ELSEVIER

Mechatronics 14 (2004) 299–326

MECHATRONICS

Adaptive fuzzy logic controller for feed drives of a CNC machine tool

Sungchul Jee [a,*], Yoram Koren [b]

[a] Department of Mechanical Engineering, Dankook University, 8 Hannam-dong, Yongsan-ku, Seoul 140-714, South Korea

[b] Engineering Research Center for Reconfigurable Machining Systems, The University of Michigan, Ann Arbor, MI 48109, USA

Received 29 April 2002; accepted 18 February 2003

Abstract

This paper introduces an adaptive fuzzy logic controller (AFLC) for precision contour machining which adjusts both input and output membership functions simultaneously. The parameters of the proposed controller are self-tuned in real-time according to a continuous measurement of the performance of the controller itself and estimated disturbance values. The adjustment of the membership functions are restricted within a stable range derived from a stability analysis. The proposed controller as well as a conventional fuzzy logic controller and a PID controller were simulated and implemented on a 3-axis milling machine in contour milling. Both the simulations and experiments show that the AFLC can provide superior performance in terms of contour tracking accuracy compared with the other two controllers particularly in machine tool systems where friction is a serious problem in the feed drives.

Keywords: Computer numerical control; Machine tools; Contour machining; Friction; Fuzzy logic control

1. Introduction

In order to achieve high precision contour machining, many efforts have been made to develop more accurate computerized numerical control (CNC) systems. In precision machining, however, the friction in the moving components of machine

* Corresponding author. Tel.: +82-2-709-2911; fax: +82-2-709-2569.
E-mail addresses: scjee@dku.edu (S. Jee), ykoren@umich.edu (Y. Koren).

0957-4158/$ - see front matter
doi:10.1016/S0957-4158(03)00031-X

tools can cause significant machining errors. In order to improve contouring accuracy in the presence of friction disturbances, we proposed to control the machine with a rule-based controller [1] using fuzzy logic rather than model-based compensation techniques [2]. The reason for this rule-based controller is that the implementation of model-based compensation controllers for friction has limitations due to the complexity and the nonlinearities of the friction characteristics. The fuzzy logic controller (FLC) can reduce the effect of cutting force disturbances as well. The FLC, whose control rules are designed with respect to phase plane determined by error and change of error, can be considered as an extension of a sliding mode controller and if properly designed, it is suitable for uncertain and complicated processes which have model uncertainties, nonlinearity and other ill-defined properties. It is asserted that fuzzy logic control is less sensitive to variation in system parameters than conventional controls and can cope with disturbances which deteriorate system performance. Therefore, in cases where disturbances such as guideway friction are a serious problem, the application of fuzzy logic control to servo-control of CNC machine tools shows promise for better contour tracking performance.

Although a fixed parameter FLC can be recommended as an alternative to a conventional controller especially in the presence of large and variable friction disturbances, it does not have the capability of self-adjusting its parameters. This inability to self-adjust its parameters has two drawbacks: (i) it requires a complex tuning procedure for each machine when installing the controller, and (ii) it cannot cope with changing operating conditions in machining.

This paper deals with a fuzzy logic approach to compensate for friction in order to improve tracking performance in high precision contour machining. In this paper, we introduce a fuzzy logic control strategy that contains a new adaptation method. The proposed adaptation method automatically tunes the controller by shifting and changing the shapes of the fuzzy logic membership functions according to a real-time performance measure. The adjustment of the membership functions are restricted within a stable range derived from a stability analysis for the fuzzy logic control system. This strategy, which only adds a small computation load, can efficiently self-tune the controller parameters at the time of installation and adapt them to time-varying processes. In addition, in order to reduce the contour errors (namely, deviations from the desired path) due to stiction in the guideways, a low-velocity friction compensation strategy is also contained in the proposed method, and is based on adjusting the output membership functions according to estimated friction values [1]. To our knowledge, this is the first time that an adaptive fuzzy logic controller (AFLC) (of any type) has been implemented on a CNC machine in contour machining applications.

This paper is composed of six sections. A conventional fuzzy logic control is briefly explained in Section 2. In Section 3, the stability analysis for the fuzzy logic control system is presented to determine a stable range for the controller parameters, limiting the location of membership functions during adaptation. In Section 4, the principles of the proposed AFLC are described. The proposed method is verified through simulation analyses and actual contour tracking experiments in Section 5. Finally, conclusions are given in Section 6.

2. Fuzzy logic control

This section briefly describes the elements of a conventional FLC (that is included in the proposed AFLC) and explains its structure.

2.1. Membership functions

A fuzzy set is characterized by a membership function whose value represents a degree of membership to the fuzzy set having a value between 0 and 1. The membership function can be defined with parameters. In general, an error (which is a difference between a desired process state and an actual process output) and the change in the error are used as inputs to an FLC. The membership functions used in this study for the error can be parametrized by their left edges (A_i^{l}), centers (A_i^{c}), and right edges (A_i^{r}) describing the fuzzy sets A_i (see Fig. 1(a)). The shape shown in Fig. 1(a) can be expressed mathematically by

$$\mu_{A_i}(e) = \begin{cases} 0 & \text{for } e < A_i^{\mathrm{l}} \\ (e - A_i^{\mathrm{l}})/(A_i^{\mathrm{c}} - A_i^{\mathrm{l}}) & \text{for } A_i^{\mathrm{l}} \leqslant e < A_i^{\mathrm{c}} \\ (A_i^{\mathrm{r}} - e)/(A_i^{\mathrm{r}} - A_i^{\mathrm{c}}) & \text{for } A_i^{\mathrm{c}} \leqslant e < A_i^{\mathrm{r}} \\ 0 & \text{for } e \geqslant A_i^{\mathrm{r}} \end{cases} \tag{1}$$

Similarly, the membership functions of fuzzy sets B_j for the change in the errors Δe can be defined in the same manner with parameters B_j^{l}, B_j^{c} and B_j^{r}.

The support of a fuzzy set is the set which contains all the elements in the universe of discourse (i.e., the space where a fuzzy set is defined) that have a nonzero grade of membership (see Fig. 1(a)). In the particular case where the support of a fuzzy set is a single point (where the grade of membership is 1), the fuzzy set is called a fuzzy singleton (see Fig. 1(b)). The output fuzzy sets C_{ij} can also be defined identically to the input fuzzy sets. To remove the computational burden for real-time control, however, we used a simplified center of area defuzzification method, by which only the centroids are considered for the corresponding output membership functions. In other words, for simplicity, we used fuzzy singletons for the fuzzy outputs, and they can be represented by

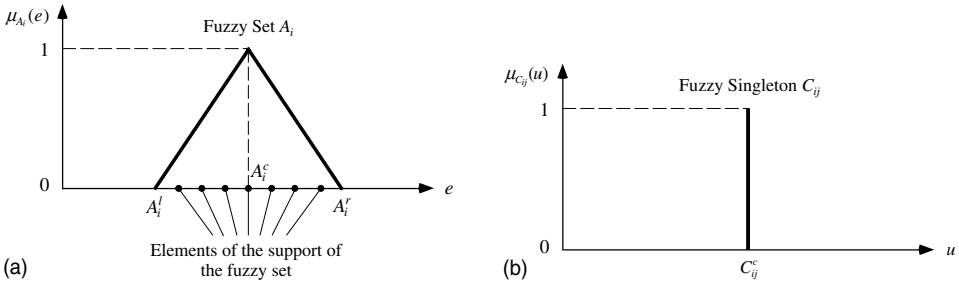

Fig. 1. Parameters describing a membership function: (a) typical; (b) singleton.

$$\mu_{C_{ij}}(u) = \begin{cases} 1 & \text{if } u = C_{ij}^{c} \\ 0 & \text{if } u \neq C_{ij}^{c} \end{cases} \tag{2}$$

where the control command u defined in a discrete universe in our case, has a quantized value and C_{ij}^{c} are the centroids of the output membership functions.

2.2. Control rules

The operation of an FLC is based on fuzzy control rules that are contained in a control-rule base. These rules utilize the linguistic values of A_i (fuzzy sets for the position error), B_j (fuzzy sets for the change in the error) and C_{ij} (fuzzy sets for the control action), and have the following form:

$$\text{If } e \in A_i \quad \text{and} \quad \Delta e \in B_j \text{ then } u = C_{ij} \tag{3}$$

A larger magnitude of $|A_i^{l} - A_i^{r}|$ increases the probability that the corresponding control rule will be activated. Therefore, by changing the values A_i^{l} and A_i^{r}, one can change the effectiveness of a control rule. In our controller, for each A_i, B_j and C_{ij}, we assign one of the following seven linguistic labels: negative large (NL), negative medium (NM), negative small (NS), nearly zero (ZR), positive small (PS), positive medium (PM), and positive large (PL). An example is shown in Fig. 2 for the position error e. The variable Δe has similar fuzzy values. Since in our controller there are two control inputs (e and Δe) and seven fuzzy sets are defined for each input, there are a total of 49 control rules that we have determined and stored in the rule base. The control rules used in the proposed controller are shown in Table 1 and are clarified in Appendix A.

2.3. Structure of fuzzy logic control

A block diagram of a conventional FLC is shown in the upper block of Fig. 6 (see Section 4). An FLC is composed of three main parts: fuzzification, inference engine with a rule base, and defuzzification. We used as inputs the position error at the current time step, $e(k)$, and the change in the position errors between the previous and current steps, $\Delta e(k)$. Through fuzzification, the controller inputs are converted

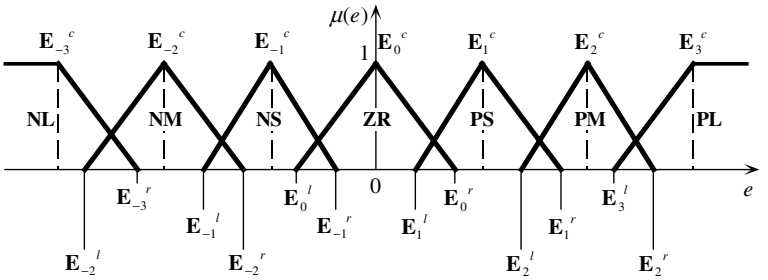

Fig. 2. Parameters of the membership functions for the position error.

Table 1
The fuzzy control rule base inside the proposed controller (rule base I)

Control	Actions	If $\Delta e \in$						
		NL	NM	NS	ZR	PS	PM	PL
If $e \in$	NL	NL	NL	NL	NL	NL	NL	NL
	NM	NL	NL	NM	NM	NS	NS	NS
	NS	NL	NM	NM	NS	NS	NS	ZR
	ZR	ZR	ZR	ZR	ZR	ZR	ZR	ZR
	PS	ZR	PS	PS	PS	PM	PM	PL
	PM	PS	PS	PS	PM	PM	PL	PL
	PL	PL	PL	PL	PL	PL	PL	PL

to fuzzy variables (A_i and B_j), where each fuzzy variable has a corresponding linguistic label. After fuzzification, the converted fuzzy input variables are transformed into a fuzzy output variable (C_{ij}) through the inference engine aided with a control rule base. The inference engine produces the overall fuzzy output (C) from individually activated control rules. The defuzzification unit receives the fuzzy variable C and generates a crisp (nonfuzzy) control action u_0 which is the control output.

3. Stability analysis of the FLC system

From a controller design point of view, we performed an analysis of the proposed FLC system. This enables us to predict and determine the controller parameters in a systematic way so that they guarantee stable system behavior. We utilized an approach for the FLC system based on Jury and Lee's theorem [3] which is an extension of Popov's method of stability analysis [4] to a discrete-time system with multi-nonlinearities. This method provides sufficient conditions for the absolute stability of nonlinear control systems. The proposed fuzzy logic control system belongs to the category of a general discrete control system shown in Fig. 3. $\mathbf{r}(k)$ is a reference input vector and $\mathbf{y}(k)$ is a system output vector. $\mathbf{e}(k)$ is an error vector and corresponds to inputs to a nonlinear controller φ which is a set of m nonlinear gain elements. The proposed FLC output for each nonlinear element can be represented by

$$u_i(k) = \varphi_i[e_i(k)], \quad i = 1, 2 \tag{4}$$

where $\varphi_i[e_i(k)]$ is the output of the ith nonlinear element in $\boldsymbol{\varphi}$ and has a bounded derivative for each nonlinear element and e_i is the ith component of the controller

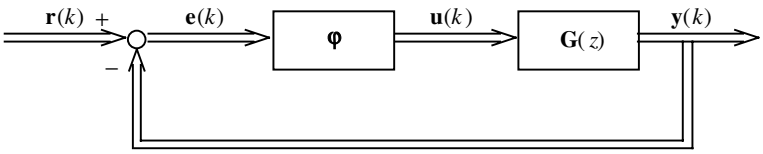

Fig. 3. Discrete control system with multi-nonlinearities in the controller.

input vector $\mathbf{e}(k)$. $\mathbf{G}(z)$ is the representation of the system transfer function matrix with a zero-order-hold.

In order to perform the stability analysis, we need a mathematical formulation of the FLC. To make the execution of an analytic formulation facile, we defined the input membership functions such that an end point of a membership function coincides with a centroid of a neighboring membership function. In addition, we let the centroids of the input membership functions be evenly spaced in their universes of discourse, and defined the constant distances between the membership functions as follows:

$$\begin{aligned} \alpha &= E_{i+1} - E_i \\ \beta &= \Delta E_{j+1} - \Delta E_j \end{aligned} \tag{5}$$

where E_i and ΔE_j are the centroids of the membership functions for e and Δe, respectively.

As a general case, we assume that $e(k)$ and $\Delta e(k)$ are located in the following range:

$$\begin{aligned} E_i &\leqslant e(k) \leqslant E_{i+1} \\ \Delta E_j &\leqslant \Delta e(k) \leqslant \Delta E_{j+1} \end{aligned} \tag{6}$$

If we use the product operation as a fuzzy implication and the simplified defuzzification method, the FLC output in the proposed approach can be represented by

$$u(k) = U_{i,j} + (1/\alpha)(U_{i+1,j} - U_{i,j})[e(k) - E_i] + (1/\beta)(U_{i,j+1} - U_{i,j})[\Delta e(k) - \Delta E_j] \tag{7}$$

where $U_{i,j}$, a centroid of the output membership function, denotes an FLC output when $e = E_i$ and $\Delta e = \Delta E_j$.

For the sake of analysis, the FLC output may be decomposed into two parts as follows:

$$u(k) = u_1(k) + u_2(k) \tag{8}$$

where $u_1(k)$ is the control output component issued only by the position error, and $u_2(k)$ is the remaining part of the control output (i.e., $u - u_1$) due to both the position error and the change in the position error. Especially when $\Delta e = 0$ (equivalently $\Delta E_j = 0$ with $j = 0$), the FLC output is caused only by the position error and is defined as

$$u_1(k) = U_{i,0} + (1/\alpha)(U_{i+1,0} - U_{i,0})[e(k) - E_i] \tag{9}$$

Accordingly, $u_2(k)$ is defined as follows:

$$\begin{aligned} u_2(k) = {} & (U_{i,j} - U_{i,0}) + (1/\alpha)[(U_{i+1,j} - U_{i+1,0}) - (U_{i,j} - U_{i,0})][e(k) - E_i] \\ & + (1/\beta)(U_{i,j+1} - U_{i,j})[\Delta e(k) - \Delta E_j] \end{aligned} \tag{10}$$

Therefore, the control output components are related by

$$u_1(k) = \varphi_1[e(k)] \quad \text{and} \quad u_2(k) = \varphi_2[e(k), \Delta e(k)] \tag{11}$$

where φ_1 and φ_2 are nonlinear gain elements. If we define the gains which confine $u_1(k)$ and $u_2(k)$ as K_1 and K_2, respectively, then

$$0 < u_1(k)e(k) < K_1[e(k)]^2 \quad \text{and} \quad 0 < u_2(k)\Delta e(k) < K_2[\Delta e(k)]^2 \tag{12}$$

which restrict the nonlinear elements φ_1 and φ_2 to sectors $[0, K_1]$ and $[0, K_2]$, respectively. K_1 is the maximum slope of $\varphi_1(e)$ with respect to the origin, and u_1 is piecewise linear between the centroids of the membership functions for e. Therefore, without loss of generality, K_1 can be determined by finding the maximum of the slopes at points where $e = E_i$ (i.e., centroids) (see Fig. 5). From Eq. (9) with $e = E_i$, the output from φ_1 is $u_1 = U_{i,0}$. Since u_2 is piecewise linear between the centroids of the membership functions for e and Δe, respectively, the maximum slope of $\varphi_2(e, \Delta e)$, K_2, can be derived from the maximum of the slopes at points where $\Delta e = \Delta E_j$ for all possible position errors, equivalently for all E_i. From Eq. (10) with $e = E_i$ and $\Delta e = \Delta E_j$, $u_2 = U_{i,j} - U_{i,0}$. In summary, in relation to the FLC parameters, K_1 and K_2 can be expressed as follows:

$$K_1 = \max_i \frac{\varphi_1(E_i)}{E_i} = \max_i \frac{U_{i,0}}{E_i} \quad \text{for } i \neq 0 \tag{13}$$

$$K_2 = \max_j \frac{\varphi_2(\Delta E_j)}{\Delta E_j} = \max_{i,j} \frac{U_{i,j} - U_{i,0}}{\Delta E_j} \quad \text{for } j \neq 0 \tag{14}$$

The nonlinear gain φ_2 depends on e as well as Δe, and in general it is a different function according to the value of e. However, with the rule base in Table 1, all these functions are confined to the region $0 \leqslant \varphi_2 \leqslant K_2 \Delta e$ (see Fig. 17(a)).

In order to apply Jury and Lee's theorem to the proposed FLC system, we define

$$\mathbf{G}(z) = \begin{bmatrix} G_{11}(z) & G_{12}(z) \\ G_{21}(z) & G_{22}(z) \end{bmatrix} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix} \tag{15}$$

with

$$\mathbf{e}(k) = \begin{bmatrix} e(k) \\ \Delta e(k) \end{bmatrix} \quad \text{and} \quad \mathbf{u}(k) = \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \tag{16}$$

In Eq. (15),

$$\begin{aligned} G_{11}(z) &= G_{12}(z) = G(z) \\ G_{21}(z) &= G_{22}(z) = (1 - z^{-1})G(z) \end{aligned} \tag{17}$$

where $G(z)$ is the open-loop transfer function in the discrete domain with a zero-order-hold. If we represent the feed drive system by the following transfer function

$$G(s) = \frac{K}{s(\tau s + 1)} \tag{18}$$

where K is the open-loop gain multiplied by the encoder gain and τ is the time constant, the discrete transfer function becomes

$$G(z) = K_0 \frac{z + a}{(z - 1)(z - b)} \tag{19}$$

where

$$K_0 = K \frac{T/\tau - 1 + b}{1/\tau}, \quad a = \frac{1 - b - (T/\tau)b}{T/\tau - 1 + b} \quad \text{and} \quad b = \mathrm{e}^{-T/\tau} \tag{20}$$

With Eqs. (15)–(17), (19) and (20), Jury and Lee's theorem yields the following absolute stability conditions for the FLC system.

Condition (i):

$$\frac{1}{K_1} > \max_{x \in [-1,1]} \left\{ \frac{K_0[2ax^2 + (1 + a)(1 - b)x - a - (1 + b) + ab]}{2(x - 1)(1 + b^2 - 2bx)} \right\} \tag{21}$$

Condition (ii):

$$K_2(g_3x^3 + g_2x^2 + g_1x + g_0) < h_2x^2 + h_1x + h_0 \quad \text{for all } x \in [-1, 1] \tag{22}$$

where

$$\begin{aligned}
g_3 &= 16K_0a \\
g_2 &= 8K_0[(1 - ab) - 2a] \\
g_1 &= 2K_0[K_0K_1a - 4(1 - ab) - 4(a + b)] \\
g_0 &= K_0[K_0K_1(a^2 + 1) + 8(a + b)] \\
h_2 &= 8(K_0K_1a + 2b) \\
h_1 &= 4[K_0K_1(1 + a)(1 - b) - 2(1 + b)^2] \\
h_0 &= 4\{2(1 + b^2) + K_0K_1[-a - (1 + b) + ab]\}
\end{aligned}$$

In our case, $K = 28.3$, $\tau = 0.055$ and $T = 0.01$, and the resulting stable range for the sectors is illustrated in Fig. 4. For the control rule base in Table 1, with the evenly

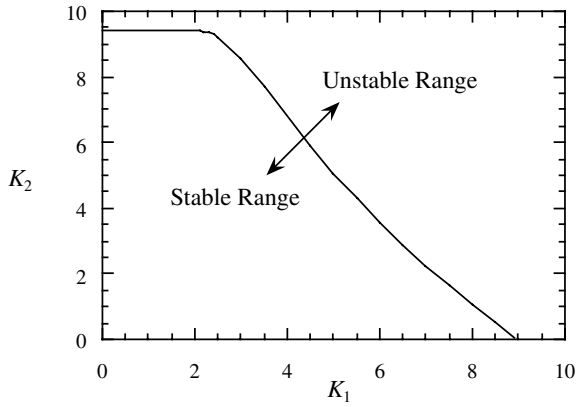

Fig. 4. Stable range for the FLC sectors.

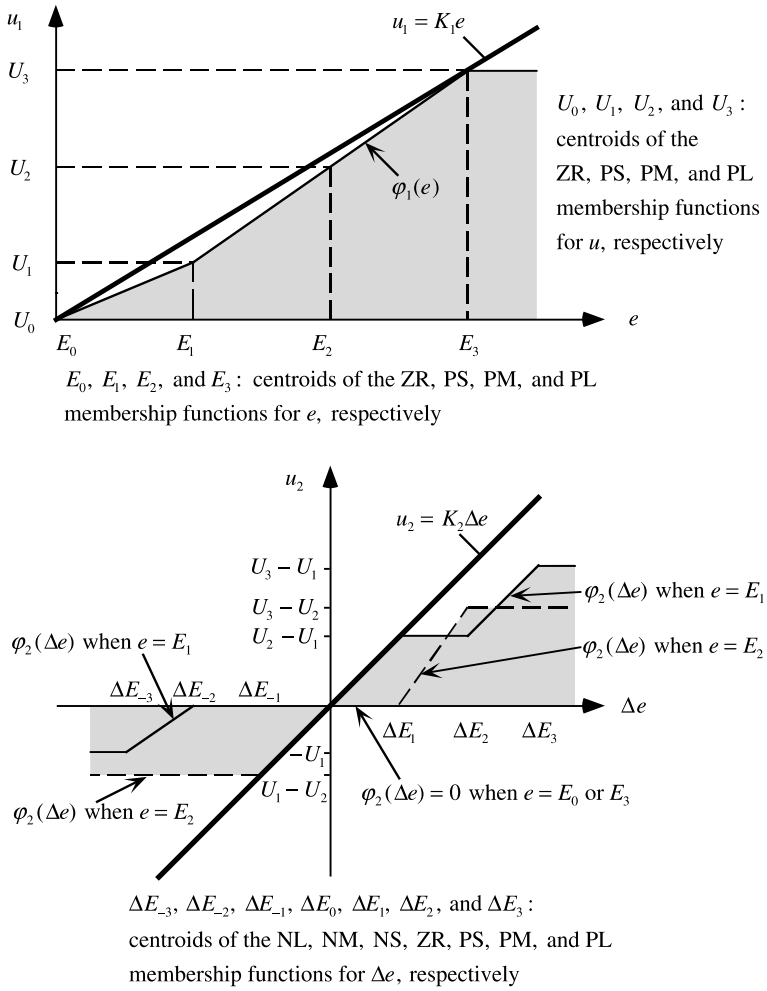

Fig. 5. Relationship between the FLC inputs and outputs.

spaced centroids of the input membership functions (i.e., A_i^c and B_j^c) in their universes of discourse, the relationships between the controller inputs and outputs can be represented as shown in Fig. 5. From Fig. 5, we can see that the sector K_2 for the change in the error is determined by the slopes corresponding to the projections on the curves: $(U_2 - U_1)/\Delta E_1$ $(= (U_1 - U_2)/\Delta E_{-1})$, $(U_3 - U_2)/\Delta E_2$, $(U_3 - U_1)/\Delta E_3$ and $-U_1/\Delta E_{-3}$. With Eq. (5), the magnitudes of the first three slopes above are mutually related to the sign of $3U_2 - 2U_1 - U_3$. That is, if $3U_2 - 2U_1 - U_3 \geqslant 0$, $(U_2 - U_1)/\Delta E_1$ is the maximum among the three, and otherwise $(U_3 - U_2)/\Delta E_2$ is the maximum. The slope $-U_1/\Delta E_{-3}$, equivalently $U_1/\Delta E_3$, is not directly relevant to the others and needs to be compared with the maximum of the other slopes. Again, with Eq. (5), it can be induced that $U_1/\Delta E_3$ is larger than $(U_2 - U_1)/\Delta E_1$ and $(U_3 - U_2)/\Delta E_2$ when

$3U_2 - 4U_1 < 0$ and $3U_3 - 3U_2 - 2U_1 < 0$, respectively. In conclusion, K_2 reduces to the following equation:

$$\begin{aligned}
&\text{If } 3U_2 - 2U_1 - U_3 \geqslant 0, \text{ then} \\
&\quad \text{if } 3U_2 - 4U_1 \geqslant 0, \text{ then } K_2 = (U_2 - U_1)/\Delta E_1 \\
&\quad \text{else } K_2 = U_1/\Delta E_3 \\
&\text{else} \\
&\quad \text{if } 3U_3 - 3U_2 - 2U_1 \geqslant 0, \text{ then } K_2 = (U_3 - U_2)/\Delta E_2 \\
&\quad \text{else } K_2 = U_1/\Delta E_3
\end{aligned} \tag{23}$$

where U_k ($k = 1, 2, 3$) are the centroids of the PS, PM and PL membership functions for the controller output. Based on Eqs. (13) and (23), and the predetermined stable range (Fig. 4), we established constraints on the range of movement for the output membership functions and restricted the location of the membership functions within the stable range.

4. The proposed adaptive fuzzy logic control

The performance of a conventional FLC is dependent on pre-defined fuzzy sets (i.e., membership functions) and a set of control rules. Thus, if the fuzzy sets or the control rules are not defined adequately, or if the controlled process behavior changes, the controller is not effective. To make it effective, the membership functions and the control rules must be modified and re-tuned in real time. Methods which do this re-tuning automatically are called adaptive fuzzy logic control (AFLC). Many adaptation methods have been proposed [5–15], and they can be classified as follows [6]:

(a) methods that directly change the set of control rules [13];
(b) methods that adjust the shapes or individual range of membership function [6,8];
(c) methods that change the elements describing the universe of discourse [5,7–12, 14,15].

The AFLC in [8] is based on a combination of methods (b) and (c). This adaptation algorithm modifies the parametrized input membership functions. Depending on the evaluation of activated control rules at the last time step, it modifies the corresponding input membership functions by either shifting them or adjusting the areas covered by them. Recently, fuzzy model reference learning control (FMRLC) has been proposed in [9,11] which uses a reference model to automatically modify the knowledge base by adjusting the output membership functions based on the difference between the plant and reference model outputs. This approach corresponds to method (c), which can be performed by either (i) adjusting a scaling factor designated to each control input/output variable or (ii) shifting an individual membership function. Especially, dynamically focused learning (DFL) [7] was proposed to en-

hance the performance of the FMRLC. The DFL adjusts both controller input and output ranges so that learning can be effectively focused on a current operating region.

Our adaptation method also combines methods (b) and (c), but it adjusts the output membership functions (i.e., changes the values C_{ij}^{c}) in addition to the input membership functions (i.e., A_i^{l}, A_i^{r}, B_j^{l}, and B_j^{r}). The adjustment of input membership functions may provide only a limited degree of control rule modification, thereby resulting in a limited improvement in performance. However, by also adjusting the output membership functions, which directly affect a control action, we can modify the control rules more comprehensively and improve the performance significantly. All these adjustments are made automatically according to (i) a performance index (PI) that we have introduced and (ii) a comparison between each activated control action and the overall control action. The PI (denoted by $p(k)$ at time step k) is defined as an indicator for degree of rate of error change due to the previously activated control rules and it measures the effectiveness of the last control action on the error reduction. The comparison evaluates the contribution of each activated control rule in the last action to the overall control action. Our proposed strategy, which includes real-time calculation of a PI and evaluation of an activated control rule and adjusts both input and output membership functions, can efficiently adapt the fuzzy controller to changing operating conditions and to time-varying processes. Further, these advantages are achieved with less computation load and less data storage compared to the conventional direct rule-modification strategy (e.g., method (a) above).

The proposed adaptation mechanism that adjusts the membership functions is illustrated in the lower block of Fig. 6 and described in detail below.

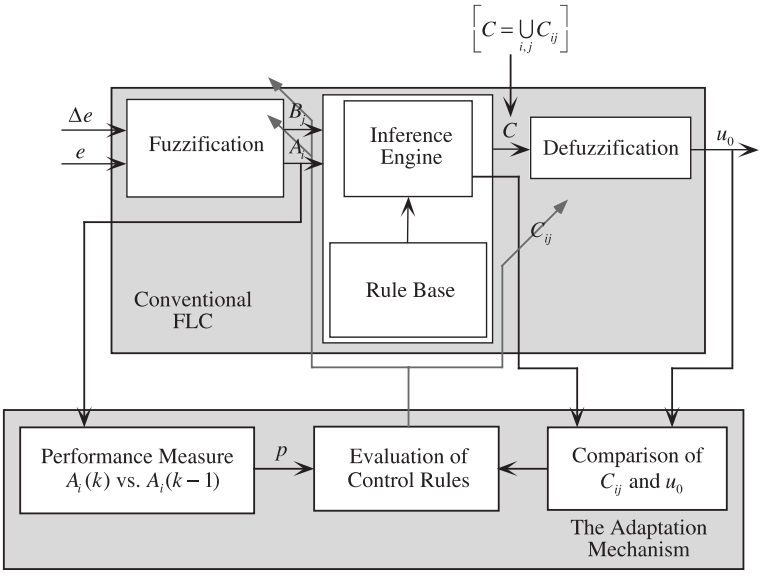

Fig. 6. Block diagram of the proposed AFLC.

4.1. Performance evaluation of control rules

As mentioned before, the proposed controller improves its own performance by self-evaluating its control rules. In order to evaluate a control rule which was activated at a former time step, $k-1$, two variables are used: the error $e(k-1)$ at step $k-1$, and the error $e(k)$ at the current step k. The controller determines how good each activated rule has changed the error between the former and current time steps. The evaluated performance $p(k)$ (at the current step k) of each control rule activated at step $k-1$, has one of five index values $\{-2,-1,0,1,2\}$ which indicate the rate of change of the error due to the activated rule: 2 represents the fastest performance and -2 represents the slowest or that the error diverges rather than converges. The PI 0 denotes that the error is tending to decrease at a moderate rate or into a desirable error range (in this case, no correction is necessary in the following step). For instance, let us consider four cases when $e(k-1)$ was PS: (i) if $e(k)$ is PL, then $p(k)$ is -2 (too slow: in fact, the error is increasing); (ii) if $e(k)$ is PS, then $p(k)$ is -1 (the error has not decreased); (iii) if $e(k)$ is NS, then $p(k)$ is 0 (adequate: the error is decreasing at a moderate rate); and (iv) if $e(k)$ is NL, then $p(k)$ is 1 (a little too fast: the error has decreased more than desired). For another example, when $e(k-1)$ was PL and if $e(k)$ is NL, then $p(k)$ corresponds to 2 (too fast: the error has decreased excessively and consequently becomes too far from the desired position in the opposite direction).

One of the innovative aspects of the proposed method is the principle that we have developed for determining the PI that we defined. The principle could be extended to similar systems having a different control rule base and/or fuzzy sets. For convenience, we defined $\mathbf{E}_i$ $(i=-3,-2,-1,0,1,2,3)$ as membership functions for the position errors, where -3 is NL, -2 is NM, -1 is NS, 0 is ZR, 1 is PS, 2 is PM, and 3 is PL. In addition, we defined $\mathbf{E}_{-3}$, $\mathbf{E}_{-1}$, $\mathbf{E}_1$ and $\mathbf{E}_3$ as odd membership functions, and $\mathbf{E}_{-2}$, $\mathbf{E}_0$ and $\mathbf{E}_2$ as even membership functions. We set outer limits for the left and right edges of the membership functions such that these points do not exceed the centers of neighboring membership functions. Then, neither the odd nor the even membership functions ever overlap each other. The principle for determining the PI are described below.

For a control rule which provides error reduction to an immediately neighboring smaller membership function, the performance is considered desirable and the index is defined as 0. First, we defined the PI between either only odd membership functions or only even membership functions. If a corresponding membership for the position error has changed between two consecutive time steps from $\mathbf{E}_{2m}$ to $\mathbf{E}_{2n}$ $(m,n=-1,0,1)$ or from $\mathbf{E}_{2m-1}$ to $\mathbf{E}_{2n-1}$ $(m,n=-1,0,1,2)$, a PI $p(k)$ can be represented by

$$p(k)=\begin{cases}\frac{1}{2}[x-y-2] & \text{for } x>0\\ \frac{1}{2}[y-x-2] & \text{for } x<0\end{cases} \tag{24}$$

where $(x,y)=(2m,2n)$ or $(2m-1,2n-1)$. The PI becomes 0 with error reduction by two membership function states which is the possible minimum change. Then, for

Table 2
The performance indices

Performance	Index	If $e(k)$ is						
		$\mathbf{E}_{-3}$	$\mathbf{E}_{-2}$	$\mathbf{E}_{-1}$	$\mathbf{E}_0$	$\mathbf{E}_1$	$\mathbf{E}_2$	$\mathbf{E}_3$
If $e(k-1)$ was	$\mathbf{E}_{-3}$	−1	0	0	0	1	2	2
	$\mathbf{E}_{-2}$	−1	−1	0	0	1	1	2
	$\mathbf{E}_{-1}$	−2	−1	−1	0	0	1	1
	$\mathbf{E}_0$	*	*	*	0	*	*	*
	$\mathbf{E}_1$	1	1	0	0	−1	−1	−2
	$\mathbf{E}_2$	2	1	1	0	0	−1	−1
	$\mathbf{E}_3$	2	2	1	0	0	0	−1

the other cases when the membership functions have changed from odd to even or vice versa, the PI is given by

$$p(k) = \begin{cases} \frac{1}{2}[x - y - 1] & \text{for } x > 0 \\ \frac{1}{2}[y - x - 1] & \text{for } x < 0 \end{cases} \tag{25}$$

where $(x, y) = (2m - 1, 2n)$ or $(2m, 2n - 1)$. In these cases, error reduction by one membership function state corresponds to the PI of 0 for the same reason (i.e., possible minimal change) as above. With this index of 0 as a datum line, the magnitude of performance indices for other undesirable cases is chosen as proportional to deviation from the above desirable cases. According to Eqs. (24) and (25), the number of performance indices may need to be increased as that of the membership functions increases. Nevertheless, the basic principle described by the equations can still be applied to the expanded set of performance indices. It should be noted, however, that there is an exception to the above principle, i.e., when the error has changed from $\mathbf{E}_{-3}$ or $\mathbf{E}_3$ to $\mathbf{E}_0$ (represented by boxes in Table 2), we defined the PI as 0 instead of 1 (since the error decreased into a desirable range). The performance indices are summarized in Table 2. Note that when $e(k-1)$ was $\mathbf{E}_0$, only no change in the error (i.e., $e(k)$ is only $\mathbf{E}_0$) is desirable; for the other cases (denoted by $*$), the PI does not need to be defined nor considered, and the contribution of the corresponding control rule toward the overall control action is made relatively smaller in the following step by adjusting the membership functions.

4.2. Adjustment of the membership functions

The modification of the fuzzy input membership functions is conducted by simultaneously moving the left (A_i^{l} and B_j^{l}) and right edges (A_i^{r}, and B_j^{r}) of the membership functions, thereby making the areas covered by the membership functions narrower or wider. For the fuzzy output membership functions, the modification is performed by shifting the centroids (C_{ij}^{c}) of the output membership functions. The principle of this membership function adjustment is described below with two examples.

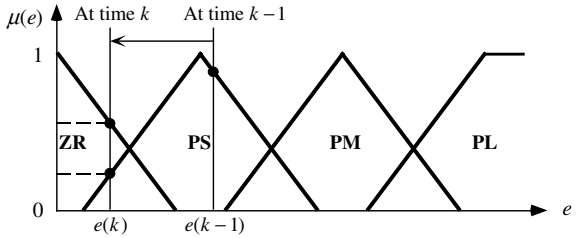

Fig. 7. Error change in Example 1.

Example 1. Let us assume that only one control rule was activated at the former step $k-1$ and it was

$$\text{If } e(k-1) = \text{PS} \quad \text{and} \quad \Delta e(k-1) = \text{PS} \text{ then } u(k-1) = \text{PM} \tag{26}$$

As a result of the corresponding control action, the error changed from $e(k-1) =$ PS to $e(k)$ that belongs to two membership functions ZR and PS (as shown in Fig. 7). Therefore, the rule in Eq. (26) has two performance indices: one PI is 0 because the error changed from $e(k-1) = \text{PS}$ to $e(k) = \text{ZR}$ (see Table 2), and the other index is -1, since the error still remains as $e(k) = \text{PS}$. However, to make a decision, the controller needs to assign only a single PI to a rule. Since $e(k) = \text{ZR}$ has a higher degree of membership than $e(k) = \text{PS}$ (i.e., closer to 1), the controller decides that the PI which represents this rule is 0. Therefore, no modification is necessary in the parameters during the next iteration.

Example 2. In the second example, the error $e(k-1)$ corresponds to two membership functions. Let us assume that the following control rules were activated at time $k-1$.

$$\begin{aligned} &\text{(a) If } e(k-1) = \text{PS} \quad \text{and} \quad \Delta e(k-1) = \text{ZR} \text{ then } u(k-1) = \text{PS} \\ &\text{(b) If } e(k-1) = \text{PM} \quad \text{and} \quad \Delta e(k-1) = \text{ZR} \text{ then } u(k-1) = \text{PM} \end{aligned} \tag{27}$$

The values corresponding to $e(k-1)$ are shown in Fig. 8(a). At time $k-1$, the controller calculates the output degree of membership for each rule (see Fig. 8(b)). These two outputs are combined to give a single overall control action. The overall control action from these rules results in an increase in the error at time k ($e(k)$) as shown in Fig. 8(a). Performance indices are assigned to each activated rule in the same way as in Example 1. For rule (a) in Eq. (27), since $e(k)$ is in both PM and PL, the PI can be either -1 (from PS to PM) or -2 (from PS to PL). However, PL has a higher degree of membership than PM (see Fig. 8(a)), and therefore the PI for rule (a) is -2. For rule (b) in Eq. (27), the same index, -1 applies to both $e(k) = \text{PM}$ and $e(k) = \text{PL}$ (see Table 2). In summary, neither of the indices of these control rules (-1 and -2) shows desirable performance, and the parameters of both rules need to be modified.

From Fig. 8(b), we can see that the control action resultant from rule (a) pulls the overall control action $u_0(k-1)$ to the left (i.e., smaller u), and therefore substantially

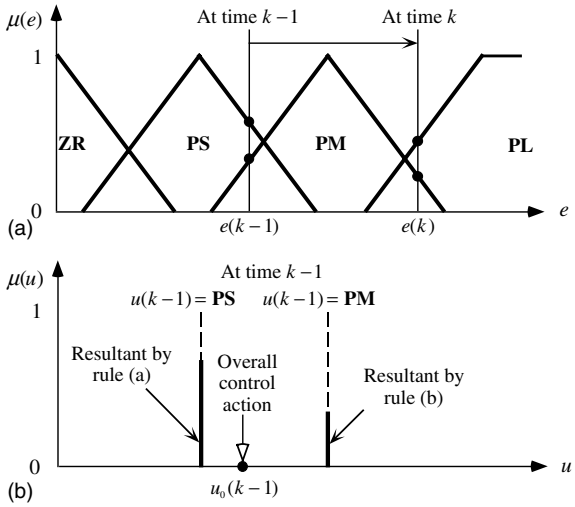

Fig. 8. Membership functions in Example 2: (a) error change; (b) fuzzy outputs.

deteriorates the overall control action (note that $u_0(k-1)$ needed to be larger to reduce the error). Since this control rule (a) suggests an action that makes a negative contribution to the overall control action, namely making it smaller, the rule is negatively reinforced by making narrower the input fuzzy sets which participated in determining this control rule (i.e., narrower membership functions for $e(k-1) = \text{PS}$ and $\Delta e(k-1) = \text{ZR}$). As for rule (b), as can be seen from Fig. 8(b), its control action is larger than the control action from rule (a), and accordingly larger than the overall control action. Therefore, the input membership functions which participate in rule (b) (i.e., $e(k-1) = \text{PM}$ and $\Delta e(k-1) = \text{ZR}$) need to be modified so that the resulting control action PM can have a higher degree of membership, therefore making a bigger contribution to the overall control action, and consequently enabling a larger u_0. For this reason, both input membership functions (PM for $e(k-1)$ and ZR for $\Delta e(k-1)$) become wider. For the fuzzy output singleton of each rule, we modify only such fuzzy output singletons that have a negative effect on the overall control action compared with other fuzzy output singletons activated together. Therefore, the fuzzy output singleton from rule (a) (i.e., $u(k-1) = \text{PS}$) is shifted to the right to have a larger value of u, and the output singleton from rule (b) (i.e., $u(k-1) = \text{PM}$) is not moved. To summarize the example, based on (i) the PI and (ii) the comparison between a single control action from an activated control rule and the overall control action, we can determine if a certain control rule suggests an action that makes a positive or negative contribution to the overall control action.

In general, the degree of modification (i.e., the magnitude of the movement of the left and right edges for the input membership functions, and of the shifting of the fuzzy output singletons) can be determined from the PI. In other words, if the PI for a control rule is ± 1, the left and right edges for corresponding input membership functions are moved by 1 step, and the corresponding fuzzy output is also shifted by

1 step. If the PI is ±2, the above parameters are adjusted by 2 steps. If the PI is 0, no modification is needed.

The AFLC main algorithm is summarized in Appendix B. In the algorithm, the constants K_A, K_B and K_C denote the magnitudes of adjustment steps for the membership functions of error and the change in the error, and for the fuzzy output singletons, respectively.

4.3. Low-velocity friction compensation

In the adaptation algorithm, we have also included a part of an estimated friction model [1]. That is, we excluded the Coulomb friction component (which is constant with respect to velocity) of the friction model, and considered only the remainder of the model. In other words, only the nonlinear component of the friction model, due to the effect of start and negative viscous friction, was used for friction compensation. To avoid the contour errors due to the nonlinear friction effect, we have defined a low-velocity range (under 12 mm/s on our machine) where the friction values are high with large start friction and negative viscous friction. For the low-velocity range, we defined the effect as a function of velocity V (in mm/s):

$$\begin{aligned}
&\text{For the } X\text{-axis:}\\
&u_d(V) = \begin{cases} 0.05V^2 - 1.16V + 6.54 & \text{for } 0 \leqslant V < 12 \\ -0.04V^2 - 1.00V - 6.43 & \text{for } -12 \leqslant V < 0 \end{cases}\\
&\text{For the } Y\text{-axis:}\\
&u_d(V) = \begin{cases} 0.04V^2 - 1.12V + 7.23 & \text{for } 0 \leqslant V < 12 \\ -0.02V^2 - 0.68V - 5.04 & \text{for } -12 \leqslant V < 0 \end{cases}
\end{aligned} \tag{28}$$

We adjusted the output membership functions using a velocity signal in a feedforward manner. In other words, u_d is added to each fuzzy output singleton $C_{ij}^c(k)$, which is already adjusted by the proposed adaptation mechanism, as represented by

$$C_{ij}^c(k) = C_{ij}^c(k) + u_d(V) \tag{29}$$

This results in adding a compensation signal u_d to the overall control action u_0. Consequently, the fuzzy output membership functions are tuned according to the performance measure and the velocity.

As an extension of the above adjustment, we also added the following conditions into the algorithm to compensate for stiction. These conditions enable the compensation to be activated only while the system has an actual feedback and issues a control command.

$$\begin{aligned}
&\text{If } (V \approx 0 \text{ and } |e(k)| \geqslant 2 \text{ BLUs}) \text{ then } u_d = u_d\\
&\text{If } (V \approx 0 \text{ and } |e(k)| < 2 \text{ BLUs}) \text{ then } u_d = 0
\end{aligned} \tag{30}$$

A basic length-unit (BLU) is the resolution unit (0.01 mm in our case); in the region -1 BLU $\leqslant e \leqslant 1$ BLU the system is in open loop. The system has a feedback and may generate a motion command only when the absolute value of the error exceeds

1 BLU. Therefore, only a signal of $|e| \geqslant 2$ BLUs shows that the control system issues a motion command.

5. Evaluation of adaptive fuzzy logic control

The objectives of this section are: (i) to compare the performance of the FLC and the AFLC, and (ii) to study the effect of adaptation on the AFLC performance. We performed both simulations and actual contour tracking experiments on a 3-hp CNC milling machine using the proposed FLC and AFLC algorithms. This machine is controlled by a general purpose computer (a 33 MHz 80486-based PC), thereby enabling us to implement various interpolation and control software. The control computer is interfaced with linear encoders and a digital-to-pulse width modulation (PWM) converter through a quadrature decoder board and a digital I/O board, respectively. The linear encoders are attached on each axis for the table position feedback to the controller, and the digital-to-PWM converter generates a corresponding 5-V PWM signal from an 8-bit digital control command for each axis. The 5-V PWM signal for each axis is amplified through a power amplifier on the machine and sent to each DC servo-motor. Fig. 9 shows a schematic diagram of the proposed

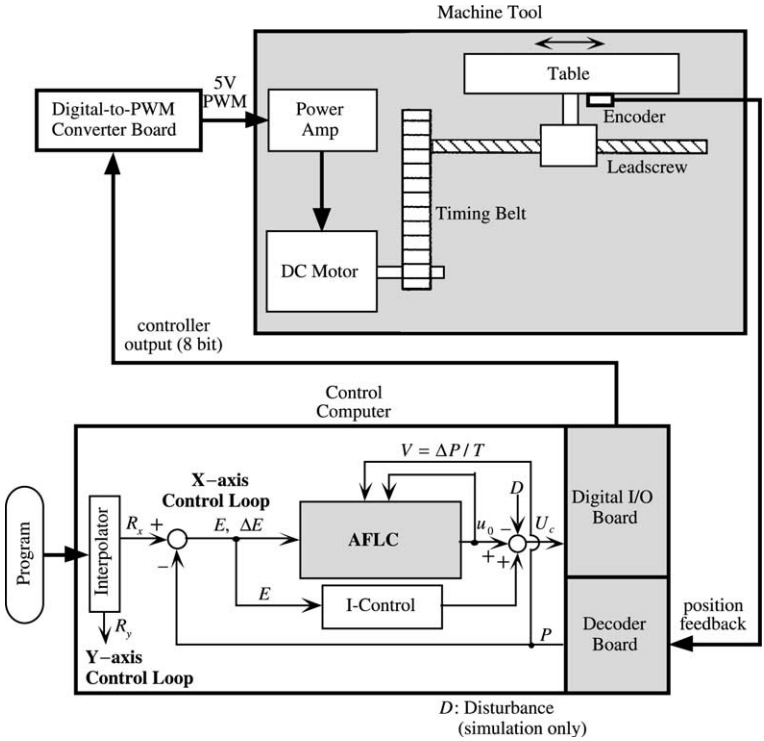

Fig. 9. Overall structure of the experimental testbed.

control system for one axis. The controls were implemented for two axes. In Fig. 9, R is a position reference input. P is a position measurement from an encoder. V is a velocity feedback calculated from dividing a position increment ΔP between two consecutive time steps by the controller sampling time T (10 ms in our system). In addition to the main AFLC algorithm, we combined an integral (I) control action to improve the steady-state tracking accuracy. U_c represents the combined control command to the motor. The AFLC, FLC and a conventional PID controllers were implemented on the milling machine.

In the proposed AFLC, the axial position error, the change in the error, the velocity feedback and the fuzzy controller output were used as inputs to the adaptation algorithm of the AFLC (see Fig. 9). The fuzzy input membership function boundaries (i.e., left and right edges) for the error were moved inward or outward by steps of 0.5 BLU (i.e., $K_A = 0.5$) while those for the change in the error were moved by steps of 0.125 BLU (i.e., $K_B = 0.125$) in proportion to the magnitude of the PI (see Appendix B). Likewise, for the fuzzy output membership functions, the singletons were shifted by one unit (i.e., $K_C = 1$) of control command to the power amplifier. In addition, the singletons were also adjusted based on the estimation of friction effect [1] in the experiments. It should also be noted that in order to avoid unstable system behavior, we added constraints on the range of movement of the controller parameters based on the stability analysis in Section 3. That is, the parameters were adjusted only within the stable range derived from the stability analysis.

The initial parameter values for the membership functions in the fuzzy logic controller were set to be the same for both the conventional FLC (without the adaptation mechanism) and the AFLC, and they were intentionally not well-tuned. We used the same initial parameter values for input membership functions in both simulations and experiments.

5.1. Simulation analyses

In order to investigate the effect of our adaptation mechanism, we performed simulations of the AFLC and the FLC under the same conditions. To make the simulation more realistic, we estimated friction values on the real machine [1] and added these values to the simulated machine as a disturbance to the system.

5.1.1. Effect of adaptation

Fig. 10 shows the simulation results when the machine performs a counter-clockwise circular motion with a radius of 40 mm, and a feedrate of 0.754 m/min. To investigate the pure effect of the proposed adaptation mechanism, we set the integral control gain to zero in this simulation. One full circular motion (namely, one cycle) takes 20 s. During this period, the AFLC adjusts itself to the environment and self-tunes the controller parameters. If another circle is needed, the performance would be better. In order to show the effect of adaptation, we simulated a motion of three successive circles. In Fig. 10(a), the *axial position errors* during the first three cycles (i.e., from 0 to 60 s) in the X-direction are presented; a similar result was obtained for the Y-direction. At time 0, both the AFLC and the FLC have the same values for the

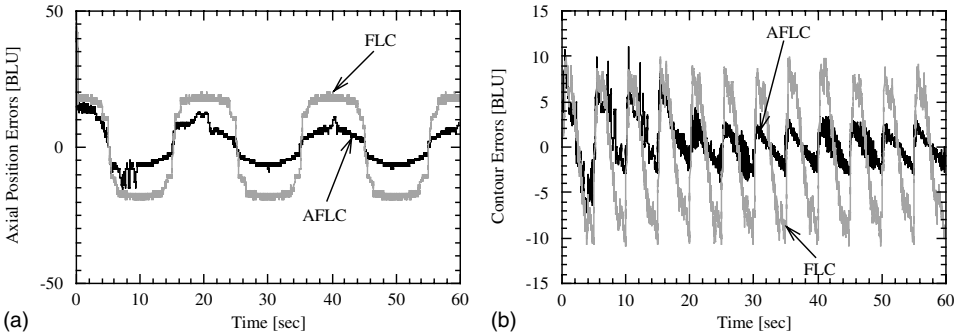

Fig. 10. Simulation comparison of (a) axial position errors and (b) contour errors for a circular contour (radius = 40 mm; feedrate = 0.754 m/min).

controller parameters. With the AFLC, the error was reduced as the motion cycle advanced, and the root mean square (RMS) error was reduced from 9.6 BLUs during the first cycle to 5.6 BLUs at the third cycle (i.e., from 40 to 60 s). Note that for the regular FLC the RMS error remained almost the same (approximately 16.2 BLUs). This shows the effect of adaptation in the AFLC algorithm.

A comparison of the *contour errors* shows that the adaptation mechanism effectively reduced the contour error after 15 s (corresponding to a three-quarter circle during the first cycle). In Fig. 10(b) we have compared the contour errors for the three cycles, and a big difference may be seen in the performance of the two controllers. In conclusion, with the adaptation mechanism, the RMS contour error is reduced after the first cycle and stays at a level lower by a factor of 4 compared to the contour error obtained with the conventional FLC. When we continued to operate with the tuned parameters, the contour error remained at the same magnitude as in the third cycle.

In Fig. 11, we have represented the simulation results for a straight line motion where the feedrate was 0.6 m/min and the trajectory had an angle of 26.6° with the

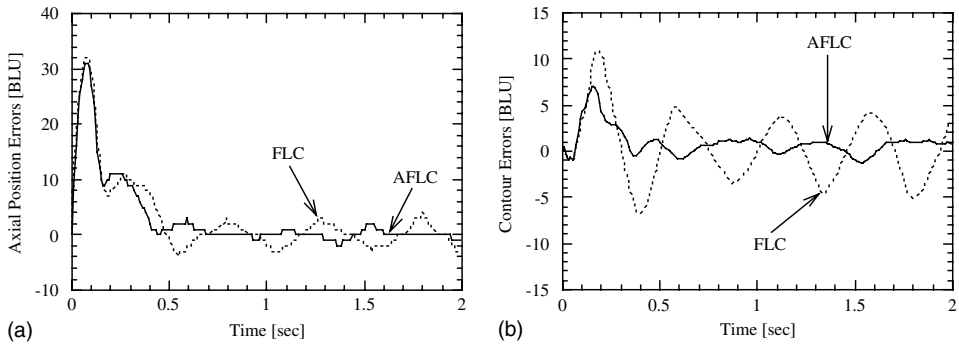

Fig. 11. Simulation comparison of (a) axial position errors and (b) contour errors for a linear contour (feedrate = 0.6 m/min).

X-axis. In this simulation, the integral control gain was set to 8.0 after the AFLC was designed. This gain was adequately chosen to eliminate the steady-state position error and to reduce the maximum contour error caused by friction disturbances, but in such a way that it does not deteriorate system stability. With the FLC the axial position error did not converge, but with the AFLC the error converged to zero steady-state error. The difference between the FLC and the AFLC is even more impressive when observing the contour error. With the FLC the contour error continues to oscillate at an amplitude of ±5 BLUs, whereas the contour error was reduced to almost zero with the AFLC. Note that an error amplitude of ±1.5 BLUs is the best that a controller can achieve.

5.1.2. Friction disturbance rejection

In order to have a baseline for the investigation of the effect of friction disturbances on the AFLC, we simulated the AFLC for a different circular contour (with a radius of 20 mm and a lower feedrate of 0.377 m/min to increase the effect of friction disturbances), with and without the disturbances in the simulator program. We compared the contour errors of both cases in Fig. 12(a) after the controller parameters were tuned respectively for each case. With the disturbances, the RMS contour error increased slightly from 1.1 to 1.5 BLUs with the disturbances. This difference, however, is much smaller, compared with the contour error increase when using a conventional PID controller in the presence of friction disturbances: we also performed the simulation with the PID controller under the same conditions as above and depicted the results in Fig. 12(b). The controller gains were derived based on the pole placement method, and they were tuned to provide a fast response while guaranteeing small overshoot. The proportional, integral and derivative gains were 1.5, 8.1 and 0.1, respectively. Without the disturbances, the PID controller also shows good contouring accuracy with an RMS contour error of 1.5 BLUs. The RMS contour error with the disturbances, however, increased to 3.4 BLUs, and the maximum contour error due to stiction repeated at every 90° was considerable and up to 15.6 BLUs. The conclusion is that the proposed AFLC has an excellent disturbance rejection performance.

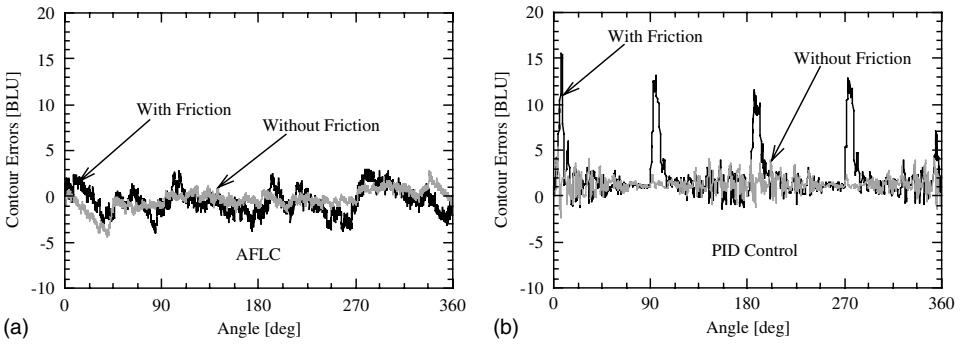

Fig. 12. Simulation comparison of contour errors with and without the friction disturbances: (a) proposed AFLC; (b) PID control.

5.2. Experimental tests

In the experimental system, in addition to the adaptation strategy as in the simulation analyses, we have also included a low-velocity friction compensation strategy as mentioned in Section 4. Therefore, for the fuzzy output singletons, in addition to adjusting them based on performance, we have also adjusted them according to the estimation of the friction effect.

In order to investigate the effect of our adaptation method, we implemented the AFLC and the FLC and compared their contour errors. Some typical results are shown in Figs. 13 and 14, where the machine moves along a linear contour $y = 10x$ with a feedrate of 0.8 m/min (Fig. 13) and a circular contour with a radius of 40 mm and a feedrate of 0.754 m/min (Fig. 14). In Fig. 14(a) and (b), we compared the

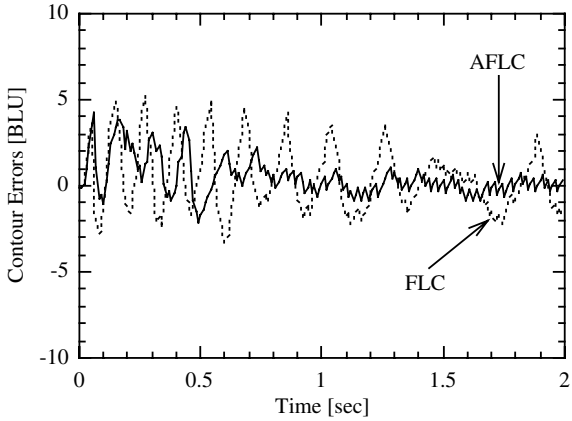

Fig. 13. Experimental comparison of linear contour errors (feedrate = 0.8 m/min).

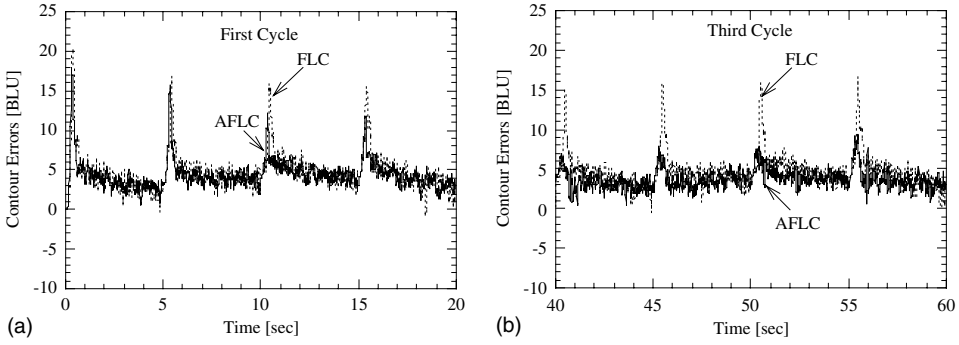

Fig. 14. Experimental comparison of circular contour errors (feedrate = 0.754 m/min): (a) first cycle; (b) third cycle.

Table 3
Initial parameter values for the membership functions

Membership functions for e	Left edge	Center	Right edge
NL	$-\infty$	–21	–16
NM	−19	−14	−9
NS	−12	−7	−2
ZR	−5	0	5
PS	2	7	12
PM	9	14	19
PL	16	21	∞
Membership functions for Δe			
NL	$-\infty$	−4.5	−3.5
NM	−4.0	−3.0	−2.0
NS	−2.5	−1.5	−0.5
ZR	−1.0	0.0	1.0
PS	0.5	1.5	2.5
PM	2.0	3.0	4.0
PL	3.5	4.5	∞
Membership functions for u	Centroid		
NL	−32		
NM	−16		
NS	−8		
ZR	0		
PS	8		
PM	16		
PL	32		

contour errors for the first and third cycles, respectively. The parameter values for the membership functions in the FLC, which were the initial parameter values for the AFLC, are listed in Table 3.

As can be seen from the figures, the adaptation mechanism substantially reduces the oscillations in the contour errors that occur with the regular FLC. The RMS value of contour errors of the above experiments is reduced by a factor of 2 as seen in Table 4. For the straight line motion, if we compare the RMS contour errors after 1 s, the error is reduced from 1.5 BLUs (with the FLC) to 0.4 BLU (with the AFLC). For the circular motion, the maximum contour error at every 90° is reduced from 14.3 to 6.4 BLUs. With the proposed AFLC, there exists some degree of oscillation in the contour errors. Although any chattering effect was not perceived during the experiments, the oscillation needs to be reduced for better surface

Table 4
Comparison of the RMS contour errors (unit: 10 μm)

Contour	AFLC	FLC
Straight line	0.9	1.7
Circle	1.9	4.0

finish. The oscillation can be reduced by increasing the number of input and output membership functions resulting in increased number of control rules which renders finer control actions. Alternatively, the reduction of the oscillation can be achieved efficiently by dynamically changing the range of input and output membership functions through adjusting scaling factors designated to the control inputs and output. This strategy of "dynamic hedging" can adjust the input and output range such that all predefined control rules are activated at any operating range, resulting in finer control actions.

In order to examine the performance of the proposed adaptation method under cutting force disturbances, we also conducted actual machining tests with the AFLC. Fig. 15 shows a typical experimental result under end milling (slotting) with an aluminum workpiece. The radius of the desired circular contour was 20 mm and the feedrate was 0.3 m/min. Depth and width of cut were 2 and 10 mm, respectively, and spindle speed was 1600 rpm. Even under the actual cutting conditions, as the cycle advanced, the controller parameters were automatically tuned by the proposed adaptation mechanism, resulting in reduction of the contour errors. In Fig. 15, we depicted the contour errors for the first cycle (Fig. 15(a)) and the third cycle (Fig. 15(b)). As seen from the figures, the maximum contour error was reduced approximately by a factor of 2 with the aid of the adaptation method.

In Fig. 16, we compared the contour errors of the AFLC (with tuned parameters through the proposed adaptation mechanism) with those of a well-tuned conventional PID controller for producing circular and linear contours, respectively. The PID controller gains were identical to those in the simulation. For circular motions, all our experiments show that the contour tracking performance of the AFLC is far better than that of the PID controller. For example, for the circular motion in Fig. 16, with the AFLC the RMS contour error was reduced by a factor of 2.6 compared to the PID controller, and the maximum error was reduced by a factor of 2.9. For straight line motions, the steady-state contour errors are not significantly different, but the initial contour errors of the PID controller due to friction are reduced considerably by the AFLC (by a factor of 3 in Fig. 16(b)). Therefore, if a contour has many short segments, the improvement achieved by the AFLC will be significant.

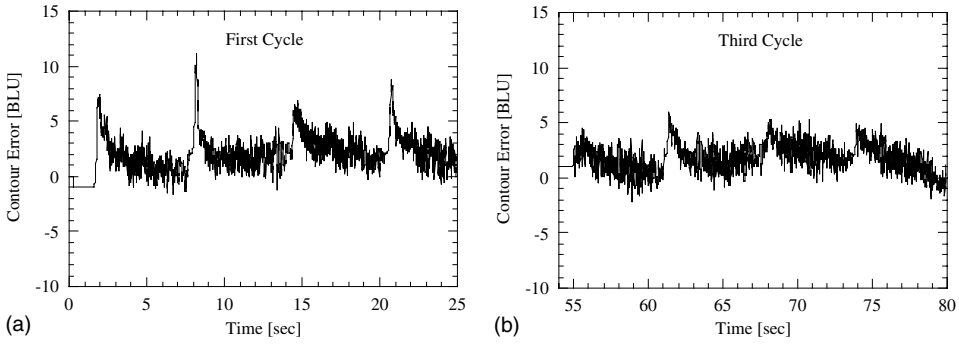

Fig. 15. Experimental contour errors for circular slot milling: (a) first cycle; (b) third cycle.

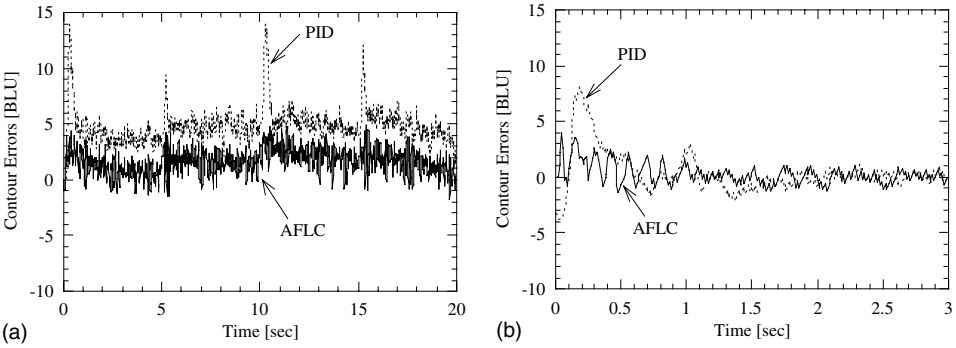

Fig. 16. Experimental comparison of contour errors: (a) circular motion; (b) linear motion.

6. Conclusions

To achieve good contouring accuracy in the presence of disturbances, such as cutting forces or friction in the feed drives of machine tools, we have developed an AFLC. The parameters for both input and output membership functions of this AFLC are tuned in real time, within a stable range, based on the performance of each control rule. In our method, this performance is checked in real time. The computation time for each axis is approximately 0.4 ms at every sampling interval (with a 33 MHz 80486-based PC), whereas 0.3 ms is required for the conventional FLC. Therefore, it appears that the additional computation load is small and the AFLC can be applied to multi-axis machines.

Through simulations and experiments we have demonstrated that the proposed AFLC can effectively adjust the controller parameters, and substantially improve the contouring accuracy compared with a conventional FLC. We have also shown that the proposed controller is robust for friction disturbances and considerably reduces the contour errors due to the disturbances. This indicates that the proposed AFLC can be a good alternative of conventional controllers, such as a PID controller, in machine tool systems especially where friction is a serious problem in the feed drives.

Appendix A. Clarification of the proposed control rules

To clarify the proposed control rules, we compared the performance of the rule base with that of two other rule bases shown in Tables 5 and 6, respectively. Table 5 shows a linear control rule base except the saturation in the controller output states. Table 6 is the classical rule base suggested by MacVicar-Whelan [16]. The differences in the three rule bases, if the controller outputs are divided into two parts (see Section 3 for details), can be considered that they have different nonlinear derivative gains while having the same nonlinear proportional gains. The different derivative

Table 5
Rule base II

Control	Actions	If $\Delta e \in$						
		NL	NM	NS	ZR	PS	PM	PL
If $e \in$	NL	NL	NL	NL	NL	NM	NS	ZR
	NM	NL	NL	NL	NM	NS	ZR	PS
	NS	NL	NL	NM	NS	ZR	PS	PM
	ZR	NL	NM	NS	ZR	PS	PM	PL
	PS	NM	NS	ZR	PS	PM	PL	PL
	PM	NS	ZR	PS	PM	PL	PL	PL
	PL	ZR	PS	PM	PL	PL	PL	PL

Table 6
Rule base III

Control	Actions	If $\Delta e \in$						
		NL	NM	NS	ZR	PS	PM	PL
If $e \in$	NL	NL	NL	NL	NL	NM	NS	ZR
	NM	NL	NL	NM	NM	NS	ZR	PS
	NS	NL	NM	NS	NS	ZR	PS	PM
	ZR	NM	NM	NS	ZR	PS	PM	PM
	PS	NM	NS	ZR	PS	PS	PM	PL
	PM	NS	ZR	PS	PM	PM	PL	PL
	PL	ZR	PS	PM	PL	PL	PL	PL

gains are depicted in Fig. 17, where E_i $(i = 0, 1, 2, 3)$ represent the centroids of the ZR, PS, PM and PL membership functions for the error, respectively. Since the rule bases are skew-symmetric (i.e., $C_{ji} = -C_{ij}$), the derivative gains are shown only for the positive range of errors. From the figure, it can be seen that overall the rule base in Table 1 has relatively lower derivative gains than the other rule bases and that the rule base in Table 6 has intermediate gains. Step responses for the rule bases are compared in Fig. 18, which shows that the rule base I outperforms the other rule bases. In addition, it appears that the rule base III gives slightly worse performance than the rule base II because the derivative gains of the rule base III are excessively low (i.e., zero) specially in the range where the error is PS or PM and the error change is PS.

Appendix B. AFLC algorithm

If a control rule activated at the former time step $k-1$ *was*

"*If* $e(k-1) = A_i(k-1)$ *and* $\Delta e(k-1) = B_j(k-1)$ *then* $u(k-1) = C_{ij}(k-1)$"
and if the resulting PI for this rule is $p(k)$*, then:*

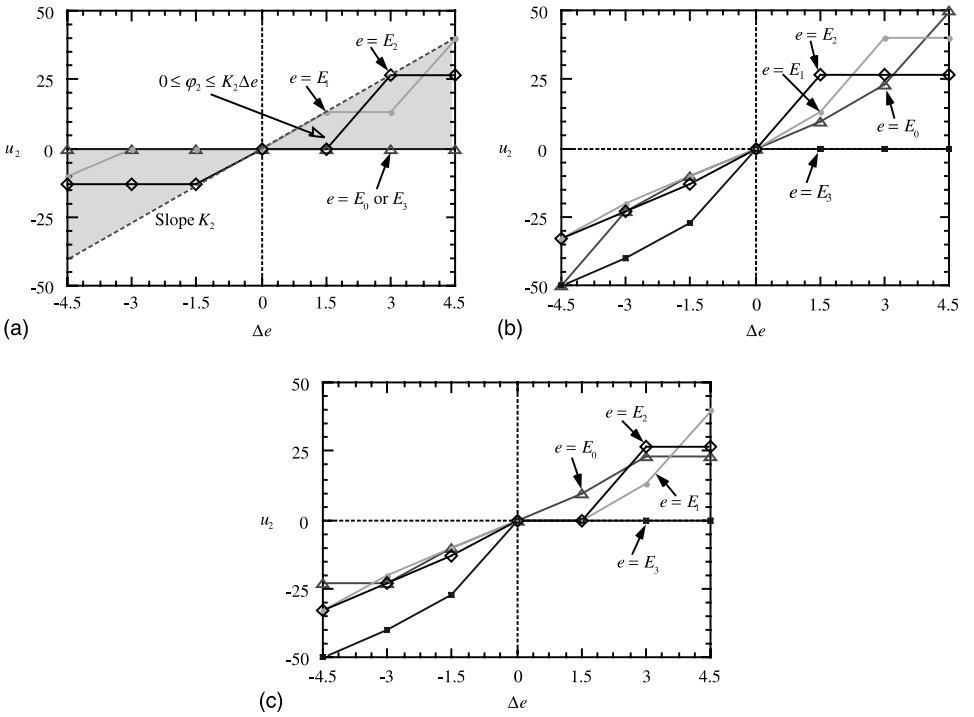

Fig. 17. Derivative gains for the rule bases: (a) Table 1; (b) Table 5; (c) Table 6.

(I) *if* $e(k-1) > 0$ *and* $p(k) > 0$, *then*
if $C_{ij}(k-1) \geqslant u_0(k-1)$ *(overall control action from all of the activated rules), then* [1]

$$A_i^{\mathrm{l}}(k) = A_i^{\mathrm{l}}(k-1) + K_{\mathrm{A}}p(k); \quad A_i^{\mathrm{r}}(k) = A_i^{\mathrm{r}}(k-1) - K_{\mathrm{A}}p(k)$$
$$B_j^{\mathrm{l}}(k) = B_j^{\mathrm{l}}(k-1) + K_{\mathrm{B}}p(k); \quad B_j^{\mathrm{r}}(k) = B_j^{\mathrm{r}}(k-1) - K_{\mathrm{B}}p(k)$$
$$C_{ij}^{\mathrm{c}}(k) = C_{ij}^{\mathrm{c}}(k-1) - K_{\mathrm{C}}p(k)$$

else

$$A_i^{\mathrm{l}}(k) = A_i^{\mathrm{l}}(k-1) - K_{\mathrm{A}}p(k); \quad A_i^{\mathrm{r}}(k) = A_i^{\mathrm{r}}(k-1) + K_{\mathrm{A}}p(k)$$
$$B_j^{\mathrm{l}}(k) = B_j^{\mathrm{l}}(k-1) - K_{\mathrm{B}}p(k); \quad B_j^{\mathrm{r}}(k) = B_j^{\mathrm{r}}(k-1) + K_{\mathrm{B}}p(k)$$
$$C_{ij}^{\mathrm{c}}(k) = C_{ij}^{\mathrm{c}}(k-1)$$

(II) *if* $e(k-1) > 0$ *and* $p(k) < 0$, *then*
if $C_{ij}(k-1) > u_0(k-1)$, *then*

$$A_i^{\mathrm{l}}(k) = A_i^{\mathrm{l}}(k-1) + K_{\mathrm{A}}p(k); \quad A_i^{\mathrm{r}}(k) = A_i^{\mathrm{r}}(k-1) - K_{\mathrm{A}}p(k)$$

[1] Indicates that the control action from this rule played a role in making the overall control action larger and consequently changed a positive error at the former time step excessively to a negative error. Therefore, the corresponding (positive) output singleton is shifted to the left to make it smaller and the corresponding input membership functions are contracted.

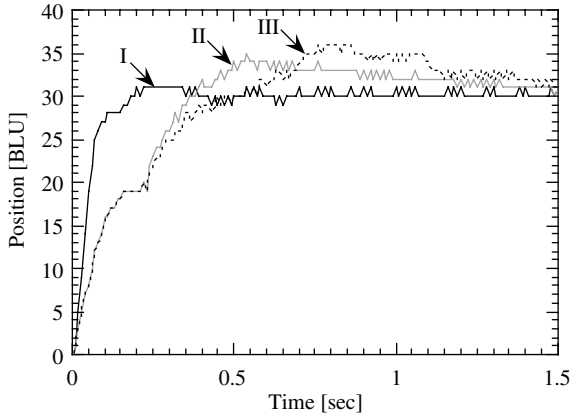

Fig. 18. Comparison of step responses for different rule bases.

$$B_j^{\mathrm{l}}(k) = B_j^{\mathrm{l}}(k-1) + K_{\mathrm{B}}p(k); \quad B_j^{\mathrm{r}}(k) = B_j^{\mathrm{r}}(k-1) - K_{\mathrm{B}}p(k)$$
$$C_{ij}^{\mathrm{c}}(k) = C_{ij}^{\mathrm{c}}(k-1)$$

else

$$A_i^{\mathrm{l}}(k) = A_i^{\mathrm{l}}(k-1) - K_{\mathrm{A}}p(k); \quad A_i^{\mathrm{r}}(k) = A_i^{\mathrm{r}}(k-1) + K_{\mathrm{A}}p(k)$$
$$B_j^{\mathrm{l}}(k) = B_j^{\mathrm{l}}(k-1) - K_{\mathrm{B}}p(k); \quad B_j^{\mathrm{r}}(k) = B_j^{\mathrm{r}}(k-1) + K_{\mathrm{B}}p(k)$$
$$C_{ij}^{\mathrm{c}}(k) = C_{ij}^{\mathrm{c}}(k-1) - K_{\mathrm{C}}p(k)$$

(III) *if* $e(k-1) < 0$ *and* $p(k) > 0$, *then*

if $C_{ij}(k-1) > u_0(k-1)$, *then*

$$A_i^{\mathrm{l}}(k) = A_i^{\mathrm{l}}(k-1) - K_{\mathrm{A}}p(k); \quad A_i^{\mathrm{r}}(k) = A_i^{\mathrm{r}}(k-1) + K_{\mathrm{A}}p(k)$$
$$B_j^{\mathrm{l}}(k) = B_j^{\mathrm{l}}(k-1) - K_{\mathrm{B}}p(k); \quad B_j^{\mathrm{r}}(k) = B_j^{\mathrm{r}}(k-1) + K_{\mathrm{B}}p(k)$$
$$C_{ij}^{\mathrm{c}}(k) = C_{ij}^{\mathrm{c}}(k-1)$$

else

$$A_i^{\mathrm{l}}(k) = A_i^{\mathrm{l}}(k-1) + K_{\mathrm{A}}p(k); \quad A_i^{\mathrm{r}}(k) = A_i^{\mathrm{r}}(k-1) - K_{\mathrm{A}}p(k)$$
$$B_j^{\mathrm{l}}(k) = B_j^{\mathrm{l}}(k-1) + K_{\mathrm{B}}p(k); \quad B_j^{\mathrm{r}}(k) = B_j^{\mathrm{r}}(k-1) - K_{\mathrm{B}}p(k)$$
$$C_{ij}^{\mathrm{c}}(k) = C_{ij}^{\mathrm{c}}(k-1) + K_{\mathrm{C}}p(k)$$

(IV) *if* $e(k-1) < 0$ *and* $p(k) < 0$, *then*

if $C_{ij}(k-1) \geqslant u_0(k-1)$, *then*

$$A_i^{\mathrm{l}}(k) = A_i^{\mathrm{l}}(k-1) - K_{\mathrm{A}}p(k); \quad A_i^{\mathrm{r}}(k) = A_i^{\mathrm{r}}(k-1) + K_{\mathrm{A}}p(k)$$
$$B_j^{\mathrm{l}}(k) = B_j^{\mathrm{l}}(k-1) - K_{\mathrm{B}}p(k); \quad B_j^{\mathrm{r}}(k) = B_j^{\mathrm{r}}(k-1) + K_{\mathrm{B}}p(k)$$
$$C_{ij}^{\mathrm{c}}(k) = C_{ij}^{\mathrm{c}}(k-1) + K_{\mathrm{C}}p(k)$$

else

$$A_i^{\mathrm{l}}(k) = A_i^{\mathrm{l}}(k-1) + K_{\mathrm{A}}p(k); \quad A_i^{\mathrm{r}}(k) = A_i^{\mathrm{r}}(k-1) - K_{\mathrm{A}}p(k)$$
$$B_j^{\mathrm{l}}(k) = B_j^{\mathrm{l}}(k-1) + K_{\mathrm{B}}p(k); \quad B_j^{\mathrm{r}}(k) = B_j^{\mathrm{r}}(k-1) - K_{\mathrm{B}}p(k)$$
$$C_{ij}^{\mathrm{c}}(k) = C_{ij}^{\mathrm{c}}(k-1)$$

References

[1] Jee S, Koren Y. Friction compensation in feed drive systems using an adaptive fuzzy logic control. In: Proceedings of the ASME Dynamic Systems and Control, vol. 2. Chicago, USA; 1994. p. 885–93.

[2] Kurakake M, Sakamoto K. Servomotor control apparatus. US Patent No. 4,916,375, April 1990.
[3] Jury EI, Lee BW. The absolute stability of systems with many nonlinearities. Automat Rem Contr 1965;26:943–61.
[4] Popov VM. Absolute stability of nonlinear systems of automatic control. Automat Rem Contr 1961;22:857–75.
[5] Bare WH, Mulholland RJ, Sofer SS. Design of a self-tuning rule based controller for a gasoline refinery catalytic reformer. IEEE Trans Auto Contr 1990;35(2):156–64.
[6] Batur C, Kasparian V. Fuzzy adaptive control. Int J Syst Sci 1993;24(2):301–14.
[7] Kwong WA, Passino KM. Dynamically focused fuzzy learning control. IEEE Trans Syst Man Cybernet 1996;26(1):53–74.
[8] Langari, G, Tomizuka M. Self organizing fuzzy linguistic control with application to arc welding. In: Proceedings of the IEEE International Workshop on Intelligent Robots and Systems. Tsuchiura-shi, Japan; 1990. p. 1007–14.
[9] Layne JR, Passino KM. Fuzzy model reference learning control. J Intell Fuzzy Syst 1996;4(1):33–47.
[10] Li HX, Gatland HB. Conventional fuzzy control and its enhancement. IEEE Trans Syst Man Cybernet 1996;26(5):791–7.
[11] Moudgal VG, Kwong WA, Passino KM, Yurkovich S. Fuzzy learning control for a flexible-link robot. IEEE Trans Fuzzy Syst 1995;3(2):199–210.
[12] Mudi RK, Pal NR. A robust self-tuning scheme for PI- and PD-type fuzzy controllers. IEEE Trans Fuzzy Syst 1999;7(1):2–16.
[13] Procyk TJ, Mamdani EH. A linguistic self-organizing process controller. Automatica 1979;15:15–30.
[14] Yurkovich S, Widjaja M. Fuzzy controller synthesis for an inverted pendulum system. IFAC Contr Eng Pract 1996;4(4):455–69.
[15] Zhao ZY, Tomizuka M, Sagara S. A fuzzy tuner for fuzzy logic controllers. In: Proceedings of the American Control Conference. Chicago, USA; 1992. p. 2268–72.
[16] MacVicar-Whelan PJ. Fuzzy sets for man-machine interactions. Int J Man-Mach Stud 1977;8:687–97.