Proceedings of the
American Control Conference
Seattle, Washington • June 1995

A SELF-ORGANIZING FUZZY LOGIC CONTROL FOR FRICTION COMPENSATION IN FEED DRIVES

Sungchul Jee and Yoram Koren

Department of Mechanical Engineering and Applied Mechanics
The University of Michigan
Ann Arbor, MI 48109

Abstract

This paper introduces a new friction compensation strategy which utilizes a rule-based fuzzy logic controller whose parameters are self-tuned according to the previous performance of the controller and a friction model in the low-velocity range. 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 for contour milling. The simulations and experiments show that the proposed self-organizing fuzzy logic controller has superior performance in terms of the contour accuracy compared with the other two controllers.

1. Introduction

In precision machining, friction in the moving components of machine tools can cause significant errors. Many efforts have been made to quantify friction and build models to compensate for friction in motor drives, robot arms and machine tools based on quantified friction models [1,4,7]. However, the model-based compensation methods have limitations since the characteristics of friction are very complex and depend on many parameters that vary during the process.

In order to address this problem, we suggest the use of a rule-based friction compensation strategy rather than a model-based approach. In this study, we adopted a self-organizing fuzzy logic control to compensate for friction in a CNC milling machine. Fuzzy logic control does not need an exact process model and is known as robust for disturbances, large uncertainty and variation in the process behavior. However, to cope with changing operating conditions and to adjust for an ill-defined control rule base, it is necessary to equip the fuzzy logic control with a self-organizing mechanism.

In this study, we have adopted a self-organizing method based on shifting and changing the shapes of the membership functions of the fuzzy controller. This strategy can efficiently adapt the fuzzy controller to changing set points and time-varying processes with a small computation load. Since both changing the shape of a membership function and shifting it can correct the corresponding membership value of an element defined on the universe of discourse, it appears that this self-organizing method can also modify the control rules. In addition, in order to reduce the contour errors due to stiction, a low-velocity friction compensation strategy is included where the output membership functions were adjusted according to the estimated friction values.

2. Fuzzy Logic Control

There are three main types of fuzzy logic controllers which are commonly used:

$$
\begin{aligned}
u &= F(e) \\
\Delta u &= F(e, \Delta e) \\
u &= F(e, \Delta e)
\end{aligned}
\qquad (1)
$$

where e is an error, Δe is the change of the error, u is a control command, and Δu is an increment in the control command. The nonlinear functional relation F includes the fuzzy reasoning and the defuzzification process. The first, second, and third controllers correspond to proportional (P), proportional-integral (PI), and proportional-derivative (PD) controllers, respectively. The third one can provide a faster transient response than the other controllers.

In this study, we have adopted the third type of fuzzy logic controller as shown in Figure 1. In other words, the controller inputs are the axial position errors at the current time step (e_k) and the change in these errors between the previous and current steps (Δe_k). The control action (u_k) is determined according to the error change rate and its direction as well as the magnitude of a current position error.

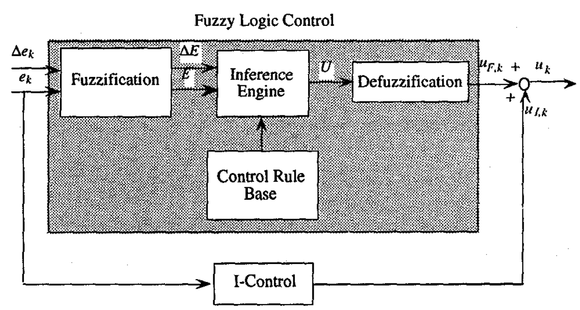


Figure 1 The controller structure.

Since in a discrete time system, a conventional PD control, with a proportional gain K_P and a derivative gain K_D, can be approximated by:

$$u_k = K_P e_k + K_D(e_k - e_{k-1}) \qquad (2)$$

this fuzzy logic controller can be regarded as a nonlinear proportional-derivative (PD) controller. We added to this fuzzy logic controller an integral (I) controller to improve the steady-state behavior. Thus, the overall control command at each time step is:

$$
\begin{aligned}
u_k &= u_{F,k} + u_{I,k} \\
&= u_{F,k} + u_{I,k-1} + K_I e_k
\end{aligned}
\qquad (3)
$$

where K_I is an integral gain, and u_F and u_I denote the fuzzy control and the integral control command, respectively. Consequently, a nonlinear type of proportional-integral-derivative (PID) fuzzy logic controller was created to compensate for friction in the machine tool feed drive system.

We defined seven fuzzy sets for each control input, and accordingly the control rule base consists of forty-nine control rules which are *if-then* conditional statements. The seven sets are:

PL :Positive Large	NS :Negative Small
PM :Positive Medium	NM :Negative Medium
PS :Positive Small	NL :Negative Large
ZR :Near Zero	

In the proposed controller, only immediately neighboring membership functions are allowed to overlap.

3. Self-Organizing Fuzzy Logic Control

The performance of a fuzzy logic controller is dependent on the pre-defined fuzzy sets or their membership functions and control rules. Thus, if the membership functions or the control rules are not defined adequately or if the controlled process behavior changes, then the established controller needs to be modified and re-tuned. Several self-organizing fuzzy logic controllers (SOFLC) have been proposed [2,3,8,10], which can be classified as follows [3]:

(a) changing a set of control rules;
(b) adjusting membership functions;
(c) changing the finite set of values describing the universe of discourse.

In this study, we have combined the methods (b) and (c) into a self-organizing method based on changing the shapes of membership functions and shifting the membership functions for the fuzzy controller inputs and output according to the performance measure. This strategy can efficiently adapt the fuzzy controller to changing set points and time-varying processes with less computation or data storage load compared to the rule-modification strategy (method (a) above). Since both changing the shape of a membership function and shifting it can correct the corresponding membership value of an element defined on the universe of discourse, it appears that this self-organizing method can also modify the control rules. However, it should be noted that the intensities or the speeds of modification are different for the two cases: shifting the membership corrects the control rules more rapidly than changing the shape does, although the extent of modification depends on the shifting amount. The proposed self-organizing mechanism is described in detail below.

First of all, in order to evaluate a control rule which was activated at the former time step k-1, two variables are used: the error e_{k-1} at step k-1, and the change in the error between step k-1 and the current step k, Δe_{k-1}. The evaluated performance of each activated control rule, p_k, has one of five values, {-2, -1, 0, 1, 2}, which indicate the change rate of the error due to the activated rule: 2 represents the fastest performance and -2 is the slowest. The performance index 0 denotes that the error is tending to decrease at a moderate rate (in this case, no correction is necessary in the following rule modification step). For instance, if e_{k-1} was *PS* and e_k is *PL*, then p_k corresponds to -2; if e_k is *PS*, then p_k is -1; if e_k is *NS*, then p_k is 0; if e_k is *NL*, then p_k is 1.

The next step is the control rule modification through the correction of membership functions for the fuzzy controller inputs and output. The principle of the control rule modification is that if a single control rule suggests an action that does not fit the overall control action, the fuzzy subsets which participated in determining that control rule are negatively reinforced (i.e., a narrower membership function). Conversely, if a control rule acts in the direction that fits the overall control action, that control rule is positively reinforced. The modification or reinforcement is conducted by making the area covered by the membership functions narrower or wider, or by simply shifting the membership functions. The modification rate or the degree of modification can be determined from the performance evaluation and the comparison of the magnitudes of the control action for each rule and the overall control action. The procedure can be summarized as follows. Here, the error and the change of the error may be intermediate between two of seven fuzzy sets, and at most four control rules are activated simultaneously. The shifted amount of the respective control action, ΔC_k has a positive sign, and the shifted amount and the contracted/expanded amount are determined based on the performance index value, D_k.

```
If e_{k-1} is A_{i,k-1} and de_{k-1} is B_{j,k-1} and u_{k-1} is C_{ij,k-1} and
  p_k is D_k, then
    if e_{k-1} > 0 and D_k > 0, then
      if C_{ij,k-1} > u_{F,k-1} (overall control action), then
          C_{ij,k} = C_{ij,k-1} - ΔC_k
          A_{i,k-1} and B_{j,k-1} are contracted
      else
          C_{ij,k} = C_{ij,k-1}
          A_{i,k-1} and B_{j,k-1} are expanded
    else if e_{k-1} > 0 and D_k < 0, then
      if C_{ij,k-1} > u_{F,k-1}, then
          C_{ij,k} = C_{ij,k-1} + ΔC_k
          A_{i,k-1} and B_{j,k-1} are contracted
      else
          C_{ij,k} = C_{ij,k-1}
          A_{i,k-1} and B_{j,k-1} are expanded
    else if e_{k-1} < 0 and D_k > 0, then
      if C_{ij,k-1} > u_{F,k-1}, then
          C_{ij,k} = C_{ij,k-1}
          A_{i,k-1} and B_{j,k-1} are expanded
      else
          C_{ij,k} = C_{ij,k-1} + ΔC_k
          A_{i,k-1} and B_{j,k-1} are contracted
    else if e_{k-1} < 0 and D_k < 0, then
      if C_{ij,k-1} > u_{F,k-1}, then
          C_{ij,k} = C_{ij,k-1} - ΔC_k
          A_{i,k-1} and B_{j,k-1} are contracted
      else
          C_{ij,k} = C_{ij,k-1}
          A_{i,k-1} and B_{j,k-1} are expanded
where i, j = 1, 2.
```

We have also included an estimated friction model in the self-tuning algorithm. To avoid the contour errors due to stiction, we have defined a low-velocity range where the friction values are high, and adjusted the output membership functions in this range as follows. As the velocity goes to zero, the centroids of the membership functions are shifted to have larger values. Consequently, the output membership functions are tuned according to the performance measure and the velocity feedback.

4. Simulation and Experimental Analyses

We have performed simulations and actual contour tracking experiments using the SOFLC. Figure 2 shows the schematic diagram of the control system for one axis. The compensation control was implemented for two-axis motions, but only one is shown in the figure. The system parameters used in the simulations are similar to those in the real experimental system. The experiments have been performed on a 3-hp CNC milling machine. The positions are measured with linear encoders, and the difference between the reference and the position feedback is the position error. The position error, the change of the error, the velocity feedback and the fuzzy control command are the inputs to the fuzzy logic controller, and the controller output is sent to the amplifier, which drives the motor.

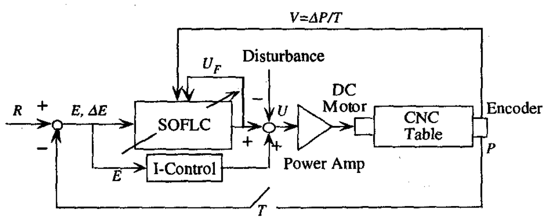


Figure 2 The overall control system.

4.1 Simulation Analyses

In the simulations, the axial position error, the change in the error and the fuzzy controller output were used in tuning the controller (the velocity feedback was not utilized). Estimated friction values [5] were used in the simulation as a disturbance to the system. The membership function boundaries for the error were expanded or contracted by 0.5 or 1.0 unit while those for the change in the error were expanded or contracted by 0.2 or 0.4 unit according to the performance index.

Effect of Self-Organizing: The simulation results when the CNC performs a circular motion in the X-Y plane are shown in Figure 3 and 4, where the radius of the circle was 40 mm, and the feedrate was 0.754 m/min. In Figure 3, the axial position errors for the third cycle (1 cycle corresponds to 20 seconds) in the X-direction are represented, and a similar result was obtained for the Y-direction. The error was reduced as the motion cycle advanced, and at the third cycle the error came to settle within a reasonable range. With this SOFLC, the root mean square (RMS) error was reduced from 3.31 to 1.82 BLU for 3 cycles while without the self-tuning the RMS error remained almost the same (i.e., 6.36 and 6.08 BLU for the first and the third cycles, respectively).

In Figure 4, we have compared the contour errors for the third cycle with the SOFLC and the conventional FLC at a feedrate of 0.754 m/min. The self-organizing mechanism effectively reduced the contour error after 15 seconds (during the first cycle), which corresponds to a 3-quarter circle, and the quadrant glitches were considerably reduced. While the RMS contour errors of the FLC were 1.89, 1.93 and 1.87 BLU for the three cycles, those of the SOFLC were 1.39, 1.24 and 1.23 BLU, respectively. We may conclude that with the self-organizing mechanism, the RMS contour error is reduced after the first cycle, and stays at a level 50 to 60% lower than the contour error achieved with the conventional FLC.

For a different feedrate, 0.377 m/min, which is adequate for an aluminum cutting, the RMS contour error of the SOFLC was still lower than that of the FLC, but the improvement is smaller as shown in Table 1. From the comparison of the results for the two feedrates, it appears that *the SOFLC is not sensitive to the feedrate,* while the conventional FLC causes larger contour errors with higher feedrates.

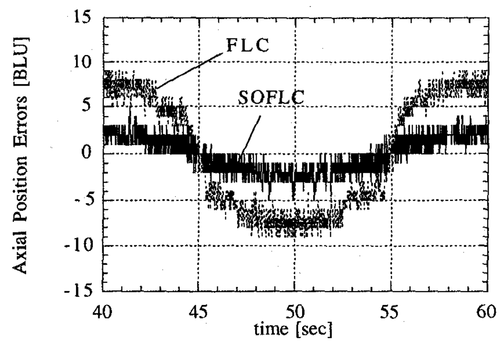
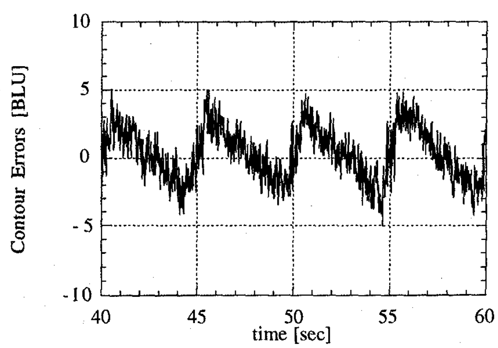
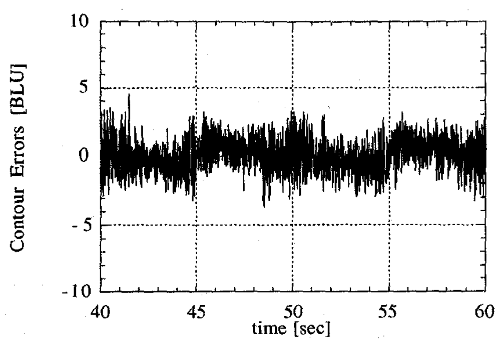


Figure 3 Comparison of the axial position errors with the SOFLC and the FLC.



(a) With the FLC



(b) With the SOFLC

Figure 4 Comparison of the contour errors with the SOFLC and the FLC (feedrate = 0.754 m/min).

Table 1 Comparison of the RMS contour errors for low feedrate (0.377 m/min). unit = 1 BLU = 10 μm

The Cycle Number	SOFLC	Conventional FLC
1	1.35	1.60
2	1.45	1.56
3	1.44	1.59

Friction Disturbance Rejection: In order to investigate the effect of friction disturbance on the SOFLC, we performed the same simulations of circular motions as above, with and without the disturbance, and compared the contour errors for the three cycles. As shown in Table 2, for the first cycle (more precisely for the first three-quarter circle), the contour error with the disturbance was larger than that without the disturbance, because the self-tuning mechanism was being shaped during that period. However, for the second and the third cycles, the RMS contour errors with the disturbance were no longer larger than that without the disturbance. In addition, for the lower feedrate (0.377 m/min), the simulation results are summarized in Table 3, and it appears that the contour errors with the disturbance are slightly larger than the contour errors without the disturbance. However, compared with the contour error increase when using a conventional PID control in the presence of friction disturbance [5], these differences are significantly small. Therefore, it is evident that this SOFLC has good disturbance rejection performance and it can be applied to the friction compensation control.

Table 2 Comparison of the RMS contour errors for the higher feedrate (0.754 m/min). unit = 1 BLU= 10 μm

The Cycle Number	With Disturbance	Without Disturbance
1	1.39	1.49
2	1.24	1.50
3	1.23	1.50

Table 3 Comparison of the RMS contour errors for the lower feedrate (0.377 m/min). unit = 1 BLU = 10 μm

The Cycle Number	With Disturbance	Without Disturbance
1	1.35	1.06
2	1.45	1.21
3	1.44	1.21

4.2 Experimental Tests

The same self-tuning algorithm used in the simulations was executed in these experiments, and the adaptation rate was set to the same value as in the simulations (i.e., 0.5 or 1.0 unit for the membership functions of the error and 0.2 or 0.4 unit for those of the change in the errors).

When we tuned the output membership functions according to the self-tuning used in the simulations, oscillatory motions occurred, although there were no oscillations in the simulation analyses. This difference can be explained by a discrepancy between the modeled system and the real system. For example, there are unmodeled dynamics such as the rubber belt which transfers the motor power to the leadscrew on our machine.

Therefore, in the experimental system, instead of adjusting the output membership functions based on performance, we have adjusted them based on the friction values. We used the estimated friction model for the X- and Y-axes in the adjustments particularly for the low velocity range (lower than 12 mm/sec), because in this range large friction values and negative viscous friction characteristics exist. For this velocity range, we adjusted the output membership functions, i.e., added compensation signals which correspond to the estimated friction values, according to the velocity which is available from either the tachometer reading or two consecutive encoder readings. In order to compensate for stiction, we have used the following strategy:

$$\text{If } (V \approx 0 \text{ and } |E| > 2 \text{ BLU}) \text{ then } U_c = F_{ds}$$
$$\text{If } (V \approx 0 \text{ and } |E| < 2 \text{ BLU}) \text{ then } U_c = 0.$$

Here, F_{ds} represents the static friction value which should be overcome to start a motion. A BLU (i.e., the resolution unit) corresponds to 0.01 mm in our system, and $|E| > 2$ BLU shows that the control system issues a motion command. We chose 2 BLU instead of 1 BLU because 1 BLU corresponds to a noise level and cannot be used as an indicator of the motion command.

Effect of Low-Velocity Friction Compensation: In Figure 5, we compared the contour errors, with and without friction adjustment for the output membership functions, to examine the effect of the above stiction compensation strategy. From the experimental result, it can be seen that with this compensation method the contour errors due to stiction are reduced by approximately 3 BLU.

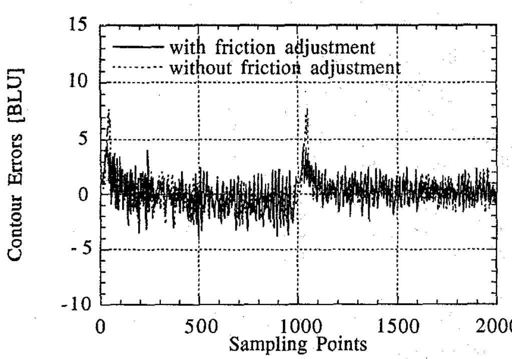


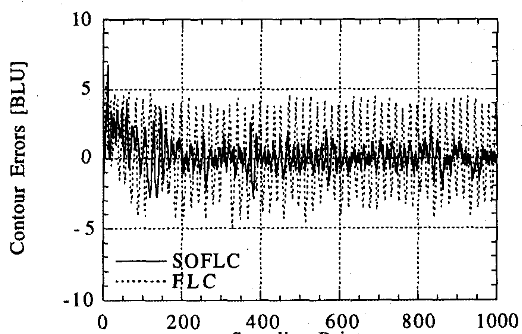
Figure 5 The effect of stiction compensation.

Effect of Self-Organizing: In Figure 6, in order to investigate the effect of the self-tuning algorithm, we compared the contour errors for the SOFLC and for a conventional FLC. This figure represents the results for a linear contour, $y = 10x$. As can be seen from the figure, the self-organizing mechanism reduces the large oscillations in the contour errors.

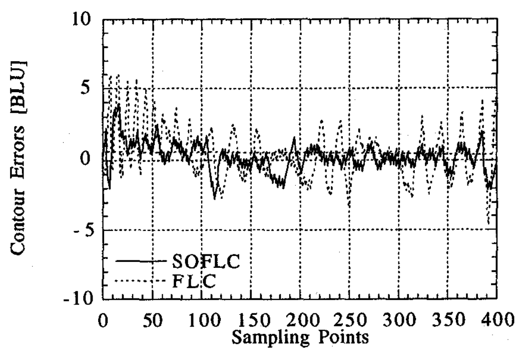
Comparison with Conventional Control: In Figure 7, we compared the contour errors of the SOFLC with those of a conventional PID control for producing a circular contour. With the SOFLC the RMS contour errors of the PID control were reduced by the ratios of 1.4 : 1 and 2.6 : 1 for the lower (0.377 m/min) and the higher feedrate (0.754 m/min), respectively. Hence, it appears that the contour tracking performance of the SOFLC is much better than that of the PID control in a high feedrate range.

5. Conclusion

Through simulation and experimental analyses we demonstrated that the self-organizing fuzzy logic control (SOFLC) is robust for friction disturbance and is not sensitive to change in feedrates. By contrast, a conventional PID control produces larger contour errors when cutting with larger feedrates. In the experiments, we have reduced the contour errors due to stiction by a low-velocity friction compensation strategy. Therefore, the SOFLC has a dramatic effect on error reduction in machine tool systems where friction is a serious problem in the feed drives. In addition, it can be applied to high-speed machining, thereby achieving small errors and high productivity.

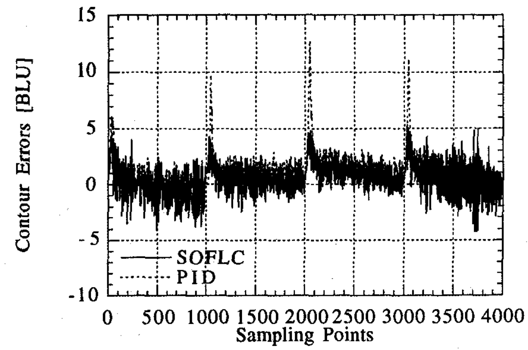


(a) For feedrate = 0.4 m/min

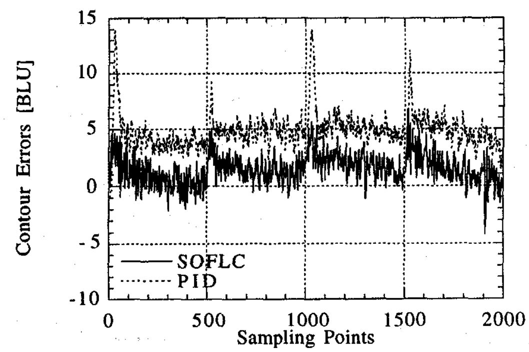


(b) For feedrate = 0.8 m/min

Figure 6 Comparison of contour errors of fuzzy logic control with and without the self-organizing mechanism for straight line motions.



(a) For feedrate = 0.377 m/min



(b) For feedrate = 0.754 m/min

Figure 7 Comparison of contour errors of the SOFLC and a PID control for circular motions.

References

[1] Armstrong, B., "Friction: Experimental Determination, Modeling and Compensation," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pp. 1422-1427, Philadelphia, PA., 1988.

[2] Bare, W. H., Mulholland, R. J., and Sofer, S. S., "Design of a Self-Tuning Rule Based Controller for a Gasoline Refinery Catalytic Reformer," *IEEE Transactions on Automatic Control*, Vol. 35, No. 2, pp. 156-164, 1990.

[3] Batur, C. and Kasparian, V., "Fuzzy Adaptive Control," *International Journal of Systems Science*, Vol. 24, No. 2, pp. 301-314, 1993.

[4] Canudas de Wit, C. and Seront, V., "Robust Adaptive Friction Compensation," *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pp. 1383-1388, Cincinnati, OH., 1990.

[5] Jee, S. and Koren, Y., "Friction Compensation in Feed Drive Systems Using an Adaptive Fuzzy Logic Control," *1994 ASME Winter Annual Meeting*, DSC-Vol. 55-2, pp. 885-893, Chicago, IL., 1994.

[6] Koren, Y., *Computer Control of Manufacturing Systems*, McGraw-Hill, 1983.

[7] Kubo, T., Anwar, G., and Tomizuka, M., "Application of Nonlinear Friction Compensation to Robot Arm," *Proceedings of the 1986 IEEE International Conference on Robotics and Automation*, pp. 722-727, San Francisco, CA., 1986.

[8] Langari, G. and Tomizuka, M. "Self Organizing Fuzzy Linguistic Control with Application to Arc Welding," *Proceedings of the 1990 IEEE Workshop on Intelligent Robots and Systems*, Tsuchiura-shi, Japan, 1990.

[9] Procyk, T. J. and Mamdani, E. H., "A Linguistic Self-Organizing Process Controller," *Automatica*, Vol. 15, pp. 15-30, 1979.

[10] Zhao, Z.-Y., Tomizuka, M., and Sagara, S., "A Fuzzy Tuner for Fuzzy Logic Controllers," *Proceedings of the 1992 American Control Conference*, pp. 2268-2272, Chicago, IL., 1992.