View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 54 (2015) 463 – 471
Procedia Computer Science

Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Artificial Neuro Fuzzy Logic PID Controller based on BF-PSO Algorithm

Rahul Patel* and Vijay Kumar

Department of Electronics and Communication Engineering, IIT Roorkee, Roorkee 247 667, India

Abstract

A PID (Proportional Integral Derivative) controller is most widespread controller and it is mostly used in industries due to its simple tuning procedure. But conventional PID controller can't be used for non-linear system and in real world all system are non-linear. So, this paper represents an artificial neuro fuzzy logic PID controller with Bacteria Foraging oriented by PSO (BF-PSO) and with the help of simulation we will reveal that ANFLC gives better result than conventional controllers.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)

Keywords: ANFLC (artificial neuro fuzzy logic controller); BF (bacterial foraging); Hybrid learning; PID control; PSO (particle swarm optimization).

1. Introduction

PID controller is elementary controller which is used in industries on account of its simple implementation, robust nature and only three tuning parameters[1] but when conventional techniques fails, fuzzy techniques are used. This technique is also simple and main advantage of fuzzy control is less number of variables and less mathematical calculation in the designing of the controller.

For controlling of the any system with the help of fuzzy technique designer needs control rules which are defined by an expert with the help of their prior knowledge about the system and this is the prime deficiency of fuzzy technique. In this method, controller is massively affected by the expert who is providing knowledge about the system. In another fuzzy technique, a reference model is applied to control the system which is replica of the system and it always gives ideal response.

Techniques to tune PID have so much botheration. To avoid these botheration, we are suggesting a simple as well as effective way to generate the fuzzy rules itself according to the system response. This technique is known as Artificial Neuro Fuzzy Logic control and it requires off line training.

To optimize the system, we will adopt Bacteria foraging oriented PSO which is motivated by the nature of Escherichia Coli bacteria. We will use the combination of PSO and Escherichia Coli based optimization to tune controller.

*Corresponding author.
E-mail address: rahul444patel@gmail.com

1877-0509 © 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of organizing committee of the Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)
doi:10.1016/j.procs.2015.06.053

2. The PID Controller

There are so many forms exists to describe the PID but one of the best and versatile way to represent the PID is on view

$$y(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt} \tag{1}$$

where,

$e(t)$ = difference between output and input
K_p = proportional gain
K_i = integral gain
K_d = differential gain

3. Generation of Fuzzy Rule Base for Fuzzy PID

Our purpose is control the system in human like way because human control the system in much easier way than artificial controller due to their imagination. They do not go for exact calculation else they just approximate the required measurement but produces exact result. In generation of rule base, two major problems exists (i) to generate the rule base according to the system knowledge (ii) to generate effective rule base and less number of rules because as number of rules increase complication in the controller increase. But here our aim is to solve first problem.

3.1 Rule generating algorithm

Let us take PD controller to generate rule base

$$y(t) = K_p e(t) + K_d \frac{de(t)}{dt} \tag{2}$$

where, K_p = proportional gain, K_d = derivative gain, $e(t)$ = output − input & $u(t)$ = output of the controller. K_d can also expressed as

$$K_d = \frac{K_p * T_d}{T} \tag{3}$$

T_d = derivative time constant & T = sampling period.
For digital control system equation (2) can be written as

$$y(n) = K_p x_1(n) + K_d x_2(n) \tag{4}$$

where $u(n)$ = controller's output at time n, $x_1(n) = e(n)$ and $x_2(n) = \Delta e(n) = e(n) - e(n-1)$.
After fuzzifying the eqn. (4)

$$U(n) = K_p^* E(n) + K_d^* CE(n) \tag{5}$$

$U(n) \in U$, $E(n) \in X_1(n)$ and $CE(n) \in X_2(n)$ where U, X_1 and X_2 are universe of discourse (uod) for control output, error and change in error respectively. U, E &CE fuzzy sets can be shown as

$$U(n) = \{-M, \ldots\ldots, -1, 0, 1 \ldots\ldots\ldots + M\}$$
$$E(n) = \{-N, \ldots\ldots, -1, 0, 1 \ldots\ldots\ldots + N\}$$
$$CE(n) = \{-N, \ldots\ldots, -1, 0, 1 \ldots\ldots\ldots + N\}$$

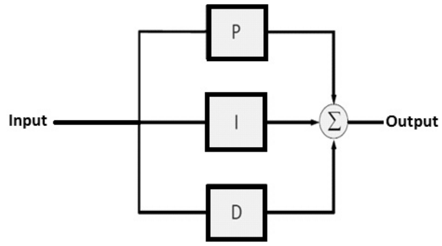

Fig. 1. PID controller.

where N, M are fuzzy linguistic number to limit the uod of the input and control output and this is the main limitation in fuzzy system because if input is not in the uod, it can't generate the result. K_p^* & K_d^* can be represented as

$$K_p^* = k * a,$$
$$K_d^* = K * (1 - \alpha)\ a \in [0, 1]$$

α is weighting factor to regulate the parameters. So eqn. (5) can be written as

$$U = \beta K[\alpha * E + (1 - \alpha) * CE] \tag{6}$$

where β is a constant to match the uod of control with uod of inputs and $\beta = M/N$. Equation (6) gives the relation between control output and inputs and it will help to generate the control rules for the controller. Format to derive the rule base is

$$R_i : \text{ If } e \text{ is } E_i \text{ and } \Delta e \text{ is } C_i \text{ then } u \text{ is } Ui. \tag{7}$$

R_i represents i^{th} rule base for the controller and result will be obtained with the help of eqn. (6). 'If' shows the causes on the controller while 'then' show their consequences of the inputs on the controller. E_i, CE_i and U_i must belong to their respective uod.

General steps to derive fuzzy rules are:

a) Decide fuzzy sets E, CE and U and their uod X_1, X_2 and U respectively.
b) Select the weighting factor α.
c) Obtain fuzzy rule based on eqn. (7).
d) Adjust α and β for better results.

3.2 Parameter tuning

To make controller's output in their uod weighting factor α and β must be predetermined. So for different values of α and β, it will produce different rules. If α is constant then it generates linear control surface and for variable α it generates NL control surface as in Fig. 1. For NL control surface,

$$\alpha = \frac{|E|}{N}(\alpha_2 - \alpha_1) + \alpha_1; \quad 0 \le \alpha_1 \le \alpha_2 \le 1 \tag{8}$$

It is clear from eqn. (8) that α depends upon $|E|$ due to this depends controller becomes self-adjusting for any deviation in error and change in error. So system becomes adaptive for error.

4. Decision Making Based on Rule Base

The task after rule base making is to generate appropriate control output according to the rule base designed. This process has two stages:

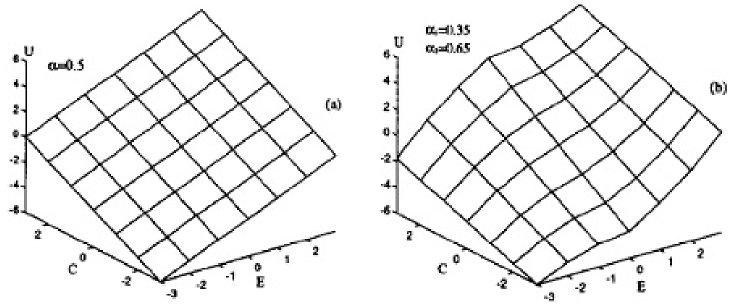

Fig. 2. Control surface for variable values of α.

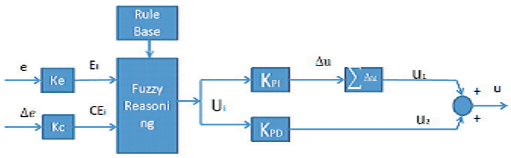

Fig. 3. Hybird Fuzzy (PI+PD) control schematic.

a) decide which rule should be used on the basis of given input and system condition.
b) calculate output from each rule and conclude all the result to generate the control output.

To calculate the control output there are numerous techniques such as COG, COA, LOM, SOM and Bisector methods but we go for COG generally. For COG (Centre of Gravity)

$$U_{\text{Output}} = \frac{\sum_i \mu_c(x_i) \cdot x_i}{\sum \mu_i(x_i)} \tag{9}$$

where μ_c represents membership grade x^i is point where i^{th} rule occurs in uod. To increase the efficiency of the controller, we can change mf, number of mf or defuzzification method but it may lead entire calculation to a cumbersome process. So for desired response, settlement between parameters is needed.

5. Learning Algorithm for Single Neuron

For intelligent i.e. neuro control online adaption of parameters is an important feature. To tune any fuzzy control, we have multiple strategies: a) mf adjustment, b) modification of scaling factor c) change in rule base. Every single strategy is useful for tuning the controller. To make controller online adaptive, MLP can be used for adjusting scaling factors. In hybrid fuzzy controller four scaling factors: K_e, K_{ce}, $K_{PI}\&K_{PD}$ for tuning are present but $K_e\&K_{ce}$ are adjusted by heuristic rules[3] while $K_{PI}\&K_{PD}$ are tuned by MLP because these parameters affects the controller significantly. If we replace MLP by unbiased single neuron with linear activation function, $K_{PI}\&K_{PD}$ will be self-learning. The hybrid neuro-fuzzy schematic is:

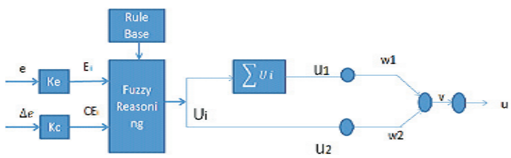

Fig. 4. Hybird neuro fuzzy control schematic.

In the Fig. 4 $K_{PI}\&K_{PD}$ are represented by w_1 $\&w_2$ and the output eqn. is

$$u = u_1 w_1 + u_2 w_2 \tag{10}$$

For learning mechanism, we need reference model to match the transient and steady state characteristics. Let us take a 2$^{\text{nd}}$ order system with a time delay

$$T_R(s) = \frac{\omega_n^2 e^{-\tau} d^s}{s^2 + 2\xi\omega_n^2 s + \omega_n^2} \tag{11}$$

Cost function E_d is minimized to derive the adaptive mechanism,

$$E_d = \frac{1}{2}[y_d^2(t) - y^2(t)]^2 = \frac{1}{2}e_d^2(t) \tag{12}$$

BP algorithm can be adopted for the tuning of scaling weights by BP learning formula[10]

$$\Delta w_i(t) = -\eta \frac{\partial E_d(t)}{\partial t} + \lambda \Delta w_i(t-1), \lambda \in [0,1], \quad \eta \in [0,1] \tag{13}$$

where,

$w_i(t)$ = synaptic weight of i^{th} neuron,
η = learning rate
λ = momentum factor

derivative of E_d w.r.t. w_i is

$$\frac{\partial E_d}{\partial w_i} = \frac{\partial E_d}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial v}\frac{\partial v}{\partial w_i} \tag{14}$$

From the eqn. (12)

$$\frac{\partial E_d}{\partial y} = -(y_d - y) = -e_d \tag{15}$$

$$\frac{\partial u}{\partial v} = f'(v)\&\frac{\partial v}{\partial w_i} = u_i \tag{16}$$

From the eqn. (14), (15) & (16) eqn. (13) can be written as

$$\Delta w_i(t) = \eta[y_d(t) - y(t)]u_i \frac{\partial y}{\partial u} + \lambda \Delta w_i(t-1) \tag{17}$$

We have to estimate $\frac{\partial y}{\partial u}$ each time w.r.t. present change in output to the change in input because generally quantitative analysis is not present.

If the system is monotonic decreasing or increasing

$$\frac{\partial y}{\partial u} < 0 \text{ or } \frac{\partial y}{\partial u} > 0. \tag{18}$$

System will be automatically optimized if the search algorithm is in right direction. So, $\frac{\partial y}{\partial u} > 0$ can be replaced by sgn function in the BP learning algorithm.

$$\Delta w_i(t) = \eta\left(\text{sgn}\frac{\partial y}{\partial u}\right) e_d u_i + \lambda \Delta w_i(t-1) \tag{19}$$

The learning speed of BP algorithm is very slow because of slow convergence. To remove this problem, modified controller is suggested as

$$\Delta w_i(t) = \text{sgn}\left(\frac{\partial y}{\partial u}\right) u_i (K_p e_d(t) + K_d \Delta e_d(t) + \sum_{j=1}^{r} e_d(j) \tag{20}$$

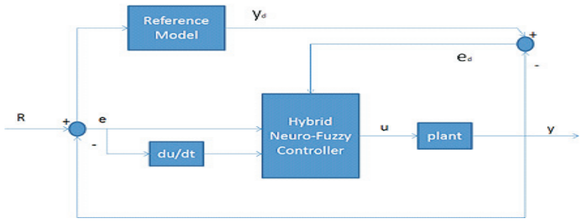

Fig. 5. Adaptive neuro fuzzy controller.

where $\Delta e_d(t) = e_d(t) - e_d(t-1)$ and in eqn (18), η and λ are substituted K_p and integral & derivative terms. Derivative and integral terms are used to remove large gradient rate and offset respectively. A non-linear system is used to analyse the learning efficiency as in Fig. 5.

6. Tuning of Neuro Fuzzy PID by BF-PSO Algorithm

BF-PSO is a combination of two algorithms – BF (Bacterial Foraging) and PSO (Particle Swarm Optimization). The fundamental problem with BF technique is its very low convergence speed. So, to conquer this problem PSO is combined. This combination provides search ability as PSO and ability to acquire new solution. These are steps to obtain optimized solution from BF-PSO:

1) Initialization

The parameters such as $s, Ns, Nc, Ned, Nre, Ped, a_1, a_2, w, v_{\max}$ are inputted.
Where,

s = no. of Bacteria
N_s = no. of bacteria tumbling step
N_c = no. of bacteria chemotaxis
N_{ed} = no. of dispersed bacteria
N_{re} = no. of reproduced bacteria
P_{ed} = dispersal probability
a_1, a_2, w = control parameters
$v_{\max}$ = velocity of bacteria

2) Reproduction

Generate N_{re} for $i = 1$;

3) Chemotaxis

Target to be calculated as N_c for $i = 1$;

4) Tumbling

If $Nc < Ns$, update variables and their current positions.

5) Dispersal

If rand < Ped, relocate the initial position.

6) Update the position, velocity and local minima according to

$$v_{id}(t+1) = wv_{id}(t) + a_1\O_1(p_{id}(t) - y_{id}(t)) + a_2\O_2(p_{gd}(t) - y_{id}(t)) \tag{21}$$

$$y_{id}(t+1) = y_{id}(t) + v_{id}(t) \tag{22}$$

where

P_{gd} = previous best stage of group
P_{id} = previous best stage of particle
$Ø_1, Ø_2$ = any two random numbers

and return to step (3).

7) Return to step (2).
8) Thus we obtained optimized result.

The obtained values after every invocation must limit to upper and lower boundaries.

7. Simulation and Results

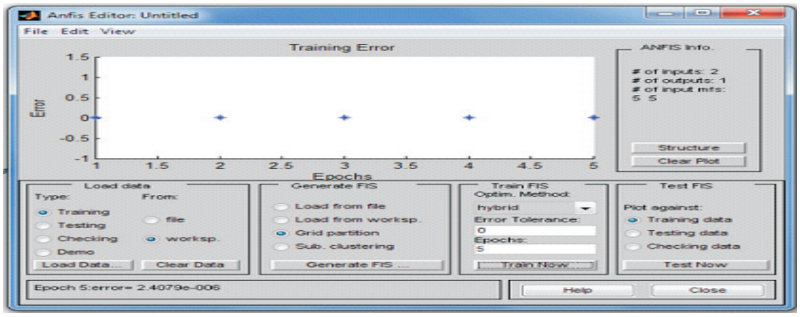

Fig. 6. ANFIS editor layout on MATLAB.

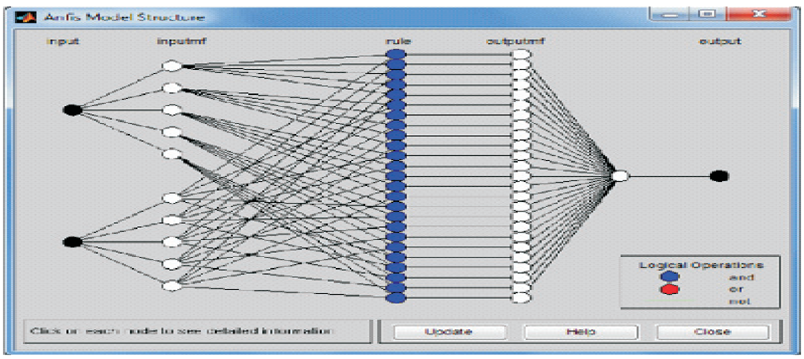

Fig. 7. Structure of ANFIS.

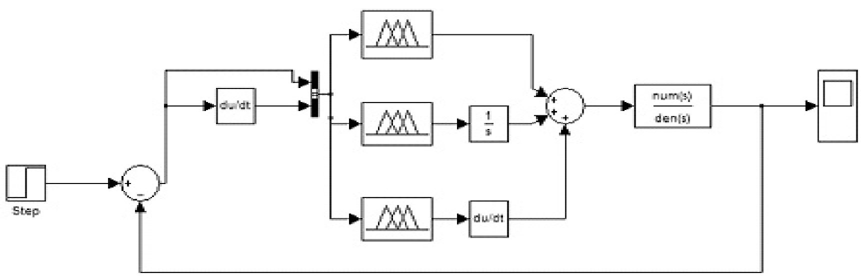

Fig. 8. Simulink model for adaptive neuro fuzzy PID controller.

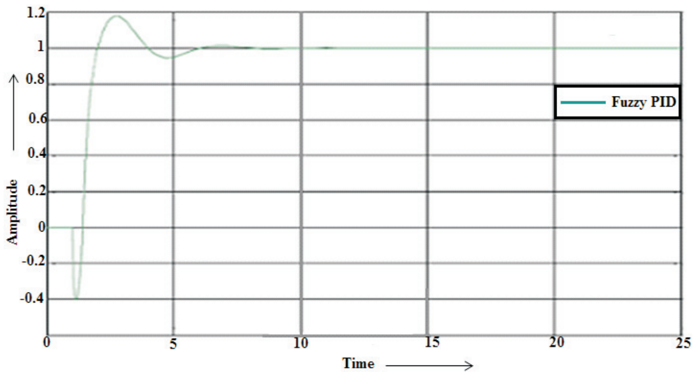

Fig. 9. Response for fuzzy PID.

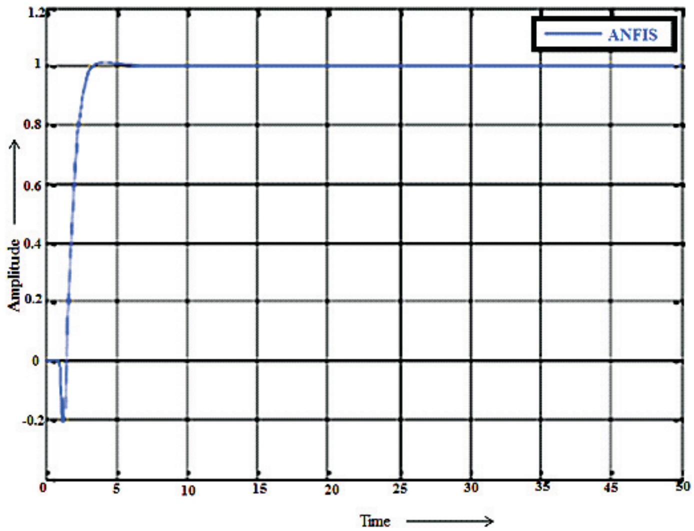

Fig. 10. Response for ANFIS PID controller.

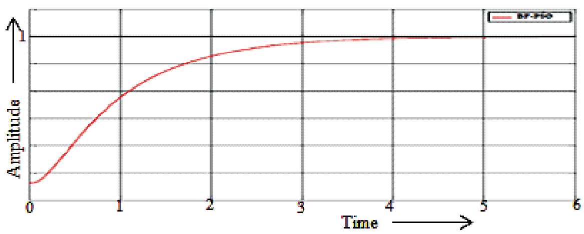

Fig. 11. Response of ANFIS PID with BF-PSO.

The step response for a system with various methods is in figure to analyse the result.

To analyze the result, we can compare some parameters such that rise time (t_r), steady state error (e_{ss}), settling time (t_s) and percentage peak overshoot (M_p) by Table 1.

Table 1. Analysis of various parameters for various tuning methods of PID.

Parameters	Name of Tuning Methods			
	ZN	Fuzzy	ANFIS	BF-PSO
Rise Time (t_r)	1.789	1.865	2.578	4.125
Settling Time (t_s)	3.745	5.624	7.125	4.220
Overshoot ($\%M_p$)	20.05	17.95	1.0149	0
Steady State Error (e_{ss})	0	0	0	0

We can wind up the table by:

- Every performance characteristics of the system improves excluding rise time but we don't bother about it because our basic requirements are less settling time and less peak overshoot and that are attained.
- It is obvious ANFIS produces best result because of combine features of fuzzy and neuro logics and main advantage is e_{ss} is zero for all tuning methods.

8. Conclusion

In the whole analysis, we simply analyzed the various fuzzy, neuro and ZN analogies of PID tuning. We used ANFIS tuning method as neuro technique and BF-PSO for the optimization and these results are acquired for 2nd order system with no poles and zeros in right half i.e. stable system. Retrieved MATLAB results conclude that best output is processed for ANFIS controller for the characteristics overshoot (M_v) and settling time (t_s). Optimization algorithm BF-PSO provides modest result if we don't bother rise time. In general plants we don't need best results in terms of all the characteristics but we go for modest result according to the necessity of the plant output.

References

[1] Meng Joo Er and Ya Lei, Hybrid Fuzzy Proportional–Integral Plus Conventional Derivative Control of Linear and Nonlinear Systems, *IEEE Transactions on Industrial Electronics*, vol. 48, no. 6, pp. 1109–1117, December (2001).

[2] Youxin Luo, The Controlling Parameters Tuning and its Application of Fractional Order PID Bacterial Foraging Based Oriented by Particle Swarm Optimization, *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, vol. 1, pp. 4–7, (2009).

[3] D. Seborg, T. F. Edgar and D. A. Mellichamp, Process Dynamics & Control, Wiley India Edition, (2005).

[4] K. Sinthipsomboon, W. Pongaen and P. Pratumsuwan, A Hybrid of Fuzzy and Fuzzy Self-Tuning PID Controller for Servo Electro-Hydraulic System, *6th IEEE Conference on Industrial Electronics and Applications*, pp. 220–225, (2011).

[5] K. S. Tang, K. F. Man, G. Chen and S. Kwong, An Optimal Fuzzy PID Controller, *IEEE Transactions On Industrial Electronics*, vol. 48, no. 4, pp. 757–765, August (2001).

[6] J. C. Basilio and S. R. Matos, Design of PI and PID Controllers with Transient Performance Specification, *IEEE Transactions on Education*, vol. 45, no. 4, pp. 364–370, November (2002).

[7] R. M. Pidaparti, Estimation of Safety Index of Aging Structure through Neural Networks and Particle Swarm Optimization, *Journal of Intelligent Material Systems and Structures*, vol. 17 no. 11, pp. 997–1005, November (2006).

[8] C. T. Lin and C. S. G. Lee, Neural Network based Fuzzy Logic Control and Design System, *IEEE Trans. Comput.*, vol. 40, pp. 1320–1326, (1991).

[9] S. Tzefestas and N. P. Papanikolopoulos, Incremental Fuzzy Expert PID Control, *IEEE Trans. Ind. Electron.*, vol. 37, pp. 365–371, (1990).

[10] S. Baker and Y. Le Cun, Improving the Convergence of Back Propagation Learning with Second Order Methods, *Proc. Connectionist Model Summer School (Morgan Kaufmann, San Mateo)*, pp. 29–37, (1988).

[11] C. T. Lin, A Neural Fuzzy Control System with Structure and Parameter Learning, *Fuzzy Sets and Systems*, pp. 183–212, (1995).