Fifth IFSA World Congress (1993), 899-902

Neural Model Predictive Control for Nonlinear Chemical Processes

Jeong Jun Song and Sunwon Park†
Process Systems Laboratory, Dept. of Chem. Eng.,
Korea Advanced Institute of Science & Technology
373-1 Kusong-dong, Yusung-gu, Taejon 305-701, Korea
Phone:+82-42-869-3920 , Fax:+82-42-869-3910

Abstract

A neural model predictive control strategy combining a neural network for plant identification and a nonlinear programming algorithm for solving nonlinear control problems is proposed. A constrained nonlinear optimization approach using successive quadratic programming combined with neural identification network is used to generate the optimum control law for complex continuous chemical reactor systems that have inherent nonlinear dynamics. The neural model predictive controller (NMPC) shows good performances and robustness.

† To whom all correspondence should be addressed.

Introduction

Since the landmark paper of Cutler and Ramaker [4], the model predictive control (MPC) techniques have been continuously developed and applied to process industries. Early versions of them are MAC (Model Algorithmic Control [24,25]), DMC (Dynamic Matrix Control [4]), IMC (Internal Model Control [8]), etc. Though the above technologies differ from each other in detail, they have some similarities and they can be categorized as MPC. The complete review of MPC is given in Garcia *et al.* [9] MPC is a multi step predictor method and has been shown to perform well in controlling processes in unstable operating regions. In addition, both input and output constraints are explicitly included in the MPC controller's design, thus facilitating the solution of more physically realistic control problems and allowing process operation in complex state space regions. One of the major issues on application of MPC is *how to get the plant model of the controlled process*. Chemical processes are inherently nonlinear and their complicated dynamics often prohibit the successful implementation of MPC in chemical industries. In this situation, the use of a neural network may be considered as a reasonable alternative for solving problems in modeling of those complex dynamics. Neural networks provide a unique computing architecture whose potential has been broadly tested and have been used to address problems that are intractable or cumbersome with traditional methods. Neural networks are massively parallel systems that rely on dense arrangements of interconnections and surprisingly simple processors. [5] In this paper, we develop a neural model predictive control (NMPC) for handling the model uncertainty and nonlinearity problems in chemical process control. Based on system identification techniques using neural network, we build a nonlinear model predictor for model predictive control. Using this *neuro-model predictor*, the constrained nonlinear programming algorithm generates optimal control input for a given process. The developed control algorithm is tested on chemical reactor systems and it shows good performances on nonlinear, model-plant mismatch, and even time varying cases.

Process Identification using Neural Network

Identification using a neural network is a problem of *nonlinear mapping* between an input and an output space. It is equivalent to the problem of synthesizing an associative memory that retrieves the appropriate output when presented with the input and generalizes when presented with new inputs. [22] For identification of nonlinear, uncertain processes, we use the neural network as a nonlinear *function approximator*. After training, the neural network learns the forward system dynamics, and it behaves as the nonlinear process. A learning dataset for the neural network can be constructed with randomly varying manipulative inputs within the desired operating range and the process outputs according to those random manipulative inputs. Thus the trained neural network presents the open loop nonlinear input-output mapping relations of a given process. It should be noted that to use the neural network as a function approximator, the manipulative inputs should be generated as random patterns or pseudo random binary sequence (PRBS) signal patterns, so that the neural network can learn the input-output relations of the process within the full range of the operating domain. This discriminates our identification neural network from the conventional model learning procedures that

use the fixed trajectory of manipulated inputs or controlled output in some robotics or mechanics fields. Good examples of training neural networks can be found in previous works. [1,21,10,23] Another one is that for function approximation, scaling of the learning dataset should be required since the activation function typically used in the neural network ranges between 0 and 1 (when standard sigmoid function is used) or between -1 and 1 (when hyperbolic tangent like function is used). We use a feedforward network with modified hyperbolic tangent [15] as activation functions and natural logarithm for scaling function for the training data. The network is trained using error back-propagation algorithm with Rumelhart's momentum method [26], and initial distribution of the network's weights is done by random value generating between -1 and 1. The network consists of one input layer, one hidden and one output layer. The input nodes consist of measurable process states or process output variables and manipulated variables at each time step t=k. The teaching patterns are the state variables in the next time step t=k+1. When the current values of the process state variables and manipulative variables are presented to the network, the network generates the predicted values of the state variables at the next sampling instant. Thus the proposed neural identification network acts like the one step ahead predictor of the process output. This one step ahead predictor is *iteratively* used for generating multi-step prediction of process outputs. Though, at the learning stage of neural network, it is possible to construct multi step predictor, the one step predictor involves less error in its predictions. [14] The structure of the neural network is shown in Figure 1.

Control System Design

Strong nonlinearity and complicated dynamics of chemical processes presents challenging control problems and currently many researchers are developing control techniques that are based on nonlinear systems concepts. Mainly those approaches are classified as *internal model approaches* by Economou *et al.* [6,7], which are extensions of internal model control to a nonlinear system (NLIMC) using operator theory, *differential geometric approaches* by Kravaris and Kantor [12,13], and *predictive control approaches* by Li *et al.* [16,17,18,19], Bequette [3], Sistu and Bequette [27] and many other researchers. An extensive review on nonlinear control of chemical processes are done by Bequette. [2] According to Bequette, predictive control strategies have been well received by industry because they are intuitive and explicitly handle constrains. One limitation to the existing method is that they are based on linear systems theory and may not perform well on highly nonlinear systems. In this paper, we use the nonlinear predictive control approach for generating optimum manipulated input at each sampling time. The successive quadratic programming (SQP) code is used for solving optimization problem in cooperation with neural network model described in the previous section. The objective of nonlinear predictive control is to select a set of future control moves (control horizon, M) to minimize a cost function based on a desired output trajectory over a prediction horizon (P). The optimization problem at sampling time $t = t_k$ is setup as follows:

$$\min \Phi(u) = \int_{t_k}^{t_k+T_p} e^2 dt = \sum_{i=k+1}^{k+P} [y_{sp}(i) - y_{pred}(i)]^2 \quad (1)$$

$$u(k), \ldots, u(k+M-1)$$

subject to

$$x^{\bullet} = f(x,u,p,t) \quad ; \textit{dynamic model constraints} \quad (1\text{-}1)$$

$$y_m = g(x) \quad ; \textit{model outputs that are functions of the state variables} \quad (1\text{-}2)$$

Although the optimization is based on the control horizon, only the first control action is implemented. After the first control action is implemented, plant output measurements are obtained. Compensation of model-plant mismatch is performed, and the optimization is performed again. Since the model-predicted output y_m is not exactly equal to the process output at the end of the time horizon, an additive output disturbance may be included after the process identification is performed. [2] The compensation of model-plant mismatch is as follows:

- the additive output disturbance $d(k)$ is calculated as

$$d(k) = y(k) - y_m(k) \quad (2)$$

where $y(k)$ is the actual plant output and $y_m(k)$ is the model prediction value.

- the corrected model prediction is calculated as

$$y_{pred}(k+i) = y_m(k+i) + d(k) \text{ , for } i = 1 \text{ to } P \quad (3)$$

where P is the prediction horizon of the controller.

The two major problems to be solved for applying nonlinear predictive control technique are *the choice of constrained optimization technique* to solve the equation (1) and *how to solve the dynamic model constraints (1-1)*. If we have a precise plant model, we can solve the dynamic model constraints using several methods such as sequential, simultaneous, intermediate, or linear approximation techniques. [2] But how can we get the dynamic model for highly nonlinear processes? One possible answer to this question is the use of neural network to model the forward dynamics of the process. In this paper, we use the neural network discussed in previous section as a forward process model. There are several important benefits for using neural network as a process model. Practically, the elimination of integration of ordinary differential equations greatly reduces the overall computing time of nonlinear controllers. Second, we can train the network within wide operating range, that

enhances the interpolation capability of the neural process model. Third, we can get the process output predictions based on the actual process open loop characteristics and can get more realistic information compared with the use of ill-posed material and/or energy balance model. Several optimization algorithms can be used for generating optimum control input such as GRG2, SQP, QP, and even unconstrained nonlinear algorithms. Among them, the constrained nonlinear optimization algorithm group has eminent benefits for explicit handling of the constraints on manipulated or output variables, fast convergence, and numerical stability. The previous work [29] deals the problem with unconstrained optimization algorithm and penalty cost function. The internal signal communications between SQP and neural network are shown in Figure 2.

Application Example

The CSTR model in this example is a part of a larger test system introduced by Williams and Otto. [20,28] The CSTR system supports the following multiple reactions:

$A + B \xrightarrow{k1} C$,

$C + B \xrightarrow{k2} P + E$, $P + C \xrightarrow{k3} G$

The desired product is P, while G, C, and E are byproducts subject to quality and environmental constraints. Reactants A and B enter as pure components in separate streams with flow rates *Fai* and *Fbi*, respectively. The manipulated variable is the flow rate *Fai* and cooling water temperature T. The input stream *Fbi* is considered to be a disturbance variable. The equations describing the kinetic behavior of the above reactions and the dynamic mass balance for the CSTR represent a coupled set of nonlinear algebraic and ordinary differential equations. A description of these equations has been provided by Williams and Otto:

$$dXa/dt = Fai/Fr - rx1 - Xa \quad \text{where, } Fr = Fai + Fbi$$

$$dXb/dt = Fbi/Fr - rx1 - rx2 - Xb$$

$$dXc/dt = 2rx1 - 2rx2 - rx3 - Xc \,, \quad dXe/dt = 2rx2 - Xe$$

$$dXg/dt = 1.5rx3 - Xg \,, \quad dXp/dt = rx2 - 0.5rx3 - Xp$$

$$rx1 = 5.9755E9 \exp(-12000/T) XaXb\rho V/(60Fr)$$

$$rx2 = 2.5962E12 \exp(-15000/T) XbXc\rho V/(60Fr)$$

$$rx3 = 9.6283E15 \exp(-20000/T) XcXp\rho V/(60Fr)$$

The control objective of this system is to maximize the yield of the desired product P using regulation of two related state variables Xc and Xg. In this example, our neural network has a 5-8-3 morphology and the input layer consists of two state variables (Xc, Xg) and two manipulated variables (*Fai*, T), and one bias node. The bias node has a value +1. The one step ahead prediction performance of the trained neural network model is shown in Figure 3. We set the prediction horizon P=1 in this example. The control performance is shown in Figure 4. In these figures, the control performance of Xc seems better than that of Xg. In MIMO systems, these kinds of interactions are rather natural and there is a trade-off for improving the control performance among coupled controlled variables. The controller designer can handle this problem by imposing proper auxiliary constraints to desired controlled variable in the optimization formulation.

Conclusion

The neural model predictive control strategy using neural network and constrained nonlinear optimization technique is discussed and tested by controlling a SISO and a MIMO CSTR system. The use of neural network for system identification is a feasible alternative when model equations are not known or only historical input-output data are available. The features of the neural model predictive control (NMPC) are:

- It combines the *mapping capability* of a neural network with the *nonlinear optimization* technique for controlling nonlinear processes that have model uncertainty, hard constraints on manipulative or controlled variables.
- It learns the uncertain process dynamics using *only available* plant states through *open-loop trends* with random signal or *existing control history*
- Appropriate selection of the topology of a neural network can compensate the insufficient information and can increase mapping capability
- The optimizing control inputs can be calculated fast because the equation solving step required in MPC does not exist in NMPC.

The NMPC shows good control performance for processes with nonlinearities and model-plant mismatch.

Literature Cited

1) Bhat, N. and T.J. McAvoy, *Computers chem. Engng*, **14**(4/5), 573-582 (1990).

2) Bequette, B.W., *Ind. Eng. Chem. Res.*, **30**, 1391-1413 (1991a).

3) Bequette, B.W., *The Canadian Journal of Chemical Engineering*, **69**, 136-143 (1991b).

4) Cutler, C.R. and B.L. Ramaker, *AIChE 86th National Meeting*, paper No. 51B, Houston, TX, April (1979).

5) Dayhoff, J.E., *Neural Network Architectures*, Van Nostrand Reinhold, (1990).

6) Economou, C.G. and M. Morari, *Proceedings of the 24th conference on Decision and Control*, Ft. Lauderdale, FL, 1361-1366 (1985).

7) Economou, C.G., M. Morari, and B.O. Palsson, *Ind. Eng. Chem. Process Des. Dev.*, **25**, 403-411 (1986).

8) Garcia, C.E. and M. Morari, *Ind. Eng. Chem. process. Dev.*, **21**(2), 308-323 (1982).

9) Garcia, C.E., D.M. Prett, and M. Morari, *Automatica*, **25**(3), 335-348 (1989).

10) Hernandez, E. and Y. Arkun, *Computers chem. Engng*, **16**(4), 227-240 (1992).

11) Kravaris, C. & Raymond A. Wright, *AIChEJ.* **35**(9), 1535-1542 (1989).

12) Kravaris, C. and J.C. Kantor, *Ind. Eng. Chem. Res.*, **29**, 2295-2310 (1990a).

13) Kravaris, C. and J.C. Kantor, *Ind. Eng. Chem. Res.*, **29**, 2310-2324 (1990b).
14) Lapeds, A. and R. Farber, *Los Alamos National Laboratory Technical Report LA-UR-87-2662*, July (1987).
15) Lee, Y.S. and K.B. Woo, *Trans. KIEE*, **39**(8), 856-867, in Korean (1990).
16) Li, W.C. and L.T. Biegler, *Ind. Eng. Chem. Res. Des.*, **27**, 1421-1433 (1988).
17) Li, W.C. and L.T. Biegler, *Chem. Eng. Res. Des.*, **67**, 562-577 (1989).
18) Li, W.C. and L.T. Biegler, *Ind. Eng. Chem. Res.*, **29**, 1647-1657 (1990a).
19) Li, W.C., Biegler L.T., Economou C.G., and M. Morari, *Computers chem. Engng*, **14**, 451-465 (1990b).
20) McFarlane, C.R. & D.W. Bacon, *Ind. Eng. Chem. Res.*, **28**, 1828-1845 (1989).
21) Psichogios, D.C. and L.H. Ungar, *Ind. Eng. Chem. Res.*, **30**, 2564-2573 (1991).
22) Poggio, T. and F. Girosi, *Proc. IEEE*, **78**(9), 1481-1497 (1990).
23) Pollard, J.F. *et al.*, *Computers chem. Engng*, **16**(4), 253-270 (1992).
24) Richalet, J.A., A. Rault, J.L. Testud, and J. Panton, *Automatica*, **14**, 413-428 (1978).
25) Rouhani, R. and R.K. Mehra, *Automatica*, **18**(4), 401-414 (1982).
26) Rumelhart, D.E., Hinton, G.E., and Williams, R.J., *Parallel Distributed Processing*, **1**, 318, Cambridge, MA: MIT Press (1986).
27) Sistu P.B. and B.W. Bequette, *AIChEJ.* **37**(11), 1711-1723 (1991).
28) Williams T.I. and R.E. Otto, *Trans. Am. Inst. Elect. Engr.*, 79, 458-473 (1960).
29) Willis, M.J. *et al.*, *IEE Proceedings-D*, 138(3), 256-266 (1991).

Fig.1 Topology of the neural network used in system identification

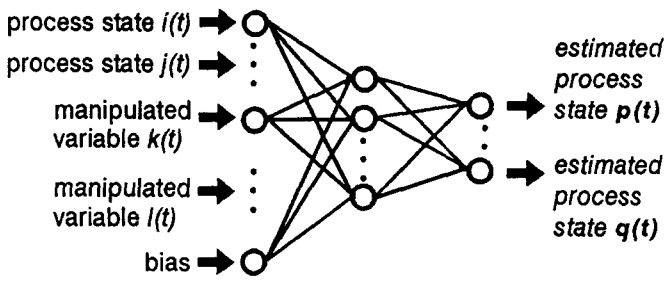
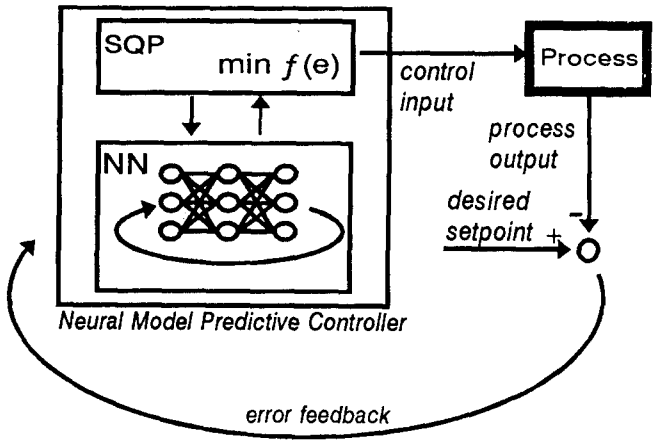


Fig.2 Neural Model Predictive Control (NMPC) Structure



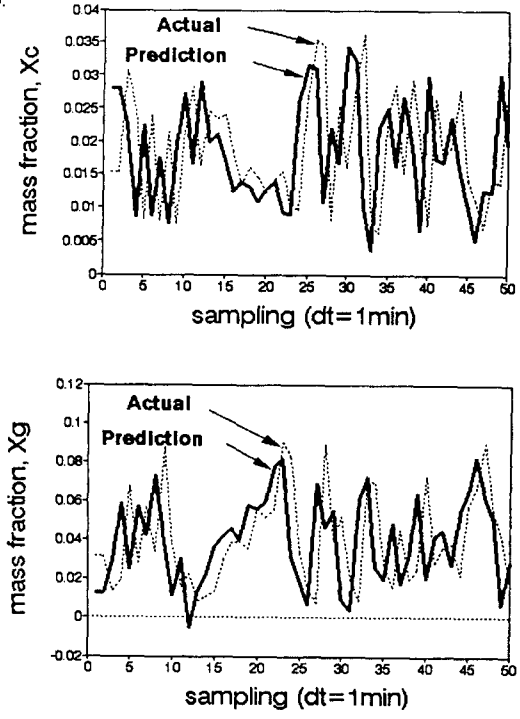


Fig.3 One step ahead prediction (Xc and Xg)

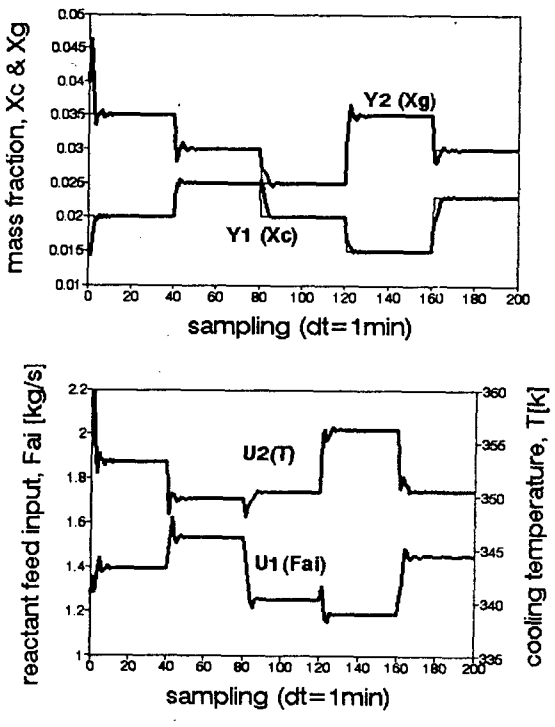


Fig.4 Control performance of NMPC (process output & control action)