52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

Optimal Tracking Control for Linear Discrete-time Systems Using Reinforcement Learning

Bahare Kiumarsi-Khomartash, Frank L. Lewis, *Fellow, IEEE*, Mohammad-Bagher Naghibi-Sistani, and Ali Karimpour

Abstract— **This paper presents an online solution to the infinite-horizon linear quadratic tracker (LQT) using reinforcement learning. It is first assumed that the value function for the LQT is quadratic in terms of the reference trajectory and the state of the system. Then, using the quadratic form of the value function, an augmented algebraic Riccati equation (ARE) is derived to solve the LQT. Using this formulation, both feedback and feedforward parts of the optimal control solution are obtained simultaneously by solving the augmented ARE. To find the solution to the augmented ARE online, policy iteration as a class of reinforcement learning algorithms, is employed. This algorithm is implemented on an actor-critic structure by using two neural networks and it does not need the knowledge of the drift system dynamics or the command generator dynamics. A simulation example shows that the proposed algorithm works for a system with partially unknown dynamics.**

Key words: linear quadratic tracker, reinforcement learning, policy iteration, algebraic Riccati equation.

I. INTRODUCTION

The linear quadratic tracker (LQT) problem is concerned with the design of an optimal control law such that the output of a linear system follows a reference trajectory. The optimality is achieved by minimizing a quadratic performance index. In contrast to solutions to the linear quadratic regulator (LQR), which require solving an algebraic Riccati equation (ARE), traditional solutions to the LQT consist of solving an ARE and a noncausal difference equation simultaneously [1]. The ARE is solved to find the feedback part of the optimal controller and a noncausal difference equation is solved to find the feedforward part of the optimal control. This is an offline design procedure that requires full knowledge of the system dynamics.

Reinforcement learning (RL) [2]-[5], as a class of machine learning methods, has been effectively used to seek solutions to the optimal regulator problem [6]-[13]. In particular, a class of RL algorithms called policy iteration (PI) is used to solve the LQR [14]. PI techniques start with an admissible control policy and then successively alternate between policy evaluation and policy improvement steps until converge to the solution of the ARE. On the other hand, because of additional complexity caused by computing the feedforward term in the LQT, to our knowledge, RL techniques have not been used to solve the LQT.

This paper develops RL algorithms to solve the infinite-horizon LQT for discrete-time systems. First, it is assumed that the LQT value function is quadratic in the state of the system and the reference trajectory. Then, based on this quadratic form, we derive a Bellman equation and an augmented ARE whose solution gives solution to the LQT. Finally, the LQT Bellman equation is used to develop a policy iteration algorithm to solve online the LQT. This algorithm is implemented as an actor-critic structure by using two neural networks. A simulation shows that the LQT problem can be solved online without requiring knowledge of the full system dynamics or reference trajectory dynamics. Convergence to the optimal control solution of the LQT is verified.

The rest of the paper is laid out as follows. Review of the standard solution for the LQT is given in Section II. An alternative approach for formulating the infinite-horizon LQT in a causal manner is presented in Section III. In Section IV we develop offline and online PI algorithms to solve the LQT. In Section V, neural network is used to approximate value function and control input. The simulation results of applying the RL algorithms to find the optimal controller are presented in Section VI.

II. LINEAR QUADRATIC TRACKER AND REVIEW OF ITS STANDARD SOLUTION

In this section, the linear quadratic tracker (LQT) problem for discrete-time (DT) systems and its standard solution are presented [1].

Consider the linear time-invariant discrete-time (DT) system in the form of

$$\begin{aligned} x_{k+1} &= A x_k + B u_k \\ y_k &= C x_k \end{aligned} \tag{1}$$

where $x_k \in R^n$ is the measured state, $u_k \in R^m$ is the control input, $y_k \in R^p$ is the output and A, B and C are constant matrices with compatible dimensions.

This work is supported by Supported by NSF grant ECCS-1128050, NSF grant IIS-1208623, ONR grant N00014-13-1-0562, AFOSR EOARD Grant #13-3055, China NNSF grant 61120106011, and China Education Ministry Project 111 (No.B08015).

Bahare Kiumarsi-khomartash is with the Department of Electrical Engineering, Ferdowsi University of Mashhad, Iran (phone: 915-382-6015; fax: 581-2233789; e-mail: ba.kiumarsi@stu-mail.um.ac.ir).

Frank L. Lewis is with University of Texas at Arlington Research Institute, 7300 Jack Newell Blvd. S., Ft. Worth, TX 76118, USA (e-mail: lewis@uta.edu).

M.b Naghibi-Sistani and Ali karimpour are with the Department of Electrical Engineering, Ferdowsi University of Mashhad, Iran (e-mails: mb-naghibi@um.ac.ir; karimpor@um.ac.ir)

The objective for the LQT problem is to find the optimal control sequence, u_k^*, so as to make the linear system (1) to track a reference trajectory r_k in an optimal manner.

This can be obtained by minimizing the finite-horizon performance index

$$\begin{aligned} J_k &= \frac{1}{2}\left(y_N - r_N\right)^T E\left(y_N - r_N\right) \\ &+ \frac{1}{2}\sum_{i=k}^{N-1}\left[\left(y_i - r_i\right)^T Q\left(y_i - r_i\right) + u_i^T R u_i\right] \end{aligned} \tag{2}$$

where $E \geq 0$, $Q \geq 0$ and $R > 0$ are symmetric matrices.

The standard solution of the finite-horizon LQT problem is given as

$$u_k = -K_k x_k + K_k^v v_{k+1} \tag{3}$$

where v_{k+1} is acquired with solving difference equation

$$v_k = \left(A - B K_k\right)^T v_{k+1} + C^T Q r_k, \qquad v_N = C^T E r_N \tag{4}$$

K_k and K_k^v are

$$K_k = \left(B^T S_{k+1} B + R\right)^{-1} B^T S_{k+1} A \tag{5}$$

$$K_k^v = \left(B^T S_{k+1} B + R\right)^{-1} B^T \tag{6}$$

where

$$\begin{aligned} S_k &= A^T S_{k+1}\left(A - B K_k\right) + C^T Q C \\ &= A^T S_{k+1} A - A^T S_{k+1} B\left(B^T S_{k+1} B + R\right)^{-1} B^T S_{k+1} A + C^T Q C, \\ S_N &= C^T E C \end{aligned} \tag{7}$$

Note that the first term of the control input (3) is a feedback control part that depends linearly on the state of the system and the second term is a feedforward control part that depends on the reference trajectory. The standard solution of the LQT is noncausal because it is needed to solve for v_k backwards in time using (4).

In [1], it was shown that the optimal value of the performance index using control (3) on the interval $[k, N]$ is given by

$$J_k = \frac{1}{2} x_k^T S_k x_k - x_k^T v_k + w_k \tag{8}$$

where the signal w_k satisfies the backward recursion

$$\begin{aligned} w_k &= w_{k+1} + \frac{1}{2} r_k^T Q r_k - \frac{1}{2} v_{k+1}^T B\left(B^T S_{k+1} B + R\right)^{-1} B^T v_{k+1} \\ w_N &= \frac{1}{2} r_N^T E r_N \end{aligned} \tag{9}$$

The infinite-horizon LQT problem is the concern of this paper. In the infinite-horizon case *N* tends to infinitely. Then the performance index (2) for fixed control policies becomes the value function

$$V\left(x_k, r_k\right) = \frac{1}{2}\sum_{i=k}^{\infty} U_i \tag{10}$$

where

$$U_i = \left(C x_i - r_i\right)^T Q\left(C x_i - r_i\right) + u_i^T R u_i$$

In the infinite-horizon case, the Riccati recursion (7) becomes the algebraic Riccati equation (ARE)

$$\begin{aligned} S &= A^T S\left(A - B K\right) + C^T Q C = \\ &= A^T S A - A^T S B\left(B^T S B + R\right)^{-1} B^T S A + C^T Q C \end{aligned} \tag{11}$$

Sufficient conditions for existence of a solution $S = S^T > 0$ to the ARE are as follows.

Assumption 1. The pair (A, B) is stabilizable and the pair is $\left(A, \sqrt{Q}C\right)$ observable [1].

The control gains (5) and (6) become constants

$$K = \left(B^T S B + R\right)^{-1} B^T S A \tag{12}$$

$$K^v = \left(B^T S B + R\right)^{-1} B^T \tag{13}$$

Then the control input is

$$u_k = -K x_k + K^v v_{k+1} \tag{14}$$

With

$$v_k = \left(A - B K\right)^T v_{k+1} + C^T Q r_k \tag{15}$$

Finally, the optimal value of the performance index is given by the value function

$$J = V\left(x_k, r_k\right) = \frac{1}{2} x_k^T S x_k - x_k^T v_k + w_k \tag{16}$$

with

$$w_k = w_{k+1} + \frac{1}{2} r_k^T Q r_k - \frac{1}{2} v_{k+1}^T B\left(B^T S B + R\right)^{-1} B^T v_{k+1} \tag{17}$$

III. CAUSAL SOLUTION TO THE LQT PROBLEM

This section presents the new results of this paper. First, it is assumed that the value function of the LQT problem can be expressed as a quadratic form in terms of x_k and r_k. Then, a Bellman equation and an augmented LQT ARE are given based on this value function. This structure is used in Section IV to solve online the LQT using RL.

The following assumptions are required.

Assumption 2. The reference trajectory for the LQT problem is generated by

$$r_{k+1} = F r_k \tag{18}$$

It is assumed that the command generator dynamic *F* in (18) is Hurwitz.

The augmented system is performed Based on the system dynamics (1) and the reference trajectory dynamics (18) as following

$$X_{k+1} = \begin{bmatrix} x_{k+1} \\ r_{k+1} \end{bmatrix} = \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & F \end{bmatrix} \begin{bmatrix} x_k \\ r_k \end{bmatrix} + \begin{bmatrix} B \\ \mathbf{0} \end{bmatrix} u_k \equiv T X_k + B_1 u_k \tag{19}$$

where the augmented state is

$$X_k = \begin{bmatrix} x_k \\ r_k \end{bmatrix} \tag{20}$$

Assumption 3. For the infinite-horizon LQT problem, under Assumption 2, the value function (10) can be written as

$$V(x_k, r_k) = V(X_k) = \frac{1}{2} X_k^T P X_k \tag{21}$$

for any fixed stabilizing policy

$$u_i = K_x x_i + K_r r_i \tag{22}$$

for some symmetric $P > 0$.

On the basis of (10) we have

$$V(x_k, r_k) = \frac{1}{2}(C x_k - r_k)^T Q(C x_k - r_k) + \frac{1}{2} u_k^T R u_k + \frac{1}{2} \sum_{i=k+1}^{\infty} \left[(C x_i - r_i)^T Q(C x_i - r_i) + u_i^T R u_i \right] \tag{23}$$

which yields the LQT Bellman equation

$$V(x_k, r_k) = \frac{1}{2}(C x_k - r_k)^T Q(C x_k - r_k) + \frac{1}{2} u_k^T R u_k + V(x_{k+1}, r_{k+1}) \tag{24}$$

Using (21) in (24), we obtain the LQT Bellman equation in terms of value function kernel matrix P

$$X_k^T P X_k = X_k^T Q_1 X_k + u_k^T R u_k + X_{k+1}^T P X_{k+1} \tag{25}$$

where

$$Q_1 = \begin{bmatrix} C^T Q C & -C^T Q \\ -Q C & Q \end{bmatrix} \tag{26}$$

Define the LQT Hamiltonian

$$H(X_k, u_k) = X_k^T Q_1 X_k + u_k^T R u_k + X_{k+1}^T P X_{k+1} - X_k^T P X_k \tag{27}$$

or equivalently

$$H(X_k, u_k) = X_k^T Q_1 X_k + u_k^T R u_k + V(X_{k+1}) - V(X_k) \tag{28}$$

The next theorem shows how the LQT problem can be solved in a causal manner using an augmented ARE.

Theorem 1. ARE for causal solution of the LQT

Under Assumptions 2 and 3, the optimal policy for the LQT problem is

$$u_k = -K_1 X_k \tag{29}$$

where

$$K_1 = \left(R + B_1^T P B_1 \right)^{-1} B_1^T P T \tag{30}$$

and P satisfies the augmented LQT ARE

$$Q_1 - P + T^T P T - T^T P B_1 \left(R + B_1^T P B_1 \right)^{-1} B_1^T P T = 0 \tag{31}$$

Proof. A necessary condition for optimality [1] is

$$\frac{\partial H(X_k, u_k)}{\partial u_k} = 2 R u_k + \frac{\partial X_{k+1}}{\partial u_k}^T \frac{\partial V(X_{k+1})}{\partial X_{k+1}} = 2 R u_k + 2 B_1^T P X_{k+1} = 0$$

Then

$$u_k = -\left(R + B_1^T P B_1 \right)^{-1} B_1^T P T X_k \tag{32}$$

Substituting (19) and (32) in Bellman equation (25), results in LQT ARE (31). ∎

The next result shows that the standard LQT in Section II is equivalent to the causal LQT formulation in Theorem 1.

Corollary 2. Relation between standard LQT and causal formulation for LQT

In the infinite-horizon LQT, the standard LQT solution (11)-(15) and its value function (16) , (17) are equivalent to the LQT solution in Theorem 1 and its value function (21) with

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \in R^{(n+p)\times(n+p)} \tag{33}$$

if

$$S = P_{11} \tag{34}$$

$$v_{k+1} = -P_{12} F r_k = -P_{12} r_{k+1} \tag{35}$$

$$w_k = \frac{1}{2} r_k^T P_{22} r_k \tag{36}$$

Proof. Putting (19) , (20) and (33) in (32), gives

$$u_k = -\left(R + B^T P_{11} B \right)^{-1} \left[B^T P_{11} A x_k + B^T P_{12} F r_k \right] \tag{37}$$

By comparing (14) and (37), the standard control input and casual control input are equivalent if (34) and (35) are satisfied. Also, using (34) and (35) in (16) and equating the value functions (16) and (21), gives (36). In the standard LQT, S , v_k and w_k are obtained by the ARE (11) and the difference equations (15) and (17), respectively. In the following, it is shown that if (34)-(36) are satisfied, the same equation as (11), (15) and (17) are obtained by causal solution of the LQT. by expanding the LQT ARE, the upper

left-hand and upper-right hand sides of the LQT ARE become

$$C^T Q C - P_{11} + A^T P_{11} A - A^T P_{11} B\left(R + B^T P_{11} B\right)^{-1} B^T P_{11} A = 0 \quad (38)$$

$$-C^T Q - P_{12} + A^T P_{12} F - A^T P_{11} B\left(R + B^T P_{11} B\right)^{-1} B^T P_{12} F = 0 \quad (39)$$

Equation (39) is equal to the ARE (11) provide that (34) is satisfied and (39) is equivalent to (15) provided that (35) is satisfied.

The time signal w_t in the standard value function satisfies (17). By expanding the LQT ARE, the lower right-hand side of it becomes

$$Q - P_{22} + F^T P_{22} F - F^T P_{21} B\left(R + B^T P_{11} B\right)^{-1} B^T P_{12} F = 0 \quad (40)$$

where is equal to (17) if (36) is satisfied.

IV. SOLVING THE LQT PROBLEM ONLINE USING REINFORCEMENT LEARNING

In this section, the LQT problem is solved using RL based on new formulation of the LQT problem in Section III.

The augmented LQT Bellman equation (25) for an arbitrary stabilizing gain K_1 becomes

$$Q_1 - P + K_1^T R K_1 + \left(T - B_1 K_1\right)^T P\left(T - B_1 K_1\right) = 0 \quad (41)$$

which is a Lyapunov equation.

Instead of directly solving the LQT ARE (31) to obtain optimal control input, Lyapunov equation (41) can be solved iteratively. Two iterative PI algorithms are provided as follows.

Algorithm 1. Offline Lyapunov iteration

Select a stabilizing initial control policy K_1.

1. Policy evaluation

$$P^{j+1} = Q_1 + \left(K_1^j\right)^T R K_1^j + \left(T - B_1 K_1^j\right)^T P^{j+1}\left(T - B_1 K_1^j\right) \quad (42)$$

2. Policy improvement

$$K_1^{j+1} = \left(R + B_1^T P^{j+1} B_1\right)^{-1} B_1^T P^{j+1} T \quad (43)$$

This algorithm is implemented offline and the system dynamics $\left(T, B_1\right)$ are required to solve (42). It is an extension of Hewer's method [15] to the LQT. The policy iteration algorithm must be suitably initialized to converge.

The next algorithm uses the LQT Bellman equation (25) to solve the LQT online.

Algorithm 2. Online policy iteration

Select a stabilizing initial control policy K_1.

1. Policy evaluation

$$X_k^T P^{j+1} X_k = X_k^T \left(Q_1 + \left(K_1^j\right)^T R K_1^j\right) X_k + X_{k+1}^T P^{j+1} X_{k+1} \quad (44)$$

2. Policy improvement

$$K_1^{j+1} = \left(R + B_1^T P^{j+1} B_1\right)^{-1} B_1^T P^{j+1} T \quad (45)$$

Policy iteration Algorithm 2 can be implemented online using Least-squares (LS) [16] using the data set X_k, X_{k+1} and ρ_k measured along the system trajectory with $\rho_k = X_k^T \left(Q_1 + \left(K_1^j\right)^T R K_1^j\right) X_k$.

Note that in Algorithms 1 and 2, it is needed to knowing full knowledge of the system dynamics or command generator dynamics. In the next section, it is shown how to solve the LQT problem without knowing the system internal dynamics matrix *T*.

V. SOLVING THE LQT PROBLEM USING NEURAL NETWORK APPROXIMATION

In this section, an actor-critic structure is given to implement Algorithm 2 online. It is well known that the neural networks (NN) can be used to approximate smooth function on a compact set [17]. Therefore, to solve the LQT problem $V(X)$ is approximated at each iterations by a critic NN

$$\hat{V}_j(X) = \sum_{i=1}^{L} w_{vj}^i \phi_i(X) = W_{Vj}^T \phi(X) \quad (46)$$

where $W_{Vj} \in R^L$ is the critic NN weights and *L* is the number of hidden-layer neurons. The vector $\phi(X) = \left[\phi_1(X) \;\; \phi_2(X) \;\; \cdots \;\; \phi_L(X)\right]^T$ is the vector activation function.

Using value function approximation (46), we can develop a PI algorithm that does not need the knowledge of the *T* in system dynamics (19). Using value function approximation to solve the Bellman equation (44) in policy evaluation step of Algorithm 2, the Bellman equation becomes

$$W_{Vj+1}^T \left(\phi(X_k) - \phi(X_{k+1})\right) = X_k^T Q_1 X_k + \left(u_k^j\right)^T R\left(u_k^j\right) \quad (47)$$

This is solved for the weights W_{Vj+1} using least mean square (LMS). For this step the dynamics $\left(T, B_1\right)$ can be unknown as they are not needed.

Note that the policy improvement step based on (43) or (45) requires full knowledge of the system dynamics $\left(T, B_1\right)$. This problem is solved by introducing a second neural network for the control policy, known as the actor NN [18]. Therefore, introduce an actor parametric approximator structure

$$\hat{u}^j(X) = \sum_{i=1}^{M} w_{uj}^i \sigma_i(X) = W_{Uj}^T \sigma(X) \quad (48)$$

where $W_{Uj} \in R^M$ is the actor NN weights and *M* is the number of hidden-layer neurons. The vector $\sigma(X) = \left[\sigma_1(X) \;\; \sigma_2(X) \;\; \cdots \;\; \sigma_M(X)\right]^T$ is the vector activation function.

After convergence of the critic NN parameters to W_{Vj+1} in step 1 Algorithm 2, it is required to perform the policy improvement in step 2 Algorithm 2

$$\hat{u}^{j+1}(X_k) = \arg\min_{u(0)}\left(X_k^T Q_1 X_k + \left(\hat{u}_k^j\right)^T R\left(\hat{u}_k^j\right) + W_{Vj+1}^T \phi(X_{k+1})\right) \quad (49)$$

To achieve this one may use an LMS algorithm. The weight update is therefore

$$W_{Uj+1}\big|_{m+1} = W_{Uj+1}\big|_{m} - \alpha\sigma(X_k)\left(2R\hat{u}^{j+1} + B_1^T \frac{\partial\phi(X_{k+1})}{\partial X_{k+1}} W_{Vj+1}\right) \quad (50)$$

where α is a positive step size and m is the iteration number for the LMS algorithm.

Remark 3. Note that in using two NNs to implement Algorithm 2, it is not necessary to know the system matrix T in (19). Only information about B_1 is used to implemented Algorithm 2 step 2 using (50).

VI. SIMULATION

To illustrate the effectiveness of the proposed approach, a simulation example is provided.

Consider the following linear discrete-time system

$$x_{k+1} = \begin{bmatrix} -1 & 2 \\ 2.2 & 1.7 \end{bmatrix} x_k + \begin{bmatrix} 2 \\ 1.5 \end{bmatrix} u_k \quad (51)$$

$$y_k = \begin{bmatrix} 1 & 2 \end{bmatrix} x_k$$

The open-loop poles are $z_1 = -2.1445$ and $z_1 = 2.8445$, so the system is unstable. The performance index is defined by (10) with $Q = 6$ and $R = 1$.

The reference trajectory is given by

$$r_{k+1} = 0.7\, r_k \quad (52)$$

The optimal value kernel matrix P is found by solving (31) to be

$$P^* = \begin{bmatrix} 157.3743 & 22.0537 & -20.3400 \\ 22.0537 & 25.7351 & -13.0253 \\ -20.3400 & -13.0253 & 7.5197 \end{bmatrix} \quad (53)$$

First, we use offline PI Algorithm 1 implemented as in (42) and (43). Fig. 1 shows that the P matrix parameters converge to their optimal values.

$$P = \begin{bmatrix} 157.3743 & 22.0537 & -20.3400 \\ 22.0537 & 25.7351 & -13.0253 \\ -20.3400 & -13.0253 & 7.5197 \end{bmatrix}$$

Fig. 2 shows that by applying control (43), the system becomes stable and the output y_k tracks the reference trajectory r_k. The optimal control signal input is also shown in Fig. 3.

Next, we use online PI Algorithm 2 implemented as in (44) and (45). PE was ensured by adding a small probing noise to the control input and reference trajectory. Fig. 4 shows that the P matrix parameters converge to their optimal values.

$$P = \begin{bmatrix} 157.3743 & 22.0537 & -20.3400 \\ 22.0537 & 25.7351 & -13.0253 \\ -20.3400 & -13.0253 & 7.5197 \end{bmatrix}$$

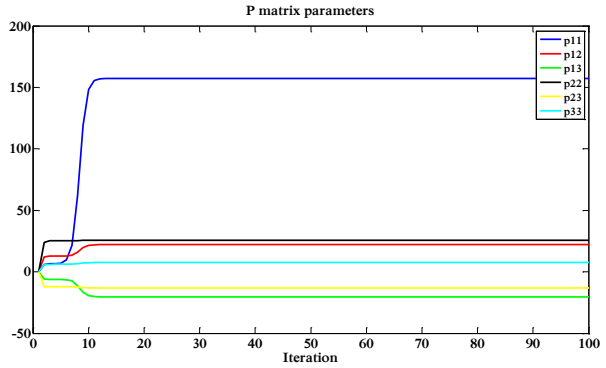

Figure 1. Convergence of the P matrix parameters to their optimal values for offline PI Algorithm 1

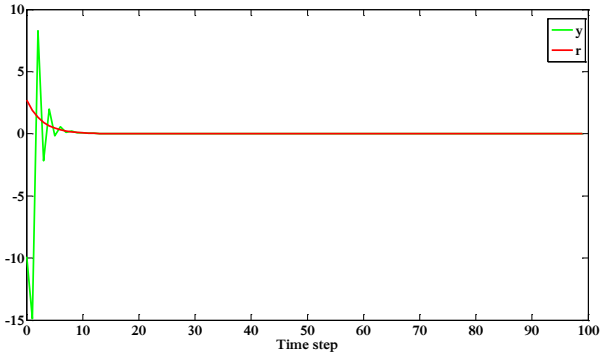

Figure 2. Evaluation of the output and reference trajectory for offline PI Algorithm 1

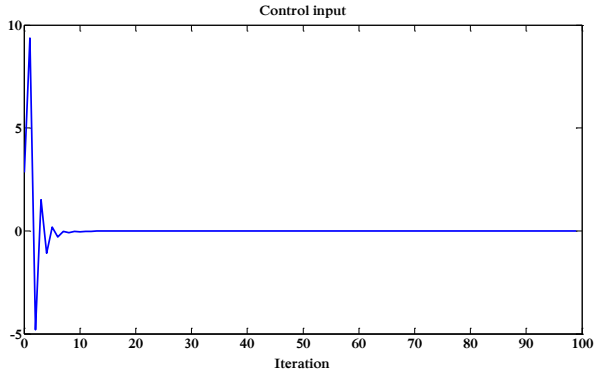

Figure 3. The control input signal

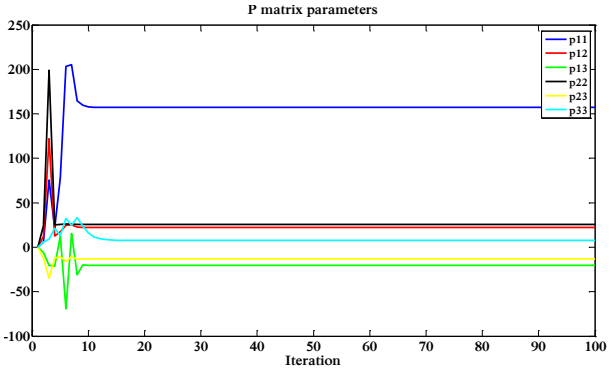

Figure 4. Convergence of the P matrix parameters to their optimal values for online PI Algorithm 2

Comparing this P matrix with the P^* matrix, it is seen that the online PI Algorithm 2 converges to the optimal controller.

Finally, it is shown that using two NN as in Section V allows one compute the optimal value and control online without knowing the system matrix T.

The optimal P matrix for this problem by solving the ARE (31) is (53). Optimal control $u^* = -K_1 X_k$, where K_1 is

$$K_1 = \begin{bmatrix} -0.2021 & 1.0189 & -0.0514 \end{bmatrix}$$

It is known that for the LQT the value is quadratic in terms of the augmented state of the system and control is linear. Therefore, we select linear activation functions for the actor NN and quadratic polynomial activations for the critic NN. The approximation of control is given by (48), where W_u the weight vector of actor NN is

$$W_u^T = \begin{bmatrix} w_u^1 & w_u^2 & w_u^3 \end{bmatrix} \tag{54}$$

and the $\sigma(X_k)$ is the vector of activation functions for the actor NN and is given by

$$\sigma(X_k) = \begin{bmatrix} x_1 & x_2 & r \end{bmatrix}^T \tag{55}$$

The value function is approximated as (46), where W_V is the weight vector of the critic NN given by

$$W_V^T = \begin{bmatrix} w_v^1 & w_v^2 & w_v^3 & w_v^4 & w_v^5 & w_v^6 \end{bmatrix} \tag{56}$$

and the $\phi(X_k)$ is the vector of activation functions for the critic NN given by

$$\phi(X_k) = \begin{bmatrix} x_1^2 & x_1 x_2 & x_1 r & x_2^2 & x_2 r & r^2 \end{bmatrix}^T \tag{57}$$

That P matrix parameters (53) are related to weights of value function as follows

$$\begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} = \begin{bmatrix} w_v^1 & 0.5w_v^2 & 0.5w_v^3 \\ 0.5w_v^2 & w_v^4 & 0.5w_v^5 \\ 0.5w_v^3 & 0.5w_v^5 & w_v^6 \end{bmatrix}$$

The value function weights converge to

$$W_V^T = \begin{bmatrix} 155.6308 & 43.8919 & -40.3151 \\ 25.7284 & -26.0281 & 7.5001 \end{bmatrix}$$

The control input converge to

$$W_u = \begin{bmatrix} -0.2001 & 1.0191 & -0.0515 \end{bmatrix}$$

Comparing value function weights with the P^* matrix, it is seen that the proposed algorithm converges to the optimal controller. Also the control weights converge to the optimal control as expected.

VII. Conclusion

In this paper, an alternative formulation for the LQT problem was proposed. On the basis of this formulation, the LQT problem was solved by using a LQT ARE. The ARE equation is solved by using a reinforcement learning algorithm implemented on an actor-critic structure online and without requiring the drift system dynamics or the command generator dynamics. The simulation example confirmed the validity of the tracking scheme.

Acknowledgment

This work is supported by Supported by NSF grant ECCS-1128050, NSF grant IIS-1208623, ONR grant N00014-13-1-0562, AFOSR EOARD Grant #13-3055, China NNSF grant 61120106011, and China Education Ministry Project 111 (No.B08015).

References

[1] F. L. Lewis, D. Vrabie, and V. Syrmos, Optimal Control. New York: John Wiley, 2012.

[2] W. B. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality. John Wiley, 2009.

[3] R. S. Sutton, and A. G. Barto, Reinforcement learning – an introduction. Cambridge, MT: MIT Press, 1998.

[4] J. Si. A. Barto, W. Powell, and D. Wunch, Handbook of Learning and Approximate Dynamic Programming. Wiley, 2004.

[5] D. P. Bertsekas, and J. N. Tsitsiklis, Neuro-dynamic programming. MA: Athena Scientific, 1996.

[6] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," Automatica, vol. 43, no. 3, pp. 473–481, Mar. 2007.

[7] P. He, and S. Jagannathan, "Reinforcement learning neural-network-based controller for nonlinear discrete-time systems with input constraints," IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 37, no. 2, pp. 425-436, Apr. 2007.

[8] Z. Chen, and S. Jagannathan, "Generalized Hamilton-Jacobi-Bellman formulation-based neural network control of affine nonlinear discrete time systems," IEEE Trans. Neural Networks, vol. 19, no. 1, pp. 90–106, Jan. 2008.

[9] P. J. Werbos, "Neural networks for control and system identification," in Proc. 28th IEEE Conf. Decision and Control, pp. 260-265, Dec, 1989.

[10] P. J. Werbos, *A menu of designs for reinforcement learning over time.* In Neural Networks for Control. W. T. Miller, R.S. Sutton, & P. J. Werbos. (Eds.). Cambridge, MA: MIT Press, pp. 67-95, 1991.

[11] P. J. Werbos, *Approximate dynamic programming for real-time control and neural modeling.* In D. A. White, & D. A. Sofge (Eds.), Handbook of intelligent control. New York: Van Nostrand Reinhold, 1992.

[12] F. L. Lewis and D. Vrabie, "Reinforcement learning and daptive dynamic programming for feedback control," *IEEE Circuit Syst. Mag*., vol. 9, no. 3, pp. 32–50, Sep, 2009.

[13] H. G. Zhang, Y. H. Luo, and D. Liu, "Neural network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraint," *IEEE Trans*. Neural Networks, vol. 20, no. 9, pp. 1490–1503, Sep, 2009.

[14] F. L. Lewis, and K. Vamvoudakis, "Reinforcement learning or partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Trans*. Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 41, no. 1, pp. 14-23, Feb, 2011.

[15] G. A. Hewer, "An iterative technique for the computation of steady state gains for the discrete optimal egulator," *IEEE Trans*. Automatic Control, vol. 16, no. 4, pp. 382–384, Aug, 1971.

[16] F. L. Lewis, D. Vrabie, and K. Vamvoudakis, "Reinforcement learning and feedback control using natural decision methods to design optimal adaptive controllers," *IEEE Syst.* Mag., vol. 32, no. 6, pp. 76-105, Nov, 2012.

[17] K. M. Hornik and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," Neural Network, vol. 3, pp. 551-560, 1990.

[18] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-Time Nonlinear HJB Solution Using Approximate Dynamic Programming: Convergence Proof," *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics,* vol. 38, no. 4, pp. 943-949, Aug, 2008.