Internal Model Control Based on Dynamic Fuzzy Neural Network

ZhiJun Zhang , XueMiao Wang
School of Electronics and Information Engineering, Dalian University of Technology
Dalian, China
zzj@dl.cn

Abstract

A novel internal model control (IMC) approaches is proposed to control for process with large time delays. A dynamic fuzzy neural network (DFNN) was applied to model the process and its mathematical inverse to control the process. The presented strategy can overcome the unstable problem caused by time delays and the problem of time consuming and complex and incapable obtaining mathematical inverse of model using the classical internal model control. The automatic configuration and learning of the networks is carried out by using new adding techniques and on-line learning algorithm. The effectiveness of the proposed control scheme is verified by simulated results.

1. Introduction

The internal model control (IMC) design method is a general linear controller design methodology that provides a transparent framework. It has received much attention over the past decade, has widely used in control of many practical process. For example, design methods for tuning PID controllers using the IMC structure are proposed in Gorez [1] and a modified IMC structure to deal with unstable processes with time delays is given in Tan, Marquez, and Chen [2], and a linear matrix inequality (LMl) IMC-based strategy to design and tune a robust tunable controller is proposed in Boulet, Duan, and Michalska [3].The key advantage of the IMC controller synthesis technique is that:

(1) It is simple and can easily be arranged into the traditional feedback controller.

(2) It can handle manipulated variable constraints (unlike traditional PID controllers), and can also handle dead time.

(3) The nominal closed-loop stability is guaranteed, for an open-loop stable process, and stable controller.

However, with the IMC design method obtaining an inverse of many complex continuous systems is extremely difficult, the problem becomes more computationally worse and uncontrollable for the uncertain nonlinear system with time delays, the IMC structure cannot be implemented when inverse of the system does not exist. In addition, inversion of static nonlinearities could be very sensitive to the uncertainties in the nonlinear functions. Hence, there is a need for a control method that approximates the inverse function more effectively. This problem is resolved when neural network based techniques are applied to internal model control .Neural networks have been successfully used in many process control applications [4][5][6][7]. Their ability to approximate arbitrary nonlinear vector functions, combined with dynamic elements has yielded powerful tool to model nonlinear dynamical systems [8][9]. Most of the non-linear control algorithms based on neural networks imply the minimization of a cost function, by using computational methods for obtaining the optimal command to be applied to the process [10]. Where as the neural network model is frequently complex, calculation of its mathematical inverse is difficult and time consuming. The neural network models which have been proposed for process control have several nodes in the input and hidden layers, as well as a large number of weights and bias terms, these weights and bias terms are usually initialized randomly, using no prior process knowledge to reduce network training time. Two major drawbacks arise. First, it is difficult to obtain the control law in an analytical way. A nonlinear optimization routine must be used to calculate the control sequence. Therefore, computational effort and convergence velocity are two key issues. Second, the abundant theory and experience with tuning linear controllers cannot easily be used for the tuning of nonlinear controllers. Thus, it may not be possible to train a NN to reach a desired level of performance, if the network does not have enough computational units, or if the learning algorithm fails to find the optimal network parameters. Moreover, these networks generally require long training times and convergence to the minimum error solution cannot be guaranteed if the network training is performed by the back propagation algorithm, which is a commonly used NN training algorithm. The implementation of the non-linear model control algorithms becomes very difficult for real-time control because the minimization algorithm must converge at least to a sub-optimal solution and the operations involved must be completed in a very short time (corresponding to the sampling period). Therefore, a configuration of a NN in addition to establishing optimal network parameters by using a suitable learning algorithm is an extremely useful tool. In this paper, a new internal model control based on dynamic fuzzy neural network (DFNNIMC) design method is proposed. The present

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00

work focuses on developing a fuzzy neural network and modifying the model on-line. The dynamic adaptation training of fuzzy neural network may start with none or a single fuzzy rule or neuron. A period training, the network structure grows and concurrently the parameters of antecedents and consequents are adapted to obtain the desired modeling accuracy. It not only increases the computation velocity but also decrease the computation steps. The capabilities and performance of all these new techniques is suitably demonstrated by their application to the modeling and control of a realistic simulation of the typical numerical example. Examples show that the structure is easy to tune and can achieve better tradeoffs between time domain performance and robustness.

2. Internal model control architecture

2.1. Model Design

Considering the signal input and signal output system can be given as a discrete time model

$$\begin{aligned} y(k+1) = f[&y(k), y(k-1), ..., y(k-n), \\ &u(k), u(k-1), ...u(k-m)] \end{aligned} \tag{1}$$

Where $y(k+1)$ is the process output at the $k+1$ time instant, and $u(k)$ is the input at the k time instant; $f(.)$ is the unknown function, n and m respectively are the time delay of the output and input. Assuming the system is reversible .The internal model is developed as a function of the past and present inputs, as well as the outputs to yield the future output of the system. In terms of the mathematics this is represented as

$$\begin{aligned} \hat{y}(k+1) = \hat{f}(&y(k), y(k-1), \cdots, y(k-i), \\ &u(k), u(k-1), \cdots u(k-j)) \end{aligned} \tag{2}$$

Where $y(k)$ is the present output of the system and $y(k+1)$ is the one step ahead output of the system, y(k-1) to y(k-i) are the past outputs up to the ith time scale. Similarly the corresponding inputs are represented as $u(k)$ for the present time and, u(k-1) to $u(k-i)$ are the past inputs to the jth level in time.

The fuzzy neural network based on Takagi-Sugeno be used to construct the above model. The structure of the fuzzy neural network is a five-layer, and the first layer is the input layer which each neuron correspond to an input variable, the second layer is the Gaussian fuzzy membership function layer which each neuron represents an premise of a fuzzy rule, and the third layer is the normalized layer, and the four layer is the weighted layer ,and the five layer is the output layer whose output can be expressed as follows

$$y(x) = \sum_{j=1}^{p} f_j = \frac{\sum_{j=1}^{p} T_j \exp\left[-\sum_{i=1}^{r} \frac{(x_i - c_{ij})^2}{2\sigma_{ij}^2}\right]}{\sum_{k=1}^{p} \exp\left[-\sum_{i=1}^{r} \frac{(x_i - c_{ij})^2}{2\sigma_{ik}^2}\right]} \tag{3}$$

Where y is the value of an output variable, $T_j = W_{j0} + W_{j1}x_1 + ... + W_{jr}x_r$,c_{ij} is the center of the ith membership function in the jth neuron, σ_{ij} is the width of the ith membership function in the jth neuron, r is the number of input variables, p is the number of neurons in

2.2. Controller Design

The controller is designed based on inverse model technique. Inverse model of the dynamical system plays a vital role in many control strategies and structures. These inverse models can be directly used as the controllers in direct inverse and internal model control schemes. Obtaining an inverse of many complex continuous systems is extremely difficult. The inverse model is constructed by fuzzy neural network in this paper. It via neural networks is obtained by the direct manipulation of the data sets, achievable through considering the inputs of the system as the outputs. This, however, can only be achieved in the region of the plant's operation where one to one functional mapping between the inputs and the outputs exist. During the training of the inverse models the neural network is fed with the required future or the reference output/set point value together with past inputs and the past outputs to predict the current input of the system i.e. $u(k)$. Thus the equation for the forward model can be altered accordingly and inverse model expression (4) can be obtained as

$$u(k) = \hat{f}^{-1}(y(k), y(k-1), \cdots; y(k-i), u(k), u(k-1), \cdots u(k-j)) \tag{4}$$

Where the nomenclature is the same as explained earlier for forward model, except for $\hat{f}^{-1}$ which is the inverse map of the forward model.

3. The Training Methodology

Learning algorithms of DFNN are composed of two phases: the structure learning and the parameter learning. The structure learning attempts to achieve an economical network size.

3.1. The structure learning

Step 1 Initialize structure of network.
Let

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE

$$c_1 = x_1^T \tag{5}$$

And

$$\sigma_1 = [\sigma_0 \quad \sigma_0 \quad \cdots \quad \sigma_0]^T \tag{6}$$

Where c_1 is the center of vector whose dimension is $r \times 1$, x_1 is the first training pattern enters the network, σ_1 t is the width vector whose dimension is $r \times 1$, σ_0 is a predefined initial width ,respectively, $0 \le w_{ij} \le 1$, w_{ij} is the weighted coefficient .

Step 2 Modify the structure rule. The structure that can train in the above-mentioned structure, the node number (the rule number) of the hidden layer carries on changing structure according to the criterion [11].

3.1.1. Error criterion. The system error $\varepsilon(k)$ is defined as

$$|\varepsilon(k)| = |d_k - y_k| \tag{7}$$

Where d_k is the desired output, y_k is the output of the network with the current framework.

3.3.2. If-part rule criterion. For the jth neuron in the hidden layer, the output is

$$\xi_j = \exp\left[-\sum_{i=1}^{r} \frac{(x_{ik} - c_{ij})^2}{2\sigma_{ij}^2}\right] \tag{8}$$

Now define

$$\xi(k) = \arg\max_j (\xi_i) \tag{9}$$

If

$$|\varepsilon(k)| > \delta \tag{10}$$

Where δ is the predefined error tolerance. And

$$\xi(k) < 0.1354 \tag{11}$$

Each node in these sets is checked by (11) and (12) ,If it is satisfied ,a new neuron should be considered for adding in the network. The center c_{new} of and width σ_{new} of new neuron take $c_{new} = x_k, \sigma_{new} = 2\min(dis)$, respectively, where *dis* is the distance of input vector .The network is adapted by adding nodes. If it is not satisfied, then the following the parameters learning should be do.

3.2. The parameter learning

The parameters which include the center (c_{ij}) and the width (σ_{ij})of the Gaussian basis function, and the weighted coefficient (w_{ij}) are updated by gradient descent method.

In learning process, the fuzzy network reaches a local minimum in error. The optimizing implement rely on the performance index, it is given as the following

$$E = \frac{1}{2}[d_k - y_k]^2 \tag{12}$$

If E is less than the setting, the parameters of the network are not adjusted and the structure is also not modified .When E larger than the setting, the on-line optimizing algorithm can be expressed as

$$w_{ij}(k) = w_{ij}(k-1) - \eta_1 \frac{\partial E}{\partial w_{ij}} \tag{13}$$

Where

$$\frac{\partial E}{\partial w_{ij}} = (d_k - y_k)\varphi_j x_i \tag{14}$$

$$c_{ij}(k) = c_{ij}(k-1) + \eta_2 (d_k - y_k) \frac{\partial f}{\partial \xi_j} \frac{\partial \xi}{\partial c_{ij}} \tag{15}$$

$$\sigma_{ij}(k) = \sigma_{ij}(k-1) + \eta_3 (d_k - y_k) \frac{\partial f}{\partial \xi_j} \frac{\partial \xi}{\partial \sigma_{ij}} \tag{16}$$

Where

$$\frac{\partial f}{\partial \xi_j} = \frac{w_j(k) - y_k}{\sum_{j=1}^{m} \xi_j} \tag{17}$$

$$\frac{\partial \xi_j}{\partial c_{ij}} = \frac{(x_{in} - c_{ij}(k-1))}{\sigma_{ij}^2(k-1)} \xi_j \tag{18}$$

$$\frac{\partial \xi_j}{\partial \sigma_{ij}} = \frac{(x_{ij} - c_{ij}(k-1))^2}{\sigma_{ij}^3} \xi_j, j = 1, 2, ..., p \tag{19}$$

η_1, η_2, η_3 is step of learning , η_1, η_2, η_3 choose at [0,1] randomly.

The parameter of the new network should also be updated using the same procedure above.

4. Simulation Result

To indicate the approach proposed capability to the system with larger delay time and the rapidly convergence of the learning algorithm, two examples is studied.

1) example 1

A typical non-linear function is expressed as follows

$$y(k+1) = \frac{y(k)y(k-1)[y(k)+2.5]}{1+y^2(k)+y^2(k-1)} + u(k) + 1.2u(k-1) \tag{20}$$

In Simulation, the training set consist of 200 input/output pairs, let $u(-0.5, +0.5)$ random input signal,

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE

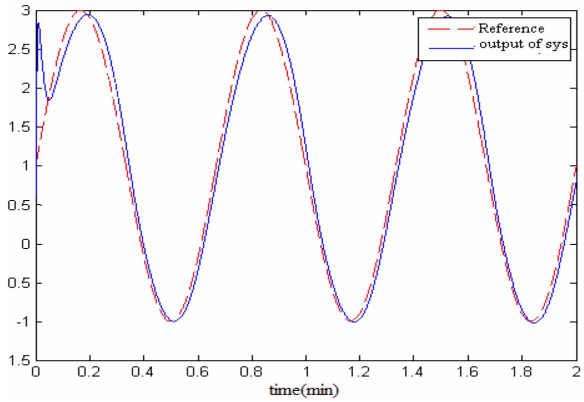

Fig. 1. **The Response of System by IMC**

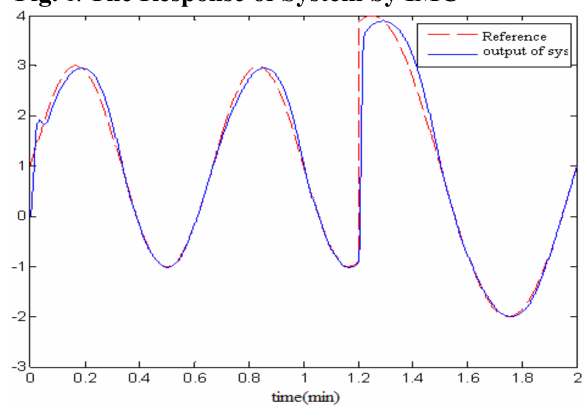

Fig. 2. Response of System controlled by IMC

A sampling time of 0.1s, the trial length is 2.0 minutes. Fig.1. and Fig.2.displays the convergence properties of the schemes, respectively, and displays that it works well for any set point change, the set point tracking error observed is less than 0.5% for the closed-loop. Obviously, the control system has rapid tracking ability.

2) example 2

The approach proposed is used to control continuous stirred tank reactor (CSTR). The dynamic model is expressed as follows

$$\dot{C}_A = \frac{q}{V}\left(C_{Af} - C_A\right) - k_0 C_A \exp\left[-\frac{E}{RT(t)}\right] \tag{21}$$

$$\dot{T} = \frac{q}{V}\left(T_f - T(t)\right) - \frac{\Delta H k_0 C_A}{\rho C_p}\exp\left[-\frac{E}{RT(t)}\right] + \frac{\rho_c C_{pc}}{\rho C_p V} q_c(t)\left[1 - \exp\left[-\frac{hA}{q_c(t)\rho_c C_{pc}}\right]\right]\left(T_{cf} - T(t)\right) \tag{22}$$

Where $C_A(t)$ is concentration of product compound, $T(t)$ is the temperature of mixture, $q_c(t)$ is coolant flow rate, C_{Af} is the inlet concentration, q is process flow rate, v is reactor volume , T_f and T_{Cf} is the inlet feed and coolant temperatures respectively, all of which are assumed constant at nominal condition. Likewise $k_0, E/R, \Delta H, hA, \rho, \rho_c, C_p, C_{cp}$ are thermodynamic and chemical constants. The objective of control is to keep the concentration by manipulating the flow rate $q_c(t)$, ensuring product conforms to the requirement. The range of concentration of product compound is about 0.08-0.12*mol/L*.The nominal operating conditions of the CSTR are shown in Table 1.

Tab. 1 **.Nominal operating conditions of the CSTR**

$q = 100L/\min$	$E/R = 9.95\times10^3 K$
$V = 100L$	$\rho_C, \rho = 1\times10^3 g/L$
$C_{Af} = 1$mol/L	$C_{pc}, C_c = 1cal/gk$
$T_f = 350K$	$hA = 7\times10^5 cal/\min K$
$T_{cf} = 350K$	$q_c = 103.41 l/\min$
$k_0 = 7.2\times10^{10}/\min$	$C_A = 8.36\times10^{-2} mol/l$
$\Delta H = -2\times10^5 cal/mol$	$T = 440.2K$

To provide a basis for comparison, The FNNIMC and conventional PID controller are employed to control the process. The simulated results are shown in figure 3 and figure 4. It is obvious that the system reaches the desired setpoint with the proposed FNNIMC design, while the simple PID design leads to an oscillatory response. The performance of the FNNIMC is much superior to the conventional PID controller. The disturbance signal $d(k) = 0.01\cos(3k)$ was added to the output of the process. The simulated result is shown as figure.5. It can be observed that the concentration of product compound reached the reference values within 0.1 min in every case of changing the reference values. Again, no overshooting was detected. The result shows the approach proposed capability to overcome disturbance.

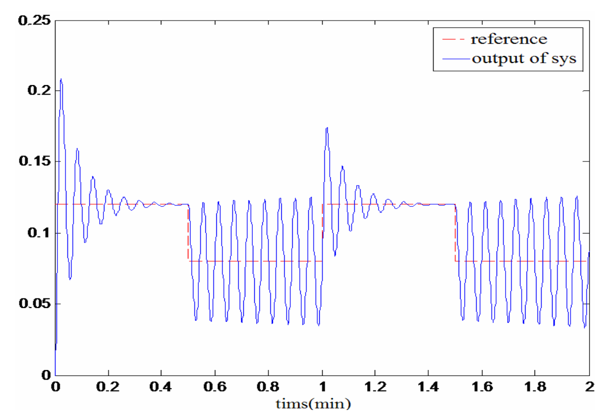

Fig.3**. Response of CSTR controlled by PID**

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE

The while the process Parameter $\frac{\rho_c C_{pc}}{\rho C_p V}$ was changed $0.85\frac{\rho_c C_{pc}}{\rho C_p V}$, the results of this test is shown in Fig.6. It is satisfying to observe that the concentration of product compound reached the set-point values without overshooting, the approach proposed has the well robustness.

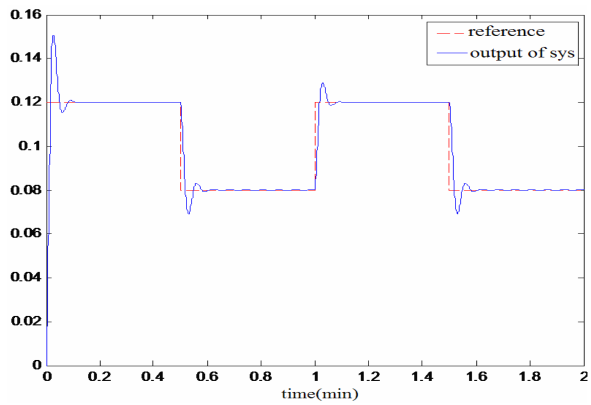

Fig.4.Response of System controlled by IMC based FNN

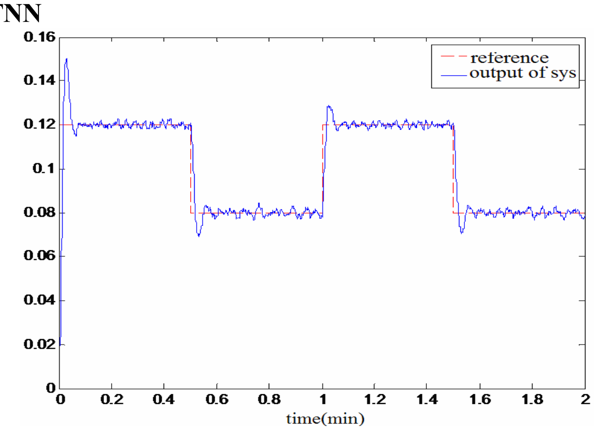

Fig.5. Output response of System controlled by IMC with disturbance

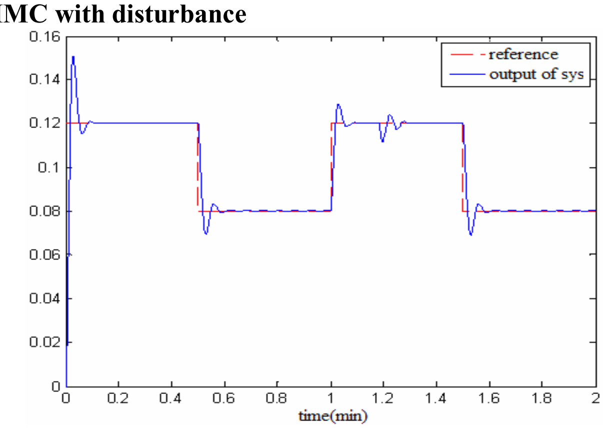

Fig.6. Output of System controlled by IMC with process Parameter Varying

4. Conclusion

A new IMC strategy based on dynamic fuzzy neural network technique has developed and applied to control the system with larger delay time. The result indicated presented scheme and on-line optimizing algorithm sufficiently improved the system performance, quickened computation velocity, therefore sped up system convergence, this is helpful to the real-time control implementation. Moreover, the controller has the good compatibility and robustness on account of the parameters of the fuzzy network can rapidly adapt themselves with changes of operating conditions. Simulation results illustrate the proposed approach is correctness and effectiveness.

5. Reference

[1] Gorez, R., New design relations for 2-D of PID-like control systems, Automatica, Vol.39(5). (2003) 901–908.

[2] Tan,W., Marquez, H. J., & Chen, T, IMC design for unstable processes with time delay, Journal of Process Control, Vol. 13. (2003).203–213.

[3] Boulet, B., Duan, Y. & Michalska, H, An LMI approach to IMC-based robust tunable control, In Proceedings of the American control conference Vol. 1. (2003) 821–826

[4] Dias Fernando, M., Mota Alexandre, M, A comparison between a PID and internal model control using neural networks.5th WorldMulti-Confe rence on Systemics, Cybernetics and Informatics, Vol. 5, Orlando, EUA,(2001) 268–273.

[5] Kiguchi, K., Tanaka, T., Fukuda, T.: Neuro-fuzzy control of a robotic exoskeleton with EMG signals. IEEE Trans. Fuzzy Systems 12 (2004) 481-490

[6] Wang, L.P., Frayman, Y.: A Dynamically-generated fuzzy neural network and its application to torsional vibration control of tandem cold rolling mill spindles. Engineering Applications of Artificial Intelligence 15 (2003) 541-550

[7] Lin, C.-M., Hsu, C.-F.: Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings. IEEE Trans. Fuzzy Systems, 12 (2004) 733-742

[8] Hunt, K. J., & Sbarbaro, D, Neural networks for nonlinear internal model control, IEE Proceedings—Part D: Control Theory and Applications,Vol. 138(5). (1991) 431–438.

[9] Frayman, Y., Wang, L.P.: A Dynamically-constructed fuzzy neural controller for direct model reference adaptive control of multi-input-multi-output nonlinear processes. Soft Computing, 6 (2002) 244-253

[10] Lightbody, G., & Irwin, G. W, Nonlinear control structures based on embedded neural system models, IEEE Transactions on Neural Networks, 8(3). (1997). 553–567.

[11] Lee,C.C, Fuzzy logic in control systems: fuzzy logic controller-parts .IEEE Transactions on systems,Man,and cybernetics,Vol.20.(1990) 404-435

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE