Machine Intelligence & Robotic Control, Vol. 5, No. 4, 129–135 (2003)

Paper

A Study of Interactive Genetic Algorithm for Human-Friendly Trajectory Generation of a Robot Arm*

Indra Adji Sulistijono†§ and Naoyuki Kubota‡¶

Abstract: This work deals with human-friendly trajectory generation of a robot arm. Various methods for the trajectory generation have been proposed so far, but robots must deal with environments including human operators. In this situation, the robot should take a suitable action/motion to the individual operator. This work applies an interactive genetic algorithm for the trajectory generation using human evaluation. Basically human evaluation is very important for generating robotic behavior, but the detail of the human evaluation is not clear. Therefore, the robot must estimate human evaluation through the optimization process, and use a state-value function used often in reinforcement learning. Furthermore, the effectiveness of the proposed method through some experiments of the robot arm will be discussed.

Keywords: Interactive Genetic Algorithm, Human-friendly Trajectory Generation

1. Introduction

RECENTLY, human-friendly robots and partner robots have been developed for human society of the next generation [1]–[3]. These robots require intelligent capabilities concerning human-robot interactions. At least, the robot should generate behaviors peculiar to its owner to realize the communication with the owner [1], [3]. Kristensen *et al.* proposed a direct human teaching method for specific scenario like the coffee serving [3]. Arsenio proposed strategies to acquire such competencies based on human-robot interactions. Interaction with human teachers facilitates robot learning of new objects and their functionality, or the acquisition of new competencies as an actor [4].

Evolutionary algorithms have been applied to path-planning problems. Xiao proposed an adaptive evolutionary planner/navigator using various operators to evolve and improve candidate paths [23]. A hierarchical trajectory planning method on the pseudo-potential space for manufacturing has been proposed. This method has basically two layers of trajectory generator and configuration generator using virus-evolutionary genetic algorithms (GAs) [11]–[13], [23]. Nowadays, various adaptive robotic systems have been proposed to adapt under dynamic or unknown environments.

Furthermore, reinforcement learning has been applied to build an agent that maximizes their expected utility only using reward or punishment from the environment. That is, the reinforcement learning does not use explicit teaching signals, but uses evaluative feedback obtained through the interaction with the environment. However, the search is difficult as state space is large. If the lookup table is used, three-dimensional workspace is decomposed into $N \times N \times N$ cells where N denotes the number of size of cells in each axis workspace dimension. Furthermore, the sequential search of the candidate trajectory is difficult owing to the dynamics of the robot arm. Incorporation of human evaluation is required to realize human-friendly arm robot that means arm moving is safe for human and reasonably efficient, in the following points:

(1) a trajectory realizing minimum distance,
(2) a trajectory moving through several given points,
(3) a trajectory farthest away from obstacles,
(4) a trajectory contacting with objects, with minimizing energy functions and
(5) a trajectory moving with high or low speed motor.

Then a robot is easy to use and easy to interact with. Finally, a robot selects a behavior and makes a decision to react to human manner and human feeling. Therefore, we propose a trajectory planning method using a state-value function and an interactive genetic algorithm (IGA). The state-value function is used for estimating the human evaluation. In this research, the sum of squares of the distance between two configurations, the sum of squares of the difference among each joint angle between two configurations and the sum of the estimated evaluation values using the state-value function will be proposed. A human-friendly robot should generate its trajectory based on human evaluation in addition to the above factors.

This paper is organized as follows. Section 2 proposes arm robot including kinematics and arm robot tasking. Section 3 shows a trajectory planning method including IGA for generating a candidate trajectory and state-value function for estimating human evaluation. Section 4 shows some experimental results of trajectory generation. Section 5 concludes this paper and describes the future works.

2. A Robot Arm

In general, the trajectory planning problem is to solve the inverse kinematics of a robot arm. Since the dimensions of the configuration space are equal to the DOF of the robot arm, a trajectory planning problem can result in

* Received on January 16, 2004; accepted on July 26, 2004.
† Department of Human and Artificial Intelligent Systems, University of Fukui, 3-9-1 Bunkyo, Fukui 910-8507, Japan. E-mail: indra@iicx.ia.his.fukui-u.ac.jp
§ Electronics Engineering Polytechnic Institute of Surabaya–ITS, Kampus ITS Sukolilo, Surabaya 60111, Indonesia.
‡ Department of Mechanical Engineering, Tokyo Metropolitan University, 1-1 Minami-Osawa, Hachioji, Tokyo 192-0397, Japan. E-mail: kubota@comp.metro-u.ac.jp
¶ PREST, Japan Science and Technology Corporation (JST)

Paper No. 1345–2681/03/040129-7

a path planning problem from an initial configuration to a final configuration without the collision between the robot arm and obstacles. Here a configuration $\boldsymbol{\theta}$ is expressed by a set of joint angles:

$$\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)^T \in \Re^n \tag{1}$$

where n denotes the DOF of a robot arm.

The kinematic equation for the robot arm was found from the Denavit-Hartenberg (D-H) transformation matrix:

$$\begin{aligned} {}^0T_i &= {}^0A_1 {}^1A_2 \ldots {}^{i-1}A_i \\ &= \prod_{j=1}^{i} {}^{j-1}A_j \text{ for } i = 1, 2, \ldots, n. \end{aligned} \tag{2}$$

Here, T denotes a homogeneous transformation matrix

$${}^0T_i = \left[\begin{array}{c|c} {}^0R_i & {}^0P_i \\ \hline 0 & 1 \end{array}\right] \tag{3}$$

where

- ${}^0R_i$: orientation matrix of the ith coordinate system established at link i with respect to the base coordinate system;
- ${}^0P_i$: position vector which points from the origin of the base coordinate system to the origin of the ith coordinate system

and A denotes a transformation matrix

$${}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where

- θ_i : the joint angle from the x_{i-1} axis to the x_i axis about the z_{i-1} axis (using the right-hand rule);
- α_i : the offset angle from the z_{i-1} axis to the z_i axis about the x_i axis (using the right-hand rule);
- a_i : the offset distance from the intersection of the z_{i-1} axis with the x_i axis to the origin of the ith frame along the x_i axis;
- d_i : the distance from the origin of the $(i-1)$th to the intersection of the z_{i-1} axis with the x_i axis along the z_{i-1} axis.

The position of the end-effector is equal to the position vector ${}^0P_i$. The upper right 3×1 partitioned matrix of ${}^0T_i$ denotes respectively the position of end-effector Pef_x, Pef_y and Pef_z. It points from the origin of the base coordinate system to the end of effector.

For the arm robot in **Fig. 1**, the robot arm link parameters are shown in **Table 1**.

The robot arm position vector ${}^0P_i$ is shown

$$\begin{aligned} Pef_x &= a_2C_1C_2 + a_3C_1C_{23} - d_3S_1 \\ Pef_y &= a_2S_1C_2 + a_3S_1C_{23} + d_3C_1 \\ Pef_z &= -a_2S_2 - a_3S_{23} \end{aligned} \tag{5}$$

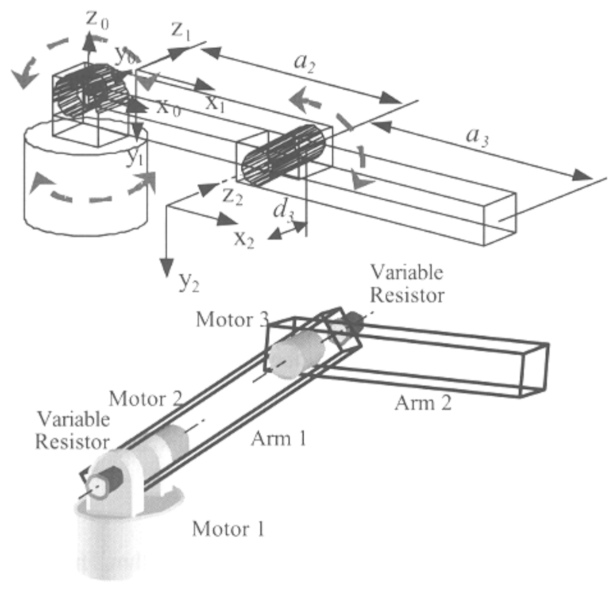


Fig. 1 A robot arm

Table 1 Robot arm link parameters

Joint i	θ_i	α_i	a_i (mm)	d_i (mm)	Joint range
1	90°	0°	0	0	0°–90°
2	0°	−90°	200	0	0°–145°
3	0°	0°	210	−50	10°–190°

where $C_i \equiv \cos\theta_i$, $S_i \equiv \sin\theta_i$, $C_{ij} \equiv \cos(\theta_i + \theta_j)$, and $S_{ij} \equiv \sin(\theta_i + \theta_j)$.

Most of tasks given to the robot arm are generally to move along a path and to move between several given points. These are corresponding to continuous path control and point-to-point (PTP) control, respectively. The trajectory planning for robot arms must satisfy the constraints such as the bounded velocity and acceleration, maximal torque, and task-dependent constraints, and also must avoid collision with obstacles.

The task of this robot arm presented in this paper is to approach toward the face of the human. This kind of task will be used in amusement such as robot is giving food to mouth etc.

3. Interactive Trajectory Generation

3.1 Trajectory planning by genetic algorithm

GAs proposed by Holland were used originally as a mechanism of natural adaptation [21]. In a GA, a candidate solution called phenotype is encoded into a finite length string called genotype, and the search is performed in the genotype space. However, since we use numerical coding, binary encoding is not used here. Furthermore, the search of a trajectory requires human evaluations, we use IGA. To reduce the number of the human evaluations, we use a state-value function estimating the human evaluation.

Therefore, we can search for good trajectory candidates with the estimated human evaluation. The procedure of the IGA for trajectory generation is shown in **Fig. 2**.

The representation of a candidate trajectory is shown in **Fig. 3**. While C_0 and C_m show the starting point and the

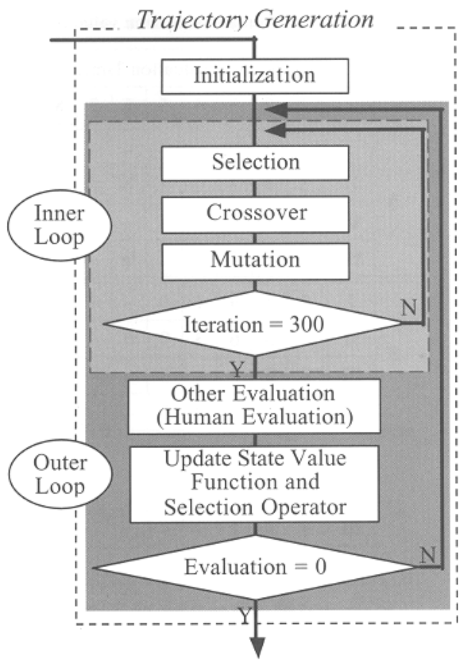


Fig. 2 The procedure of interactive genetic algorithm (IGA)

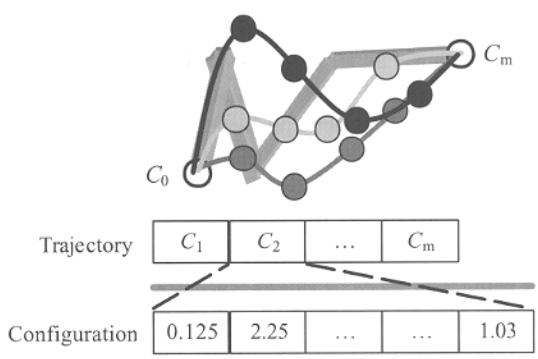


Fig. 3 The representation of a candidate trajectory

final point of the end-effector, each trajectory configuration represented the angular of each joint angle.

Initialization randomly generates an initial population of candidate trajectory and fitness value is assigned to each individual according to the evaluation function. 'Delete Least Fitness' (DLF) is used as selection scheme, which removes the worst individual selected from two host individuals. Next, one individual is randomly selected from the population.

Here we use an elite crossover incorporating some genetic information from the best individual with crossover probability. Consequently, the worst individual is replaced with the individual generated by the elite crossover. Mutation is performed by using the normal random variables with mean zero,

$$x_{i,j} \leftarrow x_{i,j} + \left(\delta_j \cdot \frac{fit_i - fit_{\min}}{fit_{\max} - fit_{\min}} + \epsilon_j\right) \cdot N(0,1) \quad (6)$$

where $x_{i,j}$ denotes the jth joint angle of the ith individual; $fit_{\min}$ and $fit_{\max}$ are the minimal and maximal values of the fitness values, respectively. δ_j denotes coefficient, ϵ_j is offset value. $N(0,1)$ denotes the normal distribution with mean of 0 and variance of 1. These processes are repeated

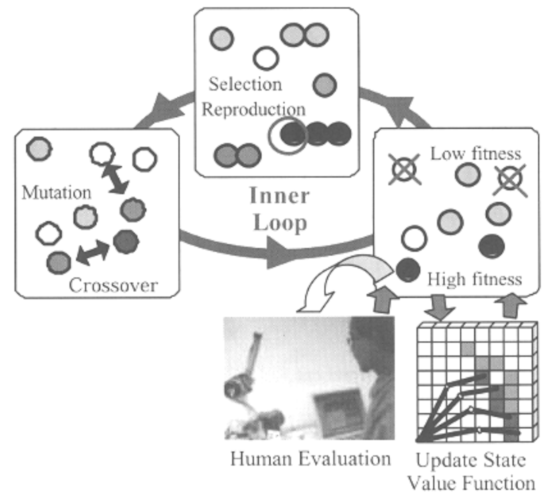


Fig. 4 The concept of interactive genetic algorithm (IGA)

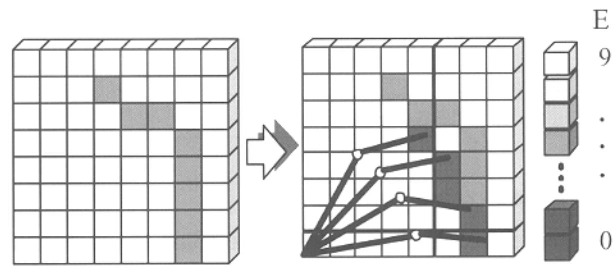


Fig. 5 The update of state-value function based on the path of the end of effector

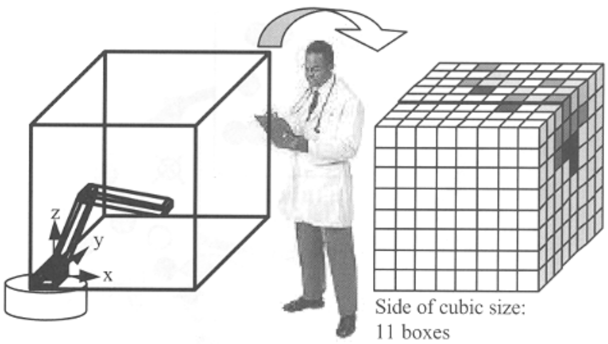
until the local termination condition is satisfied. Here we use the maximal number of generations in inner optimization as the termination condition. After the genetic process, the best trajectory with the minimal fitness value is displayed to the human. And then, the human evaluates the trajectory and scores a value within 0 and 9 and 0 is excellent. Next, the state-value function is updated (see **Fig. 4**).

The objective is to generate a trajectory realizing the minimum distance from the initial configuration to the final configuration and high human evaluation. To achieve the objective, we use the fitness function given by

$$fit_i = w_1 f_p + w_2 f_d + w_3 f_V \quad (7)$$

where w_1, w_2, and w_3 are weight coefficients. The first term, f_p, denotes the sum of squares of the distance between two configurations. The second term, f_d, denotes the sum of squares of the difference among each joint angle between two configurations. The third term, f_V, denotes the sum of the estimated evaluation values using the state-value function. Therefore, this trajectory planning problem can result in a minimization problem.

Figure 5 describes the update of state-value function based on the path of the end of effector in two dimension. The left one shows the path result in the previous action. After updating the state-value function, the path of the end-effector of the next motion is described by using the dark blue color. E denotes human evaluation. In this figure, the darkness blue color shows the good value evaluation or better result.



(a) An actual environment (b) Discrete cell space

Fig. 6 State-value function based on human evaluations

Table 2 Parameters of IGA

Population Size	100
Chromosome Length	15
Crossover Probability	0.5
Mutation Probability	1.0
Local Evaluations (inner loops)	300

3.2 A state-value function

We use a lookup table of a state-value function. The lookup table with three-dimensional arrays corresponds to the discrete cell space. By using the human evaluation, the elements of lookup table are updated according to the path of the end-effector of the robot arm (Fig. 5). The updating rule of the state-value function is as follows:

$$value(x, y, z) \leftarrow (1-\gamma)value(x, y, z) + \gamma E \quad (8)$$

where $value(x, y, z)$ is the estimated evaluation value of the cell (x, y, z); γ is the step size and E is human evaluation. Therefore, the sum of the estimated evaluation values f_V in Eq. (7) is described as follows:

$$f_V = \sum_{(x,y,z)\in H} value(x, y, z) \quad (9)$$

where H indicates cells on the path of the end-effector of the robot arm.

Figure 6 describes the state-value function based on human evaluation. Actual environment on the left figure is defined as a discrete cell space on the right figure. We define that the cubic size is $11 \times 11 \times 11$ boxes of cell space, which means the number of boxes for composing the cubic space.

4. Experimental Results

This section shows some experimental results of the interactive trajectory generation of a robot arm shown in Fig. 1. The task of the robot arm is to approach toward the face of the human as shown in **Fig. 7**. **Table 2** shows the parameters of IGA.

A set condition of the experiment was taken with the condition weight coefficients of fitness function in Eq. (7), $w_1 + w_2 + w_3 = 1$. There are three conditions chosen in this research and for each condition, four experiments were conducted.

Table 3 The history steps of human evaluation value for each condition

		Human Evaluation Times							
		1	2	3	4	5	6	7	8
A	I	1	3	0	–	–	–	–	–
	II	3	4	3	2	1	4	2	0
	III	1	1	2	0	–	–	–	–
	IV	2	2	3	1	1	0	–	–
B	I	6	5	9	1	0	–	–	–
	II	5	3	6	3	9	0	–	–
	III	9	2	1	1	0	–	–	–
	IV	2	3	1	7	0	–	–	–
C	I	1	1	3	9	0	–	–	–
	II	0	–	–	–	–	–	–	–
	III	9	3	1	4	0	–	–	–
	IV	2	0	–	–	–	–	–	–

A: $w_1 = 0.01, w_2 = 0.34, w_3 = 0.65$
B: $w_1 = 0.01, w_2 = 0.01, w_3 = 0.98$
C: $w_1 = 0.25, w_2 = 0.50, w_3 = 0.25$

Table 3 shows the history steps of human evaluation value for each condition. The value of the human evaluation represented by 0 is for good value and the increasing value shows the bad valued evaluation.

Figure 7 shows a set condition of experiment of arm robot. The number above the picture shows the initial configuration and the final configuration of each experiment on each condition. At the final trial, the state-value function estimates the human evaluation to the trajectory of the robot arm. The sizes of boxes show the human evaluation value that is produced by IGA, where the bigger size is for good evaluation whereas the smaller one is for bad evaluation.

Figures 8 and 9 show the history steps for each experiment in each condition versus the human evaluation, the fitness function, the trajectory length, the potential value of joint angle and the sum of angle of each joint angle.

The number of the history steps in condition A (i.e., $w_1 = 0.01, w_2 = 0.34, w_3 = 0.65$) having the most step is 8; in condition B (i.e., $w_1 = 0.01, w_2 = 0.01, w_3 = 0.98$) it is 6; and in condition C (i.e., $w_1 = 0.25, w_2 = 0.50, w_3 = 0.25$) it is 5. In the condition C, one of them is only one step; it means the first configuration is the last configuration because the arm motion is directly satisfied toward the human face.

As shown in Figs. 8 and 9, although the trajectory length becomes longer than the previous one but the trajectory is satisfied for human. From the equation of fitness function in (7), the second term is the most affected to the history steps.

5. Conclusion

This paper has proposed an interactive trajectory generation method using genetic algorithm and state-value function. Genetic algorithms (GAs) as known as a 'beam search' are working well to generate the trajectory in unknown space. The search of a trajectory requires human evaluations. GAs offer us multiple sets of candidate tra-

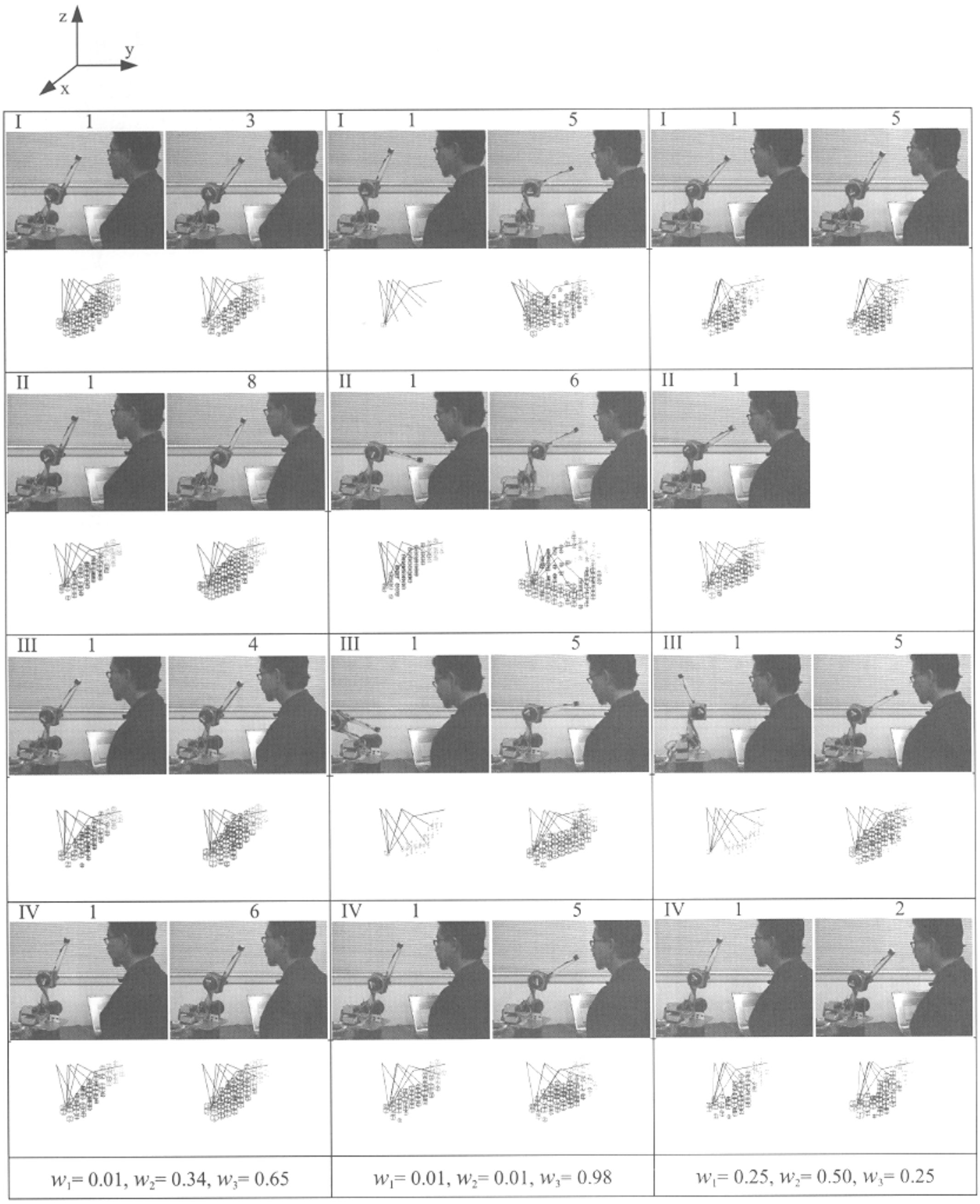


Fig. 7 A set condition of experiment of arm robot, where the first pictures of each block show the initial configuration and the second ones are the final configuration; at the final configuration, the state-value function estimates the human evaluation to the trajectory of the robot arm, in which the sizes of boxes (big size is for good evaluation and small size is for bad evaluation) show the human evaluation value that is produced by IGA

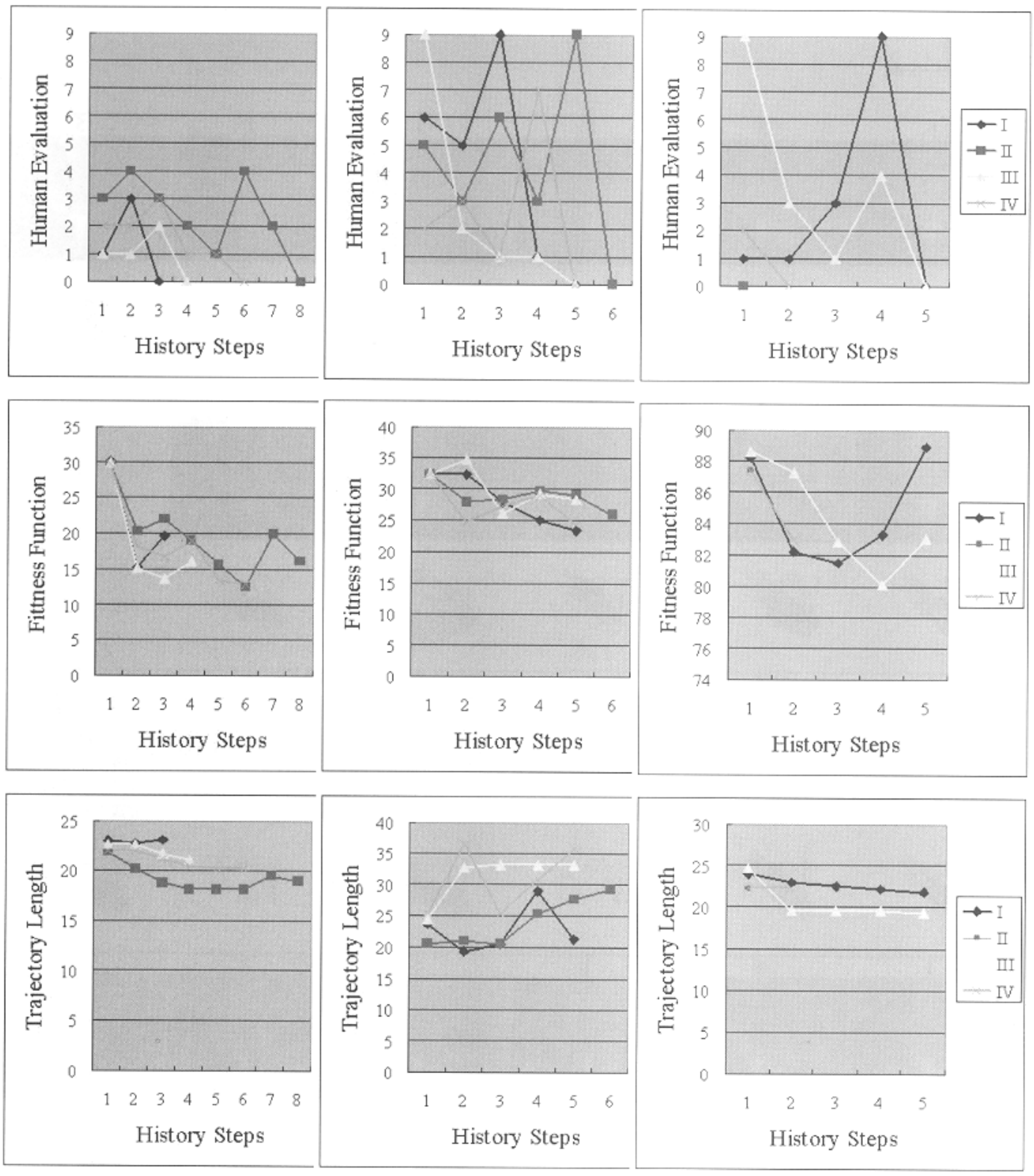
jectory and then the best one is shown to the human to be evaluated. Furthermore, to reduce the number of the human evaluations, we use a state-value function estimating the human evaluation. Therefore, we can search for good trajectory candidates with the estimated human evaluation. The experimental results show that the number of human evaluations can be reduced and furthermore, the state-value function can estimate the human evaluation.

The best condition for weight parameters is $w_1 = 0.25, w_2 = 0.50, w_3 = 0.25$. The history step of this condition is less than that of other conditions. The corresponding trajectory length is also decreased.

As future works, we will discuss how to extract human feeling through interaction with human and how to estimate the effectiveness of human evaluation.

Acknowledgements

The authors express the thanks to Shuzo Yamaji and Masanori Ishiguro for their help in the development of the robot arm. The author (Indra) also wishes to express the deepest thanks to the Japanese Government for Monbukagakusho Scholarship during taking the graduate study in Japan.



$w_1 = 0.01$, $w_2 = 0.34$, $w_3 = 0.65$ $w_1 = 0.01$, $w_2 = 0.01$, $w_3 = 0.98$ $w_1 = 0.25$, $w_2 = 0.50$, $w_3 = 0.25$

Fig. 8 The history steps versus the human evaluation, the fitness function, and the trajectory length

References

[1] O. Khatib, K. Yokoi, O. Brock, K.-S. Chang, and A. Casal, "Robots in human environments," *Archives of Control Sciences*, vol. 11, no. 3/4, pp. 123–137, 2001.

[2] N. Kubota, D. Hisajima, F. Kojima, and T. Fukuda, "Fuzzy and neural computing for communication of a partner robot," *Journal of Multi-Valued Logic and Soft Computing*, vol. 9, no. 2, pp. 221–239, 2003.

[3] S. Kristensen, S. Horstmann, J. Klandt, F. Lohnert, and A. Stopp, "Human-friendly interaction for learning and cooperation," in *Proc. of the 2001 IEEE ICRA*, 2001, pp. 2590–2595.

[4] A. M. Arsenio, "Embodied vision—Perceiving objects from actions," in *Proc. of the 12th IEEE Workshop Robot and Human Interactive Communication RO-MAN 2003*, 2003, pp. 365–371.

[5] J.-S. R. Jang, C. -T. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[6] S. J. Russell and P. Norvig, *Artificial Intelligence*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[7] J. A. Anderson and E. Rosenfeld, *Neurocomputing—Foundations of Research*. Cambridge, MA: The MIT Press, 1988.

[8] J. C. Bezdek, "What is computational intelligence?," in *Computational Intelligence—Imitating Life*, J. M. Zurada, R. J. Marks II, C. J. Robinson, Eds. Piscataway, NJ: IEEE Press, 1994, pp. 1–12.

[9] R. P. Paul, *Robot Manipulators: Mathematics, Programming, and Control*. Cambridge, MA: The MIT Press, 1981.

[10] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics: Control, Sensing, Vision, and Intelligence*. New York, NY: McGraw-Hill, 1987.

[11] T. Fukuda, N. Kubota, and T. Arakawa, "GA algorithms in intelligent robots," in *Fuzzy Evolutionary Computation*, Dortrecht, The Netherlands: Kluwer Academic Publishers, 1997, pp. 81–105.

[12] T. Arakawa and T. Fukuda, "Natural motion trajectory generation of biped locomotion robot using genetic algorithm through energy optimization," in *Proc. of The IEEE Int. Conf. on Systems, Man, and Cybernetics*, 1996, pp. 1495–1500.

[13] N. Baba and N. Kubota, "Collision avoidance planning of a robot manipulator by using genetic algorithm—A consideration for the problem in which moving obstacles and/or several robots are in the workspace," in *Proc. of The First IEEE Conf. on Evolutionary Computation*, 1994, pp. 714–719.

[14] T. Lozano-Perez, "Automatic planning of manipulator transfer movements," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 11, pp. 681–698, 1981.

[15] R. A. Brooks, "Planning collision-free motions for pick-and-place operation," *The Int. J. of Robotics Research*, vol. 2, no. 4, pp. 19–44, 1983.

[16] O. Khatib, "Real-time obstacle avoidance for manipulators and mo-

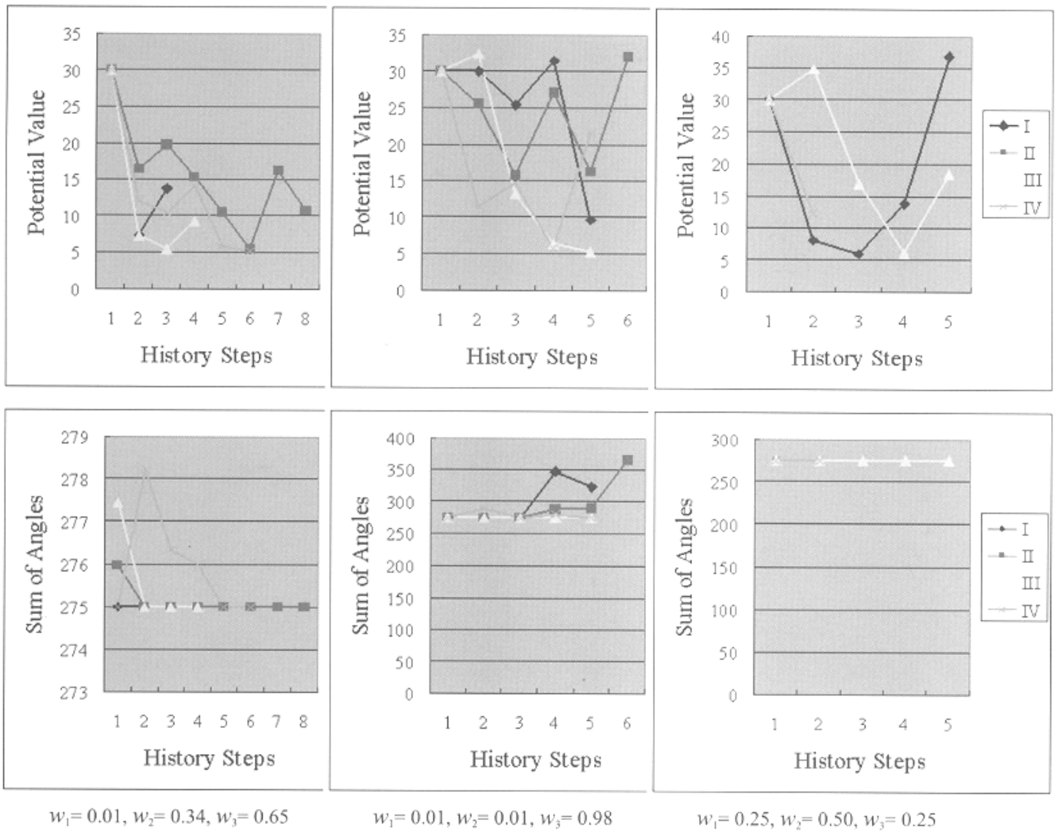


Fig. 9 The history steps versus the potential value of joint angle and the sum of squares of the difference between joint angles for two configurations

bile robots," *The Int. J. of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[17] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.

[18] T. Tsuji, S. Nakayama, and K. Ito, "Parallel and distributed trajectory generation of redundant manipulators through cooperation and competition among subsystems," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 27, no. 3, pp. 498–509, 1997.

[19] R. A. Brooks, "Intelligence without representation," *Artificial Intelligence*, vol. 47, no. 1/3, pp. 139–159, 1991.

[20] D. B. Fogel, *Evolutionary Computation*. Piscataway, NJ: IEEE Press, 1995.

[21] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

[22] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA: Addison Wesley, 1989.

[23] J. Xiao, Z. Michalewicz, L. Zhang, and K. Trojanowski, "Adaptive evolutionary planner/navigator for mobile robots," *IEEE Trans. Evolutionary Computation*, vol. 1, no. 1, pp. 18–28, 1998.

[24] N. Kubota, T. Fukuda, and K. Shimojima, "Trajectory planning of cellular manipulator system using virus-evolutionary genetic algorithm," *Robotics and Autonomous Systems*, vol. 19, pp. 85–94, 1996.

[25] N. Kubota, I. A. Sulistijono, and F. Kojima, "Interactive genetic algorithm for trajectory generation of a robot manipulator," in *Proc. of the 4th Asia-Pacific Conf. on SEAL*, 2002, pp. 146–150.

[26] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

Biographies

Indra Adji Sulistijono received a B.E. degree from the Institute of Technology Sepuluh November Surabaya (ITS), Indonesia in 1993. From 1994, he joined with the Electronics Engineering Polytechnic Institute of Surabaya–ITS. From April 2003, he joined Graduate School of Engineering, University of Fukui, Japan. His research interests are in intelligent robotic system and mechatronics. He is a student member of The Institute of Electrical and Electronic Engineers (IEEE).

Naoyuki Kubota received a B.E. degree from the Osaka Kyoiku University in 1992, and M.E. degree from the Hokkaido University in 1994, and a D.E. degree from the Nagoya University in 1997. He received the Best Paper Award of IECON 96 and the Best Paper Award of CIRA 97. From April 1997, he has been with the Department of Mechanical Engineering at Osaka Institute of Technology. From 1999 to 2004, he joined with the Department of Human and Artificial Intelligent Systems at University of Fukui, Japan. Currently, he is an Associate Professor in Department of Mechanical Engineering, Tokyo Metropolitan University, Japan. His research interests are in computational intelligence, intelligent robotic systems and intelligent manufacturing systems. He is a member of the Robotics Society of Japan (RSJ), the Japan Society of Mechanical Engineers (JSME), the Society of Instrument and Control Engineers (SICE), The Institute of Electrical and Electronic Engineers (IEEE), etc.