Q-learning for Optimal Control of Continuous-time Systems

Article · October 2014
Source: arXiv

3 authors, including:

Tingwen Huang
Texas A&M University at Qatar
464 PUBLICATIONS  8,332 CITATIONS

Some of the authors of this publication are also working on these related projects:

- Boolean network View project
- SUPP smart grid View project
Q-learning for Optimal Control of Continuous-time Systems

Biao Luo, Derong Liu, Fellow, IEEE and Tingwen Huang

Abstract—In this paper, two Q-learning (QL) methods are proposed and their convergence theories are established for addressing the model-free optimal control problem of general nonlinear continuous-time systems. By introducing the Q-function for continuous-time systems, policy iteration based QL (PIQL) and value iteration based QL (VIQL) algorithms are proposed for learning the optimal control policy from real system data rather than using mathematical system model. It is proved that both PIQL and VIQL methods generate a nonincreasing Q-function sequence, which converges to the optimal Q-function. For implementation of the QL algorithms, the method of weighted residuals is applied to derived the parameters update rule. The developed PIQL and VIQL algorithms are essentially off-policy reinforcement learning approaches, where the system data can be collected arbitrary and thus the exploration ability is increased. With the data collected from the real system, the QL methods learn the optimal control policy offline, and then the converged control policy will be employed to real system. The effectiveness of the developed QL algorithms are verified through computer simulation.

Index Terms—Q-learning; model-free optimal control; off-policy reinforcement learning; the method of weighted residuals.

I. INTRODUCTION

Q-LEARNING (QL) is a popular and powerful off-policy reinforcement learning (RL) method, which is a great breakthrough in RL researches [1]–[4]. QL was proposed by Watkins [5], [6] that can be used to optimally solve Markov decision processes (MDPs). The major attractions of QL are its simplicity and that it allows using arbitrary sampling policies to generate the training data rather than using the policy to be evaluated. Till present, Watkins’ QL [5], [6] has been extended and some meaningful results have been reported [6]–[12] in machine learning community. Such as, the theoretical analysis of QL was studied in [7], [10]–[13]. An incremental multi-step QL [14] was proposed, named Q(λ)-learning, which extends the one-step QL by combining it with TD(λ) returns for general λ in a natural way for delayed RL. A GQ(λ) algorithm was introduced in [11], which works to a general setting including eligibility traces and off-policy learning of temporally abstract predictions. By using two-timescale stochastic approximation methodology, two QL algorithms [15] were proposed. Observe that these results about QL are mainly for MDPs, which is highly related to the optimal control problem in control community. Thus, it is possible and promising to introduce the basic QL framework for addressing the optimal control design problem.

For the optimal control problem in control community, it usually depends on the solution of the complicated Hamilton-Jacobi-Bellman equation (HJBE) [16]–[18], which is extremely difficult and requires the accurate mathematical system model. However, for many practical industrial systems, due to their large scale and complex manufacturing techniques, equipment and procedures, it is usually impossible to identify the accurate mathematical model for optimal control design, and thus the explicit expression of the HJBE is unavailable. On the other hand, with the development and extensive applications of digital sensor technologies, and the availability of cheaper measurement and computing equipment, more and more system information could be extracted for direct control design. In the past few years, the model-free optimal control problem has attracted researchers’ extensive attention in control community [19], [20], and has also brought new challenges to them. Some model-free or partially model-free RL methods [21]–[32] have been developed for solving the optimal control design problem by using real system data. For example, RL approaches were employed to solve linear quadratic regulator (LQR) problem [24], optimal tracking control problem [31] and zero-sum game problem [30] of linear systems. For nonlinear optimal control problem, Modares et al. [28] developed an experience-replay based integral RL algorithm for nonlinear partially unknown constrained-input systems. Zhang et al. [22] presented a data-driven robust approximate optimal tracking control scheme for nonlinear systems, but it requires a prior model identification procedure and the approximate dynamic programming method is still model-based. Globalized dual heuristic programming algorithms [23], [26] were developed by using three neural networks (NNs) for estimating system dynamics, cost function and its derivatives, and control policy, where model NN construction error was considered.

Most recently, some QL techniques have been introduced for solving the optimal control problems [33]–[37]. Such as, for linear discrete-time systems, the optimal tracking control problem [37] and $H_\infty$ control problem [33], [34] were studied with QL. Online QL algorithms [35], [36] were investigated for solving the LQR problem of linear continuous-time systems. However, these works are just for simple linear optimal control problem [35]–[37] or discrete-time systems [33], [34], [36], [37]. To the best of our knowledge, the QL method and its theories are still rarely studied for general nonlinear continuous-time systems.

In this paper, we consider the model-free optimal con-
control problem of general nonlinear continuous-time systems, and two QL algorithms proposed: policy iteration based QL (PIQL) and value iteration based QL (VIQL). The rest of this paper is arranged as follows. Section II presents the problem description. PIQL and VIQL algorithms are proposed in Sections III and IV, which are then simplified for LQR problem in Section V. Subsequently, the computer simulation results are demonstrated in Section VI and a brief conclusion is given in Section VII.

Notation: $\mathbb{R}^n$ is the set of the $n$-dimensional Euclidean space and $\|\cdot\|$ denotes its norm. The superscript $T$ is used for the transpose and $I$ denotes the identify matrix of appropriate dimension. $\nabla \triangleq \partial/\partial x$ denotes a gradient operator notation. For a symmetric matrix $M, M > (\geq)0$ means that it is a positive (semi-positive) definite matrix. $\|v\|^2_M \triangleq v^TMv$ for some real vector $v$ and symmetric matrix $M > (\geq)0$ with appropriate dimensions. $C^1(\mathcal{X})$ is a function space on $\mathcal{X}$ with first derivatives are continuous. Let $\mathcal{X}$ and $\mathcal{U}$ be compact sets, denote $\mathcal{D} \triangleq \{(x,u,x')|x,x' \in \mathcal{X}, u \in \mathcal{U}\}$. For column vector functions $s_1(x,u,x')$ and $s_2(x,u,x')$, where $(x,u,x') \in \mathcal{D}$, define the inner product $\langle s_1(x,u,x') \rangle_{\mathcal{D}} \triangleq \int_\mathcal{D} s_1^1(x,u,x')s_2(x,u,x')d(x,u,x')$ and the norm $\|s_1(x,u,x')\|_{\mathcal{D}} \triangleq \langle s_1(x,u,x'), s_1(x,u,x') \rangle_{\mathcal{D}}^{1/2}$.

II. PROBLEM DESCRIPTION

Let us consider the following general nonlinear continuous-time system:

$$\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0 \quad (1)$$

where $x = [x_1 \ldots x_n]^T \in \mathcal{X} \subset \mathbb{R}^n$ is the state, $x_0$ is the initial state and $u = [u_1 \ldots u_m]^T \in \mathcal{U} \subset \mathbb{R}^m$ is the control input. Assume that $f(x,u)$ is Lipschitz continuous on the set $\mathcal{X} \times \mathcal{U}$ that contains the origin, i.e., $f(0,0) = 0$. The system is stabilizable on $\mathcal{X}$, i.e., there exists a continuous control function $u(x)$ such that the system is asymptotically stable on $\mathcal{X}$.

For the model-free optimal control problem considered in this paper, the model $f(x,u)$ of system (1) is completely unknown. The objective of optimal control design is to find a state feedback control law $u(t) = u(x(t))$, such that the system (1) is closed-loop asymptotically stable, and minimize the following generalized infinite horizon cost functional:

$$V_u(x_0) \triangleq \int_0^{+\infty} \left(S(x(t)) + W(u(t))\right)dt \quad (2)$$

where $S(x)$ and $W(u)$ are positive definite functions, i.e., for $\forall x \neq 0, u \neq 0, S(x) > 0, W(u) > 0$, and $S(x) = 0, W(u) = 0$ only when $x = 0, u = 0$. Then, the optimal control problem is briefly presented as

$$u(t) = u^*(x) \triangleq \arg\min_u V_u(x_0). \quad (3)$$

III. POLICY ITERATION BASED Q-LEARNING

In this section, a PIQL algorithm and its convergence are established for solving the model-free optimal control problem of the system (1). Before starting, the definition of admissible control is necessary.

Definition 1. (Admissible control) For the given system (1), $x \in \mathcal{X}$, a control policy $u(x)$ is defined to be admissible with respect to cost function $C(x)$ on $\mathcal{X}$, denoted by $u(x) \in \Omega(\mathcal{X})$, if: 1) $u$ is continuous on $\mathcal{X}$, 2) $u(0)=0$, 3) $u(x)$ stabilizes the system, and 4) $V_u(x) < \infty, \forall x \in \mathcal{X}$. \hfill $\square$

A. Policy Iteration Based Q-learning

Noting that the mathematical system model $f(x,u)$ is unknown, a PIQL algorithm is proposed to learn the optimal control policy from real system data directly. For notations simplicity, denote $t' \triangleq t + \Delta t$ for $\forall t, \Delta t > 0$, $x_0 \triangleq x(t)$ and $x_1 \triangleq x(t')$. For an admissible control policy $u(x) \in \Omega(\mathcal{X})$, define its cost function

$$V_u(x_t) \triangleq \int_t^{+\infty} R(x(\tau), u(\tau))d\tau \quad (4)$$

where $R(x,u) \triangleq S(x) + W(u)$, and $V_u(0) = 0$. Define the Hamilton function

$$H(x,u,\nabla V) \triangleq [\nabla V(x)]^Tf(x,u) + R(x,u) \quad (5)$$

for some cost function $V(x) \in C^1(\mathcal{X})$. Taking derivative on both sides of expression (4) along with the system (1) yields

$$[\nabla V_u(x)]^Tf(x,u) = -R(x,u)$$

i.e.,

$$H(x,u,\nabla V_u) = 0 \quad (6)$$

which is linear partial differential equation.

Let $V^*(x) \triangleq V^*_u(x)$ be the optimal cost function, then the optimal control law (3) is given by

$$u^*(x) = \arg\min_u H(x,u,\nabla V^*). \quad (7)$$

Substituting (7) into (6) yields the following HJBE

$$H(x,u^*,\nabla V^*) = 0 \quad (8)$$

which is nonlinear partial differential equation. It is observed that the optimal control policy $u^*$ relies on the solution $V^*$ of the HJBE (8), while the unavailability of the system model $f(x,u)$ prevents using model-based approaches for control design.

To derive the PIQL algorithm, it is necessary to introduce an action-state value function, named Q-function. For $\forall u(x) \in \Omega(\mathcal{X})$, define its Q-function as

$$Q_u(x_t, \mu) \triangleq \int_t^{t'} R(x(\tau), \mu(\tau))d\tau + \int_{t'}^{+\infty} R(x(\tau), u(\tau))d\tau. \quad (9)$$

where $(x_t, \mu) \in \mathcal{X} \times \mathcal{U}$ and $Q_u(0,0) = 0$. It is found that $Q_u(x,u) = V_u(x)$, then the Q-function (9) is rewritten as

$$Q_u(x_t, \mu) = \int_t^{t'} R(x(\tau), \mu(\tau))d\tau + Q_u(x_t', u)$$

$$= \int_t^{t'} R(x(\tau), \mu(\tau))d\tau + V_u(x_t'). \quad (10)$$

Note that the Q-function $Q_u(x,\mu)$ for a state $x$ and control action $\mu$ represents the value of the performance metric obtained when action $\mu$ is used in state $x$ and the control
policy $u$ is pursued thereafter. For the optimal control policy $u^*(x)$, the associate optimal Q-function $Q^*(x, \mu)$ is given by

$$Q^*(x_t, \mu) = \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + Q_{u^*}(x_t', u^*)$$

$$= \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + V^*(x_t').$$ \hspace{1cm} (11)

Then,

$$Q^*(x_t, u^*) = \min_{\mu} Q^*(x_t, \mu)$$

$$= \min_{\mu} \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + V^*(x_t')$$

$$= \int_t^{t'} \mathcal{R}(x(\tau), u^*(\tau))d\tau + V^*(x_t')$$

$$= V^*(x_t).$$ \hspace{1cm} (12)

According to the expressions (3) and (12), the optimal control policy $u^*(x)$ can also be presented as

$$u^*(x) = \arg\min_{\mu} V_u(x) = \arg\min_{\mu} Q^*(x, \mu).$$ \hspace{1cm} (13)

Now, we give the PIQL method as follows:

\textbf{Algorithm 1.} Policy iteration based Q-learning

- **Step 1:** Let $u^{(0)}(x) \in \mathcal{U}(\mathcal{X})$ be an initial control policy, and $i = 0$;
- **Step 2:** (Policy evaluation) Solve the equation
  $$Q^{(i)}(x_t, \mu) = \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + Q^{(i)}(x'_t, u^{(i)})$$
  for unknown Q-function $Q^{(i)} \triangleq Q_u^{(i)}$;
- **Step 3:** (Policy improvement) Update control policy with
  $$u^{(i+1)}(x) = \arg\min_{\mu} Q^{(i)}(x, \mu);$$ \hspace{1cm} (15)
- **Step 4:** Let $i = i + 1$, go back to Step 2 and continue. □

\textbf{Remark 1.} The PIQL Algorithm 1 involves two basic operators: policy evaluation and policy improvement. The policy evaluation is to evaluate the current control policy $u^{(i)}$ for its Q-function $Q^{(i)}$, and the policy improvement is to get a better control policy $u^{(i+1)}$ based on the Q-function $Q^{(i)}$. By giving an initial admissible control policy, PIQL algorithm implement policy evaluation and policy improvement alternatively to learn the optimal Q-function $Q^*$ and the optimal control policy $u^*$. Note that the proposed PIQL algorithm has two main features. First, it is a data-based control design approach, where the system model $f(x, u)$ is not required. Second, it is an off-policy learning approach [2], [38], [39], which refers to evaluate a target policy $u^{(i)}$ for its Q-function $Q^{(i)}$ while following another policy $\mu$, known as behavior policy that can be exploratory. Thus, the proposed PIQL algorithm is exploration insensitive, which means that it is independent of how the behaves while the data is being collected. □

\textbf{B. Theoretical Analysis}

Some theoretical aspects of the PIQL method (i.e., Algorithm 1) are analyzed in this subsection. Its convergence is proved by demonstrating that the sequences $\{Q^{(i)}\}$ and $\{u^{(i)}\}$ generated by Algorithm 1 will converge to the optimal Q-function $Q^*$ and the optimal control policy $u^*$, respectively.

\textbf{Theorem 1.} Let $u^{(i+1)}(x)$ be given by (15), then

$$u^{(i+1)}(x) = \arg\min_{\mu} H(x, \mu, \nabla V^{(i)}).$$ \hspace{1cm} (16)

\textbf{Proof.} According to (10), equation (14) can be rewritten as

$$Q^{(i)}(x_t, \mu) = \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + V^{(i)}(x_t)$$

where $V^{(i)}(x_t) \triangleq V_{u^{(i)}}(x_t) = Q^{(i)}(x_t', u^{(i)})$. Subtracting $V^{(i)}(x_t)$ on both sides of equation (17) yields

$$Q^{(i)}(x_t, \mu) - V^{(i)}(x_t) = \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + \int_t^{t'} \frac{dV^{(i)}(x)}{d\tau}d\tau$$

$$= \int_t^{t'} \left(\mathcal{R}(x(\tau), \mu(\tau)) + [\nabla V^{(i)}(x(\tau))]^T f(x(\tau), \mu(\tau))\right) d\tau$$

$$= \int_t^{t'} H(x(\tau), \mu(\tau), \nabla V^{(i)}(x(\tau)))d\tau.$$

\hspace{1cm} i.e.,

$$Q^{(i)}(x_t, \mu) = V^{(i)}(x_t) + \int_t^{t'} H(x(\tau), \mu(\tau), \nabla V^{(i)}(x(\tau)))d\tau.$$

Then,

$$\min_{\mu} Q^{(i)}(x_t, \mu) = \min_{\mu} V^{(i)}(x_t)$$

$$+ \min_{\mu} \int_t^{t'} H(x(\tau), \mu(\tau), \nabla V^{(i)}(x(\tau)))d\tau$$

$$= V^{(i)}(x_t) + \min_{\mu} \int_t^{t'} H(x(\tau), \mu(\tau), \nabla V^{(i)}(x(\tau)))d\tau.$$ \hspace{1cm} (18)

Let

$$\mu_1(\tau) = \arg\min_{\mu} \int_t^{t'} H(x(\tau), \mu(\tau), \nabla V^{(i)}(x(\tau)))d\tau$$ \hspace{1cm} (19)

and

$$\mu_2(\tau) = \arg\min_{\mu} \int_t^{t'} H(x(\tau), \mu(\tau), \nabla V^{(i)}(x(\tau))).$$ \hspace{1cm} (20)

Thus, it follows from (20) that

$$H(x, \mu_2(\tau), \nabla V^{(i)}(x)) = \min_{\mu} H(x, \mu, \nabla V^{(i)}(x))$$

$$\leq H(x, \mu_1(\tau), \nabla V^{(i)}(x))$$

for $\forall \tau \in [t, t']$, then integrating on interval $[t, t']$ yields

$$\int_t^{t'} \int_t^{t'} H(x(\tau), \mu_2(\tau), \nabla V^{(i)}(x(\tau)))d\tau$$

$$\leq \int_t^{t'} \int_t^{t'} H(x(\tau), \mu_1(\tau), \nabla V^{(i)}(x(\tau)))d\tau.$$ \hspace{1cm} (21)
On the other hand, based on (19),
\[
\int_t^{t'} H(x(\tau), \mu_1(\tau), \nabla V^{(i)}(x(\tau)))d\tau
\]
\[
= \min_{\mu} \int_t^{t'} H(x(\tau), \mu(\tau), \nabla V^{(i)}(x(\tau)))d\tau
\]
\[
\leq \int_t^{t'} H(x(\tau), \mu_2(\tau), \nabla V^{(i)}(x(\tau)))d\tau. \tag{22}
\]
According to (21) and (22),
\[
\int_t^{t'} H(x(\tau), \mu_1(\tau), \nabla V^{(i)}(x(\tau)))d\tau
\]
\[
= \int_t^{t'} H(x(\tau), \mu_2(\tau), \nabla V^{(i)}(x(\tau)))d\tau
\]
i.e.,
\[
\min_{\mu} \int_t^{t'} H(x(\tau), \mu(\tau), \nabla V^{(i)}(x(\tau)))d\tau
\]
\[
= \int_t^{t'} \min_{\mu} H(x(\tau), \mu(\tau), \nabla V^{(i)}(x(\tau)))d\tau. \tag{23}
\]
Then, it follows from equations (18) and (23) that
\[
\min Q^{(i)}(x_i, \mu) = V^{(i)}(x_i)
\]
\[
+ \int_t^{t'} \min_{\mu} H(x(\tau), \mu(\tau), \nabla V^{(i)}(x(\tau)))d\tau. \tag{24}
\]
which means that
\[
\arg \min_{\mu} Q^{(i)}(x, \mu) = \arg \min_{\mu} H(x, \mu, \nabla V^{(i)}) = u^{(i+1)}(x). \tag{25}
\]
The proof is completed. \(\square\)

From Theorem 1, it is indicated that the policy improvement with (15) for learning a control policy \(u^{(i+1)}\) that minimizes the Q-function \(Q^{(i)}\), is theoretically equivalent to obtain a control policy that minimizes the associated Hamilton function \(H(x, \mu, \nabla V^{(i)})\).

**Theorem 2.** Let \(u^{(0)}(x) \in \Omega(\mathcal{X})\), the sequence \(\{u^{(i)}(x)\}\) be generated by Algorithm 1. Then, \(u^{(i)}(x) \in \Omega(\mathcal{X})\) for \(\forall i = 0, 1, 2, \ldots\).

**Proof.** The proof is by mathematical induction. First, \(u^{(0)}(x) \in \Omega(\mathcal{X})\), i.e., Theorem 2 holds for \(i = 0\).

Assume that Theorem 2 holds for index \(i = l\), that is, \(u^{(l)}(x) \in \Omega(\mathcal{X})\). Then, according to (6), the cost function \(V^{(l)}(x)\) associated with \(u^{(l)}(x)\) satisfies the following Hamilton function equation
\[
H(x, u^{(l)}, \nabla V^{(l)}) = 0. \tag{26}
\]
Next, we should prove that Theorem 2 still holds for index \(i = l + 1\). Under the control policy \(u^{(l+1)}(x)\), the closed-loop system is
\[
\dot{x} = f(x, u^{(l+1)}). \tag{27}
\]
Selecting \(V^{(l)}(x)\) be the Lyapunov function, and taking derivative of \(V^{(l)}(x)\) along the state of the closed-loop system (27) yields
\[
\dot{V}^{(l)} = \nabla V^{(l)} f(x, u^{(l+1)})
\]
\[
= \nabla V^{(l)} f(x, u^{(l+1)}) + S(x) + W(u^{(l+1)})
\]
\[
- S(x) - W(u^{(l+1)})
\]
\[
= H(x, u^{(l+1)}, \nabla V^{(l)}) - S(x) - W(u^{(l+1)}). \tag{28}
\]
Based on the expressions (16) and (26),
\[
H(x, u^{(l+1)}, \nabla V^{(l)}) = \min_{\mu} H(x, \mu, \nabla V^{(l)})
\]
\[
\leq H(x, u^{(l)}, \nabla V^{(l)}) = 0. \tag{29}
\]
It follows from (28) and (29) that
\[
\dot{V}^{(l)} \leq -S(x) - W(u^{(l+1)}) \leq 0.
\]
which means that the closed-loop system (27) is asymptotically stable. Thus, \(u^{(l+1)}(x) \in \Omega(\mathcal{X})\), i.e., Theorem 2 holds for index \(i = l + 1\). \(\square\)

Theorem 2 shows that giving an initial admissible control policy, all policies generated by the PIQL Algorithm 1 are admissible.

**Theorem 3.** For \(\forall (x, \mu) \in \mathcal{X} \times \mathcal{U}\), the sequences \(\{Q^{(i)}(x, \mu)\}\) and \(\{u^{(i)}(x)\}\) are generated by Algorithm 1. Then,
1) \(Q^{(i)}(x, \mu) \geq Q^{(i+1)}(x, \mu) \geq Q^*(x, \mu)\)
2) \(Q^{(i)}(x, \mu) \rightarrow Q^*(x, \mu)\) and \(u^{(i)}(x) \rightarrow u^*(x)\) as \(i \rightarrow \infty\).

**Proof.** For \(\forall i = 0, 1, 2, \ldots\), define a new iterative function sequence
\[
\bar{V}^{(i)}(x_t) = \min_{\mu} Q^{(i)}(x_t, \mu). \tag{30}
\]
Then, it follows from the expressions (14) and (30) that
\[
\bar{V}^{(i)}(x_t) \leq Q^{(i)}(x_t, u^{(i)})
\]
\[
= \int_t^{t'} \mathcal{R}(x(\tau), u^{(i)}(\tau))d\tau + V^{(i)}(x_t')
\]
\[
= V^{(i)}(x_t). \tag{31}
\]
For \(\forall x_t \in \mathcal{X}\), according to (4),
\[
V^{(i+1)}(x_t) = \int_t^{t+\Delta t} \mathcal{R}(x(\tau), u^{(i+1)}(\tau))d\tau + V^{(i+1)}(x_t')
\]
\[
= \int_t^{t+\Delta t} \mathcal{R}(x(\tau), u^{(i+1)}(\tau))d\tau + V^{(i)}(x_t')
\]
\[
- V^{(i)}(x_t') + V^{(i+1)}(x_t')
\]
\[
= \bar{V}^{(i)}(x_t) - V^{(i)}(x_t') + V^{(i+1)}(x_t')
\]
\[
\leq \bar{V}^{(i)}(x_t) - V^{(i)}(x_t') + V^{(i+1)}(x_t')
\]
\[
= \int_t^{t+\Delta t} \mathcal{R}(x(\tau), u^{(i)}(\tau))d\tau + V^{(i+1)}(x_t'). \tag{32}
\]
Similar with (32), there is
\[
V^{(i+1)}(x_{t+k\Delta t}) \leq \int_{t+k\Delta t}^{t+(k+1)\Delta t} \mathcal{R}(x(\tau), u^{(i)}(\tau))d\tau
\]
\[
+ V^{(i+1)}(t + x_{(k+1)\Delta t}). \tag{33}
\]
for \(\forall k\). Note from Theorem 2 that \(u^{(i+1)}(x) \in \mathcal{U}(\mathcal{X})\), then \(x_t = 0\) as \(t \to \infty\), that is \(V^{(i+1)}(x_{t \to \infty}) = 0\). Thus, it follows from (32) and (33) that

\[
V^{(i+1)}(x_t) \leq \int_t^{t+\Delta t} R(x(\tau), u^{(i)}(\tau))d\tau + \int_{t+\Delta t}^{t+2\Delta t} R(x(\tau), u^{(i)}(\tau))d\tau + V^{(i+1)}(x_{t+2\Delta t})
\]

\[
\leq \int_t^{t+\Delta t} R(x(\tau), u^{(i)}(\tau))d\tau + \int_{t+\Delta t}^{t+2\Delta t} R(x(\tau), u^{(i)}(\tau))d\tau + \cdots + \int_{t+(k-1)\Delta t}^{t+k\Delta t} R(x(\tau), u^{(i)}(\tau))d\tau + \cdots + V^{(i+1)}(x_{t \to \infty})
\]

\[
= V^{(i)}(x_t).
\] (34)

For \(\forall (x_t, \mu) \in \mathcal{X} \times \mathcal{U}\), by using the expressions (14) and (34), there is

\[
Q^{(i+1)}(x_t, \mu) = \int_t^{t'} R(x(\tau), \mu(\tau))d\tau + V^{(i)}(x_t')
\]

\[
\leq \int_t^{t'} R(x(\tau), \mu(\tau))d\tau + V^{(i)}(x_t') = Q^{(i)}(x_t, \mu).
\]

According to (10) and (11),

\[
Q^{(i)}(x_t, \mu) = \int_t^{t'} R(x(\tau), \mu(\tau))d\tau + V^{(i)}(x_t') \geq \int_t^{t'} R(x(\tau), \mu(\tau))d\tau + V^*(x_t') = Q^*(x_t, \mu).
\]

The part 1) of Theorem 3 is proved.

2) From the part 1) of Theorem 3, \(\{Q^{(i)}(x_t, \mu)\}\) is a nonincreasing sequence and bounded below by \(Q^*(x_t, \mu)\). Considering that a bounded monotone sequence always has a limit, denote \(Q^{(\infty)}(x_t, \mu) \triangleq \lim_{i \to \infty} Q^{(i)}(x_t, \mu)\) and then \(u^{(\infty)}(x) \triangleq \arg\min_{\mu} Q^{(\infty)}(x, \mu)\).

Based on the proof of the part 1) in Theorem 3, \(\{V^{(i)}(x)\}\) is also a nonincreasing sequence and bounded below by \(V^*(x)\). Denote \(V^{(\infty)}(x) \triangleq \lim_{i \to \infty} V^{(i)}(x)\). It follows from Theorem 1 (by simplify replacing \(V^{(i)}\) with \(V^{(\infty)}\)) that

\[
u^{(\infty)}(x) = \arg\min_{\mu} Q^{(\infty)}(x, \mu) = \arg\min_{\mu} H(x, \mu, \nabla V^{(\infty)}),
\] (35)

Since \(u^{(\infty)}(x) \in \mathcal{U}(\mathcal{X})\), it is based on (6) that \(H(x, u^{(\infty)}, \nabla V^{(\infty)}) = 0\), which means that \(V^{(\infty)}(x)\) satisfies the HJBE (8). According to the uniqueness of the HJB ESs solution, \(V^{(\infty)}(x) = V^*(x)\).

Then, from (14),

\[
Q^{(\infty)}(x_t, \mu) = \lim_{i \to \infty} Q^{(i)}(x_t, \mu)
\]

\[
= \lim_{i \to \infty} \int_t^{t'} R(x(\tau), \mu(\tau))d\tau + \lim_{i \to \infty} Q^{(i)}(x_t', u^{(i)}(x_t'))
\]

\[
= \int_t^{t'} R(x(\tau), \mu(\tau))d\tau + Q^{(\infty)}(x_t', u^{(\infty)}(x_t'))
\]

\[
= \int_t^{t'} R(x(\tau), \mu(\tau))d\tau + V^{(\infty)}(x_t')
\]

\[
= \int_t^{t'} R(x(\tau), \mu(\tau))d\tau + V^*(x_t') = Q^*(x_t, \mu).
\] (36)

The substitution of (36) into (35) yields \(u^{(\infty)}(x) = \arg\min_{\mu} Q^*(x, \mu) = u^*(x)\). The proof is completed. □

In Theorem 3, the convergence of the proposed PIQL algorithm is proved. It is demonstrated that \(\{Q^{(i)}(x_t, \mu)\}\) is a bounded nonincreasing sequence that converges to the optimal Q-function \(Q^*(x_t, \mu)\), and then the control sequence \(\{u^{(i)}(x)\}\) converges to the optimal control policy \(u^*(x)\).

C. The Method of Weighted Residuals

In the PIQL algorithm (i.e., Algorithm 1), the policy evaluation requires the solution of the equation (14) for unknown Q-function \(Q^{(i)}(x_t, \mu)\). In this subsection, the method of weighted residuals (MWR) is developed. Let \(\Psi(x, \mu) \triangleq \{\psi_j(x, \mu)\}_{j=1}^\infty\) be complete set of linearly independent basis functions, such that \(\psi_j(0,0) = 0\) for \(\forall j\). Then, the solution \(Q^{(i)}(x_t, \mu)\) of the iterative equation (14) can be expressed as linear combination of basis function set \(\Psi(x, \mu)\), i.e., \(Q^{(i)}(x_t, \mu) = \sum_{j=1}^\infty \theta_j^{(i)} \psi_j(x_t, \mu)\) which are assumed to converge pointwise in \(\mathcal{X} \times \mathcal{U}\). The trial solution for \(Q^{(i)}(x_t, \mu)\) can be respectively taken by truncating the series to

\[
\hat{Q}^{(i)}(x_t, \mu) = \sum_{j=1}^L \theta_j^{(i)} \psi_j(x_t, \mu) = \Psi^T_L(x_t, \mu)\theta^{(i)}
\] (37)

where \(\theta^{(i)} \triangleq [\theta_1^{(i)} ... \theta_L^{(i)}]^T\) is an unknown weight vector, \(\Psi_L(x, \mu) \triangleq [\psi_1(x, \mu) ... \psi_L(x, \mu)]^T\). By using (15) and (37), the estimated solution for \(Q^{(i)}(x_t, \mu)\) is given by

\[
\hat{u}^{(i)}(x) = \arg\min_{\mu} \hat{Q}^{(i-1)}(x_t, \mu).
\] (38)

Due to the truncation error of the trial solution (37), the replacement of \(Q^{(i)}(x_t, \mu)\) and \(u^{(i)}(x)\) in the iterative equation (14) with \(\hat{Q}^{(i)}(x_t, \mu)\) and \(\hat{u}^{(i)}(x)\) respectively, yields the following residual error:

\[
\sigma^{(i)}(x_t, \mu, x_t') \triangleq \hat{Q}^{(i)}(x_t, \mu) - Q^{(i)}(x_t', \mu) - \int_t^{t'} R(x(\tau), \mu(\tau))d\tau
\]

\[
= \Psi_L(x_t, \mu) - \Psi_L(x_t', \mu)\hat{u}^{(i)}(x_t')\|^T \theta^{(i)}
\]

\[
- \int_t^{t'} R(x(\tau), \mu(\tau))d\tau
\]

\[
= \hat{\Psi}^{(i)}(x_t, \mu, x_t')\theta^{(i)} - \pi(x_t, \mu)
\] (39)
where $\pi^{(i)}(x_t, \mu, x_t') = [p_1^{(i)}(x_t, \mu, x_t') \cdots p_L^{(i)}(x_t, \mu, x_t')]$ and $\pi(x_t, \mu) \triangleq \int_{t}^{t'} R(x(\tau), \mu(\tau)) d\tau$, with $p_j^{(i)}(x_t, \mu, x_t') = \psi_j(x_t, \mu) - \psi_j(x_t', \mu(t))$, $j = 1, \ldots, L$. In the MWR, the unknown constant vector $\theta^{(i)}$ can be solved in such a way that the residual error $\sigma^{(i)}(x_t, \mu, x_t')$ (for $\forall t, t' \geq 0$) of (39) is forced to be zero in some average sense. The weighted integrals of the residual are set to zero:

$$\left\langle W_L^{(i)}(x, \mu, x'), \sigma^{(i)}(x, \mu, x') \right\rangle_D = 0.$$  

(40)

where $W_L^{(i)}(x, \mu, x') \triangleq [\omega_1^{(i)}(x, \mu, x') \cdots \omega_L^{(i)}(x, \mu, x')]^T$ is named the weighted function vector. Then, the substitution of (39) into (40) yields,

$$\left\langle W_L^{(i)}(x, \mu, x'), \theta^{(i)} \right\rangle_D - \left\langle W_L^{(i)}(x, \mu, x'), \pi(x, \mu) \right\rangle_D = 0$$

where the notations $\left\langle W_L^{(i)}, \theta^{(i)} \right\rangle_D$ and $\left\langle W_L^{(i)}, \pi \right\rangle_D$ are given by

$$\left\langle W_L^{(i)}, \theta^{(i)} \right\rangle_D \triangleq \left[ \left\langle \omega_1^{(i)}, \theta_1^{(i)} \right\rangle_D \ldots \left\langle \omega_L^{(i)}, \theta_L^{(i)} \right\rangle_D \right]$$

and

$$\left\langle W_L^{(i)}, \pi \right\rangle_D \triangleq \left[ \left\langle \omega_1^{(i)}, \pi \right\rangle_D \ldots \left\langle \omega_L^{(i)}, \pi \right\rangle_D \right]^T.$$  

(41)

and thus $\theta^{(i+1)}$ can be obtained with

$$\theta^{(i+1)} = \left( W_L^{(i)}, \pi \right)^{-1}_D \left( W_L^{(i)}, \pi \right)_D.$$  

Note that the computations of $\left\langle W_L^{(i)}, \theta^{(i)} \right\rangle_D$ and $\left\langle W_L^{(i)}, \pi \right\rangle_D$ involve many numerical integrals on domain $D$, which are computationally expensive. Thus, the Monte-Carlo integration method [40] is introduced, which is especially competitive on multi-dimensional domain. We now illustrate the Monte-Carlo integration for computing $\left\langle W_L^{(i)}(x, \mu, x'), \theta^{(i)}(x, \mu, x') \right\rangle_D$. Let $I_D \triangleq \int_D d(x, \mu, x')$, and $S_M \triangleq \{(x[k], \mu[k], x[k]') | (x[k], \mu[k], x[k]') \in D, k = 1, \ldots, M\}$ be the set that sampled on domain $D$, where $M$ is size of sample set $S_M$. Then, $\left\langle W_L^{(i)}(x, \mu, x'), \theta^{(i)}(x, \mu, x') \right\rangle_D$ is approximately computed with

$$\left\langle W_L^{(i)}(x, \mu, x'), \theta^{(i)}(x, \mu, x') \right\rangle_D = \frac{1}{M} \sum_{k=1}^{M} W_L^{(i)}(x[k], \mu[k], x'[k]) \theta^{(i)}(x[k], \mu[k], x'[k])$$

$$= \frac{I_D}{M} \sum_{k=1}^{M} W_L^{(i)}(x[k], \mu[k], x'[k]) \theta^{(i)}(x[k], \mu[k], x'[k])$$

$$= \frac{I_D}{M} (W^{(i)})^T \theta^{(i)}.$$  

(42)

Similarly,

$$\left\langle W_L^{(i)}(x, \mu, x'), \pi(x, \mu) \right\rangle_D = \frac{I_D}{M} \sum_{k=1}^{M} W_L^{(i)}(x[k], \mu[k], x'[k]) \pi(x[k], \mu[k])$$

$$= \frac{I_D}{M} (W^{(i)})^T \pi.$$  

(43)

where $\pi \triangleq \left[ \pi(x[1], \mu[1]) \ldots \pi(x[M], \mu[M]) \right]^T$. Then, the substitution of (42) and (43) into (41) yields,

$$\theta^{(i)} = \left( (W^{(i)})^T \pi \right)^{-1} (W^{(i)})^T \pi.$$  

(44)

It is observed that the sample set $S_M$ is arbitrary on domain $D$, based on which $W^{(i)}$, $Z^{(i)}$ and $\pi$ can be computed and then the unknown parameter vector $\theta^{(i)}$ is obtained with the expression (44) accordingly.

The above MWR is for solving an iterative equation (15) in the PIQL algorithm. Based on the expression (44), the implementation procedure of the PIQL algorithm is given as follows:

**Algorithm 2. Implementation of PIQL**

- **Step 1:** Collect real system data $(x_k, \mu_k, x_k')$ for sample set $S_M$, and then compute $\Psi_L(x_k, \mu_k)$ and $\eta$;
- **Step 2:** Let $u^{(0)}(x) \in U(X)$, and $i = 0$;
- **Step 3:** Compute $W^{(i)}$ and $Z^{(i)}$, and then update parameter vector $\theta^{(i)}$ with (44);
- **Step 4:** Update control policy $\hat{u}^{(i+1)}(x)$ based on (38) with index $i + 1$;
- **Step 5:** If $||\theta^{(i)} - \theta^{(i-1)}|| \leq \xi$ (i.e., $\xi$ is a small positive number), stop iteration, else, let $i = i + 1$ and go back to Step 3. $\square$

**IV. VALUE ITERATION BASED Q-LEARNING**

Generally, RL involves two basic frameworks: policy iteration and value iteration. Section III gives a PIQL method for model-free optimal control design of the system (1). In this section, we proposed a value iteration based QL (VIQL) algorithm, which is presented as follows:

**Algorithm 3. Value iteration based Q-learning**

- **Step 1:** Let $Q^{(i)}(x, \mu) \geq 0$ be an initial Q-function. Let $i = 1$;
- **Step 2:** (Policy improvement) Update control policy with:

$$u^{(i)}(x) \triangleq \arg \min_{\mu} Q^{(i-1)}(x, \mu);$$  

(45)

- **Step 3:** (Policy evaluation) Solve the iterative equation

$$Q^{(i)}(x_t, \mu) \triangleq \int_t^{t'} R(x(\tau), \mu(\tau)) d\tau + Q^{(i-1)}(x_t', u^{(i)});$$  

(46)

for unknown Q-function $Q^{(i)}$. 

The above MWR is for solving an iterative equation (15) in the PIQL algorithm. Based on the expression (44), the implementation procedure of the PIQL algorithm is given as follows:
Step 4: Let $i = i + 1$, go back to Step 2 and continue. □

**Theorem 4.** For $\forall (x, \mu) \in X \times U$, let $\{Q^{(i)}(x, \mu)\}$ be the sequence generated by Algorithm 3. Let $Q^{(0)}(x, \mu)$ be an arbitrary Q-function of a stable control policy, then

1) $Q^{(i)}(x, \mu) \geq Q^{(i+1)}(x, \mu) \geq Q^*(x, \mu)$
2) $Q^{(i)}(x, \mu) \rightarrow Q^*(x, \mu)$ and $u^{(i)}(x) \rightarrow u^*(x)$ as $i \rightarrow \infty$.

**Proof.** 1) The proof is by mathematical induction.

Without loss of generality, let $Q^{(0)}(x, \mu)$ denotes the Q-function of a stable control policy $\psi(x)$. Then, it follows from the definition of Q-function (9) and (10) that

$$Q^{(0)}(x_t, \mu) = Q_v(x_t, \mu) = \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + Q^{(0)}(x'_t, v). \quad (47)$$

According to (45)-(47),

$$Q^{(1)}(x_t, \mu) = \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + Q^{(0)}(x'_t, u^{(1)})$$

$$\leq \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + \min_{\mu} Q^{(0)}(x'_t, \mu)$$

$$\leq \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + Q^{(0)}(x'_t, v)$$

$$= Q^{(0)}(x_t, \mu)$$

which means that the inequality $Q^{(i)}(x, \mu) \geq Q^{(i+1)}(x, \mu)$ holds for $i = 0$.

Assume that the inequality $Q^{(i)}(x, \mu) \geq Q^{(i+1)}(x, \mu)$ holds for $i = l - 1$, i.e.,

$$Q^{(l)}(x, \mu) \leq Q^{(l-1)}(x, \mu). \quad (48)$$

Based on the expression (45),

$$Q^{(l)}(x, u^{(l)}) = \min_{\mu} Q^{(l)}(x, \mu) \leq Q^{(l)}(x, \mu) \quad (49)$$

for $\forall x, \mu$. Then, it follows from (46), (48) and (49) that

$$Q^{(l+1)}(x_t, \mu) = \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + Q^{(l)}(x'_t, u^{(l+1)})$$

$$\leq \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + \min_{\mu} Q^{(l)}(x'_t, \mu)$$

$$\leq \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + Q^{(l)}(x'_t, u^{(l)})$$

$$\leq \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + Q^{(l-1)}(x'_t, u^{(l)})$$

$$= Q^{(l)}(x_t, \mu) \quad (50)$$

which means that the inequality $Q^{(i)}(x, \mu) \geq Q^{(i+1)}(x, \mu)$ holds for $i = l$.

Next, we will prove $Q^{(i)}(x, \mu) \geq Q^*(x, \mu)$. Considering $Q^{(i-1)}(x, \mu) \geq Q^{(i)}(x, \mu)$ holds for $\forall (x, \mu) \in X \times U$, then

$$Q^{(i-1)}(x'_t, u^{(i)}) \geq Q^{(i)}(x'_t, u^{(i)}) \quad (51)$$

Combining (46) and (51), there is

$$Q^{(i)}(x_t, \mu) = \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + Q^{(i-1)}(x'_t, u^{(i)})$$

$$\geq \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + Q^{(i)}(x'_t, u^{(i)})$$

$$= \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + V^{(i)}(x'_t)$$

$$\geq \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + V^*(x'_t)$$

$$= Q^*(x_t, \mu). \quad (52)$$

The part 1) of Theorem 4 is proved.

2) The part 2) of Theorem 4 can be proved similarly as that for the part 2) of Theorem 3, and it is omitted for briefness. The proof is completed. □

It is noted that an iterative equation (46) should be solved in each iteration of the VIQL algorithm. Similar with Subsection III-C, the MWR can be used to solve the equation (46). To avoid repeat, we omit the detailed derivation of the MWR and give the parameter update law directly as

$$\theta^{(i)} = [WTZ]^{-1}W^T(\eta + Z^{(i)}\theta^{(i-1)}) \quad (53)$$

where the notations are given by

$$W = [W_L(x|1, \mu|1) \cdots W_L(x|M, \mu|M)]^T$$

$$Z = [\tilde{p}^T(x|1, \mu|1) \cdots \tilde{p}^T(x|M, \mu|M)]^T$$

$$Z^{(i)} = [\tilde{p}^{(i)}(x'_t, \tilde{u}^{(i)}(x'_t|1)) \cdots \tilde{p}^{(i)}(x'_t, \tilde{u}^{(i)}(x'_t|M))]^T$$

$$\eta = [\pi(x|1, \mu|1) \cdots \pi(x|M, \mu|M)]^T$$

with $W_L(x, \mu) = [\omega_1(x, \mu) \cdots \omega_L(x, \mu)]^T$ be the weighted function vector, $\tilde{p}_j(x, \mu) = [\tilde{p}_j(x, \mu) \cdots \tilde{p}_j(x, \mu)]$ with $\tilde{p}_j(x, \mu) = \tilde{p}_j(x, \mu), j = 1, ..., L$, and $\pi(x, \mu) = \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau$.

Based on the parameter update law (53), the implementation procedure of the VIQL algorithm is presented below:

**Algorithm 4. Implementation of VIQL**

Step 1: Collect real system data $(x_k, \mu_k, x'_k)$ for sample set $S_M$, and then compute $W, Z$ and $\eta$;

Step 2: Let $\theta^{(0)}$ be the initial parameter vector, and $i = 1$;

Step 3: Update control policy $\tilde{u}^{(i)}(x)$ with (38);

Step 4: Compute $Z^{(i)}$, and then update parameter vector $\theta^{(i)}$ with (53);

Step 5: If $||\theta^{(i)} - \theta^{(i-1)}|| \leq \xi$ (\(\xi\) is a small positive number), stop iteration, else, let $i = i + 1$ and go back to Step 3. □

**Remark 2.** There are several similarities and differences between PIQL and VIQL algorithms, which are summarized as follows: 1) Both of them generate a non-increasing Q-function sequence, which converges to the optimal Q-function.

2) Both algorithms are model-free method, which learns the optimal control policy from real system data rather than using...
V. Q-LEARNING FOR LQR PROBLEM

In the section, we simplify the developed PIQL and VIQL methods for solving the model-free linear quadratic regulation (LQR) problem. Consider a linear version of system (1):
\[ \dot{x}(t) = Ax(t) + Bu(t), x(0) = x_0 \] (54)
where \( A \in \mathbb{R}^{n \times n} \) and \( B \in \mathbb{R}^{n \times m} \) are unknown matrices, and the quadratic cost function:
\[ J(x_0, u) = \int_0^{\infty} \left( \|x(t)\|^2_S + \|u(t)\|^2_W \right) dt \]
(55)
where matrices \( S, W > 0 \).

For the LQR problem of system (54) with cost function (55), its optimal Q-function can be given by
\[ Q^*(x, \mu) = \begin{bmatrix} x \\ \mu \end{bmatrix}^T G \begin{bmatrix} x \\ \mu \end{bmatrix} \]
(56)
where \( G \geq 0 \) is a block matrix denoted as
\[ G \triangleq \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}. \]
According to (13), its optimal control policy is given by
\[ u^*(x) = \arg \min_{\mu} Q^*(x, \mu) = -G_{22}^{-1} G_{12}^T x. \]
(57)
Thus, denote the Q-function \( Q^{(i)}(x, \mu) \) as
\[ Q^{(i)}(x, \mu) = \begin{bmatrix} x \\ \mu \end{bmatrix}^T G^{(i)} \begin{bmatrix} x \\ \mu \end{bmatrix} \]
(58)
where \( G^{(i)} \) is given by
\[ G^{(i)} \triangleq \begin{bmatrix} G^{(i)}_{11} & G^{(i)}_{12} \\ G^{(i)}_{21} & G^{(i)}_{22} \end{bmatrix}. \]

A. PIQL for LQR Problem

By using the expression (58), the PIQL Algorithm 1 can be simplified for solving the model-free LQR problem of system (54), which is given as follows:

Algorithm 5. PIQL for LQR problem

- **Step 1:** Let \( u^{(0)}(x) \in \mathfrak{U}(\mathcal{X}) \) be an initial stabilizing control policy, and \( i = 0 \);

- **Step 2:** (Policy evaluation) Solve the equation
\[ \begin{bmatrix} x_t \\ \mu \end{bmatrix}^T G^{(i)} \begin{bmatrix} x_t \\ \mu \end{bmatrix} = \int_t^\infty R(x(\tau), \mu(\tau)) d\tau + \begin{bmatrix} \dot{x}_t \\ u^{(i)}(t) \end{bmatrix}^T G^{(i)} \begin{bmatrix} \dot{x}_t \\ u^{(i)}(t) \end{bmatrix} \]
(59)
for unknown unknown matrix \( G^{(i)} \);

- **Step 3:** (Policy improvement) Update control policy with
\[ u^{(i+1)}(x) = -\left( G^{(i)}_{22} - 1 \right) G^{(i)}_{12}^T x ; \]
(60)

- **Step 4:** Let \( i = i + 1 \), go back to Step 2 and continue.

According to Theorem 3, Algorithm 5 generates a non-increasing matrix sequence \( \{ G^{(i)} \} \) that converges to \( G \).

For each iteration of Algorithm 5, the iterative equation (59) should be solved for the unknown unknown matrix \( G^{(i)} \). It is observed that \( G^{(i)} \) is a \((n+m) \times (n+m) \) symmetric matrix, which has \((n+m)(n+m+1)/2 \) unknown parameters. Letting
\[ \Psi_L(x, \mu) = [x_1^T, x_1, x_2, \ldots, x_1 \mu_1, \ldots, x_1 \mu_m, x_2^T, \ldots, \mu_m^2]^T \]
(61)
and
\[ \theta^{(i)} = [(g^{(i)}_{11})^2, 2 g^{(i)}_{11} g^{(i)}_{12}, \ldots, 2 g^{(i)}_{11} g^{(i)}_{n+1}, \ldots, 2 g^{(i)}_{11} g^{(i)}_{n+m}, (g^{(i)}_{22})^2, 2 g^{(i)}_{22} g^{(i)}_{n+1}, \ldots, (g^{(i)}_{n+m} g^{(i)}_{n+m+1})^2]^T \]
(62)
the iterative equation (59) can be equivalently rewritten as
\[ \psi_L(x_t, \mu) - \psi_L(x_t, \tilde{u}^{(i)}(x_t^*))^T \theta^{(i)} = \pi(x_t, \mu) \]
(63)
where \( \pi(x_t, \mu) \triangleq \int_t^\infty R(x(\tau), \mu(\tau)) d\tau \). By using the data set \( \mathcal{S}_m \) collected from real system, the following least-square scheme is obtained for updating the unknown parameter vector \( \theta^{(i)} \)
\[ \theta^{(i)} = \left[(Z^{(i)})^T Z^{(i)}\right]^{-1} (Z^{(i)})^T \eta \]
(64)
where
\[ Z^{(i)} = \psi_L(x_{[1]}, \mu_{[1]}) - \psi_L(x_{[1]}, \tilde{u}^{(i)}(x_{[1]}^*)) \]
\[ \ldots \psi_L(x_{[M]}, \mu_{[M]}) - \psi_L(x_{[M]}, \tilde{u}^{(i)}(x_{[M]}^*)) \]
\[ \eta = [\pi(x_{[1]}, \mu_{[1]}) \ldots \pi(x_{[M]}, \mu_{[M]})]^T. \]

With the least-square scheme (64), the implementation procedure of Algorithm 5 will be obtained similarly with Algorithm 2.

Remark 3. For the LQR problem of the linear system (54), there is no residual error in the expression (63) because the Q-function \( Q(x, \mu) \) can be exactly represented by basic function vector (62). □

B. VIQL for LQR Problem

With the expression (58), the VIQL Algorithm 3 can be simplified for solving the model-free LQR problem of system (54), which is given as follows:

- **Step 1:** Let \( u^{(0)}(x) \in \mathfrak{U}(\mathcal{X}) \) be an initial stabilizing control policy, and \( i = 0 \);

- **Step 2:** (Policy evaluation) Solve the equation
\[ \begin{bmatrix} x_t \\ \mu \end{bmatrix}^T G^{(i)} \begin{bmatrix} x_t \\ \mu \end{bmatrix} = \int_t^\infty R(x(\tau), \mu(\tau)) d\tau + \begin{bmatrix} \dot{x}_t \\ u^{(i)}(t) \end{bmatrix}^T G^{(i)} \begin{bmatrix} \dot{x}_t \\ u^{(i)}(t) \end{bmatrix} \]
(59)
for unknown unknown matrix \( G^{(i)} \);

- **Step 3:** (Policy improvement) Update control policy with
\[ u^{(i+1)}(x) = -\left( G^{(i)}_{22} - 1 \right) G^{(i)}_{12}^T x ; \]
(60)

- **Step 4:** Let \( i = i + 1 \), go back to Step 2 and continue. □
Algorithm 6. VIQL for LQR problem

\[ \text{Step 1: } \text{Let } G^{(0)} \geq 0 \text{ and } i = 1; \]
\[ \text{Step 2: (Policy improvement)} \text{ Update control policy with: } u^{(i)}(x) = -[G^{(i-1)}_{22}]^{-1}[G^{(i-1)}_{12}]^T x; \quad (65) \]
\[ \text{Step 3: (Policy evaluation)} \text{ Solve the iterative equation } \]
\[ \begin{bmatrix} x_t \\ \mu \end{bmatrix}^T G^{(i)} \begin{bmatrix} x_t \\ \mu \end{bmatrix} = \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau + \begin{bmatrix} x_t' \\ u^{(i)}(x_t) \end{bmatrix}^T G^{(i-1)} \begin{bmatrix} x_t' \\ u^{(i)}(x_t) \end{bmatrix}, \quad (66) \]

for unknown unknown matrix \( G^{(i)} \);
\[ \text{Step 4: Let } i = i + 1, \text{ go back to Step 2 and continue.} \]

Based on Theorem 4, Algorithm 6 generates a non-increasing matrix sequence \( \{G^{(i)}\} \) that converges to \( G \). By using the expressions (61) and (62) the iterative equation (66) can be equivalently rewritten as
\[ \Psi_N(x_t, \mu)\theta^{(i)} = \pi(x_t, \mu) + \Psi_N(x_t', \eta(x_t'))\theta^{(i-1)} \quad (67) \]
where \( \pi(x_t, \mu) = \int_t^{t'} \mathcal{R}(x(\tau), \mu(\tau))d\tau \).

With the least-square scheme (68), the implementation procedure of Algorithm 6 will be obtained similarly with Algorithm 4, which is omitted for brevity.

VI. SIMULATION STUDIES

To test the effectiveness of the proposed QL algorithms, computer simulation studies are conducted on a linear F-16 aircraft plant and a numerical nonlinear system. For all simulations, the algorithm stop accuracy is set as \( \xi = 10^{-5} \), the system data set \( S_M \) is collected randomly and its size is set as \( M = 100 \).

A. Example 1: Linear F-16 Aircraft Plant

Consider a linear F16 aircraft plant [41]–[43], where the system dynamics is described with (54), and the system matrices given by:
\[ A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \]
where the system state vector is \( x = [\alpha \ q \ \delta_e]^T \), \( \alpha \) denotes the angle of attack, \( q \) is the pitch rate, \( \delta_e \) is the elevator deflection angle, and the control input \( u \) is the elevator actuator voltage. Select \( S, W \) as unit matrices for quadratic cost function (55). Then, solve the associated Riccati algebraic equation with the MATLAB command CARE, we can obtain the optimal control policy given by \( u^*(x) = Kx \), where \( K = [-0.1352 \ -0.1501 \ 0.4329] \) is the optimal control gain.

For the LQR problem, it follows from (61) and (62) that the basic function set is
\[ \Psi_L(x, \mu) = [x_1^2 \ x_1 x_2 \ x_1 x_3 \ x_1 \mu \ x_2^2 \ x_2 x_3 \ x_2 \mu \ x_3^2 \ x_3 \mu \ \mu^2]^T \quad (69) \]
and the parameter vector is
\[ \theta^{(i)} = [(g_1^{(i)})^2 \ 2g_{1,2}^{(i)} \ 2g_{1,3}^{(i)} \ (g_{1,4}^{(i)})^2 \ 2g_{2,3}^{(i)} \ 2g_{2,4}^{(i)} \ (g_{3,3}^{(i)})^2 \ 2g_{3,4}^{(i)} \ (g_{4,4}^{(i)})^2]^T. \]

According to the policy improvement rule (60), the iterative control gain is \( K^{(i)} = \left[ k_1^{(i)} \ k_2^{(i)} \ k_3^{(i)} \right] = -(g_{4,4}^{(i)})^{-1}[g_{1,4}^{(i)} \ g_{2,4}^{(i)} \ g_{3,4}^{(i)}] = -0.5(\theta_{10}^{(i)})^{-1}[\theta_{1}^{(i)} \ \theta_{7}^{(i)} \ \theta_{9}^{(i)}]. \]

First, the PIQL algorithm (i.e., Algorithm 2) with parameters update law (64) is used to solve this model-free LQR
Fig. 3. For example 1, the iterative control gain $K^{(i)}$ obtained by PIQL algorithm.

Fig. 4. For example 1, all parameters of vector $\theta^{(i)}$ obtained by VIQL algorithm.

Fig. 5. For example 1, the norm $\|\theta^{(i)}\|$ obtained by VIQL algorithm.

Fig. 6. For example 1, the iterative control gain $K^{(i)}$ obtained by VIQL algorithm.

Problem. Figures 1-3 gives the simulation results, where the PIQL algorithm achieves convergence at $i = 5$ iteration. Figures 1 and 2 show the parameter vector $\theta^{(i)}$ and its norm $\|\theta^{(i)}\|$ respectively, where the parameter vector converges to $\theta^{(5)} = [1.4250 \ 2.3374 \ -0.2734 \ -0.0068 \ 1.4355 \ -0.3034 \ -0.0076 \ 0.4375 \ 0.0216 \ 0.0254]^T$.

Figure 3 shows the control gain $K^{(i)}$ at each iteration, where the dot lines represent idea value of the optimal control gain $K$. It is observed that $K^{(i)}$ converges to $K^{(5)} = [0.1343 \ 0.1488 \ -0.4256]$ which approaches to the optimal control gain $K$.

From the simulation results, it is found that the PIQL algorithm achieves much faster convergence than VIQL algorithm.

B. Example 2: Numerical Nonlinear System

This numerical example is constructed by using the converse HJB approach [44]. The system model is given as follows:

\[
\dot{x} = \begin{bmatrix}
-x_1 + x_2 \\
-0.5(x_1 + x_2) + 0.5x_1^2x_2
\end{bmatrix} + \begin{bmatrix}
0 \\
x_1
\end{bmatrix} u
\] (70)

where $x_0 = [0.1 \ 0.1]^T$. With the choice of $Q(x) = x^T x$ and $W(u) = u^2$ in the cost function (2), the optimal control policy is $u^*(x) = -x_1x_2$. 

Select the basic function vector as
\[
\Psi_L(x, \mu) = \begin{bmatrix} x_1^2 & x_1x_2 & x_2^2 & x_1^3 & x_1^2x_2 & x_1x_2^2 \\
x_3^3 & x_1^3 & x_1^2x_2 & x_1x_2^2 & x_1x_2 & x_2^3 \end{bmatrix}^T
\]
of size \( L = 18 \), and the weighted basic function vector as \( W_L = \Psi_L \). According to the policy improvement rule \((15)\), the iterative control policy is
\[
u^{(i)}(x) = K^{(i)}[x_1 \ x_2 \ x_1^2 \ x_1x_2 \ x_2^2]^T
\]
with control gain \( K^{(i)} = [k_1^{(i)} \ k_2^{(i)} \ k_3^{(i)} \ k_4^{(i)} \ k_5^{(i)}] = -0.5[\theta_8^{(i)} - 1][\theta_{13}^{(i)} \ \theta_{14}^{(i)} \ \theta_{15}^{(i)} \ \theta_{16}^{(i)} \ \theta_{17}^{(i)}] \). Consider that the optimal control policy is \( \nu^*(x) = -x_1x_2 \), which can be represented as \( \nu^*(x) = K[x_1 \ x_2 \ x_1^2 \ x_1x_2 \ x_2^2]^T \) with the idea optimal control gain \( K = [0 \ 0 \ 0 \ -1 \ 0] \).

First, the PIQL algorithm (i.e., Algorithm 2) is used to solve the model-free optimal control problem of the system \((70)\). It is observed that the algorithm achieves convergence at \( i = 7 \) iteration, where the parameter vector \( \theta^{(i)} \) converges to
\[
\theta^{(7)} = \begin{bmatrix} 0.5097 & 0.0078 & 0.9977 & -0.0189 & -0.0142 \\
-0.0060 & 0.0120 & 0.0091 & 0.0038 & 0.0215 \\
-0.0057 & -0.0078 & -0.0003 & 0.0000 & 0.0006 \\
0.0497 & 0.0004 & 0.0252 \end{bmatrix}^T.
\]
and the iterative control gain \( K^{(i)} \) converges to
\[
K^{(7)} = \begin{bmatrix} 0.0067 & -0.0009 & -0.0112 & -0.9860 & -0.0076 \end{bmatrix}.
\]

Figure 7-8 give the the parameter vector \( \theta^{(i)} \) and the norm \( \|\theta^{(i)}\| \) at each iteration, respectively. Figure 9 shows the control gain \( K^{(i)} \) at each iteration, where the dot lines represent idea value of the optimal control gain \( K \). It is noted that good convergence is achieved. Then, the convergent control policy \( \hat{\nu}^{(7)}(x) \) is used for real control of the system \((70)\). It is observed that the cost is 0.0150. Figure 10 shows the closed-loop trajectories of system state and control signal.

Next, by using the same basic function vector \((71)\), the VIQL algorithm (i.e., Algorithm 4) is employed to solve the model-free optimal control problem of system \((70)\). Figures
Fig. 11. For example 2, all parameters of vector $\theta^{(i)}$ obtained by VIQL algorithm.

Fig. 12. For example 2, the norm $\|\theta^{(i)}\|$ obtained by VIQL algorithm.

11-13 give the simulation results, where the VIQL algorithm achieves convergence at $i = 390$ iteration. Figures 11 and 12 show the parameter vector $\theta^{(i)}$ and its norm $\|\theta^{(i)}\|$ respectively, where the parameter vector converges to

$$\theta^{(390)} = \begin{bmatrix} 0.4958 & 0.0132 & 0.9983 & -0.0021 & 0.0022 \\ -0.0214 & 0.0112 & 0.0029 & -0.0102 & 0.0219 \\ 0.0047 & -0.0089 & -0.0003 & -0.0001 & 0.0007 \\ 0.0495 & 0.0006 & 0.0252 \end{bmatrix}^T.$$  

Figure 13 shows the control gain $K^{(i)}$ at each iteration, where the dot lines represent idea value of the optimal control gain $K$. It is observed that $K^{(i)}$ converges to

$$K^{(390)} = \begin{bmatrix} 0.0055 & 0.0016 & -0.0130 & -0.9813 & -0.0117 \end{bmatrix}$$

which approaches to the optimal control gain $K$. With the convergent control policy $\hat{u}^{(390)}(x)$, closed-loop simulation is conducted on the real system (70). It is observed that the cost is 0.0150, and the the closed-loop trajectories of system state and control signal are almost the same as that in Figure 10, which is omitted here.

VII. CONCLUSIONS

In this paper, the model-free optimal control problem of general nonlinear continuous-time systems is considered, and two QL methods are developed. First, PIQL and VIQL algorithms are proposed and their convergence is proved. For implementation purpose, the MWR is employed to derive a update law for unknown parameters. Both PIQL and VIQL algorithms are model-free off-policy RL methods, which learn the optimal control policy offline from real system data and then be used for real-time control. Subsequently, PIQL and VIQL algorithms are simplified to solve the LQR problem of linear systems. Finally, a linear F-16 aircraft plant and a numerical nonlinear system are used to test the developed QL algorithms, and the obtained simulation results demonstrate their effectiveness.

REFERENCES


