Adaptive Neural Network-based Synchronization Control for Dual-drive Servo System

Suprapto¹
¹Graduate School of Engineering Science & Technology, National Yunlin University of Science and Technology, Douliou, Yunlin, Taiwan, R.O.C. e-mail: d10210035@yuntech.edu.tw

Wei-Lung Mao²
²Graduate School of Engineering Science & Technology and Department of Electrical Engineering, National Yunlin University of Science and Technology, Douliou, Yunlin, Taiwan, R.O.C. e-mail: wlmao@yuntech.edu.tw

Abstract—In this paper, the neural network control is proposed for master-slave method of dual-drive servo system application. The architecture of control system includes traditional PID, back propagation neural network (BPNN) and radial basis function neural network (RBFNN). The BPNN can adjust three parameters of traditional PID automatically. The RBFNN approximation can determine the characteristics of servo system from given input and output sets. By combining PID, BPNN and RBFNN structure, the adaptive neural network-based method can achieve accurate control of nonlinear systems in synchronization for dual-drive servo. It is shown that the system performance of synchronization control including the speed output, the accuracy and the robustness works well with better dynamic and static characteristics.

Keywords — Back Propagation Neural network (BPNN); Radial Basis function Neural Network (RBFNN); Dual-drive Servo; Synchronization Control.

I. INTRODUCTION

Synchronization control of dual-drive servo system has growth rapidly and plays a very important role in many industrial fields. Based on the connection, synchronization for dual-drive servo system control has two methods. They are an independent and a dependent synchronization [1]. Independent is the connection without physical link, whereas dependent needs physical link or mechanical couplings, such as using gears, shafts, etc. In industrial application, selection of the synchronization servo control is one of a very important factor to achieve an effectiveness and efficiency of the production as well as the maintenance cost.

Due to the some advantages, many industries have changed using synchronization control method without physical link. In [2] the synchronous control realized without physical links will improve tracking performance with software algorithms. Using the method has various benefits such as good flexibility, easy for maintenance, low noise, and low vibration. In addition to those benefits, using software algorithm will have satisfied result.

One of the popular control methods in industrial fields is traditional PID. It has some advantages such as simple structure, good stability and convenient adjustment [3]. In addition the advantages, it also has the disadvantages such as nonlinear control, low precision and static error [4]. The solution to solve the weakness of traditional PID, it must be combined with other method. This paper introduces using Neural Network to adjust three parameters of traditional PID. Combining between its, it will has properties such as ability to uniform approximate arbitrary input-output linear or nonlinear mapping, stability and robust.

Intelligent neural network has properties like ability to uniform approximate arbitrary input-output linear or nonlinear mapping. It is also popular known for its strong capacities of self-learning, self-adapting and self-organization, and it is outstanding for the control of nonlinear systems [5]. In this paper, the control design applied neural network to adjust three parameters of traditional PID control so it gets optimal synchronization control of dual-drive servo. The design of synchronization dual-drive servo control using neural network is comprised two algorithms, one is a back propagation neural network (BPNN) and the other is a radial basis function neural network (RBFNN). The function of BPNN is used for identification between input and output of plant.

According to [6, 7], there are several synchronous control strategies in recent year, master-slave control, cross coupling control, bi-axial cross couple control, electronics line-shafting and relative coupling strategy. Master slave synchronous control method is very simple to be realized, so we apply a algorithm of dual-drive servo method using master slave control. Synchronous control based on BPNN and RBFNN applied on dual-drive servo method using master slave control has faster dynamic response and higher steady precision.

II. DUAL-DRIVE SERVO CONTROL

The synchronous control method comprises of several motors at least two servo de motors. Connection more than one servo on synchronization control method needs a good strategy of control, which it can be achieved by closed loop control using encoder. In this paper, the design applies two servo de motors connected in master-slave mode. According to [8], the main characteristic of master-slave mode is
revolved speed output. The speed output of the master will become the reference value of the slave. The structure of the master-slave of dual servo is shown in Fig. 1.

In master-slave control mode, all instruction from master side transmitted to slave and then slave side will follow the instructions of master side. Of course, if the master side has any disturbance, the slave will reflect and follow it. On the contrary, the master will be never affected the disturbance from the slave.

III. PID CONTROLLER USING BP-NN METHOD

According to [9], there are so many techniques have been used for controller servo motor design. They are PID control, predictive control, optimal control, adaptive control, artificial intelligence control, feedback and feed forward technique. PID controller is widely used in industrial, because of the simplicity and robustness. The problem of traditional PID control can not adapt to time varying of characteristics in wide range.

Designing of combination with traditional PID and artificial intelligence control can improve control performance. Artificial intelligence neural network is chosen to adjust three parameters which the traditional PID has. This paper proposes the adaptive PID control based on BPNN. The equation for traditional PID can be written as:

$$u(k) = u(k-1) + \Delta u(k)$$

where $e(k) = r(k) - y(k)$ is error, $K_P$ is proportional gain, $K_I$ is integral gain, $K_D$ is differential gain and $u(k)$ is the controller output. Using the BPNN, three adjustable parameters can be adjusted in real time according to the condition of the system. The input of BPNN is decided by desire input, error, plant and the outputs are three adjustable parameters of PID controller.

In Fig. 2, suppose a structure of BPNN has three layers neural network. They have input nodes, hidden nodes and output nodes. The input can be defined as $X = [r(k), y(k), e(k), 1]$. The outputs of three parameters of PID controller are $K_P, K_I$ and $K_D$. The state variables can be written as:

$$x_1 = e(k) - e(k-1)$$
$$x_2 = e(k)$$
$$x_3 = e(k) - 2e(k-1) + e(k-2)$$
$$x_4 = u(k-1)$$

where $x_1, x_2, x_3, x_4$ are state variable of $X$, $e(k)$ is error and $u(k)$ is controller output. The relation between input and output of hidden layer as follows:

$$net_i^{(2)}(k) = \sum_{j=0}^{M} W_{ij}^{(2)} O_j^{(1)}$$
$$O_i^{(2)}(k) = f \left( net_i^{(2)}(k) \right), (i = 1,2,\ldots,8)$$

where $W_{ij}^{(2)}$ is the weight of hidden layer, $O_j^{(1)}$ is the output of input layer, $O_i^{(2)}(k)$ is the output of hidden layer and the superscripts of (1), (2), (3) are three layers (input, hidden and output layer). The BPNN structure can be shown in Fig. 2.

The weight coefficient of the input layer and output layer is:

$$net_i^{(3)}(k) = \sum_{j=0}^{Q} W_{ij}^{(3)} O_j^{(2)}(k)$$
$$O_l^{(3)}(k) = g \left( net_l^{(3)}(k) \right), (l = 1,2,3)$$
$$O_1^{(3)}(k) = K_P$$
$$O_2^{(3)}(k) = K_I$$
$$O_3^{(3)}(k) = K_D$$

where $O_l^{(3)}(k)$ is the output. The output $K_P$, $K_I$ and $K_D$ are used auto-adjust traditional PID. The error can be shown as:

$$E_k = \frac{1}{2} \left( d_k - O_k^{(3)} \right)^2$$

where $E_k$ is output error, $d_k$ is desire input and $O_k^{(3)}$ is the output. When the different between desire input and output is bigger, the result is worse. The total error output can be known by this equation:

$$E = \frac{1}{2} \sum_{k=1}^{3} \left( d_k - O_k^{(3)} \right)^2$$
where $d_k$ is desire input and $O^{(3)}_t$ is the output. By the gradient descent to adjust them, the equation can be calculated using a partial derivative respectively as follow:

$$
\Delta W^{(3)}_{li}(k) = -\eta \frac{\partial E(k)}{\partial W^{(3)}_{li}} + \alpha \Delta W^{(3)}_{li}(k - 1)
$$  (15)

where, $\eta$ is the learning rate, $\alpha$ is the inertia coefficient,

$$
\frac{\partial \Delta \delta_i}{\partial w^{(3)}_{li}} = \frac{\partial \Delta \delta_i}{\partial \Delta u(k)} \cdot \frac{\partial \Delta u(k)}{\partial w^{(3)}_{li}},
$$

and while

$$
\frac{\partial \text{net}^{(3)}_i}{\partial w^{(3)}_{li}} = O^{(2)}_l(k)
$$  (17)

So

$$
\frac{\partial \Delta u(k)}{\partial O^{(3)}_1(k)} = x_1 = e(k) - e(k - 1)
$$  (18)

$$
\frac{\partial \Delta u(k)}{\partial O^{(3)}_2(k)} = x_2 = e(k)
$$  (19)

$$
\frac{\partial \Delta u(k)}{\partial O^{(3)}_3(k)} = x_3 = e(k) - 2e(k - 1) + e(k - 2)
$$  (20)

In (16), $\delta^{y(k)}_{\Delta u(k)}$ is unknown, so it can be replaced by $\text{sgn} \left( \delta^{y(k)}_{\Delta u(k)} \right)$, thus

$$
\frac{\partial \delta^{y(k)}_{\Delta u(k)}}{\partial \delta_{\Delta u(k)}} = \text{sgn} \left( \delta^{y(k)}_{\Delta u(k)} \right)
$$

Thus, the weights connection from hidden-layer to output-layer can be written as:

$$
\Delta W^{(3)}_{li}(k) = \alpha \Delta W^{(3)}_{li}(k - 1) + \eta \delta^{(3)}_i O^{(2)}_i(k)
$$  (22)

$$
\delta^{(3)}_i = e(k) \text{sgn} \left( \frac{\delta^{y(k)}_{\Delta u(k)}}{\Delta u(k)} \right) \Delta u(k) \cdot \text{sgn} \left( \frac{\partial \text{net}^{(3)}_i}{\partial \Delta u(k)} \right) \text{for} (l = 1,2,3)
$$

Likewise, the weights connection from input layer to hidden layer can be gained:

$$
\Delta W^{(2)}_{li}(k) = \alpha \Delta W^{(2)}_{li}(k - 1) + \eta \delta^{(2)}_i O^{(1)}_i(k)
$$

$$
\delta^{(2)}_i = f' \left( \text{net}^{(2)}_i(k) \right) \sum_{l=1}^{3} \delta^{(3)}_l W^{(3)}_{li}(k),
$$

(i = 1,..,n - 1)

Where

$$
g'(*) = g(x)[1 - g(x)], \quad f'(*) = \frac{[1 - f^2(x)]}{2}
$$  (26)

IV. DYNAMIC RBFNN IDENTIFICATION NETWORK

According to [10], RBFNN was presented by J. Moody and C. Darken at the end of 1980s. The Radial functions are a special class of functions. The characteristic feature of RBFNN is decrease, increase, or monotonically with distance from a center point [11, 12]. RBFNN has three inputs and three hidden layers. The output is the weight sum of hidden nodes outputs. In Fig. 3, $X = [x_1, x_2, ..., x_k]^T$ is NN’s input.

In RBFNN, $y(k)$ as input $H = [h_1, h_2, ..., h_k]^T$ is the Radial Basis vector of RBF Network and $h_i$ is Gaussian function.

![Fig. 3. RBF Neural Network](image)

The hidden layer’s output of RBFNN can be written as:

$$
h_i = \exp \left( - \frac{||y(k) - c_i||}{2b_i^2} \right)
$$  (27)

Where $i = 1; \ j = 1,..,m; \ b_i > 0; \ c_i = [c_1, ..., c_m]$; and $B = [b_1, b_2, ..., b_m]^T$. The weight vector of RBF NN is expressed as:

$$
W = [w_1, w_2, ..., w_m]^T
$$  (28)

The Output of identifying can be written

$$
\hat{y}_n(k) = \sum_{n=1}^{m} w_n h_n
$$  (29)

$$
y_n(k) = [w_1 h_1 + w_2 h_2 + ... + w_m h_m]^T
$$  (30)

Where $n$ is the number of hidden layer. The criterion function of identifier can be written

$$
J_1 = \frac{1}{2} [y(k) - \hat{y}(k)]^2 = \frac{1}{2} e^2(k)^2
$$  (31)

According to eq. (31), the iterative algorithms of $w$, $b$ and $c$ are shown as below:

$$
w_i(k) = w_i(k - 1) + \eta \frac{\partial e(k)^2}{\partial w_i} h_j
$$

$$
+ \alpha [w_i(k - 1) - w_i(k - 2)]
$$

$$
b_j(k) = b_j(k - 1) + \eta \frac{\partial e(k)^2}{\partial b_j} h_i
$$

$$
+ \alpha [b_j(k - 1) - b_j(k - 2)]
$$

$$
c_i(k) = c_i(k - 1) + \eta \frac{\partial e(k)^2}{\partial c_i} h_i
$$

$$
+ \alpha [c_i(k - 1) - c_i(k - 2)]
$$
\[ c_i(k) = c_i(k-1) + \eta'(k)[y(k) - \hat{y}(k)]w_ih_i \frac{(c_j - c_{ij})}{b^2_j} + \alpha'[c_{ij}(k-1) - c_{ij}(k-2)] \] (34)

where \( i \in \{1, 2, 3\}, j = \{1, 2, 3\} \), \( \eta' \) is the learning rate, and \( \alpha' \) is the momentum factor. The gradient parameter \( \frac{\partial \hat{y}(k)}{\partial \Delta u(k)} \) as:

\[ \frac{\partial \hat{y}(k)}{\partial \Delta u(k)} = \sum_{j=1}^{3} w_ih_i \frac{c_j - \Delta u(k)}{b^2_j} \] (35)

V. SIMULATION RESULTS

The design of synchronization of dual-drive servo system based on adaptive NN can be depicted as Fig. 1. In Fig. 4., the one side block diagram of synchronization dual-drive servo control comprises a controller using BPNN and an identification using RBFNN. The controller is used to adjust three parameters of PID. The identification network is used to determine the characteristics of dual-drive servo control using a transfer function as controlled object.

The complete block diagram of synchronization dual-drive servo divided into master and slave side. The master side is given an input signal. The output of master side become the input of slave side. From the properties of the master-slave synchronization dual-drive servo control, the output of slave will be decided by master side, on contrary the master side is not decided by the slave side.

To know the ability of design, we assume two servos as controlled object for simulation. The transfer function of the controlled object is expressed as follows:

\[ G(s)_1 = \frac{1.0025129}{1.28295s + 1}, G(s)_2 = \frac{1.002362}{1.28295s + 1} \] (36)

The simulation of design, a step math function signal is given on the input of master, a random value is used for initialized the weights of neural network, the learning rate \( \eta \) is taken 0.1, the inertia coefficient \( \alpha \) is given = 0.05 and time sample is 0.02s.

By the given value, the output of three parameters adjusted traditional PID of master will be depicted in Fig. 5. In addition to output of \( K_P \), \( K_I \) and \( K_D \), the output signal of auto-tuning PID controller in Fig. 6. The outputs of them is given after the system physical parameters are selected.

In Fig. 7, the output of synchronization of dual-drive servo in master-slave method is shown. The output has three lines, one is green color, the others are red and blue color. Green line is the desire output, red line is output signal of master side and blue line is output signal of slave side. Both red line and blue line are different value at starting point, but after some seconds, they will same at the track.
From the output as Fig. 7, the quality of control can be calculated using Integral Absolute Error (IAE) which is mathematically given by:

$$I_{AE} = \int_0^\infty |e(t)| dt$$

(37)

where $t$ is the time and $e(t)$ is the error. The Error $e(t)$ is calculated as the difference between the desire output and the output either master side or slave side. The average of calculation of IAE on master side is 1.1312 and slave side is 1.6188, whereas the average of IAE of master output and slave output have error 0.4878. The result of IAE calculation can be shown in Fig. 8.

From the calculation of IAEs, errors of the master and slave are very small. It indicates that the design has good characteristics, robust and stable in speed and acceleration. Comparing with traditional PID, adaptive neural network using both BPNN and RBFNN adjusted PID is better because it has characteristic to uniform approximate and identify arbitrary input-output. The output of synchronization control at master side will guide at slave side, otherwise the slave side will not influence at master side. The slave follows of master side, either the signal control or disturbance.

VI. CONCLUSIONS

The experiment result proved using MATLAB Simulation tool, adaptive neural network could show the signal output of synchronization control for dual-drive servo. The control comprises BPNN to adjust three parameters of traditional PID and RBFNN to identify base of signal input and output given. Simulation result indicates that the system performance of synchronization for dual-drive control including the speed output, the accuracy and the robustness works well with better dynamic and static characteristics.

VII. ACKNOWLEDGEMENT

This project supported by Signal and Control Laboratory, National Yunlin University of Science and Technology, Taiwan, ROC. The first author would like to thanks for scholarship supported by the Indonesian Directorate General of Higher Education (DIKTI) and Yogyakarta State University, Indonesia.

VIII. REFERENCES