# Multi-Layered Learning Systems for Vision-based Behavior Acquisition of A Real Mobile Robot

Yasutake Takahashi[1,2], Minoru Asada[1,2]

1 Dept. of Adaptive Machine Systems,
2 Handai Frontier Research Center,
Graduate School of Engineering, Osaka University Yamadaoka 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 decomposing the large state/action space at the bottom level into several subspaces and merging those subspaces at the higher level. This allows the system to maintain computational resources assigned to the modules compact and small, to reuse the policies learned before, and therefore to avoid the curse of dimension. To show the validity of the proposed methods, we apply them to a simple soccer situation in the context of RoboCup, and show the experimental results.

**Keywords:** reinforcement learning, multi-layered learning system, hierarchical control system

## 1. Introduction

Reinforcement learning (hereafter, RL) is an attractive method for robot behavior acquisition with little or no *a priori* knowledge and higher capability of reactive and adaptive behaviors [1)]. However, single and straightforward application of RL methods to real robot tasks is considerably difficult due to its almost endless explanation which is easily scaled up exponentially with the size of the state/action spaces, that seems almost impossible from a practical viewpoint.

One approach to the problem is to adopt a hierarchical structure within leaning control system. That is, the system

1. prepares learning/control modules of one kind each of which deals with a subspace divided from a whole state/action space,
2. abstracts situations and behaviors based on the acquired learning/control modules, and
3. acquires higher level, new behaviors based on the state and action spaces constructed from already abstracted situations and behaviors.

This approach can suppress the explosion of the state and action spaces since the higher level learning/control system manages adequately small size spaces which are abstracted in the lower levels.

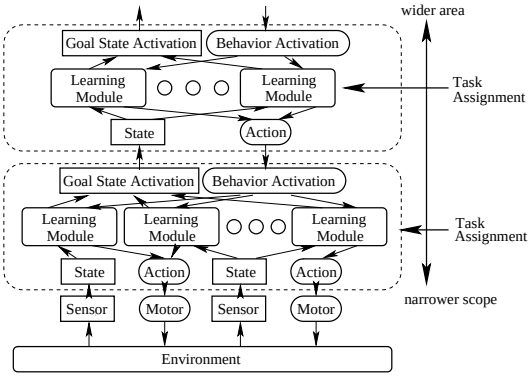
Fortunately, a long time-scale behavior might be often decomposed into a sequence of simple behaviors in general, and therefore, the search space is expected to be able to be divided into some smaller ones. Connell and Mahadevan [2)] decomposed the whole behavior into sub-behaviors each of which can be independently learned. Morimoto and Doya [5)] applied a hierarchical RL 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. Kleiner et al. [6)] proposed a hierarchical learning system in which the modules at lower layer acquires low level skills and the module at higher layer coordinates them. Hasegawa and Fukuda [3, 4)] 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.

However, in these proposed methods, 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 acquired learning/control modules. The learned modules of lower layer provide not only adaptive and reasonable behaviors but also the closeness to the goal states of the modules and feasibility of their behaviors. It is reasonable to utilize those information in order to construct state/action spaces of higher modules from already abstracted situations and behaviors of lower ones.

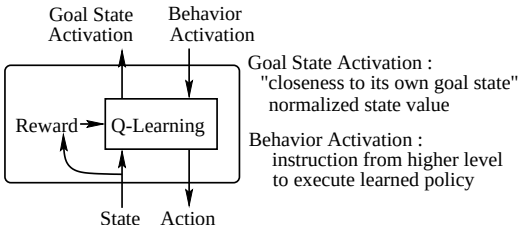
In this paper, we show a series of the proposed multi-layered learning control system which prepares learning/control modules of one kind, assigns them to subspaces divided from a whole state/action space, abstracts situations and behaviors, and acquires higher level behaviors. To show the validity of the proposed method, we apply it to a simple soccer situation in the context of RoboCup, and show the experimental results.

## 2. 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. 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




(a) A whole system




(b) A behavior learning module

Figure 1: A hierarchical learning architecture

over time, using $Q$-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 $b$ 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 the robot takes the action in the environment.

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. 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.

# 3. Behavior Acquisition on Multi-Layered System [7)]

## 3.1 Experiment System

Figure 2 shows a picture of a mobile robot we designed and built, a ball, and a goal. 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. These parameters of the system are unknown to the robot, and it tries to estimate the mapping from the sensory information to the appropriate motor commands by the method. The environment consists of the ball, the goal, and the mobile robot.

[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.

Figure 2: A mobile robot, a ball and a goal

## 3.2 Architecture

In this experiment, the robot receives the information of only one goal, for the simplicity. The state space at the bottom layer is constructed in terms of the centroids of goal 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 $162(9 \times 9 \times 2)$ and $9(3 \times 3)$, respectively. The state and action at the upper layer is constructed by the learning modules at the lower layer which are automatically assigned.

The experiment is constructed with two stages, one is the learning one and other is the task execution one using the learned result. First of all, the robot moved at random in the environment for about two hours. The system learned and constructed the four layers and one learning module is assigned at the top layer (Figure 3). 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 4 and 5 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 images. The numbers on the figures indicate the learning module numbers. The figures show that each learning module is assigned on the state space uniformly.

Figure 6 shows a rough sketch of the state transition and 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 opposite

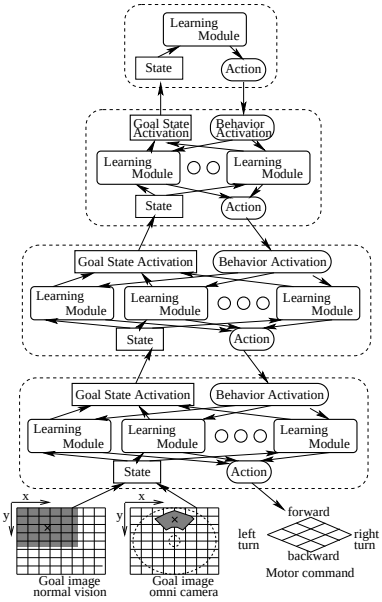



Figure 3: A hierarchical architecture of learning modules

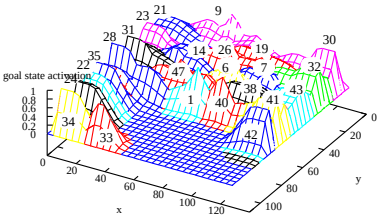



Figure 4: The distribution of learning modules at bottom layer on the normal camera image

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 set the behavior activation of the lower learning module.

## 4. State Space Integration [8)]

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 suppose that all tasks can be defined on the state space at the bottom level. And also, 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

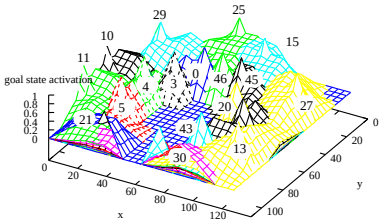



Figure 5: The distribution of learning modules at bottom layer on the omni-directional camera image

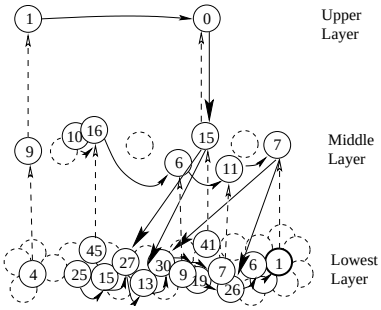



Figure 6: A rough sketch of the state transition on the multi-layer learning system

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 7).

When the higher layer constructs its state-action space based on situations and behaviors acquired by the modules of several lower layers, it should consider that the layers are independent from each other, or there is dependence between them. The layer might be basically independent from each other when the each layer's modules recognize different objects and learn behaviors for them. For example, in the case of robot in the RoboCup field, one layer's modules could be the experts of ball handling and the other layer's modules the one of navigation on the field. In such a case, the state space is constructed as direct product of module's activations of lower layers. We call this way of state space construction "a multiplicative approach".

On the other hand, there might be dependence between the layers when modules on both layers recognize the same object in the environment with different logical sensor outputs. For example, our robot recognizes an object with both perspective vision system and omni-directional one. In such a case, the system can recognize the situation complementary using plural layers' outputs even if one layer loses the object on its own state spaces. We call this way of state space construction "a complementary approach".

Figure 7 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 po-

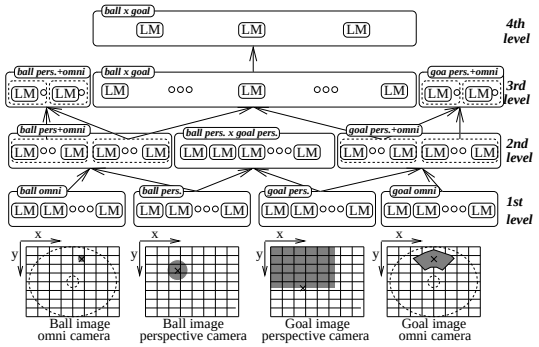



Figure 7: A hierarchy architecture of learning modules

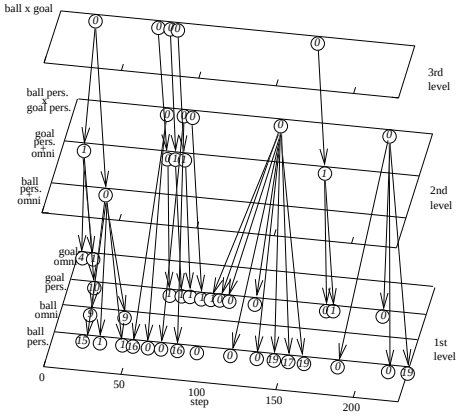



Figure 8: A sequence of the behavior activation of learning modules and the commands to the lower layer modules

sitions on the perspective camera image and omni one, and goal position on both images). At the second level, there are three learning layers in which one adopts the multiplicative approach and the others adopt the complementary approach. The multiplicative approach of the "*ball pers.*×*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 in which one adopts the multiplicative approach and the others adopt the complementary approach, again. At the levels higher than third layer, the learning layer is constructed as described in the previous section.

After the learning stage, we let our robot do a couple of tasks. One of them is shooting a ball into the goal using this multi-layer learning structure. 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 8 shows the 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

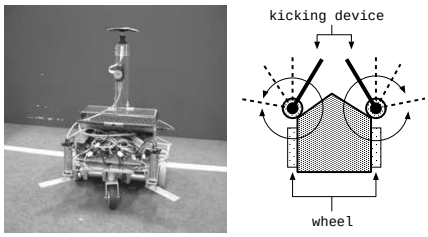



Figure 9: A Robot : it has a PWS system vehicle, pinball like kicking devices, and a small camera with a omni-directional mirror

the lower learning modules.

# 5. Behavior Segmentation and Coordination

Figure 9 shows a picture and a top view of a soccer robot for middle size league of RoboCup we designed and built, recently. The driving mechanism is PWS, and it equips a pinball like kicking device in front of the body. These days, many robots have number of actuators such as navigation devices and object manipulators, and have a capability of execution of many kinds of task by coordinating these actuators. 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.

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 time can be reduced significantly.

## 5.1 Hierarchical Learning System

**whole system** We have implemented two kind of hierarchical system to check the basic idea. Each system has been assigned a task (Figures 10 and 11). One is placing the ball in the center circle (task 1), and the other is shooting the ball into the goal (task2).

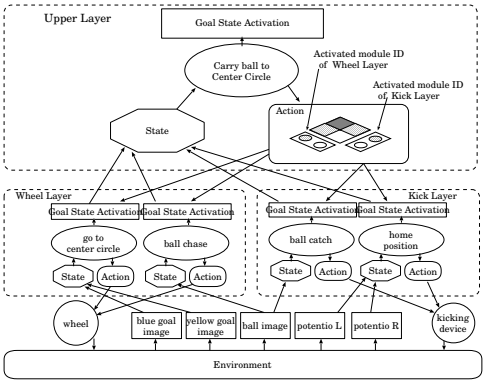



Figure 10: A hierarchical learning system for task 1

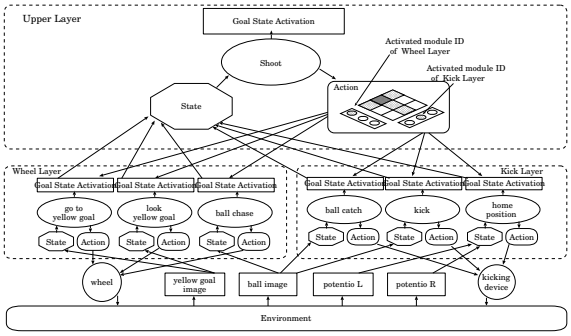



Figure 11: A hierarchical learning system for task 2

**low level behavior acquisition** We have prepared the following subtasks for the vehicle: "Chasing a ball", "Looking the goal in front of the body", "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.

**higher level behavior acquisition** After the learner acquired low level behaviors, it puts new learning modules at higher layer as shown in Figures 10 and 11 and learn two kinds of behaviors.

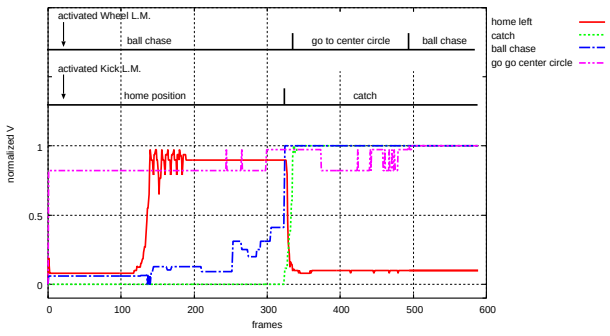



Figure 12: A sequence of the goal state activations and behavior commands (Task 1)

Figure 12 shows the sequence of the goal state activations of lower modules and behavior commands to the lower ones. At the start of this behavior, the robot activates the module of setting home position behavior for the kicking device and ball chasing module for the vehicle at lower layer. The robot reaches the ball, then it activates the module of catching the ball for kicking device and the module of reaching the center circle. Then, it achieves the task of placing a ball to the center circle.

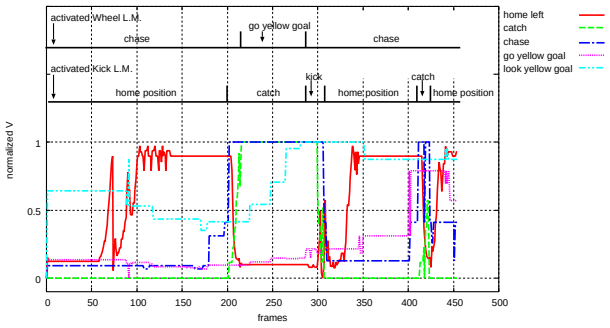



Figure 13: A sequence of the goal state activation and behavior activation (Task 2)

Figure 13 shows the sequence of the goal state activations of lower modules and behavior commands to the lower ones. At the start of this behavior, the robot activates the module of setting home position behavior for the kicking device and ball chasing module for the vehicle at lower layer (from ① to ⑧ in Figure 14). The robot reaches the ball, then it activates the module of catching the ball for kicking device and the module of reaching the goal (from ⑨ to ⑫). When the robot captures the goal in front of the body and get near to the goal, it activates the module of kicking the ball, then successfully shoots the ball into the goal (from ⑬ to ⑮).

# 6. 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 module 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 be failed if the vehicle moves.

**Self-assignment of modules** It is still a important issue to find a purposive behavior for each learning module automatically. In the paper[7], the system distributes modules on the state space uniformly, however, it is not so efficient. 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

This research was supported by the Japan Science and Technology Corporation, in Research for the the Core Research for the Evolutional Science and Technology Program (CREST) titled Robot Brain Project in the research area "Creating a brain". We would like to thank Motohiro Yuba for his efforts of real robot experiments.

## References

[1] Jonalthan H. Connell and Sridhar Mahadevan. *ROBOT LEARNING*. Kluwer Academic Publishers, 1993.

[2] Jonalthan H. Connell and Sridhar Mahadevan. *ROBOT LEARNING*, chapter RAPID TASK LEARNING FOR REAL ROBOTS. Kluwer Academic Publishers, 1993.

[3] Yasuhisa Hasegawa and Toshio Fukuda. Learning method for hierarchical behavior controller. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pages 2799–2804, 1999.

[4] Yasuhisa HASEGAWA, Hiroaki TANAHASHI, and Toshio FUKUDA. 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, pages 526–531, 2001.

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

[6] Alexander Kleiner, Markus Dietl, and Bernhard Nebel. Towards a life-long learning soccer agent. In Gal A. Kaminka, Pedro U. Lima, and Raul Rojas, editors, *The 2002 International RoboCup Symposium Pre-Proceedings*, pages CD–ROM, June 2002.

[7] Y. Takahashi and M. Asada. 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, pages 395–402, 2000.

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

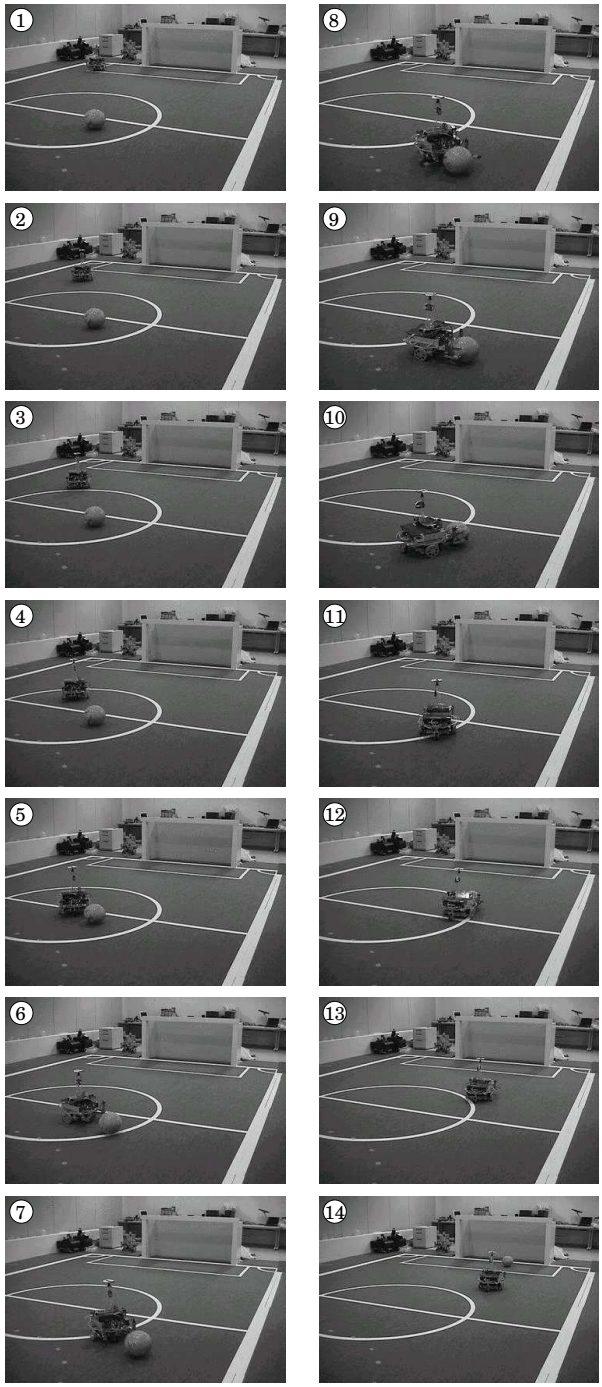



Figure 14: A sequence of an acquired behavior (Shooting)