Real-Coded Genetic Algorithm Based Fuzzy Sliding-Mode Control Design For Precision Positioning

Pai-Yi Huang, Sinn-Cheng Lin and Yung-Yaw Chen

Lab. 202, Electrical Engineering Department, National Taiwan University, Taipei, Taiwan, R.O.C.
f81085@cctwin.ee.ntu.edu.tw

Abstract

A fuzzy sliding-mode controller was optimized through real-coded genetic algorithms and successfully implemented on an industrial XY table. The fuzzy sliding-mode controller is special type of fuzzy controller. By using the sliding surface, the fuzzy rule is simpler and the entire rule base is more compact. Thus, more easily for applying self-learning schemes. The real-coded genetic algorithm uses the internal floating-point representation of the computer system. With this advantage, the finite resolution problem of traditional genetic algorithm has been solved. The experimental results show the success of this approach.

1. Introduction

Precision positioning is very important in advanced industry. Especially for IC fabrication, the more finer the resolutions means the more circuits that can be condensed into one single chip. This also means faster, lower power and cheaper electrical product that can be produced. But to increase the precision of the positional machine is not an easy task. Friction and backlash are two nonlinear phenomena found in general mechanical systems. The backlash existed in the gears will cause nonlinear behavior while the direction reversal has happened. The friction, also referred as stick-slip friction, caused by the relative movement of the two contacting surfaces is also a big problem. The magnitude of the friction is a random variable with respect to time and also is different form place to place. In conclusion, the friction is time varying and position dependent, so it is very difficult to compensate.

There are a lot of references for compensation of the friction. Among them, direct compensation for friction is used in most of the researcher [1]. But this approach is not necessary work if the parameters of the friction model are not correctly evaluated. Some adaptive control algorithm [2] developed for solving this problem. And alternatively, There are more control schemes, such as robust control [3], fuzzy control and neural network [4].

Model-based control design strategies have been used to solve most of the automatic control problems in the classical control theory. But in the real world, there are a lot of complex industrial processes whose available models can not be easily developed. Hence, design a model-based controller to control such ill-defined systems becomes a very tough job. Skilled operators, however, can control the systems successfully without keeping any mathematical model in mind. Fuzzy logic controller (FLC), a controller with linguistic rules that works like human thinking, has become an extensively researched topic in the last three decades. By organizing human expertise into fuzzy IF-THEN rules, an FLC can be constructed to control complex or ill-defined systems like human experts do. Knowledge acquisition therefore becomes the most important task in the FLC design.

The fuzzy sliding mode controller (FSMC) is a special type of fuzzy controller. The fuzzy rule is simpler and the entire rule base is more compact, the speed of fuzzy inference of FSMC therefore is faster than that of conventional FLC. In this paper, a FSMC with automatic knowledge acquisition through Real-coded Genetic Algorithm (RGA) is applied to solve this nonlinear compensation problem. Genetic training of fuzzy membership functions gathers and compensates the nonlinear behaviors through repetitive learning process. Good parameter patterns are collected and combined step by step, To cope with real parameter changing, floating-point coding other than fix-point coding is employed to resolve the coding resolution problem. Successful experimental results also gives to support this methodology.

This paper is organized as follows. Section 2 is brief description of the plant and the motivation for this research. Next, Section 3 presents the methodology of the floating-point genetic algorithm. In Section 4 the real-coded genetic algorithm combines with fuzzy sliding mode control for self-tuning is discussed. Further, Section 5 presents the enhancements added to the XY tables, preparations for the experiment and the experimental results. Finally, Section 6 gives the conclusions.

0-7803-4863-X/98 $10.00©1998 IEEE

2. Hardware Description and Motivation

The experimental hardware, XY table, and the motivation of this research are introduced in this section.

The DC-motor-driven XY table (Figure 1) was manufactured by NSK Inc. Two low-noise linear amplifiers were used to apply the driving force. The specifications of the tables are listed in table1.

Table 1. Specifications of the experimental precision table	
Stroke	200*mm* x 200*mm*
Pitch	5*mm*
Repeatability	0.003*mm*
Backlash	0.001*mm*
Encoder Resolution	1.25μm
Class	C2

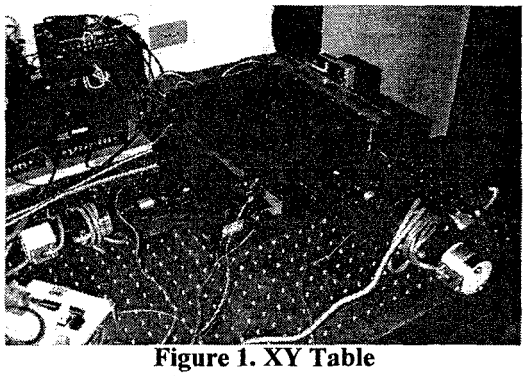

Figure 1. XY Table

The stroke defined is the available travel length of the table. The XY table had 200*mm* stroke in each axis. The definition of the 5*mm* pitch is the advancing distance in correspondent axis when the motor makes one turn. Form the certification of the manufacturer, the backlash was 1μ and the repeatability was 3μ.

A sinusoidal voltage signal was injected into the motor and the output position signal measured could be used to calculate the behavior of the friction. By labeling voltage injected as x-axis and velocity calculated as y-axis, The characteristic of the friction is now easily depicted from figure 2. Moreover, Due to imperfect alignment during manufacturing, the magnitude of the friction force may have positional dependency with cycle distance equal to one pitch. Since the reasons presented above, the friction force is hard to model very accurately.

The XY table with friction behavior as describe above is hard to control accurately. Nonlinear, position dependent and time-varying are the characteristics of friction as obtained from experimental experiences. Moreover, the more the positioning resolution increases the more difficulties to estimate the friction-model parameters. Thus, compensations for the friction are even harder. Intelligent control that will aggregate from previous experiences to optimize the control efforts is proposed. In this paper, the real-coded genetic algorithm is used for turning the fuzzy sliding mode controller (FSMC).

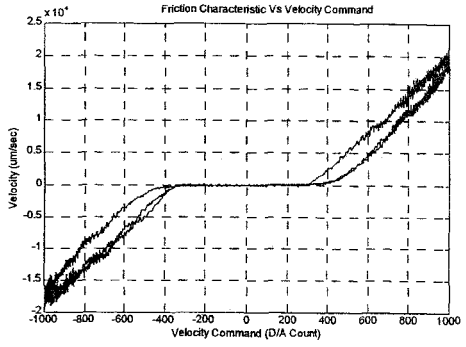

Figure 2. Friction characteristics of the table

3. Real-Coded Genetic Algorithm (RGA)

In the traditional GAs, the natural parameter set of the optimization problem needs to be coded as a finite length string. This operation maps a real quantity to a fixed length binary string. However, it will cause some information lost due to truncation. Furthermore, one must face a trade-off problem between the length of the coding string and the resolution of the parameter value. To increase the resolution, a longer binary string must be chosen and hence, will slow down the convergence rate.

RGAs [5-6] have been proposed to overcome the aforementioned problem. RGAs use real numbers for computation, and there are no encoding and decoding operations involved. Thus, theoretically, there will be no resolution problems. But in all computer systems, a real number is implemented by finite and discrete floating-point coding. In this paper, the IEEE 754 floating standard is used as the coding format [7]. Below are three important operations defined for the genetic algorithm that used in our optimization scheme? Reproduction, crossover and mutation will be discussed separately.

3.1 Reproduction

Reproduction is a process based on the principle of survival of the fittest. Individual strings are copied according to their fitness (objective) function values. The strings with a higher fitness value have a higher probability of contributing one or more offspring in the next generation. To achieve this, firstly, we must define the fitness function, which can be an appropriate nonlinear, positive definite function. Then individual fitness is calculated and normalized with the sum of all fitness values. The normalized fitness of the string is usually defined to be its survival probability in next generation.

3.2 Crossover

Crossover provides a mechanism for individual strings to exchange information via a probabilistic process. Once the reproduction operator is applied, the members in the mating pool are allowed to mate with one another. In each generation, If there are n parameters needed to be determined, we define the parents $R_i \in R^n$ and the children $R_i \in R^n, i = 1,2$. The crossover operation can be shown by following equation.

$$\begin{cases} R_1 = R_1 + k_1 \times (R_2 - R_1) \\ R_2 = R_1 + k_1 \times (R_2 - R_1) \\ -2 \le k_1, k_2 \le 1 \end{cases} \quad (1)$$

The ranges of k_1 and k_2 are determined from heuristics. Values other than -2 and 1 were tested but no significant differences were found.

3.3 Mutation

For the mutation part, the operation is more like in a binary-coded GA. At each iteration, every gene is subject to a random change with probability of the pre-assigned mutation rate. In the binary-coded case, a mutation operator changes a bit from 0 to 1 or vice versa. There is only a little difference; that is, the mutation rate should be different in different fields. In the exponent part mutation rate should be lower, because the exponent part will have strongest effect on the coded values. Sudden change of these values will cause abrupt change in the parameter fields, which will lead to loss of some established good schemata, and cause longer convergence time. The operation will not benefit search sequence and should be avoided. In the mantissa part, mutation rate could be raised to increase the ability to explore.

4. RGA based FSMC

This section proposed strategies that apply genetic algorithms to search suitable rule bases for fuzzy controllers. To utilize GAs for fuzzy rules extracting, it would encounter following problems: If the fuzzy controller is a conventional FLC, then whereas the input variables or linguistic labels increase, it will lead to exponentially increase the number of rules. Consequently, the chromosome length of GAs will increase exponentially.

An FSMC[8], which is based on sliding mode control (SMC) and fuzzy logic, has better properties and is more suitable for genetic learning than a conventional FLC. Therefore, an FSMC is used in the proposed self-learning fuzzy control system instead of an FLC.

4.1 System Description

Consider a class of nonlinear systems with the following error dynamics:

$$e^{(n)} = f(e, \dot{e}, \ldots, e^{(n-1)}) + g(e, \dot{e}, \ldots, e^{(n-1)})u \quad (2)$$

where $e \in \Re$ is the system error; $u \in \Re$ is the control input; $f(.)$ is an unknown continuous function with known upper bound, *i.e.* $|f| \le f^U$; $g(.)$ is an unknown positive definite function with known lower bound, *i.e.* $0 < g_L \le g$. Actually, equation (2) represents a general uncertain nonlinear dynamic system.

Define $x_i = e^{(i-1)}, i = 1, 2, \ldots, n$, then (2) can be represented by the following state representation:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \quad \vdots \\ \dot{x}_n = f(\underline{x}) + g(\underline{x})u \end{cases} \quad (3)$$

where $\underline{x} = [x_1\ x_2 \ldots x_n]^T \in \Re^n$ is the state vector.

4.2 Fuzzy Sliding Mode Control

The first step in both SMC and FSMC design is to define a sliding function:

$$s(x) = \underline{c}^T \underline{x} = \sum_{i=1}^{n} c_i x_i \quad (4)$$

where $\underline{c}^T = [c_1\ c_2 \ldots c_n]^T \in \Re^n$ is coefficient vector of sliding surface that has to be properly determined..

The fuzzy control law defined from the following fuzzy sliding mode control rule base:

$$R_j : IF\ s\ is\ S_j(m_j, \sigma_j)\ THEN\ u_f\ is\ U_j(\varphi_j) \quad (5)$$

where $j = 1, 2, \ldots, N$, and N is the number of rules; S_j s are the input linguistic labels, they are simply assigned as Gaussian-shaped functions here and hereafter, *i.e.*. $\mu_{S_j}(s) = \exp\left(-\left(\frac{s - m_j}{\sigma_j}\right)^2\right)$; U_j s are the output linguistic labels, especially, they are assigned to be fuzzy singleton in this paper, *i.e.*, $\mu_{U_j}(u) = \begin{cases} 1, u = \varphi_j \\ 0, u \ne \varphi_j \end{cases}$.

Suppose that the singleton fuzzification and the weighted average defuzzification methods are applied, then the control output of (5) is given by:

$$u_f(s) = \underline{\varphi}^T \underline{\rho}(s) \quad (6)$$

where $\underline{\varphi} = [\varphi_1, \varphi_2, \ldots, \varphi_N]^T$ and $\underline{\rho} = [\rho_1, \rho_2, \ldots, \rho_N]^T$ in

which $\rho_j(s) = \dfrac{\mu_{S_j}(s)}{\sum_{j=1}^{N} \mu_{S_j}(s)}$

Without loss of generality, assume that $L_i = L$ for all i. Then the number of rules in a conventional FLC with complete rule base is given by $N = L^n$. On the other hand, if the states are combined into a single variable s, the number of rules in an FSMC with complete rule base (5) is given by $N = L$ ..This fact presents that the number of fuzzy rules in an FLC will grow exponentially once the number of either input variables n or linguistic labels L increase. However, the number of rules in an FSMC is in proportion to the number of input linguistic labels L

For finding the optimal rule parameters, the RGA-based learning schemes are presented to learn the fuzzy control rule base (5). Such kinds of FSMCs are denoted as RGA-based FSMCs. To improve the performance, the genetic algorithm with elitist model was adopted. Its basic principle is to always include the most fitted individual in the population. The operations are: if the best individual generated up to generation h is not in the population of the new generation of h+1, then select one member from the new population and replace it with the elitist of the old population.

Constructing a parameter space to be searched by RGAs required transferring the fuzzy rule base (5) to a parameter representation. Assume that X_m, X_σ, X_φ are the search space of m_j s, σ_j s, φ_j s, respectively; M is the population size; h is the number of generation.

The fuzzy control law of RGA-based FSMC is given by (6) with both adjustable consequence part, $\underline{\varphi}$, and premise part, $\underline{\rho}(s)$. Both m_j s and σ_j s in the IF part and φ_j s in the THEN part of (5) are adjusted during the learning period. Hence, the parameter vector to be learned by genetic algorithm, $\underline{\theta}$, is defined as:

$$\underline{\theta} = [\underline{m}^{\mathrm{T}}\ \underline{\sigma}^{\mathrm{T}}\ \underline{\varphi}^{\mathrm{T}}]^{\mathrm{T}} = [m_1\ m_2 \cdots m_N\ \sigma_1\ \sigma_2 \cdots \sigma_N\ \varphi_1\ \varphi_2 \cdots \varphi_N]^{\mathrm{T}}$$

Every individual in a population corresponds to an FSMC candidate that has its parameters are stacked into a vector. The population size is heuristically selected to be M.

4.2 RGA-based FSMC

Basically, the learning procedure of RGA-based FSMC is described in [8]. At first, $3M$ initial parameter vectors were randomly generated,

$$\underline{m}^{(i)}(h) = [m_1^{(i)}(h)\, m_2^{(i)}(h) \cdots m_N^{(i)}(h)]^{\mathrm{T}}$$
$$\underline{\sigma}^{(i)}(h) = [\sigma_1^{(i)}(h)\, \sigma_2^{(i)}(h) \cdots \sigma_N^{(i)}(h)]^{\mathrm{T}}$$
$$\underline{\varphi}^{(i)}(h) = [\varphi_1^{(i)}(h)\, \varphi_2^{(i)}(h) \cdots \varphi_N^{(i)}(h)]^{\mathrm{T}}$$

where

$m_j^{(i)}(h) \in X_m$, $\sigma_j^{(i)}(h) \in X_\sigma$ and $\varphi_j^{(i)}(h) \in X_\varphi$ (i = 1, 2, ..., M, j = 1, 2, ..., N).

The i-th candidate of RGA-based FSMC is denoted by RGA-FSMC$^{(i)}$.The fuzzy controller became

$$\begin{cases} R_1^{(i)}: \text{IF } s \text{ is } S_1^{(i)}(m_1^{(i)}, \sigma_1^{(i)}) \text{ THEN } u_f \text{ is } U_1^{(i)}(\varphi_1^{(i)}) \\ R_2^{(i)}: \text{IF } s \text{ is } S_2^{(i)}(m_2^{(i)}, \sigma_2^{(i)}) \text{ THEN } u_f \text{ is } U_2^{(i)}(\varphi_2^{(i)}) \\ \vdots \\ R_N^{(i)}: \text{IF } s \text{ is } S_N^{(i)}(m_N^{(i)}, \sigma_N^{(i)}) \text{ THEN } u_f \text{ is } U_N^{(i)}(\varphi_N^{(i)}) \end{cases}$$

$S_j^{(i)}$ s and $U_j^{(i)}$ s are unknown linguistic labels: $m_j^{(i)}$ s, $\sigma_j^{(i)}$ s and $\varphi_j^{(i)}$ s are adjustable parameters that manipulated by RGA.

The RGA could be used for evaluating and optimizing the parameters after that. The fitness function, F, is defined in the following intuition: a controller which drives the state to reach the sliding surface as fast as possible without consuming too much control energy and then keep the state onto the surface as close as possible gets a higher score, *i.e.*

$$J_s = \sum_{k=1}^{K} t_s \cdot \left(w\|s(k)\| + v\|u(k)\|\right) \qquad (7a)$$

$$F = 1/\left(J_s + \varepsilon_0\right) \qquad (7b)$$

Where t_s denotes the reach time of sliding mode; k denotes the iteration instance. Finally, ε_0 is a small positive constant used to avoid the numerical error of divided by zero.

The genetic operators, *i.e.* reproduction, crossover and mutation, then were to applied to generate a new population $\boldsymbol{P}(h+1)$, which is always known as the offspring of $\boldsymbol{P}(h)$. The elitistism was used for preserving the best-fitted individuals in each generation. The experiment was then conducted for every of them to calculate their fitness values.

The stop criticism for the experiment was depending on users. The best parameters for controller could never be discovered, but the optimized parameters with better performance could be found. The users should decide when to stop, at once an ideal solution had been found.

5.Experimental Setup and Results

Before conducting experiment, the apparatus, XY table, had been enhanced. The 1.25μm encoder of the motor as in Table 1 was replaced by a 0.1μm laser scale.

Limit switches was installed at table movement extremes to detect any abnormal actions. The signal connected to an electrical protecting circuit that would turn off system power when extreme conditions happened. Also, an automatic computer program was developed to handle the data gathered. Detailed descriptions of them are discussed separately in the following.

5.1 Position Sensor (Laser Scale)

High precision control depends on high-resolution sensors attached on the object plant. It is a very important part of the precision system. Stable and reliable position sensors are recommended. Many precision system use laser as the position sensing device, because laser light has good stability for its' wavelength and this could be used as a reference base.

The XY table has encoder attached to the dc motors. But this position is not the real position of the moving table. Because, the table is not a rigid body, so there must exist some dynamics between the rotating shaft and the plate of the moving table. To take this dynamics into consideration, a direct measurement that translates the movement of the moving table is required. To accomplish this, laser scale is mounted on the table. As shown in Figure 1, the XY table is equipped with a 0.1 μ m laser scale manufactured by FUTABA on both axes. With the help of these measurement devices, the resolution of the tables has been upgraded to sub-micron level. Table2 is the specifications of the laser scales used in XY table.

Table 2. Specifications of Laser Scales	
Manufacturer	FUTABA
Laser Scale Type	FM20WA
Readout Length	200mm
Maximum Response Speed	40mm/sec
Resolution	0.1 μm

5.2 Protection Circuit

In order to prevent any damage of these high precision devices, a special designed logic circuit and limit switches are added at both ends of the tables. When the moving plate reaches the ends on both sides of the table, the limit switch installed will cut off the power and stop the dc motors for further movements and this will prevent extra damages to the tables. Furthermore, an emergency stop button is added for operator to stop the whole system at any time in the learning process.

These devices act as a very important role during experiments. In the development period of the prototype controller, improperly designed controller often leads to an unstable system. In this case, permanent damages will occur if the weak parts in the system have not been protected in advance.

5.3 Computer Software and Hardware

A 486 IBM Compatible Personal Computer, equipped with an interface card to received the position feedback, and a D/A converter to send control command to the dc servo, is used to implement the core of the controller. The control program is implemented as an interrupt that been triggered by a timer. In every sample period, the interrupt subroutine will read the position from the laser scale, calculate the error and evaluate the fitness for the output control command, then the signal is send to D / A converter to control the dc servo. The error of the position feedback in each time step will be recorded for the evaluation of the fitness value.

Figure 3 is the computer program structure that has been implemented. Table3 below are some specification that used in the RGA.

Table 3. The parameters of RGA	
Population Size	20
Crossover Rate	1.0
Mutation Rate (Exponent Part)	0.001
Mutation Rate (Mantissa Part)	0.01

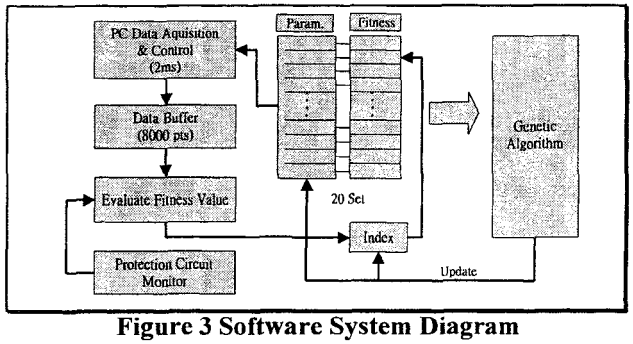

Figure 3 Software System Diagram

The experimental results are shown in Figure 4 to Figure 7. The fitness function, Figure 4, indicates the RGA-based fuzzy controller performs better and better from generation to generation.

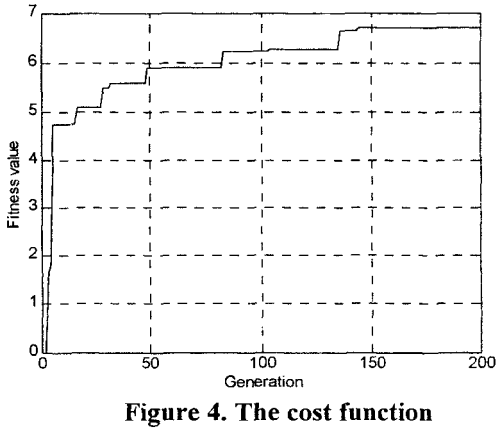

Figure 4. The cost function

In Figure 5, the position error seems to be

reduced to zero. To view the effect in tiny motion, Figure 6 plots the error from 0.15 sec. to 2 sec. We can see the steady state error is $-8\times10^{-4}mm$, or equivalently, -0.8 μm.

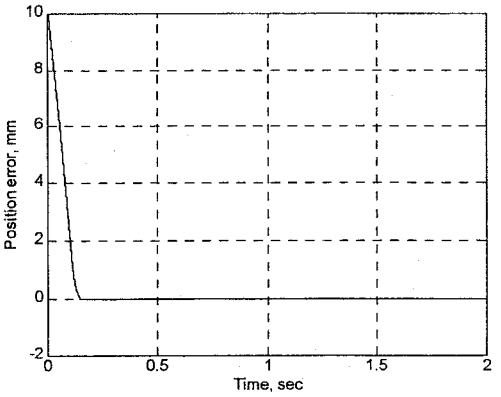

Figure 5. The position error

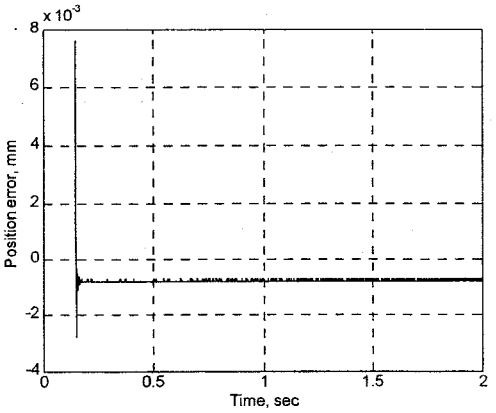

Figure 6. The position error from 0.15 sec. to 2 sec.

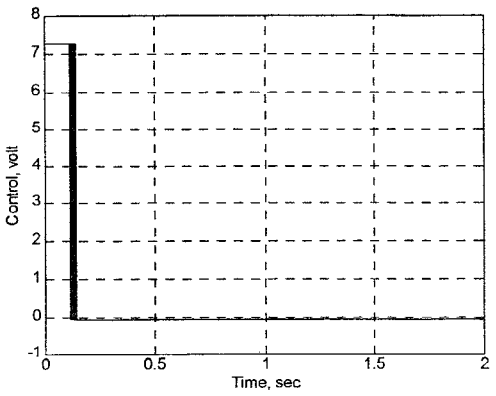

Figure 7. The control voltage

6. Conclusion

The genetic algorithm with the elitist model indeed behaves as an efficient searching tool for self-extracting the fuzzy rules of FSMCs.

The RGA-based FSMC was applied to a high precision positioning system. An XY-table with a high resolution laser scale ($0.1\,\mu m$) was used as a demonstrated plant. To avoid the truncation problem that usually presented in the conventional genetic algorithms, a read-coded genetic algorithm that uses floating-point code was employed. The experimental results shown the practicability of proposed self-learning control strategy.

Reference

[1] Armstrong, B., Dupont, P., and Canudas, C, "A Survey of Models, Analysis Tools and Compensation Methods for the Control of Machines with Friction," *Automatica*, Vol. 30, No. 7, pp. 1083-1138, 1994.

[2] Yang, S., and Tomizuka, M., "Adaptive Pulse Width Control for Precise Positioning under the Inference of Stiction and Coulomb Friction," *Transactions of the ASME Journal of Dynamic System, Measurement and Control*, Vol. 110, pp. 221-227, 1988.

[3] Mittal, S., and Menq, C. H., "Robust Compensation Techniques for DC Servomechanisms Subject to Stiction and Parametric Uncertainties Using Sliding Mode Estimation," *Proceedings of the American Control Conference*, Seattle, Washington, FA14-10:35 pp. 3306-3310, 1995.

[4] Ciliz, M. K., and Tomizuka, M., "Modeling and Compensation of Friction Uncertainties in Motion Control: A Neural Network Based Approach," *Proceedings of the American Control Conference, Seattle Washington*, FA 13-9:55 pp. 3269-3273, 1995.

[5] A. Wright, "Genetic Algorithms for Real Parameter Optimization", *In Foundations of Genetic Algorithms*, G. J. E. Rawlins (editor), Morgan Kaufmann, San Mateo, CA, pp. 205-218, 1991.

[6] H. M"uhlenbein, M. Schomish and J. Born, "The Parallel Genetic Algorithm as Function Optimizer", *Proceedings of the forth Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 271-278, 1991.

[7] Pai-Yi Huang and Yung-Yaw Chen, "Design of PID Controller for Precision Positioning Table Using Genetic Algorithms", accepted by IEEE Conference on Decision and Control 1997.

[8] Lin, S. C. and Y. Y. Chen, Design of self-learning fuzzy sliding mode controller based on genetic algorithms, *Fuzzy Sets and Systems* 86, 139-153, 1997.