

# Um método neurofuzzy para construção de modelos inferenciais

L.V. R. Arruda e Elaine Yassue Nagai

Centro Federal de Educação Tecnológica do Paraná (CEFET-PR)

Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI)

Laboratório de Automação e Sistemas de Controle Avançado (LASCA)

[eynagai@cpgei.cefetpr.br](mailto:eynagai@cpgei.cefetpr.br), [arruda@cpgei.cefetpr.br](mailto:arruda@cpgei.cefetpr.br)

**Resumo:** Este trabalho apresenta uma metodologia para construção de modelos fuzzy inferenciais baseada em redes neurais e agrupamento fuzzy. A partir de um conjunto de medidas de variáveis secundárias do sistema a ser modelado, um mapa de Kohonen é aplicado para selecionar a estrutura de um modelo fuzzy linguístico. Em seguida, a ordem deste modelo é selecionada através dos quocientes de Lipschitz. O algoritmo fuzzy *c-means* é então aplicado aos dados selecionados, gerando um modelo inicial cuja base de regras é sobredimensionada. Em seguida, as funções de pertinência similares são agrupadas, e uma análise de consistência é realizada reduzindo a base de regras final do modelo fuzzy gerado. Esta metodologia tem sido aplicada na construção de sensores virtuais para medidas de inferência em processos de destilação.

## I. Introdução

Em geral, na indústria química, modelos inferenciais podem ser utilizados para estimar a composição de produtos a partir de medidas de temperatura e de outras variáveis secundárias disponíveis on-line. Porém, considerando-se que, na maioria dos casos, processos químicos apresentam comportamento não-linear, a modelagem destes processos mostra-se complexa quando são empregadas abordagens baseadas em filtro de Kalman ou outras técnicas clássicas de estimação [1],[2]. Para superar tais dificuldades, este trabalho propõe uma metodologia para a identificação de modelos *fuzzy* inferenciais a ser utilizados na construção de *sensores virtuais* para a indústria química. Esta metodologia é composta por três etapas. A primeira etapa corresponde à identificação da estrutura do modelo *fuzzy*, onde as variáveis de entrada são selecionadas, dentre o conjunto de variáveis secundárias disponíveis do processo, aplicando-se mapas de Kohonen, e para selecionar a ordem do modelo *fuzzy* são empregados os quocientes de Lipschitz. Após esta fase inicial, o algoritmo *fuzzy c-means* é aplicado no espaço de entrada e saída de modo a encontrar uma base de

regras capaz de modelar a interação entre as variáveis do processo, assim com as suas possíveis não-linearidades. Finalizando, o modelo inicial gerado passa por um processo de simplificação, onde as funções de pertinência similares que descrevem o domínio de cada variável são agrupadas com o objetivo de diminuir a complexidade e a carga computacional e, aumentar a interpretabilidade do modelo final, tornando-o mais simples e conciso.

Este trabalho está organizado como segue. A seção 2 detalha cada procedimento que compõe a metodologia proposta. A seção 3 sumariza o procedimento proposto de identificação de modelos *fuzzy* inferenciais, a seção 4 apresenta os resultados de simulação obtidos na construção de um sensor virtual para inferir a composição do produto de topo de uma coluna de destilação binária, e concluindo o trabalho, a seção 5 apresenta as considerações finais.

## II. Identificação Neurofuzzy

A primeira fase do método proposto para a identificação de modelos *fuzzy* preocupa-se com a seleção do conjunto apropriado de entrada e a determinação da estrutura do modelo (seleção do número de atrasos utilizados para as variáveis de entradas e saída). Estes dois problemas não são independentes, porém, nesta metodologia, eles são tratados de forma independente, através de dois procedimentos distintos, como descritos a seguir.

### A. Seleção das Variáveis de Entrada

Neste trabalho a técnica proposta em [3] é adaptada para selecionar o subconjunto apropriado de variáveis de entrada do modelo *fuzzy* inferencial. O objetivo é encontrar o subconjunto de variáveis que contém a maioria das informações relevantes sobre o processo a ser modelado. Inicialmente, os subconjuntos de variáveis secundárias são especificados como segue. Seja  $p$  o número de variáveis secundárias, calcula-se o coeficiente de correlação entre cada variável secundária e a variável de saída. As  $p$  variáveis são

ordenadas, de forma crescente, conforme os respectivos valores absolutos de coeficiente de correlação, criando o conjunto  $S$  de variáveis secundárias ordenadas. Os  $p$  subconjuntos são construídos de acordo com a seqüência de variáveis em  $S$ , ou seja, o primeiro subconjunto é composto pela primeira variável secundária em  $S$  e a variáveis de saída; o segundo subconjunto é composto pelas duas primeiras variáveis em  $S$  e a variável de saída. Então, um mapa de Kohonen é construído para cada subconjunto e cada mapa é avaliado através da medida de dissimilaridade entre dois mapas distintos  $L_q$  e  $L_r$  [4], definida através das equações 1 e 2.

$$D(L_q, L_r) = E \left[ \frac{\|d_{L_q}(x) - d_{L_r}(x)\|}{d_{L_q}(x) + d_{L_r}(x)} \right], \quad (1)$$

$$d(x) = \|x(t) - w_{bmu1(x)}(t)\| \sum_{i=p_{bmu1}}^{p_{bmu2}-1} \|w_i(t) - w_{i+1}(t)\|. \quad (2)$$

A medida de dissimilaridade informa a relevância de cada combinação de variáveis secundárias em relação a variável de saída, onde  $E$  é a esperança matemática;  $p_{bmu1}$  é a posição da primeira *bmu* (*best map unit* – melhor unidade do mapa);  $p_{bmu2}$  é a posição da segunda melhor *bmu*; e  $d(x)$  é a distância de  $x$  até a segunda *bmu*, denotada por  $w_{bmu2}(t)$ , iniciando-se da primeira *bmu*, denotada por  $w_{bmu1}(t)$ .

A distância  $d(x)$  combina uma indicação da continuidade do mapeamento a partir do conjunto de dados para a estrutura do mapa com uma medida de precisão do mapa na representação do conjunto de dados [4]. O menor valor de dissimilaridade média calculado com a Equação 1 para qualquer subconjunto de variáveis secundárias indica a similaridade na qualidade e na quantidade da informação representada pelos mapas. O Quadro 1 sumariza o procedimento de seleção de variáveis secundárias.

- **Passo 1:** Seja  $U = \{u_1, \dots, u_p\}$  o conjunto de variáveis secundárias e  $y$  a variável de saída. Calcular o coeficiente de correlação entre cada variável secundária e a de saída;
- **Passo 2:** Ordenar as variáveis secundárias de acordo com o valor absoluto do coeficiente de correlação calculado no passo 1, criando o conjunto  $S = \{s_1, \dots, s_p\}$ , onde  $\|corr(s_i, y)\| \geq \|corr(s_{i+1}, y)\|$ , e  $\|\cdot\|$  é o valor absoluto;
- **Passo 3:** Definir o número  $m$  de unidades dos mapas auto-organizáveis (SOMs: *Self Organizing Maps*);
- **Passo 4:** Seja  $p$  o número de variáveis secundárias. Definir os  $p$  subconjuntos de variáveis secundárias ( $c_1, \dots, c_p$ ) a partir do conjunto  $S$ , onde  $c_1 = \{s_1, y\}$ ,  $c_2 = \{s_1, s_2, y\}$ , ...,  $c_p = \{s_1, \dots, s_p, y\}$ ;
- **Passo 5:** Definir um SOM com  $m$  unidades para cada subconjunto  $c_i$  e treinar esses SOMs com os dados de

treinamento;

- **Passo 6:** Para cada par de SOMs  $L_q$  e  $L_r$ ,  $q \neq r$ , calcular a medida de dissimilaridade (Equação 1);
- **Passo 7:** Calcular a média de dissimilaridade de cada SOM  $L_q$  com relação a todos os outros SOMs;
- **Passo 8:** O mapa com o menor valor de média de dissimilaridade contém a melhor configuração, ou seja, o melhor subconjunto de variáveis secundárias para compor o espaço de entrada do modelo *fuzzy*.

Quadro 1. Algoritmo de seleção de variáveis de entrada.

## B. Seleção da Estrutura do Modelo Fuzzy

O método proposto em [5] é aplicado para selecionar a ordem do modelo *fuzzy*. Este método utiliza os quocientes de Lipschitz, calculados a partir do conjunto de dados amostrados do processo,  $\{x(t), y(t)\}$ ,  $t = 1, \dots, N_d$ , onde  $x(t)$  é o vetor contendo todas as  $n$  variáveis de entrada (selecionadas anteriormente) e  $y(t)$  é a saída do modelo *fuzzy*, e do vetor de regressores definido pela Equação 3 e Equação 4:

$$\varphi(t) = [y(t-1), \dots, y(t-lag_y), x(t-1), \dots, x(t-lag_x)]^T, \quad (3)$$

$$Z^{N_d} = \{\{\varphi(t), y(t), \quad t = 1, \dots, N_d\}\}, \quad (4)$$

onde  $lag_y$  é o atraso da variável de saída;  $lag_x$  é o atraso da variável de entrada; e  $Z^{N_d}$  é o conjunto de  $N_d$  pares de entrada e saída. Portanto, para todas as combinações de pares de entrada-saída, o quociente de Lipschitz é calculado como na Equação 5.

$$q_{i,j} = \frac{\|y(t_i) - y(t_j)\|}{\|\varphi(t_i) - \varphi(t_j)\|}, \quad i \neq j. \quad (5)$$

O procedimento completo de seleção da ordem do modelo *fuzzy* é apresentado no quadro 2:

- **Passo 1:** Para uma dada escolha de atraso, determinar os quocientes de Lipschitz para todas as combinações de pares de entrada e saída.
- **Passo 2:** Selecionar os  $r$  maiores quocientes,  $r = 0.01N_d \approx 0.02N_d$ . Os maiores quocientes ocorrem, em geral, quando as diferenças  $\varphi(t_i) - \varphi(t_j)$  são pequenas.
- **Passo 3:** Avaliar o seguinte critério:
- $$\bar{q}^{(z)} = \left( \prod_{i=1}^r \sqrt{n} q^{(z)}(t) \right)^{\frac{1}{r}},$$
- onde  $q^{(z)}$  é o quociente referente ao atraso  $z$ .
- **Passo 4:** Repetir os cálculos para um determinado número de atrasos distintos.
- **Passo 5:** Plotar o critério em função do atraso e selecionar o número ótimo de regressores no “*knee-point*” da curva.

Quadro 2. Algoritmo de seleção da estrutura do modelo

### C. Agrupamento Fuzzy

Um algoritmo de agrupamento realiza, essencialmente, a tarefa de particionamento de um conjunto de padrões em um número de grupos (*clusters*) homogêneos, com respeito a uma determinada medida de similaridade. Os padrões pertencentes a qualquer um dos grupos são similares entre si, e os padrões de *grupos* distintos são tão “dissimilares” quanto possível. Na análise clássica de *grupos*, o limite dos diferentes grupos é *crisp*, tal que um padrão é atribuído a somente um *grupo*. Na prática, os dados geralmente não são bem distribuídos, sendo que o limite pode não ser precisamente definido. Portanto, um mesmo dado pode pertencer a dois ou mais grupos com diferentes graus de pertinência. Esta é a idéia por trás dos algoritmos de agrupamento *fuzzy*.

**Definição 1** *Partição-c Fuzzy.* Seja um conjunto de dados  $X=\{x_1, \dots, x_n\}$ , onde  $x_k \in \mathcal{R}^p$ . Seja  $P(X)$  o conjunto potência de  $X$ , ou seja, o conjunto de todos os subconjuntos de  $X$ ,  $V_c$  o conjunto de matrizes  $U=[u_{ik}] \mathcal{R}^{n \times c}$ , e  $c$  um número inteiro onde  $2 \leq c \leq n$ . O espaço da partição- $c$  fuzzy  $X$  é o conjunto:

$$M_{fc} = \left\{ U \in V_c \mid u_{ik} \in [0,1], 1 \leq i \leq c, 1 \leq k \leq n, \text{ e } \sum_{i=1}^c u_{ik} = 1, \forall k \in \{1, 2, \dots, n\} \right\} \quad (6)$$

onde  $u_{ik}$  é o valor de pertinência de  $x_k$  ao grupo  $C_i$ .

O algoritmo *Fuzzy C-Means (FCM)* é um algoritmo de agrupamento que realiza uma partição *c-fuzzy* de  $M_{fc}$ ,  $U=[u_{ik}] \in M_{fc}$ , tal que a função objetivo  $J_m(U, V)$ , dada pela Equação 7 seja minimizada.

$$J_m(U, V) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m \|x_k - v_i\|^2 \quad (7)$$

Onde  $m \in (1, \infty)$  é uma constante de ponderação que determina a sobreposição dos grupos e  $V=(v_1, \dots, v_c)$  com  $v_i \in \mathcal{R}^p$  s. A otimização aproximada de  $J_m$  realizada pelo algoritmo *FCM* é um procedimento iterativo como descrito no Quadro 3.

- **Passo 1:** Para um dado conjunto de dados  $X$ , estipule  $c \in \{2, 3, \dots, N_d - 1\}$ ,  $m \in (1, \infty)$ , e inicialize  $H^{(0)} \in M_{fc}$ .
- **Passo 2:** Na iteração  $q$ ,  $q=0, 1, 2, \dots$ , calcule os vetores  $V$ , com o novo centro de cada grupo  $C_i^q$ ;

- $$v_i^{(q)} = \frac{\sum_{k=1}^{N_d} (h_{ik}^{(q)})^m x_k}{\sum_{k=1}^{N_d} (h_{ik}^{(q)})^m}$$
- **Passo 3:** Atualize  $H^{(q)} = [h_{ik}^{(q)}]$  para  $H^{(q+1)} = [h_{ik}^{(q+1)}]$  tal que:
  - $$h_{ik}^{(q+1)} = \frac{1}{\sum_{j=1}^c \left( \frac{\|x_k - v_i^{(q)}\|}{\|x_k - v_j^{(q)}\|} \right)^{\frac{2}{m-1}}}$$
  - onde  $1 \leq i \leq c$ ,  $1 \leq k \leq N_d$ .
- **Passo 4:** Se  $\|H^{(q)} - H^{(q+1)}\| < \varepsilon$ , então pare o processo; caso contrário, incremente  $q$ ,  $q=q+1$ , e volte ao passo 2.

Quadro 3. Algoritmo de agrupamento *Fuzzy C-Means*.

### D. Geração da Base de Regras Inicial

Nesta fase, o algoritmo *FCM* é aplicado no espaço de entrada e saída e as amostras do conjunto de dados são classificadas em *ncr grupos* de modo a gerar uma base de regras superestimada, onde cada grupo corresponde a uma regra. Um valor superdimensionado do tamanho da base de regras inicial (*ncr*) é empregado na tentativa de cobrir todas as regiões importantes do espaço de entrada e saída, tornando o resultado do agrupamento menos dependente da partição inicial do domínio das variáveis [6].

As funções de pertinência referentes aos conjuntos *fuzzy* das regras são obtidas a partir da matriz de partição  $H$ . Os conjuntos *fuzzy* unidimensionais correspondentes a cada grupo/regra são obtidos a partir dos valores discretos da função de pertinência multidimensional contida na  $i$ -ésima linha da matriz  $H$ , através de sua projeção *fuzzy* no domínio de cada variável. A Figura 1 ilustra a projeção dos grupos nos domínios das variáveis  $x$  e  $y$ , gerando a seguinte base de regras:

C1: Se  $x$  é A1 Então  $y$  é B1

C2: Se  $x$  é A2 Então  $y$  é B2

C3: Se  $x$  é A3 Então  $y$  é B3

Este procedimento gera os pontos discretos das funções de pertinência em cada domínio, que podem ser aproximados por uma função de pertinência paramétrica. Neste trabalho esta função é gaussiana em que o elemento com o maior grau de função de pertinência é o centro da função gaussiana e o desvio padrão é calculado através dos valores discretos que descreve o conjunto *fuzzy*. O Quadro 4 apresenta o procedimento de geração da base de regras inicial.

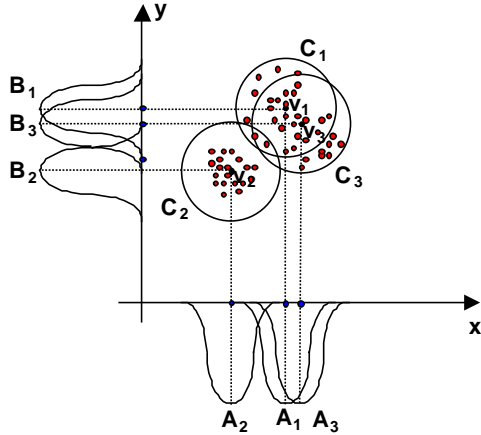


Figura 1: Projeção dos grupos nos espaços de entrada e saída.

- **Passo 1:** Estipular o número de grupos  $ncr$  (tamanho da base de regras inicial);
- **Passo 2:** Aplicar o algoritmo FCM no espaço de entrada e saída de modo a obter a localização dos grupos nesse espaço multidimensional;
- **Passo 3:** Projetar os grupos no subespaço de cada variável com a projeção *fuzzy* da matriz  $H$  obtida com o algoritmo FCM;
- **Passo 4:** Calcular os parâmetros das  $ncr$  funções de pertinência gaussianas que descrevem o domínio de cada variável.

Quadro 4. Algoritmo de geração da base de regras inicial.

### E. Simplificação da Base de Regras Inicial

A transparência de um modelo *fuzzy* é um dos aspectos que distinguem a modelagem *fuzzy* de métodos de modelagem tipo caixa-preta. Porém essa transparência não é alcançada automaticamente. Quando um modelo *fuzzy* é desenvolvido utilizando-se de conhecimento especialista, é possível controlar o número de regras e de funções de pertinência geradas. No entanto, torna-se difícil controlar a redundância e a complexidade desnecessária quando a identificação do modelo é realizada de forma automática, a partir de pares amostrados de dados de entrada e saída, como é o caso desta implementação.

Nesta metodologia, o número de funções de pertinência de cada variável não é pré-determinado, sendo possível a existência de funções de pertinência similares. Funções de pertinência similares são funções altamente sobrepostas, que descrevem quase a mesma região no domínio da variável a qual esses conjuntos pertencem (como os conjuntos  $A_1$  e  $A_3$  na Figura 1), fazendo com que o modelo utilize mais conjuntos *fuzzy* do que o necessário [7]. Assim para cada domínio, as funções de pertinência cujos valores são muito

próximos são consideradas similares e devem ser agrupadas, ou seja, os parâmetros dessas funções são substituídos por novos parâmetros, tal que o novo valor de centro é a média dos valores de centro das funções de pertinência similares, e o novo valor de desvio padrão é o maior valor de desvio padrão dentre as funções similares. Após este procedimento, as regras são atualizadas com as novas funções de pertinência. A diminuição do tamanho da base de regras é uma consequência dessa simplificação. A melhoria da estrutura realizada nesta fase desempenha um importante papel na identificação de modelos *fuzzy*, pois ela aumenta a interpretabilidade e transparência do modelo, duas características essenciais dos sistemas *fuzzy*.

### F. Tratamento de Regras Inconsistentes

Considerando-se que o número de regras geradas pelo algoritmo FCM é muito grande, é provável que existam regras inconsistentes, ou seja, regras que possuem o mesmo antecedente e conseqüentes diferentes. Além disso, a união de conjuntos *fuzzy* similares também pode causar inconsistência em regras. Para resolver esse conflito, é utilizado a medida do fator de confiança  $CF$  de uma regra. Seja  $A \rightarrow C$  uma regra, onde  $A$  é o antecedente (uma conjunção de condições) e  $C$  é o conseqüente da regra. A medida  $CF$  é dada pela Equação 8 [7], e em um grupo de regras inconsistentes, a regra com o maior valor de  $CF$  é mantida, e as demais são removidas da base de regras.

$$CF(R_i) = \frac{|A \wedge C| - 1/2}{|A|}, \quad (8)$$

onde  $|./|$  denota cardinalidade, e  $R_i$  é a  $i$ -ésima regra inconsistente.

### III. Sumário do Procedimento de Identificação

- **Passo 1:** Selecionar o conjunto de entrada do modelo *fuzzy* a partir do conjunto de variáveis secundárias disponíveis através do algoritmo descrito na Figura 2;
- **Passo 2:** Selecionar a estrutura do modelo como descreve o algoritmo da Figura 3;
- **Passo 3:** Estabelecer os conjuntos de treinamento e teste, o tamanho da base de regras inicial  $ncr$ , o parâmetro  $m$  do algoritmo FCM, e o erro aceitável  $\xi$ ;
- **Passo 4:** Gerar a base de regras inicial conforme descreve a Seção 2.4;
- **Passo 5:** Simplificar a base de regras inicial, agrupando as funções de pertinência similares, e

atualizar a base de regras com as novas funções de pertinência;

- **Passo 6:** Verificar a existência de regras inconsistentes na base de regras, calcular o CF para cada regra inconsistente. A regra com o maior valor de CF é mantida e as demais são removidas da base de regras;
- **Passo 7:** Calcular o erro ao aplicar o conjunto de teste ao modelo *fuzzy*. Se o erro é maior do que  $\xi$ , então retornar ao passo 4, caso contrário finalizar o procedimento de identificação.

#### IV. Resultados Experimentais

A metodologia proposta foi usada na construção de um sensor virtual para estimar a composição do produto de topo de uma coluna de destilação binária durante a sua fase de inicialização. Os dados foram obtidos de [8] onde esta coluna é modelada com o auxílio da ferramenta computacional HYSYS<sup>®1</sup>.

A coluna possui 20 pratos e o seu perfil de temperatura é determinado por quatro sensores posicionados nos pratos 9 e 13 acima do ponto de alimentação, e nos pratos, 16 e 18, abaixo deste ponto. Quando as temperaturas lidas nos pratos aproximam-se do perfil desejado para funcionamento em regime estacionário, o processo de inicialização chega ao final [8]. Nesta situação, a composição de destilado (metanol) retirado da coluna apresenta concentração maior que 99,9%, indicando o momento onde é possível iniciar a retirada de produtos da coluna. É importante obter estimativas das composições para não correr o risco de retirar da coluna um produto impuro (com mais de 2% de impurezas) ou iniciar a retirada do resíduo enquanto este ainda possui uma quantidade de metanol diluída maior que 2% [8]. Em ambientes industriais, utilizam-se testes laboratoriais para a determinação desta composição, o que pode levar horas, ou mesmo dias, retardando o início da retirada de produtos.

Um sensor virtual foi então desenvolvido para inferir a composição do produto de topo (metanol) e do produto de base (água) numa faixa de operação entre 90% e 99,9% de pureza para estimar a composição do metanol. Foram gerados conjuntos de dados para obtenção do sensor virtual (calibração) e um conjunto para testes correspondendo a valores das variáveis secundárias disponíveis descritas na Tabela 1 e das variáveis de entrada / saída selecionadas na Tabela 2. Esta tabela também mostra o número de funções de pertinência para cada variável do modelo (inclusive para a variável de saída) obtidas após o procedimento

de simplificação do modelo *fuzzy*. A Tabela 3 apresenta as informações relevantes sobre o modelo *fuzzy* inferencial obtido, tais como o tamanho da base de regras e o número de variáveis de modelo. Como descrito na Tabela 2 foram selecionadas sete variáveis de entrada, e considerando-se um modelo dinâmico com atraso selecionado ( $lag_y=lag_u=2$ ), o espaço de entrada do modelo *fuzzy* é composto por 15 variáveis.

Tabela 1. Descrição das variáveis secundárias.

variável	descrição
<i>feed1_pv</i>	vazão molar de alimentação
<i>reflux_pv</i>	refluxo do condensador para a coluna
<i>pv_press</i>	pressão no topo da coluna
<i>bottom_pv</i>	nível líquido do refeedor
<i>top_pv</i>	nível líquido do condensador
<i>tray9</i>	temperatura do prato 9
<i>tray13</i>	temperatura do prato 13
<i>tray16</i>	temperatura do prato 16
<i>tray18</i>	temperatura do prato 18
<i>cond_flow</i>	vazão para o condensador
<i>top_flow</i>	vazão do produto de topo (destilado)
<i>bottom_flow</i>	vazão do produto de base (resíduo)

Tabela 2. Variáveis de entrada selecionadas.

#	variáveis	n ° de funções de pertinência
u <sub>1</sub>	<i>feed1_pv</i>	8
u <sub>2</sub>	<i>top_pv</i>	8
u <sub>3</sub>	<i>tray9</i>	9
u <sub>4</sub>	<i>tray13</i>	10
u <sub>5</sub>	<i>tray16</i>	8
u <sub>6</sub>	<i>tray18</i>	7
u <sub>7</sub>	<i>cond_flow</i>	8
y <sub>7</sub>	composição do metanol	9
y	composição do metanol	11

Tabela 3. Parâmetros do modelo *fuzzy* inferencial.

descrição	valor
tamanho da base de regras inicial (ncr)	100
tamanho da base de regras simplificada	45
número de variáveis de entrada	15
número de variáveis de saída	1
atraso selecionado	2

Para validar o sensor virtual construído a partir do modelo *fuzzy* da tabela 3, foram empregados dois índices de desempenho, o *mse* (*mean-square error* – média do erro quadrático) e o *nrmse* (*normalized root mean-square error* – erro normalizado) (Nagai,

<sup>1</sup> HYSYS 3.0, HYPROTECH Ltd.

2002). Os resultados obtidos com os dados de calibração e de teste são mostrados na Tabela 4.

Tabela 4. Índices de desempenho do sensor *neurofuzzy*

	treinamento	teste	total
<i>mse</i>	0.0482	0.0505	0.0494
<i>nrmse (%)</i>	0.2287	0.2348	0.2318

A Figura 2 apresenta a composição do produto de topo inferida pelo sensor virtual (linha pontilhada) e a composição de saída fornecida pelo simulador (linha contínua) no final da fase de start-up.

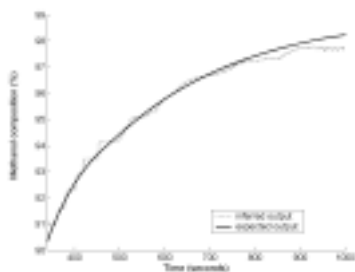


Figura 2. Inferência da composição do produto no topo da coluna.

Através da Figura 2 observa-se que a curva da saída inferida pelo sensor virtual reproduz a tendência da curva simulada, porém é possível notar que o modelo *fuzzy* apresenta deficiências em inferir as saídas próximas aos limites do universo de discurso (neste caso valores próximos de 90% e 99% de concentração de metanol). Isto se deve ao fato das regras e funções de pertinência serem geradas a partir do algoritmo FCM, que tende a enfatizar os pares de dados próximos aos centros dos grupos e, portanto, os dados longe desses centros não são tratados devidamente.

## V. Considerações Finais

Este trabalho propõe uma metodologia para a identificação de modelos *neurofuzzy* inferenciais para a construção de sensores virtuais. Esta metodologia é composta por três fases. Inicialmente as variáveis de entrada são selecionadas aplicando-se mapas de Kohonen e os quocientes de Lipschitz são utilizados para selecionar a ordem do modelo. Então o algoritmo FCM é aplicado no espaço de entrada e saída de modo a encontrar o modelo *fuzzy* inicial. Na terceira fase, as funções de pertinência similares são agrupadas, eliminando a redundância na descrição do domínio de cada variável do modelo.

A metodologia proposta foi utilizada na construção de um sensor virtual para a estimação da medida de composição do produto de topo de uma coluna de destilação binária. Os resultados apresentados indicam que o estimador de composição é consistente mas seu desempenho pode ser melhorado com o uso de técnicas de refinamento do modelo *fuzzy* inferencial, tais como um método para a sintonia dos parâmetros das funções de pertinência e um método para a avaliação da relevância de cada regra do modelo de modo a reduzir o tamanho da base de regras.

## VI. Agradecimentos

Este trabalho conta com o apoio financeiro da Agência Nacional do Petróleo - ANP - e da Financiadora de Estudos e Projetos - FINEP - por meio do Programa de Recursos Humanos da ANP para o Setor Petróleo e Gás - PRH-ANP/MCT (PRH10-CEFET-PR).

## VII. Referências

- [1] Zambogna, E., Barolo, M., Seborg, D. E. Optimal Selection Of Soft Sensor Inputs For Batch Distillation Columns Using Principal Component Analysis. *Journal Of Process Control*, V. 15, P. 39-52, 2005.
- [2] Lant, P. A., Montague, G. A., Morris, A. J., Tham, M. T. Soft-Sensors For Process Estimation And Inferential Control. *Journal of Process Control*, v. 01, p. 3-14, 1991.
- [3] Rallo, R., Ferre-Gine, J. Arenas, A., Giral, F. Neural Virtual Sensor For The Inferential Prediction Of Product Quality From Process Variables. *Computers And Chemical Eng.*, V. 26, P. 1735-1754, 2002.
- [4] Kaski, S., Lagus, K. Comparing Self-Organizing Maps. *In Proc. Of Icann'96*, 1996.
- [5] He, X., Asada, H. A New Method For Identifying Orders Of Input-Output Models For Nonlinear Dynamical Systems. *In Proc. Of The American Control Conference*, P. 2520-2523, 1993.
- [6] SETNES, M. Supervised *fuzzy* clustering for rule extraction. *IEEE Transaction on Fuzzy Systems*, v. 08, n. 04, p. 416-424, 2000.
- [7] Nagai, E. Y. *Métodos De Identificação De Modelos Fuzzy A Partir De Dados Numéricos*. Dissertação De Mestrado, Cpgei/Cefet-Pr, Curitiba-Pr-Brasil.
- [8] Fabro, J. A. *Uma Abordagem Neuro-Nebulosa Para Controle Preditivo De Processos Multi-Estágios*, Tese De Doutorado, Cpgei/Cefet-Pr, Curitiba-Pr-Brasil.