Robust Quantum-Inspired Reinforcement Learning for Robot Navigation

Daoyi Dong, *Member, IEEE*, Chunlin Chen, *Member, IEEE*, Jian Chu, and Tzyh-Jong Tarn, *Life Fellow, IEEE*

***Abstract*—A novel quantum-inspired reinforcement learning (QiRL) algorithm is proposed for navigation control of autonomous mobile robots. The QiRL algorithm adopts a probabilistic action selection policy and a new reinforcement strategy, which are inspired, respectively, by the collapse phenomenon in quantum measurement and amplitude amplification in quantum computation. Several simulated experiments of Markovian state transition demonstrate that QiRL is more robust to learning rates and initial states than traditional reinforcement learning. The QiRL approach is then applied to navigation control of a real mobile robot, and the simulated and experimental results show the effectiveness of the proposed approach.**

***Index Terms*—Probabilistic action selection, quantum amplitude amplification, quantum-inspired reinforcement learning (QiRL), robot navigation.**

I. Introduction

With the rapid development of artificial intelligence and quantum technology [1], some scientists have begun to explore the connection and interaction between quantum mechanics and artificial intelligence. For example, Benioff [2] presented the concept of quantum robots in 1998, where a quantum robot is described as a mobile quantum system that includes an on-board quantum computer and needed ancillary systems. Dong *et al*. [3] proposed a structure for quantum robots from the perspective of engineering, where quantum robots can interact with the external environment by sensing and processing information. Several ideas from quantum computation have been fused into computational intelligence and several quantum or quantum-inspired intelligent learning algorithms have been proposed. For example, the neural network version based on quantum computation has been studied from pure theory to simple architecture [4], [5]. Rigatos and Tzafestas [6] have proposed using quantum computation to speed up the fuzzy inference through the parallelization of a fuzzy logic control algorithm. Quantum characteristics have been used to improve the existing evolutionary algorithms and quantum-inspired evolutionary algorithms have been applied to several optimization problems such as the knapsack problem [7], [8]. Dong *et al*. [9] have presented the concept of quantum reinforcement learning (QRL) through the combination of quantum computation and reinforcement learning, and have also applied it to a learning control problem of quantum systems [10]. Following these results, in this paper, we will propose a novel quantum-inspired reinforcement learning (QiRL) algorithm, where a probabilistic action selection policy is inspired by the collapse phenomenon in quantum measurement (e.g., see [1] and [9]) and a new reinforcement strategy is inspired by amplitude amplification in quantum computation (e.g., see [11], [12], and [13]). The proposed QiRL approach in this paper is one of a series of results [3], [9], [10] for exploring quantum or quantum-inspired intelligent algorithms. It is worth noting that the focus of QiRL is greatly different from QRL in [9]. In QRL, we expect to explore real quantum learning algorithms from computation essence and learning mechanism, and the algorithm realization depends on practical quantum computers. However, QiRL only borrows several ideas from quantum computation and is essentially a classical learning algorithm. It can be directly used to accomplish some practical learning control tasks without the requirement of quantum computers [14]. In particular, we will apply QiRL to the navigation control of autonomous mobile robots in unknown environments.

Autonomous mobile robots are an important testbed for the research of artificial intelligence and have also been applied to many industrial and service areas. The navigation control is a fundamental task for autonomous mobile robots [15]–[19]. The methods of robot navigation can generally be classified into two classes: local navigation [20] and global navigation [21]. Local navigation (also called reactive control) is the fundamental ability for mobile robots, where the robots learn the local paths using the current sensory inputs without *a priori* complete knowledge of the environment. Several reactive control approaches [20], [22], [23] have been proposed and implemented for local navigation tasks such as obstacle avoidance, wall-following, wandering, and point to point moving. To achieve a better control performance, it is desirable to develop effective machine learning

Manuscript received May 25, 2010; revised August 9, 2010; accepted September 18, 2010. Date of publication December 17, 2010; date of current version January 9, 2012. Recommended by Technical Editor P. X. Liu. This work was supported in part by the National Creative Research Groups Science Foundation of China under Grant 60721062, in part by the National Natural Science Foundation of China under Grant 60703083 and Grant 60805029, in part by the Fundamental Research Funds for the Central Universities under Grant 2010QNA5014, and in part by the Australian Research Council under Grant DP1095540.

D. Dong is with the Institute of Cyber Systems and Control, State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China, and also with the School of Engineering and Information Technology, University of New South Wales at the Australian Defence Force Academy, Canberra, ACT 2600, Australia (e-mail: daoyidong@gmail.com).

C. Chen is with the Department of Control and System Engineering, School of Management and Engineering, Nanjing University, Nanjing 210093, China (e-mail: clchen@nju.edu.cn).

J. Chu is with the Institute of Cyber Systems and Control, State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China (e-mail: chuj@supcon.com).

T.-J. Tarn is with the Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO 63130 USA (e-mail: tarn@wuauto.wustl.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2010.2090896

methods for the reactive control of mobile robots. So far, several learning approaches have been applied to robot navigation regarding the architecture [24], learning methods [25], and specific skills [26]. Among all these approaches, reinforcement learning has been proven to be an effective method for robot navigation [15], [27]. However, it is usually sensitive to learning rates and initial settings. Moreover, reinforcement learning learns slowly for robot navigation in complex unknown environments. In this paper, we propose a novel QiRL algorithm and demonstrate, using an example of state transition, that QiRL is more robust to learning rates and initial settings than traditional reinforcement learning. Hence, it is a useful approach for the navigation control of autonomous mobile robots, which is also shown through simulated and experimental results.

This paper is organized as follows. Section II briefly introduces reinforcement learning, presents the quantized action representation, and proposes a quantum-inspired probabilistic action selection policy. Section III proposes a new reinforcement strategy inspired by amplitude amplification in quantum computation. A QiRL algorithm and its performance illustration are presented in Section IV. Section V applies QiRL to the navigation control of an autonomous mobile robot and the simulated and experimental results demonstrate the effectiveness of QiRL. Concluding remarks are given in Section VI.

II. Quantum-Inspired Probabilistic Action Selection

A. Reinforcement Learning

Reinforcement learning (RL) is an important kind of machine learning method [28]–[36]. It learns through trial and error and uses a scalar-value-named reward to evaluate the input–output pairs. Its goal is to learn a mapping from states to actions through interaction with environments. RL algorithms assume that the state set S and action set $A_{(s_n)}$ for state s_n (i.e., $S = \{s_1, s_2, \ldots, s_n, \ldots\}$) can be divided into discrete values. At a certain step t, the agent observes the state of the environment s_t, and then chooses an action a_t. After executing the action, the state of the environment will change into next state s_{t+1} and the agent receives a reward r_{t+1}, which reflects how good that action is (in a short-term sense). The agent will choose the next action a_{t+1} according to related knowledge.

The goal of reinforcement learning is to learn a mapping from states to actions. In other words, the agent is to learn a policy $\pi : S \times \cup_{s\in S} A_{(s)} \rightarrow [0, 1]$, so that the expected sum of discounted rewards of each state will be maximized

$$\begin{aligned} V^{\pi}_{(s)} &= E\{r_{(t+1)} + \gamma r_{(t+2)} + \cdots | s_t = s, \pi\} \\ &= E[r_{(t+1)} + \gamma V^{\pi}_{s_{(t+1)}} | s_t = s, \pi] \\ &= \sum_{a\in A_s} \pi(s, a)[r^a_s + \gamma \sum_{s'} p^a_{ss'} V^{\pi}_{(s')}] \end{aligned} \tag{1}$$

where $\gamma \in [0, 1]$ is a discount factor, $\pi(s, a)$ is the probability of selecting action a according to state s under policy π, $p^a_{ss'} = \Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$ is the probability for state transition, and $r^a_s = E\{r_{t+1} | s_t = s, a_t = a\}$ is the expected one-step reward. $V_{(s)}$ (or $V(s)$) is also called the value function of state s. With different definitions and updating of value functions, two classes of important reinforcement learning methods including the temporal difference (TD) algorithm [28], [29] and Q-learning [30] have been studied in depth. The widely used TD one-step updating rule of $V(s)$ (for details, see, e.g., [28] and [29]) may be described as

$$V(s) \leftarrow V(s) + \eta(r + \gamma V(s') - V(s)) \tag{2}$$

where $\eta \in (0, 1]$ is the learning rate. We have the optimal state-value function

$$V^*_{(s)} = \max_{a\in A_s} \left[r^a_s + \gamma \sum_{s'} p^a_{ss'} V^*_{(s')} \right] \tag{3}$$

$$\pi^* = \arg\max_{\pi} V^{\pi}_{(s)}, \forall s \in S. \tag{4}$$

B. Quantized Action Representation

Since the 1980s, RL has widely been used in machine learning and intelligent control, especially in robotics [37]. In practical applications, reinforcement learning is still confronted with some difficulties, such as the balancing of exploration and exploitation, slow learning speeds in complex unknown environments, and reinforcement strategies in probabilistic circumstances. Several methods have been proposed to combat those difficulties or improve the performance of reinforcement learning. For example, temporal abstraction and decomposition method has been explored to speed up learning [33]. The adaptation of Q-learning with fuzzy inference systems for problems with large state-action spaces or with continuous state spaces has also been proposed [31], [34], [37]. In spite of all these attempts, more new ideas are desirable for the exploration of more effective reinforcement strategies and learning mechanisms. In this paper, we will explore novel RL algorithms inspired by some ideas from quantum mechanics. To explore QiRL, first we will represent actions into the form of quantum superposition state (e.g., see [1]).

In quantum theory, the state of a closed quantum system is described by a quantum state $|\Phi\rangle$ (Dirac representation). $|\Phi\rangle$ is a unit vector in Hilbert space and can be expanded by a set of orthogonal bases $\{|\phi_n\rangle\}$ of a Hermitian operator (e.g., the free Hamiltonian of the closed quantum system), i.e.,

$$|\Phi\rangle = \sum_n \beta_n |\phi_n\rangle \tag{5}$$

where β_n are complex coefficients (generally called probability amplitudes) satisfying $\sum_n |\beta_n|^2 = 1$. To present the quantized representation of actions in RL, we denote an action a_n in RL as the corresponding orthogonal quantum state $|a_n\rangle$ ($|a_n\rangle$ is called an eigenaction). According to the superposition principle in quantum theory, the superposition of $|a_n\rangle$ is also a reasonable quantum state, which inspires us to represent all actions a_n (or eigenactions $|a_n\rangle$) using a quantum state $|qa\rangle$ for every state in RL, i.e.,

$$|qa\rangle = \sum_n \alpha_n |a_n\rangle \tag{6}$$

where α_n is probability amplitude satisfying $\sum_n |\alpha_n|^2 = 1$. It is worth mentioning that $|qa\rangle$ is only a representation method

and is not a practical action. The approach of quantized action representation establishes a connection between a quantum state and the action set. That is, the action set for every state in traditional reinforcement learning can correspond to a quantum state $|qa\rangle$, where eigenstates $|a_n\rangle$ in $|qa\rangle$ correspond to actions a_n in tradition RL. The quantized representation provides the possibility of applying quantum amplitude amplification as a reinforcement strategy (for details, see Section III).

C. Quantum-Inspired Probabilistic Action Selection

Action selection policy is a central issue in the design of high intelligent-behavior-based agents. In RL, the agent generally chooses actions according to accumulated rewards. One widely used action selection scheme is ε-greedy, where the optimal action is selected with probability $(1-\varepsilon)$ and a random action is selected with probability ε $(0<\varepsilon<1)$. The exploration probability ε can be reduced over time, during which the agent moves from exploration to exploitation. The ε-greedy method is simple and effective but it has one drawback in that it chooses equally among all actions when exploring. This means that it does not distinguish between choosing the worst action and choosing the next-to-best action. Another problem is that it is difficult to choose a proper parameter ε which can offer an optimal balancing between exploration and exploitation. In addition, as for the ε-decreasing strategy, it is more difficult to properly reduce the parameter ε along with the learning process.

Another kind of action selection scheme is Boltzmann exploration (including the Softmax action selection method) [28]. It uses a positive parameter τ called the temperature and chooses action with the probability proportional to $e^{Q(s,\,a)/\tau}$, where $Q(s,\,a)$ is the state-action value (the expected sum of discounted rewards of state s if the action a is chosen; for details, see [30]). It can move from exploration to exploitation by adjusting the "temperature" parameter τ. It is natural to sample actions according to this distribution, but it is very difficult to set and adjust a good parameter τ. Hence, it is desirable to develop new approaches for action selection in RL. Here, we introduce a new action selection policy inspired by the collapse phenomenon in quantum measurement.

In quantum mechanics, the measurement on a quantum system in $|\Phi\rangle=\sum_n \beta_n|\phi_n\rangle$ means that one chooses a corresponding observable $\hat{O}$ (a Hermitian operator) and executes a measurement by a set of measurement operators (for details, see [1]). If we select the same measurement bases as $\{|\phi_n\rangle\}$, after executing a measurement, the system state will change from $|\psi\rangle$ into $|\phi_n\rangle$ with probability $|\beta_n|^2$. This process is known as the collapse phenomenon in quantum measurement. In QiRL, the agent is to learn a policy $\pi: S\times\cup_{s\in S}A_{(s)}\rightarrow[0,1]$, which will maximize the expected sum of discounted reward of each state. That is to say, the mapping from states to actions is $f(s)=\pi: S\rightarrow A$. Corresponding to the action representation $|qa\rangle$, a quantum-inspired action selection process can be represented as

$$f(s)=\frac{|a_1\rangle}{|\alpha_1|^2}+\frac{|a_2\rangle}{|\alpha_2|^2}+\cdots=\sum_n\frac{|a_n\rangle}{|\alpha_n|^2}. \tag{7}$$

In (7), the expression is not for numerical computation and just means that at the state s (whose action set is $\{|a_1\rangle,\ldots,|a_n\rangle\}$), the agent will choose the eigenaction $|a_n\rangle$ with probability $|\alpha_n|^2$. This quantum-inspired action selection approach is a probabilistic action selection policy where the representation (7) provides an intuitive explanation on how to determine the probability of action selection. The specific value of the corresponding probability can be calculated using (6), whose updating is realized by quantum amplitude amplification operations which provide a mechanism for probabilistic action selection.

III. Reinforcement Strategy Using Quantum Amplitude Amplification

In reinforcement learning, when the reward and value function show that the corresponding action is "good", we should boost the selection probability of this action; i.e., reinforce good decision. In quantum computation, quantum amplitude amplification has derived success for several quantum algorithms with quadratic speedup. Intuitively, quantum amplitude amplification can amplify the amplitude of some appointed components and increase the probability of success roughly by a constant on each iteration, which is analogous to the iteration process in classical probabilistic algorithms [12]. Hence, it is possible to use the idea of quantum amplitude amplification as a new reinforcement strategy.

A. Quantum Amplitude Amplification

Quantum amplitude amplification is a useful approach for powerful quantum algorithms [12], [38]. It is a natural generalization of Grover's quantum searching algorithm that allows a speedup of several classical algorithms [12], [38]–[43]. The idea of amplitude amplification was first discovered by Brassard and Høyer [39]. The central task of quantum amplitude amplification is to find a suitable operator $\mathbf{Q}$ defined similarly to the Grover iteration operator (for details, see, e.g., [11], [40], and [42]).

Let $|\Phi\rangle$ be a quantum state and it can be represented as a superposition of orthonormal states $\mathbb{X}=\{|1\rangle,\ldots,|x\rangle,\ldots,|N\rangle\}$ in N-dimensional Hilbert space $\mathscr{H}$; i.e., $|\Phi\rangle=\sum_{x=1}^{N}c_x|x\rangle$, where $\sum_{x=1}^{N}|c_x|^2=1$. A Boolean function $\chi:\mathbb{X}\rightarrow\{0,1\}$ induces two orthogonal subspaces of $\mathscr{H}$: "good" subspace and "bad" subspace. The good subspace is spanned by the set of basis states $|x\rangle\in\mathbb{X}$ satisfying $\chi(x)=1$ and the bad subspace is its orthogonal complement in $\mathscr{H}$. Every quantum state $|\Phi\rangle$ in $\mathscr{H}$ can be decomposed as $|\Phi\rangle=|\Phi_g\rangle+|\Phi_b\rangle$, where $|\Phi_g\rangle$ denotes the projection of $|\Phi\rangle$ onto the good subspace and $|\Phi_b\rangle$ denotes the projection of $|\Phi\rangle$ onto the bad subspace. According to quantum theory, the occurrence probabilities of "good" state $|x\rangle$ $(\chi(x)=1)$ and "bad" state $|x\rangle$ $(\chi(x)=0)$ are $g=\langle\Phi_g|\Phi_g\rangle$ (assuming $g\neq 0$) and $b=\langle\Phi_b|\Phi_b\rangle$, respectively.

Let $\mathscr{U}$ be any quantum algorithm that acts on $\mathscr{H}$ without measurements and $|\Phi\rangle=\mathscr{U}|1\rangle$. Given two angles $0\leq\varphi_1,\varphi_2<\pi$, the general quantum amplitude amplification can be realized by the following operator [12]:

$$\mathbf{Q}=\mathbf{Q}(\mathscr{U},\chi,\varphi_1,\varphi_2)=-\mathscr{U}\mathscr{P}_1^{\varphi_1}\mathscr{U}^{-1}\mathscr{P}_\chi^{\varphi_2}. \tag{8}$$

The operators $\mathscr{P}_1^{\varphi_1}$ and $\mathscr{P}_\chi^{\varphi_2}$ conditionally change the phase of the amplitudes of state $|1\rangle$ and the good states, respectively [12]. They can be expressed into

$$\mathscr{P}_1^{\varphi_1} = I - (1 - e^{\iota\varphi_1})|1\rangle\langle 1| \tag{9}$$

$$\mathscr{P}_\chi^{\varphi_2} = I - \frac{1}{g}(1 - e^{\iota\varphi_2})|\Phi_g\rangle\langle\Phi_g| \,. \tag{10}$$

where $\iota = \sqrt{-1}$. The action of $\mathbf{Q}$ can be described as the following lemma [11], [12]:

Lemma 1: Let $\mathscr{U}|1\rangle = |\Phi\rangle = |\Phi_g\rangle + |\Phi_b\rangle$ and $g = \langle\Phi_g|\Phi_g\rangle$. Then

$$\mathbf{Q}|\Phi\rangle = [(1 - e^{\iota\varphi_1})(1 - g - ge^{\iota\varphi_2}) - e^{\iota\varphi_2}]|\Phi_g\rangle + [g(1 - e^{\iota\varphi_1})(e^{\iota\varphi_2} - 1) - e^{\iota\varphi_1}]|\Phi_b\rangle. \tag{11}$$

From Lemma 1, we can amplify (or shrink) the amplitude of $|\Phi_g\rangle$ (or $|\Phi_b\rangle$) by suitable selection of φ_1, φ_2 in $\mathbf{Q}$. To make this clearer, we consider the special case $\varphi_1 = \varphi_2 = \pi$ (i.e., $\mathbf{Q} = \mathbf{Q}(\mathscr{U}, \chi, \pi, \pi)$). If $\mathscr{U}|\mathbf{1}\rangle = |\Phi\rangle = |\Phi_g\rangle + |\Phi_b\rangle$, for all $L \geq 0$, we have

$$\mathbf{Q}^L\mathscr{U}|\mathbf{1}\rangle = \frac{1}{\sqrt{g}}\sin((2L+1)\theta)|\Phi_g\rangle + \frac{1}{\sqrt{b}}\cos((2L+1)\theta)|\Phi_b\rangle \tag{12}$$

where $b = 1 - g$, θ satisfies $\sin^2\theta = g$, and $0 < \theta \leq \pi/2$.

Lemma 1 provides a method for boosting the success probability. In particular, for $\mathbf{Q} = \mathbf{Q}(\mathscr{U}, \chi, \pi, \pi)$, it can boost the success probability from $g = \sin^2\theta$ to $g' = \sin^2((2L+1)\theta)$ by applying $\mathbf{Q}$ for L times.

B. Quantum-Inspired Reinforcement Strategy

From Lemma 1, we know that quantum amplitude amplification can boost the success probability g of a quantum algorithm through constructing a suitable operator $\mathbf{Q}$. Now we apply the idea as a new reinforcement strategy for reinforcement learning. Assume the number of eigenactions for state s is N and the set of eigenactions is represented with corresponding N orthonormal states $\mathbb{A} = \{|a_1\rangle, \ldots, |a_x\rangle, \ldots, |a_N\rangle\}$ in N-dimensional Hilbert space $\mathscr{H}$. Hence $|qa\rangle = \sum_{x=1}^N \alpha_x|a_x\rangle$, where $\sum_{x=1}^N |\alpha_x|^2 = 1$. A Boolean function $\chi : \mathbb{A} \to \{0, 1\}$ induces two orthogonal subspaces of $\mathscr{H}$: "good" subspace and "bad" subspace. The good subspace is spanned by the set of basis states $|a_x\rangle \in \mathbb{A}$ satisfying $\chi(a_x) = 1$ and the bad subspace is its orthogonal complement in $\mathscr{H}$. Every $|qa\rangle$ in $\mathscr{H}$ can be decomposed as $|qa\rangle = |qa_g\rangle + |qa_b\rangle$, where $|qa_g\rangle$ denotes the projection of $|qa\rangle$ onto the good subspace and $|qa_b\rangle$ denotes the projection of $|qa\rangle$ onto the bad subspace. Hence, the selection probabilities of "good" actions $|a_x\rangle$ ($\chi(a_x) = 1$) and "bad" actions $|a_x\rangle$ ($\chi(a_x) = 0$) are $g = \langle qa_g|qa_g\rangle$ and $b = \langle qa_b|qa_b\rangle$, respectively.

In RL, the agent is to learn a mapping from states to actions $f(s) = \pi : S \to A$. Based on the quantized action representation, $|qa\rangle = \sum_{x=1}^N \alpha_x|a_x\rangle$, the agent will use (7) to select an action $|a_x\rangle$ with the probability $|\alpha_x|^2$. That is, the agent may select any one eigenstate $|a_x\rangle$ in the action set $\{|a_x\rangle,\ x = 1, \ldots, N\}$ and the probability that $|a_x\rangle$ (corresponding to the action a_x in RL) is selected is $|\alpha_x|^2$. After executing an action a_x (corresponding to $|a_x\rangle$), the agent will get a reward. Then, it can amplify the probability of "good" action using (6) according to corresponding rewards. That is, the agent can change the probability $|\alpha_x|^2$ by changing the amplitude α_x. This process may correspond to a quantum amplitude amplification operation. First, let the "good" action correspond to $|\Phi_g\rangle$. We can carry out $\mathbf{Q}$ for certain times M to update the probability amplitudes according to respective rewards and value functions. Usually, it is related to specific $\mathbf{Q}$, reward r, and value function V. For example, we may denote M as a function of r and V; i.e., $M = F(r, V)$. During the process of simulated experiments, we find that it is easy to determine M for ensuring the algorithm convergent. However, the determination of optimal M is still an open problem. In the following experiments, we determine $M = \lceil 0.01(r + V(s'))\rceil$, where $\lceil x\rceil = \min\{n \in \mathbb{Z} | x \leq n\}$. The probability amplitudes will be normalized with $\sum_{x=1}^N |\alpha_x|^2 = 1$ after each updating. Since α_x indicates the occurrence probability of $|a_x\rangle$, it is no doubt that probability amplitude updating is the key toward recording the "trial-and-error" experience for learning algorithms. It is worth noting that the probability amplitude updating is very different from quantum algorithms. The objective of quantum algorithms is to find $|a_x\rangle$ by amplifying its occurrence probability to almost 1. However, the aim of amplitude amplification process in QiRL just appropriately amplifies corresponding amplitudes for "good" eigenactions.

IV. QiRL Algorithm and Performance Illustration

A. QiRL Algorithm

Based on the previous description, we can present the QiRL algorithm described in Fig. 1. In the algorithm, we can first establish a corresponding relationship between actions in RL and eigenactions in QiRL. Then, we can execute action selection according to $f(s) = \sum_{x=1}^N \frac{|a_x\rangle}{|\alpha_x|^2}$. If the agent selects an eigenaction $|a_x\rangle$, it executes this action and gives out the next state $|s'\rangle$, reward r, and state value $V(s')$. $V(s)$ is updated by the TD rule

$$V(s) \leftarrow V(s) + \eta(r + \gamma V(s') - V(s)) \tag{13}$$

where $\eta \in (0, 1]$ is the learning rate and $\gamma \in (0, 1)$ is a discount factor. Denote $\triangle V(s)$ as the difference of $V(s)$ between two neighbor iterations. A positive small real number e is predefined for the criterion of terminating an episode of learning process $|\triangle V(s)| \leq e$ when the state value $V(s)$ is updated. The convergence of QiRL can be stated as follows.

Proposition 2. For any Markov chain, QiRL algorithms converge to the optimal state value function $V^*(s)$ with probability 1 under proper exploration policy when the following conditions hold (where η_k is learning rate and nonnegative):

$$\lim_{T\to\infty}\sum_{k=1}^T \eta_k = \infty \quad \lim_{T\to\infty}\sum_{k=1}^T \eta_k^2 < \infty. \tag{14}$$

Remark 1: The proof of Proposition 2 is similar to the proof of convergence for QRL in [9]. In QiRL, we use the TD prediction for the state value updating. The convergence of QiRL

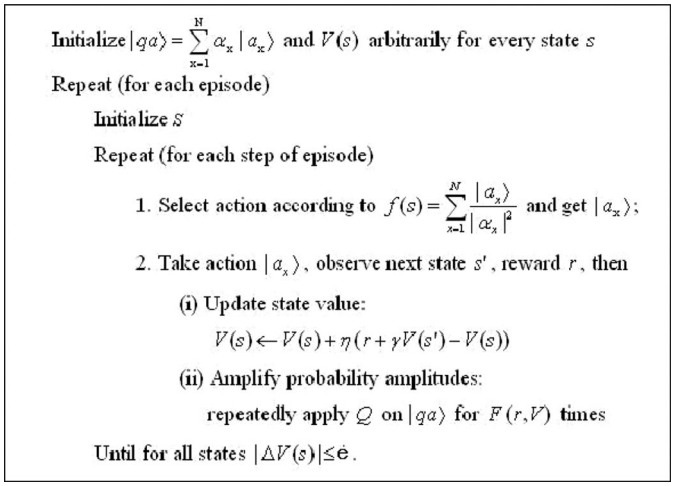


Fig. 1. QiRL algorithm.

is the same as that of traditional TD algorithms in RL. The main differences only lie in: The exploration policy in QiRL is inspired by the collapse postulate of quantum measurement while being observed, and it is a probabilistic exploration policy in essence. The modification does not affect the characteristic of convergence. The convergence of TD algorithms in RL has been provenusing the result in [44]. Hence, the QiRL algorithm converges when (14) holds.

It is clear that QiRL is derived from the TD method and is inspired by several ideas in quantum computation. The action set can be represented into a form of quantum states. The occurrence probability of every eigenaction is determined by corresponding probability amplitude, which is updated according to the rewards and value functions and is accomplished through quantum amplitude amplification. Hence, it naturally works as a proper exploration strategy to keep the balance between exploration and exploitation. When the environment changes, the learning policy also changes with the online learning process, which is difficult to implement for traditional reinforcement learning.

As for the robustness of QiRL, it is a model-free learning algorithm similar to many other kinds of RL algorithms [28]. It is robust (independent of environmental models) for different kinds of problems under unknown environments [28]. Moreover, it is also very robust to different learning rates and initial states due to its special exploration strategy. These robust characteristics will be tested in detail in the next section using a Markovian state transition example. The performance comparison between QiRL and other learning methods will also be further demonstrated in Section V.

B. Performance Illustration: A State Transition Example

To evaluate QiRL, a Markovian state transition example is demonstrated to test the robust performances of QiRL via different experimental settings.

1) Experimental Settings: Consider the Markovian state transition diagram as depicted in Fig. 2. There are 58 distinct states in this example and the narrow arrows (actions) are the state transitions between those states. From a start state, some states are reachable and some are unreachable. For each state transition, we will get a penalty reward of -1 until we get into the goal state, then we get a reward of $+100$. Our aim is to find an optimal state transition sequence that maximizes the sum of rewards between the start state and the desired goal state, as long as the two states are reachable. To compare the performance of QiRL with the existing RL methods (typically TD method), some normalized constraints are defined, i.e., for all the following experiments, a one-step updating rule is applied, the state values are all initialized as 0, and discount factor $\gamma = 0.99$.

2) Performance of QiRL for Different Initial States: In this group of experiments, QiRL is used to learn the optimal policy for a pointed goal state $S38$ from four different start states: $S1, S15, S19, S24$. The learning results are demonstrated in Fig. 3 compared with the TD algorithm (with the same learning rate). As shown in Fig. 3, for almost all these experiments with different start states, QiRL converges after about 2000 episodes (trials); while the TD algorithm is much more sensitive to start states and the number of trials varies from 4000 to 1000 for different start states. In Fig. 3, we also find that QiRL displays more oscillations before convergence than the TD algorithm. The reason is that QiRL explores more than the TD algorithm at the beginning of learning phase, but it learns faster and guarantees a better balancing between exploration and exploitation. These experiments demonstrate that QiRL is more robust to initial settings than traditional RL and possesses a steady performance for different specific learning tasks.

3) Performance of QiRL With Different Learning Rates: Furthermore, a group of experiments is taken to demonstrate the performance of QiRL with different learning rates. A wider range of learning rates is very important for RL methods in practical applications. A large learning rate will dramatically speed up learning. However, most RL algorithms are very sensitive and unstable to large learning rates, and it is very difficult to set a proper learning rate.

As shown in Fig. 4, QiRL is much more robust to a very large range of learning rates. It keeps a good performance with the learning rate η from 0.01 to 0.12. However, it is very difficult for the TD algorithm to converge when $\eta \geq 0.02$. In fact, the TD algorithm cannot find the optimal solutions with $\eta \geq 0.02$ for the same task in our experiments. This characteristic makes QiRL a very good candidate for practical learning tasks in a dynamic unknown environment, such as the mobile robot navigation task.

V. Robot Navigation Based on QiRL Algorithm

In Section IV, we have shown that QiRL learns faster and is more robust to initial settings and learning rates than traditional RL. In this section, the proposed QiRL algorithm is applied to the navigation control of an autonomous mobile robot.

A. Simulated Experiments

The mobile robot navigation is a typical sequential decision problem in dynamic environments. The main task is to drive a robot to move in an initially unknown environment with onboard sensors. The robot model used in this paper is shown in Fig. 5.

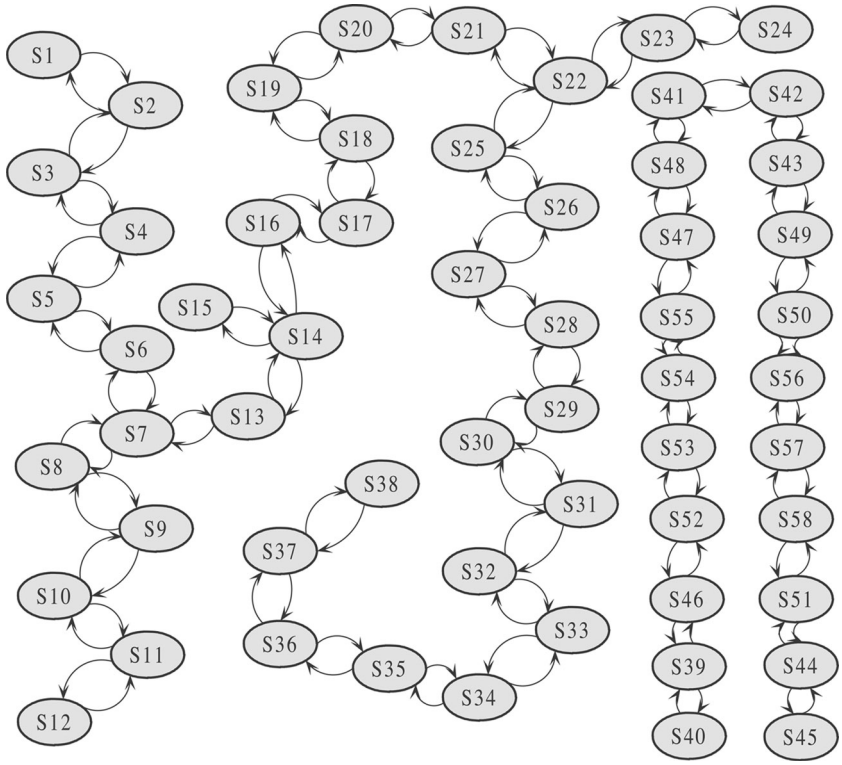


Fig. 2. Example of Markovian state transition.

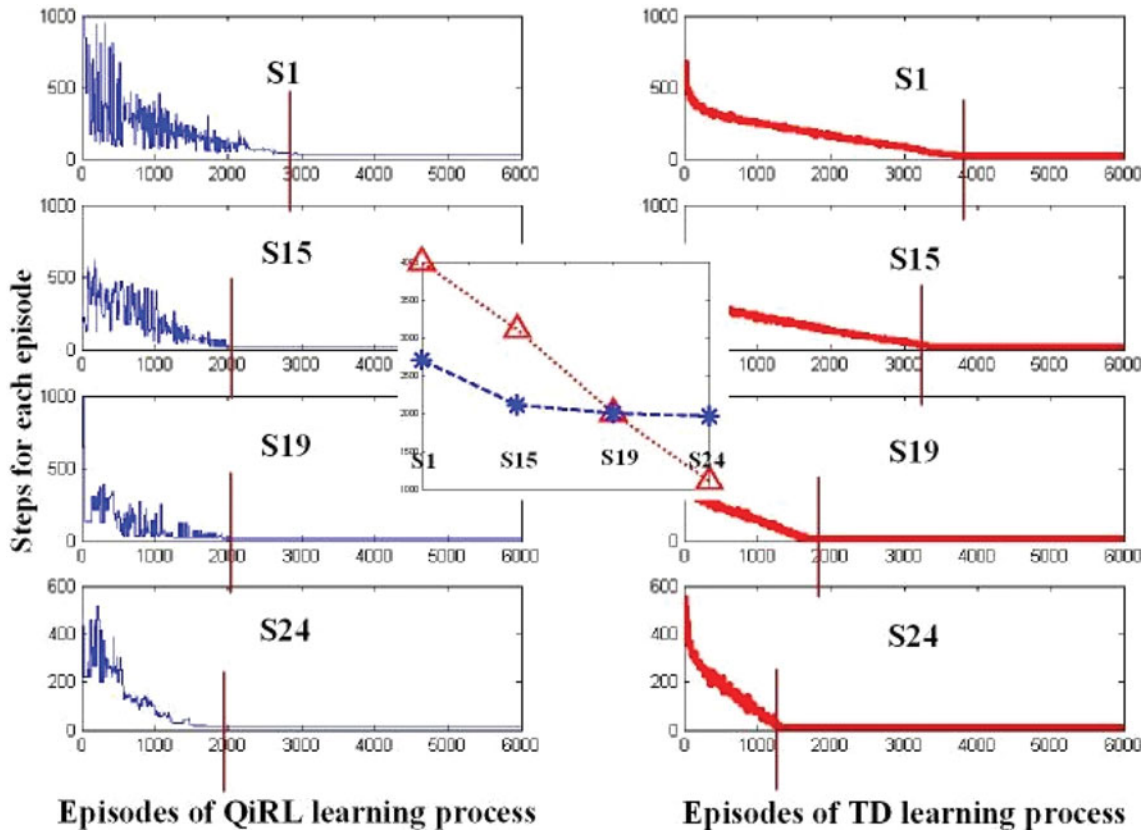


Fig. 3. Learning performance of QiRL for state transition from different start states (S1, S15, S19 and S24) to the same goal state S38.

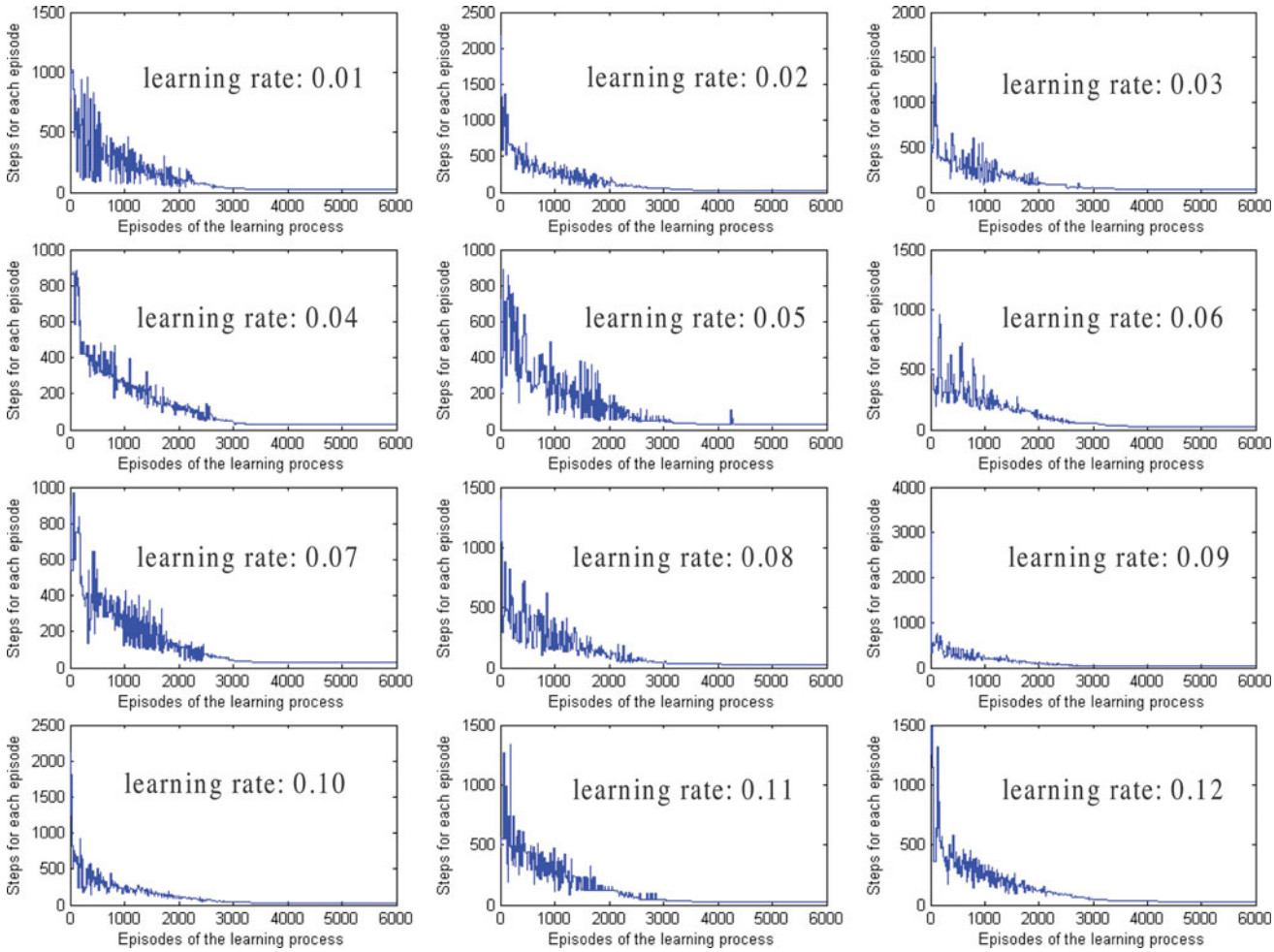


Fig. 4. Learning performance of QiRL for state transition from S1 to the goal state S38 with different learning rates.

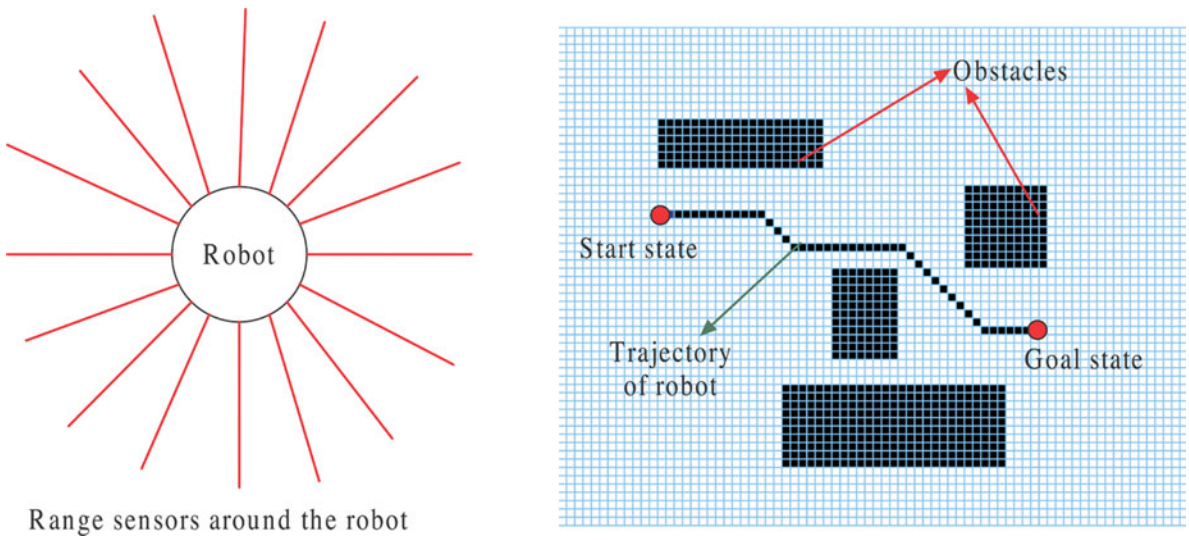


Fig. 5. Demonstration of robot navigation with range sensors in a grid world with obstacles.

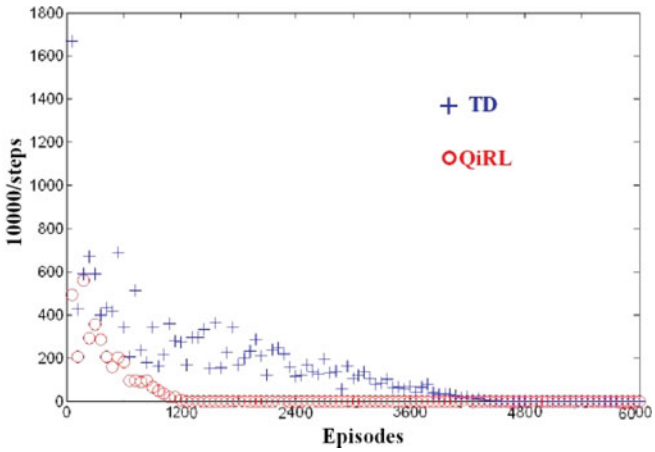


Fig. 6. Learning performance of QiRL and TD for obstacle-avoidance behavior.

There are 16 range sensors around the robot and they provide the sensory data of the obstacles in the unknown environment. Grid-based representation is adopted for the modeling of environment and groups of black grids represent obstacles. The task of the learning control system is to give an optimal trajectory from a start state to a goal state without getting into obstacles.

In this paper, a simulation environment is built up using Visual C++ 6.0 with the setting of 600×400 grids. This environment is assumed to be unknown for the robot. The parameters for the learning algorithms are as follows: discount factor $\gamma = 0.99$, learning rate $\eta = 0.01$, all state values initialized as 0, reward –1 for each step until getting into the goal state and then a reward +100 received. If the robot gets into an obstacle area before reaching the goal state, the current learning process (one episode) ends and a reward –100 is received.

To compare the performance of QiRL and TD algorithms, an important navigation task, obstacle-avoidance behavior, is first tested in this simulation environment, i.e., the robot learns to wander in an unknown environment without getting into any obstacles. Fig. 6 shows the learning performance of QiRL and TD for the obstacle-avoidance behavior. In Fig. 6, an episode is a learning process before the robot gets into an obstacle. The variable *steps* represent the number of steps for each episode. Fig. 6 shows that through learning, the robot is getting better at obstacle-avoidance and the number of steps is getting larger ($\frac{10{,}000}{steps}$ is getting smaller to zero) for each episode. But QiRL learns faster than the TD algorithm because this problem is relatively complex and QiRL is more robust than the TD algorithm.

After learning, the control results are demonstrated in Fig. 7. Fig. 7(a) shows the control performance of navigation in an environment with sparse obstacles and Fig. 7(b) shows that in clustered obstacles. The simulated results show that QiRL is robust for dynamic control problems and is effective for the navigation control of autonomous mobile robots.

Remark 2: Several machine learning methods have been applied to navigation control of autonomous mobile robots. For example, Alvarez *et al*. [45] proposed a genetic algorithm for navigation planning of an autonomous underwater vehicle in an ocean environment. Thrun [46] used artificial neural networks to learn grid-based maps for indoor mobile robot navigation. Different machine learning methods have their advantages and disadvantages depending on different tasks for robot navigation. Roughly speaking, genetic algorithms are suitable for path planning of mobile robots, where an optimal path can be learned off-line and its application usually requires some prior knowledge about environmental maps [22]. Artificial neural networks are useful for learning environmental maps and local controller in robot navigation [17], [46], where known training data are necessary. The unique advantage of QiRL is that it is model-free (where no prior knowledge and training data are required) and robust to initial conditions and learning rates. It is more suitable for online learning and lifelong adaptive learning for autonomous mobile robot navigation in unknown environments.

B. Real Experiments

The proposed methods are further applied to a real robot. The robot employed in this study is a mobile robot called MT-R and its main configurations are shown in Fig. 8. It is a two-wheel-driven robot with six pairs of range sensors and each pair of range sensor consists of an ultrasonic sensor and an infrared sensor. The configuration of the six pairs of range sensors is shown in Fig. 8. The overall structure of MT-R consists of three layers of function modules: multisensor layer, information fusion and planning layer, and execution and energy-supplying layer. The specifications of robot MT-R are listed as follows:

1) Dimensions: base ($d = 0.5$ m*), height (*0.6 m*);*

*2) Weight:*30 kg;

*3) Battery:*24 V,20 AH*;*

*4) Motors: Two MAXON motors (*24 V,70 W*) with two shaft encoders;*

*5) Motion control processor:*DSP + CPLD*;*

6) Max speed: 2.5 m/s*;*

7) Industrial PC*:* Intel PM1.8G,SATA 80G *hard disk,* DDR512 M *memory;*

*8) Wireless communication:*54 M*;*

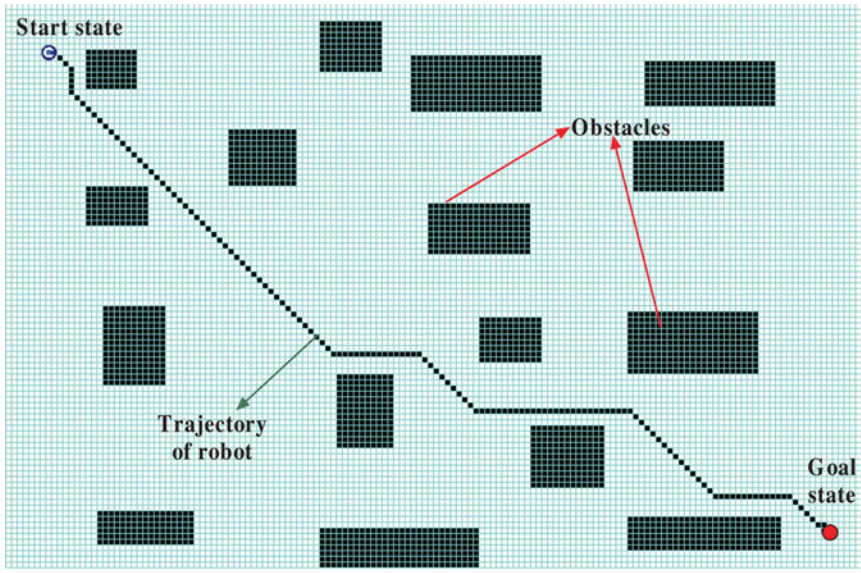
*9) Camera:*1.3 M pixel,30 frame/s,USB *interface;*

10) Sensors: Six ultrasonic sensors with sensitive range of ($0.2 \sim 7$ m*);*

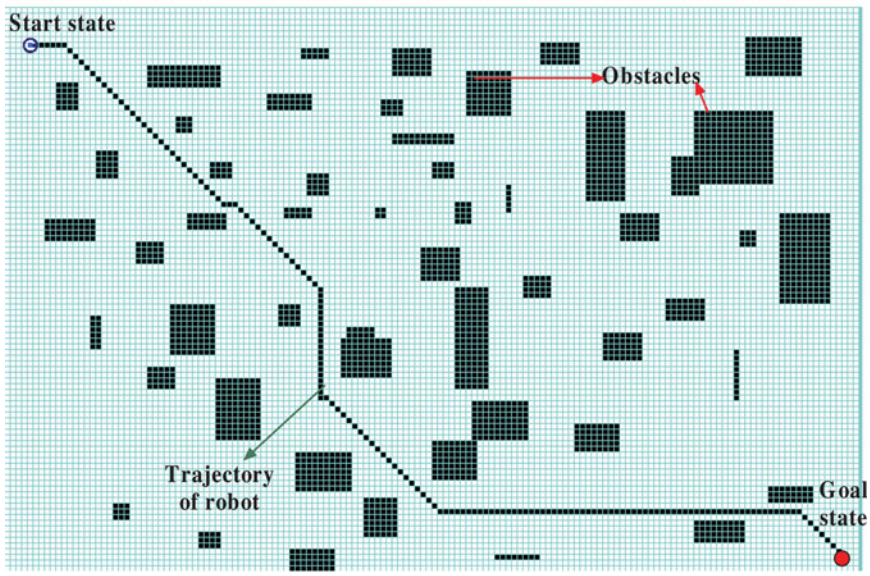
11) Infrared sensors: Six infrared sensors with sensitive range of ($0.08 \sim 0.8$ m*);*

To navigate the robot in a clustered environment, the mobile robot percepts and then decides the motion commands. The robot MT-R has six range sensor pairs and can detect the obstacles from six directions (Fig. 8). The navigation environment in the experimental studies is an office building with corridors as shown in Fig. 9. The task of the robot is to move from an initial state, walk along the corridor, and get into a pointed room. The parameter settings are the same as those in the simulated experiments. The difference lies in that the sensory inputs are six range data as shown in Fig. 8.

Fig. 9 shows the results of this navigation task. After learning, the robot successfully walks into a pointed room through a corridor while avoiding obstacles. More results also show that



(a)



(b)

Fig. 7. Robot navigation using QiRL. (a) Navigation with sparse obstacles. (b) Navigation with clustered obstacles.

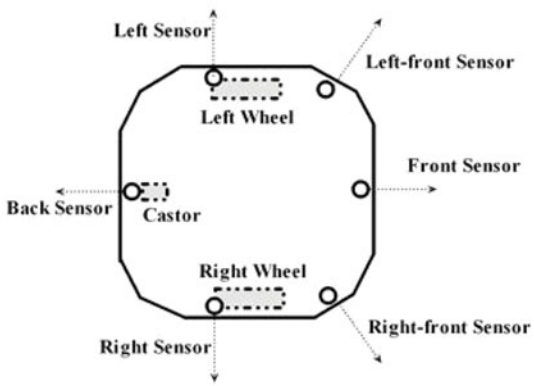


Fig. 8. Robot MT-R and its configuration.

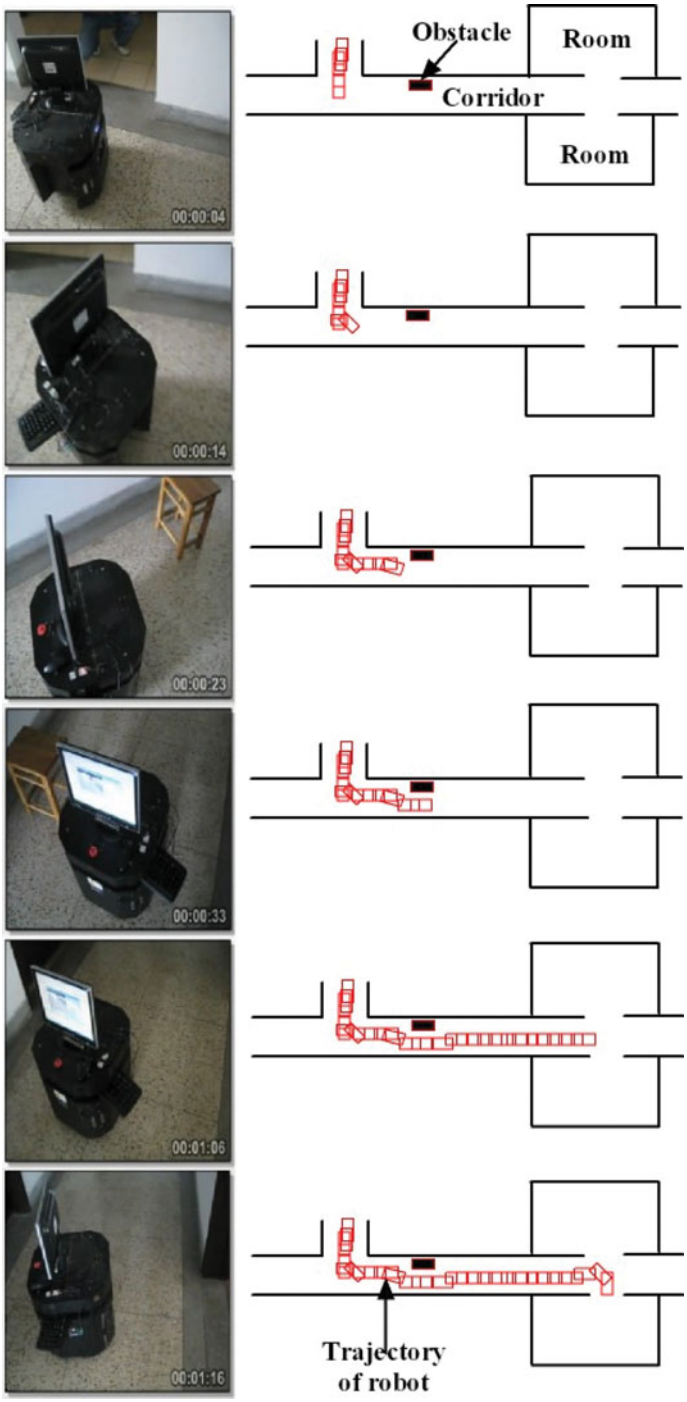


Fig. 9. Robot navigation in an unknown environment.

the QiRL algorithm is practicable and robust to such problems as sensor-based and sequential decision making for mobile robots.

VI. Concluding Remark

Both quantum computation and machine learning are rapidly developing research fields. It is a suitable chance to combine traditional learning algorithms and quantum computation methods and maybe some difficulties will be solved appropriately in a new way. In this paper, we explore the possible application of quantum computation ideas to learning tasks and propose a novel quantum-inspired reinforcement learning algorithm. In this QiRL algorithm, the action selection policy is inspired by the collapse phenomenon in quantum measurement and the reinforcement strategy is inspired by amplitude amplification in quantum computation. This action selection policy is proven useful to better balancing the tradeoff between exploration and exploitation, and can make the QiRL algorithm more robust in applications. The algorithm is used for the navigation control of an autonomous mobile robot and the results demonstrate that QiRL is robust to different learning rates and initial states and is effective for adaptive robot learning. Our future work will focus on the practicable representation methods for the state-action space of QiRL to extend it to more applications, and other quantum-inspired learning theories and algorithms.

Acknowledgment

The authors would like to thank the Technical Editor and three anonymous reviewers for helpful comments and suggestions which helped improve this paper.

References

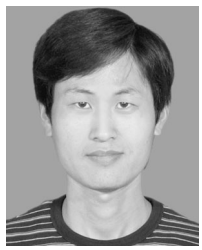
[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[2] P. Benioff, "Quantum robots and environments," *Phys. Rev. A*, vol. 58, pp. 893–904, 1998.
[3] D. Y. Dong, C. L. Chen, C. B. Zhang, and Z. H. Chen, "Quantum robot: Structure, algorithms and applications," *Robotica*, vol. 24, pp. 513–521, 2006.
[4] S. Kak, "On quantum neural computing," *Inf. Sci.*, vol. 83, pp. 143–160, 1995.
[5] A. Narayanan and T. Menneer, "Quantum artificial neural network architectures and components," *Inf. Sci.*, vol. 128, pp. 231–255, 2000.
[6] G. G. Rigatos and S. G. Tzafestas, "Parallelization of a fuzzy control algorithm using quantum computation," *IEEE Trans. Fuzzy Syst.*, vol. 10, no. 4, pp. 451–460, Aug. 2002.
[7] A. Malossini, E. Blanzieri, and T. Calarco, "Quantum genetic optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 231–241, Apr. 2008.
[8] K. H. Han and J. H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 6, pp. 580–593, Dec. 2002.
[9] D. Dong, C. Chen, H. Li, and T. J. Tarn, "Quantum reinforcement learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1207–1220, Oct. 2008.
[10] D. Dong, C. Chen, T. J. Tarn, A. Pechen, and H. Rabitz, "Incoherent control of quantum systems with wavefunction controllable subspaces via quantum reinforcement learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 4, pp. 957–962, Aug. 2008.
[11] D. Dong, C. Zhang, H. Rabitz, A. Pechen, and T. J. Tarn, "Incoherent control of locally controllable quantum systems," *J. Chem. Phys.*, vol. 129, no. 15, pp. 154103-1–154103-10, 2008.
[12] G. Brassard, P. Høyer, and A. Tapp, "Quantum counting," in *Proc. 25th Int. Colloq. Automata, Lang. Program., Lecture Notes in Computer Science*, 1998.
[13] D. Dong, C. Chen, and H. Li, "Reinforcement strategy using quantum amplitude amplification for robot learning," in *Proc. 26th Chinese Control Conf.*, Zhangjiajie, China, 2007, vol. 6, pp. 571–575.
[14] D. Dong and C. Chen, Quantum-inspired reinforcement learning for decision-making of Markovian state transition," presented at the 2010 Int. Conf. Intell. Syst. Knowl. Eng., Hangzhou, China, Nov. 15–16, 2010.
[15] C. Chen, H. Li, and D. Dong, "Hybrid control for robot navigation: A hierarchical Q-learning algorithm," *IEEE Robot. Autom. Mag.*, vol. 15, no. 2, pp. 37–47, Jun. 2008.

[16] S. Park and S. Hashimoto, "Autonomous mobile robot navigation using passive RFID in indoor environment," *IEEE Trans. Ind. Electron.*, vol. 56, no. 7, pp. 2366–2373, Jul. 2009.
[17] J. A. Fernandez-Leon, G. G. Acosta, and M. A. Mayosky, "Behavioral control through evolutionary neurocontrollers for autonomous mobile robot navigation," *Robot. Auton. Syst.*, vol. 57, no. 4, pp. 411–419, 2009.
[18] S. Shair, J. H. Chandler, V. J. Gonzalez-Villela *et al.*, "The use of aerial images and GPS for mobile robot waypoint navigation," *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 6, pp. 692–699, Dec. 2008.
[19] A. Franchi, L. Freda, G. Oriolo *et al.*, "The sensor-based random graph method for cooperative robot exploration," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 2, pp. 163–175, Apr. 2009.
[20] C. Chen and D. Dong, "Grey system based reactive navigation of mobile robots using reinforcement learning," *Int. J. Innov. Comput., Inf. Control*, vol. 6, no. 2, pp. 789–800, 2010.
[21] G. E. Jan, K. Y. Chang, and I. Parberry, "Optimal path planning for mobile robot navigation," *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 4, pp. 451–460, Aug. 2008.
[22] T. W. Manikas, K. Ashenayi, and R. L. Wainwright, "Genetic algorithms for autonomous robot navigation," *IEEE Instrum. Meas. Mag.*, vol. 10, no. 6, pp. 26–31, Dec. 2007.
[23] M. F. Selekwa, D. D. Dunlap, D. Shi, and E. G. Collins, "Robot navigation in very cluttered environments by preference-based fuzzy behaviors," *Robot. Auton. Syst.*, vol. 56, no. 3, pp. 231–246, 2008.
[24] M. A. Salichs and L. Moreno, "Navigation of mobile robot: Open questions," *Robotica*, vol. 18, pp. 227–234, 2000.
[25] M. Rodriguez, R. Iglesias, C. V. Regueiro, J. Correa, and S. Barro, "Autonomous and fast robot learning through motivation," *Robot. Auton. Syst.*, vol. 55, pp. 735–740, 2007.
[26] N. D. Ratliff, D. Silver, and J. A. Bagnell, "Learning to search: Functional gradient techniques for imitation learning," *Auton. Robots*, vol. 27, pp. 25–53, 2009.
[27] T. Kondo and K. Ito, "A reinforcement learning with evolutionary state recruitment strategy for autonomous mobile robots control," *Robot. Auton. Syst.*, vol. 46, pp. 111–124, 2004.
[28] R. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.
[29] R. Sutton, "Learning to predict by the methods of temporal difference," *Mach. Learn.*, vol. 3, pp. 9–44, 1988.
[30] C. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, 1992.
[31] D. Vengerov, N. Bambos, and H. Berenji, "A fuzzy reinforcement learning approach to control in wireless transmitters," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 4, pp. 768–778, Aug. 2005.
[32] Y. Wang and C. W. de Silva, "Sequential Q-learning with Kalman filtering for multirobot cooperative transportation," *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 2, pp. 261–268, Apr. 2010.
[33] T. G. Dietterich, "Hierarchical reinforcement learning with the Maxq value function decomposition," *J. Artif. Intell. Res.*, vol. 13, pp. 227–303, 2000.
[34] M. J. Er and C. Deng, "Online tuning of fuzzy inference systems using dynamic fuzzy Q-learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 3, pp. 1478–1489, Jun. 2004.
[35] S. Whiteson and P. Stone, "Evolutionary function approximation for reinforcement learning," *J. Mach. Learn. Res.*, vol. 7, pp. 877–917, 2006.
[36] M. Kaya and R. Alhajj, "A novel approach to multiagent reinforcement learning: Utilizing OLAP mining in the learning process," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 35, no. 4, pp. 582–590, Nov. 2005.
[37] S. G. Tzafestas and G. G. Rigatos, "Fuzzy reinforcement learning control for compliance tasks of robotic manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 1, pp. 107–113, Feb. 2002.
[38] P. Høyer, "Arbitrary phases in quantum amplitude amplification," *Phys. Rev. A*, vol. 62, pp. 052304-1–052304-5, 2000.
[39] G. Brassard and P. Høyer, "An exact quantum polynomial-time algorithm for Simon's problem," in *Proc. 5th Israeli Symp. Theory Comput. Syst.*, Los Alamitos, CA, 1997, pp. 12–23.
[40] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, pp. 325–327, 1997.
[41] G. L. Long, "Grover algorithm with zero theoretical failure rate," *Phys. Rev. A*, vol. 64, pp. 022307-1–022307-4, 2001.
[42] D. Dong and I. R. Petersen, "Sliding mode control of quantum systems," *New J. Phys.*, vol. 11, pp. 105033-1–105033-18, 2009.
[43] L. K. Grover, "Quantum computers can search rapidly by using almost any transformation," *Phys. Rev. Lett.*, vol. 80, pp. 4329–4332, 1998.
[44] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific, 1996.
[45] A. Alvarez, A. Caiti, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *IEEE J. Ocean. Eng.*, vol. 29, no. 2, pp. 418–429, Apr. 2004.
[46] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artif. Intell.*, vol. 99, pp. 21–71, 1998.

Daoyi Dong (S'05–M'06) was born in Hubei, China. He received the B.E. degree in automatic control and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively.

He was a Postdoctoral Fellow at the Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences from 2006 to 2008. He then joined the Institute of Cyber-Systems and Control, State Key Laboratory of Industrial Control Technology, Zhejiang University. Since June 2008, he has been a Research Associate at the University of New South Wales at the Australian Defence Force Academy, Canberra, Australia. His research interests include quantum control, quantum computation, and quantum intelligent information processing.

Dr. Dong received an Australian Postdoctoral Fellowship from the Australian Research Council in 2010, the Chinese Academy of Sciences K. C. Wong Postdoctoral Fellowships in 2006, and the President Scholarship of the Chinese Academy of Sciences in 2004.



Chunlin Chen (S'05–M'06) was born in Anhui, China, in 1979. He received the B.E. degree in automatic control and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively.

Currently, he is an Associate Professor in the Department of Control and System Engineering, Nanjing University, Nanjing, China. His research interests include machine learning, intelligent control, mobile robotics, and quantum algorithm.

Jian Chu was born in Zhejiang Province, China, in 1963. He received the B.Sc., M.Sc., and Ph.D. degrees in industrial process control from Zhejiang University, Hangzhou, China, in 1982, 1984, and 1989, respectively. He studied at Kyoto University, Kyoto, Japan, from 1986 to 1989.

From 1989 to 1991, he was a Postdoctoral Research Fellow at Zhejiang University. He has been a Faculty Member at Zhejiang University since 1991 and was promoted to Full Professor in 1993. He was appointed as the Yangzie Scholar Professor in 1999. He is now the Vice President, the Director of the Institute of Cyber-Systems and Control, and the Director of the State Key Laboratory of Industrial Control Technology, all at Zhejiang University. He was also a Committee Member of the Control System Design Committee, International Federation of Automatic Control (IFAC). He led a research team that developed the first hot-redundancy-technology-based distributed control system in China in 1993, and the first multifieldbus-based distributed network control systems in 1997. He led a team to constitute the National Standard GB/T 20171-2006 "EPA system architecture and communication specifications for use in industrial control and measurement systems", which was accepted as the EPA International Standard by IEC 61158-3-14/-4-14/-5-14/-6-14. He has authored or coauthored five books and more than 100 journal papers. His research interests include sensor networks, robotics, system modeling, advanced process control, and optimization for a variety of large-scale industrial systems.

Tzyh-Jong Tarn (M'71–SM'83–F'85–LF'05) received the D.Sc. degree in control system engineering from Washington University in St. Louis, St. Louis, MO.

Currently, he is a Professor in the Department of Electrical and Systems Engineering and the Director of the Center for Robotics and Automation at Washington University in St. Louis.

Professor Tarn served as the President of the IEEE Robotics and Automation Society from 1992 to 1993, the Director of IEEE Division X from 1995 to 1996, and was a member of the IEEE Board of Directors from 1995 to 1996. He received the NASA Certificate of Recognition for the creative development of a technical innovation on robot arm dynamic control by computer in 1987. The Japan Foundation for the Promotion of Advanced Automation Technology presented him with the Best Research Article Award in March 1994. He also received the Best Paper Award at the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. He is the first recipient of both the Nakamura Prize and the Ford Motor Company Best Paper Award at the Japan/USA Symposium on Flexible Automation in 1998. In addition, he was the recipient of the prestigious Joseph F. Engelberger Award of the Robotic Industries Association in 1999, the Auto Soft Lifetime Achievement Award in 2000, and the Pioneer in Robotics and Automation Award in 2003 from the IEEE Robotics and Automation Society.