The 6th Asian Control
July 18-21, 2006
Bali, Indonesia

Optimized Real-Time Path Planning for a Robot Manipulator based on PRM and Reinforcement Learning

Ji-Hun Kim*, Jung-Jun Park, Jae-Bok Song*****

* Dept. of Mechanical Eng.
Korea University, Seoul, Korea
e-mail: 1810cp@korea.ac.kr

**Dept. of Mechanical Eng.
Korea University, Seoul, Korea
e-mail: hantiboy@korea.ac.kr

*** Dept. of Mechanical Eng.
Korea University, Seoul, Korea
e-mail: jbsong@korea.ac.kr

Abstract

The Probabilistic RoadMap(PRM) method, one of the popular path planning schemes for a manipulator, can find a collision-free path by connecting the start and goal poses through the roadmap constructed by drawing random nodes in the free configuration space. The PRM shows robust performance for static environments, but rather poor performance for the dynamic environments. On the other hand, reinforcement learning, one of the behavior-based control techniques, can cope with uncertainties in the environment. The agent of reinforcement learning can establish a policy that maximizes the sum of rewards by selecting the optimal actions in any state through iterative interactions with the environment. In this paper, we propose an efficient real-time path planning by combining the PRM and reinforcement learning to cope with uncertain dynamic environments and similar environments. A series of experiments show that the hybrid path planner can generate the collision-free path even for the dynamic environment in which the objects block the pre-planned global path. It is also shown that the hybrid path planner can adapt to the similar environments learned previously without much additional learning.

1 Introduction

A service robot is a human-oriented robot which can provide various services such as education, support for labor and housework, entertainment and so on by interacting with humans. The arm of a service robot, which provides various services to humans as a means of manipulation, is more likely to collide with static obstacles as well as dynamic obstacles including humans than any other parts of the robot.

Path planning for a robot manipulator means generation of an optimized global path which can avoid collision with static or dynamic obstacles in a given workspace [1]. Path planning is conducted either in real workspace or in configuration space composed of a manipulator and obstacles. In the former case, it is advantageous that path planning is performed easily and directly without other specified mapping processes. However, singularity problems may occur because multiple solutions can exist for the given configuration of a manipulator. On the other hand, if configuration space is used for path planning, the environment information on the collision region and collision-free region can be obtained since the joint angles at which the manipulator collides with obstacles can be found. Obstacles having a uniform shape in workspace are usually deformed to an unpredictable shape by the configuration space mapping process. Therefore, it is very difficult for the path planner to cope with dynamic environments without information on accurate pose and configuration for dynamic obstacles.

Several schemes such as a roadmap approach, a cell decomposition method, a potential field method have been proposed to generate the optimal global path in a given configuration space [2]. Among them, the PRM (probability roadmap) method based on the roadmap approach can be applied to not only complex static environments but also a manipulator with high degrees of freedom [3]. Furthermore, it can be easily implemented because of its simple structure. But the PRM requires accurate information on the environment, which is difficult to obtain in practical situations, especially in dynamic environments.

Reinforcement learning (RL) has been used to handle the uncertain situations. In this paper, therefore, we propose an efficient real-time hybrid path planning scheme by combining the PRM and reinforcement learning to cope with uncertain dynamic environments. This hybrid path planner can be applied effectively to the environments similar to the previously learned environments without additional learning.

This paper is organized as follows. Section 2 gives an overview of configuration space, PRM, and reinforcement learning. Section 3 proposes a hybrid path planner based on the PRM and RL. The experimental results for both static and dynamic environments are discussed in this section. Section 4 is concerned with adaptability to

similar environments and a balance between exploration and exploitation. Finally, Section 5 presents conclusions.

2 PRM and Reinforcement Learning

A configuration of an arbitrary object is a specification of its pose (i.e., position and orientation) with respect to a fixed reference frame. The configuration space (C-space for short) is the space that is composed of all possible configurations of the object [2]. It is usually described in the Cartesian coordinate system whose axis represents each degree of freedom of a manipulator. Therefore, an arbitrary point in the C-space corresponds to one specific configuration of a manipulator and a curve connecting two points in the configuration space describes the path of a manipulator.

Path planning of a manipulator based on configuration space shows robust performance for static environments. In the static environment for which full prior information is known, a global collision-free path can be planned for the given start and goal poses. Figure 1 shows the C-space determined by a simple 2-link manipulator and static workspace with various static obstacles.

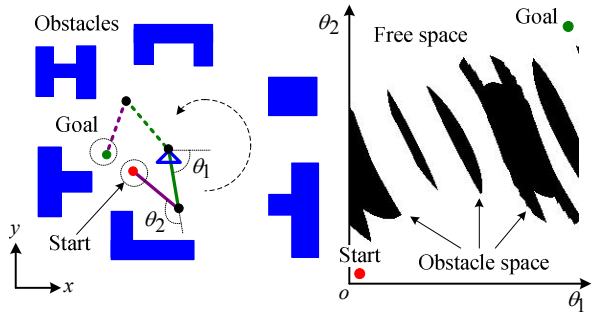

Fig. 1. 2-link manipulator in workspace (left) and its configuration space (right).

The PRM (probabilistic roadmap) planner consists of a preprocessing phase and a query phase. The preprocessing phase draws collision-free nodes called milestones randomly in the free C-space and constructs the roadmap by connecting milestones with directional two-way curves. The query phase generates an optimized global collision-free path by connecting the start and goal poses to two nodes of the roadmap. As an example, if the PRM planner is applied to the configuration space shown in Fig. 1, a global path shown in Fig. 2 can be obtained through the preprocessing and query phases.

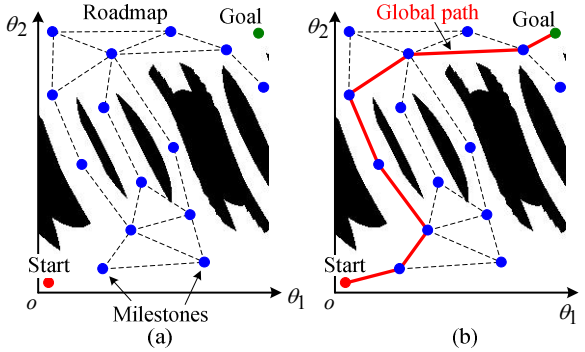

Fig. 2. PRM planner: (a) preprocessing phase, and (b) query phase.

Reinforcement learning (RL) was proposed by Minsky [4][5]. As shown in Fig. 3, the RL agent which performs actual learning interacts continuously with an environment outside the agent. The agent performs an action a_t in some state s_t and receives a real-valued reward r_t from the environment. Through this sequence, the agent learns a control policy π, which can help the agent to select the optimal action at any given state by itself.

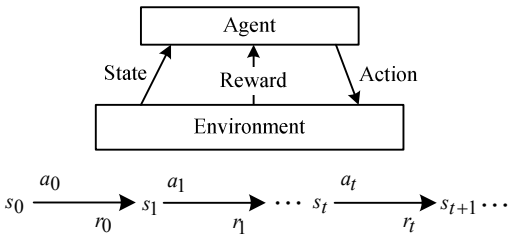

Fig. 3. Standard model of reinforcement learning.

Several conventional methods have been suggested for actual embodiment of reinforcement learning and they are classified into temporal difference learning method, dynamic programming, Monte-Carlo method [6]. In this paper, we use Q-learning (Quality learning) which is based on the temporal difference learning method that combines advantages of the dynamic programming and Monte-Carlo method. Also, Q-learning is suitable for incremental learning process.

3 Hybrid PRM-RL Path Planner

In this paper, the hybrid path planning scheme based on PRM and RL (reinforcement learning) is proposed to improve the adaptability of a PRM planner to dynamic and similar environments. This hybrid path planner is illustrated in Fig. 4.

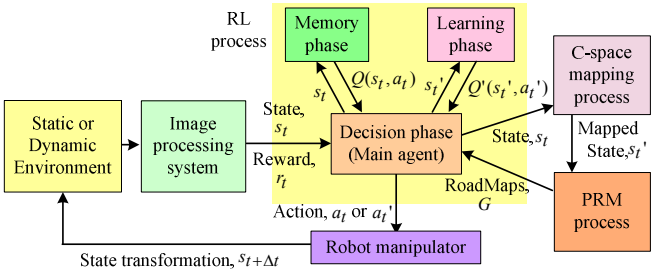

Fig. 4. Hybrid path planner based on PRM and RL.

The image processing system transfers the state information of a static or dynamic environment to the RL agent. In case of a static environment, the poses of static obstacles in workspace are recognized by extracting their color and edge information. Then the image processing system checks whether the obstacle information matches the previously given state information of the static environment. In case of a dynamic environment, the difference image between two successive images is used to detect the dynamic obstacle as shown in Fig. 5.

(a) (b) (c)

Fig. 5. Detection of dynamic obstacle based on difference image: (a) image at time t, (b) image at time $t+\Delta t$, and (c) detected obstacle during Δt.

The C-space mapping process extracts a C-space from a given workspace. The workspace associated with a manipulator with high degrees of freedom is usually mapped into a high-dimensional C-space, which is difficult to visualize and causes computational burden due to the long mapping process. To solve this problem, a dilation operation, quantization of high-dimensional C-space, and the modified slice projection based on feature extraction of obstacle are used in the C-space mapping process.

For a dilation operation, we assume that the manipulator consists of several links with an identical circular cross section but different length. Then the dilation operation is performed by expanding all obstacles in the workspace by an amount equal to the radius of a link as shown in Fig. 6. As a result of the dilation operation, the manipulator with an arbitrary shape can be easily mapped into the C-space. Also, the avoidance of collision between a manipulator and obstacles can be improved by increasing ΔT during the dilation process.

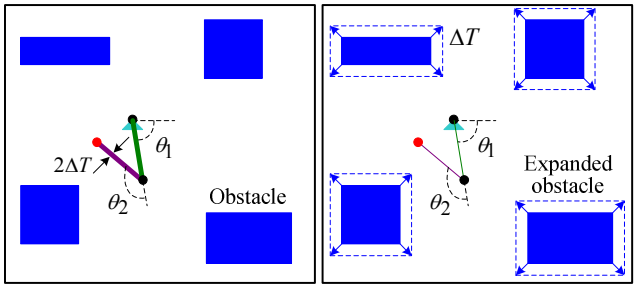

Fig. 6. Expansion of obstacles using dilation operation.

A 6 DOF manipulator usually consists of the positioning structure (joints 1, 2, 3) to control the position of an end-effector and the orienting structure (joints 4, 5, 6) to control its orientation. A mapping process into the 6 dimensional C-space requires long computation time. Furthermore, the orienting structure has little effect on collision in comparison with the positioning structure. Therefore, it is assumed that joints 4, 5, 6 are attached to joint 3 and thus the 6 dimensional C-space is quantized into 3 dimensional C-space in this research.

Figure 7 illustrates the conventional slice projection method. Suppose an obstacle is sliced at intervals of $\Delta\theta$ between θ_{1a} and θ_{1b} in a given workspace. Since an obstacle has a different cross sectional shape with respect to θ_1, the C-space mapping process has to be conducted repeatedly to accurately describe the shape, thus leading to computational burden. To cope with this problem, a modified slice projection method is proposed in this research. An angle θ_1' ought to be found at which the sectional area of an obstacle becomes maximum between θ_{1a} and θ_{1b}. Then the obstacle is assumed to have the same cross sectional area with the one at θ_1' for all θ_1 between θ_{1a} and θ_{1b}. By applying the modified slice projection method, the obstacles in workspace are deformed in configuration space. This deformed obstacle tends to overestimate the obstacle space, but it is advantageous in light of obstacle avoidance.

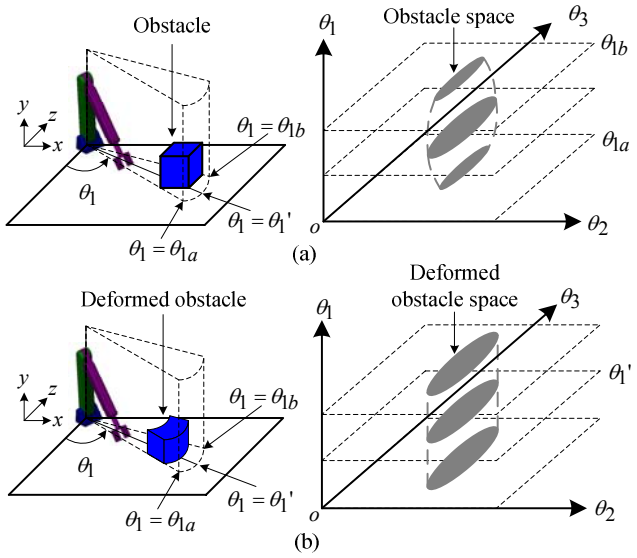

Fig. 7. (a) Conventional slice projection method, and (b) modified slice projection method based on feature extraction of obstacle.

The PRM consists of a preprocessing phase and a query phase. In this hybrid path planner, however, only a preprocessing phase of the PRM is employed to construct a roadmap in the C-space from a given workspace. This roadmap is used as state information for learning performed by the RL agent.

In applying the reinforcement learning (RL) method, the state in an environment is defined as the manipulator configuration given by joint variables θ_1 and θ_2. For example, if the current configuration is given by $\theta_1 = \theta_1'$ and $\theta_2 = \theta_2'$, $s_w(\theta_1', \theta_2')$ and $s_c(\theta_1', \theta_2')$ represent the state variable in the workspace and C-space, respectively.

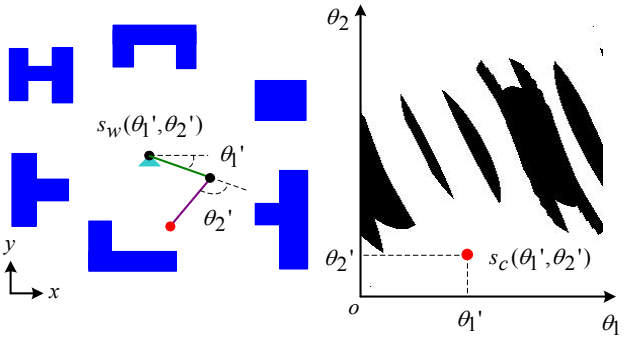

Fig. 8. Definition of state variables for RL.

The action variable which can be chosen by the agent at any arbitrary state $s_c(\theta_1, \theta_2)$ is defined as a set of joint variables which

causes the manipulator to move from the current milestone to another on the roadmap which is constructed by the preprocessing phase of the PRM. For example, if the current state is $s_c(\theta_1', \theta_2')$, then the RL agent can take either the action variable $a_c(\theta_{1a}, \theta_{2a})$ or $a_c(\theta_{1b}, \theta_{2b})$ because the states $s_c(\theta_{1a}, \theta_{2a})$ or $s_c(\theta_{1b}, \theta_{2b})$ are only two state accessible from the current state.

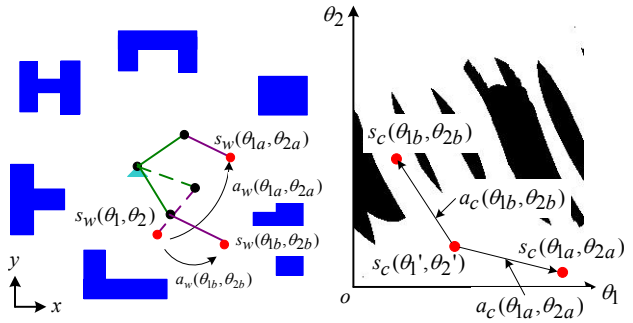

Fig. 9. Definition of the action variables for reinforcement learning.

The reward of RL is a numerical evaluation for an action selected by the agent in the current state. As shown in Fig. 10, the agent receives a numerical reward of $r_t = R$ only when the agent generates a global collision-free path from the start to the goal pose while maintaining the distance to the obstacles greater than the threshold distance throughout the path.

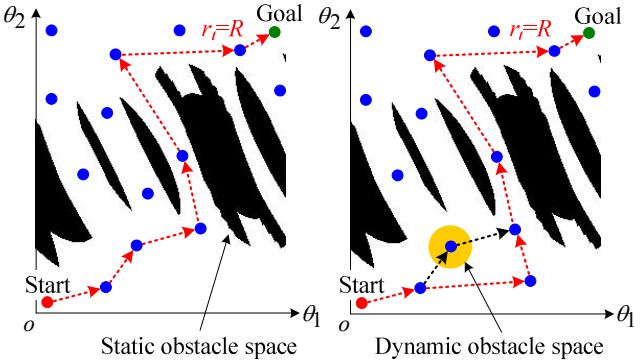

Fig. 10. Numerical reward during RL for generation of optimized global path.

The action-value function $Q(s_t, a_t)$ is defined as the numerical value which evaluates the future influence by the action a_t chosen at the current state s_t. In Q-learning, the action-value function is called a Q-value, and the purpose of Q-learning is to employ a policy π that helps the agent to select an action a_t which makes the Q-value maximum in a given state s_t[6][7]. In this paper renewal of the Q-value is performed by the undeterministic reward and action method as follows:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \cdot \left[r_t + \gamma \cdot \max_{a_{t+1}} Q(s_{t+1},a_{t+1}) - Q(s_t,a_t) \right] \quad (1)$$

where α is the learning rate ($0 \le \alpha \le 1$) determining the convergence rate of learning and γ is the discount rate ($0 \le \gamma \le 1$) which decides a relative ratio between the immediate reward at current state s_t and the delayed reward at future state s_t'. The agent performs learning on all local paths which connect each milestone on the roadmap to reach the goal pose, because a reward is given to the agent only when it reaches the goal pose through the roadmap. In this process, the Q-values for the local paths on the roadmap are renewed continually by Eq. (1). Figure 11 shows that a portion of the learning process performed by the RL agent by Eq. (1) when $\alpha = 0.5$ and $\gamma = 0.5$.

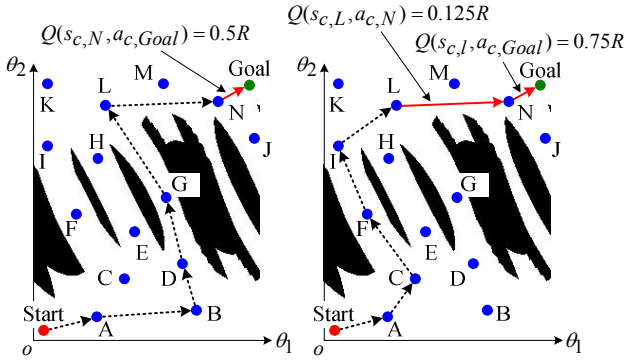

Fig. 11. Learning trial for a given roadmap by agent's policy using reinforcement.

As shown in the above example, the Q-value continues to increase with the increasing number of learning, when the agent performs learning for a given environment. If the learning process is completed, the learning data are obtained so that different weights are given to each local path on the roadmap. Therefore, it is possible to generate the optimized global path by combining local paths which have the maximum Q-values from the start to the goal pose on the roadmap in the static environment as shown in Fig. 12.

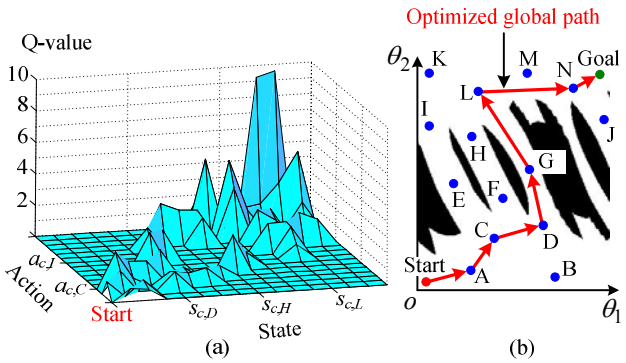

Fig. 12. Results of reinforcement learning for a given roadmap : (a) Q-values, and (b) optimal global path.

As a result of RL, the learning data and the optimal global path are generated as shown in Fig. 12. Suppose a dynamic obstacle blocking the pre-planned global path is detected at an arbitrary milestone during the real manipulation process. In this case, the RL agent updates the learning data generated previously by setting all Q-values related to that milestone to zero. The agent then regenerates another collision-free global path by using the updated Q-values. If the agent has collected sufficient learning data for a given environment, it can avoid dynamic obstacles in real-time by

using the previous learning data without additional learning as shown in Fig. 13.

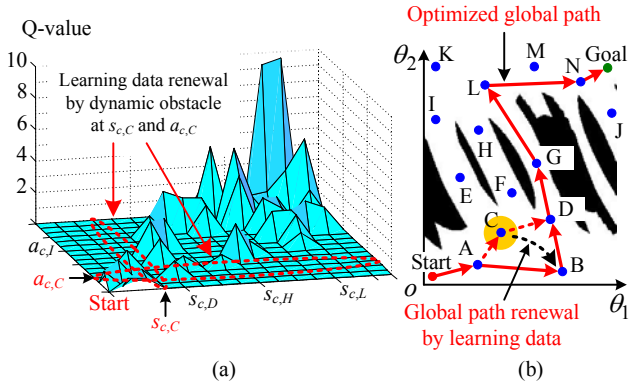

Fig. 13. Renewal of learning data: (a) renewal of Q-values, and (b) renewal of global path caused by occurrence of dynamic obstacle.

To verify the validity of the proposed hybrid PRM-RL path planer, various experiments have been conducted for the environment shown in Fig. 14. The manipulator used for experiments was Samsung FARAMAN AS-1i with 6 DOFs. The stereo camera, Videre STH-MDCS2, was installed on the ceiling to model the environment and detect dynamic obstacles. This camera is capable of providing the range data for each pixel in the image. The C-space was extracted from a given workspace by the modified slice projection method mentioned before. A total of 210 milestones were used to generate the sufficient number of collision-free nodes in the extracted C-space.

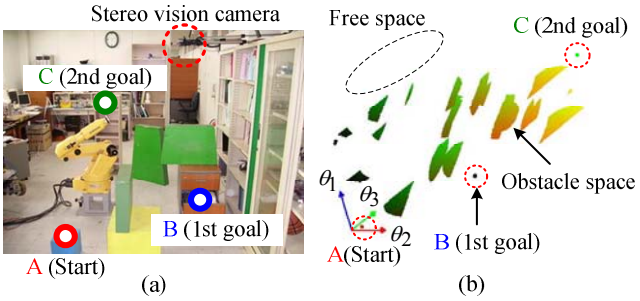

Fig. 14. Constructed environment for first experiment: (a) its workspace, and (b) its configuration space.

Fig. 15 shows the collision-free global path which was optimized from the learning data obtained by applying reinforcement learning to a roadmap constructed in the preprocessing phase of PRM for the static environment shown in Fig. 14. It is shown that the proposed hybrid PRM-RL path planner can provide smoother path than the path planner based on the PRM only.

As shown in Fig. 16, if a dynamic obstacle blocking any milestone on the pre-planned global path during the real manipulation process, the hybrid path planner regenerates another global collision-free path in real-time by resetting all Q-values to zero which are related to that milestone by detecting the positional information of dynamic obstacle from the stereo camera.

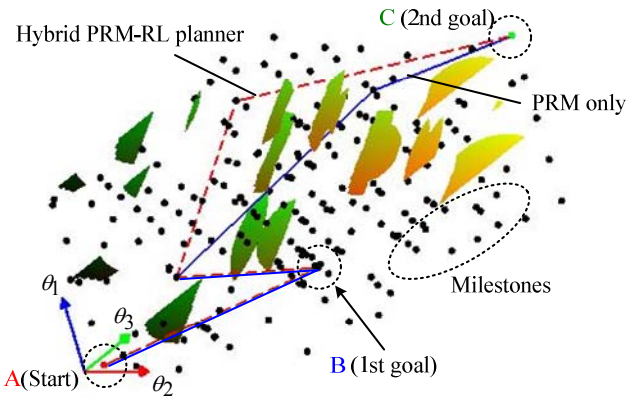

Fig. 15. Global paths for static environment (Experiment)

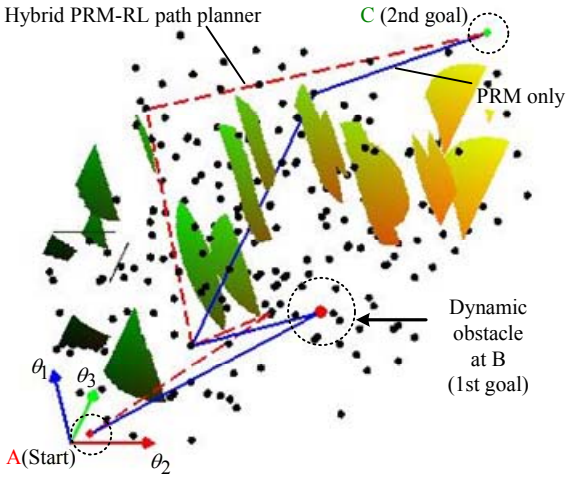

Fig. 16. Global paths for dynamic environment (Experiment)

4 Hybrid PRM-RL Path Planner

The environments in which a service robot operates tend to vary frequently for various reasons. Therefore, it is important that the path planner can adapt to the environments only with a small number of learning once they are similar to the ones learned previously.

To perform learning for a given environment, the RL agent has to collect and analyze various experiments for the current state, the action chosen by the agent, the state transition by action selection, and the best action maximizing the reward. To achieve this, the agent needs a balance between new exploration for a given environment and exploitation based on the existing learning data [7]. To do this, the reward is given by

$$r_t = R \cdot (e^{-p_r} + e^{-p_i}) \quad \text{where } p_r + p_i = 1 \tag{2}$$

where p_r is the exploration rate defined as the ratio of the number of new explorations for a given environment to the total number of learning. Likewise, the exploitation rate p_i is defined as the ratio of the number of exploitations based on the existing learning data to the total number of learning. As shown in Fig. 17, the identical reward is given independent of exploration or exploitation.

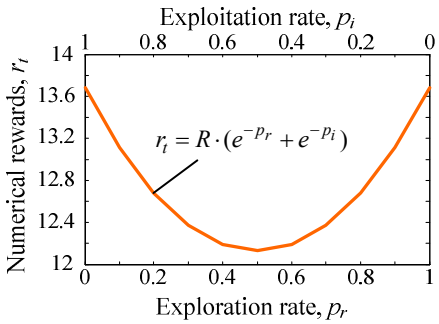

Fig. 17. Modified numerical reward for optimization of path planning.

Various experiments were conducted to verify the adaptability of the hybrid path planner to similar environments. First, the agent performed learning 100 times for environment A shown in Fig. 18. Next, it performed learning 100 times in order for environments B and C which were similar to environment A. Environments B and C were changed from A by slightly changing the poses of obstacles.

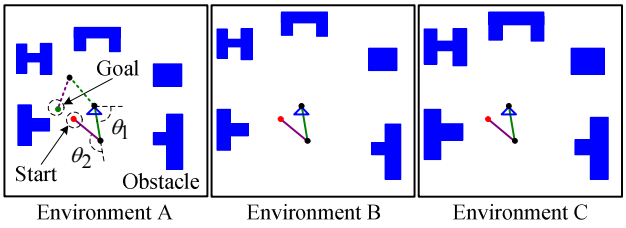

Fig. 18. Three types of environments to investigate adaptability to similar environments.

Figure 19 shows the experimental results showing the adaptability of the hybrid path planner to similar environments. Whenever the agent is given a new environment, it does not know whether the environment is a completely new (e.g., environment A) or it is similar to one of those which have been learned previously (e.g., environment B, C). As the characteristic of the environment is found in the decision period in which both exploration and exploitation are tried in identical rates, the agent performs reinforcement learning in a different way. If a given environment is completely new, the hybrid path planner performs learning with higher exploration rate than exploitation rate, thus meaning that the agent tends to make new attempts continuously for a given environment. On the other hand, if a given environment is similar to the one learned previously, the agent performs learning with higher exploitation rate than exploration rate, thus meaning that it tends to use the previous learning data.

5 Conclusions

In this paper, we propose the hybrid path planner based on the PRM and reinforcement learning to enable the manipulator to cope with both static and dynamic environments and to adapt to similar environments. From various experiments, the following conclusions are drawn:

1. The hybrid path planner can generate a collision-free optimal global path in static environments, provided the environment is known in advance.
2. The hybrid path planner can cope with dynamic environments in which an obstacle blocks any milestone on the pre-planned global path by regenerating another collision-free global path without additional learning.
3. The hybrid path planner can effectively adapt to the environments identical or similar to the ones experienced previously by autonomously adjusting a balance between exploration and exploitation.

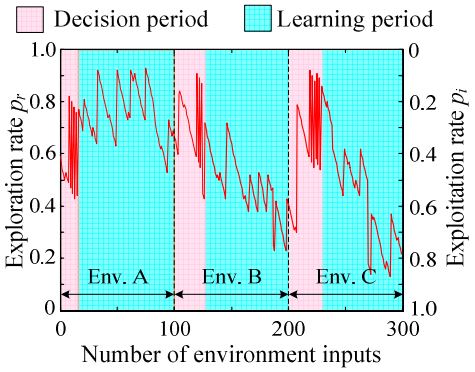

Fig. 19. Optimization of path planning by a balance between exploration and exploitation.

Acknowledgements

This research was supported by the Personal Robot Development Project funded by the Ministry of Commerce, Industry and Energy of Korea.

References

[1] P. J. McKerrow, "Robotics," Addison Wesley, pp. 507-515, 1992.

[2] J.C. Latombe, "Robot Motion Planning," Kluwer Academic Publishers, 1993.

[3] L.E. Kavraki, J.C. Latombe and M. H. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE Trans. on Robotics and Automation,* vol. 12, no. 4, Aug., 1996.

[4] M.L. Minsky, "Theory of Neural - Analog Reinforcement System and Application to the Brain - Model Problem," Ph.D. Thesis, Princeton Univ., Princeton, 1954.

[5] A.G. Barto, D.A. White and D.A. Sofge, "Reinforcement learning and adaptive critic methods," *Handbook of Intelligent Control*, pp. 469-491, 1992.

[6] R.S. Sutton and A.G. Barto, "An introduction to Reinforcement Learning: An Introduction," MIT Press, 1998.

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

