Stable Tracking Control to a Non-linear Process Via Neural Network Model

Yujia Zhai
Department of Electrical and Electronic Engineering
Xi'an Jiaotong-Liverpool University

Abstract A stable neural network control scheme for unknown non-linear systems is developed in this paper. While the control variable is optimised to minimize the performance index, convergence of the index is guaranteed asymptotically stable by a Lyapnov control law. The optimization is achieved using a gradient descent searching algorithm and is consequently slow. A fast convergence algorithm using an adaptive learning rate is employed to speed up the convergence. Application of the stable control to a single input single output (SISO) non-linear system is simulated. The satisfactory control performance is obtained.

• Key Words : Stable, Control, No-linear, Neural Network

1. Introduction

Application of neural networks (NN) in process modelling and control has been studied intensively in recent years. A number of NN based control strategies were捌 developed. In these strategies, NN were used either as process models, such as the internal control [1-2] and the neural network model based predictive control [3-4], or as controllers, such as the model reference adaptive control [5-6]. To ensure the stability of overall control system involving NNs, some stable control approaches were also developed using Lyapnov method [7-8]. Due to non-linearity of the system and the network, development of these control laws are very difficult, and the developed algorithms are complex and with strict limitations.

Recently, a few NN based control methods were proposed to minimize the error between the reference signal and the NN model output [9-11]. Since the NN models can be constructed and trained to model a continuous non-linear function to any pre-specified accuracy, provided with enough number of neurons in the hidden layer [12], then the process output is guaranteed to track the reference with only modelling error. In [9], the control variable was optimized using the gradient descent method to minimize the difference between the reference and the model output. However, the system stability was not considered. Tan and Cauwenberghe [10] considered the stability issue in a one-step-ahead control and derived a stability condition. The method used a multilayer perception network as the model and the control variable is optimized using the gradient descent method. This optimization is slow and therefore the control variable is difficult to achieve the optimal value in one sample interval. Behera et al, [11] proposed an algorithm for SISO system that the control variable was first estimated by the model inversion techniques, and then was further optimized by a stable control law. In this method, since the first estimation of the control variable was used in the performance index of the optimal control, the control performance for a multi-input multi-output (MIMO) system is greatly degraded due

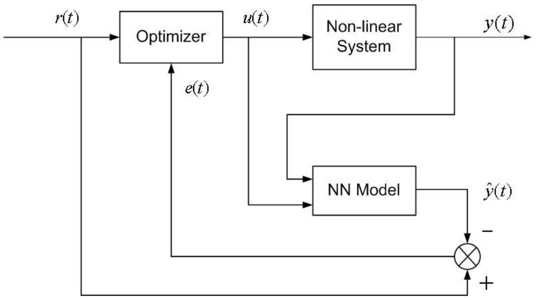
*교신저자 : Yujia Zhai(yujia.zhai@xjtlu.edu.cn)
접수일 2014년 8월 12일 수정일 2014년 10월 26일 게재확정일 2014년 11월 5일

to the first estimation error of the control variable.

In this paper, a stable control law is developed to minimize the difference between the reference signal and the NN model output, so that when the model precisely represents the system, the system output will follow the reference. It is guaranteed a Lyapnov function that the control variable asymptotically converges to the optimal value. The stable control law is derived from a modification of the performance index used in [11]. The optimization of the control variable is speed up by applying a fast convergence algorithm [13]. The developed method is applied to a SISO non-linear process and the simulation results are demonstrated. Satisfactory ontrol performance using the developed method is shown.

2. NN stable control strategy

NN Control and stability analysis for a non-linear system is very difficult, and even it is achievable it may currently not be easy for practical application due to the complexity of the algorithm and strict conditions. The control strategy developed here is to minimise the difference between the set-point and the predicted output of the NN model. This greatly simplifies the development of the stable control law. The strategy is shown as follows:



[Fig. 1] NN Stable Control Strategy.

The MIMO non-linear system in [Fig.1] is represented by a non-linear auto-regressive exogenous (NARX) model of the following form,

$$y(t)=f\left(u(t-1),\cdots u(t-n_u),y(t-1),\cdots,y(t-n_y)\right)+e_p(t) \tag{1}$$

Where $y(t)\in\Re^p$ and $u(t)\in\Re^m$ are process output and input vectors, n_y and n_u are output order and input order, respectively, $e_p(t)$ is the modelling error vector and $f(.)$ is a vector valued non-linear function. In this paper, a radial basis function network (RBFN) with the Gaussian function is used to model the system in (1). The RBFN performs a non-linear mapping and is represented by the following equations,

$$\hat{y}=W\phi \tag{2}$$

where $\hat{y}\in\Re^p$ is the network output vector, $W\in\Re^{p\times n_h}$ is the weight matrix with the element, w_{rj} connecting the j th hidden layer node to the i th output, and $\phi\in\Re^{n_h}$ the hidden layer output vector with its element given by,

$$\varphi_j=\exp(-\frac{z_j^2}{\sigma^2}),\ j=1,\cdots,n_h \tag{3}$$

with

$$z_j=\left\|x-C_j\right\|,j=1,\cdots,n_h \tag{4}$$

here, $x\in\Re^q$ is the input vector of the network involving the process input vector $u\in\Re^m$ as m elements, $C_j\in\Re^q, j=1,\cdots,n_h$ is the j th centre vector associated to the j th hidden layer node, $\|\cdot\|$ is the Euclidean norm.

To realise the control system in Fig, the performance index in the optimisation is given as,

$$J(t) = e(t)^T W_y e(t) + \xi \Delta u(t)^T W_u \Delta u(t) \tag{5}$$

where $e = r - \hat{y}$ is the model prediction error vector, r is the reference output vector, $\Delta u(t) = u(t) - u(t-1)$ is the control increment vector and t is the sample time. W_y and W_u are positive definite weighting matrices used to scale different output and input ranges and ξ is the control weight. Compare to the index used by Behera et al. the weighted norm of the control increment is used instead of norm of the error between the control to be optimised and the estimate of the control. It was found that when the estimation is not accurate (it is always the case especially for MIMO non-linear systems, otherwise the optimisation is not needed), the control performance would be greatly degraded. Inclusion of control increment in (5) tends to smoothen the control variable.

A stable control law is now developed to compute control variable, *u(t)* so that the performance index $J(t)$ is minimised asymptotically. Using the index in as a chosen Lyapnov function, obviously, $J(t) > 0$ for $e(t) \neq 0 \quad \forall t$. The derivative of $J(t)$ with respect to t is,

$$\dot{J} = (-e^T W_y \frac{\partial \hat{y}}{\partial u} + \xi \Delta u^T W_u)\dot{u} \tag{6}$$

It follows that if $\dot{u}$ is given as

$$\dot{u} = -\frac{(\xi \Delta u^T W_u - e^T W_y {}^{\partial \hat{y}}\!/\!{}_{\partial u})^T e^T W_y e}{\left\| \xi \Delta u^T W_u - e^T W_y {}^{\partial \hat{y}}\!/\!{}_{\partial u} \right\|^2} \tag{7}$$

then $\dot{J} = -e^T W_y e < 0$. This proves that if control increases along the direction given in (7), then index J will asymptotically converge to its minimal value. The calculation of is $\dot{u}(t)$ straightforward except for $\frac{\partial \hat{y}}{\partial u}$, which is according to the structure of the NN used. For the RBFN given in (2)-(4), $\frac{\partial \hat{y}}{\partial u}$ can be derived as

$$\frac{\partial \hat{y}(t+1)}{\partial u(t)} = \frac{\partial \hat{y}(t+1)}{\partial \phi(t)} \cdot \frac{d\phi(t+1)}{dz(t+1)} \cdot \frac{\partial z(t+1)}{\partial u(t)} \tag{8}$$

where

$$\frac{\partial \hat{y}(t+1)}{\partial \phi(t)} = W$$

$$\frac{d\phi}{dz} = -2 \cdot \begin{bmatrix} \frac{z_1}{\sigma_1^2} & & \\ & \ddots & \\ & & \frac{z_{n_h}}{\sigma_{n_h}^2} \end{bmatrix} \cdot \begin{bmatrix} \varphi_1 & & \\ & \ddots & \\ & & \varphi_{n_h} \end{bmatrix}$$

$$\frac{\partial z}{\partial u} = \begin{bmatrix} \frac{1}{z_1} & & \\ & \ddots & \\ & & \frac{1}{z_{n_h}} \end{bmatrix} \cdot \begin{bmatrix} u_1 - c_{11} & \cdots & u_m - c_{m1} \\ \vdots & \ddots & \vdots \\ u_1 - c_{1n_h} & \cdots & u_m - c_{mn_h} \end{bmatrix}$$

and $C_j = \left[c_{1j} \cdots c_{mj} \right]^T$ is the j th centre. Thus, we have

$$\frac{\partial \hat{y}}{\partial u} = -2 \cdot W \cdot \begin{bmatrix} \frac{1}{\sigma_1^2} & & \\ & \ddots & \\ & & \frac{1}{\sigma_{n_h}^2} \end{bmatrix} \cdot \begin{bmatrix} \varphi_1 & & \\ & \ddots & \\ & & \varphi_{n_h} \end{bmatrix} \cdot \left[u - C_1 \quad \cdots \quad u - C_{n_h} \right] \tag{9}$$

Computation of $u(t)$ according to (7) can be simply realised in discrete form,

$$u(t+1) = u(t) + \Lambda t_s \dot{u} \tag{10}$$

where t_s is the sample interval and $\Lambda \in \Re^{m \times m}$ is a diagonal matrix with its element, $\lambda_{i,i}$ being the learning rate of the i th control, u_i . However, it was found that using a fixed learning rate in the convergence was often very slow. For example, in the application of the developed control strategy to a SISO non-linear process described in the section following, when set-point has a step change, the convergence needs more than 300 iterations (will be shown later) to satisfy the given tolerance. A fast convergence algorithm by adapting learning rate [13] is used here to speed up the convergence.

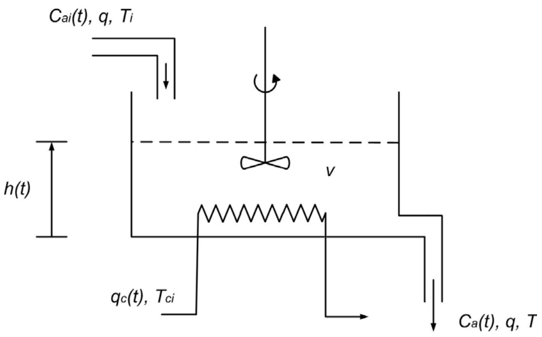
$$\begin{cases} \lambda_i(k) = \beta\lambda_i(k-1) & if \quad \dot{u}(k)\dot{u}(k-1) > 0 \quad and \quad |J(k)| < |J(k-1)| \\ \lambda_i(k) = \lambda_i(0) & if \quad \dot{u}(k)\dot{u}(k-1) < 0 \quad or \quad |J(k)| < |J(k-1)| \quad ,i = 1,\cdots,m \\ \lambda_i(k) = \lambda_i(k-1) & otherwise \end{cases} \tag{11}$$

Here k is the iterative number in the optimisation process and not the sample instant, $\beta > 1$ is an exponential adapting parameter that is chosen in the design. The first condition in (11) means that when control variable keeps increasing or decreasing, and performance index reduces in either positive side or negative side. In this case the control variable smoothly converges to its optimal value and therefore the learning rate should increase. The second condition means that either the performance index increases or the control variable is oscillatory. This may be caused by a too large learning rate and therefore it should be reset to the initial value. In the other cases the learning rate keeps the same value. β is chosen according to the application and generally in the range, .

$$\beta \in [1.1, \quad 1.5]$$

3. Process description

The SISO CSTR p滯rocess given by Morningred, et al. (1990) is chosen as a realistic example for the application in this paper. The CSTR process is a typical dynamical process used in chemical and biochemical industry. The schematic diagram of the reactor is shown in [Fig. 3.1]



[Fig. 2] SISO CSTR process.

This CSTR process is chosen because it has highly non-linear dynamics. The following equations are available to describe the process dynamics.

$$\frac{dC_a(t)}{dt} = \frac{q}{v}[C_{ai} - C_a(t)] - k_0 C_a(t) e^{\frac{-E}{RT(t)}} \tag{12}$$

$$\frac{dT(t)}{dt} = \frac{q}{v}[T_i - T(t)] - k_1 C_a(t) e^{\frac{-E}{RT(t)}} + k_2 q_c(t)\left[1 - e^{\frac{-E}{q_c(t)}}\right][T_{ci} - T(t)] \tag{13}$$

The CSTR process can be briefly described as follows. Within the CSTR, two kinds of chemicals are mixed and react to produce a product, compound A, at a concentration $C_a(t)$, with the temperature of the mixture being $T(t)$. The reaction is exothermic, producing heat, which acts to slow the reaction down. By introducing a coolant flow-rate $q_c(t)$, the temperature can be varied, and hence the product concentration controlled. $C_{ai}(t)$ is the inlet feed concentration, $q(t)$ the process flow-rate, $T_i(t)$ and $T_{ci}(t)$ the inlet and coolant temperature, respectively, all of which are assumed constant at nominal values. Likewise $k_0(t), E/R, v, k_1, k_2$ and

k_3 , are thermodynamic and chemical constants relating to this CSTR process. Numerical values for the parameters of this CSTR are given in Appendix. The input and output of this CSTR process are

$$u(t) = q_c(t) \quad , \quad y(t) = C_a(t)$$

the bounded range of the input variable is

$$q_c(t) \in [90, \quad 110] \quad (l/\min)$$

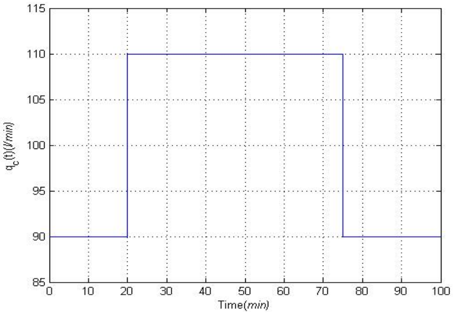
The initial conditions of the CSTR process are assumed to be a nominal steady state where the concentration and tank temperature have the following values:

$$C_a = 0.062 \quad (mol/l) \quad , \quad T = 448.752 \quad (K)$$

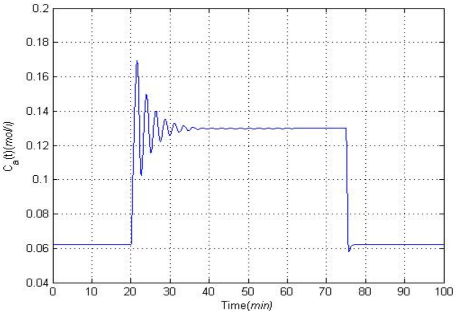
Corresponding to a constant input of

$$q_i = 90(l/mol)$$

The CSTR process open-loop response in [Fig.3.3] to a step change of exciting signal in [Fig. 3.2] shows that the process exhibits a high degree of non-linearity.



[Fig. 3] Exciting input signal to simulated SISO CSTR process.



[Fig. 4] Simulated SISO CSTR process open-loop response.

In the investigation of simulation, 3-second sample time is chosen in this SISO CSTR process. This CSTR process is simulated as an unknown non-linear dynamic process in modelling and control using NNs.

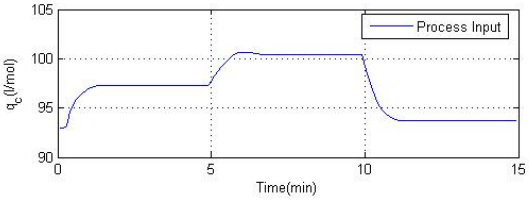
4. NN adaptive control of the process

The RBFN model based adaptive control strategy with the fast learning algorithm was applied to the Simulink model of the non-linear SISO process in [Fig.3.1] Firstly, a NN model was trained for the process. For this process, the output order and input order were estimated as 3 and 2, respectively. The input vector of the network model was therefore determined as

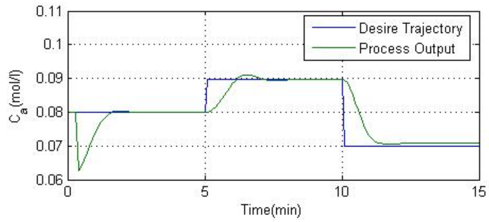
$$x(t) = [C_a(t) \quad , \quad C_a(t-1) \quad , \quad C_a(t-2) \quad , \quad q_c(t-1) \quad , \quad q_c(t)]^T.$$

The output vector of the NN model was $\hat{y}(t+1) = \hat{C}_a(t+1)$ and 20 hidden layer nodes were used, therefore, the RBFN model has structure of . A sample period, $T_s = 3\sec$, was chosen according to the knowledge of process and 1500 samples of input-output data were collected from the process Simulink model when a random amplitude signal (RAS) was used as the excitation signal. 20 centres were selected using the K-means clustering algorithm [14] from these data. The first 1000 samples then were used to train the RBFN model using a recursive orthogonal least squares training algorithm [15], while the last 500 samples where used to validate the model (the detailed modelling is not shown here for briefness).

A Matlab file was coded to calculate the optimal control in each sample interval, according to the prediction of the RBFN model to minimise the cost function in (5) for set-point tracking control. The computed optimal control variable was then sent to the Simulink model to produce the process output. This online mode of the simulation was executed for 200 samples. The set-point signal was designed from initial value of 0.08 mol/l to 0.09, and to 0.07, at 5 min intervals. In the simulation, the control parameters were set as $w_y = 1$, $w_u = 1$ as the input and output data had been scaled before optimisation, and $\xi = 0.05$. Also, the initial value of the control at each sample interval was set as the value at the last interval, $u(t)|_{k=0} = u(t-1)$ and the initial value of the learning rate, $\lambda = 0.01$ and $\beta = 1.5$. When the control variable was bounded as $u(t) \in [90, \ 110] \ \ (l/\min)$, the simulated process output, the set-point and the optimal control are displayed in [Fig.4.1] and [Fig.4.2]



[Fig. 5] Process input signal of SISO CSTR process



[Fig. 6] Process tracking response of SISO CSTR process

It can be observed in [Fig.4.2] that the process output follows step change in the set-point quickly with small overshoot and settle down quickly with very small steady-state error. When increased the control weight ξ , the control variable was smoother than that in [Fig.4.1] but the process output responded to the step change in the set point correspondingly slowly.

The use of the fast optimisation algorithm greatly reduced the time interval used for optimisation. One example was taken from the simulation and is shown here. When a tolerance of 0.001 was used for the performance index, $J < 0.001$, the optimisation stopped at the iteration 60 when using the adaptive learning rate, while the index did not reach the same tolerance after 300 iterations when using the fixed learning rate $\lambda = 0.01$.

The fast algorithm is stable because the learning rate will be reset to the initial value when the convergence gets to oscillatory. On average it can reduce computing time to 1/5 ”1/10 of the time when using the fixed learning rate algorithm in this application example.

5. Conclusions

A NN based stable control is developed for unknown SISO non-linear systems. The application of developed control strategy to a ch浥emical process is simulated. Set-point tracking is satisfactory. The optimisation in this approach is solved using the fast optimisation algorithm. The algorithm greatly reduces the computing time, which enables the control strategy to be applied to a fairly fast system.

References

[1] HUNT, K.J., SBARBARO, R., ZBIKOWSKI, R. and GAWTHROP, P.J.: ‘Neural networks for control systems - a survey’, Automatica, pp. 575-587, 1992.

[2] HUNT, K.J. and SBARBARO, D.: ‘Neural networks for non-linear internal model control’, IEE Proc.D, pp. 431-438, 1991.

[3] GOMM, J.B., EVANS, J.T. and WILLIAMS, D.: ‘Development and performance of a neural network predictive controller’, Control Engineering Practice, pp. 49-59, 1997.

[4] YU, D.L., GOMM, J.B. and WILLIAMS, D.: ‘On-line predictive control of a chemical process using neural network models’, Proc. of IFAC 14th World Congress, pp. 121-126, 1999.

[5] LIGHTBODY, G. and IRWIN, G.W.: ‘Direct neural model reference adaptive control’, IEE Proc.D, pp.

31-43, 1995.

[6] WARWICK, K., IRWIN, G.W. and HUNT, K.J. (Eds.): 'Neutral networks for control and systems' (Peter Peregrinus, Stevenage, UK, 1992.

[7] JAGANNATHAN, J. and LEWIS, F.L.: 'Discrete-time neural net controller for a class of nonlinear dynamical systems', IEEE Trans. Automatic Contro, pp. 1693-16991, 1996.

[8] FABRI, S. and KADIRKAMANATHAN, V.: 'Dynamic structure neural network for stable adaptive control of non-linear systems', IEEE Trans. Neural Networks, pp. 1151-1167, 1996.

[9] NORIEGA, J.R. and WANG, H.: 'A direct adaptive neural network control for unknown non-linear systems and it's application', IEEE Trans. Neural Networks, pp. 27-34, 1998.

[10] TAN, Y. and CAUWENBERGHE, A.V.: 'Non-linear one-step-ahead control using neural networks: control strategy and stability design', Automatica, pp. 1701-1706, 1996.

[11] BEHERA, L., GOPAL, M. and CHAUDHURY, S.: 'Inversion of RBF networks and application to adaptive control of non-linear systems', IEE Proc. D, pp. 617-624, 1995.

[12] FUNAHASHI, K.: 'On the approximate realization of continuous mappings by neural networks', Neural Networks, pp. 183-192, 1989.

[13] YU, D.L. and YU, D.W.: 'A fast optimization algorithm in neural network based control', Submitted to Electronics Letters, 2000.

[14] LEONARD, J.A. and KRAMER, M.A.: 'Radial basis functions for classifying process faults', IEEE Control Systems Magazine, pp. 31-38, 1991.

[15] YU, D.L., GOMM, J.B. and WILLIAMS, D.: 'A recursive orthogonal least squares algorithm for training RBF networks', Neural Processing Letters, pp.167-176, 1997.

[16] SENBORG, D.E., EDGAR, T.F. and MELLICHAMP, D.A.: 'Process dynamics and control' (John Wiley and Sons Inc., New York, USA, 1989)

저자소개

Yujia Zhai [정회원]

- 2001. Department of Electrical Engineering, Changchun University (Bachelor of Eng.)
- 2004. Department of Electrical Engineering and Electronics, University of Liverpool (UoL) (Master in Information and Intelligence Engineering.)
- 2009. Liverpool John Moores University (LJMU) (Ph.D. in Control Engineering.)

<Research Interest> : Nonlinear Control and Robustness, Automotive Engine Modeling and Dynamics, Analysis Artificial Intelligence