Parameter Optimization for NC Machine Tool Based on Golden Section Search Driven PSO

Sehoon Oh and Yoichi Hori
Institute of Industrial Science
University of Tokyo
4-6-1, Komaba, Meguro, Tokyo, 153-8505 Japan
Email: sehoon@horilab.iis.u-tokyo.ac.jp, hori@iis.u-tokyo.ac.jp

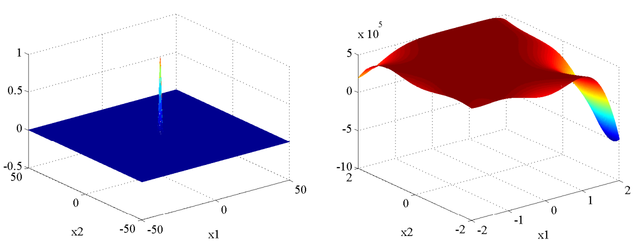
***Abstract*— We have proposed a modified PSO[1]; GPSO (Golden-section-search driven Particle Swarm Optimization) which updates only one particle in a generation based on a strategy: golden section search and steepest descent method. It was proved to be effect in various optimization problem. In this paper, first, this GPSO is revised to make clear its effectiveness. Then, the GPSO is utilized to optimize control parameters in NC machine tools. Parameters which are said to be difficult to optimize in a NC machine tool, is chosen and the roles of those parameters are scrutinized. Based on those scrutiny, fitness are defined for parameters.**

In order to verify optimization performance of the algorithms (GA, PSO, GPSO), a hardware-in-the-loop system with a NC machine tool is set up and on-line optimization experiments are conducted using the system. In experiments, the GPSO shows better optimization performance.

Key Words* : *golden section search driven particle swarm optimization (GPSO), hardware-in-the-loop system, parameter tuning, golden section search, steepest descent method, precision motion control

I. Introduction

We have proposed a modified PSO that is called golden section search driven particle swarm optimization (GPSO)[1]. In the last work, the performance of GPSO was verified by numerical experiments. As a result, it was proved that the GPSO is more appropriate for the problem which has a steep slope around the global optimal point. For example, the GPSO shows better performance in the problem (a) Easom function than (b) GP function illustrated in Figure 1.



(a) Easom function (b) GP function

Fig. 1. Two Benchmark Problems in 2 Dimensions

In our previous work, the algorithm sometimes needs two swarms to be updated by the golden ratio section search. Since this can worsen the performance, the algorithm is revised in this paper.

Also in the last work, GPSO has been suggested as one efficient optimization method of on-line control parameter optimization. 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. Our last paper suggested the characteristics of control parameter optimization in NC machine tools as below, based on the experience of the control experts:

1) Although design of a fitness 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.

Considering these characteristics, the GPSO's effective optimization performance can be said to be adequate for this optimization, since it will reduce tuning time with less size of a swarm.

However, the more related with performance the parameters are, the higher the possibility of optimization will be. In the last paper, fitness function is not so well scrutinized resulting in unclear performance improvement by the GPSO. In order to make this point clear, this paper focuses on the relationship between parameters and control performance.

Lastly, the experiments using the hardware-in-the-loop system compare the optimization perforamces of three algorithms: GA, PSO, GPSO, and verify superiority of the proposed GPSO algorithm.

II. Golden section search driven Particle swarm optimizer

A. Necessary Improvement in Conventional Optimization Methods

Various search algorithms have been used for optimization problem in high precision control [3],[4]. Those algorithms 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.

$$v_i(k+1) = \alpha v_i(k) + \beta_1 \text{rand}(x_{pbest_i} - x_i(k)) + \beta_2 \text{rand}(x_{gbest} - x_i(k)), \quad (1)$$

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 which 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.

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. The GPSO addressed this step size problem in PSO based on the golden section search and the steepest descent method.

B. Suggestion of Golden section search driven PSO Algorithm

The GPSO novel algorithm proposed in this section just changes the strategy of one particle in one generation and attempts to make improvement by those particles.

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.

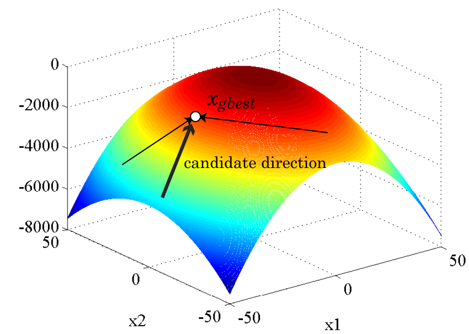


Fig. 2. Decision of the Candidate Direction

After the candidate direction is fixed, the position of one 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 $x_{\text{gss1}}, x_{\text{gss2}}, x_{\text{gss3}}, x_{\text{gss4}}$.

The distances among these GSS points are designed as below.

$$\begin{aligned} \|x_{\text{gss1}} - x_{\text{gss3}}\| : \|x_{\text{gss3}} - x_{\text{gss4}}\| &= \lambda : 1-\lambda \\ \|x_{\text{gss1}} - x_{\text{gss2}}\| : \|x_{\text{gss2}} - x_{\text{gss4}}\| &= 1-\lambda : \lambda, \end{aligned} \quad (2)$$

where λ is the golden ratio calculated by $\frac{\sqrt{5}-1}{2}$. This golden ratio is the most efficient bracket ratio to narrow the searching length[2]. The golden section search (GSS) will reduce the bracket length by changing the upper bound x_{gss4} or the lower bound x_{gss1} to x_{gss3} or x_{gss2}, based on the values evaluated at x_{gss2} and x_{gss3}. A new point to be chosen after reducing the length, will be located keeping the golden section relationship in Equation (2) with the residual three points. This repetitive selection of a new point results in finding the extremum.

Note that for each iteration, only one new point have to be constructed and only one new evaluation, have to be made. This is why the GSS algorithm is appropriate to enhance the PSO algorithm. Only one agent in the whole population will be updated based on the GSS algorithm, while the others are updated by the PSO algorithm. Moreover, the point driven by the GSS moves in the most likely direction for the extremum with the most efficient step size, addressing the step size problem in the PSO algorithm.

Figure 3 shows how to contract the searching area, where the optimal point is located between the two terminal points $x_{\text{gss1}}, x_{\text{gss4}}$. In these cases, the position of the GSS points

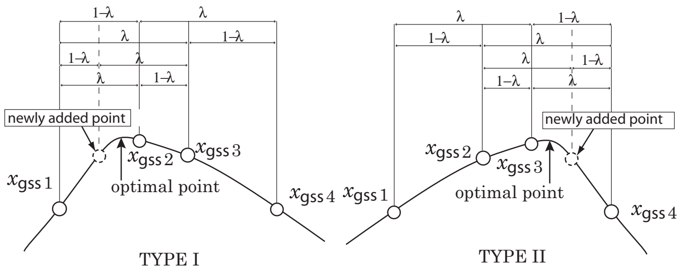


Fig. 3. Contraction by Golden Section Search

are updated as Equation (3) and (4).

$$x'_{\text{gss}2} = \lambda x_{\text{gss}2} + (1-\lambda)x_{\text{gss}4}, \quad x'_{\text{gss}1} = x_{\text{gss}2}$$
$$x'_{\text{gss}3} = x_{\text{gss}3}, \quad x'_{\text{gss}4} = x_{\text{gss}4} \quad \text{(TYPE I)} \tag{3}$$
$$x'_{\text{gss}3} = (1-\lambda)x_{\text{gss}1} + \lambda x_{\text{gss}3}, \quad x'_{\text{gss}4} = x_{\text{gss}3}$$
$$x'_{\text{gss}1} = x_{\text{gss}1}, \quad x'_{\text{gss}2} = x_{\text{gss}2} \quad \text{(TYPE II)} \tag{4}$$

Updates of the GSS points in Type I and II are general ones in the GSS algorithm which reduce the range of search space. $x'_{\text{gss}2}$ or $x'_{\text{gss}3}$ is only one new point which will replace one particle in the PSO algorithm as a GSS driven particle; that is, if the population size of one generation is N, one particle is updated by this GSS algorithm, while the other $N-1$ particles are updated by the PSO algorithm. This newly inserted GSS position has much possibility to be near the optimal point; it can improve the whole performance of PSO algorithm.

Although this effective usage of a particle produces good optimization performance, the range which includes the optimal point; the first upper bound $x_{\text{gss}4}$ and lower bound $x_{\text{gss}1}$ should be assigned. In other words, In this application of GSS to PSO, this range should be also searched at first.

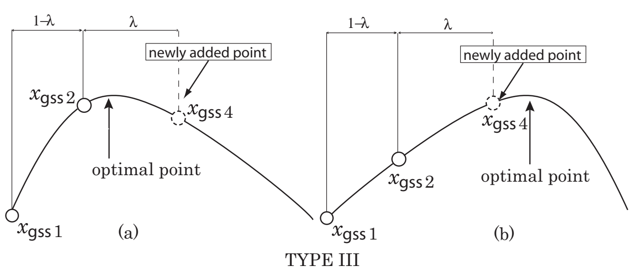


Fig. 4. Extension by Golden Section Search

Type III does not reduce the bracket length but extends the range. When right after the candidate direction is decided, only two points are useful: the gbest point and the other point that makes the steepest slope. These two points are set as $x_{\text{gss}2}$ and $x_{\text{gss}1}$ respectively.

Then as extension by golden section search, $x_{\text{gss}4}$ is calculated based on Equation (5) as a newly-inserted GSS driven particle.

$$x_{\text{gss}4} = \frac{1}{(1-\lambda)}(x_{\text{gss}2} - x_{\text{gss}1}) + x_{\text{gss}1} \text{(TYPE II)} \tag{5}$$

Fitness value of $x_{\text{gss}4}$ determines whether we should contract the search area or extend more. If fitness value of $x_{\text{gss}4}$ is more than that of $x_{\text{gss}2}$, we can assume that the optimal point is outside of the current range so that the range should be extended again. In that case, $x_{\text{gss}4}$ is set as new $x_{\text{gss}2}$ and new $x_{\text{gss}4}$ calculated based on Equation(5).

If fitness value of $x_{\text{gss}4}$ is less than that of $x_{\text{gss}2}$, the search area will be contracted. A newly-inserted GSS driven particle is calculated based on Equation (6) and set as $x_{\text{gss}3}$

$$x'_{\text{gss}3} = (1-\lambda)x_{\text{gss}1} + \lambda x_{\text{gss}3} \tag{6}$$

Since this completes the total 4 points that realize the Golden section search, the calculations (3) and (4) can be repeated afterwards.

Figure 5 is the whole procedure of the proposed optimization algorithm. This algorithm is more or less complicated

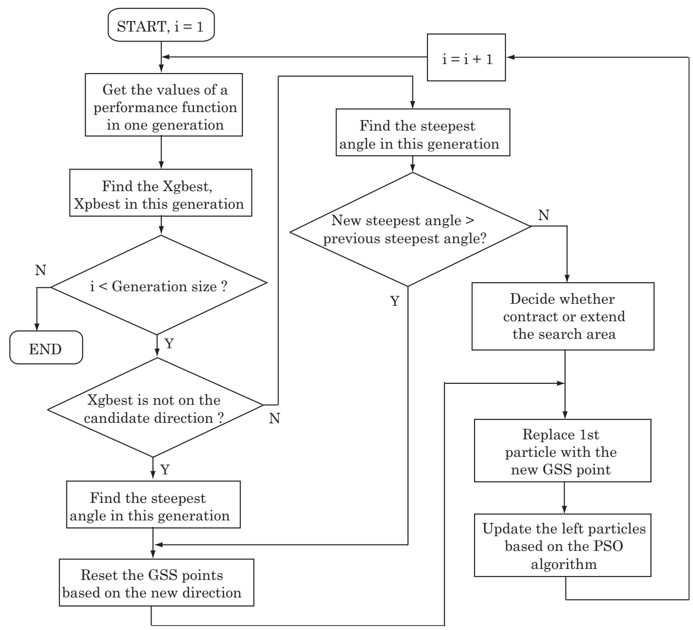


Fig. 5. Flowchart of Proposed Algorithm

than the original PSO. In the application to the real NC machine tools, however, calculation time is not a significant problem because the experiment takes much longer time than the calculation.

III. Real NC machine tool optimization with simple fitness definition

A. The Hardware-in-the-loop system Used for Experiments

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 III, and its result is shown in 8.

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.

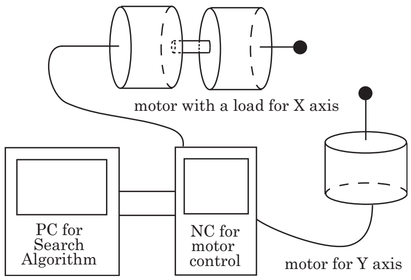


Fig. 6. Hardware-in-the-loop System for Experiments

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 and calculates the performance function based on the measurement, generating the next swarm according to the proposed algorithm. The generated swarm is fed back to the parameters of the NC controller.

B. 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. The lower figure in Figure 7 shows a zoomed trajectory around a corner where there is a large error.

Figure 8 shows an example of two trajectory errors in two trials. There are four groups of peaks where the error increases drastically. These periods correspond to the time when the trajectory changes its direction. Minimization of this trajectory is one purpose of the optimization.

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.

$$\begin{aligned} \text{Fitness} \;\; = \;\; & \mathrm{K}_e \exp\left(\mathrm{S}_e\left(\mathrm{e}_{bias} - \sum \text{trajectory error}(t)\right)\right) \\ & + \mathrm{K}_t \exp\left(\mathrm{S}_t\left(\mathrm{t}_{bias} - \text{Elapsed time}\right)\right) \end{aligned} \tag{7}$$

C. Adjustable Control Parameters to Be Optimized

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.

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.

The last parameter we consider is the timing of feedforwad control input. This timing can be illustrated as k in Figure

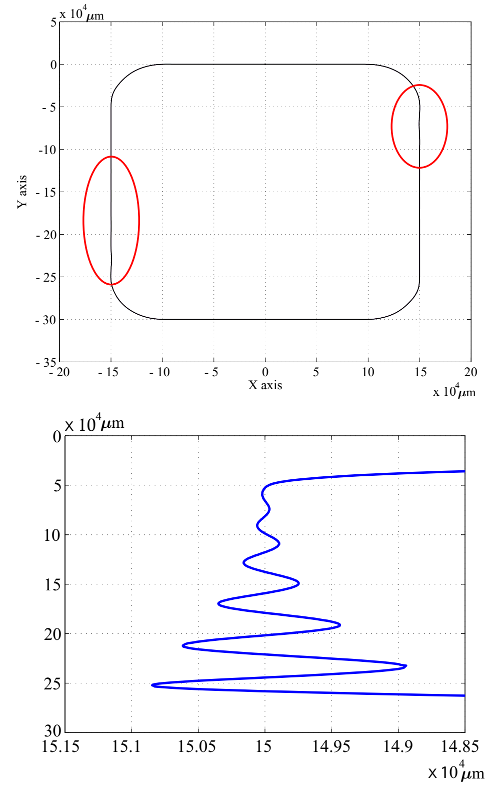


Fig. 7. Trajectory Error in Experimental Result

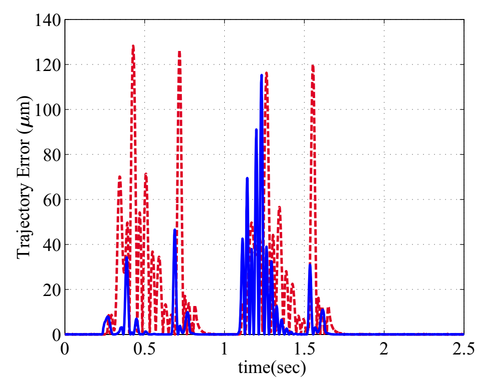


Fig. 8. Trajectory Errors in Two Trials

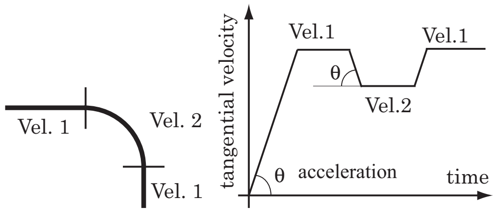


Fig. 9. Changes in the Velocity Tangential to the Trajectory

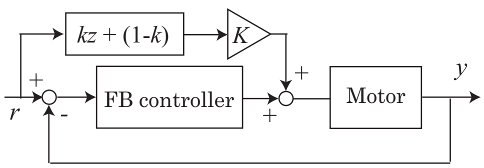


Fig. 10. Adjustment of Feedforward Control Input Timing

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

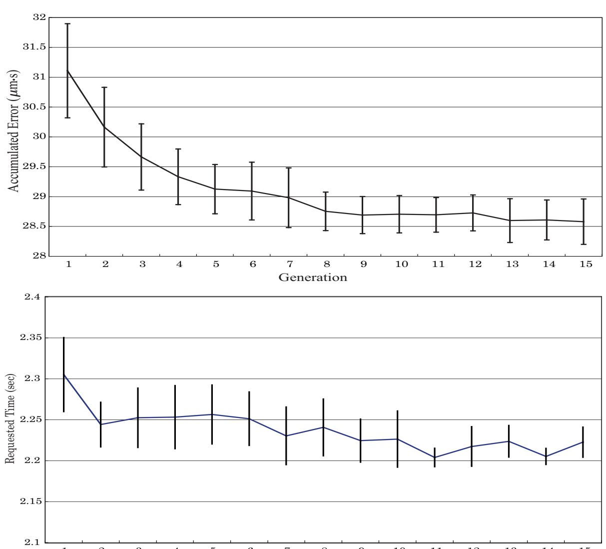


Fig. 11. Change of the Trajectory Errors and the Requested Time

The GPSO is applied to the HIL system. Figure 11 describes the result of optimization, showing the value of accumulated trajectory error and the required time of x_{gbest}. The populations size is 30. The changes in these values evaluated at each x_{gbest} in 15 generations represents the proposed algorithm optimizes the parameter as the generation goes on. The vertical lines represents the standard deviation of 12 experiments. This standard deviation also decreases with the generations.

Real systems do not yield the same control performances with the same given control parameters. Considering this point, the parameters of x_{gbest} in one generation are re-evaluated in the next generation in this experiment in order to ensure the reliability of the optimization. This is why there are some rises in the graphs. Although this re-evaluation does worsen the optimization performance, it should be conducted in a HIL system.

Table I shows the optimized parameters and there control performances. P_1, P_2 are the values of θ in x and y axis, P_3 is Vel.2 in Figure 9, P_4 is k in Figure 10 for y axis controller, and P_5, P_6 means the velocity feedforward gains in x and y axis. 4 sets of parameters which are obtained as the results of optimization are selected and re-evaluated 4 times relatively. They all provide almost same performances, proving the proposed optimization succeeded in finding optimizing parameters with reliability.

TABLE I

OPTIMIZIN PARAMETERS AND THEIR PERFORMANCES

P_1	P_2	P_3	ERR1	ERR2	ERR3	ERR4
P_4	P_5	P_6	Time1	Time2	Time3	Time4
200	164	5999	31.51	31.44	31.28	31.81
-2197	400	204	2.196	2.18	2.205	2.189
169	162	4020	31.35	31.51	31.42	31.45
-762	400	294	2.194	2.199	2.196	2.2
200	164	5999	31.53	31.72	31.56	31.66
-2164	400	204	2.202	2.185	2.159	2.161
134	162	2000	31.77	31.11	31.18	31.36
-451	400	100	2.181	2.199	2.197	2.196

From these optimizing control parameter sets, we also can tell the proposed algorithm not only can select the best parameter but also can find the optimizing parameter sets. For P_2 and P_5, 162 and 400 are found as the best parameters. For P_1, P_3, P_4, we can see they work as a set. This insight into the relationship between parameters set is what the proposed optimization can provide when it is applied to a HIL system.

IV. NC MACHINE TOOL OPTIMIZATION WITH MORE DETAILED FITNESS DEFINITION

A. *Performance Analysis of Each Control Parameter*

Figure 12 shows the configuration of controller we used in NC machine tools, which is the most general cascade control of current, velocity, and position control. As is said in Section III-C, feedfoward control parameters are main target of this optimization which is shaded in Figure 12.

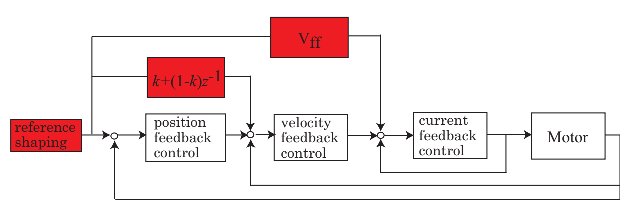


Fig. 12. Controller Configuration

Each feedfoward parameter has the roles:

Reference Shaping (P_1, P_2, P_3)

Acceleration/Deceleration of each motor specified in Figure 10 and the limitation of tangential velocity in the arc corners is characterized into reference shaping. These parameters have trade-off characteristics. If the values of acceleration and limited velocity are slow, the less trajectory error the controller produces although they also take much more time for writing one trajectory. If the values are set high, it results in small settling time and large trajectory errors.

Adjustment of Feedfoward Control Input Timing (P_4)

This is related with the trajectory error especially in a circle trajectory. The parameter can coordinate differences in timing of two motors. If the difference is too large, the circle that is tracked by two motors tends to be an ellipse.

Velocity Feedforward Gain (P_5, P_6 in each motor)

This parameter reduces the over/undershoot after changes of direction in a trajectory adjusting the velocity control input, which means it can reduce the effect of integration control and prevent windup.

In order to take these characteristics into consideration, the trajectory is divided into 8 sections like Figure 13: 4 arcs and 4 linear regions.

Parameter P_4 uses the trajectory error at the arc corner: (2), (4), (6) and (8), and P_5 and P_6 use the error peaks in the area (1), (3), (5) and (7) for their optimizations, while parameters P_1 to P_3 use the whole trajectory errors of 8 sections.

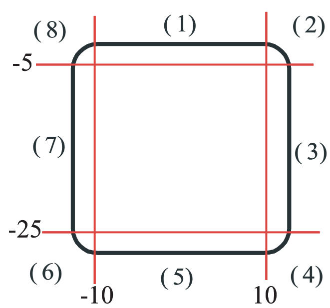


Fig. 13. Trajectory is Divided into 8 Sections

Also parameters P_4 to P_6 are not so strongly related with the settling time, the requested time is not considered in fitness definition for these three parameters. Consequently, P_4 and P_5, P_6 have their own fitness definition like below.

$$\text{Fitness}_i = \exp\left(\text{S}_i\left(\text{e}_{bias_i} - \sum \text{trajectory error}_i(t)\right)\right) \quad (8)$$

Experiment with this new fitness definition is our future work; however, this fitness definition will improve optimization performance.

B. Optimization Improvement by GPSO

In our last work, the effectiveness of the GPSO in NC parameter tunings is not proved. Since the GSS driven particle is limited to one particle in this paper, the effectiveness can be verified focusing on the performance of that particle. To this end, pbest which is the personal best particle in its own history of each particle is compared. If the fitness of the GSS driven particle's pbest is higher than that of the other particles, the superiority of the GSS driven particle can be proved statistically. In these experiments, the population size is set to 16 and the generation size is set to 10.

TABLE II

FITNESS AVERAGE OF PBESTS IN 4 EXPERIMENTS

1st(GSS)	2nd	3rd	4th	5th	6th	7th	8th
1.00	0.85	0.94	0.79	0.96	0.85	0.94	0.87
1.00	0.87	0.92	0.92	0.92	0.97	0.79	0.95
1.00	0.90	0.97	0.88	0.96	0.93	0.98	0.94
1.00	0.92	0.96	0.97	0.96	0.98	0.89	0.94
9th	**10th**	**11th**	**12th**	**13th**	**14th**	**15th**	**16th**
0.78	0.77	0.92	0.81	0.89	0.86	0.77	0.91
0.92	0.85	0.90	0.77	0.92	0.90	0.89	0.68
0.86	0.88	0.97	0.86	0.92	0.92	0.86	0.99
0.94	0.90	0.93	0.86	0.96	0.93	0.92	0.86

Table II illustrates the fitness values of pbests. Experiments were conducted 4 times, and the fitness values during 10 generations were averaged. If the fitness values are normalized to make the fitness value of the GSS driven particle 1, the other particles' fitness are all lower than 1, which proves the GSS driven particle is more successful in finding optimal parameters.

V. CONCLUSION

The GPSO algorithm is revised in this paper to make clear its performance. The number of GSS driven particle is limited to one. By experiments, the superiority of that GSS particle in finding optimal point is verified.

Parameters which are said to be difficult to optimize is chosen and the roles of those parameters are scrutinized. Based on those scrutinies, fitness is defined for parameters; trajectory error is divided into some sections based on its shape. The relationship between those trajectory errors and the parameters are emphasized in new fitness definition. Experiment with the new fitness definition is future work. In that experiment, the repeatability problems will be also addressed.

REFERENCES

[1] Sehoon Oh and Yoichi Hori "Control Parameter Optimization in the Hardware-in-the-loop System using Novel Search Algorithm," *IECON*, 2006

[2] Press, W.H., Teukolsky, S.A., Vetterling, W. T., and Flannery, B.P. *Numerical Recipes in C, The Art of Scientific Computing, second edition*, 1999.

[3] 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.

[4] Byunghoon Chang and Yoichi Hori "Research related to the Parameter Auto-tuning of Two Mass Control System," *IEE of Japan Technical Meeting Record*, IIC-05-47, 2005.