Reconstructing Dynamics of Fuzzy PID Controller Based on Elman NN

Bingxiang Zhong\textsuperscript{1} Debiao Wang\textsuperscript{1} Yingying Su\textsuperscript{2} Ling Nie\textsuperscript{1} Taifu Li\textsuperscript{1}

\textsuperscript{1} Dept. of Electronic Information Eng., Chongqing Univ. of Sci. and Tech., Chongqing 400050, P. R. China
\textsuperscript{2} Dept. of Electronic Information and Automation, Chongqing Inst. of Tech., Chongqing 400050, P. R. China

Abstract

Fuzzy PID controller is difficult in implementation for its computational complexity. In this paper, for the dynamic modeling and parallel computation ability of Elman NN (Neural Network), the authors utilized an equivalent Elman NN to accurately approximate the dynamics of a known fuzzy PID controller. Consequently, the same process model was controlled by the fuzzy PID controller and the remodeling Elman NN, respectively. The results show that the control qualities of two different controllers were extremely similar. Thus, the fuzzy PID controller can be simplified to a remodeling Elman NN in purpose of reducing the computational complexity, solving the curse of dimensionality and improving the real-time performance.

Keywords: Fuzzy PID, Elman neural network, Function approximation, Reconstructing, Dynamics

1. Introduction

Thus far, fuzzy PID controllers with adjustable gain or self-regulating is not only used in linear systems, but also effectively applied in lots of controlled process which are complex, nonlinear, high at order, time-delay and so on [1]. In a fuzzy PID controller, three gain coefficients are all two-dimension fuzzy systems with respect to error $e$ and change in error $ec$. It leads to the hard implementation in DSP, VHDL language, and MCU, etc, and the poor real-time performance by industrial computer [2]. Someone has solved the problem with transforming three fuzzy systems $K_p$, $K_i$, $K_d$ into look up table in advance, but it is still hard to implement because of large computation for three fuzzy rule bases.

It is said that NN isn’t only good at nonlinear approximating ability in $L^2$ norm, but also fast in parallel computation, which can decrease computing complexity [3]. It equals that NN is better in real-time performance and easier implemented in parallel machine languages than fuzzy systems. Therefore, in order to utilize the advantages of fuzzy PID controller and effectively overcome its shortcoming, authors have presented methodology of reconstructing dynamics in fuzzy PID controller with NN.

Since conventional NN, such as BP, RBF, etc are actually static mapping model, aimed at reconstructing the dynamics of fuzzy PID controller, a recurrent NN is seriously demanded. Elman NN is dynamic with feedback connections from hidden layer to input. So we reconstructed the dynamics of fuzzy PID controller with Elman NN through lots of dynamic input/output data pairs. The method tries to reduce the computation complexity in fuzzy PID controller, and widen the applications of fuzzy PID with hardware implementation in DSP, VHDL language, etc.

2. Fuzzy PID Controller

In the process of industrial production, the characteristic parameters or structures of many controlled plants would be varying resulted from changing load and disturbances. Nowadays, PID algorithm is still widely applied in industrial production and its parameter tuning is mainly implemented in manual. Combining the classical PID control with the advanced expert system, if all these tuning methods are described with fuzzy rules, PID optimal parameters could be automatically regulated with the help of fuzzy inference engine. A fuzzy PID controller takes error $e$ and change in error $ec$ as inputs, and it could meet requirements of PID parameters self-regulating for different $e$ and $ec$, whose structure [3] is shown in Fig.1.

![Fig.1: Structure of fuzzy PID controller.](image-url)
The PID controller is given by
\[ u(k) = K_p e(k) + K_i \sum e(k) + K_d e\Delta c(k) \]  
(1)

In a fuzzy PID, \( K_p, K_i \) and \( K_d \) are all the fuzzy systems with respect to error \( e \) and change in error \( e\Delta c \). The fuzzy sets and domains \(^4\) of error \( e \) and change in error \( e\Delta c \) are defined in Fig.2, Fig.3:

Fig.2: Membership function of error \( e \)

Fig.3: Membership function of change in error \( e\Delta c \)

Where, \( K_p, K_i \) and \( K_d \) are all composed of initial regulating parameters and modified parts which are fuzzy system with respect to error \( e \) and change in error \( e\Delta c \).

\[ K_p = K_{p0} + \Delta K_p \]
\[ = K_{p0} + f_p(e, e\Delta c) \]  
(2)
\[ K_i = K_{i0} + \Delta K_i \]
\[ = K_{i0} + f_i(e, e\Delta c) \]  
(3)
\[ K_d = K_{d0} + \Delta K_d \]
\[ = K_{d0} + f_d(e, e\Delta c) \]  
(4)

In which, domains and fuzzy sets of \( \Delta K_p, \Delta K_i \) and \( \Delta K_d \) are shown in Fig.4, Fig.5 and Fig.6.

![Fig.4: Membership function of \( \Delta K_p \)](image)

![Fig.5: Membership function of \( \Delta K_i \)](image)

![Fig.6: Membership function of \( \Delta K_d \)](image)

The fuzzy rules of \( \Delta K_p, \Delta K_i \) and \( \Delta K_d \) are shown in Table 1, Table 2 and Table 3:

| e | e\Delta c | \( \Delta K_p \) | \( \Delta K_i \) | \( \Delta K_d \) |
|---|---|---|---|---|
| \( \text{NB} \) | \( \text{NB} \) | \( \text{NB} \) | \( \text{NB} \) | \( \text{NB} \) |
| \( \text{NM} \) | \( \text{PB} \) | \( \text{PB} \) | \( \text{PM} \) | \( \text{PB} \) |
| \( \text{NS} \) | \( \text{PM} \) | \( \text{PM} \) | \( \text{PM} \) | \( \text{PM} \) |
| \( \text{ZO} \) | \( \text{PM} \) | \( \text{PS} \) | \( \text{PS} \) | \( \text{ZO} \) |
| \( \text{PS} \) | \( \text{PS} \) | \( \text{PS} \) | \( \text{ZO} \) | \( \text{NS} \) |
| \( \text{PM} \) | \( \text{ZO} \) | \( \text{ZO} \) | \( \text{NS} \) | \( \text{NM} \) |
| \( \text{PB} \) | \( \text{ZO} \) | \( \text{NS} \) | \( \text{NS} \) | \( \text{NM} \) |

Table 1: Fuzzy rules of \( \Delta K_p \)
Table 2: Fuzzy rules of $\Delta K_p$

| e | NB | NM | NS | ZE | PS | PM | PB |
|---|---|---|---|---|---|---|---|
| NB | PS | NS | NB | NB | NB | NM | PS |
| NM | PS | NS | NB | NM | NM | NS | ZO |
| NS | ZO | ZO | NS | NS | NS | NS | ZO |
| ZO | ZO | NS | NS | NS | NS | PS | PS |
| PS | ZO | ZO | NS | NS | ZO | ZO | ZO |
| PM | PB | PS | PS | PS | PS | PS | PB |
| PB | PB | PM | PM | PM | PM | PS | PB |

Table 3: Fuzzy rules of $\Delta K_i$

| e | NB | NM | NS | ZE | PS | PM | PB |
|---|---|---|---|---|---|---|---|
| NB | NB | NB | NM | NM | NS | ZO | ZO |
| NM | NB | NB | NM | NS | NS | ZO | ZO |
| NS | NB | NM | NS | NS | ZO | PS | PS |
| ZO | NM | NM | NS | ZO | PS | PM | PM |
| PS | NM | NS | ZO | PS | PS | PM | PB |
| PM | ZO | ZO | PS | PS | PM | PB | PB |
| PB | ZO | ZO | PS | PS | PM | PB | PB |

Three fuzzy systems of $\Delta K_p$, $\Delta K_i$, and $\Delta K_d$ are all chosen as singleton fuzzification, center average defuzzification, Mamdani product inference engine, correspondingly mathemathic expressions are shown in (5), (6) and (7).

$$\Delta K_p = f_x(e, ec)$$
$$\Delta K_p = \sum_{i=1}^{m} \mu_{E_i} (e) \cdot \mu_{E_{ec}} (ec)$$

(5)

$$\Delta K_i = f_x(e, ec)$$
$$\Delta K_i = \sum_{i=1}^{m} \mu_{E_i} (e) \cdot \mu_{E_{ec}} (ec)$$

(6)

$$\Delta K_d = f_x(e, ec)$$
$$\Delta K_d = \sum_{i=1}^{m} \mu_{E_i} (e) \cdot \mu_{E_{ec}} (ec)$$

(7)

3. Reconstructing dynamics of Fuzzy PID Controller with an equivalent Elman NN

For a conventional Fuzzy-PID controller above-mentioned, the inputs are usually error and change in error. The adjustable parameters $K_p$, $K_i$, $K_d$ are all the fuzzy systems related to them, which leads to the computation complexity. It results in the time-delay and the implementation problems in hardware. According to Kolmogorov theorem, an Elman NN with one hide-layer could approach every nonlinear continuous mapping $\mathbb{R}^n \rightarrow \mathbb{R}^n$ [5]. Furthermore, Elman NN is a dynamic modeling method with feedback connections from the outputs of hidden layer to inputs. In this way, Elman NN could effectively approximate the dynamics of Fuzzy PID controller and simplify the computation complex. The reconstructing diagram is shown in Fig. 7. Besides, the structure of Elman NN is in Fig. 8.

Fig.7: The diagram of reconstructing fuzzy-PID with Elman NN Estimator

Fig.8: The structure of Elman NN

Suppose that $o(t)$ indicates the weighted sum of input $e(t)$ and feedback $h_i(t)$ from hidden-layer with one order time-delay, in this way the output $u(t)$ of the Elman NN [6] is:

$$u(t+1) = \frac{1}{2} f(c_{W} \cdot h(t+1) + \gamma h)$$

(8)

where, $h(t+1) = \frac{1}{2} f(o(t+1))$

$$o(t+1) = W \cdot e(t) + W \cdot h_i(t) + \beta$$
For an Elman NN, there are usually two layers. The first layer is hidden layer with nonlinear continuous functions, such as 'sigmoid', 'log-sigmoid'. The last layer is linear. Here, we chose sigmoid and linear as the transfer function, respectively. In other words,

$$1 f(x) = \frac{2}{1 + e^{-x}} - 1$$
$$2 f(x) = x$$

The error energy function is:

$$J(t) = \sum_{t=1}^{n} E(t) = \frac{1}{2} \sum_{t=1}^{n} (y_j(t) - y(t))^2 = \frac{1}{2} \sum e^2(t)$$

where, $y_j(t)$, $y(t)$ are the target and the output of Elman NN, respectively.

The modified method here adopts the dynamic back propagation algorithm given as following:

$$\Delta^1w_{ij} = -\eta \delta_j \mathcal{O}_i(t) = -\eta \delta_j f(x(t)) \mathcal{O}_i(t)$$
$$\Delta^2w_{ij} = -\eta \delta_j \mathcal{W}_i(t) \cdot f'(x(t)) \mathcal{W}_i(t) = -\eta \delta_j \mathcal{W}_i(t) \frac{\partial x(t)}{\partial w_{ij}}$$
$$\Delta^1w_{ij} = -\eta \delta_j \mathcal{W}_i(t) \cdot f'(x(t)) \mathcal{W}_i(t) u_j(t-1)$$
$$= -\eta \delta_j \mathcal{W}_i(t) u_j(t-1)$$

where, $\delta_j(t) = \frac{\partial E(t)}{\partial x_j(t)} = \delta(t) r_{ij}(t)$.

In order to reconstruct fuzzy PID controller, complete training data pairs of 441 were effectively obtained from the input/output of fuzzy PID controller. The input of Elman NN is error between target and actual output, and the output is control variable. With the above discussion, some Elman NNs were trained when increasing the neurons of hidden layer. In the end, comprehensively considering the training accuracy and convergence speed, the structure is determined as 1(10)-10-1. The training processing is shown in Fig.9, where the final error square sum is $\sum_{i=1}^{441} (y_j(i) - y(i)) = 0.0907$.

$$i \mathcal{W}_{ij}(j = 1,2,3,109,110,i = 1) = [0.7031 0.2933 0.1075 0.0450 0.2695 -0.3310 0.4220 0.3446 -0.1238 -0.1354 0.1706 -0.3267 0.0751 -0.4123 0.0258 0.0978 0.0379 -0.2787 -0.3113 -0.2362 0.3200 0.4077 -0.4134 0.0691 0.0951 -0.3598 0.3133 0.3153 -0.4880 0.2589 -0.1204 0.0567 -0.3971 -0.0840 -0.1619 0.4970 -0.4971 0.0982 -0.0722 -0.4006 -0.3043 -0.0953 0.3928 -0.3016 0.0318 -0.0880 -0.0779 0.1328 0.4203 -0.4300 0.2158 -0.3003 -0.3264 -0.2888 -0.2953 -0.0662 -0.3245 -0.0988 -0.3440 -0.1465 -0.1221 -0.1460 0.5727 -0.1894 0.0029 0.4365 -0.0369 0.0545 0.3227 0.0819 0.011 -0.0102 0.0733 -0.2874 -0.3028 0.0491 -0.4303 -0.1368 -0.1542 -0.1493 0.5365 0.1667 0.7184 -0.1189 -0.2445 0.3510 -0.1417 0.3386 0.2606 0.2846 0.3697 -0.0002 0.0364 0.6670 0.2678 -0.4260 0.2248 -0.0967 -0.0148 0.0459 0.3512 -0.2543 0.3197 -0.0663;$$

$$i \mathcal{b}_{ij}(k = 1, j = 1,2,3,109,110) = [-0.2042 -0.2665 0.6140 -0.3381 -0.6411 0.1973 -0.4096 0.0416 0.8089 0.7703];$$

$$i \mathcal{b}_{ij}(k = 1) = [0.2802];$$

Here, training NN has completed. Then, we will verify model precision of the equivalent Elman NN approximating the fuzzy PID. So we may get outputs of fuzzy PID controller and that of the equivalent Elman NN model with the same input data, which are represented with symbol $u_{F-PID}$ and $u_{Em-NN}$, respectively in Fig.10.

![Fig. 10: Verification of equivalent Elman NN](image)

In Fig.10, the outputs from two different controllers are mainly similar, which indicates that the equivalent Elman NN can satisfy the precision requirement. Because the data is so dense in the figure, it isn't easy to directly observe the equivalence, we
further plot an approximation error curve between the two controllers in Fig.11.

Fig. 11: Error curve between $u_{PID}$ and $u_{Elm-NN}$

In Fig.11, the error between fuzzy PID and equivalent Elman NN doesn’t mainly exceed 0.2. It was only about $\frac{0.2}{20} \times 100\% \approx 0.5\%$ of output in Fig.10. Thus, the precision of equivalent Elman NN model can satisfy the control requirement. It can accurately reconstruct the dynamics of fuzzy PID controller.

4. Control Simulation

In order to farther verify the purpose of remodeling the PID controller with an equivalent Elman NN, we chose the same plant, and controlled it with the two controllers, respectively. The transfer function of plant is given by equation (11):

$$G(s) = \frac{523500}{s^3 + 87.35s^2 + 10470s} \quad (11)$$

The fuzzy PID controller and the equivalent Elman NN model were used to trace the step input with sample interval 1ms, respectively. The simulation figures in step response are shown in Fig.12 to Fig.14.

In Fig.12, the broken line and real line represent the step response curves of fuzzy PID and the equivalent Elman NN, respectively. The two lines are almost coincident, which indicated that the control effects from two controllers are extremely similar.

Fig.12: The response curves from fuzzy PID and the equivalent Elman NN in step input

In Fig.13, the broken line and real line represent the step response error curves of fuzzy PID and the equivalent Elman NN, respectively. The error curves from two different controllers are also extremely similar, so it farther verifies the equivalent control effects between them.

Fig.13: The error curves from fuzzy PID and equivalent Elman NN in step input

In Fig.14, the broken line and real line represent the output control signal curves of fuzzy PID and the equivalent NN in step response, respectively. The two lines are still very close, which farther verify the equivalent control effects between them.

Fig.14: The control output from fuzzy PID and equivalent Elman NN in step input.

In Fig.14, the broken line and real line represent the output control signal curves of fuzzy PID and the equivalent NN in step response, respectively. The two lines are still very close, which farther verify the equivalent control effects between them.

5. Conclusions
In this paper, in order to decrease the complex implementation in fuzzy PID controller, an effectively dynamic modeling method with Elman NN is applied to accurately approximate a known fuzzy PID controller. Model precision verification and control simulation show that an equivalent Elman NN is extremely similar to a known fuzzy PID controller. In this way, a fuzzy PID controller may be simplified to an equivalent Elman NN model to solve actual control complexity.

Therefore, the study, which is to reconstruct the dynamics of fuzzy PID controller with an equivalent Elman NN can utilize the advantages of fuzzy PID controller, effectively avoid its computational complexity, and widen its applications in hardware implementation with DSP, VHDL language, etc.

Acknowledgement

The authors gratefully acknowledge the support of Chongqing Educational Committee Sci. & Tech. Research Project (Grant No. KJ071411).

References

[1] Z.Y. Zhao , M.Tomizuta, and S.Isaka, Fuzzy gain scheduling of PID controllers, *IEEE Trans. on Systems, Man, and Cybernetics*, 23: 1392-1398, 1993.

[2] T.F. Li, B. Guo and R.M. Deng, Exploration into integrated realization of a special kind of complex system, *Advances in Systems Science and Applications*, 4: 226-230, 2004.

[3] T.F. Li, Y.Y. Su and B.X. Zhong, Remodeling for fuzzy PID controller based on neural networks, *Advance in Soft Computing*, Springer-Verlag, pp.714-725, 2007.

[4] C. T. Chao, Y. J. Chen and C.C. Teng, Simplification of fuzzy-neural systems using similarity analysis, *IEEE Trans. On Systems, Man, and Cybernetics*, 26: 344-354, 1996.

[5] L. X. Wang, Analysis and design of hierarchical fuzzy systems, *IEEE Trans. On Fuzzy Systems*, 7: 617-624, 1999.

[6] T.F. Li, Y.J. Shen and B.W. Wang, Identification of nonlinear dynamical system by using neural network with multi-output neural model, *Advances in Systems Science and Applications*, 4: 456-461, 2004.