Proceedings of the 2002 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
EPFL, Lausanne, Switzerland • October 2002

# Online Robot Learning by Reward and Punishment for a Mobile Robot

Dejvuth Suwimonteerabuth, Prabhas Chongstitvatana

*Department of Computer Engineering*
*Chulalongkorn University, Bangkok, Thailand*
*prabhas@chula.ac.th*

## Abstract

*The existing robot learning methods required specifically defined goals. We aim to produce a more plastic behavior. We present our work which a human observer can influence the robot behavior. The robot learns from reward and punishment from a human in real-time. To examine the developed approach, we evolved a control system for a color-following task as an example. A physical robot is used to perform the experiments. Experimental results show the emergence of learned behaviors. We discussed the factors that influence the learning process.*

## 1. Introduction

In the field of Evolutionary Robotics, researchers have mainly tried to show the validity of their approaches by evolving some pre-specified behaviors. For example, the example-based approach is used for evolving robot controllers in the collision avoidance task [1]. The locally weighted learning (LWL) algorithms is applied on the learning of devil-sticking, pole-balancing of a humanoid robot arm, and inverse-dynamic learning for a seven degree-of-freedom robot [2]. The task primitives in robot learning from observation are employed to play a virtual and an actual air hockey game [3]. Genetic programming is also utilized to evolve the behavior primitives and behavior arbitrators for box-pushing task [4, 5]. A perceptron is required in evolving the exploration and homing behavior [6]. In these works, the desired task of robot must be clearly defined.

In contrast, this paper proposes a different approach: the desired behavior of robot is not obtained from prior knowledge but from knowledge that is given after learning process begins. Such knowledge is directly acquired from a human observer using *reward and punishment.* We aim to realize a robot controller that make robot perform the behavior that will receive reward and avoid the behavior that will receive punishment. Therefore, no prior knowledge about the desired task is needed. This approach is very advantageous to the learning problems since it is realized that building a robot behavior by iteratively dealing with the robot-environment interactions is a tough and tedious job. Especially when the complexity increases, it would go beyond a designer's capability to construct all the behaviors. Our work is similar to "robot shaping" [7, 8] that we use learning to translate feedback from an external observer into a control strategy. The strategy that we used is comparable to a simple case of reinforcement learning [9].

The robot used in this work is illustrated in Figure 1. The robot is capable of detecting color of the floor and able to receive reward and punishment signal from a remote transmitter. Furthermore, as another important feature, the robot must move around the designated area without colliding walls. This is accomplished by using front and rear infrared sensor.

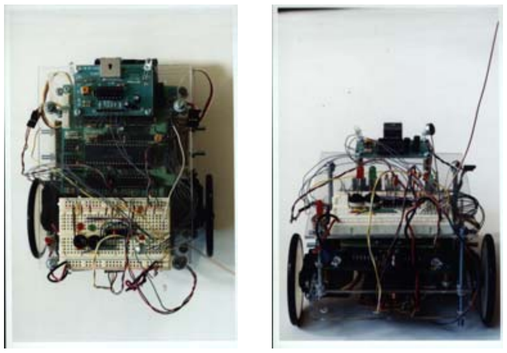
***Figure 1:*** *The mobile robot used in this work*

A *Finite-State Machine (FSM)* is used as robot controller, which determines the behavior of robot.

A method for synthesizing a FSM from sequences of partial input/output is based on *Genetic Algorithms (GA)* [10, 11]. GA is a search and optimization algorithm that simulates the natural evolution. GA performs search in a population, a set of individuals that represents points in the search space. At each generation, a sequence of genetic operations called selection, reproduction, crossover, and mutation transforms the existing individuals into a new set of solutions. The quality of solution is evaluated in terms of fitness in which *fitness function* must be defined for each problem. The individuals are probabilistically selected to the next generation proportional to their fitness. Examples of using GA to synthesize FSM can be found in [12, 13].

Learning using a physical robot takes a long time because of its limit on mechanical speed. To help speed up parameter tuning process we wrote a simulator and use this simulator to find out the appropriate parameters for GA. The physical robot is used online in real time to study the learning characteristics and the robot behaviors with feedback from a human trainer.

Moreover, we study a number of factors that are relevant to the quality of learning. The relevant factors are: no reward/punishment, start position, maximum number of reward/punishment, size of population, and the person that give reward/punishment signals. All details will be described later.

## 2. The Experiments

### 2.1 Tasks

The environment experienced by the robot is a plain rectangular floor, size 1.5 × 2.2 m., surrounded by walls. The robot is able to detect walls and stays inside the designated floor all the time. The floor is painted with two colors: black and white; which is divided exactly at the middle of the floor. The goal of this experiment is that the robot will learn to stay in the color, which reward signals are presented, and also learn to move out of the color which punishment signals are given. For example, if the robot receives rewards when it is on a white floor, and receives punishments when it is on a black floor, the robot will eventually learn to stay in the white floor and move out of the black floor.

### 2.2 Control programs

The program used to control the robot has a single input, which is the color of the floor. Zero means the robot is on white floor and one means the robot is on black floor. Two-bit output is used to control motions of robot, as described in Table 1.

***Table 1:*** *Outputs of program and their corresponding motions.*

| Output | Motions |
|---|---|
| 00 | Move forward |
| 01 | Turn left |
| 10 | Turn right |
| 11 | Move backward |

### 2.3 Genetic Algorithms

The learning behaviors of robot are based on GA. The algorithm slightly adapted from [14] using combined rank to promote diversity. The rank constant is 0.6666. Due to real time requirement we set the population size quite small (any smaller population causes GA to get stuck at local minima). The population size is 20. We use elitism and recombine 6 individuals. The mutation is performed on 13 individuals with the probability 0.01.

### 2.4 Encoding Scheme

Each individual represents an FSM by its state transition diagram. The state transition diagram is represented by concatenating all of the outputs followed by the next states to form a fixed length binary string. Figure 2 shows an example. Since we assume no prior knowledge about the tasks, the number of internal states needed to produce a complete solution is unknown. Thus, we let the number of states of an individual to be larger than the minimum required by the task, as 8 states are used. The solution may contain redundant states and unreachable states. A conventional method can be used to optimize them.

### 2.5 Fitness Function

The fitness of an individual is evaluated by the following steps:

```
fitness, $f = 0$
Reset individual (FSM) to start state
Initialize time, $\tau = 1000$
While $\tau > 0$ Do
```

```
    If a reward detected, $f = f + 1$
    If a punishment detected, $f = f - 1$
    Move the robot according to current FSM
    $\tau = \tau - 1$
End while
```

Each state machine has about 30 seconds execution time in the real world during which the robot behavior is evaluated by a human trainer. The human observer who gives reward and punishment signals is directly control the fitness value. Therefore, that person must clearly understand the promising behaviors in order to teach the robot to accomplish the desired tasks. Some important reward and punishment giving techniques can be used to accelerate the learning tendency, such as

- If robot is on a white floor and moving in the way that will not go to the black floor, give more rewards.
- If robot is on a black floor and moving in the way that will go out of the black floor, also give some rewards.

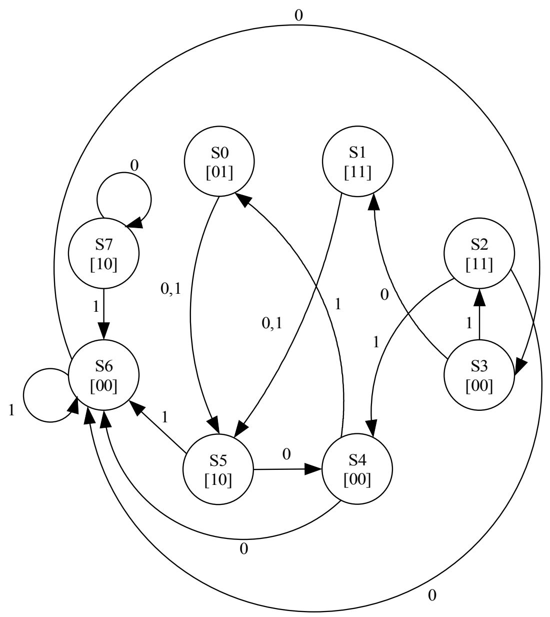

***Figure 2:*** *An example FSM representing "01 11 11 00 00 10 00 10 101101 101101 110100 001010 110000 100110 011110 111110".*

### 2.6 Details of the Experiments

The experiments were carried out in order to study factors that are relevant to the quality of learning. The experiments are divided into 6 problems, that is, giving no reward/punishment, changing the start position, maximum number of reward/punishment, size of population, and the person who give reward/ punishment signals. Assume that we want the robot to stay in the white floor and move out of the black floor.

For each problem, unless otherwise stated, experiments are performed a couple of times, 60 minutes each. The measurement of how good the robot performed it task is the time that it stays in the designated color. A result is logged every 5 minutes, which is the time the robot stays in the white floor. Dividing this number by 5 results in a *robot-on-white-floor ratio* indicating the learning tendency in the last 5 minutes. Consequently, the learning behavior is illustrated by a graph, which x-axis represents time and y-axis represents robot-on-white-floor ratio. Full details of each problem are described along with its results.

## 3. The Experimental Results

### 3.1 Learning behavior when giving no reward/punishment

The purpose of this experiment is to show that the learning behavior does not exist when giving no reward/punishment. Since GA use fitness function to determine the quality of each individuals, in this case, all individuals' fitness are equal to zero. As a result, no learning occurs. Figure 3 indicates the result that was experimented 3 times, 40 minutes each.

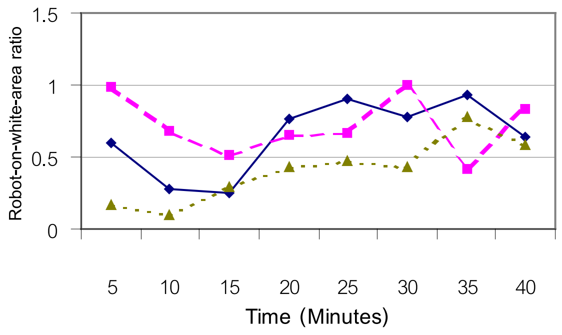

***Figure 3:*** *The learning behavior when giving no reward/punishment.*

### 3.2 Learning behavior when start position is in white area

Figure 4 shows the learning result when normally given reward/punishment, and the robot starts learning on the white floor. The average value of 3-times experiments is also provided. At first, the robot stayed mostly on the white floor, but for some later time (about 15 minutes), it moved and stayed mostly on the black floor. This behavior is caused by the initial randomness of FSM. When some more time had passed (about 25-30 minutes), the learning behavior was arisen. The learning process continued until about the minute 45 the robot completed its learning and stayed in the white floor almost all the time.

During the experiments, we observed that some robot movements result in few displacements, such as moving forward and backward repeatedly. If such movements had appeared when the robot was on the black floor, it would not have tried to move out of the unwanted black floor, and, of course, decreased the robot-on-white-floor ratio (the minute 15). Many punishments were given in this case; therefore, in some later time (the minute 30), the movement was altered in the way that more displacements were conducted, such as continuously moving forward. On the other hand, we want the robot to stay in the white floor, so we gave more rewards to individuals that tended to exhibit less displacement. The task was finally achieved when the robot's displacement was approaching zero when it was on the white floor (the minute 45).

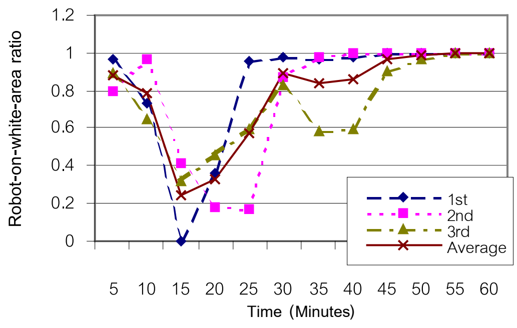

***Figure 4:*** *The learning behavior when start position is in white area.*

### 3.3 Learning behavior when start position is in black area

Experiments were carried out 2 times, as depicted in Figure 5. In the first time, the learning behavior was rapidly come into view because of some random individuals in the initial population exhibit such behavior. The second time was dissimilar. The learning still took place, but in slower manner than first run. The explanation is that the robot only learned to escape the black area in the former part of experiment, but not learned to stay in the white area. The learning progressed gradually in the latter part.

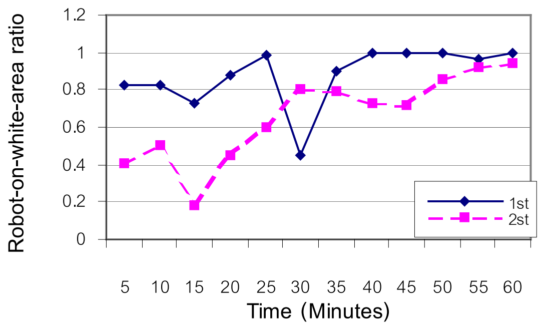

***Figure 5:*** *The learning behavior when start position is in black area.*

### 3.4 Learning behavior when limiting maximum number of reward/punishment

As stated above, the reward and punishment directly control fitness value, and thus, the learning behavior must be changed if the number of reward and punishment are changed. This experiment was performed twice, varying the maximum number that each individual will receive reward and punishment to 3 and 6 times compared with the average in problem B in which no limit is presented. If a individual received more than the indicated value, its fitness value is left unchanged.

The results are explained in Figure 6, pointing out that the rate of learning is reduced when the maximum number of reward and punishment is decreased. Consequently, when limit equals to six the quality of learning is better than when limit equals to three, but no better than when no limit defined.

### 3.5 Learning behavior when varying size of population

Size of population has an effect on the diversity of all individuals. The larger population increases the sample size of finding good solutions. This experiment is performed twice, varying the size of population to 5 and 10 individuals compared with the average in problem B that use 20 individuals.

Figure 7 illustrated the results. In the case of 5 individuals, the learning quality was low, but still can partially learned in the latter part of experiment. The size of population used here is too small, hence good solutions cannot be maintained even when GA can find ones.

However, when the size of population is 10, the result was much better. The learning rate is better than the case of 20 individuals whereas the size is half of it. The reason is that when the size of population is appropriately reduced, time used for searching solutions in each generation is accordingly decreased. The appearance of good solutions and the disappearance of bad solutions are faster. For example, if the current solution makes the robot exhibits undesired behavior (receive many punishments), this solution will be wiped out in a short period of time.

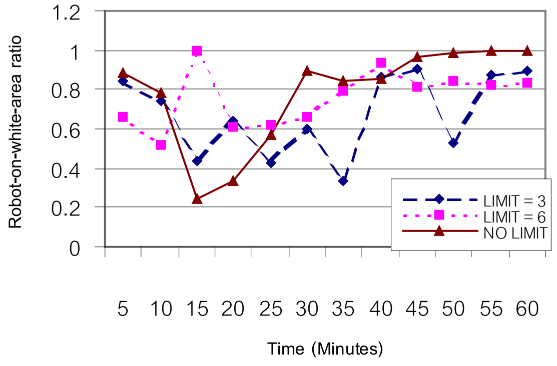

***Figure 6:*** *The learning behavior when limiting number of reward/punishment.*

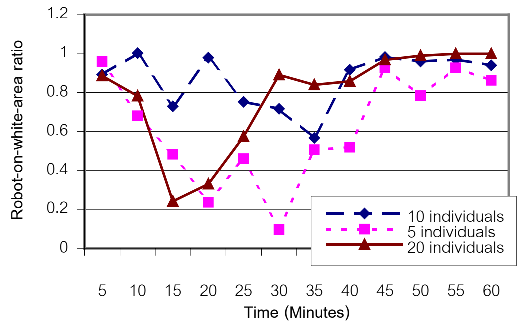

***Figure 7:*** *The learning behavior when varying size of population.*

### 3.6 Learning behavior when changing person who gives reward/punishment

Since the knowledge of person who gives reward and punishment is an important factor for the quality of learning, this experiment suggests how the learning behavior is changed when the person who gives reward and punishment changed as illustrated in Figure 8. The experiment was performed 3 times by three different human teachers. Comparing the result with the average in problem B. The result shows the promising learning behavior in the first and second runs, but not in the third run.

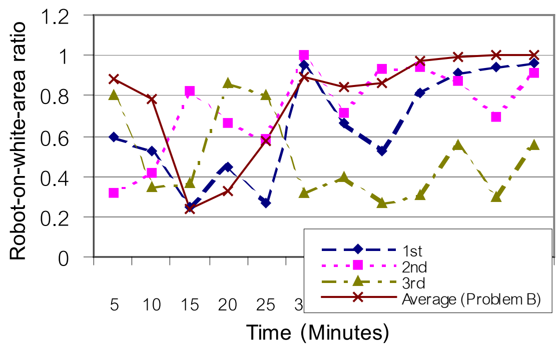

***Figure 8:*** *The learning behavior when changing person who gives reward/punishment*

## 4. Conclusions

This work proposes an empirical study of the learning behavior using reward and punishment. The goal of this work is that the robot must learn the behavior that will receive reward and avoid the behavior that will receive punishment. To verify the proposed approach, we evolved controllers for a color-following task and the preliminary results show the promise of our approach. Moreover, this work also presents the analysis of the quality of learning when some learning-relevant factors are changed.

The experiments use genetic algorithms in order to find the controller in the form of a finite state machine. At first, we adjusted all GA-relevant parameters in a computer simulation, and used these parameters in a real mobile robot. Then, we studied the learning rate and motion characteristics of a robot for each problem.

Our work presented here points to some prospects of future research. The most obvious way is to use the proposed approach to evolve controllers for more complicated tasks to further examine its generality.

## 5. References

[1] W.-P. Lee and S.-F. Lai. "An Example-Based Approach for Evolving Robot Controllers", In *Proceedings IEEE International Conference on System, Man, and Cybernetics*, 1999, Volume: 5, pp. 618-623.

[2] S. Schaal, C. G. Atkeson, and Vijayakumar. "Real-Time Robot Learning With Locally Weighted Statistical Learning", In *Proceeding IEEE International Conference on ICRA '00.*, Volume: 1 , 2000.

[3] D. C. Bentivegna, C. G. Atkeson. "Learning From Observation Using Primitives", In *Proceedings IEEE International Conference on 2001 ICRA*, Volume: 2 , 2001.

[4] W.-P. Lee, J. Hallam, H. H. Lund. "Applying Genetic Programming to Evolve Behavior Primitives and Arbitrators for Mobile Robots", In *Proceeding of IEEE International Conference on Evolutionary Computation* , 1997, pp. 495-499.

[5] J. R. Koza and J. P. Rice, "Automatic programming of robots using genetic programming", AAAI-92 Proc. 10th National Conf. on AI, 1992, pp. 194-201.

[6] H. H. Lund, J. Hallam. "Evolving Sufficient Robot Controllers", In *Proceeding of IEEE International Conference on Evolutionary Computation*, 1997.

[7] M. Dorigo, "*ALECSYS and the AutonoMouse: Learning to control a real robot by distributed classifier systems*", Machine learning, vol. 19, 1995, pp.209-240.

[8] M. Dorigo and M. Colombetti, "*Robot shaping: an experiment in behavior engineering*", MIT Press, 1998.

[9] L. P. Kaelbling and M. L. Littman and A. P. Moore, "*Reinforcement Learning: A* Survey", Journal of Artificial Intelligence Research, vol. 4, pp.237-285, 1996.

[10] D. E. Goldberg. *Genetic Algorithm in search, optimization, and machine learning.* Addison-Wesley, 1989.

[11] J. H. Holland, "*Adaptation in natural and artificial systems*", Ann Arbor, MI: University of Michigan Press, 1975.

[12] P. Chongstitvatana and C. Aporntewan. "Improving Correctness of Finite-State Machine Synthesis from Multiple Partial Input/Output Sequences". In *Proceedings of the first NASA/DoD Workshop on Evolvable Hardware*, , 1999, pp. 262-266.

[13] C. Aporntewan and P. Chongstitvatana. "An On-Line Evolvable Hardware for Learning Finite-State Machine", *Proc. of Int. Conf. on Intelligent Technologies*, Bangkok, December 13-15, 2000, pp.125-134.

[14] P. H. Winston. *Artificial Intelligence*. Addision-Wesley. 1992.