'94 KACC (1994. 10. 17 ~ 20)

Control of A Cart System Using Genetic Algorithm

Sung-Soo Kim*
*Dept. of Elec. Eng.,Ruhr-Universität. Bochum,BRD

Kwang-Bang Woo†
†Elec. Eng.,Yonsei University

Abstract

So far many researchers have studied to control a cart system with a pole on the top of itself(forwards we call it simply a cart system) which is movable only to the directions to which a cart moves, using neural networks and genetic algorithms. Especially when it was solved by genetic algorithms, it was possible to control a cart system more robustly than ordinary methods using neural networks but it had problems too,i.e., the control time to be achieved was short and the processing time for it was long. However we could control a cart system using standard genetic algorithm loger than ordinary neural network methods (for example error backpropagation) and could see that robust control was possible. Computer simulation was performed through the personal computer and the results showed the possibility of real time control because the cpu time which was occupied by processes was relatively short.

1 Introduction

The history of GA(Genetic Algorithm) which was studied first in 1960s by J.H.Holland and his students, is a little over 30 years old[1][2][3]. It is possible for GA to be applied to the optimization problem, the learning of neural networks, the control of a complicated system and the pattern recognition etc. And its performance is superior to the other methods.

If we regard the mechanism of higher animals among the living things which were created by god as the best one in the world, we can say that a system which can approach to such a mechanism and its processing method, are one of the important objects for us human beings to study about it. From this point of view, GA is thought valuable to be studied from now on.

Among the physical systems which exist in the real world there are control problems of unstable nonlinear system of which we can't estimate the outputs of the next state of a plant and many control theoriests tried to solve them. One of the representative examples of such problems is a cart system. We tried to solve the classic control system using GA longer and more robustly in a shorter time than other methods to be studied till now.

A cart system is shown in Fig.1. It has two subjects. One is to center a cart within a given track, and the other is to balance a pole within a given angle. That is, a cart system can move to the right or left direction within a given track. A pole hinged upon the cart is able to move freely only in a vertical plane which is formed by a pole and the directions to which a cart moves. The control object of a cart system is to give a constant force to the right or left side of the plant every discrete time - it is a bang bang control problem - to keep the pole angle within a specific range and at the same time to keep the cart to the center of a predefined track as long as possible. This system is a dynamic system and has four state variables:

x : position of a cart
θ : angle of a pole
$\dot{x}$: velocity of a cart
$\dot{\theta}$: angular velocity of a cart

So far many people have selected a cart system as a test bed to see the performances of their own GAs in the control field of nonlinear systems[4][5][6][7][8][9][10]. We also adopted this problem as an object to be controled to see the adaptability and the robustness of standard GA to the nonlinear systems in arbitrary control circumstances. Especially we selected this problem as our subject because it is easy to compare our experimental results with so far achieved ones.

The experimental results which were achieved so far are summarized briefly in Table 1 and Table 2. Table 1 shows the results using GAs and Table 2 using other methods.

2 The layout of a cart control system

We constructed a cart control system like Fig.2. The mathematic model of this system is already knwon by R.H.Cannon,Jr[11]. His model is given in eq.(1)-(2) :

$$\ddot{\theta} = \frac{g\sin\theta - \frac{\cos\theta(F + m_p l\dot{\theta}^2 \sin\theta)}{m_p + m_c}}{\frac{4l}{3} - \frac{l m_p \cos^2\theta}{m_p + m_c}} \tag{1}$$

$$\ddot{x} = \frac{F + m_p l(\dot{\theta}^2 \sin\theta - \ddot{\theta}\cos\theta)}{m_p + m_c} \tag{2}$$

And eqs.(3)-(6) are acquired from eqs.(1)-(2) using Euler's approximation equations :

$$\theta(t+1) = \theta(t) + \dot{\theta}(t)T \tag{3}$$
$$\dot{\theta}(t+1) = \dot{\theta}(t) + \ddot{\theta}(t)T \tag{4}$$
$$x(t+1) = x(t) + \dot{x}(t)T \tag{5}$$
$$\dot{x}(t+1) = \dot{x}(t) + \ddot{x}(t)T \tag{6}$$

Our control system consists of a plant, a controller and a genetic processor. The controller consists of 4 converters A,B,C and D which convert the values of state variables from a plant into binary digits; a decoder which decodes binary digits from converts into decimal numbers; a box which outputs binary digits according to the outputs from the decoder; a supervisor. Every discrete time the supervisor checks the outputs of the plant to see if the given conditions are content. And if the conditions are not content, it sends the genetic processor a false signal(we used 0) but if the conditions are content, it sends to the processor a permit signal 1. If the genetic processor receives a failure signal from the supervisor, it sends the plant a reset signal so that the plant restarts its control from the initial conditions. And whenever the genetic processor receives a fail signal, it sends a renewal signal to the box so that the box resets its values with the genes of a new chromosome. Converter A for x outputs 1 bit, converter B for θ 4 bits, converter C for $\dot{x}$ 2 bits and converter D for $\dot{\theta}$ 2 bits.

3 The control of a cart system

For control of a cart system we defined an evaluation function as eq.(7) :

$$\text{fitness} \equiv \sum P \quad (7)$$

where P : permit signal

We defined the summation of the permit signals from the supervisor to the genetic processor as a fitness of each chromosome. Accordingly it is possible to say that the purpose of this paper is to seek a chromosome of which the evaluation function is the greatest one among all the chromosomes.

The controller that we devised, doesn't know about the dynamics of itself and doesn't have any a priori knowledges of the plant control. Only it receives the state variables of the present state as inputs. The function of the plant can be described as eqs. (1)-(2). The state variables from the plant are sent to the 4 converters A,B,C and D, where according to the Fig. 3(in the case of experiment 2 Fig. 4) they are converted into 9 bits $b_8b_7b_6b_5b_4b_3b_2b_1b_0$ and they are sent to the decoder. The decoder decodes them into decimal digits and sends it to the box. The box contains always the 512 binary values of a chromosome of a population. Whenever the box receives a decimal number as input, it sends a binary value(but actually it is transformed into decimal values by the genetic processor) to the plant. If the read value from the box is 0, it outputs 10[N](in the case of experiment 2, it outputs 6[N]) and if the read one is 1, it outputs -10[N](in the case of the experiment 2, it outputs - 6[N]) to the plant. However if the plant doesn't receive the reset signal, this procedure continues during the given generation or until it reaches to the objective value. If all of the chromosomes in a population don't search the optimal value or receive failure signals, then the genetic processor creates a new population through crossover and mutation procedure and a new generation begins. This procedure continues for the given generations or until the optimal value is sought. And whenever the box receives a renewal signal from the genetic processor, it removes the stored values and stores the genes of a new chromosome in itself.

3.1 The experiment 1 and its results

We selected the initial values of state variables in controling a cart system using genetic algorithm :

x : ±2.0 [m]
θ : ±1.0 [deg]
$\dot{x}$: 0.0 [m/sec]
$\dot{\theta}$: 0.0 [deg/sec]

And the control parameters used in the experiment are as follows :

discrete time interval T : 0.02[sec]
force F : 10[N]
mass of a cart m_c : 1 [kg]
mass of a pole m_p : 0.1 [kg]
length of a pole (2l) : 1 [m]
gravitational acceleration g : 9.8 [m/ sec^2]
length of a track : 4.8 [m]

We adopted 3 kinds of initial populations (forwards we call them simply populations) for the experiment and 10 kinds of initial values of state variables. Using the same population under the same initial conditions we made 10 runs with 10 different seeds. We simulated the experiment using a personal computer IBM 486 DX compatible adopting UNIX as os. The simulation results are shown in Table 3. Fig.5(a) - (c) show the survive time in this experiment. Fig.5 (a) shows the results using initial population 1, Fig.5 (b) shows the results using initial population 2 and Fig.5 (c) shows the results using initial population 3.

From the above experiment it was revealed that in the case using initial population 1 the result are best and in the case using initial population 2 worst. It is because the initial values affect the results of our system. Especially the initial values of angle affected the results more than other ones. If the local optimum is met too early, it took a long time to find a global optimum. Because the chromosome which have a local optimum, yields more children in the next generation. A method to avoid this phenomenon is to restrict the numbers of children of the chromosome which has a local optimum.

We found that the seed affects the results greatly. The reason is that we restricted the generation number to 100. In the paper we report that we could control at most 2 hours but actually we could control very long time in many cases.

The total experimental results are shown in Table 3. Averagely we could control a cart system for 1 hour 13 minutes 15.23 seconds using 1 hour 6 minutes 50.65 seconds of cpu time(in 68 generations). These results are superior to the other results of Table 1 and Table 2,i.e., the used cpu time is shorter and the controlled time is longer compared to the previous ones.

3.2 The experiment 2 and its results

We selected the initial values of state variables in controling a cart system using genetic algorithm :

x : ±1.0 [m]
θ : ±1.0 [deg]
$\dot{x}$: 0.0[m/sec]
$\dot{\theta}$: 0.0[deg/sec]

And the control parameters used in the experiment are as follows :

discrete time interval T : 0.05[sec]
force F : 6[N]
mass of a cart m_c : 0.8 [kg]
mass of a pole m_p : 0.1 [kg]
length of a pole (2l) : 0.4 [m]
gravitational acceleration g : 9.8 [m/ sec^2]
length of a track : 6.0 [m]

We adopted a kind of population for the experiment and 5 kinds of initial values of state variables. Using the same population under the same initial conditions we made 5 runs with 5 different seeds. In this experiment the objective value is 3 hours and we have not restricted the generation numbers,i.e., the program was not stopped until we get the objective value. The simulation results are shown in Table 4. Fig.6 shows the cpu time used in the experiment.

According to the experimental results we could control a cart system for 3 hours of simulation time using 1 hour 35 minutes 20.60 seconds(in 202 generations) of cpu time. The results of the experiment shows that our system can be controlled more robustly under the harsher conditions compared to the other ones. And we observed that if we adopt the harsher initial conditions, it is possible to control the system for 3 hours. But in the case it requires more cpu time.

4 Conclusion and discussion

We could control a cart system using standard genetic algorithm in a relatively short time for a given time successfully, which was regarded as one of the difficult nonlinear control systems. It says that the complicated nonlinear systems are able to be controlled successfully using genetic algorithm simply. However there are also some subjects to be solved. One is to decide the appropriate values of crossover probability and mutation probability. Another is to modify the genetic algorithm according to the specific system circumstances. That means that we must adopt the evolution programming method ,i.e.,we must change the representation of each chromosome instead of changing the problem.

If we divide the search space into more regions, probably we can acquire the better results but because of the longer processing time of computer it must be restricted. Therefore according to the control conditions appropriate division of search space must be considered carefully. And for the real time control it is recommended to adopt the so far achieved various genetic techniques for example varying of mutation probability[12].

References

1. J.D.Bagley, The Behavior Of Adaptive Systems Which Employ Genetic And Correlation Algorithms, doctoral thesis, Dept. of Computer and Communication Sciences, University of Michigan, 1967.

2. D.R.Frantz, Non-Linearities In Genetic Adaptive Search, doctoral thesis, Dept. of Computer and Communication Sciences, University of Michigan, 1972.

3. J.H.Holland, Adaptation in Natural and Artificial Systems : An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, University of Michigan, 1975.

4. M.O.Odetayo,D.R.McGregor, " Genetic Algorithm For Inducing Control Rules For A Dynamic System",Proc. 3rd International Conf. on Genetic Algorithms, Morgan Kaufmann,pp.177-182,1989.

5. D.Whitley,S.Dominic,R.Das," Genetic Reinforcement Learning with Multilayer Neural Networks", Proc. 4th International Conf. on Genetic Algorithms, Morgan Kaufmann,pp.562-569,1991

6. D.Michie,R.A.Chambers," BOXES:AN EXPERIMENT IN ADAPTIVE CONTROL", Machine Intelligence 2, E.Dale and D.Michie, Eds.Edinburgh : Oliver and Boyd, pp.137-152,1968.

7. A.G.Barto,R.S.Sutton,C.W.Anderson," Neuronlike adaptive elements that can solve difficult learning control problems", IEEE Transactions on Systems, Man, and Cybernetics SMC-13,pp.834-846,1983.

8. M.E.Connell,P.E.Utgoff," Learning to Control a Dynamic Physical System", Proceedings of the 6th National Conference on Artificial Intelligence, Los Altos, CA,Morgan Kaufmann,pp.456-460,1987.

9. C.W.Anderson," Learning to Control an Inverted Pendulum Using Neural Networks",IEEE Control Systems Magazine, vol.9, no.3, pp.31-37, 11989.

10. E.Grant,B.Zhang," A Neural-Net Approach to Supervised Learning of Pole Balancing", Proceedings IEEE International Symposium on Intelligent Control 1989,pp.123-129,1989.

11. R.H.Cannon,Jr.,Dynamics of Physical Systems, McGraw-Hill Book Company, pp.703 – 705, 1967.

12. S.S.Kim,K.B.Woo," A Study On Improvement of Performance in Genetic Algorithm", submitted to the KIEE, January 1994.

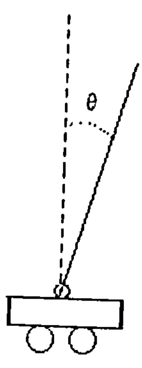

Fig. 1 A cart system

A :	$-2.4 \le x < 0.0$	-> $b_0 = 0$
	$0.0 \le x \le 2.4$	-> $b_0 = 1$
B :	$-0.25 > \dot{x}$	-> $b_1 = 0, b_2 = 0$
	$-0.25 \le \dot{x} < 0.0$	-> $b_1 = 1, b_2 = 0$
	$0.0 \le \dot{x} \le 0.25$	-> $b_1 = 0, b_2 = 1$
	$0.25 < \dot{x}$	-> $b_1 = 1, b_2 = 1$
C :	$-12.0 \le \theta < -10.0$	-> $b_3 = 0, b_4 = 0, b_5 = 0, b_6 = 0$
	$-10.0 \le \theta < -5.0$	-> $b_3 = 1, b_4 = 0, b_5 = 0, b_6 = 0$
	$-5.0 \le \theta < -2.5$	-> $b_3 = 0, b_4 = 1, b_5 = 0, b_6 = 0$
	$-2.5 \le \theta < -1.5$	-> $b_3 = 1, b_4 = 1, b_5 = 0, b_6 = 0$
	$-1.5 \le \theta < -1.0$	-> $b_3 = 0, b_4 = 0, b_5 = 1, b_6 = 0$
	$-1.0 \le \theta < -0.5$	-> $b_3 = 1, b_4 = 0, b_5 = 1, b_6 = 0$
	$-0.5 \le \theta < -0.25$	-> $b_3 = 0, b_4 = 1, b_5 = 1, b_6 = 0$
	$-0.25 \le \theta < 0.0$	-> $b_3 = 1, b_4 = 1, b_5 = 1, b_6 = 0$
	$0.0 \le \theta < 0.25$	-> $b_3 = 0, b_4 = 0, b_5 = 0, b_6 = 1$
	$0.25 \le \theta < 0.5$	-> $b_3 = 1, b_4 = 0, b_5 = 0, b_6 = 1$
	$0.5 \le \theta < 1.0$	-> $b_3 = 0, b_4 = 1, b_5 = 0, b_6 = 1$
	$1.0 \le \theta < 1.5$	-> $b_3 = 1, b_4 = 1, b_5 = 0, b_6 = 1$
	$1.5 \le \theta < 2.5$	-> $b_3 = 0, b_4 = 0, b_5 = 1, b_6 = 1$
	$2.5 \le \theta < 5.0$	-> $b_3 = 1, b_4 = 0, b_5 = 1, b_6 = 1$
	$5.0 \le \theta < 10.0$	-> $b_3 = 0, b_4 = 1, b_5 = 1, b_6 = 1$
	$10.0 \le \theta \le 12.0$	-> $b_3 = 1, b_4 = 1, b_5 = 1, b_6 = 1$
D :	$-50.0 \le \dot{\theta} < -5.0$	-> $b_7 = 0, b_8 = 0$
	$-5.0 \le \dot{\theta} < 0.0$	-> $b_7 = 1, b_8 = 0$
	$0.0 \le \dot{\theta} < 5.0$	-> $b_7 = 0, b_8 = 1$
	$5.0 \le \dot{\theta} \le 50.0$	-> $b_7 = 1, b_8 = 1$

Fig. 3 Internal functions of converters used in experiment 1

A :	$-3.0 \le x < 0.0$	-> $b_0 = 0$
	$0.0 \le x \le 3.0$	-> $b_0 = 1$
B :	$-0.25 > \dot{x}$	-> $b_1 = 0, b_2 = 0$
	$-0.25 \le \dot{x} < 0.0$	-> $b_1 = 1, b_2 = 0$
	$0.0 \le \dot{x} \le 0.25$	-> $b_1 = 0, b_2 = 1$
	$0.25 < \dot{x}$	-> $b_1 = 1, b_2 = 1$
C :	$-24.0 \le \theta < -21.0$	-> $b_3 = 0, b_4 = 0, b_5 = 0, b_6 = 0$
	$-21.0 \le \theta < -18.0$	> $b_3 = 1, b_4 = 0, b_5 = 0, b_6 = 0$
	$-18.0 \le \theta < -15.0$	> $b_3 = 0, b_4 = 1, b_5 = 0, b_6 = 0$
	$-15.0 \le \theta < -12.0$	> $b_3 = 1, b_4 = 1, b_5 = 0, b_6 = 0$
	$-12.0 \le \theta < -9.0$	> $b_3 = 0, b_4 = 0, b_5 = 1, b_6 = 0$
	$-9.0 \le \theta < -6.0$	-> $b_3 = 1, b_4 = 0, b_5 = 1, b_6 = 0$
	$-6.0 \le \theta < -3.0$	-> $b_3 = 0, b_4 = 1, b_5 = 1, b_6 = 0$
	$-3.0 \le \theta < 0.0$	-> $b_3 = 1, b_4 = 1, b_5 = 1, b_6 = 0$
	$0.0 \le \theta < 3.0$	-> $b_3 = 0, b_4 = 0, b_5 = 0, b_6 = 1$
	$3.0 \le \theta < 6.0$	-> $b_3 = 1, b_4 = 0, b_5 = 0, b_6 = 1$
	$6.0 \le \theta < 9.0$	-> $b_3 = 0, b_4 = 1, b_5 = 0, b_6 = 1$
	$9.0 \le \theta < 12.0$	-> $b_3 = 1, b_4 = 1, b_5 = 0, b_6 = 1$
	$12.0 \le \theta < 15.0$	-> $b_3 = 0, b_4 = 0, b_5 = 1, b_6 = 1$
	$15.0 \le \theta < 18.0$	-> $b_3 = 1, b_4 = 0, b_5 = 1, b_6 = 1$
	$18.0 \le \theta < 21.0$	-> $b_3 = 0, b_4 = 1, b_5 = 1, b_6 = 1$
	$21.0 \le \theta \le 24.0$	-> $b_3 = 1, b_4 = 1, b_5 = 1, b_6 = 1$
D :	$-50.0 \le \dot{\theta} < -5.0$	-> $b_7 = 0, b_8 = 0$
	$-5.0 \le \dot{\theta} < 0.0$	-> $b_7 = 1, b_8 = 0$
	$0.0 \le \dot{\theta} < 5.0$	-> $b_7 = 0, b_8 = 1$
	$5.0 \le \dot{\theta} \le 50.0$	-> $b_7 = 1, b_8 = 1$

Fig. 4 Internal functions of converters used in experiment 2

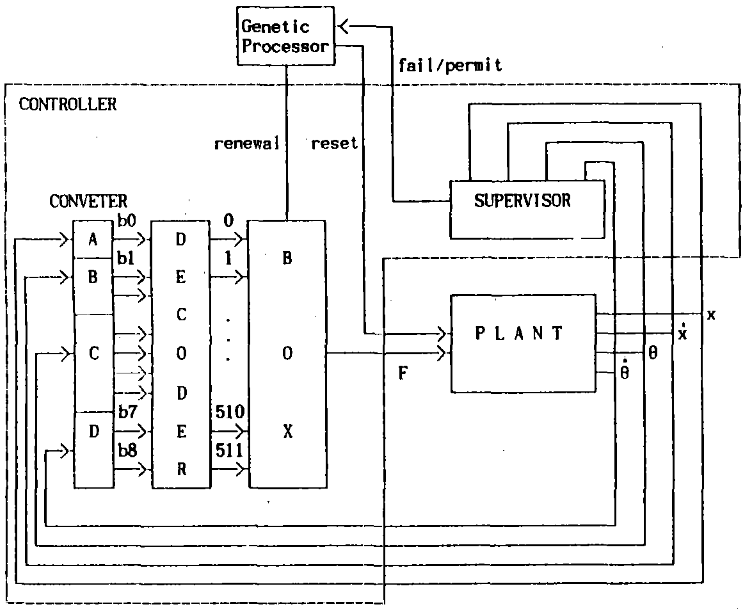

Fig. 2 The configuration of a cart control system

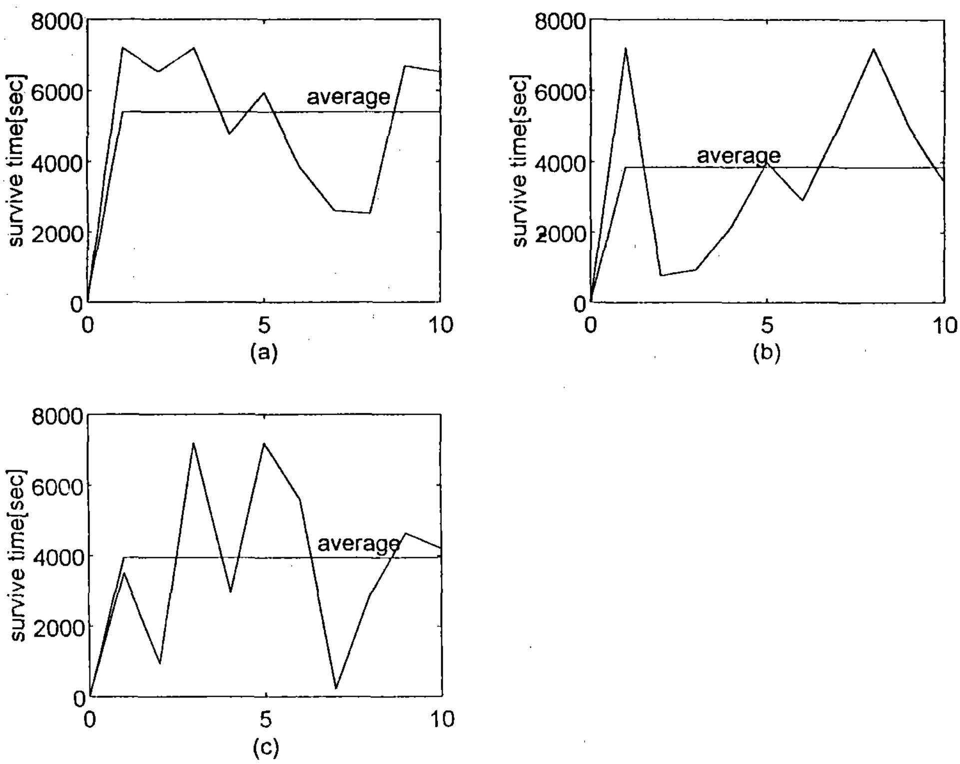

Fig. 5 The survive time of experiment 1

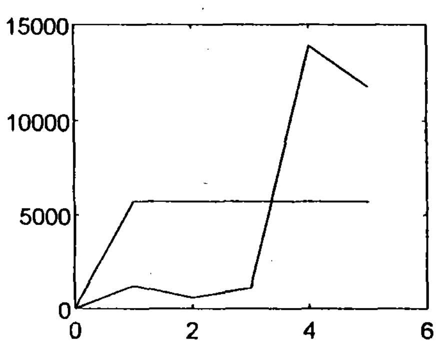

Fig. 6 The cpu time of experiment 2

Table 3 Total results of experiment 1

success/ run	minimal survive time [sec] hh:mm:ss generation	maximal survive time [sec] hh:mm:ss generation	average survive time [sec] hh:mm:ss generation
166 / 300	19.50 00:02:16.60 100	* 7200.00 09:24:34.00 48 $00:00:55.33	4395.23 01:06:50.65 68

$: survive time which was searched most rapidly
* : survive time which was searched most slowly

Table 4 Total results of experiment 2

success/ run	minimal survive time [sec] hh:mm:ss generation	maximal survive time [sec] hh:mm:ss generation	average survive time [sec] hh:mm:ss generation
25 / 25	$10800.00 00:04:59.70 92	*10800.00 10:51:13.00 908	10800.00 01:35:20.06 202

$: survive time which was searched most rapidly
* : survive time which was searched most slowly

Researchers	average control time	learning method	cpu time hh:mm:ss	used computer	chromosome size	population size	probability of mutation	probability of crossover	remarks
D.Whitley et al.	40 minutes	GENITOR variant	3148 trials	not known	35	50	large	not known	real number gene was used
M.O.Odetayo et al.	200 second	GA	00:17:04 (9 generations)	not known	54	300	0.01	not known	integer gene was used

Table 1 So far achieved results using GA for a cart control system

Table 2 So far achieced results using other methods for a cart control system

Researchers	average control time	learning method	cpu time hh:mm:ss	used computer	remarks
D.Michie et al	about 1 hour	BOX	600 trials	not known	225 boxes were used. T = 0.05[sec] result for test run D
A.G.Barto et al	about 27 minutes	AHC	75 trials	not known	ASE/ACE is used
E.Grant et al	200 second	error backpropagation	about 10 minutes	SUN 3	T = 0.01[sec] off-line processing
M.E.Connell et al	214 second	CART	not known	not known	
C.W.Anderson	24 minutes	AHC	10000 trials	not known	2 layer network was used