Identification and Control of Systems with Friction Using Accelerated Evolutionary Programming

Jong-Hwan Kim, Hong-Kook Chae, Jeong-Yul Jeon, and Seon-Woo Lee

This article proposes a novel evolutionary algorithm, called *accelerated evolutionary programming* (AEP), which improves evolutionary programming in terms of convergence speed and diversity. Comparison between the proposed algorithm and evolutionary programming is carried out for five widely used test functions to show the effectiveness of the proposed algorithm. The proposed algorithm is applied to the identification of a seven-parameter friction model of an X-Y table, which is adopted from the results of recent tribology studies. Based on the identified friction model, a compensator is designed for the control of the X-Y table without stick-slip motion at very low velocity. Experimental results on the X-Y table demonstrate the effectiveness of the proposed scheme, especially for very-low-speed tracking.

Introduction

Friction is a natural phenomenon that is quite hard to model. The Coulomb + static + viscous friction model is most commonly used in engineering, which is based on the static mapping between the relative velocity and friction force. The friction parameters may be identified either off-line, following a data gathering experiment, or continuously, on-line as part of operation of the machine. In a Coulomb + static + viscous friction model, all the parameters are used in a linear fashion for friction modeling and may thus be identified by a standard identification technique [1, 2].

Since the linear model does not represent the friction characteristics sufficiently, we might consider as an alternative a seven-parameter friction model developed using theory and experiments by many researchers [3]. In other words, the parameters that represent a nonlinear relationship with the friction torque cannot be identified by linear identification schemes. To identify these parameters, nonlinear techniques should be used. Cheok et al. [4] employed the simplex method as a nonlinear identification technique to determine the parameters of a Karnopp friction model [5]. They pointed out that multiple local minima were observed and would have arrested a gradient technique. The presence of noise in the data was thought to have introduced local minima.

In this article, we first consider an identification problem of the seven-parameter friction model of an X-Y table with ball-screw type mechanism by using an evolutionary algorithm [6]. Identification of the friction parameters is formulated as a nonlinear optimization problem. To solve this problem, an improved evolutionary algorithm, called "accelerated evolutionary programming" (AEP), is proposed as a nonlinear identification algorithm. Comparison between the proposed algorithm and MetaLN EP [7] is carried out for five widely used test functions to illustrate the relative effectiveness. Based on the identified parameters of the seven-parameter friction model of the X-Y table, a compensation scheme is designed for the control of the X-Y table without stick-slip motion at very low velocity. The control law is composed of the conventional PD (proportional and derivative) control and the friction compensator which applies a force/torque command equal and opposite to the estimated instantaneous friction force. The applicability of the proposed scheme is demonstrated by experiments on the X-Y table for very-low-speed tracking without stick-slip motion.

Evolutionary Optimization Problem

A simple parameter optimization problem may be stated as follows:

Determine the values of the ordered set of n parameters

$$\mathbf{z} = [z_1, z_2, z_3, \ldots, z_n]^T \tag{1}$$

which minimize/maximize the cost function $f(\mathbf{z})$.

For convenience, here we consider the optimization algorithm as a function minimizer. When optimizing a function, one generally has to find a good tradeoff between convergence and diversity. The convergence means fast convergence even to a local optimum. On the other hand, the diversity guarantees high probability of finding the global optimum. The first part of this article focuses on finding an optimization algorithm which satisfies both convergence and diversity.

Recently, evolutionary algorithms which include genetic algorithms (GA), evolution strategies (ES), evolutionary programming (EP), etc., have emerged as practical, robust optimization and search methods. The binary representation traditionally used in GA has some drawbacks when applied to multidimensional, high-precision numerical optimization problems [8]. When addressing real-valued optimization problems, the solution is most appropriately represented in a real-valued vector as in ES or EP.

Based on a paper presented at the IEEE Singapore International Conference on Intelligent Control and Instrumentation, Singapore, July 1995. *J.-H. Kim, J.-Y., Jeon and S.-W. Lee are with the Department of Electrical Engineering, KAIST, Taejon-shi, 305-701, Korea. H.-K. Chae is with the Research Institute of Science and Technology, Pohang, Korea. Professor Kim can be reached by email at johkim@vivaldi.kaist.ac.kr.*

 0272-1708/96/$05.00©1996IEEE

Both EP and ES use mutations as search mechanisms and selection to direct the search toward the prospective regions in the search space. One can control either the convergence or the diversity by changing the generation method of the offspring from the parents and/or by modifying the selection mechanism. Many modifications have been suggested to the standard EP or ES to improve its performance.

Accelerated Evolutionary Programming

For the problem above, AEP based on EP is developed for the improvement of the convergence speed without decreasing the diversity among the individuals. The proposed AEP uses two variation operators according to the evaluation conditions. One is a direction operator which determines the direction of the search according to the fitness score. The other is a zero-mean Gaussian operator, which is used as a perturbation and added to a parent in order to generate an offspring as in the EP. Another variable, "age," is introduced which enhances the diversity of the search and prevents individuals from remaining in the local minima. The Gaussian operator is incorporated with both the direction operator and the variable age. A selection mechanism of the AEP is also different from that of the EP. In AEP, only one child is generated from a parent. New parents are selected by one-to-one competition between a child and its parent. An offspring is selected if it wins in the competition with its parent. This scheme does not eliminate the individuals with lower rank which have a lower fitness score in the hope of generating a global optimum.

The proposed AEP is a stochastic optimization technique for finding the extrema of an arbitrary function. The typical AEP paradigm is an iterative procedure, consisting of the evaluation of a function at an initial set of solutions in parameter space, generation of a child from a parent through random perturbation, one-to-one competition between a child and its parent, and the retention of survivals for future solution generation and functional re-evaluation. The process is then repeated. The AEP has several positive aspects, including the facility to encode real-valued parameters (phenotype) and the ability to escape local minima of the function as well as a fast convergence compared to EP.

AEP incorporates the "direction" of evolution and the "age" concept into the vector to be optimized in order to improve the convergence speed without losing a diversity among the individuals. The vector of the solution can be extended for the *i*th vector as:

$$\mathbf{z}^i = \left[z_1^i, \ldots, z_n^i, dir\left(z_1^i\right), \ldots, dir\left(z_n^i\right), age^i\right]^T \tag{2}$$

where $dir\left(z_j^i\right) \in \{-1, 1\}$ for all z_j^i is the evolving "direction" of the *j*th parameter z_j^i in the *i*th vector, and age^i denotes the duration of the life in an integer type in the *i*th vector. The following two rules are employed for perturbing the parents to generate their offspring.

Rule 1:

If $f\left(\mathbf{z}^i[k]\right) < f\left(\mathbf{z}^i[k-1]\right)$,

then: $dir\left(z_j^i[k]\right) = sgn\left(z_j^i[k] - z_j^i[k-1]\right)$,

$age^i[k] = 1$;

else: $age^i[k] = age^i[k-1] + 1$,

$\forall i \in \{1, 2, \ldots, N_p\}, \forall j \in \{1, 2, .., n\}$

where $z_j^i[k]$ denotes the *j*th parameter in the *i*th vector among N_p vectors at the *k*th generation, $dir\left(z_j^i[k]\right)$ denotes the evolving direction of $z_j^i[k]$, and "*sgn*" is a sign function.

Based on Rule 1, the mutation occurs as follows:

Rule 2:

If $age^i[k] = 1$,

then: $\sigma^i = \beta_1 \cdot f(\mathbf{z}^i[k])$,

$z_j^i[k] = z_j^i[k-1] + dir\left(z_j^i\right) \cdot \left|N\left(0, \sigma^i\right)\right|$;

else: $\sigma^i = \beta_2 \cdot f(\mathbf{z}^i[k]) \cdot age^i$,

$z_j^i[k] = z_j^i[k-1] + N\left(0, \sigma^i\right)$,

$\forall i \in \{1, 2, \ldots, N_p\}, \forall j \in \{1, 2, \ldots, n\}$

where $|\cdot|$ denotes an absolute value, $dir\left(z_j^i\right) \cdot \left|N\left(0, \sigma^i\right)\right|$ is a realization of a Gaussian-distributed random variable which is polarized in the direction of $dir\left(z_j^i\right)$, and β_i, $i = 1, 2$, are positive constants.

Rule 1 indicates that if the performance of a newly generated child is better than that of its parent, the previous search direction is retained toward the prospective regions in the search space by memorizing the sign of each parameter's evolution direction as in "then" part of Rule 2. The "then" part shows that the random variable is generated only in one region, either a positive or negative region, depending on the sign of the direction. In this case, the age of the child is set to 1.

On the other hand, if the performance of the parent is better than or equal to that of its newly generated child, the parent's age is increased by 1 as in "else" part of Rule 1. Associated with the increased age is the standard deviation of the Gaussian-distributed random variable. As the age increases, its standard deviation becomes larger as "else" part of Rule 2 shows. The "else" parts of the two conditions give an individual a chance to get out of the local minima, i.e., to have a high probability of finding the global minima.

In general, if an individual is caught in a local minimum, its standard deviation becomes small because it depends on the relatively small fitness score $f(\mathbf{z}^i)$. Thus, it can not easily escape from the local minimum. In AEP, however, the standard deviation increases gradually as the duration time, i.e., the age of the individual in local minimum, increases, which guarantees a higher probability of finding the global minimum. This means that the algorithm with "age" has more diversity than the algorithm without "age." However, it should be noted that the age concept used here is different from that in GAVaPS [9]. They introduced the concept of age of a chromosome, which is equivalent to the number of generations the chromosome stays "alive," to the selection mechanism.

Fig. 1 illustrates the selection mechanism. In AEP, only one child is generated from a parent. By one-to-one competition between a child and its corresponding parent, new parents are selected. When a child survives its corresponding parent, evolving directions of its parameters are computed according to Rule 1 and its age is set to 1. In Fig. 1, for example, parents have been

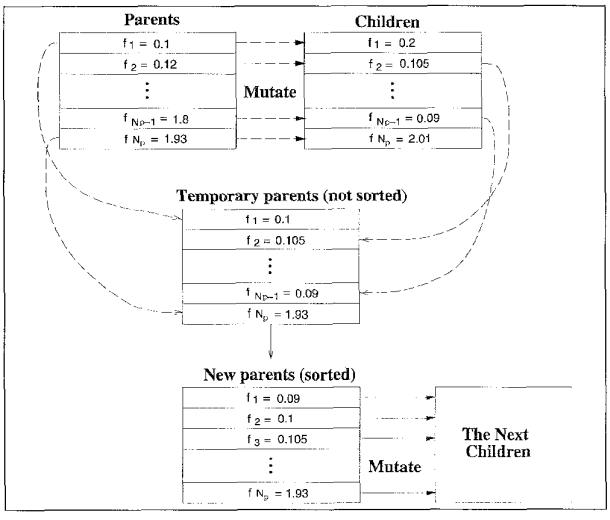


Fig. 1. Illustration of the selection mechanism.

sorted from the rank of 1st to the rank of N_pth according to the fitness scores from 0.1 to 1.93. The parents are mutated by Rules 1 and 2 such that their children are generated. By one-to-one competition between a child and its parent, a parent with fitness score 0.1 survives its child with fitness score 0.2 for the first case, a child with fitness score 0.105 survives its parent with fitness score 0.12 for the second case, and so forth. In other words, the better one is always selected. Through the one-to-one competition, temporary parents, which are not sorted, are generated. After being sorted according to their fitness scores, new parents are defined and go through the same procedure as above. For further details, the reader is referred to [10].

The procedure of the AEP is similar to that of the EP, which is summarized as follows:

1. (Initialization) Generate an initial population of N_p trial solutions with a uniform distribution within the given domain for the ith individual vector

$$\mathbf{z}^i = \left[z_1^i, \ldots, z_n^i, dir\left(z_1^i\right), \ldots, dir\left(z_n^i\right), age^i\right]^T,$$
$$\forall i \in \left\{1, 2, \ldots, N_p\right\} \tag{3}$$

where z_j^i and $dir\left(z_j^i\right), \forall j \in \{1, 2, \ldots, n\}$ are randomly initialized and age^i is initially set to 1.

2. (Evaluation) Evaluate the fitness of each parent solution by the given cost function $f(\mathbf{z}^i)$.

3. (Mutation) For each of the N_p parents $\mathbf{z}^i$, generate a child by using Rules 1 and 2 successively.

4. (Evaluation) Evaluate the fitness of each new offspring.

5. (Selection by one-to-one comparison) If the newly generated child is better than its parent, then select the child and discard the parent, else vice versa. New parents are composed of N_p vectors selected among $2N_p$ vectors by one-to-one comparison. The new N_p parents are sorted by fitness score.

6. (Termination check) Proceed to step 3 unless available execution time is exhausted or acceptable solution has been discovered.

Algorithm Comparison

Performance evaluation of probabilistic search algorithms is not an easy task in itself. To evaluate them, a performance measure and a representative suite of test functions are needed. As a performance measure, an average cost value of the best individuals was used. In this subsection we describe an evaluation of the proposed algorithm and MetaLN EP [7] for the following test problems.

Problem 1: Minimize

$$f_1(z_1, z_2, z_3) = \sum_{i=1}^{3} z_i^2$$

subject to:

$$-6.0 \le z_i \le 6.0, \quad i = 1, 2, 3.$$

Problem 2: Minimize

$$f_2(z_1, z_2) = 100\left(z_1^2 - z_2\right)^2 + (1 - z_1)^2$$

subject to:

$$-6.0 \le z_i \le 6.0, \quad i = 1, 2.$$

Problem 3: Minimize

$$f_3(z_1, \ldots, z_5) = 30.0 + \sum_{i=1}^{5} \lfloor z_i \rfloor,$$

subject to:

$$-6.0 \le z_i \le 6.0, \quad i = 1, \ldots, 5,$$

where $\lfloor \cdot \rfloor$ takes only an integer part from the argument.

Problem 4: Minimize

$$f_4(z_1, \ldots, z_{30}) = \sum_{i=1}^{30} z_i^4 + N(0, 1)$$

subject to:

$$-1.28 \le z_i \le 1.28, \quad i = 1, \ldots, 30.$$

where $N(0, 1)$ is a realization of a Gaussian random variable with mean zero and standard deviation one.

Problem 5: Minimize

$$f_5(z_1, z_2) = \frac{1}{500 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} \left(z_i - a_{ji}\right)^6}}$$

subject to:

$$-65 \le z_i \le 65, \quad i = 1, 2$$

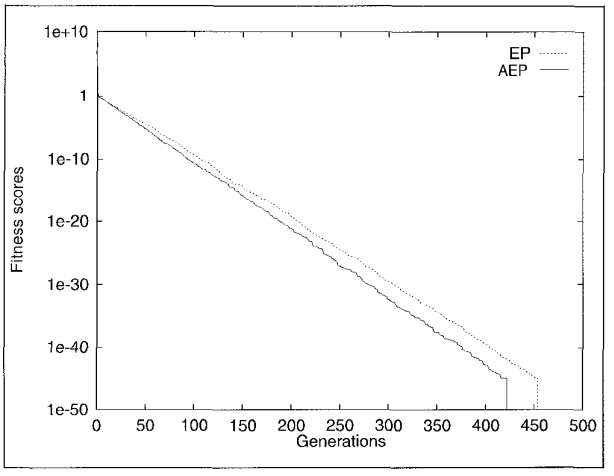


Fig. 2. The mean best score in the population averaged over 50 trials executed on function f1.

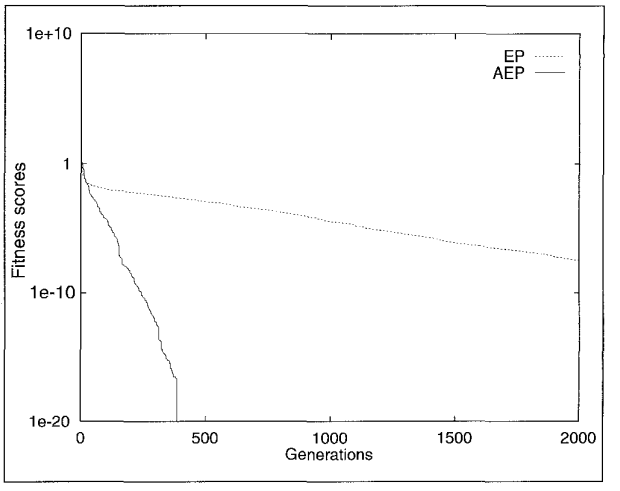


Fig. 3. The mean best score in the population averaged over 50 trials executed on function f2.

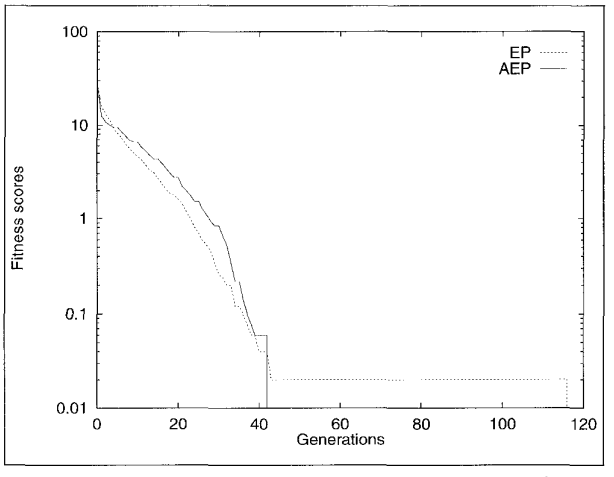


Fig. 4. The mean best score in the population averaged over 50 trials executed on function f3.

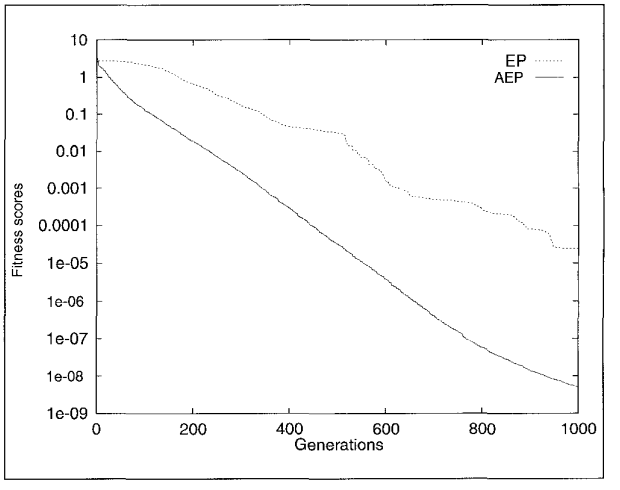


Fig. 5. The mean best score in the population averaged over 50 trials executed on function f4.

where

a_{j1} = {-32,-16,0,16,32,-32,-16,0,16,32,-32,-16,0,16,32, -32,-16,0,16,32,-32,-16,0,16,32},

a_{j2} = {-32,-32,-32,-32,-32,-16,-16, -16,-16,-16,0,0,0, 0,0,16,16,16,16,16,32,32,32,32,32}.

Problems 1-5 have been widely used for GA benchmarking [11]. The function f_1 tests simple sum of squares with a minimum at $z_i = 0$, $i = 1, 2, 3$. The function f_2 is the classical two-dimensional function of Rosenbrock and Chebyquad that is unimodal, yet difficult to minimize. The next, f_3, is the plateau function generated as a sum of integer threshold values. The five-dimensional space has one minimum and is discontinuous. The function f_4 is a noisy quartic function of 30 variables. While the intent of this function is to determine an optimizer's performance in the presence of noise, this function is perhaps flawed, as no definite global minimum exists. f_5 spans a 2-dimensional space with a global minimum ≈ 0.998004, which has 25 local minima.

Figs. 2 through 6 illustrate the performances of the two algorithms on corresponding functions f_1 through f_5. The solid lines represent the results of the AEP, and the dotted lines denote the results of the MetaLN EP. The abscissa indicates the number of generations, while the ordinate shows the mean best score in the population averaged over 50 trials. For all test problems, the population size, N_p, was fixed at 40. No determination was made to find optimal population sizes. To remove the effect of the different random initial starting points, both algorithms used the same initial population. Unless specified otherwise, $\beta_1 = 5$ and $\beta_2 = 0.05$ were used for the AEP and $\sigma^i[0] = 2.5$ was used as an initial value for the MetaLN EP, which is the same condition as in [7].

In Problem 1, both algorithms easily found a global minimum 0.0.

In Problem 2, EP found a solution (1.00002796 , 1.00005506) and the value of the objective function is $f_2 = 1.96246028 \times 10^{-7}$ in 2,000 generations. The AEP, however, found a solution (1.00000000 , 1.00000000) with $f_2 = 1.95556153 \times 10^{-29}$. Clearly, Fig. 3 shows that AEP performs better than EP.

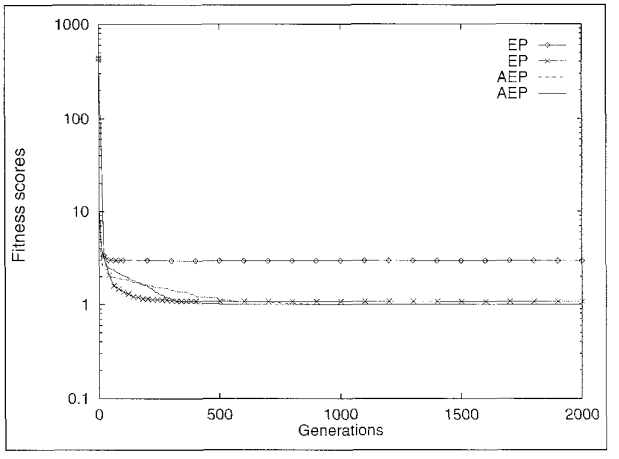


Fig. 6. The mean best score in the population averaged over 50 trials executed on function f5.

Next, in Problem 3, both algorithms found a global minimum 0.0 after around 40 generations, but EP found the global minimum in 118 generations once out of the 50 runs, as shown in Fig 4.

Since EP could not minimize the function f_4 with the given initial parameter, we replaced it by $\sigma^i[0] = 0.25$, which was the same case as in [7]. The mean best fitness scores of f_4 were $2.42449246 \times 10^{-5}$ and $4.99929185 \times 10^{-9}$, which were returned by EP and AEP, respectively, in 1,000 generations. Fig. 5 shows better performance of AEP than EP.

In Fig. 6 for Problem 5, a line with "∘" marks represents the result of EP with the preassigned parameters. In this case, EP could not find the global minimum. We applied another set of parameters ($\{\sigma^i[0] = 30,\ \tau = \left(\sqrt{2\sqrt{N_p}}\right)^{-1},\ \tau' = \left(\sqrt{2N_p}\right)^{-1}\}$) to the same function f_5 which is represented by the line with "x" marks. Then, EP could find a global minimum quickly, but still failed to find the global minimum three times out of the 50 runs. However, AEP shows very good performance in Fig. 6 regardless of the variation of the "strategy" parameters β_1 and β_2, where a solid line represents the result of the AEP with the preassigned parameter values and a dot-dashed line, for comparison purpose only, denotes the result of the AEP with $\{\beta_1 = 30,\ \beta_2 = 0.3\}$, respectively. As the figure shows, AEP is quite robust to the parameter variation.

All through the test problems, always the proposed algorithm found the global minimum. In summary, the simulation results on the five test functions show that AEP is effective in terms of convergence, diversity, and solution accuracy for function optimization.

Friction Compensator Design

In this section, we describe a seven-parameter friction model which is to be identified by using the AEP described in the previous section. The compensation scheme is designed based on the identified friction model.

Seven-Parameter Friction Model

The 1 degree of freedom (DOF) mechanical plant under investigation is a mass constrained to move in one dimension with friction present between the mass and the supporting surface, as shown in Fig 7. The equation for this model is :

$$m\ddot{x}(t) + F_f(\cdot) = F_u(t) \tag{4}$$

where m is a mass, $x(t)$ is a relative displacement, $F_f(\cdot)$ is a friction force, and $F_u(t)$ is a force applied to m.

The friction model used in this article is the seven-parameter friction model, which consists of pre-sliding displacement and Coulomb + viscous + Stribeck friction terms. The model is summarized as follows [3]:

- (pre-sliding displacement):

$$F_f(x) = k_t x \tag{5}$$

- (Coulomb + viscous + Stribeck):

$$F_f\left(\dot{x}, t_2, F_{s,a}\right) = \left(F_c + F_v|\dot{x}| + F_{s,b}\left(\gamma, t_2, F_{s,a}\right) \frac{1}{1+\left(\frac{\dot{x}(t-\tau_L)}{\dot{x}_s}\right)^2} \right) sgn(\dot{x}) \tag{6}$$

with

$$F_{s,b}\left(\gamma, t_2, F_{s,a}\right) = F_{s,a} + \left(F_{s,\infty} - F_{s,a}\right)\frac{t_2}{t_2+\gamma} \tag{7}$$

where $F_f(\cdot)$ is the instantaneous friction force, F_C is the Coulomb friction force*, F_v is the viscous friction force*, $F_{s,b}$ is the magnitude of the Stribeck friction, $F_{s,a}$ is the magnitude of the Stribeck friction at the end of the previous sliding period, $F_{s,\infty}$ is the magnitude of the Stribeck friction after a long time at rest*, k_t is the tangential stiffness of the static contact*, $\dot{x}_s$ is the velocity of the Stribeck friction*, τ_L is the time constant of frictional memory*, γ is the temporal parameter of the rising static friction*, t_2 is the dwell time, i.e., time at zero velocity, x is the relative displacement, and

$$sgn(\dot{x}) = \begin{cases} +1 & \dot{x} > 0 \\ 0 & \dot{x} = 0 \\ -1 & \dot{x} < 0. \end{cases}$$

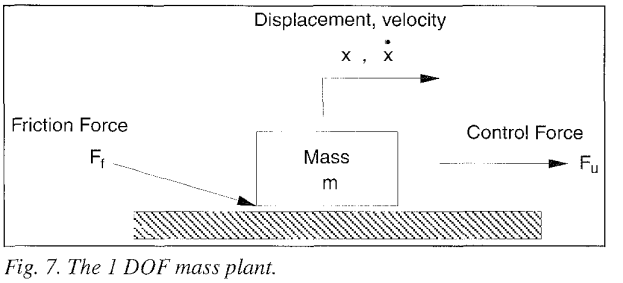


Fig. 7. The 1 DOF mass plant.

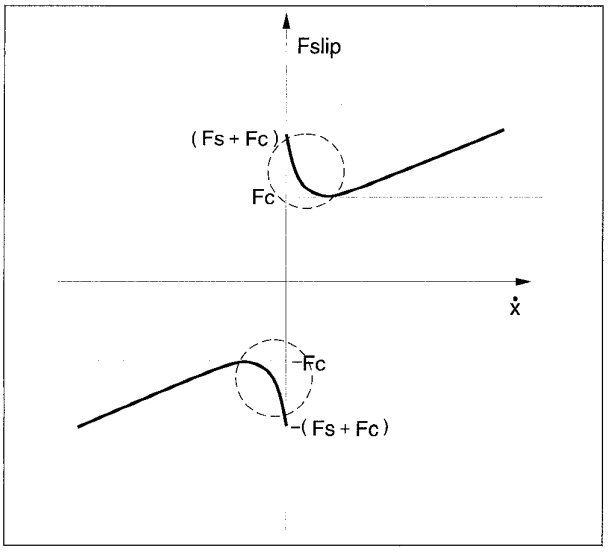


Fig. 8. The Stribeck effect (negative viscous friction).

The parameters marked with (*) are friction model parameters, and the other variables are state variables. The friction model parameters are to be identified by using the AEP. The friction model above is briefly described in the following.

1. Pre-sliding displacement. It is well known that contacts are compliant in both the normal and tangential directions. Johnson [12] and Dahl [13], studying experimental observations of friction in small rotation of ball bearings, concluded that for a small motion a junction in static friction behaves like a spring. There is a displacement (pre-sliding displacement) which is an approximately linear function of the applied force up to a critical force, at which breakaway occurs. When a control force $F_u(t)$ is applied, the asperities will deform, but recover when the force is removed, as does a spring. The tangential force is governed by

$$F_u(t) = k_t x$$

where $F_u(t)$ is the applied tangential force, k_t is the tangential stiffness of the static contact, and x is the displacement away from the equilibrium position. $F_u(t)$ and x refer to the force and displacement in the contact before sliding begins. When the applied force exceeds the breakaway force, the junctions break and true sliding begins.

2. Stribeck effect. The Stribeck effect is shown in Fig. 8. Because of the Stribeck effect, the friction force decreases with increasing of the relative velocity within the low-velocity sliding region. This phenomenon is due to partial fluid lubrication in solid-to-solid contact and may be one of the main reasons of stick-slip motion [14]. The curves in the dashed circles of Fig. 8 are represented mathematically as:

$$F_s(\dot{x}) = \left(F_{static} \frac{1}{1 + \left(\frac{\dot{x}}{\dot{x}_s} \right)^2} \right) sgn(\dot{x}) \tag{9}$$

where $\dot{x}_s$ is the characteristic velocity of the Stribeck friction.

3. Frictional memory. The Stribeck curve shows a dependence of friction upon relative velocity. If there is a change in velocity, one might presume the corresponding change in friction force to occur simultaneously. In fact there is a delay in the effect on the friction. This phenomenon is dominant in the lower velocity region and can be included into Equation (9) in the manner

$$F_s(\dot{x}) = \left(F_{static} \frac{1}{1 + \left(\frac{\dot{x}(t - \tau_L)}{\dot{x}_s} \right)^2} \right) sgn(\dot{x}) \tag{10}$$

where τ_L represents a frictional memory.

4. Rising static friction. Let us consider a 1 DOF plant that exhibits cyclic stick-slip motion. The dwell time t_2 is defined as the time duration when relative velocity $\dot{x} = 0$. The static friction force increases with increasing of the dwell time t_2, and this accounts for the larger limit cycle at a lower velocity. The relationship between the static friction force and dwell time t_2 is:

$$F_{s,b}(\gamma, t_2) = F_{s,a} + \left(F_{s,\infty} - F_{s,a} \right) \frac{t_2}{t_2 + \gamma} \tag{11}$$

where $F_{s,b}$ is the magnitude of the Stribeck friction, $F_{s,a}$ is the magnitude of the Stribeck friction at the end of the previous sliding period, $F_{s,\infty}$ is the magnitude of the Stribeck friction at $t_2 = \infty$, and γ is the temporal parameter of the rising static friction.

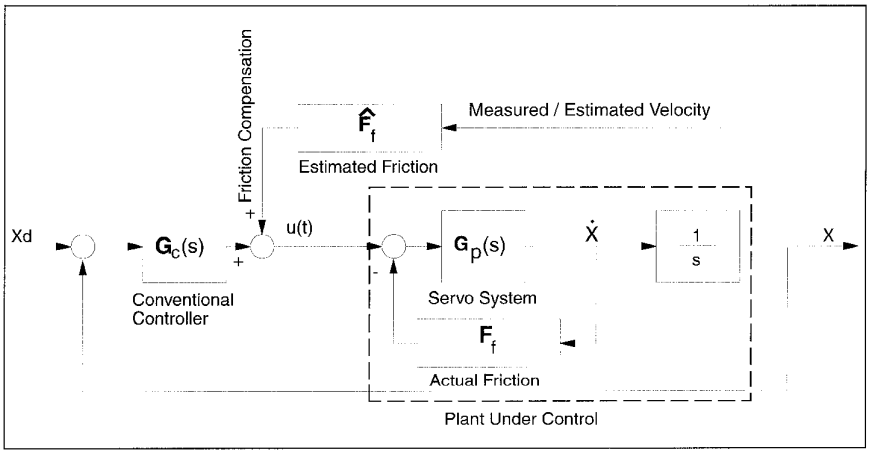


Fig. 9. The friction model-based compensation scheme.

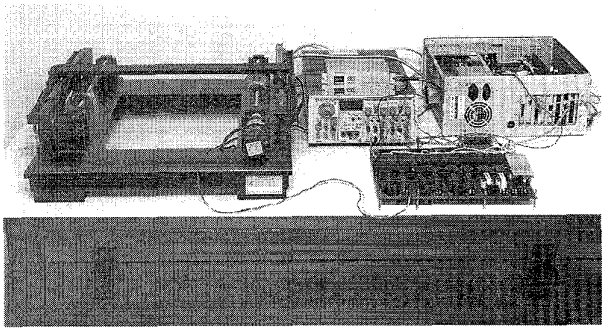


Fig. 10. Experimental setup.

Friction Compensation Scheme

When an identified friction model is available, it is possible to compensate friction by applying a force/torque command equal and opposite to the estimated instantaneous friction force as shown in Fig. 9. We should know the values of parameters $\dot{x}$, t_2, and $F_{s,a}$ to evaluate the friction force expressed by Equations (7-9). In this subsection, we first describe how to obtain the parameters from the transition of motion states. The motion state "sw" is defined as:

$$sw = \begin{cases} 0 & \text{if } |\dot{x}| \le V_{cr}\text{: stick} \\ 1 & \text{otherwise: slip} \end{cases} \tag{12}$$

where V_{cr} is pre-specified by the resolution of a velocity sensor.

From the above equation, t_2 and $F_{s,a}$ are obtained as follows:

$$\begin{cases} sw\text{: } 0 \mapsto 0, & t_2 \Leftarrow t_2 + T_s \\ sw\text{: } 0 \mapsto 1, & \text{compute } F_{s,b} \\ sw\text{: } 1 \mapsto 0, & F_{s,a} \Leftarrow F_{s,b}(\gamma, t_2)\dfrac{1}{1+\left(\dfrac{\dot{x}(t-\tau_L)}{\dot{x}_s}\right)^2} \\ & \text{and } t_2 \Leftarrow 0 \\ sw\text{: } 1 \mapsto 1, & \text{No operation} \end{cases} \tag{13}$$

where T_s is the sampling time of the control system, and "$\mapsto$" denotes the state transition.

Thus, the measured velocity $\dot{x}(t)$ can be used to evaluate the friction force. Consequently, the control input $u(t)$ is

$$u(t) = u_{PD}(t) + \hat{F}_f\left(\dot{x}(t), t_2, F_{s,a}\right) \tag{14}$$

where $\hat{F}_f(\cdot)$ is the estimated friction and $u_{PD}(t)$ is the output of the conventional PD controller. Fig. 9 shows a friction model-based compensation scheme.

Application

In this section, we describe the application of the scheme developed in the previous section to the identification and control of the X-Y table with ball-screw type positioning mechanism for very-low-speed tracking without stick-slip phenomena.

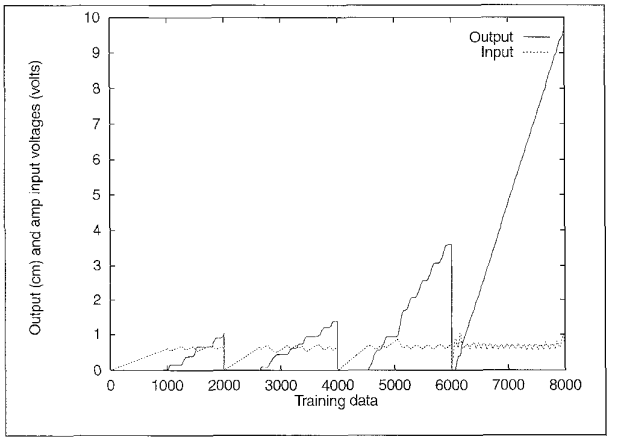


Fig. 11. Training I/O sample data.

Experimental Environment

The experimental setup is shown in Fig. 10. The plant consists of the positioning mechanism, a position sensor system, a servo amplifier, and an IBM-compatible PC equipped with a custom board containing a 24-bit counter and a 1-channel DA conversion circuit for 1-channel analog voltage output. As shown in Fig. 10, the X-Y table has two linear motion mechanisms which are composed of a DC servo motor, a screw, and a sensor. The DC servo motor is connected with ball-screw through a coupler. A linear encoder with one-micrometer (μm) resolution, equipped in the X-axis mechanism, is used as a position sensor.

We can obtain the voltage-current characteristics of the current servo amplifier by measuring the current supplied to the DC motor. The amplifier model is included into the mathematical plant model in computer simulation. The main control algorithm is implemented via the PC with the Intel i486DX-66 microprocessor. The sampling time used here is 5 (msec). The proposed algorithm is written in the C language.

Identification Problem

Problem Statement

Let a vector $\mathbf{z}$ denote a set of unknown plant parameters in Equation (4). The vector $\mathbf{z}$ consists of an equivalent mass m_e and seven friction parameters that were discussed in the previous section:

$$z = \left[m_e, F_C, F_v, F_{s,\infty}, \gamma, \tau_L, \dot{x}_s, k_t\right]^T, \; z \in \Re^8 \tag{15}$$

where all the parameters are positive. Let us define the mismatched displacement error between the plant output and the identification model output as follows:

$$e(\hat{z}, t_i) = x(\mathbf{z}, t_i) - x_m(\hat{z}, t_i), \;\; i = 1, 2, \ldots, N_s \tag{16}$$

where $\hat{\mathbf{z}}$ denotes the estimate of $\mathbf{z}$, $x(\mathbf{z}, t_i)$ is a displacement of the plant, $x_m(\hat{\mathbf{z}}, t_i)$ is a displacement of the identification model at a sample instant t_i, which depends on the actual $\mathbf{z}$ and the estimated vector $\hat{\mathbf{z}}$, respectively, and N_s is the total number of sample data. Consider the cost function of the squared errors given by

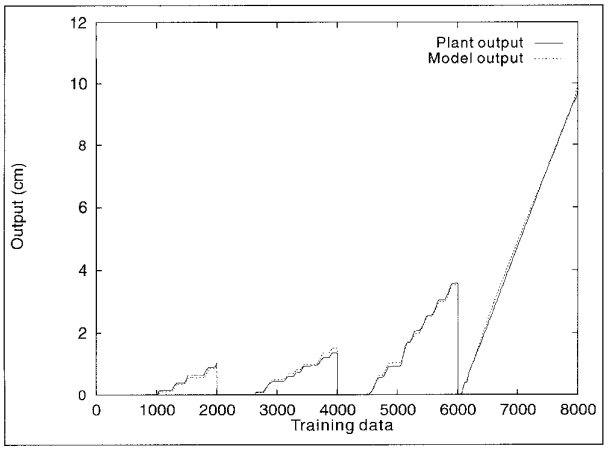


Fig. 12. Outputs of the X-Y table and the identified model after 250 generations.

$$J_e(\hat{\mathbf{z}}) = \sum_{i=1}^{N_s} e^2(\hat{z}, t_i) . \tag{17}$$

Then, the parameter identification problem can be posed as an optimization problem as follows:

$$\underset{\hat{\mathbf{z}}}{\text{Min}}\ J_e(\hat{\mathbf{z}}) .$$

Choice of Training I/O Sample Data

The choice of the training I/O sample data is an interesting aspect of both linear and nonlinear system identification problems. Successful identification of the plant parameters requires the proper choice of training I/O data. There are two different kinds of parameters in the model under investigation: one associated with linear characteristics and the other with nonlinear characteristics. A solution to the suitable choice of the training I/O data would be a set of I/O data which exhibits the stick-slip motion in the region of lower velocity, i.e., nonlinear characteristics, and the fast motion for parameters with linear characteristics. Considering this aspect, training I/O sample data, N_s = 8,000, were collected from four operating regions, each consisting of 2,000 sample training points, as shown in Fig. 11. The first of three blocks of sample data in Fig. 11 are mainly for the nonlinear characteristics, and the last one is for the linear characteristics of the X-Y table. These I/O sample data could be obtained through the application of PD control to the X-Y table. Since the sampling time is 5 (msec), the run time of each block of 2,000 sample training points is 10 (sec).

Results of Identification

We tried to identify the parameters of the system by using the MetaLN EP for the same 8,000 training points; however, the obtained results showed that the EP could find only the local minima even by many trials. The parameters of the local minima caused a large identification error, so the wrongly identified model could not be used in the control.

On the other hand, the result of identification using the AEP is shown in Fig. 12, and the equivalent mass and the estimated friction parameters after 250 generations are:

m_e = 0.034 Kg
F_C = 0.4040 Kgf
F_v = 0.4244 Kgf · sec/cm
$F_{s,\infty}$ = 0.4260 Kgf
$\dot{x}_s$ = 0.0001 cm/sec
τ_L = 25 msec
γ = 0.0089 msec
k_t = 7173 Kgf/cm .

From Fig. 12, we can see that the response of the identified model is almost the same as that of the X-Y table for the wide range of operating points which can reveal both linear and nonlinear characteristics. This indicates that the friction of the X-Y table is well identified by the integrated friction model. The estimated friction will be used in a control law to eliminate the actual friction present in the X-Y table, whereas the estimated equivalent mass m_e is used only for the identification of the friction model.

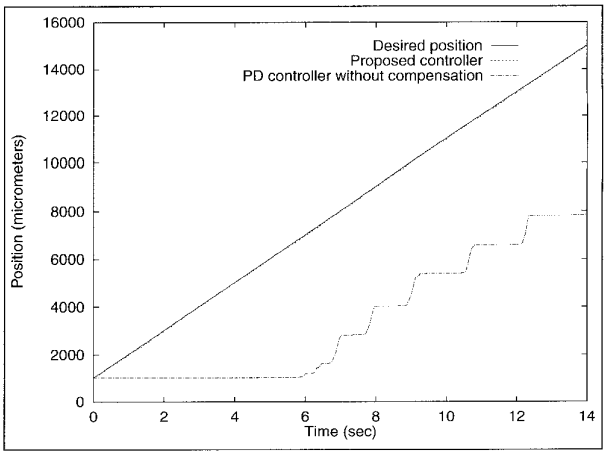


Fig. 13. Output responses for a ramp input with the slope of 0.1 cm/sec.

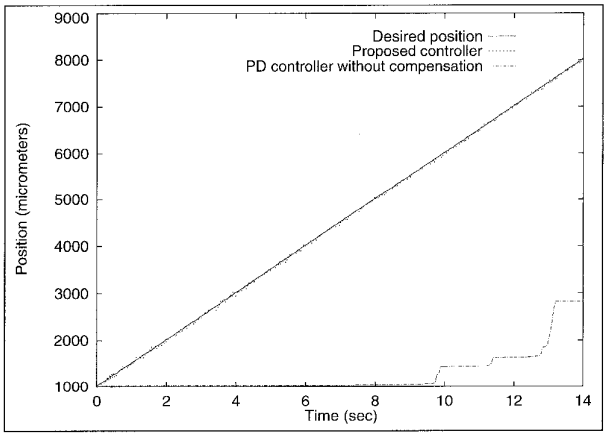


Fig. 14. Output responses for a ramp input with the slope of 0.05 cm/sec.

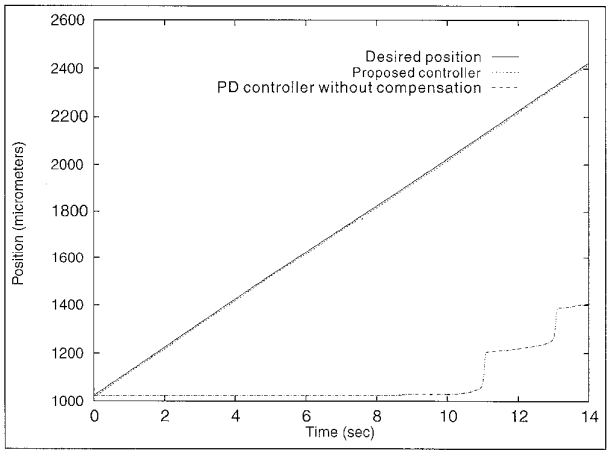


Fig. 15. Output responses for a ramp input with the slope of 0.01 cm/sec.

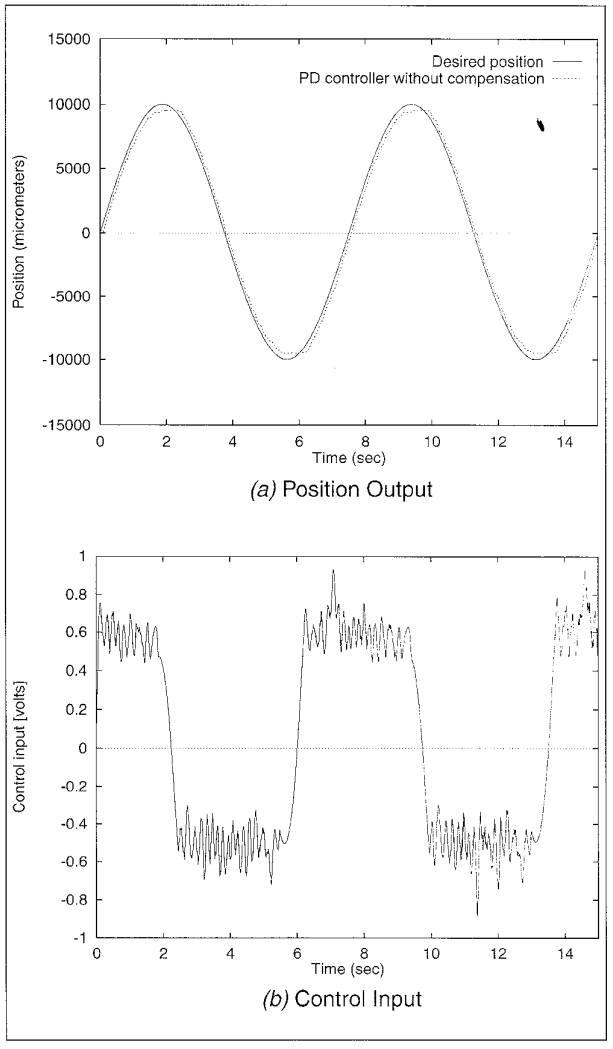


Fig. 16. Experimental results of PD control without friction compensation.

Control with the Friction Compensator

For simplicity, we applied the proposed control scheme only to the X-axis positioning system. As we used only a position sensor with one-micrometer resolution and 5 (msec) sampling time, we could detect a slip motion with a resolution of 0.02 (cm/sec). Considering the noise sensitivity, however, the V_{cr} which is used to determine the state of the plant in Equation (12) was set to 0.06 (cm/sec), and the desired velocity $\dot{x}_d$ was used to evaluate the control input instead of $\dot{x}$ in Equation (14). The gains of PD controller used here were $K_P = 0.1$ and $K_D = 0.1$.

The results of the PD control and the proposed compensation scheme are shown in Figs. 13, 14, and 15 for $\dot{x}_d(t) = 0.1$ (cm/sec), $\dot{x}_d(t) = 0.05$ (cm/sec), and $\dot{x}_d(t) = 0.01$ (cm/sec), respectively, to verify that the estimated friction can represent the actual properties of the friction in the ball-screw contact. Note that the linear velocities 0.1, 0.05, and 0.01 (cm/sec) correspond to 12, 6, and 1.2 (rpm), respectively, because a lead pitch of the ball-screw is 5 (mm), where a lead pitch is defined as a linear distance per motor revolution. From the experimental results, we can see that the proposed friction compensation scheme eliminates the stick-slip output responses of the PD controller at lower tracking

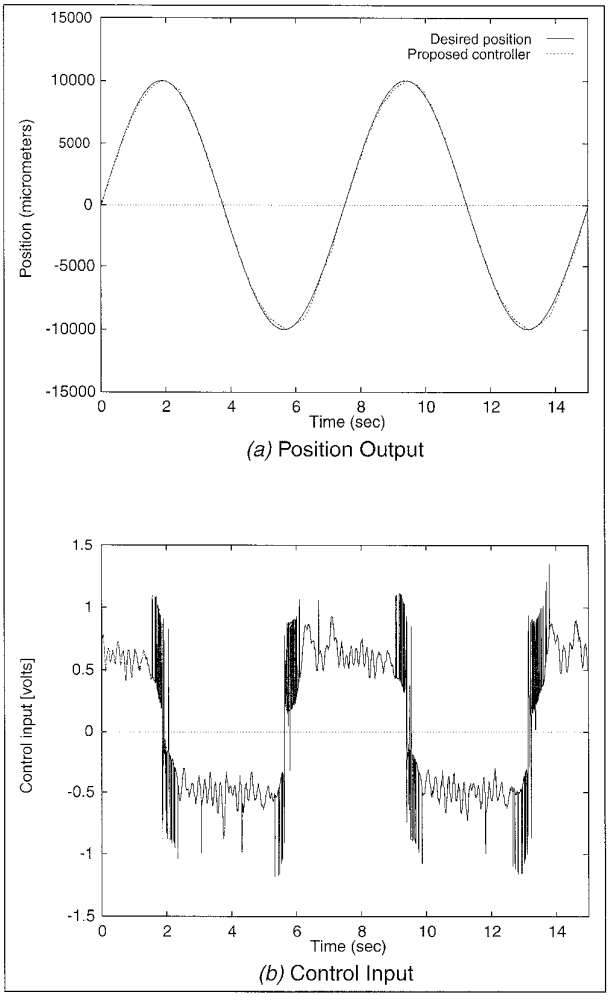


Fig. 17. Experimental results of the proposed scheme.

velocity. This means that most of the friction present in the X-Y table system is well compensated by the proposed scheme so that the motion dynamics are governed by a simple linear algebraic equation. Thus, the control problem of low-speed tracking becomes trivial.

The proposed scheme can be also applied to eliminate stick-slip at standstill, which occurs with velocity reversal. From Fig. 16, we can see there exist standstill phenomena with velocity reversal when PD control is applied without friction compensation. However, the stick-slip at standstill is almost avoided by the proposed scheme as shown in Fig. 17.

Conclusions

We have proposed an evolutionary optimization algorithm, called *accelerated evolutionary programming*, which can be applied to the identification and control of a broad class of nonlinear systems. To show the effectiveness of the AEP, we have compared the AEP with the MetaLN EP for the widely used test problems. The friction compensation scheme has been proposed based on the identified friction model obtained by the AEP. We have shown the effectiveness and applicability of the proposed control scheme via experiments on the X-Y table positioning control of very-low-speed tracking without stick-slip phenomena.

References

[1] C. Canudas, K.J. Åström, and K. Braun, "Adaptive Friction Compensation in DC Motor Drives," *IEEE J. of Robotics and Automation*, vol. 3, pp. 681-685, December 1987.

[2] C. Canudas, P. Noel, A. Aubin, and B. Brogliato, "Adaptive Friction Compensation in Robot Manipulators: Low Velocities," *The Int. J. of Robotics Research*, vol. 10, pp. 189-199, June 1991.

[3] B. Armstrong-Hélouvry, "A Survey of Models, Analysis Tools and Compensation Methods or the Control of Machines with Friction," *Automatica*, vol. 30, pp. 1083-1138, October 1994.

[4] K.C. Cheok, H. Hu, and N.K. Loh, "Modeling and Identification of a Class of Servomechanism Systems with Stick-Slip Friction," *J. of Dyn. Sys., Measure., and Control*, vol. 110, pp. 324-328, September 1988.

[5] D. Karnopp, "Computer Simulation of Stick-Slip Friction in Mechanical Dynamic Systems," *ASME J. Dyn. Sys. Meas. Contr.*, vol. 107, pp. 100-103, March 1985.

[6] J.-H. Kim, H.-K. Chae, J.-Y. Jeon, and S.-W. Lee, "Friction Parameter Identification and Compensation Scheme of x-y Table by Using Evolutionary Algorithm," in *Proc. of IEEE Singapore International Conference on Intelligent Control and Instrumentation*, pp. 167-172, Singapore, July 1995.

[7] N. Saravanan and D.B. Fogel, "Learning Strategy Parameters in Evolutionary Programming: an Empirical Study," in *Proc. of Third Annual Conf. on Evolutionary Programming* (A.V. Sebald and L.J. Fogel, eds.), pp. 269-280, San Diego, February 1994.

[8] Z. Michalewicz, "Evolutionary Operators for Continuous Convex Parameters Spaces," in *Proc. Third Annual Conf. on Evolutionary Programming* (A.V. Sebald and L.J. Fogel, eds.), pp. 84-97, San Diego, February 1994.

[9] J. Arabas, Z. Michalewicz, and J. Mulawka, "GAVaPS—A Genetic Algorithm with Varying Population Size," in *Proc. of the First Conference on Evolutionary Computation* (Z. Michalewicz, H. Kitano, D. Schaffer, H.-P. Schwefel, and D. Fogel, eds.), pp. 73-78, Orlando, June 1994.

[10] J.-H. Kim, J.-Y. Jeon, and K. Koh, "A Novel Evolutionary Algorithm with Fast Convergence," in *Proc. of IEEE International Conference on Evolutionary Computation*, Perth, Australia, November 1995.

[11] K. DeJong, *An Analysis of the Behavior of a Class of Genetic Adaptive System*, Ph.D. thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI, 1975.

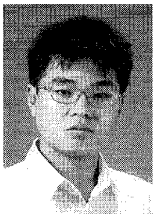
[12] K.L. Johnson, *Contact Mechanics*, Cambridge: Cambridge University Press, 1987.

[13] P.R. Dahl, "Measurement of Solid Friction Damping of Mechanical Vibrations," *AIAA J.*, vol. 14, pp. 1675-1682, December 1976.

[14] D.P. Hess and A. Soom, "Friction at a Lubricated Line Contact Operating at Oscillating Sliding Velocities," *J. of Tribology*, vol. 112, no. 1, pp. 147-152, 1990.

[15] C. Canudas, H. Olsson, K.J. Åström, and P. Lischinsky, "Dynamic Friction Models and Control Design," in *Proc. of American Control Conference*, pp. 10-12, San Francisco, June 1993.

Jong-Hwan Kim received the B.S., M.S., and Ph.D. degrees in electronics engineering from Seoul National University, Seoul, Korea, in 1981, 1983, and 1987, respectively. From 1984 to 1988, he was a research assistant in the Department of Electronics Engineering at Seoul National University. Since 1988, he has been with the Department of Electrical Engineering at the Korea Advanced Institute of Science and Technology (KAIST), where he is currently an associate professor. He was a visiting scholar at Purdue University from September 1992 to August 1993. His research interests are in the areas of adaptive control, intelligent machines, and soft computing, including evolutionary computation. Dr. Kim was the recipient of the 1988 ChoonGang Young Investigator Award from ChoonGang Memorial Foundation. He is an associate editor of the *International Journal of Intelligent and Fuzzy Systems*.



Hong-Kook Chae was born in Ulsan, Korea, in 1970. He received the B.S. degree in electronic engineering from Hanyang University and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology in 1993 and 1995, respectively. He is currently a researcher at the Research Institute of Science and Technology, Pohang, Korea. His research interests are in the areas of intelligent control, neural networks, and evolutionary computation.

Jeong-Yul Jeon was born in Kongju, Korea, in 1968. He received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Taejeon, Korea in 1990 and 1993, respectively. He is currently a candidate for the Ph.D. degree in the Department of Electrical Engineering at the Korea Advanced Institute of Science and Technology and a researcher in LG Industrial Systems Co., Anyang, Korea. His research interests are in the areas of adaptive control, intelligent control, and precision control.

Seon-Woo Lee was born in Seoul, Korea, in 1967. He received the B.S. and M.S. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology, Taejeon, Korea, in 1990 and 1992, respectively. He is currently a candidate for the Ph.D. degree in the Department of Electrical Engineering at Korea Advanced Institute of Science and Technology. His research interests are in the areas of intelligent control, fuzzy logic-based control, and adaptive control.