International Journal of Innovative Computing, Information and Control
Volume **9**, Number **10**, October **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 3903–3914

ADAPTIVE PSO-BASED SELF-TUNING PID CONTROLLER FOR ULTRASONIC MOTOR

Alrijadjis Djoewahir[1,2], Kanya Tanaka[2] and Shota Nakashima[2]

[1]Electronic Engineering Polytechnic Institute of Surabaya
Kampus ITS, Keputih, Sukolilo, Surabaya 6011, Indonesia
r501we@yamaguchi-u.ac.jp

[2]Department of Electrical and Electronic Engineering
Yamaguchi University
2-16-1, Tokiwadai, Ube, Yamaguchi 755-8611, Japan
{ s-naka; ktanaka }@yamaguchi-u.ac.jp

Received September 2012; revised January 2013

Abstract. *An adaptive particle swarm optimization based proportional-integral-derivative (APSO-PID) controller is proposed to control an ultrasonic motor (USM). Since the dynamic characteristic of the USM is difficult to obtain and the motor parameters are time varying, the APSO algorithm is applied to automatically tune the gains of PID controller without using any plant model of the USM. In the proposed method, an inertia weight of PSO is adaptively adjusted based on the swarm condition. The personal and the global best position of particles is taken and used to calculate the inertia weight during the searching process by using feedback mechanism. The aim of this study is to present an optimal self-tuning PID controller using APSO for compensating the dynamic characteristic changes and nonlinearity of USM. To demonstrate the performance of the APSO-PID controller, it is tested on USM servo system and compared with the previous methods. Results show that the performance of the proposed method is satisfactory in term of convergence speed and position accuracy of USM.*
Keywords: Feedback mechanism, PID controller, Particle swarm optimization (PSO), Ultrasonic motor (USM), Self-tuning

1. **Introduction.** Ultrasonic motors (USM) have received much attention these last years. The USM is a special type motor, which is driven by the ultrasonic vibration force of piezoelectric elements. It has excellent performance and many other useful features, which are not present in other electromagnetic type motor, e.g., compactness, light-weight, no-running sound, high torque even at a low speed, high retention torque within the stop condition, quick response, no emitted electromagnetic interferences, high holding torque without supply [1-4]. In actual applications, the USM has been used as actuator of cameras, as direct drive actuators for articulated robot, control valve, in medical equipment and in strong magnetic field area. The drive principle of USM has a complex mechanism. The rotor is moved by ultrasonic vibration force of piezoelectric elements on the stator, the resonant frequency and mechanical characteristics of the motor vary with the driving condition, such as temperature, input frequency, load, and disturbances. That is, the USM has time-varying and non-linear characteristics.

For this reason, it is difficult to derive accurate mathematical model of the USM based on physical analysis. Though some models for USM have been proposed, they were not satisfactory because the given model is not covering all properties of the USM and use parameters simplification [5-7]. Therefore, it is difficult to predict the performances and the characteristics of USM under various working conditions. Under these conditions, the controller for USM, which can achieve required high-accuracy in spite of non-linearity and uncertain effects, is more difficult.

There is no perfect control scheme for USM, especially in the precise speed and position servo systems. Aimed at these problems, much research work has recently been done. Due to the fact that fuzzy logic control (FLC) is especially preferred when the mathematical model of a control plant is difficult to obtain, Bal et al. have developed a position control using fuzzy logic control, which can give superior speed and position characteristics for the USM [8]. Li et al. have presented a precise position control using fuzzy logic control with dead-zone compensation for USM. The dead-zone effect may reduce the precision of USM and by using FLC with higher gain; this effect can be addressed [9]. Yoshida et al. have realized speed sensor-less control of USM using neural network (NN), which performs well characteristics. The nonlinear identification ability of the NN has realized enhanced control performance which includes nonlinearities associated with load torque and surface temperatures of USM [10]. Since a PID controller can be built even if there is no plant model, PID controller has been widely used as the control method for USM [11,12]. Moreover, the PID controller has a simple structure, efficient, effective, easy implementation and quite robust. PID controller has been used for a long time and many systems that used PID controller showed the satisfactory performance. PID controller has been proven to be able to overcome the nonlinearity properties and complexity of plant. The main problem related with USM servo system is how to optimally tune the gains of PID controller, i.e., the determination of PID gains for satisfactory control performance. In order to get better dynamic performance, guarantee security and sustainable utilization of equipment and plants, PID gains must be properly tuned. Many methods have been reported for improving the setting performance of PID gains of USM in the past decades, such as fuzzy based-auto tuning PID [13], GA based-self-tuning PID [14], BPNN based-PID [15], and PSO based-PID [16]. Recently, PSO is attractively used for self-tuning PID controller because it has superior properties such as easy implementation, simple algorithm, fast and stable convergence and efficient in time calculation. PSO gives a better result than the previous methods in term of convergence speed and solution accuracy. Although PSO has superior properties, they have some inherent problems. PSO may have a tendency to prematurely converge and fall into local optima. In order to overcome those problems, many researchers have attempted to improve the PSO algorithm [17-21]. However, reports on the improvement of PSO for self-tuning PID controller of USM are relatively few.

The main contribution of this study is to propose a modified particle swarm optimization algorithm that we have called adaptive particle swarm optimization (APSO). The proposed method is applied to automatically determine the gains of PID controller of USM in order to get the best optimal performance. The benefit of the proposed method is its simplicity to address the problems of PSO by using feedback mechanism to adjust adaptively the balancing between exploration-exploitation abilities based on the swarm condition. To validate and prove the performance and the effectiveness of our method, we have tested in the real USM servo system and compared with the previous methods.

This paper is organized as follows. In Section 2, we describe the PSO algorithm and its variants. Section 3 talks about the proposed APSO. Section 4 describes the implementation of APSO in PID controller for USM. Section 5 shows the experimental results. Finally, Section 6 shows the conclusions.

2. **PSO Algorithm and Its Variants.**

PSO is a population based stochastic optimization method using the concept of cooperation inspired by the behavior of organism, such as birds flocking, in search for food [15]. The outline for PSO is marked as follows. Let consider the optimization problem of maximizing the evaluation function f: $M \to M' \subset R$ for variable $x \in M \subset R^n$. Let there be N particles (mass point) on M dimensional space, where the position vector and velocity vector of i $(= 1, 2, 3, \ldots, N)$-th particle for k searching number are x_i^k and v_i^k. The best position for each particle in the evaluation function $f(x)$ of $x_i^1, x_i^2, \ldots, x_i^k$ searching point is represented as P_{bi} ($Pbest$), while the best position of $f(x)$ in the searching point for the whole particle is represented as g_b ($gbest$). The particles are manipulated according to the following recurrence equations:

$$v_i^{k+1} = w.v_i^k + c_1.R.\left\{P_{bi} - x_i^k\right\} + c_2.R.\left\{g_b - x_i^k\right\} \tag{1}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{2}$$

where w is the inertia weight; c_1 and c_2 are cognitive and social constant; R is uniform random numbers from the interval [0, 1].

The example for optimized solution search using PSO is shown in Figure 1. The movement of particles is governed by three parts: (a) the inertial part, $w.v_i^k$; (b) the cognitive part, $(P_{bi} - x_i^k)$; (c) the social part, $(g_b - x_i^k)$. The velocity vector of v_i^{k+1} is formed based on three vectors as shown in (1). The first one is inertia vector, which is the vector from weighting factor w and the velocity vector v_i^k. The remaining two are vectors for each $(P_{bi} - x_i^k)$ and $(g_b - x_i^k)$, which formed from learning factor c_1 as well as c_2, and also [0, 1] of uniform random numbers R. From those interactions, velocity vector v_i^{k+1} act so that the particle moves to new position, x_i^{k+1}.

2.1. **PSO with linearly decreasing inertia weight (PSO-LDW).** A linearly decreasing inertia weight was first introduced by Shi and Eberhart in 1998 [20]. At beginning of iterations, a larger inertia weight is applied for global searching. During the search, inertia weight is becoming smaller and smaller for local searching, which leads to significant improvement in the performance of original PSO. The linearly decreasing inertia weight is modified as:

$$w^k = w_{\max} - \left(\frac{w_{\max} - w_{\min}}{k_{\max}}\right) \cdot k \tag{3}$$

And the velocity is modified as:

$$v_i^{k+1} = w^k.v_i^k + c_1.R.\left\{P_{bi} - x_i^k\right\} + c_2.R.\left\{g_b - x_i^k\right\} \tag{4}$$

where $w_{\max}$ is the initial inertia weight, $w_{\min}$ is the final inertia weight, and $k_{\max}$ is the number of maximum iteration. Normally, $w_{\max}$ is set to 0.9 and $w_{\min}$ is set to 0.4.

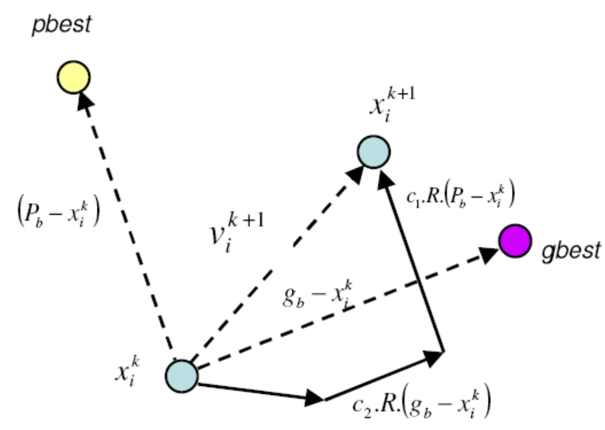

Figure 1. Search example of PSO

2.2. **PSO with nonlinearly decreasing inertia weight (PSO-NDW).** A nonlinearly decreasing inertia weight is proposed to investigate the influence of path variant of decreased inertia weight [17]. Nonlinearity degree of the path can be adjusted by nonlinear index number. This number is a new parameter in PSO-NDW which can significantly affect the performance of PSO. The nonlinearly decreasing inertia weight is modified as:

$$w^k = w_{\min} + (w_{\max} - w_{\min}) \cdot \left(\frac{k_{\max} - k}{k_{\max} - 1} \right)^x \tag{5}$$

where x is the nonlinear index number. Normally, x is set to 1.5 for the best performance.

2.3. **PSO with random inertia weight (PSO-RIW).** A random inertia weight is proposed to provide both abilities of PSO in the same time during iteration; thus both abilities of PSO can be processed in the same time in order to eliminate the lack of exploitation ability at the beginning of iteration and exploration ability at the end of iteration (this is a shortcoming of PSO-LDW and PSO-NDW) [18]. Due to this strategy, the capability of PSO to avoid premature convergence and falling into local optima is better than the previous methods. The random inertia weight is modified as:

$$w = w_{\min} + (w_{\max} - w_{\min}) \cdot R \tag{6}$$

3. **The Proposed APSO.** The experiments indicate that inertia weight is most important parameter to balance the global search ability (exploration) and local search ability (exploitation), when inertia weight is higher, the global search ability is strong, but the local search ability is low; whereas the local search ability is strong, but global search ability is low [20]. This balancing is a key role to improve the performance of PSO. However, the method of selecting inertia weight is not easy and need to be further investigated [21]. The linearly and nonlinearly decreasing inertia weight can make PSO adjust global search ability and local search ability, but it has shortcomings: firstly, due to the lack of local search ability at beginning of iteration and global search ability at the end of iteration, there is a possibility for to get stuck in local optima; secondly, improper selected initial inertia weight ($w_{\max}$), final inertia weight ($w_{\min}$) and nonlinear index number may decrease the performance of PSO [16,17]. Moreover, major experiments show that particles can accumulate at point in searching area, but it is not a global optimum solution. The particles have stagnated and lost the ability to find the global optima solution. Especially, in dynamic environment, so the particles will often fall in local optima. The random inertia weight can overcome these problems, but the selection of lower and upper limit of inertia weight is necessary to get the best performance [18].

Noticeably, the used strategy in previous research is still based on iteration and the performance improvement is limited because of no information about the condition of particles in population. The inertia weight is only the function of iteration for linear or nonlinear strategy and random function. In this study, we proposed a new strategy, which inertia weight is adjusted adaptively based on the swarm condition. We called it as PSO-AIW (particle swarm optimization with adaptive inertia weight) or APSO (adaptive particle swarm optimization). The basic idea of APSO is that the global best and the personal best position of particle always changes over iteration and tends to the similar value if the swarm has approached the solution. The values of *pbest* and *gbest* are taken and are used to adjust the value of inertia weight by using feedback mechanism. We need an initial inertia weight whose value is greater than one (e.g., 1.4). If the particles are too far from the solution, the fitness of *pbest* is greater than *gbest*, so the comparison between *gbest* and *pbest* is smaller than one. In this condition, the inertia weight should be set to larger value. If the particles are too close to the solution, the fitness of *pbest* is similar

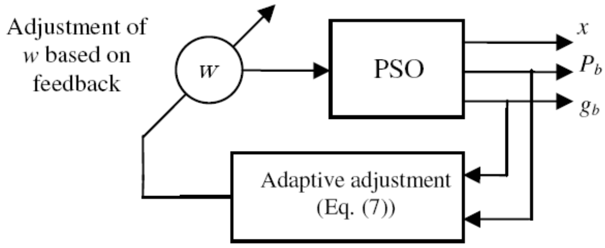

Figure 2. PSO using feedback

to *gbest*, so the comparison between *gbest* and *pbest* is close to one. In this condition, the inertia weight should be set to smaller value. Due to this strategy, the balancing between exploration-exploitation abilities can be controlled based on the swarm condition. The adaptive inertia weight is modified as:

$$w = \left\{ w_o - \frac{gb}{Pb_i} \right\} \tag{7}$$

where w_o is initial value of inertia weight. The feedback mechanism in PSO algorithm is shown in Figure 2.

4. **Implementation APSO-PID Controller.** A PID controller optimized with APSO algorithm was developed for USM. It was also called the APSO-PID controller. The transfer function of the PID controller is:

$$C(s) = K_p + \frac{K_i}{s} + K_d.s \tag{8}$$

where the K_p is proportional constant, K_i is integral constant and K_d is derivative constant. Design of APSO-PID controller for USM is shown in Figure 3. In this system, three PID parameters (K_p, K_i, K_d) will be tuned automatically by APSO algorithm. The error signal $e(k)$ will be entered for APSO and subsequently evaluated in the fitness function to guide the particles during the optimization process. The fitness function for the proposed method is given as:

$$\text{Fitness} = \frac{1}{1 + e(k)^2} \tag{9}$$

Fitness shows the following-up of evaluation function for the object input. The purpose is to decrease the steady-state error by maximizing the function. The fitness is updated by each millisecond according to the value of $e(k)$.

The USM control for clockwise (CW) rotation and the counter clockwise (CCW) rotation use the different PSO in tracking the object input. Since the characteristics of USM are different depending on the rotation direction, we evaluate both rotations separately.

4.1. **USM servo system.** The USM servo system is shown in Figure 4. USM, the electromagnetic brake and the encoder are connected on a same axis. The position information from an encoder is transmitted to the counter board embedded into a Personal Computer (PC). Meanwhile, according to error resulted from the comparison between the output and reference signal, the control input signal which is calculated in PC is transmitted to the driving circuit through the I/O board and oscillator. In each experiment, the load is added or not is discussed to observe the changes of the USM characteristics. While the voltage of 12 [V] is imported, the force of 0.25 [N.m] could be loaded to the shaft of the USM. The specification of USM, encoder and electronic brake is listed in Table 1.

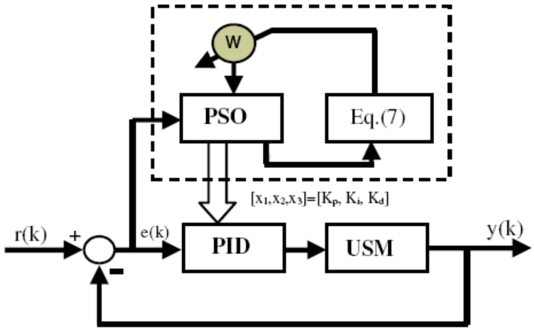

Figure 3. APSO-PID controller

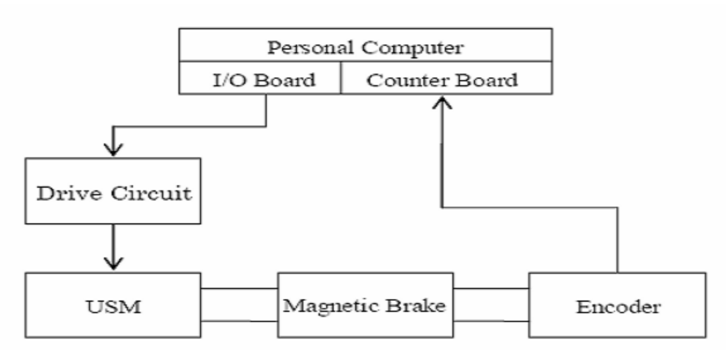

Figure 4. USM servo system

Table 1. Specification of USM servo system

USM	Rated rotational speed : 100 [rpm]
	Rated torque : 0.5 [N.m]
	Holding torque : 1.0 [N.m]
Encoder	Resolution : 0.0011 [deg]
Load	0 to 0.5 [N.m]

5. **Experimental Results.** Some experimental results are provided in this section to verify the effectiveness of the proposed APSO-PID controller for USM. We also have compared our method with previous methods, i.e., fixed-gain PID, PSO-LDW based PID, PSO-NDW based PID, and PSO-RIW based PID, with the same system condition. The reference input $r(k)$ is a rectangular signal. The amplitude is set from +45 [deg] or clockwise (CW) rotation to –45 [deg] or counter clockwise (CCW) rotation. The period is 4 [sec]. Two test conditions are provided in the experimentation, which are the unloaded condition and the loaded condition. The loaded condition is the addition of load from electronic brake with 0.25 [N.m]. Each method has been performed for 10 trials of CW rotation and 10 trials of CCW rotation. The parameters of previous PSO and APSO are set as follows: particle number, $n = 5$; cognitive constant, $c_1 = 1.0$; social constant, $c_2 = 1.0$; $w_{\max} = 0.8$; $w_{\min} = 0.3$; $w_o = 1.4$.

Firstly, the conventional fixed-gain PID must be tuned by trial-and-error method or hand-tuned method [22]. We found that the gains of PID controller are $K_p = 0.3692$; $K_i = 12.175$ and $K_d = 0.000085$. The position estimation error in histogram of USM servo system is controlled by the conventional fixed-gain PID, PSO-LDW based PID, and APSO based PID controller are shown in Figures 5-10, respectively. Each bucket of the

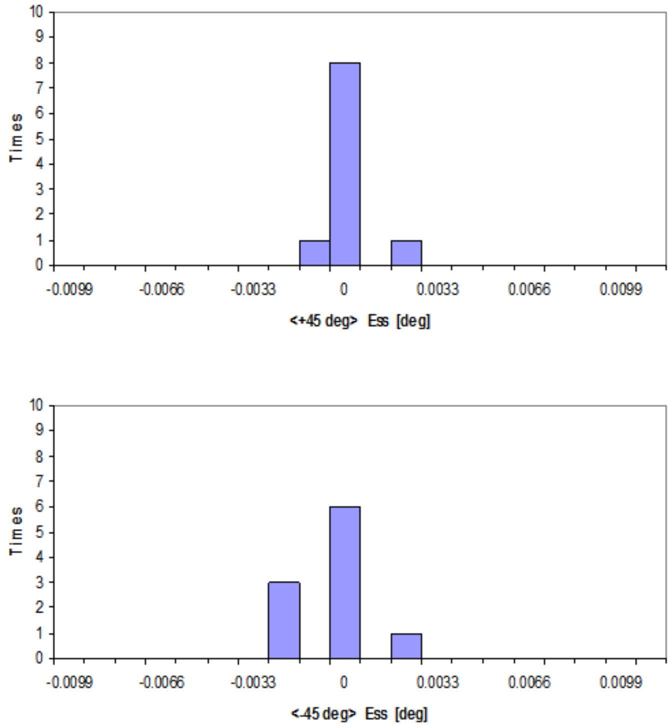

Figure 5. Position estimation error of the fixed-gain PID (unloaded)

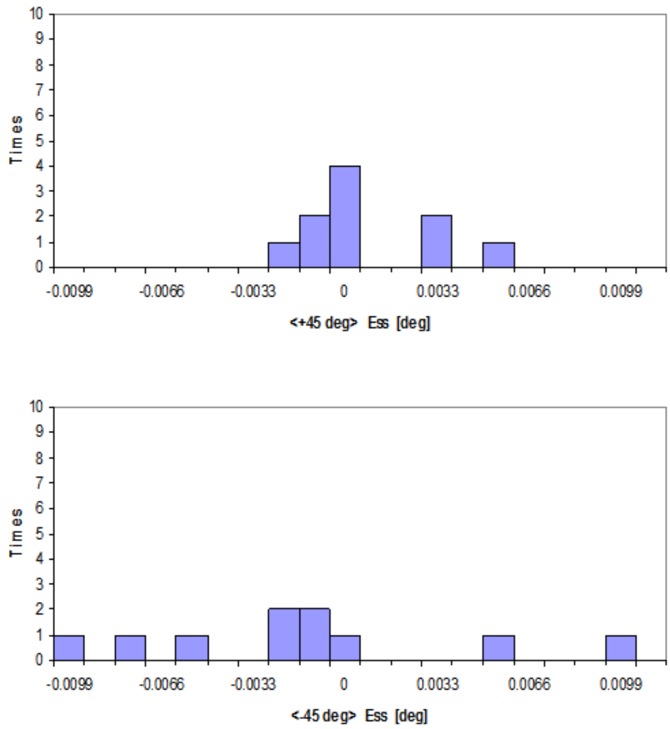

Figure 6. Position estimation error of the fixed-gain PID (loaded)

histogram is set to a width of 0.0011 [deg]. It is a resolution of the encoder. In Figures 5 and 6, it seems a high density around the zero for the unloaded condition and to be distributed uniformly (away from the zero) for the loaded condition. We can say that the conventional fixed-gain PID controller shows enough good accuracy in the unloaded condition, but becomes poor and inaccurate in the loaded condition. The determined gains are only suitable for the unloaded condition. If the plant's behavior is changed (i.e., due to the loading), it is necessary to re-tune PID and it is drawback of the fixed-gain PID. The conventional fixed-gain PID cannot compensate the characteristics changes of USM during operation. The position estimation error of USM is improved by using PSO-LDW based PID and better when using APSO based PID controller as shown in Figures 7-10, respectively. The proposed method shows a high density around the zero both the unloaded and the loaded conditions. It means that the APSO algorithm can solve the optimization problem in tuning the PID controller with better results than previous methods. The proposed APSO algorithm can improve the performance the original PSO algorithm. It can be seen clearly that self-tuning PID controller can compensate the characteristic changes of USM due to the loading effect. The gains PID are automatically adjusted according to the plant's behavior. There are characteristic differences between the CW and the CCW rotation. The position estimation error of the CW rotation is better than the CCW rotation. It may be caused by the use of one vibration source to generate the traveling wave in the stator of USM.

The statistical analysis of all methods in term of the mean of error (Ess_mean), the standard deviation of steady-state error (Ess_std), and the frequency of zero-error (Zero_Err) in 20 trials is listed in Table 2. The meaning of zero-error is the error whose value is

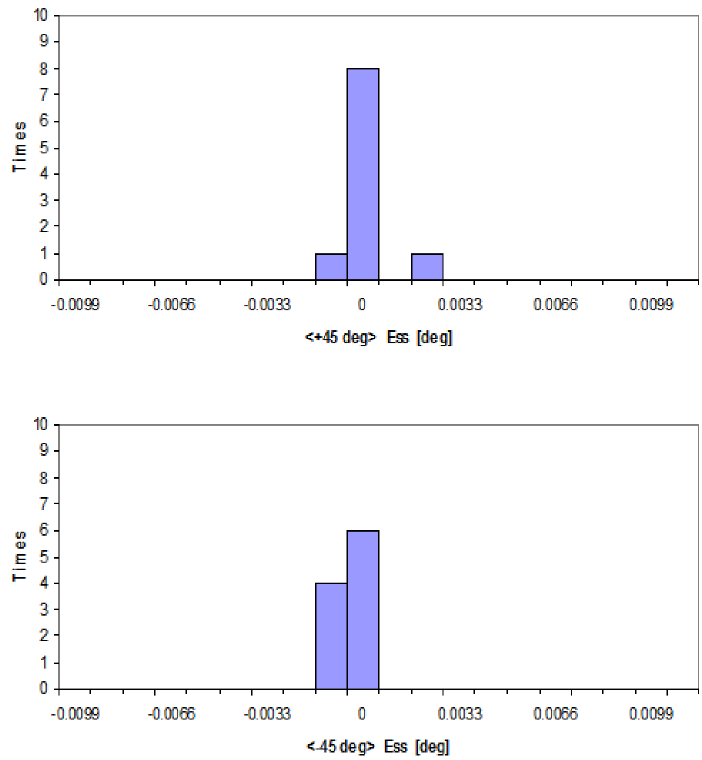

FIGURE 7. Position estimation error of the PSO-LDW based PID (unloaded)

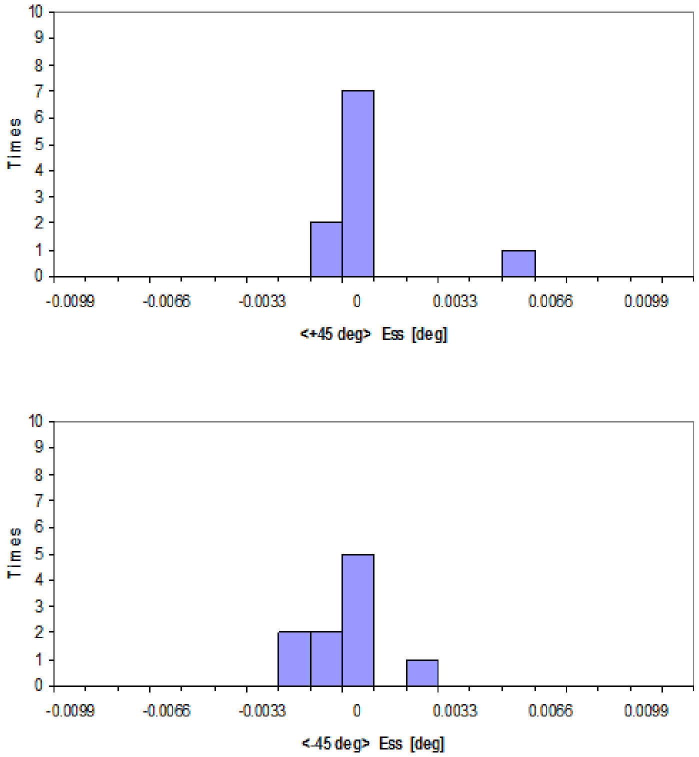

FIGURE 8. Position estimation error of the PSO-LDW based PID (loaded)

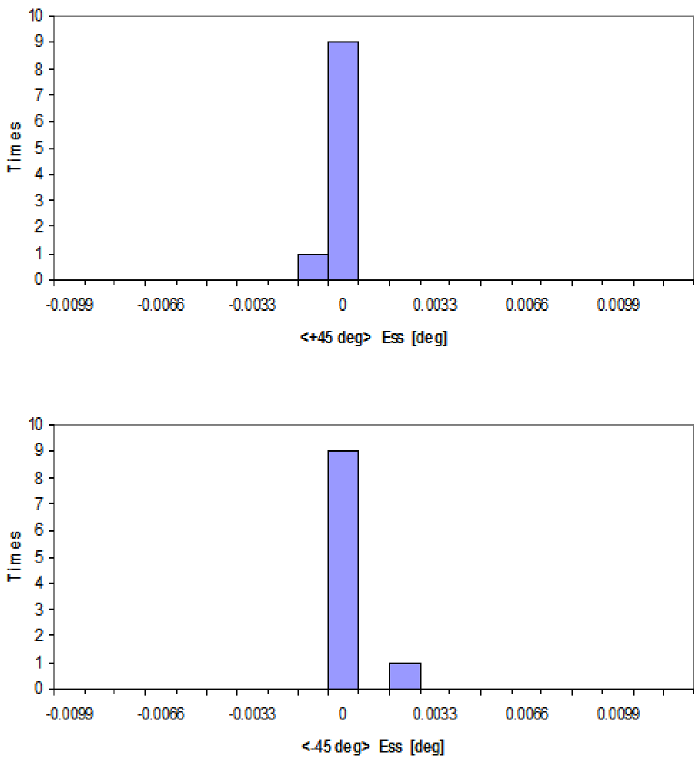

FIGURE 9. Position estimation error of the APSO based PID (unloaded)

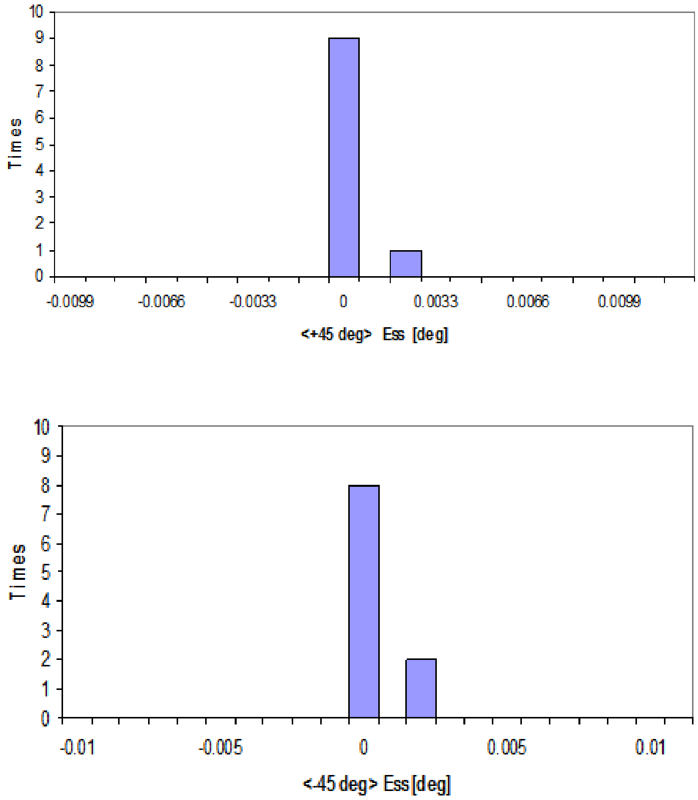

FIGURE 10. Position estimation error of the APSO based PID (loaded)

TABLE 2. Comparison results of all methods for performance test

Method	**Ess_mean**		**Ess_std**		**Zero_Err**	
	Unloaded	Loaded	Unloaded	Loaded	Unloaded	Loaded
PID	5.78E-04	3.31E-03	9.49E-04	3.62E-03	14	5
PSO-LDW-PID	4.50E-04	7.33E-04	7.46E-04	9.73E-04	14	12
PSO-NDW-PID	2.94E-04	4.28E-04	5.24E-04	5.98E-04	15	13
PSO-RIW-PID	1.83E-04	3.06E-04	4.47E-04	5.43E-04	17	15
APSO-PID	1.22E-04	1.83E-04	3.76E-04	4.48E-04	18	17

smaller than the resolution of encoder (i.e., 0.0011 [deg]). It is also called as success-rate (SR). From Table 2, it is easy to see that the proposed APSO-PID controller shows better performance in all parameters than the previous methods both the unloaded and loaded condition. The position estimation error of the proposed APSO presents a mean of 1.22×10^{-4} [deg] in the unloaded condition and 1.83×10^{-4} [deg] in the loaded condition, a standard deviation of 3.76×10^{-4} [deg] in the unloaded condition and 4.48×10^{-4} [deg] in the loaded condition, a zero-error of 18 or SR of 90% in the unloaded condition and 17 or SR of 85% in the loaded condition. According to those results, the proposed APSO algorithm may guarantee a good sense of accuracy. Compared with the original PSO-LDW-based PID controller, the proposed APSO-PID controller can increase performance up to 72.9% in the unloaded condition and 75.0% in loaded condition or 73.95% in both the unloaded and loaded condition.

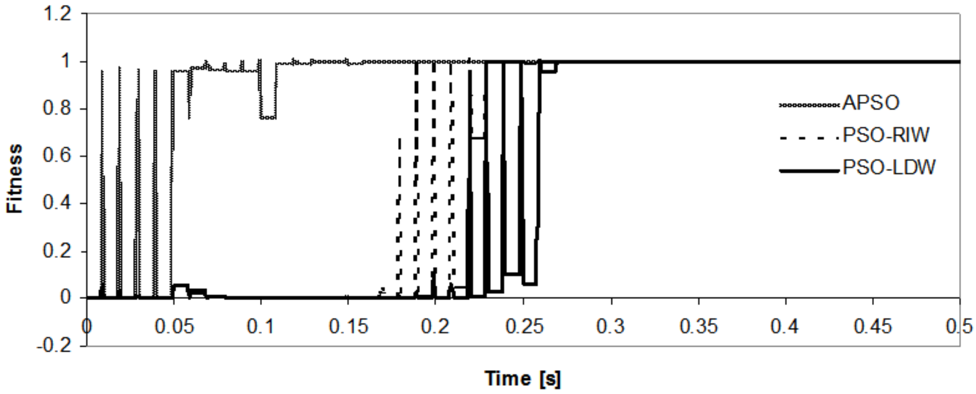

Figure 11. Fitness convergence

The fitness convergence characteristics of PSO-LDW, PSO-RIW and APSO are shown in Figure 11. It saw clearly that the particles APSO achieve faster convergence than the previous methods. The particles of APSO achieved convergence in 0.11 seconds, while the particles of PSO-RIW and PSO-LDW achieved convergence in 0.23 seconds and 0.27 seconds, respectively.

6. **Conclusions.** The adaptive PSO algorithm has been successfully applied to the optimal self-tuning PID controller in order to compensate the characteristic changes and nonlinearity of USM. The proposed APSO algorithm is motivated by the fact that it is difficult to adjust the value of inertia weight in order to achieve the proper balancing between exploration-exploitation abilities. The balancing is a key role to improve the performance of the original PSO algorithm. The proper balancing can reduce the risk of premature convergence and falling into local optima. The main advantage of our proposed method is its simplicity to overcome the weakness of the original PSO algorithm using feedback mechanism to adjust the value of inertia weight adaptively based on the swarm condition. During searching process, the personal and the global best position of particle are taken and used to calculate the value of inertia weight. By using this strategy, the proper balancing can be achieved easily. It has an important significance to engineering design optimization. The performance of our proposed method has been investigated and compared with previous methods (i.e., fixed-gain PID, PSO-LDW based-PID, PSO-NDW based-PID, and PSO-RIW based PID) by experimenting with USM servo system. We could conclude that the proposed APSO based PID method would have the superiority to the previous methods on both convergence speed and position accuracy in practical application.

Acknowledgment. The authors acknowledge the support of Directorate General of Higher Education, Ministry of National Education, Republic of Indonesia and Electronics Engineering Polytechnic Institute of Surabaya (EEPIS – ITS).

REFERENCES

[1] T. Kenjo and T. Sashida, *An Introduction of Ultrasonic Motor*, Oxford Science Publications, 1993.
[2] K. Adachi, Actuator with friction drive: Ultrasonic motor, *Journal of the JSME, IEEE International Conference on Robotic and Automation*, Barcelona, Spain, vol.108, no.1037, pp.4851, 2005.
[3] I. Yamano and T. Maeno, Five-fingered robot hand using ultrasonic motor and elastic element, *Proc. of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[4] M. Yang, M. Zhu, C. R. Robert, C. L. Martin, G. W. Peter and W. Kevin, Design and evaluation of linear ultrasonic motor for a cardiac compression assist device, *Sensor and Actuator, A: Physical*, vol.119, pp.214-220, 2005.
[5] B. Nogarede and E. Piecourt, Modelization of a traveling wave piezoelectric motor by equivalent electrical circuit, *Proc. of the International Conference on Electrical Machines*, pp.128-133, 1994.
[6] P. Hagedorn and J. Wallaschek, Traveling wave ultrasonic motors, Part 1: Working principle and mathematical modelling of the stator, *Journal of Sound and Vibration*, vol.155, no.1, pp.31-46, 1992.
[7] N. Bigdali and M. Haeri, Simplified modeling and generalized predictive position control of an ultrasonic motor, *ISA Transactions*, vol.44, pp.273-282, 2005.
[8] G. Bal, E. Bekiroglu, S. Dumirbas and I. Colak, Fuzzy logic based DSP controlled servo position control for ultrasonic motor, *Energy Convertion and Management*, vol.45, pp.3139-3153, 2004.
[9] H. Li, C. Zhao and C. Gu, Precise position control of ultrasonic motor using fuzzy control with dead-zone compensation, *Journal of Electrical Engineering*, vol.56, no.1-2, pp.49-52, 2005.
[10] T. Yoshida, T. Senjyu, M. Nakamura, N. Urasaki, T. Funabashi and H. Sekine, Speed sensorless control of ultrasonic motor using neural network, *Journal of Power Electronics*, vol.6, no.1, 2006.
[11] N. Suda, *PID Control*, Asakura Publishing Co., 1992.
[12] S. Yamamoto and N. Katou, *Base and Application of PID Control*, Asakura Publishing Co., 1997.
[13] Z. Sun, R. Xing, C. Zhao and W. Huang, Fuzzy auto-tuning PID control of multiple joint robot driven by ultrasonic motors, *Ultrasonic*, vol.46, pp.303-312, 2007.
[14] K. Tanaka, Y. Wakasa, T. Akashi and M. Oka, PI control adjusted by GA for ultrasonic motor, *Electrical Engineering in Japan*, vol.9, no.1, pp.663-668, 2009.
[15] F. Cheng, K.-C. Fan, J. Miao, B.-K. Li and H.-Y. Wang, A BPNN-PID based long-stroke nano-positioning control scheme driven by ultrasonic motor, *Precision Engineering*, vol.36, pp.485-493, 2012.
[16] F. A. Rahman, Y. Murata, K. Tanaka, Y. Nishimura, S. Uchikado and Y. Osa, Variable gain type PID control using PSO for ultrasonic motor, *The 5th International Workshop on Computational Intelligence and Application IEEE SMC Hiroshima Chapter*, Hiroshima University, 2009.
[17] A. Djoewahir, K. Tanaka and S. Mu, A modified particle swarm optimization with nonlinear decreasing inertia weight based PID controller for ultrasonic motor, *International Journal of Innovation, Management and Technology*, vol.3, no.3, pp.198-201, 2012.
[18] A. Djoewahir and K. Tanaka, Self-tuning PID controller based on PSO-RIW for ultrasonic motor, *International Journal of Engineering Innovation and Management*, vol.2, pp.33-40, 2012.
[19] J. Kennedy and C. Eberhart, Particle swarm optimization, *Proc. of IEEE International Conference on Neural Networks*, pp.1942-1945, 1995.
[20] Y. Shi and R. C. Eberhart, A modified particle swarm optimization, *IEEE Int. Conf. Evol. Comput.*, Anchorage, Alaska, 1998.
[21] I. C. Trelea, The particle swarm optimization algorithm: Convergence analysis and parameter selection, *Information Processing Letters*, vol.85, pp.317-325, 2003.
[22] G. Ellis, *Control System Design Guide*, Academic Press, London, 1991.