ELSEVIER

Fuzzy Sets and Systems 128 (2002) 95–106

FUZZY sets and systems

www.elsevier.com/locate/fss

Tuning of PID controllers for unstable processes based on gain and phase margin specifications: a fuzzy neural approach

Ching-Hung Lee[a], Ching-Cheng Teng[b,*]

[a] *Department of Electrical Engineering, Yuan Ze University, Chungli, Taoyuan 320, Taiwan, ROC*
[b] *Department of Electrical & Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, ROC*

Received 5 February 1999; received in revised form 1 October 2000; accepted 2 February 2001

Abstract

This paper presents a PID tuning method for unstable processes using an adaptive-network-based-fuzzy-inference system (ANFIS) for given gain and phase margin (GPM) specifications. PID tuning methods are widely used to control stable processes. However, PID controller for unstable processes is less common. In this paper, the PID controller parameters can be determined by the ANFIS. Because the definitions of gain and phase margin equations are complex, an analytical tuning method for achieving specified the gain and phase margins is not yet available. In this paper, the ANFIS is adopted to identify the relationship between the gain-phase margin specifications and the PID controller parameters. Then, it is used to automatically tune the PID controller parameters for different gain and phase margin specifications so that neither numerical methods nor graphical methods need be used. A simple method is also developed to estimate the stabilizing region of PID controller parameters and valid region for gain-phase margin. Even for unreasonable specifications, out of the valid region, the ANFIS can still find suitable PID controller to guarantee the stability of the closed-loop system. Simulation results show that the ANFIS can achieve the specified values efficiently.

Keywords: ANFIS; Unstable process control; PID controller tuning; Gain and phase margin

1. Introduction

Several methods for determining PID controller parameters have been developed over the past 50 years. Some employ information about open-loop step response, for example, the Coon–Cohen reaction curve method [7]; other methods use knowledge of the Nyquist curve, for example, the Ziegler–Nichols frequency-response method. However, these tuning methods use only a small amount of information about the dynamic behavior of the system, and often do not provide good tuning. It is known that gain margin and phase margin have served as important measures of robustness. From classical control theories, phase margin is related to the

* Corresponding author. Tel.: +886-3-5712121/ext: 54345; fax: +886-3-5715998.
E-mail addresses: chlee@saturn.yzu.edu.tw (C.-H. Lee), ccteng@cn.nctu.edu.tw (C.-C. Teng).

PII: S0165-0114(01)00051-3

damping of the system, and therefore also serves as a performance measurement. Their solutions are normally obtained numerically or graphically by trial-and-error use of Bode plots. Controllers designed to satisfy gain margin and phase margin (GP/GM) criteria are not new approaches [2,8−11,19]. In 1984, Astrom and Hagglund first proposed a tuning method for PID controllers based on phase and amplitude specifications [1]. Then, Ho et al. presented a tuning method for stable and unstable processes [9−11]. They adopt linear equations to approximate the arctan function as to simplify the gain-phase margin formulas. Due to the approximation of arctan function, this method may result in unstable controllers or unstable systems for some specifications. In this paper, a fuzzy neural network approach is presented to solve this problem. The approach determines the PID controller parameters that guarantee the stability of controller and the closed-loop system.

With the development of fuzzy logic controllers and the more recent hybrid controllers which use both fuzzy logic and neural network methodology, the possibility exists that one or both of these methods could perform as a feedback controller [5,6,15,17,18]. The fuzzy logic toolbox [16] implements one of the hybrid schemes known as the adaptive-network-based-fuzzy-inference system (ANFIS). The ANFIS has proven to be an excellent function approximation tool and can be as good or better than a plain feedforward neural network for some situations. Although various kinds of fuzzy logic controllers (FLCs) [20,21] are widely used nowadays and have certain advantages over conventional PID controllers, relatively few theoretical analysis that explain why they can achieve better performance are available. In literature [6], we have presented a tuning method that uses the ANFIS based on gain and phase margin specifications, to tune the PI controller parameters processes efficiently. This approach enjoys the advantage of functionally mapping the ANFIS, and gives better performance than GPM [9]. The purpose of this paper is to extend this approach to unstable processes and solve the unsuitable results of [11]. The stabilizing region of controller parameters and the valid region of specifications (A_m, ϕ_m) for PID controller are also estimated.

The arrangement of this paper is as follows. In Section 2, we briefly introduce gain and phase margins, and the used fuzzy neural network (ANFIS). Section 3 proposes the structure of PID controller using the ANFIS and the tuning method. Section 4 describes a procedure for estimating the stabilizing region of controller parameters and valid region of GPM specifications for PID controller. Section 5 gives the simulation results and discusses the advantages of the proposed approach as compared with other methods. Finally, conclusions are summarized in Section 6.

2. Preliminaries

2.1. Gain margin and phase margin

Consider the n-order unstable process with time-delay

$$G_{\mathrm{p}}(s) = \frac{K(1+w_{n1}s)^{n1}(1+w_{n2}s)^{n2}\cdots(1+w_{nq}s)^{nq}}{(1+w_{d1}s)^{d1}(1+w_{d2}s)^{d2}\cdots(1+w_{dp}s)^{dp}}\mathrm{e}^{-Ls}, \tag{1}$$

where at least one of w_{di} is negative and $n=\sum_{i=1}^{p} di$. The open-loop step response of the process is unbounded, since it has a pole in the right-half plane. Figs. 1(a) and 1(b) show the Bode and Nyquist diagrams of an unstable first-order plus delay process with PI control. Note that unstable plant have more than one GM/PM. As the definitions of the GM/PM [14], A_{m1} and A_{m2} are called *gain margin* (or upward gain margin) and *gain reduction margin* (or downward margin). In addition, applying the Nyquist criterion for stability, the Nyquist diagram should encircle the point $(-1, 0)$ [in the $G(jw)$ plane] exactly once in the anti-clockwise direction. Based on the stability criterion, the gain margin A_{m1} is chosen in this literature. Here, the PID tuning for unstable plant is detailed in [11]. We followed the same line of [11].

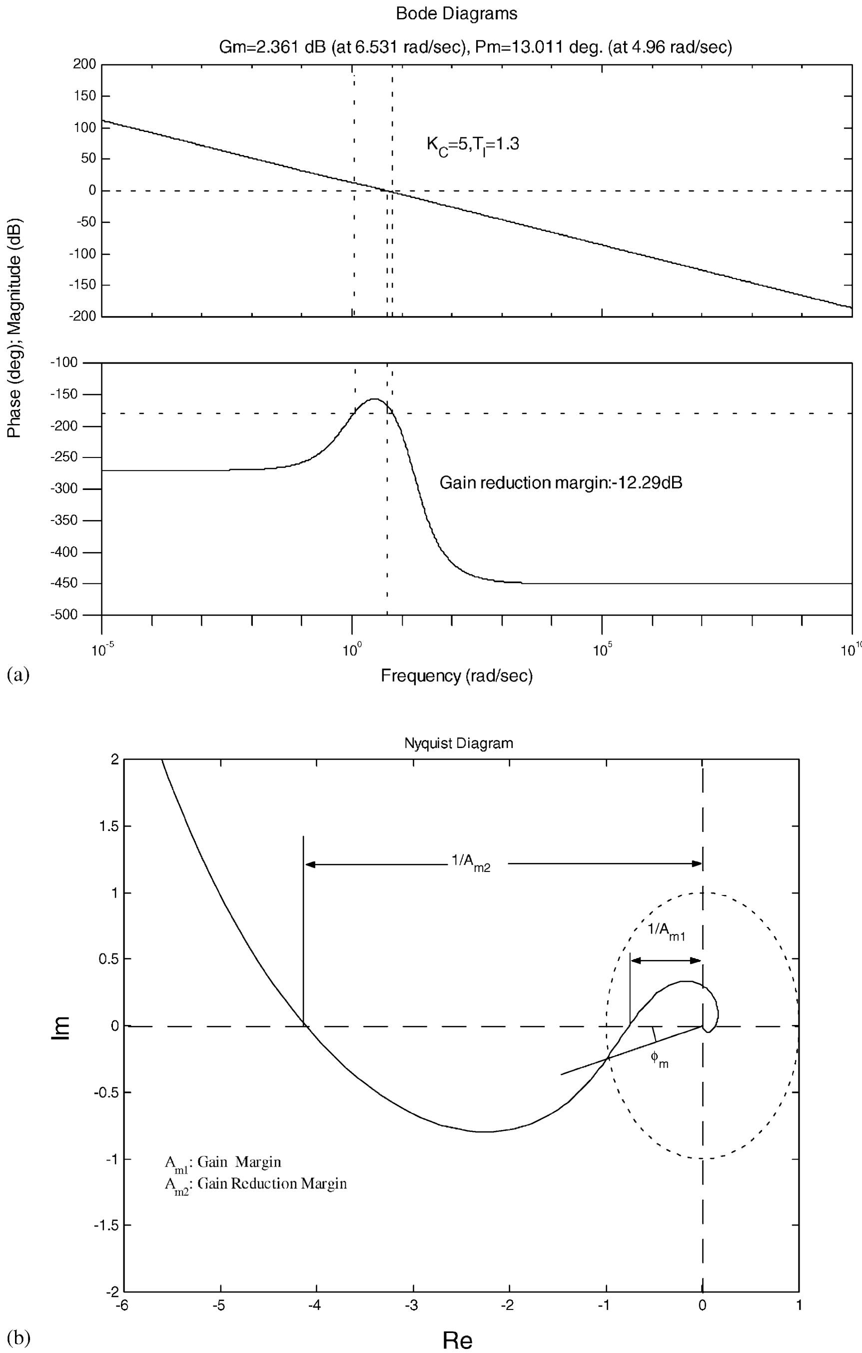

Fig. 1. Bode and Nyquist diagrams of unstable first-order plus delay process with PI control.

The PID controller given by

$$G_c(s) = K_P + \frac{K_I}{s} + K_D s, \tag{2}$$

must be used to satisfy the Nyquist criterion. Let the specified gain and phase margins be denoted by A_m and ϕ_m, respectively. The formulas for gain and phase margins are as follows:

$$\arg[G_C(jw_p)G_P(jw_P)] = -\pi \tag{3}$$

$$A_m = \frac{1}{|G_C(jw_P)G_P(jw_P)|}, \tag{4}$$

$$|G_C(jw_g)G_P(jw_g)| = 1, \tag{5}$$

$$\phi_m = \arg[G_C(jw_g)G_P(jw_g)] + \pi, \tag{6}$$

where the gain margin is defined by Eqs. (3) and (4), and the phase margin by Eqs. (5) and (6). Here w_p and w_g denote the phase crossover frequency and gain crossover frequency, respectively. The loop transfer function is obtained from

$$G_c(s)G_p(s) = \frac{K(K_I + K_P s + K_D s^2)(1 + w_{n1}s)^{n1}(1 + w_{n2}s)^{n2}\cdots(1 + w_{nq}s)^{nq}}{s(1 + w_{d1}s)^{d1}(1 + w_{d2}s)^{d2}\cdots(1 + w_{dp}s)^{dp}} e^{-Ls}.$$

Substituting the above equation into Eqs. (3)–(6), we have

$$\tfrac{1}{2}\pi + \tan^{-1}(w_p w_{c1}) + \tan^{-1}(w_p w_{c2}) + n_1 \tan^{-1}(w_p w_{n1}) + \cdots + n_q \tan^{-1}(w_p w_{nq}) - w_p L$$
$$-d_1 \tan^{-1}(w_p w_{d1}) - d_2 \tan^{-1}(w_p w_{d2}) - \cdots - d_p \tan^{-1}(w_p w_{dp}) = 0, \tag{7}$$

$$A_m K = w_p \frac{\sqrt{(1 + w_p^2 w_{n1}^2)^{n1}}\sqrt{(1 + w_p^2 w_{n2}^2)^{n2}}\cdots\sqrt{(1 + w_p^2 w_{nq}^2)^{nq}}}{\sqrt{1 + w_p^2 w_{c1}^2}\sqrt{1 + w_p^2 w_{c2}^2}\sqrt{(1 + w_p^2 w_{d1}^2)^{d1}}\cdots\sqrt{(1 + w_p^2 w_{dp}^2)^{dp}}}, \tag{8}$$

$$K = w_g \frac{\sqrt{(1 + w_g^2 w_{d1}^2)^{d1}}\sqrt{(1 + w_g^2 w_{d2}^2)^{d2}}\cdots\sqrt{(1 + w_g^2 w_{dp}^2)^{dp}}}{\sqrt{(1 + w_g^2 w_{c1}^2)}\sqrt{1 + w_g^2 w_{c2}^2}\sqrt{(1 + w_g^2 w_{n1}^2)^{n1}}\cdots\sqrt{(1 + w_g^2 w_{nq}^2)^{nq}}}, \tag{9}$$

$$\phi_m = \tfrac{1}{2}\pi + \tan^{-1}(w_g w_{c1}) + \tan^{-1}(w_g w_{c2}) + n_1 \tan^{-1}(w_g w_{n1}) + \cdots + n_q \tan^{-1}(w_g w_{nq}) - w_g L$$
$$- d_1 \tan^{-1}(w_g w_{d1}) - d_2 \tan^{-1}(w_g w_{d2}) - \cdots - d_p \tan^{-1}(w_g w_{dp}), \tag{10}$$

where w_{c1} and w_{c2} are the roots of $(K_I + K_P s + K_D s^2)$. For a given process $(K, w_{n1}, \ldots, w_{nq}, w_{d1}, \ldots, w_{dp}, L)$ and specifications (A_m, ϕ_m), Eqs. (7)–(10) can be solved for the PID controller parameters (K_P, K_I, K_D) and crossover frequencies (w_g, w_p) numerically but not analytically because of the presence of the $\tan^{-1}$ function. For stable processes, controllers such as the IMC [4] and GPM [9,10] that are based on gain and phase margins cannot efficiently meet specifications within a 10% error margin owing to the approximation of the $\tan^{-1}$ function. In addition, a similar controller based on GPM for an unstable process improves performance but still can only meet the specifications within 5% error [11]. Using this approximated method [11], unstable results (unstable controller or unstable closed-loop system) occurred due to the approximation of $\tan^{-1}$ (for details, see Remark 3). Therefore, another approach using the ANFIS for general processes is considered here. This approach yields high accurate tuning formulas for controllers including P, PI, PD and PID controllers of stable and unstable processes with time delay.

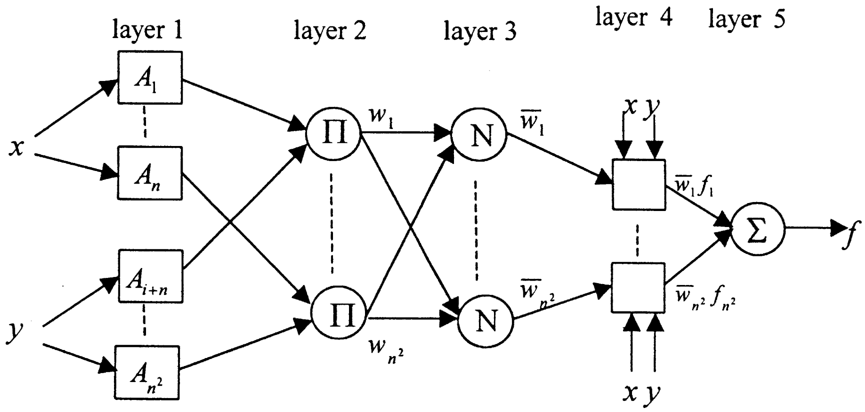

Fig. 2. The architecture of the ANFIS.

Remark 1. It is well known that the model

$$G(s) = \frac{K}{1+sT}\,\mathrm{e}^{-sL} \tag{11}$$

is the most common process model used in paper on PID controller tuning [2]. As the statement of [2,13] the following processes were chosen that are representative for the dynamics of typical industrial processes:

$$
\begin{aligned}
G_1(s) &= \frac{\mathrm{e}^{-s}}{(1+sT)^2}, \quad T = 0.1,\ldots,10,\\
G_2(s) &= \frac{1}{(1+s)^n}, \quad n = 3,4,8,\\
G_3(s) &= \frac{1}{(1+s)(1+\alpha s)(1+\alpha^2 s)(1+\alpha^3 s)}, \quad \alpha = 0.2, 0.5, 0.7,\\
G_4(s) &= \frac{1-\alpha s}{(1+s)^3}, \quad \alpha = 0.1, 0.2, 0.5, 1.2.
\end{aligned} \tag{12}
$$

The test batch (12) does not include the transfer function (11) because this model is not representative for typical industrial processes [2]. Therefore, we present here our approach in transfer function (1) that includes the test batch (12) and model (11) as model (11) is the most common process model used in the paper on PID controller tuning.

2.2. *Fuzzy neural network* (*ANFIS*)

The used ANFIS [15–17] architecture is shown in Fig. 2. The inputs are given by (x, y) and have R^i $(i = 1,\ldots,n^2)$ implications, then the value of f is implied as follows.

Layer 1: Here we denote the output node i in this layer as $O_{l,i}$. Every node is an adaptive node with a node output defined by

$$
\begin{aligned}
O_{1,i} &= \mu_{A_i}(x) \quad \text{for } i = 1,\ldots,n\\
O_{1,i+n} &= \mu_{A_{i+n}}(y),
\end{aligned}
$$

where x is the input and A_i is a fuzzy set associated with this node. In other words, outputs of this layer are the membership values of the premise part. Here the membership function can be characterized by the generalized bell-shaped function:

$$\mu_{A_i}(x) = \frac{1}{1+[(x_i c_i/a_i)^2]^{b_i}},$$

where $\{a_i, b_i, c_i\}$ is in the parameter set. Parameters in this layer are referred to as *premise parameters*.

Layer 2: Every node in this layer is a fixed node labeled Π, which multiplies the incoming signals and outputs the product,

$$O_{2,k} = w_k = \mu_{A_i}(x) \times \mu_{A_j}(y), \quad i,j = 1,\dots,n, \;\; k = 1,\dots,n^2.$$

Each node output represents the firing strength of a rule.

Layer 3: Every node in this layer is a fixed node labeled N. The ith node calculates the ratio of the ith rule's firing strength to the sum of all rule's firing strengths:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + \cdots + w_{n^2}}, \quad i = 1,\dots,n^2.$$

For convenience, the outputs from this layer are called *normalized firing strengths*.

Layer 4: Every node in layer 4 is an adaptive node with a node function

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i(p_1^i x + p_2^i y + p_0^i), \quad i = 1,\dots,n^2,$$

where $\bar{w}_i$ is the output of layer 3 and $\{p_0^i, p_1^i, p_2^i\}$ is in the parameter set. Parameters in this layer are called as *consequent parameters*.

Layer 5: The single node in this layer is a fixed node labeled $\sum$ that computes the overall outputs as the summation of all incoming signals, i.e.,

$$f = O_{5,1} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}, \quad i = 1,\dots,n^2.$$

3. PID controller using the ANFIS

To obtain parameters (K_P, K_I, K_D) for a PID controller more exactly, without using the approximation of arctan functions, we use the ANFIS [15–17] based on gain and phase margins to model these equations analytically.

Considering the nonlinear coupled Eqs. (7)–(10), we find that there are five parameters $(w_p, w_g, K_P, K_I, K_D)$ in those four equations. If we are given gain margin and phase margin specifications (A_m, ϕ_m), it may not be possible to solve for the five parameters analytically because the equations are nonlinear. Now, let us consider another approach. First, it is possible to give randomly controller parameters (K_P, K_I, K_D) as the input of these equations. Using Eq. (7), we can solve for w_p then substitute it into Eq. (8) to get A_m. And using Eq. (9), we can calculate w_g then substitute it into Eq. (10) to obtain ϕ_m. Hence we obtain the parameters (w_p, w_g, A_m, ϕ_m) that correspond to the controller parameters (K_P, K_I, K_D), respectively. Fig. 3 summarizes the approach. In preparation for training the ANFIS, we assign randomly points (K_P, K_I, K_D), obtain the corresponding (A_m, ϕ_m) points, and set them as the training data. That is, the input data are (A_m, ϕ_m) and the output are (K_P, K_I, K_D). Note that the training data satisfy the stability condition, i.e., $A_m > 0$ and $\phi_m > 0$. Thus, we can get our training data for the ANFIS. This approach avoids the possibility of not finding a solution to nonlinear Eqs. (7)–(10), and reduces the overall task. Furthermore, this approach is useful for all processes

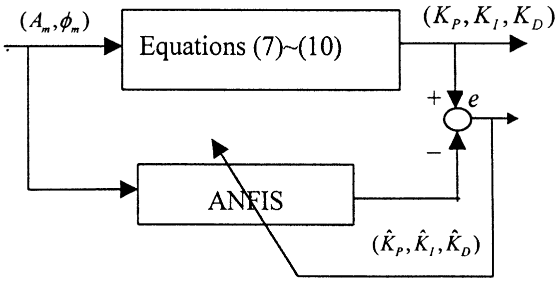

Fig. 3. Block diagram of function mapping using ANFIS.

(stable, unstable, higher-order, under-damped response, etc.). In Section 5, simulation results demonstrate the effectiveness of this approach.

Fig. 3 illustrates the block diagram of the function mapping of Eqs. (7)–(10) using the ANFIS. Suppose we are given (A_m, ϕ_m) and have R^i $(i=1,\ldots,n^2)$ implications, then the value of $y \in \{K_P, K_I, K_D\}$ is implied as follows.

3.1. Tuning of the ANFIS

We note that when the values of the premise parameters are fixed, the overall output $f = \{K_P, K_I, K_D\}$ can be expressed as a linear combination of consequent parameters. In symbols, the output f in Fig. 2 can be written as

$$f = \frac{w_1}{w_1 + \cdots + w_{n^2}} f_1 + \cdots + \frac{w_{n^2}}{w_1 + \cdots + w_{n^2}} f_{n^2} = \bar{w}_1 f_1 + \cdots + \bar{w}_{n^2} f_{n^2}$$
$$= (\bar{w}_1 A_m) p_1^1 + (\bar{w}_1 \phi_m) p_2^1 + (\bar{w}_1) p_0^1 + \cdots + (\bar{w}_{n^2} A_m) p_1^{n^2} + (\bar{w}_{n^2} \phi_m) p_2^{n^2} + (\bar{w}_{n^2}) p_0^{n^2},$$

which is linear in the consequent parameters $\{p_0^1, p_1^1, p_2^1, \ldots, p_0^{n^2}, p_1^{n^2}, p_2^{n^2}\}$. Note that if a fuzzy neural network output or its transformation is linear in some of the network's parameters, then we can identify these linear parameters using the well-known linear least-squares method [12]. Therefore, we use an off-line learning (the recursive least-square algorithm) to update the parameters of ANFIS. After the parameters are updated for each data presentation, we have an on-line learning scheme. This learning strategy [3] is vital to on-line parameter identification by systems with changing characteristics. In this learning scheme, we use back-propagation learning [21] to update the premise parameters $\{a_i, b_i, c_i\}$. Details for tuning the ANFIS can be found in [6,15–17].

4. Stabilizing region and valid region for PID controller

Some of the equations that appeared in the derivation are useful for assessing what is achievable by PID control. Firstly there are some restrictions on the choice of the gain and phase margins. One usual requirement is that the controller parameters $K_P > 0$, $K_I > 0$ and $K_D > 0$. Therefore, the suitable choice (specification) of gain and phase margins for the unstable process must be determined. In literature [10], Ho and Xu used the linear equation to approximate the $\tan^{-1}$ function that reduces the complex of Eqs. (7)–(10). Therefore, they found the relationship between A_m, ϕ_m, time-delay, and unstable-pole. An analysis method was developed to find the suitable choice of gain-phase margins. However, there are unsuitable specifications by using the result of [11], see Example 1. A simple method is proposed to estimate the valid region of A_m, ϕ_m for PID controller and the stabilizing region.

4.1. Procedure for estimating the stabilizing region and valid region

Step 1: Estimate the stabilizing range Ω roughly for controller parameters ($K_P > 0$, $K_I > 0$ and $K_D > 0$).

Step 2: Choose randomly (or uniformly) the testing data (K_P^i, K_I^i, K_D^i, $i = 1, \ldots, n$) from the region Ω and calculate the corresponding gain and phase margins using Eqs. (7)–(10).

Step 3: Find the data set Ω_P that every testing data in Ω results a stable closed-loop system ($A_m > 0$ and $\phi_m > 0$).

Step 4: Estimate the stabilizing region Ω_P^* of parameter (K_P, K_I, K_D) from these points, i.e., find the boundary of Ω_P (here we can omit the isolated point that are far from the grouped points).

Step 5: Choose randomly on the closure of Ω_P^* and calculate the corresponding A_m and ϕ_m using Eqs. (7)–(10). Then, the valid region of gain and phase margin specification for the PID controller can be obtained.

Remark 2. It is known that choosing proper training data is important for neural network system. In this case, gain and phase margins are input data while the PID parameters are the output data. In the preceding discussion, we explained that we get our training data by giving the PID parameters randomly to derive the desired output (gain margin and phase margin). The stability of the closed-loop system depends on the PID parameters we choose. Thus, the above method for finding the stabilizing region that guarantees the validity of the training data.

Example 1. Unstable plant $G_p(s) = e^{-0.2s}/s - 1$ with PI controller [11].

First, we roughly give the stabilizing range of K_P and K_I as [0, 10] and [0 10]. Then the testing data are chosen randomly. For each pair (K_P, K_I), the corresponding gain and phase margin can be obtained. Then, find the points set that satisfy $A_m > 0$ and $\phi_m > 0$. We omit the isolated points that far from the grouped points and estimate the stabilizing region Ω_P^* of parameter (K_P, K_I) from these points. Fig. 4 shows the estimated stabilizing region of PI controller for the unstable plant $G_p(s) = e^{-0.2s}/s - 1$. Finally, we would estimate the valid region of gain and phase margins using the information provided by the stabilizing region. Points chosen randomly on the closure of the stabilizing region (K_P, K_I) are used to calculate the corresponding gain and phase margins using Eqs. (7)–(10). Then, the valid region of gain and phase margin specification for the PI controller can be obtained. Fig. 5 shows the estimated valid range for the unstable plant $G_p(s) = e^{-0.2s}/s - 1$ with PI controller. Note that, the form of the controller Ho et al. [9–11] used was $G_c(s) = K_c(1 + (1/sT_I))$. By comparing these two forms of controllers, we have $K_C = K_P$ and $T_I = K_P/K_I$.

Remark 3. Denote Ω_1 (dash–dotted line) and Ω_2 (solid-line) as the estimated valid regions for the result in [11] and our approach. From Fig. 5 and Table 1, it is clear that $P_1, P_2, P_3 \notin (\Omega_1 \cup \Omega_2)$ and $P_4, \ldots, P_{10} \in \Omega_1$ but $P_4, \ldots, P_{10} \notin \Omega_2$. By testing these data, we obtain that these 10 specifications by using the approximated method [11] give unavailable results (at least one of K_C, T_I, A_m, ϕ_m is negative, see the shadow items in Table 1) because of the approximation of $\tan^{-1}$ function. Since $P_1, P_2, P_3 \notin \Omega_1$, we got $T_I < 0$. On the other hand, $P_4, \ldots, P_{10} \in \Omega_1$ we have parameters $K_C, T_I > 0$ and wrong phase margin ($\phi_m < 0$ or ∞). Note that, even for these unreasonable specifications, the ANFIS provides suitable controller parameters that guarantee the closed-loop system stability. In this paper, the system plant is directly used to design the controller. Therefore, we avoid the above results using the ANFIS. In the following section, we will show the comparison of simulations between the results of [11] and our approach.

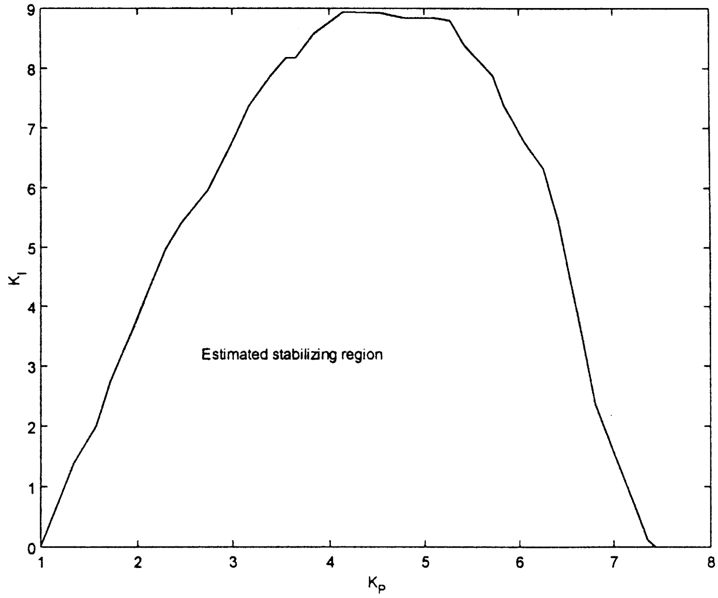

Fig. 4. Estimated stabilizing region of (K_P, K_I).

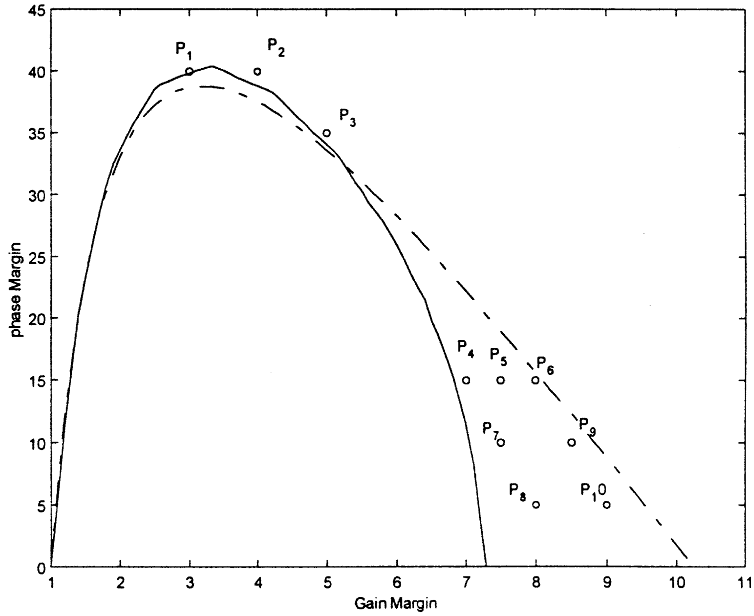

Fig. 5. Valid region of (A_m, ϕ_m) for PI controller: solid line, our result; dash–dotted line: result of [10] (P_1–P_{10} are outside the estimated valid region).

Table 1
Comparison result of unreasonable specifications

	Specifications	Results of [11]				ANFIS results			
		A_m	ϕ_m	K_C	T_I	A_m	ϕ_m	K_C	T_I
P_1	(3, 40)	3.0420	41.8603	2.3998	**−17.3642**	3.0536	39.9748	2.3818	96.6836
P_2	(4, 40)	4.0559	42.0461	1.8035	**−13.0428**	3.9295	38.3010	1.8476	37.5771
P_3	(5, 35)	5.0700	35.9405	1.4363	**−29.5354**	4.8060	33.1555	1.5076	21.7432
P_4	(7, 15)	7.0952	**−2.6003**	1.0090	8.5340	6.9303	9.7452	1.0478	40.3184
P_5	(7.5, 15)	7.6038	**−13.3516**	0.9477	16.4203	6.9303	9.7452	1.0478	40.3184
P_6	(8, 15)	8.1117	∞	0.8934	101.3638	6.9303	9.7452	1.0478	40.3184
P_7	(7.5, 10)	7.6012	**−12.6772**	0.9398	7.3770	6.9303	9.7452	1.0478	40.3184
P_8	(8, 5)	8.1072	**−20.2941**	0.8796	6.6594	6.9303	9.7452	1.0478	40.3184
P_9	(8.5, 10)	8.6184	∞	0.8390	32.6130	6.9303	9.7452	1.0478	40.3184
P_{10}	(9, 5)	9.1249	∞	0.7909	20.7626	6.9303	9.7452	1.0478	40.3184

Table 2
Different PI controllers for $G_p(s) = 100e^{-0.01s}/(s^2 + 10s - 5)$

Specifications		Result						Error	
A_m	ϕ_m°	K_P	K_I	A_m^*	ϕ_m^*	w_g	w_p	Error of A_m (%)	Error of ϕ_m° (%)
2	30	0.5894	0.003808	1.9356	29.8200	8.3540	5.0432	3.220	0.600
3	45	0.3741	0.0024667	3.0500	44.5976	8.3539	3.3631	1.667	0.894
4	50	0.2784	0.0176454	4.0734	49.8999	8.3170	2.5372	1.835	0.200

5. Simulation results

In this section, we give a specific performance comparison with GPM [11] because both were designed based on gain and phase margin specifications.

Example 2. PI controller for a second-order process.
The process is given as follows:

$$G_p(s) = \frac{100e^{-0.05s}}{s^2 + 10s - 5}.$$

Since the process is not a first-order type, the GPM cannot be applied. Various gain and phase margins are specified for this model in Table 2. The ANFIS yields less than 3.5% and 0.9% for desired gain and phase margin specifications.

Example 3. PI controller for a first-order with time-delay process.
The process is given as

$$G_p(s) = \frac{e^{-0.2s}}{s - 1}, \quad L/\tau = 0.2 < 1.$$

The results for different specifications in this example are illustrated in Table 3, which shows that even when the plant is first-order with time-delay, the proposed ANFIS approach has better performance than GPM.

Table 3
Different PI controllers for $G_p(s) = e^{-0.2s}/(s-1)$

Tuning method	Specifications		Result						Error	
	A_m	ϕ_m°	K_C	T_I	A_m^*	ϕ_m^*	w_g	w_p	Error of A_m (%)	Error of ϕ_m° (%)
Results of [11]	3	35	2.3453	6.6120	3.2074	35.9332	0.4442	2.1181	6.91	2.67
	4	30	1.7453	4.5773	4.2593	29.4323	0.5411	1.4408	6.48	1.89
	5	30	1.4181	12.0744	5.3666	27.8533	0.3257	0.9813	7.33	7.16
ANFIS	3	35	2.3829	4.9050	2.9866	34.8414	0.5188	2.1716	0.45	0.45
	4	30	1.7636	4.1793	4.0181	29.6371	0.5648	1.4773	0.45	1.21
	5	30	1.4452	10.8200	4.9882	29.1893	0.3441	1.0469	0.24	2.70

Table 4
Results for $G_p(s) = e^{-0.2s}/(s-1)$ with PID controllers under different specifications

Specifications		Results					Errors	
A_m	ϕ_m	K_P	K_I	K_D	A_m^*	ϕ_m^*	Error of A_m (%)	Error of ϕ_m° (%)
2	20	5.1641	0.1081	0.2915	2.0278	20.3724	1.390	1.862
3	30	5.5592	0.0135	0.3102	2.9998	30.3809	0.007	1.270
4	40	2.6842	0.1798	0.0684	3.9787	40.5597	0.5325	1.400

ANFIS yields less than a 2% error but GMP's is greater than 5%. Also, Table 1 shows the comparison with the result of [11] for unreasonable specifications.

Remark 4. It is clear that the results of ANFIS all satisfy the stability conditions $K_C, T_I > 0$ and $A_m, \phi_m > 0$. Even if someone gives unreasonable specifications (outside the valid region Ω_2) for PI controller, we have a stable system using the ANFIS. This guarantees the stabilization of the proposed PI controller.

Example 4. PID controller for a first-order with time-delay process.
The process is given as

$$G_p(s) = \frac{e^{-0.2s}}{s-1}, \quad L/\tau = 0.2 < 1.$$

In this example, we use the PID controller to compensate the first-order unstable process. ANFIS also yields less than a 2% error in this case. Table 4 shows the simulation results for different specifications. However, the ANFIS gives acceptable errors for the specified gain and phase margins.

6. Conclusion

This paper has investigated the PID tuning method using fuzzy neural system (ANFIS) based on gain and phase margin specifications. The proposed method has been generalized to determine the PID controller parameters for general processes that include the test batch and common used model of the typical industrial processes. There are two advantages to use the ANFIS for formulating gain and phase margin problems. First, the trained ANFIS automatically tunes the PID controller parameters for different gain and phase margin specifications so that neither numerical methods nor graphical methods need be used. Second, the ANFIS can also find the relationship between PID controllers (K_P, K_I, K_D) and specifications (A_m, ϕ_m) in the weighting parameters in the networks. Therefore, the proposed method is simple and systematic in reducing the complexity of the problem presented in this paper. A simple method was also developed to estimate the stabilizing region of controller parameters and valid region for gain-phase margin specification. The ANFIS can still find suitable PID controller parameters that guarantee the stabilization even for unreasonable specifications. That is, the ANFIS can provide controller parameters for guaranteeing the stability of the closed-loop system. Simulation results have shown that the ANFIS can achieve the specified values efficiently.

References

[1] K.J. Astrom, T. Hagglund, Automatic tuning of simple regulators with specifications on phase and amplitude margins, Automatica 20 (1984) 645–651.

[2] K.J. Astrom, T. Hagglund, PID Controllers: Theory, Design, and Tuning, Research Triangle Park, NC, ISA, 1995.

[3] A.G. Barto, R.S. Sutton, C.W. Anderson, Neuron like adaptive elements that can solve difficult learning control problems, IEEE Trans. Systems Man Cybernet. SMC-13 (1983) 834–846.

[4] I.-L. Chien, P.S. Fruehauf, Consider IMC tuning to improve controller performance, Chem. Eng. Progr. 86 10 (1990) 33–41.

[5] Y.C. Chen, C.C. Teng, A model reference control structure using a fuzzy neural network, Fuzzy Sets and Systems 73 (1995) 291–312.

[6] S.Y. Chu, C.C. Teng, Tuning of PID controllers based on gain and phase margin specifications using fuzzy neural network, Fuzzy Sets and Systems 101 (1999) 21–30.

[7] G.H. Cohen, G.A. Coon, Theoretical consideration of retarded control, Trans. ASME 75 (1953) 827–834.

[8] G.F. Franklin, J.D. Powell, A.E. Baeini, Feedback Control of Dynamic Systems, Addison-Wesley, Reading, MA, 1986.

[9] W.K. Ho, C.C. Hang, L.S. Cao, Tuning of PID controllers based on gain and phase margin specification, Automatica 31 (1995) 497–502.

[10] W.K. Ho, C.C. Hang, J. Zhou, Self-tuning PID control of a plant with under-damped response with specifications on gain and phase margins, IEEE Trans. Control Systems Technol. 5 (1997) 446–452.

[11] W.K. Ho, W. Xu, PID tuning for unstable processes based on gain and phase-margin specifications, IEE Proc.—Control Theory Appl. 145 (1998) 392–396.

[12] T.C. Hsia, System Identification: Least-Squares Methods, Heath, New York, 1977.

[13] R. Isermann, Digital Control Systems, Springer, Berlin, 1981.

[14] L. Jan, Robust Multivariable Feedback Control, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[15] J.S. Jang, ANFIS: Adaptive-Network-Based Fuzzy Inference System, IEEE Trans. Systems Man Cybernet. 23 (1993) 665–685.

[16] J.S. Jang, N. Gulley, Fuzzy Logic Toolbox User's Guide, The Math Works Inc., January 1995.

[17] J.S. Jang, C.T. Sun, Neuro-Fuzzy Modeling and Control, Proceedings of IEEE 83 (1995) 378–405.

[18] C.T. Lin, C.S.G. Lee, Neural-network-based fuzzy logic control and decision system, IEEE Trans. Comput. 40 (12) (1991) 1320–1336.

[19] K. Ogata, Modern Control Engineering, 2nd Edition, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[20] J.X. Xu, C. Liu, C.C. Hang, Tuning and analysis of a fuzzy logic controller based on gain and phase margins, Proc. of the American Control Conference, June 1995, pp. 3234–3238.

[21] J.X. Xu, C. Liu, C.C. Hang, Tuning of fuzzy PI controllers based on gain/phase margin specifications and ITAE index, ISA Trans. 35 (1996) 79–91.