International Journal of Robotics and Automation (IJRA)
Vol. 3, No. 1, March 2014, pp. 1~12
ISSN: 2089-4856

RSMDP-based Robust Q-learning for Optimal Path Planning in a Dynamic Environment

Yunfei Zhang and Clarence W. de Silva
Industrial Automation Laboratory, Department of Mechanical Engineering
The University of British Columbia, Vancouver, Canada
Email: yfzhang@mech.ubc.ca

Article Info

Article history:

Received Aug 11, 2013
Revised Jan 10, 2014
Accepted Jan 26, 2014

Keyword:

Markov decision process
Online Q-learning
Optimal path planning
Probabilistic roadmap
Regime switching
Unknown dynamic obstacles

ABSTRACT

This paper presents arobust Q-learning method for path planningin a dynamic environment. The method consists of three steps: first, a regime-switching Markov decision process (RSMDP) is formed to present the dynamic environment; second a probabilistic roadmap (PRM) is constructed, integrated with the RSMDP and stored as a graph whose nodes correspond to a collision-free world state for the robot; and third, an onlineQ-learning method with dynamic stepsize, which facilitates robust convergence of the Q-value iteration, is integrated with the PRM to determine an optimal path for reaching the goal. In this manner, the robot is able to use past experience for improving its performance in avoiding not only static obstacles but also moving obstacles, without knowing the nature of the obstacle motion. The use ofregime switching in the avoidance of obstacles with unknown motion is particularly innovative. The developed approach is applied to a homecare robot in computer simulation. The results show that the online path planner with Q-learning is able torapidly and successfully converge to the correct path.

Corresponding Author:

Yunfei Zhang,
Industrial Automation Laboratory, Department of Mechanical Engineering,
The University of British Columbia,Vancouver, BC, V6T 1Z4, Canada
Email: yfzhang@mech.ubc.ca

1. INTRODUCTION

During the past decade, researchers in robotics have increasingly redirected their attention fromtraditional industrial robots operating in structured or stationary environments to the area of service robotics.In this regard, homecare robotics for the elderly and the disabledhas a special significance for its contribution to improving the quality of life and reducing the cost and caregiver burden.

Path planning is a critical function for a homecare robot, which moves in anenvironment of unknown dynamics. In this function, the path of the navigating robot from the start location to the goal location is planned according some criteria (e.g., shortest path, quickest path, path of minimum energy) while avoiding collisions with static and moving obstacles, and subject to some constraints (e.g., robot capabilities with respect to its possible movements). However, path planning becomes more complex in the present application since the home environment is dynamic and unstructured due to moving objects such as humans and pets and their actions (e.g., rearrangement of furniture) [1]-[3]. Therefore, it is important that such robots learn how to deal with dynamic environments so that when they repeat those or similar tasks, advantage can be taken of the prior experience.

Sampling-based methods such as Probabilistic Roadmap [4]-[6] and Rapidly-exploring Random Trees (RRT) [4], [7] became very popular since the nineties. These algorithms have proved to be very effective for path planning in high-dimension since they rely on a collision-checking module instead of explicit representation of the environment; however, convergence to optimal solutions with probability one

cannot be guaranteed. Recently [7], a series of variants to the sampling-based methods (PRM*, RRT*) have been proposed to provide probabilistic completeness. This means the probability of discovering the optimal solution converges to one as the sampled number of states increases to infinity, if a solution exists for a particular plan. All these methods do not explicitly consider moving obstacles when asymptotically converging to the optimal path.

In practical applications, however, most challenges in path planning come from factors of uncertainty in dynamic environments such as varying environment and unknown moving obstacles. Inspired by the consideration of static and moving obstacle separately and the configuration-time state space, Van den Berg proposed a hybrid approach which first constructs a path using PRM in the configuration-time state space based on the stationary obstacles in the environment, and subsequently plans a collision-free path from the original path by taking into account the moving obstacles, which are modeled as time discs [8], [9]. A particular advantage of his approach is that it does not need to exactly know the specific movements of the obstacles, such as speed and direction, and the planner is able to generate a path on-line while taking into account changes in the environment during the period of deliberation. However, his approach also has some disadvantages. One is the assumption that the maximum speed of obstacle motion must be less than that of the robot. Another is that the safety buffer of the time disc module sacrifices part of the collision-free space. Furthermore, the approach does not make use of the previous planning experience, which can help to reduce computational burden.

The present paper presents a new path planning approach, which incorporatesonline reinforcement learning integrated with RSMDP (regime-switching Markov Decision Process) [10] for a mobile robot moving in a dynamic environment. The considered dynamic environment is unstructured and has uncertainty due to lack of knowledge of the behavior of moving obstacles. Hence it is rather difficult to model the surrounding environment and the unpredictable movements of the obstacles. To resolve this problem, a novel framework called the RSMDP scheme is introduced to represent the dynamic environment. In addition, an online reinforcement learning approach is integrated into the RSMDP scheme to resolve the uncertainty in a model-free environment, and PRM (Probabilistic Roadmap)—a sample-based method—is used to resolve the "curse of dimensionality" that arises with reinforcement learning when facing a continuous state and action space. The main contributions of the present paper, in the context of the related previous work ([11], [12]), are as follows. First, unlike [11] and [12], which consider only static obstacles, the present path planner is able to return a globally optimal path in the presence of unknown moving obstacles, with regard to balancing the shortest path and obstacle avoidance. Second, through PRM, both state space and control space can be constrained to a low-dimensional finite space. Third, and most importantly, the reinforcement learning is used in an online formation, using the concept of regime-switching [13],[14] to represent the changing environment caused by moving obstacles, where value iteration is robust to parameter changes. This appears to be the first application of regime switching to solve path planning problems in dynamic and unstructured environment.

The remainder of this paper is organized as follows: after introducing the related work in Section 1, Section 2 formally defines the problem of study. Section 3 briefly introduces the concept of RSMDP and the methods of reinforcement learning and PRM. It rationalizes the combination of these two methods for the application in online path planning. In addition, a path planner is proposed for a mobile robot operating in a dynamic environment, which uses the"experience base" built on-line through robust Q-learning, for generating the optimal path. The developed approach is implemented in a simulated homecare robot. The results presented in Section 4 validate the developed hybrid planner. In Section 5, conclusions are drawn for the present work

2. PROBLEM DESCRIPTION

Consider an autonomous robot that is navigating to a goal location through a complex, dynamic, and unstructured environment in which there are stationary and moving obstacles. The uncertainty comes from the fact thatthe movements of the obstacles cannot be predicted or estimated in advance. It is presumed that the changes of the environment and of the stationary obstacles can be completely known through a global camera. A local camera and laser sensor are able to localize the robot. A moving obstacle may be considered as a static obstacle at the time when it is detected by the sensors of the robotic system, but will not be taken into account during planning path until it is detected again by the sensors. In practice, the autonomous robot is defined in a 6-dimensional (6D) configuration space and in a 2-dimensional (2D) workspace using: three mobile coordinates, x, y, θ_1 and three manipulator coordinates $\theta_2, \theta_3, \theta_4$. So, the entire configuration space has 6 dimensions $s = [x, y, \theta_1, \theta_2, \theta_3, \theta_4]$. Consequently the resulting robot system is a high-dimensional one, which will suffer from the curse of dimensionality. Hence, a sample-based algorithm is used to approximate

the extensive state and action space. In the present paper, for simplicity of description and simulation, a 2D configuration space [*x, y*] is used. Here *x* and *y* represent the two Cartesian axes of the 2D workspace and also the position of the mobile robot in the workspace. A point mass is used to represent the mobile robot instead of considering its kinematic model. The PRM, which is a sample-based method, is used in the present approach, which can be extended to RRT and its variants as well, if a kinematic modelof the robot is integrated. The aim of the present work is find an optimal path plan for a mobile that takes into account not only the shortest length but also the capability of obstacle avoidance under uncertain conditions.

3. METHODOLOGY

3.1. Regime-Switching Markov Decision Process (RSMDP)

The scenario of path planning problem described in Section 2 can be formulated as an MDP because it has the Markov property that the future state depends only the current state and has no dependence on the past states. An MDP is defined by a tuple with five elements: $M = (S,A,P,R,TC)$ where S is a set of states, A is a set of actions depending on X, $P(\cdot|\cdot,\cdot): S\times S\times A\to \square_{\geq 0}$ is a transition probability function that satisfies $\sum_{s'\in S} P(s'|s,a)=1$ for all $s\in S$ and $a\in A$, $R(\cdot,\cdot,\cdot): S\times A\times S\to\square$ is an immediate cost function for all $s\in S$ and $a\in A$,and $TC: S\to\square$ is a terminal cost function denoting the sign of an end of an MDP; an absorbing state with cost zero is usually used in a path planning problem . Whether or not a control policy $\pi: S\to A$ of one MDP is a good process is determined by its corresponding expected value function, which usually can be obtained by solving the *Bellman equation* given by:

$$V^{\pi}(s_t)=E_{\pi}(\sum_{i=0}^{\infty}\gamma^i r_{t+i}\,|s_t)=E_{\pi}(r_t+\gamma\sum_{i=0}^{\infty}\gamma^i r_{t+i+1}\,|s_t)=E_{\pi}(r_t+\gamma V^{\pi}(s_{t+1})) \quad (1)$$

where $\gamma\in[0,1]$ is the step size that corresponds to the iteration rate of the Bellman equation, and $s_{t+1}=\delta(s_t,a_t)$ is the transition function according to $P(\cdot|\cdot,\cdot): S\times S\times A\to\square_{\geq 0}$. The goal is to find an optimal policy $\pi^*(s)=\arg\max_a V^{\pi}(s)$ that minimizes (or maximizes, depending on specific definition of the cost function *R*) the expected value (1) for every initial state s_0. In this paper a sample-based method is applied to overcome the curse of dimensionality. Specifically, the finite horizon discount version of MDP where $i<\infty$, is used for this purpose.

Next regime-switching is integrated into MDP to represent a dynamic environment. Fromexperience it is known that a home environment can change between static and dynamic states. A regime ψ is defined as the time/step period between the last changes and the current changes. Therefore, the state, action and the transition probability of MDP stay the same in one regime and vary from one regime to the next. Consider a countable collection Ψ of changing regimes of cost-minimizing MDP problems. Associate each regime $\psi_k\in\Psi$ with one period of static MDP to be one regime-switching MDP (RSMDP) $M_{\psi_k}=(S_{\psi_k},A_{\psi_k},P_{\psi_k},R_{\psi_k},TC_{\psi_k})$, where $k\in N$ denotes the index of each discrete static time/step period of the changing environment. Consequently, the goal becomes finding the optimal policy $\pi_{\psi_k}{}^*(s)=\arg\max_a V^{\pi}{}_{\psi_k}(s)$ where $V^{\pi}{}_{\psi_k}(s)$ is given by:

$$V^{\pi}{}_{\psi_k}(s_{\psi_k,t})=E_{\psi_k,\pi}(\sum_{i=0}^{\infty}\gamma_{\psi_k}{}^i r_{\psi_k,t+i}\,|s_{\psi_k,t})=E_{\psi_k,\pi}(r_{\psi_k,t}+\gamma_{\psi_k}\sum_{i=0}^{\infty}\gamma_{\psi_k}^{\,j} r_{\psi_k,t+i+1}\,|s_{\psi_k,t})=E_{\psi_k,\pi}(r_{\psi_k,t}+\gamma_{\psi_k}V^{\pi}{}_{\psi_k}(s_{\psi_k,t+1})) \quad (2)$$

It is seen that the optimal policy $\pi_{\psi_k}{}^*(s)$ varies from one regime to another. Hence in RSMDP, the problem of tracking the optimal policy $\pi_{\psi_k}{}^*(s)$ corresponding to each regime ψ_k is considered. The only assumptions that is needed for ψ_k is as follow:

Assumption 1: For each regime $\psi_k\in\Psi, E[T_{\psi}]>>\sup E[t_i]$, where T_{ψ} represents the duration of regime ψ_k and t_i represents the duration of each iteration in the Bellman equation.

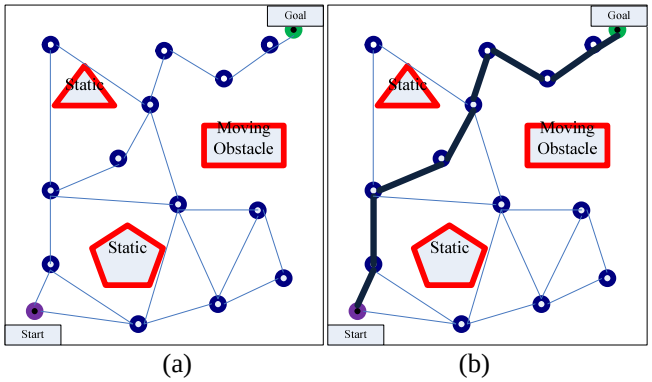
Assumption 1 implies that the requirement of successfully converging to $\pi_{\psi_k}^*(s)$ is that the regime does not change too often when compared with the time used for each iteration step in (2). This is satisfied the practical scenario of path planning that is considered in the present work. In the following subsection, the way to express the path planning problem by incorporating PRM into RSMDP is described.

3.2. Probabilistic Roadmap for RSMDP

A home environment is arguably unstructured. For example, furniture may be cluttered and unorganized, and it is difficult to determine the structure of such furniture using sensors. Furthermore, this will impose a huge computational burden when building an accurate model to represent the environment. Sampling-based methods have adequately resolved the problem of computational burden, because these methods rely on a collision-checking module instead of using an explicit representation of the environment. Probabilistic Roadmap (PRM) and its variants [5], [6], provide effective methods of path planning that are sampling-based.

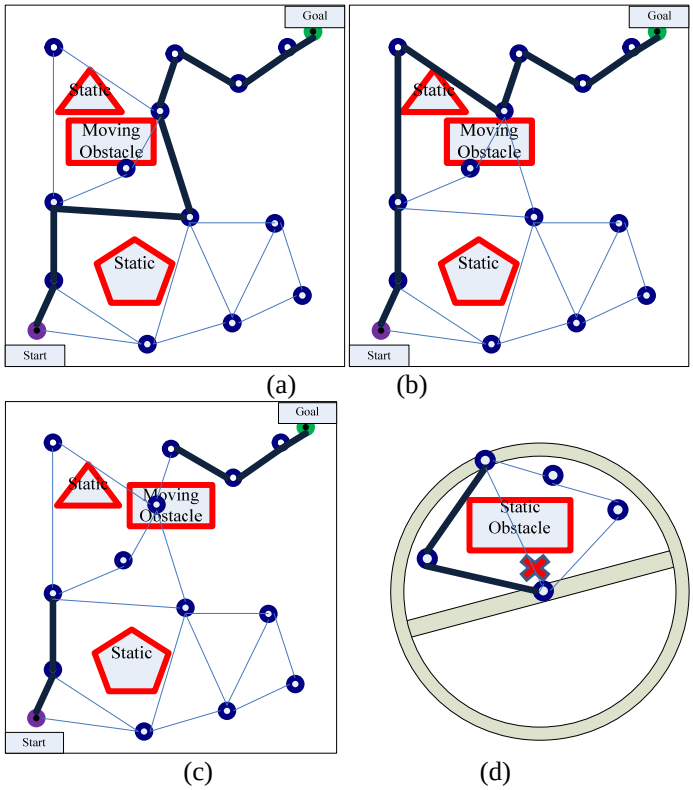
PRM is a network of simple curve segments, or arcs, that meet at nodes. Each node corresponds to a configuration in the configuration space(C-space). Each arc between two nodes corresponds to a collision free path between two configurations. It comprises a preprocessing phase and a query phase. In the following, let *C* denote the robot's C-space, C_f the free C-space, *N*the node set, and *E* the edge set. First, initiate a graph *R* = (*N*, *E*) that is empty. The preprocessing phase constructs the free C-space, giving the global picture, as shown in Figure 1(a). The query phase, shown in Figure 1(b), generates an optimal global collision-free path (bold line in Figure 1(b)) by connecting the start and goal nodes to the roadmap, where heuristic methods are usually used (*Q*-learning will be used in the present paper). Details are found in [5] and [6].

As defined, state set S_{ψ_k} in RSMDP corresponds to node set *N* in PRM, and action set A_{ψ_k} corresponds to edge set *E*. If the system is continuous, index *t* denotes the time interval; otherwise index *t* denotes the step interval. In this paper*t*is considered as the step interval without losing generality since the dynamic environment is formulated as a discrete RSMDP and each step is very short relative to the entire C-space. Therefore, the action subset $\bar{A}_{\psi_k} \in A_{\psi_k}$ associated with a state $s_{\psi_k,i} \in S_{\psi_k}$ at step *i* under regime ψ_k, is countable and corresponds to those edges connecting to the associated node in PRM. Hence which action should be chosen at a certain state in the learning process is governed by a stochastic behavior due to the unpredictable motion of the obstacles, although the available actions are countable. The final goal of the present work is to find the optimal policy $\pi_{\psi_k}^*(s)$ iteratively after the*Q*-learning process, as described in subsection 3.3. Regime ψ_k will not be changed unless the moving obstacles affect the current $\pi_{\psi_k}^*(s)$. Figure 2 shows two common scenarios of regime change where one moving obstacle is detected using some distance threshold. As it blocks the current optimal path, the transition probability (although not known in advance) of the corresponding states and actions will be changed so that the current regime ψ_k will be changed into next regime ψ_{k+1}. Then the *Q*-learning process will choose another available path according to the new regime. Sometimes,once the chosen path is blockedin the current regime, there may notbe available path to choose from due to lack of sampled nodes in PRM. For example in Figure 2(c), an obstacle moves to block all possible paths to the goal node and stays there for a long time. Here again, PRM is imported to build a local roadmap around the robot and the moving obstacle, in order to determine a feasible path. In particular, as shown in Figure 2(d), first a semicircular local region is built centered on the location of the previous state of the blocked edge. Its radius is calculated from the distance between the locations of two states connected by the same blocked edge. Then, a roadmap is generated within this semicircle by the same PRM method as before. Clearly the extra sampled nodes generated by the local roadmap will change the structure of the current PRM and consequently change the current regime ψ_k into the next regime ψ_{k+1}.



(a) (b)

Figure 1. The PRM process in the C-space. (a) Preprocessing stage; (b) Query state (Polygons represent static and moving obstacles, and the blank space represents the collision-free C-space. In order to represent a robot as a point as it moves on the ground, the standard practice is to expand the obstacles corresponding to the size reduction of the robot, as shown by bold sideline of polygons. A uniformly random sample method is used to construct deterministic nodes in the free C-space. Then, such nodes are collision-free nodes. The roadmap is constructed using the collision-free nodes).



(a) (b) (c) (d)

Figure 2. Local roadmap generation: (a)an original path generated by PRM;(b) an alternative pathselected if the original path is blocked; (c)(d)if no alternative path available, a semicircle is used to a generate new collision-free path.

3.3. Path Planner with Online Q-learning

PRM works well in a static environment, but it cannot adapt to a dynamic environment where there are moving obstacles. Re-planning might be an intuitive alternative, in the presence of moving obstacles, but it would be impractical in general. For example, a moving obstacle might rapidly change its position after the path planner re-calculates a path based on the previous sensor information of the moving obstacle, and that new position of the moving obstacle might still block the new path. Considering such problems, the present

paper incorporatesthe reinforcement learning method, *Q*-learning, into the query phase of the PRM in the RSMDP formulation. In this manner, when the optimal path is blocked by a moving obstacle, the path planner is able to quickly choose another optimal path, using the previous experience about the map, as determined by the *Q*function value.

In the 1990s reinforcement learning (RL) was proposed to solve MDP problems [4]. In RL, an agent learns its behavior through trial-and-error interactions with a dynamic environment, while receiving rewards for good actions and penalties for bad actions. Specifically, the agent performs an action a_t in state s_t and receives a real-valued reward $r_t = r(s_t, a_t) \in R$ from the environment. Through this process, the agent learns anoptimal policy $\pi^*(s) = \arg\max_a V^\pi(s)$ where $V^\pi(s)$ is equal to 1, which maps the state set *S* into the action set *A*, and arrives at its next state $s_{t+1} = \delta(s_t, a_t)$. The policy should be able to maximize the cumulative reward according to $V^\pi(s)$.

Q-learning is a popular version of off-policy reinforcement learning which, regardless of the policy being followed, always estimates the optimal *Q*-function that is defined as $Q(s_t, a_t) : S \times A \to$. *Q*-learning has two main advantages when compared with other approaches of reinforcement learning. First, it does not require a model of the environment, which is advantageous when dealing with an unknown environment. For example, in an unknown environment $r_t = r(s_t, a_t)$, $s_{t+1} = \delta(s_t, a_t)$ are nondeterministic functions. Then, *r* and *s*are initiated arbitrarily and the algorithm will eventually converge to the optimal $Q^*(s,a)$ value in view of its mathematical basis. Second, it is able to update the estimates using partially learned estimates without waiting for completing the entire episode, which means it bootstraps. These aspects are discussed under MDP formulation. The core algorithm of *Q*-learning in RSMDP is given by:

$$Q_{\psi_k}(s_t, a_t) \leftarrow Q_{\psi_k}(s_t, a_t) + \gamma_{\psi_k, t}[r_{\psi_k, t}(s_t, a_t) + \alpha_{\psi_k} \max_a Q_{\psi_k}(s_{t+1}, a_{t+1}) - Q_{\psi_k}(s_t, a_t)] \quad (3)$$

The optimal action policy $\pi^*_{\psi_k}$ is given by:

$$\pi^*_{\psi_k}(s) = \arg\max_a Q^*_{\psi_k}(s, a) \quad (4)$$

There are two conditions that should be satisfied to guarantee the convergence of*Q*-learning to optimal $Q^*_{\psi_k}(s,a)$ with probability one [28].

Condition 1: All the state-action pairs $Q(s_t, a_t)$ are visited infinitely often as the number of transitions approaches infinity.

Condition 2: The stepsize γ_t should satisfy $\gamma_{\psi_k, t} > 0, \forall k$, $\sum_{t=0}^{\infty} \gamma_{\psi_k, t} = \infty$, $\sum_{t=0}^{\infty} \gamma^2_{\psi_k, t} < \infty$.

Condition 1 is called exploration, which requires that*Q*-learning has nonzero probability of choosing any action when it also needs to exploit its current knowledge in order to perform well by selecting greedy actions in the current *Q*-function. A popular method to balance exploration with exploitation is the $\varepsilon - greedy$ approach:

$$a_t \leftarrow \begin{cases} an\ uniform\ random\ action\ in\ A_{\psi_k}, with\ probability\ \varepsilon_k \\ a \in \arg\max_a Q_{\psi_k}(s_t, a_t), with\ probability\ 1 - \varepsilon_k \end{cases} \quad (5)$$

Condition 2 implies that the stepsize should meet the requirement $\lim_{t \to 0} \gamma_{\psi_k, t} = 0$. There is also a tradeoff problem when choosing $\gamma_{\psi_k, t}$ in regime ψ_k. In order forQ-learningto converge to optimal $Q^*_{\psi_k}(s,a)$ quickly, the stepsize $\gamma_{\psi_k, t}$ has to be large; however, the stepsize $\gamma_{\psi_k, t}$ should be small in order to minimize the magnitude of the fluctuations of the Q-function within a given regime. In traditional Q-learning, this tradeoff does not considerably affect the system since the speed is usually adequate to solve related problems ina static situation. However, in obstacle avoidance inthe RSMDP framework, the way how Q-function converges to the optimal value greatly affects the robot system.A large $\gamma_{\psi_k, t}$ is expected to produce a fast convergence speed, butthe high-magnitude fluctuationscaused by small $\gamma_{\psi_k, t}$ will lead to incorrect optimal Q-

function, possibly causing the robot to collide with obstacles. At high speed, safety should be given more attention. The strategy to achieve these performance requirements isto make the Q-function iterationprocess robust to $\gamma_{\psi_k,t}$. Then, accurate optimal value can be achieved at satisfactory speed. Therefore, In the current work,by setting and resetting $\gamma_{\psi_k,t} = \gamma_{\max}^t$,the path planner always selects the largest possible stepsize for the current regime and makes it converge to zero within the same regime. But the step size is resetto the largest possible value again when the regime changes. In this way, $\gamma_{\psi_k,t}$ is able to eventually converge to zero, but it is set to a large value in the beginning of the iteration so thatthe Q-function iterationis robust to the changes in $\gamma_{\psi_k,t}$,whilehaving sufficient speed of converging to the new optimal Q-function $Q_{\psi_k}^*(s,a)$ to adapt to the new regime ψ_{k+1} . When to reset $\gamma_{\psi_k,t}$ is critical in the present approach. In view of Assumption 1in subsection3.3, the changing frequency of the dynamic environment should not be very highalthough the moving obstacle may always make the environment to change. To this end, it is assumed that the regime is changed only when the current path has been blocked by obstacles that enter the robot's dangerous area as defined by some threshold, rather than when moving obstacles change the PRM.

The online path planner with*Q*-learning, which is used in the present paper, chooses the optimal path according to the maximum *Q* value with respect to each state-action pair. The*Q* value of each state-action pair is obtained by taking into account both the shortest path and obstacle avoidance in the cost function $r_{\psi_k,t}(s_t,a_t)$ defined by:

$$r_{\psi_k,t}(s_t,a_t) = \omega f(s_t,s_{t+1}) + (1-\omega)h(s_t) \tag{6}$$

Here $f(s_t,s_{t+1}) = \|s_t,s_{t+1}\|_\delta$ denotes a distance function defined by δ norm, ω is the weighting parameter used to balance $f(s_t,s_{t+1})$ and $h(s_t)$, and $h(s_t)$ denotes the reward function according to moving obstacles.

Once an optimal path is obtained in the current regime, the robot begins to move. When a moving obstacle blocks the recurrent optimal path, the stepsize resetting will be made and the path planner will chooses another available optimal path by quickly converging to the new optimal policy. It is seen that although the stepsize changes every time when the regime changes, the learning rate within the new regime will be faster than in the previous regimes since Q-value for each state-action pair is saved as the knowledge for the new regime. That is the reason why the present path planner is able to adapt to a dynamic environment.

4. SIMULATION STUDIES

This section presents simulation studies to validate the online path planner developed in the present paper. The system is simulated using Microsoft Visual Studio 2008 and open-source software OpenCV. Figure 3 shows an image of a simulated environment obtained from a presumed global camera before the mobile robot is started. The white rectangles represent static obstacles and the white ellipse representsa moving obstacle. The world state is made up of the pixel coordinates of the nodes of the roadmap within a 640×480 image plane, which is represented as $s(x,y)$ where x = 0, 1, … , 640; y =0 ,1, 2 …, 480. After obtaining the collision-free states, the starting point and the goal point of the robot are set at fixed positions indicated by arrows. These two points are added into the collision-free roadmap in the same way the roadmap is generated, except that loop connection is used instead of incremental connection in order to provide more possibilities of path connecting between the starting point and the goal point. The thin lines represent available paths and the thicker lines represent optimal paths obtained during the learning process. The number close to each edge is the *Q* value for each state-action pair corresponding to the edge. Euclidian distance is used by setting $\delta = 2, \omega = 10/(640+480)$ in (6) and setting the reward function as follows:

$$h(s_t) = \begin{cases} R = 0; \text{ when the robot reaches the gaol} \\ R = -10; \text{ when the robot touches the obstacle} \\ R = -5; \text{in any other situation.} \end{cases} \tag{7}$$

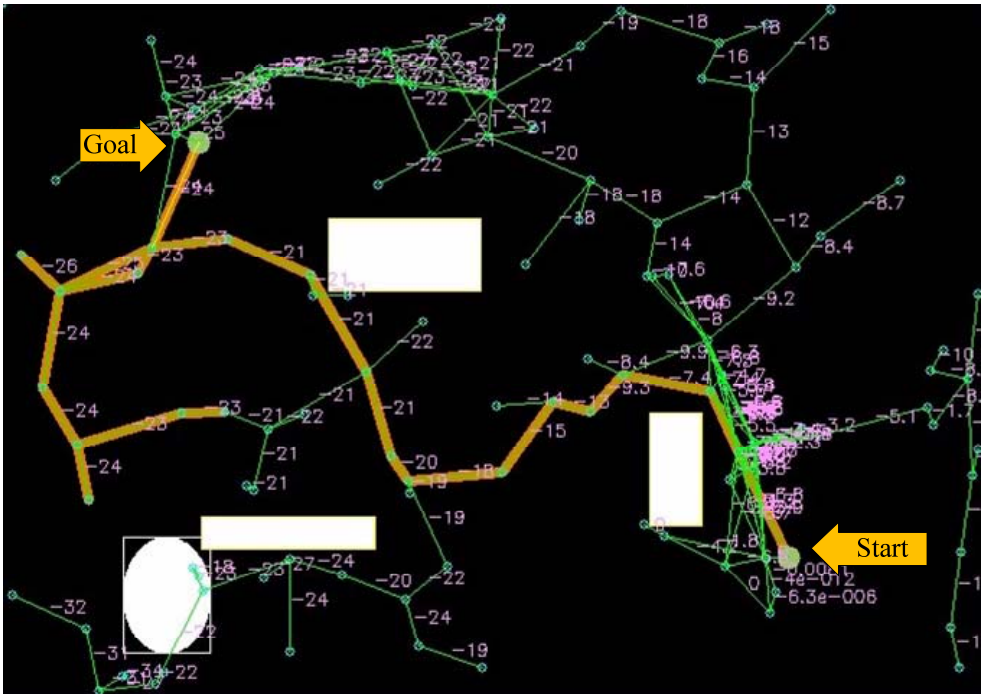


Figure 3. Simulated environment for the mobile robot in a 640×480 image plane, in a learning process.

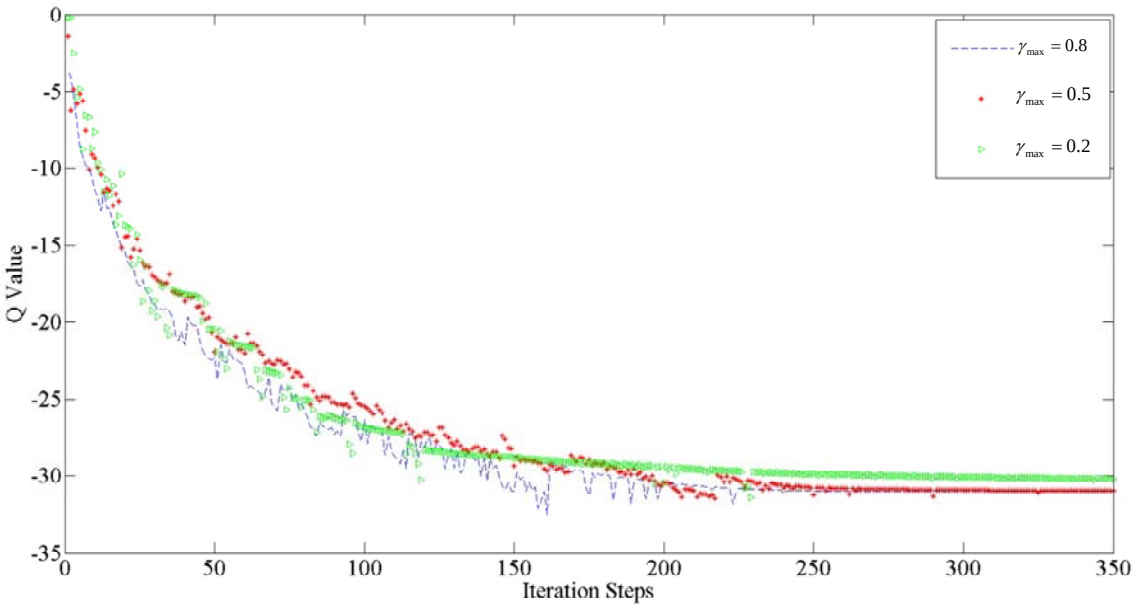


Figure 4. History of *Q*-valuewith $\gamma_{max} = 0.8, 0.5, 0.2$ and 100 sampled points in PRM.

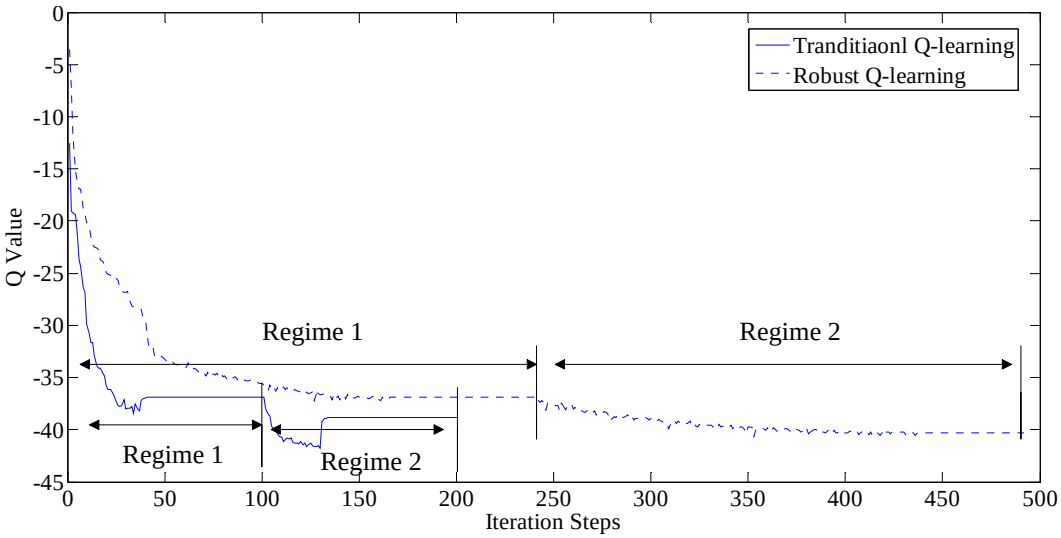


Figure 5. Performance of traditional Q-learning and Robust Q-learning with $\gamma_{\max} = 0.98$.

The first simulation (shown in Figure 4) tests how the*Q* value converges to its optimal value in terms of possible maximum beginning value ofthe stepsize when using the strategy $\gamma_{\psi_k,t} = \gamma_{\max}^t$. The *Q* value of the edge close to the goal point is chosen under three beginning step-size values, $\gamma_{\max} = 0.8, 0.5, 0.2$ (shown by the dotted line, star line and triangle line,respectively, in the figure). It is seen that the smallest value $\gamma_{\max} = 0.2$ can bring up less intense fluctuations, which is in accordance with Condition 2 in subsection3.3. Nevertheless, it is seen that the optimal Q-value and the learning rate do not change appreciably for different maximum beginning values of the stepsize $\gamma_{\max}$. In this sense, the Q-value iteration process is robust for choosing $\gamma_{\max}$ .Specifically, $\gamma_{\max}$ can be chosen as high as possible in order to obtain the fastest possible learning rate or the fastest convergence speed. Figure 5 shows the performance of the iteration process when $\gamma_{\max}$ is set at 0.98 for robust Q-learningand traditional Q-learning. In Figure 5 it is seen that both robust Q-learning(dotted line) and traditional Q-learning (solid line) are able tosuccessfully converge to a new optimal Q-value when the regime changes from regime 1 into regime 2, caused by moving obstacles. Although tranditional Q-leanring is faster than robust Q-learning, robust Q-learning converges to the optimal Q-value very smoothly comparing with the traditional Q-learning. This propertywill help the robot to choose a more accurateoptimal action if the optimal Q-values of the actions with respect to one state intersect with each other when the regime changes quickly and there is not enough time for the Q-value iteration to converge to the optimal value.Therefore, the safety can be ganranteed as much as possible at the cost of low speed of convergence. This low speed is still the fastest one in such conditions and can be achieved using a modern computer.

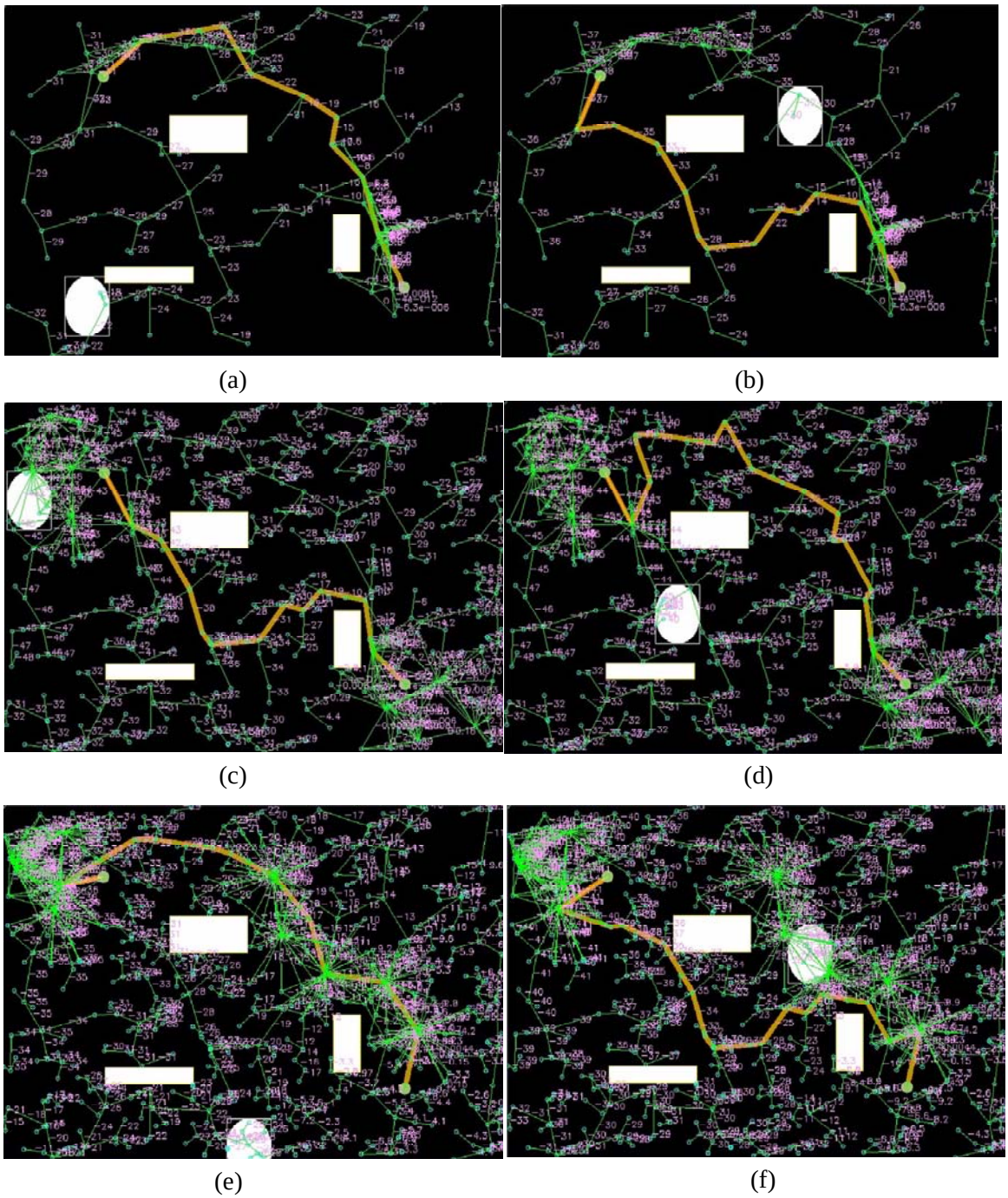


Figure 6. Obstacle avoidance in dynamic environment. (a), (c), (e) original optimal path in regime 1; (b), (d), (f) scenarios for regime 2 where moving obstacle is blocking the current path, hence choose another optimal path. The top row of the pictures corresponds to 100 sampled points in PRM, center row corresponds to 400 sampled points, and the bottom row corresponds to 600 sampled points.

The second simulation (shown in Figure 6) verifies that the present algorithm is able to successfully avoid both static and moving obstacles under the RSMDP and robust Q-learning framework. Keyboard controller is used to control the moving obstacle and make it move to block the obtained optimal path. With the cost function as defined in (6) and (7), the simulated robot reaches the goal point by choosing the shortest available path and avoiding obstacles, which is considered as regime 1. When the robot detects that the moving obstacle is blocking its current optimal path, it quickly finds another optimal path by using the

learning experience, which is considered as regime 2. It is noted that although increasing the number of PRM nodes will generate more available paths, the time spent to learn a new optimal path will also increase. Hence, there is a tradeoff between the number of nodes and the time taken to avoid obstacles. In the present case, having 400 sampled nodes can provide the fastest speed to adapt to a dynamic environment.
The online robust Q-learning method of the present paper is a behavior-based decision-making process. The robot continuously observes the world states and selects the action having the optimal *Q* value among the possible actions in the current state, as given by the *Q* function. This is different from a traditional behavior-based system where the rule base of behavior is designed entirely by a human expert in advance. The rule base of Q-learning is learned autonomously when the robot interacts with its environment during the training process. The curse of dimensionality is a serious challenge in this process because, in theory, an infinite number of iterations would be needed to guarantee convergence to the optimal value. The method proposed in the paper overcomes this problem by incorporating a PRM roadmap as the world state for the robot. The safety is another challenge, when dealing with rapidly moving obstacles in a dynamic environment. The robust Q-learning in the present paper guarantees smooth convergence for Q-value iteration so that arelatively accurate action is chosen when the regime changes.

5. CONCLUSION

This paper presented anonline robust Q-learning path planning method for a mobile robot in a dynamic environment with unpredictable moving obstacles. Dynamic stepsize strategy in online robust Q-learning is central when using RSMDP to represent a dynamic environment. This strategy makes the Q-value iteration robust to the choice of maximum initial stepsize. PRM contributes to overcoming the curse of dimensionality which is a common problem in MDP and reinforcement learning. Simulation experiments presented in the paper showed that the developed path planner could rapidly and safely find an optimal path in a dynamic environment with both static and moving obstacles. The method presented in the paper can be directly extended to robots represented by kinematic models and in higher dimensions since a continuous space may be presented by an infinite discrete space.

ACKNOWLEDGEMENTS

This work has been supported by research grants from the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Canada Foundation for Innovation (CFI), the British Columbia Knowledge Development Fund (BCKDF), and the Canada Research Chair in Mechatronics and Industrial Automation held by C.W. de Silva.

REFERENCES

[1] H Choset, et al. *Principles of Robot Motion: Theory, Algorithms, and Implementation.* MIT Press, June 2005.
[2] SM LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, 2006.
[3] J Pineau, et al. "*Probablistic control of human robot interaction: Experiments with a robotic assistant for nursing homes*". Proceeedings of the 2nd IARP/IEEE-RAS Joint Workshop on Technical Challenge for Dependable Robots in Human Environments. 2002: 11-19.
[4] R Luna, et al. "*Anytime Solution Optimization for Sampling-Based Motion Planning*". In Proceedings of the IEEE International Conference in Robotics and Automation, 2013. To be published.
[5] L Kavraki, et al. "Analysis of probabilistic roadmaps for path planning". *IEEE Transactions on Robotics and Automation.* 1998; 14: 166-171.
[6] L Kavraki, et al. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". *IEEE Transactions on Robotics and Automation*. 1996; 12: 566-580.
[7] S Karaman, et al. "Anytime motion planning using the RRT*". *IEEE Conference on Robotics and Automation.* 2011.
[8] J van den Berg et al. "*Anytime Path Planning and Replanning in Dynamic Environments*". Proceedings of IEEE International Conference on Robotics and Automation. 2006; 2366-2371.
[9] J van den Berg and M Overmars. "Planning Time-Minimal Safe Paths Amidst Unpredictably Moving Obstacles". *International Journal of Robotics Research.* 2008; 27(11-12): 1274-1294.
[10] RS Sutton and AG Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, 1998.
[11] R Alterovitz, et al. "*The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty*". Proceedings of Robotics: Science and Systems. June 2007.
[12] V Huynh, et al. "An incremental sampling based algorithm for stochastic optimal control". *In IEEE Conference on Robotics and Automation*. 2012.
[13] G Yin, et al. "Regime switching stochastic approximation algorithms with application to adaptive discrete stochastic optimization". *SIAM Journal on Optimization.* 2004; 14(4): 1187-1215.

[14] A Costa and FJ Vázquez-Abad. "Adaptive stepsize selection for tracking in a regime-switching environment". *Automatica*. 2007; 43(11): 1896-1908.
[15] R Brooks and T Lozano-Perez. "*A subdivision algorithm in configuration space for findpath with rotation*". Proceedings of International Joint Conference on Artificial Intelligence. 1983; 799-806.
[16] O Khatib. "Real-time obstacle avoidance for manipulators and mobile robots". *The International Journal of Robotics Research*. 1986; 5: 90–98.
[17] Y Koren and J Borenstein. "Potential field methods and their inherent limitations for mobile robot navigation". *IEEE Conference on Robotics and Automation*. 1991; 2: 1398-1404.
[18] L Zeng and GM Bone. "Mobile Robot Navigation for Moving Obstacles with Unpredictable Direction Changes, Including Humans". *Advanced Robotics*. 2012; 26(16): 1841-1862.
[19] L Kavraki and JC Latombe. "Randomized preprocessing of configuration space for fast path planning". *IEEE International Conference on Robotics and Automation*. 1994; 3: 2138-2145.
[20] F Von Hundelshausen, et al. "Driving with tentacles- Integral structures of sensing and motion". *Journal of Field Robotics*. 2008; 25: 640–673.
[21] S Quinlan and O Khatib. "Elastic bands: connecting path planning and control". *IEEE Conference on Robotics and Automation*. 1993.
[22] J Minguez, et al. "Motion planning and obstacle avoidance". *Springer Handbook of Robotics*. B. Siciliano, O. Khatib (Eds.), Springer. 2008: 827–852.
[23] T Wada, et al. "A deceleration control method of automobile for collision avoidance based on driver's perceptual risk". *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2009; 4881-4886.
[24] A Ohya, et al. "Vision-based navigation by a mobile robot with obstacle avoidance using a single-camera vision
[25] F. Lamiraux, et al. "Reactive path deformation for nonholonomic mobile robots". *IEEE Transactions on Robotics*. 2004; 20: 967–977.
[26] L Lapierre, et al. "Simultaneous path following and obstacle avoidance control of a unicycle-type robot". *IEEE Conference on Robotics and Automation*, 2007.
[27] MW Spong, et al. *Robot Modeling and Control*. Wiley Press, New York. 2006.
[28] DP Bertsekas. *Dynamic Programming and Optimal Control, Vol. II, 4th Edition: Approximate Dynamic Programming*. Belmont: Athena Scientific, 2012.