Proceedings of the 2009 IEEE
International Conference on Robotics and Biomimetics
December 19 -23, 2009, Guilin, China

Weeding Manipulator Exploiting Its Oscillatory Motion for Force Generation: Parameter Optimization of VDP Oscillator using Real-Coded Genetic Algorithm

Jun Kobayashi

***Abstract*— This paper addresses parameter optimization of Van der Pol (VDP) oscillator driving a weeding manipulator, which exploits its own oscillatory motion to achieve efficient force generation. The author has confirmed the effectiveness of the exploitation in force generation using simulations. The simulation results have shown that a method using the VDP oscillator attains a superior performance to one using a sinusoidal wave function. In the simulations, however, the parameters of the VDP oscillator have been chosen after several attempts. Therefore, if the parameters are adjusted by an appropriate way, the performance of the force generation method will be improved. In this paper, real-coded genetic algorithm (RCGA), which is an optimization technique, is applied to the parameter optimization of the VDP oscillator because it is easily implementable for nonlinear systems. Simulation results show that RCGA succeeded in finding out sets of better parameters than ones used in the previous study.**

I. Introduction

Weeding is a necessary task to keep a garden and a park beautiful, but you probably hate to be involved in such a task because of its tediousness. Therefore, a robot will make you happy if it can execute the task without human intervention.

The weeding manipulator dealt with in this paper pulls out weeds one by one, unlike mowing machines that have been already used out there. Mowing machines leave the roots of weeds in the ground, but the weeding manipulator try to remove weeds with the roots in the same way human beings do. This approach prevents regrowth of weeds.

The weeding robot is required to generate a large force because some weeds have roots spreading deeply and tightly. Performance of the robot could be improved by providing powerful actuators, but such a robot would consume energy uselessly owing to the heavy weight of the powerful actuators. Efficiency is an important factor especially for robots working outdoors because their energy sources are usually batteries. Inefficient robots could not execute enough tasks within its uptime.

Some studies have been conducted to realize efficient robots that exploit their characteristics to the maximum to achieve an efficient behavior. Papadopoulos and Gonthier have introduced the force workspace, which is a map indicating the locations where a robot can apply a given force [1]. With the workspace, they have showed that force capabilities of manipulators can be improved by employing base mobility and manipulator redundancy. Imamura and Kosuge have proposed virtually unactuated joints so that a manipulator could generate a larger force than the load capacity [2]. Kobayashi, Kishida and Ohkawa have formulated an optimization problem and solved it to find out work postures in which a robot manipulator realizes a force as large as possible [3].

In terms of efficient force generation, the author has demonstrated that a manipulator exploiting its own oscillatory motion can efficiently generate a large force [4]. In the study, van der Pol (VDP) oscillator was used to drive the manipulator in order to achieve efficient force generation. Thanks to an entrainment property of the VDP oscillator, a command signal to the manipulator can synchronize with the motion of the manipulator, and then the synchronization leads the manipulator to generate a large force by smaller amount of joint driving torque. It reduces copper loss of motors of the manipulator, so the energy efficiency would be increased. However, the author determined the parameters of the VDP oscillator after several attempts; the parameters are not the optimum ones presumably. Therefore, it would be possible to improve the performance by using an appropriate way that finds out a set of better parameters.

In this paper, Real-Coded Genetic Algorithm (RCGA), which is a Genetic Algorithm (GA) handling real-valued vectors instead of bit strings, is adopted as the appropriate way for the parameter optimization. It is difficult to apply an optimization method using the information about the weeding manipulator driven by the VDP oscillator because of the complexity of its mathematical expressions. However, RCGA can be implemented without difficulty because it is a metaheuristic optimization method that requires no prior information about the robotic system. The validity of the parameter optimization method using RCGA was verified by simulations.

II. Simulation Models

This section explains a weeding manipulator model and a weed model that were used in simulations. Open Dynamics Engine (ODE) was used to create the models [5]. ODE is an open source, high performance library for simulating rigid body dynamics. All simulations described in this paper are carried out based on the weeding manipulator model and the

J. Kobayashi is with the Department of Systems Design and Informatics, Kyushu Institute of Technology, Kawazu 680-4, Iizuka, Fukuoka, 820-8502, Japan (corresponding author to provide phone: +81-948-29-7747; fax: +81-948-29-7709; e-mail: jkoba@ces.kyutech.ac.jp).

weed model created using rigid body objects of ODE. The version of ODE used in the simulations is 0.11.

A. Weeding Manipulator Model

The model of the weeding manipulator created on the simulation program using ODE is shown in Fig. 1. The weeding manipulator consists of four rigid links, four joints and a base. It is assumed in the simulation program that the stiffness of each joint is variable. Variable stiffness joints have been realized in several ways, e.g. [6] and [7], and the author also has been developing such a joint using nonlinear springs. One reason for the assumption is that the robot exploiting its own oscillatory motion needs elasticity in its dynamics. The other reason is that the variable stiffness joints will probably be an essential property for upcoming robots that work in a living environment together with people for safety. In the manipulator model, when it try to pull out a weed, the third joint becomes a free joint and the fourth joint becomes a spring joint, thereby two distal links painted yellow work like a spring. This condition is feasible under the assumption that the stiffness of the joints is variable. On the other hand, when a typical task is assigned to the manipulator, all four links are used ordinarily. For example, if the typical task is a positioning of an object, the stiffness of the all four joints is set to rigid to achieve the task accurately.

In the following, the position of the tip of the manipulator that is connected to the weed model is designated by symbols x_t and y_t, and the position of the third joint that is a connecting point between the distal links that work like a spring and the proximal links is designated by symbols x_{t2} and y_{t2}. The spring constant of the fourth joint is set to 1.0 Nm/rad. The other parameters of the manipulator are set as follows: the length and the mass of the first and second links are 0.1 m and 0.5 kg respectively, the length and the mass of the third and fourth links is 0.05 m and 0.25 kg respectively. The damping coefficient of all joints is 0.01 Nms/rad. The torques driving the joints are limited to 1.0 Nm.

B. Weed Model

Since an actual restoring force of a weed is complicated, it is impossible to express it completely in a mathematical way. Hence, a simple restoring force model was adopted as a feasible way in the simulation model.

A weed is modeled using a rectangular solid object of ODE. The green object shown in Fig. 1 is the weed model. When the weed model is moved from its base point, the restoring force is generated as shown in Fig. 2. The red arrow indicates the vector of the restoring force. The magnitude of the force vector depends on the displacement from the base point, which is designated by Δx in equations described later. The restoring force acts on the weed model and the direction of the force vector points toward the base point; the weed model is brought back to the base point by the restoring force.

The restoring force is decomposed into an elastic force and a viscous force. Equation (1) and (2) describe the elastic and viscous restoring forces respectively.

$$f_{elastic} = \begin{cases} 0.0 & (\Delta x < d_{\min}) \\ k \cdot (\Delta x - d_{\min}) & (d_{\min} \le \Delta x \le d_{\max}) \\ 0.0 & (\Delta x > d_{\max}) \end{cases} \tag{1}$$

$$f_{viscous} = \begin{cases} c \cdot d\Delta x / dt & (\Delta x \le d_{\max}) \\ 0.0 & (\Delta x > d_{\max}) \end{cases} \tag{2}$$

The d_{min} is a radius of the dead zone of the elastic restoring force, which is represented by the circular area in light blue in Fig. 2. The viscous restoring force acts even in the dead zone. The dead zone eliminates unusual vibratory motion from the weed model. The blue circle represents the effective range of the two restoring forces; out of the range the restoring forces have no effect. The radius of the effective range is d_{max}. The weed displacement of more than d_{max} means that the weed has been pulled out by the manipulator.

The parameters of the weed model are set as follows: the mass of the weed model is 0.08 kg, the radius of the dead zone d_{min} is 0.003 m, and the effective range of the restoring force d_{max} is 0.01 m. The coefficients k and c are set to $f_{max}/(d_{max}-d_{min})$ and 100 Ns/m respectively, where f_{max} is 50 N.

III. Control Scheme

A control scheme to oscillate the weeding manipulator for force generation is discussed in the following subsections.

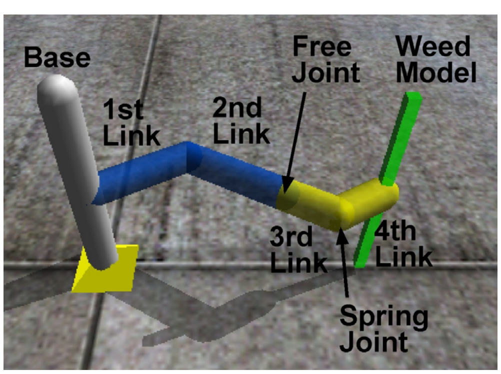

Fig. 1. Weeding manipulator model & weed model created using ODE.

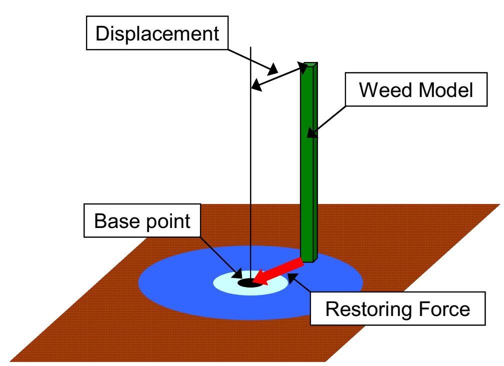

Fig. 2. Schematic diagram of the restoring force of the weed model.

A. Outline

This subsection outlines the flow of the control scheme. First, a desired tip trajectory, $x_{t2d}(t)$ and $y_{t2d}(t)$, for oscillating the manipulator is generated. Since the trajectory generation process is the most important part of this scheme, the process is elaborated on in the next subsection.

Second, desired joint angles, $q_{1d}(t)$ and $q_{2d}(t)$, are derived from the desired tip trajectory using the following inverse kinematics equations.

$$q_{1d}(t) = \cos^{-1}\left(\frac{x_{t2d}^2(t) + y_{t2d}^2(t) + l_1^2 - l_2^2}{2l_1\sqrt{x_{t2d}^2(t) + y_{t2d}^2(t)}}\right) + \tan^{-1}\left(\frac{y_{t2d}(t)}{x_{t2d}(t)}\right) \tag{3}$$

$$q_{2d}(t) = -\cos^{-1}\left(\frac{x_{t2d}^2(t) + y_{t2d}^2(t) - l_1^2 - l_2^2}{2l_1 l_2}\right) \tag{4}$$

If the desired tip position is out of a reachable region of the manipulator, the desired joint angles calculated one time step before are substituted for the current desired joint angles.

Finally, the torques driving the joints to achieve the desired joint angles are calculated by the following simple control law. The proportional gain K_p is set to 100.

$$\tau_1(t) = K_P \cdot \{q_{1d}(t) - q_1(t)\} \tag{5}$$

$$\tau_2(t) = K_P \cdot \{q_{2d}(t) - q_2(t)\} \tag{6}$$

The manipulator following this control scheme becomes an oscillatory state; thereby it can generate a large force.

B. Trajectory Generation Using VDP Oscillator

The trajectory generation method uses the VDP oscillator. The desired tip trajectory for force generation is determined based on the state of the VDP oscillator. Since the VDP oscillator has a frequency entrainment property [8], the desired tip trajectory synchronizes with the motion of the manipulator. This adjustability achieves better performance than the method using a sinusoidal wave function [5].

This trajectory generation method is expressed by the following equations.

$$x_{t2d}(t) = x_{t2}(0) - a_x \cdot t \tag{7}$$

$$y_{t2d}(t) = \begin{cases} y_{t2}(0) & (t < t_{osc}) \\ y_{t2}(0) + a_y \cdot (t - t_{osc}) \cdot \dot{x}_v(t) & (t \geq t_{osc}) \end{cases} \tag{8}$$

$$\ddot{x}_v + \varepsilon(x_v^2 - 1)\dot{x}_v + \omega_v^2 x_v = G_{in}\dot{y}_{t2} \tag{9}$$

The symbol of t_{osc} is a time to start oscillating the manipulator, which is set to 0.5 s in this study. The a_x and a_y are set to 0.005 and 0.001 respectively. As explained in the equations, during the first 0.5 s, the manipulator only draws the weed toward its base along the x-axis without oscillation. Then the oscillation of the tip of the manipulator along the y-axis is started to pull out the weed.

With respect to the y-axis motion, the desired tip trajectory is calculated using the state of the VDP oscillator as shown in (8). The differential equation in (9) expresses the dynamics of the VDP oscillator, where x_v is its state, ε is a positive parameter controlling the damping term, f_v is its natural frequency, $\omega_v = 2\pi f_v$, and G_{in} is an input gain. The gain G_{in} should be high so that the VDP oscillator can entrain to the motion of the manipulator. These parameters are set as follow: $\varepsilon = 1.0$, $f_v = 0.5$ Hz, $G_{in} = 2000$. The initial state of the VDP oscillator is set to $x_v = 1.0$, $dx_v/dt = 0.0$. Using the derivative of the x_v and y_{t2} in this method has the effect of removing any DC components from these signals [9].

Although the parameters, f_v, ε, G_{in}, and a_y have been chosen after several attempts here, they are adjusted by RCGA in the next section in order to improve the performance of the control scheme.

C. Simulation Results

Simulations were conducted to evaluate the effectiveness of the control scheme using the manipulator model and the weed model shown in Fig. 1. In the ODE simulations, the time step size for numeric integration was set to 1 ms.

To begin with, let us describe a simulation in which the manipulator keeps only drawing the weed toward its base; in this case the manipulator does not exploit its oscillatory motion. Fig. 3 shows the force generated by the manipulator only drawing the weed without an oscillatory motion. In this case, the weeding manipulator failed to pull out the weed model because the generated force was approximately 25.8 N, which is too small to pull out the weed model with the physical parameters described in Section II-B.

Fig. 4 shows the force generated by the manipulator driven by the control scheme using the VDP oscillator. In this case, the manipulator was able to generate a larger force, and then succeeded in weeding. Fig. 5 shows the state of the VDP oscillator and the tip velocity along the y-axis of the manipulator. Thanks to the entrainment property of the VDP oscillator, its state almost synchronized with the tip motion of the manipulator. Since the desired tip trajectory was calculated from the synchronized state, the large force generation was achieved. A discussion about the efficiency of the control scheme can be found in [5].

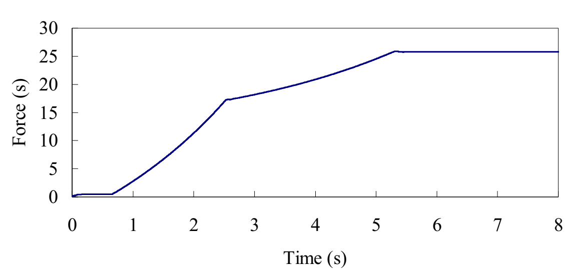

Fig. 3. Force generated by the manipulator not exploiting its oscillatory motion. The manipulator only drew the weed without oscillatory motion.

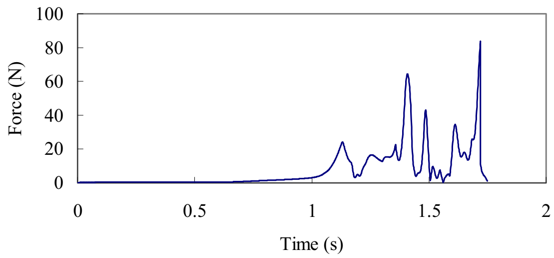

Fig. 4. Force generated by the manipulator driven by the method using the VDP oscillator: f_v = 0.5 Hz, ε = 1.0, G_{in} = 2000, a_y = 0.001.

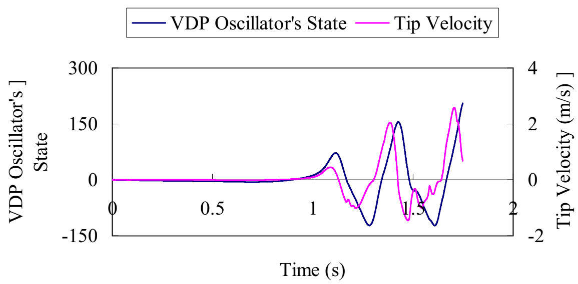

Fig. 5. VDP oscillator's state dx_v/dt and the tip velocity of the manipulator dy_{t2}/dt: f_v = 0.5 Hz, ε = 1.0, G_{in} = 2000, a_y = 0.001.

IV. Parameter Optimization Using RCGA

In the previous section, the parameters of the VDP oscillator have been chosen after several attempts. Therefore, the performance of the control scheme could be improved by adjusting the parameters in a more proper way.

In this study, RCGA is adopted as the way to seek a set of better parameters. There are two reasons for the adoption. The first reason is complexity of mathematical expressions of the weeding manipulator and the VDP oscillator. Since GA is a metaheuristic optimization method, it requires no prior information about the complicated mathematical expressions. Therefore, RCGA can be implemented without any difficulty. The second reason is that RCGA can directly handle real-valued vectors, while general GAs use bit strings to describe candidates for a solution of an optimization problem. Using RCGA, you don't need to decode the parameters of the VDP oscillator into a bit string.

In reality, it is almost impossible to execute RCGA for a real weeding manipulator to find out the optimum parameters because the process is huge time-consuming for experiments. However, parameters obtained by RCGA in simulations could be effectively used as initial parameters for a real weeding manipulator, which might need a minor adjustment after the first implementation.

In the following subsections, RCGA used in this study is described, and finally simulation results that verified the validity of the parameter optimization using the RCGA are shown.

A. Individuals

The flowchart shown in Fig. 6 explains the procedure of general GAs. First of all, an initial population, a set of individuals representing candidates for a solution of an optimum problem, is generated. After that, three processes, selection, crossover, and mutation, are repeatedly executed until you get a satisfying solution of an optimization problem.

In RCGA, the individual is a real-valued vector. The following equation shows an individual dealt with in this study. The individual consists of the parameters of the VDP oscillator.

$$\boldsymbol{x} = \left(f_v,\ \varepsilon,\ G_{in},\ a_y\right) \tag{10}$$

Based on the knowledge obtained from the former simulation results, the range of the four parameters are restricted as follows:

$$f_v \in [0, 10],\ \varepsilon \in [0, 10],\ G_{in} \in [0, 10000],\ a_y \in [0, 0.1].$$

B. Selection & Fitness Function

In the selection process, better individuals survive and worse individuals are culled out of the population. In this study, a tournament selection is used to pick the superior individuals among the population. Specifically, new generation is produced by the following procedure.

Step 1: make the pairs of individuals
Step 2: evaluate their respective values of fitness
Step 3: select the winner based on the values of fitness
Step 4: keep the winners and replace the losers with new individuals

The new individuals in the step 4 are produced from the pairs by crossover and mutation described in the following subsections.

The individuals are evaluated based on an elapsed time for weeding; if an individual finishes a weeding task faster than another one, a higher value of fitness will be given to the individual. The fitness is calculated by

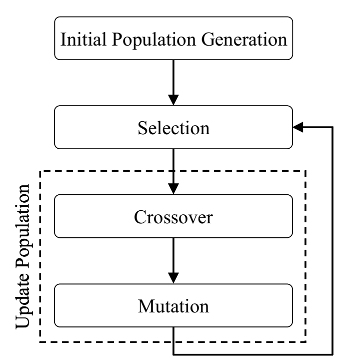

Fig. 6. Flowchart of Genetic Algorithm.

$$fitness = \begin{cases} t_e^{-1} & (t_e \le 5.0) \\ 0.0 & (t_e > 5.0) \end{cases} \tag{11}$$

where t_e is an elapsed time for weeding.

C. Crossover

There are some crossover methods developed specially for RCGA, such as UNDX [10] and SPX [11]. Because of its simplicity, the BLX-α [12] is adopted for the parameter optimization of the VDP oscillator.

Let us give an illustration of the BLX-α for parental individuals, $\boldsymbol{x} = (x_1, x_2)$ and $\boldsymbol{y} = (y_1, y_2)$ in two dimensional space. The area in gray in Fig. 7 is a pool of the candidates for their child, where $I_1 = |x_1 - y_1|$, $I_2 = |x_2 - y_2|$, and α is a user specified GA-parameter. The parameter α is set to 0.3 in the simulations described later. The child individual vector is randomly picked from the gray area. For example, when the individual $\boldsymbol{y}$ is a looser and the individual $\boldsymbol{z}$ is chosen as a child of the parents, the looser $\boldsymbol{y}$ is replaced with the child $\boldsymbol{z}$.

D. Mutation

Mutation is another way to produce the new individual, in which an element of individuals is altered randomly at a certain rate. The rate is called mutation rate, which is set to 0.25 in the simulation. As a result, one of the elements of the individual will be probably changed because the real-valued vectors dealt with in this study have four elements.

E. Simulation Results

In the simulation program written in C, drand48() library function was used to generate uniformly distributed pseudo-random numbers between 0.0 and 1.0. The property of the weed model f_{max} was set to 50N, 45N and 40N. The RCGA process was iterated 10000 times.

Fig. 8 shows time histories of the maximum values of fitness in each simulation. The maximum values of fitness converged at around 1.28.

Fig. 9 shows elapsed times for weeding t_e. The parameter optimization method using RCGA shortened the elapsed time clearly. In the figure, "Manual" indicates the VDP oscillator's parameters used in the simulation described in the section III; the parameters were chosen manually. "RCGA(50N)" is a set of the parameters adjusted by RCGA in the simulation where the weed model property f_{max} was set to 50N. Fig. 10 shows the force generated by the manipulator driven by the method using the VDP oscillator with RCGA(50N). Fig. 11, 12, 13, and 14 show the adjusted parameters, f_v, ε, G_{in}, and a_y, respectively. The parameter optimization method using RCGA succeeded in modifying the parameters for rapid weeding.

The all adjusted parameters except for the natural frequency depend on the weed model property f_{max}. TABLE I shows elapsed times for weeding under some conditions, and the mark 'X' means that the weeding was failed. According to TABLE I, RCGA(40N) that was optimized under the condition f_{max} = 40 N were also useful for the weeds with f_{max} = 45 N and 50 N. However, the weeding manipulators driven by the VDP oscillator with RCGA(45N) and RCGA(50N) failed to pull out the weed with f_{max} = 40 N. This result suggests that the condition of the environment, in which the optimization is executed, affects the optimization result. To solve this problem, RCGA parameter optimization in changing environment will be introduced in the future work.

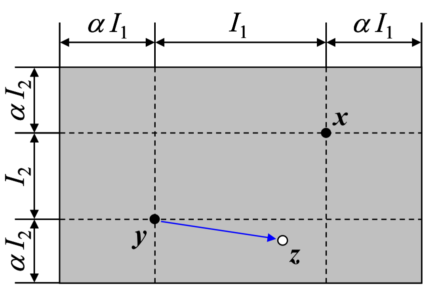

Fig. 7. Diagram of BLX-α.

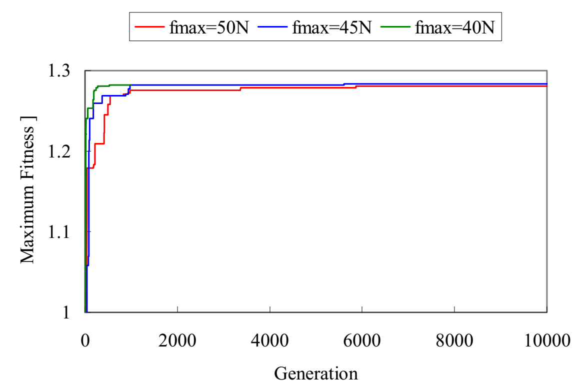

Fig. 8. Time Histories of Maximum Values of Fitness.

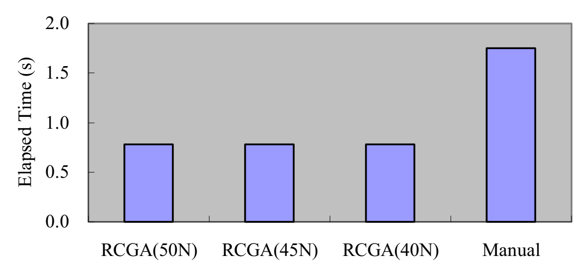

Fig. 9. Elapsed Times for Weeding.

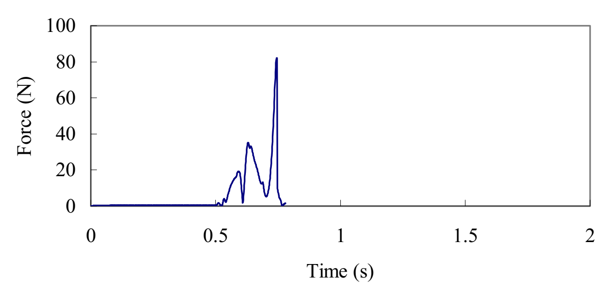

Fig. 10. Force generated by the manipulator driven by the method using the VDP oscillator with RCGA(50N) parameter set.

V. CONCLUSIONS

The weeding manipulator driven by the VDP oscillator exploits its own oscillatory motion for large force generation. The VDP oscillator put out a command signal synchronized with the motion of the manipulator; thereby the manipulator can efficiently generate a large force for weeding. In this paper, the parameter optimization method using RCGA was introduced to adjust the four parameters of the VDP oscillator. Although the mathematical expressions of the manipulator driven by the VDP oscillator are complicated, the optimization method was easily implemented because RCGA is a metaheuristic optimization technique that requires no prior information about the complicated mathematical expressions. The simulation results showed that the elapsed time for weeding was reduced using the set of the VDP oscillator's parameters adjusted by RCGA.

REFERENCES

[1] E. Papadopoulos and Y. Gonthier, "A Framework for Large-Force Task Planning of Mobile and Redundant Manipulator," *Journal of Robotic Systems*, vol. 16, no. 3, 1999, pp. 151-162.

[2] J. Imamura and K. Kosuge, "Handling of an Object Exceeding Load Capacity of Dual Manipulators Using Virtually Unactuated Joints, " *In Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, 2002, pp. 989-994.

[3] J. Kobayashi, S. Kishida, and F. Ohkawa, "Analysis of Suitable Postures for Robot Manipulator Applying Force using Numerical Optimization Method," *In Proceedings of International IEEE Conference Mechatronics & Robotics*, Part II, 2004, pp. 277-282.

[4] J. Kobayashi and F. Ohkawa, "Efficient Oscillation Method with Nonlinear Oscillator for Large Force Generation of Manipulator Exploiting Oscillatory Motion," *In Proceeding of the 2006 IEEE International Conference on Robotics and Biomimetics*, 2006, pp. 351-356.

[5] J. Kobayashi, "Weeding Manipulator Exploiting Its Oscillatory Motion for Force Generation: Verification of the Effectiveness by Simulations using Open Dynamics Engine," *In Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics*, 2009, pp. 1285-1290.

[6] G. A. Pratt and M. M. Williamson, "Series Elastic Actuators," *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1995, pp. 399-406.

[7] B. Bill and I. R. Harvey, "Programmable Spring: Developing Actuators with Programmable Compliance for autonomous Robots," *Robotics and Autonomous Systems*, vol. 55, no. 9, 2007, pp. 728-734.

[8] D. W. Jordan and P. Smith, *Nonlinear Ordinary Differential Equations: An Introduction to Dynamical Systems (Third Edition)*, Oxford University Press, 1999, pp. 253-257.

[9] P. Veskos and Y. Demiris, "Developmental acquisition of entrainment skills in robot swinging using van der Pol oscillators, " *In Proceedings of the Fifth International Workshop on Epigenetic Robotics*, 2005, pp. 87-93.

[10] H. Kita, I. Ono, and S. Kobayashi, "Theoretical Analysis of the Unimodal Normal Distribution Crossover for Real-coded Genetic Algorithm," *In Proceedings of the 1998 IEEE Conference on Evolutionary Computation*, 1998, pp. 529-534.

[11] S. Tsutsui, M. Yamamura, and T. Higuchi, "Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms," *In Proceedings if the 1999 Genetic and Evolutionary Computation Conference*, 1999, pp. 657-664.

[12] L. J. Eshelman and J. D. Schaffer, "Real-Coded Genetic Algorithms and Interval-Schemata," *Foundations of Genetic Algorithms*, 2, Morgan Kaufman Publishers, 1993, pp. 187-202.

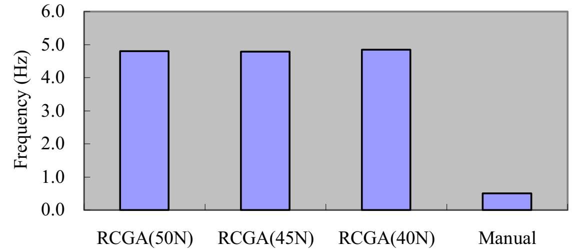

Fig. 11. Adjusted Parameters (Natural Frequency f_v).

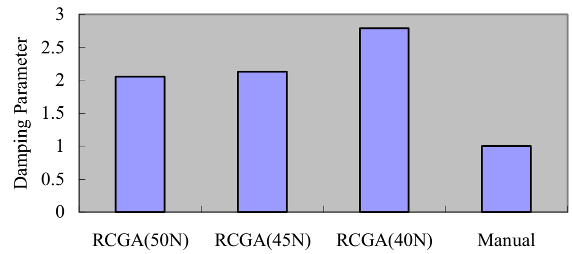

Fig. 12. Adjusted Parameters (Damping Parameter ε).

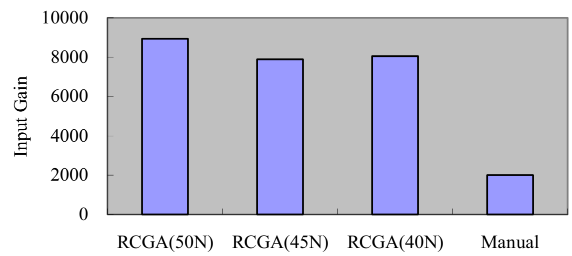

Fig. 13. Adjusted Parameters (Input Gain G_{in}).

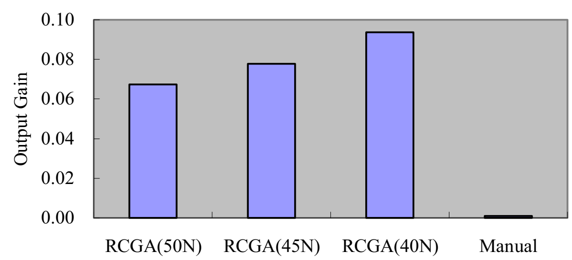

Fig. 14. Adjusted Parameters (Output Gain a_y).

TABLE I
ELAPSED TIMES FOR WEEDING

	f_{max}		
	50 N	45 N	40 N
RCGA(50N)	0.781	X	X
RCGA(45N)	0.786	0.779	X
RCGA(40N)	0.785	0.781	0.778