IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 8, 2020, accepted July 18, 2020, date of publication July 22, 2020, date of current version July 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3011194

Reinforcement Q-Learning Incorporated With Internal Model Method for Output Feedback Tracking Control of Unknown Linear Systems

CONG CHEN, WEIJIE SUN, (Member, IEEE), GUANGYUE ZHAO, AND YUNJIAN PENG, (Member, IEEE)
School of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China

Corresponding author: Yunjian Peng (pengyj@scut.edu.cn)

This work was supported in part by the National Natural Science Foundation (NNSF) of China under Grant 61573154, and in part by the Science and Technology Planning Project of Guangdong Province under Grant 2015A010106003 and Grant 2017A010101009.

ABSTRACT This paper investigates the output feedback (OPFB) tracking control problem for discrete-time linear (DTL) systems with unknown dynamics. With the approach of augmented system, the tracking control problem is first turned into a regulation problem with a discounted performance function, the solution of which relies on the Q-function based Bellman equation. Then, a novel value iteration (VI) scheme based on reinforcement Q-learning mechanism is proposed for solving the Q-function Bellman equation without knowing the system dynamics. Moreover, the convergence of the VI based Q-learning is proved by indicating that it converges to the Q-function Bellman equation and it brings out no bias of solution even under the probing noise satisfying the persistent excitation (PE) condition. As a result, the OPFB tracking controller can be learned online by using the past input, output, and reference trajectory data of the augmented system. The proposed scheme removes the requirement of initial admissible policy in the policy iteration (PI) method. Finally, effectiveness of the proposed scheme is demonstrated through a simulation example.

INDEX TERMS Adaptive dynamic programming (ADP), optimal control, Bellman equation, on-policy, internal model.

I. INTRODUCTION

For controller design problem, optimization of performance costs has been an important concern since it may lead to reduction in energy effort which leads to positive consequences on earth environment. The practical need has greatly promoted the development of optimal control [1]. The key to optimal control problem is the solution of a equation, which is called Ricatti equation for linear systems. For the linear case, the solution of Ricatti equation can be efficiently obtained by the iteratively computational algorithms [2], [3], which are only applicable to the cases where complete knowledge of system dynamics is known. However, it is often desirable in control engineering to design online learning controllers without resorting to the system dynamics [4]–[8]. Notice that a data-based method has been proposed in [9] to analysis the controllability and observability of DTL systems without knowing system parameters.

Reinforcement learning (RL) was developed to focus on obtaining the optimal reward from the interaction with environment [10]. On the other hand, adaptive control has been studied to design controllers for systems with uncertain parameter models [11], [12]. Thanks to RL techniques, adaptive control with optimal design criterions can be found by sequentially updating the controller parameters based on the reward signal reflecting the controller's performance [13]. Generally speaking, reinforcement learning provides adaptive optimal control design philosophy, which brings new insight into the Control System Community [13], [14]. It has given development to an alternative optimal control strategy known as adaptive dynamic programming (ADP), which is a (partially) model-free method achieving the optimal performance index [14]–[18]. Solutions to optimal control based on the idea of ADP have been extensively investigated for

The associate editor coordinating the review of this manuscript and approving it for publication was Nasim Ullah.

both linear quadratic regulator (LQR) and linear quadratic tracking (LQT) problems, see [19]–[22] and the references therein. The learning scheme under RL framework generally has two iterative steps to find optimal control policy, one for policy evaluated and the other for policy updated with the latter being an improvement of the first. Notice that RL based on value function approximation (VFA) would ruin the algorithm convergence, which results from the exploration noise added intentionally to the evaluated policy for sufficiently exciting the system [22]–[24]. Besides, the policy iteration (PI) scheme in ADP framework needs an initial admissible policy, which requires a prior knowledge of unknown system to design a robust controller [21], [25]. To overcome such a requirement, value iteration (VI) scheme has been studied in recent work [22], [26] using VFA method.

Most of the existing studies rely on the available measurement of full state information, see [22], [27] and the references therein. However, it may not be feasible in practical implementations [28], and therefore, it is desirable to design output feedback (OPFB) learning controllers. Dynamic OPFB controllers have been studied in [29] for the Q-learning based LQR control of DTL systems. The state parametrization method was presented to reconstruct system states based on the filtered input and output signals. On the other hand, the static OPFB design owes its popularity due to its simplicity in structure and is proposed to solve the LQR problem in [30]. Extension to H_∞ control problem can be seen in the recent work [31], [32], where an initial stabilizing control policy is required. To find the static OPFB controller, the full state knowledge is needed during the learning stage and the neural network based model-free state estimation technique should be employed [31], [32]. OPFB control can also be achieved by employing model-free state reconstruction method. The advantage comes from the measured data of past input, output, and reference trajectory as an alternative to the unavailable system states. It was first presented in [23] to learn OPFB LQR controller. The same method has also been used to solve OPFB LQT problem [24] by employing VFA technique. In recent studies, the model-free state reconstruction technique was also used to develop OPFB Q-learning PI scheme for H_∞ control problem [33], [34].

In this paper, a novel VI based Q-learning method is proposed to design OPFB tracking controller for DTL systems with unknown dynamics. The main contributions are summarized as follows.

1) As opposed to a PI design, VI algorithm removes the requirement of initial admissible policy such that it allows a more general condition. Under the Q-learning scheme, the convergence analysis of VI algorithm is given.

2) The OPFB Q-learning tracker design is achieved without knowing the full state vector by collecting past input, output, and reference trajectory data.

3) In [25], a H_∞ robust stabilization controller was applied to provide initial data for LQR problem, but we extend this design philosophy to LQT problem. An internal model controller is proposed to collect the unavailable few data at first. Then the OPFB Q-learning controller is used to achieve the tracking issue.

II. PROBLEM STATEMENT

In this section, the infinite-horizon LQT problem is first reviewed for DTL systems. Then, some basic results are presented for solving a discrete-time Bellman equation. Consider a DTL time-invariant system described by

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, and $y_k \in \mathbb{R}^p$ are the state, input, and output, respectively. The denotations A, B, C are constant matrices with appropriate dimensions, where the pairs (A, B) and (A, C) are respectively controllable and observable.

The reference trajectory is produced by the exogenous system

$$r_{k+1} = Fr_k \tag{2}$$

where $r_k \in \mathbb{R}^p$ and F is a constant matrix of proper dimension.

Define the tracking error as

$$e_k = y_k - r_k. \tag{3}$$

The objective is to design an optimal control policy u_k such that the output y_k could track the reference trajectory r_k in an optimal sense by minimizing the following discounted performance index

$$J(x_k, r_k) = \frac{1}{2}\sum_{i=k}^{\infty} \gamma^{i-k}(e_i^T Q e_i + u_i^T R u_i) \tag{4}$$

where Q and R are positive definite weighting matrices, and $0 < \gamma \leq 1$ is the discount factor.

Remark 1: As argued in [35], the discount factor γ in (4) allows a more general solution than the standard LQT problem. The matrix F is not assumed to be stable. Thus, it satisfies a much larger class of reference signals for tracking control problem with an quadratic performance index. On the other hand, both feedback and feedforward parts of the control input can be optimized simultaneously, which provides a causal manner for the solution of infinite-horizon LQT problem. Notice that the discount factor γ loses no generality in the sense that it can be chosen as $\gamma = 1$ when F is Hurwitz, and for this case, the LQT problem reduces into an LQR problem with some specified output trajectory decaying to zero exponentially.

A. OFFLINE SOLUTION FOR LQT

Denoting $X_k = \begin{bmatrix} x_k^T & r_k^T \end{bmatrix}^T$ yields an augmented system

$$\begin{aligned} X_{k+1} &= TX_k + B_1 u_k \\ e_k &= C_1 X_k \end{aligned} \tag{5}$$

where $T = \begin{bmatrix} A & 0 \\ 0 & F \end{bmatrix}$, $B_1 = \begin{bmatrix} B \\ 0 \end{bmatrix}$ and $C_1 = \begin{bmatrix} C & -I \end{bmatrix}$.

If $u_k = -KX_k$ with $K = \begin{bmatrix} K_x & K_r \end{bmatrix}$, the discounted performance index (4) is in a quadratic form as follows

$$V(x_k, r_k) = V(X_k) = \frac{1}{2} X_k^T P X_k \tag{6}$$

for some matrix $P = P^T > 0$, as can be seen from Lemma 1 of [35].

Using (4), one has

$$J(x_k, r_k) = \frac{1}{2}(e_k^T Q e_k + u_k^T R u_k) + \frac{1}{2} \sum_{i=k+1}^{\infty} \gamma^{i-(k+1)} (e_i^T Q e_i + u_i^T R u_i) \tag{7}$$

According to (6), $J(x_k, r_k)$ can be expressed as $V(x_k, r_k)$. Thus, we have

$$V(x_k, r_k) = \frac{1}{2} e_k^T Q e_k + \frac{1}{2} u_k^T R u_k + \gamma V(x_{k+1}, r_{k+1}). \tag{8}$$

Putting (6) into (8), we have LQT Bellman equation with respect to P

$$X_k^T P X_k = X_k^T \Pi X_k + u_k^T R u_k + \gamma X_{k+1}^T P X_{k+1}, \tag{9}$$

where $\Pi = \begin{bmatrix} C^T Q C & -C^T Q \\ -QC & Q \end{bmatrix}$.

Define the LQT Hamiltonian

$$\frac{1}{2} H(X_k, u_k) = \frac{1}{2} X_k^T \Pi X_k + \frac{1}{2} u_k^T R u_k + \gamma V(X_{k+1}) - V(X_k).$$

By solving the stationary condition [35], [36], i.e.,

$$\frac{\partial H(X_k, u_k)}{\partial u_k} = 0$$

one has

$$u_k = -KX_k = -K_x x_k - K_r r_k \tag{10}$$

where $K = (R + \gamma B_1^T P B_1)^{-1} \gamma B_1^T P T$ with P satisfying the following augmented algebraic Riccati equation (ARE)

$$\Pi - P + \gamma T^T P T - \gamma^2 T^T P B_1 (R + \gamma B_1^T P B_1)^{-1} B_1^T P T = 0 \tag{11}$$

Remark 2: The augmented ARE (11) has a unique positive definite solution P if the pair $(A, \sqrt{Q}C)$ is observable and $\gamma^{1/2} F$ is stable [24]. Moreover, a lower bound has been given for the discount factor to guarantee the stability of augmented system [37].

Due to the nonlinear relationship in the unknown parameter, it is difficult to directly obtain the solution of (11). Instead of solving (11), substituting (10) into (9) leads to the augmented LQT Lyapunov equation

$$\Pi - P + K^T R K + \gamma (T - B_1 K)^T P (T - B_1 K) = 0. \tag{12}$$

In this respect, an offline PI algorithm [35] was proposed to iteratively compute the solution of (12). However, it requires the complete knowledge of augmented system dynamics. To obviate this requirement, a Q-learning scheme [35] has been developed to solve model-free LQT problem.

B. Q-FUNCTION BELLMAN EQUATION

Let $Z_k = \begin{bmatrix} X_k^T & u_k^T \end{bmatrix}^T$ and define a discrete-time Q-function as

$$Q(Z_k) = \frac{1}{2} X_k^T \Pi X_k + \frac{1}{2} u_k^T R u_k + \gamma V(X_{k+1}) \tag{13}$$

Using the augmented system dynamics (5) in (13) gives

$$Q(Z_k) = \frac{1}{2} Z_k^T \tilde{H} Z_k \tag{14}$$

where

$$\tilde{H} = \begin{bmatrix} \Pi + \gamma T^T P T & \gamma T^T P B_1 \\ \gamma B_1^T P T & R + \gamma B_1^T P B_1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{H}_{XX} & \tilde{H}_{Xu} \\ \tilde{H}_{uX} & \tilde{H}_{uu} \end{bmatrix}$$

for kernel matrix $\tilde{H} = \tilde{H}^T$.

Applying $\frac{\partial Q(Z_k)}{\partial u_k} = 0$, we can get

$$u_k = -(\tilde{H}_{uu})^{-1} \tilde{H}_{uX} X_k \tag{15}$$

Noticing that $Q(Z_k) = V(X_k)$ leads to the Q-function Bellman equation

$$Z_k^T \tilde{H} Z_k = X_k^T \Pi X_k + u_k^T R u_k + \gamma Z_{k+1}^T \tilde{H} Z_{k+1}. \tag{16}$$

C. PI BASED Q-LEARNING FOR LQT

Based on (16) in terms of Q-function, the PI based Q-learning solution for LQT problem can be implemented by Algorithm 1 without resorting to the system dynamics [35].

Algorithm 1 PI Q-learning Algorithm for LQT

Initialization

Start with an admissible control policy u_k^0 with $\tilde{H}^0$

Procedure

1: (Policy Evaluation) For $j = 0, 1, \ldots$, collect data samples under u_k^j

to solve $\tilde{H}^{j+1}$ using the Q-function Bellman equation:

$$Z_k^T \tilde{H}^{j+1} Z_k = X_k^T \Pi X_k + (u_k^j)^T R (u_k^j) + \gamma Z_{k+1}^T \tilde{H}^{j+1} Z_{k+1} \tag{17}$$

2: (Policy Improvement) Compute the improved control policy as follows:

$$u_k^{j+1} = -(\tilde{H}_{uu}^{j+1})^{-1} \tilde{H}_{uX}^{j+1} X_k \tag{18}$$

3: (Stopping Criterion) Stop the iteration if $\|\tilde{H}^{j+1} - \tilde{H}^j\| < \varepsilon$ for some specified small positive number ε. Otherwise, let $j = j + 1$ and go back to iteration.

End Procedure

In Algorithm 1, repeated iteration between (17) and (18) will be performed until convergence. In contrast to offline algorithm [35], the policy improvement step is conducted by the learned kernel matrix $\tilde{H}^{j+1}$. Therefore, the objective of finding optimal control policy is achieved with completely unknown dynamics.

III. VALUE ITERATION Q-LEARNING FOR OUTPUT FEEDBACK LQT

In this section, the VI learning mechanism is introduced into Q-learning to solve Q-function Bellman equation arising in the model-free LQT problem for DTL systems.

A. COMPARISON FOR PI AND VI SCHEME

In ADP framework [10], PI and VI are two basic ways to solve Bellman equation. Generally, the solution is obtained by the repeated iteration between policy evaluation and policy improvement until the desired convergence criterion is met. PI method has been extensively investigated in a variety of situations [20], [32]. For the PI based Q-learning in Algorithm 1, it begins with an initial allowable control policy u_k^0. Then, update Q-function and control policy by

$$\begin{aligned} Z_k^T \tilde{H}^{j+1} Z_k &= X_k^T \Pi X_k + (u_k^j)^T R(u_k^j) \\ &\quad + \gamma Z_{k+1}^T \tilde{H}^{j+1} Z_{k+1} \qquad (19) \\ u_k^{j+1} &= \underset{u_k}{argmin} \left\{ X_k^T \Pi X_k + u_k^T R u_k \right. \\ &\quad \left. + \gamma Z_{k+1}^T \tilde{H}^j Z_{k+1} \right\} \qquad (20) \end{aligned}$$

To show the drawback, rewrite (19) as

$$\begin{aligned} Z_k^T \tilde{H}^{j+1} Z_k &= X_k^T \Pi X_k + (u_k^j)^T R(u_k^j) \\ &\quad + \gamma Z_{k+1}^T \tilde{H}^{j+1} Z_{k+1} \\ &= X_k^T \Pi X_k + (u_k^j)^T R(u_k^j) \\ &\quad + \gamma \left(X_{k+1}^T \Pi X_{k+1} + (u_{k+1}^j)^T R(u_{k+1}^j) \right) \\ &\quad + \gamma^2 Z_{k+2}^T \tilde{H}^{j+1} Z_{k+2} \\ &= \dots \\ &= \sum_{l=k}^{\infty} \gamma^{l-k} \left(X_l^T \Pi X_l + (u_l^j)^T R(u_l^j) \right) \qquad (21) \end{aligned}$$

Thus, it can seen that $Z_k^T \tilde{H}^{j+1} Z_k$ is in terms of an infinite-sum. If the stabilizing control policy is not used, it may go to infinity which makes no sense. The convergence of Algorithm 1 under an given initial admissible policy has been proven in [35]. Following the same line in [38], it is concluded that if an initial stabilizing control policy is used, all improved control policies u_j would be stabilizing.

In contrast to PI scheme, VI learning mechanism admits a more relaxed initial condition. It begins with an initial Q-function. Then, update control policy and Q-function by

$$\begin{aligned} u_k^j &= \underset{u_k}{argmin} \left\{ X_k^T \Pi X_k + u_k^T R u_k \right. \\ &\quad \left. + \gamma Z_{k+1}^T \tilde{H}^j Z_{k+1} \right\} \qquad (22) \\ Z_k^T \tilde{H}^{j+1} Z_k &= X_k^T \Pi X_k + (u_k^j)^T R(u_k^j) \\ &\quad + \gamma Z_{k+1}^T \tilde{H}^j Z_{k+1} \qquad (23) \end{aligned}$$

It is seen from (23) that $Z_k^T \tilde{H}^{j+1} Z_k$ is only the sum of a one-step utility function and the previous Q-function $Z_{k+1}^T \tilde{H}^j Z_{k+1}$. The finiteness of $Z_k^T \tilde{H}^{j+1} Z_k$ can be assured by a finite $Z_{k+1}^T \tilde{H}^j Z_{k+1}$. Hence, VI removes the requirement of initial stabilizing control policy.

B. VI Q-LEARNING FOR OPFB LQT

Algorithm 1 requires an initial admissible control policy, and thus may demand a prior knowledge of controlled system to conduct a robust design [21], [25]. As seen from the above subsection, VI learning mechanism would be more general since it meets a more free initial condition than PI method. Based on (23), we propose the VI based Q-learning Algorithm 2 for model-free LQT solution. The convergence property of Algorithm 2 can be concluded by the convergence analysis of Algorithm 3, as will be stated in the following subsection.

Algorithm 2 VI Q-learning Algorithm for LQT

Initialization

Select an initial policy u_k^0 not necessary stabilizing with $\tilde{H}^0$

Procedure

1: (Value Function Update) For $j = 0, 1, \dots$, collect data samples under

u_k^j to solve $\tilde{H}^{j+1}$ using the Q-function Bellman equation:

$$Z_k^T \tilde{H}^{j+1} Z_k = X_k^T \Pi X_k + (u_k^j)^T R(u_k^j) + \gamma Z_{k+1}^T \tilde{H}^j Z_{k+1} \qquad (24)$$

2: (Policy Improvement) Compute the improved control policy as follows:

$$u_k^{j+1} = -(\tilde{H}_{uu}^{j+1})^{-1} \tilde{H}_{uX}^{j+1} X_k \qquad (25)$$

3: (Stopping Criterion) Stop the iteration if $\|\tilde{H}^{j+1} - \tilde{H}^j\| < \varepsilon$ for some specified small positive number ε. Otherwise, let $j = j + 1$ and go back to iteration.

End Procedure

In Algorithm 2, the full state information x_k is required, which might not be always available in practice. In what follows, we will improve Algorithm 2 using the measured data by the past input, output, and reference trajectory sequence. The developed VI algorithm uses measured data for value function update and policy improvement, which is different from that in Algorithm 2 with state information. For this purpose, recall the following state reconstruction lemma [23].

Lemma 1: Under the condition that the pair (A, C) is observable, the augmented system state X_k can be expressed by past input, output, and reference trajectory sequence as

$$X_k = \begin{bmatrix} M_u & M_y & M_r \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ r_{k-N} \end{bmatrix} \qquad (26)$$

where $\bar{u}_{k-1,k-N}$ and $\bar{y}_{k-1,k-N}$, $N \leq n$, are the sequences of input and output signals over the time interval

$[k-N, k-1]$, respectively, defined by

$$\bar{u}_{k-1,k-N} = \begin{bmatrix} u_{k-1}^T & u_{k-2}^T & u_{k-3}^T & \cdots & u_{k-N}^T \end{bmatrix}^T$$
$$\bar{y}_{k-1,k-N} = \begin{bmatrix} y_{k-1}^T & y_{k-2}^T & y_{k-3}^T & \cdots & y_{k-N}^T \end{bmatrix}^T$$

and the coupling matrices are

$$M_u = \begin{bmatrix} U_N - A^N W_N^+ D_N \\ 0 \end{bmatrix}, \quad M_y = \begin{bmatrix} A^N W_N^+ \\ 0 \end{bmatrix}, \; M_r = \begin{bmatrix} 0 \\ F^N \end{bmatrix}$$

with

$$U_N = \begin{bmatrix} B_1 & AB_1 & \cdots & A^{N-1}B_1 \end{bmatrix},$$
$$W_N = \begin{bmatrix} (CA^{N-1})^T & \cdots & CA & C \end{bmatrix}^T,$$
$$D_N = \begin{bmatrix} 0 & CB & CAB & \cdots & CA^{N-2}B \\ 0 & 0 & CB & \cdots & CA^{N-3}B \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & CB \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and $W_N^+ = (W_N^T W_N)^{-1} W_N^T$.

Now, we can transform (16) into a Q-function Bellman equation defined in terms of the measured data by the past input, output, and reference trajectory sequence.

First, using (26) in (14) gives

$$Q(Z_k) = \frac{1}{2} Z_k^T \tilde{H} Z_k = \frac{1}{2} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ r_{k-N} \\ u_k \end{bmatrix}^T H \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ r_{k-N} \\ u_k \end{bmatrix}$$
$$\triangleq \frac{1}{2} z_k^T H z_k \tag{27}$$

with $z_k = \begin{bmatrix} \bar{u}_{k-1,k-N}^T & \bar{y}_{k-1,k-N}^T & r_{k-N}^T & u_k^T \end{bmatrix}^T$ and

$$H = H^T = \begin{bmatrix} H_{\bar{u}\bar{u}} & H_{\bar{u}\bar{y}} & H_{\bar{u}r} & H_{\bar{u}u} \\ H_{\bar{y}\bar{u}} & H_{\bar{y}\bar{y}} & H_{\bar{y}r} & H_{\bar{y}u} \\ H_{r\bar{u}} & H_{r\bar{y}} & H_{rr} & H_{ru} \\ H_{u\bar{u}} & H_{u\bar{y}} & H_{ur} & H_{uu} \end{bmatrix}$$

where the partitioned matrices are

$$\begin{aligned}
H_{\bar{u}\bar{u}} &= M_u^T(\Pi + \gamma T^T P T) M_u = M_u^T \tilde{H}_{XX} M_u, \\
H_{\bar{u}\bar{y}} &= M_u^T(\Pi + \gamma T^T P T) M_y = M_u^T \tilde{H}_{XX} M_u, \\
H_{\bar{u}r} &= M_u^T(\Pi + \gamma T^T P T) M_r = M_u^T \tilde{H}_{XX} M_u, \\
H_{\bar{u}u} &= \gamma M_u^T T^T P B_1 = M_u^T \tilde{H}_{Xu}, \\
H_{\bar{y}\bar{y}} &= M_y^T(\Pi + \gamma T^T P T) M_y = M_y^T \tilde{H}_{XX} M_y, \\
H_{\bar{y}r} &= M_y^T(\Pi + \gamma T^T P T) M_r = M_y^T \tilde{H}_{XX} M_r, \\
H_{\bar{y}u} &= \gamma M_y^T T^T P B_1 = M_y^T \tilde{H}_{Xu}, \\
H_{rr} &= M_r^T(\Pi + \gamma T^T P T) M_r = M_r^T \tilde{H}_{XX} M_r, \\
H_{ru} &= \gamma M_r^T T^T P B_1 = M_r^T \tilde{H}_{Xu}, \\
H_{uu} &= R + \gamma B_1^T P B_1 = \tilde{H}_{uu}.
\end{aligned} \tag{28}$$

It is observed from (27) that the Q-function is in a new terms of previous input, output and reference trajectory data. Performing the minimization of (27) with respect to u_k, we can obtain the optimal controller using input, output and reference trajectory sequence

$$\begin{aligned}
u_k^* &= -(H_{uu})^{-1}(H_{u\bar{u}}\bar{u}_{k-1,k-N} + H_{u\bar{y}}\bar{y}_{k-1,k-N} + H_{ur} r_{k-N}) \\
&= -(H_{uu})^{-1} \begin{bmatrix} H_{u\bar{u}} & H_{u\bar{y}} & H_{ur} \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ r_{k-N} \end{bmatrix} \\
&= -K^* \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ r_{k-N} \end{bmatrix}.
\end{aligned} \tag{29}$$

Applying (27) into (16), we have a Q-function Bellman equation in the input-output form

$$z_k^T H z_k = \tau_k^T \Gamma \tau_k + u_k^T R u_k + \gamma z_{k+1}^T H z_{k+1} \tag{30}$$

where $\tau_k = \begin{bmatrix} y_k \\ r_k \end{bmatrix}$ and $\Gamma = \begin{bmatrix} Q & -Q \\ -Q & Q \end{bmatrix}$.

It is now ready to propose the VI based Q-learning Algorithm 3 to learn the OPFB LQT controller.

Algorithm 3 VI Q-learning Algorithm for OPFB LQT

Initialization

Select an initial policy u_k^0 not necessary stabilizing with H^0

Procedure

1: (Value Function Update) For $j = 0, 1, \ldots$, collect data samples under

u_k^j to solve H^{j+1} using the Q-function Bellman equation:

$$z_k^T H^{j+1} z_k = \tau_k^T \Gamma \tau_k + (u_k^j)^T R u_k^j + \gamma z_{k+1}^T H^j z_{k+1}. \tag{31}$$

2: (Policy Improvement) Compute the improved control policy as follows:

$$\begin{aligned}
u_k^{j+1} &= -(H_{uu}^{j+1})^{-1} \\
&\quad \times (H_{u\bar{u}}^{j+1}\bar{u}_{k-1,k-N} + H_{u\bar{y}}^{j+1}\bar{y}_{k-1,k-N} + H_{ur}^{j+1} r_{k-N})
\end{aligned} \tag{32}$$

3: (Stopping Criterion) Stop the iteration if $\|\tilde{H}^{j+1} - \tilde{H}^j\| < \varepsilon$ for some specified small positive number ε. Otherwise, let $j = j + 1$ and go back to iteration.

End Procedure

Using Least-squares (LS) method, we can calculate H^{j+1} in the value function update step (31). For this purpose, we denote the linearly parameterized expression of $z_k^T H^{j+1} z_k$ as

$$z_k^T H^{j+1} z_k = (\bar{H}^{j+1})^T \bar{z}(k) \tag{33}$$

with

$$\begin{aligned}
\bar{H}^{j+1} &= vec(H^{j+1}) \in \mathbb{R}^{l(l+1)/2} \\
&\triangleq [H_{11}^{j+1}, 2H_{12}^{j+1}, \cdots, 2H_{1l}^{j+1}, H_{22}^{j+1}, \cdots, 2H_{2l}^{j+1}, \\
&\quad \cdots, H_{ll}^{j+1}]^T
\end{aligned}$$

where, the number H_{ik}^{j+1} denotes the element in ith row and kth column of the matrix H^{j+1}, $i,k = 1,\cdots,l$, and $l = mN + pN + m$. The regression vector $\bar{z}(k) \in \mathbb{R}^{l(l+1)/2}$ represented by $\bar{z}_k = z_k \otimes z_k$ is defined as the quadratic basis set formed by the Kronecker product $\bar{z} \triangleq [z_1^2, z_1z_2, \cdots, z_1z_l, z_2^2, z_2z_3, \cdots, z_2z_l, \cdots, z_l^2]$.

For the right hand side of (31), we denote the first tow terms as $r(\tau_k, u_k^j)$, i.e.,

$$r(\tau_k, u_k^j) = \tau_k^T \Gamma \tau_k + (u_k^j)^T R u_k^j \tag{34}$$

Using the denotations of (33) and (34), we can simplify (31) into

$$(\bar{H}^{j+1})^T \bar{z}_k = r(\tau_k, u_k^j) + \gamma (\bar{H}^j)^T \bar{z}_{k+1} \tag{35}$$

Since H^{j+1} is a symmetric $l \times l$ matrix with $l(l+1)/2$ independent elements, we need to collect $N \geq l(l+1)/2$ data samples $\bar{z}_k$ such that, the solution to (35) in the LS sense can be obtained by

$$\bar{H}^{j+1} = ((\Phi^j)^T(\Phi^j))^{-1}(\Phi^j)^T(\Upsilon^j + \gamma \Psi^j \bar{H}^j). \tag{36}$$

where

$$\Phi^j = \begin{bmatrix} \bar{z}_k \\ \bar{z}_{k+1} \\ \vdots \\ \bar{z}_{k+L-1} \end{bmatrix}, \quad \Upsilon^j = \begin{bmatrix} r(\tau_k, u_k^j) \\ r(\tau_{k+1}, u_{k+1}^j) \\ \vdots \\ r(\tau_{k+L-1}, u_{k+L-1}^j) \end{bmatrix},$$

$$\Psi^j = \begin{bmatrix} \bar{z}_{k+1} \\ \bar{z}_{k+2} \\ \vdots \\ \bar{z}_{k+L} \end{bmatrix}$$

with $\Phi^j \in \mathbb{R}^{L \times l(l+1)/2}$, $\Upsilon^j \in \mathbb{R}^{L \times 1}$ and $\Phi^j \in \mathbb{R}^{L \times l(l+1)/2}$.

Remark 3: When $k \leq N$, the input and output data $\bar{u}_{k-1,k-N}, \bar{y}_{k-1,k-N}$ are not available. To solve this technical dilemma, the internal model principle will be employed to collect the unavailable data. The detail of design is given in the following section. The internal model principle is effective to accommodate parameter variations of controlled system [39] while achieving the asymptotic tracking. Therefore, it is expected that the generated data would contain more inherent information for learning the optimal control solution.

Remark 4: In (36), $\bar{H}^{j+1}$ denotes the jth estimate of unknown vector H^{j+1} under the current policy. Based on the entries of vector $\bar{H}^{j+1}$, we can obtain the components of matrix H^{j+1}. The matrix H^{j+1} can then be used to compute u_k^{j+1} in the policy improvement step (32), which in turn are applied to generate the learned N data samples in the following $j+1$ iteration. To guarantee the unique solution to (36), the persistently exciting (PE) condition [19], [20] should be satisfied by imposing the exploration noise into control input. However, it results in a bias of optimal solution if VFA method is employed [22], [24].

We now show the benefits of employing Q-learning scheme which creates no bias in the parameter estimates leading to no bias of optimal solution.

Theorem 1: The exploration noise does not result in any bias in the Q-function estimates.

Proof: By Lemma 1, we know that the Q-function (27) in input-output form is equivalent to the original Q-function (14). Due to the excitation noise, the actual control input to collect data is $\hat{u}_k = u_k + w_k$ with w_k being the probing noise signals.

Hence, the Q-function (13) is rewritten as

$$Q(X_k, \hat{u}_k) = \frac{1}{2} X_k^T \Pi X_k + \frac{1}{2} \hat{u}_k^T R \hat{u}_k + \gamma V(X_{k+1}). \tag{37}$$

Let $\hat{H}$ be the estimate of $\tilde{H}$ obtained using the input $\hat{u}_k$. It then follows from (13) that

$$\begin{aligned} &\frac{1}{2}\begin{bmatrix} X_k \\ \hat{u}_k \end{bmatrix}^T \hat{H} \begin{bmatrix} X_k \\ \hat{u}_k \end{bmatrix} \\ &\quad = \frac{1}{2} X_k^T \Pi X_k + \frac{1}{2} \hat{u}_k^T R \hat{u}_k \\ &\qquad + \frac{1}{2} \gamma (TX_k + B_1 \hat{u}_k)^T P (TX_k + B_1 \hat{u}_k). \end{aligned}$$

Expanding both sides of the above equation yields

$$\begin{aligned} &\frac{1}{2}\begin{bmatrix} X_k \\ u_k \end{bmatrix}^T \hat{H} \begin{bmatrix} X_k \\ u_k \end{bmatrix} + \frac{1}{2}\gamma X_k^T T^T P B_1 w_k + \frac{1}{2}\gamma w_k^T B_1^T P T X_k \\ &\quad + \frac{1}{2} w_k^T (R + \frac{1}{2}\gamma B_1^T P B_1) u_k + \frac{1}{2} u_k^T (R + \frac{1}{2}\gamma B_1^T P B_1) w_k \\ &\quad + \frac{1}{2} w_k^T (R + \frac{1}{2}\gamma B_1^T P B_1) w_k \\ &= \frac{1}{2} X_k^T \Pi X_k + \frac{1}{2} u_k^T R u_k + \frac{1}{2} w_k^T R w_k + \frac{1}{2} w_k^T R u_k \\ &\quad + \frac{1}{2} u_k^T R w_k + \frac{1}{2}\gamma (TX_k + B_1 u_k)^T P (TX_k + B_1 u_k) \\ &\quad + \frac{1}{2}\gamma (TX_k + B_1 u_k)^T P B_1 w_k + \frac{1}{2}\gamma w_k^T B_1^T P B_1 w_k \\ &\quad + \frac{1}{2}\gamma (B_1 w_k)^T P (TX_k + B_1 u_k). \end{aligned}$$

We can observe that both sides of the equation have the same terms containing w_k, and therefore, we are left with

$$\begin{aligned} &\frac{1}{2}\begin{bmatrix} X_k \\ u_k \end{bmatrix}^T \hat{H} \begin{bmatrix} X_k \\ u_k \end{bmatrix} \\ &\quad = \frac{1}{2} X_k^T \Pi X_k + \frac{1}{2} u_k^T R u_k \\ &\qquad + \frac{1}{2}\gamma (TX_k + B_1 u_k)^T P (TX_k + B_1 u_k) \end{aligned} \tag{38}$$

Comparing (38) with (13) gives $\hat{H} = \tilde{H}$, that is

$$Q(X_k, u_k) = \frac{1}{2} X_k^T \Pi X_k + \frac{1}{2} u_k^T R u_k + \gamma V(X_{k+1}) \tag{39}$$

By (27), we have

$$z_k^T H z_k = \tau_k^T \Gamma \tau_k + u_k^T R u_k + \gamma z_{k+1}^T H z_{k+1}$$

Thus, we have obtained the Q-function Bellman equation in the absence of exploration noise, and it is the same as the one given in (30). Hence, there is no bias of Q-function after adding excitation noise. This completes the proof.

C. CONVERGENCE ANALYSIS

In this section, we propose a convergence analysis for the VI based Q-learning Algorithm 3, which shows that the policy matrix H^{j+1} converges to the optimal value H^* in each iteration. Then, according to (29), $u_k^j \to u_k^*$ as $j \longrightarrow \infty$. The convergence properties of VI scheme with state feedback have been investigated in recent studies based on VFA method [22], [26], [40]. In what follows, we will extend them to the OPFB Q-learning approach.

First of all, (31) can be rewritten as

$$\begin{aligned} Q^{j+1}(z_k) &= \min_{u_k}\left\{\frac{1}{2}r(\tau_k,u_k)+\gamma Q^j(z_{k+1})\right\} \\ &= \frac{1}{2}r(\tau_k,u_k^j)+\gamma Q^j(z_{k+1}) \end{aligned} \tag{40}$$

where $r(\tau_k,u_k^j) = \tau_k^T\Gamma\tau_k + (u_k^j)^T R u_k^j$ and $u_k^j = \arg\min_{u_k}\left\{\frac{1}{2}r(\tau_k,u_k)+\gamma Q^j(z_{k+1})\right\}$. Then, based on (40), the following two lemmas are proposed. The proof follows from the same line in [22].

Lemma 2: For system (5), given any arbitrary sequence of control policy $\{\vartheta^j\}$, where $j = 0, 1, \cdots, \infty$, let $\Lambda^j(z_k)$ be formulated as

$$\Lambda^{j+1}(z_k) = \frac{1}{2}\tau_k^T\Gamma\tau_k + \frac{1}{2}(\vartheta^j(z_k))^T R\vartheta^j(z_k) + \gamma\Lambda^j(z_{k+1}) \tag{41}$$

Let $\{u_k^j\}$ and $\{Q^j(z_k)\}$ be the sequences defined by (40). If $Q^0(z_k) = \Lambda^0(z_k) = 0$, then $Q^j(z_k) \le \Lambda^j(z_k)$, $\forall j$.

Proof: According to (40), $Q^{j+1}(z_k)$ is the minimum value with respect to u_k^j. Since $\Lambda^{j+1}(z_k)$ is the result of (41) in terms of arbitrary control policy ϑ^j, we have $Q^{j+1}(z_k) \le \Lambda^{j+1}(z_k)$. Therefore, if $Q^0(z_k) = \Lambda^0(z_k) = 0$, it follows that $Q^j(z_k) \le \Lambda^j(z_k)$, $\forall j$.

Lemma 3: Consider the sequence $\{Q^j(z_k)\}$ defined by (40). If system (5) is controllable, there exists an upper bound Υ such that $0 \le Q^j(z_k) \le \Upsilon$, $\forall j$.

Proof: Let $\{\upsilon_k\}$ be an arbitrary admissible control policy for LQT problem and $Q^j(z_k)$ be formulated as in (40). Define M^j as

$$M^{j+1}(z_k) = \frac{1}{2}\tau_k^T\Gamma\tau_k + \frac{1}{2}(\upsilon_k)^T R\upsilon_k + \gamma M^j(z_{k+1}) \tag{42}$$

with $Q^0(\cdot) = M^0(\cdot) = 0$. Then, it can be calculated that

$$\begin{aligned} M^{j+1}(z_k) - M^j(z_k) &= \gamma(M^j(z_{k+1}) - M^{j-1}(z_{k+1})) \\ &= \gamma^2(M^{j-1}(z_{k+2}) - M^{j-2}(z_{k+2})) \\ &= \gamma^3(M^{j-2}(z_{k+3}) - M^{j-3}(z_{k+3})) \\ &\vdots \\ &= \gamma^j(M^1(z_{k+j}) - M^0(z_{k+j})) \end{aligned} \tag{43}$$

Since $M^0(z_{k+j}) = 0$, it follows that

$$\begin{aligned} M^{j+1}(z_k) &= \gamma^j M^1(z_{k+j}) + M^j(z_k) \\ &= \gamma^j M^1(z_{k+j}) + \gamma^{j-1}M^1(z_{k+j-1}) + M^{j-1}(z_k) \\ &= \gamma^j M^1(z_{k+j}) + \gamma^{j-1}M^1(z_{k+j-1}) \\ &\quad + \gamma^{j-2}M^1(z_{k+j-2}) + \cdots + M^1(z_k) \end{aligned} \tag{44}$$

Hence, we have

$$\begin{aligned} M^{j+1}(z_k) &= \sum_{i=0}^{j}\gamma^i M^1(z_{k+i}) \\ &= \sum_{i=0}^{j}\gamma^i\left(\frac{1}{2}\tau_{k+i}^T\Gamma\tau_{k+i} + \frac{1}{2}(\upsilon_{k+i})^T R\upsilon_{k+i}\right) \\ &\le \sum_{i=0}^{\infty}\gamma^i\left(\frac{1}{2}\tau_{k+i}^T\Gamma\tau_{k+i} + \frac{1}{2}(\upsilon_{k+i})^T R\upsilon_{k+i}\right) \end{aligned} \tag{45}$$

Notice that υ_k is an admissible tracking control policy. Therefore, there exists an upper bound Υ such that

$$\forall j : M^{j+1}(z_k) \le \sum_{i=0}^{\infty}\gamma^i M^1(z_{k+i}) \le \Upsilon \tag{46}$$

Since Q and R are both positive definite, $r(\tau_k,u_k^0)$ in (40) is nonnegative. Considering $Q^0(z_k) = 0$, we have $Q^1(z_k) \ge 0$. By the mathematical induction, it is easy to obtain $Q^j(z_k) \ge 0$, $\forall j$. By Lemma 2 and $Q^0(\cdot) = M^0(\cdot) = 0$, it can be concluded that

$$0 \le Q^j(z_k) \le M^j(z_k) \le \Upsilon \tag{47}$$

Using Lemma 3 gives the following theorem.

Theorem 2: let $\{Q^j(z_k)\}$ be the sequence defined by (40) with $Q^0(\cdot) = 0$. Then, the Q-function sequence $\{Q^j(z_k)\}$ is monotonically non-decreasing such that $Q^{j+1}(z_k) \ge Q^j(z_k)$, $\forall j$. Moreover, it follows that $Q^j(z_k) \to Q^*(z_k)$ as $j \to \infty$ with $Q^*(z_k)$ the solution of Bellman optimization equation, that is

$$Q^*(z_k) = \min_{u_k}\left\{\frac{1}{2}\tau_k^T\Gamma\tau_k + \frac{1}{2}u_k^T R u_k + \gamma Q^*(z_{k+1})\right\} \tag{48}$$

Proof: Since $Q^0(\cdot) = 0$, we have $Q^1(z_k) \ge 0$. Thus, $Q^{j+1}(z_k) \ge Q^j(z_k)$ holds when $j = 0$. By the mathematical induction method, assume that $Q^{j+1}(z_k) \ge Q^j(z_k)$ holds when $j = \rho$, $\rho \ge 0$. Then, one has $Q^{\rho+1}(z_k) \ge Q^\rho(z_k)$. Take $j = \rho + 1$, one has

$$\begin{aligned} Q^{\rho+2}(z_k) &= \min_{u_k}\left\{\frac{1}{2}\tau_k^T\Gamma\tau_k + \frac{1}{2}u_k^T R u_k + \gamma Q^{\rho+1}(z_{k+1})\right\} \\ &\ge \min_{u_k}\left\{\frac{1}{2}\tau_k^T\Gamma\tau_k + \frac{1}{2}u_k^T R u_k + \gamma Q^{\rho}(z_{k+1})\right\} \\ &= Q^{\rho+1}(z_k) \end{aligned} \tag{49}$$

Therefore, $Q^{j+1}(z_k) \ge Q^j(z_k)$ holds when $j = \rho + 1$. And we can conclude that $\{Q^j(z(\cdot))\}$ is a non-decreasing sequence with $Q^{j+1}(z_k) \ge Q^j(z_k)$, $\forall j$. Moreover, as can be seen in Lemma 3, $Q^j(z_k)$ is upper bounded. Hence, $Q^j(z_k) \to Q^\infty(z_k)$ as $j \to \infty$.

In this respect, we are ready to prove that $Q^j(z_k) \to Q^*(z_k)$ as $j \to \infty$. Obviously, it is equivalent to show that Q^∞ satisfies the Bellman optimality equation, i.e.,

$$Q^\infty(z_k) = \min_{u_k}\left\{\frac{1}{2}\tau_k^T\Gamma\tau_k + \frac{1}{2}u_k^T R u_k + \gamma Q^\infty(z_{k+1})\right\} \tag{50}$$

On the one hand, by putting $Q^j(z_k) \leq Q^\infty(z_k)$ into (40), we can have

$$\forall j : Q^j(z_k) \leq \min_{u_k} \left\{ \frac{1}{2}\tau_k^T \Gamma \tau_k + \frac{1}{2}u_k^T R u_k + \gamma Q^\infty(z_{k+1}) \right\}$$

yielding

$$Q^\infty(z_k) \leq \min_{u_k} \left\{ \frac{1}{2}\tau_k^T \Gamma \tau_k + \frac{1}{2}u_k^T R u_k + \gamma Q^\infty(z_{k+1}) \right\} \quad (51)$$

as $j \to \infty$.

On the other hand, since $Q^\infty(z_k) \geq Q^j(z_k)$, we can obtain

$$\forall j : Q^\infty(z_k) \geq \min_{u_k} \left\{ \frac{1}{2}\tau_k^T \Gamma \tau_k + \frac{1}{2}u_k^T R u_k + \gamma Q^{j-1}(z_{k+1}) \right\}$$

Let $j \to \infty$, then

$$Q^\infty(z_k) \geq \min_{u_k} \left\{ \frac{1}{2}\tau_k^T \Gamma \tau_k + \frac{1}{2}u_k^T R u_k + \gamma Q^\infty(z_{k+1}) \right\} \quad (52)$$

Based on (51) and (52), it can be concluded that $Q^\infty(z_k)$ is the solution of Bellman optimality equation (50). This completes the proof.

Remark 5: It has been proved that the Q-function will converge to the optimal value in infinite iterations. Besides, in finite iterations, the Q-function will arrive in a compact set around $Q^*(z_k)$, i.e., $\forall \varepsilon, \exists N, |Q^j(z_k) - Q^*(z_k)| < \varepsilon$ when $j > N$. As shown in (40), $Q^{j+1}(z_k) = r(\tau_k, u_k^j) + \gamma Q^j(z_{k+1})$. Then, we have $Q^{j+1}(z_k) = r(\tau_k, u_k^j) + \gamma r(\tau_{k+1}, u_{k+1}^{j-1}) + \ldots + \gamma^{K-k} r(\tau_K, u_K^{j-K+k})$, which means that the value of Q-function at $(j+1)$th iteration is the sum of utility functions at different iterations and different times. The previous recent iterations give more contribution and the selecting of γ has an influence on the convergence speed of iterative Q-function.

IV. INTERNAL MODEL CONTROLLER DESIGN

In this section, we review the tracking controller design for DTL systems based on internal model principle. The internal model-based design claims that if a closed-loop system incorporates in the feedback path a reduplicated model of the references, it can achieve the asymptotic tracking under the parameter variations within a small range [41].

In the following, the process of internal model design is described for the sinusoidal reference tracking. For more details, one can refer to [42]. Consider the DTL system (1) with exogenous system (2) where

$$r_k = \begin{bmatrix} r_{k1} \\ r_{k2} \end{bmatrix}, \quad F = F(\sigma) = \begin{bmatrix} 0 & \sigma \\ -\sigma & 0 \end{bmatrix}$$

with σ the frequency of sinusoidal signal. Then the tracking control problem can be formulated as

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \\ r_{k+1} &= F(\sigma) r_k \\ e_k &= y_k - r_{k1} \\ r_{k1} &= \begin{bmatrix} 1 & 0 \end{bmatrix} r_k \end{aligned} \quad (53)$$

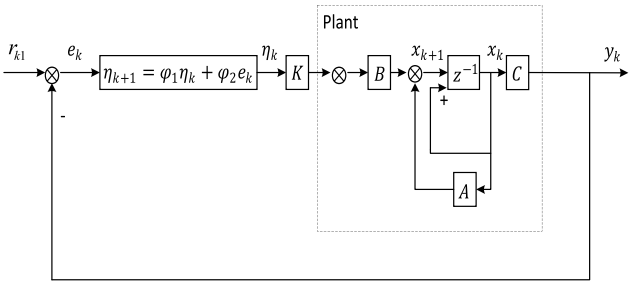

FIGURE 1. Dynamic output feedback control design with internal model.

According to the internal model principle, the dynamic output feedback control law [42]

$$\begin{aligned} u_k &= K\eta_k \\ \eta_{k+1} &= \begin{bmatrix} A + BK_x - LC & BK_\eta \\ 0 & G_1 \end{bmatrix} \eta_k + \begin{bmatrix} L \\ G_2 \end{bmatrix} e_k \\ &\triangleq \varphi_1 \eta_k + \varphi_2 e_k \\ G_1 &= \Psi \begin{bmatrix} 0 & 1 \\ -\sigma^2 & 0 \end{bmatrix} \Psi^{-1} \\ G_2 &= \Psi \begin{bmatrix} 0 & 1 \end{bmatrix}^T \end{aligned} \quad (54)$$

solves the tracking control problem regardless of small variations of the entries in the matrices A, B, and C. In (54), Ψ is chosen as a two-dimensional identity matrix and L is selected such that $A - LC$ is Schur. The matrix $K = \begin{bmatrix} K_x & K_\eta \end{bmatrix}^T$ is appropriately designed such that

$$\begin{bmatrix} A + BK_x & BK_\eta \\ G_2 C & G_1 \end{bmatrix} \quad (55)$$

is Hurwitz, where the classic methods such as Lyapunov method and pole placement can be applied. Until now, the tracking error e_k can converge to zero under some parameter variations of the controlled system. The design scheme of internal model principle is shown in Figure 1.

The internal model principle has the potential to maintain asymptotic tracking under parameter changes, and therefore, it allows the unknown system to have a good tendency to track the reference signal. By such a process of collecting unavailable measured data, not only it solves the dilemma as seen in Remark 3, but also it may be more efficient to make the unknown system track the given signal when utilizing the Q-learning method.

V. SIMULATION RESULTS

In this section, we propose a simulation example to verify the effectiveness of developed OPFB Q-learning algorithm based on VI scheme. Consider a DTL system described by

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1.69 & -0.67 \\ 1 & 0 \end{bmatrix} x_k + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_k \\ y_k &= \begin{bmatrix} 1 & -0.5 \end{bmatrix} x_k \end{aligned} \quad (56)$$

with $x_0 = \begin{bmatrix} 0.65 & 0 \end{bmatrix}$. It can be calculated that the open-loop poles are $z_1 = 1.0548$ and $z_2 = 0.6352$ such that the system

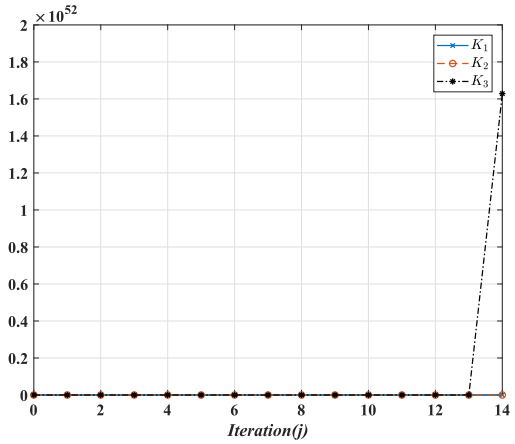

FIGURE 2. Control gain obtained by Algorithm 1.

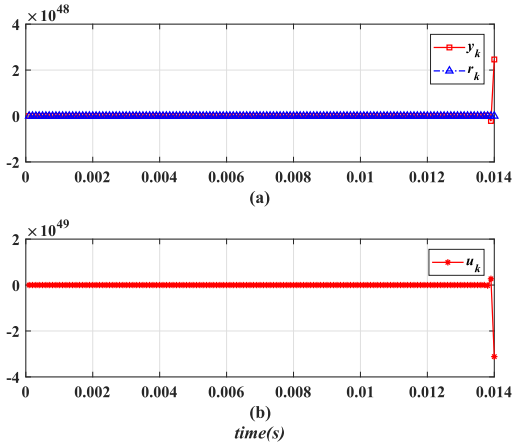

FIGURE 3. Trajectories of output and input by Algorithm 1.

is unstable. The PI based Algorithm 1 requires an initial admissible control policy as well as the available full-state information. By employing VI scheme combined with the state reconstruction method, both of these two requirements have been relaxed in the proposed Algorithm 3.

A. COMPARATIVE STUDIES

In this subsection, comparative studies are conducted on Algorithm 1 and Algorithm 2 without the initial stabilizing control policy, i.e., $u_k^0 = 0$. Compared with the PI-based Algorithm 1, it is verified that the VI-based Algorithm 2 removes the requirement of initial stabilizing control policy.

The desired reference trajectory is generated by $r_{k+1} = 0.95\ r_k$ with $r_0 = 0.5$. The performance function (4) is considered with $Q = 1, R = 0.001$ and $\gamma = 1$. Thus the optimal state feedback gain can be obtained as $K_s^* = \begin{bmatrix}1.1887 & -0.6693 & -0.9488\end{bmatrix}$. During the learning phase, the injected excitation signal is $w_k = 0.7\ sin(k) + 0.5\ cos(2k) + 0.9\ sin(8k) + 0.2\ cos(6k)$. First, Algorithm 1 is employed and it is found that the convergence can not be achieved. Hence, we terminate the learning at $j = 14$th iteration. Figure 2 gives the evolution of control parameters.

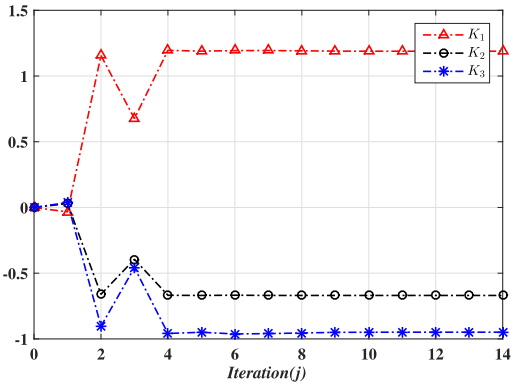

FIGURE 4. Control gain obtained by Algorithm 2.

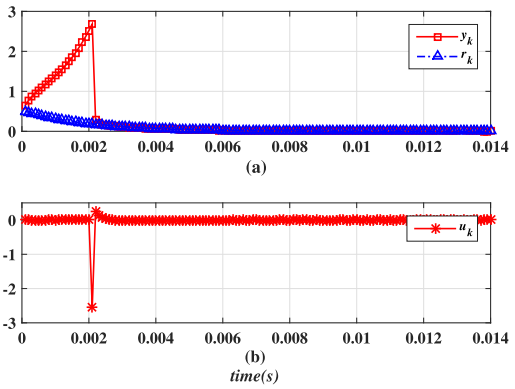

FIGURE 5. Trajectories of output and input by Algorithm 2.

Figure 3 shows that the output cannot track the reference whilst the trajectories of output and input diverge as time increases.

The corresponding results for using Algorithm 2 are shown in Figures 4 and 5. For comparison fairness, we perform Algorithm 2 with the same parameters and excitation signal as Algorithm 1. It is observed from Figure 4 that Algorithm 2 achieves convergence of the optimal control gain. This is mainly because the VI mechanism allows learning to be performed without an initial admissible policy. In Figure 5, the trajectories of output and reference are shown, which illustrates the guaranteed tracking.

B. VI Q-LEARNING UNDER PARAMETER VARIATIONS

In this subsection, further simulations are conducted to study the effectiveness of Algorithm 3. Specifically, the parameter variations of controlled system would be considered. For system (56), we first perform the simulation with the internal model controller designed under the nominal parameters. Let y_r be the desired reference trajectory generated by

$$\begin{aligned}\dot{x}_r &= \begin{bmatrix}0 & 100\pi\\ -100\pi & 0\end{bmatrix} x_r\\ y_r &= \begin{bmatrix}1 & 0\end{bmatrix} x_r\end{aligned} \tag{57}$$

with the initial condition $x_r(0) = \begin{bmatrix}0 & 1\end{bmatrix}^T$. Hence, the reference trajectory is a sinusoidal waveform with a amplitude of one and a frequency of 50 Hz. Under the specified sampling

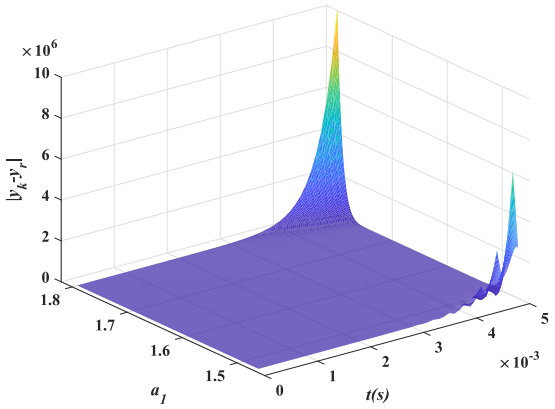

FIGURE 6. Error $|y_k - y_r|$ with different a_1 by internal model design.

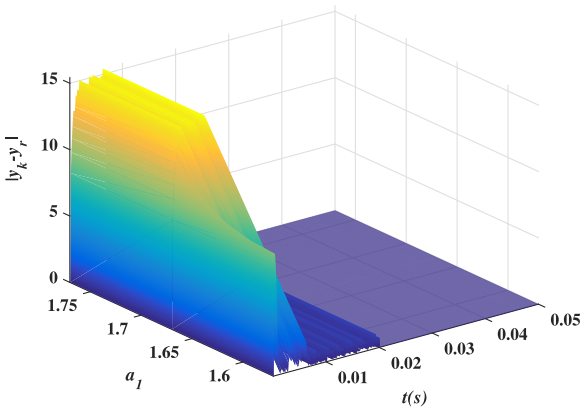

FIGURE 7. Error $|y_k - y_r|$ with different a_1 by VI based OPFB Q-learning.

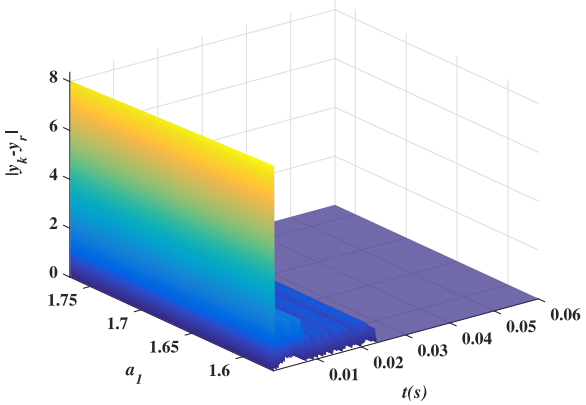

FIGURE 8. Error $|y_k - y_r|$ with different a_1 by VI based OPFB Q-learning.

rate $f = 10000\ Hz$, the discrete counterpart of (57) can be obtained in the form of (2) with

$$F = \begin{bmatrix} 0.9995 & 0.0314 \\ -0.0314 & 0.9995 \end{bmatrix} \tag{58}$$

The matrices

$$L = \begin{bmatrix} -4.0433 & -13.6667 \end{bmatrix}^T$$

and

$$K = \begin{bmatrix} 3.3990 & -1.7860 & 1.1058e8 & 2.5900e4 \end{bmatrix}$$

are obtained by employing the pole placement method. Figure 6 shows the profile of tracking error when the element $a_1 = 1.69$ in system (56) ranges from 1.47 to 1.80. It can be observed that when the parameter gets away from its nominal value, the tracking error displays a sizable deviation from zero. Thus we have seen that the tracking performance is only robust with respect to small parameter variations.

In contrast to the robust design based on internal model, it is expected that a Q-learning controller can maintain its tracking performance with optimal criterion in the presence of parameter variations. Moreover, the learning data usually cannot be obtained from the full-state information. With the above observations, we use VI Q-learning Algorithm 3 based on measured data along the system trajectories. The performance function (4) is considered with $Q = 1, R = 0.001$ and $\gamma = 0.09$. The excitation signal injected in the control channel is $w_k = 0.7\ sin(k) + 0.5\ cos(2k) + 0.9\ sin(8k) + 0.2\ cos(6k)$. The simulation using Algorithm 3 without an initial admissible policy is performed. Figure 7 shows that the controlled output y_k achieves tracking of the reference trajectory y_r under different values a_1.

With the same parameters and the same excitation signal, we also perform Algorithm 3 using an initial stabilizing control policy obtained under the nominal system, i.e.,

$$u_k^0 = -\begin{bmatrix} 1.6000 \\ -1.0500 \\ 3.0100 \\ -1.6170 \\ 0_{2\times 1} \end{bmatrix}^T \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ r_{k-N} \end{bmatrix} \tag{59}$$

Figure 8 gives the evolution of tracking error with initial policy (59). As can be seen, the output tracking of reference signal is assured. Furthermore, the overshoot in the tracking performance is improved and it may result from using the initial stabilizing policy.

From the simulation results, it is observed that the VI based OPFB Q-learning can achieve tracking no matter whether an initial admissible policy is given or not. The developed learning algorithm is able to maintain the tracking property regardless of parameter variations. As claimed in Remark 3, the internal model controller based on nominal parameters is used to provide the unavailable initial data during the OPFB learning.

C. INFLUENCE OF THE EXPLORATION NOISE

In the following, the unbiasedness of Q-learning algorithm would be shown, and therefore, we can verify that the convergence to optimal tracking control gain is completely immune to the excitation noise. The simulation is conducted on Algorithm 3 with the same parameters as the above subsection and is considered without the initial admissible policy.

Four different excitation noises are respectively injected:

*Case*1 :

$$w_k = 0.5sin(2.0k)^2cos(10.1k) + 0.9sin(1.102k)^2 \times cos(4.001k) + 0.3sin(1.99k)^2cos(7k)$$

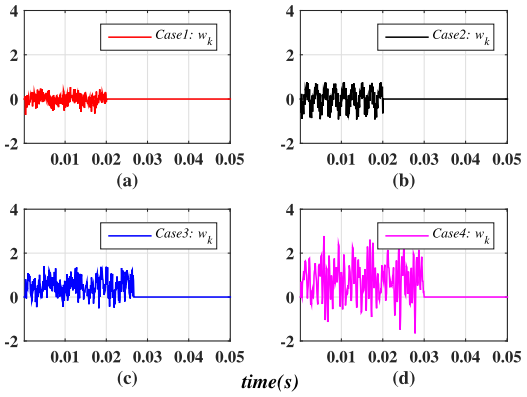

FIGURE 9. Excitation noises during the learning process.

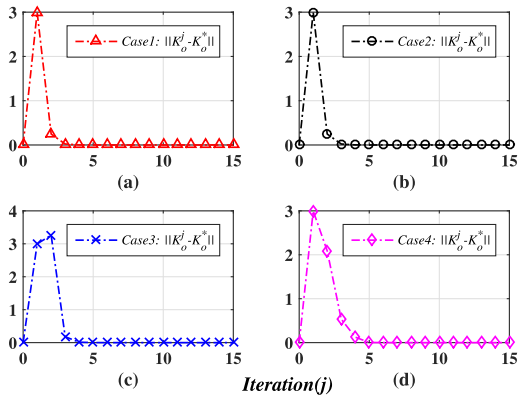

FIGURE 10. Convergence of OPFB control gain matrix under different excitation noises.

$$+0.3sin(10.0k)^3cos(9.65k)$$

*Case*2 :

$$w_k = 0.7sin(2k) + 0.5cos(3k) + 0.9sin(6k) + 0.2cos(8k)$$

*Case*3 :

$$\begin{aligned} w_k = {} & sin(100k)^2cos(100k) + sin(2k)^2cos(0.1k) \\ & + sin(-1.2k)^2cos(0.5k) + sin(k)^2 \\ & + sin(1.12k)^2 \end{aligned}$$

*Case*4 :

$$\begin{aligned} w_k = {} & sin(1.009k) + 2cos(0.538k)cos(0.538k) \\ & + sin(0.9k) + cos(100k) + sin(2.781k)^2 \\ & + sin(0.157k)^3 + cos(0.349k)sin(4.199k)^2 \end{aligned}$$

Figure 9 gives the excitation noises injected to the control channel during the learning phase. It can be observed that the excitation noises are in different frequencies and magnitudes. However, Figure 10 shows that, although different excitation noises result in different convergence time, the learned controller gain can converge to the theoretical optimal value for all four cases. Unlike VFA method, the simulation results confirm that the Q-learning mechanism would not bring bias on learning solution under the excitation noise.

VI. CONCLUSION

In this paper, we have studied the VI based Q-learning algorithm for model-free OPFB tracker design of DTL systems. Using the augmented system approach, we have transformed this problem into a regulation problem with a discounted performance function, which relies on the Q-function Bellman equation. For the solution of Bellman equation, we have employed the VI learning mechanism to remove the requirement of initial admissible policy, which involves the measurement of past input, output, and reference trajectory data. Therefore, it provides a novel solution without the state measurement. Furthermore, the internal model is incorporated to provide the unavailable initial data. The effectiveness of proposed design is shown by the application to a simulation example. Future work would include extending this result to the unknown discrete-time multi-agent systems. It is also interesting to investigate how to classify the initial data provided by internal model controller in a practical experimental platform as well as how to choose the value of discount factor for a practical unknown systems.

REFERENCES

[1] F. L. Lewis, D. L. Vrabie, and V. L. Syrmos, *Optimal Control*, 3rd ed. Hoboken, NJ, USA: Wiley, 2015.
[2] G. Hewer, "An iterative technique for the computation of the steady state gains for the discrete optimal regulator," *IEEE Trans. Autom. Control*, vol. AC-16, no. 4, pp. 382–384, Aug. 1971.
[3] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*. London, U.K.: Oxford Univ. Press, 1995.
[4] S.-L. Dai, C. Wang, and M. Wang, "Dynamic learning from adaptive neural network control of a class of nonaffine nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 111–123, Jan. 2014.
[5] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 3, pp. 334–344, Mar. 2016.
[6] N. T. Luy, "Robust adaptive dynamic programming based online tracking control algorithm for real wheeled mobile robot with omni-directional vision system," *Trans. Inst. Meas. Control*, vol. 39, no. 6, pp. 832–847, Jun. 2017.
[7] W. He, T. Meng, X. He, and S. S. Ge, "Unified iterative learning control for flexible structures with input constraints," *Automatica*, vol. 96, pp. 326–336, Oct. 2018.
[8] M.-B. Radac and R.-E. Precup, "Data-driven model-free tracking reinforcement learning control with VRFT-based adaptive actor-critic," *Appl. Sci.*, vol. 9, no. 9, pp. 1807-1–1807-24, 2019.
[9] Z. Wang and D. Liu, "Data-based controllability and observability analysis of linear discrete-time systems," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2388–2392, Dec. 2011.
[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1998.
[11] Q. Shen, P. Shi, J. Zhu, S. Wang, and Y. Shi, "Neural networks-based distributed adaptive control of nonlinear multiagent systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 3, pp. 1010–1021, Mar. 2020.
[12] Q. Shen, P. Shi, J. Zhu, and L. Zhang, "Adaptive consensus control of leader-following systems with transmission nonlinearities," *Int. J. Control*, vol. 92, no. 2, pp. 317–328, Feb. 2019.
[13] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Syst.*, vol. 12, no. 2, pp. 19–22, Apr. 1992.
[14] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Aug. 2009.
[15] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

[16] Z.-P. Jiang and Y. Jiang, "Robust adaptive dynamic programming for linear and nonlinear systems: An overview," *Eur. J. Control*, vol. 19, no. 5, pp. 417–425, Sep. 2013.
[17] K. Zhang, H.-G. Zhang, Y. Cai, and R. Su, "Parallel optimal tracking control schemes for mode-dependent control of coupled Markov jump systems via integral RL method," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 3, pp. 1332–1342, Jul. 2020.
[18] K. Zhang, H. Zhang, Y. Mu, and C. Liu, "Decentralized tracking optimization control for partially unknown fuzzy interconnected systems via reinforcement learning method," *IEEE Trans. Fuzzy Syst.*, early access, Jan. 13, 2020, doi: 10.1109/TFUZZ.2020.2966418.
[19] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, Feb. 2009.
[20] Y. Jiang and Z.-P. Jiang, "Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics," *Automatica*, vol. 48, no. 10, pp. 2699–2704, Oct. 2012.
[21] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Trans. Autom. Control*, vol. 59, no. 11, pp. 3051–3056, Nov. 2014.
[22] X. Li, L. Xue, and C. Sun, "Linear quadratic tracking control of unknown discrete-time systems using value iteration algorithm," *Neurocomputing*, vol. 314, pp. 86–93, Nov. 2018.
[23] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 14–25, Feb. 2011.
[24] B. Kiumarsi, F. L. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input-output measured data," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2770–2779, Dec. 2015.
[25] W. Gao, M. Huang, Z.-P. Jiang, and T. Chai, "Sampled-data-based adaptive optimal output-feedback control of a 2-degree-of-freedom helicopter," *IET Control Theory Appl.*, vol. 10, no. 12, pp. 1440–1447, Aug. 2016.
[26] G. Xiao, H. Zhang, K. Zhang, and Y. Wen, "Value iteration based integral reinforcement learning approach for H_∞ controller design of continuous-time nonlinear systems," *Neurocomputing*, vol. 285, pp. 51–59, Apr. 2018.
[27] M.-B. Radac and T. Lala, "Learning output reference model tracking for higher-order nonlinear systems with unknown dynamics," *Algorithms*, vol. 12, no. 6, pp. 121-1–121-23, 2019.
[28] P. Shi and Q. K. Shen, "Observer-based leader-following consensus of uncertain nonlinear multi-agent systems," *Int. J. Robust Nonlinear Control*, vol. 27, no. 17, pp. 3794–3811, 2017.
[29] S. A. A. Rizvi and Z. Lin, "Output feedback Q-learning control for the discrete-time linear quadratic regulator problem," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1523–1536, Oct. 2019.
[30] L. M. Zhu, H. Modares, G. O. Peen, F. L. Lewis, and B. Yue, "Adaptive suboptimal output-feedback control for linear systems using integral reinforcement learning," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 264–273, Jan. 2015.
[31] R. Moghadam and F. L. Lewis, "Output-feedback H_∞ quadratic tracking control of linear systems using reinforcement learning," *Int. J. Adapt. Control Signal Process.*, vol. 33, no. 2, pp. 300–314, Feb. 2019.
[32] A. Parviz Valadbeigi, A. Khaki Sedigh, and F. L. Lewis, "H_∞ static output-feedback control design for discrete-time systems using reinforcement learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 396–406, Feb. 2020.
[33] S. A. A. Rizvi and Z. Lin, "Output feedback Q-learning for discrete-time linear zero-sum games with application to the H-infinity control," *Automatica*, vol. 95, pp. 213–221, Sep. 2018.
[34] Y. Peng, Q. Chen, and W. Sun, "Reinforcement Q-learning algorithm for H_∞ tracking control of unknown discrete-time linear systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Dec. 20, 2019, doi: 10.1109/TSMC.2019.2957000.
[35] B. Kiumarsi, F. L. Lewis, H. Modares, A. Karimpour, and M. B. Naghibi-Sistani, "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics," *Automatica*, vol. 50, no. 4, pp. 1167–1175, 2014.
[36] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.
[37] B. Kiumarsi and F. L. Lewis, "Output synchronization of heterogeneous discrete-time systems: A model-free optimal approach," *Automatica*, vol. 84, pp. 86–94, Oct. 2017.
[38] D. Kleinman, "On an iterative technique for Riccati equation computations," *IEEE Trans. Autom. Control*, vol. 13, no. 1, pp. 114–115, Feb. 1968.
[39] W. Sun and J. Huang, "On a robust synchronization problem via internal model approach," *Asian J. Control*, vol. 12, no. 1, pp. 103–109, 2010.
[40] J. Shi, D. Yue, and X. Xie, "Adaptive optimal tracking control for nonlinear continuous-time systems with time delay using value iteration algorithm," *Neurocomputing*, vol. 396, pp. 172–178, Jul. 2020.
[41] B. A. Francis and W. M. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, Sep. 1976.
[42] J. Huang, *Nonlinear Output Regulation: Theory and Applications*. Philadelphia, PA, USA: SIAM, 2004.

CONG CHEN received the B.Eng. degree in automation from Shandong University, Jinan, China, in 2019. He is currently pursuing the M.S. degree in control engineering with the South China University of Technology, Guangzhou, China.

His research interests include adaptive and reinforcement learning and power electronics.

WEIJIE SUN (Member, IEEE) received the B.Eng. degree in thermal engineering from Jiangsu University, Zhenjiang, China, in 2003, the M.S. degree in control theory and control engineering from Fuzhou University, Fuzhou, China, in 2006, and the Ph.D. degree in control theory and control engineering from the South China University of Technology, Guangzhou, China, in 2009.

He is currently an Associate Professor at the School of Automation Science and Engineering, South China University of Technology, Guangzhou. His research interests include adaptive and learning control, robotics and automation, and PV power system modeling and control.

GUANGYUE ZHAO received the B.Eng. degree in automation control from the Hunan University of Science and Technology, Xiangtan, China, in 2017. He is currently pursuing the M.S. degree in control engineering with the South China University of Technology, Guangzhou, China.

His research interests include adaptive and reinforcement learning and PV power system modeling and control.

YUNJIAN PENG (Member, IEEE) received the B.Eng. degree in geotechnical investigation engineering from the Central South University of Technology, Changsha, China, in 1996, the M.S. degree in control theory and control engineering from Central South University (CSU), Changsha, in 2002, and the Ph.D. degree in control theory and control engineering from the South China University of Technology, Guangzhou, China, in 2007.

He is currently an Associate Professor at the School of Automation Science and Engineering, South China University of Technology, Guangzhou. His research interests include adaptive and learning control, stochastic dynamic system analysis and control, PV power system modeling and control, and information system engineering.