Nonlinear Internal Model Control and Model Predictive Control using Neural Networks

D. C. Psichogios and L. H. Ungar

Department of Chemical Engineering
University of Pennsylvania

Abstract

Two nonlinear model based control strategies, Internal Model Control (IMC) and Model Predictive Control (MPC), were modified to incorporate the use of neural networks and applied to the control of a nonlinear SISO exothermic CSTR. An IMC-type neural network controller in which the process model was replaced by a neural network gave very good performance, even when only partial state data was available. An MPC-type neural controller, extended to include feedback, also gave excellent performance. These results indicate that neural networks can learn accurate models and give good nonlinear control when model equations are not known.

1. Introduction

Neural networks have been successfully used to identify dynamical systems which exhibit complicated behavior ([1], [10]) and they are starting to be used as models in model based process control ([7], [8], [9], [11]). Using neural networks in model based control offers the advantage of learning valuable information about the dynamical system which can be used in the controller design, thus improving performance and minimizing the need for expert intervention. Linear control strategies have been shown to work poorly when applied to certain nonlinear systems [4]; neural networks promise to provide a flexible basis for adaptive nonlinear controllers. This paper studies the ramifications of incorporating neural networks in two model based control architectures, namely Internal Model Control (IMC) and Model Predictive Control (MPC).

The details of the IMC control strategy for linear systems can be found in the landmark paper of Garcia and Morari [5]. This approach was recently extended to nonlinear systems by Economou and Morari [4], whose work we use as a base for our development. By incorporating networks into an established framework, a host of results about IMC and MPC [6] can be directly applied.

The control of the output concentration of a nonlinear SISO chemical reactor (Fig. 1) was chosen as a test problem. The network architecture we used was the feedforward backpropagation network [12] with sigmoidal activation functions. Section two of this paper describes the development of a neural network analog to the conventional IMC design, discusses the controller behavior and presents control architectures necessary to improve controller performance. Section three examines the performance of the neural network controller under less restrictive assumptions. In section four a neural network analog to the conventional MPC design is developed and tested; suggestions for improving performance are also presented. Finally, in section five, the results are discussed and suggestions for further improvement are presented.

2. Internal Model Control scheme

2.1 Implementation

A non-isothermal CSTR (Fig. 1) with a first order reversible reaction \( A \leftrightarrow R \) was used as a test problem. The process model, which is open-loop stable, is given by the following equations [4]:

\[
\frac{dA}{dt} = \frac{1}{t} \cdot (A_0 - A) - k_1 \cdot A + k_{-1} \cdot R
\]

(1)

\[
\frac{dR}{dt} = \frac{1}{t} \cdot (R_0 - R) + k_1 \cdot A - k_{-1} \cdot R
\]

(2)

\[
\frac{dT}{dt} = \left( \frac{-\Delta H_R}{\rho \cdot c_p} \right) \cdot (k_1 \cdot A - k_{-1} \cdot R) + \frac{1}{t} \cdot (T_0 - T)
\]

(3)

\[
k_1 = C_1 \cdot \exp\left( -\frac{Q}{R \cdot T} \right)
\]

\[
k_{-1} = C_{-1} \cdot \exp\left( -\frac{Q}{R \cdot T} \right)
\]

The same parameter values as in [4] were used. The steady state conversion as a function of temperature has a maximum; the control objective was to operate the process close to this maximum conversion point. The product concentration \( R \) was chosen to be the controlled variable and the inlet feed temperature \( T_i \) the manipulated variable; the resulting control problem is nonlinear (i.e., the controlled output - manipulated input relationship is nonlinear) and ill-posed since the input-output mapping is not invertible; for a given output concentration there are two values of the manipulated variable that can produce the desired output (a low temperature and a high temperature). Because of the maximum conversion, training of the networks had to be restricted to a region where the inverse mapping is unique. The same problem was circumvented by Economou and Morari by including a filter in the control design such that the error signal in the controller does not become larger than this maximum value. For the same reason a PI controller failed to control the process.

In order to obtain a training dataset for the neural networks, the differential equations describing the physical system were integrated over time with the manipulated input \( T_i \) varying...
An IMC type control scheme where the inlet feed temperature was the manipulated variable and both the controller and the model were replaced by neural networks was implemented (Fig. 3). The system was subjected to a -2% constant disturbance in the inlet feed concentration $A_i$ from $t=0$ to $t=20$ min. The dual neural network controller was unable to handle the disturbance (Fig. 4), contrary to the conventional IMC control scheme where the operator describing the system's behavior (Eqn. 1-3) was inverted using Newton's method. The inferior performance of the neural network controller was due to the fact that the network used to represent the inverse process dynamics was not the exact inverse of the process model network.

---

2.2 Dual neural network controller

An IMC type control scheme where the inlet feed temperature was the manipulated variable and both the controller and the model were replaced by neural networks was implemented (Fig. 3). The system was subjected to a -2% constant disturbance in the inlet feed concentration $A_i$ from $t=0$ to $t=20$ min. The dual neural network controller was unable to handle the disturbance (Fig. 4), contrary to the conventional IMC control scheme where the operator describing the system's behavior (Eqn. 1-3) was inverted using Newton's method. The inferior performance of the neural network controller was due to the fact that the network used to represent the inverse process dynamics was not the exact inverse of the process model network.

---

2.2 Dual neural network controller

An IMC type control scheme where the inlet feed temperature was the manipulated variable and both the controller and the model were replaced by neural networks was implemented (Fig. 3). The system was subjected to a -2% constant disturbance in the inlet feed concentration $A_i$ from $t=0$ to $t=20$ min. The dual neural network controller was unable to handle the disturbance (Fig. 4), contrary to the conventional IMC control scheme where the operator describing the system's behavior (Eqn. 1-3) was inverted using Newton's method. The inferior performance of the neural network controller was due to the fact that the network used to represent the inverse process dynamics was not the exact inverse of the process model network.

---

Figure 1. Non-isothermal CSTR

Figure 2. Model network’s performance on test data (dotted line : network’s prediction, solid line : actual value)

Figure 3. IMC structure

Figure 4a. Inlet feed temperature vs time (dual network controller; dotted line : setpoint)
2.3 Inversion of the process model network

One solution to the problem posed by having an inaccurate inverse process model is to directly invert the network describing the process model. Inversion is simplified by noting that the network representing the forward model implicitly contains the Jacobian of the process [9]; i.e., one can find the derivative of the network’s output with respect to the network’s inputs. Newton’s method was used to solve the nonlinear operator equation

\[ f(u_n, \hat{x}_n) - \bar{x}_{2,t+1} = 0 \]

where the goal was to find the value of the manipulated variable \( u_n \) that drives the system from the current state \( \hat{x}_n \) to the target value of the controlled variable \( \bar{x}_{2,t+1} \). A sequence of manipulated inputs can be generated through the following recursive relationship

\[ u_{n+1}^* = u_n^* - \frac{f(u_n, \hat{x}_n) - \bar{x}_{2,t+1}}{\partial f(u_n, \hat{x}_n) / \partial u_n} \]

where \( f() \) is the nonlinear mapping relating the network’s inputs to its outputs.

The described method provided a rigorous way of constructing the inverse and was tested on the same control problem. The system was subjected to a constant disturbance of -2% on the inlet reactant concentration from \( t = 0 \) to \( t = 20 \) min. This neural network controller showed very good disturbance rejection behavior and zero steady state offset (Fig. 5); its performance was thus comparable to the conventional approach with the additional advantage of less computation required.

3. Training using only historical data

In the previous development it was postulated that all state variables were measurable and thus available. This assumption is often unrealistic; usually the only data available are input-output process measurements. The objective of this section is to evaluate the performance of an IMC-type controller under such less restrictive assumptions.

A historical database of input-output measurements was already available by the method described in the previous section. A network was trained to represent the process dynamics in the following way: A point in the database was chosen as the center of a window. The inputs to the network were the current and previous values of the controlled and manipulated variables (within that window), and the target was the value of the controlled output at the next sampling interval. This process was repeated by moving the window through the database, either randomly or sequentially. It is evident that the learning task in this case was more difficult, and the network
required approximately 1500 passes through the dataset in order to converge (compared to roughly 100 in the previous section).

With this network used as the process model, an IMC neural network controller as described in section 2.3 was implemented and tested on the same control problem (-2% disturbance on the inlet feed concentration from t=0 to 20 min). The controller exhibited good disturbance rejection behavior and zero steady state offset (Fig. 6) as before, when all state variables were measured. Thus the effect of reducing the amount of information available to the controller did not seem to affect its performance; the only (qualitative) difference was the increased oscillation of the manipulated input during the transient parts of the process operation.

The MPC law follows from the solution of the following optimization problem

\[
\min \sum_{i=1}^{N} q_i \left( y_i^{\text{act}} - y_i^{\text{ref}} \right)^2
\]

\[
u_j = \begin{cases} u_j+1, & j=M,...,N-1 \text{ for } N > M \\ u_j, & j=1,...,M \end{cases}
\]

\[y_{\text{min}} \leq y_i^{\text{act}} \leq y_{\text{max}}, \quad i=1,2,...,N\]

\[u_{\text{min}} \leq u_i \leq u_{\text{max}}, \quad i=1,2,...,N\]

\[|u_i - \Delta u| \leq \Delta u_{\text{max}}, \quad i=1,2,...,M-1\]

where the sum of the squared deviations of the model predictions \(y^{\text{act}}\) from some prespecified setpoint \(y^{\text{ref}}\) over a horizon of \(N\) future sampling intervals is minimized. The decision variables are the control moves over a manipulated input horizon \(M\) (in general not equal to \(N\)) and are kept constant for the remaining sampling intervals (the first set of constraints in the above formulation). The formulation involves constraints on output and input (the second and third set of constraints respectively) and also bounds on the allowed control moves between sampling intervals (the fourth set of constraints).

The neural network developed in section 3, which used only historical input-output data to predict the future value of the controlled output was used as the process model. Prediction of the controlled output over the horizon of \(N\) future sampling times was achieved by iterating the network. The gradients of the objective function with respect to the decision variables (control moves) were also readily available as a result of the discussion in section 2.2. The optimization problem was solved by using successive quadratic programming [3]; a controlled output horizon of six time steps and a manipulated input horizon of three time steps were chosen. The lower and upper bounds on the control variables were 420 and 435 K respectively, and the maximum allowable change was 1 K.

4.1 Extended MPC with feedback

The control problem was the same as in section 3, to allow direct comparison of the performance of the two control techniques. The resulting MPC controller's behavior was poor and exhibited steady state offset. The reason was the inherent modeling error involved in training a network to represent the process dynamics; as reported elsewhere [2], the MPC's performance is quite sensitive to modeling errors. In order to amend the situation, the objective function was reformulated in the following way:

\[
\min \sum_{i=1}^{N} q_i \left( y_i^{\text{act}} - (y_i^{\text{ref}} + d) \right)^2
\]
where $d$ was the difference between the actual value of the controlled output at the current sampling instant and that predicted by the model; it was assumed that this value was constant over the optimization horizon $N$. This correction is in a sense an extension of the IMC design and accounts for the modeling errors and unmeasured disturbances entering the process. The controller's behavior in this case is shown in Fig. 7; as can be seen, including the feedback yields superior performance over the previous case and gives practically zero steady state offset.

Some interesting results can be drawn by comparing the behavior of the IMC and MPC controllers (Fig. 6 and 7 respectively). The qualitative behavior is similar, but for the MPC controller the inlet feed temperature reaches its upper bound for the duration of the disturbance; this was expected, since only by increasing the feed temperature as much as possible is the objective function minimized. On the other hand, the MPC controller results in a much smoother behavior of the manipulated input compared to the IMC method and no oscillations are observed. The disadvantage of the MPC method is the increased computation needed, although a good initial guess for the decision variables facilitates convergence.

5. Discussion

Two nonlinear control strategies, IMC and MPC, incorporating neural networks were developed and tested on the control of a SISO non-isothermal CSTR. An IMC scheme using networks for both forward and inverse process model failed because the learned inverse model network was not the exact inverse of the learned forward model network. When the exact inverse was calculated the neural controller performed as well as the conventional one. A neural network IMC-type controller where the process model was learned using only historical input-output data also gave very good performance. The latter process model was incorporated in an MPC scheme appropriately extended to account for modeling errors and unmeasured disturbances, and the performance of the two controllers was compared. Both worked very well, but which is best may depend on the specific problem at hand.

Many issues of incorporating neural networks in model based control remain unexplored. We are currently investigating possible areas of improvement of the controller's performance, including the use of networks to forecast future disturbances. This may lead to improved performance, especially in situations where the disturbance affects the future control moves in a nonlinear fashion, thus rendering the constant additive disturbance concept inefficient. Another area of investigation is the development of alternative network architectures in order to produce more accurate process models and the incorporation of approximately known models. However, these first results indicate that neural networks can learn accurate models from historical input-output data and perform well when incorporated in nonlinear control schemes.

Acknowledgements

This paper also appears at the Proceedings of the Fifth IEEE International Symposium on Intelligent Control. We wish to acknowledge the financial support of the NSF Presidential Young Investigator Award CBT 86-57899.

References