NONLINEAR MODEL PREDICTIVE CONTROL USING A NEURAL NETWORK

Ridong Zhang Ping Li

(School of Information Engineering, Liaoning University of Petroleum & Chemical Technology, Liaoning, Fushun 113001, P.R.China)

Abstract A neural network model-based generalized predictive control for a class of nonlinear discrete-time systems is presented with the local linearization of nonlinear activation function. The method converts the nonlinear multi-step predictions into a series of local linear multi-step predictions and uses linear GPC method to gain the control law. The method avoids the shortcomings of some past predictive algorithms , it doesn't need any assumptions and give a direct and effective multi-step predictive method. It also avoids the complicated nonlinear optimization and computation burden is not serious. A simulation result is presented in the article.

Keywords: neural-network models; generalized predictive control; nonlinearity; linearization; adaptive control.

1. INTRODUCTION

Generalized Predictive Control (Clarke D W et al.,1987) has been greatly used in the control of many industrial processes because of its excellent control performance and robustness due to its three basic features: predictive model, feedback correction and rolling optimization. However, for a nonlinear system, it is not easy to apply GPC because of the difficulty of getting an accurate nonlinear model. Since the mid-1980s, neural networks have been internationally studied to model and control nonlinear systems, and there are more and more neural network based predictive control algorithms, too. K Chao-Chee and Y L Kang(1995) presented a diagonal recurrent neural network based control strategy for dynamic systems. Among the results of nonlinear predictive control, an analytical predictive control law was presented (Furong Gao et al., 2000). A control strategy based on two assumptions was presented (Jian Guo et al., 2001). Two neural networks with an algorithm using the reverse dynamic technique was given (Qibing Jin et al., 1999), it avoids the nonlinear optimization, however, both the two networks need training and therefore its algorithm is complicated. Saint Donat J et al (1991) presented a neural net based model predictive control algorithm. A neural network predictive control strategy assuming that the process can be described as a linear part plus a nonlinear part was given (Yupu Yang et al., 1999), it uses a dynamic recurrent network to model the two parts of the system.There is also an algorithm using a global linearized model (Jun Liu et al., 2000).

The above results show that it needs to solve the following problems when applying neural networks based predictive control algorithms to nonlinear systems: (1) give a direct and effective method of multi-step prediction. (2) try to avoid the complicated nonlinear optimization. (3) reduce the sum of neural networks so as to cut down computation burden.

In this paper, using the local linearization of nonlinear activation function, a new control strategy is presented. The method converts the nonlinear multi-step predictions into a series of local linear multi-step predictions and uses linear GPC method to gain the control law. A simulation result is also given in the paper, evidencing that the controller presents a fairly good performance.

2. NONLINEAR SYSTEMS AND THEIR REPRESENTATION

Consider the following SISO nonlinear discrete-time system described by the following model:

$$y(t) = f(y(t-1),..., y(t-n), u(t-1),..., u(t-m)) \quad (1)$$

where n, m are the orders of its output and input respectively.

The system can be described by a three-layer BP neural network as follows:

$$y(t) = g\{\sum_{i=1}^{I} w3(i) g[\sum_{j=1}^{m} w2(i,j)u(t-j) + \sum_{j=m+1}^{m+n} w2(i,j)y(t+m-j)]\} \quad (2)$$

where $w3(i), w2(i,j)$ $(i=1,...,I \;\; j=1,...,m+n)$ are the linking weights, $n+m$ is the sum of input nodes, I the sum of hidden nodes and there is one output node. And "g" is the activation function:

$$g(x) = \frac{1}{1+e^{-x}} \quad (3)$$

In order to get a multi-step predictor, the following method is used:

Let

$$s_3(t) = \sum_{i=1}^{I} w3(i) g[\sum_{j=1}^{m} w2(i,j)u(t-j) + \sum_{j=m+1}^{m+n} w2(i,j)y(t+m-j)] \quad (4)$$

$$s_{2i}(t) = \sum_{j=1}^{m} w2(i,j)u(t-j) + \sum_{j=m+1}^{m+n} w2(i,j)y(t+m-j) \quad (5)$$

where $i=1,...,I$ .And then $y(t) = g[s_3(t)]$ **,and** $s_3(t) = \sum_{i=1}^{I} w3(i) g[s_{2i}(t)]$ .Using the Taylor expansion technique :

$$\begin{aligned} y(t) &= g(s_{31}) + g'(s_{31})[s_3(t) - s_{31}] + F_1(\theta(t)) \\ &= g'(s_{31})s_3(t) + g(s_{31}) - g'(s_{31})s_{31} + F_1(\theta(t)) \\ &= g'(s_{31})\sum_{i=1}^{I} w3(i) g[s_{2i}(t)] \\ &\quad + g(s_{31}) - g'(s_{31})s_{31} + F_1(\theta(t)) \end{aligned} \quad (6)$$

where $\theta(t)$ refers to $[y(t-1),..., y(t-n), u(t-1),..., u(t-m)]$, F_1 is a function symbol, and s_{31} is the center of the expansion . Also define $\theta(t+i)$ as: $[y(t+i-1),..., y(t+i-n), u(t+i-1),..., u(t+i-m)]$. In general, let the center s_{31} =0. The same technique can also be employed on $s_{2i}(t)$, likewise , it derives:

$$g[s_{2i}(t)] = g(s_{2i}) + g'(s_{2i})[s_{2i}(t) - s_{2i}] + F_{2i}(\theta(t)) \quad (7)$$

where s_{2i} $(i=1,...,I)$ are also the centers , also let s_{2i} =0 $(i=1,...,I)$,and F_{2i} $(i=1,...,I)$ are function symbols. Substitute eq.(7) into eq.(6) and combine the nonlinear parts $F_1(\theta(t))$ and $F_{2i}(\theta(t))$ $(i=1,...,I)$ into one nonlinear part $F_3(\theta(t))$ leads to the following:

$$\begin{aligned} y(t) &= \sum_{i=1}^{I} w3(i) g'(s_{31})\{g(s_{2i}) + g'(s_{2i})[s_{2i}(t) - s_{2i}]\} \\ &\quad + g(s_{31}) - g'(s_{31})s_{31} + F_3(\theta(t)) \\ &= \sum_{i=1}^{I} w3(i) g'(s_{31}) g'(s_{2i}) s_{2i}(t) \\ &\quad + \{\sum_{i=1}^{I} w3(i) g'(s_{31})[g(s_{2i}) - g'(s_{2i})s_{2i}] \\ &\quad + g(s_{31}) - g'(s_{31})s_{31}\} + F_3(\theta(t)) \end{aligned} \quad (8)$$

where $F_3(\theta(t))$ is the nonlinear part:

$$F_3(\theta(t)) = \sum_{i=1}^{I} w3(i) g'(s_{31}) F_{2i}(\theta(t)) + F_1(\theta(t)) \quad (9)$$

For simplicity , let $M_i = w3(i) g'(s_{31}) g'(s_{2i})$ $(i=1,...,I)$ **, and** represent the second part in eq.(8) as N , substitute eq.(5) into eq.(8) and gives:

$$y(t) = \sum_{i=1}^{I} M_i s_{2i}(t) + N + F_3(\boldsymbol{\theta}(t))$$

$$= \sum_{i=1}^{I} M_i [\sum_{j=1}^{m} w2(i, j)u(t-j) \quad (10)$$

$$+ \sum_{j=m+1}^{m+n} w2(i, j) y(t-j+m)] + N + F_3(\boldsymbol{\theta}(t))$$

The discrete differential equation of $y(t)$ can be written as:

$$\begin{aligned} y(t) &= a_1 y(t-1) + ... + a_n y(t-n) \\ &+ b_0 u(t-1) + ... + b_{m-1} u(t-m) \\ &+ N + F_3(\boldsymbol{\theta}(t)) \end{aligned} \quad (11)$$

Compare eq.(10) with eq.(11) leads to:

$$a_1 = \sum_{i=1}^{I} M_i w2(i, m+1) \quad b_0 = \sum_{i=1}^{I} M_i w2(i,1)$$

$$a_2 = \sum_{i=1}^{I} M_i w2(i, m+2) \quad b_1 = \sum_{i=1}^{I} M_i w2(i,2)$$

$$\vdots$$

$$a_n = \sum_{i=1}^{I} M_i w2(i, m+n) \quad b_{m-1} = \sum_{i=1}^{I} M_i w2(i, m) \quad (12)$$

Now the system model has been divided into two parts: a linear part and a nonlinear part. The coefficients of the linear part are calculated by eq.(12).

3. CONTROL SYSTEM DESIGN

Note that N is a constant, eq.(11)can be written as follows:

$$\begin{aligned} y(t) &= A_1 y(t-1) + ... + A_{n+1} y(t-n-1) \\ &+ B_{1,0} \Delta u(t-1) + ... + B_{1,m-1} \Delta u(t-m) \\ &+ \Delta F_3(\boldsymbol{\theta}(t)) \end{aligned} \quad (13)$$

where:

$$A_1 = 1 + a_1, A_i = a_i - a_{i-1} (i = 2,..., n), A_{n+1} = -a_n,$$

$B_{1,i} = b_i \ (i = 0,..., m-1)$, and D is the differencing operator $1 - q^{-1}$.

Now, divide the optimal predictions into three parts, one is determined by past inputs and outputs, this is represented by $\boldsymbol{Y_p}$, another is determined by present and future inputs, it is represented by $\boldsymbol{GU}$, the other is the prediction error, it consists of the nonlinear error $\boldsymbol{E_1}$ and the error caused by external disturbances $\boldsymbol{E_2}$, then it derives:

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{Y_p} + \boldsymbol{GU} + \boldsymbol{E} \\ &= \boldsymbol{Y_p} + \boldsymbol{GU} + \boldsymbol{E_1} + \boldsymbol{E_2} \end{aligned} \quad (14)$$

where:

$$\boldsymbol{Y} = [y(t+1/t), y(t+2/t),..., y(t+p/t)]^T$$
$$\boldsymbol{Y_p} = [y_p(t+1), y_p(t+2),..., y_p(t+p)]^T$$
$$\boldsymbol{U} = [\Delta u(t), \Delta u(t+1),..., \Delta u(t+p-1)]^T$$
$$\boldsymbol{E}_1 = [\varepsilon(\boldsymbol{\theta}(t+1)), \varepsilon(\boldsymbol{\theta}(t+2)),..., \varepsilon(\boldsymbol{\theta}(t+p))]^T$$
$$\boldsymbol{E_2} = [y(t) - \hat{y}(t), y(t) - \hat{y}(t),..., y(t) - \hat{y}(t)]^T$$

$$\boldsymbol{G} = \begin{bmatrix} B_{1,0} & & & \\ B_{2,0} & B_{1,0} & 0 & \\ ... & ... & & \\ B_{p,0} & B_{p-1,0} & ... & B_{1,0} \end{bmatrix} \quad (15)$$

Here the control horizon and prediction horizon are both p, $y(t)$ is the output of the system, $\overset{L}{y}(t)$ is the output of its neural network model.

Since $\boldsymbol{Y_p}$ is the "free response" of the system, it can be calculated by the neural network model. The elements in $\boldsymbol{G}$ are calculated as follows:

$$B_{1,0} = b_0$$
$$B_{k,0} = b_{k-1} + \sum_{j=1}^{k-1} A_j B_{k-j,0}, k = 2,...,p \quad (16)$$

Moreover, let the reference trajectory be as follows:

$$\begin{aligned} y_r(t) &= y(t) \\ y_r(t+k) &= a^k y(t) + (1 - a^k) y_s \end{aligned} \quad (17)$$

where y_s is the set-point, $a \in (0,1)$.

Let the vector form of the reference trajectory and the cost function be the follows respectively:

$$\boldsymbol{Y_r} = [y_r(t+1), y_r(t+2),..., y_r(t+p)]^T \qquad (18)$$

$$J = \min\{(\boldsymbol{Y_r} - \boldsymbol{Y})^T(\boldsymbol{Y_r} - \boldsymbol{Y}) + \beta^2 \boldsymbol{U}^T \boldsymbol{U}\}$$

where b^2 is the weighting factor.

Note that because the system is nonlinear and $\boldsymbol{E_1}$ is function of $\boldsymbol{\Theta}$, here $\boldsymbol{\Theta}$ refers to $[\varepsilon(\theta(t+1)),...,\varepsilon(\theta(t+p))]^T$, so $\boldsymbol{E}_1$ is not known. So the control law cannot be calculated by eq.(14).

However, the following method is used to get the control law:

First, let $\boldsymbol{E_{10}}$ =0, where $\boldsymbol{E_{10}}$ is the initial value of $\boldsymbol{E}_1$, then, from $\frac{\partial J}{\partial \boldsymbol{U}} = 0$, $\boldsymbol{U_0}$ can be calculated:

$$\boldsymbol{U_0} = (\boldsymbol{G}^T\boldsymbol{G} + \beta^2 \boldsymbol{I})^{-1}\boldsymbol{G}^T(\boldsymbol{Y_r} - \boldsymbol{Y_p} - \boldsymbol{E_2} - \boldsymbol{E_{10}}) \qquad (19)$$

And $\boldsymbol{U_0}$ is the initial value of $\boldsymbol{U}$. Define the vector form of the multi-step predictions based on neural network model as $\boldsymbol{Y_m}$:

$$\boldsymbol{Y_m} = [y_m(t+1/t), y_m(t+2/t),..., y_m(t+p/t)]^T$$

And its value of step j is $\boldsymbol{Y_{mj}}$, the value of $\boldsymbol{Y}$ of step j is $\boldsymbol{Y_j}$, the value of $\boldsymbol{U}$ of step j is $\boldsymbol{U_j}$, the value of $\boldsymbol{E_1}$ of step j is $\boldsymbol{E_{1j}}$. Thus the optimal $\boldsymbol{U}$ can be gained by the following method:

$$\boldsymbol{Y_j} = \boldsymbol{Y_p} + \boldsymbol{G}\boldsymbol{U_j} + \boldsymbol{E_2} + \boldsymbol{E}_{1j} \quad j = 1,2,... \qquad (20)$$

Substitute $\boldsymbol{U_j}$ into its neural network model eq.(2) and $\boldsymbol{Y_{mj}}$ can be gained, then:

$$\boldsymbol{E_{1j+1}} = F(\boldsymbol{E_{1j}}) = \boldsymbol{E_{1j}} + \delta[\boldsymbol{Y_{mj}} - \boldsymbol{Y_j}] \qquad (21)$$

$$\boldsymbol{U_{j+1}} = (\boldsymbol{G}^T\boldsymbol{G} + \beta^2 \boldsymbol{I})^{-1}\boldsymbol{G}^T(\boldsymbol{Y_r} - \boldsymbol{Y_p} - \boldsymbol{E_2} - \boldsymbol{E_{1j+1}}) \qquad (22)$$

where $\boldsymbol{E_{10}}$ =0, $d \in (0,1)$. Note that if the function $F(\bullet)$ is compressed mapping, the above iterative process is convergent. And when $\boldsymbol{Y}$ equals $\boldsymbol{Y_m}$, the control sequence $\boldsymbol{U}$ gained based on eq.(22) is the optimal one. However, if $\|\boldsymbol{E_{1j+1}} - \boldsymbol{E_{1j}}\|$ < a desired tolerance , $\boldsymbol{U_{j+1}}$ can be thought of as the optimal control law, define $\boldsymbol{q}^T$ as the first row of $(\boldsymbol{G}^T\boldsymbol{G} + \beta^2 \boldsymbol{I})^{-1}\boldsymbol{G}^T$, then the control law is:

$$u(t) = u(t-1) + \boldsymbol{q}^T(\boldsymbol{Y_r} - \boldsymbol{Y_p} - \boldsymbol{E_2} - \boldsymbol{E_{1j+1}}) \qquad (23)$$

Hence, the algorithm for the nonlinear predictive control method can be summarized as follows:

Step 1. Use BP algorithm to train the weights of the network and gain an initial estimate of their values.

Step 2. Pick the output $y(t)$ and give the network an on-line training so as to adjust the weights adaptively.

Step 3. Divide eq.(2) into two parts using the method of the second section.

Step 4. Compute the free response $\boldsymbol{Y_p}$ of the system.

Step 5. Get the reference trajectory using eq.(18).

Step 6. Gain the optimal $u(t)$ by using eq.(23).

Step 7. Return to step 2.

4. SIMULATION RESULT

In this section, an example is given to illustrate the above method. The system is represented by the following model:

$$y(t) = \frac{0.91y(t-1) + u(t-1)}{1 + y(t-2)u(t-2)} + e(t)/D$$

Also white noise was added to the output. Its amplitude is tenth of the set-point. The control parameters are selected as follows:

p =5, b^2 =1, $a = 0.65$, the system response can be seen in Fig.1.

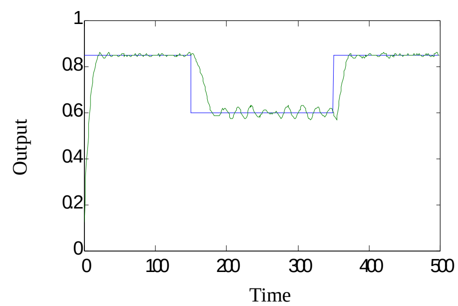

Fig. 1. Output response

5. CONCLUDING REMARKS

In this work, a new neural network based nonlinear predictive control algorithm is conducted and applied to a nonlinear system. It gives a direct and effective predictive method and avoids nonlinear optimization. In the algorithm, only one neural network is used, so the computation burden is not serious.

Acknowledgements---This research was supported by 863 program of China under Grant No. 2001AA413110.

REFERENCES

Clarke D W, Mohtadi C and Tuffs P S (1987). Generalized Predictive Control Part I. The

Basic Algorithm. *Automatica,* **Vol. 23, No. 2,** pp. 137-148

Furong Gao, Fuli Wang and Mingzhong Li (2000). An analytical predictive control law for a class of non-linear processes. *Ind.End.Chem.Res.,* **Vol. 39,** pp.2029-2034.

Jian Guo, Rijun zhu and Weili Hu (2001). Generalized Predictive Control for a Class of Nonlinear Systems. *Control and Decision*, **Vol. 16, No. 3,** pp. 358-361

Jun Liu, Xing He and Xiaoming Xu (2000). Extension of DMC Predictive Control Using Neural Network Base Nonlinear Modelsand Multi-step Operration Point Respones. *Control and Decision,* **Vol. 15, No. 3,** pp.342-344

Chao-Chee K and Kang Y L (1995). Diagonal recurrent neural networks for dynamic systems control. *IEEE trans on Neural Networks*, **Vol. 6, No. 1,** pp. 144-156

Qibing Jin, Jianhui Wang and Shusheng Gu (1999). Multi-step Prediction Cost Function based reverse-dynamic method. *Control and Decision*, **Vol. 14, No. 4,** pp.308-312

Saint Donat J, Dhat N and McAvoy T J (1991). Neural net based model predictive control. *Int J Control*, **Vol. 54, No. 6,** pp.1453-1468

Wei Wang and Jianjun Yang (1997). Generalized Predictive Control, Theory, Algorithm and Applications. *Control Theory and Applications,* **Vol. 14, No. 6,** pp.778-785

Yupu Yang, Xinmin Huang and Xiaoming Xu (1999). Nonlinear Muti-step Predictive Control Using Compound Neural Networks. *Control and Decision*, **Vol. 14, No. 4,** pp.314-318