Hardware accelerator for evolutionary robotics

MASAYA YOSHIKAWA[(1)], HIDEKAZU TERAI[(2)]

[(1)] Department of Information Engineering,
Meijo University
1-501, Shiogamaguchi, Tenpaku, Nagoya, 468-8502, JAPAN

[(2)]Department of VLSI System Design,
Ritsumeikan University
1-1-1, Nojihigashi, Kusatsu, Shiga, 525-8577, JAPAN

Abstract: - Evolution robotics is one of the autonomous learning and its approaches have been applied to various fields. In this paper, we discuss new evolutionary robotics technique. The proposed evolutionary robotics acquires their behavior using genetic-based machine learning. It adopts new if-then rules for learning algorithm. Moreover, it introduces novel hardware accelerator in order to reduce simulation time for calculation, since the genetics-based machine learning requires very long computational time. Experiments using quasi-ecosystem demonstrate not only the effectiveness of the proposed learning algorithm, but also that of the proposed hardware accelerator.

Key-Words: - Evolutionary robotics, Hardware accelerator, If-then rules, Genetic-based machine learning, Quasi-ecosystem

1 Introduction

Recently, intelligent robots have been applied to various fields[1]. An intelligent robot can generally recognize its environment, make decisions, take actions. That is, an intellectual robot's action is learned autonomously. However, it is impossible to design a robot's controller in consideration of all situations in advance. Therefore, there are some researches which make a robot learn action using neural network and fuzzy system. Evolution robotics is one of the autonomous learning and its approaches are based on Genetic Algorithm (GA)[2],[3]. The GA was proposed by Holland as an algorithm for probabilistic search, learning, and optimization. It is based in part on the mechanism of biological evolution and applied to various fields[4]-[6]. The evolutionary robotics in quasi-ecosystems[7]-[9] have been proposed and have achieved the measure success. However, GA has the inherent problem of requiring substantial processing time, because they are multi-point search algorithms. Therefore, the computer simulation of evolutionary robotics in the quasi-ecosystem requires a very long computational time.

In this paper, we propose a novel approach of evolutionary robotics in a quasi-ecosystem using a hardware accelerator in order to realize high-speed calculation. We deal with multiple robots interacting with a quasi-ecosystem model composed of two species of fishes and plankton. The fish has energies and consume them with progress of time. And then, the energies are supplemented by eating plankton. The fish will die if their energies are lost. However, if the number of planktons exceeds a threshold, the inside of water will be short of oxygen and all fish will become extinct. On the other hand, the plankton increases according to regulation. Basically, the aim of the proposed evolutionary robotics is to maintain the numerical balance of fishes and plankton using multiple robots. A relationship among the fishes, plankton and robots is shown in Fig.1.

Experiments using quasi-ecosystem demonstrate not only the effectiveness of the proposed learning algorithm, but also that of the proposed hardware accelerator.

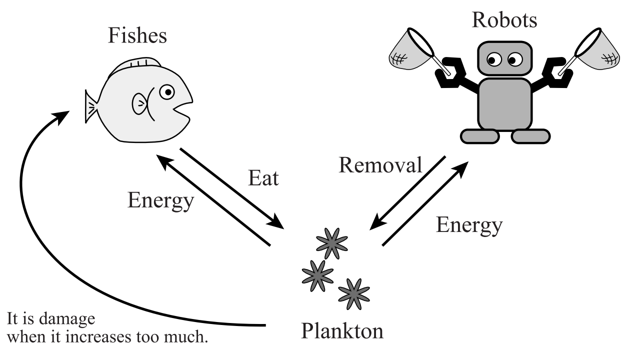


Fig.1 Relationship of the quasi-ecosystem

2 Preliminaries

2.1 Related work

Examples of GA hardware have been proposed by Scott *et al.* [10], Graham *et al.* [11], and Imai *et al.* [12]. Scott *et al.* developed a hardware-based GA and demonstrated its superiority to software in speed and solution quality. And then, Imai *et al.* developed a processor element of GA for parallel processing and proved the effectiveness of the parallel technique. However, most of these previous works deal with small-scaled problems. Moreover, no studies have ever seen the hardware accelerator for evolutionary robotics. The proposed hardware accelerator, implemented on the field-programmable gate array (FPGA), achieves high-speed the genetics-based machine learning.

2.2 Model of quasi-ecosystem

We used a modified quasi-ecosystem [7] in order to correspond to the proposed learning model. The quasi-ecosystem consists of grids of two dimensions, and the fishes and plankton exist on these grids as shown in Fig.2.

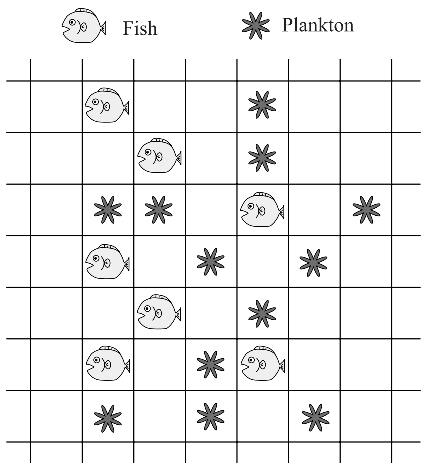


Fig.2 Example of quasi-ecosystem

3 Evolutionary robotics

The proposed evolutionary robotics acquires their behavior using genetic-based machine learning. Moreover, simulation mechanism of each behavior achieves the behavior acquisition which considers the situation in the future.

Fig.3 shows relationship between the behavior acquisition and the environment.

The measurement range of a sensor of the robot is 3x3 centering on itself to recognize their surroundings. And then, moving ranges are adjoining four directions as shown in Fig.4. On the other hand, all fishes can observe surroundings (5x5) as shown in Fig5 and decide their actions based on their surroundings.

An action of a fish has six kinds which are movement of four directions, eating, and stillness. The movement consumes fish's energy and the energy is supplemented by eating plankton.

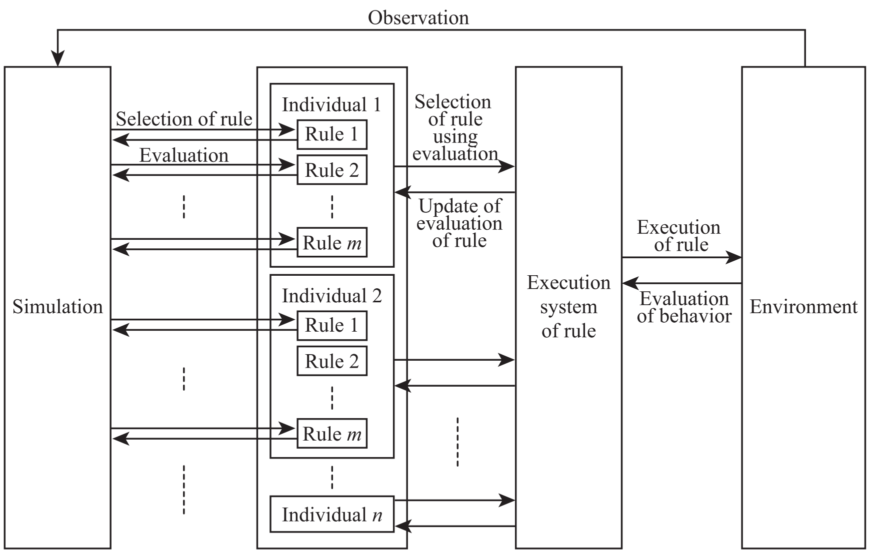


Fig.3 Relationship between the behavior acquisition and the environment

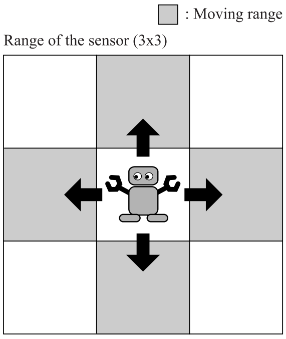


Fig.4 Range of sensor and moving direction of robot

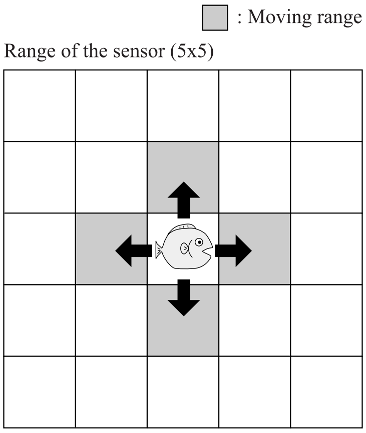


Fig.5 Range of sensor and moving direction of fish

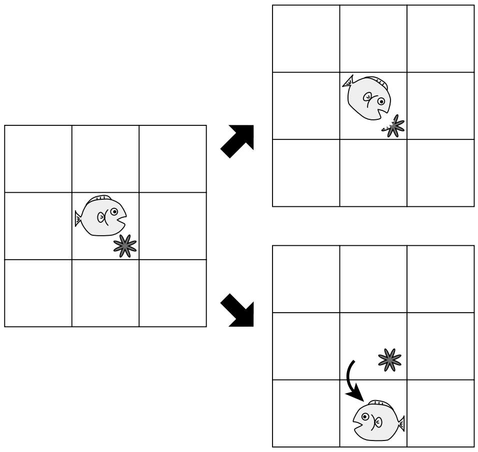
Regarding decision of an action in each fish, first, all fishes observe surroundings. The surroundings have two cases as follows.

Case1: A fish and plankton exist on the same grid. If fish's energy is less than threshold, the fish eats the plankton. In contrast, if it's energy exceeds threshold, it decides whether to eat or not to eat at random.

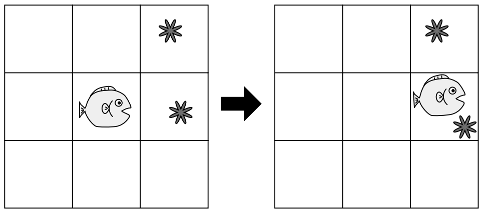
Case2: A fish and plankton does not exist on the same grid. The fish moves to the direction of which more plankton exists.

Fig.6 shows examples of the behavior rule of the fish.

Moreover, the condition of which the fish dies is decided by the two conditions; (1) the total number of plankton and (2) the energy amount of fish. Concerning (1), the condition is decided by the total number of plankton in the whole grid. That is, the number of plankton is counted in each step, and if it exceeds the threshold, all fish die out. Concerning (2), a fish dies out when energy of the fish decreases more than the threshold.



(1) Case1: A fish and plankton exist on the same grid



(2)Case2: A fish and plankton exist on the same grid

Fig.6 Examples of the behavior rule of the fish

The energy of a fish is calculated as follows.

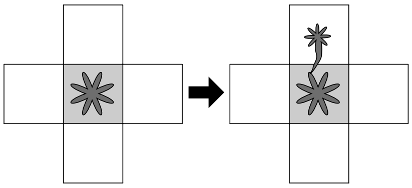
$$E(t+1) = E(t) + a \tag{1}$$

$$E(t+1) = E(t) - b \tag{2}$$

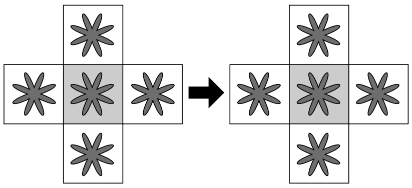
Here, equation (1) represents a case of which a fish eats plankton, and equation (2) represents a case of movement with no eating. In these equation, *E(t)* represents the energy on step *t*, and "*a*" and "*b*" represent parameters.

On the other hand, the plankton proliferates to either in the adjacent grid in each *n* step. However, it doesn't proliferate when the plankton already exists in all adjacent grids. Fig.7 shows examples of the behavior rule of the plankton.

It needs to model for adopting GA at optimization problem. Regarding coding, a set of decision rules of each robot is regarded as an individual in GA. That is, each robot has a candidate rule set. Therefore, a chromosome for rule set is represented as shown in Fig.8. Each rule set includes if-then rules. In this way, the rule set is evolved in the robot itself.



(1) Example of proliferation of plankton



(2) Example of existence in all adjacent grids

Fig.7 Examples of the behavior rule of the plankton

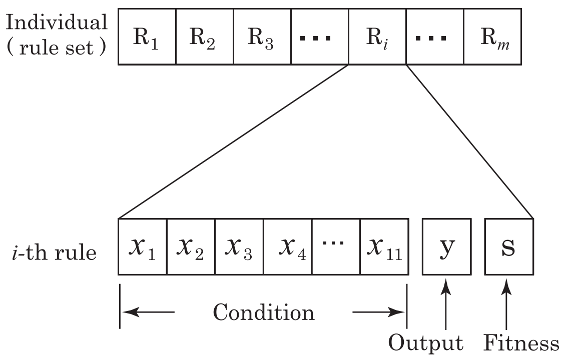


Fig.8 Example of chromosome for the rule set

The robots also have energies and consume them with moving. The energies are supplemented by eating plankton as well as the fish.

The input to the robot has eleven kinds from x_1 to x_{11}. These inputs of 11 pieces are expressed respectively by one bit of 0 or 1. The inputs from x_1 to x_9 indicate surroundings of which the robot is centered as shown in Fig.9.

Here, if the plankton exists, the corresponding bit is assigned to 1. Conversely, if the plankton does not exist, that is assigned to 0. And then, x_{10} represents the amount of energy of a present robot. Concretely, if an amount of the remainder of energy exceeds the threshold, the corresponding bit is assigned to 1. Conversely, if an amount of the remainder of energy is less than the threshold, it is assigned to 0.

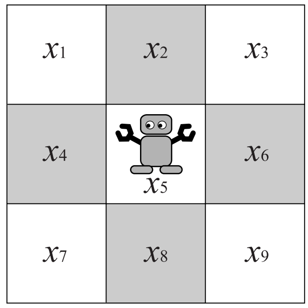


Fig.9 Examples of inputs from x_1 to x_9

x_{11} represents plankton's existence history. Concretely, if the total of surrounding plankton at step *t* is more than the threshold, the corresponding bit is assigned to 1. In contrast, if the total of surrounding plankton at step *t* is less than the threshold, it is assigned to 0.

A basic procedure of the behavior acquisition is as follows.

Step1. Each robot observes surrounding plankton

Step2. The input is decided by the surroundings, energy, and the plankton's existence history.

Step3. The rule is compared with the input of *Step2*.

Step4. When they are the same, the output is the corresponding rule. Conversely, when they are not corresponding, the output is special remark.

The proposed evolutionary robotics assigns several mask data to the input of *Step 2*. The corresponding bits to the mask data are ignored. These mask data realizes the robustness of learning. The rule which has the highest evaluation is selected when there are two or more rules corresponding by *Step4*. The calculation of evaluation value is as follows.

$$F(t+1) = F(t) + a \times P(t) + b \times (\rho(t+1) - \rho(\mathrm{t})) + c \times (E(t+1) - E(\mathrm{t})) \quad (3)$$

Here, *F(t)* represents evaluation value at step *t*, *P(t)* represents the total of surrounding plankton at step *t*, *E(t)* represents the robot's energy at step *t*. And then, *a*, *b*, and *c* represent parameters of positive integer. Equation (3) indicates that the robot obtain more high evaluation when it has little energy. Furthermore, moving to where there is a lot of plankton gives the high evaluation, because the transition of the number of plankton is applied as reward.

Regarding genetic operation, we adopt several operations. Roulette wheel selection, tournament selection, and elitism are adopted as selection operator. Elitism is applied at all generations, and either roulette wheel selection or tournament selection is applied at random. One-point crossover, two points crossover, and uniform crossover are adopted as crossover operator. These crossover operators are applied at random. One-point mutation is applied as mutation operator. It replaces the gene with another gene at random. Fig.10 and Fig.11 show the flow of the behavior acquisition and the whole flow of the proposed evolutionary robotics respectively.

4 Hardware accelerator

In order to reduce the calculation time, we developed a hardware accelerator. The block diagram of the proposed hardware accelerator is shown in Fig.12.

It consists of interface circuit, two controllers for input/output, and holding of output data circuit and robot circuits. In the hardware accelerator, these 10 robot circuits are implemented and parallel processing is performed. The processing procedure of the proposed hardware accelerator is as follows (see Fig.12).

Step1: The data from the software side transfers to the interface circuit and the controller-1 (for input data). The interface circuit checks the error of data.

Step2: The controller for input data sends the individual data and environmental data to each robot circuit.

Step3: Each robot circuit is simulated.

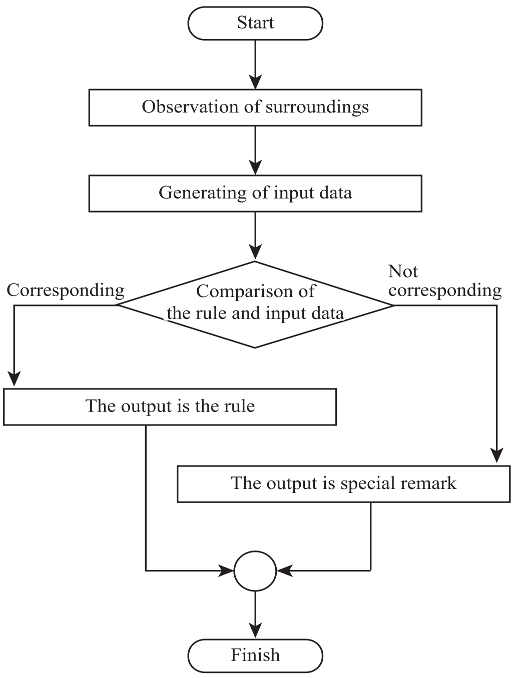


Fig.10 Flow of the behavior acquisition in each robot

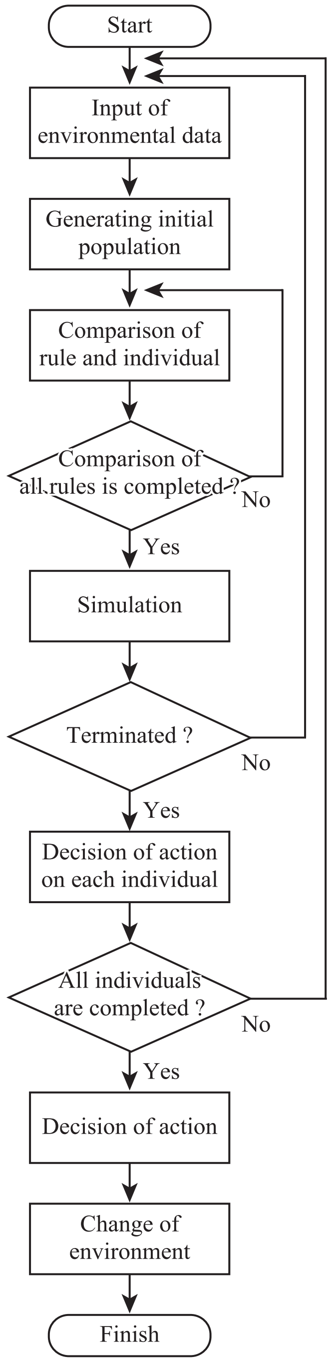


Fig.11 The whole flow of the proposed evolutionary robotics

Step4: Each robot circuit sends simulation results to the holding circuit.

Step5: The simulation results are sent to the controller-2(for output), after all robot circuit have completed simulations.

Step6: The controller-2 (for output) sends simulation results to the interface circuit.

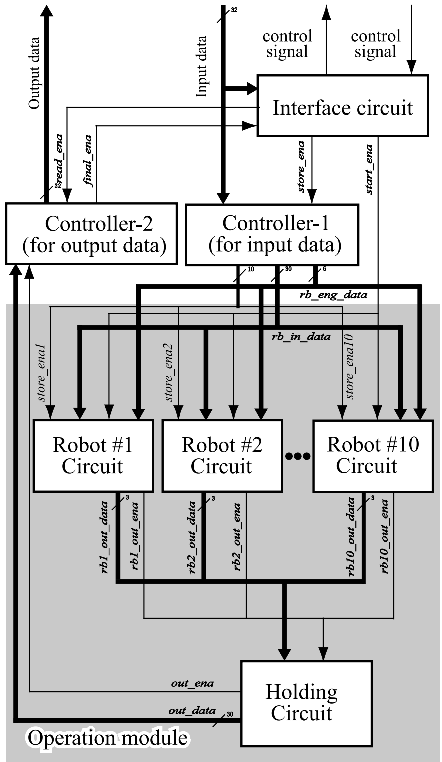


Fig.12 Block diagram of hardware accelerator

The robot circuit consists of memory controller, robot state circuit, comparison of action circuit, change of robot state circuit, calculation of evaluation circuit, change of environment circuit and controller circuit as shown in Fig.13. Moreover, the evolutionary processing is conducted in the change of robot state circuit of the robot circuit.

The processing procedure of the hardware accelerator is as follows (see Fig.13).

Step1: The robot circuit receives the individual's data and the environmental data, and stores these data in each RAM (individual RAM and environmental RAM).

Step2: The data loaded from the environmental RAM is compared with rule of individual's data in the comparison action circuit.

Step3: The data, which is corresponding to the rule, is evaluated in the calculation of evaluation circuit, and it updates the evaluation value with the rule.

Step4: The rule is simulated and evaluated. A regulated frequency repeats the simulation from *Step2* to *Step3*.

Step5: The rule of the individual with the highest evaluation value is sent to the control circuit as an output.

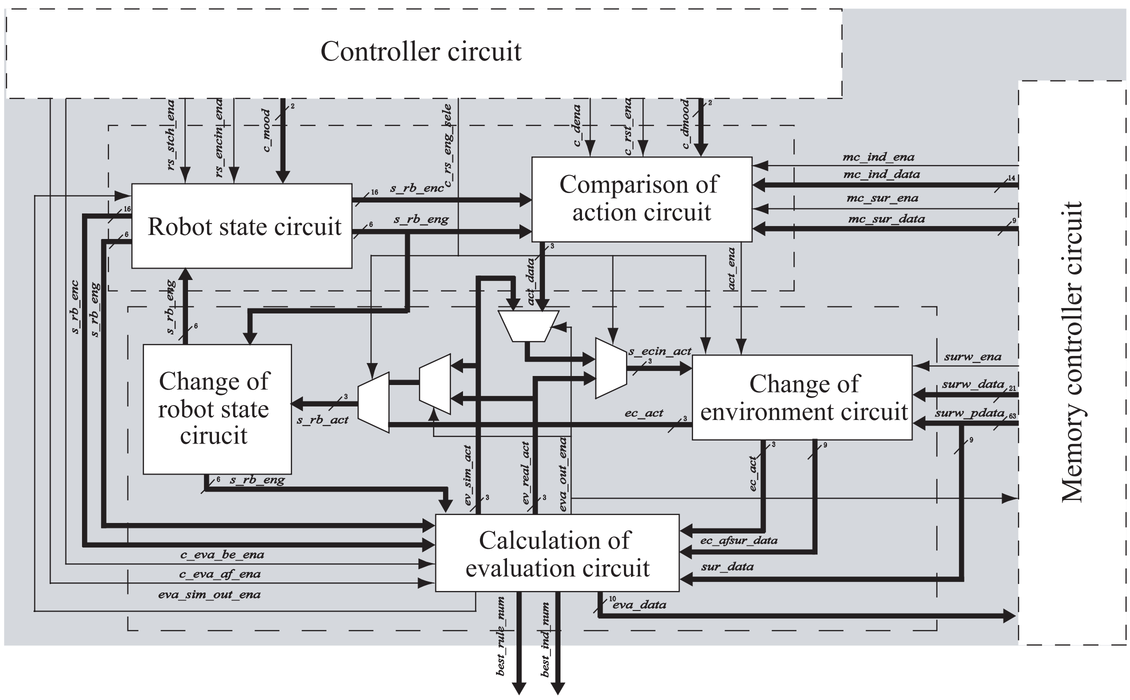


Fig.13 Block diagram of robot circuit

Fig.14 shows the change of robot state circuit. It consists of three sub-circuits (selection, crossover, and mutation circuit). Fig.15 and Fig.16 show selection circuit, and crossover and mutation circuit, respectively.

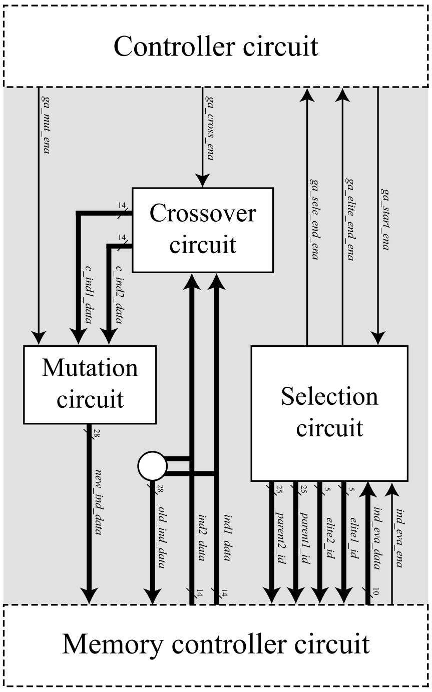


Fig.14 Block diagram of the change of robot state circuit

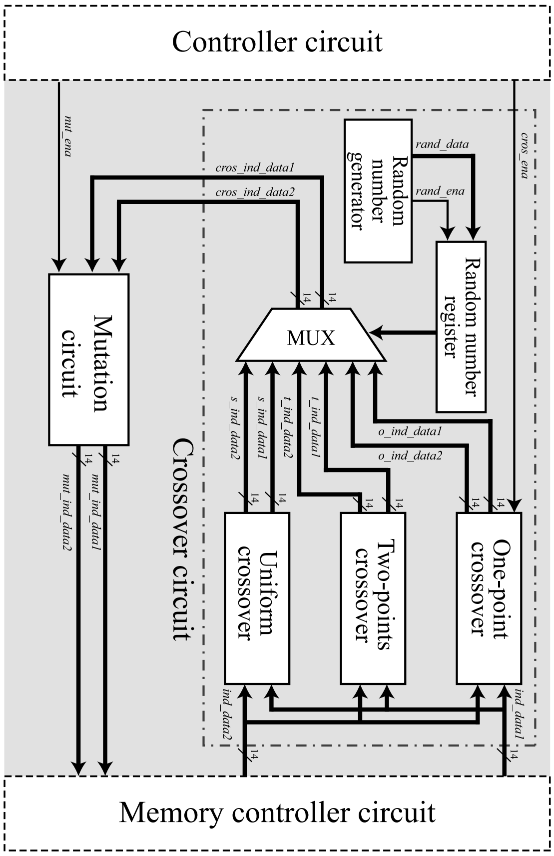


Fig.16 Block diagram of the crossover and mutation circuit

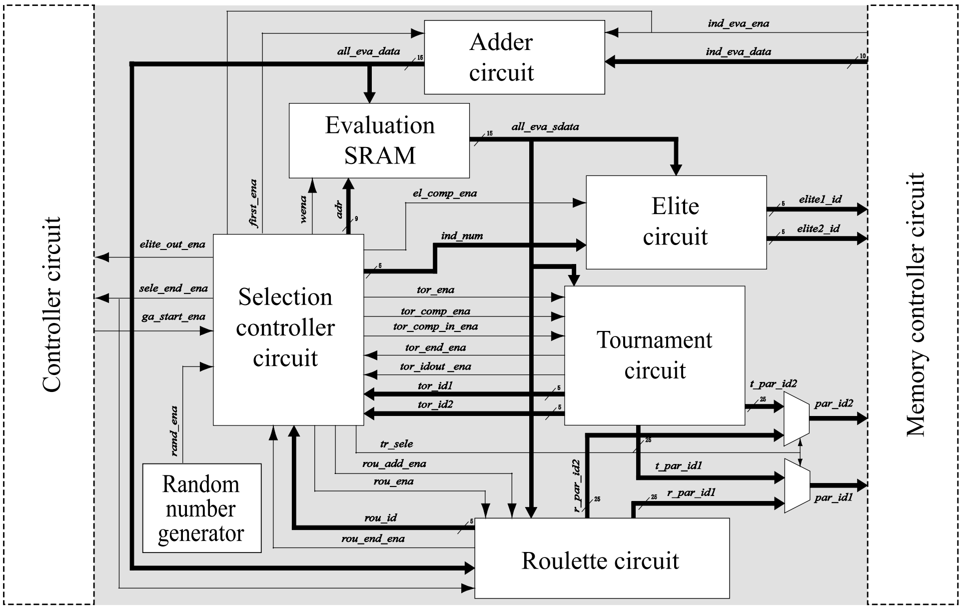


Fig.15 Block diagram of the selection circuit

5 Experimental results and discussion

The proposed hardware accelerator has been designed by Verilog-HDL and synthesized by the Synplicity Synplify. The frequency of hardware accelerator is set up with 33 MHz.

We implemented the hardware accelerator on an single board, which consists of two FPGAs (X2CV3000 and X2CV6000) as shown in Fig.17. Table.1 shows the gate size. In Table.1, LUTs represent combinational logics, and block rams represents memory blocks. We conduct preliminary experiments in order to decide the initial state of plankton.

Fig.18 shows the results of preliminary experiments. The horizontal axis of Fig.18 shows the existence ratio of plankton in the quasi-ecosystem. The initial state is set up with 35% form Fig.18.

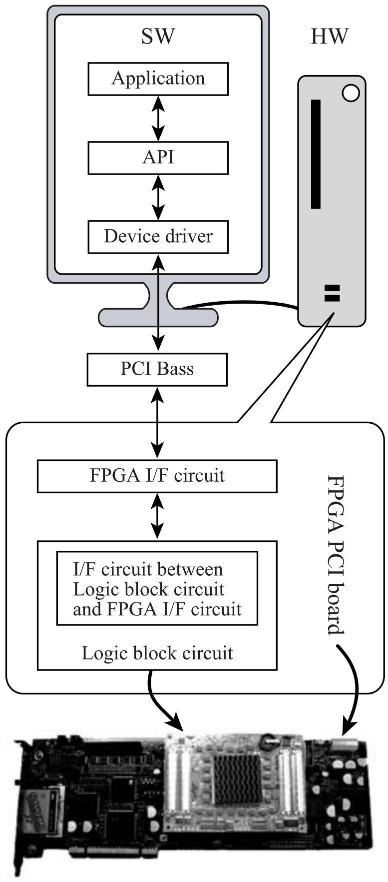


Fig.17 Hardware accelerator on an shingle board

Table.1 Gate size

Resource	Used	Utilization
LUTs	35467	52%
Block Rams	108	75%
Slice	30368	89%

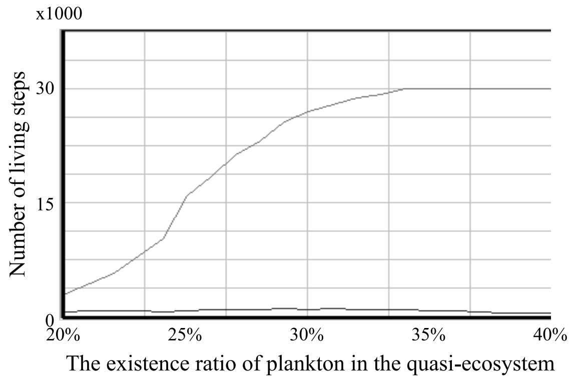


Fig.18 Results of preliminary experiments

Experimental result comparison with software processing is shown in Table.2. In this experiment, ten robots are implemented.

The hardware accelerator achieves 50 times the speed on average compared with software processing.Fig.19 shows the state in the quasi-ecosystem. The state after 1000th steps is the same as the initial state as shown in Fig.19. Lastly, Fig.14 shows a relationship the number of plankton and fish, and generations. The proposed hardware accelerator realized acquisition of the behavior which maintains balance as shown in Fig.20.

6 Conclusion

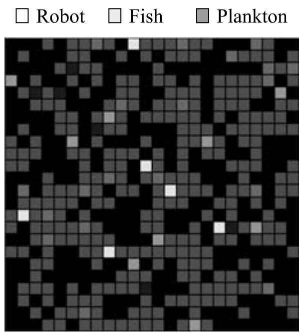
In this paper, we proposed the hardware accelerator for simulation of evolutionary robotics in the quasi-ecosystem. The quasi-ecosystem model composed of fish, plankton and robots, which were in a relationship of parasitism, was simulated on the discrete cell space. The if-then rules were introduced and applied genetics-based machine learning for acquiring a strategy of the robots.

Table.2 Comparison of run time (200 Steps)

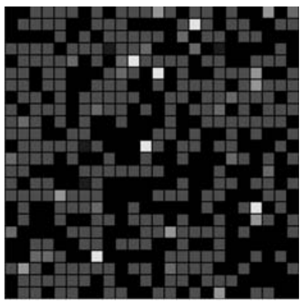
Software processing	Hardware processing
7.324933 (s)	0.143067 (s)

The proposed hardware accelerator achieved 50 times the speed on average compared with software processing, and realized acquisition of the behavior which maintains balance between fish and plankton.

In relation to future works, simulation in complex quasi-ecosystem is the most important priority. We will also apply the proposed evolutionary robotic to simulation of artificial life.



(1) Initial state



(2) 1000th steps

Fig.19 State in the quasi-ecosystems

Acknowledgments

This study was partially supported by the telecommunication advancement foundation. The authors would like to thank their supports.

References:

[1] A.Ishiguro, S.Tokura, T.Kondo, Y.Uchikawa, P.Eggenberger, "Reduction of the gap between simulated and real environments in evolutionary robotics: a dynamically-rearranging neural network approach", Proc. of IEEE conf. Int. on Systems, Man, and Cybernetics, Vol3, pp.239-244, 1999.

[2] J.Holland: Adaptation in Natural Artificial Systems, the University of Michigan Press (Second edition; MIT Press), 1992.

[3] D.E.Goldberg: Genetic algorithms in search optimization, and machine learning; Addison Wesley, 1989.

[4] M.Yoshikawa, T.Kihira, H.Terai, "Q-learning based on hierarchical evolutionary mechanism", WSEAS Trans. on Systems and Control, Issue 3, Vol.3, pp.219-228, 2008.

[5] M.Alfonceca, M.Cebrian, A.Ortega, "A Fitness Function for Computer-Generated Music using Genetic Algorithms",WSEAS Trans. on Information Science & Applications, Issue 3, Vol.3, pp.518-525, 2006.

[6]M.Yoshikawa, H.Terai, "Bus-Oriented Floorplanning Technique Using Genetic Algorithm", WSEAS Trans. on Circuits and System, Issue2, Vol.6, pp.253-258, 2007.

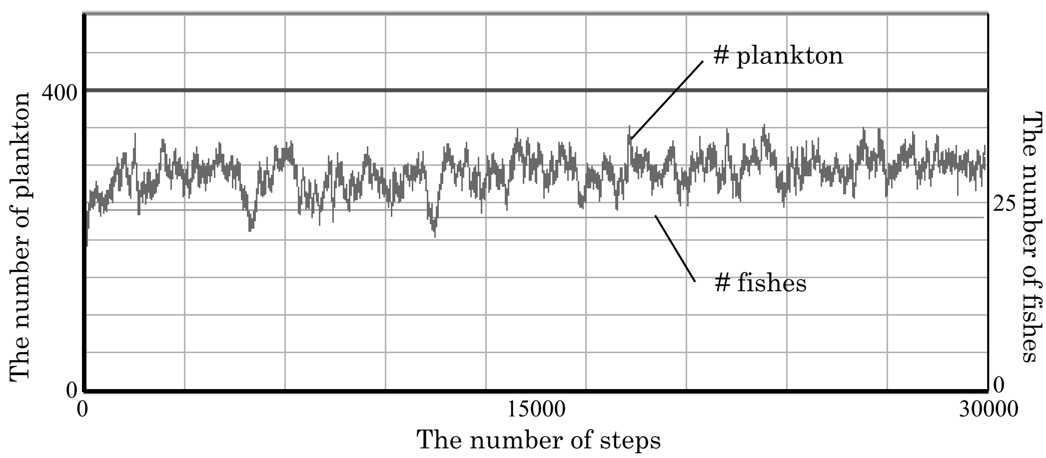


Fig.20 Transition of the number of fish and planktons

[7] N.Kubota, M.Mihara, F.Kojima, " Evolutionary robotics for quasi-ecosystem", Proc. of Congress on Evolutionary Computation, Vol.1, pp.115-120, 2001.
[8] N.Kubota, M.Ogishi, F.Kojima, "Learning of multiple robots in quasi-ecosystem", Proc. of 26th Annual Conference of the IEEE Industrial Electronics Society, Vol.3, pp.2105-2110, 2000.
[9] N.Kubota, M.Mihara, "Multi-objective behavior coordination of multiple robots interacting with a dynamic environment", Proc. of The 12th IEEE International Conference on Fuzzy Systems, Vol.1, pp.248-253, 2003.
[10] S.D.Scott, A.Samal, S.Seth, "HGA:A Hardware -Based Genetic Algorithm", Proc. of International Symposium on Field-Programmable Gate Array, pp.53-59, 1995.
[11] P.Graham. B.Nelson, "Genetic Algorithm in Software and in Hardware A performance Analysis of Workstation and custom Computing Machine Implementations", FPGAs for Custom Computing Machines, pp.216-225, 1996.
[12] T.Imai, M.Yoshikawa, H.Terai, H.Yamauchi, "Scalable GA-Processor Architecture and Its Implementation of Processor Element", Proc. of IEEE, International Conference on Acoustics, Speech, and Signal Processing, Vol.3, pp.3148-3151, 2002.