

Aplicações de redes neurais no Controle de Processos: Uma Perspectiva de substituição do modelo e do controlador

Autores: Rafael Domingos Nobre de Araújo^{*}; Darci Odloak

Departamento de Engenharia Química, Escola Politécnica, Universidade de São Paulo

Autor Correspondente: *rafaeldnobre@hotmail.com

Resumo

O Controle avançado de processos, desenvolvido na década de 1980, é amplamente utilizado em indústrias químicas para otimizar a produção e atender às limitações de processo. Entretanto, encontra problemas em alguns sistemas, seja na velocidade de computação do controlador, seja na modelagem desse processo, que pode ser imperfeita dadas as condições de operação e a transitoriedade decorrente de mudanças temporais. A Inteligência Artificial pode auxiliar nesses problemas, permitindo a criação de um controlador preditivo onde o modelo do processo com dinâmica não-linear é aproximado por uma rede neural que é utilizada na previsão da saída do sistema. Paralelamente, no caso de sistemas muito rápidos onde a solução de um problema de otimização não é compatível com o período de amostragem, a rede neural pode ser usada para substituir o controlador avançado que é baseado em um modelo linear ou não-linear. Neste trabalho são apresentadas modelagens para um sistema não-linear, uma para cada uma dessas abordagens, que tiveram resultados satisfatórios, permitindo o controle do sistema de maneira eficiente, sem a presença de offset.

Palavras-chave

Controle preditivo, Inteligência artificial, Redes neurais.

1. Introdução

Na década de 1970 surgiu o *Model Predictive Control* (MPC), metodologia que permitiu aplicar os modelos de processo representados por equações matemáticas, diretamente no controlador. Com base nessa modelagem, pode-se implantar técnicas mais avançadas de controle, permitindo controle por faixas e a garantia de atendimento a restrições de processo (Limon et al, 2006), aumentando consideravelmente o número de aplicações desses controladores em indústrias químicas e petroquímicas.

Com os avanços na teoria de MPC, novas aplicações e sistemas começaram a ser atendidos por esses controladores. Assim, sistemas com dinâmicas mais rápidas, de maiores dimensões, não-lineares e com outras particularidades começaram a ser controlados, garantindo mais flexibilidade e usabilidade para as ferramentas.

Nos últimos anos, uma nova fase do controle de processos começou a se revelar, baseada na utilização de técnicas de Inteligência Artificial e Aprendizado de Máquina. O objetivo da utilização dessas ferramentas é permitir controlar sistemas que não são controlados por MPCs comuns, como sistemas de grandes dimensões ou dinâmica muito rápida, como sistemas mecânicos (Cavagnari et al, 1999), ou com altos graus de não-linearidade, bem como, melhorar a performance dos controladores comuns através dessas técnicas (Kumar et al, 2021). Além disso, existe outro ramo de trabalhos que aborda a definição do modelo do processo via redes neurais (Wang et al, 2023)

Este trabalho tem como objetivo estudar a implantação de técnicas de Inteligência Artificial em um sistema de relativa complexidade para substituir o controlador MPC convencional, ou para substituir a modelagem do sistema estudado. Para isso, foram desenvolvidas redes neurais que aprendem a lógica do modelo do sistema e do controlador em malha fechada para substituí-los em aplicações práticas.

2. Metodologia

2.1. Sistema estudado

O sistema não-linear utilizado como base para esse trabalho considera um processo com duas entradas e duas saídas, no qual as variáveis de entrada são a vazão de alimentação da solução B concentrado e a vazão de alimentação de B diluído. As variáveis de saída são o nível de líquido no reator e a concentração de B no reator. As equações que descrevem o sistema estão descritas em (1). A definição das variáveis e seus valores estão dispostas na Tabela 1.

$$\begin{aligned}
\frac{dx_1}{dt} &= u_1 + u_2 - 0,2 * x_1^{0,5} \\
\frac{dx_2}{dt} &= (C_{B_1} - x_2) * \frac{u_1}{x_1} + (C_{B_2} - x_2) * \frac{u_2}{x_1} - \frac{k_1 x_2}{(1 + k_2 x_2)^2} \\
y &= x
\end{aligned} \tag{1}$$

Tabela 1: Variáveis e Constantes do Modelo.

Símbolo	Descrição	Valor
x_1	Nível de líquido no reator	...
x_2	Concentração de B no reator	...
u_1	Vazão de alimentação de B concentrado	...
u_2	Vazão de alimentação de B diluído	...
C_{B_1}	Concentração de B na corrente 1	24,9
C_{B_2}	Concentração de B na corrente 2	0,1
k_1	Constante cinética 1	1
k_2	Constante cinética 2	1

O controlador considerado inicialmente é um NMPC convencional sem restrições, definido pela equação (2), sujeito à restrição (3), onde, np representa o horizonte de predição, m o horizonte de controle, k o instante atual da simulação, \hat{y} representa a saída prevista, y^{sp} o set-point, Q é a matriz de ponderação das saídas, Δu é a variação no vetor de entradas e R é a matriz de ponderação das entradas.

$$\min_{\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+m-1)} \sum_{l=k+1}^{k+np} \{ [\hat{y}(l) - y^{sp}(l)]^T Q [\hat{y}(l) - y^{sp}(l)] + \Delta u(l-1)^T R \Delta u(l-1) \} \tag{2}$$

$$\Delta u(k+m) = \Delta u(k+m+1) = \dots = \Delta u(k+np-1) = 0 \tag{3}$$

As etapas envolvidas nesse controlador incluem a linearização e discretização do modelo, além da utilização desse modelo na predição das saídas do processo para o cálculo das entradas futuras através de um problema de otimização quadrática. Esse controlador não considera restrições nas saídas, apenas os erros em relação aos set-points, então para garantir a exploração de toda a região de operação, é necessário alterar esses valores.

2.2. Rede substituindo o modelo – NN System Identification

Inicialmente, propomos representar o modelo definido em (1) por uma rede neural. Para modelar uma rede que substitua o processo, são necessárias etapas sequenciais para a geração dos dados, treinamento da rede e simulação final. A primeira dessas etapas é a geração de dados em malha aberta para treinamento da rede, através de uma sequência aleatória de variações das entradas em uma faixa predefinida. O tempo de duração de cada uma dessas etapas é ajustado para que seja possível alcançar o estado estacionário por pelo menos 30 instantes para as duas saídas, mas não tão longo que represente um estado estacionário definitivo, o que poderia prejudicar o entendimento da dinâmica.

Em seguida são definidos os vetores de treinamento da rede, que são janelas com as últimas entradas no sistema e os últimos valores das saídas do sistema. Essa combinação de dados deve ser suficiente para treinar uma rede que consiga capturar a dinâmica do sistema e calcular o estado do sistema no próximo instante de tempo.

Após a criação do conjunto de dados, a rede, do tipo FFNN (*feedforward neural network*) foi treinada com funções de ativação lineares, como a *purelin* através de um problema de otimização com o objetivo

de minimizar a métrica de performance do ajuste, visto que performances próximas de zero apontam uma identificação perfeita do sistema.

Por fim, esse novo modelo do processo é posto em malha fechada com o controlador já definido, com o intuito de simular o processo em malha fechada. Nesse novo sistema, a rede neural é utilizada para prever o estado do sistema nos próximos instantes de tempo, permitindo assim ao controlador, com base nesse horizonte de estados calculados pela rede, definir uma variação ótimas nas entradas do sistema para levá-lo a um ponto de operação desejado e que atenda a todas as restrições.

2.3. Rede Substituindo o controlador – NN Controller

de Substituindo o controlador – NN Controller Em uma segunda etapa, propomos usar uma rede neural para substituir o controlador MPC, ou seja, a rede definirá qual o valor da manipulação das entradas será aplicado no sistema. Para se conseguir uma rede neural que simule as ações de controle definidas pelo controlador e atenda a todas as restrições do processo associadas às entradas e saídas do processo, é necessário treinar a rede em uma amplitude de cenários que cubram todas as possíveis condições operacionais. Nesse caso, o set-point é adotado e são definidas alterações percentuais ao redor desse valor para serem executadas em determinados períodos, obrigando o controlador original a sempre agir tentando atingir o set-point.

Os dados obtidos nessa sequência de simulações são armazenados em vetores que contêm o erro da saída do processo em relação ao set-point e a última ação de controle aplicada pelo controlador no processo como valores de input para a rede. O valor da variação da ação de controle calculada pelo controlador convencional é usado como saída para a rede. Esses vetores são usados para ajustar uma FFNN com funções de ativação não-lineares e que pretende substituir o controlador.

Para utilizar essa rede é necessário replicar a parte da interface do sistema com o controlador original, porém sendo possível remover toda a parte de modelagem do sistema e solução do problema de controle, bem como a definição de restrições, montagem de matrizes auxiliares, definição do hessiano e dos termos contantes no problema de otimização, além de outros elementos.

Com essas etapas realizadas, é necessário gerar um vetor para alimentar a rede com as mesmas informações utilizadas na modelagem da rede para gerar suas saídas que são as ações de controle a serem aplicadas no sistema. Com isso, espera-se que o sistema seja capaz de alcançar os set-points e satisfazer às restrições com mais adaptabilidade e com uma resposta mais veloz.

3. Resultados

3.1. Rede substituindo o modelo

A utilização de uma rede neural para substituir o modelo definido em (1) leva a um controlador tipo NMPC. Com isso, ou nós linearizamos a rede neural e resolvemos um problema de QP, ou resolvemos o problema de controle via uma NLP. A linearização da rede não foi tentada neste trabalho e optamos por resolver o NMPC diretamente via uma NLP. A solução desse problema via NLP com a rotina `fmincon` do Matlab se mostrou instável. Assim, optamos pela solução do NMPC usando um método baseado no método do gradiente descendente, que tem como função objetivo a equação (4) e obedece as restrições de (5), onde y_p representa o set-point, y representa o vetor de predições das saídas, Q e R são matrizes de ponderação e du representa a variação das entradas, du_{min} , du_{max} , u_{min} e u_{max} são as restrições das variações das entradas entre cada instante e os limites para as variáveis controladas, respectivamente.

Além disso, o modelo da rede é disposto na equação (6), onde k representa o instante atual, x o estado do sistema, u as entradas do sistema, na e nb representam a janela temporal considerada para cada termo e f representa a rede neural.

$$\min_{du} (y^{sp} - y)^T Q (y^{sp} - y) + du^T R du \quad (4)$$

$$\Delta u_{min} < \Delta u < \Delta u_{max} \mid U_{min} < U < U_{max} \quad (5)$$

$$x(k) = f(x(k-1), x(k-2), \dots, x(k-na), u(k-1), u(k-2), \dots, u|(k-nb)) \quad (6)$$

O treinamento da rede neural que representa o modelo dinâmico do reator foi feito com o vetor de dados de entrada para a rede contendo os valores das entradas saídas passadas do reator, onde $na = 10$ e $nb = 7$. A rede tem como saída o valor da saída do reator no instante atual, e uma arquitetura com duas camadas escondidas, a primeira com 50 nós e a segunda com 10 nós, ambas com funções de ativação lineares. A definição desses parâmetros foi feita de maneira empírica, com o intuito de balancear o tempo de treinamento da rede, que cresce consideravelmente com a quantidade de nós e o tamanho dos vetores de treinamento, e a performance resultante, que caso muito pequena, pode representar um fenômeno de *overfitting*, não ideal para o nosso trabalho.

O treinamento da rede foi feito com base em uma simulação de 100000 instantes de tempo, no qual as entradas do processo variavam a cada 100 instantes. A Figura 1 apresenta um trecho dos primeiros mil dados gerados para o treinamento da rede, onde são representadas as saídas e entradas do sistema. A performance da rede foi avaliada por meio da métrica de MSE (*mean squared error*) e apresentou um valor de $2,28 \times 10^{-8}$, indicando um excelente ajuste.

A simulação da malha fechada com o controlador NMPC baseado no modelo do reator representado pela rede neural obtida acima, tem como resultado a Figura 2. Nesse caso, os set points das saídas sofreram uma variação em degrau e foi imposta uma variação bastante suave das variáveis de entrada do reator. Um resultado importante dessa simulação é a observação que as duas variáveis controladas tendem para os set-points, sem a presença de off-set, apesar de um *overshoot* em y_2 .

É importante notar que apesar de o sistema se estabilizar no *set-point*, a resposta dos dois controladores é consideravelmente diferente, com o controlador original gerando picos nas duas entradas, enquanto o controlador baseado no sistema da rede apresenta respostas mais suaves, que se estabilizam inclusive em outros patamares, essa diferença na resposta é a razão para o *overshoot* em y_2 .

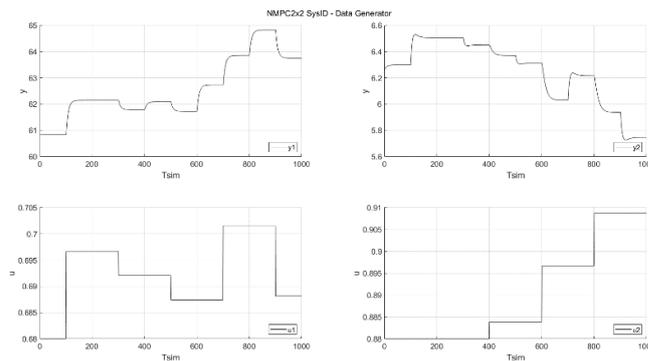


Figura 1: Geração de dados para treinamento da rede – NN System Identification

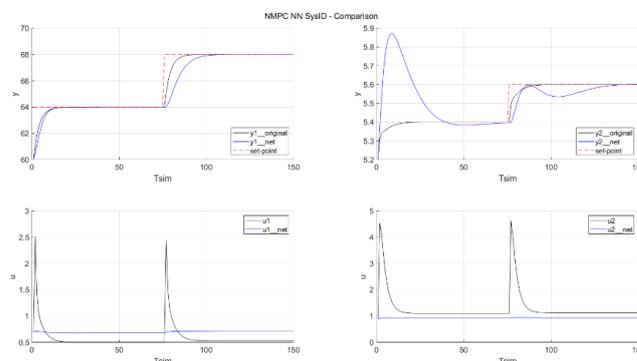


Figura 2: Simulação em malha fechada com modelo de rede neural

3.2. Rede substituindo o controlador

Para a geração dos dados de treinamento da rede que deveria substituir o controlador NMPC convencional foi definido um intervalo de 5000 instantes, onde o valor do set-point foi alterado aleatoriamente

a cada 50 instantes em torno de um set-point pré-definido, a rede tinha uma arquitetura com duas camadas ocultas, a primeira com 100 nós e a segunda com 20 nós, ambas com funções de ativação não-lineares.

O Modelo dessa rede é apresentado em (7), onde f representa a rede, $u(k)$ representa a entrada do sistema no instante atual, $u(k - 1)$ representa a entrada no instante anterior e $e(k - 1)$ representa o erro entre o estado real e o set-point no instante anterior.

$$u(k) = f(e(k - 1), u(k - 1)) \quad (7)$$

Os primeiros 1000 dados gerados são apresentados na Figura 3, que representam a simulação em malha fechada. A performance dessa rede foi de $1.7 * 10^{-3}$, indicando um ajuste aceitável. A Figura 4 apresenta a simulação da malha fechada com o NNMPC, onde é possível observar que o controlador tem um comportamento bastante adequado e temos uma rápida estabilização do sistema e ausência de *off-set*, apesar da presença de um *overshoot*, visualizado principalmente em y_2 , decorrente de uma variação muito grande em u_2 no sentido oposto ao do controlador original.

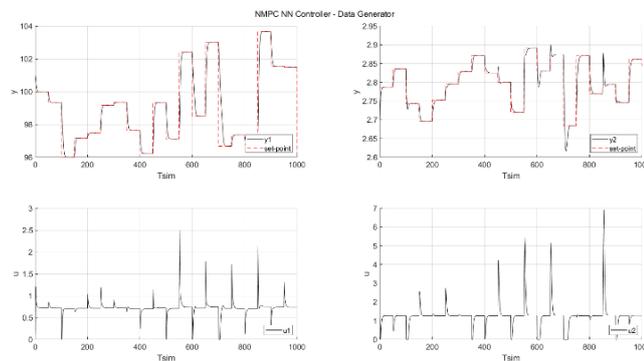


Figura 3: Geração dos dados para treinamento da rede – NN Controller

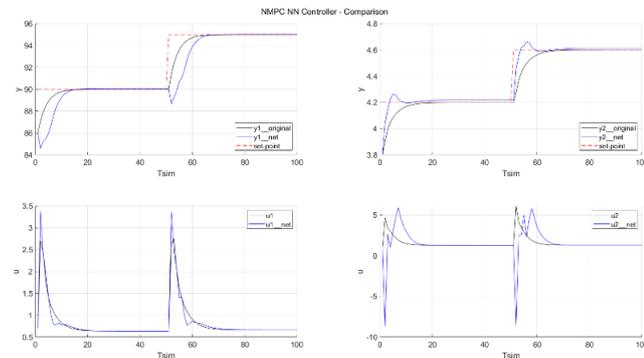


Figura 4: Simulação com NN Controller

4. Conclusão

Os resultados deste trabalho, ainda preliminar, indicam a possibilidade da integração da inteligência artificial com o controle de processos em duas frentes importantes, apresentando respostas em malha fechada sem *off-set*, com rápida convergência e capacidade de adaptação ao longo do tempo.

Referências

Cavagnari, L.; Magni, L.; Scattolini, R. Neural Network Implementation of Nonlinear Receding- Horizon Control. *Neural Computing & Applications*, vol. 8, p. 86-92, 1999. Springer-Verlag London Limited.

Limon, D. et al. On the Stability of Constrained MPC Without Terminal Constraint. *IEEE Transactions on Automatic Control*, 2006. Disponível em: <https://doi.org/10.1109/TAC.2006.875014>.

Kumar, P.; Rawlings, J. B.; Wright, S. J. Industrial, large-scale model predictive control with structured neural networks. *Computers and Chemical Engineering*, v. 150, p. 107291, 2021. DOI: <https://doi.org/10.1016/j.compchemeng.2021.107291>.

WANG, Yaxin; XU, Baochang. A Hybrid Strategy Enhanced Sub-Sampled Recursive Second Order Algorithm for Chemical Process Intelligent Identification. *International Journal of Computational Intelligence Systems*, 2023. Disponível em: <https://doi.org/10.1007/S44196-023-00296-5>.