Proceedings of the 2000 IEEE International Workshop
on Robot and Human Interactive Communication
Osaka, Japan - September 27-29 2000

Interactive Classifier System for Real Robot Learning

D. Katagami S. Yamada

CISS, IGSSE, Tokyo Institute of Technology
4259 Nagatsuta, Midori-ku Yokohama 226-8502, JAPAN
{katagami,yamada}@ymd.dis.titech.ac.jp

Abstract

In this paper, we describe a fast learning method for a mobile robot which acquires autonomous behaviors from interaction between a human and a robot. We develop a behavior learning method ICS (Interactive Classifier System) using evolutionary computation and a mobile robot is able to quickly learn rules so that a human operator can directly teach a physical robot. Also the ICS is a novel evolutionary robotics approach using an adaptive classifier system to environmental changes. The ICS has two major characteristics for evolutionary robotics. For one thing, it can speedup learning by means of generating initial individuals from human-robot interaction. For another, it is a kind of incremental learning methods which adds new acquired rules to priori knowledge by teaching from human-robot interaction at any time.

1 Introduction

In previous robot learning studies, optimization of parameters has been applied to acquire suitable behaviors in a real environment. Also in most of such studies, a model of human evaluation has been used for validation of learned behaviors. However, since it is very difficult to build human evaluation function and adjust parameters, a system hardly learns behavior intended by a human operator.

In contrast with modeling human evaluation analytically, we introduce another approach in which a system learns suitable behavior using human direct evaluation without its modeling. Such an interactive method with *Evolutionary Computation (EC)* as a search algorithm is called *Interactive EC*[1][2], and a lot of studies on it have been done thus far[3][4][6].

The most significant issue of *Interactive EC* is how it reduces human teaching load. The human operator needs to evaluate a lot of individuals at every generation, and this evaluation makes him/her so tired. Specially in the *interactive EC* applied to robotics, the execution of behaviors by a robot significantly costs and a human operator can not endure such a boring task.

Additionally reinforcement learning has been applied to robot learning in a real environment[5]. Unfortunately the learning takes pretty much time to converge. Furthermore, when a robot hardly gets the first reward because of no priori knowledge, the learning convergence becomes far slower. Since most of the time that are necessary for one time of action moreover is spent in processing time of sense system and action system of a robot, the reduction of learning trials is necessary to speedup the learning.

In our study, we build an *interactive EC* framework which quickly learns rules with operation signal of a robot by a human operator as teacher signal. Our objective is to make initial learning more efficient and learn the behaviors that a human operator intended through interaction with him/her. To the purpose, we utilize a classifier system as a learner because it is able to learn suitable behaviors by the small number of trials, and also extend the classifier system to be adaptive to a dynamic environment.

In this system, a human operator instructs a mobile robot while watching the information that a robot can acquire as sensor information and camera information of a robot shown on the screen top. In other words, the operator acquire information from a viewpoint of a robot instead of a viewpoint of a designer. We call such a framework **Interactive Classifier System (ICS)**, and Fig.2 shows its overview.

Operator performs teaching with joystick by direct operating a physical robot. the **ICS** inform operator about robot's state by a robot send a vibration signal of joystick to the **ICS** according to inside state.

Our **ICS** is significantly different from previous passive interactive techniques in which a human selects good individuals from given ones. The **ICS** is an active interactive technique to learn suitable rules because a human is allowed to actively operate it. This technique reduces the psychological load of an operator and does learning without getting tired. Because a user must evaluate each aquwired rule at every steps in previous wroks with *EC*. This is a problem because of the psychological load and a physical one. However the user always add new rules by which operate a mobile robot directly at any time in **ICS**. For this reason, this technique reduces the psychological load.

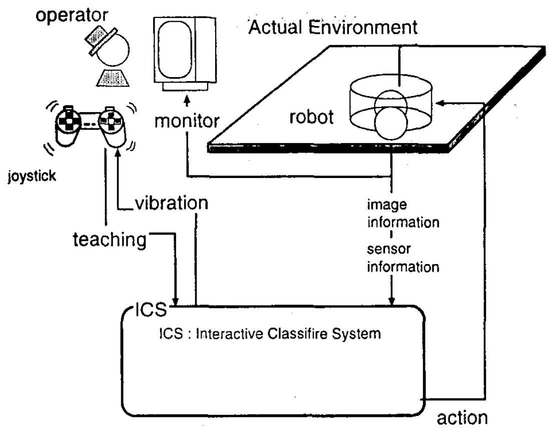

Figure 1 Overview of Interactive Classifier System

We investigate the effectiveness of **ICS** by a task of wall-following and soccer game in which real robots are used.

2 Classifier system

(Learning) Classifier System (CS, LCS) is a kind of method of Genetic Algoritm-Based Machine Learning (GBML). LCS consists in Performance Function based on Production System, Reinforcement Learning based on Credit Assignment and Rule Generation Function based on GA.

In this study, we used *XCS*[7][8] as a classifier system for **ICS**. *XCS* is a classifier system in which each classifier maintains a prediction of expected payoff, but the classifier's fitness is given by a measure of the prediction's accuracy. These aspects of *XCS* result in its population tending to form a complete and accurate mapping $X \times A \rightarrow P$ form inputs and actoins to payoff predictions.

3 Interactive Classifier System

ICS consists of a rule generation component (RGC), a sensor processing component (SPC), a display component (DC) and a reinforcement component (RC).

The rule generation component make a new classifier from teaching by an operator. The SPC processes each information of infrared sensors, optical sensors and CCD camera, and through it for the RGC. The DC displays by GUI interface and processes the input from a joystick. Finally the RC performs learning by updating parameters in **ICS**.

At first, a human operates robots with a joystick by viewing sensor information displayed on GUI interface, and the DC processes the information. Next, the SPC gets interaction and sensor information. The RGC make a new rules from them and adds them into a rule list. When nothing is input from the operotor, a mobile robot executes autonomous behaviors from interaction. Finally, the RC reinforces the classifiers by updating their parameters in the actions which were previously executed.

ICS differs from *IEC* in that operators have the load for evaluation of individuals at every time. However the opeartor always can operate a mobile robot directly, and such direct operation of a robot is considered for a human operator far less than the fitness evaluation in IEC. Therefore the operator can do teaching without the load, and always can do conentrative additional learning for sub-tasks difficult to achievement. Fig.1 shows overview of **ICS**.

3.1 Rule Generation Component (RGC)

In previous methods, the initial individuals are generated randomly and learn at an early stage through trial and error. Our method speedup the learning at an early stage by automatically generating the initial individuals from interacton between a robot and a human operator. Whenever this component generates a new classifier, the population is scanned to see if the new classifier has the same conditon and action as any existing classifier. If there is a classifier with identical condition and action, the classifier is incremented by one. If there is no existing classifier with identical condition and action, the new classifier is added to the population. Consequently, our method can subjectively evaluate as same as *IEC* which integrate human evaluation into our system.

Fig.2 shows the schematic illustration of **ICS**.

3.2 Display Component (DC)

As mentioned before, our system consists of the RGC which generates a new classifier from human-robot interaction, the SPC which processes infrared sensors and images from a CCD camera and the DC which has GUI interface and processes input from a joystick. This system is developed with C langage and GTK+ on Linux and using Video4Linux for image processing. Fig.3 shows developed GUI interface on DC.

3.3 Sensor Processing Component (SPC)

We use the joystick for a SONY PlayStation via parallel port as a joystick on SPC for teaching by operator. The specification of the joystick is as followings.

- It has fourteen buttons and the system can use all buttons in degital mode.
- Moreover, the system can use two buttons in analog mode.
- It has two vibration modes.

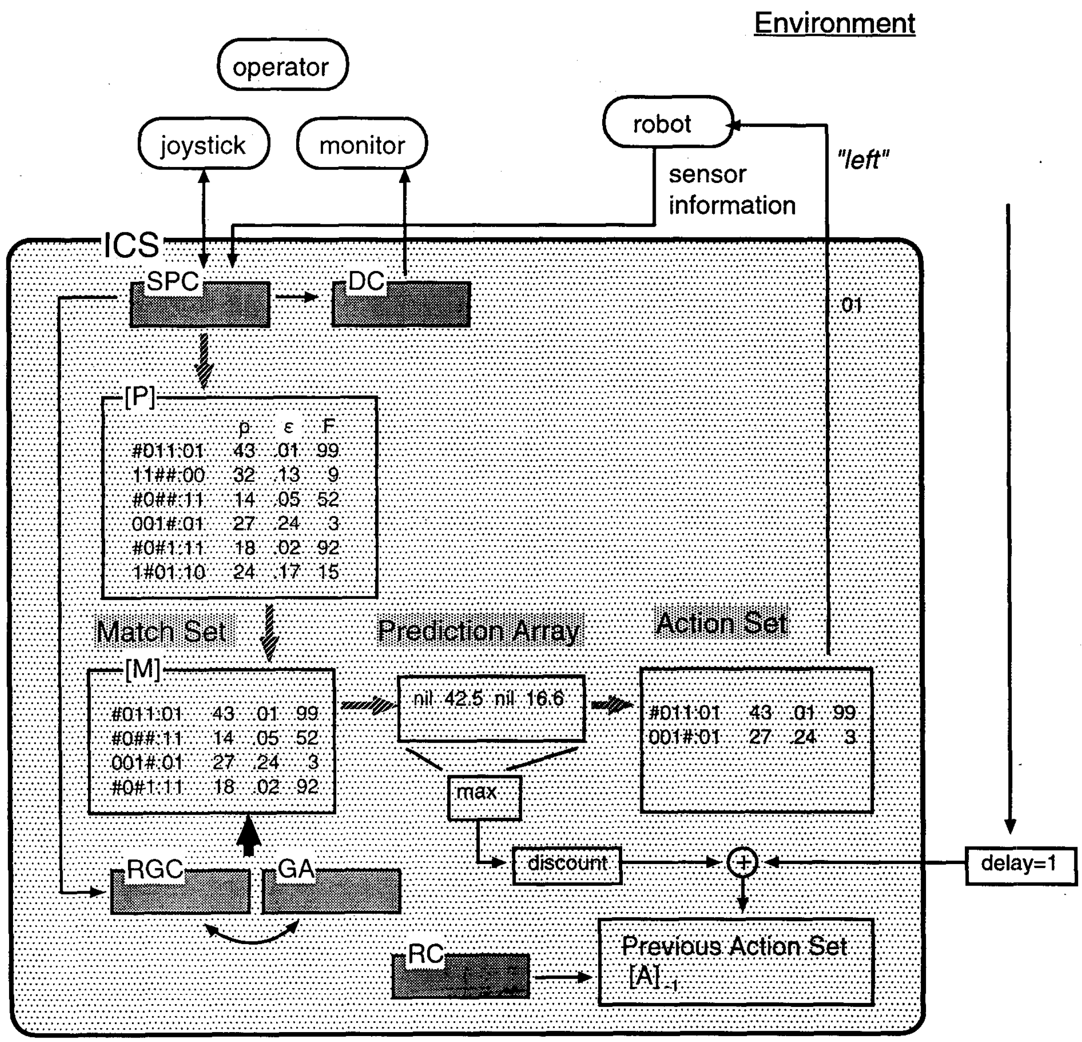

Figure 2 Schematic illustration of **ICS**

Because of this two kinds of vibration, the system can communicate stimulus to an operator directly. To take an example use vibration functions, an operator can feel directly robot's sensations because the system communicate stimulus by vibration when the mobile robot approach obstacles. Hence, we think the system have an effect on acquisition of a reflective behavior and an instinctive one by teaching.

3.4 Renforcement Component (RC)

the RC reinforces the classifiers by updating their parameters in the actions which were previously executed.

4 Some charactaristics of ICS

4.1 Initial individuals from human-robot interaction

In previous works of robot learning in a real environment, the learning takes pretty much time to converge because of learning by trial and error. In this work, we assumed that this learning by trial and error is a probrem in a real environment. However, it hardly prepare suitable priori knowledge in a environment in advance. For this reason, the **ICS** generate initial individuals by teaching from human-robot interaction. We can perform efficiently initial learning in this way.

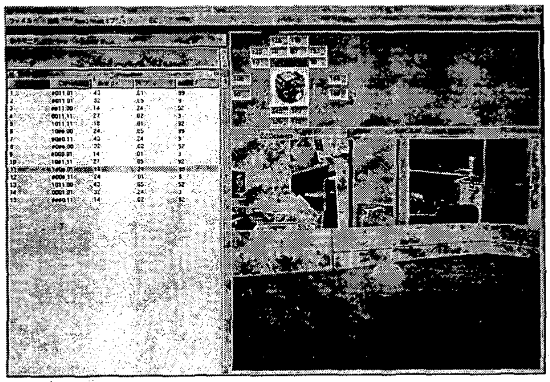

Figure 3 User Interface

4.2 Adding rules at any time

In previous wroks with *EC*, a user must evaluate each aquwired rule at every steps. This is a problem because of a mental load and a physical one. In contrast with EC, the user always add new rules by which operate a mobile robot directly at any time in **ICS**. Therefore the user can do teaching without the load, and always can do conentrative incremental learning for sub-tasks difficult to achievement.

4.3 Teaching method

ICS performs two modes: a teaching mode and an autonomous behavior mode by turns. The procedures of the two modes are shown in the following.

Teaching mode

Step1 Opeator have teaching to mobile robot by performing tasks practically.

Step2 ICS generate a new rule from human-robot interaction and sensor information.

Step3 If there is no existing classifier with identical condition and action, the new classifier is added to the population.

Step4 If there is existing classifier with identical condition and action, the existing classifier is incremented by one.

This procedure is as same as classifier generation method of *XCS* called for *Macroclassifiers*[8].

Autonomous behaviour mode

Step1 The robot behaves by conforming to stored rules in Rule List.

Step2 If there is no rule to match input in Rule List, perform a new rule generated from any other rules.

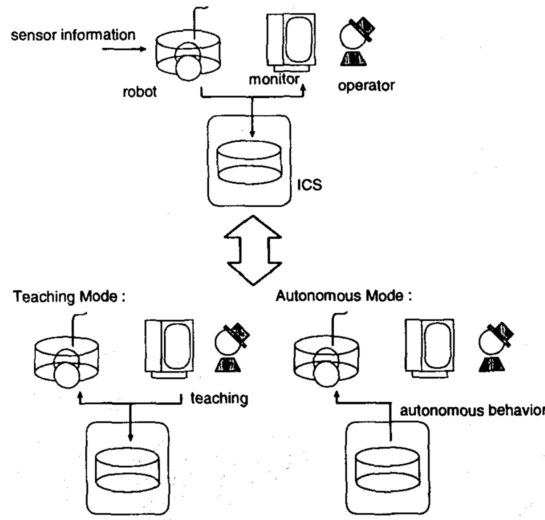

Figure 4 Teaching Mode and Autonomous Mode

Step3 The genetic algorithm acts on the match set after fixed time steps.

In the autonomous behavior mode, the system finds a similar stored rule and executes it. If the system don't find a similar rule, it generates a new rule integrated with any human-robot interaction. Fig.4 shows the overview of a teaching mode and an autonomous behavior mode.

4.4 Mining and recycling of rules

ICS extract basic rules in each environment from acqueired rules by ICS. We use datamining tools to realize. In contrast with the learning by trial and error, **ICS** can perfrom initial learning in a real environment by recycling acquwired rules as initial rules acquwired rules in new niche envrinment.

5 Experiment environment

5.1 Environment settings

The experiments were performed with a standard autonomous miniature robot Khepera (Fig.5). It is equipped with eight infrared proximity sensors. The mobile robot has a circular shape, a diameter of 6 cm and a height of 5 cm. It possesses two motors and onboard power supply. The motors can be independently controlled by a PID controller. The eight infrared sensors are distributed around the robot in a circular pattern (Fig.5). They emit infrared light, receive the reflected light and measure distances in a short range: 2-5 cm. The robot is also equipped with a Motorola 68331

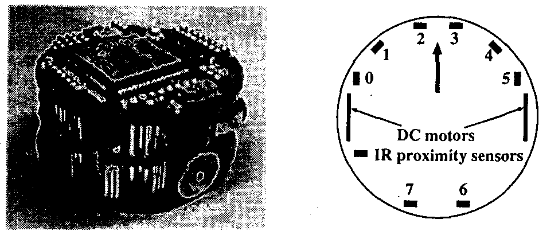

Figure 5 A mobile robot: Khepera

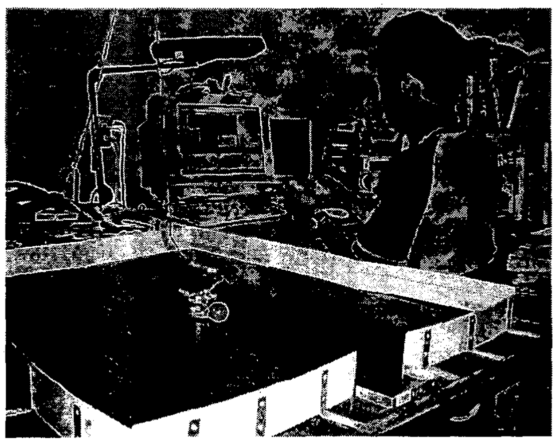

Figure 6 Teaching based on sensor information

micro-controller which can be connected to a computer via serial cable.

5.2 Teaching method

Teaching based on sensor information An operator performs teaching to the mobile robot, not from a view point of overlooking the envronment, but teach as seeing displayed sensor information (in other words, from robot's view point) on GUI interface and images from a CCD camera. As a consequence, operator performs teaching within the bounds of information the robot is able to acquire. Thus **ICS** generates suitable robot program easily.

Fig.6 shows the appearance of teaching based on sensor information.

Teaching based on observation In contrast to teaching based on sensor information, an operator observes the mobile robot directly from a view point of overlooking the envronment and considers next operation for teaching. Thus, the system can perform teaching save labor for tasks in need of a detailed teaching.

Fig.7 shows the appearance of teaching based on observation by human.

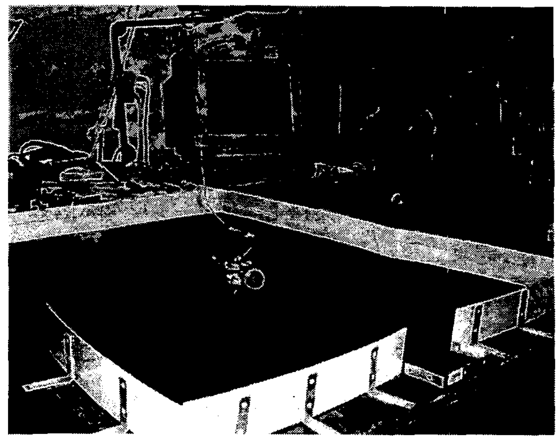

Figure 7 Teaching based on observation

5.3 Tasks of experiments

We will make experiments with two kinds of tasks. As exmanle tasks, a wall following task with a single learning robot and a simplified soccer game with multiple learning robots.

A wall following task In a wall following task, the purpose is to acquire a wall following behavior. Several human operators perform teaching to a robot respectively. If each acquired robot program is same each other, we consider it is a solution of a wall following task. We will make experiments with above two teaching and compare with previous work.

A soccer game with multi robots In a simplified soccer game with multiple learning robots, **ICS** extracts the difference of acquired rules because of cooperative and competitive behaviors.

Fig.8 shows overview of experimental environment.

we analyze rules acquired in experiment and consider it.

6 Conclusion

We proposed a fast learning method based on **ICS** for mobile robots which acquire autonomous behaviors from experience of interaction between a human and a robot.

ICS has two major characteristics. For one thingm, the **ICS** generate initial individuals by teaching from human-robot interaction. We can perform efficiently initial learning in this way. For another, the user always add new rules by which operate a mobile robot directly at any time in **ICS**. Therefore the user can do teaching without the load, and always can do conentrative incremental learning for sub-tasks difficult to achievement.

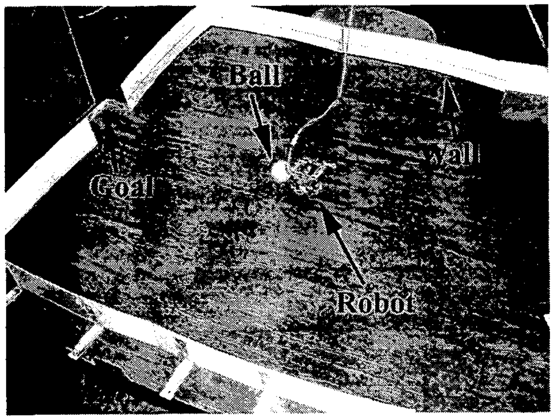

Figure 8 Experimental environment for soccer tasks

Consequently, ICS can speedup robot learning in a real environment and extract complexed robot program which human hardly generate.

References

[1] R. Dawkins. *The Blind Watchmaker.* Longman, Essex, 1986.

[2] R. Dawkins. The Evolution of Evolvability. In Langton, C. G., editor, Artificial Life, pages 201–220. Addison-Wesley, 1989.

[3] Y. Nakanishi. Capturing Preference into a Function Using Interactions with a Manual Evolutionary Design Aid System. Genetic Programming, pages 133–140, 1996.

[4] M. Ohsaki, H. Takagi, and T. Ingu. Methods to Reduce the Human Burden of Interactive Evolutionary Computation. Asian Fuzzy System Symposium (AFSS'98), pages 495–500, 1998.

[5] E. Uchibe, M. Asada, and K. Hosoda. Behavior coordination for a mobile robot using modular reinforcement learning. In IEEE/RSJ International Conference on Intelligent Robots and Systems 1996 (IROS96), pages 1329–1336, 1996.

[6] T. Unemi. A Design of Multi-Field User Interface for Sumulated Brdding. Asian Fuzzy System Symposium (AFSS'98), pages 489–494, 1998.

[7] S.W. Wilson. ZCS: a zeroth order classifier system. Evolutionary Computation, 2:1–18, 1994.

[8] S.W. Wilson. Classifier fitness based on accuracy. Evolutionary Computation, 3(2):149–175, 1995.