=1}^n y_i(x_i) = \sum_{i=1}^n f_i(x_i) \quad (3)$$

Os valores y_i são determinados a partir de um conjunto equivalente de regras *fuzzy* do tipo *se-então*. Considerando funções de pertinência complementares, os valores das funções de pertinência são diferentes de

zero para, no máximo, duas funções vizinhas. Isto equivale a dizer que, para cada valor da variável de entrada x_i , no máximo duas das p regras são ativadas, sendo estas indexadas por k_i e k_i+1 . Com isto, utilizando o esquema de inferência *max-min* e o centro de gravidade para efetuar a *defuzzificação*, os valores de $f_i(x_i)$ podem ser determinados da seguinte forma:

$$f_i(x_i) = \frac{\sum_{j=1}^p \mu_{ij}(x_i) \cdot w_{ij}}{\sum_{j=1}^p \mu_{ij}(x_i)} \quad (4)$$

$$= \frac{\mu_{ik_i}(x_i) \cdot w_{ik_i} + \mu_{ik_i+1}(x_i) \cdot w_{ik_i+1}}{\mu_{ik_i}(x_i) + \mu_{ik_i+1}(x_i)}$$

Como as funções de pertinência são complementares, a soma de duas funções sucessivas é igual a 1. Portanto, tem-se a seguinte expressão:

$$f_i(x_i) = \mu_{ik_i}(x_i) \cdot w_{ik_i} + \mu_{ik_i+1}(x_i) \cdot w_{ik_i+1} \quad (5)$$

4.2. Treinamento on-line

O algoritmo de treinamento consiste na atualização dos pesos, para cada padrão de entrada, de forma a minimizar o erro E_t :

$$E_t = \frac{1}{2} (y_t - y_t^d)^2 = E_t(w_{ij}) \quad (6)$$

Utilizando o método do gradiente a atualização é feita da seguinte maneira:

$$w_{ik_i}^{j+1} = w_{ik_i}^j - \alpha^j \cdot \frac{\partial E_t(w_{ik_i}^j)}{\partial w_{ik_i}^j} \quad (7)$$

$$w_{ik_i}^{j+1} = w_{ik_i}^j - \alpha^j \cdot (y_t^j - y_t^d) \mu_{ik_i}(x_{ti}) \quad (8)$$

O tamanho do passo, o termo α , que no caso de treinamento de redes neurais ou *neurofuzzy* é chamado de taxa de aprendizado, pode ser determinado empiricamente, ou utilizando algum método de busca unidirecional indireto (por exemplo secção áurea, Fibonacci e outros) ou direto (fazendo aproximação polinomial para a função a ser minimizada). Considerando que a função a ser minimizada é quadrática e convexa, propõe-se em [14] uma expressão para calcular α ótimo tal que, em apenas um passo, obtém-se um conjunto de pesos que proporcionam erro nulo empregando a equação (9).

$$\alpha = \frac{1}{\sum_{i=1}^n \mu_{ik_i}(x_{ti})^2 + \mu_{ik_i+1}(x_{ti})^2} \quad (9)$$

4.3. Aplicação em identificação do sistema de acionamento C.C.

Para exemplificar, considere-se uma RNF para identificação do sistema de acionamento C.C. com a seguinte estrutura:

Entradas:

$$\begin{bmatrix} \hat{i}_a(k-1) & \hat{i}_{fd}(k-1) & \hat{w}_{k-1}(k-1) & v_a(k) & v_{fd}(k) \\ \hat{y}_1(k-1) & \hat{y}_2(k-1) & \hat{y}_3(k-1) & u_1(k) & u_2(k) \end{bmatrix} =$$

Saídas: $[\hat{y}_1(k) \quad \hat{y}_2(k) \quad \hat{y}_3(k)]$

Nas figuras 5 e 6 são mostradas as curvas da variável de saída y_1 (corrente de armadura) estimada e real, bem como o peso w_{121} da RNF, para o primeiro e décimo ciclo de treinamento, respectivamente. Nota-se que no primeiro ciclo, a rede consegue aproximar com erro nulo, com grandes variações nos pesos. À medida que a rede aprende, estas variações dos pesos são menores. Para os resultados apresentados, o número máximo de partições *fuzzy* para cada entrada foi igual a três. As entradas do sistema de acionamento C.C., u_1 e u_2 , foram geradas aleatoriamente.

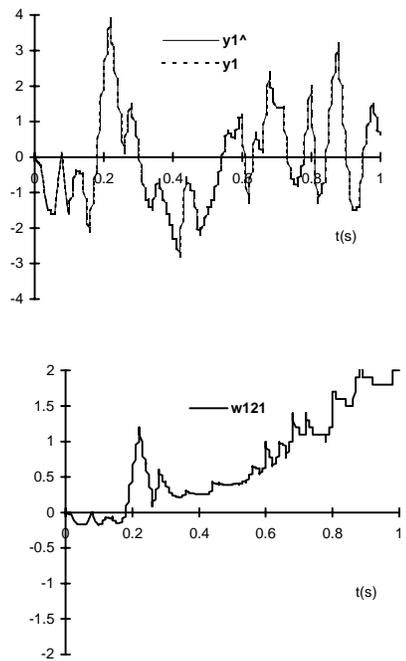


Figura 5: Saída y_1 estimada e real e peso w_{121} da RNF primeiro ciclo de treinamento

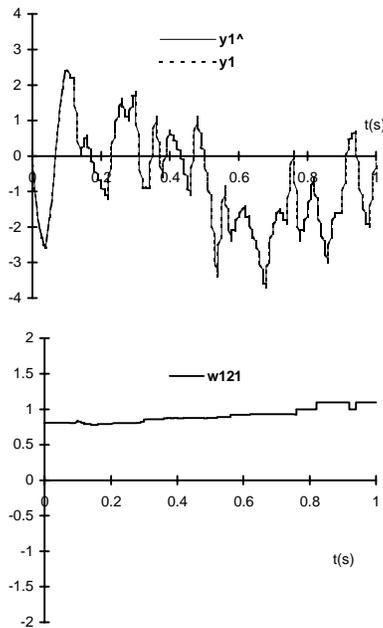


Figura 6: Saída y_1 estimada e real e peso w_{121} da RNF: Décimo ciclo de treinamento

5. Detecção de Falhas no Sistema de Acionamento C.C.

Conforme visto na figura 2, a topologia de detecção de falhas utiliza a variação dos pesos da RNF para geração de resíduos. Os pesos da RNF \mathbf{W}^0 (vetor) são determinados através do treinamento *off-line* da rede. Nesta etapa, os dados fornecidos para o treinamento referem-se à operações do sistema sem ocorrência de falhas. Já os pesos \mathbf{W} são obtidos via treinamento *on-line*. Quando ocorrem falhas no sistema, os pesos \mathbf{W} serão diferentes de \mathbf{W}^0 , sendo que essa diferença é usada para gerar os resíduos, que são coordenadas dos padrões do sistema de classificação. Estas variações nos pesos são justificadas pelo fato que a falha provoca uma alteração do modelo que a RNF tinha aprendido.

Para ilustrar, considere os seguintes resíduos:

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} w_{121} - w_{121}^0 \\ w_{222} - w_{222}^0 \\ w_{313} - w_{313}^0 \end{bmatrix} \quad (10)$$

lembrando que para o peso w_{ijl} , os subscritos i, j e l são os indexadores da entrada da rede, partição *fuzzy* e saída da rede, respectivamente. Os gráficos das simulações são mostrados nas figuras 7 a 10. Observa-se que, quando ocorre uma falha, em $t=0,5$ segundos, para as três simulações há variações nos pesos da RNF. Na figura 7 são mostrados os resíduos para uma falha do

sensor de velocidade. Esta falha leva o valor da velocidade medida a zero. Quando ocorre a falha, há variações nos pesos da RNF, principalmente naqueles que estão associados a saída y_3 , que é a velocidade. A abertura da fonte da armadura provoca variação brusca no peso que está relacionado a entrada $y_3(k-1)$ e a saída $y_3(k)$, figura 8. Para um curto-circuito da fonte de campo, o peso mais afetado está relacionado a corrente de campo y_2 , figura 9. Na figura 10 é mostrada a curva do conjugado de carga para a situação de curto-circuito da fonte de campo. Nota-se que os resíduos são robustos às distúrbios de carga. Para tal, no treinamento da RNF este conjugado foi alterado de forma aleatória.

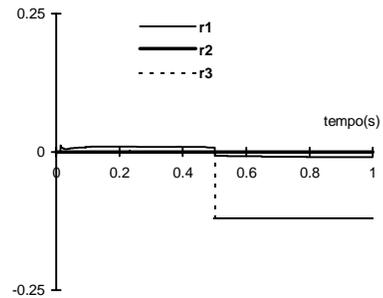


Figura 7: Comportamento dos resíduos para a falha no sensor de velocidade

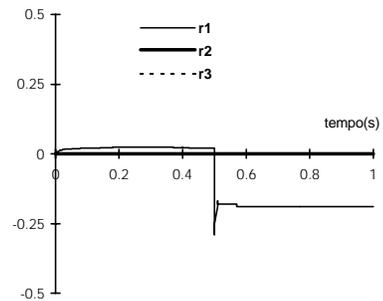


Figura 8: Comportamento dos resíduos para a falha: abertura da fonte da armadura

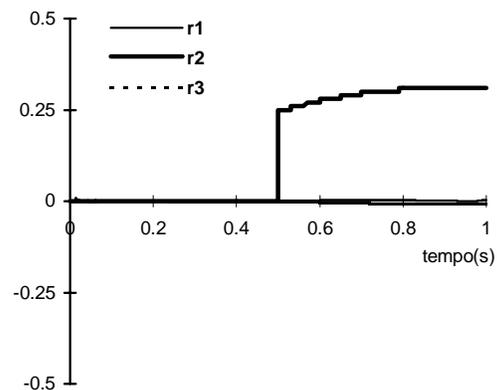


Figura 9: Comportamento dos resíduos para a falha: curto-circuito da fonte de campo

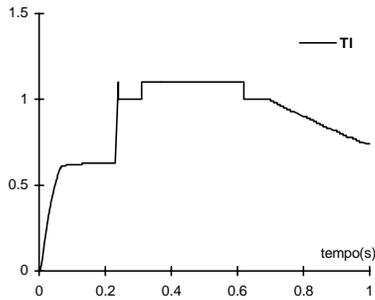


Figura 10: Curva do conjugado de carga para a falha: curto-circuito da fonte de campo

Com as informações dos resíduos é possível apenas detectar a falha. Para permitir o diagnóstico, são necessárias outras informações, tais como valores das correntes e velocidade. Com isto, o sistema de classificação utilizado possui a seguinte topologia:

$$\text{Entradas: } [r_1 \quad r_2 \quad r_3 \quad x_1 \quad x_2 \quad x_3]$$

Número de partições *fuzzy* para cada entrada: três
 Número de saídas: oito (classes)

Para avaliar o desempenho do sistema de detecção e diagnóstico de falhas foram feitas duas mil simulações. Para simular o sistema foi adotado a seguinte metodologia: tempo máximo de 1 segundo; o índice da falha foi gerado de forma aleatória, para cada simulação; o tempo de ocorrência da falha gerado aleatoriamente; distúrbio de carga de amplitude aleatória, cujo valor máximo é 50% do nominal e ruído de medição aleatório e de amplitude 2% do valor nominal da grandeza. Todas as falhas foram detectadas e diagnosticadas corretamente, ou seja, não ocorreu alarmes falsos e nenhuma falha deixou de ser detectada. O tempo média de detecção das falhas foi de 8 ms, que corresponde a 4 períodos de amostragem.

6. Conclusões

Foi proposta uma estratégia para detecção e diagnóstico de falhas em sistemas dinâmicos, empregando uma rede neurofuzzy. Esta estratégia foi utilizada para detectar e diagnosticar falhas em um sistema de acionamento elétrico em corrente contínua. A estratégia proposta é baseada em identificação do modelo da planta através da RNF. Os coeficientes do modelo são os pesos da RNF. As variações nesses coeficientes são usadas na geração de resíduos, que são empregados para detectar as falhas no sistema. A vantagem dessa estratégia, com relação àquelas que utilizam identificação de parâmetros empregando, e.g. o método de mínimos quadráticos recursivo [6], é que o número de operações utilizando da RNF é menor. Além

disso, variações na planta são traduzidas de forma imediata em variações dos pesos da RNF. Este fato proporciona que falhas bruscas sejam detectadas de maneira quase instantânea.

Referências

- [1] P. Frank, and R. Seliger. Fault Detection and Isolation in Automatic Process. *Control and Dynamic Systems*, 49, pp. 241-287, 1991.
- [2] R. Clark. A simplified Instruments Failure Detection Scheme. *IEEE Trans. Aerospace Electron. Syst.*, Vol. 14, pp. 558-563, 1978.
- [3] P. Frank and L. Keller. Sensitivity Discriminating Observer Design for Instrument Failure Detection. *IEEE Trans. Aerospace Electron. Syst.*, Vol. AES-16, pp. 460-467. 1980.
- [4] W. Ge and C. Fang. Detection of Faulty Components Via Robust Observation. *Int. J. Control*, 47: pp. 581-599, 1988.
- [5] P. Frank and J. Wünnenberg. Robust Fault Diagnosis Using Unknown Input Observer Schemes *Em: Fault Diagnosis in Dynamic Systems, Theory and Applications*. R. J. Patton; P. M. Frank & R. N. Clark, Eds), pp. 47-98, 1989.
- [6] R. Iserman. Process Fault Detection Based on Modeling and Estimation Methods – A Survey. *Automatica*, 20, pp. 387-404, 1984.
- [7] H. Camargo. Detecção e Isolação de Falhas Utilizando Estimacão de Parâmetros de Modelos Contínuos e Limites de Confiança Adaptativos - Aplicação a uma Planta Piloto. Dissertação de Mestrado, PPGEE/UFMG, 1997.
- [8] M. Pistauer. Neural Network Design for a Process Fault-Diagnosis Module with Fuzzy System. *Proc. Sixth International Fuzzy Systems Association World Congress - IFSA '95*, São Paulo, Brazil, 2: pp. 591-594, 1995.
- [9] M. Polycarpou and A. Vemuri. Learning Methodology for Failure Detection and Acomodation. *IEEE Control Systems*, June, pp. 16-24, 1995.
- [8] R. Milne. Artificial Intelligence for Online Diagnosis. *IEE Proceedings*, Vol. 134, No. 4, pp. 238-244, 1987.
- [9] M. Ulieru. Diagnosis by Approximate Reasoning on Dynamic Fault Trees. *Proc. 3th IEEE International Conference on Fuzzy Systems*, Orlando, Florida, USA, pp. 2051-2056, 1994.
- [10] T. Yamakwa, E. Uchino, T. Miki and T. Kusanagi. A Neo Fuzzy Neuron and its Applications to System Identification and Predictions to System Behavior. *Proc. Of the 2nd IZUKA, Iizuka-Japan*, pp. 477-483, 1992.
- [11] W. Caminhas, H. Tavares, F. Gomide and W. Pedrycz. Fuzzy Set Based Neural Networks: Structure, Learning and Application, *Journal Of Advanced Computational Intelligence*, 3(3), p. 151-157, 1999.
- [12] W. Caminhas and F. Gomide. A Fast Learning Algorithm for Neofuzzy Networks. *8th IPMU*, Madrid, Spain, 1(1): pp. 1784-1790, 2000.

Agradecimento: O último autor agradece ao CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico, pelo apoio via bolsa pesquisa # 300729/86-03.