Multi-Layered Learning System for Real Robot Behavior Acquisition

Yasutake Takahashi and Minoru Asada
Department of Adaptive Machine Systems,
Graduate School of Engineering, Osaka University
Yamadagaoka 2-1, Suita, Osaka, 565-0871, Japan
{yasutake, asada}@ams.eng.osaka-u.ac.jp

Abstract

This paper presents a series of the studies of multi-layered learning system for vision-based behavior acquisition of a real mobile robot. The work of this system aims at building an autonomous robot which is able to develop its knowledge and behaviors from low level to higher one through the interaction with the environment in its life. The system creates leaning modules with small limited resources, acquires purposive behaviors with compact state spaces, and abstracts states and actions with the learned modules. To show the validity of the proposed methods, we apply them to simple soccer situations in the context of RoboCup (Asada *et al.* 1999) with real robots, and show the experimental results.

Introduction

One of the main concern about autonomous robots is how to implement a system with learning capability to acquire both varieties of knowledge and behaviors through the interaction between the robot and the environment during its life time. There have been a lot of work on different learning approaches for robots to acquire behaviors based on the methods such as reinforcement learning, genetic algorithms, and so on. Especially, reinforcement learning has recently been receiving increased attention as a method for behavior learning with little or no a priori knowledge and higher capability of reactive and adaptive behaviors. However, simple and straightforward application of reinforcement learning methods to real robot tasks is considerably difficult due to its almost endless exploration of which time is easily scaled up exponentially with the size of the state/action spaces, that seems almost impossible from a practical viewpoint.

One of the potential solutions might be application of so-called "mixture of experts" proposed by Jacobs and Jordan (Jacobs *et al.* 1991), in which a set of expert modules learn and one gating system weights the output of the each expert module for the final system output. This idea is very general and has a wide range of applications. However, we have to consider the following two issues to apply it to the real robot tasks:

- Task decomposition: how to find a set of simple behaviors and assign each of them to a learning module or an expert in order to achieve the given initial task. Usually, human designer carefully decompose the long time-scale task into a sequence of simple behaviors such that the one short time-scale subtask can be accomplished by one learning module.
- Abstraction of state and/or action spaces for scaling up: the original "mixture of experts" consists of experts and and gate for expert selection. Therefore, no more abstraction beyond the gating module. In order to cope with complicated real robot tasks, abstraction of the state and/or action spaces is necessary.

Connell and Mahadevan (Connell & Mahadevan 1993) decomposed the whole behavior into sub-behaviors each of which can be independently learned. Morimoto and Doya (Morimoto & Doya 1998) applied a hierarchical reinforcement learning method by which an appropriate sequence of subgoals for the task is learned in the upper level while behaviors to achieve the subgoals are acquired in the lower level. Hasegawa and Fukuda (Hasegawa & Fukuda 1999; Hasegawa, Tanahashi, & Fukuda 2001) proposed a hierarchical behavior controller, which consists of three types of modules, behavior coordinator, behavior controller and feedback controller, and applied it to a brachiation robot. Kleiner et al. (Kleiner, Dietl, & Nebel 2002) proposed a hierarchical learning system in which the modules at lower layer acquires low level skills and the module at higher layer coordinates them. However, in these proposed methods, the task decomposition has been done by the designers very carefully in advance, or the constructions of the state/action spaces for higher layer modules are independent from the learned behaviors of lower modules. As a result, it seems difficult to abstract situations and behaviors based on the already acquired learning/control modules.

A basic idea to cope with the above two issues is that any learning module has limited resource constraint, and this constraint of the learning capability leads us to introduce a multi-module and multi-layered learning system; one learning module has a compact state-action space and acquires a simple map from the states to the actions, and a gating system enables the robot to select one of the behavior modules depending on the situation. More generally, the higher mod-

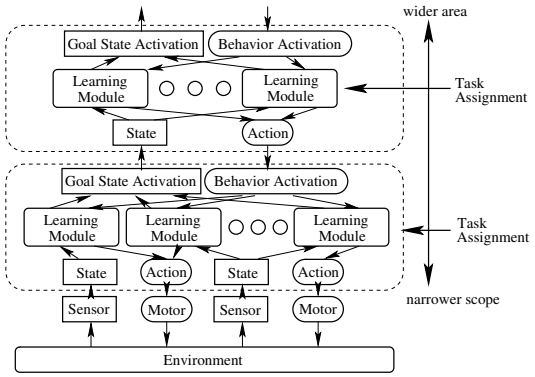
ule controls the lower modules depending on the situation. The definition of this situation depends on the capability of the lower modules because the gating module selects one of the lower modules based on their acquired behaviors. From the another viewpoint, the lower modules provide not only the rational behaviors but also the abstracted situations for the higher module; how feasible the module is, how close to its subgoal, and so on. It is reasonable to utilize such information in order to construct state/action spaces of higher modules from already abstracted situations and behaviors of lower ones. Thus, the hierarchical structure can be constructed with not only experts and gating module but more layers with multiple homogeneous learning modules.

In this paper, we show a series of studies towards the construction of such hierarchical learning structure developmentally. The first one (Takahashi & Asada 2000) is automatic construction of hierarchical structure with purely homogeneous learning modules. Since the resource (and therefore the capability, too) of one learning module is limited, the initially given task is automatically decomposed into a set of small subtasks each of which corresponds to one of the small learning modules, and also the upper layer is recursively generated to cover the whole task. In this case, the all learning modules in the one layer share the same state and action spaces although some modules need the part of them. Then, the second work (Takahashi & Asada 2001) and third one (Takahashi & Asada 2003) focused on the state and action space decomposition according to the subtasks to make the learning much more efficient. Further, the forth one (Takahashi, Hikita, & Asada 2003) realized unsupervised decomposition of a long time-scale task by finding the compact state spaces, which consequently leads the subtask decomposition. We have applied these methods to simple soccer situations in the context of RoboCup (Asada *et al.* 1999) with real robots, and show the experimental results.

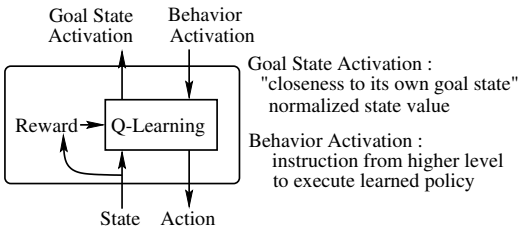
Multi-Layered Learning System

The architecture of the multi-layered reinforcement learning system is shown in Figure 1, in which (a) and (b) indicate a hierarchical architecture with two levels and an individual learning module embedded in the layers, respectively. Each module has its own goal state in its state space, and it learns the behavior to reach the goal, or maximize the sum of the discounted reward received over time based on the Q-learning method. The state and the action are constructed using sensory information and motor command, respectively at the bottom level. The input and output to/from the higher level are the goal state activation and the behavior activation, respectively, as shown in Figure 1(b). The goal state activation g is a normalized state value [1], and $g = 1$ when the situation is the goal state. When the module receives the behavior activation from the higher modules, it calculates the optimal policy for its own goal, and sends action commands to the lower module. The action command at the bottom level is translated to an actual motor command, then

[1]The state value function estimates the sum of the discounted reward received over time when the robot takes the optimal policy, and is obtained by Q learning.



(a) A whole system



(b) A behavior learning module

Figure 1: A hierarchical learning architecture

the robot takes the action in the environment.

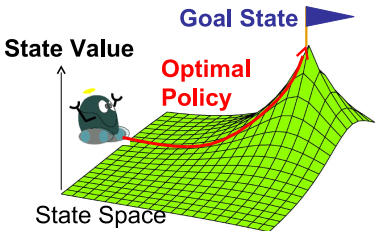


Figure 2: A sketch of a state value function

One basic idea is to use the goal state activations g of the lower modules as the representation of the situation for the higher modules. Figure 2 shows a sketch of a state value function where a robot receives a positive reward one when it reach to a specified goal. The state value function can be regarded as closeness to the goal of the module. The states of the higher modules are constructed using the patterns of the goal state activations of the lower modules. In contrast, the actions of the higher level modules are constructed using the behavior activations to the lower modules.

Behavior Acquisition on Multi-Layered System (Takahashi & Asada 2000)

An Experiment System

Figure 3 shows a picture of the environment where a mobile robot we designed and built, a ball, and a goal are in-

Figure 3: A mobile robot, a ball and a goal

cluded. It has two TV cameras: one has a wide-angle lens, and the other a omni-directional mirror. The driving mechanism is PWS (Powered Wheels Steering) system, and the action space is constructed in terms of two torque values to be sent to two motors that drive two wheels.

Architecture and Results

In this experiment, the robot receives the information of only one goal for simplicity. The state space at the bottom layer is constructed in terms of the centroids of goal region on the images of the two cameras and is tessellated both into 9 by 9 grids each. The action space is constructed in terms of two torque values to be sent to two motors corresponding to two wheels and is tessellated into 3 by 3 grids. Consequently, the numbers of states and actions are $162(9\times9\times2)$ and $9(3\times3)$, respectively. The state and action spaces at the upper layer are constructed by the learning modules at the lower layer which are automatically assigned.

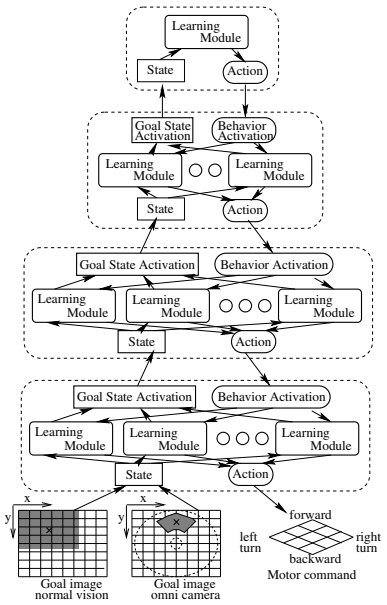


Figure 4: A hierarchical architecture on a monolithic state space

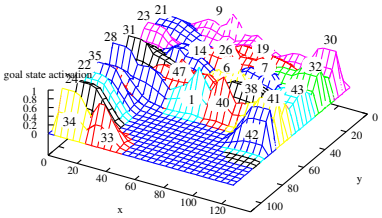


Figure 5: The distribution of learning modules at bottom layer on the normal camera image

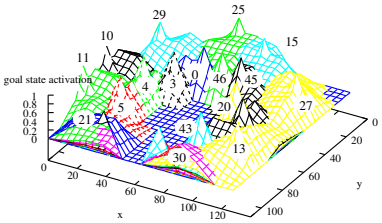


Figure 6: The distribution of learning modules at bottom layer on the omni-directional camera image

The experiment is constructed with two stages: the learning stage and the task execution one based on the learned result. First of all, the robot moves at random in the environment for about two hours. The system learns and constructs the four layers and one learning module is assigned at the top layer (Figure 4). We call each layer from the bottom, "bottom", "middle", "upper", and "top" layers. In this experiment, the system assigned 40 learning modules at the bottom layer, 15 modules at the middle layer, and 4 modules at the upper layer. Figures 5 and 6 show the distributions of goal state activations of learning modules at the bottom layer in the state spaces of wide-angle camera image and omni-directional mirror image, respectively. The x and y axes indicate the centroid of goal region on the images. The numbers in the figures indicate the corresponding learning module numbers. The figures show that each learning module is automatically assigned on the state space uniformly.

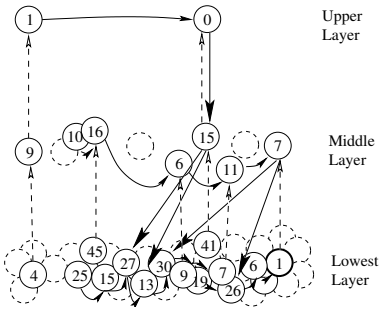


Figure 7: A rough sketch of the state transition on the multi-layer learning system

Figure 7 shows a rough sketch of the state transition and

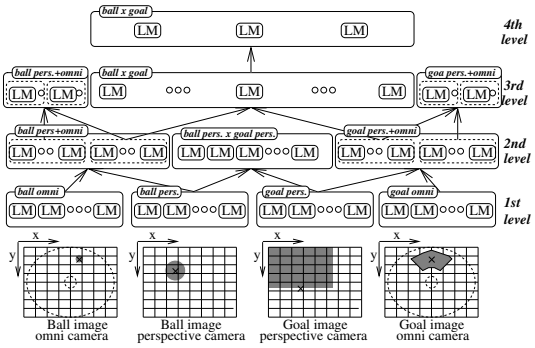


Figure 8: A hierarchy architecture on decomposed state spaces

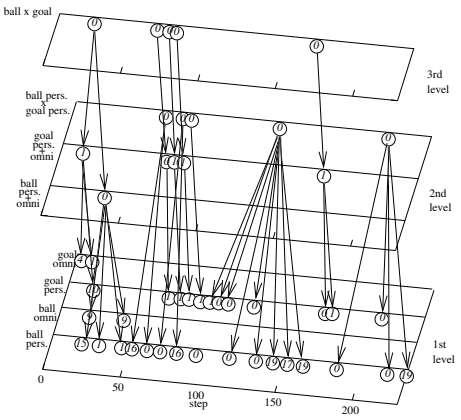


Figure 9: A sequence of the behavior activation of learning modules and the commands to the lower layer modules

the commands to the lower layer on the multi-layer learning system during navigation task. The robot was initially located far from the goal, and faced the opposite direction to it. The target position was just in front of the goal. The circles in the figure indicate the learning modules and their numbers. The empty up arrows (broken lines) indicate that the upper learning module recognizes the state which corresponds to the lower module as the goal state. The small solid arrows indicate the state transition while the robot accomplished the task. The large down arrows indicate that the upper learning module sends the behavior activation to the lower learning module.

State Space Decomposition and Integration (Takahashi & Asada 2001)

The system mentioned in the previous section dealt with a whole state space from the lower layer to the higher one. Therefore, it cannot handle the change of the state variables because the system supposes that all tasks can be defined on the state space at the bottom level. Further, it is easily caught by a curse of dimension if number of the state variables is large.

Here, we introduce an idea that the system constructs a whole state space with several decomposed state spaces. At the bottom level, there are several decomposed state spaces in which modules are assigned to acquire the low level behavior in the small state spaces. The modules at the higher level manage the lower modules assigned to different state spaces. In this paper, we define the term "layer" as a group of modules sharing the same state space, and the term "level" as a class in the hierarchical structure. There might be several layers at one level (see Figure 8).

Figure 8 shows an example hierarchical structure. At the lowest level, there are four learning layers, and each of them deals with its own logical sensory space (ball positions on the perspective camera image and omni one, and goal position on both images). At the second level, there are four learning layers. The "*ball pers.*$\times$*goal pers.*" layer deals with lower modules of "*ball pers.*" and "*goal pers.*" layers. The arrows in the figure indicate the flows from the goal state activations to the state vectors. The arrows from the action vectors to behavior activations are eliminated. At the third level, the system has three learning layers, again.

After the learning stage, we let our robot do a couple of tasks, for example, chasing a ball, moving in front of the goal, and shooting a ball into the goal, using this multi-layer learning structure. When the robot behaves chasing a ball, the system uses "*ball pers.*" and "*ball omni*" layers at 1st level, "*ball pers.*+*omni*" at 2nd level, and "*ball pers.*+*omni*" at 3nd level. When the robot behaves moving in front of the goal, the system uses "*goal pers.*" and "*goal omni*" layers at 1st level, "*goal pers.*+*omni*" at 2nd level, and "*goal pers.*+*omni*" at 3nd level. And when the robot shoots a ball into a goal, the system uses all 4 layers at the 1st level, all 3 layers at 2nd level, "*ball x goal*" layer at the 3rd level, and the layer at the 4th level. All layers at the 1st level and "*ball pers.*+*omni*" and "*goal pers.*+*omni*" layers are reused among the three behaviors. In the case of the shooting behavior, the target situation is given by reading the sensor information when the robot pushes the ball into the goal; the robot captures the ball and goal at center bottom in the perspective camera image. As an initial position, the robot is located far from the goal, faced opposite direction to it. The ball was located between the robot and the goal.

Figure 9 shows a sequence of the behavior activation of learning modules and the commands to the lower layer modules. The down arrows indicate that the higher learning modules fire the behavior activations of the lower learning modules.

Action Space Decomposition and Coordination (Takahashi & Asada 2003)

Figure 10 shows a picture and a top view of another soccer robot for middle size league of RoboCup we designed and built. The driving mechanism is same as the last one, and it equips a pinball like kicking device in front of the body. If one learning module has to manipulate all actuators simultaneously, the exploration space of action scales up exponentially with the number of the actuators, and it is impractical to apply a reinforcement learning system.

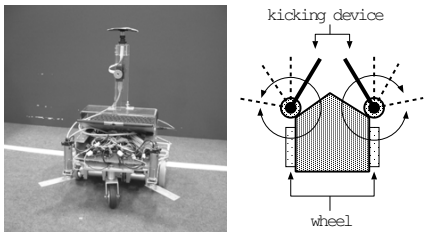


Figure 10: A Robot : it has a PWS system vehicle, pinball like kicking devices, and a small camera with a omnidirectional mirror

Fortunately, a complicated behavior which needs many kinds of actuators might be often generally decomposed into some simple behaviors each of which needs small number of actuators. The basic idea of this decomposition is that we can classify them based on aspects of the actuators. For example, we may classify the actuators into navigation devices and manipulators, then the some of behaviors depend on the navigation devices tightly, not on the manipulators, while the others depend on manipulators, not on the navigation. The action space based on only navigation devices seems to be enough for acquisition of the former behaviors, while the action space based on manipulator would be sufficient for the manipulation tasks. If we can assign learning modules to both action spaces and integrate them at higher layer, much smaller computational resources is needed and the learning can be accelerated significantly.

Architecture and Results

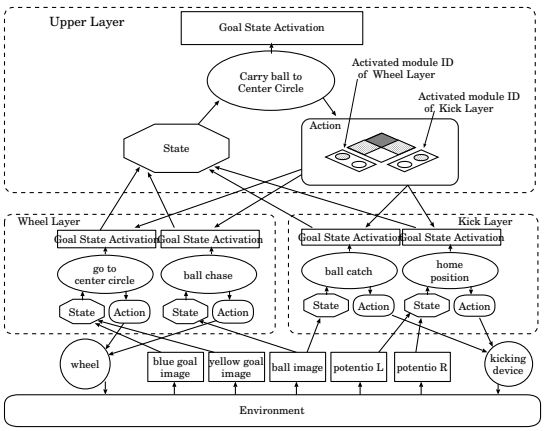


Figure 11: A hierarchical learning system for the behavior of placing the ball in the center circle (task1)

We have implemented two kinds of hierarchical systems to check if the basic idea can be realized. Each system has been assigned a task (Figures 11 and 12). One is placing the ball in the center circle (task 1), and the other is shooting the ball into the goal (task2).

We have prepared the following subtasks for the vehicle: “Chasing a ball”, “Looking at the goal in front”, “Reaching the center circle”, and “Reaching the goal”. We have also prepared the following subtasks for the kicking device: “Catching the ball”, “Kicking the ball”, and “Setting the kicking device to the home position”. Then, the upper layer modules integrates these lower ones. After the learner acquired low level behaviors, it puts new learning modules at higher layer as shown in Figures 11 and 12 and learn two kinds of behaviors. We let our robot learn the behavior for the task 1 (placing a blal in the center circle) first. The robot acquired “Chasing a ball” and “Reaching the center circle” behaviors for the vehicle, and “Catching the ball” and “Setting the kicking device to the home position” behaviors for the kicking device. Then the robot learned the behavior for the task 2 (shooting the ball into the goal). It reused the behaviors of “Chasing a ball”, “Catching the ball”, and “Setting the kicking device to the home position”, and learned the other new behaviors. Figure 13 shows a sequence of shooting a ball into the goal with the hierarchical learning system (see also Figure 12).

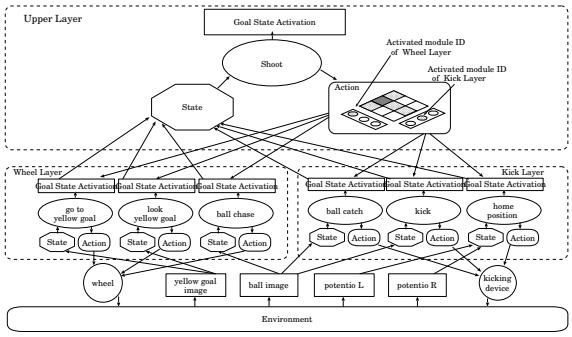


Figure 12: A hierarchical learning system for the behavior of shooting the ball into the goal (task 2)

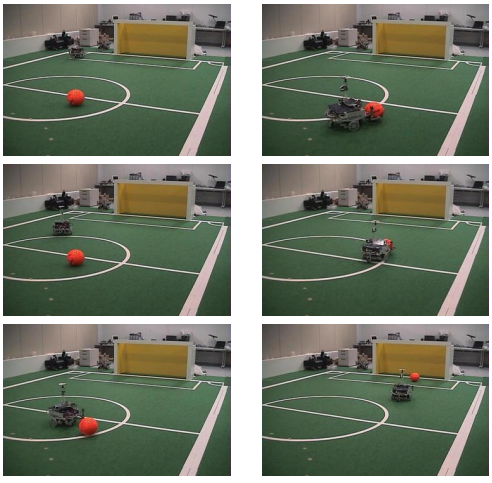


Figure 13: A sequence of an acquired behavior (Shooting)

Task Decomposition based on Self-Interpretation of Instructions by Coach (Takahashi, Hikita, & Asada 2003)

When we develop a real robot which learns various behaviors in its life, it seems reasonable that a human instructs or shows some example behaviors to the robot in order to accelerate the learning before it starts to learn. We proposed a behavior acquisition method based on hierarchical multi-module leaning system with self-interpretation of coach instructions. The proposed method enables a robot to

1. decompose a long term task into a set of short term subtasks,
2. select sensory information needed to accomplish the current subtask,
3. acquire a basic behavior to each subtask, and
4. integrate the learned behaviors to a sequence of the behaviors to accomplish the given long term task.

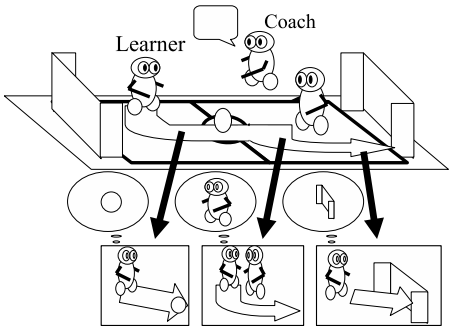
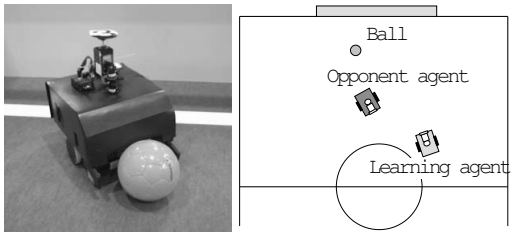


Figure 14: Basic concept: A coach gives instructions to a learner. The learner follows the instruction and find basic behaviors by itself.

Figure 14 shows a rough sketch of the basic idea. There are a learner, an opponent, and a coach in a simple soccer situation. The coach has *a priori* knowledge of tasks to be played by the learner. The learner does not have any knowledge on tasks but just follows the instructions. In Figure 14, the coach shows a instruction of shooting a ball into a goal without collision to an opponent. After some instructions, the learner segments the whole task into a sequence of subtasks, acquires a behavior for each subtask, finds the purpose of the instructed task, and acquire a sequence of the behaviors to accomplish the task by itself. When the coach gives new instructions, the learner reuses the learning modules for familiar subtasks, generates new learning modules for unfamiliar subtasks at lower level. The system generates a new module for a sequence of behaviors of the whole instructed task at the upper level. The details are described in (Takahashi, Hikita, & Asada 2003).

Experiments

Figure 15 (a) shows the mobile robot. The robot has an omni-directional camera system. A simple color image processing is applied to detect the ball area and an opponent one



(a) A real robot and a ball (b) Top view of the field

Figure 15: Real robot and environment

in the image in real-time (every 33ms). Figure 15 (b) shows a situation with which the learning agent can encounter.

The robot receives instructions for the tasks in the order as follows:

Task 1: chasing a ball

Task 2: shooting a ball into a goal without obstacles

Task 3: shooting a ball into a goal with an obstacle

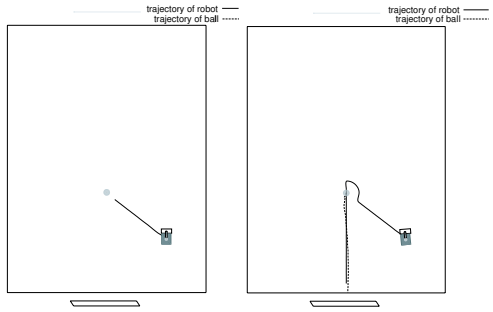
Figures.16 (a), (b), and (c) show the one of the example behaviors for each task. Figures17 show the constructed systems after the learning of each task. First of all, the coach gives some instructions for the ball chasing task. The system produce one module which acquired the behavior of ball chasing. At the second stage, the coach gives some instructions for the shooting task. The learner produces another module which has a policy of going around the ball until the directions to the ball and the goal become same. At the last stage, the coach gives some instructions for the shooting task with obstacle avoidance. The learner produces another module which acquired the behavior of going to the intersection between the opponent and the goal avoiding the collision. Figure18 shows a sequence of an experiment of real robots for the task.

Conclusions and Future Works

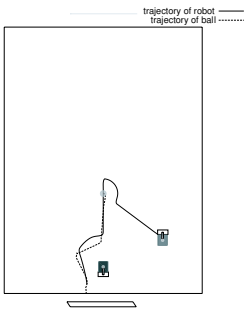
We showed a series of approaches to the problem of decomposing the large state action space at the bottom level into several subspaces and merging those subspaces at the higher level. As future works, there are a number of issues to extend our current methods.

Interference between modules One module behavior might have inference to another one which has different actuators. For example, the action of a navigation module will disturb the state transition from the view point of the kicking device module; the catching behavior will be success if the vehicle stays while it will fail if the vehicle moves.

Self-assignment of modules It is still an important issue to find a purposive behavior for each learning module automatically. In the paper (Takahashi & Asada 2000), the system distributes modules on the state space uniformly,



(a) Task 1 (b) Task 2



(c) Task 3

Figure 16: Example behaviors for tasks

however, it is not so efficient. In the paper (Takahashi, Hikita, & Asada 2003), the system decomposes the task by itself, however, the method uses many heuristics and needs instruction from a coach. In many cases, the designers have to define the goal of each module by hand based on their own experiences and insights.

Self-construction of hierarchy Another missing point in the current method is that it does not have the mechanism that constructs the learning layer by itself.

Acknowledgments

We would like to thank Motohiro Yuba and Kouichi Hikita for their efforts of real robot experiments.

References

Asada, M.; Kitano, H.; Noda, I.; and Veloso, M. 1999. Robocup: Today and tomorrow – what we have learned. *Artificial Intelligence* 193–214.

Connell, J. H., and Mahadevan, S. 1993. *ROBOT LEARNING*. Kluwer Academic Publishers. chapter RAPID TASK LEARNING FOR REAL ROBOTS.

Hasegawa, Y., and Fukuda, T. 1999. Learning method for hierarchical behavior controller. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, 2799–2804.

Hasegawa, Y.; Tanahashi, H.; and Fukuda, T. 2001. Behavior coordination of brachation robot based on bahavior phase shift. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume CD-ROM, 526–531.

Jacobs, R.; Jordan, M.; S, N.; and Hinton, G. 1991. Adaptive mixture of local expoerts. *Neural Computation* 3:79–87.

Kleiner, A.; Dietl, M.; and Nebel, B. 2002. Towards a life-long learning soccer agent. In Kaminka, G. A.; Lima, P. U.; and Rojas, R., eds., *The 2002 International RoboCup Symposium Pre-Proceedings*, CD–ROM.

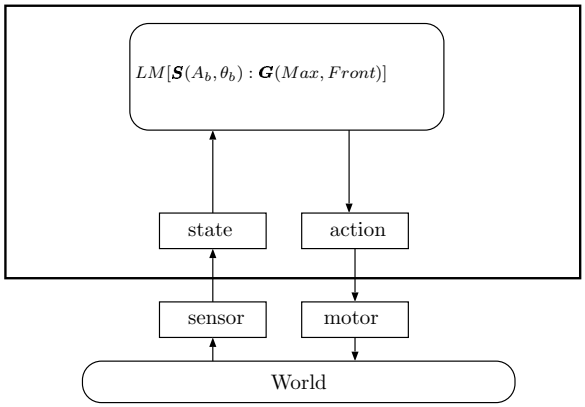
Morimoto, J., and Doya, K. 1998. Hierarchical reinforcement learning of low-dimensional subgoals and high-dimensional trajectories. In *The 5th International Conference on Neural Information Processing*, volume 2, 850–853.

Takahashi, Y., and Asada, M. 2000. Vision-guided behavior acquisition of a mobile robot by multi-layered reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, 395–402.

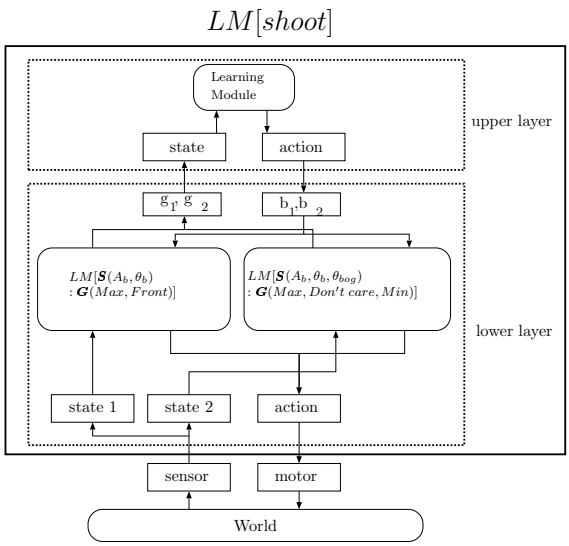
Takahashi, Y., and Asada, M. 2001. Multi-controller fusion in multi-layered reinforcement learning. In *International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI2001)*, 7–12.

Takahashi, Y., and Asada, M. 2003. Multi-layered learning systems for vision-based behavior acquisition of a real mobile robot. In *Proceedings of SICE Annual Conference 2003 in Fukui*, volume CD-ROM, 2937–2942.

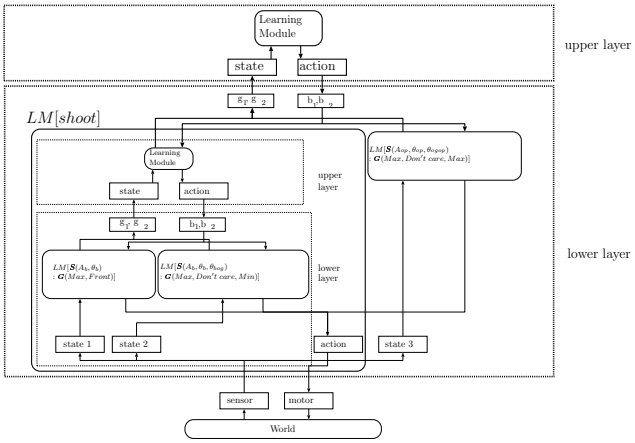
Takahashi, Y.; Hikita, K.; and Asada, M. 2003. Incremental purposive behavior acquisition based on self-interpretation of instructions by coach. In *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, CD–ROM.



(a) Task 1



(b) Task 2



(c) Task 3

Figure 17: The acquired hierarchical structure

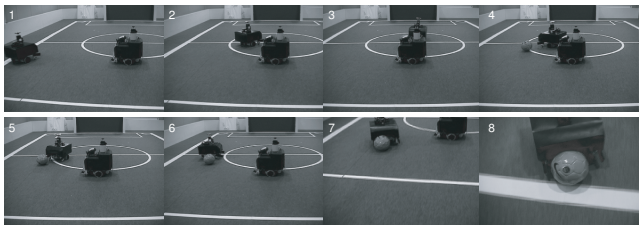


Figure 18: A sequence of an experiment of real robots : shooting a ball into a goal with an obstacle (task3)