Path Planning for a Robot Manipulator based on Probabilistic Roadmap and Reinforcement Learning

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

Abstract: The probabilistic roadmap (PRM) method, which is a popular path planning scheme, for a manipulator, can find a collision-free path by connecting the start and goal poses through a roadmap constructed by drawing random nodes in the free configuration space. PRM exhibits robust performance for static environments, but its performance is poor for dynamic environments. On the other hand, reinforcement learning, a behavior-based control technique, can deal with uncertainties in the environment. The reinforcement learning agent 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 efficient real-time path planning by combining PRM and reinforcement learning to deal with uncertain dynamic environments and similar environments. A series of experiments demonstrate that the proposed hybrid path planner can generate a collision-free path even for dynamic environments in which objects block the pre-planned global path. It is also shown that the hybrid path planner can adapt to the similar, previously learned environments without significant additional learning.

Keywords: Path planning, probabilistic roadmap, reinforcement learning, robot manipulator.

1. INTRODUCTION

A service robot is a human-oriented robot that can provide various services such as education, support for labor and housework, entertainment, and so on by interacting with humans. Among all the parts of a service robot, its arm, which can be manipulated to provide various services to humans, is the most likely to collide with static as well as dynamic obstacles including humans.

Path planning for a robot manipulator requires the generation of an optimized global path that can avoid collisions with static or dynamic obstacles in a given workspace [1]. Path planning is conducted either in a real workspace or in a configuration space (C-space) comprising a manipulator and obstacles. The former case advantageous since path planning is performed easily and directly without other specified mapping processes. However, singularity problems may occur because multiple solutions can exist for a given configuration of the manipulator. To cope with these drawbacks, a collision avoidance solution that used the back propagation neural network was proposed in [2], but it still had the uncertainty depending on the training set.

On the other hand, in the latter case, environment information on the collision and collision-free regions can be obtained since the joint angles at which the manipulator collides with obstacles can be determined. Obstacles having a uniform shape in the workspace are usually deformed to an unpredictable shape by the C-space mapping process. Therefore, it is very difficult for the path planner to deal with dynamic environments without accurate information on the pose and configuration of dynamic obstacles.

Several schemes such as a roadmap approach, cell decomposition method, potential field method have been proposed to generate an optimal global path in a given C-space [3,4]. Among these, the PRM (probabilistic roadmap) method based on the roadmap approach can be applied to complex static environments as well as to a manipulator with high degrees of freedom [5]. Furthermore, it can be easily implemented because of its simple structure. However, it requires accurate information on the environment, which is difficult to obtain in practical situations, especially in dynamic environments.

Since most environments involve uncertainties due to various causes, practical path planning should deal with such uncertainties. Reinforcement learning (RL) has been used to handle uncertain situations in various applications. Therefore, in this paper, we propose an

Manuscript received July 16, 2006; revised July 7, 2007; accepted September 19, 2007. Recommended by Editorial Board member Jang Myung Lee under the direction of Editor Jae Weon Choi. This research was supported by the Personal Robot Development Project funded by the Ministry of Commerce, Industry and Energy of Korea.

Jung-Jun Park, Ji-Hun Kim, and Jae-Bok Song are with the Department of Mechanical Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul 136-713, Korea (e-mails: {hantiboy, 1810cp, jbsong}@korea.ac.kr).

* Corresponding author.

efficient real-time hybrid path planning scheme that combines PRM and reinforcement learning to deal with uncertain dynamic environments. The real-time operation of the path planner is another important issue. When the information on obstacles is given in a workspace, the slice projection method is used to convert the workspace into the C-space, which requires a large computation time. Therefore, in this research, the use of a modified slice projection algorithm is proposed to reduce this computational burden. A series of experiments demonstrate that the proposed hybrid path planner can generate a collision-free path even for a dynamic environment in which objects block the pre-planned global path. It is also shown that the hybrid path planner can adapt to the similar, previously learned environments without significant additional learning.

This paper is organized as follows. Section 2 provides an overview of the 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 discusses the adaptability to similar environments and a balance between exploration and exploitation. Finally, Section 5 presents the conclusions.

2. PRM AND REINFORCEMENT LEARNING

The 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) is the space that comprises all possible configurations of the objects [3]. It is usually described in the Cartesian coordinate system whose axes represent each degree of freedom of a manipulator. Therefore, an arbitrary point in the C-space corresponds to one specific configuration of the manipulator and a curve connecting two points in the C- space exhibits the path of the manipulator.

The path planning of a manipulator based on the C-space exhibits robust performance for static environments. In a static environment for which

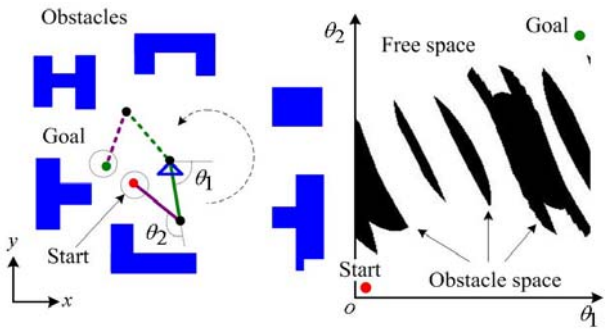

Fig. 1. Two-link manipulator in a workspace (left) and its configuration space (right).

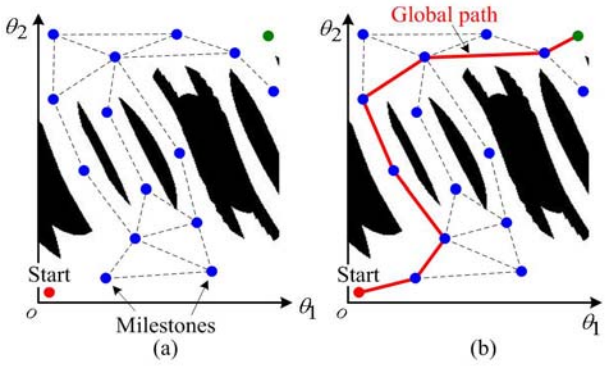

Fig. 2. PRM planner: (a) preprocessing phase and (b) query phase.

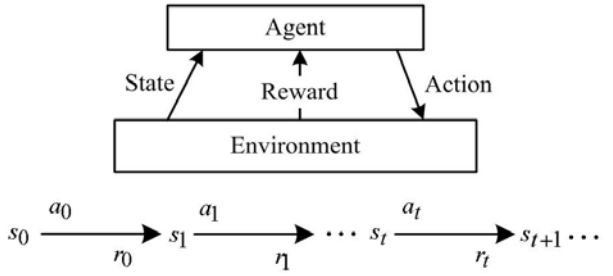

Fig. 3. Standard model of reinforcement learning.

complete prior information is known, a global collision-free path can be planned for the given start and goal poses. Fig. 1 shows the C-space determined by a simple two-link manipulator and a static workspace with various static obstacles.

The PRM planner comprises a preprocessing phase and a query phase. The preprocessing phase randomly draws collision-free nodes, called *milestones*, in the free C-space and constructs the roadmap by connecting the 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. For example, if the PRM planner is applied to the C-space shown in Fig. 1, the global path shown in Fig. 2 is obtained through the preprocessing and query phases.

Reinforcement learning (RL) was proposed by [6,7]. As shown in Fig. 3, the RL agent that performs the actual learning interacts continuously with the 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 process, the agent learns a control policy π that enables it to select the optimal action at any given state by itself.

Several conventional methods such as the temporal difference learning method, dynamic programming, and Monte-Carlo method [8] have been suggested for the actual realization of reinforcement learning. In this paper, we use Q-learning (quality learning) which is based on the temporal difference learning method that combines the advantages of dynamic programming

and the Monte-Carlo method. Further, Q-learning is suitable for incremental learning processes.

3. HYBRID PRM/RL PATH PLANNER

3.1. Path planner based on PRM and RL algorithms

In this paper, a hybrid path planning scheme based on PRM and RL is proposed to improve the adaptability of a PRM planner to dynamic and similar environments. This hybrid path planner is shown in Fig. 4. The components comprising this hybrid path planner are described below in detail below.

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

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

For a dilation operation, we assume that the manipulator comprises several links with an identical circular cross section but different lengths. Then, dilation 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 this operation, a manipulator with an arbitrary shape can be easily mapped into the C-space. Further, the collision avoidance between a manipulator and obstacles can be improved by increasing ΔT during the dilation process.

A 6-DOF manipulator usually comprises a positioning structure (joints 1, 2, and 3) to control the position of an end-effector and an orienting structure (joints 4, 5, and 6) to control its orientation. A mapping process into the six-dimensional C-space requires a substantial amount of computation. Furthermore, the orienting structure has a minimal effect on the collision as compared to the positioning structure. Therefore, it is assumed in this research that joints 4, 5, and 6 are attached to joint 3 and thus the six-dimensional C-space is quantized into a three-dimensional C-space.

Fig. 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

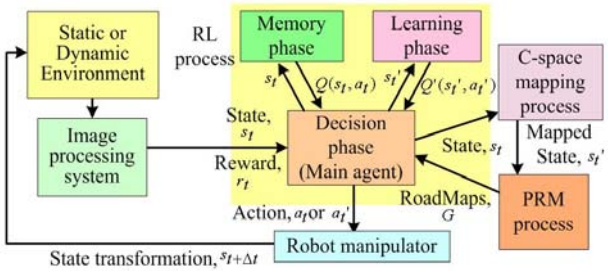

Fig. 4. Hybrid path planner based on PRM and RL.

(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.

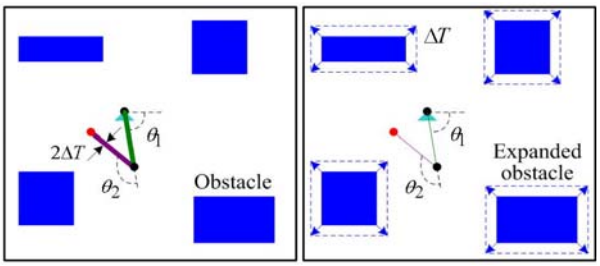

Fig. 6. Expansion of obstacles using dilation operation.

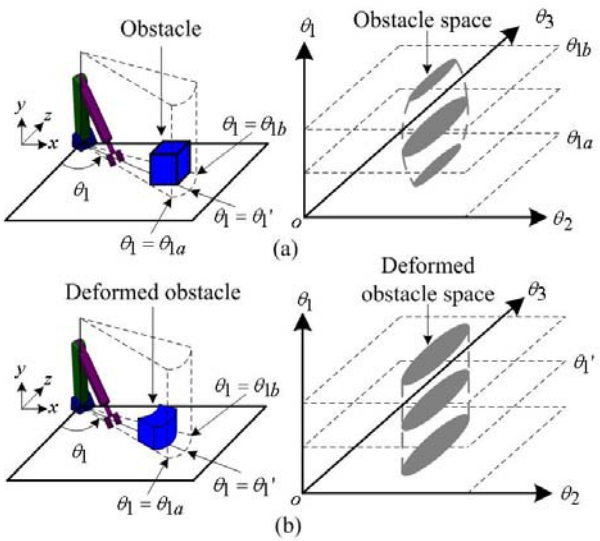

Fig. 7. (a) Conventional slice projection method and (b) modified slice projection method based on feature extraction of obstacles.

an obstacle has a different cross-sectional shape with respect to θ_1, the C-space mapping process has to be performed repeatedly to accurately describe the shape, thus leading to a computational burden. In order to deal with this problem, a modified slice projection method is proposed in this research. The angle θ_1' at which the sectional area of an obstacle becomes maximum between θ_{1a} and θ_{1b} must be found. Then, the obstacle is assumed to have the same cross-sectional area as 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 a workspace are deformed in the configuration space. This deformed obstacle tends to overestimate the obstacle space; however, it is advantageous in terms of obstacle avoidance.

PRM comprises a preprocessing phase and a query phase. However, in this hybrid path planner, only the preprocessing phase of PRM is employed to construct a roadmap in the C-space from a given workspace. This roadmap is used as the state information for the learning performed by the RL agent. When applying the RL method, the state in an environment is defined as the manipulator configuration given by the joint variables θ_1 and θ_2. For example, if the current configuration is given by $\theta_1 = \theta_1'$ and $\theta_2 = \theta_2'$, then s_w (θ_1', θ_2') and $s_c(\theta_1', \theta_2')$ represent the state variables in the workspace and C-space, respectively.

The action variable, which can be selected by the agent at any arbitrary state $s_c(\theta_1, \theta_2)$, is defined as a set of joint variables that causes the manipulator to move from the current milestone to another on the

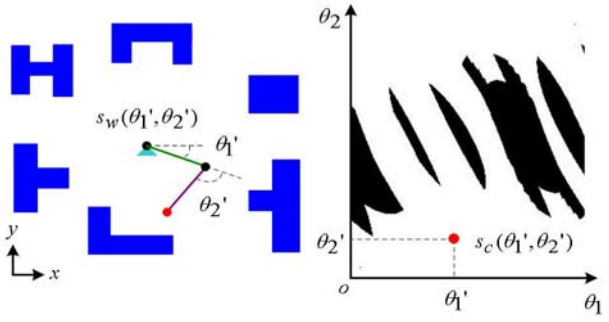

Fig. 8. Definition of state variables for RL.

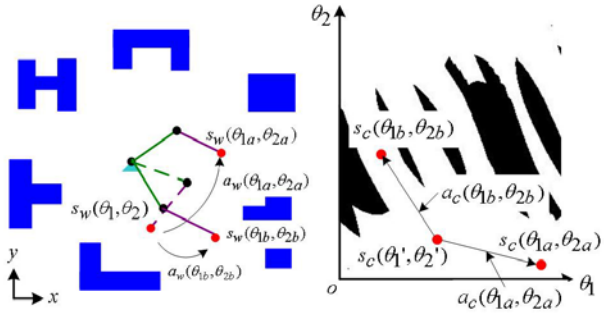

Fig. 9. Definition of the action variables for reinforcement learning.

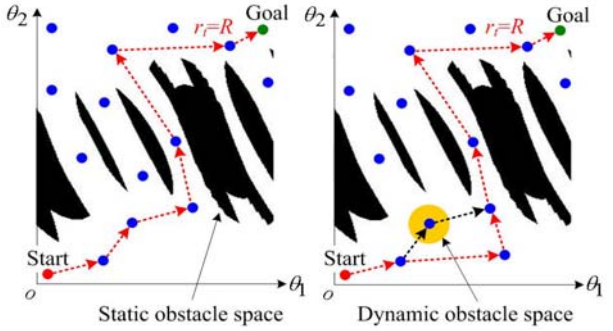

Fig. 10. Numerical reward during RL for generation of an optimized global path.

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

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 it generates a global collision-free path from the start to the goal pose while maintaining the distance to the obstacles that are greater than the threshold distance throughout the path.

3.2. Q-Learning

The action-value function $Q(s_t, a_t)$ is defined as the numerical value that 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 that makes the Q-value maximum in a given state s_t [8,9]. In this paper, the 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 ($0 \le \alpha \le 1$) that determines the convergence rate of learning and γ is the discount rate ($0 \le \gamma \le 1$) that decides the relative ratio between the immediate reward at the current state s_t and the delayed reward at the future state s_t'. The agent performs learning on all local paths that connect each milestone on the roadmap to reach the goal pose because it is rewarded 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 continuously by (1). Fig. 11 shows a portion of the learning process performed by the RL agent using (1) when $\alpha = 0.5$ and $\gamma = 0.5$.

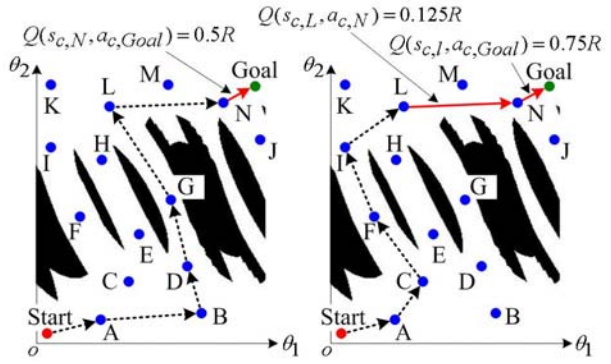

Fig. 11. Learning trial for a given roadmap by the agent's policy using reinforcement.

As shown in the above example, the Q-value increases with the amount 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 can be assigned to each local path on the roadmap. Therefore, it is possible to generate the optimized global path by combining the local paths that have the maximum Q-values from the start to the goal pose on the roadmap in the static environment, as shown in Fig. 12.

The learning data and 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 (e.g., 16 learning data), it can avoid dynamic obstacles in real-time by using the previous learning data without additional learning as shown in Fig. 13.

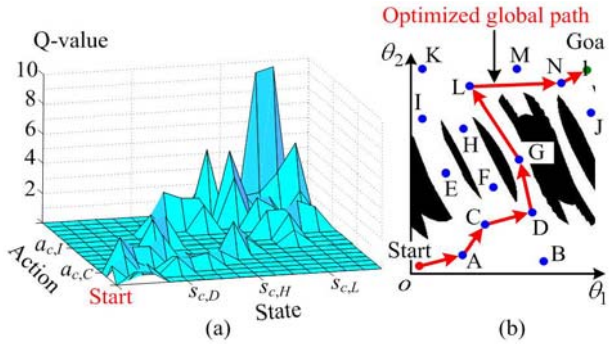

Fig. 12. Results of reinforcement learning for a given roadmap: (a) Q-values, and (b) optimal global path.

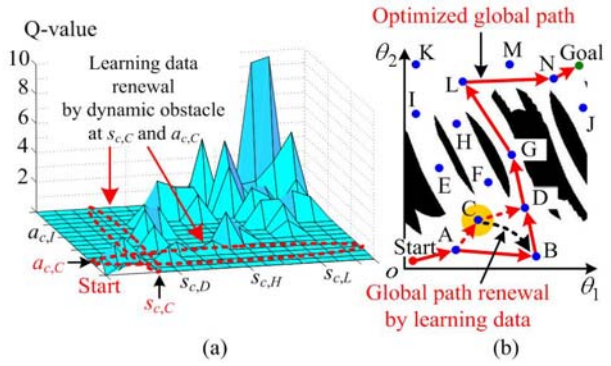

Fig. 13. Renewal of learning data: (a) renewal of Q-values and (b) renewal of global path caused by the occurrence of a dynamic obstacle.

3.3. Experimental results

In order to verify the validity of the proposed hybrid PRM/RL path planner, various experiments have been conducted for the environment shown in Fig. 14. The manipulator used for the experiments was a Samsung FARAMAN AS-1i with 6 DOFs. A stereo camera, Videre STH-MDCS2, was installed on the ceiling to model the environment and detect dynamic obstacles. This camera can provide 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 previously. A total of 210 milestones were used to generate a sufficient number of collision-free nodes in the extracted C-space.

Fig. 15 shows the collision-free global path that 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 a smoother path than the path planner based on only PRM which is difficult to visualize.

As shown in Fig. 16, if a dynamic obstacle blocks any milestone on the pre-planned global path during the real manipulation process, a manipulator executed by the PRM alone is likely to collide with this obstacle since PRM does not include any procedure that causes the manipulator to move to the nearby milestones where a collision can be avoided. However,

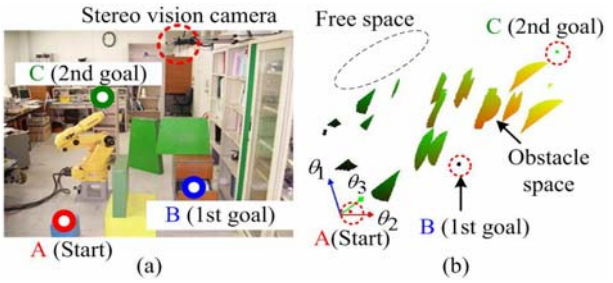

Fig. 14. Constructed environment for first experiment: (a) its workspace and (b) its configuration space.

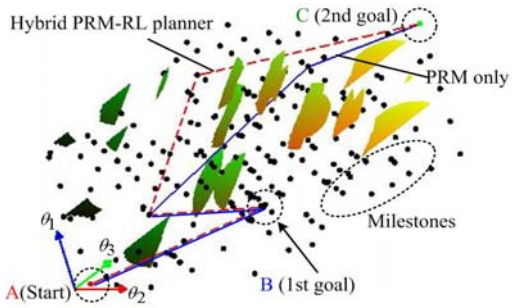

Fig. 15. Global paths for static environment (experiment).

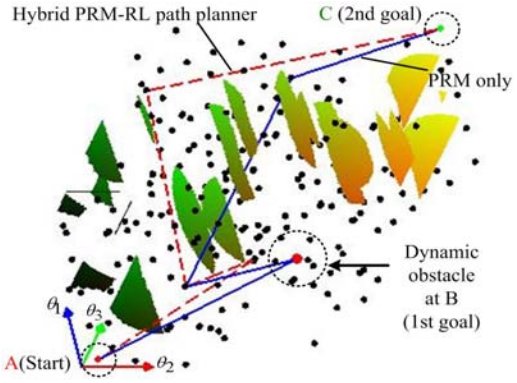

Fig. 16. Global paths for a dynamic environment (experiment).

the hybrid path planner regenerates another global collision-free path in real-time by resetting all Q-values related to the milestone occupied by the dynamic obstacle to zero by detecting the positional information of the dynamic obstacle from the stereo camera. Since the hybrid path planner extracts the optimal path based on the Q-values updated above, it can generate a new collision-avoidance path without performing learning again for this dynamic environment.

4. ADAPTABILITY TO SIMILAR ENVIRONMENTS

The environments in which a service robot operates tend to vary frequently for various reasons. Therefore, the path planner must be capable of adapting to new environments with minimal learning once they are similar to the ones learned previously.

In order to perform learning for a given environment, the RL agent must 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. The agent must strike a balance between new exploration for a given environment and exploitation based on the existing learning data [9]. The reward for this procedure is given by

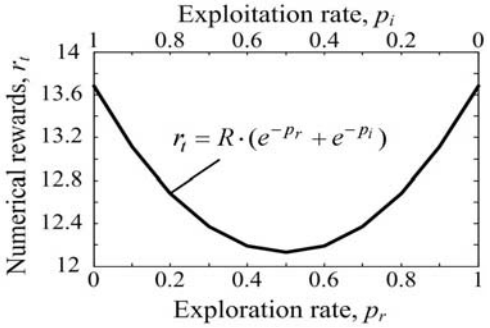

Fig. 17. Modified numerical reward for optimization of path planning.

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

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

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 for environments B and C, which were similar to environment A. Environments B and C were varied from A by making slight changes to the positions of the obstacles.

Fig. 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 this environment is a completely new (e.g., environment A) or is similar to one that has already been learned (e.g., environments B and C). Since the characteristics of the environment are determined in the decision period during which both exploration and exploitation are performed to identical extents, the agent performs reinforcement learning in a different manner. If a given environment is completely new, the hybrid path planner performs learning with a higher exploration rate and lower exploitation rate, implying that the

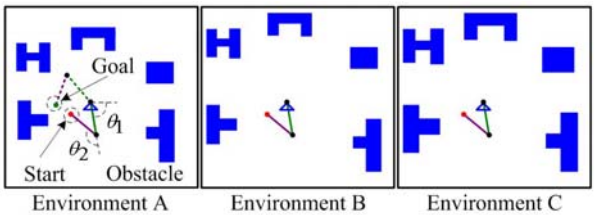

Fig. 18. Three slightly different environments used for investigating the adaptability to similar environments.

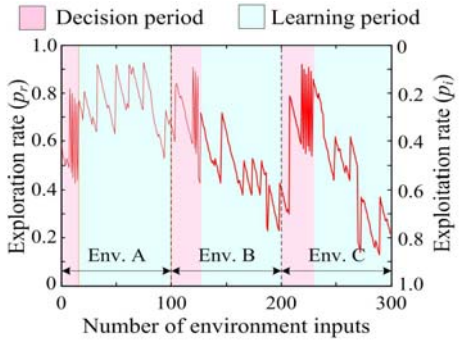

Fig. 19. Optimization of path planning by striking a balance between exploration and exploitation.

agent tends to make new attempts continuously for a given environment. On the other hand, if a given environment is similar to one learned previously, the agent performs learning with a higher exploitation rate and lower exploration rate, implying that it tends to exploit the previous learning data.

5. CONCLUSIONS

In this paper, we propose a hybrid path planner based on PRM and reinforcement learning to enable the manipulator to deal 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. If a sufficient amount of learning can be performed and the Q-values can be imposed on the local paths on the roadmap, the optimal global path can be generated by combining the local paths having the maximum Q-values.

2. The hybrid path planner can deal 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 and robustly adapt to the environments that are identical or similar to the ones learned by autonomously adjusting a balance between exploration and exploitation.

REFERENCES

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

[2] S. F. M. Assal, K. Watanabe, and K. Izumi, "Fuzzy hint acquisition for the collision avoidance solution of redundant manipulators using neural network," *International Journal of Control, Automation, and Systems*, vol. 4, no. 1, pp. 17-29, 2006.

[3] J. C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, 1993.

[4] R. Al-Hmouz, T. Gulrez, A. Al-Jumaily, "Probabilistic road maps with obstacle avoidance in cluttered dynamic environment," *Proc. of Intelligent sensor, Sensor Networks and Information Processing Conference*, pp. 241-245, 2004.

[5] L. E. Kavraki, P. Svestka, 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, pp. 566-580, 1996.

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

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

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.

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

Jung-Jun Park received the M.S. degree in Mechanical Engineering from Korea University in 2005. He is now a Ph.D. candidate in Mechanical Engineering at Korea University. His research interests include robotic manipulation and a safe robot arm.

Ji-Hun Kim received the M.S. degree in Mechanical Engineering from Korea University in 2006. Since 2006, he has been working for Doosan Infracore as a Researcher. His research interests include robotic manipulation.

Jae-Bok Song received the B.S. and M.S. degrees in Mechanical Engineering from Seoul National University in 1983 and 1985, respectively. Also, he received the Ph.D. degree in Mechanical Engineering from MIT in 1992. He is currently a Professor of Mechanical Engineering, Korea University, where he is also the Director of the Intelligent Robotics Laboratory from 1993. His current research interests lie mainly in mobile robot, design and control of intelligent robotic systems, and haptics.