mathematics

MDPI

Article

The Adaptive Optimal Output Feedback Tracking Control of Unknown Discrete-Time Linear Systems Using a Multistep Q-Learning Approach

Xunde Dong [1], Yuxin Lin [1], Xudong Suo [2], Xihao Wang [1] and Weijie Sun [3,*]

[1] School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China; audxd@scut.edu.cn (X.D.); 202130462486@mail.scut.edu.cn (Y.L.); 202321017516@mail.scut.edu.cn (X.W.)

[2] Intelligent Mobile Robot Research Institute (Zhongshan), Zhongshan 528478, China; suoxudong@imrobotri.com

[3] School of Automation Science and Engineering, Key Laboratory of Autonomous Systems and Networked Control, Ministry of Education, Guangdong Engineering Technology Research Center of Unmanned Aerial Vehicle System, South China University of Technology, Guangzhou 510641, China

* Correspondence: auwjsun@scut.edu.cn

Abstract: This paper investigates the output feedback (OPFB) tracking control problem for discrete-time linear (DTL) systems with unknown dynamics. To solve this problem, we use an augmented system approach, which first transforms the tracking control problem into a regulation problem with a discounted performance function. The solution to this problem is derived using a Bellman equation, based on the Q-function. In order to overcome the challenges of unmeasurable system state variables, we employ a multistep Q-learning algorithm that surpasses the advantages of the policy iteration (PI) and value iteration (VI) techniques and state reconstruction methods for output feedback control. As such, the requirement for an initial stabilizing control policy for the PI method is removed and the convergence speed of the learning algorithm is improved. Finally, we demonstrate the effectiveness of the proposed scheme using a simulation example.

Keywords: tracking; Q-learning; optimal control; output feedback; UPS

MSC: 49M25

check for updates

Citation: Dong, X.; Lin, Y.; Suo, X.; Wang, X.; Sun, W. The Adaptive Optimal Output Feedback Tracking Control of Unknown Discrete-Time Linear Systems Using a Multistep Q-Learning Approach. *Mathematics* **2024**, *12*, 509. https://doi.org/10.3390/math12040509

Academic Editor: Juan Ramón Torregrosa Sánchez

Received: 2 January 2024
Revised: 23 January 2024
Accepted: 31 January 2024
Published: 6 February 2024

1. Introduction

The optimization of performance costs has always been a crucial concern in controller design problems as it can lead to energy savings and, subsequently, have a positive impact on the environment. The development of practical requirements has significantly contributed to the advancement of optimal control [1–4]. The key challenge in optimal control lies in solving the Riccati equation for linear systems. In the case of linear systems, computationally efficient iterative algorithms [5,6] can be employed to obtain the solution to the Riccati equation. However, this method is only applicable when a comprehensive understanding of the system dynamics is available. In control engineering, online learning controllers have commonly been designed without complete knowledge of the system dynamics [7–11]. Notably, a data-based approach was proposed in [12] for analyzing the controllability and observability of discrete-time linear (DTL) systems without the precise knowledge of system parameters.

Reinforcement learning (RL) is a powerful method for optimizing rewards via interactions with the environment [13]. Utilizing RL techniques, controller performance can be enhanced based on reward signals [14] and controller parameters can be updated to achieve optimal design criteria for adaptive control. Consequently, RL has provided valuable insights into the field of control systems [14,15], augmented by the introduction of the adaptive dynamic programming (ADP) approach, which aims to achieve optimal performance indices for (partially) model-free scenarios [15–19]. Extensive research has been conducted on developing optimal control schemes based on the ADP concept, particularly for applications in linear quadratic regulator (LQR) and linear quadratic tracking (LQT) problems, which was outlined comprehensively in [2,20–23] and other related references. It is worth noting that learning schemes in reinforcement learning generally involve two iterative steps: policy evaluation and policy update (with the latter focusing on policy improvement). However, it is essential to acknowledge that reinforcement learning based on value function approximation (VFA) introduces deliberate exploration noise to fully investigate systems, thereby undermining the algorithm's convergence [23–25]. Furthermore, the policy iteration (PI) scheme within the adaptive dynamic programming (ADP) framework necessitates an initially admissible policy, which demands a priori knowledge of unknown systems to design robust controllers [22,26]. To overcome this requirement, recent studies have adopted value iteration (VI) methods [23,27,28] within value function approximation (VFA) schemes. Recently, event-triggered control approaches have also been applied to solve the adaptive optimal output regulation problem using PI and VI methods with one-step learning [29].

Most current studies in the field of control engineering have relied on the ability to measure the complete state information of systems [23,30], which is often challenging to achieve in practical engineering applications [31]. As such, the development of output feedback learning controllers has become essential. In the literature, dynamic output feedback controllers have been investigated [32], which rely on the Q-learning algorithm to solve the LQR control problem for discrete-time linear systems. Additionally, a state parameterization method for reconstructing system states based on filtered input and output signals has been proposed. In contrast, static output feedback designs are popular due to their simplicity and have been used to solve the LQR problem for continuous-time linear systems [33]. However, obtaining static output feedback controllers requires not only the complete state variable information of systems during the learning phase but also model-free state estimation techniques based on neural networks [34,35]. An alternative approach, first proposed in [24], is to use the measurements of past inputs, outputs, and reference trajectories in a system as substitutes for the unmeasurable system state to learn the output feedback LQR controller. This approach has also been extended to solve the output feedback LQT problem by employing the VFA technique [25]. Furthermore, model-free state reconstruction techniques have recently been applied to solve output feedback Q-learning PI schemes for H_∞ control problems [36,37].

In this paper, we propose a tracking control approach that utilizes a static output feedback multistep Q-learning algorithm in conjunction with state reconstruction techniques. A separate adaptation mechanism was introduced in [38] to estimate unknown feedforward tracking terms. However, the static OPFB design we propose owes its popularity to its simplicity in terms of structure.

The key contributions of this work can be summarized as follows:

- Compared to the results reported in [23,39], the proposed approach does not apply an actor–critic structure, which are dependent on actor and critic NNs, to approximate control policy or value function. Moreover, the proposed model-free learning approach removes the requirement for the measurability of system state variables by collecting past input, output, and reference trajectory data. This is particularly advantageous in practical scenarios in which obtaining full state information may be challenging or costly;
- VFA-based learning [23,39] can ruin algorithm convergence due to the exploration noise that is intentionally added to evaluated policies to sufficiently excite systems. However, we apply the Q-learning scheme [40], which creates no biases in the estimated parameters of Q-function Bellman equations;
- Using the proposed multistep Q-learning technique [41], which surpasses the advantages of PI and VI methods, we are able to remove the requirement for an initial

stabilizing control strategy. Moreover, this combination improves the convergence speed of the algorithm, leading to more efficient control performance.

The rest of this paper is organized as follows: Section 1 formulates the problem statement, Section 2 presents the proposed methodology, Section 3 displays the simulation results, and finally, Section 4 concludes the paper with some discussion and future research directions.

2. Problem Statement

This section will first review the problem of infinite-horizon LQT for DTL systems. Then, we will present some fundamental results for solving a discrete-time Bellman equation.

Consider a time-invariant DTL system described by the following state and output equations:

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

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

The reference trajectory is generated by the exogenous system:

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

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

The tracking error is defined as follows:

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

The goal is to create an optimal control policy, u_k, that allows the output, y_k, to track the reference trajectory, r_k, in an optimal way. This is achieved 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 $0 < \gamma \leq 1$ represents the discount factor.

Remark 1. *As stated in [40], the discount factor γ in (4) allows for a more general solution to the LQT problem compared to the standard setting. Importantly, the matrix F need not be stable, thus permitting a broader range of permissible reference signals for the tracking control problem with the quadratic performance index. Additionally, this framework allows for simultaneous optimization of both feedback and feedforward components of the control input, leading to a causal solution to the infinite-horizon LQT problem. It is worth noting that the use of the discount factor γ does not sacrifice generality, as one can set $\gamma = 1$ when F is Hurwitz, reducing the LQT problem to an LQR problem with the specified output trajectory exponentially decaying to zero.*

2.1. Offline Solution for LQT

By denoting $X_k = \begin{bmatrix} x_k^T & r_k^T \end{bmatrix}^T$, we obtain the following 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 $C_1 = \begin{bmatrix} C & -I \end{bmatrix}$.

It can be shown by Lemma 1 of [40] that, with the choice of $u_k = -KX_k$, where $K = \begin{bmatrix} K_x & K_r \end{bmatrix}$, the discounted performance index (4) can be expressed in a quadratic form as follows:

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

where $P = P^T > 0$.

Using Formula (4), the cost function can be expressed as follows:

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

Using Equation (6), the cost function $J(x_k, r_k)$ can be rewritten as $V(x_k, r_k)$, which can be expressed as follows: H = r + Cv

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

Substituting Equation (6) into Equation (8) yields the LQT Bellman equation for 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 as

$$\frac{1}{2} H(X_k, u_k) = \frac{1}{2} X_k^T \Pi X_k + \frac{1}{2} u_k^T R u_k + \gamma V(X_{k+1}) - V(X_k) \tag{10}$$

By solving the stationary condition [40,42], i.e.,

$$\frac{\partial H(X_k, u_k)}{\partial u_k} = 0 \tag{11}$$

we can find the optimal control input

$$u_k = -KX_k = -K_x x_k - K_r r_k \tag{12}$$

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

$$\Pi - P + \gamma T^T P T - \gamma^2 T^T P B_1 (R + \gamma B_1^T P B_1)^{-1} B_1^T P = 0 \tag{13}$$

Remark 2. *The augmented ARE (13) has a unique, positive definite solution P if the pair* $(A, \sqrt{Q}C)$ *is observable and* $\gamma^{1/2} F$ *is stable [25]. Additionally, a lower bound has been established for the discount factor to ensure the stability of the augmented system [43].*

A direct solution to (13) is challenging due to the nonlinear relationship in the unknown parameter. Instead, we substitute (12) into (9) to obtain the augmented LQT Lyapunov equation:

$$\Pi - P + K^T R K + \gamma (T - B_1 K)^T P (T - B_1 K) = 0. \tag{14}$$

To address this issue, an offline PI algorithm [5] has been proposed as an iterative approach to compute the solution to (14). However, it requires complete knowledge of the augmented system dynamics. To overcome this limitation, a Q-learning scheme [40] was developed to solve the model-free LQT problem.

2.2. Q-Function Bellman Equation

Let $Z_k = \begin{bmatrix} X_k^T & u_k^T \end{bmatrix}^T$; then, the discrete-time Q-function can be defined as follows:

$$Q(Z_k) = \frac{1}{2} X_k^T \Pi X_k + \frac{1}{2} u_k^T R u_k + \gamma V(X_{k+1}) \tag{15}$$

By substituting the augmented system dynamics (5) into (15), we obtain:

$$Q(Z_k) = \frac{1}{2} Z_k^T \tilde{H} Z_k \tag{16}$$

where

$$\tilde{H} = \begin{bmatrix} \Pi + \gamma T^T P T & \gamma T^T P B_1 \\ \gamma B_1^T P T & R + \gamma B_1^T P B_1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{H}_{XX} & \tilde{H}_{Xu} \\ \tilde{H}_{uX} & \tilde{H}_{uu} \end{bmatrix} \tag{17}$$

and $\tilde{H}$ is a kernel matrix and $\tilde{H} = \tilde{H}^T$.

By applying $\frac{\partial Q(Z_k)}{\partial u_k} = 0$, we can solve for u_k as follows:

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

Furthermore, noticing that $Q(Z_k) = V(X_k)$ leads to the Q-function Bellman equation:

$$Z_k^T \tilde{H} Z_k = X_k^T \Pi X_k + u_k^T R u_k + \gamma Z_{k+1}^T \tilde{H} Z_{k+1} \tag{19}$$

This equation expresses the connection between the Q-function and the kernel matrix $\tilde{H}$.

2.3. PI-Based Q-Learning for LQT

Based on the Q-function Bellman equation (19), the PI-based Q-learning solution for the LQT problem can be implemented using Algorithm 1, without relying on the system dynamics [40].

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 samples under $u_k^j$ to solve $\tilde{H}^{j+1}$
using the Q-function Bellman equation:
$Z_k^T \tilde{H}^{j+1} Z_k = X_k^T \Pi X_k + (u_k^j)^T R (u_k^j) + \gamma Z_{k+1}^T \tilde{H}^{j+1} Z_{k+1}$
2: Policy Improvement: Compute the improved control policy as follows:
$u_k^{j+1} = -(\tilde{H}_{uu}^{j+1})^{-1} \tilde{H}_{uX}^{j+1} X_k$
3: Stopping Criterion: Stop the iteration if $\|\tilde{H}^{j+1} - \tilde{H}^j\| < \varepsilon$ for some specified
small positive number $\varepsilon$. Otherwise, let $j = j + 1$ and go back to iteration.
End Procedure
```

Algorithm 1 performs repeated iterations between policy evaluation and policy improvement until convergence. In contrast to the offline algorithm [40], Algorithm 1 conducts the policy improvement step using the learned kernel matrix $\tilde{H}^{j+1}$. This allows finding the optimal policy even under completely unknown dynamic conditions.

3. Methods

3.1. Multistep Q-Learning

Lemma 1. [24] *When the pair (A, C) of the DTL system (1) is observable, the state X_k of the augmented system can be reconstructed from the past input, output, and reference signal trajectories:*

$$X_k = \begin{bmatrix} M_u & M_y & M_r \end{bmatrix} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ r_{k-N} \end{bmatrix} \tag{20}$$

where $\bar{u}_{k-1,k-N} = [u_{k-1}^T, u_{k-2}^T, \cdots, u_{k-N}^T]^T$ and $\bar{y}_{k-1,k-N} = [y_{k-1}^T, y_{k-2}^T, \cdots, y_{k-N}^T]$, $N \le n$ are the sequences of input and output signals over the time interval $[k-N, k-1]$, respectively, and

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

$$U_N = \begin{bmatrix} B_1 & AB_1 & A^2B_1 & \cdots & A^{N-1}B_1 \end{bmatrix}$$

$$W_N = \begin{bmatrix} (CA^{N-1})^T & (CA^{N-2})^T & \cdots & CA & C \end{bmatrix}^T$$

$$D_N = \begin{bmatrix} 0 & CB & CAB & \cdots & CA^{N-2}B \\ 0 & 0 & CB & \cdots & CA^{N-3}B \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & CB \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$W_N^+ = \left(W_N^T W_N\right)^{-1} W_N^T$$

Lemma 1 states that the Q-function Bellman equation (19) can be transformed by using the past input, output, and reference trajectory sequences. By substituting Equation (20) into Equation (16), we obtain

$$\begin{aligned} Q(Z_k) = \tfrac{1}{2} Z_k^T \tilde{H} Z_k = \tfrac{1}{2} \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ r_{k-N} \\ u_k \end{bmatrix}^T H \begin{bmatrix} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ r_{k-N} \\ u_k \end{bmatrix} \\ \triangleq \tfrac{1}{2} z_k^T H z_k \end{aligned} \tag{21}$$

where

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

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

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

According to the principle of optimality, the optimal control policy should satisfy $\frac{\partial Q(z_k)}{\partial u_k} = 0$. Solving for u_k from Equation (21) yields the optimal control policy u_k^* as:

$$
\begin{aligned}
& u_k^* = -(H_{uu})^{-1}\left(H_{u\bar{u}}\bar{u}_{k-1,k-N} + H_{u\bar{y}}\bar{y}_{k-1,k-N} + H_{ur}r_{k-N}\right) \\
& = -(H_{uu})^{-1}\left[\begin{array}{ccc} H_{u\bar{u}} & H_{u\bar{y}} & H_{ur} \end{array}\right]\left[\begin{array}{c} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ r_{k-N} \end{array}\right] \\
& = -K^*\left[\begin{array}{c} \bar{u}_{k-1,k-N} \\ \bar{y}_{k-1,k-N} \\ r_{k-N} \end{array}\right]
\end{aligned} \tag{22}
$$

where $K^* = (H_{uu})^{-1}\left[\begin{array}{ccc} H_{u\bar{u}} & H_{u\bar{y}} & H_{ur} \end{array}\right]$.

By substituting Equation (21) into Equation (19) with the utility function $r(\tau_k, u_k) = \tau_k^T \Gamma \tau_k + u_k^T R u_k$, we have the Q-function Bellman equation incorporating input, output, and reference trajectory sequences, which is expressed as follows:

$$
z_k^T H z_k = r(\tau_k, u_k) + \gamma z_{k+1}^T H z_{k+1} \tag{23}
$$

where $\tau_k = \left[\begin{array}{c} y_k \\ \\ r_k \end{array}\right]$ and $\Gamma = \left[\begin{array}{cc} Q & -Q \\ -Q & Q \end{array}\right]$.

Define the optimal value function $V^*(z_k) \triangleq V_{u^*}(z_k)$. According to the optimal control theory [1], $V^*(z_k)$ satisfies the following Bellman equation:

$$
V^*(z_k) = \min_{u(z)}\{r(\tau_k, u_k) + \gamma V^*(z_{k+1})\} \tag{24}
$$

and the optimal control is

$$
u_k^* = arg\min_{u(z)}\{r(\tau_k, u_k) + \gamma V^*(z_{k+1})\} \tag{25}
$$

It is known that the policy evaluation step in the VI scheme is expressed as follows [23]:

$$
z_k^T H^{j+1} z_k = r\left(\tau_k, u_k^j\right) + \gamma z_{k+1}^T H^j z_{k+1} \tag{26}
$$

Transforming Equation (26) yields

$$\begin{aligned} z_k^T H^{j+1} z_k &= r\left(\tau_k, u_k^j\right) + \gamma z_{k+1}^T H^j z_{k+1} \\ &= r\left(\tau_k, u_k^j\right) + \gamma\left(r\left(\tau_{k+1}, u_{k+1}^j\right) + \gamma z_{k+2}^T H^j z_{k+2}\right) \\ &= r\left(\tau_k, u_k^j\right) + \gamma r\left(\tau_{k+1}, u_{k+1}^j\right) + \gamma^2\left(r\left(\tau_{k+2}, u_{k+2}^j\right) + \gamma z_{k+3}^T H^j z_{k+3}\right) \\ &\ \vdots \\ &= \sum_{i=k}^{k+N_j-1} \gamma^{i-k} r\left(\tau_i, u_i^j\right) + \gamma^{N_j} z_{k+N_j}^T H^j z_{k+N_j} \end{aligned} \tag{27}$$

Thus, the convergence of the VI method [23] can be accelerated by introducing a multistep utility function in the policy evaluation. The resulting multistep Q-learning VI algorithm based on output feedback is described as follows:

- **Step 1. Initialization:** Set $j = 0$ and iterate from any initial control policy u_k^0, which does not need to be stable or controllable, and H^0.
- **Step 2. Multistep policy evaluation:** Use the Q-function Bellman equation to solve H^{j+1}, where

$$z_k^T H^{j+1} z_k = \sum_{i=k}^{k+N_j-1} \gamma^{i-k} r\left(\tau_i, u_i^j\right) + \gamma^{N_j} z_{k+N_j}^T H^j z_{k+N_j} \tag{28}$$

- **Step 3. Policy improvement:** Update the control policy u_k^{j+1} as follows:

$$u_k^{j+1} = -\left(H_{uu}^{j+1}\right)^{-1}\left(H_{u\bar{u}}^{j+1}\bar{u}_{k-1,k-N} + H_{u\bar{y}}^{j+1}\bar{y}_{k-1,k-N} + H_{ur}^{j+1} r_{k-N}\right) \tag{29}$$

- **Step 4. Termination condition:** Check if $\left\|H^{j+1} - H^j\right\| \leq l$, where l is a very small threshold with pre-set algorithmic accuracy. If this condition is satisfied, terminate the iteration and obtain the optimal control policy u_k^{j+1}. Otherwise, return to Step 2 and repeat the iteration.

Remark 3. *As indicated in Equation (28), it is observed that the resultant value function in each policy evaluation step is the sum of the one-step utility function and the previous value function. When $N_j = 1$, Equation (28) simplifies to the policy evaluation step, which uses the VI framework for one-step learning [23,28]. The difference in policy evaluation leads to contrasting merits in value iteration. Unlike the traditional value iteration method, the proposed multistep Q-learning VI algorithm takes advantage of a finite-sum utility function instead of a one-step calculation as in value iteration. Consequently, the proposed algorithm leads to an improvement on the learning convergent speed, as demonstrated in the simulation example.*

3.2. Adjustment Rules for Step Size N_j

During each iteration of the multistep policy evaluation (28), the step size is adjusted. The value iteration algorithm [23] employs a one-step policy evaluation to eliminate the requirement for an initial stabilizing control policy. On the other hand, the policy iteration algorithm uses an infinite-step strategy evaluation to speed up convergence [39]. However, the speed of convergence of the multistep Q-learning algorithm depends on the chosen step size. Initially, a small step size is used in the iteration to avoid the need for an initial stabilizing control policy. Then, the step size is gradually increased to speed up convergence. To adaptively adjust the step length, we use the following rule [41]:

$$N_j = 1 + [\beta\sqrt{j}] \tag{30}$$

where $\beta \geq 0, [\cdot]$ means rounding down. When $\beta = 0$ and $N_j = 1$, it is equivalent to the VI method with one-step policy evaluation [23].

3.3. Implementation

By using the least squares method, the linear parametric expression of $z_k^T H^{j+1} z_k$ is given as follows:

$$z_k^T H^{j+1} z_k = \left(\bar{H}^{j+1}\right)^T \bar{z}(k) \tag{31}$$

where

$$\bar{H}^{j+1} = \mathrm{vec}\left(H^{j+1}\right) \in \mathbb{R}^{l(l+1))/2} \equiv \left[H_{1l}^{j+1}, 2H_{12}^{j+1}, \cdots, 2H_{1l}^{j+1}, H_{21}^{j+1}, \cdots, 2H_{2l}^{j+1}, \cdots, H_{ll}^{j+1}\right]^T$$

Here, H_{ik}^{j+1} represents the element in the i-th row and k-th column of matrix H^{j+1}, where $i, k = 1, 2, \cdots, l$, and $l = mN + pN + m$. The Kronecker product $\bar{z}_k = z_k \otimes z_k$ is defined as $\left[z_1^2, z_1 z_2, \cdots, z_1 z_l, z_2^2, z_2 z_3, \cdots, z_2 z_l, \cdots, z_l^2\right] \in \mathbb{R}^{l(l+1)/2}$.

Combining Equation (31) with Equation (28), it can be simplified as:

$$\left(\bar{H}^{j+1}\right)^T \bar{z}_k = \sum_{i=k}^{k+N_j-1} \gamma^{i-k} r\left(\tau_i, u_i^j\right) + \gamma^{N_j} \left(\bar{H}^j\right)^T \bar{z}_{k+N_j} \tag{32}$$

The symmetric matrix H^{j+1} has dimensions of $l \times l$, resulting in a total of $l(l+1)/2$ independent elements. Consequently, Equation (32) requires collecting at least $L \geq l(l+1)/2$ sets of $\bar{z}_k$ for solving.

The least squares expression for Equation (28) is

$$\bar{H}^{j+1} = \left\{\left\{\Phi^j\right\}^T \left\{\Phi^j\right\}\right\}^{-1} \left\{\Phi^j\right\}^T \left\{\Upsilon^j + \gamma^{N_j} \Psi^j \bar{H}^j\right\} \tag{33}$$

where

$$\Phi^j = \begin{bmatrix} \bar{z}_k \\ \bar{z}_{k+1} \\ \vdots \\ \bar{z}_{k+L-1} \end{bmatrix} \in \mathbb{R}^{L \times l(l+1)/2}$$

$$\Upsilon^j = \begin{bmatrix} \sum_{i=k}^{k+N_j-1} \gamma^{i-k} r\left(\tau_i, u_i^j\right) \\ \sum_{i=k}^{k+N_j} \gamma^{i-k} r\left(\tau_i, u_i^j\right) \\ \vdots \\ \sum_{i=k}^{k+N_j+L-1} \gamma^{i-k} r\left(\tau_i, u_i^j\right) \end{bmatrix} \in \mathbb{R}^{L \times 1}$$

$$\Psi^j = \begin{bmatrix} \bar{z}_{k+N_j} \\ \bar{z}_{k+N_j+1} \\ \vdots \\ \bar{z}_{k+N_j+L-1} \end{bmatrix} \in \mathbb{R}^{L \times l(l+1)/2)}$$

Remark 4. *When $k \leq N$, the input–output data $\bar{u}_{k-1,k-N}$ and $\bar{y}_{k-1,k-N}$ will be unavailable. To address this issue, the internal model principle can be utilized to collect the missing data. Additionally, the internal model principle allows for asymptotic tracking control in the presence of small variations in system parameters, resulting in data that contain more intrinsic information for learning the optimal control solution.*

Remark 5. *In Equation (33), the vector $\bar{H}^{j+1}$ represents the jth estimated value of H^{j+1} under the current control policy. By using the components of $\bar{H}^{j+1}$, we can infer the corresponding components of matrix H^{j+1} and use them, along with the policy update step (29), to solve for the next step's control policy. This updated policy can then be used to gather a new set of data for each iteration until obtaining an optimal control policy. To ensure uniqueness in solving Equation (33), a persistent excitation condition is proposed in the literature [20,21], where probing noise w_k is added to the control input, ensuring that Φ^j is full rank and $(\Phi^j)^T(\Phi^j)$ is invertible. However, using the VFA method may result in bias in finding an optimal solution [23,25]. On the other hand, the Q-learning approach does not produce bias during the parameter estimation process and, hence, does not lead to bias in finding an optimal solution.*

3.4. Convergence Analysis

In reference [41], a multistep Q-learning algorithm based on state feedback is proposed for solving the optimal output regulation problem of DTL systems, and the convergence of the proposed algorithm is derived. This paper investigates a multistep Q-learning algorithm based on output feedback to solve the optimal output tracking control problem of discrete-time linear systems. Unlike the optimal output regulation problem studied in reference [41], this paper introduces a discount factor γ into the performance index function of the optimal output tracking control problem, resulting in changes to the corresponding Bellman equation. The system state is reconstructed using input, output, and reference signals. Therefore, it is necessary to verify the convergence of the output feedback multistep Q-learning algorithm for solving the optimal output tracking control problem of DTL systems.

The convergence of the algorithm illustrated in Section 2.1 can be proven by first noticing that Equation (28) can be rewritten as follows:

$$Q^{j+1}(z_k) = \frac{1}{2}\sum_{i=k}^{k+N_j-1} \gamma^{i-k} r\left(\tau_i, u_i^j\right) + \gamma^{N_j} Q^j\left(z_{k+N_j}\right) \tag{34}$$

Here, we are ready to obtain the following theorem, which indicates that the policy matrix H^{j+1} converges to the optimal value H^*.

Theorem 1. *Let $\left\{Q^j(z_k)\right\}$, where$Q^j(z_k) = \frac{1}{2}z_k^T H^j z_k$, be the sequence generated by the multistep Q-learning algorithm. If $N_j \geq 1$ and*

$$Q^0(z_k) \geq \min_{u(z)}\left\{\frac{1}{2}r(\tau_k, u_k) + \gamma Q^0(z_{k+1})\right\} \tag{35}$$

holds, then

(i) *For any j,*

$$Q^{j+1}(z_k) \leq \min_{u(z)}\left\{\frac{1}{2}r(\tau_k, u_k) + \gamma Q^j(z_{k+1})\right\} \leq Q^j(z_k) \tag{36}$$

holds;

(ii) $\lim_{j\to\infty} Q^j(z_k) = Q^*(z_k)$, *where $Q^\infty(z_k)$ is the optimal solution to the Q-function Bellman equation.*

Proof. (i) We will use mathematical induction to prove the result (36). From Equations (34) and (35), we have

$$
\begin{aligned}
Q^1(z_k) &= \frac{1}{2}\sum_{i=k}^{k+N_0-1} \gamma^{i-k} r\left(\tau_i, u_i^0\right) + \gamma^{N_0} Q^0\left(z_{k+N_0}\right) \\
&= \frac{1}{2}\sum_{i=k}^{k+N_0-2} \gamma^{i-k} r\left(\tau_i, u_i^0\right) + \frac{1}{2}\gamma^{N_0-1} r\left(\tau_{k+N_0-1}, u_{k+N_0-1}^0\right) + \gamma^{N_0} Q^0\left(z_{k+N_0}\right) \\
&= \frac{1}{2}\sum_{i=k}^{k+N_0-2} \gamma^{i-k} r\left(\tau_i, u_i^0\right) + \gamma^{N_0-1} \min_{u(z)}\left\{\frac{1}{2} r\left(\tau_{k+N_0-1}, u_{k+N_0-1}\right) + \gamma Q^0\left(z_{k+N_0}\right)\right\} \\
&\leq \frac{1}{2}\sum_{i=k}^{k+N_0-2} \gamma^{i-k} r\left(\tau_i, u_i^0\right) + \gamma^{N_0-1} Q^0\left(z_{k+N_0-1}\right) \\
&\vdots \\
&\leq \frac{1}{2} r\left(\tau_k, u_k^0\right) + \gamma Q^0\left(z_{k+1}\right) \\
&= \min_{u(z)}\left\{\frac{1}{2} r(\tau_k, u_k) + \gamma Q^0(z_{k+1})\right\} \leq Q^0(z_k)
\end{aligned}
\tag{37}
$$

which means that Equation (36) holds for $j = 0$.

Next, assume that Equation (36) is satisfied for $j-1$, i.e.,

$$
Q^j(z_k) \leq \min_{u(z)}\left\{\frac{1}{2} r(\tau_k, u_k) + \gamma Q^{j-1}(z_{k+1})\right\} \leq Q^{j-1}(z_k)
$$

Then,

$$
\begin{aligned}
Q^j(z_k) &= \frac{1}{2}\sum_{i=k}^{k+N_{j-1}-1} \gamma^{i-k} r(\tau_i, u_i^{j-1}) + \gamma^{N_{j-1}} Q^{j-1}(z_{k+N_{j-1}}) \\
&\geq \frac{1}{2}\sum_{i=k}^{k+N_{j-1}-1} \gamma^{i-k} r(\tau_i, u_i^{j-1}) + \gamma^{N_{j-1}} \min_{u(z)}\left\{\frac{1}{2} r(\tau_{k+N_{j-1}}, u_{k+N_{j-1}}) + \gamma Q^{j-1}(z_{k+N_{j-1}+1})\right\} \\
&= \frac{1}{2} r(\tau_k, u_k^{j-1}) + \frac{1}{2}\sum_{i=k+1}^{k+N_{j-1}-1} \gamma^{i-k} r(\tau_i, u_i^{j-1}) + \gamma^{N_{j-1}} Q^{j-1}(z_{k+N_{j-1}+1}) \\
&= \frac{1}{2} r(\tau_k, u_k^{j-1}) + \gamma\left\{\sum_{i=k+1}^{k+N_{j-1}} \frac{1}{2}\gamma^{i-(k+1)} r(\tau_i, u_i^{j-1}) + \gamma^{N_{j-1}} Q^{j-1}(z_{k+N_{j-1}+1})\right\} \\
&= \frac{1}{2} r(\tau_k, u_k^{j-1}) + \gamma Q^j(z_{k+1}) \geq \min_{u(z)}\left\{\frac{1}{2} r(\tau_k, u_k) + \gamma Q^j z_{k+1}\right\}
\end{aligned}
\tag{38}
$$

Using Equations (34) and (38), we have

$$
\begin{aligned}
Q^{j+1}(z_k) &= \sum_{i=k}^{k+N_j-1} \frac{1}{2}\gamma^{i-k} r\left(\tau_i, u_i^j\right) + \gamma^{N_j} Q^j\left(z_{k+N_j}\right) \\
&= \sum_{i=k}^{k+N_j-2} \frac{1}{2}\gamma^{i-k} r\left(\tau_i, u_i^j\right) + \gamma^{N_j-1} \min_{u(z)}\left\{\frac{1}{2} r\left(\tau_{k+N_j-1}, u_{k+N_j-1}\right) + \gamma Q^j\left(z_{k+N_j}\right)\right\} \\
&\le \sum_{i=k}^{k+N_j-2} \frac{1}{2}\gamma^{i-k} r\left(\tau_i, u_i^j\right) + \gamma^{N_j-1} Q^j\left(z_{k+N_j-1}\right) \\
&\vdots \\
&\le \frac{1}{2} r\left(\tau_k, u_k^j\right) + \gamma Q^j(z_{k+1}) \\
&= \min_{u(z)}\left\{\frac{1}{2} r(\tau_k, u_k) + \gamma Q^j(z_{k+1})\right\}
\end{aligned}
\tag{39}
$$

Thus, Equation (36) holds for all j.

(ii) According to the conclusion (36), $\left\{Q^j(z_k)\right\}$ is a monotonically non-increasing sequence with a lower bound of $Q^j(z_k) \ge 0$. For a bounded monotone sequence, we can always have a limit denoted by $Q^\infty(z_k) \triangleq \lim_{j\to\infty} Q^j(z_k)$. Take the limit of Equation (36):

$$
Q^\infty(z_k) \le \min_{u(z)}\left\{\frac{1}{2} r(\tau_k, u_k) + \gamma Q^\infty(z_{k+1})\right\} \le Q^\infty(z_k) \tag{40}
$$

Here, we have

$$
Q^\infty(z_k) = \frac{1}{2} r(\tau_k, u_k) + \gamma Q^\infty(z_{k+1}) \tag{41}
$$

Notice that Equation (41) is the solution to the Q-function Bellman equation (23). As a result, $Q^\infty(z_k) = Q^*(z_k)$.

□

Remark 6. *It is worth noting the choice of the initial value function Q^0 in multistep Q-learning algorithms. According to Theorem 1, Q^0 must satisfy Condition (35). However, (35) is only a sufficient condition, not a necessary one. Therefore, in practical systems, Q^0 can be a positive definite function over a large range and can be chosen through trial and error.*

4. Simulation Experiment

4.1. Controlled Object

To validate the proposed algorithm, a simulation experiment was conducted on a single-phase voltage source uninterruptible power supply (UPS) inverter, an essential component of the smart grid. With the development of new energy technologies, it is important for the control engineer to design a controller that makes UPS provide efficient and stable sinusoidal output voltages with optimal performance even in the presence of unknown loads. The circuit diagram of the single-phase voltage source UPS inverter is illustrated in Figure 1.

The dynamic equations of the inverter can be expressed as follows:

$$
\begin{aligned}
C_f \tfrac{dv_o}{dt} &= i_L - i_o \\
L_f \tfrac{di_L}{dt} + r i_L &= u V_s - v_o
\end{aligned}
\tag{42}
$$

where L_f represents the filter inductance; C_f represents the filter capacitance; r represents the inductance resistance; i_L represents the filter inductance current; v_0 represents the output voltage of the inverter, which is the output voltage of the pulse width modulation (PWM) inverter bridge represented as uV_s; $i_o = \frac{v_o}{R_o}$ represents the output current; and R_o represents the resistance value of the power grid.

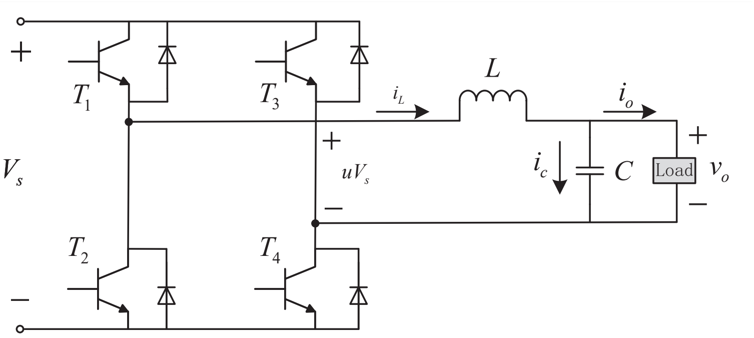

Figure 1. Circuit diagram of a single-phase voltage source UPS inverter.

Choosing v_0 and i_L as the system state variables, uV_s as the system input, and v_0 as the system output, we can obtain the state space representation of the single-phase voltage source UPS inverter as follows:

$$\begin{gathered}\dot{x} = \bar{A}x + \bar{B}u = \begin{bmatrix} -\frac{1}{R_o C_f} & \frac{1}{C_f} \\ -\frac{1}{L_f} & -\frac{r}{L_f} \end{bmatrix} x + \begin{bmatrix} 0 \\ \frac{1}{L_f} \end{bmatrix} u \\ y = \bar{C}x = \begin{bmatrix} 1 & 0 \end{bmatrix} x \end{gathered} \tag{43}$$

In the above equations, the inverter model parameters are as follows: $L_f = 3.56\ mH$, $C_f = 9.92\ \mu F$, $r = 0.4\ \Omega$, and $R_o = 50\ \Omega$. The initial values for the capacitor voltage and inductor current are set as $x_0 = [0, 0]^T$.

By discretizing Equation (43), the state space representation of the discrete-time system can be obtained as follows:

$$\begin{gathered} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{gathered} \tag{44}$$

where $A = e^{\bar{A}T}$, $B = \int_0^T e^{\bar{A}\tau} d\tau \bar{B}$, and $C = \bar{C}$. The sampling interval is $T = 10^{-4} s$. Substituting the inverter model parameters into the equations, we have:

$$A = \begin{bmatrix} 0.6969 & 806545 \\ -0.0241 & 0.8603 \end{bmatrix}, B = \begin{bmatrix} 0.1290 \\ 0.0267 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{45}$$

A sinusoidal signal is a typical type of reference signal in power electronics control. The state space representation of a continuous-time system with a sinusoidal signal of magnitude $220\sqrt{2}V$ and frequency $f = 50Hz$ is given by:

$$\begin{gathered} \dot{x}_d = \begin{bmatrix} 0 & 2\pi f \\ -2\pi f & 0 \end{bmatrix} x_d = \begin{bmatrix} 0 & 100\pi \\ -100\pi & 0 \end{bmatrix} x_d \\ r_d = \begin{bmatrix} 1 & 0 \end{bmatrix} x_d \end{gathered} \tag{46}$$

where the initial state $x_d(0) = [0, 1]^T$. The state space expression for the DTL system corresponding to Equation (46) can be described in the form shown in Equation (2), where

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

4.2. Experiment

We select $Q = 0.1, R = 0.001$, and $\gamma = 0.009$ as the parameter values. For each step, we set $\beta = 4$ for N_j. The detection noise w_k is defined as follows:

$$w_k = 0.001\big(7\sin(k) + 5\cos(2k) + 9\sin(8k) + 2\cos(6k)\big) \tag{47}$$

The controlled system used in the simulation experiment has 28 independent variables in $\bar{H}^{j+1}$, as shown in Equation (33). Therefore, a minimum of 28 sets of data are required for each iteration. In this simulation, we collected 30 sets of data. The following are the simulation results obtained using MATLAB.

The tracking curve of the output feedback multistep Q-learning algorithm is depicted in Figure 2. In this figure, r_d represents the sinusoidal reference signal, and y_o represents the actual output of the controlled system. After a certain number of iterations, the system output y_o successfully tracks the reference signal r_d. The corresponding tracking error curve is illustrated in Figure 3. From this figure, it can be observed that the tracking error reaches zero at 0.012 s, achieving the tracking goal. Figure 4 illustrates the norm of the difference between the Q-function matrices after two consecutive iterations in the output feedback multistep Q-learning algorithm. In the figure, it is shown that $\left|H^{j+1} - H^j\right| < 10^{-6}$ in the third iteration, indicating that the desired control accuracy has been achieved, and the algorithm has successfully converged. The simulation for the one-step learning case when $N_j = 1$ has also been conducted for clarity. It can be observed from Figure 5 that $\left|H^{j+1} - H^j\right| \approx 4.12 \times 10^{-5}$ for the third iteration, which is larger than the value of 9.01×10^{-7} in Figure 4. Therefore, the proposed multistep Q-learning scheme improves the learning convergence speed.

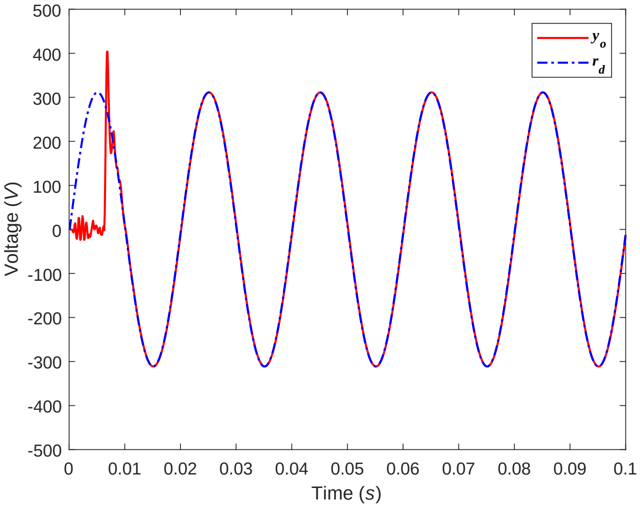

Figure 2. Effects of reference trajectory tracking with the multistep Q-learning algorithm.

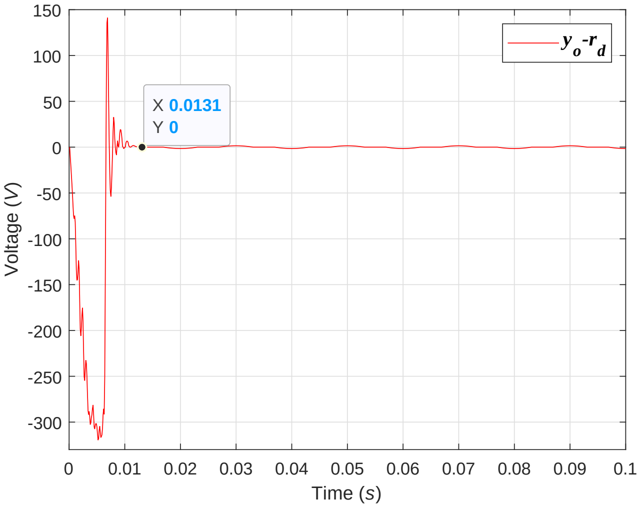

Figure 3. Tracking error curve.

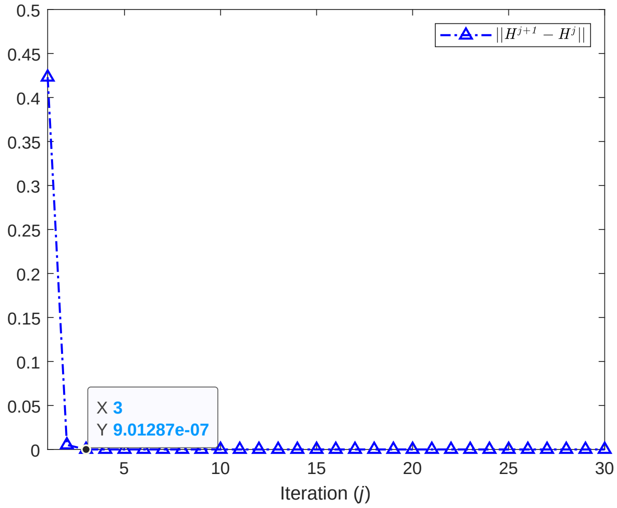

Figure 4. The norm of the difference between the Q-function matrices using multistep learning.

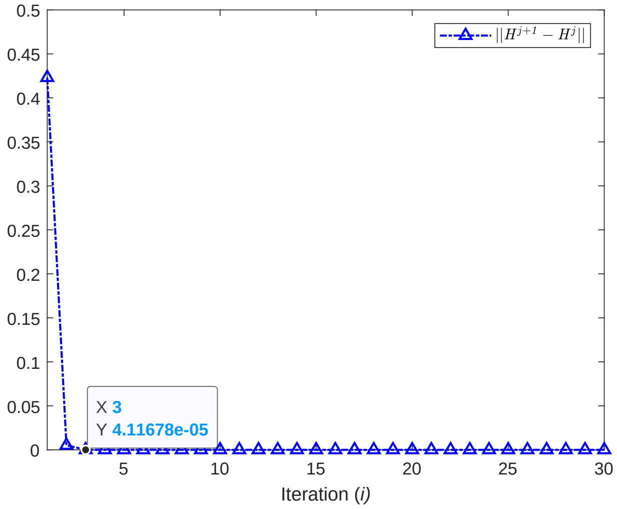

Figure 5. The norm of the difference between the Q-function matrices using one-step learning.

Figure 6 shows the variations in each component of the control gain K during each iteration process of the output feedback multistep Q-learning algorithm. Through iterations, it converges to

$$K^3 = \begin{bmatrix} 0.03769 & 0.02188 & 0.1879 & -0.1472 & -0.1163 & -0.01107 \end{bmatrix}$$

with

$$H^3 = \begin{bmatrix} 0.001758 & 0.001603 & 0.0056 & -0.01079 & -0.01316 & -0.0008375 & 3.563e-5 \\ 0.01603 & 0.001473 & 0.01897 & -0.009919 & -0.01215 & -0.00077 & 2.243e-5 \\ 0.02056 & 0.01897 & 0.2449 & -0.1277 & -0.1569 & -0.009916 & 0.0001926 \\ -0.01079 & -0.00919 & -0.1277 & 0.06677 & 0.08181 & 0.005183 & -0.0001508 \\ -0.01316 & -0.01215 & -0.1569 & 0.8181 & 0.1005 & 0.006352 & -0.001192 \\ -0.008375 & -0.0077 & -0.009916 & 0.05183 & 0.006352 & 0.004024 & -1.135e-5 \\ 3.863e-5 & 2.243e-5 & 0.001926 & -0.001508 & -0.001192 & -1.135e-5 & 0.001025 \end{bmatrix}$$

By solving the algebraic Riccati equation (13), the theoretical optimal values for the control gain K and the matrix H can be obtained as follows:

$$K^* = \begin{bmatrix} 0.0377 & 0.0219 & 0.1879 & -0.1472 & -0.1163 & -0.0111 \end{bmatrix}$$

and

$$H^* = \begin{bmatrix} 0.001758 & 0.001603 & 0.02056 & -0.01079 & -0.01316 & -0.0008375 & 3.828e-5 \\ 0.001603 & 0.001473 & 0.01897 & -0.009919 & -0.01215 & -0.00077 & 2.221e-5 \\ 0.02056 & 0.01897 & 0.2449 & -0.1277 & -0.1569 & -0.009916 & 0.0001909 \\ -0.01079 & -0.009919 & -0.1277 & 0.06677 & 0.08181 & 0.005183 & -0.0001495 \\ -0.01316 & -0.01215 & -0.1569 & 0.08181 & 0.1005 & 0.006352 & -0.0001181 \\ -0.0008375 & -0.00077 & -0.009916 & 0.005183 & 0.006352 & 0.0004024 & -1.125e-5 \\ 3.828e-5 & 2.221e-5 & 0.0001909 & -0.0001495 & -0.0001181 & -1.125e-5 & 0.001016 \end{bmatrix}$$

The Q function matrix H^3 obtained using the proposed learning algorithm is observed to be nearly equal to the theoretical optimal value H^*, indicating the effectiveness of the proposed output feedback multistep Q-learning algorithm for model-free tracking control.

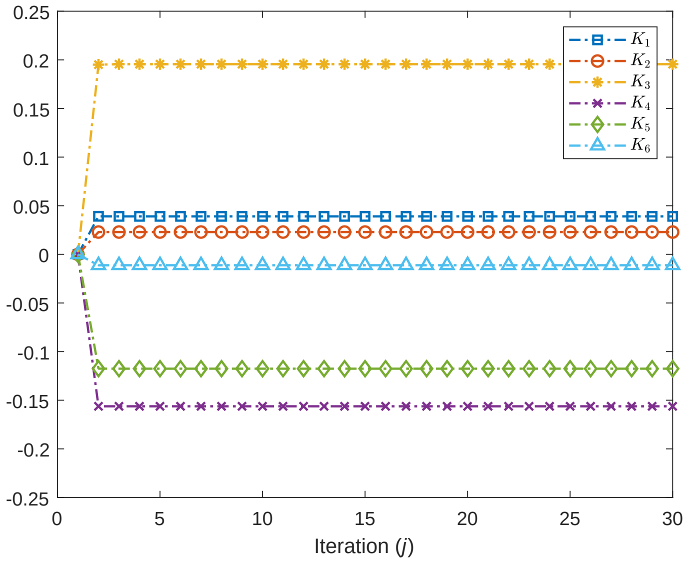

Figure 6. Convergence of control gain K with number of iterations.

Figure 7 shows the input signal trajectory of the actual tracking control system, which is a sinusoidal signal after the first four iterations. Figure 8 depicts the waveform of the excitation noise signal, which becomes zero at 0.0091 s, indicating the end of the algorithm learning phase without further noise excitation input. Figure 9 illustrates the tracking error of the system as the value of the system resistance parameter varies within the range of $40\ \Omega \leq R_o \leq 60\ \Omega$. With the change in resistance values, the system maintains its ability to achieve asymptotic tracking, demonstrating the adaptive characteristic of the algorithm. Figure 10 shows the variation in the step size N_j in the multistep Q-learning algorithm. It can be observed that N_j gradually increases as the iterations increase.

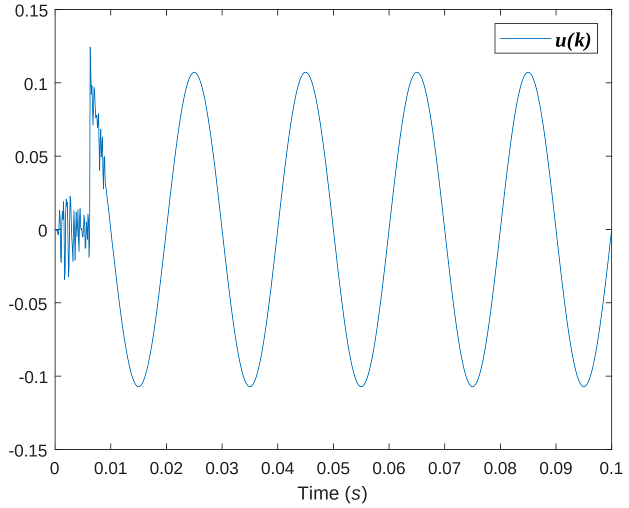

Figure 7. Input signal of the actual tracking control systems.

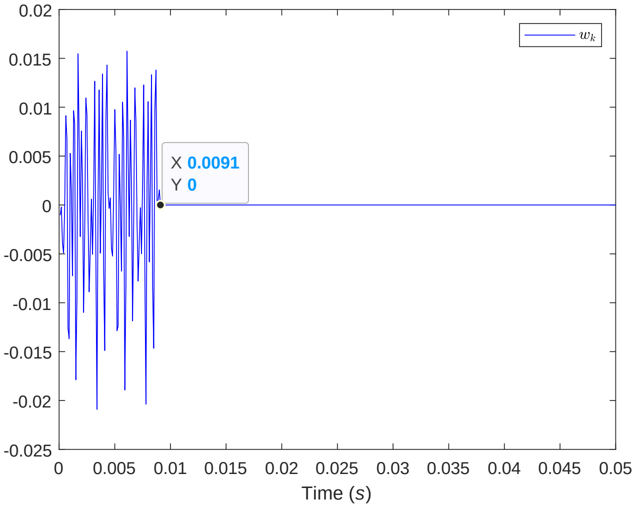

Figure 8. Excitation noise signal.

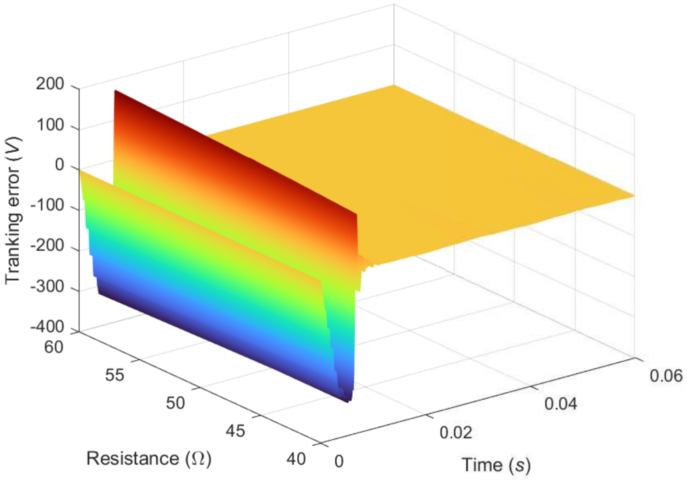

Figure 9. Tracking error under multistep Q-learning with different resistance values R_o.

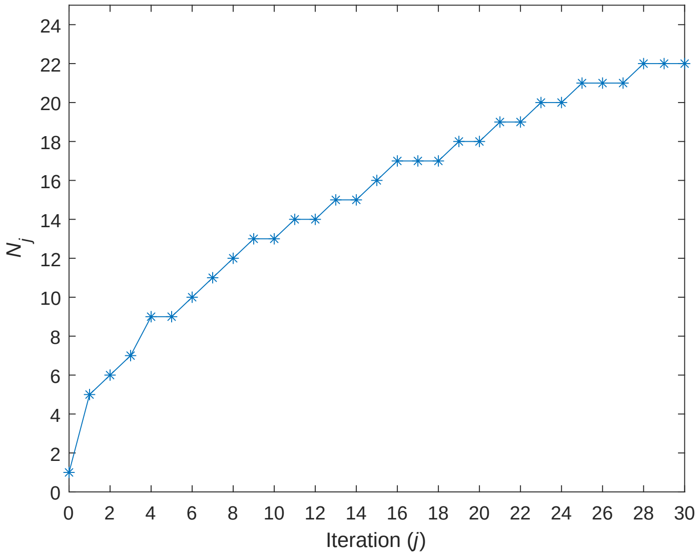

Figure 10. Variation of N_j with number of iterations.

5. Conclusions

In this paper, we investigate a value iteration (VI)-based multistep Q-learning algorithm for model-free optimal tracking controller design of unknown discrete-time linear (DTL) systems. By utilizing the augmented system approach, we transform this problem into a regulation problem with a discounted performance function, that depends on the Q-function Bellman equation. To solve the Bellman equation, we employ the VI learning mechanism and develop a multistep Q-learning algorithm that eliminates the need for an initial admissible policy and only requires measurements of past input, output, and reference trajectory data. As a result, our proposed approach offers a novel solution that does not require state measurements and has improved convergence learning speed. To validate the effectiveness of the proposed design, we demonstrate its application through a simulation example. Future work will involve extending the proposed multistep Q-learning scheme to unknown discrete-time systems with time delays and/or sampling errors. Additionally, it would be interesting to explore how to balance the computational demands of the algorithm with the available arithmetic power in practical experimental platforms.

Author Contributions: Conceptualization, W.S. and X.D.; methodology, W.S.; software, W.S. and X.S.; validation, X.D.; formal analysis, X.D.; investigation, X.D., Y.L. and X.W.; writing—original draft preparation, X.D.; writing—review and editing, W.S.; visualization, X.D., Y.L. and X.W.; supervision, W.S.; project administration, W.S.; funding acquisition, X.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant numbers 62003141 and U20A20224), the Natural Science Foundation of Guangdong Province, China (grant number 2021A1515011598), and the Fundamental Research Funds for the Central Universities (grant number 2022ZYGXZR023).

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lewis, F.L.; Vrabie, D.; Syrmos, V.L. *Optimal Control*, 3rd ed.; Wiley: Hoboken, NJ, USA, 2012.
2. Luo, R.; Peng, Z.; Hu, J. On model identification based optimal control and its applications to multi-agent learning and control. *Mathematics* **2023**, *11*, 906.
3. Chen, Y.H.; Chen, Y.Y. Trajectory tracking design for a swarm of autonomous mobile robots: a nonlinear adaptive optimal approach. *Mathematics* **2022**, *10*, 3901.

4. Banholzer, S.; Herty, M.; Pfenninger, S.; Zügner, S. Multiobjective model predictive control of a parabolic advection-diffusion-reaction equation. *Mathematics* **2020**, *8*, 777.
5. Hewer, G. An Iterative Technique for the Computation of the Steady State Gains for the Discrete Optimal Regulator. *IEEE Trans. Autom. Control* **1971**, *16*, 382–384.
6. Lancaster, P.; Rodman, L. *Algebraic Riccati Equations*; Oxford University Press: Oxford, UK, 1995.
7. Dai, S.; Wang, C.; Wang, M. Dynamic Learning From Adaptive Neural Network Control of a Class of Nonaffine Nonlinear Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2014**, *25*, 111–123.
8. He, W.; Dong, Y.; Sun, C. Adaptive Neural Impedance Control of a Robotic Manipulator With Input Saturation. *IEEE Trans. Syst. Man, Cybern. Syst.* **2016**, *46*, 334–344.
9. Luy, N.T. Robust adaptive dynamic programming based online tracking control algorithm for real wheeled mobile robot with omni-directional vision system. *Trans. Inst. Meas. Control.* **2017**, *39*, 832–847.
10. He, W.; Meng, T.; He, X.; Ge, S.S. Unified iterative learning control for flexible structures with input constraints. *Automatica* **2018**, *96*, 326–336.
11. Radac, M.B.; Precup, R.E. Data-Driven model-free tracking reinforcement learning control with VRFT-based adaptive actor-critic. *Appl. Sci.* **2019**, *9*, 1807.
12. Wang, Z.; Liu, D. Data-Based Controllability and Observability Analysis of Linear Discrete-Time Systems. *IEEE Trans. Neural Netw.* **2011**, *22*, 2388–2392.
13. Sutton, R.S.; Barto, A.G. *Reinforcement Learning*; MIT Press: Cambridge, MA, USA, 1998.
14. Sutton, R.S.; Barto, A.G.; Williams, R.J. Reinforcement learning is direct adaptive optimal control. *IEEE Control Syst. Mag.* **1992**, *12*, 19–22.
15. Lewis, F.L.; Vrabie, D. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits Syst. Mag.* **2009**, *9*, 32–50.
16. Wang, F.; Zhang, H.; Liu, D. Adaptive Dynamic Programming: An Introduction. *IEEE Comput. Intell. Mag.* **2009**, *4*, 39–47.
17. Jiang, Z.P.; Jiang, Y. Robust adaptive dynamic programming for linear and nonlinear systems: An overview. *Eur. J. Control* **2013**, *19*, 417–425.
18. Zhang, K.; Zhang, H.; Cai, Y.; Su, R. Parallel Optimal Tracking Control Schemes for Mode-Dependent Control of Coupled Markov Jump Systems via Integral RL Method. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1332–1342.
19. Zhang, K.; Zhang, H.; Mu, Y.; Liu, C. Decentralized Tracking Optimization Control for Partially Unknown Fuzzy Interconnected Systems via Reinforcement Learning Method. *IEEE Trans. Fuzzy Syst.* **2020**, *29*, 917–926.
20. Vrabie, D.; Pastravanu, O.; Abou-Khalaf, M.; Lewis, F.L. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica* **2009**, *45*, 477–484.
21. Jiang, Y.; Jiang, Z.P. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica* **2012**, *48*, 2699–2704.
22. Modares, H.; Lewis, F.L. Linear Quadratic Tracking Control of Partially-Unknown Continuous-Time Systems Using Reinforcement Learning. *IEEE Trans. Autom. Control.* **2014**, *59*, 3051–3056.
23. Li, X.; Xue, L.; Sun, C. Linear quadratic tracking control of unknown discrete-time systems using value iteration algorithm. *Neurocomputing* **2018**, *314*, 86–93.
24. Lewis, F.L.; Vamvoudakis, K.G. Reinforcement Learning for Partially Observable Dynamic Processes: Adaptive Dynamic Programming Using Measured Output Data. *IEEE Trans. Syst. Man, Cybern. Part B (Cybern.)* **2011**, *41*, 14–25.
25. Kiumarsi, B.; Lewis, F.L.; Naghibi-Sistani, M.B.; Karimpour, A. Optimal Tracking Control of Unknown Discrete-Time Linear Systems Using Input-Output Measured Data. *IEEE Trans. Cybern.* **2015**, *45*, 2770–2779.
26. Gao, W.; Huang, M.; Jiang, Z.; Chai, T. Sampled-data-based adaptive optimal output-feedback control of a 2-degree-of-freedom helicopter. *IET Control. Theory Appl.* **2016**, *10*, 1440–1447.
27. Xiao, G.; Zhang, H.; Zhang, K.; Wen, Y. Value iteration based integral reinforcement learning approach for H_∞ controller design of continuous-time nonlinear systems. *Neurocomputing* **2018**, *285*, 51–59.
28. Chen, C.; Sun, W.; Zhao, G.; Peng, Y. Reinforcement Q-Learning Incorporated With Internal Model Method for Output Feedback Tracking Control of Unknown Linear Systems. *IEEE Access* **2020**, *8*, 134456–134467. https://doi.org/10.1109/ACCESS.2020.3011194.
29. Zhao, F.; Gao, W.; Liu, T.; Jiang, Z.P. Adaptive optimal output regulation of linear discrete-time systems based on event-triggered output-feedback. *Automatica* **2022**, *137*, 110103.
30. Radac, M.B.; Lala, T. Learning Output Reference Model Tracking for Higher-Order Nonlinear Systems with Unknown Dynamics. *Algorithms* **2019**, *12*, 121.
31. Shi, P.; Shen, Q.K. Observer-based leader-following consensus of uncertain nonlinear multi-agent systems. *Int. J. Robust Nonlinear Control.* **2017**, *27*, 3794–3811.
32. Rizvi, S.A.A.; Lin, Z. Output Feedback Q-Learning Control for the Discrete-Time Linear Quadratic Regulator Problem. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 1523–1536.
33. Zhu, L.M.; Modares, H.; Peen, G.O.; Lewis, F.L.; Yue, B. Adaptive suboptimal output-feedback control for linear systems using integral reinforcement learning. *IEEE Trans. Control. Syst. Technol.* **2015**, *23*, 264–273.
34. Moghadam, R.; Lewis, F.L. Output-feedback H_∞ quadratic tracking control of linear systems using reinforcement learning. *Int. J. Adapt. Control. Signal Process.* **2019**, *33*, 300–314.

35. Valadbeigi, A.P.; Sedigh, A.K.; Lewis, F.L. H_∞ Static Output-Feedback Control Design for Discrete-Time Systems Using Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 396–406.
36. Rizvi, S.A.A.; Lin, Z. Output feedback Q-learning for discrete-time linear zero-sum games with application to the H-infinity control. *Automatica* **2018**, *95*, 213–221.
37. Peng, Y.; Chen, Q.; Sun, W. Reinforcement Q-Learning Algorithm for H_∞ Tracking Control of Unknown Discrete-Time Linear Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *50*, 4109–4122. https://doi.org/10.1109/TSMC.2019.2957000.
38. Rizvi, S.A.A.; Lin, Z. Experience replay-based output feedback Q-learning scheme for optimal output tracking control of discrete-time linear systems. *Int. J. Adapt. Control. Signal Process.* **2019**, *33*, 1825–1842.
39. Luo, B.; Liu, D.; Huang, T.; Liu, J. Output Tracking Control Based on Adaptive Dynamic Programming With Multistep Policy Evaluation. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2155–2165. https://doi.org/10.1109/TSMC.2017.2771516.
40. Kiumarsi, B.; Lewis, F.L.; Modares, H.; Karimpour, A.; Naghibi-Sistani, M.B. Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics. *Automatica* **2014**, *50*, 1167–1175.
41. Luo, B.; Wu, H.N.; Huang, T. Optimal output regulation for model-free quanser helicopter with multistep Q-learning. *IEEE Trans. Ind. Electron.* **2017**, *65*, 4953–4961.
42. Lewis, F.L.; Vrabie, D.; Vamvoudakis, K.G. Reinforcement Learning and Feedback Control: Using Natural Decision Methods to Design Optimal Adaptive Controllers. *IEEE Control. Syst. Mag.* **2012**, *32*, 76–105.
43. Kiumarsi, B.; Lewis, F.L. Output synchronization of heterogeneous discrete-time systems: A model-free optimal approach. *Automatica* **2017**, *84*, 86–94.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.