TELKOMNIKA, Vol. 11, No. 7, July 2013, pp. 3611 ~ 3617
e-ISSN: 2087-278X

Adaptive Neural Network Generalized Predictive Control for Unknown Nonlinear System

Lin Niu*, Junsheng Li
College of Engineering, Honghe University, Mengzi, Yunnan, China
Corresponding author, e-mail: cdunl@hotmail.com, hhlijsh@gmail.com

Abstract

This paper presents an adaptive neural control design for a class of unknown nonlinear systems. Novel state variables and the corresponding transform are introduced, such that the state-feedback control of a pure-feedback system can be viewed as the output-feedback control of a canonical system. An adaptive predictor incorporated with a neural network observer is proposed to obtain the future system states predictions, which are used in the control design to circumvent the input delay and nonlinearities. The proposed predictor, observer and controller are all online implemented, and the closed-loop system stability is guaranteed. The conventional backstepping design and analysis for pure-feedback systems are avoided, which renders the developed scheme simpler in its synthesis and application. Practical guidelines on the control implementation and the parameter design are provided. The applicability in nonlinear system is demonstrated by simulation experiments.

Keywords: *adaptive control, neural network, nonlinear system, predictive control*

1. Introduction

Great progress has been witnessed in neural network (NN) control of nonlinear systems in recent years, which has evolved to become a well-established technique in advanced adaptive control. Adaptive NN control approaches have been investigated for nonlinear systems with matching [1-4] and nonmatching conditions [5, 6] as well as systems with output feedback requirement [7-11]. The main trend in recent neural control research is to integrate NN, including multilayer networks [2], radial basis function networks [12], and recurrent 1ones [13-16], with main nonlinear control design methodologies. Such integration significantly enhances the capability of control methods in handling many practical systems that are characterized by nonlinearity, uncertainty, and complexity.

It is well known that NN approximation-based control relies on universal approximation property in a compact set in order to approximate unknown nonlinearities in the plant dynamics. The widely used structures of neural network based control systems are similar to those employed in adaptive control, where a neural network is used to estimate the unknown nonlinear system, the network weights need to be updated using the network's output error, and the adaptive control law is synthesized based on the output of networks.

Therefore the major difficulty is that the system to be controlled is nonlinear with its diversity and complexity as well as lack of universal system models. It has been proved that the neural network is a complete mapping. Using this characteristic, an adaptive predictive control algorithm is developed to solve the problems of tracking control of the systems.

2. Problem Statement

Assume that the unknown nonlinear system to be considered is expressed by

$$y(t+1) = f(y(t), y(t-1), \cdots, y(t-n), u(t), u(t-1), \cdots, u(t-m)) \tag{1}$$

where $y(t)$ the scalar output of the system, $u(t)$ is the scalar input to the system, $f(\cdots)$ is unknown nonlinear function to be estimated by a neural network, and n and m are the known

Received January 17, 2013; Revised April 5, 2013; Accepted April 15, 2013

structure orders of the system. The purpose of our control algorithm is to select a control signal $u(t)$, such that the output of the system $y(t)$ is made as close as possible to a prespecified setpoint $r(t)$.

Figure 1 shows the overall structure of the closed-loop control system which consists of the system (1), a feedforward neural network which estimates $f(\cdots)$ and a controller realized by an optimizer.

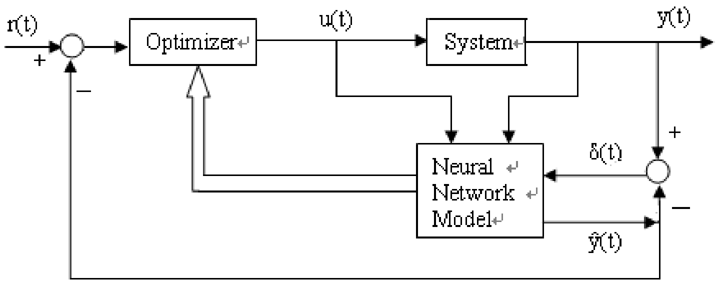


Figure 1. Neural-network control system

Figure 2 shows the neural network architecture. A two-layer neural network is used to learn the system and the standard backpropagation algorithm is employed to train the weights. The activation functions are hyperbolic tangent for the first layer and linear for the second layer. Since the input to the neural network is

$$p = [y(t), y(t-1), \cdots, y(t-n), u(t), u(t-1), \cdots, u(t-m)] \quad (2)$$

The neural model for the unknown system (1) can be expressed as

$$\hat{y}(t+1) = \hat{f}[y(t), y(t-1), \cdots, y(t-n), u(t), u(t-1), \cdots, u(t-m)) \quad (3)$$

Where $\hat{y}(t+1)$ is the output of the neural network and $\hat{f}$ is the estimate of f. Since the backpropagation training algorithm guarantees that

$$[y(t+1) - \hat{y}(t+1)]^2 = \min \quad (4)$$

$\hat{y}(t+1)$ is also referred to as a predicted output of the system (1). Therefore the control signal can be selected such that $\hat{y}(t+1)$ is made as close as possible to $r(t)$.

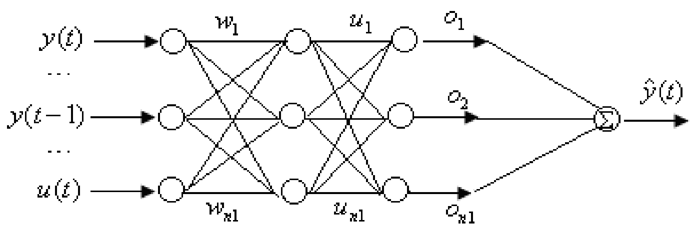


Figure 2. The neural-network structure

3. Adaptive Algorithm

Take an objective function J as

$$J = \frac{1}{2} e^2 (t+1) \tag{5}$$

where $e(t+1) = r(t+1) - \hat{y}(t+1)$ (6)

The control signal $u(t)$ should therefore be selected to minimize J. Using the neural network structure, (3) can be rewritten to give

$$\hat{y}(t+1) = w_2[\tanh(w_1 p + b_1)] + b_2 \tag{7}$$

where w_1, w_2, b_1 and b_2 are the weights and biases matrices of the neural network. To minimize J, the $u(t)$ is recursively calculated via using a simple gradient descent rule

$$u(t+1) = u(t) - \eta \frac{\partial J}{\partial u(t)} \tag{8}$$

where $\eta > 0$ is a learning rate. It can be seen that the controller relies on the approximation made by the neural network. Therefore it is necessary that $\hat{y}(t+1)$ approaches the real system output $y(t+1)$ asymptotically. This can be achieved be keeping the neural network training online. Differentiating (5) with respect to $u(t)$, it can be obtained that

$$\frac{\partial J}{\partial u(t)} = -e(t+1) \frac{\partial \hat{y}(t+1)}{\partial u(t)} \tag{9}$$

where $\partial\hat{y}(t+1)/\partial u(t)$ is known as the gradient of the neural network model with respect to $u(t)$. Substituting (9) into (8), we have

$$u(t+1) = u(t) + \eta e(t+1) \frac{\partial \hat{y}(t+1)}{\partial u(t)} \tag{10}$$

The gradient can then be analytically evaluated by using the known neural network structure (7) as follows:

$$\frac{\partial \hat{y}(t+1)}{\partial u(t)} = w_2[\mathrm{sec}\,h^2(w_1 p + b_1)]w_1 \frac{dp}{du} \tag{11}$$

where $\frac{dp}{du} = [0, 0, \cdots, 0, 1, 0, \cdots 0]'$ (12)

is the derivative of the input vector p respect to $u(t)$. Finally, (10) becomes

$$u(t+1) = u(t) + \eta e(t+1) w_2[\mathrm{sec}\,h^2(w_1 p + b_1)]w_1 \frac{dp}{du} \tag{13}$$

Equation (13) can now be used in a computer program for real-time control. To summarize, we have the adaptive algorithm:

1) produce $\hat{y}(t+1)$ using (7);
2) find $e(t+1)$ using (6);
3) update the weights using backpropagation algorithm;
4) compute new control signal from (13);
5) feed $u(t+1)$ to the system;
6) go to step 1).

4. Adaptive Predictive Control Algorithm

The algorithm described in section 3 can be improved by using the technique in generalized predictive control theory [5], which considers not only the design of the instant value of the control signal but also its future values. As a result, future values of setpoint and the system output are needed to formulate the control signal. Since the neural network model (3) represents the plant to be controlled asymptotically, it can be used to predict future values of the system output. For this purpose, let T be a prespecified positive integer and denot

$$R_{t,T} = [r(t+1), r(t+2), \cdots, r(t+T)]' \tag{14}$$

as the future values of the setpoint and

$$\hat{Y}_{t,T} = [\hat{y}(t+1), \hat{y}(t+2), \cdots, \hat{y}(t+T)]' \tag{15}$$

as the predicted output of the system using the neural network model (7), then the following error vector.

$$E_{t,T} = [e(t+1), e(t+2), \cdots, e(t+T)]' \tag{16}$$

can be obtained where

$$e(t+i) = r(t+i) - \hat{y}(t+i) \tag{17}$$

Defining the control signals to be determined as

$$U_{t,T} = [u(t+1), u(t+2), \cdots, u(t+T)]' \tag{18}$$

and assuming the following objective function

$$J_1 = \frac{1}{2}[E_{t,T}^T E_{t,T}] \tag{19}$$

then our purpose is to find $U_{t,T}$ such that J_1 is minimized. Using the gradient decent rule, it can be obtained that

$$U_{t,T}^{k+1} = U_{t,T}^{k} - \eta \frac{\partial J_1}{\partial U_{t,T}^{k}} \tag{20}$$

where $$\frac{\partial J_1}{\partial U_{t,T}^{k}} = E_{t,T} \frac{\partial \hat{Y}_{t,T}}{\partial U_{t,T}^{k}} \tag{21}$$

and

$$\frac{\partial \hat{Y}_{t,T}}{\partial U_{t,T}^{k}} = \begin{bmatrix} \frac{\partial \hat{y}(t+1)}{\partial u(t)} & 0 & \cdots & 0 \\ \frac{\partial \hat{y}(t+2)}{\partial u(t)} & \frac{\partial \hat{y}(t+2)}{\partial u(t+1)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \hat{y}(t+T)}{\partial u(t)} & \frac{\partial \hat{y}(t+T)}{\partial u(t+1)} & \cdots & \frac{\partial \hat{y}(t+T)}{\partial u(t+T-1)} \end{bmatrix} \tag{22}$$

It can be seen that each element in the above matrix can be found by differentiating (3) with respect to each element in (18). As a result, it can be obtained that

$$\frac{\partial \hat{y}(t+n)}{\partial u(t+m-1)} = \frac{\partial \hat{f}(p)}{\partial u(t+m-1)} + \sum_{i=m}^{n-1} \frac{\partial \hat{f}(p)}{\partial \hat{y}(t+i)} \left[\frac{\partial \hat{y}(t+i)}{\partial u(t+m-1)} \right] \quad (23)$$
$$for \quad n = 1, 2, \cdots, T \quad m = 1, 2, \cdots, T$$

Equation(22) is the well-known Jacobian matrix which must be calculated using (23) every time a new control signal has to be determined. This could result in a large computational load for a big T. Therefore a recursive form for calculating the Jacobian matrix is given in the following so that the algorithm can be applied to the real-time systems with fast responses.

$$\frac{\partial \hat{y}(t+n)}{\partial u(t+m-1)} = \frac{\partial \hat{y}(t+n-1)}{\partial u(t+m-1)} \left[1 + \frac{\partial \hat{f}(p)}{\partial \hat{y}(t+n-1)} \right] \quad (24)$$

From (24) it can be seen that in order to find all the elements in the Jacobian matrix it is only necessary to calculate the diagonal elements and the partial derivatives of the $\hat{f}(\cdots)$ with respect to the previous predicted output. Therefore less calculation is required to formulate the Jacobian matrix. The two derivative terns in (24) are given by

$$\frac{\partial \hat{y}(t+n-1)}{\partial u(t+m-1)} = w_2 \left[\sec h^2(w_1 p + b_1)]w_1 \frac{dp}{du} \right] \quad (25)$$

$$\frac{\partial \hat{f}(p)}{\partial \hat{y}(t+n-1)} = w_2 \left[\sec h^2(w_1 p + b_1)]w_1 \frac{dp}{d\hat{y}} \right] \quad (26)$$

where $\frac{dp}{d\hat{y}} = [1, 0, \cdots, 0, 0, 0, \cdots, 0]'$ (27)

Equations (25) and (26) can now be used in a computer program to calculate the Jacobian matrix and the adaptive predictive algorithm is summarized as follows:

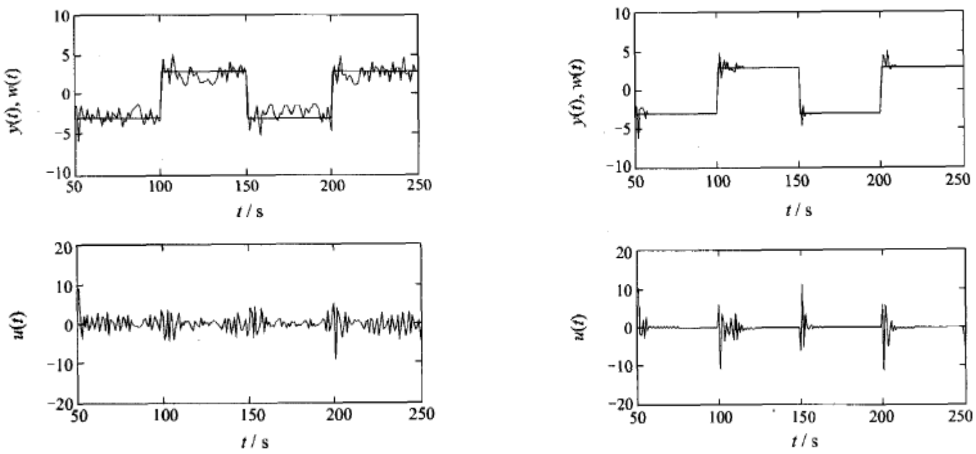
1) Select a T;
2) Find new predicted output $\hat{y}(t+1)$ using (7);
3) Calculate $\partial \hat{y}(t+n-1)/\partial u(t+m-1)$ and $\partial \hat{f}(p)/\partial \hat{y}(t+n-1)$ via (25) and (26);
4) Update vector p, using new $\hat{y}(t+1)$ calculated in step 2 and $u(t+n-1)$ from the vector of future control signals (18);
5) Use (24) and the results obtained in Step 3) to calculate the off-diagonal elements of the Jacobian;
6) Use (20) form a new vector of future control signals;
7) Apply $u(t+1)$ found in step 6) to close the control loop;
8) Return to step 1).

5. Simulation

The following simulation with the adaptive algorithm and the adaptive predictive algorithms gives the situations of systems tracking square-wave. Considering a SISO nonlinear system:

$$y(t+1) = 2.6y(t) - 1.2y(t-1) + u(t) + 1.2u(t-1) + \sin(u(t) + u(t-1) + y(t) + y(t-1) - \frac{u(t) + u(t-1) + y(t) + y(t-1)}{1 + u^2(t) + u^2(t-1) + y^2(t) + y^2(t-1)} \quad (28)$$

Figure 3(a) and (b) show the simulation results with the adaptive control and the adaptive predictive control respectively. It can be seen that the adaptive predictive controller has excellent control performance with better stability and convergence, less study parameters and small calculation



(a) (b)

Figure 3. (a) Adaptive tracking control, (b) Adaptive predictive tracking control

6. Conclusion

A neural network based adaptive control strategy is presented in this paper for general unknown nonlinear systems, where a simplified formulation of the control signals is obtained using the combination of a feedforward neural network and an optimization scheme. The neural network is used online to estimate

The system and the backproagation training algorithm is applied to train the weights. Taking the resulting neural network estimation as a known nonlinear dynamic model for the system, control signals can be directly obtained using the well-established gradient descent rule. An improved algorithm is also included where both instant and future values of control signals are derived. Simulations have successfully demonstrated the use of the proposed method.

Acknowledgements

The project was supported by the Applied Basic Research Programs of Yunnan Provincial Science and Technology Department (Grant No. 2011ZF192) and Honghe University discipline construction foundation (Grant No. 081203) as well as the science foundation of Honghe University (Grant No. 10XJZ104).

References

[1] Lin CM, Ting AB, Chen TY. Neural-network-based robust adaptive control for a class of nonlinear systems. *Neural Computing & Applications*. 2011; 20(4): 557-562.
[2] Hsu CF, Chiu CJ, Tsai JZ. Indirect adaptive self-organizing RBF neural controller design with a dynamical training approach. *Expert Systems with Applications*. 2012; 39(1): 564-571.
[3] Chen M, Ge SS, Ren B. Adaptive tracking control of uncertain MIMO nonlinear systems with input constraints. *Automatica*. 2011; 47(3): 452-460.
[4] Li TS, Li RH, Wang D. Adaptiveneural control of nonlinear MIMO systems with unknown time delays. *Neurocomputing*. 2012; 78(1): 83-89.
[5] Zhang TP, Zhu QQ. Adaptive neural network control of nonlinear time-varying delay systems. *Control and Decision*. 2011; 26(2): 263-269.

[6] Yu J, Chen B, Yu HS, Gao JW. Adaptive fuzzy tracking control for the chaotic permanent magnet synchronous motor drive system via backstepping. *Nonlinear Analysis: Real World Applications*. 2011; 12(1): 671-678.
[7] Yao SJ. Adaptive neural output-feedback control for a class of non-linear systems with unknown time-varying delays. *Control Theory & Applications*. 2012: 6(1): 130-137.
[8] Wang HQ, Chen B, Lin C. Direct adaptive neural control for strict-feedback stochastic nonlinear systems.*Nonlinear Dynamics*. 2012; 67(4): 2703-2709.
[9] Wang T, Tong SC, Li YM. Adaptive neural network output feedback control of stochastic nonlinear systems with dynamical uncertainties. *Neural Computing & Applications*. DOI: 10.1007/s00521-012-1099-7Online First™. 2012.
[10] Hayakawa T, Haddad WM, Hovakimyan N. Neural Network Adaptive Control for a Class of Nonlinear Uncertain Dynamical Systems With Asymptotic Stability Guarantees. *IEEE Transactions on Neural Networks*. 2008; 19(1): 80-88.
[11] Chen PC, Lin PZ, Wang CH, Lee TT. Robust Adaptive Control Scheme Using Hopfield Dynamic Neural Network for Nonlinear Nonaffine systems. *LNCS*. 2010; 6064: 497-505.
[12] Zhu YH, Feng Q, Wang JH. Neural Network-based Adaptive Passive Output Feedback Control for MIMO Uncertain System. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(6): 1263-1272.
[13] Li JJ, Qu R, Chen Y. Construction Equipment Control Research Based on Predictive Technology. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(5): 960-967.
[14] You J, Yang QM, Lu JG. Nonlinear MPC Design for Identified Nonlinear Parameter Varying Model. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2012; 10(3): 514-523.
[15] Deng H, Li HX, Wu YH. Feedback-Linearization-Based Neural Adaptive Control for Unknown Nonaffine Nonlinear Discrete-Time Systems. *IEEE Transactions on Neural Networks*. 2008; 9(9): 1615-1621.
[16] Li T, Feng G, Wang D, Tong S. Neural-network-based simple adaptive control of uncertain multi-imput multi-output nonlinear systems. *Control Therory & Application*. 2010; 4(9): 1543-1549.