2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.1.35

Design and Application of Neural Network PID Decoupling Controller

Guijin Han[+]

Automation College, Xi'an University of Posts & Telecommunications, Xi'an, China

Abstract. Aiming at the multivariant, nonlinear, time variat-ion and strong coupling plant, and neural network PID decoupling controller is proposed. Firstly, a DRNN-PID controller is constructed based on diagonal recurrent neural network and then several DRNN-PID controllers are adopted in parallel as the neural network decoupling controller, which accomplish decoupling and controlling simultaneously. Finally, the stability condition of the controller is presented based on the Lyapunov theory. Simulation results show that the proposed decoupling controller is effectively.

Keywords: PID; decoupling controller; DRNN; stability condition

1. Introduction

For MIMO nonlinear systems, due to the coupling among various inputs and outputs, the control problem is very complicated. It is well known that the neural network owns the capability of self-learning and self-organization, and theoretical works have proven that, even with one hidden layer, neural networks can uniformly approximate any continuous function over a compact domain [1], so it is suitable for the nonlinear coupling control system. Over the past several years, neural networks, especially feed-forward networks, have been widely applied to identify and control nonlinear system. Compared to feed-forward neural network, because Recurrent neural networks(RNN) owns feedback neurons in hidden layers, it is more suitable to identify and control dynamic systems [2][3]. In this paper, we use an RNN's simplification form-Diagonal Recurrent Neural Network (DRNN) as plant identifier.

Up to now, PID control algorithms are still used widely in industrial control systems because of simple control structure, and many new types of PID controller have been proposed[4][5]. In this paper, we construct a nonlinear PID controller based on DRNN and adopt several controllers in parallel as the neural network decoupling controller. The neural network decoupling controller not only can overcome the drawback of difficulty in tuning the parameters of conventional PID controller, but also can accomplish decoupling function. Finally, using the Lyapunov stability theory, the stability condition of the controller is analyzed theoretically. Simulation shows that the algorithm is effective and practical for nonlinear multivariable process control.

2. Diagonal Recurrent Neural Network

2.1 Ntwork structure

The DRNN structure is shown in Figure1, which consists of one input layer, one hidden layer and one output layer, where $x(k)\in R^n$ is the input vector, $h(k)\in R^q$ is the hidden layer output vector and $\hat{y}(k)\in R^p$ is the network output vector. $W^h \in R^{q\times(n+1)}$, $W^d \in R^{v\times q}$ and $W^y \in R^{p\times(q+1)}$ are connection weight matrices between input layer and hidden layer, within hidden layer and between hidden layer and output layer, respectively.

[+] Corresponding author.
E-mail address: hgjin123@126.com.

The structure of one hidden layer neuron with feedback is shown in Figure2, the feedback vector within the hidden layer includes 1 to v time lags to give delayed values of the hidden layer output, which is used to internally present the dynamics of the process to be modelled.

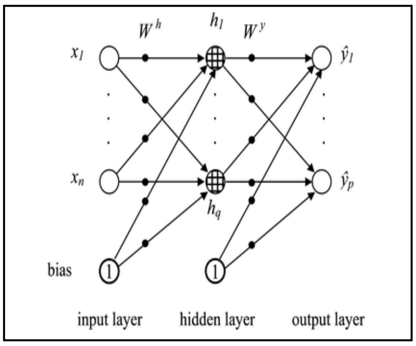

Figure 1. DRNN structure

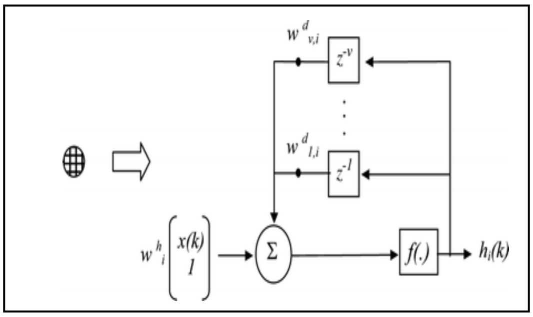

Figure 2. Hidden layer neuron structure

In mathematical terms, the DRNN with q hidden layer nodes used in this research is governed by the following equations.

$$\hat{y}(k) = W^y \begin{bmatrix} h(k) \\ 1 \end{bmatrix} \tag{1}$$

$$h(k) = \frac{1}{1+e^{-z(k)}} \tag{2}$$

$$\begin{aligned} z(k) &= W^h \begin{bmatrix} x(k) \\ 1 \end{bmatrix} + \left[\begin{bmatrix} w_{1,1}^d & & 0 \\ & \ddots & \\ 0 & & w_{1,q} \end{bmatrix} \cdots \begin{bmatrix} w_{v,1}^d & & 0 \\ & \ddots & \\ 0 & & w_{v,q}^d \end{bmatrix} \right] \begin{bmatrix} h(k-1) \\ \vdots \\ h(k-v) \end{bmatrix} \\ &= W^h \begin{bmatrix} x(k) \\ 1 \end{bmatrix} + \left[diag(w_1^d) \cdots diag(w_v^d \right] \begin{bmatrix} h(k-1) \\ \vdots \\ h(k-v) \end{bmatrix} \end{aligned} \tag{3}$$

In (2) a sigmoid function is used as the nonlinear activation function in the hidden layer.

2.2Learning algorithms

Firstly, the target function is defined as follow:

$$J(k) = \frac{1}{2}\left[Y(k) - \hat{Y}(k)\right]^T \left[Y(k) - \hat{Y}(k)\right] \tag{4}$$

where $Y(k)$ is the expected output vector of hidden layer, and $\hat{Y}(k)$ is the real output vector of hidden layer.

Define $E(k) = Y(k) - \hat{Y}(k)$, we can rewrite Eq.(4) as the following:

$$J(k) = \frac{1}{2} E(k)^T E(k) \tag{5}$$

The usual method of modulating weight value of DRNN is gradient descent algorithm with momentum term (GDM). To optimize the algorithm, we add proportional and derivative function into GDM according to PID control theory, then get PIDGDM as Eq.(8) and Eq.(9). We can use Z-D method to adjust parameter η, η_p and η_d, and then adjust parameter ∂, $0 < \partial < 1$ for assuring algorithm's asymptotic stablility[6]. PIDGDM not only has faster track and convergence characteristic than GDM, but also has bigger possibility of jumping out local extremum, which has been proved[7]. The PIDGDM is as follows:

$$\Delta W(k) = -\eta \cdot \frac{\partial J(k)}{\partial W(k)} = -\eta \cdot \frac{\partial E^T(k)}{\partial W(k)} \cdot E(k) \\ = \eta \cdot \frac{\partial \hat{Y}^T(k)}{\partial W(k)} \cdot E(k) \tag{6}$$

$$W(k) = W(k-1) + \Delta W(k) + \eta_p \cdot (\Delta W(k) - \Delta W(k-1)) + \\ \eta_d \cdot (\Delta W(k) - 2\Delta W(k-1) + \Delta W(k-2)) + \partial \cdot (W(k-1) - W(k-2)) \tag{7}$$

where η is the learning rate, η_p, η_d is the proportional and derivative parameter respectively, ∂ is the inertia coefficient.

3. Drnn Identifier

Suppose the controlled nonlinear MIMO system with n inputs and n outputs can be described by a discrete-time equation

$$Y(k) = f\left[Y(k-1),...,Y(k-n_y-1),U(k-1),...,U(k-n_u-1)\right] \tag{8}$$

where $Y(k) = [y_1(k),\cdots,y_n(k)]^T$, $U(k) = [u_1(k),\cdots,u_n(k)]^T$. is output and input vectors respectively. $f(.)$ is a smooth nonlinear function, n_y and n_u are the orders of outputs and inputs. In general, the mathematical model of a plant is unknown and an identifier based on DRNN (NNI) is used

$$\hat{Y}(k) = f_{NN}\left[Y(k-1),...,Y(k-n_y-1),U(k-1),...,U(k-n_u-1),W(k)\right] \tag{9}$$

where $f_{NN}(.)$ is the nonlinear mapping function provided by the DRNN, $W(k)$ is the weight vector of the whole network, and $\hat{Y}(k)$ is the output vector of NNI.

The target functions for identifier as follow:

$$J(k) = \frac{1}{2}\left[Y(k) - \hat{Y}(k)\right]^T\left[Y(k) - \hat{Y}(k)\right] \tag{10}$$

Define $E_I(k) = Y(k) - \hat{Y}(k)$, we can rewrite Eq.(10) as the following:

$$J(k) = \frac{1}{2}E_I(k)^T E_I(k) \tag{11}$$

4. Neural Network PID Decoupling Controller

4.1 Neural network PID decoupling controller

If we set the input vector of DRNN to $X(k) = [x_{i1}(k), x_{i2}(k), x_{i3}(k)] = [e_i(k), \Delta e_i(k), \Delta^2 e_i(k)]$, with $e_i(k) = r_i(k) - y_i(k)$, $\Delta e_i(k) = e_i(k) - e_i(k-1)$, $\Delta^2 e_i(k) = e_i(k) - 2e_i(k-1) + e_i(k-2)$, $e_i(k)$, $r_i(k)$ is the i th reference trajectory. The output of DRNN is the corresponding control signal $u_i(k)$. This can be expressed as $u_i(k) = g_{NN}[e_i(k), \Delta e_i(k), \Delta^2 e_i(k), V_i(k)]$, where $g_{NN}(\cdot)$ is the nonlinear mapping function provided by NNC, $V_i(k)$ s the weight vector of the whole network. So this network can be regarded as a nonlinear PID controller. In the multivariable plant control, n DRNN-PID controllers are adopted in parallel. The structure is shown in Figure3.

The target function for the i th controller as follow:

$$J_i(k) = \frac{1}{2}\left[r_i(k) - y_i(k)\right]^2 + \frac{1}{2}\sum_{j=1}^{n,j\neq i}\left[r_j(k) - y_j(k)\right]^2 \\ = \frac{1}{2}\sum_{i=1}^{n}\left[r_i(k) - y_i(k)\right]^2 \tag{12}$$

In Eq.(14), the second term serve as the decoupling part.

Define $E_C(k) = [e_1(k),\cdots,e_n(k)]^T$, we can rewrite Eq.(12) as the following:

$$J_i(k)=\frac{1}{2}E_C(k)^T E_C(k) \tag{13}$$

we apply PIDGDM as follows to train i th controller

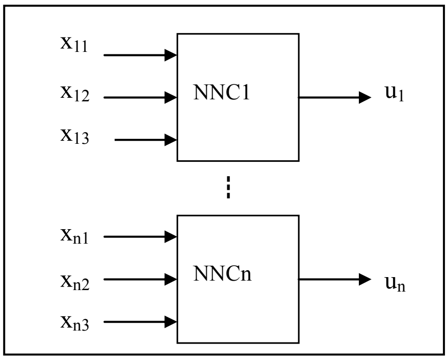

Figure 3. Decoupling Controller Structure

$$\begin{aligned}\Delta V_i(k)&=-\eta_i\cdot\frac{\partial J_i(k)}{\partial V_i(k)}=-\eta_i\cdot\frac{\partial J_i(k)}{\partial u_i(k)}\cdot\frac{\partial u_i(k)}{\partial V_i(k)}\\&=\eta_i\cdot E_C^T(k)\cdot\frac{\partial Y(k)}{\partial u_i(k)}\cdot\frac{\partial u_i(k)}{\partial V_i(k)}\end{aligned} \tag{14}$$

$$\begin{aligned}V_i(k)&=V_i(k-1)+\Delta V_i(k)+\eta_{pi}\cdot(\Delta V_i(k)-\Delta V_i(k-1))\\&+\eta_{di}\cdot(\Delta V_i(k)-2\Delta V_i(k-1)+\Delta V_i(k-2))\\&+\partial_i\cdot(V_i(k-1)-V_i(k-2))\end{aligned} \tag{15}$$

Let $Y_U=\partial Y/\partial U^T$. $\partial Y(k)/\partial u_i(k)$ is the i th column of Y_U , be written as Y_U^i . While we regard $\hat{Y}(k)\approx Y(k)$, then $\hat{Y}_U\approx\hat{Y}$, Eq.(15 can be written as

$$\Delta V_i(k)=\eta_i\cdot E_C^T(k)\cdot\hat{Y}_U^i\cdot\frac{\partial u_i(k)}{\partial V_i(k)} \tag{16}$$

4.2 Control system structure

The control system structure is shown in Figure.4, where the controlled plant is a coupling plant with n inputs and n outputs, NNI is DRNN identifier, NNC is DRNN-PID controller. $u_1,\cdots,u_n$ is control inputs, $y_1,\cdots,y_n$ is real outputs, $\hat{y}_1,\cdots,\hat{y}_n$ is the outputs of NNI, and $r_1,\cdots,r_n$ is reference inputs.

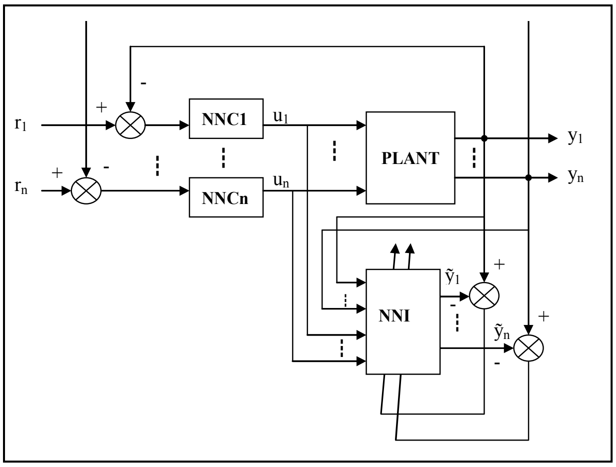

Figure 4. Control System Structure

5. Convergence Analysis

Using the Lyapunov stability theory, the stability condition of the controller can be analyzed theoretically.

Condition Suppose $V_i(k)$ of the i th nonlinear PID controller is updated along Eq. (15), η_i is the learning rate for the weights. Then this neural network will be convergent in exponential speed, if

$$0<\eta_i<\frac{2}{\left\|\hat{Y}_U^i\right\|^2\cdot\left\|\frac{\partial u_i(k)}{\partial V_i(k)}\right\|^2} \tag{17}$$

where $\|\bullet\|$ is Euclidean norm.

Proof Construct discrete-time Lyapunov function

$$L_i(k)=\frac{1}{2}E_C(k)^T E_C(k)$$

Then, the change of the Lyapunov function due to the training process is obtained

$$\Delta L_i(k)=L_i(k+1)-L_i(k)=[E_C(k)+\Delta E_C(k)/2]^T\cdot\Delta E_C(k) \quad \Delta E_C(k)=\frac{\partial E_C(k)}{\partial V_i^T(k)}\cdot\Delta V_i(k)=-\frac{\partial Y(k)}{\partial u_i(k)}\cdot\left[\frac{\partial u_i(k)}{\partial V_i(k)}\right]^T\Delta V_i(k)$$

$$=-\hat{Y}_U^i\cdot\left[\frac{\partial u_i(k)}{\partial V_i(k)}\right]^T\cdot\Delta V_i(k)$$

From Eq. (16), this equation can be written as

$$\Delta E_C(k)=-\eta_i\cdot\hat{Y}_U^i\cdot E_C^T(k)\cdot\hat{Y}_U^i\cdot\left\|\frac{\partial u_i(k)}{\partial V_i(k)}\right\|^2$$

Then

$$\Delta L_i(k)=-\eta_i\cdot[E_C^T(k)\cdot\hat{Y}_U^i]^2\cdot\left\|\frac{\partial u_i(k)}{\partial V_i(k)}\right\|^2+$$

$$\frac{1}{2}\eta_i^2\cdot\left\|\hat{Y}_U^i\right\|^2\cdot\left[E_C^T(k)\cdot\hat{Y}_U^i\right]^2\left\|\frac{\partial u_i(k)}{\partial V_i(k)}\right\|^4$$

To ensure NNC be convergent in exponential speed, it should be $\Delta_i L(k)<0$, which leads to Eq. (17)). This completes the proof.

Equations (17) give an upper bound for the learning rate of NNC to guarantee the neural etworks be convergent in exponential speed.

6. Simlation Exsmole

For testing the proposed control strategy above, we consider the following system with 2-inputs and 2-outputs

$$y_1(k)=1.0/(1+y_1(k-1))^2\cdot(0.8y_1(k-1)+u_1(k-2)+0.2u_2(k-3)) \tag{18}$$

$$y_2(k)=1.0/(1+y_2(k-1))^2\cdot(0.9y_2(k-1)+0.3u_1(k-3)+u_2(k-2)) \tag{19}$$

The structure of NNI are selected as 2-9-2, and the structure of two DRNN-PID controller are all selected as 3-7-1, and $\nu=2$, sampling time $T=1s$. First, we take the setting value of the system output are 1,0 respectively. Simulation results is shown in Figure5 and Figure6 is simulation results using PID decoupling control strategy [5] .

We can see from the Figure5 and Figure6, compared with PID decoupling control, the proposed internal decoupling control based DRNN can trace the setting value faster.

Secondly, we take pulse signal as setting value of the system outputs for testifying validity of the proposed control strategy better. Simulation results as shown in Figure7.

We can see from Figure5 and Figure7, after the initial learn online, both the outputs realize the tracing to the setting value respectively, this achieves the function of the internal model control. At the same time, system realizes the function of the decoupling control. System has better static state and dynamic state characteristic, and the output is stable and the error is small. Above of all the control effects is good, the control strategy is valid.

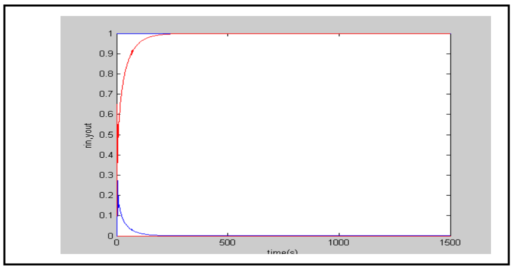

Figure 5. Results of Neural Network PID decoupling control

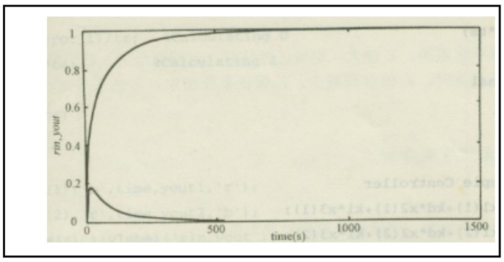

Figure 6. Results of PID decoupling control

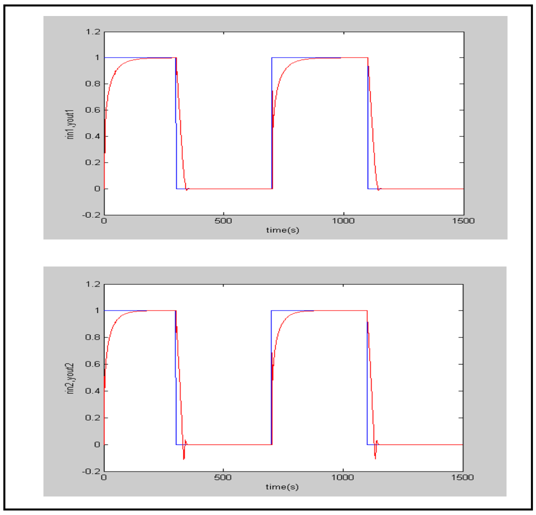

Figure 7. Results of Neural Network PID decoupling control

7. Conclusions

In this paper, a neural network PID decoupling controller based on diagonal recurrent neural network is presented. The main result is that several DRNN-PID controllers are adopted in parallel as the neural network decoupling controller to accomplish decoupling and controlling simultaneously. The stability condition of the controller is presented based on the Lyapunov theory. For testing the validity of proposed control strategy, a nonlinear coupling plant is studied using the proposed control strategy, simulation results show the control strategy is valid.

8. References

[1] Narendra, K.S., Zafiriou, E., “Identification and control of dynamical systems using neural networks”, IEEE Trans.Neural Networks, vol. 1, pp. 4-27, January 1990

[2] Delgado, A., “Dynamic recurrent neural networks for systerm identification and control”, IEE Proc. Control Theory Appl， vol. 142, pp. 307-314, April 1995.

[3] Parlos, A.G., Rais, O.T., Atiya, A.F.,” Multi-step-ahead prediction using dynamic recurrent neural network”, Neural Networks, vol. 13, pp. 765-786, May 2000.

[4] Jinkun Liu, “Advanced PID control and its MATLAB simulation”. Bei Jing: publishing house of electronics industry，2003，pp. 124-156

[5] Chen, Z.Q., Mao, Z.X., Du, Sh.Zh., "Analysis of robustness of PID-GPC based on IMC structure", Chinese J. Chem.Eng., vol. 11,pp. 55-61, January 2003 .

[6] Qing Yang, Xuanju Dang, “A Multivariable Decoupling Control Based DRNN”, Control theory and application, vol. 23, pp. 14-17, March, 2004.

[7] Wenxiu Zhang, “Rough set theory and method”, Bei jing: Science Press, 2003, pp. 100-145.