Efficient Neuro-predictive Control of a Chemical Plant

Morteza Mohammadzaheri[1], Mehdi Ghanbari[2], Ali Mirsepahi[3], Fariba Behnia-Willson[4]
morteza.mohammadzaheri@adelaide.edu.au
[1]Islamic Azad University-Semnan Branch, [2]University of South Australia,
[3]Adelaide University, [4]Flinders University
Name of the Presenter: Morteza Mohammadzaheri

Abstract

In nonlinear predictive control, there usually exist two components, a nonlinear model to predict the behaviour of the system, and an optimisation algorithm to generate the control command to minimize the performance function (which is highly influenced by current and predicted errors). If artificial neural networks are used as nonlinear model; the controller is called “neuro-predictive”. In neuro-predictive control, second-order derivative-based optimisation methods, particularly, Levenberg-Marquardt method are employed to achieve a better performance. Using such optimisation methods, rather than steepest descent (first order ones), leads to better performance of control system; although, they need much more computation in comparison to first-order methods, as an important drawback. In this paper, an optimisation algorithm is developed by the combination of fuzzy logic and steepest descent method, particularly for neuro-predictive control purposes (not ordinary optimisation problems). The usage of proposed method in neuro-predictive control leads to a control performance roughly as good as the control performance with Levenberg-Marquardt (LM) method; at the same time, as the main advantage, the computation time with the proposed method is ten times shorter than with LM. That is, it is around ten times more efficient than LM.

Key words: Neuro-predictive, control, fuzzy, optimisation, CSTR, process plant

1. Introduction

In feedback control, as the most common type of control, the control command is generated using the error which has already occurred, whereas, in predictive control the predicted error (which is going to occur) is utilized to generate control command to avoid the error before appearing (Camacho and Bordons 2004); to do so a model (for predict the system’s response) and a control algorithm (to generate the control command) are needed.

Predictive control was initially introduced as the classical model predictive control needing a linear state space model of system (Camacho and Bordons 2004),(Bemporad, Morari et al. 2010)]. However, the nonlinearity of many systems is not negligible. In such occasions, approximate fully (Nagy, Mahn et al. 2007) or piecewise (Magni and Scattolini 2007) linear models may be used. But, in general, nonlinear models are needed to predict the output(s) of nonlinear systems for control purposes. There are some physics/chemistry-based methods which define the model of some systems entirely (Holenda, Domokos et al. 2008) or partially (Harnischmacher and Marquardt 2007). Artificial neural networks (Mohammadzaheri and Chen 2010) (Mohammadzaheri, Chen et al. 2009) (Al Seyab and Cao 2008) and fuzzy inference systems (neuro-fuzzy networks) (Ghaffari, Mehrabian et al.

2007; Karer, Music et al. 2007; Mohammadzaheri and Chen 2008) also can model dynamic systems. In this paper recurrent neuro-fuzzy and neural networks are designed/trained to model Catalytic Continuous Stirred Tank Reactor which is MIMO and nonlinear. It is found that a neural network (perceptron) can do this job better, so it is used as the model. As a result, this paper, the design of a neuro-predictive control system is addressed.

When nonlinear models are used, nonlinear optimisation methods are needed as control algorithms in predictive control. Several methods have been used for optimisation, such as evolutionary algorithms (Karer, Music et al. 2007), control weighting (Altinten 2007), automatic differentiation (Al Seyab and Cao 2008) and chaos optimisation (Song, Chen et al. 2007). Among all, Levenberg-Marquardt method, as one of the best derivative based methods is commonly used in neuro-predictive control (Demuth, Beale et al. 2010) (Tan and VanCauwenberghe 1996). In this paper, a combination of steepest descent method and fuzzy logic is proposed as control or optimisation algorithm, and the results are compared to LM results.

2. Neuro-predictive Control

In In neuro-predictive control, an artificial neural network is used to predict the behaviour of nonlinear systems, and an optimisation method defines the control command (represented by u) based on minimising a performance function involving the predicted errors. (1) is a typical performance function (represented by J) in neuro-predictive control.

$$J(k)=\sum_{i=1}^{N}[y_s(k+i)-y_d]^2+\rho[u'(k)-u(k-1)]^2. \quad (1)$$

y_s and y_d are the estimated and desired outputs of the system respectively, and u' and u are tentative and actual control inputs; ρ represents the importance of the constancy of control input.

As the first stage of the definition of tentative control command, the performance function (J) should be calculated. To do so, the output values should be predicted for N future instants (see (1)), so the nonlinear model should be used N times. N is called <u>the horizon of prediction.</u>

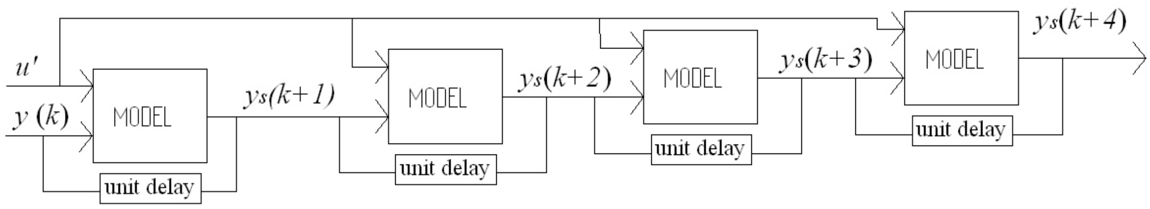

Fig 1: Prediction of output values with the horizon of 4

If current output and previous output/input of the system (the measured data) and u' are known, all other arguments of J will be definitely known (see Fig.1). So these arguments can not be subject to modification by optimisation algorithms. However u' can be changed arbitrarily freely from the measured input/output signals and this change affects other arguments of J, then the performance function itself. Therefore, in the optimisation for control purposes, it can be assumed:

$$J = J(u'). \quad (2)$$

3. Case Study

The case study is a Catalytic Continuous Stirred Tank Reactor (CSTR). A diagram of the process is shown in the following figure:

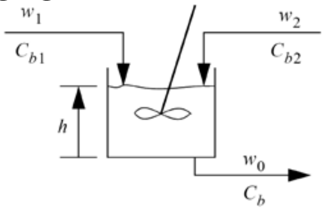

Fig 2: A schematic of the studied CSTR (Demuth, Beale et al. 2010)

Two flows of liquid enter the reactor with the concentration of $C_{b1} = 24.9\,(kmol/m^3)$ and $C_{b2} = 0.1$ $(kmol/m^3)$. The flow rate of input flows are named w_1 and w_2 .The reactor outlets another flow of liquid with the concentration of C_b and the flow rate of w_0. Another important variable is the height of liquid in the reactor (*h*).
A simplified mathematical model of system, achieved by mass equilibrium equations, is:

$$\frac{dh(t)}{dt} = w_1(t) + w_2(t) - 0.2\sqrt{h(t)}, \tag{3}$$

$$\frac{dC_b(t)}{dt} = (C_{b1} - C_b(t))\frac{w_1(t)}{h(t)} - (C_{b2} - C_b(t))\frac{w_2(t)}{h(t)} - \frac{k_1 C_b(t)}{(1 + k_2 C_b(t))^2} \ , \tag{4}$$

where k_1 and k_2 are parameters relevant to valves' resistance.

If the concentration of outlet flow and the height of liquid are considered as the outputs ($w_0 = 0.2\sqrt{h}$), the total system can be shown as below:

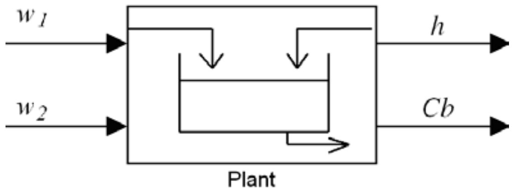

Fig 3: The studied CSTR as a MIMO system

In predictive control one control command is often used, the flow rate of the second input flow (with the concentration of 0.1 $kmol/m^3$) is set to the constant value of 0.1 $litres/min$.

4. Intelligent Modeling of the Case Study

The order of two is considered for the model and 8000 set of data (including w_1, *h* and C_b) with sampling time of 0.2 second are utilized in training. Then two coupled single-output adaptive neuro-fuzzy networks (ANFIS) and a perceptron are designed and trained to model the system. Prediction accuracy is defined as modeling criterion.

$$\text{PAN} = \sum_{i=1}^{N} |\, \hat{C}_b(i) - C_b(i) \,| \,. \tag{5}$$

Table 1 shows PA10 and PA30 (the sum of absolute error of prediction for 10 and 30 future instants or next 2 or 6 seconds), for two different series of checking data. Further information is available in (Mohammadzaheri and Chen 2008).

Criterion	PA10 ($kmol/m^3$)		PA30($kmol/m^3$)	
Checking data	1st series	2nd series	1st series	2nd series
Neuro-fuzzy model (two ANFIS)	0.060	0.036	0.752	0.658
Perceptron (double output)	0.018	0.022	0.051	0.033

Table 1. Prediction accuracy for different trained models.

5. Optimisation

From section 2, it is found that for nonlinear predictive control purposes:

$$J = J(u'). \tag{2}$$

Now, u' should be so determined that J has its minimal value. To do so, Taylor's series is written for performance function up to the first and second order:

$$J(u' + \Delta u') \cong J(u') + \frac{\partial J(u')}{\partial u'}(\Delta u'), \tag{6}$$

$$J(u' + \Delta u') \cong J(u') + \frac{\partial J(u')}{\partial u'}(\Delta u') + \frac{1}{2}\frac{\partial^2 J(u')}{\partial u'^2}(\Delta u')^2, \tag{7}$$

Based on these expansions, two different methods are used for optimisation in this paper, Levenberg-Marquardt method, based on (7), which is a very good derivative based optimisation method (Tan and VanCauwenberghe 1996; Jang, Sun et al. 2006). This method is currently used for neuro-predictive control (Demuth, Beale et al. 2010). The second method, based on (6), is a combination of steepest descent and fuzzy logic.

5-1. Levenberg Marquardt (LM)

In practice following relation is used for LM optimisation (Jang, Sun et al. 2006):

$$\Delta u' = u'_{new} - u'_{old} = -\eta[\frac{\partial^2 J(u')}{\partial u'^2} + \lambda]^{-1}\frac{\partial J(u')}{\partial u'}. \tag{8}$$

where; $\lambda = d \times \dfrac{\partial^2 J(u')}{\partial u'^2}.$ (9)

An initial value is assigned to *d* (i.e. 0.001), then δ is generated:

$$\delta = \frac{\dfrac{\partial J(u')}{\partial u'}}{(d+1)\dfrac{\partial^2 J(u')}{\partial u'^2}}, \tag{10}$$

Then: if $E(u' + \delta) < E(u')$ then *d=d/10*; otherwise *d=d×10;*

(10 is a modification factor, it can be of another value)

As a value of d is repeated, it is the result.

(8) represents Levenberg-Marquardt method for the optimisation of a single variable function (Jang, Sun et al. 2006):

In this method, g_k is numerically considered as performance function gradient:

$$\frac{\partial J(u')}{\partial u'} = g_k = \frac{J(k) - J(k-1)}{u'(k) - u(k-1)}; \tag{11}$$

moreover, G_k is defined as:

$$[\frac{\partial^2 J(u')}{\partial u'^2}]^{-1} = G_k = \frac{u'(k) - u(k-1)}{(g_k - g_{k-1})}. \tag{12}$$

So (11) can be rewritten in this form:

$$\Delta u' = u'_{new} - u'_{old} = -\eta(1+d)G_k g_k. \tag{13}$$

Using (15), we will have:

$$J(u'_{new}) = J(u'_{old} - \eta(1+d)G_k g_k), \tag{14}$$

or: $Argument\, of\ J = u'_{old} - \eta(1+d)G_k g_k$. (15)

Both u'_{old} and $(1+d)G_k g_k$ are known in this stage, then, with changing η, $Argument\, of\ J$ moves along a line. There is an optimum point on this line that minimizes J. Such an optimization problem is classified as a linear search. Backtracking method, introduced by Dennis and Schnabel (Dennis and Schnabel 1989) is selected for linear search. The modified u' (u'_{new}) is used as new control input.

5-2. Fuzzy Steepest Descent (FSD)

A fuzzy-derivative optimisation method is specially designed to suit optimisation tasks for predictive control purposes.

In (6), if $\Delta u'$ is replaced by $-\eta\frac{\partial J(u')}{\partial u'}$ then:

$$J(u' + \Delta u') \cong J(u') - \eta\left(\frac{\partial J(u')}{\partial u'}\right)^2. \tag{16}$$

If $\eta > 0$, it can be concluded (approximately) that:

$$J(u' + \Delta u') \le J(u'), \tag{17}$$

Any positive value can be used for η. In this research, for nonlinear predictive purposes, η is generated by a fuzzy inference system (FIS) to reduce the alteration of control system's response when it is approaching the setpoint/reference. This FIS has two rules (*e* is error):

Rule 1: if $|e|$ is LOW then $\eta = 0$

Rule 2: if $|e|$ is $HIGH$ then $\eta = 1$

LOW and *HIGH* membership functions are defined as:.

$$\begin{cases} \mu_{HIGH} = |e|, \ \text{for} |e| < 1 \\ \mu_{HIGH} = 1, \ \text{for} |e| \ge 1 \end{cases} \tag{18}$$

$$\begin{cases} \mu_{LOW} = -2|e| + 1, \text{ for } |e| < 0.5 \\ \mu_{LOW} = 0 \qquad , \text{ for } |e| \geq 0.5 \end{cases} \tag{19}$$

The fuzzy inference system is a Sugeno-type FIS, so (Mohammadzaheri and Chen 2008):

For $|e| < 0.5$ $\quad \eta = \dfrac{[0 \times (-2|e| + 1)] + [1 \times |e|]}{(-2|e| + 1) + |e|} = \dfrac{|e|}{1 - |e|}$

Moreover, for $|e| \geq 0.5$, $\quad \eta = \dfrac{[0 \times 0] + [1 \times \mu_{HIGH}]}{(0) + \mu_{HIGH}} = 1$

Or in summary:

$$\begin{cases} |e| < 0.5 \Rightarrow \eta = \dfrac{|e|}{1 - |e|} \\ |e| \geq 0.5 \Rightarrow \eta = 1 \end{cases} \tag{20}$$

6. Simulation Results

In neuro-predictive control of process plants, two design criteria are usually considered; the error integral and the constancy of input (usually in the form a flow rate):

$$EI = \frac{\int_0^{\tau} |e| dt}{\tau}, \tag{21}$$

$$IC = \frac{\int_0^{\tau} |u(t + \Delta t) - u(t)| dt}{\tau}. \tag{22}$$

Levenberg-Marquardt optimisation method leads to good results from both aspects. The proposed method (fuzzy steepest descent), offers results roughly as good as Levenberg-Marquardt (the currently prevalent method); at the same time it is several times more efficient than the first method. If steepest-descent optimisation method is used for optimization (without fuzzy inference system), the controller is practically impossible to implement because of substantial change of control input during operation. Fig.9 and Table III show the results when Levenberg-Marquardt (LM),fuzzy steepest descent (FSD) or pure steepest decent method (SD) method is used as the optimiser part of nonlinear predictive controller. In Table III, ‘Time’ is the time needed to simulate 100 seconds of operation of the closed loop system, with sample time of 0.2 s, using a dual core processor (4200 MHz) and MATLAB software. In this example, setpoints changes very quickly to test the capabilities of controllers.

	EI (*kmol/* m^3)	IC (*litres*)	Time (*s*)
LM	0.3651	0.0939	225
FSD	0.3729	0.0944	25
SD	0.7738	0.2553	24

Table 2. Simulation results with different optimisation algorithms

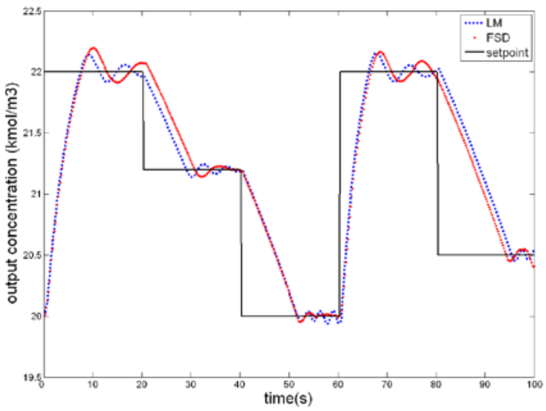

Fig 4: the response of system with LM and FSD optimisation algorithms

6. Conclusion

In this paper, neuro-predictive control of a Catalytic Continuous Stirred Tank Reactor of process plants is addressed. Particularly, in term of “efficiency”; the time needed for “computation”. A nonlinear predictive controller includes a modeling tool and an optimisation method. A perceptron neural is selected as the model for this case study in terms of offering more accurate prediction in comparison to neuro-fuzzy model. For optimisation, well-known method of Levenberg-Marquardt (LM) is used. Furthermore, a combination of steepest descent method and fuzzy logic is developed as an intelligent optimisation method, especially designed for nonlinear predictive control purposes. Offering a performance nearly as good as LM, the newly introduced method, needs around ten times shorter time for computation of control input; as a result, it suits better for real application.

References

Al Seyab, R. K. and Y. Cao (2008). "Differential recurrent neural network based predictive control." *Computers & Chemical Engineering*, 32(7), 1533-1545.

Altinten, A. (2007). "Generalized predictive control applied to a pH neutralization process." *Computers & Chemical Engineering*, 31(10), 1199-1204.

Bemporad, A., M. Morari, et al. (2010). *Model Predictive Control Toolbox 3, User’s Guide*, Mathworks.

Camacho, E. and C. Bordons (2004). *Model Predictive Control*. London, Springer-Verlag.

Demuth, H., M. Beale, et al. (2010). *Neural Networks Toolbox 7, User’s Guide*, The MathWorks.

Dennis, J. and R. Schnabel (1989). *Optimization, Chapter I, A view of unconstrained optimization*. Amsterdam Elsevier Science Publishers.

Ghaffari, A., A. R. Mehrabian, et al. (2007). "Identification and control of power plant de-superheater using soft computing techniques." *Engineering Applications of Artificial Intelligence*, 20(2), 273-287.

Harnischmacher, G. and W. Marquardt (2007). "A multi-variate Hammerstein model for processes with input directionality." *Journal of Process Control*, 17(6), 539-550.

Holenda, B., E. Domokos, et al. (2008). "Dissolved oxygen control of the activated sludge wastewater treatment process using model predictive control." *Computers & Chemical Engineering*, 32(6), 1270-1278.

Jang, J. R., C. Sun, et al. (2006). *Neuro-Fuzzy and Soft Computing*. New Delhi, Prentice-Hall of India.

Karer, G., G. Music, et al. (2007). "Hybrid fuzzy modelling for model predictive control." *Journal of Intelligent & Robotic Systems*, 50(3), 297-319.

Magni, L. and R. Scattolini (2007). "Tracking of non-square nonlinear continuous time systems with piecewise constant model predictive control." *Journal of Process Control*, 17(8), 631-640.

Mohammadzaheri, M. and L. Chen (2008). Intelligent Modelling of MIMO Nonlinear Dynamic Process Plants for Predictive Control Purposes. *The 17th World Congress of The International Federation of Automatic Control*, Seoul, Korea.

Mohammadzaheri, M. and L. Chen (2010). "Intelliegnt Predictive Control of Model Helicopters' Yaw Angle." *Asian Journal of Control*, 12(6), 1-13.

Mohammadzaheri, M., L. Chen, et al. (2009). "A combination of linear and nonlinear activation functions in neural networks for modeling a de-superheater." *Simulation Modelling Practice and Theory*, 17(2), 398-407.

Nagy, Z. K., B. Mahn, et al. (2007). "Evaluation study of an efficient output feedback nonlinear model predictive control for temperature tracking in an industrial batch reactor." *Control Engineering Practice*, 15(7), 839-850.

Song, Y., Z. Q. Chen, et al. (2007). "Neural network nonlinear predictive control based on tent-map chaos optimization." *Chinese Journal of Chemical Engineering*, 15(4), 539-544.

Tan, Y. H. and A. R. VanCauwenberghe (1996). "Optimization techniques for the design of a neural predictive controller." *Neurocomputing*, 10(1), 83-96.