

Application Study of ILC with Fuzzy Neural Network in Shaking Table Control System

Jianqiu Chen^{1,2} Xinzheng Zhang¹ Ping Tan² Fulin Zhou²

¹ Department of Automation, University of Guangdong Technology, Guangzhou 510006, P. R. China

² Guangdong Key Laboratory of Earthquake Engineering & Applied Technique, University of Guangzhou, Guangzhou 510006, P. R. China

Abstract

This paper proposes a new approach to improve the control precision of shaking table control system, in which the fuzzy neural network (FNN) technique and iterative learn control (ILC) are combined and developed a new control technique. A FNN inverse model is built and is identified through a white noise with appropriate peak values and frequency range. Then better control effect is obtained by ILC than Remote Parameter Control (RPC). This proposed technique is capable of improving the system precision and adaptability, and reducing the effect of structural load's dynamic characteristic.

Keywords: FNN, ILC, System identification, RPC, Shaking table

1. Introduction

Shaking table test technology is developing rapidly worldwide with the development of structural test technique. There are a lot of factors affecting the tracking accuracy of earthquake wave input in the shaking table control system, which is nonlinear and affected by load dynamic characteristic, belonging to hydraulic servo control system. The current technique of shaking table control system, namely, Remote Parameter Control (RPC) cannot achieve a good tracking effect, as the tracking error is dependent on the physical parameters of the entire system (including flexibility of foundation, oil retractility of actuator), the types of the hydraulic control system, and load dynamic characteristic, etc. Firstly, the shaking table is simulated with produced artificially signal or recorded signal (using artificial white noise in order to guarantee full driving), and system transfer function of the entire experimental system (including shaking table system and experimental object) is achieved. Secondly, initial driving signal is computed with the system transfer function, with which the shaking table is simulated and the new transfer function and errors are computed to modify the driving signal. Thirdly, with the revising driving signal the experiment is

started. In this paper, adopting fuzzy neural network (FNN) technique and iterative learn control (ILC) improves the control precision of shaking table control system.

2. Identification of FNN inverse model of nonlinear system

Fuzzy neural network (FNN) is a new technology, which combines fuzzy technology with neural network. It overcomes their disadvantages and integrates their advantages. On the one hand, it enhances the model interpreting ability of the neural network by making use of the interpreting ratiocination ability of the fuzzy system; on the other hand, it overcomes the dependences of the fuzzy technology on the advice of experts and the non-self-adaptability of the fuzzy concourse by making use of the self-learning functions of the neural network. And it can close to the inverse mapping of any continuous mapping with any precision. In this paper, we will discuss the structure of the fuzzy neural network used to identify t inverse model of shaking table system, as Fig. 1.

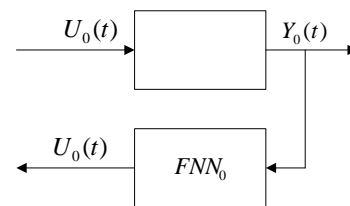


Fig. 1: Identification of FNN inverse model of the shaking table system.

Defines $U(k)$ and $Y(k)$ as Input-Output time series of given system with determined initialization values. The NARMAX model of Input-Output system can be expressed by difference equation, which is a stabilized and d-rank time delay reversible system:

$$Y(k) = g[Y(k-1), \dots, Y(k-n), U(k-d), \dots, U(k-(d+m))] \quad (1)$$

Where, Y is output and U is input. The difference equation of inverse system of the d-rank time delay nonlinear system can be expressed by (2):

$$U(k-d) = g^{-1}[Y(k), Y(k-1), \dots, Y(k-n), U(k-(d+1)), \dots, U(k-(d+m))] \quad (2)$$

The output of fuzzy neural network inverse model is:

$$\hat{U}(k) = Ng^{-1}[Y(k+d), Y(k+d-1), \dots, Y(k+d-n), U(k-1), \dots, U(k-m) : V] \quad (3)$$

Where, V is adjustable parameter of fuzzy neural network identifier.

2.1. Structural identification of fuzzy neural network

For identified system with only input and output data, no other information, clustering algorithm is an effective method constructing the structure of fuzzy system. Commonly, fuzzy c means (FCM) algorithm is used to construct initialization fuzzy rule, however, the clustering sort number of FCM algorithm should be given first, thus need abundant experience for engineers. Therefore, the unsupervised clustering algorithm is employed herein [1]-[2], in which vectors with high relational same characteristics can be grouped into a cluster automatically.

The basic idea of this algorithm is that picking reference vector first, and then searching it in the sample space. If the degree of relation is high, the reference vector is replaced by the mean of these vectors as clustering center. The detail of the algorithm is presented in reference [2].

The unsupervised algorithm can be described as follows:

Let $X = \{x^{(1)}, x^{(2)}, \dots, x^{(p)}\}$ be a set of p vectors in a $(n+1)$ -dimensional sample space, where

$$x^{(k)} = \{x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}, x_{n+1}^{(k)}\} \quad (4)$$

$$(k = 1, 2, \dots, p)$$

is a vector. The preceding n scalars are input vectors of the k sample point, and the $n+1$ scalar is the corresponding output.

Step 1 Define p movable vectors $v^{(k)}$ ($k = 1, 2, \dots, p$) and let $v^{(k)} = x^{(k)}$, that is, $x^{(k)}$ is the initial value of $v^{(k)}$.

Step 2 Calculate the relational grades between the reference vector $v^{(k)}$ and the comparative vector $v^{(l)}$ by

$$r_{kl} = \exp\left[-\frac{\|v^{(k)} - v^{(l)}\|^2}{2b^2}\right], k = 1, 2, \dots, p \quad (5)$$

Where $\|v^{(k)} - v^{(l)}\|$ represents the Euclidean distance between $v^{(k)}$ and $v^{(l)}$, and b is the width of Gaussian function.

Step 3 Modify the relational grades between the reference vector $v^{(k)}$ and the comparative vector $v^{(l)}$ by:

$$r_{k1} = \begin{cases} 0, & \text{if } r_{k1} < \xi \\ r_{k1}, & \text{otherwise} \end{cases} \quad (6)$$

Where ξ is a small constant.

Step 4 Calculate $z^{(k)} = \{z_1^{(k)}, z_2^{(k)}, \dots, z_{n+1}^{(k)}\}$ by:

$$z^{(k)} = \sum_{l=1}^p r_{kl} v^{(l)} / \sum_{l=1}^p r_{kl}, k = 1, 2, \dots, p \quad (7)$$

Step 5 If all the vectors $z^{(k)}$ are the same as $v^{(k)}$, $k = 1, 2, \dots, p$, then go to Step 6; otherwise let $v^{(k)} = z^{(k)}$ and go to Step 2.

Step 6 Based on the final results $v^{(k)}$, we can determine that the number of clusters is equal to the number of convergent vector, the original data with the same convergent vector are grouped into the same cluster, and the convergent vector is the cluster center.

In this way the given data set is grouped into m clusters, and the corresponding cluster centers $c = (c_{1j}, c_{2j}, \dots, c_{nj}, c_{n+1,j})$ $j = 1, 2, \dots, m$ are obtained, thus the rule-base of an initial fuzzy model can be constructed as follows:

$$R^j: \text{If } x_1 \text{ is } A_1^j(x_1) \\ \dots, \\ x_n \text{ is } A_n^j(x_n) \text{ and} \\ \text{then } \omega_j, \\ j = 1, 2, \dots, m \quad (8)$$

Where m is the number of fuzzy rules, and n is the number of input variables. The subordinate function of the premise part in rules is Gaussian function (corresponded to the membership generation layer of Fig.2).

$$\mu_{ij} = \exp[-((x_i - a_{ij})/b_{ij})^2], \\ (1 \leq i \leq n; 1 \leq j \leq m) \quad (9)$$

Where a_{ij} is the center of the Gaussian function,

$a_j = (a_{1j}, a_{2j}, \dots, a_{nj}) = (c_{1j}, c_{2j}, \dots, c_{nj})$, and b_{ij} is its width, $b_j = (b_{1j}, b_{2j}, \dots, b_{nj})$. b_{ij} can be computed by the

j input data points and the errors of clustering center. The real value ω_j of the consequent part is expressed

as $\omega_j = c_{n+1,j}$.

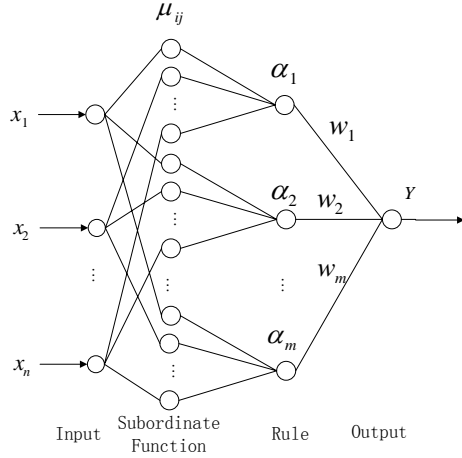


Fig. 2: Fuzzy neural network model.

2.2. Parameter identification of fuzzy neural network

In order to matches with fuzzy reasoning mechanism of the constructed fuzzy model, this paper has designed a 4 layers fuzzy neural network. This network structure is shown in Fig. 2, consisting of input layer, subordinate function layer, rule layer and the output layer (defuzzification layer).

The 1st layer (input layer): each input neuron receives signal x_j , and transmits it to 2nd layer.

The 2nd layer (subordinate function): consists of m groups (expressed m rules), Each group has n neurons (expressed n rule preconditions). The i -th neuron of the j -th group ($i = 1, 2, \dots, n; j = 1, 2, \dots, m$) only connects with the i -th neuron of the 1st layer, and its production subordinate function is expressed by (5).

The 3rd layer (rule layer): Also is premise set layer, which is used for to realize each rule premise match. This layer has m neurons, in which the j -th neuron only receives input coming from the j -th neuron of the 2nd layer, and its output is:

$$\alpha_j = \prod_{i=1}^n \mu_{ij} \quad (10)$$

The 4th layer (output layer): defuzzification layer, or rule set layer. As model total output is the linear combination of all rule conclusions, therefore this neuron output is simply defined as:

$$y = \sum_{j=1}^m \alpha_j \omega_j = \sum_{j=1}^m \omega_j \prod_{i=1}^n \mu_{ij} \quad (11)$$

Although BP algorithm is the most commonly used network study algorithm, this algorithm has many deficiencies in practical application, such as

longer training time, slow convergence speed, and converging to the partial minimum point. This paper uses Levenberg-Marquardt (LM) algorithm [3], which is a union of the gradient drop law and Gauss-Newton, and also an improvement form of Gauss-Newton. This algorithm possesses both the Gauss-Newton's partial astringency and the overall situation characteristic of the gradient drops method. Since the LM algorithm has used approximate second time derivative information, it is much more efficient than the gradient method.

Supposes $\omega^{(k)}$ to express the vector composed by the weight and the threshold value of k time iteration, so the vector $\omega^{(k+1)}$ composed by the new weight and the threshold value can be obtained according to following rules:

$$\omega^{(k+1)} = \omega^{(k)} + \Delta\omega \quad (12)$$

where:

$$\Delta\omega = -[J^T(\omega)J(\omega) + \mu I]^{-1} J(\omega)e(\omega) \quad (13)$$

$$J(\omega) = \begin{bmatrix} \frac{\partial e_1(\omega)}{\partial \omega_1} & \dots & \frac{\partial e_1(\omega)}{\partial \omega_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_n(\omega)}{\partial \omega_1} & \dots & \frac{\partial e_n(\omega)}{\partial \omega_n} \end{bmatrix} \quad (14)$$

The (14) is Jacobian matrix.

The erroneous target function is:

$$E(\omega) = \frac{1}{2} \sum_{i=1}^N e_i^2(\omega) \quad (15)$$

The computation step of LM algorithm is as follows:

- 1) Gives the training error permissible value ε , β , μ_n , as well as the initialization weight and the threshold value vector $\omega^{(0)}$, $k = 0, \mu = \mu_n$;
- 2) Computes network output and erroneous target function $E(\omega^{(k)})$;
- 3) Computes Jacobian matrix $J(\omega)$ according to (14);
- 4) Separately computes $\Delta\omega$ and $E(\omega^{(k)})$ according to (13) and (15);
- 5) If $E(\omega^{(k)}) < \varepsilon$ returns to 7), otherwise, take $\omega^{(k+1)}$ as weight and threshold value computing erroneous target function;
- 6) If $E(\omega^{(k+1)}) < E(\omega^{(k)})$, then makes $k = k + 1$, $\mu = \mu / \beta$, and returns to 2); Otherwise this time does not renew the weight and the threshold value, and makes $\omega^{(k+1)} = \omega^{(k)}$, $\mu = \mu\beta$, then returns to 4);
- 7) Stop.

3. Iterative learning control

The shaking table control system may be regarded as an interactive motion control system. This system satisfies following condition approximately [4]:

- 1) Regarding the known ideal output $Y_d(t)$, the system has only one input $U_d(t)$ inevitably which corresponds with it;
- 2) Each tracking has fixed time-gap ($T > 0$) and fixed ideal track $Y_d(t) t \in [0, T]$;
- 3) System ideal track $Y_d(t)$ is realizable.

Therefore, the iterative learning tool can be used to achieve the control goal for the shaking table control system.

Through the control system inverse model identification mentioned above, the initial input signal $U_0(t)$ of the controlled subject may be obtained by ideal output signal $Y_d(t)$. And from this input signal, the actual output signal of controlled subject $Y_0(t)$ may be measured.

The goal of iterative learning is seeking to a recursion algorithm:

$$U_{k+1}(t) = F(U_k(t), \Delta z_k(t)) \quad (16)$$

where, F represents a kind of recursion algorithm.

With increasing in learning times, k of the control system, the actual output $Y_k(t)$ will converge into the ideal output $Y_d(t)$.

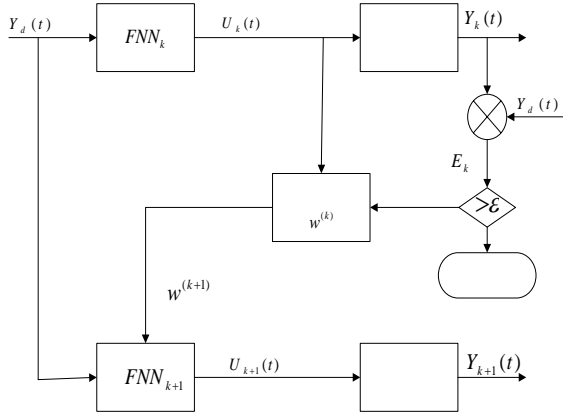


Fig. 3: Iterative learning control strategy of system.

The control strategy of iterative learning control is expressed as follows (Fig. 3):

- 1) Computing error of the k -th time:

$$E_k = \frac{1}{N} \sum_{i=1}^N (Y_d(i) - Y_k(i)) / Y_d \quad (17)$$

in which N is the sampling data length. If the error is within the permissible level ($E_k > \epsilon$), the next step will go on, otherwise, the iteration stops;

- 2) Regarding k -th system output $Y_k(t)$ and input $U_k(t)$ as those of the k -th fuzzy neural network inverse model (FNN_k), and then renew weighting of this model $\omega_i^{(k)}$;
- 3) Calculating $(k+1)$ -th weighting $\omega_i^{(k+1)}$ of the system through FNN_k , the $(k+1)$ -th fuzzy neural network inverse model FNN_{k+1} can be obtained;
- 4) The ideal output $Y_d(t)$ is reused as the input of the $(k+1)$ -th fuzzy neural network inverse model (FNN_{k+1}), so the $(k+1)$ -th input signal $U_{k+1}(t)$ can be computed;
- 5) Measuring the $(k+1)$ -th output $Y_{k+1}(t)$ of system and repeating the first step;
- 6) Stop.

4. Application

For earthquake shaking table control system with flexible load simulation, white noise is used as the exciting input of the system with the frequency range from 1 to 30Hz and the PGA of 0.12g (Fig. 4 and Fig. 5). The white noise input and the output of are used as the input and output of the fuzzy neural network, then the parameters of fuzzy neural network inverse model are determined. The ideal input of the system is calculated through the ideal output of the control system (El-Centro earthquake wave), finally the ideal output in the allowance error scope is obtained after 2 steps iteration. The control performance can be observed from Figure 6 and figure 7, and the control effect comparison between two kinds of control strategies see figure 8 and table 1. In which the relative error based on RPC control strategy is 10%, while the relative error based on FNN and ILC control strategy is 4%, so FNN and ILC control strategy is better than RPC. The proposed procedure can meet the requirement of the earthquake simulation experiments and improves control precision of the original control system.

Control Strategies	RPC	FNN and ILC
Relative Error (%)	10	4

Table 1: The control effect comparison between two kinds of control strategies.

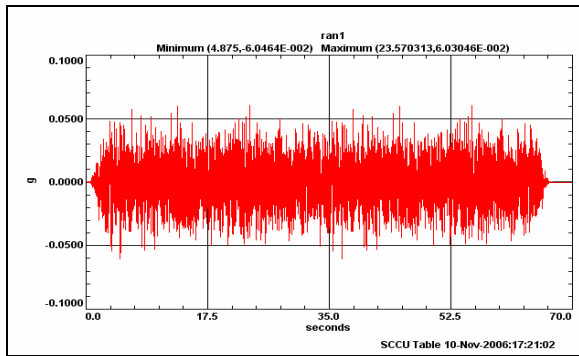


Fig. 4: The time history signal of white noise.

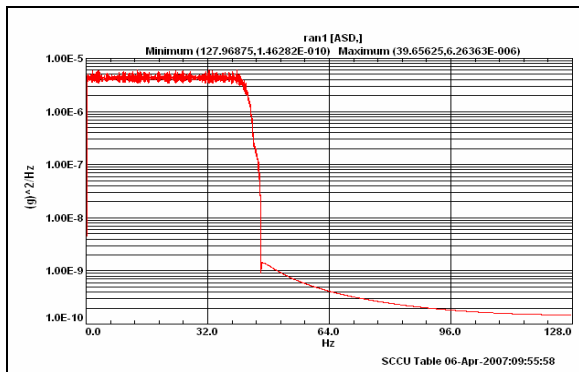


Fig. 5: The frequency domain signal of white noise.

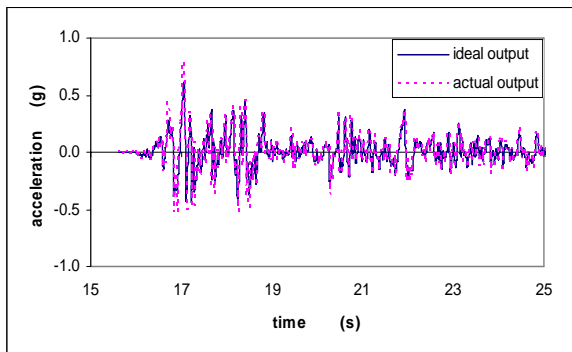


Fig. 6: The control effect based on RPC control strategy.

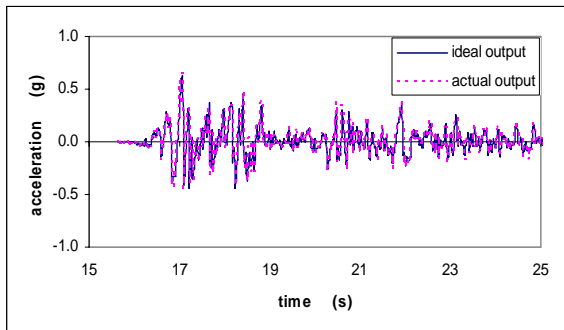


Fig. 7: The control effect based on FNN and ILC control strategy.

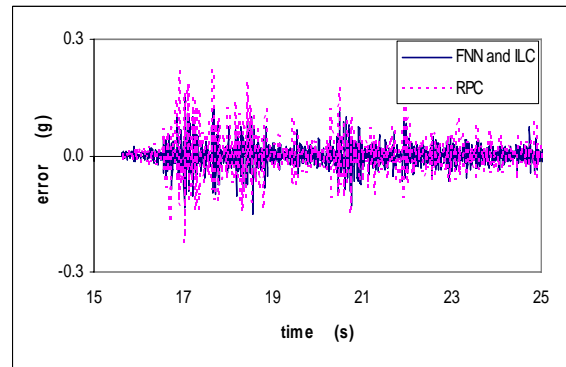


Fig. 8: The control effect comparison between two kinds of control strategies

5. Conclusions

This paper combined the fuzzy neural network technology and the iterative learn control technology into the earthquake shaking table control system and has obtained good control performance. However, the effect of the load structure dynamic characteristic on the control system has not been included in the control algorithm, which must be considered for better performance in the future.

Acknowledgement

This work is partially supported by National Nature Science Foundation of China (Grant No. 60574052), National Nature Science Foundation of China (Grant No. 50578046), Technical Department Special Project of China (Grant No. 2004CCA03300), Guangzhou Science and Technology Plan Project (Grant No. 2004Z1-E0051).

References

- [1] C.C. Wang, C. C. Chen, A hybrid clustering and gradient descent approach for fuzzy modeling, *IEEE Transactions on system, man and Cybernetics- part B*, pp. 686-693, 1999.
- [2] Y. Li, B. D. Bai, L. C. Jiao, A model identification approach of nonlinear systems based on fuzzy neural networks, *Journal of Electronics and Information Technology*, 24: 355-360, 2002.
- [3] H.C. Huang, Y. J. Huang, G. I. Yang, Neural network system identification based on levenberg-marquardt algorithm, *Combined Machine and automatic machining technique*, pp. 6-8, 11, 2003.

- [4] X. Li, Study on nonlinear motion model-free learning algorithm. *Information and Control*, 32: 391-394, 417, 2003.
- [5] J. Nie, Constructing fuzzy model by self-organizing counter propagation network, *IEEE Trans. on System, Man and Cybernetics*, pp. 963-970, 1995.
- [6] Y.B. He, G. R. Yan, J. X. Xu, Neural Network Based Fast and Robust Tracking Control for Electro- Hydraulic Servo Structural Testing System, *Control Theory and Applications*, 16: 776-778, 1999.
- [7] K. Hamamoto, T. Sugie, An iterative learning control algorithm within prescribed Input-output subspace, *Automatica*, 37: 1803-1809, 2001.