15th Triennial World Congress, Barcelona, Spain

NEURO-FUZZY BASED CONTROL LOOP TUNER

Yul Y. Nazaruddin and Edna Caroline

Department of Engineering Physics, Institut Teknologi Bandung
Jl. Ganesa no. 10, Bandung 40132, INDONESIA
Phone/fax. : +(62) (22) 2508138 ; e-mail : yul@tf.itb.ac.id

Abstract: An alternative approach for intelligent tuning of a control loop will be presented in this paper. The objective is to design an algorithm which will tune the controller employing a neuro-fuzzy based algorithm. Structure and design method based on this approach will be explored, which include an adaptive network employed as building block, the back-propagation gradient method and least square estimator as a hybrid learning rule, and its integration with tuning algorithm. The effectiveness and the performance of the proposed intelligent tuning method will be demonstrated by on-line controller tuning of a process mini-plant. Experimental results show some potential benefits on applying the proposed technique using the real-world process plant. *Copyright ©2002 IFAC*

Keywords : neuro-fuzzy, tuning, real-time control, hybrid learning, PI controller

1. INTRODUCTION

The tuning of controllers is a task that requires a considerable amount of expertise in order to achieve an adequate performance in term of process control. Until now, simple controllers of the Proportional Integral (Derivative) / PI(D) type are still going through an interesting development. This type of controllers is the major practical control technology which is widely used in the process industries at present. Although PI(D) control has been succesfully applied to process control and many other control problems, it still has some major limitation. Its performance depends heavily on the plant operating parameters. Once these parameters change, a significant amount of effort is required to retune the controller manually. Various approaches have been developed to design PI(D) controllers which have the ability to adapt to a changing operating environment automatically (Koivo and Tanntu 1991, Aström and Hagglund 1995). In recent years, there have been several attempts to incorporate the intelligent methodologies, including neural network, expert and fuzzy system, into the control loops (see e.g. Aström 1991, Gupta and Sinha 1996). Various industrial application of intelligent methodologies, either for modeling and control, have been developed succesfully in several areas of application.

Recent research efforts have also been shown to combine the neural network methodology and the fuzzy systems methodology. The primary motivation is the integration of the strength of both methodologies in order to achieve learning and adaptation capabilities and knowledge representation via fuzzy if-then rules, producing the so-called neuro-fuzzy systems. The main advantage of the neuro-fuzzy controller is its learning capability from the numerical data obtained from the measurements and hence no mathematical model of the plant to be controlled is needed, which is very advantageous for the plants where its mathematical models are difficult to derive.

Owing to the effectiveness of the neuro-fuzzy approach in its learning capabilities (Nazaruddin and Yamakita 1999, Alturki and Abdennour 1999) has given a motivation that an alternative method to automatically tune the controller using neuro-fuzzy approach could be explored. An architecture of the so-called adaptive neuro-fuzzy inference system has been further investigated. The effectiveness and the performance of the proposed intelligent tuning method will be demonstrated by on-line controller tuning of a process mini-plant. Performance comparison was also made if the plant was tuned using the conventional Ziegler-Nichols method.

2. ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM SCHEME

By using the neuro-fuzzy scheme, the fuzzy inference system can be tuned with a neural network algorithm based on some collection of input-output data, which then allow the fuzzy system to learn. An architecture of neuro-fuzzy described as Adaptive Neuro-Fuzzy Inference Systems/ANFIS (Jang, *et.al.*, 1997) is illustrated in Fig. 1. In this architecture, fuzzy if-then rules, obtained from human experts to describe the input-output behaviour of complex systems can be refined and if human expertise is not available, reasonable membership functions are set-up intuitively and the learning process is initiated to generate a set of fuzzy if-then rules to approximate a desired data set.

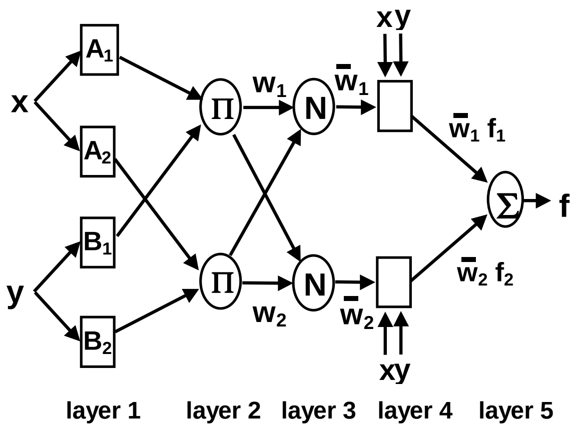


Fig. 1. Adaptive neuro-fuzzy inference systems scheme

The basic *learning rule* concerns with how to recursively obtain a gradient vector in which each element is defined as the derivative of an error measures with respect to parameters. A solution called hybrid learning rule which combines the gradient method and the least square estimator is applied to avoid a slow convergence and possibility to be trapped to local minima as experienced by the conventional back-propagation learning rule. A fuzzy inference system of Tagaki-Sugeno of the first order which has two inputs and one output can be described by the following rule base

Rule i-th : If x is A_i and y is B_i Then $f_i = p_i\, x + q_i\, y + r_i$ $i = 1,2,...,m$

where m denotes the number of rules. Fig. 1 describes a neuro-fuzzy structure which is equivalent to 2 rules. The architecture consists of 5 layers with different functions in every layer. The description and its function in every layer can be summarized as shown in Table 1.

In the above structure, the adaptive network is manifestated only by layer 1 and 4. The parameters in layer 1 and 4 are referred to as *premise parameters* and as *consequent parameters* respectively. In the 1st. layer, the adaptive parameter is the parameter of the membership function of input fuzzy set, which is nonlinear function of the system output. The parameters in the 4th. layer are the linear function of the system output, assuming that the parameter of the membership function is fixed. In general, the structure has nonlinear adaptive parameter in the 1st. layer and linear adaptive parameter in the 4th. layer. Due to the linear relationship with regard to the output parameters, then a least-square estimator (LSE) can be applied for the learning process. Suppose that S_1 is a set of nonlinear parameters and S_2 is a set of linear parameters in the architecture. The learning process applies the gradient descend and the least-square algorithm (Jang, *et.al.*, 1997) to update the parameters in S_1 and S_2 respectively.

Table 1. The description and its function in every layer

Layer 1 :	adaptive node with node function : $O_{1,i} = \mu_{A_i}(x)$ and $O_{1,i} = \mu_{B_i}(y)$
Layer 2 :	fixed node with firing strength of a rule : $O_{2,i} = w_i = \mu_{A_i}(x)\,\mu_{B_i}(y)$, i =1,2
Layer 3 :	fixed node with normalized firing strength : $O_{3,i} = \overline{w}_i = \dfrac{w_i}{w_1 + w_2}$, i =1,2
Layer 4 :	adaptive node with node function : $O_{4,i} = \overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i)$
Layer 5 :	fixed node which computes the summation of signals : $O_{5,i} = \overline{w}_1 f_1 + \overline{w}_2 f_2$

The Learning Process for the Linear Parameters :

If the parameters in S_1 in figure 1 is fixed, then the output of the system can be written as

$$f = \overline{w}_1 f_1 + \overline{w}_2 f_2 = (\overline{w}_1 x)p_1 + (\overline{w}_1 y)q_1 + (\overline{w}_1)r_1 + (\overline{w}_2 x)p_2 + (\overline{w}_2 y)q_2 + (\overline{w}_2)r_2 \quad (1)$$

From the above equation, it can be seen that the consequent parameters are linear parameters with respect to the systems output. If P learning data is applied to the Eq. (1), it can be shown that it can be represented by $A\theta = y$, where θ is an unknown vector whose elements are parameters in S_2 and y is the output vector whose elements are P learning data. Using the least-square estimator, the best solution of this equation will be given by

$$\theta_{i+1} = \theta_i + P_{i+1} a_{i+1} (y_{i+1}^T - a_{i+1}^T \theta_i) \quad (2.a)$$

$$P_{i+1} = P_i - \frac{P_i a_{i+1} a_{i+1}^T P_i}{1 + a_{i+1}^T P_i a_{i+1}} \quad (2.b)$$

with a_i^T is row vector of matrix A , y_i is the i-th element of y, and P_i is the covariance matrix.

The Learning Process for the Nonlinear Parameters:

The learning process uses the simple steepest descend method, where each parameters are updated using the relation $\alpha_{next} = \alpha_{now} - \eta(\partial^+ E / \partial\alpha)$, causing

$E(\alpha_{next}) < E(\alpha_{now})$, where α is the node parameter and η is the learning constant and $\partial^{+}E / \partial\alpha$ denotes the ordered derivative of the error signals $E(\theta) = (y_d - y)^2$ (i.e. the difference between an actual trajectory and a given desired trajectory) with respect to the node parameter α.

3. NEURO-FUZZY BASED TUNING OF PI CONTROLLER

In general, the transfer function of PI controller is given as $G_s(s) = K_P + K_I / s$ with K_p and K_I are proportional and integral gain respectively. In another form, PI controller is formulated as $G_s(s) = K_p(1 + 1/(T_{Ii}/s))$ with $T_I = K_P/K_I$ is known as integral time constant. Further, the discrete version of the controller can be written as $u(k) = K_P e(k) + K_I T_I \sum e(k)$, with *u(k)* represents control signal and *e(k)* represents error between process output and setpoint.

Basically, the automatic tuning procedure is performed on the two controller parameters, i.e. proportional gain K_p and integral time constant T_i until transient system response meets the desired performance specification. Similar to the tuning strategies in Yen *et.al*. (1995), the rules used can be stated in the form

If *performance-measure* is X_1 then ΔK_p is Y_1
If *performance-measure* is X_2 then ΔK_I is Y_2

where X_1 and X_2 are fuzzy sets describing the performance measure, and Y_1 and Y_2 are fuzzy sets describing the amount of correction to the scaling factor of the controller parameters. The performance measure used for criteria of tuning is percent maximum overshoot (OV), defined as
$\max . OV = ((c(t_p) - c(\infty)) / c(\infty)).100\%$.

Based on above rules, for the adaptive neuro-fuzzy inference system mechanism, learning data with 2n input and n output are collected, in the format of $\lfloor OV_a \quad OV_d \quad \Delta K_p \rfloor$ to tune the component of K_p and format $\lfloor OV_a \quad OV_d \quad \Delta K_I \rfloor$ to tune the parameter K_I, where OV_a is the actual OV and OV_d is the desired OV. These data are collected by varying P and I parameters of the controller and registering the overshoot, for different operating conditions of the plant. Tuning procedure is then performed iteratively until the desired transient response is achieved. In each step, controller parameters are updated as follows

$$\begin{aligned} K_p(i+1) &= K_p(i) + \Delta K_p \\ T_I(i+1) &= T_I(i) + \Delta T_I \end{aligned} \qquad (3)$$

4. PROCESS MINI-PLANT DESCRIPTION

The proposed neuro-fuzzy based PI controller tuner was tested on a real-time fluid level control of a laboratory scaled process mini-plant. The tuning and controller algorithm was implemented as software which developed using C++ language and run on a personal computer connected *on-line* to the process mini-plant. The plant consists of a tank reactor containing fluid which is to be controlled and several industrial devices to measure fluid flow and level. An interface card is attached to the personal computer to enable direct data acquisition from the level transmitter and to send control signal to the control valve. View of the process mini-plant and its process flow diagram are shown in Fig 2. and 3, respectively.

Fig 2. View of process mini-plant

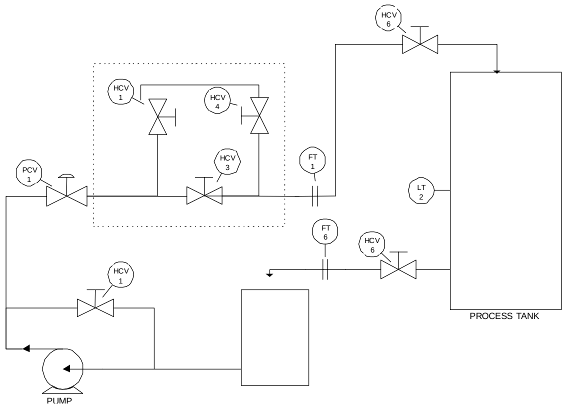


Fig. 3. Process flow diagram of the mini-plant to be controlled

To observe the plant characteristics and how it behaves for different scattering data of K_p and K_I, which will be used for learning data, initial step response test was conducted on the process mini-plant. The test was performed using set-point of 50% level and a sampling period of 1 sec. The plant response is illustrated in Fig 4. and the performance measure for plant characteristics, given in its percent overshoot (OV) and rise time (RT), is given in Table 2. Based on this result, learning data for parameter K_p was taken for the range of K_p between 0.3 and 5.

Table 2. Plant performance for various values of K_p

K_p	OV(%)	RT (sec.)
0.3	55.88	21
1	14.62	20
5	Oscillate	32

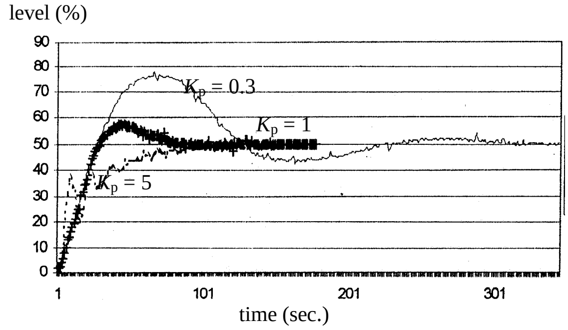


Fig. 4. Plant response to a step input of 50%

5. EXPERIMENTAL RESULTS AND EVALUATION

Tuning of PI controller using the proposed adaptive neuro-fuzzy approach was performed based of 161 pairs of input-output data, which were taken from 5 different conditions of mini-plant characteristics. Based-on observation of plant performance test, the initial condition for K_p was set to 0.3 and T_I to 10 sec., resulting an overshoot (OV) of 77.62% and rise time (RT) of 13 sec. The tuning procedure was then performed iteratively using the adaptive neuro-fuzzy scheme, explained above, so that a desired overshoot of 5% could be achieved. As performance measure, the root means square of the error (RMSE), defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^{k} e(nT)^2}{k}} \qquad (4)$$

and the integral of time multiplied absolute error (ITAE), defined as

$$\text{ITAE} = k\sum_{n=1}^{k} |e(nT)| \qquad (5)$$

where $e(.)$ denotes the error at time k and T is the sampling instant, were used in all experimental studies. Table 3 shows the results of the tuning after few iterations. Plant response after tuning procedure with K_p = 3.1 and T_I =10 sec. and with K_p = 3.1 and T_I =17.26 sec. are shown in Fig. 5.

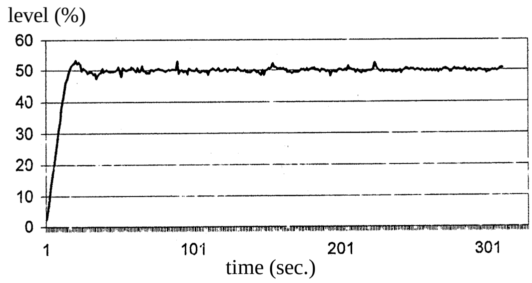
Table 3. Result of tuning PI controller using the adaptive neuro-fuzzy scheme

Itera-tion	K_p	T_I	**OV (%)**	**RT (sec.)**
1	1.28	10	32.5	10
2	2.344	10	12.88	12
3	3.1	10	6.5	12
4	3.1	17.26	3.12	27

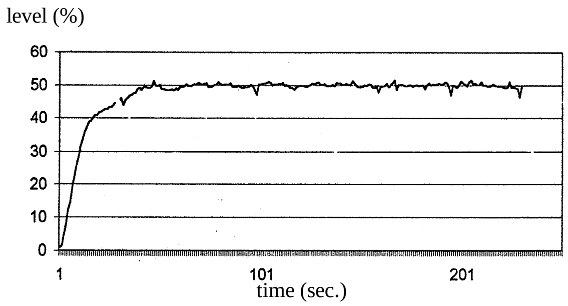
Iteration	**1**	**2**	**3**	**4**
RMSE	9.05	7.851	8.367	9.01
ITAE	27272	14130	12143	15818

As comparison, a tuning procedure based on conventional Ziegler-Nichols closed-loop method was also conducted in the experiment. The procedure consists of estimating the *ultimate gain* K_u, i.e. value of gain which result in critically stable system while using only gain controller, and measuring the *ultimate period* P_u, i.e. oscillation period while closed-loop system is critically-stable.

Based on these two information and using an empirical formula, PI controller were determined, which yields parameters setting parameters for PI controller, i.e K_p = 0.567 and T_I =51.7 sec. The plant response using this controller parameters is illustrated in Fig. 6. As observed from the transient response, an overshoot of 9.6% and rise time of 31 sec. are produced by using this type of controller.



(a). K_p = 3.1 and T_I =10 sec.



(b). K_p = 3.1 and T_I =17.26 sec.

Fig. 5. Plant response using neuro-fuzzy based controller tuner

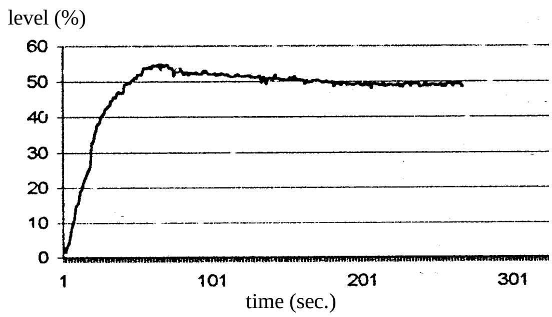


Fig. 6. Plant response using Ziegler-Nichols tuning procedure

As can be observed, better performances are shown using the controller tuned with the designed adaptive neuro-fuzzy based tuner compared to the conventional Ziegler-Nichols method. As not in the case of the latter method, the neuro-fuzzy based controller tuner can also be tuned based on the desired performance criteria, which in this case is the percent overshoot, which was set to 5% in the experiments. The best performance was given if K_p = 3.1 and T_I =10 sec. was used as also confirmed by the value of RMSE and ITAE.

6. CONCLUSIONS

An alternative technique for control loop tuner using adaptive neuro-fuzzy inference mechanism has been presented and succesfully implemented in tuning a controller on real-time control of a process mini-plant. Better performance was also shown from the experimental results compared to the conventional tuning using Ziegler-Nichols method. The controller parameters can be tuned based on desired transient response specifications, namely overshoot percentage and *rise-time.* Further, compared to Ziegler-Nichols method, tuning of neuro-fuzzy based PI controller works iteratively and will not bring the system into critical-stable condition.

REFERENCE

Alturki, F.A. and A.B. Abdennour (1999), Neuro-Fuzzy Control of a Steam Boiler-Turbine Unit, Proceed. of *IEEE International Conference on Control Applications (CCA'99),* Kohala Coast, Island of Hawaii, Hawaii, USA

Aström, K.J. (1991), Directions on Intelligent Control, *Proceed. of Intelligent Tuning and Adaptive Control (ITAC'91),* Singapore

Aström, K.J. and T. Hagglund (1995), *PID Controllers: Theory, Design and Tuning,* 2^{nd} Edition, Instrument Society of America, North Carolina

Gupta, M.M. and N.K. Sinha (ed.) (1996), *Intelligent Control Systems*, IEEE Press, New Jersey

Jang, J.S.R., C.T. Sun and E. Mizutani (1997), *Neuro-fuzzy and Soft Computing*, Prentice-Hall Inc., Englewood Cliffs, New Jersey

Koivo, H.N. and J.T. Tanttu (1991), Tuning of PID Controllers : Survey of SISO and MIMO Techniques, *Proceed. Intelligent Tuning and Adaptive Control (ITAC'91*), Singapore

Nazaruddin,Y.Y. and M. Yamakita (1999), Neuro-Fuzzy Based Modeling of Vehicle Suspension System, Proceed. of *IEEE International Conference on Control Applications (CCA'99),* Kohala Coast, Island of Hawaii, Hawaii, USA

Yen, J., R. Langari and L.A. Zadeh (1994), *Industrial Applications of Fuzzy Logic and Intelligent Systems*, IEEE Press, New York