Reconstructing Dynamics of Fuzzy PID Controller Based on Elman NN

Bingxiang Zhong¹ Debiao Wang¹ Yingying Su² Ling Nie¹ Taifu Li¹

¹ Dept. of Electronic Information Eng., Chongqing Univ. of Sci. and Tech., Chongqing 400050, P. R. China
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 Kp, Ki, Kd 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.
The PID controller is given by
\[ u(k) = K_p e(k) + K_i \sum e(t) + K_d ec(k) \quad (1) \]

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 \( ec \).
The fuzzy sets and domains [4] of error \( e \) and change in
error \( ec \) are defined in Fig.2, Fig.3:

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 \( ec \).

\[
K_p = K_{p0} + \Delta K_p \\
= K_{p0} + f_p(e, ec) \quad (2)
\]

\[
K_i = K_{i0} + \Delta K_i \\
= K_{i0} + f_i(e, ec) \quad (3)
\]

\[
K_d = K_{d0} + \Delta K_d \\
= K_{d0} + f_d(e, ec) \quad (4)
\]

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.

<table>
<thead>
<tr>
<th>e</th>
<th>ec</th>
<th>( \Delta K_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NB</td>
<td>NB</td>
</tr>
<tr>
<td>NM</td>
<td>PB</td>
<td>PB</td>
</tr>
<tr>
<td>NS</td>
<td>PM</td>
<td>PM</td>
</tr>
<tr>
<td>ZO</td>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>PS</td>
<td>PS</td>
<td>PS</td>
</tr>
<tr>
<td>PM</td>
<td>ZO</td>
<td>ZO</td>
</tr>
<tr>
<td>PB</td>
<td>ZO</td>
<td>ZO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>e</th>
<th>ec</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NB</td>
</tr>
<tr>
<td>NM</td>
<td>NB</td>
</tr>
<tr>
<td>NS</td>
<td>NB</td>
</tr>
<tr>
<td>ZO</td>
<td>NB</td>
</tr>
<tr>
<td>PS</td>
<td>NB</td>
</tr>
<tr>
<td>PM</td>
<td>NB</td>
</tr>
<tr>
<td>PB</td>
<td>NB</td>
</tr>
</tbody>
</table>

### Table 3: Fuzzy rules of $\Delta K_d$

<table>
<thead>
<tr>
<th>e</th>
<th>ec</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NB</td>
</tr>
<tr>
<td>NM</td>
<td>NB</td>
</tr>
<tr>
<td>NS</td>
<td>NB</td>
</tr>
<tr>
<td>ZO</td>
<td>NB</td>
</tr>
<tr>
<td>PS</td>
<td>NB</td>
</tr>
<tr>
<td>PM</td>
<td>NB</td>
</tr>
<tr>
<td>PB</td>
<td>NB</td>
</tr>
</tbody>
</table>

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 mathematic expressions are shown in (5), (6) and (7).

\[
\Delta K_p = f_p(e, ec) = \frac{\sum_{i=1}^{49} \Delta K_{p,i}' \cdot \mu_{e,i}(e) \cdot \mu_{ec,i}(ec)}{\sum_{i=1}^{49} \mu_{e,i}(e) \cdot \mu_{ec,i}(ec)} \tag{5}
\]

\[
\Delta K_i = f_i(e, ec) = \frac{\sum_{i=1}^{49} \Delta K_{i,i}' \cdot \mu_{e,i}(e) \cdot \mu_{ec,i}(ec)}{\sum_{i=1}^{49} \mu_{e,i}(e) \cdot \mu_{ec,i}(ec)} \tag{6}
\]

\[
\Delta K_d = f_d(e, ec) = \frac{\sum_{i=1}^{49} \Delta K_{d,i}' \cdot \mu_{e,i}(e) \cdot \mu_{ec,i}(ec)}{\sum_{i=1}^{49} \mu_{e,i}(e) \cdot \mu_{ec,i}(ec)} \tag{7}
\]

#### 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 $R^n \rightarrow R^n$ [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.

![Reconstructing diagram](image)

![Structure of Elman NN](image)

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:

\[
u(t+1) = f(h(t+1))
\]

where, $h(t+1) = f(o(t+1))$

\[
o(t+1) = W \cdot e(t) + W \cdot h_i(t) + b
\]
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,

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

\[ f(x) = x \]

The error energy function is:

\[ J(t) = \sum_{i=0}^{t} E(t) = \frac{1}{2} \sum(y_d(t) - y(t))^2 = \frac{1}{2} \sum e^2(t) \]

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

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

\[ \Delta^2 w_j = -\eta \delta_j \cdot \delta_i \cdot \frac{\partial e}{\partial \delta_i} \]

\[ \Delta^1 w_j = -\eta \delta_j \cdot \delta_i \cdot \frac{\partial e}{\partial \delta_i} \]

\[ \Delta^1 b_j = -\eta \delta_j \cdot \delta_i \cdot \frac{\partial e}{\partial \delta_i} \]

where, \( \delta_i \) is the error in the output layer.

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 \( \sum_{i=1}^{441}(y_i - y(t))^2 = 0.0907 \).

\[ \lambda = h(t-1) = \int f(\lambda(t-1)) \]

![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 $u_{f-PID}$ 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 $\frac{0.2}{20} \times 100\% = 0.5\%$ 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):

$$G(s) = \frac{523500}{s^3 + 87.35s^2 + 10470s}$$ (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.13: The error curves from fuzzy PID and 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.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


