1

Learning for Cooperative Transportation by Autonomous Humanoid Robots

Yutaka Inoue, Takahiro Tohge, and Hitoshi Iba

Department of Frontier Informatics, Graduate School of Frontier Sciences,
The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan,
inoue@iba.k.u-tokyo.ac.jp, http://www.iba.k.u-tokyo.ac.jp

In this chapter, we describe a cooperative transportation to a target position with two humanoid robots and introduce a machine learning approach to solving the problem. The difficulty of the task lies on the fact that each position shifts with the other's while they are moving. Therefore, it is necessary to correct the position in a real-time manner. However, it is difficult to generate such an action in consideration of the physical formula. We empirically show how successful the humanoid robot HOAP-1's cooperate with each other for the sake of the transportation as a result of Q-learning. Furthermore, we show a result of the experiment that transports an object cooperatively to a target position using those robots.

1.1 Introduction

In this chapter, we first clarify the practical difficulties we face from the cooperative transportation task with two bodies of humanoid robots. Afterwards, we propose a solution to these difficulties and empirically show the effectiveness both by simulation and by real robots.

In recent years, many researches have been conducted upon various aspects of humanoid robots [1] [2]. Since humanoid robots have physical features similar to us, it is very important to let them behave intelligently like humans. In addition, from the viewpoint of AI or DAI (Distributed AI), it is rewarding to study how cooperatively humanoid robots perform a task just as we humans can. However, there have been very few studies on the cooperative behaviors of multiple humanoid robots. Thus, in this chapter, we describe the emergence of the cooperation between humanoid robots so as to achieve the same goal. The target task we have chosen is a cooperative transportation, in which two bodies of humanoids have to cooperate with each other to carry and transport an object to a certain goal position.

As for the transportation task, several researches have been reported on the cooperation between a human and a wheel robot [3][4] and the cooperation among multiple wheel robots [5][6]. However, in most of these studies, the goal was to let a robot perform a task instead of a human.

Research to realize collaboration with a legged robot includes lifting operations of an object with two robots [7] and box-pushing with two robots [8]. However, few studies have addressed cooperative work using similar legged robots. It is presumed that body swinging during walking renders cooperative work by a legged robot difficult [9]. Therefore, it is more difficult for a humanoid robot to carry out a transportation task, because it is capable of motions that are more complicated and less stable than a usual legged robot.

In leader-follower type control [10][11], which is often used for cooperative movement, it is essential that a follower robot acquire information such as the position and velocity of an object fluctuated by the motion of a leader robot. This information is usually obtained by a force sensor or wireless communication. Such a method is considered to be effective for a robot with a stable center of gravity operating with less information for control. However, much information must be processed simultaneously to allow a humanoid robot to perform complicated actions, such as transporting an object cooperatively, with its difficulty to control caused by its unstable body balance. It would be expensive to build a system that carries out optimal operation using this information.

One hurdle in the case where multiple humanoid robots move carrying an object cooperatively is the disorder of cooperative motion by body swinging during walking. Therefore in this chapter, learning is carried out to acquire behavior to correct a mutual position shift generated by this disorder of motion. For this purpose, we use two kinds of methods: (i) Classifier System [12] and (ii) Q-learning [13]. We will show that behavior to correct a position shift can be acquired based on the simulation results of this study. Moreover, according to this result, the applicability to a real robot is investigated. Furthermore, cooperative transportation to a target position is conducted.

This chapter is organized as follows. The next section explains the clarified problem difficulties with the cooperative transportation. After that, Section 1.3 proposes our method to solve the problem. Section 1.4 presents an experimental result in the simulation and real robots environment. Then Section 1.5 shows an experimental result of cooperative transportation with real robots. Section 1.6 discusses these results and future researches. Finally, a conclusion is given in Section 1.7.

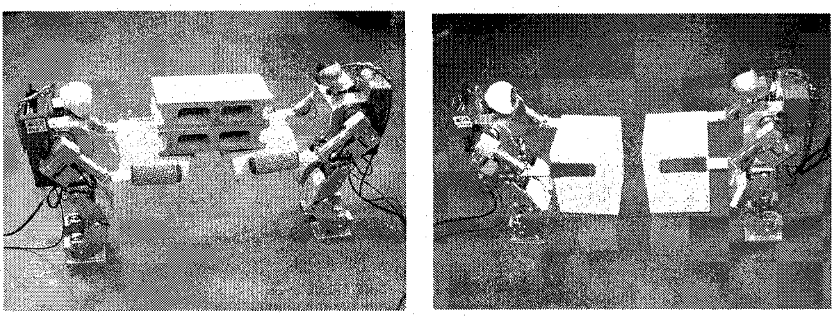
1.2 Problem in cooperative Transportation by humanoid Robots

Cooperative transportation by humanoid robots involves solving many difficult problems. It is different from the transportation by a single robot, in which another robot motion is negligible. On the other hand, in case of the cooperative transportation, one robot's motion has an influence on another robot to some extent. Thus, it is necessary to synchronize both robots' motions. However, the synchronization is not easily achieved because precise motions are not expected by humanoids due to the load weight or the floor friction.

We conducted an experiment assuming tasks to transport a lightweight object all around, aiming to extract specific problems from using two humanoid robots: HOAP-1 (manufactured by Fujitsu Automation Limited). Dimensions of a HOAP-1 are 223 x 139 x 483 mm (width, depth, and height) with a weight of 5.9 kg. It has two arm joints with 4 degrees of freedom each, and two leg joints with 6 degrees of freedom each: 20 degrees of freedom in all for right and left.

Actually, when a package is transferred, it seems to be more practical for two robots to have a single object. However, unless both robots move synchronously in the desirable direction, too much load will be given to the arms of robots, which may often cause the mechanical trouble in the arm and the shoulder. It is assumed in experiment that the arm movement can cancel the position shift, and that the distance and angle that can be cancelled would be in the space between two objects.

We assume the following task situation (see Fig. 1.1a): Each robot raises its platform, on which a brick, i.e., a transportation target, is to be placed. However, as a first step, we have removed the target for the sake of simplicity (Fig. 1.1b). The platform each robot raises is made of foam polystyrene and about 80 gram weigh. The size is about 150 mm wide, 150 mm deep and 200 mm high. This platform is larger than a conventional one because it has to bear the weight of the transportation target. A sponge grip is attached on each robot arm, so that an object would not slip off the arm during the experiment.

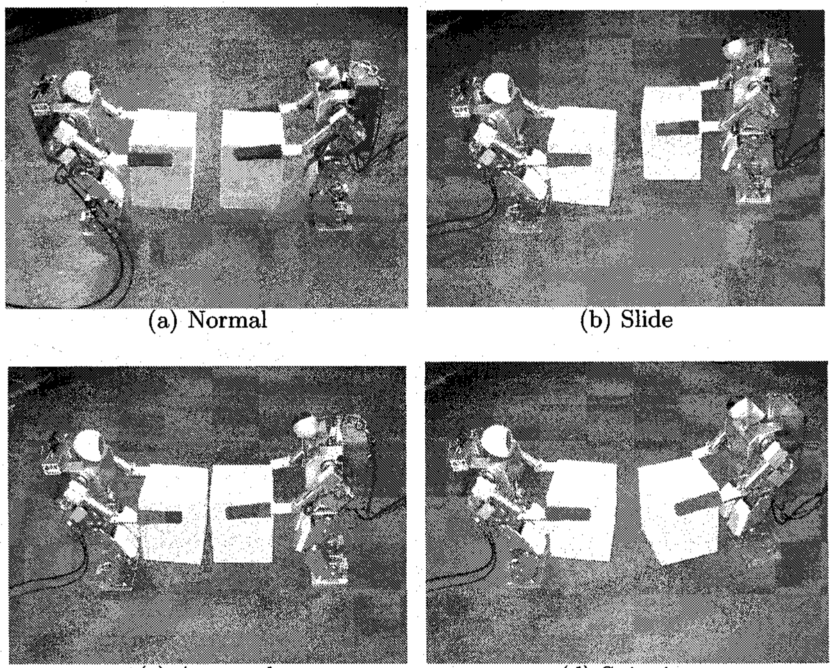


(a) Trunk-based transfer (b) Simplified transportation.

Fig. 1.1. The target of cooperative transportation.

The two robots operate in Master-Slave mode. That is, the Master robot transmits data corresponding to each operation created in advance to the Slave robot; the two robots start a motion in the same direction simultaneously. The created basic motions consist of the following 12 patterns: forward, backward, rightward, leftward, half forward, half backward, half rightward, half leftward, right turn, left turn, pick up, and put down. These basic motions are combined to allow the two robots to transport an object.

The experiment of several times was conducted using each motion. The initial position in this experiment is shown in Fig. 1.2a. The results indicated that unintentional motions such as lateral movement (Fig. 1.2b) and back-and-forth movement (Fig. 1.2c) by sliding from the normal position, and rotation (Fig. 1.2d) occur frequently in basic transportation motions such as forward, backward, rightward, and leftward. This is considered mainly to result from swinging during walking and the weight of the object.



(a) Normal (b) Slide

(c) Approach (d) Spinning

Fig. 1.2. Normal positions and different kinds of positional shifts.

The following three factors can be considered the causes of these shifts in motion.

- Swing when the robot moves

- Shift of the center of gravity by having an object
- Initialization error of robot's joint motors

Especially, in case of humanoid robots, we can think of motor vibration due to the body motion as its cause. This may affect the robot's translation or direction. In addition, the gravity change resultant from carrying an object may possibly cause some errors in the movement.

When activating a robot, it is necessary to set the initial positions of each joint's motors manually. Thus, setting those ini-tial values wrongly may result in fatal errors. In order to investigate the error of initial setting, we performed experiments in the fundamental mode of motions: forward, backward, rightward and leftward. More precisely, a robot was forced to make five steps in each direction so as to measure the final position. In these experiments, the initial setting was used twice in each of test patterns, and the experiments were repeated 10 time, which means that 20 trials were performed in to-tal for each setting. Note that the same robot was used for these experiments.

The moving distance to front and back, right and left of each experiment is shown in Fig. 1.3. The moving distance in two initial setups is expressed by a circle and a triangle. As shown in Fig. 1.3a and Fig. 1.3b, when the robot moves front-ward or backward, the error occurs to the right incline. On the other hand, Fig. 1.3c and Fig. 1.3d show that rightward or leftward movements resulted in the errors in the frontward incline. From the results, it is evident that coincident ini-tial positioning of two robots is very difficult, and error occurs in moving distance or in direction.

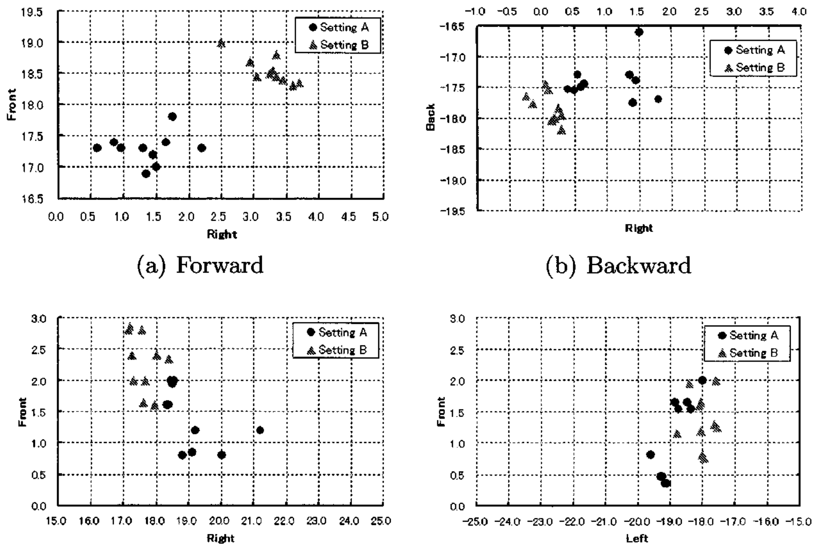
Such a position shift can be cancelled, if only slight, by installing a force sensor on a wrist and moving arms in the load direction. However, a robot's position must be corrected in case of a shift beyond the limitation of an arm. Improper correction may cause failure of an arm or a shoulder joint and breakage of an object.

1.3 Approach of Transportation Control

The practical problem of transporting an object is the possibility that a robot falls during movement, due to loss of body balance in connection with a load on the arm by a mutual position shift after moving. Therefore, it is important to acquire behavior for correcting the position shift generated from movement by learning algorithms.

One of the advantages of using reinforcement learning is its easiness of revising the system due to the change of input-output information and its possibility to select an appropriate action in response to various information.

A situation is assumed in which two robots move face to face while maintaining the distance within a range to transport an object stably. This motion can be divided into two stages: one in which the two robots move simultaneously, and one in which one robot corrects its position. Simultaneous movement of two robots is controlled by wireless communication. A shift over a



(a) Forward (b) Backward

(c) Rightward (d) Leftward

Fig. 1.3. Experimental results of initial setting.

certain limit of distance or angle in this motion will be corrected by one robot according to behavior acquired by learning.

In order to recognize an object or a state, the Master robot is equipped with an active camera, while the Slave robot carries a static one. The active camera works with a pan angle of ± 90[deg]Cand a tilt angle of ± 90[deg]. The robots rotate these cameras and recognize their goal so that they can transport the target object to the goal. The static camera is used to observe the current state of two robots. The obtained information is used as the input to the learning system.

Fig. 1.4 shows the motion overview for conducting a transportation task. In the first stage, the Master robot performs a motion programmed in advance; simultaneously, it issues directions to perform the same motion to the Slave robot. If there is no position shift after movement, the process forwards to the next stage; otherwise, the position is corrected with the learning system. We have tried to realize a cooperative transportation task by repeating the series of this flow.

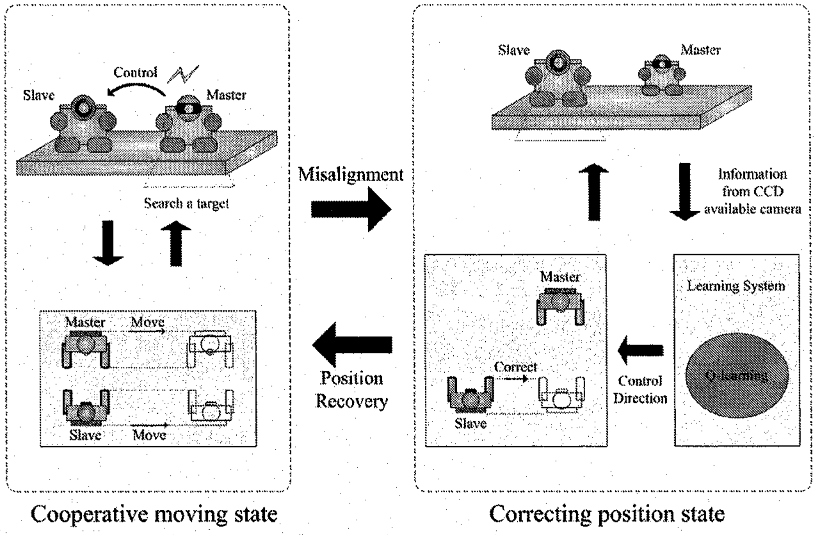


Fig. 1.4. Steps of the cooperative transportation.

1.4 Learning to Correct Positioning

1.4.1 Learning Model

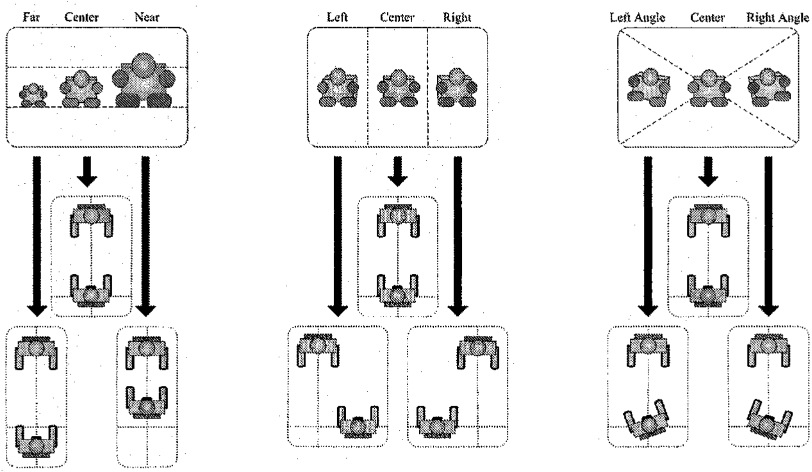
The learning for position correction is carried out with Q-learning and Classifier System.

Q-learning guarantees that the state transition in the environment of a Markov decision process converges into the optimal direction [14]. However, it requires much time until the optimal behavior obtains a reward in the early stage of learning. Thus, it takes time for the convergence of learning. Furthermore, because all combinations of a state and behavior are evaluated for a predetermined Q value, it is difficult to follow environmental change. Therefore, learning by a real robot is extremely difficult because of the processing time.

On the other hand, Classifier System can learn a novel classification and to maintain the diversity by means of GA, which evolves a rule including # (don't care symbol). Thus, it enables the learning with relatively few trials so that the evolved robot may adapt the dynamic environment more effectively. However, too much generalization might result in the poor performance due to the overfitting.

We use these above two methods for the sake of simulation-based learning of the position correction and compare the obtained results.

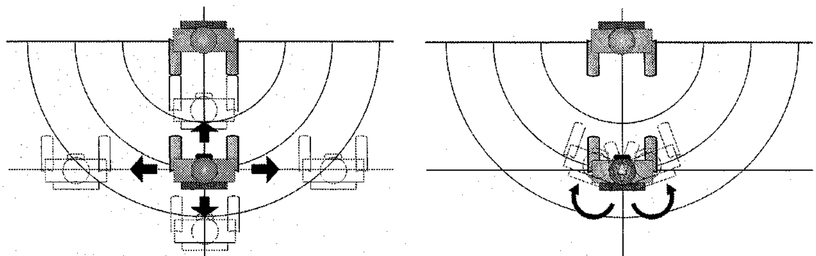
The effective division of states and the selection of actions are very essential for the sake of efficient Q-learning and Classifier System. A static camera is attached to one robot to obtain information required for learning from the external environment. The external situation is evaluated with images from this static camera. Based on the partner robot's position projected on the image acquired by the static camera, a state space is arranged as shown in Fig. 1.5. It is divided into three states: vertical, horizontal, and angular. Hence, the total number of states of the environment is 27. If the vertical, horizontal, and angular positions are all centered, the goal will be attained.



(a) Vertical alignment (b) Horizontal alignment (c) Angular alignment

Fig. 1.5. Different states (27-states).

We assumed six behaviors which a robot can choose among the 12 patterns mentioned in Section 1.2. They are the especially important motions of forward, backward, rightward, leftward, right turn, and left turn. Fig. 1.6 depicts all these motions.



(a) Move forward, backward, rightward and leftward (b) Turn right and left

Fig. 1.6. Different actions (6-actions).

1.4.2 Learning in Simulator

The learning model stated in the preceding subsection has been realized in a simulation environment. This simulator sets a target position at a place of constant distance from the front of the partner robot, which is present in a plane. A task will be completed if the learning robot reaches the position and faces the partner robot.

The target position here ranges in distance movable in one motion. In this experiment, back-and-forth and lateral distances and the rotational angle movable in one motion are assumed to be constant. That is, if the movable distance in one step is about 10 cm back-and-forth and 5 cm laterally, the range of the target point will be 50 cm^2. In this range, the goal will be attained if the learning robot is in place where it can face the partner robot with one rotation motion.

The Q-learning parameters for the simulation were as follows: the initial Q value, Q_0, was 0.0, the learning rate α was 0.01, the reduction ratio γ was 0.8 and the reward was 1.0 for the task achievement. We used the following parameters for Classifier Systems and GA: the initial value for a rule is 0.1, the tax is 0.001, the bid value is 0.01, the crossover rate is 0.95, the mutation ratio is 0.05, and the population size is 1,024.

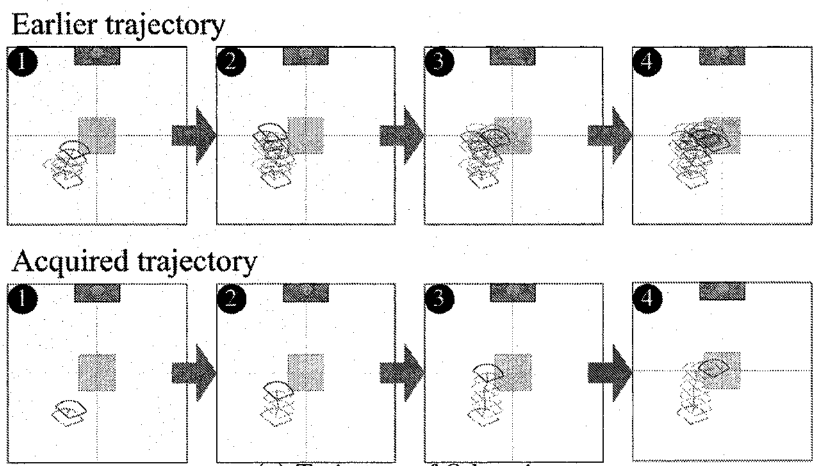
A certain noise is added to the motion. This is to establish the learning scheme in consideration of uncertain factors, such as translation errors due to the motion or different operational characteristics of robots. More precisely, 5% error is given to a motion at one time as noise.

1.4.3 Result of Simulator Learning

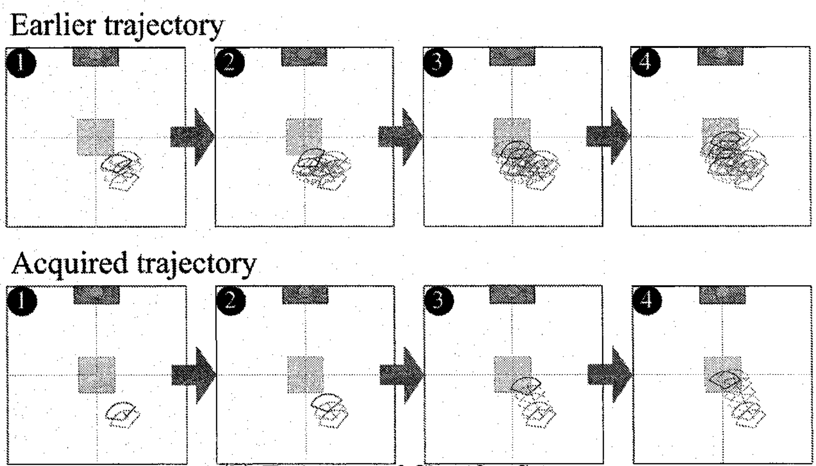
Behavior patterns obtained by simulation with the Q-learning approach in the early stage and acquired by learning are shown in Figs. 1.7a and 1.7b, respectively. In the early stage, motions are observed such as walking to the same place repeatedly and going to a direction different from the target position. Behavior approaching the target position is gradually observed as learning progresses; finally, behavior is acquired to move to the target position and turn to the front with relatively few motions.

As can be seen Classifier System simulation by in Figs. 1.7c and 1.7d, the trajectory divergence occurred at the earlier stage of learning. However, at the later generations, the effective actions were acquired so as to face the goal correctly.

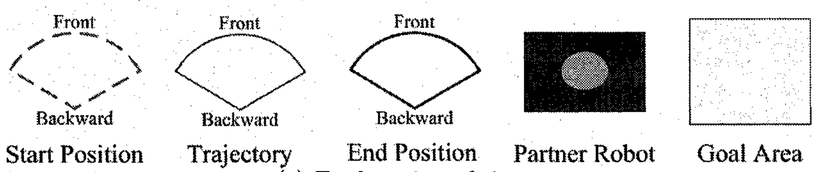
Fig. 1.8a plots the success rate of learning for 1,000 steps. Fig. 1.8b gives the number of successful motions with generations. Both data were averaged over 10 runs. As can be seen, Q-learning is superior. This may be because it enables hill-climbing local search. Classifier System's performance goes up and down irregularly. However, this is considered to show the superiority in terms of the robust learning. As a result of this, numbers of motions are almost the same for both methods as the later stage of learning.



(a) Trajectory of Q-learning.

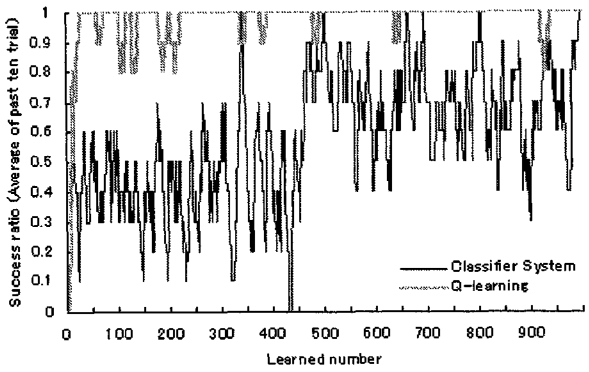


(b) Trajectory of Classifier System.

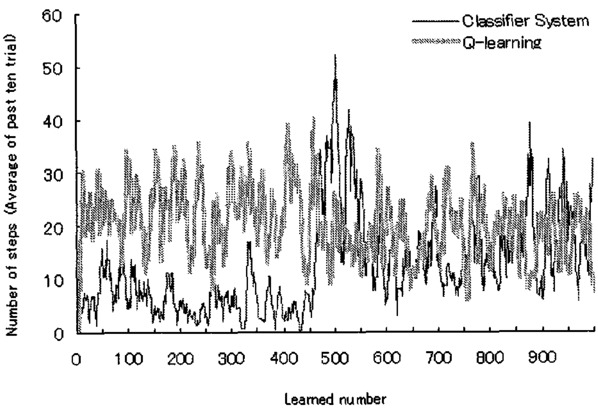


(c) Explanation of signs.

Fig. 1.7. Results of a simulation with Q-learning and Classifier System.



(a) Success ratio.



(b) Moved number.

Fig. 1.8. Q-learning vs. Classifier System.

1.4.4 Experiments with Real Robots

Following the simulation results described in the previous subsection, we conducted an experiment with real robots to con-firm their applicability. In this experiment, we have used the learning data obtained from Q-learning, because Q-learning acquired the relatively more precise behaviors than Classifier System in the previous simulation.

For the recovery from the horizontal left (right) slide, a humanoid robot was initially shifted leftward (rightward) against the opponent robot by 5.2 cm. On the other hand, it was initially moved forward (backward) from the correct position by 3.2 cm for the recovery from front (back) position. In case of the rotation failure, the robot was shifted either leftward or rightward by

5.2 cm and rotated toward the opponent by 20 degrees. The images of the static camera in each pattern are shown in Fig. 1.9. The actions used for the recovery were of six kinds, i.e., half forward, half backward, half rightward, half leftward, right turn and left turn.

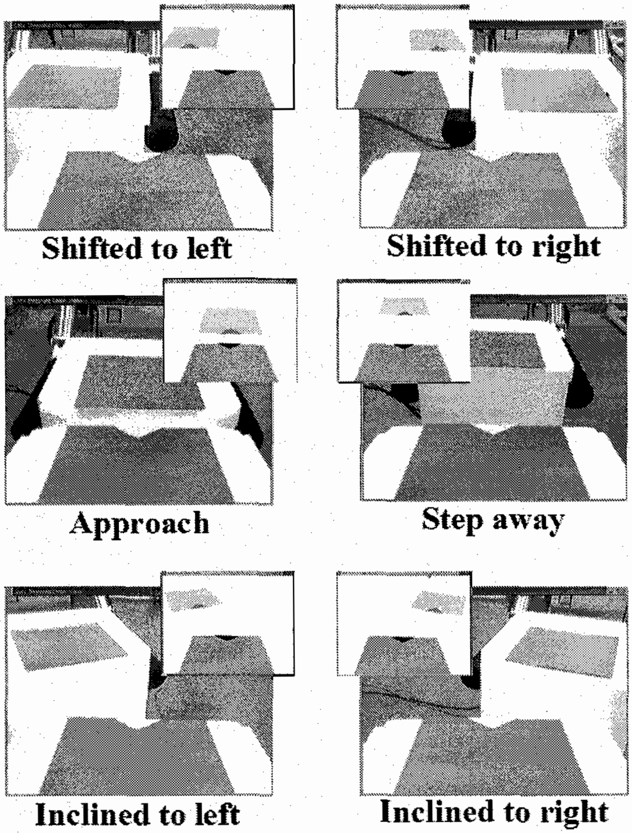


Fig. 1.9. Type of the experiments.

For this experiment, robots started from one of the three patterns shown in Figs. 1.2b, 1.2c and 1.2d, which were classified as the failure of actions (see Section 1.2). We employed two HOAP-1's, one of which used the learning results, i.e., the acquired Q-table, so as to generate actions for the sake of recovery from the failure. Q-learning was conducted by simulation with different numbers of iterations, i.e., 1,000, 10,000, and 100,000 iterations. The learning parameters were the same as in the previous subsection.

1.4.5 Experimental results

Table 1.1 shows the averaged numbers of actions for the sake of recovery from the above three failure patterns. In Table 1.1: RL represents the slide recovery from the right, LR is the slide recovery from the left, NF stands for the distance recovery from the front, FN is defined as the distance recovery from the back, RLS and LRS are respectively the angle recovery from the right and from the left. The averaged numbers of required actions were measured over five runs for each experimental condition, i.e., with different Q-learning iterations.

Table 1.1. Numbers of average movement.

		Q-learning Iterations		
Failure	Recovery	1,000times	10,000times	100,000times
Horizontal slide	RL	4.6	4.4	4.4
	LR	5.4	5.2	5.2
Approach and away	NF	6.6	1.8	1.8
	FN	2.0	2.0	1.4
Spinning around	RLS	9.4	9.4	8.6
	LRS	11.4	16.8	10.2

For slide motion, the robot learned an effective motion after 1,000 time steps. This is explained in the following way. A gap usually occurs even when a robot corrects a position. However, correcting a slide position requires only a simple sequence of actions, as a result of which the gap rarely occurs.

With 1,000 iterations, more actions were needed to recover from the front position to the back. This is because the robot had acquired the wrong habit of moving leftward when the opponent robot was approaching (see Fig. 1.10). This habit has been corrected with 10,000 iterations, so that much fewer actions were required for the purpose of repositioning.

The recovery from "spinning around" seems to be the most difficult among the three patterns. For this task, the movement from the slant to the front (see Fig. 1.11) was observed with 10,000 iterations, which resulted in the increase of required actions. This action sequence was not observed with 1,000 iterations. This is considered that the phenomenon is caused by the difference between simulation and a real-world environment.

1.5 Cooperative Transportation to Target Position

1.5.1 Experiments with Real robots

The cooperative transportation task, i.e., two humanoid robots cooperate with each other to transport an object to a certain goal, is carried out by using the

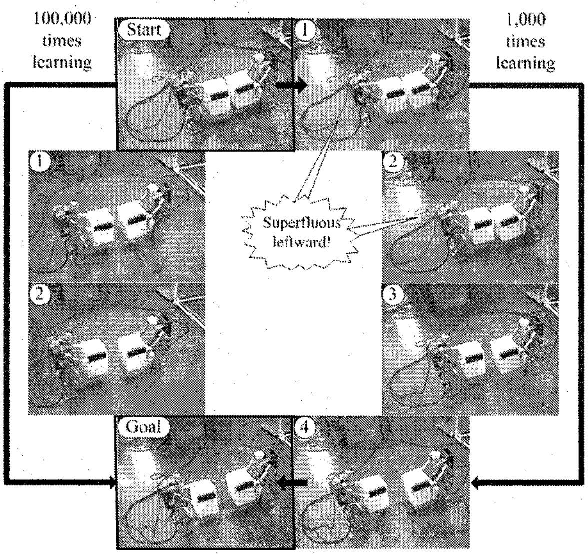


Fig. 1.10. Behavior of NF with short-time learning and full learning.

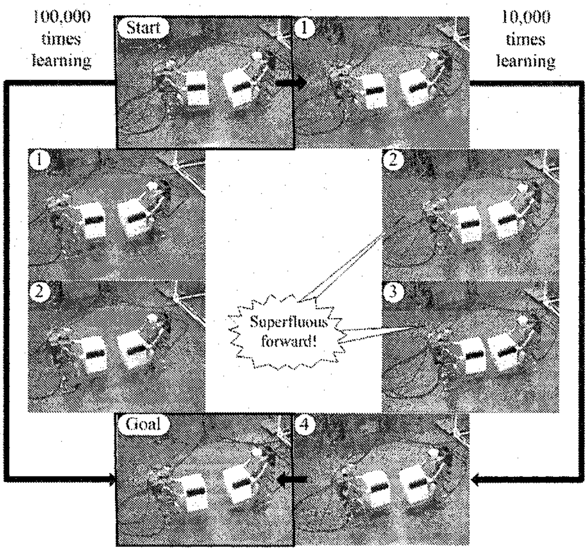


Fig. 1.11. Behavior of LRS with short-time learning and full learning.

obtained Q-learning data shown in the previous section. The transportation target is a sphere made of foam polystyrene. Its diameter is about 25 cm and 63 gram weigh. The goal is positioned in a place about 1m distant from each humanoid robot and is marked for the purpose of recognition.

The Master robot finds its mark using the active camera, and decides the transportation path to the destination. The path is derived as follows:

1. Move the Master robot forward or backward so that it is next to the goal.
2. Move the Master robot left or right to a position adjacent to the mark.

In the meantime, if a positional shift occurs, the Slave robot recognizes its type and tries to recover from it. Afterward, the Master robot searches for a new path again and the transportation is restarted according to the new path.

1.5.2 Experimental results

Fig. 1.12 shows the transportation process with some recovery actions. As can be seen, two recovery actions were performed in case of side motions. As a result, the robots achieved the task successfully. In case of a position shift, the path to the goal was slightly changed. This was caused by each other's shift and its recovery. In order to reduce this anomaly and re-calculation of the path, two robots need to revise their positions simultaneously.

Moreover, when the goal is seen overlapped with the opponent robot, the mark is difficult to recognize. In order to solve this difficulty, two robots should rotate cooperatively with the object on the platform or both robots should be equipped with active cameras for the recognition.

1.6 Discussion

We have established a learning system for the cooperative transportation by simulation and confirmed its real-world applicability by means of real robots. Furthermore, we have conducted cooperative transportation including acquired behavior to correct position using real robots.

The effective actions were acquired for the sake of recovery from the position failure as a result of simulation learning. In a real environment, at the earlier stage of learning, we have often observed the unexpected movement to a wrong direction by real humanoid robots; which was also the case with the simulation. In the middle of learning, the forward movement was more often observed from the slant direction. These types of movements, in fact, had resulted in the better learning performance by simulation, whereas in a real environment they prevented the robot from moving effectively. This is considered to be the distinction between simulation and a real-world environment. We have confirmed the success of the cooperative transportation by real robots, i.e., both robots cooperatively transported an object to a goal while revising their position shift effectively.

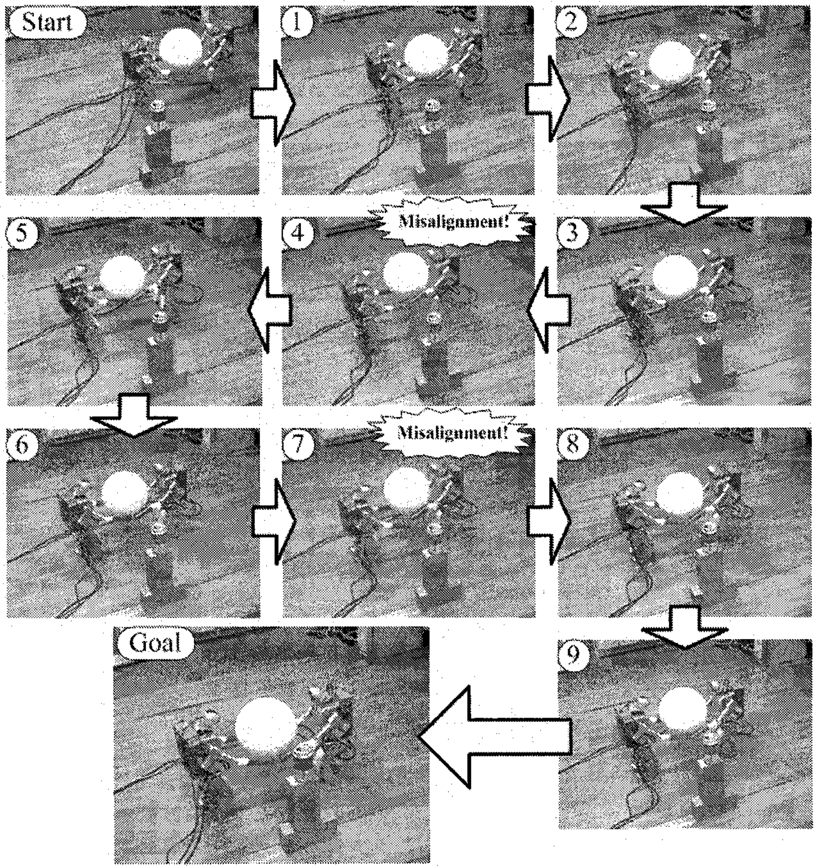


Fig. 1.12. Result of an experiment with real robots.

In this chapter, the position recovery was carried out by one robot. It is more desirable and efficient if both robots can do so. For this purpose, the learning of two robots in a real environment is essential. This is also important to nullify the difference between simulation and real-world environment. However, it is not easy using Q-learning because of the frequent loss of a goal or an opponent in the early state of the learning in a real environment. Thus, we can conclude Classifier System is superior to Q-learning for the purpose of the cooperative learning in a real-world environment.

Moreover, we are now developing a methodology of filtering learning result by means of camera information from difference devices, for the purpose of applying the obtained result in a simulator to a real environment. This method is based on the evolutionary computation and probabilistic estimation.

In order to solve the difficulty with the distinction, learning in the real world is essential. For this purpose, we are currently working on the integration of GP and Q-learning in a real robot environment [15]. This method does not need a precise simulator, because it is learned with a real robot. In other words, the precision requirement is met only if the task is expressed properly. As a result of this idea, we can greatly reduce the cost to make the simulator highly precise and acquire the optimal program by which a real robot can perform well. We especially showed the effectiveness of this approach with various types of real robots, e.g. SONY AIBO or HOAP-1.

1.7 Conclusion

Specific problems were extracted in an experiment using a practical system in an attempt to transport an object cooperatively with two humanoid robots. The result proved that both body swinging during movement and the shift in the center of gravity, by transporting an object, caused a shift in the position after movement.

We investigated the behavior of fundamental motions to make sure the impact of initial positioning on the robot operation. Consequently, it is found that position matching of motors is very difficult even using the same robot and even in the same motion, there occur errors in moving distance and direction.

Therefore, we have proposed a learning method to revise a position shift while the cooperative transportation, and established a learning framework in a simulation. In addition, the obtained results were verified by using real robots in a real environment.

In order to move towards the target position efficiently, it is necessary to perform the real learning by two robots. Therefore, it is important to discuss the approach for efficient movement and perform experiment with real robots. Since huge time is required for learning in real robots, it is important to reduce the time of learning in real environment using learning data in the simulator.

In our future work, we want to study how robots can more to the target in the shortest path when there is an obstacle in the path or how to more in an L-shaped path.

References

1. K. Yokoi, et al., "Humanoid Robot's Application in HRP", In *Proc. of IARP International Workshop on Humanoid and Human Friendly Robotics*, pp.134-141, 2002.
2. H. Inoue, et al., "Humanoid Robotics Project of MITI", *The 1st IEEE-RAS International Conference on Humanoid Robots*, Boston, 2000.
3. O. M. AI-Jarrah and Y. F. Zheng, "Armmanipulator Coordination for Load Sharing using Variable Compliance Control", In *Proc. of the 1997 IEEE International Conference on Robotics and Automation*, pp.895-900, 1997.

4. M. M. Rahman, R. Ikeura and K. Mizutani, "Investigating the Impedance Characteristics of Human Arm for Development of Robots to Cooperate with Human Operators", In *CD-ROM of the 1999 IEEE International Conference on Systems, Man and Cybernetics*, pp.676-681, 1999.
5. N. Miyata, J. Ota, Y. Aiyama, J.Sasaki and T. Arai, "Cooperative Transport System with Regrasping Car-like Mobile Robots", In *Proc. of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.1754-1761, 1997.
6. H. Osumi, H.Nojiri, Y.Kuribayashi and T.Okazaki, "Cooperative Control of Three Mobile Robots for Transporting A Large Object", In *Proc. of International Conference on Machine Automation (ICMA2000)*, pp.421-426, 2000.
7. M. J. Matarić, M. Nillson and K. T. Simsarian, "Cooperative Multi-robot Box-pushing", In *Proc. of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.556-561, 1995.
8. H.Kimura and G.Kajiura, "Motion Recognition Based Cooperation between Human Operating Robot and Autonomous Assistant Robot", In *Proc. of the 1997 IEEE International Conference on Robotics and Automation*, pp.297-302, 1997.
9. Y. Inoue, T. Tohge and H. Iba, "Cooperative Transportation by Humanoid Robots - Learning to Correct Positioning -", In *Proc. of the Hybrid Intelligent Systems (HIS2003)*, pp.1124-1133, 2003.
10. J. Ota, Y. Buei, T. Arai, H. Osumi and K. Suyama, "Transferring Control by Cooperation of Two Mobile Robots", *The Journal of the Robotics Society of Japan (JRSJ)*, vol.14, no.2, pp.263-270, 1996.
11. K. Kosuge, T. Osumi and H. Seki, "Decentralized Control of Multiple Manipulators Handling an Object", In *Proc. of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.318-323, 1996.
12. L. B. Booker, D. E. Goldberg, and J. H. Holland, "Classifier Systems and Genetic Algorithms", In *Machine Learning: Paradigms and Methods*, *MIT Press*, 1990.
13. R. S. Sutton and A. G. Barto, "Reinforcement Learning", *MIT Press*, Boston, 1998.
14. C. J. C. H. Watkins and P. Dayan, "Q-learning", *Machine Learning*, Vol. 8, pp.279-292, 1992.
15. S. Kamio, H. Mitsuhashi and H. Iba, "Integration of Genetic Programming and Reinforcement Learning for Real Robots", In *Proc. of the Genetic Computation Conference (GECCO2003)*, pp.470-477, 2003.