



COBEQ

XIII Congresso Brasileiro de Engenharia Química
XIX Interamerican Congress of Chemical Engineering

Brazilian Congress of Phase Equilibrium and Fluid Properties for Chemical Process Design

COMPARAÇÃO SISTEMÁTICA ENTRE REDES DE MODELOS LOCAIS E REDES NEURAIAS

M. S. Posser¹, J. O. Trierweiler^{2*} and A. R. Secchi³

Departamento de Engenharia Química, Universidade Federal do Rio Grande do Sul,

Rua Marechal Floriano, 501/81, CEP 90020-061 - Porto Alegre (RS), Brasil

e-mail: {posser¹, jorge², arge³}@enq.ufrgs.br

Resumo. As Redes de Modelos Locais (RML) e as Redes Neurais (RN) do tipo feedforward são comparadas segundo sua capacidade de utilização em controle de processos. Devido a possibilidade de gerar resposta inversa e mudar o sinal do ganho, utilizou-se nessa comparação, como estudo de caso, a reação de Van de Vusse. Os resultados obtidos nesse exemplo demonstram a superioridade das RMLs frente as RNs. As RMLs foram capazes de capturar melhor a resposta inversa apresentada pelo sistema.

Palavras-chave: Rede de modelos locais, Redes neurais, Identificação de sistemas, Sistemas não lineares, Controle de processos.

1. INTRODUÇÃO

A modelagem de sistemas dinâmicos não lineares a partir de dados provenientes dos próprios processos é uma das áreas de grande interesse na engenharia. Duas das metodologias que utilizam este conceito são as Redes Neurais (RN) e as Redes de Modelos Locais (RML). Essas metodologias apresentam algumas similaridades, como por exemplo a necessidade de se estimar parâmetros com o objetivo de minimizar o erro existente entre os dados estimados pela rede e os dados provenientes do processo; a relevância dos sinais utilizados na perturbação do sistema para coleta de informações nas regiões de operação e a incapacidade de uma extrapolação satisfatória. Como principais diferenças pode-se ressaltar a dificuldade de obtenção de informações sobre o sistema bem como inserção de conhecimentos prévios na construção e utilização das Redes Neurais sendo, portanto, designadas de “caixas pretas”; enquanto que nas Redes de Modelos Locais, “caixas cinzas”, já possuem uma flexibilidade maior, permitindo a inserção de conhecimentos em praticamente todos os níveis da sua metodologia de construção e aplicação. Outra diferença importante está na relativa facilidade de se obter um controlador para as RML, que pode possuir uma estrutura similar a do controlador do tipo *Gain Scheduling*, enquanto que para as RN uma das possibilidades seria montar uma nova rede baseada no inverso da planta.

2. REDE DE MODELOS LOCAIS

A Rede de Modelos Locais é uma ferramenta dita híbrida, pois incorpora a possibilidade de introdução de conhecimentos prévios além do conjunto de dados para a modelagem do processo. A idéia de RML é uma extensão das Redes de Função Base Radial (FBR), que é considerada um tipo de Rede Neural. Ao contrário das FBR, onde as saídas são uma combinação de várias funções de ativação radiais onde os pesos são parâmetros constantes

* A quem a correspondência deve ser submetida.

após terem sido determinados, as Redes de Modelos Locais possuem pesos variáveis, que são função da região de operação. Desta forma, a rede pode ser descrita do seguinte modo:

$$y = f(x_1, x_r) = \sum_{i=1}^n f_i(x_i) \rho_i(x_r) \quad (1)$$

onde $f_i(x_i)$: modelos locais; x_i : variáveis utilizadas nos modelos locais; $\rho_i(x_r)$: função peso; x_r : variáveis que definem o ponto de operação do sistema.

A forma da rede da Eq.(1) é apresentada na Fig.1. Esta estrutura pode ser vista como a decomposição de um sistema não linear complexo em uma série de n sub-modelos locais válidos para uma dada região, cujo modelo global pode ser obtido através da ponderação das funções bases associadas de tal forma que ocorra uma transição suave de um modelo à outro [Trierweiler e Neumann, 1998].

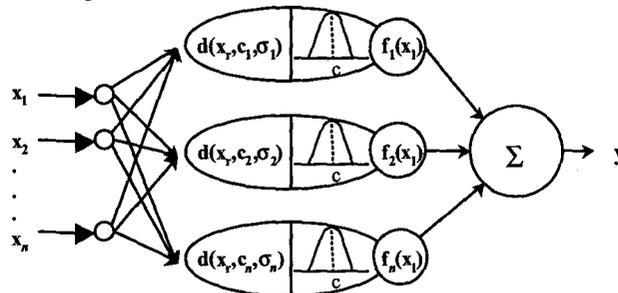


Figura 1. Representação esquemática da Rede de Modelos Locais.

2.1 Construção da Rede

A construção da rede é uma das tarefas mais importantes na metodologia da RML, pois é nesta etapa que muito do conhecimento prévio sobre o processo pode ser inserido. Para definir a topologia da rede basta determinar onde se localizarão as funções peso ou seja, seus centros, e o raio de espalhamento das mesmas no espaço. A metodologia utilizada propõe fazer uma seleção das variáveis que definirão o espaço de operação bem como os seus respectivos intervalos de variação. Após concluída esta etapa, o espaço é dividido em sub-regiões e são escolhidas as variáveis usadas nos modelos locais, podendo coincidir ou não com as variáveis que caracterizam o espaço de operação. Por fim, é feita a escolha da estrutura dos modelos locais e a determinação dos parâmetros utilizados nos modelos, que serão posteriormente combinados para a obtenção de um modelo global para o processo.

Para a identificação dos centros e raios de espalhamento no caso de não se ter informações sobre o processo, pode-se utilizar o algoritmo LOLIMOT [Nelles, 1997], que consiste basicamente em sucessivas divisões do espaço em hiperquadriláteros buscando um modelo global que resulte no menor erro. Os modelos locais utilizados neste algoritmo são lineares e do tipo ARX devido a facilidade de cálculo dos parâmetros e também porque se este modelo não for suficientemente satisfatório na descrição de uma dada região, significa que existe um certo grau de não linearidade nela sugerindo, portanto, uma nova divisão. O algoritmo LOLIMOT apresenta a seguinte estrutura:

1. gerar um hiperquadrilátero que inclua todos os pontos experimentais.
2. para cada uma das D dimensões que o hiperquadrilátero i possuir, deve-se proceder da seguinte maneira, isto é, $j=1, \dots, D$:
 - a. divida o hiperquadrilátero i no meio ao longo da dimensão j
 - b. colocar uma função peso normalizada no centro de cada um dos novos hiperquadriláteros, sendo σ (raio de espalhamento) proporcional as dimensões do hiperquadrilátero

- c. determinar os parâmetros dos respectivos modelos locais
 - d. calcular o erro produzido pela predição da rede
 3. determinar qual das D divisões feitas no passo 2 produz o menor erro
 4. efetivar a mais promissora das divisões, utilizando os resultados obtidos em (2b) e (2c).
 5. calcular os erros locais produzidos por cada um dos modelos locais através da ponderação do erro total pela função peso normalizada
 6. selecionar o hiperquadrilátero com o maior erro relativo para a próxima divisão
 7. caso não tenha sido satisfeito um critério de parada, retornar para 2, caso contrário parar.
- Desta forma, a cada iteração o subsistema com maior erro é dividido em duas partes iguais de forma a produzir o melhor aprimoramento do resultado.

2.2 Funções Peso

As funções peso são as responsáveis pelas transições suaves entre os modelos locais na composição do modelo global. Para assegurar a homogeneidade do espaço, as funções peso devem satisfazer a condição de normalidade, isto é

$$\sum_{i=1}^m \rho_i(x_r) = 1, \forall x_r \quad (2)$$

Em geral, as funções que quantificam a influência de uma dado centro no espaço não satisfazem esta condição, e neste caso ela pode ser normalizada através

$$\rho_i(x_r) = \frac{\mu_i(x_r)}{\sum_{k=1}^m \mu_k(x_r)} \quad (3)$$

onde $\mu_i(x_r)$ representa a função que está sendo normalizada e m o número total de centros. Uma escolha usual para $\mu_i(x_r)$, tanto para as FBR quanto para as RML é a função gaussiana.

$$\mu_i = \exp\left(-\frac{1}{2} \left(\sum_{r=1}^d \left(\frac{x_r - c_{i,r}}{\sigma_{i,r}} \right)^2 \right)^{1/n}\right) \quad (4)$$

que assuma a seguinte forma após ser normalizada:

$$\rho_i = \frac{\exp\left(-\frac{1}{2} \left(\sum_{r=1}^d \left(\frac{x_r - c_{i,r}}{\sigma_{i,r}} \right)^2 \right)^{1/n}\right)}{\sum_{k=1}^m \exp\left(-\frac{1}{2} \left(\sum_{r=1}^d \left(\frac{x_r - c_{k,r}}{\sigma_{k,r}} \right)^2 \right)^{1/n}\right)} \quad (5)$$

A variável c_k é a localização do centro k e σ_k equívale ao raio de espalhamento da função ou o desvio padrão, que quantifica o quão longe do centro k está um ponto no espaço definido pelas variáveis que caracterizam o espaço.

Dependendo da localização dos centros e do seu raio de espalhamento ao longo das dimensões, alguns problemas devido a normalização podem surgir, como por exemplo o ressurgimento da função peso em uma região na qual ela não existia antes ou então a redução do valor das ponderações em determinadas regiões, principalmente as próximas ao centro onde a ponderação deveria ser a máxima, como pode ser observado na Fig.2 apresentada a seguir.

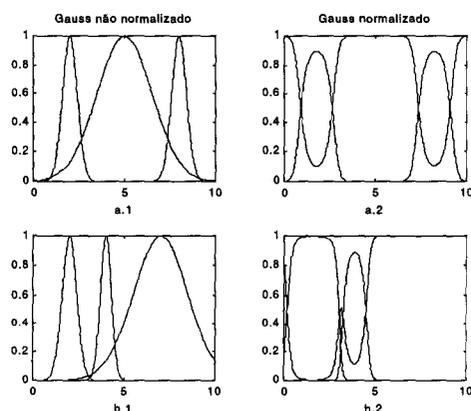


Figura 2. Funções Peso do tipo gaussiana normalizadas e não normalizadas para centros equidistantes e desvios padrões diferentes (a.1 e a.2) e para centros não equidistantes com desvios padrões diferentes (b.1 e b.2).

Para minimizar estes efeitos, pode-se escolher uma outra função peso denominada de Gauss generalizada, Eq.(6). Esta função além de apresentar menos problemas com a normalização, possui no topo um patamar, fazendo com que uma região maior em torno de um centro receba uma ponderação maior.

$$\mu_i = \frac{1}{1 + \left(\sum_{r=1}^d \left(\frac{x_r - c_{i,r}}{\sigma_{i,r}} \right)^{2nP} \right)^{1/n}} \quad (6)$$

Normalizando a função do tipo Gauss generalizada, obtém-se:

$$\rho_i = \frac{\sum_{k=1}^m \left(1 + \left(\sum_{r=1}^d \left(\frac{x_r - c_{k,r}}{\sigma_{k,r}} \right)^{2nP} \right)^{1/n} \right)}{1 + \left(\sum_{r=1}^d \left(\frac{x_r - c_{i,r}}{\sigma_{i,r}} \right)^{2nP} \right)^{1/n}} \quad (7)$$

onde P representa o parâmetro referente ao tamanho desejado para o patamar.

2.3 Metodologia RML

A metodologia desenvolvida consiste em cinco etapas básicas: 1) a geração de um sinal que perturbe o processo como um todo para identificar as regiões críticas de operação, ou seja, de maior não linearidade e dividi-las em sub-regiões; 2) determinar a topologia da rede, seja via conhecimentos prévios do sistema ou utilizando o algoritmo LOLIMOT; geração de

sinais que estimulem especificamente as sub-regiões determinadas no passo anterior; 3) escolha e cálculo dos modelos locais referentes as regiões, podendo ser lineares ou não; 4) ponderação dos modelos locais através das funções peso normalizadas compondo o modelo global e 5) por fim a validação da rede. Na Fig.3 pode-se ver esquematicamente a metodologia proposta.

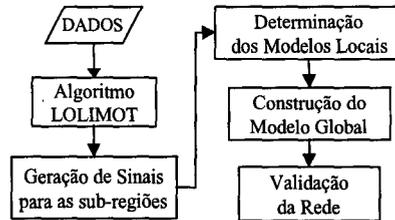


Figura 3. Esquema da metodologia básica para a construção da RML.

A etapa de construção de sinais para perturbar sub-regiões específicas tem por objetivo gerar um conjunto de dados pertencentes a uma região do espaço os quais serão utilizados para identificação de um modelo local, desta forma, o modelo identificado consegue capturar melhor as respostas do sistema, apesar de muitas vezes não conseguir capturar a real dinâmica do processo, como zeros positivos por exemplo.

3. REDES NEURAIS

As Redes Neurais são estruturas do tipo caixa preta não linear capazes de descrever virtualmente qualquer não linearidade de sistemas dinâmicos. A estrutura da rede é constituída basicamente de entradas que recebem uma ponderação e um valor base que serão avaliadas por funções chamadas de funções de ativação que geram um conjunto de saídas, onde cada uma destas funções e os vetores de ponderação respectivos a elas recebem o nome de neurônios, por analogia às conexões existentes com os neurônios biológicos. Esta estrutura pode repetir-se em diversas camadas sendo que a última camada sempre será constituída por um número de neurônios equivalentes ao número de saídas do sistema analisado. A Fig.4 apresenta uma estrutura esquemática de uma Rede Neural tipicamente utilizada para descrever processos não lineares.

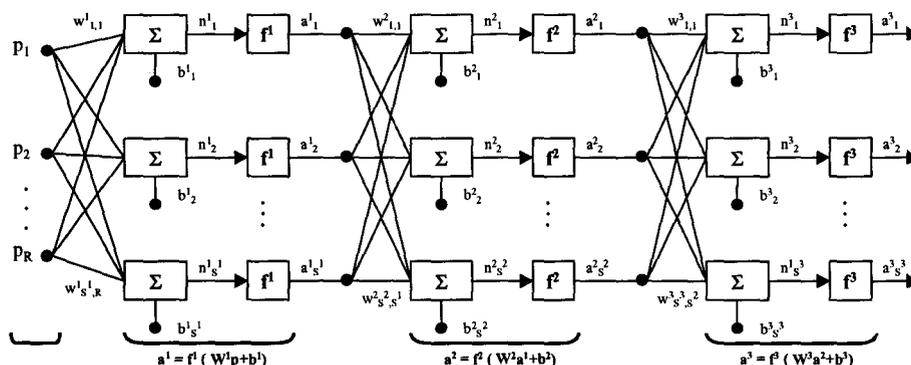


Figura 4. Rede com três camadas.

Os vetores p_i ($i = 1, 2, \dots, R$) mostrados na Fig. 4 são as entradas do sistema que serão ponderados pelos vetores $w_{j,k}$ pertencentes à matriz W dos pesos e depois acrescidos de um valor base $b_{j,k}$. O resultado destas operações seguem pela rede sendo avaliados pelas funções de ativação f^j gerando novas saídas que servirão de entradas para as camadas seguintes (com S^n neurônios) até que seja atingida a última camada. Desta maneira, a saída a^3 assume a seguinte forma:

$$a^3 = f^3(W^3 f^2(W^2 f^1(W^1 p + b^1) + b^2) + b^3) \quad (8)$$

Então na estrutura apresentada tem-se R entradas, S^1 neurônios na primeira camada, S^2 na segunda camada e S^3 na última camada, o que corresponde ao número de saídas do sistema.

3.1 Funções de Ativação

As funções de ativação ou também conhecidas por funções de transferência no jargão das redes neurais, são as responsáveis pelas características desejadas para a rede. No caso de se precisar de uma rede para proceder com seleção entre padrões, por exemplo, pode-se utilizar funções de ativação do tipo

$$\begin{cases} a = 0, \forall n \leq 0 \\ a = 1, \forall n > 0 \end{cases} \quad (9)$$

ou variações da mesma.

Na área de controle de processos o que interessa são redes capazes de capturar a não linearidade dos sistemas, o que pode ser conseguido utilizando redes com no mínimo duas camadas, sendo que a interna deve possuir uma função de ativação que confira o grau de não linearidade à rede, enquanto que a externa é conveniente que satisfaça todo o espaço das variáveis de saída. As funções que preenchem estes requisitos são as seguintes:

$$\textit{log-sigmoidal} \quad a = \frac{1}{1 + e^{-n}} \quad (10)$$

$$\textit{tangente hiperbólica-sigmoidal} \quad a = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (11)$$

$$\text{e para a camada de saída é usual se utilizar uma função } \textit{linear} \text{ do tipo } a = n \quad (12)$$

Construindo uma rede com duas camadas, a interna utilizando neurônios com a função de ativação *tangente hiperbólica-sigmoidal* e a última com a função *linear*, pode-se representar qualquer não linearidade, desde que o conjunto de dados seja suficiente e pertença a região de interesse [Hagan *et al.*, 1996].

3.2 Determinação dos Parâmetros da Rede

Como pode ser observado, para construir uma Rede Neural, basicamente o que é preciso determinar é o número de camadas internas e os respectivos números de neurônios. O número de neurônios deve ser o menor possível para evitar problemas de *overfit*. O *overfit* ocorre quando o número de parâmetros é tão elevado que a rede consegue apenas estimar de forma satisfatória as saídas para o próprio conjunto de dados para o qual foi construída. Depois de determinada a topologia da rede, procede-se com a determinação dos seus parâmetros W e b , o que é conhecido por “treinamento da rede”. Existem vários algoritmos para o ajuste dos parâmetros, no caso particular de se utilizar uma rede *feedforward*, na qual não existe retroalimentação, um dos algoritmos eficientes é o da retropropagação do erro (*back propagation*). Basicamente o que este algoritmo faz é o ajuste dos parâmetros da última camada e passa o erro para a camada anterior e assim sucessivamente até a primeira camada. Isto é possível devido a estrutura da rede, que equivale a aplicação da regra da cadeia em cada camada na hora do cálculo das derivadas para minimizar o erro quadrático médio.

4. REDE DE MODELOS LOCAIS vs. REDES NEURAIS – EXEMPLO

Nesta seção será apresentado um exemplo de um reator CSTR (*continuous stirred tank reactor*) onde ocorre a reação de Van de Vusse, um problema que geralmente é utilizado para testes de algoritmos de controle para processos não lineares [Chen *et al.*, 1995].

4.1 Reação de Van de Vusse

A reação de Van de Vusse consiste na produção de ciclopentanol (*B*) a partir do ciclopentadieno (*A*), tendo como subprodutos o ciclopentanodiol (*C*) e o dicitlopentadieno (*D*) (para maiores detalhes veja [Trierweiler, 1997]).



A Fig.5 apresenta esquematicamente o reator e as variáveis envolvidas no processo.

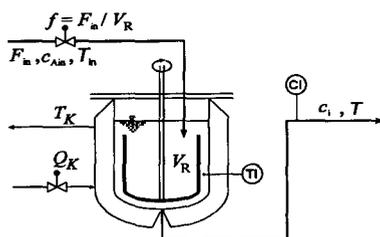


Figura 5. Representação esquemática do reator CSTR.

Assumindo que a massa específica é constante, a dinâmica do sistema pode ser descrita pelas cinco equações diferenciais apresentadas a seguir, que derivam do balanço de massa e energia para o reator e a camisa de resfriamento.

$$\frac{dV_r}{dt} = F_{in} - F_{out} \quad (14)$$

$$\frac{d(V_r c_A)}{dt} = F_{in} c_{Ain} - F_{out} c_A - V_r [k_1(T) c_A + k_3(T) c_A^2] \quad (15)$$

$$\frac{d(V_r c_B)}{dt} = -F_{out} c_B + V_r [k_1(T) c_A - k_2(T) c_B] \quad (16)$$

$$\frac{d(V_r T)}{dt} = F_{in} T_{in} - F_{out} T_{out} + \frac{k_w A_r}{\rho C_p} (T_k - T) \quad (17)$$

$$\begin{aligned} & - \frac{V}{\rho C_p} [k_1(T) c_A \Delta H_1 + k_2(T) c_B \Delta H_2 + k_3(T) c_A^2 \Delta H_3] \\ \frac{dT_k}{dt} &= \frac{1}{m_k C_{pk}} [Q_k + k_w A_r (T - T_k)] \end{aligned} \quad (18)$$

O reagente ciclopentadieno (*A*) é alimentado no reator com uma concentração igual a c_{Ain} e temperatura T_{in} . As vazões volumétricas de entrada e saída no reator são, respectivamente, F_{in} e F_{out} . Devido a forma de retirada de produtos, segue que $F_{in} = F_{out}$ e o volume V_r do reator é constante. As concentrações do ciclopentadieno (*A*) e do ciclopentanol (*B*) são, respectivamente, c_A e c_B . A temperatura no reator é denotada por T , enquanto que a temperatura da camisa de resfriamento é representada por T_k . A taxa específica de reação k_i é dependente da temperatura de acordo com a equação de Arrhenius.

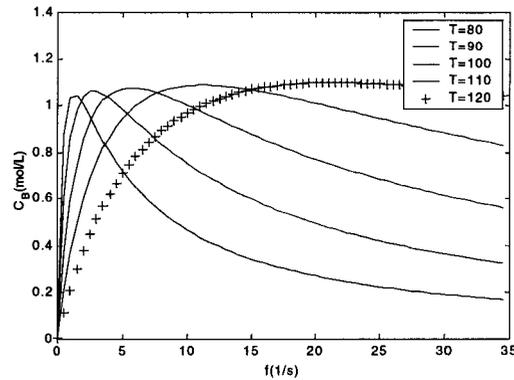


Figura 6. c_B vs f para várias temperaturas ($^{\circ}\text{C}$).

Fazendo um gráfico dos valores da concentração do componente B no estacionário variando com a taxa de diluição f para várias temperaturas T (Fig.6), pode-se constatar a presença de duas regiões distintas de operação, uma com o ganho positivo, a baixas taxas de diluição e outra com ganho negativo, a vazões mais elevadas.

4.2 Rede de Modelos Locais vs Rede Neural

Para que a comparação fosse a mais justa possível, utilizou-se o mesmo conjunto de dados para montar ambas as redes e depois um outro conjunto para validá-las. No caso da Rede de Modelos Locais, foram feitas duas simulações, uma utilizando modelos lineares identificados através dos dados e outra utilizando os modelos linearizados nos centros identificados pelo algoritmo LOLIMOT, que dividiu o espaço em 3 regiões, duas na região de fase não mínima e outra na região de fase mínima. Essas divisões foram feitas nas variáveis f e Q_k , o que seria de se esperar analisando o modelo linearizado. A Rede Neural construída é do tipo *feedforward*, com uma camada interna e cinco neurônios com funções de ativação tangente hiperbólica sigmoidal e a camada de saída é constituída por dois neurônios com a função do tipo linear. Para a simulação das RMLs, foi utilizado um horizonte de predição infinito, enquanto que para a RN o horizonte de predição foi de 20.

Comparando as saídas previstas pelas redes com os valores obtidos através de simulação, obteve-se os seguintes resultados:

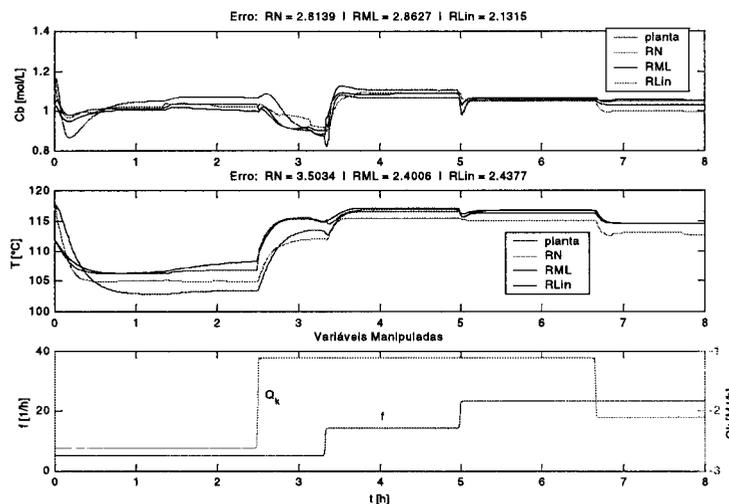


Figura 7. Validação das redes para conjunto de dados que perturbam regiões de interesse.

A Tabela 1 a seguir mostra o erro obtido em cada uma das previsões das redes em relação aos valores esperados para as saídas.

Tabela 1. Erros dos modelos preditos pelas redes em relação aos dados simulados.

	Rede Neural	RML com modelos identificados	RML com modelos linearizados
erro em c_B	2,8139	2,8627	2,1315
erro em T	3,5034	2,4006	2,4377

Em termos de comparação, o número de parâmetros a serem determinados para a rede neural foi de 77 parâmetros, enquanto que para a rede de modelos locais com os modelos identificados foi de 60 parâmetros.

As perturbações na entrada foram feitas de forma a mostrar os resultados da previsão para as duas regiões distintas do modelo, a de fase mínima e a de fase não mínima. O melhor conjunto de dados estimados para a Rede Neural está apresentado na Fig. 7, onde pode-se observar que esta rede consegue capturar a dinâmica do processo, bem como as outras redes.

4.3 Análise de um ponto de operação de fase não mínima

Um modelo para ser considerado como bom em termos de controle de processos deverá ter as seguintes características: 1) ser capaz de representar o comportamento de fase não mínima do sistema, i.e., tempo morto e resposta inversa, 2) ter o sinal correto do ganho estacionário do processo e 3) capturar as dinâmicas principais do sistema, das quais se destacam o tempo de subida e a sobre elevação (*overshoot*). Fig. 8 compara as respostas produzidas pelas RN e RML quando submetidas a uma perturbação tipo degrau nas variáveis manipuladas. As perturbações levam o sistema de uma condição inicial onde c_B é máximo para uma região onde o comportamento do sistema apresenta resposta inversa, ou seja, para um ponto situado a esquerda do máximo. Para eliminar os efeitos estáticos e melhor representar os aspectos dinâmicos da resposta, optou-se por mostrar Δc_B e ΔT . Note que a RML foi capaz de capturar melhor os aspectos dinâmicos do processo que a RN, indicando que a sua utilização para fins de controle apresentará um desempenho superior.

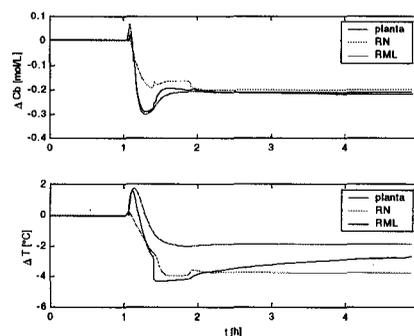


Figura 8. Δc_B e ΔT quando f passa bruscamente $14,2 \text{ h}^{-1}$ para $5,2 \text{ h}^{-1}$ e a temperatura inicial do reator é de $117 \text{ }^\circ\text{C}$.

4.4 Predição com horizonte unitário para RN e ARX221

Outro aspecto crítico para a aplicação de um modelo na área de controle é sua capacidade de previsão. O modelo para ser utilizado como base de um algoritmo de controle terá que ser capaz de prever a resposta do sistema num horizonte de controle de no mínimo 20 passos

futuros. Observe que todas as simulações com RN e RML foram realizadas utilizando-se respectivamente um horizonte de predição de 20 e infinito.

A Fig. 9 compara a resposta da RN com a predição de um único modelo linear do tipo ARX221 para um horizonte de predição unitário. Como pode ser observado na Fig. 9, os dois modelos concordam muito bem com os dados de planta (gerados para validação), pois com um horizonte de predição unitário o modelo se auto corrige, ou seja, ele utiliza os próprios dados de saída da planta, não precisando usar as saídas preditas pelo modelo, resultando numa qualidade que pode muitas vezes induzir a interpretações errôneas. Ainda é importante salientar, que o modelo ARX-221 é facilmente gerado, sendo os seus parâmetros determinados analiticamente.

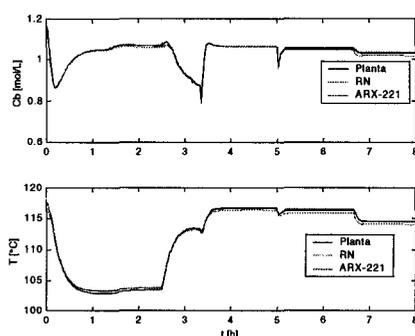


Figura 9. Validação dos modelos de RN e ARX-221.

5. CONCLUSÃO

Em termos de controle de processos os modelos gerados pelas RMLs foram superiores aos das RNs. Naturalmente, melhores resultados poderiam ser obtidos com ambas as redes caso se dispusesse de um maior conjunto de dados. Justamente nesse ponto as RMLs podem levar uma vantagem considerável frente as RNs, pois permitem a fácil inserção do conhecimento prévio do sistema. Para que a comparação fosse justa, não se fez uso desse conhecimento nesse trabalho. Adicionalmente, outra vantagem apresentada pelas RMLs é a sua relativa transparência, pois os seus parâmetros estão diretamente relacionados com as funções de transferências válidas em cada uma das regiões que definem o modelo podendo, dessa forma, serem direta e isoladamente analisados aplicando-se os conhecimentos básicos da teoria de controle. Infelizmente, a mesma transparência não se verifica na RNs. Note que para se proceder um refinamento da RN é necessário que se varie o número de neurônios e/ou o número de camadas consistindo em um procedimento com elevada carga computacional, uma vez que as informações referentes as parâmetros de uma rede não podem ser eficientemente transferidos para a nova topologia de rede.

REFERÊNCIAS

- Chen, H., A. Kremling, F. Allgöwer (1995), "Nonlinear Predictive Control of a Benchmark CSTR", *Proc. of 3rd ECC, Rome*, pp. 3247-3252.
- Hagan, M. T., Demuth, B. H. e Beale. M. (1996); "Neural Network Design"; PWS Publishing Company, EUA. pp. 2.1-3.9, 11.1-11.39
- Nelles, O. (1997), "LOLIMOT- Lokale, lineare Modelle zur Identifikation nicht-linearer, dynamischer Systeme, Automatisierungstechnik", pp.163-174.
- Trierweiler J. O. (1997), "A Systematic Approach to Control Structure Design", Ph.D. Thesis, Univ. of Dortmund.
- Trierweiler, J. O. e Neumann, U. (1998), "Rede de Modelos Locais: Uma solução simples para problemas complexos", COBEQ 98, Trab338, Porto Alegre, Brasil.