SICE-ICASE International Joint Conference 2006
Oct. 18-21, 2006 in Bexco, Busan, Korea

Development of Golden Section Search Driven Particle Swarm Optimization and its Application

Sehoon Oh[1] and Yoichi Hori[1]

[1]Institute of Industrial Science, University of Tokyo, Tokyo, Japan
(Tel : +81-3-5452-6289; E-mail: sehoon@horilab.iis.u-tokyo.ac.jp)
[2]Institute of Industrial Science, University of Tokyo, Tokyo, Japan
(Tel : +81-3-5452-6289; E-mail: hori@iis.u-tokyo.ac.jp)

Abstract: The Particle Swarm Optimization (PSO), although it has been widely used in various fields, has a step-size problem, which deteriorates optimization performance. This problem is resolved using the Golden Section Search (GSS) and the Steepest descent method. We also design a filter that will improve optimization performance of the proposed algorithm. The effectiveness of the proposed algorithm, including for which type of problems the proposed algorithm is adequate, is verified using some benchmark problems. Moreover, a hardware-in-the-loop system which consists of a NC system with two motors and a computer that optimizes control parameters of the NC controller is introduced as an industrial application.

Keywords: particle swarm optimization, parameter tuning, golden section search, steepest descent method, precision motion control, hardware-in-the-loop

1. INTRODUCTION

Recently, optimization algorithms have been used for control parameter optimizations[1],[2]. This paper suggests a novel search algorithm which improves performance of the conventional algorithms used for this parameter optimizations.

1.1 Search Algorithms used for Optimization of Control Parameter

Using various search algorithms, large numbers of reseaches optimize a variety of parameters in controllers: optimization of gains and orders of controller, estimation of physical characteristics of a plant. Among the algorithms, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are the most popular methods.

GA adopts selection, cross-over, and mutation as its optimization tools. It is an analogy of the optimization procedure of the nature and easy to understand. Yet, it has several problems : GA largely depends on the initial parameters' value and it can lead to a failure in optimization; The update of parameters is mainly done by cross-over which cannot get over the range of initial parameters' code. Mutation is the only way to get over the range, however, the direction of mutation is quite random, and it takes too much time to reach the optimum by that mutation.

On the other hand, PSO which uses the concept of velocity described in Equation (1) can easily get over the range of initial parameters.

$$\begin{aligned} v_i(k+1) &= \alpha v_i(k) \\ &\quad +\beta_1 \mathrm{rand}(x_{pbest_i} - x_i(k)) + \beta_2 \mathrm{rand}(x_{gbest} - x_i(k)), \end{aligned} \tag{1}$$

where x_{gbest} means the position of a global best obtained so far by any particle in the population, and x_{pbest_i} means the position of the best solution an i th particle has achieved so far. The positions or parameters of particles in next generation are determined by adding this velocity to the positions of particles in current generation. By choosing appropriate coefficients α, β_1 and β_2, the range in which next particles search around can be adjusted. This is a big difference between GA and PSO. Also, this velocity concept is quite useful in the multi-dimensional optimization problems which do not have any mathematical model. It enables the particle to move toward the optimum with high possibility of getting there.

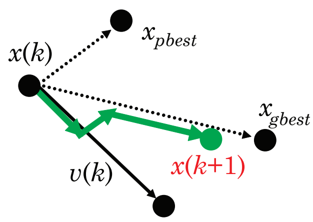


Fig. 1 Illustration of a Position Update in PSO

The only problem of this PSO is that the step size in each update is quite random, which decreases the possibility of finding the optimum and makes the convergence more or less slow.

This paper suggests quite a new idea on the deciding of the step size in PSO based on the golden section search and the steepest descent method. This new algorithm is proposed in Section 2.and verified by numerical experiments in Section 3.

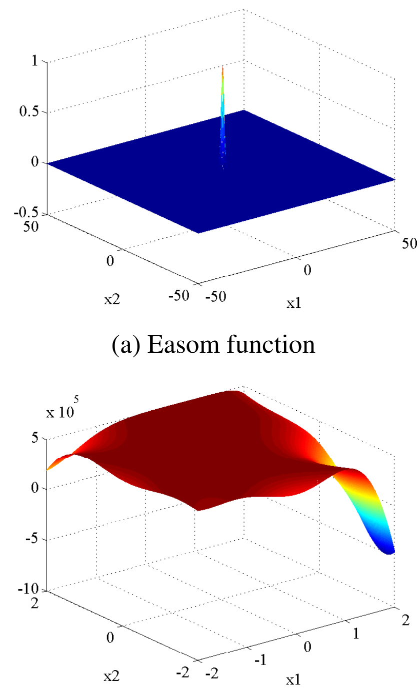
1.2 Parameter Tuning Problem in Controller Design

In industry, parameters of controllers should be tuned according to the purpose and environment of the system. Especially in the high precision control, this tuning is difficult and time-consuming work for general users and has been the work of sophisticated experts. The search algorithm, however, can be a good alternative to this experts's work. In order to apply the search algorithms as auto-tuning methods for control parameters, a hardware-in-the-loop system is employed in this paper. An NC system which is composed of motors, NC controller and servo amplifier produces a value that represent the performance of controller, and a search algorithms optimizes control parameters based on that value. The whole configuration of experimental setup is explained in Section 4. and its preliminary result is shown in 4.4

The parameter optimization of this NC system has some characteristics as below.

1. Although design of a function that indicates the performance of controller is not uniquely determined, it is not likely to have lots of local optima, and roughly can be considered as a unimodal function.
2. Since it uses real hardware such as motors, it should not search a parameter space where the hardware can be broken during parameter searches. Mutation in the GA is hard to be favored in this application. Additionally, parameters should be optimized with fewer experiments.
3. It needs fine adjustment of parameters. The algorithm should find excellent parameters, not fairly good parameters. Therefore, the algorithm should find the possible search space, search that space in detail, and find excellent parameters quickly. This means that it should have a strategy to search a space after it comes near the optimal point.

These characteristics are assumed based on the experience of the control experts. A novel PSO algorithm that is appropriate for these characteristics is proposed in next section.



(a) Easom function

(b) GP function

Fig. 2 Two Benchmark Problems in Two Dimension

2. SUGGESTION OF NOVEL PSO ALGORITHM

2.1 Problem in Conventional PSO Algorithm

As introduced in the previous section, PSO has a step-size problem. This results in frequent failure in a certain kind of optimization problem. Two benchmark problems explain these characteristics. It is true the α, β_1, β_2 in Equation (1) are factors that decide that step size [3]. However, those factors adjust just the trends of step size and there remains considerable room for progress.

$$ES(x_1, x_2) = -\cos(x_1)\cos(x_2)\exp(-((x_1-\pi)^2+(x_1-\pi)^2)) \quad (2)$$

$$GP(x_1, x_2) = \Big(1+ (x_1+x_2+1)^2\,(19-14x_1+3x_1^2-14x_2 +6x_1x_2+3x_2^2)\Big)\Big(30+(2x_1-3x_2)^2(18-32x_1+12x_1^2 +48x_2-36x_1x_2+27x^2)\Big) \quad (3)$$

The Easom (ES) function in Equation (2) and the Goldstein and Price (GP) function in Equation (3) are adopted to verify the characteristics of PSO. The ranges of each parameter are set from -50 to 50 in the ES function and -2 to 2 in the GP function. (π, π) and $(0, -1)$ are the global optimum of two functions. Figure 2 illustrates that there is a big difference between two functions; The slope around the optimal point of the ES function is quite steep, on the other hand, the slope of the GP function is smooth.

This difference makes a difference in the performance of PSO optimization. To see this, numerical experiments are conducted; Success of optimization is defined and the rates of that success in 2000 trials are compared. If the optimized point by PSO is around the global optimal point with the precision as below, the optimization is counted as a success.

$$\|x_{\text{global best}} - x_{\text{best by PSO}}\| < \frac{\text{the full range}}{50} \quad (4)$$

The full range is 100 in the ES function and 4 in the GP function. The population size is 10, and the generation size is 20. α is 0.9, β_1 is 0.4, and β_2 is 0.9.

The average success rate of the ES function was 60.1% and the rate of the GP function was 37.3 %. We can easily assume that the lower success rate of the GP function is due to the slope around the optimal point. Since the PSO is good at finding the direction toward optimal point, it has more possibility to find that point in the problem where the optimal point has steep slope around the optimal point. Meanwhile, where there is smooth slope around the optimal point, it is highly likely that the PSO algorithm just searches around the global optimal position and has difficult stopping at the position.

2.2 Suggestion of Golden-section-search-driven PSO Algorithm

Several improvements have been made for the PSO algorithm[3], [4]. Strictly speaking, however, they are not an improvement of PSO, but mixtures with several algorithms as they use more particles in one generation to hybridize algorithms. The novel algorithm proposed in this section just changes the strategy of one or two particles in one generation and attempts to make improvement by those particles.

In the application to the hardware-in-the-loop system, the value to be optimized is obtained by a real experiment using a hardware. Since this experiment takes time, the size of a swarm should be less. For this reason, it can be said that our proposed method is more favorable for auto-tuning of parameters of an NC controller.

In the PSO, decision of the direction and step size for the update of a generation is important. We adopt the steepest descent method for the direction decision and golden section search for step size decision. In the whole procedure of the PSO, one direction is selected as a candidate direction where the global optimum is assumed to be located. The direction in which the slope to the x_{gbest} has been steepest so far is selected as the candidate direction, therefore, when x_{gbest} is changed or there comes steeper slope, this direction is reset.

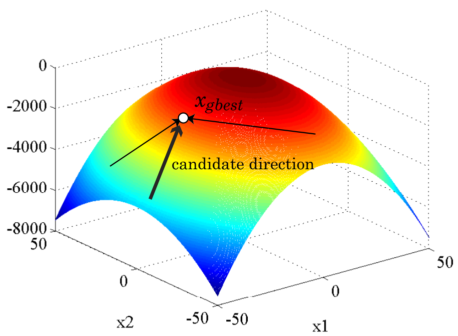


Fig. 3 Decision of the Candidate Direction

After the candidate direction is fixed, the position of one or two particle is selected along this direction with the step size decision by the golden section search. For the golden section search, four positions in one direction are necessary. Let us denominate these four points as GSS points and represent them as $x_{\text{gss}1}, x_{\text{gss}2}, x_{\text{gss}3}, x_{\text{gss}4}$.

The distances among these GSS points are designed as below.

$$\begin{aligned} \|x_{\text{gss}1} - x_{\text{gss}3}\| : \|x_{\text{gss}3} - x_{\text{gss}4}\| &= \lambda : 1-\lambda \\ \|x_{\text{gss}1} - x_{\text{gss}2}\| : \|x_{\text{gss}2} - x_{\text{gss}4}\| &= 1-\lambda : \lambda, \quad (5) \end{aligned}$$

where λ is the golden ratio calculated by $\frac{\sqrt{5}-1}{2}$.

Based on the values of the performance function of these four points, the GSS points in the next generation ($x'_{\text{gss}1}, x'_{\text{gss}2}, x'_{\text{gss}3}, x'_{\text{gss}4}$) are decided reducing or shifting the range of $x_{\text{gss}1}$ to $x_{\text{gss}4}$. How to move the range is determined by how the values of a performance function are located. Figure 4 describes the possible placement types of the function of these four GSS points.

Type I and Type III are the well-known function-value placement in the golden section search where the optimal point is located between the two terminal points

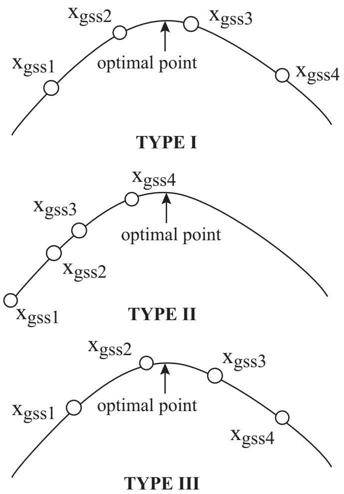


Fig. 4 Four-types Function-value Placement GSS Points

$x_{\text{gss}1}, x_{\text{gss}4}$. In these cases, the position of the GSS points are updated as Equation (6) and (8).

$$\begin{aligned} x'_{\text{gss}2} &= \lambda x_{\text{gss}2} + (1-\lambda)x_{\text{gss}4}, \quad x'_{\text{gss}1} = x_{\text{gss}2} \\ x'_{\text{gss}3} &= x_{\text{gss}3}, \quad x'_{\text{gss}4} = x_{\text{gss}4} \quad (\text{TYPE I}) \quad (6) \\ x'_{\text{gss}3} &= (1-\lambda)x_{\text{gss}1} + \lambda x_{\text{gss}3}, \quad x'_{\text{gss}4} = x_{\text{gss}3} \\ x'_{\text{gss}1} &= x_{\text{gss}1}, \quad x'_{\text{gss}2} = x_{\text{gss}2} \quad (\text{TYPE III}) \quad (7) \end{aligned}$$

This update in Type I and III reduces the range of search space. Since only a $x'_{\text{gss}2}$ or $x'_{\text{gss}3}$ is a new position, this position replace only one particle in the next generation; that is, if the population size of one generation in PSO is N, only one particle is replaced with this new GSS position, and the other $N-1$ particles are updated by the PSO algorithm. This newly inserted GSS postion has much possibility to be near the optimal point, it can improve the whole performance of PSO algorithm.

Type II is a rare case in the general golden section search. However, it will frequently happen in the proposed algorithm because we do not know where the optimal point is. In this case II, we assume that the optimal point is outside the current range, and therefore the range should be shifted to the direction in which the optimal point is likely to be. This shift is given as

$$\begin{aligned} x'_{\text{gss}4} &= -\frac{(1-\lambda)}{\lambda}x_{\text{gss}2} + \frac{1}{\lambda}x_{\text{gss}4}, \quad x'_{\text{gss}1} = x_{\text{gss}2} \\ x'_{\text{gss}2} &= \lambda x'_{\text{gss}1} + (1-\lambda)x'_{\text{gss}4}, x'_{\text{gss}3} = x_{\text{gss}4} (\text{TYPE II}) \\ & \quad (8) \\ & \quad (9) \end{aligned}$$

This update uses the exterior division to make the ratio among the GSS points the golden ratio.

Figure 5 is the whole procedure of the proposed optimization algorithm. This algorithm is more or less complicated than the original PSO, however, in the application to the HIL system, the calculation time is not a significant problem because the experiment takes much longer time than the calculation.

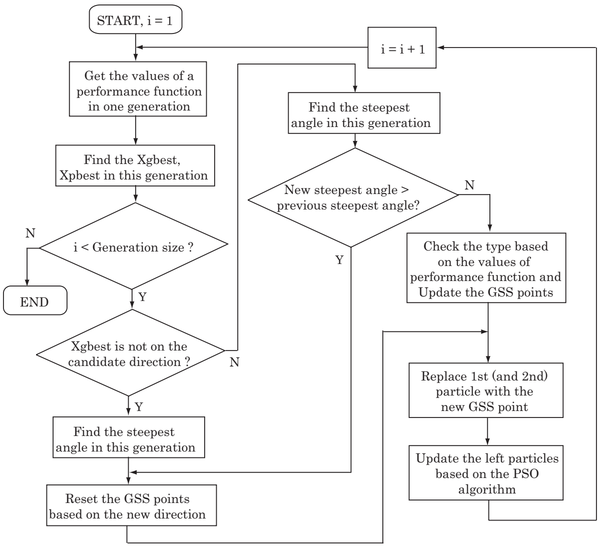


Fig. 5 Flowchart of Proposed Algorithm

3. VERIFICATION OF THE EFFECTIVENESS OF THE PROPOSED ALGORITHM

The benchmark problems tested in Section 2.1 are used again with the same parameter setting, to verify the effectiveness of the proposed algorithm.

3.1 Numerical Experiments with Two-dimensional Benchmark Problems

The proposed algorithm is applied to the ES and GP problems, and the success ratio is calculated. In the ES problem, the success ratio in 2000 trials is improved to 64.7% from 60.1% To compare more in detail, The average error from the optimal position and its variance is compared. The average error is 0.243, and the covariance is 3.56 in the original PSO. On the other hand, in the proposed PSO, 0.135 is the average error, and 1.18 is the covariance. From this we can verify that not only the success rate but also the quality of the optimization is improved by the proposed algorithm.

However, the proposed method proved not to be so effective in the GP problem. It just changes the success ratio from 39.2% to 39.3%. The average error from the optimal point is 0.0114 in the original PSO and 0.0108 in the proposed PSO. The covariance is 0.00495 in the original PSO and 0.00455 in the proposed PSO. The values are almost same, and it is certain that the proposed method can not improve this kind of optimization. In the GP problem, due to the smooth slope around the optimal point, the steepest descent method can not work so well, moreover, this is a two-dimensional problem which is said to be not good for the steepest descent method. This can be one reason.

3.2 Effectiveness of the Proposed Algorithm

In the last section, we verified that the proposed algorithm becomes effective in the problem where the slope around the optimal point is steep.

To fully utilize these characteristics, we adopt a filter which molds the value of a performance function. A filter which can bend the function space without changing the ordinal relationship between the values of a function can improve the optimization performance. In this paper, an exponential filter in Equation (10) is employeed.

$$f_{new} = \exp(p \cdot f_{original} + q) \tag{10}$$

p and q are arbitrarily deciede according to the values of $f_{original}$. Table 1 shows the whole results of numerical experiments. Adding to the previous four numerical experiments, another two experiments with the filter is conducted in the GP function.

Table 1 Results of Numerical Experiments

Func.	Method	SR	Error	Variance
ES	Org. PSO	60.1	0.243	3.56
ES	Prop. PSO	64.7	0.135	1.18
GP	Org. PSO	39.2	0.0114	0.00495
GP	Prop. PSO	39.3	0.0108	0.00455
GP	Org. PSO w/ filter	36.8	0.0111	0.00443
GP	Prop. PSO w/ filter	53.5	0.0083	0.00364

The last two results show that the filter in Equation (10) improves the performance. Not only the success ratio but also the error and variance are improved by adopting this filter. It can be said that this kind of bending space can result in wrong optimization where there are quite similar local optimal points, since those values also will be emphasized by this filter. However, although the GP and ES function have several local optima too, the proposed method shows an excellent optimization performance with this size of local optima. Consequently, this filter should be employed taking the characteristics of a target problem into consdireation.

3.3 Numerical Experiments with a High Dimensional Benchmark Problem

Poor performance in the GP function can be due to the small dimension. In order to demonstrate the performance in high dimensional problems, the proposed algorithm is applied to the Shekel function of which dimension is four[3]. The results are listed in Table 2.

Table 2 Results in the Shekel Function

Func.	Method	SR	Error	Variance
Shekel	Org. PSO Tr.1	23.8	0.192	9.57e-5
Shekel	Org. PSO Tr.2	23.07	0.206	1.51e-4
Shekel	Prp. PSO Tr.1	26.94	0.185	2.34e-4
Shekel	Prp. PSO Tr.2	24.93	0.186	1.67e-4
Shekel	Prp. PSO Tr.1 w/ filter	26.57	0.184	7.34e-05
Shekel	Prp. PSO Tr.2 w/ filter	28.95	0.181	8.51e-05

This results verifies effectiveness of the proposed algorithm in high dimensional problems. The filter in Equation (10) also improved the performance.

With all experimental results, the proposed algorithm is not effective for all optimization problem. In problems where there are big globla optima, the algorithm can easily fails to find the global optimal. However, the algorithm is proved to be quite useful to other problems.

4. APPLICATION TO THE HARDWARE-IN-THE-LOOP SYSTEM

In order to verify the effectiveness of the proposed algorithm in industrial applications, an experiment using a hardware-in-the-loop (HIL) system is introduced.

4.1 The Hardware-in-the-loop system Used for Experiments

This HIL system is set up in an attempt to build an auto-parameter-tuning of a NC system. Figure 6 is the configuration of the HIL system used in this research.

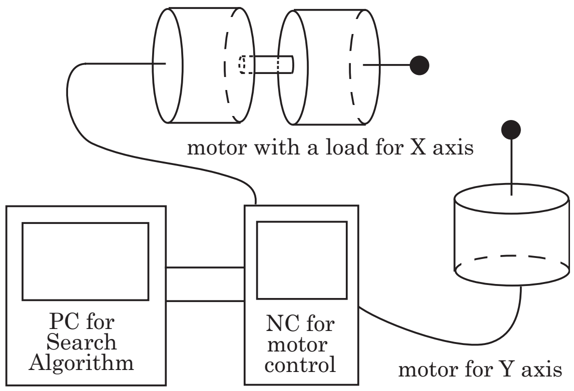


Fig. 6 Hardware-in-the-loop System for Experiments

The NC system has two motors to conduct two-dimensional motion. The NC controller and motors are one closed system; NC controller has all control parameters in it, and it also measures necessary information on motor motions. This closed NC system is attached to a computer that will optimize the control parameters using search algorithms. The performance function to be optimized is obtained through the real experiments by hardware; this is the main purpose of this proposed HIL system.

The computer that is connected to the NC system obtains a result of one experiment calculates the performance function based on the measurements. It also generates the next swarm according to the proposed algorithm based on this calulated value. The generated swarm is fed back to the parameters of the NC controller.

4.2 Target Motion

As an object of the optimization, a two-dimensional trajectory described in Figure 7 is chosen. This rectangle with the four arc corner is said to be difficult trajectory to be tuned as it deals with the timing problem between two axes.

Minimization of the trajectory is one purpose of the parameter tuning. Figure 8 shows an example of trajectory error in one trial. There are four groups of peaks where the error increases drastically. These periods correspond to the time when the trajectory changes its direction. These peaks should be decreased.

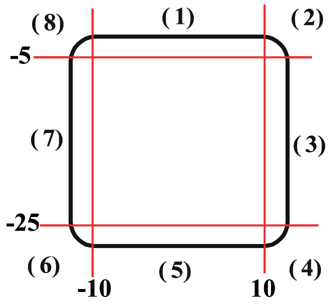


Fig. 7 Target Trajectory: Rectangle with the Four Arc Corner

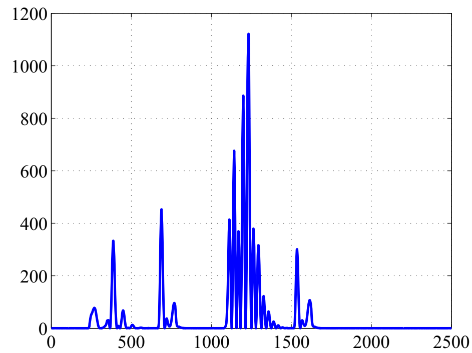


Fig. 8 Trajectory Error in One Trial

The other purpose is reduction of the elapsed time to draw the target trajectory. As a result, the performance function is defined as follows. Let us call this function as the fitness.

$$\text{Fitness} = \frac{K_{err}}{\sum \text{trajectory error}(t)} + \frac{K_{time}}{\text{Elapsed time}} \tag{11}$$

4.3 Adjustable Control Parameters to Be Optimized

The control parameters to be optimized should be strongly linked with the control performance that is represented in the fitness. As a first step, this research adopts four control parameters in order to optimize the fitness defined above.

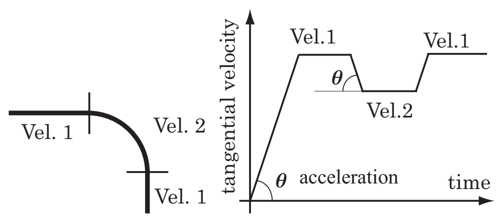


Fig. 9 Changes in the Velocity Tangential to the Trajectory

Figure 9 describes a reference velocity profile which is tangential to the trajectory. Vel.1 is the velocity of a motor when the point is located on the side of the rectangle. Vel.2 is the velocity when the point is on the corner. On the corner the tangential velocity is limited less than

Vel.1 to suppress the acceleration in the normal direction which can lead to the vibration after escaping the corner. This limited velocity Vel.2 is a parameter to be optimized in this research.

To satisfy this velocity limitation, a motor should decrease its velocity before entering the corner and increase the velocity after the corner. The magnitude of acceleration/deceleration that is represented as the angle θ in Figure 9 is another parameter to be optimized. This angle is set separately in each motor.

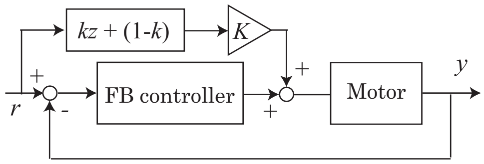


Fig. 10 Adjustment of Feedforward Control Input Timing

The last parameter we consider is the timing of feedforwad control input. This timing can be illustrated as k in Figure 10. As the target motion is two-dimensional, timing between two motors becomes an important factor to realize precise trajectory. k is an effective parameter which decides this timing.

Consequently, we have four parameters as objects of optimization.

- limit of tangential velocity in the corner (Vel.2 in Fig. 9)
- magnitudes of acceleration in two motors (θ in Fig. 9)
- timing of feedforward control in one motor (k in Fig. 10)

These four parameters are optimized using the proposed algorithm in next section.

4.4 Preliminary Experimental Results

The algorithm proposed in Section 2is applied to the HIL system. Figure 11 is the calculated trajectory errors and corresponding fitness. The results with the fitness 1.395 and 4.88103 are obtained with the first randomly chosen parameters. The result with the fitness 14.4216 is obtained by the optimized parameters. Especially the first and the third peak groups are reduced by the optimization. The second and fourth peak groups are not improved. This means the four parameters we used is not enough and cannot affect these two peaks. Zoomed error shows that the optimized parameters reduces the last overshoot.

These are preliminary results. To choose another function as fitness and applying a filter to this fitness can be applied to improve the performance. These are future works.

5. CONCLUSION

A novel PSO algorithm was proposed in this paper and verified by several numerical experiments. This algorithm is developed to be applied to auto-tuning of the NC control parameters. The relationship between the control performance and the parameters to be optimized and how to find an appropriate fitness function should be more clarified, which can make more effective performance function. It is necessary to verify the effectiveness of the proposed algorithm by comparing with the results of other algorithms. These are our future works.

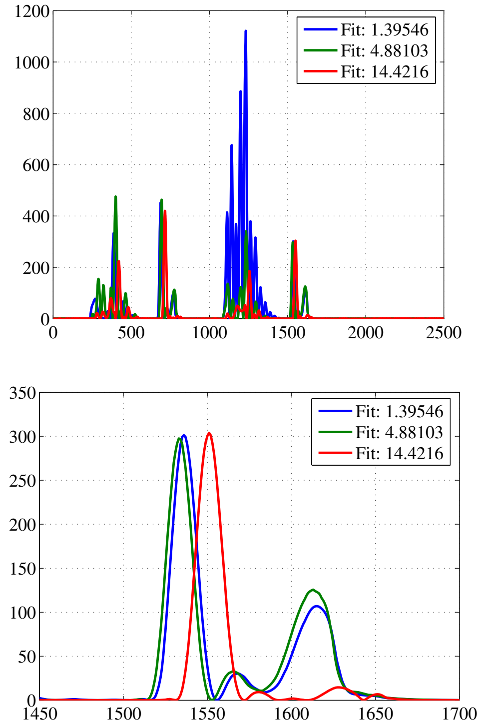


Fig. 11 Trajectory Error Changes

REFERENCES

[1] Ito, K., Iwasaki, M., Matsui, N., “GA-based practical compensator design for a motion control system,” *IEEE/ASME Transactions on Mechatronics,* Vol. 6, No. 2, pp.143-148, 2001.

[2] Byunghoon Chang and Yoichi Hori “Research related to the Parameter Auto-tuning of Two Mass Control System,” (in Japanese) *IEE of Japan Technical Meeting Record*, IIC-05-47, 2005.

[3] Hshu-kai S. Fan, Yun-chia Liang and Erwie Zahara, “Hybrid Simplex Search and Particle Swarm Optimization for the Global Optimization of Multimodal Functions,” *Engineering Optimization*, Vol. 36, No. 4, pp.401-418, 2004,

[4] Yu Liu, Zheng Qin, Zhewen Shi, “Hybrid particle swarm optimizer with line search,” *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 4, pp.3751 - 3755, 2004.