

Dynamic Model of Heat Exchanger Based on Neural Network

Min Guo¹, Xinkui Han², ShouJun Zhou¹

¹ School of Thermal Engineering, Shandong institute of Architecture and Engineering, Jinan 250014, China

² HuanTai Power Plant, Jinan 250014, China

Abstract: Neural network is used to establish dynamic model of heat exchanger system in this paper, gray-box model is brought forward on the base of analyzing heat exchanger's operating characteristic. Matrix operations are easy to be understood, the result of prediction is accurate.

Key words: heat exchanger, Neural Network, dynamic model, BP algorithm

1. INTRODUCTION

Along with the production scale expansion and the production technology modernization, the heat exchanger have to satisfy each kind of peculiar circumstance and the special request, the heat exchanger operation principle is becoming more and more complex, displays for highly non-linear and time lag and so on. So it's difficult to describe system dynamic model with precise mathematics modelling method. However, the neural network has the strong ability of pattern recognition and can approach the highly nonlinear function map, this has brought the new method for the unknown nonlinear system dynamic modelling. In this paper, establish the heat exchanger dynamic model through connecting heat exchanger operation characteristic with NN theory, model prediction result is accurate by computer programming with VB assembler language.

2. HEAT EXCHANGER PERATION CHARACTERISIC BASED ON OPERATING POINT

In the heat exchanger system control process, the process input and the output usually change nearby operating point, generally decided by operation load. Nearby some operating point, the relation between the heat exchanger outlet temperature increment with the hot water flow rate increment may approximately be regarded as second order plus time delay

$$W(s) = \frac{\Delta T_G(s)}{\Delta G_L(s)} = \frac{K}{(1+T_c s)^2} e^{-\tau s} = \frac{K}{T_c^2 s^2 + 2T_c s + 1} e^{-\tau s} \quad (1)$$

K —static gain of system operating point

T_c —time constant of system operating point

τ —delay time of system operating point

3. PRAY_BOX DYNAMIC MODEL OF HEAT EXCHANGER BASED ON OPERATING POINT

3.1 Pray_box dynamic model structure of heat exchanger based on operating point

Using the self_learning and nonlinear function analog characteristic of static Neural Network and forecasting K and T_c in the different load of operating point, thus may recognize the heat exchanger dynamic model with linear second_order system. Because this kind of model fully used the known knowledge of heat exchanger, so it is called the pray_box dynamic model, the structure of dynamic model is depicted in Fig 1.

As shown in Fig 1, G_c is the cold water flow rate, G_L is the heat water flow rate, $T_G(k)$ is actual system output, k and T_c is heat exchanger characteristic parameter predicted by NN model, $G_L(k)$ is mean flow_rate of the heat exchanger in current period of time, $T_G(k)$ is outlet temperature average value of the heat exchanger in current period of time, $\Delta T_G(k)$ and $\Delta T_c(k)$ are actual and predicted outlet temperature increment at time k , the error of both is minimized to zero through adjusting the NN weight.

For second_order, using bilinear transform method,

$$\text{For Eq(1), let } s = \frac{2}{T} \cdot \frac{1-z^{-1}}{1+z^{-1}}, \text{ getting pulse}$$

transfer function

$$W(z) = \frac{\Delta T_G(z)}{\Delta G_L(z)} = \frac{b_0 z^{-2} + b_1 z^{-1} + b_2}{a_0 z^{-2} + a_1 z^{-1} + 1} z^{-d}$$

where, $\tau = dT$ and $d=1$, getting Eq(2) through inverse transfer

$$\begin{aligned} \Delta \hat{T}_G(k) &= -a_1 \Delta T_G(k-1) - a_0 \Delta T_G(k-2) + \\ b_2 \Delta G_L(k-1) &+ b_1 \Delta G_L(k-2) + b_0 \Delta G_L(k-3) \end{aligned}$$

$$\text{where, } b_2 = \frac{K}{A}, b_1 = 2b_2, b_0 = b_2,$$

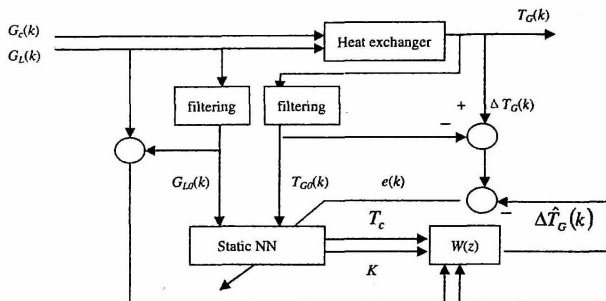


Fig.1 system structure of pray_box dynamic model

$$a_1 = \frac{2T^2 - 8T_c^2}{T^2 A},$$

$$a_0 = \frac{4T_c^2 - 4TT_c + T^2}{T^2 A},$$

$$A = \frac{4}{T^2} T_c^2 + \frac{4}{T} T_c + 1 \quad (2)$$

T is the sampling period, the relation of instantaneous value, average value and increment as follows:

$$T_G = T_{G0} + \Delta T_G, \quad G_L = G_{L0} + \Delta G_L,$$

$$\hat{T}_G = T_{G0} + \Delta \hat{T}_G$$

3.2 Pray_box dynamic model BP algorithm based on operating point

Pray_box dynamic NN model may be described by chart 2:

$$\begin{cases} x_h(k) = f_1(s_i) = \tan \operatorname{sig} \left(\sum_{i=1}^q w_{hi}^{HI} \cdot u_i(k) \right) \\ y_o(k) = \sum_{h=1}^n w_{oh}^{OH} \cdot x_h(k) \end{cases} \quad (i=1, 2 \dots q; \quad o=1, 2 \dots m;)$$

In which $u_i(k)$ stands for input signal of i th input neuron, $x_h(k)$ stands for output signal of h th hidden neuron, $y_o(k)$ stands for output signal of o th output neuron, w_{hi}^{HI} stand for weight vector from the input

layer to the hidden layer, w_{oh}^{OH} stand for weight vector from the hidden layer to the output layer. Weight adjustment algorithm as follows:

Assuming the error function

$$E_k = \frac{1}{2} (T_G(k) - \hat{T}_G(k))^2 = \frac{1}{2} e^2$$

(a) Δw_{oh} stand for weight adjustment from output layer to the hidden layer

$$\Delta w_{oh} = -\eta \cdot \frac{\partial E_k}{\partial w_{oh}}$$

$$\frac{\partial E_k}{\partial w_{oh}} = \frac{\partial E_k}{\partial \hat{T}_G} \cdot \frac{\partial \hat{T}_G}{\partial y_o^{(o)}} \cdot \frac{\partial y_o^{(o)}}{\partial s_o^{(o)}} \cdot \frac{\partial s_o^{(o)}}{\partial w_{oh}^{(o)}} = -e \cdot \frac{\partial \hat{T}_G}{\partial y_o^{(o)}} \cdot f'_1(s_o^{(o)}) \cdot x_h^{(h)} \quad (o=1, 2, \dots, m; h=1, 2, \dots, n), \quad (4)$$

(b) Δw_{hi} stand for weight adjustment from hidden

layer to the input layer: $\Delta w_{hi} = -\eta \cdot \frac{\partial E_k}{\partial w_{hi}}$

$$\begin{aligned} \frac{\partial E_k}{\partial w_{hi}^{(H)}} &= \frac{\partial E_k}{\partial \hat{T}_G} \cdot \left(\sum_{o=1}^m \frac{\partial \hat{T}_G}{\partial y_o^{(o)}} \cdot \frac{\partial y_o^{(o)}}{\partial x_h^{(H)}} \right) \cdot \frac{\partial x_h^{(H)}}{\partial w_{hi}^{(H)}} \\ &= -e \cdot \left(\sum_{o=1}^m \frac{\partial \hat{T}_G}{\partial y_o^{(o)}} \cdot w_{oh}^{(o)} \right) \cdot f'_1(s_h^{(H)}) \cdot \frac{\partial s_h^{(H)}}{\partial w_{hi}^{(H)}} \\ &= -e \cdot \left(\sum_{o=1}^m \frac{\partial \hat{T}_G}{\partial y_o^{(o)}} \cdot w_{oh}^{(o)} \right) \cdot f'_1(s_h^{(H)}) \cdot u_i(k) \end{aligned} \quad (5)$$

When the input sample is:

$$[u_1(k) = G_{G0}(k), \quad u_2(k) = G_{L0}(k), \quad T_{G0}(k), \quad T_G(k), \quad \Delta \hat{T}_G(k-1), \quad \Delta \hat{T}_G(k-2), \quad \Delta G_L(k-1), \quad \Delta G_L(k-2), \quad \Delta G_L(k-3)]$$

applying the input sample to Eq(2) we get the following relation :

$$\begin{aligned} \frac{\partial \hat{T}_G(k)}{\partial y_o^{(o)}} &= -\Delta \hat{T}_G(k-1) \cdot \frac{\partial y_1}{\partial y_o^{(o)}} - \Delta \hat{T}_G(k-2) \cdot \frac{\partial y_2}{\partial y_o^{(o)}} \\ &\quad - a_1 \cdot \frac{\partial \hat{T}_G(k-1)}{\partial y_o^{(o)}} - a_0 \cdot \frac{\partial \hat{T}_G(k-2)}{\partial y_o^{(o)}} \\ &\quad + \Delta G_L(k-1) \cdot \frac{\partial y_2}{\partial y_o^{(o)}} + \Delta G_L(k-2) \cdot \frac{\partial y_1}{\partial y_o^{(o)}} + \Delta G_L(k-3) \cdot \frac{\partial y_2}{\partial y_o^{(o)}} \end{aligned}$$

if $o=1$, from Eq(2) we know that :

$$\begin{aligned}\frac{\partial a_1}{\partial y_1^{(0)}} &= \frac{\partial a_1}{\partial K} = 0, \quad \frac{\partial a_0}{\partial y_1^{(0)}} = \frac{\partial a_0}{\partial K} = 0, \\ \frac{\partial b_2}{\partial y_1^{(0)}} &= \frac{\partial b_2}{\partial K} = \frac{1}{A}, \quad \frac{\partial b_1}{\partial y_1^{(0)}} = 2 \cdot \frac{\partial b_2}{\partial y_1^{(0)}} = \frac{2}{A}, \\ \frac{\partial b_0}{\partial y_1^{(0)}} &= \frac{\partial b_2}{\partial K} = \frac{1}{A}\end{aligned}$$

and

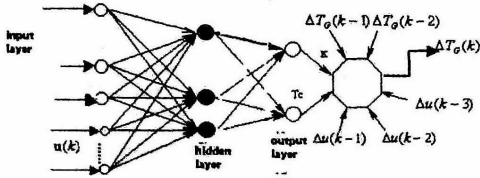


Fig.2 Pray_box dynamic model BP algorithm

$$\begin{aligned}\frac{\partial \hat{T}_c(k)}{\partial y_1^{(0)}} &= -a_1 \frac{\partial \hat{T}_c(k-1)}{\partial K} - a_0 \frac{\partial \hat{T}_c(k-2)}{\partial K} + \frac{1}{A} \Delta G_L(k-1) \\ &+ \frac{2}{A} \Delta G_L(k-2) + \frac{1}{A} \Delta G_L(k-3)\end{aligned}\quad (6)$$

if $\alpha=2$, similarly as in the previous relation, we can also get:

$$\begin{aligned}\frac{\partial \hat{T}_c(k)}{\partial y_1^{(0)}} &= \frac{\partial \hat{T}_c(k)}{\partial T_c} = \frac{16T_c^2 T_c A - (8T_c^2 - 2T_c^2)(8T_c + 4T)}{T^4 A^2} \Delta \hat{T}_c(k-1) \\ &+ \frac{(4T_c^2 - 4T_c + T^2)(8T_c + 4T) - A(8T_c^2 T_c - 4T^3)}{T^4 A^2} \Delta \hat{T}_c(k-2) \\ &- a_1 \frac{\partial \hat{T}_c(k-1)}{\partial T_c} - a_0 \frac{\partial \hat{T}_c(k-2)}{\partial T_c} - \frac{K(8T_c + 4T)}{T^2 A^2} \Delta G_L(k-1) \\ &- \frac{K(16T_c + 8T)}{T^2 A^2} \Delta G_L(k-2) - \frac{K(8T_c + 4T)}{T^2 A^2} \Delta G_L(k-3)\end{aligned}\quad (7)$$

From Eq(4) to Eq(7), getting pray_box dynamic model weight adjustment algorithm as follows :

$$\begin{aligned}\Delta W^{OH} &= \eta \epsilon \begin{bmatrix} \frac{\partial \hat{T}_c(k)}{\partial K} \\ \frac{\partial \hat{T}_c(k)}{\partial T_c} \end{bmatrix} [x_1(k) x_2(k) \dots x_n(k)] \\ \Delta W^{HI} &= \eta \epsilon \begin{bmatrix} \frac{\partial \hat{T}_c(k)}{\partial K} & \frac{\partial \hat{T}_c(k)}{\partial T_c} \end{bmatrix} \begin{bmatrix} w_{11}^{OH} & w_{12}^{OH} & \dots & w_{1n}^{OH} \\ w_{21}^{OH} & w_{22}^{OH} & \dots & w_{2n}^{OH} \end{bmatrix} \frac{\partial}{\partial y_1^{(0)}} u_1(k-1) \leftarrow \text{第 } i \text{ 行} \\ \Delta W^{II} &= [\Delta W^{II}]_{i,j}\end{aligned}$$

4. REAL DATA CONFIRMATION AND CONCLUSION

The pray_box dynamic model described in this paper is confirmed by using actual operation data of Shandong Provincial Communications Department, the result of prediction is depicted in Fig3

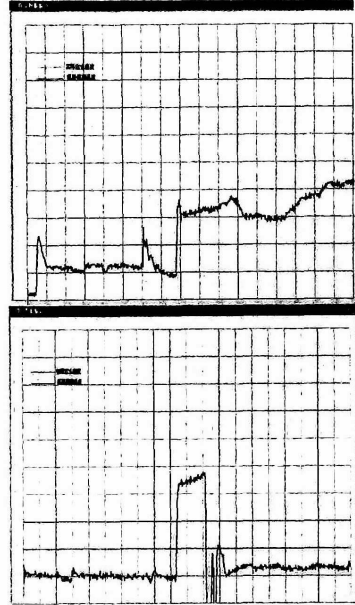


Fig.3 The temperature for NN predictions and actual data

As shown in Fig3, the NN pray_box dynamic model can track the heat exchanger dynamic process well. The purpose of this paper was to demonstrate the reliability of artificial intelligence in modeling of industrial processes, the identified neural network model presents fairly good results in prediction. The results are quite general and are applicable to most classes of other nonlinear system model.

REFERENCES

- [1] Norega R, Wang Hong. A direct adaptive Neural Network Control for unknown nonlinear systems and its application[J]. IEEE TRANSACTIONS ON NEURAL NETWORKS, 1998, 19(1): 27-34.
- [2] K.Funahashi, On the approximate realization of continuous mappings by neural networks, Neural Networks 2(1989) 183-192
- [3] G.Cohen, Neural network implementations to control real-time manufacturing systems, Comput. Manuf. Syst. 11(1998) 243-251.
- [4] L.N.Smith, L.V.Dihoru, Optimisation of powder packing densities using image analysis and neural network modelling, Met. Powder Rep. 55(3) (2000)