

UM MÉTODO NEUROFUZZY PARA CONSTRUÇÃO DE SENSORES VIRTUAIS

L. V. R. Arruda¹
Flávio Neves-Jr.
Elaine Y. Nagai

Universidade Tecnológica Federal do Paraná
80230-901 Curitiba-PR Brasil
Email: lvrarruda@utfpr.edu.br

Resumo – Este trabalho apresenta uma metodologia para construção de sensores virtuais baseada em técnicas de inteligência computacional. O sensor virtual é projetado em duas etapas. Na primeira, a partir de um conjunto de medidas de variáveis secundárias do sistema a ser modelado e de uma análise de correlação, um mapa de Kohonen é aplicado para selecionar a estrutura de um modelo *fuzzy* lingüístico. Em seguida, a ordem deste modelo é selecionada através dos quocientes de Lipschitz. Na segunda etapa, um algoritmo de agrupamento do tipo *fuzzy c-means* é 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 com sucesso na construção de sensores virtuais para medidas de composição em processos de refino.

Abstract – This paper presents an approach to build a soft sensor based on computational intelligence techniques. The goal is to identify fuzzy models from numerical data. The soft sensor design is carried out in two steps. First, the fuzzy model input variables are selected from the secondary variables set by means of non-linear correlation analysis and Kohonen maps. Then, the Lipschitz quotients are used to select the lag structure of the fuzzy model. In the second step, a fuzzy clustering algorithm is applied to find an initial rule base, and, to conclude the identification process, this initial rule base is simplified by merging the similar membership functions. The validity of the proposed identification method is demonstrated by the development of a soft sensor to infer product compositions of refining process.

Palavras Chaves – Identificação, mapa de Kohonen, sistema *fuzzy*, sensor virtual.

1 Introdução

Nos últimos anos, um novo paradigma tem surgido na comunidade industrial de controle de processos: o conceito de sensor virtual. Um sensor virtual é uma ferramenta de software capaz de estimar qualquer variável de um processo ou de medir a qualidade do produto através de um modelo matemático. Este modelo substitui o sensor real e realiza suas medidas a partir de dados obtidos de outros sensores [1]. Desta forma, o desenvolvimento de sensores virtuais está relacionado às áreas de identificação e validação de modelos para controle de processos [2].

Em geral, na indústria química, os sensores virtuais podem ser utilizados para estimar variáveis de processos quando sensores por hardware não estão disponíveis. Tais estimadores inferenciais são também úteis quando limitações técnicas e/ou financeiras impedem o uso *in-line* de sensores por hardware. Os sensores virtuais possibilitam uma estimação em tempo real das variáveis, evitando os atrasos introduzidos por equipamentos muito lentos tais que cromatógrafos e analisadores, melhorando assim o desempenho das estratégias de controle [1]. De um modo específico, os sensores virtuais podem ser utilizados para inferir a composição de produtos em processo químicos a partir de medidas de temperatura e de outras variáveis secundárias disponíveis no processo. Porém, considerando-se que, na maioria dos casos, os 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 [3,4].

Por estes motivos, abordagens alternativas baseadas em técnicas de inteligência computacional têm sido recentemente propostas para o projeto de sensores virtuais. Uma boa revisão destas abordagens é dada em [5,6] e as aplicações mais recentes são relacionadas em [1]. Esses trabalhos destacam a utilização de redes neurais artificiais e de sistemas *fuzzy* / *neurofuzzy* como metodologias apropriadas para a construção de modelos a partir de dados numéricos. As vantagens das redes neurais são a habilidade de aprendizado adaptativa, a capacidade de processamento paralelo e boas propriedades de generalização. Entretanto, os valores das conexões (pesos) os quais caracterizam uma rede neural são difíceis de entender e explicar, conferindo pouca interpretabilidade ao modelo obtido. Por outro lado, a combinação de redes neurais e sistemas *fuzzy*, gerando um sistema *neurofuzzy*, parece ser o meio mais fácil de inserir conhecimento a priori dentro de um sensor virtual. Um problema fundamental desta abordagem é o da dimensão do sistema final. Quando a dimensão do espaço de entrada aumenta, a base de regras *fuzzy* aumenta exponencialmente levando a um aumento no custo computacional, na memória do sensor e na quantidade de dados necessária para o treinamento ou calibração do sensor [7]. Além disso, outros aspectos importantes devem ser considerados quando se identificam modelos a partir de dados reais [8]:

- Flexibilidade: a estrutura do modelo deve ser capaz de representar as diferentes dinâmicas apresentadas na aplicação.
- Parcimônia: o modelo deve conter o menor número de parâmetros livres necessário a uma representação adequada do sistema.
- Acurácia/Precisão: o modelo deve ser capaz de inferir valores com erros mínimos e com igual precisão em todas as faixas de operação do sistema, mesmo em presença de perturbações.

Apesar do problema de dimensão acima citado, os sistemas *fuzzy* são úteis na modelagem dos aspectos qualitativos do conhecimento e do raciocínio humanos, lançando mão de uma análise quantitativa precisa [9]. Em muitas aplicações, o conhecimento necessário pode não estar facilmente disponível e os usuários são incapazes de extrair este conhecimento de uma quantidade massiva de dados. Esta situação motiva o desenvolvimento de técnicas computacionais para a extração e representação automática do conhecimento utilizando sistemas *fuzzy* baseados em regras. Apesar de importante, a acurácia não é o único aspecto a ser priorizado no desenvolvimento de modelos *fuzzy*. Nos últimos anos, a interpretabilidade tem sido objeto de especial atenção [10]. No entanto ainda não existe uma definição formal de interpretabilidade [11,12]. Neste artigo, interpretabilidade é considerada uma combinação de três características básicas: consistência, compacticidade e visibilidade. A consistência de uma base de regras presume a ausência de regras conflitantes, isto é, regras como o mesmo antecedente e com diferentes conseqüentes. Compacticidade está relacionada com o conceito de parcimônia e é medida pelo número de regras *fuzzy* presentes na base de regras. Visibilidade está relacionada com as partições do universo de discurso a fim de assegurar a granularidade dos domínios sem intervalos ou sobreposição total das funções de pertinência. Assim um modelo *fuzzy* que combine uma boa precisão/acurácia com um número reduzido de regras pode ser uma solução promissora para a representação de sensores virtuais.

Nesse contexto, o presente artigo propõe uma metodologia para a identificação de modelos *fuzzy* lingüísticos a ser utilizada na construção de sensores virtuais, também chamados de modelos inferenciais, para a indústria química. Esta metodologia é implementada em duas etapas. A primeira etapa corresponde à identificação da estrutura do modelo *fuzzy* e ocorre em três passos. Inicialmente as variáveis secundárias disponíveis do processo são avaliadas e ordenadas através de uma análise de correlação não-linear, gerando sub-conjuntos de variáveis de entrada. Estes sub-conjuntos são analisados através mapas de Kohonen e medidas de dissimilaridade, selecionando-se o conjunto de variáveis de entrada para o modelo. No terceiro passo são empregados os quocientes de Lipschitz na seleção da ordem do modelo *fuzzy*. A segunda etapa, que corresponde à identificação dos parâmetros do modelo também ocorre em três passos. Inicialmente, o algoritmo *fuzzy c-means* é aplicado nos espaços de entrada e saída de modo a encontrar as funções de pertinência e uma base de regras capaz de modelar a interação entre as variáveis do processo, assim como as suas possíveis não-linearidades. Em seguida, 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. Finalmente, uma análise de consistência é realizada a fim de assegurar a acurácia do modelo identificado.

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 experimental. Finalmente, a seção 5 apresenta as considerações finais, concluindo o artigo.

2 Identificação Neurofuzzy

Qualquer procedimento de identificação de sistemas que leve à construção de modelos a partir de dados, ou seja, modelos caixa-preta ou caixa-cinza, deve contemplar quatro aspectos fundamentais [2,8]:

- A obtenção de um conjunto de dados que represente adequadamente o sistema a ser modelado (seleção de variáveis de entrada e saída).
- A seleção da estrutura do modelo.
- Um método de identificação de parâmetros.
- Um conjunto de testes que servem para avaliar a qualidade do modelo final.

O modelo desenvolvido neste artigo pode ser considerado um modelo não linear do tipo caixa cinza, uma vez que a base de regras de um modelo *fuzzy* lingüístico representa um mapeamento compreensível (interpretável) entre as variáveis de entrada e saída de um sistema. Além disso, para modelos *fuzzy*, os aspectos de seleção da estrutura do modelo e identificação de parâmetros são interdependentes e envolvem as seguintes escolhas:

- Estrutura das regras. Esta escolha está relacionada ao tipo de modelo (lingüístico, relacional ou Takagi-Sugeno) e a forma dos antecedentes da regra.

- Número e forma das funções de pertinência para cada variável. Estas definições são relacionadas à acurácia/precisão do modelo final e determinam a partição (granularidade) dos universos de discursos para cada uma das variáveis.
- Mecanismo de inferência e método de defuzificação. Estas definições estão associadas às normas utilizadas para representar as operações de implicação, agregação de antecedentes e regras e, operação de defuzificação, os quais dependem do tipo de modelo utilizado (lingüístico, relacional ou Takagi-Sugeno).

Os vários aspectos acima citados são considerados para o desenvolvimento do método de construção de sensores virtuais proposto neste artigo e detalhado a seguir.

2.1. Seleção das Variáveis de Entrada / Saída

A maior parte dos processos industriais é equipada com medidores *on-line* de temperatura, pressão, vazão e outros. Assim, um banco de dados com uma grande quantidade de dados históricos, medidos da planta, encontra-se disponível para a construção de modelos inferenciais. No entanto, parte destes dados acaba tornando-se irrelevante na elaboração dos sensores virtuais e para melhorar a acurácia do modelo devem ser eliminados da base de dados. Além disto, se estas variáveis irrelevantes são incluídas no modelo, elas funcionam como ruído dificultando as tarefas de medição. Desta forma, a primeira fase do método proposto para a identificação de modelos *fuzzy* e construção de sensores virtuais preocupa-se com a seleção do conjunto apropriado de variáveis de entrada e com a determinação da estrutura do modelo, isto é, a escolha da ordem e dos atrasos utilizados para cada uma das variáveis de entrada e saída. Estes dois problemas são interdependentes, porém, nesta metodologia, eles são tratados de forma independente, através de dois procedimentos distintos, como descritos a seguir.

Um procedimento eficiente para seleção de variáveis é um fator de sucesso para a construção de sensores virtuais. Em geral esta seleção de variáveis é feita utilizando técnicas de projeção ou análise estatística tais que análise de componentes principais, estatística descritiva, análise de correlação entre outras. A metodologia consiste em mapear as relações entre todas as variáveis e escolher um novo subconjunto de variáveis de entrada a partir de várias combinações de variáveis relacionadas. Este procedimento resulta em um novo espaço de entrada de baixa dimensão sem perda de informação significativa. O problema desta abordagem é que as novas variáveis são difíceis de interpretar em termos das variáveis primitivas do processo, prejudicando assim a interpretabilidade do modelo obtido. Analogamente a decomposição ortogonal pode não ser a melhor abordagem para reduzir o espaço de entrada [13]. Um método alternativo baseado em mapas de Kohonen foi proposto por Rallo e colaboradores [13] para mapear as relações entre as variáveis do processo e extrair subconjuntos com características relevantes a partir do espaço completo de entrada. Estes mapas auto-organizáveis (*Self Organizing Map* – SOM) preservam a topologia durante o seu agrupamento e são avaliados/classificados a partir de uma medida de similaridade.

Neste trabalho a técnica proposta em [13] é modificada 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. O método original considera somente relações lineares entre as variáveis de entrada e a saída [13]. Se estas relações forem não-lineares, os subconjuntos formados por este método podem ser inconsistentes [6]. Neste artigo, quatro tipos de relações entre as variáveis são considerados: linear, exponencial, potência e logarítmico. Para cada variável secundária, quatro modelos são gerados (um para cada tipo de relação) para representar o mapeamento entre as variáveis. O modelo que tiver o maior valor de correlação é o escolhido para representar este mapeamento.

Inicialmente, os subconjuntos de variáveis secundárias são especificados como segue. Seja p o número de variáveis secundárias, calculam-se os quatro coeficientes de correlação entre cada variável secundária e a variável de saída. Para cada variável é escolhido o maior valor de correlação para representar o mapeamento. Em seguida, as p variáveis são ordenadas, de forma crescente, conforme os respectivos valores absolutos de coeficiente de correlação escolhido, criando o conjunto S de variáveis secundárias ordenadas. Em seguida, 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 variável de saída; o segundo subconjunto é composto pelas duas primeiras variáveis em S e a variável de saída, e assim sucessivamente. Então, um mapa de Kohonen ou SOM é construído para cada subconjunto e, a qualidade de cada mapa é calculada através da medida de dissimilaridade entre dois mapas distintos.

O SOM pode ser considerado como uma técnica de extração de característica que realiza um mapeamento de um espaço de entrada de dimensão alta para um espaço de saída de dimensão menor, preservando a topologia do espaço original. O espaço de saída é formado por uma grade regular de unidades de mapeamento (neurônios). Cada unidade do mapa é representada por seu vetor de pesos $w_k = [w_{k1}, \dots, w_{kn}]$, onde n é igual a dimensão do espaço de entrada. Cada unidade do mapa tem uma vizinhança topológica determinada pelo reticulado da grade. O algoritmo de treinamento do SOM utiliza um processo de aprendizado do tipo “o vencedor leva tudo” [14]. A cada passo do treinamento, um vetor $z=[x,y]$ é escolhido aleatoriamente do conjunto de dados de treinamento. Uma medida de similaridade entre este vetor e cada um dos vetores de peso é calculada e escolhe-se o neurônio com menor distância, como melhor unidade do mapa ou unidade *bmu* (*best matching unit*). Neste artigo utilizou-se a distância euclideana como medida de similaridade. De posse do *bmu* e dos neurônios em sua vizinhança topológica, os pesos são atualizados na direção apropriada do espaço de busca [13]. Uma vez calculado todos os mapas, a

escolha do melhor subconjunto de variáveis é feita para cada dois mapas distintos L_q e L_r [15], a partir de uma medida de dissimilaridade definida através das Equações 1 a 3.

$$Mdiss(L_q) = \frac{\sum_{r=1}^{p-1} D(L_q, L_r)}{p-1}, \quad (1)$$

$$D(L_q, L_r) = E \left[\frac{\|d_{L_q}(x) - d_{L_r}(x)\|}{d_{L_q}(x) + d_{L_r}(x)} \right], \quad (2)$$

$$d(x) = \|x(t) - w_{bmu1(x)}(t)\| \sum_{i=p_{bmu1}}^{p_{bmu2}-1} \|w_i(t) - w_{i+1}(t)\| \quad (3)$$

A medida de dissimilaridade informa sobre a relevância de cada combinação de variáveis secundárias em relação a variável de saída, onde E é a esperança matemática, $\|\cdot\|$ é a distância euclideana; p_{bmu1} é a posição da primeira bmu ; 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 [15]. 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 os coeficientes de correlação entre cada variável secundária e a variável de saída para cada um dos modelos: linear, exponencial, potência, e logaritmo;
 - **Passo 2:** Escolher para cada variável secundária o modelo que apresentar o maior coeficiente de correlação.
 - **Passo 3:** 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 4:** Definir o número m de unidades dos mapas auto-organizáveis e montar o conjunto com N_d dados de treinamento $Z=\{[x_i, y_i] \mid x_i = [u_{1,i}, \dots, u_{p,i}]\}$ para $i=1, \dots, N_d$.
 - **Passo 5:** 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 6:** Definir um SOM com m unidades para cada subconjunto c_i e treinar esses SOMs com os dados de treinamento;
 - **Passo 7:** Para cada par de SOMs L_q e L_r , $q \neq r$, calcular a medida de dissimilaridade (Equação 2);
 - **Passo 8:** Calcular a média de dissimilaridade de cada SOM L_q com relação a todos os outros SOMs (equação 1);
 - **Passo 9:** 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.

O algoritmo proposto neste artigo não tem seu uso restrito à metodologia de identificação neurofuzzy aqui desenvolvida. Ele pode ser usado para seleção de variáveis com qualquer outro método de identificação, seja em abordagens clássicas tais que Filtro de Kalman, Mínimos Quadrados [2], Análise de Componentes Principais, Mínimos Quadrados Parciais [16], ou em abordagens baseadas em inteligência computacional como redes neurais, sistemas fuzzy e outros [17,19].

2.2. Seleção da Ordem do Modelo Fuzzy

O método proposto por He e Asada [20] 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 ($n \leq p$) e $y(t)$ é a saída do modelo *fuzzy*, e do vetor de regressores definido pela Equação 4 e Equação 5:

$$\varphi(t) = [y(t-1), \dots, y(t-lag_y), x(t-1), \dots, x(t-lag_x)]^T \quad (4)$$

$$Z^{N_d} = \{\{\varphi(t), y(t), \quad t = 1, \dots, N_d\}\} \quad (5)$$

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 6.

$$q_{i,j} = \frac{\|y(t_i) - y(t_j)\|}{\|\varphi(t_i) - \varphi(t_j)\|}, \quad i \neq j \quad (6)$$

Convém ressaltar que a inserção de regressores supérfluos (ordem superestimada) não tem um impacto significativo nos valores dos coeficientes de Lipschitz, no entanto para um modelo *fuzzy* este aumento do conjunto de regressores implica no aumento da base de regras e consequentemente na complexidade do modelo. O procedimento completo de seleção da ordem do modelo *fuzzy* é apresentado no quadro 2.

-
- **Passo 1:** Para uma dada escolha de atraso $a = lag_y = lag_x$, 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:

$$\bar{q}^{(a)} = \left(\prod_{i=1}^r \sqrt{n} q^a(t) \right)^{\frac{1}{r}},$$

onde $q^{(a)}$ é o quociente referente ao atraso a .

- **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

2.3. Partição dos espaços de entrada/saída

Uma vez escolhida a estrutura do modelo, isto é, as variáveis de entrada e os regressores a elas associados, o problema de identificação de sistemas dinâmicos com modelos *fuzzy* é transformado em um problema de regressão não linear estática $y = F(x)$ em que a função desconhecida $F(x)$ é uma função suave a qual define uma hipersuperfície. O modelo *fuzzy* a ser identificado deve aproximar esta hipersuperfície o melhor possível (acurácia). Para isto é necessário, decompor o domínio das variáveis de entrada e saída em regiões ou partições *fuzzy* do universo de discurso, isto é, escolher o número e a forma das funções de pertinência por variável. Em seguida é preciso determinar as relações entre as diferentes partições do universo de discurso, isto é, determinar a base de regras do modelo *fuzzy*. As abordagens baseadas em agrupamento *fuzzy* fornecem ferramentas para determinar apropriadamente estes parâmetros, conforme será apresentado a seguir.

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*. Formalmente [21]:

Definição 1: Partição C-Fuzzy. Seja um conjunto de dados $X = \{x_1, \dots, x_{N_d}\}$, onde $x_k \in \mathcal{R}^n$, V_c um conjunto de matrizes $H = [h_{ik}] \in \mathcal{R}^{c \times n}$, e c o número de agrupamentos a serem formados ($2 \leq c \leq n$). O espaço da partição C-fuzzy de X é o conjunto:

$$M_{fc} = \left\{ H \in V_c \mid h_{ik} \in [0,1], 1 \leq i \leq c, 1 \leq k \leq N_d, e \sum_{i=1}^c h_{ik} = 1, \forall k \in \{1, 2, \dots, N_d\} \right\} \quad (7)$$

onde h_{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 $M_{fc} H = [h_{ik}] \in M_{fc}$, tal que a função objetivo $J_m(H, V)$, dada pela equação 8 seja minimizada.

$$J_m(U, V) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m \|x_k - v_i\|^2 \quad (8)$$

Onde $m \in (1; \infty)$ é uma constante de ponderação que determina a sobreposição dos grupos, $V = (v_1, \dots, v_c)$ com $v_i \in \mathcal{R}^n$, e v_i é o centro do cluster i . A otimização aproximada de J_m realizada pelo algoritmo *FCM* é um procedimento iterativo como descrito no quadro 3 [21].

-
- **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 de V , com o novo centro de cada grupo C^i ;

$$v_i^{(q)} = \frac{\sum_{k=1}^{N_d} (h_{ik}^{(q)})^m x_k}{\sum_{k=1}^{N_d} (h_{ik}^{(q)})^m}$$

- **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 $\|H^{(q+1)} - H^{(q)}\| < \varepsilon$, então pare o processo; caso contrário, incremente q , e volte ao passo 2.
-

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

2.4. 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 [22].

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. Esses pontos discretos devem ser aproximados no domínio de cada variável por uma função de pertinência paramétrica contínua conhecida. A Figura 1 ilustra a projeção dos grupos nos domínios das variáveis x e y e a base de regras gerada por esta projeção.

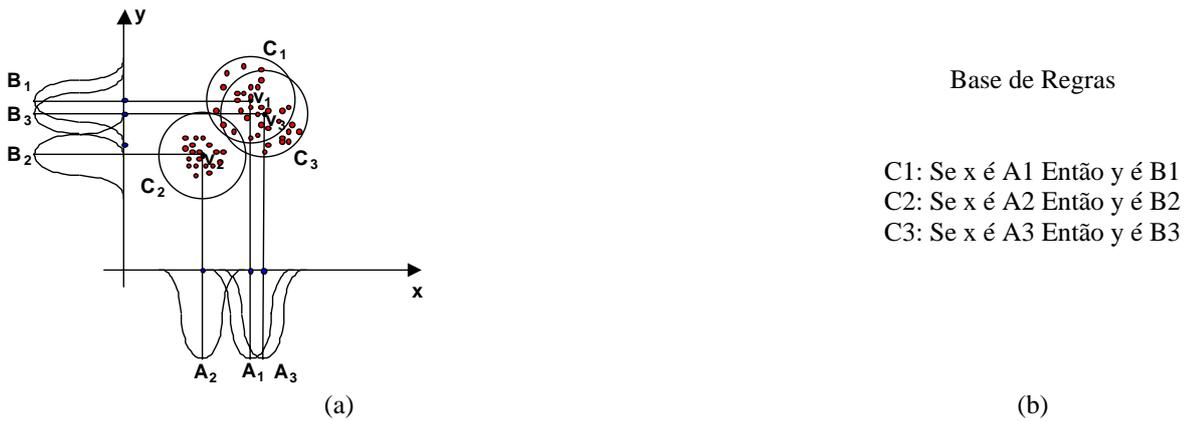


Figura 1: Projeção dos grupos nos espaços de entrada e saída (a) e base de regras associada (b).

Para a aproximação destas funções, foi utilizada neste trabalho a função de pertinência *gbell* (*generalized bell*), dada pela equação 9, devido à sua notação concisa e à suavidade de sua curva. A vantagem do uso de funções *gbell* sobre as funções gaussianas está no parâmetro σ , que fornece um grau adicional de liberdade para ajuste dos pontos de cruzamento entre as funções de pertinência que descrevem o universo de discurso [23]. Além disso, esse parâmetro permite a representação de regiões de platô, ou seja, regiões onde mais de um elemento possui grau de pertinência unitário. Esta flexibilidade não é possível quando se usa funções gaussianas [6].

$$f(x, \sigma, \gamma, \omega) = \left[1 + \left| \frac{x - \omega}{\sigma} \right|^{2\gamma} \right]^{-1} \quad (9)$$

onde o parâmetro ω é o centro da função *gbell*, $(\omega - \sigma)$ e $(\omega + \sigma)$ são os pontos de cruzamento, cujo grau de pertinência é igual a 0,5 e o parâmetro γ controla a inclinação destes pontos de cruzamento. O quadro 4 apresenta os passos do procedimento de geração da base de regras inicial.

-
- **Passo 1:** Estipular o número de grupos *ncr* (tamanho da base de regras inicial);
 - **Passo 2:** Aplicar o algoritmo FCM (quadro 3) 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 do tipo *gbell* que descrevem o domínio de cada variável.
-

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

2.5. Simplificação da Base de Regras Inicial

A interpretabilidade lingüística de um modelo *fuzzy* é um dos aspectos que distinguem a modelagem *fuzzy* de métodos de modelagem tipo caixa-preta. Porém essa interpretabilidade não é alcançada automaticamente. Quando um modelo *fuzzy* é desenvolvido utilizando-se 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. A simplificação da base de regras inicial é assim uma fase importante durante o processo de identificação de modelos *fuzzy* inferenciais, pois o compromisso entre a precisão e a generalização do modelo é considerado durante esta fase.

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 A1 e A3 na Figura 1), fazendo com que o modelo utilize mais conjuntos *fuzzy* do que o necessário [6].

Neste trabalho a similaridade entre conjuntos *fuzzy* é medida através da proximidade entre os centros das funções *gbell* que os representam. Para verificar a existência de conjuntos *fuzzy* similares os parâmetros das funções de pertinência são

verificados da seguinte forma. Considerando-se o domínio de cada variável j , duas funções *gbell*, cujos centros são ω_{fj} e ω_{gj} são consideradas similares se:

$$|\omega_{fj} - \omega_{gj}| \leq \zeta_{\omega} \quad (10)$$

Onde $|\cdot|$ é o valor absoluto e ζ_{ω} é a diferença mínima entre os centros e deve ser definida a priori. Se a desigualdade for satisfeita, então as respectivas funções de pertinência são substituídas por uma nova função de pertinência *gbell* cujo centro é a média aritmética dos centros das funções originais e novo valor de σ é calculado considerando os maiores valores de pontos de cruzamento à esquerda e à direita. Após este procedimento, as regras são atualizadas com as novas funções de pertinência. De um modo indireto, o parâmetro ζ_{ω} controla o número de funções de pertinência das variáveis do modelo *fuzzy*. Assim é possível aumentar a precisão do modelo ao diminuir o valor de ζ_{ω} . Por outro lado, a diminuição de ζ_{ω} provoca o aumento do número de funções de pertinência das variáveis, aumentando em consequência o tamanho da base de regras. Em resumo, o compromisso entre precisão/accurácia e interpretabilidade depende da escolha do valor do parâmetro ζ_{ω} .

O quadro 5 (passos 1 e 2) sumariza o procedimento de simplificação da base de regras inicial. Este procedimento de simplificação agrupa as funções de pertinência que descrevem o mesmo local no domínio de uma determinada variável. 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 do modelo, uma das características essenciais dos sistemas *fuzzy* [24].

2.6. Tratamento de Regras Inconsistentes

Considerando-se que o número de regras geradas pelo algoritmo FCM é 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 utilizada 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 denotada pela Equação 11 [6], 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.

$$CF(R_i) = \frac{|A \wedge C| - 1/2}{|A|} \quad (11)$$

onde $|\cdot|$ denota cardinalidade e R_i é a i -ésima regra inconsistente. Os passos 3 e 4 do quadro 5 representam esse tratamento de regras inconsistentes e finaliza o procedimento de geração de base de regras.

-
- **Passo 1:** Identificar as funções de pertinência similares no domínio de cada variável;
 - **Passo 2:** Substituir os pares de funções similares por uma nova função de pertinência *gbell*;
 - **Passo 3:** Para as regras com os mesmos antecedentes, formar os grupos de regras inconsistentes;
 - **Passo 4:** Para cada grupo, calcular o *CF* das regras e reter na base de regras apenas aquela de maior *CF*.
-

Quadro 5. Algoritmo de simplificação da base de regras.

Segundo Babuska e colaboradores [25] grande parte dos métodos de simplificação de base de regras existentes na literatura é baseada em medidas de similaridade para eliminar redundâncias na descrição do universo de discursos das variáveis. A principal desvantagem dessa abordagem é que para verificar a similaridade entre duas funções de pertinência é necessário calcular o grau de pertinência de todos os pontos do universo de discurso em relação a essas funções de pertinência. Uma vez que duas funções são consideradas similares elas são substituídas por uma nova função capaz de representá-las adequadamente. Então o procedimento se repete até que não haja mais similaridade entre funções na base de regras. A carga computacional e a lentidão desse procedimento aumentam de acordo com número de variáveis do modelo *fuzzy* e também com a quantidade de funções de pertinência que descreve o domínio de cada variável. Diante desta dificuldade, o algoritmo apresentado no quadro 5 é uma alternativa para verificar similaridade entre funções de pertinência de forma simplificada, realizada diretamente através da comparação dos parâmetros das funções de pertinência em questão. Os procedimentos

propostos para a simplificação da base de regras e para o tratamento de regras inconsistentes do quadro 5 são gerais e podem ser aplicados a qualquer modelo *fuzzy* lingüístico.

2.7. Validação do modelo final

A etapa de validação determina se o modelo identificado é bom o suficiente de acordo com um determinado critério de qualidade estabelecido a priori. Para modelos construídos a partir de dados, independente da metodologia utilizada, a sua validação é sempre feita com um conjunto de dados diferentes (dados de validação) do conjunto de dados utilizados para a obtenção do modelo (dados de treinamento).

Neste trabalho, a validação do modelo *fuzzy* é realizada através da análise da estrutura do modelo gerado e do desempenho alcançado durante a aplicação dos dados de validação. O compromisso entre a precisão do modelo, na geração das estimativas, e a sua capacidade de generalização é fator determinante na escolha do modelo final. Além disso, é necessário manter a interpretabilidade do modelo *fuzzy*, pois se trata de uma das principais vantagens e características deste tipo de modelo. Para isto, quanto menor o número de funções de pertinência no universo de discurso de cada variável e / ou de regras descrevendo o comportamento do processo, maior é a interpretabilidade do modelo. Por outro lado, a precisão das estimativas pode ser comprometida pelo número reduzido de regras e funções de pertinência.

Outro aspecto que deve ser avaliado na etapa de validação é a propriedade de completude da base de dados e da base de regras gerada. Essa propriedade refere-se à capacidade do modelo gerado inferir para qualquer entrada, uma saída apropriada. Isto significa que o modelo é capaz de representar o sistema para todos os estados possíveis. Para a base de dados, esta propriedade determina que a união dos conjuntos suporte de todos os predicados de uma variável deve cobrir todo o universo de discurso desta variável. Para a base de regras, a propriedade garante que sempre existirá ao menos uma regra que será disparada para qualquer entrada possível [26].

Para avaliar a capacidade de generalização do modelo *fuzzy* foram empregados dois índices de desempenho: o erro quadrático médio (*mse*: *mean-square error*) e a média do erro absoluto em percentual (*mape*: *mean absolute percentage error*) os quais são dados respectivamente pelas equações (13) e (14) [6]:

$$mse = \frac{1}{N} \sum_{k=1}^N (y(k) - yf(k))^2 \quad (13)$$

$$mape = \frac{1}{N} \sum_{k=1}^N \left| \frac{y(k) - yf(k)}{y(k)} \right| \quad (14)$$

Onde y é a saída medida, yf é a saída inferida pelo modelo *fuzzy*, e N é o número de amostras do conjunto de treinamento e/ou teste. Os valores destes índices de desempenho devem ser inferiores a limites pré-estabelecidos (ξ_{mse} e ξ_{mape}) para aceitação do modelo.

3. Sumário do Procedimento de Identificação

Na metodologia proposta neste artigo, modelos *fuzzy* são identificados automaticamente a partir de um conjunto de dados amostrados do sistema sem o conhecimento de especialistas. Alguns aspectos referentes à estrutura do modelo são estabelecidos a priori. O uso de modelos lingüísticos e funções de pertinência do tipo *gbell* foram escolhidos a fim de aumentar a interpretabilidade do modelo conforme discutido acima. Para o módulo de inferência *fuzzy*, foram escolhidos o operador de agregação de antecedentes t-norma “mínimo”, operador de agregação de regras s-norma “máximo”, o operador de implicação t-norma “mínimo”, e o método de defuzificação por bissetor da área. Estes operadores foram selecionados de forma empírica conforme o desempenho por eles alcançado.

A metodologia de identificação de modelos *fuzzy* inferenciais proposta neste artigo pode ser resumida através dos seguintes passos:

- **Passo 1:** Selecionar o conjunto de entrada do modelo *fuzzy* a partir do conjunto de variáveis secundárias disponíveis através do algoritmo descrito no quadro 1;
- **Passo 2:** Selecionar a estrutura do modelo como descreve o algoritmo do quadro 2;
- **Passo 3:** Estabelecer os conjuntos de treinamento e teste, o tamanho da base de regras inicial *ncr* (número de agrupamentos), o parâmetro de sobreposição dos agrupamentos “*m*” do algoritmo FCM, e os limites de erro aceitável para os índices de desempenho ξ_{mse} e ξ_{mape} ;
- **Passo 4:** Gerar a base de regras inicial conforme descreve o quadro 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 (quadro 5);

- **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 (quadro5);
- **Passo 7:** Checar a interpretabilidade e a completude da base de regras.
- **Passo 8:** Calcular os índice de desempenho aplicando o modelo gerado no conjunto de teste. Se os índices de desempenho forem superiores aos seus limites ($mse > \xi_{mse}$ e $mape > \xi_{mape}$) então retornar ao passo 4, caso contrário finalizar o procedimento de identificação.

O procedimento acima descrito foi encapsulado em um programa, existindo duas versões, uma em MATLAB e outra em Builder C++. As informações inseridas no passo 3, tamanho da base de regras, sobreposição dos agrupamentos e limites para o erro são os únicos parâmetros a serem especificados pelo usuário. Na etapa de validação (passos 7 e 8) o usuário pode aceitar ou não o modelo identificado, retornando a este passo (passo 3) para refazer suas escolhas.

4. Resultados

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 (figura 2) durante a sua fase de inicialização. Os dados para implementação da coluna foram obtidos de [27] onde esta coluna é modelada com o auxílio da ferramenta computacional HYSYS^{®1}.

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 [27]. 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, ou seja, a coluna entrou em regime de operação. 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% [27]. 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.

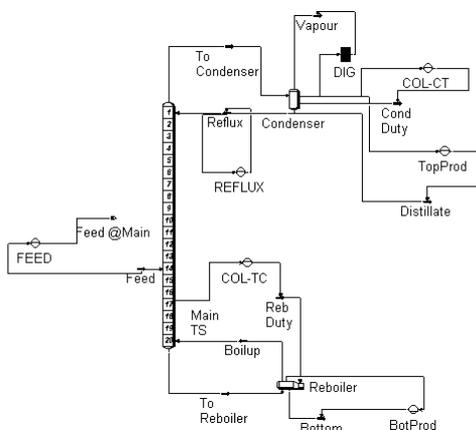


Figura 2: Coluna de destilação binária.

Um sensor virtual foi então desenvolvido para inferir a composição do produto de topo (metanol) numa faixa de operação entre 90% e 99,9% de pureza. Para obtenção do sensor virtual, foi gerado um conjunto de dados com 550 amostras do tipo $z = (u_1 \dots u_{12}, y)$ correspondendo aos valores das variáveis secundárias e da saída descritas na tabela 1.

Tabela 1. Descrição das variáveis secundárias.

Variável	Nome da variável	Descrição	Valor desejado em regime estacionário [27]	Coefficiente de correlação	Posição
u_1	feed1_pv	vazão molar de alimentação	236kmol/h	0.8524	4°.
u_2	reflux_pv	refluxo do condensador para a coluna	247kmol/h	0.5265	9°.
u_3	pv_press	pressão no topo da coluna	101,3kPa	0.1987	11°.
u_4	bottom_pv	nível do refervedor	50%	0.1948	12°.
u_5	top_pv	nível do condensador	20%	0.9909	1°.
u_6	tray9	temperatura do prato 9	54°C	0.4062	10°.

¹ HYSYS 3.0, HYPROTECH Ltd.

u_7	tray13	temperatura do prato 13	41°C	0.7924	6°.
u_8	tray16	temperatura do prato 16	3°C	0.8945	2°.
u_9	tray18	temperatura do prato 18	3°C	0.8933	3°.
u_{10}	cond_flow	vazão para o condensador	315kmol/h	0.8319	5°.
u_{11}	top_flow	vazão do produto de topo (destilado)	85kmol/h	0.6973	8°.
u_{12}	bottom_flow	vazão do produto de fundo (resíduo)	151 kmol/h	0.7743	7°.
y	top_comp	Composição do produto de topo	99,9%	-	Saída

Na etapa de seleção das variáveis de entrada, inicialmente são calculados quatro coeficientes de correlação para cada variável secundária do conjunto de dados, como mostrado na figura 3. O modelo com maior valor de coeficiente de correlação (tabela 1) é usado para representar o mapeamento entre a variável e saída. De acordo com o valor do seu coeficiente, as variáveis secundárias são ordenadas como mostrado na última coluna da tabela 1 e são construídos os subconjuntos de variáveis de entrada, mostrados na tabela 2.

Tabela 2: Subconjunto de variáveis

Subconjunto	Variáveis selecionadas
L1	(u_5)
L2	(u_5, u_8)
L3	($u_5, u_8, u_9,$)
L4	(u_5, u_8, u_9, u_1)
L5	($u_5, u_8, u_9, u_1, u_{10}$)
L6	($u_5, u_8, u_9, u_1, u_{10}, u_7$)
L7	($u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}$)
L8	($u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}, u_{11}$)
L9	($u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}, u_{11}, u_2$)
L10	($u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}, u_{11}, u_2, u_6$)
L11	($u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}, u_{11}, u_2, u_6, u_3$)
L12	($u_5, u_8, u_9, u_1, u_{10}, u_7, u_{12}, u_{11}, u_2, u_6, u_3, u_4$)

Em seguida a medida de dissimilaridade de cada subconjunto é calculada gerando o gráfico da figura 4. Da análise desse gráfico, o conjunto com a menor dissimilaridade é o L6. Na etapa de seleção da ordem do modelo *fuzzy*, os coeficientes de Lipschitz são calculados de acordo com o algoritmo do quadro 2. Este procedimento indica que um atraso unitário é suficiente para representar o sistema. Assim a variável de saída $y(t-1)$ é adicionada ao conjunto de variáveis de entrada que passa a ter a seguinte composição $x(k) = [u_5(k-1), u_8(k-1), u_9(k-1), u_1(k-1), u_{10}(k-1), u_7(k-1), y(k-1)]$. A partir do conjunto Z com 550 amostras foram extraídos os pares ordenados $(x(k), y(k))$ criando um novo conjunto com 549 pares. Este conjunto foi normalizado no intervalo [0, 1] e em seguida dividido em duas partes para compor o conjunto de dados de treinamento do sensor e um conjunto de validação (calibração).

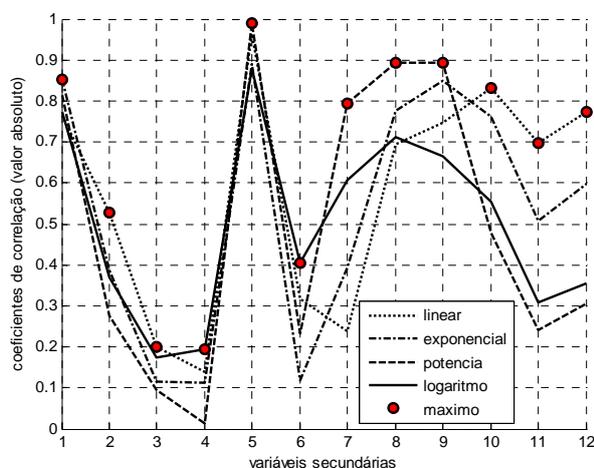


Figura 3: Coeficientes de correlação para cada modelo

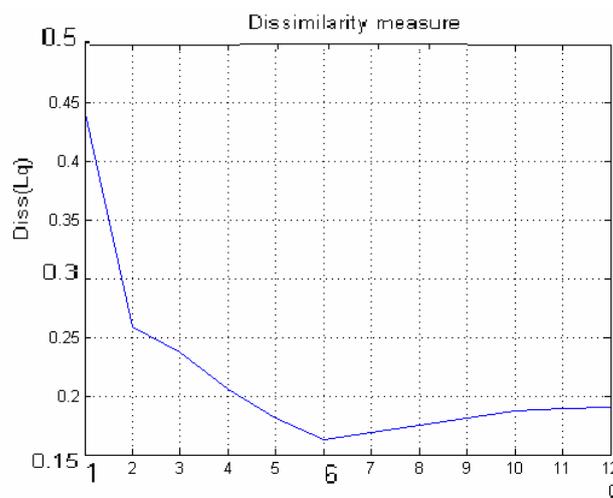


Figura 4: Medida de dissimilaridade

Na etapa de partição do espaço de entrada e saída foi definida que o número inicial de grupos (clusters), o qual corresponde ao tamanho da base de regras inicial, é equivalente a quarta parte do número de amostras no conjunto de

treinamento, isto é, 68 grupos ou regras. De acordo com a figura 1, as funções de pertinência para cada variável são obtidas através da projeção de cada grupo no universo de discurso de cada variável, assim cada variável possui 68 funções de pertinência. A base de regras inicial é então simplificada através do algoritmo dado no quadro 5, reduzindo o número de funções de pertinência conforme mostrado na tabela 3. Essas partições antes (linha tracejada) e depois da simplificação (linha contínua) são mostradas nas figuras 5 a 12 para cada um das variáveis *fuzzy* utilizadas no modelo final. O procedimento de eliminação das regras inconsistentes é também aplicado gerando uma base com apenas 42 regras.

Tabela 3. Número de funções de pertinência por variável no modelo final.

#	Variável	Funções de pertinência
x_1	Nível de líquido no condensador (u_5)	14
x_2	Temperatura do prato 18 (u_9)	11
x_3	Temperatura do prato 16 (u_8)	10
x_4	Vazão molar de alimentação (u_1)	9
x_5	Vazão para o condensador (u_{10})	10
x_6	Temperatura no prato 13 (u_7)	11
x_7	Composição do metanol – realimentação (y_1)	13
y_1	Composição do metanol - saída	13

A tabela 4 resume as informações relevantes sobre o modelo *fuzzy* inferencial obtido, tais como o tamanho da base de regras e o número de variáveis de modelo.

Tabela 4. Parâmetros do modelo *fuzzy* inferencial.

descrição	valor
tamanho da base de regras inicial (ncr)	68
tamanho da base de regras simplificada	42
Número de regras inconsistentes encontradas	0
número de variáveis de entrada	7
número de variáveis de saída	1
atraso selecionado	1

Para validar o sensor virtual construído a partir do modelo *fuzzy* da tabela 4, foram empregados os dois índices de desempenho citados anteriormente: o erro quadrático médio (*mse*) e a média do erro absoluto em percentual (*mape*). Os resultados obtidos com os dados de treinamento e de teste são mostrados na Tabela 5.

Tabela 5. Índices de desempenho do sensor *neurofuzzy*

	treinamento	teste
<i>mse</i>	0.0078	0.0096
<i>mape</i> (%)	0.0714	0.0787

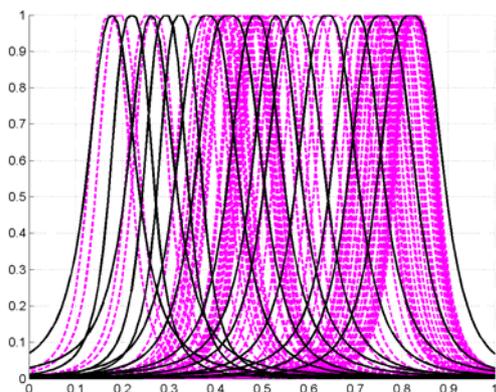


Figura 5. Funções de pertinência para a variável Nível de líquido no condensador (x_1)

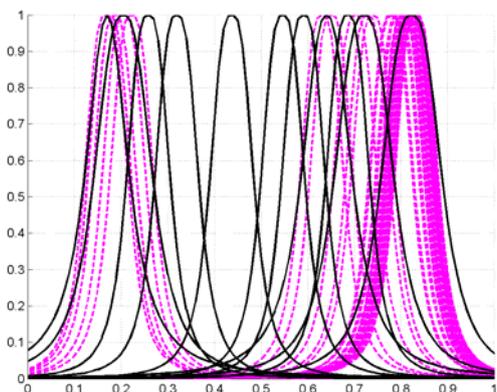


Figura 6. Funções de pertinência para a variável Temperatura do prato 18 (x_2)

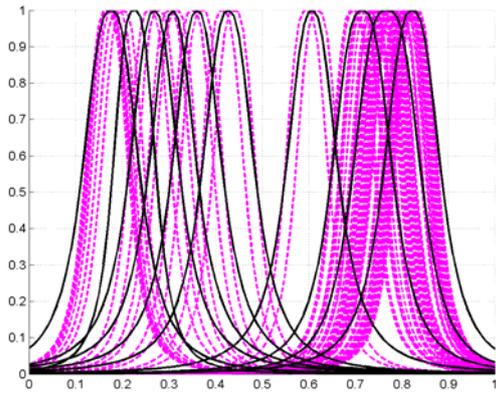


Figura 7. Funções de pertinência para a variável Temperatura do prato 16 (x_3)

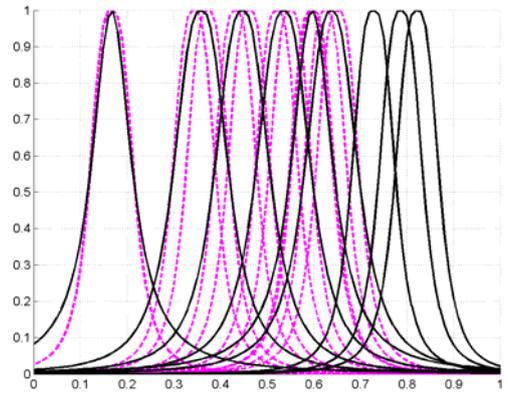


Figura 8. Funções de pertinência para a variável Vazão molar de alimentação (x_4)

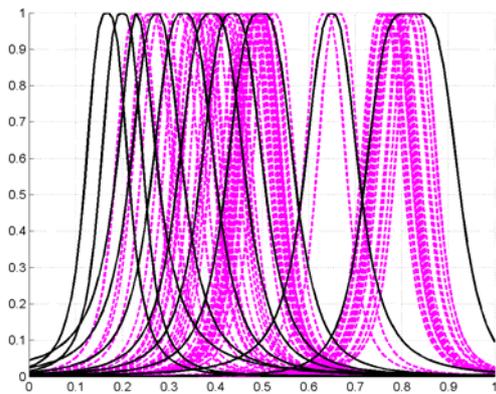


Figura 9. Funções de pertinência para a variável Vazão para o condensador (x_5)

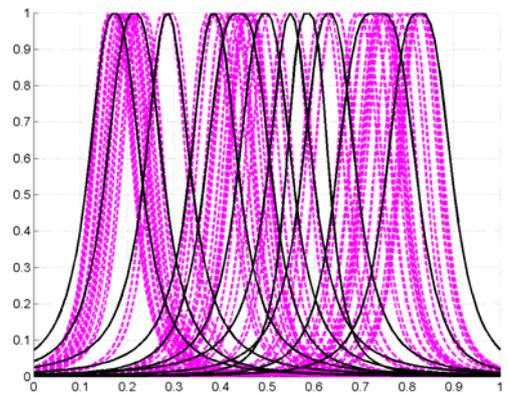


Figura 10. Funções de pertinência para a variável Temperatura no prato 13 (x_6)

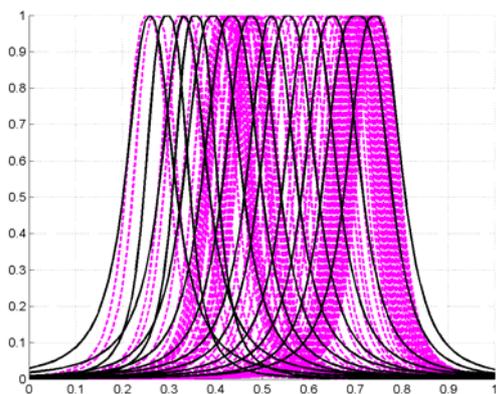


Figura 11. Funções de pertinência para a variável Composição do metanol - realimentação (x_7)

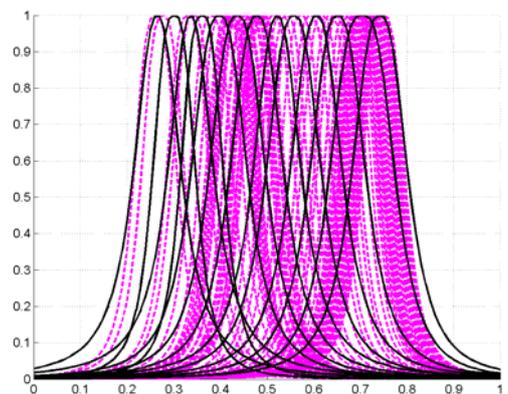


Figura 12. Funções de pertinência para a variável Composição do metanol (y_1)

A fim de validar o sensor virtual desenvolvido comparou-se também o desempenho do modelo inferencial *fuzzy* com o de modelos obtidos através de regressão nas componentes principais (*PCR – Principal Component Regression*) e redes neurais. Dois modelos baseados em PCR foram obtidos: o PCR6 o qual utiliza as seis primeiras componentes principais (*PCA – principal Components Analysis*) e o PCR12 que utiliza todas as componentes principais do sistema. Estes métodos foram

implementados como descritos em [16]. O modelo neural foi obtido com uma rede direta sem conexões recorrentes (*MANN – Moving Average Neural Network*) conforme recomendado em [1]. Esta rede foi treinada pelo método de Levenberg-Marquadt [28], com 12 neurônios na camada de entrada, 8 neurônios na camada intermediária e 1 neurônio de saída. Estes valores foram escolhidos por tentativa e erro. Foram realizadas 20 simulações com diferentes inicializações e foram selecionados os valores de pesos que resultaram em menor erro quadrático médio (*mse*).

A tabela 6 traz uma comparação de desempenho entre estes modelos para o conjunto de validação. A figura 13 apresenta a composição do produto de topo inferida pelos quatro modelos (linha pontilhada) e a composição de saída medida na coluna (linha contínua) no final da fase de *start-up*, considerado o conjunto completo de dados (550 amostras).

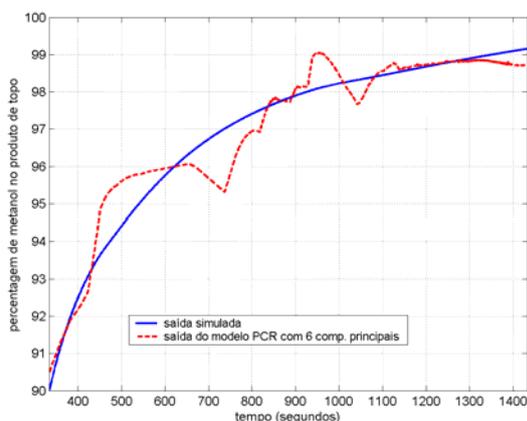
Tabela 6. Índices de desempenho para os vários sensores

Método	PCR6	PCR12	MANN	Modelo <i>fuzzy</i>
<i>mse</i>	0.3443	0.2374	0.0010	0.0096
<i>mape</i>	0.4323%	0.3710%	0.0215%	0.0787%

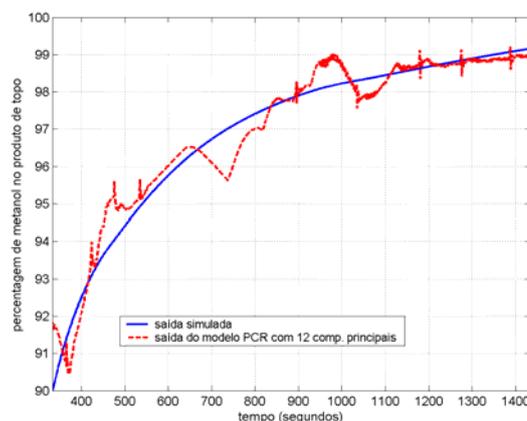
Os dados da tabela 6 e os gráficos da figura 13 revelam que os modelos obtidos com regressão por componentes principais (PCR6 e PCR12) mostram desempenho inferior aos obtidos com os modelos *fuzzy* e neural. Quanto ao modelo neural, este apresenta melhor desempenho do que o modelo *fuzzy* proposto, no entanto a obtenção deste modelo demandou um trabalho empírico para determinação da arquitetura da rede, além da realização de vários experimentos para eliminar os efeitos aleatórios de inicialização da rede.

Uma das vantagens do modelo *fuzzy* sobre o modelo neural é a interpretabilidade. Por exemplo, as regras do modelo obtido podem ser escritas na forma: **Se** o nível do condensador está próximo do seu valor de regime **e** a temperatura do prato 18 está abaixo do valor de regime **e** ... **então** a composição do produto de topo está abaixo do valor de regime. Com isto o engenheiro de processos pode ter uma avaliação qualitativa do estado atual do processo.

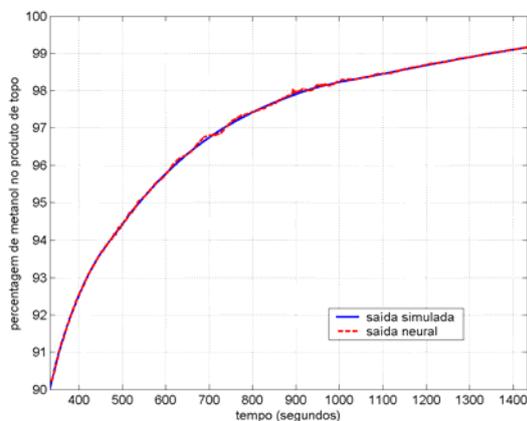
Através da figura 13 observa-se ainda que a curva da saída inferida pelo modelo *fuzzy* reproduz a tendência da curva simulada, porém é possível notar que o modelo *fuzzy* apresenta deficiências em inferir as saídas próximas aos limites do universo de discurso (neste caso valores próximos de 90% e 99% de concentração de metanol) da variável de saída. Nestes limites obteve-se aproximadamente 0,20% (limite inferior) e 0,25% (limite superior) de erro percentual absoluto. 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. Este fato é possível de ser visualizado na figura 12, em que nota-se a ausência de funções de pertinência nestes limites.



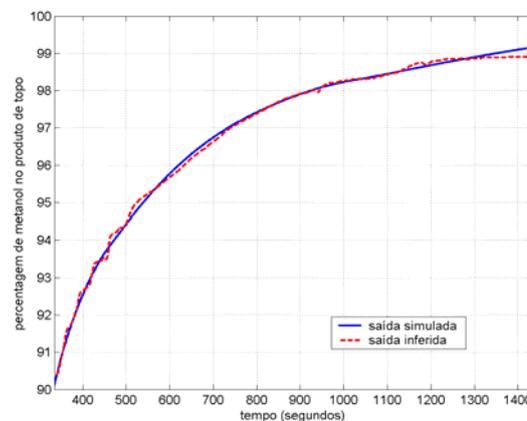
a) Resposta do modelo PCR6



b) Resposta do modelo PCR12



c) Resposta do modelo neural



d) Resposta do modelo fuzzy

Figura 13. Inferência da composição do produto de topo y_1 para os vários sensores.

5. Conclusões

Este trabalho propõe uma metodologia *neurofuzzy* para a identificação de modelos inferenciais a serem utilizados em sensores virtuais. Esta metodologia é desenvolvida em duas etapas. Na primeira etapa, as variáveis de entrada do modelo são selecionadas aplicando-se análise de correlação, mapas de Kohonen e medidas de similaridade. Também os quocientes de Lipschitz são utilizados para selecionar a ordem do modelo. Na segunda etapa, de identificação dos parâmetros do modelo, o algoritmo de agrupamento *fuzzy c-means* é aplicado no espaço de entrada e saída de modo a encontrar o modelo *fuzzy* inicial e as partições do universo de discurso de cada variável do modelo. As funções de pertinência similares são agrupadas, eliminando a redundância na descrição do domínio de cada variável do modelo. Por fim o modelo final é validado a partir de análise de completude, interpretabilidade e precisão/accurácia alcançados.

De forma especial, o algoritmo para seleção de variáveis baseada em modelos não-lineares de correlação e mapas de Kohonen é uma proposta original desta metodologia, que pode ser utilizada com qualquer outro método de identificação. Também o algoritmo de simplificação da base de regras é uma proposta original e pode ser utilizada em qualquer modelo *fuzzy* para verificar similaridade entre regras.

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 e seu desempenho pode ser melhorado com o uso de técnicas de refinamento do modelo *fuzzy* inferencial. Entre estas técnicas citam-se métodos para a sintonia dos parâmetros das funções de pertinência e/ou métodos para a avaliação da relevância de cada regra do modelo de modo a reduzir o tamanho da base de regras. Uma das desvantagens da metodologia proposta é que a modelagem é baseada em algoritmos de agrupamento os quais são dependentes da qualidade e quantidade dos dados utilizados. Em especial o algoritmo *fuzzy c-means* tem dificuldades em modelar os dados próximos aos limites do universo de discurso.

Para finalizar, pode-se concluir que a metodologia de identificação *neurofuzzy* apresentada neste artigo fornece uma ferramenta eficiente e flexível para o projeto de sensores virtuais, enquanto mantém o compromisso entre complexidade do modelo identificado e dificuldade de projeto dentro de limites aceitáveis.

Agradecimentos

Este trabalho contou 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 para o Setor Petróleo e Gás – PRH-ANP/MCT (PRH10-UTFPR) e do CNPq (processo nº 303517/2004-2).

Referências Bibliográficas

- [1] L. Fortuna, S. Granziani, A. Rizzo and M. G. Xibilia: Soft sensors for monitoring and control of industrial processes, Springer - Verlag, London, 2007.
- [2] L. Ljung: System identification: theory for the user, 2nd edn, Prentice-Hall, Upper Saddle River, NJ, 2000.

- [3] E. Zambroga, M. Barolo and D. E. Seborg: "Optimal selection of soft sensor inputs for batch distillation columns using Principal Component Analysis", *Journal of Process Control*, vol. 15, pp. 39-52, 2005.
- [4] M. T. Tham, G. A. Montague, A. J. Morris and P. A. Lant: "Soft-sensors for process estimation and inferential control", *Journal of Process Control*, vol. 01, pp. 3-14, 1991.
- [5] Y. Dote and S. J. Ovaska: "Industrial applications of soft computing: a review", *Proceedings of IEEE*, vol. 89, pp 1243-1265, 2001.
- [6] E. Nagai: "Identificação Automática de Modelos *Fuzzy* Inferenciais", Tese de Doutorado, Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná, Curitiba-Pr-Brasil, 2006.
- [7] J. X. Luo and H. H. Shao: "Developing soft sensors using hybrid soft computing methodology: a neurofuzzy system based on rough set theory and genetic algorithms", *Soft Computing*, vol. 10, pp. 54-60, 2006.
- [8] T. Soderström and P. Stoica: *System identification*, 1st edn, Prentice Hall, Upper Saddle River, NJ, 1989.
- [9] L. A. Zadeh: "Fuzzy sets", *Information and Control*, vol. 8, pp. 338-352, 1965.
- [10] M. R. Delgado, F. Von Zuben and F. Gomide: "Hierarchical genetic fuzzy systems: accuracy, interpretability and design autonomy", in: *Interpretability Issues in Fuzzy Modeling - Studies in Fuzziness and Soft Computing*, Physica-Verlag, Berlin, pp. 379-405, 2003.
- [11] J. Espinosa and J. Vandewalle: "Constructing *fuzzy* models with linguistic integrity from numerical data-AFRELI algorithm", *IEEE Transactions on Fuzzy Systems*, vol. 8, pp. 591-600, 2000.
- [12] Y. Jin: "Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement", *IEEE Transactions on Fuzzy Systems*, vol. 8, pp. 212-221, 2000.
- [13] R. Rallo, J. Ferre-Gine, A. Arenas and F. Giral: "Neural virtual sensor for the inferential prediction of product quality from process variables", *Computers and Chemical Engineering*, vol. 26, P. 1735-1754, 2002.
- [14] S. Haykin: *Neural Networks*, New Jersey: Prentice-Hall, 1999
- [15] S. Kaski and K. Lagus: "Comparing self-organizing maps", in *Proceedings of ICANN*, pp 809 - 814, 1996.
- [16] P. Geladi and B. R. Kowalski: "Partial least-squares regression: a tutorial", *Analytica Chimica Acta*, vol. 185, pp 1 - 17, 1986.
- [17] L. S. Coelho and V. C. Mariani: "Identificação de sistemas não-lineares usando uma abordagem de sistema nebuloso com otimização evolutiva", *Learning and Nonlinear Models*, vol. 3, no. 1, pp. 27-39, 2005.
- [18] J. A. Ruiz Vargas and E.M. Hemerly: "Convergência do erro de predição em identificação neural sujeita a distúrbios e parâmetros variantes no tempo", *Learning and Nonlinear Models*, vol. 3, no. 2, pp. 84-97, 2005.
- [19] L. G. M. Souza and G. A. Barreto: "Nonlinear system identification using local ARX models based on the self-organizing map", *Learning and Nonlinear Models*, vol. 4, no. 2, pp. 112-123, 2006.
- [20] X. He and H. Asada: "A new method for identifying orders of input-output models for nonlinear dynamical systems", In *Proceedings of the American Control Conference*, pp. 2520-2523, 1993.
- [21] J. C. Bezdek: *Pattern recognition with fuzzy objective function algorithms*, Plenum Press, New York, 1987.
- [22] M. Setnes: "Supervised fuzzy clustering for rule extraction", *IEEE Transaction on Fuzzy Systems*, vol. 8, n° 04, pp 416-424, 2000.
- [23] J. S. R. Jang, C. T. Sun and E. Mizutani: *Neuro-fuzzy and soft computing*, Prentice Hall, Upper Saddle River, NJ, 1997.
- [24] S. Guillaume: "Designing fuzzy inference systems from data: an interpretability-oriented review", *IEEE Transactions on Fuzzy Systems*, vol. 09, n° 03, pp. 426-443, 2001.
- [25] R. Babuska, U. Kaymak, H. R. van Nauta and M. Setnes: "Similarity measures in fuzzy rule base simplification, *IEEE Trans. on Man, Systems and Cybernetics - Part B*, vol. 28, n° 03, pp. 376 - 386, 1998.
- [26] C. C. Lee: "Fuzzy logic in control systems: Fuzzy logic controller - part I and II", *IEEE Trans. on Man, Systems and Cybernetics - Part B*, vol. 20, n° 02, pp. 404 - 418, 1990.
- [27] J. A. Fabro, L. V. R. Arruda and F. Neves Jr: "Startup of a distillation column using intelligent control techniques", *Computers and Chemical Engineering*, vol. 30, pp 309 - 320, 2005.
- [28] R. Fletcher: *Practical Methods of Optimization*, Wiley, New York, 1987.