IEMS Vol. 3, No. 1, pp. 78-84, April 2004.

Optimizing Movement of A Multi-Joint Robot Arm with Existence of Obstacles Using Multi-Purpose Genetic Algorithm

Yoshiaki Toyoda†
College of Science and Engineering
Aoyama Gakuin University, Kanagawa 229-8558, JAPAN
Tel: +81-42-759-6000, Fax: +81-42-759-6560, E-mail: toyoda@ise.aoyama.ac.jp

Fumihiko Yano
College of Humanities
Obirin University, Tokyo 194-0294, JAPAN
Tel: +81-42-797-9629, Fax: +81-42-797-1778, E-mail: yano@obirin.ac.jp

Abstract. To optimize movement of a multi-joint robot arm is known to be a difficult problem, because it is a kind of redundant system. Although the end-effector is set its position by each angle of the joints, the angle of each joint cannot be uniquely determined by the position of the end-effector. There exist the infinite number of different sets of joint angles which represent the same position of the end-effector.
This paper describes how to manage the angle of each joint to move its end-effector preferably on an X-Y plane with obstacles in the end-effector's reachable area, and how to optimize the movement of a multi-joint robot arm, evading obstacles. The definition of "preferable" movement depends upon a purpose of robot operation. First, we divide viewpoints of preference into two, 1) the standpoint of the end-effector, and 2) the standpoint of joints. Then, we define multiple objective functions, and formulate it into a multi-objective programming problem. Finally, we solve it using multi-purpose genetic algorithm, and obtain reasonable results. The method described here is possible to add appropriate objective function if necessary for the purpose.

Keywords: multi-purpose programming, genetic algorithm, robot, multi-joint robot arm, optimal control, adaptive control, optimal end-effector trajectory

1. INTRODUCTION

An industrial robot has multi joints and links which compose its arm, and an end-effector which is installed at the tip of the arm. As increasing the number of joints and links, movement flexibility increases, however, robot control gets more complex. Therefore, considering both the flexibility of the movement and the easiness of the control, an arm of a normal industrial robot appearing on the market usually has five or six joints and links, and moves in a three dimensional space. However, two joints out of the five or six are used for fine positioning of an end-effector. Therefore, the number of joints regarding movement of an arm is three or four. In addition, one out of them is used to increase the reachable dimensions from two to three.

Generally, the angle of the joints cannot be uniquely determined by the position of the end-effector even if a robot arm has only two joints and two links, and the end-effector at the tip of the arm moves on an two dimensional X-Y plane, although the position of the end-effector can be calculated using the angle of each joint. Therefore, a robot control system is thought to be a redundant system. These facts express the difficulties of a robot control.

Several researches regarding control and generation of a multi-joint robot trajectory have appeared in technical journals (Akutsu, 1997; Davidor, 1991; Freund, 1998; Wang and Zalzala, 1996). Those researches mainly focus not on the movement of whole robot arm but on the control of each joint to move the end-effector along a predetermined trajectory as close as possible, for example, along the straight line between two points. However, we deal with a problem to optimize movement of a whole multi-joint robot arm, when the end-effector

† : Corresponding Author

moves from a starting point to an ending point.

In this paper, a robot with an arm which is composed of two joints and two links is considered, and it is supposed that an end-effector at the tip of the arm moves on an X-Y plane. Here, we discuss the case that there exist obstacles in the reachable area of the end-effector, and describe that the formulation works well with genetic algorithm.

We first define normal preferable movement of a whole robot arm into four different objective functions from the standpoint of the end-effector and from the standpoint of each joint (Yano and Toyoda, 1999). Then, we evaluate each of the objective functions, using it as a fitness function of genetic algorithm. Finally, we formulate the problem as a multi-purpose programming problem, and obtain a preferable solution by genetic algorithm. Also we suggest possibility to add various objective functions necessary under different environments.

2. PROBLEM DESCRIPTION

Here, we consider a robot which has an arm with two joints and two links. The end-effector installed at the tip of the arm is supposed to move on an X-Y plane. Weight, bend, acceleration, etc. of the arm are not considered.

As shown in Figure 1, let the lengths of the links 1 and 2, the angle between link 1 and X-axis, the angle between links 1 and 2, and the position of the end-effector be 1, 1, α_i , β_i , $P_i = (x_i, \ y_i)$ respectively. For easy understanding, restrictions, $0 \le \alpha_i \le \pi/2$ and $0 \le \beta_i \le \pi/2$, are added. These restrictions do not reduce its generality, because the method described here is easily extended the number of joints and links to three or four, and the number of the reachable dimensions to three. The number of the joints and links, and the number of the dimensions are not problems as long as the position of the end-effector is determined by the angle of each joint.

Under this restriction, the reachable area of the end-effector is shown in Figure 2. However, in this paper, we consider cases that there exist obstacles in the reachable area, that is, obstacle areas are infeasible zones. This means that the feasible reachable area, which is hereafter called the "feasible area," is reduced to smaller one as shown in Figure 3. Although the end-effector can freely move in the feasible area, it goes from a starting point to an ending point evading obstacles. The area where the end-effector is reachable gets smaller with obstacles, but still the number of positions where the end-effector arrives is infinitive, that is, the number of the set of angles of each joint is infinitive.

A robot control problem here is far different from the problem that the best combination is searched among the quite many but the limited number of combinations, such as a traveling salesman problem. In our robot control problem, we have to create a new combination of passing points, and find the best one among those under a certain restriction. Even when the end-effector moves

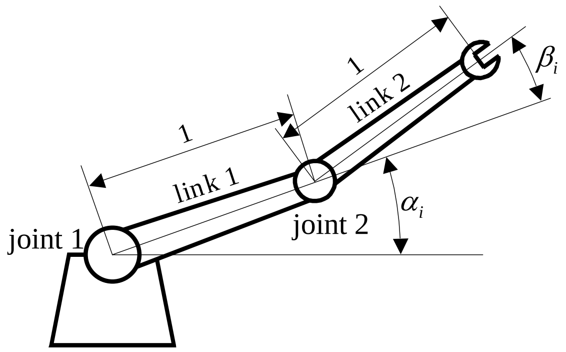

Figure 1. The multi-joint robot under consideration

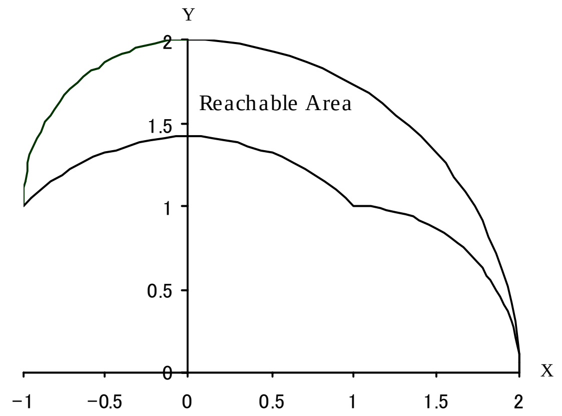

Figure 2. The reachable area

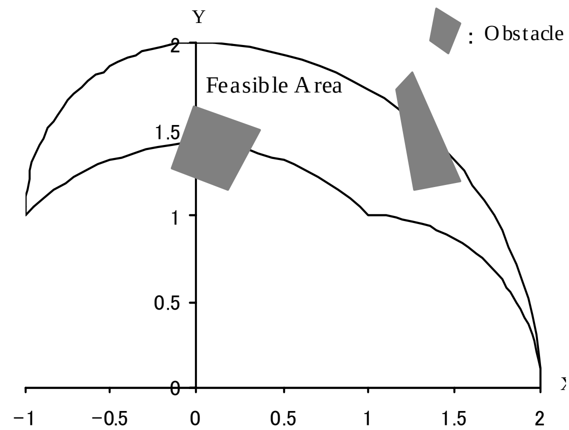

Figure 3. The feasible area

on the straight line between two points, a joint may greatly rotate at one time, and the end-effector may go out of a feasible area at another time.

Consider that the end-effector moves from a starting point, $P_0=(x_0,\ y_0)$, determined by α_0 and β_0, to an ending point, $P_n=(x_n,\ y_n)$, determined by α_n and β_n, passing n-1 points, altering the angles of the joints by $\alpha_i-\alpha_{i-1}$ and $\beta_i-\beta_{i-1}$ in a unit time period. The starting point, the ending point, all the passing points and all the routes of the end-effector should be in the feasible area. Here, the starting and ending points are fixed, however, other points, P_1, P_2, …, P_{n-1}, may be any point in the feasible area where the trail between P_{i-1} and P_i is also in the feasible area.

Let P_i' be a point on the straight line between P_{i-1} and P_i. Then, P_i' is expressed by the linear combination of P_i and P_{i-1}, or

$P_i'=(qx_{i-1}+(1-q)x_i, qy_{i-1}+(1-q)y_i)$

(where i=1,2,…,n, $0\le q\le 1$).

As all the passing points and routes should be in the feasible area, E, then $P_i\in E$ and $P_i'\in E$. Here, we try to obtain preferable passing points of an end-effector between starting and ending points evading existing obstacles under not a single preferable condition but multiple ones. This problem can be formulated into a multi-purpose programming problem. Then, a robot can determine preferable passing points for itself under a certain restriction and move its end-effector along with the passing points. This is not a problem to find an optimal combination of the finite number of enumerated passing points, that is, a general combinatorial problem. It is to create a new passing point among the infinite number of passing points (i. e., the infinite number of combinations) and to find optimal passing points under a certain restriction.

3. OBJECTIVE FUNCTIONS

In this paper, we define the four preferable movements and their objective functions. Then, we formulate those into a multi-purpose programming problem.

The preferable movements and their objective functions are defined as follows:

1) The minimum total distance of the end-effector movement.

The distance, d_i, between two adjoining passing points, P_{i-1} and P_i, is described as

$$d_i=\sqrt{(x_i-x_{i-1})^2+(y_i-y_{i-1})^2}\ .$$

Therefore, an objective function, s, to minimize the total distance of the end-effector movement is formulated as:

$$s=\sum_{i=1}^{n}d_i=\sum_{i=1}^{n}\sqrt{(x_i-x_{i-1})^2+(y_i-y_{i-1})^2}\ .\ \to\ \min$$

If there exists no obstacle on the straight line between the starting and ending points, and any point on the straight line is in the feasible area, the optimal value is $\min s=\sqrt{(x_n-x_0)^2+(y_n-y_0)^2}$, when the end-effector moves on the straight line between the starting and ending points.

Here, normalize the objective function, s, as follows and let h_1 be a new objective function.

$$h_1=\frac{s-\sqrt{(x_n-x_0)^2+(y_n-y_0)^2}}{s}=1-\frac{\sqrt{(x_n-x_0)^2+(y_n-y_0)^2}}{s}\ \to\ \min \tag{1}$$

Then, the range of h_1 is $0\le h_1\le 1$.

2) Uniform velocity of the end-effector.

That the end-effector moves between starting and ending points with uniform velocity means that it moves constant distance in a unit time period. As an end-effector moves between adjoining two points among passing points in a unit time period, the distances between those two points may be all equal. The distance between two adjoining passing points, P_{i-1} and P_i, is expressed as

$$d_i=\sqrt{(x_i-x_{i-1})^2+(y_i-y_{i-1})^2}\ .$$

Therefore, the objective function, d, to attain uniform velocity of the end-effector is described as

$$d=\max_{i=1,2,\dots,n}d_i-\min_{i=1,2,\dots,n}d_i\ \to\ \min\ .$$

Its optimal value is min d=0, that is, the distances between two adjoining points are all equal. Here, normalize d as follows and let h_2 be a new objective function:

$$h_2=\frac{\max_{i=1,2,\dots,n}d_i-\min_{i=1,2,\dots,n}d_i}{\max_{i=1,2,\dots,n}d_i}=1-\frac{\min_{i=1,2,\dots,n}d_i}{\max_{i=1,2,\dots,n}d_i}\ \to\ \min \tag{2}$$

The range of h_2 is $0\le h_2\le 1$.

3) The minimum total rotate angle of joints.

From the standpoint of each joint, it is preferable that total rotate angle of joints is smaller. Regarding the joint 1, it rotates from α_{i-1} to α_i to move the end-effector from P_{i-1} to P_i. As the joint displacement angle is expressed by $\alpha_i-\alpha_{i-1}$, the objective function, λ_α, is formulated by

$$\lambda_\alpha=\sum_{i=1}^{n}\left|\alpha_i-\alpha_{i-1}\right|\ \to\ \min$$

The optimal value of this function is

min $\lambda_\alpha = |\alpha_n - \alpha_0|$.

Here, if $\alpha_i - \alpha_{i-1} \geq 0$ for all i's, then $\lambda_\alpha = (\alpha_n - \alpha_0)$, and if $\alpha_i - \alpha_{i-1} \leq 0$ for all i's, then $\lambda_\alpha = (\alpha_0 - \alpha_n)$. That is, regardless with the positions of passing points, when $\alpha_i - \alpha_{i-1} \geq 0$ or $\alpha_i - \alpha_{i-1} \leq 0$ for all i's, the objective function obtains the optimal value. Normalizing the objective function, λ_α, as follows, and let $h_{\alpha 3}$ be a new objective function.

$$h_{\alpha 3} = \frac{\lambda_\alpha - |\alpha_n - \alpha_0|}{\lambda_\alpha} = 1 - \frac{|\alpha_n - \alpha_0|}{\lambda_\alpha} \rightarrow \min$$

Now, the range of $h_{\alpha 3}$ is $0 \leq h_{\alpha 3} \leq 1$.

Similarly, we can obtain $h_{\beta 3}$ for the joint 2 by replacing α in the case of joint 1 with β.

Here, let a linear combination of $h_{\alpha 3}$ and $h_{\beta 3}$ be a new objective function.

$$\begin{aligned} & h_3 = u_\alpha h_{\alpha 3} + u_\beta h_{\beta 3} \\ & (\text{where } u_\alpha + u_\beta = 1,\ u_\alpha, u_\beta \geq 0) \end{aligned} \tag{3}$$

The range of h_3 is $0 \leq h_3 \leq 1$.

4) Uniform angular velocity of each joint.

It is preferable that angular velocity at each joint is uniform. Regarding the joint 1, the objective function, δ_α, is expressed as follows:

$$\delta_\alpha = \max_{i=1,2,\dots,n} (\alpha_i - \alpha_{i-1}) - \min_{i=1,2,\dots,n} (\alpha_i - \alpha_{i-1}) \rightarrow \min$$

The optimal value of this objective function is min $\delta_\alpha = 0$, that is, $\alpha_i - \alpha_{i-1}$ for all i's are equal. Here, normalize the objective function δ_α as follows and let $h_{\alpha 4}$ be a new objective function.

$$h_{\alpha 4} = \frac{\delta_\alpha}{\pi} \rightarrow \min$$

The range of $h_{\alpha 4}$ is $0 \leq h_{\alpha 4} \leq 1$.

Similarly, we can obtain $h_{\beta 4}$ for the joint 2 by replacing α in the case of joint 1 with β.

Here, let a linear combination of $h_{\alpha 4}$ and $h_{\beta 4}$ be a new objective function.

$$\begin{aligned} & h_4 = v_\alpha h_{\alpha 4} + v_\beta h_{\beta 4} \\ & (\text{where } v_\alpha + v_\beta = 1,\ v_\alpha, v_\beta \geq 0) \end{aligned} \tag{4}$$

The range of h_3 is $0 \leq h_4 \leq 1$.

Each objective function described above has its own characteristics and has some conflicts each other. It means that even if each objective function independently obtains a good fitness value, it does not always lead to a totally preferable movement of the end-effector. Therefore, we try to formulate this problem into a multi-purpose programming problem and solve it using genetic algorithm. Here, the multi-purpose objective function is organized by the dynamically weighted objective functions. Let the sum of the weighted objective functions defined above,

$$h = \sum_{j=1}^{4} w_j h_j \ (\text{where } w_j \geq 0,\ \sum_{j=1}^{4} w_j = 1), \tag{5}$$

be a multi-purpose objective function to be minimized. Then, the multi-purpose programming problem is formulated as follows:

$$\left.\begin{aligned} & h = \sum_{j=1}^{4} w_j h_j \rightarrow \min \\ & \text{subject to} \\ & 0 \leq \alpha_i \leq \frac{\pi}{2},\ 0 \leq \beta_i \leq \frac{\pi}{2},\ i = 0, 1, 2, \dots, n \\ & x_i = \cos(\alpha_i) + \cos(\alpha_i + \beta_i) \\ & y_i = \sin(\alpha_i) + \sin(\alpha_i + \beta_i),\ i = 0, 1, \dots, n \\ & P_i' \in E,\ i = 1, 2, \dots, n \\ & w_j \geq 0,\ j = 1, 2, 3, 4,\ \sum_{j=1}^{4} w_j = 1 \end{aligned}\right\} \tag{6}$$

The weight, w_j, varies depending upon the purpose of the problem. In this paper, w_j is assigned as

$$w_j = \frac{h_j}{\sum_{j=1}^{4} h_j}$$

so that all the h_j's are balanced.

Object functions depend upon purposes of problems. According to the purposes, addition, reduction or change of objective functions is possible, and appropriate weighting are given to those objective functions.

Consider the link 3 is added at the tip of the link 2, which length is also 1. If the angle between the links 2 and 3 is γ_i, then the position of the end-effector at the tip of the link 3 is expressed as follows:

$$x_i = \{\cos(\alpha_i) + \cos(\alpha_i + \beta_i) + \cos(\alpha_i + \beta_i + \gamma_i)\}$$

$$y_i = \{\sin(\alpha_i) + \sin(\alpha_i + \beta_i) + \sin(\alpha_i + \beta_i + \gamma_i)\}$$

Similarly, additional joints and links can be increased.

To move the end-effector in a three-dimensional X-Y-Z space, an joint is added at the root of the link 1, which rotates the whole arm around the Y-axis. If the angle of the joint from X-axis is θ_i, the position of the end-effector with two links is expressed as follows:

$$x_i = \{\cos\alpha_i + \cos(\alpha_i + \beta_i)\}\cos\theta_i$$

$$y_i = \{\sin\alpha_i + \sin(\alpha_i + \beta_i)\}$$

$$z_i = \{\cos\alpha_i + \cos(\alpha_i + \beta_i)\}\sin\theta_i$$

In this manner, addition of links and dimensions is not a problem, but results in more computational time.

4. FORMULATION INTO GENETIC ALGORITHM

Let us consider a multi-purpose programming problem which is formulated as the equation (6) in the previous chapter.

1) Expression of a position of the end-effector.

First of all, we define an angle unit (α_i, β_i) of two joints as a gene. As a gene (α_i, β_i) expresses an arm configuration at *i*-th passing point, a passing point P_i of the end-effector can be expressed by a function of α_i and β_i. A set of *n*+1 genes including a starting point and an ending point, $\{(\alpha_0, \beta_0), (\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n)\}$, describes a chromosome, that is, a chromosome expresses a series of arm configurations and a single chromosome expresses an individual.

2) Generation of initial individuals.

At the initial generation, we give constant angle values at a starting point P_0 and an ending point P_n, and random angle values within a feasible area at passing points to generate initial parent genes. In the case that a preferable end-effector trajectory is predetermined and the fitness function is defined to minimize the variance of the actual pass from the predetermined trajectory, it is reasonable to give similar passing points to the trajectory as described in Davidor's paper (Davidor, 1991). In this paper, no preferable trajectory is given. Therefore, we have to generate initial individuals randomly.

3) Definition of the fitness function.

Here, we use objective functions defined by the equations (1-5) in the previous chapter as fitness functions.

4) Mechanism of Selection.

In genetic algorithm, the general reproduction strategy of new generations is that parents with better fitness have more reproduction chance. However, in the optimization problem which we discuss, we have to create new combinations of passing points (arm configurations) among the infinite number of passing points, and find out the optimal or approximate solution among those combinations without falling into a local solution. Therefore, in this paper, next generation is constructed by selecting about 4-8% of fitter chromosomes and others among all chromosomes randomly allowing repetition.

5) Crossover.

Two parent chromosomes to be crossovered are randomly selected. In a general crossover method, it occurs that the end-effector greatly goes forward sometime and backward another time. This crossover method is not preferable in this kind of research. Therefore, Davidor introduced a variable gene length which composes individuals and let two parents swap at *i* and *j* which satisfy the equation.

$$\min_{\substack{i=1,2,\dots n-1 \\ j=1,2,\dots,n-1}} \sqrt{\left(x_{i1}-x_{j2}\right)^2+\left(y_{i1}-y_{j2}\right)^2}$$

where (x_{i1}, y_{i1}) is the *i*-th passing point of one parent and (x_{j2}, y_{j2}) is the *j*-th passing points of the other parent (Davidor, 1991). This crossover method is able to eliminate the destruction of chromosome. However, according to increase of the number of generations, the length of a gene becomes shortening and finally only the starting and the ending points are left. Davidor introduced deletion and addition operators to his genetic algorithm in order to improve the loss of arm-configurations and trajectory diversity in case of the varying gene length (Davidor, 1991). In our method, we fix the gene length and swap two parents at each *i*-th passing point which satisfies the equation.

$$\min_{i=1,2,\dots n-1} \sqrt{\left(x_{i1}-x_{i2}\right)^2+\left(y_{i1}-y_{i2}\right)^2}$$

where (x_{i1}, y_{i1}) and (x_{i2}, y_{i2}) are the *i*-th passing points of two parents. The gene length does not change and crossover is done at close genes, then the situation that the end-effector greatly goes forward sometime and backward another time is prevented. In Davidor's method, a cross site is determined according to the genotypic character rather than to the genotypic position. However, our crossover method chooses a cross site according to both the genotypic character and genotypic position.

6) Mutation.

Mutation is carried out by changing the angle of each joint randomly at given probability. Mutation is applied to the individuals to keep their diversity and not to fall into a local optimum. Generally, high mutation rate makes the fitness value of individuals worse (Kitano, 1993; Kobayashi, 1995). In most cases, the mutation rate is set at a small value such as an approximate reciprocal of the gene length. Using only crossover, it is hard to create the best or better combinations of passing points, because initial individuals are generated using random numbers. Therefore, we need the high mutation rate to create new combinations of passing points with good fitness values. Through several simulations, in our method, the mutation rate is set about 0.7.

5. NUMERICAL EXPERIMENTS

In a multi-purpose programming problem regarding the optimization of the robot arm movement, if the weights in the multi-purpose objective function are set at appropriate values, a combination of arm configuration approximating to a preferable movement can be obtained. Figure 4 shows the best initial solution without obstacle.

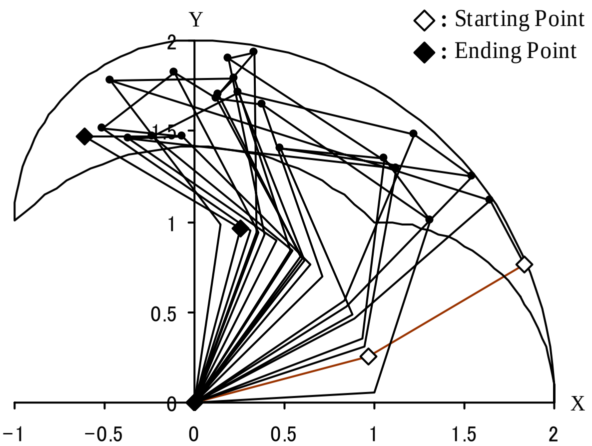

Figure 4. The best initial solution without obstacle

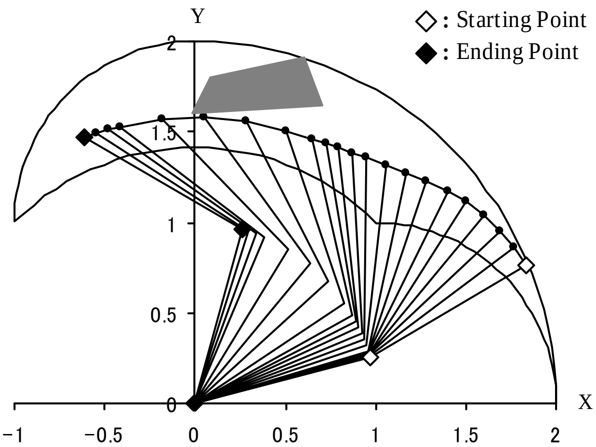

Figure 7. A preferable movement with one obstacle at the 20,000th generation

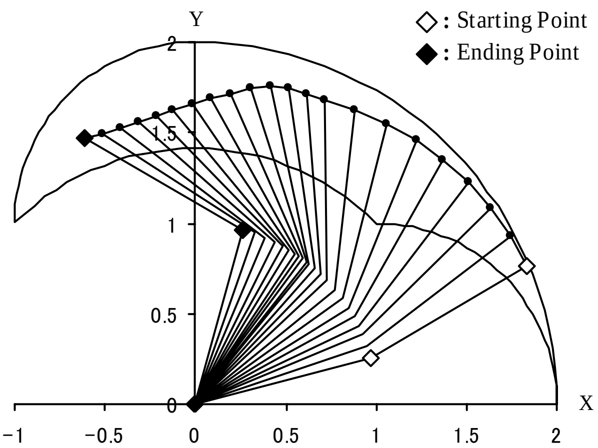

Figure 5. A preferable movement without obstacle at the 20,000th generation

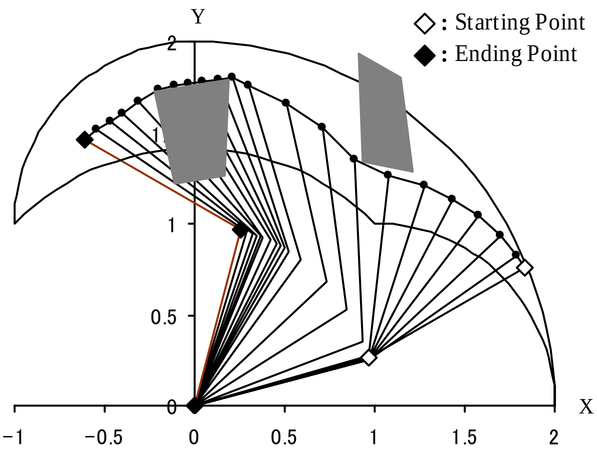

Figure 8. A preferable movement with two obstacles at the 50,000th generation (1)

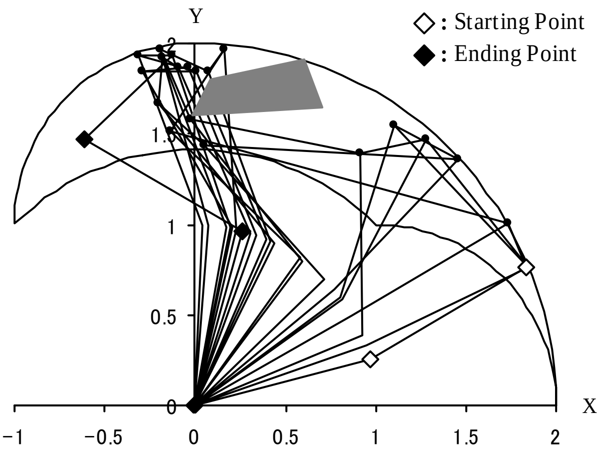

Figure 6. The best initial solution with one obstacle

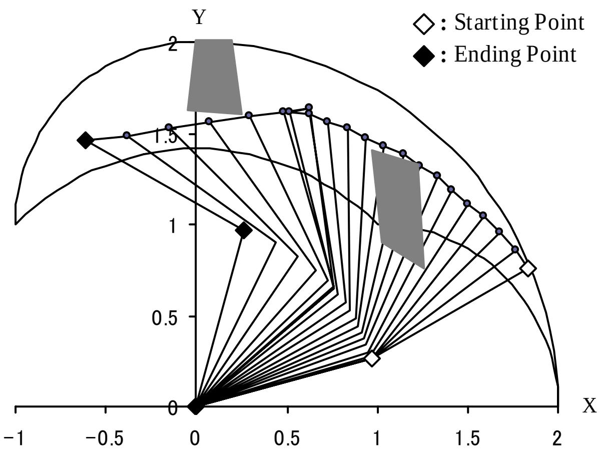

Figure 9. A preferable movement with two obstacles at the 50,000th generation (2)

Figure 5 shows a preferable movement of the end-effector without obstacle solved by our genetic algorithm at the 20,000th generation. Figure 6 shows the best initial solution with one obstacle. We tried various weighting techniques and found out a preferable dynamically weighting method in which a supplement fitness function, $h' = \sum_{j=1}^{4} w'_j h_j$ (where $w'_j \geq 0$, $\sum_{j=1}^{4} w'_j = 1$) , is formulated. Values of w'_j are determined as:

1) if $h_3 \geq 0.20$, then w'_1=0.00, w'_2=0.00, w'_3=0.70, w'_4=0.30
2) if $0.10 \leq h_3 < 0.20$, then w'_1=0.10, w'_2=0.10, w'_3=0.40, w'_4=0.40,
3) if $0.01 \leq h_3 < 0.10$, then w'_1=0.05, w'_2=0.20, w'_3=0.55, w'_4=0.20, and
4) if $h_3 < 0.01$, then $w'_j = \dfrac{h'_j}{\sum_{j=1}^{4} h'_j}$.

Figure 7 shows an obtained result of an end-effector trajectory and an arm configuration in the case of existence of one obstacle at the 20,000th generation. The value of the fitness function is h=0.48 with h_1=0.10, h_2=0.61, h_3=0.07 and h_4=0.04. Figure 8 shows an obtained result in the cases of existence of two obstacles at the 50,000th generation. The values of the fitness function are h=0.47 with h_1=0.15, h_2=0.66, h_3=0.14 and h_4=0.08. Figure 9 shows another result in the cases of existence of two obstacles at the 50,000th generation. The values of the fitness function are h=0.37 with h_1=0.18, h_2=0.52, h_3=0.08 and h_4=0.07.

Many similar problems are examined and all results are almost same.

6. CONCLUSION

In this paper, we deal with not a problem to move an end-effector along a certain predefined trajectory but a problem to control a whole robot arm from a standpoint of the movement optimization of an overall arm. Although various optimization techniques to multi-purpose programming problems have been proposed (Kitano, 1995; Nishikawa, 1982), most of them are not fit for our research. Then, we applied genetic algorithm to solve the problem, and made various experimental simulations. Typical results are shown in the previous chapter. Other results are almost same and these results can be considered to be reasonable. It may be said that the end-effector trajectory is quite desirable, however, the fitness function, h_3, is necessary to be improved to a better value.

The methods described in this paper can be extended to the cases that the end-effector moves in a three-dimensional space, that the number of links increases, etc.

REFERENCES

Akutsu, K. (1997) Generation of the path for robot by the genetic algorithm. *Report of Machinery and Metallurgy Research Institute of Chiba Prefecture* **26**(1), 67-74.

Davidor, Y. (1991) *Genetic Algorithm and Robotics, A Heuristic Strategy for Optimization*, World Scientific Publishing, London.

Freund, E. (1998) Automatic trajectory generation for multi-joint robot systems. *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, Aachen, **4**, 2216-2221.

Kitano, H. (1993) *Genetic Algorithm*, Sangyo Tosho, Tokyo.

Kitano, H. (1995) *Genetic Algorithm 2*, Sangyo Tosho, Tokyo.

Kobayashi, S., and Yamamura, M. (1995) Search and learning by genetic algorithms. *Journal of the Robotics Society of Japan*, **13**, 57-62.

Nishikawa, Y., Sannomiya, N., and Ibaraki, T. (1982) *Optimization*, Iwanami Shoten, Tokyo.

Wang, Q., Zalzala, A. M. S. (1996) Genetic control of near time-optimal motion for an industrial robot arm. *Proceedings of the 1996 13th IEEE International Conference on Robotics and Automation*, Minneapolis, MN, 2592-2597.

Yano, F., and Toyoda, Y. (1999) Preferable movement of a multi-joint robot arm using genetic algorithm. *Proceedings of SPIE Conference on Intelligent Robots and Computer Vision XVIII: Algorithms, Techniques, and Active Vision*, Boston, MA, **3837**, 80-88.