


## Article

# Reinforcement Q-Learning for PDF Tracking Control of Stochastic Systems with Unknown Dynamics

Weiqing Yang <sup>1,2</sup>, Yuyang Zhou <sup>3,\*</sup>, Yong Zhang <sup>1,2,\*</sup>  and Yan Ren <sup>1,2</sup>

<sup>1</sup> School of Automation and Electrical Engineering, Inner Mongolia University of Science and Technology, Baotou 014010, China; yangweiqing1998@stu.imust.edu.cn (W.Y.); ren0831@imust.edu.cn (Y.R.)

<sup>2</sup> Key Laboratory of Synthetical Automation for Process Industries at Universities of Inner Mongolia Autonomous Region, Inner Mongolia University of Science and Technology, Baotou 014010, China

<sup>3</sup> School of Computing Engineering and Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, UK

\* Correspondence: y.zhou@napier.ac.uk (Y.Z.); zhangyong@imust.edu.cn (Y.Z.)

**Abstract:** Tracking control of the output probability density function presents significant challenges, particularly when dealing with unknown system models and multiplicative noise disturbances. To address these challenges, this paper introduces a novel tracking control algorithm based on reinforcement Q-learning. Initially, a B-spline model is employed to represent the original system, thereby transforming the control problem into a state weight tracking issue within the B-spline stochastic system model. Moreover, to tackle the challenge of unknown stochastic system dynamics and the presence of multiplicative noise, a model-free reinforcement Q-learning algorithm is employed to solve the control problem. Finally, the proposed algorithm's effectiveness is validated through comprehensive simulation examples.

**Keywords:** tracking control; probability density function; reinforcement learning; B-spline model; Q-learning; model-free

**MSC:** 93E35



**Citation:** Yang, W.; Zhou, Y.; Zhang, Y.; Ren, Y. Reinforcement Q-Learning for PDF Tracking Control of Stochastic Systems with Unknown Dynamics. *Mathematics* **2024**, *12*, 2499. <https://doi.org/10.3390/math12162499>

Academic Editor: Duarte Valério

Received: 8 July 2024

Revised: 6 August 2024

Accepted: 9 August 2024

Published: 13 August 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The surge in interest in stochastic control stems from its applicability to diverse real-world systems, including aerospace, chemical, textile, and maritime machinery [1,2]. Gaussian processes in control can be managed through mean and variance manipulation [3], but non-Gaussian processes, such as particle size distribution in paper processing [4], flame shape dynamics [5], and chemical polymerization molecular weight distribution [6], necessitate a more comprehensive approach. Wang's 1996 innovation introduced probability density function (PDF) control [7–9], utilizing B-spline functions to bypass the Gaussian assumption limitations [10]. This has led to a range of stochastic control frameworks in both theoretical and practical realms, particularly in target tracking, where systems often aim to track distributions rather than values [11]. In detail, this method applies a B-spline model to represent the system's dynamic probability density function (PDF) by mapping the weights of the B-spline function to a dynamic state-space model. This approach shifts the focus from controlling the shape of the PDF to aligning the weights with predefined values. Under this framework, numerous significant papers have been published. For example, Luan's output PDF control tackles static tracking [12]. However, the controller design in most of the papers within this framework heavily relies on precise knowledge of the PDF model. Specifically, it requires the state-space model, derived from the PDF via the B-spline network, to be accurate for effective controller design. Meeting this requirement is challenging in many real-world industrial systems, limiting the practical application and development of this framework.

Multiplicative noise, which poses significant challenges due to its signal-dependent nature and non-linear characteristics, appears in various applications. However, a great number of studies, including those utilizing the algorithms proposed under Wang's framework, do not account for the influence of multiplicative noise in their system modeling. Although Yang's fully probabilistic design under Wang's framework addresses this issue by incorporating multiplicative noise into the model [13], it necessitates precise knowledge of the system's model parameters. Consequently, there is still a gap between this method and practical real-world applications.

Another important aspect often neglected in this framework is the consideration of time-varying targets. In many industrial contexts, time-varying targets are essential and present additional challenges for controller design. However, most studies within this framework tend to overlook this factor, which further complicates the practical application and effectiveness of the framework in real-world scenarios [9].

To address these barriers, this paper employs Wang's B-spline framework to develop a model-free control approach that enables dynamic weights to adapt to a time-varying target pattern. In our design, we utilize Q-learning-based Linear Quadratic Tracking Control (LQT) as a controller to achieve the tracking goal. Traditional LQT often struggle with the complexities of real-world systems, highlighting the necessity for model-free control strategies. Thus, reinforcement learning, particularly strategy iteration, is employed here to handle control and scheduling tasks without requiring complete model information [14]. As a model-free technique, reinforcement learning optimizes control by learning under given constraints [11,15]. The concept of L-Extra-Sampled (Les)-dynamics was introduced in [16], providing a new perspective for addressing reinforcement learning problems in partially observable linear Gaussian systems. However, the research in this paper primarily focuses on linear Gaussian systems, which may limit the method's applicability to non-Gaussian systems. In [17], an algorithm based on Q-learning was proposed to handle time-varying linear discrete-time systems with complete dynamic uncertainties. Although the algorithm is model-independent, it may require more computational resources to compute the Q-function and update the policy when dealing with high-dimensional systems. Ref. [18] introduced extreme value theory into reinforcement learning, proposing a new online and offline maximum entropy reinforcement learning update rule [19]. This rule avoids the difficulty of estimating the maximum Q-value in continuous action space. In [20], an efficient offline Q-learning method was proposed to solve the data-driven discrete-time linear quadratic regulator problem. This approach does not require knowledge of the system dynamic model and demonstrates advantages over existing methods in simulations. In [21], a Q-learning based iterative learning control (ILC) framework for fault estimation (FE) and fault-tolerant control (FTC) was proposed to address the actuator fault problem in multiple-input multiple-output (MIMO) systems. However, traditional reinforcement learning methods are inefficient, consuming significant time and requiring continuous adjustment. The efficiency of reinforcement learning can be enhanced by integrating it with optimal control techniques. For instance, in [22], the successful application of Q-learning to handle discrete systems with uncertain parameters was demonstrated, improving tracking control performance. Similarly, Xue's extension to two-time-scale systems showed favorable outcomes [23]. However, these findings are based on deterministic models and neglect the system randomness and uncertainties. By solving the Riccati equation for quadratic optimal control of linear stochastic systems with unknown parameters, we improve learning efficiency and achieve model-free tracking control. This approach allows for effective control even in the presence of random disturbances, addressing the limitations of traditional methods and enhancing the practical applicability of our framework.

In summary, this paper utilizes Wang's B-spline framework to develop a model-free control approach, enabling dynamic weights to adapt to a target time-varying pattern. In the case of an unknown stochastic system model, we can change the output PDF shape by controlling the weights. This approach expands the scope of application for stochastic systems in output PDF control. Output PDF control is employed to monitor and manage

the distribution of key parameters during the production process, which helps enhance product quality and decrease scrap rates. For instance, in the semiconductor manufacturing industry, controlling the distribution of parameters such as etching depth and doping concentration during wafer fabrication ensures consistent chip performance. Similarly, in the automobile manufacturing industry, controlling the distribution of coating thickness during the painting process improves surface quality and anti-corrosion performance. By leveraging the B-spline basis functions and the model-free reinforcement Q-learning algorithm, we can effectively handle unknown system dynamics and multiplicative noise, achieving precise control over the output PDF shape. This method facilitates accurate PDF shape tracking and establishes a theoretical foundation for PDF monitoring in non-Gaussian stochastic systems, making it highly applicable to real-world industrial contexts. The contributions of this paper can be summarized as follows:

- 1 By integrating the reinforcement learning method with the LQR control algorithm, the control framework proposed in this paper effectively addresses the challenge of PDF tracking control in stochastic systems with unknown parameters, eliminating the need for precise knowledge of the system's model parameters.
- 2 By utilizing B-spline functions to approximate the PDF, our method converts the PDF tracking problem into a state tracking problem with dynamic weights.
- 3 The multiplicative noise is being considered while modeling the PDF under the B-spline framework, reflecting a more accurate representation of complex and realistic uncertainties.
- 4 The consideration of time-varying PDF target, which are crucial in many real-world applications but often overlooked in previous studies, enhances the practical applicability of the framework.
- 5 The framework combines optimal control principles with reinforcement learning, specifically Q-learning, to significantly enhance the performance and accelerate the learning speed of RL algorithms.

The remainder of this paper is organized as follows: Section 2 provides a detailed description of the problem and the B-spline model of the PDF. In Section 3, the optimal control law is derived based on the performance metrics, and the implementation algorithm is presented. Section 4 demonstrates the application of the controller through two numerical examples. Finally, Section 5 summarizes the conclusions and outlines potential future work.

## 2. Problem Description

### 2.1. PDF Description Based on B-Spline

Modeling the output PDF of a controlled system by solving partial differential equations can be challenging when using first principles, complicating the development of an effective control strategy [24]. To overcome this, the B-spline approach can be employed to approximate the PDF curve by mapping weights to basis functions. B-spline basis functions are a flexible and widely used class of functions that can adapt to various interpolation and fitting requirements. They can be classified by order: first-order, second-order, third-order and fourth-order B-spline basis functions. Among these, third-order B-spline basis functions are the most commonly used for cubic polynomial functions because they strike a good balance between smoothness and computational complexity. Specifically, given a known interval  $[a, b]$ , where the output PDF  $\gamma(y)$  is continuous and bounded, the PDF can be expressed using  $n$  B-spline basis functions as follows:

$$\gamma(y) = \sum_{i=1}^n w_i B_i(y), \quad (1)$$

where  $w_i$  (with  $i = 1, 2, \dots, n$ ) represents the weight and  $B_i(y)$  represents pre-selected  $n$  basis functions, which can include Gaussian, radial basis, or wavelet functions. Given

that  $\gamma(y)$  is a PDF defined on the interval  $[a, b]$ , it is subject to the following mathematical constraints:

$$\int_a^b \gamma(y) dy = 1. \quad (2)$$

To satisfy Equation (2), only  $n - 1$  weights are independent, which allows us to express the distribution in the following form:

$$\gamma(y) = \mathbf{C}_0(y)\mathbf{x} + L(y), \quad (3)$$

$$\mathbf{x} = [w_1, w_2, \dots, w_{n-1}]^T \in \mathbb{R}^{n-1 \times 1}, \quad (4)$$

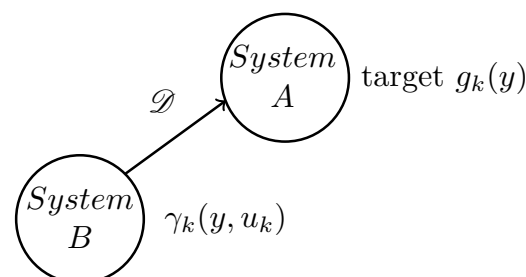
$$L(y) = \left( \int_a^b B_n(y) dy \right)^{-1} B_n(y) \in \mathbb{R}^{1 \times 1}, \quad (5)$$

$$\mathbf{C}_0(y) = \begin{bmatrix} B_1(y) - \frac{B_n(y)}{\int_a^b B_n(y) dy} \int_a^b B_1(y) dy \\ B_2(y) - \frac{B_n(y)}{\int_a^b B_n(y) dy} \int_a^b B_2(y) dy \\ \vdots \\ B_{n-1}(y) - \frac{B_n(y)}{\int_a^b B_n(y) dy} \int_a^b B_{n-1}(y) dy \end{bmatrix}^T \in \mathbb{R}^{1 \times n-1}, \quad (6)$$

where  $\mathbf{x}$  denotes the weight set,  $\mathbf{C}_0(y)$  represents the vector of basis functions, and  $L(y)$  is a scalar associated with the basis functions. Based on Equations (5) and (6), we can see that the choice of basis functions determines  $\mathbf{C}_0(y)$  and  $L(y)$ . From Equation (1) to Equation (6), it is evident that the B-spline model enables the control of the output PDF shape by manipulating  $n - 1$  independent weight vectors [25].

## 2.2. PDF Tracking Control Problem

The tracking problem is a prevalent issue in the field of control, including in stochastic control systems. In conventional control fields, the objective often involves directing the system to follow a predefined value. Conversely, in stochastic control systems, the goal shifts to having the system track a predetermined probability density function (PDF). Figure 1 illustrates the tracking diagram where system  $B$  tracks system  $A$ . System  $A$ , despite its unknown structure, can monitor its output in real time, allowing access to the output PDF distribution  $g_k(y)$  at any given time  $k$ . On the other hand, system  $B$  is a controlled system with established dynamics and employs a control input  $u$  to produce its output  $\gamma_k(y, u_k)$ . The objective is to align the output distribution of system  $B$  with that of system  $A$ , with  $\mathcal{D}$  quantifying the disparity between the two distributions. The details of the system and tracking control issues outlined above are further elaborated in the subsequent section.



**Figure 1.** Diagram of the tracking system.

Consider the stochastic system with output PDF  $\gamma_k(y)$ , whose dynamics is formed as follows:

$$\gamma_{k+1}(y) = f(\gamma_k(y), u_k), \quad (7)$$

where the distribution of the system output  $y$  is denoted by  $\gamma_k(y)$  and  $u_k$  represents the system input.

Based on the B-spline model (3), denoting  $x_k$  as the weights corresponding to the basis functions, the output PDF  $\gamma_k(y)$  of the tracking system B can be represented as:

$$\gamma_k(y) = C_0(y)x_k + L(y). \quad (8)$$

The tracking target  $g_k(y)$  is a dynamic PDF with the following manner:

$$g_k(y) = C_0(y)V_k + L(y), \quad (9)$$

where  $V_k$  in Equation (9) represent the pre-set target weights corresponding to each basis function.

Subsequently, the shaping of the output PDF  $\gamma(y, u_k)$  over the interval  $[a, b]$  can be achieved by controlling the weight state  $x_k$ . Within the framework of the B-spline model outlined in [8], the dynamics of the weight states  $x_k$  for the B-spline model-based PDF are described as follows:

$$x_{k+1} = Gx_k + Hu_k + Dx_kE_k, \quad (10)$$

where  $G \in \mathbb{R}^{n-1 \times n-1}$  and  $H \in \mathbb{R}^{n-1 \times 1}$  are the corresponding weight system parameters and  $E_k$  represents the state-based model randomness, whose distribution is given by:

$$E_k \sim (0, M), \quad (11)$$

where  $M$  is the variance of  $E_k$ .

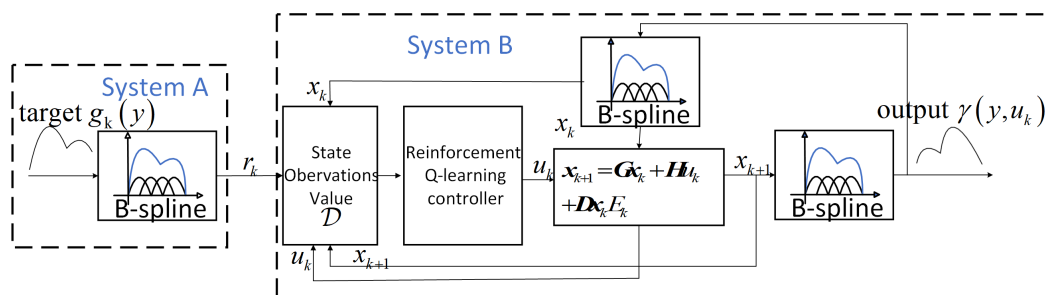
From Equations (7) to (10), we observe that the system's PDF is dynamic, leading to the derivation of a state-space model for the weights to represent the PDF's dynamics. However, obtaining precise model parameters,  $G$ ,  $H$ , and  $D$ , is challenging. Existing control strategies often rely on precise model parameters, necessitating that  $G$ ,  $H$ , and  $D$  in Equation (10) are known accurately. This assumption is difficult to meet in many real-world industrial processes.

To address this limitation, this study explores the application of data-driven methods, specifically reinforcement learning within machine learning, to mitigate the dependence on model parameters. Reinforcement learning, characterized by its objective to maximize reward through iterative optimization, offers a promising solution to the optimal control problem [26]. By employing Q-learning, one of a reinforcement learning algorithm, the need for explicit model knowledge is alleviated, as it enables the determination of optimal control strategies based solely on system operational data. This approach enables the controlled variable to effectively track the desired trajectory without requiring knowledge of the model parameters.

The control flowchart is depicted in Figure 2. After selecting the B-spline basis functions, the time-varying target weight  $r_k$  in Figure 2 is determined based on the target distribution using the B-spline principle. The system input  $u_k$  is derived by assessing both the target and system weights using a reinforcement Q-learning control, which will be detailed in the subsequent section. The weight  $x_{k+1}$  is then updated through B-spline principle modeling and the input  $u_k$ . The output distribution is obtained by correlating the weight with the basis function. It is important to note that the model error component  $E_k$  is characterized by multiplicative noise, and  $D$  represents the appropriately dimensioned weight matrix. The weights are iteratively updated according to the model to control the output distribution.

There are many methods to address multiplicative noise, and it is necessary to choose the appropriate method for different application scenarios to effectively reduce the noise's impact. Techniques such as logarithmic transformation, adaptive filtering, wavelet transformation, statistical methods, and specially designed filters can significantly improve the quality and reliability of signal processing. However, within a reinforcement learning

framework, multiplicative noise can be directly learned, thus eliminating the need for these additional processing methods.



**Figure 2.** System control structure diagram.

Building on this framework, the control objective for such stochastic system is to design a state feedback controller that enables the weight  $x_k$  to track the target weight  $r_k$ . Consequently, the PDF shape tracking control problem is transformed into a weight tracking control problem. The details of the controller design will be presented in the next section.

### 3. Control Algorithm

In this section, we introduce the reinforcement Q-learning control algorithm to achieve the tracking objective of the weight state. The primary motivation for choosing reinforcement learning control is to attain optimal tracking control in the presence of unknown model parameters. Additionally, the reinforcement Q-learning method is employed to establish an optimal control iterative solution structure, which proves to be more efficient than traditional reinforcement learning algorithms. The specific details are elaborated below.

#### 3.1. General Control Solution of LQT for Systems with Multiplicative Noise

In this section, we consider the infinite-horizon linear quadratic tracking problem with multiplicative noise. This developed algorithm will then be utilized as a foundation to create our model-free Q-learning control method.

Denoting the target weight for the LQT problem which are generated by the B-spline model based on the expected PDF output is  $r_k$ , based on the system dynamics Equation (10), we construct the augmented system:

$$X_{k+1} = \begin{bmatrix} x_{k+1} \\ r_{k+1} \end{bmatrix} = \begin{bmatrix} G & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix} \begin{bmatrix} x_k \\ r_k \end{bmatrix} + \begin{bmatrix} H \\ \mathbf{0} \end{bmatrix} u_k + \begin{bmatrix} D & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_k \\ r_k \end{bmatrix} E_k \equiv \bar{A}X_k + \bar{B}u_k + \bar{D}X_k E_k, \quad (12)$$

where the augmented state  $X_k$  is given by:

$$X_k = \begin{bmatrix} x_k \\ r_k \end{bmatrix}. \quad (13)$$

For the regular infinite-horizon LQT problem, the objective is to design an optimal controller for the system in Equation (10), ensuring that the weight  $x_k$  tracks a reference trajectory  $r_k$ . This can be achieved by minimizing the following infinite-horizon performance index:

$$J = \mathbb{E} \left\{ \sum_{i=0}^{\infty} \frac{1}{2} \left[ (x_i - r_i)^T Q (x_i - r_i) + u_i^T R u_i \right] \right\}, \quad (14)$$

where  $\mathbb{E}$  denotes the mathematical expectation over the noise  $\{E(0), E(1), \dots\}$ , and  $Q > 0$  and  $R > 0$  are symmetric matrices. The performance index in Equation (14) can be rewritten using Equation (13) as:

$$J = \mathbb{E} \left\{ \sum_{i=0}^{\infty} \frac{1}{2} \left[ X_i^T Q_1 X_i + u_i^T R u_i \right] \right\}, \quad (15)$$



where  $Q_1 = C_1^T Q C_1$  with  $C_1 = [I, -I]$ . The stochastic system (12) is said to be mean-square stabilizable if there exists a state feedback control [27]:

$$u_k^* = -KX_k. \quad (16)$$

When the model parameters are known, there are multiple methods to determine the optimal feedback gains and the associated optimal cost. The optimal cost can be derived from the solution of the generalized algebraic Riccati Equation (GARE) [28]:

$$P = Q_1 + \bar{A}^T P \bar{A} + M \bar{D}^T P \bar{D} - \bar{A}^T P \bar{B} (R + \bar{B}^T P \bar{B})^{-1} \bar{B}^T P \bar{A}. \quad (17)$$

The optimal gain matrix is as follows:

$$K^* = - \left( R + \bar{B}^T P \bar{B} \right)^{-1} \bar{B}^T P \bar{A}. \quad (18)$$

As verified in [29], the standard conditions for the uniqueness and existence of a solution to the standard ARE require that  $(\bar{A}, \bar{B})$  is stabilizable and  $(\bar{A}, \sqrt{Q_1})$  is detectable [30].

### 3.2. Data-Driven Reinforcement Learning for LQT Problems

In this section, we establish the action-value Q-function for reinforcement learning to iteratively solve the Riccati equation for quadratic optimal control of stochastic systems with unknown model parameters using data. The Q-function is equivalent to the cost function in optimal control, and we need to optimize this Q-function to achieve the optimal value through the control strategy [31]. We construct our Q-function according to the standard control method. Firstly, we need to defined the Q-function. If the control policy, such as in Equation (16), is a mean-square stabilization control policy, then the Q-function should be defined in the desired form. Thus, the Q-function is defined as follows:

$$Q(X_k, u_k) = \mathbb{E} \left\{ \sum_{i=0}^{\infty} \frac{1}{2} \left[ X_i^T Q_1 X_i + u_i^T R u_i \right] \right\}. \quad (19)$$

where  $Q_1$  is defined in Equation (15). The Bellman equation transforms an infinite sum of terms into a simpler, future-term form. Thus, based on the discrete-time LQT Bellman equation and Equation (19):

$$\begin{aligned} Q(X_k, u_k) &= \mathbb{E} \left\{ \frac{1}{2} X_k^T Q_1 X_k + \frac{1}{2} u_k^T R u_k + \Gamma Q(X_{k+1}, u_{k+1}) \right\} \\ &= \mathbb{E} \left\{ \frac{1}{2} X_k^T P X_k \right\}, \end{aligned} \quad (20)$$

where  $P$  is defined in Equation (17). The Q-function can be further extended as follows:

$$Q(X_k, u_k) = \mathbb{E} \left\{ \frac{1}{2} X_k^T Q_1 X_k + \frac{1}{2} u_k^T R u_k + \frac{1}{2} \Gamma X_{k+1}^T P X_{k+1} \right\}, \quad (21)$$

where  $0 < \Gamma \leq 1$  is the discount factor. This discount factor is crucial because it prevents the reward from increasing to infinity as the time step approaches infinity, thereby making the infinitely long control process evaluable. The discount factor represents the expectation of future rewards. A smaller discount factor places more emphasis on the reward in the current state, while a larger discount factor emphasizes future rewards. However, in the context of an infinite-time control process, the condition  $\Gamma = 1$  holds if and only if the reference trajectory  $r_k$  is Schur stable. The selection of the discount factor depends on the

desired effect, involving a balance between exploration and exploitation of the algorithm. By incorporating augmented system dynamics Equation (12), the Q-function becomes:

$$Q(X_k, u_k) = \mathbb{E} \left\{ \frac{1}{2} X_k^T Q_1 X_k + \frac{1}{2} u_k^T R u_k + \frac{1}{2} \Gamma (\bar{A} X_k + \bar{B} u_k + \bar{D} X_k E_k)^T P (\bar{A} X_k + \bar{B} u_k + \bar{D} X_k E_k) \right\}. \quad (22)$$

To simplify the computation of the Q-function, we introduce a kernel matrix  $\mathcal{H} = \mathcal{H}^T$ , emphasizing its quadratic nature in the variables:

$$\begin{aligned} Q(X_k, u_k) &= \mathbb{E} \left\{ \frac{1}{2} \begin{bmatrix} X_k \\ u_k \end{bmatrix}^T \mathcal{H} \begin{bmatrix} X_k \\ u_k \end{bmatrix} \right\} \\ &= \mathbb{E} \left\{ \frac{1}{2} \begin{bmatrix} X_k \\ u_k \end{bmatrix}^T \begin{bmatrix} \mathcal{H}_{XX} & \mathcal{H}_{Xu} \\ \mathcal{H}_{uX} & \mathcal{H}_{uu} \end{bmatrix} \begin{bmatrix} X_k \\ u_k \end{bmatrix} \right\}, \end{aligned} \quad (23)$$

$$\mathcal{H}_{XX} = Q_1 + \Gamma (\bar{A}^T P \bar{A} + M \bar{D}^T P \bar{D}), \quad (24)$$

$$\mathcal{H}_{Xu} = \Gamma \bar{A}^T P B_1, \quad (25)$$

$$\mathcal{H}_{uX} = \Gamma \bar{B}^T P \bar{A}, \quad (26)$$

$$\mathcal{H}_{uu} = R + \Gamma B_1^T P B_1, \quad (27)$$

The optimal control action can then be derived by setting the gradient of Q-Function with respect to to zero, leading to a policy formula based on the Riccati-type solution. Applying the condition  $\frac{\partial Q(X_k, u_k)}{\partial u_k} = 0$  to Equation (23), we can obtain:

$$u_k = -\mathcal{H}_{uu}^{-1} \mathcal{H}_{uX} X_k, \quad (28)$$

Substituting Equations (26) and (27) into Equation (28), we have:

$$u_k = -(R + \Gamma \bar{B}^T P \bar{B})^{-1} \Gamma (\bar{B}^T P \bar{A}) X_k. \quad (29)$$

Define the extended state  $Z_k$  as:

$$Z_k = \begin{bmatrix} X_k \\ u_k \end{bmatrix}. \quad (30)$$

The Q-function can then be written as the following form by substituting Equation (30) into Equation (20):

$$Q(X_k, u_k) = \mathbb{E} \left\{ \frac{1}{2} \begin{bmatrix} X_k \\ u_k \end{bmatrix}^T \mathcal{H} \begin{bmatrix} X_k \\ u_k \end{bmatrix} \right\} = \mathbb{E} \left\{ \frac{1}{2} Z_k^T \mathcal{H} Z_k \right\}. \quad (31)$$

where

$$Z_k^T \mathcal{H} Z_k = X_k^T Q_1 X_k + u_k^T R u_k + \Gamma Z_{k+1}^T \mathcal{H} Z_{k+1}. \quad (32)$$

The multiplicative noise is then incorporated into the kernel matrix  $\mathcal{H}$  to participate in the policy iteration without requiring additional consideration.

The algorithm is structured around two main components:

#### 1 Policy Evaluation:

$$\mathbb{E} \left\{ Z_k^T \mathcal{H}^{j+1} Z_k \right\} = \mathbb{E} \left\{ X_k^T Q_1 X_k + (u_k^j)^T R (u_k^j) + \Gamma Z_{k+1}^T \mathcal{H}^{j+1} Z_{k+1} \right\}. \quad (33)$$

Estimating the Q-function based on current policy and updating the kernel matrix  $\mathcal{H}$  using collected data by Equation (33).



## 2 Policy Improvement:

$$u_k^{j+1} = -(\mathcal{H}_{uu}^{-1})^{j+1} \mathcal{H}_{uX}^{j+1} X_k. \quad (34)$$

Adjusting the control policy to minimize future Q-function values thereby optimizes system performance over time. Policy iteration can be implemented using least squares (LS) with the data tuple [22].

Through these steps, the reinforcement Q-learning algorithm iteratively refines its estimates and policy, adapting to changes in system dynamics and performance criteria without requiring prior knowledge of the underlying system model. The multiplicative noise is embedded within the kernel matrix using the data-driven method. This approach not only enhances the flexibility of the control system, but also improves its robustness in handling real-world operational variabilities.

Algorithm 1 provides a structured approach to implement the proposed control framework, ensuring that each step is clearly defined and actionable.

---

### Algorithm 1: Tracking control framework with output probability density function

---

**Input:** Target PDF  $\gamma_g(y, k)$  with time dynamics

**Output:** PDF

```

1 Choose  $n$  B-spline basis functions and a stabilization control policy  $u_0$ ;
2 Model the PDF and get the weight dynamic using B-spline models as Equation (5);
3 Initialize: In the Q-function  $Q$  and  $R$ , nuclear matrix  $\mathcal{H}_0$ , discount factor  $\Gamma$ , error
  expected  $\sigma$  and multiplicative noise  $E_k$  here denotes random generation;
4 for  $k = 0$  do
5   Update  $x_{k+1}$  according to Equation (12)
6   Update  $\mathcal{H}_{k+1}$  according to Equation (33)
7   Update  $u_{k+1}$  according to Equation (34)
8   if  $\|x_k - r_k\| > \sigma$  then
9      $k = k + 1$ ; Back to step 5
10  else
11    break; Termination of learning
12  end
13 end
14 return result

```

---

## 4. Simulation Result

In this section, we demonstrate the effectiveness of the model-free PDF tracking control algorithm through two numerical simulations. During the simulation process, the algorithm does not have any knowledge of the system model and relies solely on numerical calculations.

We start by choosing the B-spline basis functions, which are crucial for transforming the original system into a form suitable for our control algorithm. B-spline basis functions are flexible and can be tailored to various interpolation and fitting requirements. Specifically, we use third-order B-spline basis functions because they provide a good balance between smoothness and computational complexity. The B-spline basis functions selected are as follows:

$$\begin{aligned}
 B_1(y) &= 0.5(y^2 + 6y + 9)I_1 + (-y^2 - 3y - 1.5)I_2 + 0.5y^2I_3, \\
 B_2(y) &= 0.5(y^2 + 4y + 4)I_2 + (-y^2 - y + 0.5)I_2 + 0.5(y^2 - 2y + 1)I_4, \\
 B_3(y) &= 0.5(y^2 + 2y + 1)I_3 + (-y^2 + y + 0.5)I_4 + 0.5(y^2 - 4y + 4)I_5, \\
 B_4(y) &= 0.5y^2I_4 + (-y^2 + 3y - 1.5)I_5 + 0.5(y^2 - 6y + 9)I_6,
 \end{aligned} \quad (35)$$

$$\text{where } I_i = \begin{cases} 1 & y \in [i-4, i-3] \\ 0 & \text{others} \end{cases}, i = 1, \dots, 6.$$

As per constraint (2), only the corresponding weights for three of the four B-spline basis functions are necessary.

### Example 1

The fourth weight is linearly dependent on the first three weights, thereby reducing the model order to three. For example, 1, we use the same model as in reference [32] for comparison to demonstrate the difference between model-based and model-free approaches. By comparing the results, we can highlight the advantages and limitations of the proposed model-free reinforcement Q-learning algorithm against traditional model-based methods. Thus, the coefficient matrix of the system model is given by:

$$x_{k+1} = Gx_k + Hu_k + Dx_kE_k, \quad (36)$$

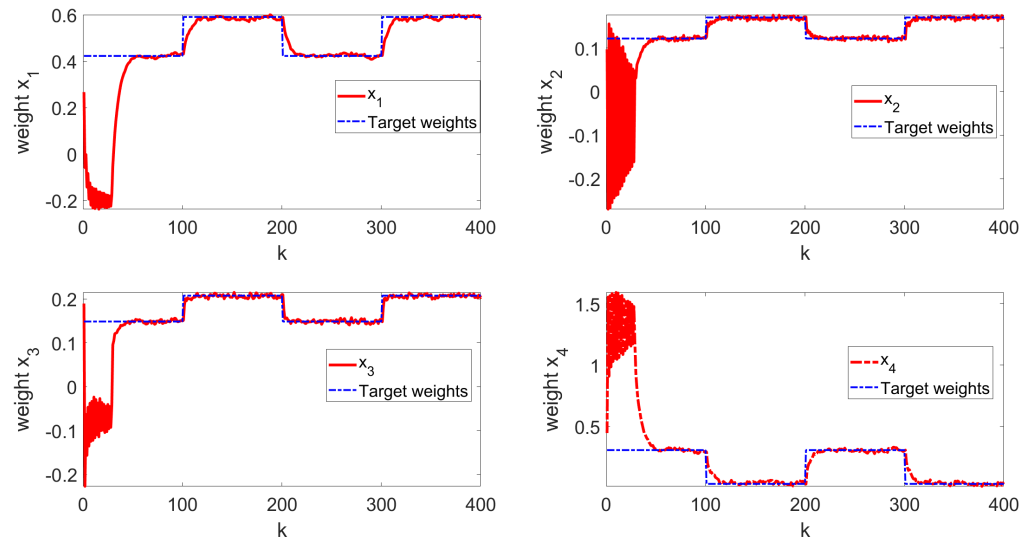
$$\text{with } G = \begin{bmatrix} 0.555 & -0.098 & -0.041 \\ -0.1 & -0.734 & 0.181 \\ -0.292 & 0.02 & 0.291 \end{bmatrix}, H = \begin{bmatrix} 0.275 \\ 0.302 \\ 0.302 \end{bmatrix}, D = \begin{bmatrix} 0.41 & 1.66 & 0.51 \\ -0.11 & 0.215 & 0.16 \\ 0.31 & 0.02 & -1.005 \end{bmatrix}.$$

In the described model,  $G$  represents the state weight matrix,  $H$  is the control matrix,  $D$  is the random weight matrix of the noise term, and  $E_k$  is the Gaussian noise. Note that the parameters  $G$ ,  $H$ , and  $D$  are assumed unknown. The target weights change every 100 steps, alternating between  $r_k = [0.4229 \ 0.1217 \ 0.1487]^T$  and  $r_k = [0.5908 \ 0.1701 \ 0.2077]^T$ . Additionally, the initial state of the system is set at  $x_0 = [0.2673 \ 0.0969 \ 0.1897]^T$ . The noise covariance follows the distribution given by  $E_k \sim N(0, 0.004)$ . The performance index, as indicated in Equation (14), employs  $Q = \text{diag}[20 \ 20 \ 20]$  and  $R = 0.1$ , with a discount factor  $\Gamma = 0.8$ . The parameter  $\sigma$  is chosen to be a suitably small value = 0.01. The initial kernel matrix  $\mathcal{H}_0$  is given by:

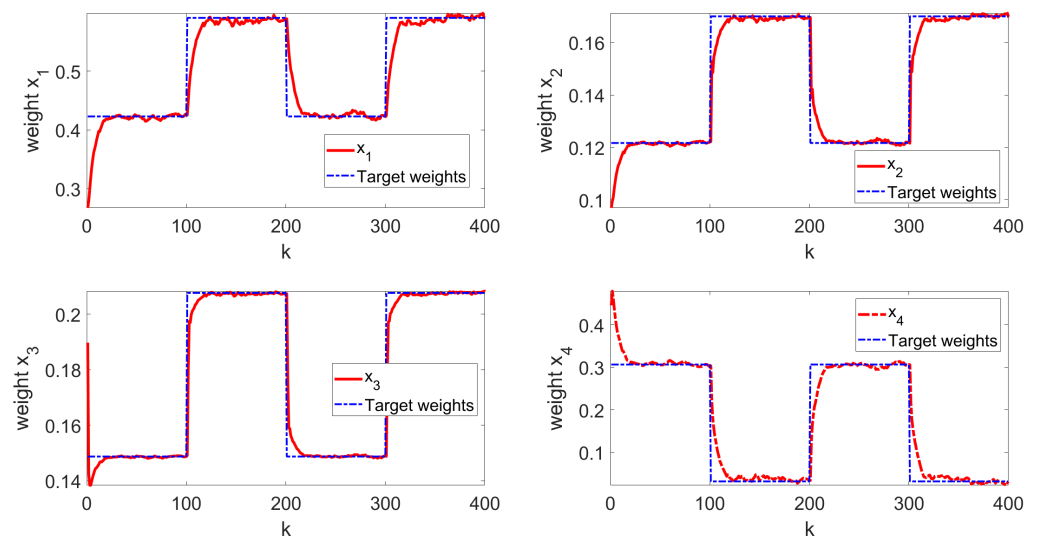
$$\mathcal{H}_0 = \begin{bmatrix} 65.9612 & 16.9868 & 12.0472 & 80.0261 & -12.02978 & 3.0209 & -27.0793 \\ 49.9990 & 86.9108 & -13.9571 & 54.1188 & -23.0361 & -25.0965 & 74.9031 \\ 3.9340 & -17.9644 & 38.9088 & 70.9414 & 13.0338 & -32.9645 & 79.0096 \\ 71.9722 & 8.9283 & 52.9810 & 33.9975 & 16.9669 & 69.9581 & 15.0152 \\ 96.0250 & -14.9325 & 93.0874 & 34.9930 & 69.0667 & 35.0675 & 63.9573 \\ -30.0573 & 86.0075 & 70.0079 & -0.0794 & -10.0277 & -39.0610 & -23.9975 \\ -18.9535 & 6.0832 & -31.9131 & 37.9693 & 66.9357 & 99.0435 & -24.9186 \end{bmatrix}$$

The simulation results are shown in Figures 3–7. Figure 3 illustrates the weight tracking effect during the online learning process. It is evident that the algorithm begins with exploration and converges around the 40th iteration, demonstrating its computational efficiency. Notably, in Figure 4, the red line represents the weight tracking curve under the enhanced Q-learning algorithm, while the blue dashed line represents the target reference curve. The weights  $r_1$ ,  $r_2$ , and  $r_3$  are controlled states, whereas  $r_4$ , which is linearly related to the other weights and excluded from control due to constraints, is also shown. Figure 4 further shows the control of the four weights and their corresponding target reference curves after the learning is completed. This demonstrates that the reinforcement Q-learning algorithm successfully tracks the specified target weights after learning. Figure 5 presents the system control input, with the red line indicating the control input curve under the reinforcement Q-learning algorithm. This curve aligns synchronically with the target configuration change and is smooth enough for practical use. Figure 6 illustrates the output curve of the reinforcement Q-learning algorithm, which consistently follows the desired PDF shape at each time instant. It also presents the results of the target weights integrated with the PDF output derived from the B-spline model. For example, in controlling the flame shape, this curve represents the flame shape at each moment. Despite the unknown system parameters, the output curve achieves the desired PDF shape, highlighting the effective utilization of the data. Figure 7 shows the tracking error curve of the four states. It can be seen from the figure that, despite the presence of multiplicative noise interference, the tracking effect remains very good. The tracking error only experiences a jump when

the tracking weight changes, and in the remaining cases, it is essentially near 0. Finally, the mean square error of the tracking is calculated to be very small, specifically 0.0285, indicating that the tracking effect is good. These results indicate that the proposed control framework effectively enables the system's PDF to track a predefined PDF shape without requiring specific knowledge of the system parameters.



**Figure 3.** Online learning process state curve.



**Figure 4.** Learning ending state curve.

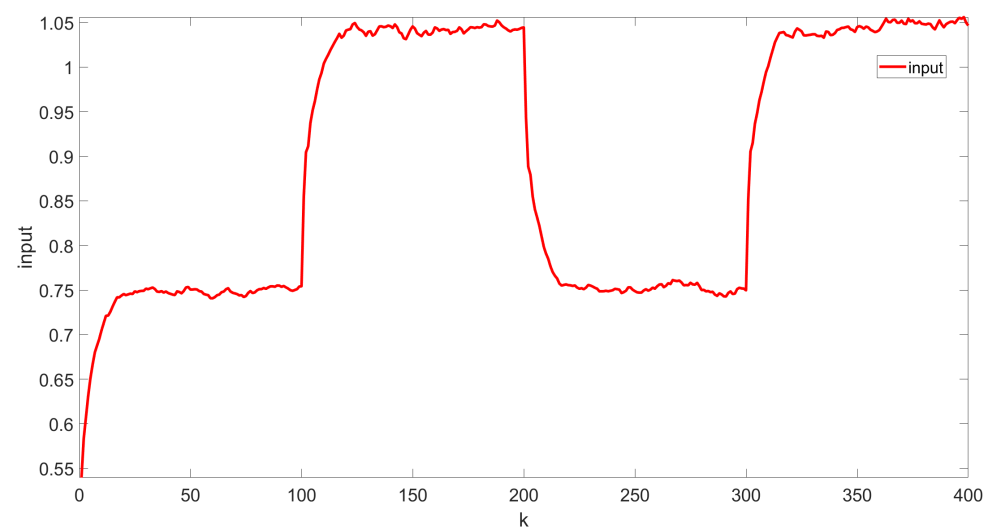


Figure 5. Control input curve.

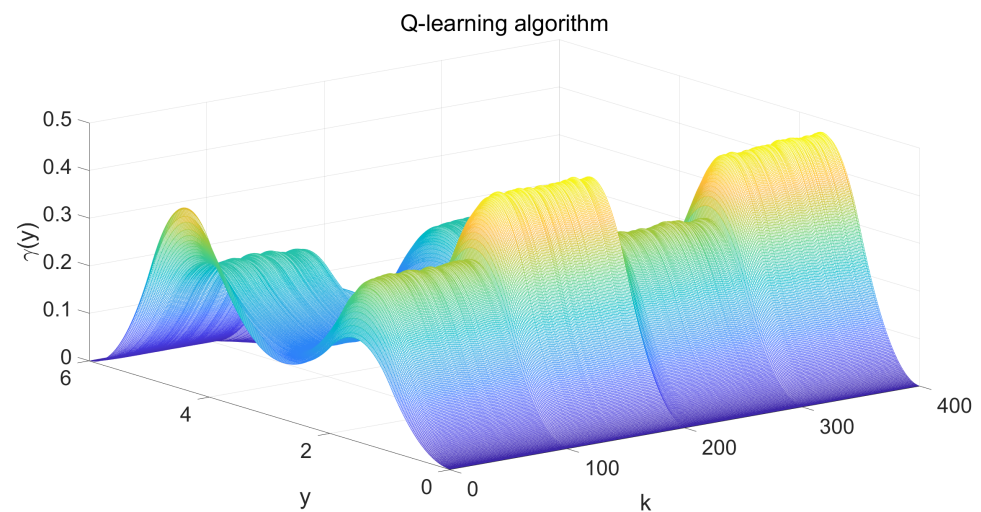


Figure 6. System output PDF of 3D drawings.

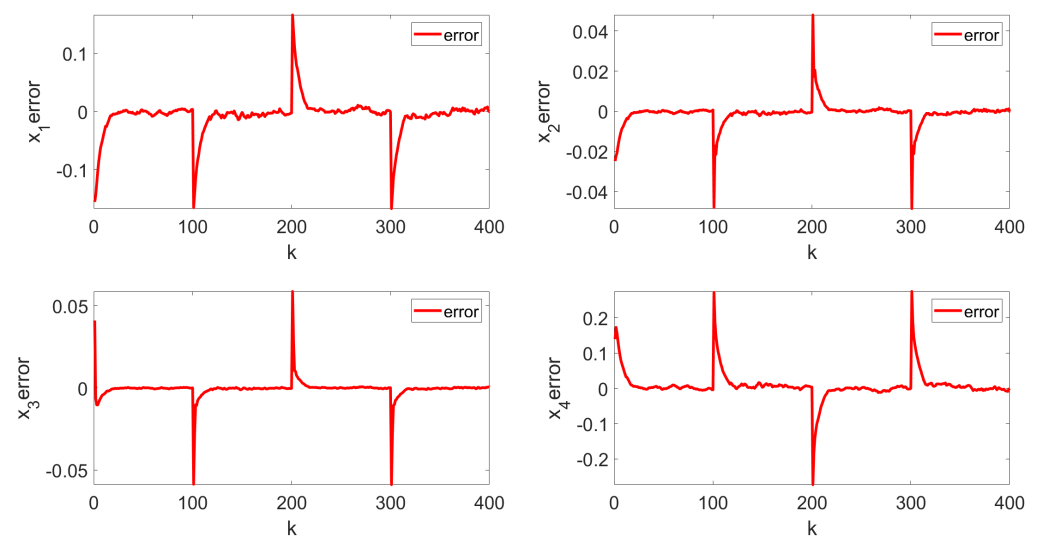
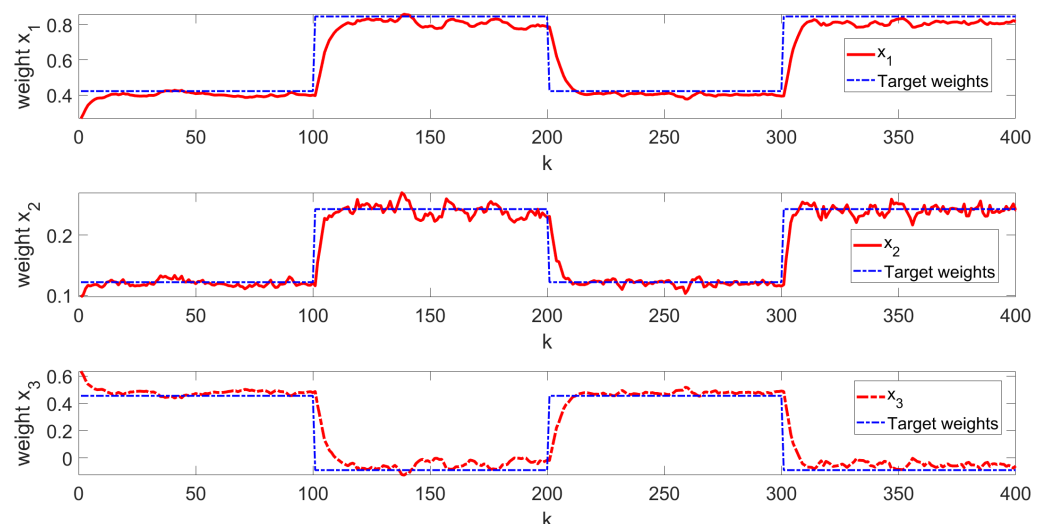


Figure 7. Tracking error.

### Example 2

To strengthen the argument for the algorithm's applicability in model-free scenarios, we provide another numerical simulation example, with  $G = \begin{bmatrix} 0.6969 & 0.6545 \\ -0.0241 & 0.8603 \end{bmatrix}$ ,  $H = \begin{bmatrix} 0.1298 \\ 0.0787 \end{bmatrix}$ ,  $D = \begin{bmatrix} 0.0081 & 0.6649 \\ 0.1665 & 0.1786 \end{bmatrix}$ . The target weights change every 100 steps, alternating between  $r_k = [0.4229 \ 0.1217]^T$  and  $r_k = [0.8458 \ 0.2434]^T$ . Additionally, the initial state of the system is set at  $x_0 = [0.2673 \ 0.0969]^T$ . The noise covariance follows the distribution given by  $E_k \sim N(0, 0.04)$ . The initial kernel matrix is  $\mathcal{H}_0 = \begin{bmatrix} 66.00452 & 19.9959 & 95.0028 & -29.996 & 42.9970 \\ 23.0017 & 32.9949 & 60.9976 & 54.99558 & -33.9953 \\ 37.9961 & 55.9946 & -36.0101 & 77.9975 & 74.0099 \\ 70.9958 & -19.0035 & 91.0042 & 31.0001 & 88.0092 \\ 33.9968 & 16.0032 & 57.0020 & 2.0017 & 31.9961 \end{bmatrix}$ .

The simulation results are illustrated in Figure 8. In this figure, the red solid line represents the system state under our control, while the blue dashed line indicates the target weight. As shown, the system states successfully tracks the pre-set time-varying references. Additionally, the mean square error is calculated to be 0.1561, demonstrating the effective control performance. This result clearly shows that even without a model, the second-order stochastic system can accurately track the target weight.



**Figure 8.** Learning ending state curve.

Through two numerical simulation examples, we observe that the overall tracking error is minimal, the response speed is rapid, and the learning efficiency is high in both systems. This indicates that our control method is effective across different systems. Additionally, the mapping relationship between the systems is established using B-spline function fitting, which allows for controlling the PDF shape by tracking the target weight. Furthermore, the presented approach does not require prior knowledge of the model, enabling flexible back-and-forth switching control of the PDF shape.

### 5. Conclusions

This paper addresses the challenge of output distribution shape tracking in stochastic distribution systems by developing a learning-based, model-free control algorithm within the B-spline model framework to approximate the PDF. This method simplifies the complex issue of PDF shaping into a more manageable problem of dynamic weight modification, treating the system's inherent randomness and inaccuracies as state-dependent noises,

which closely mirror real-world complexities. To handle time-varying targets and reduce dependency on precise model knowledge, an extended Reinforcement Q-learning algorithm is applied in this framework. Simulation results confirm the method's effectiveness, demonstrating its ability to accurately track varying distribution shapes. Using the data-driven method to control, it greatly removes the limitation that most control methods need system model parameters. Additionally, this approach eliminates the need to address complexities arising from multiplicative noise issues.

**Author Contributions:** Conceptualisation, Y.Z. (Yong Zhang); Methodology, Y.Z. (Yuyang Zhou); Validation, W.Y.; Writing—original draft preparation, W.Y.; Resources, Y.Z. (Yong Zhang); Writing—review and editing, Y.Z. (Yuyang Zhou); funding acquisition, Y.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China under Grant 62263026, Grant 62063027, the Fundamental Research Funds for Inner Mongolia University of Science and Technology under Grant 2024QNJS003, the Inner Mongolia Natural Science Foundation under Grant 2023MS06001, the Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region under Grant NJYT22057, the Fundamental Research Funds for Inner Mongolia University of Science and Technology under Grant 2023RCTD028, and the Inner Mongolia Autonomous Region Control Science and Engineering Quality Improvement and Cultivation Discipline Construction Project.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are contained within the article.

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

## References

1. Ren, M.; Zhang, Q.; Zhang, J. An introductory survey of probability density function control. *Syst. Sci. Control Eng.* **2019**, *7*, 158–170. [\[CrossRef\]](#)
2. Lu, J.; Han, L.; Wei, Q.; Wang, X.; Dai, X.; Wang, F.Y. Event-triggered deep reinforcement learning using parallel control: A case study in autonomous driving. *IEEE Trans. Intell. Veh.* **2023**, *8*, 2821–2831. [\[CrossRef\]](#)
3. Filip, I.; Dragan, F.; Szeidler, I.; Albu, A. Minimum-variance control system with variable control penalty factor. *Appl. Sci.* **2020**, *10*, 2274. [\[CrossRef\]](#)
4. Li, M.; Zhou, P. Predictive PDF control of output fiber length stochastic distribution in refining process. *Acta Autom. Sin.* **2019**, *45*, 1923–1932.
5. Sun, X.; Xun, L.; Wang, H.; Dong, H. Iterative learning control of singular stochastic distribution model of jet flame temperature field. *J. Beijing Univ. Technol.* **2013**, *33*, 523–528.
6. Cao, L.; Wu, H. MWD modeling and control for polymerization via B-spline neural network. *J. Chem. Ind. Eng. China* **2004**, *55*, 742–746.
7. Wang, H.; Yue, H. Output PDF control of stochastic distribution systems: Modelling control and applications. *Control Eng. China* **2003**, *10*, 193–197.
8. Wang, H.; Zhang, J. Bounded stochastic distributions control for pseudo-ARMAX stochastic systems. *IEEE Trans. Autom. Control.* **2001**, *46*, 486–490. [\[CrossRef\]](#)
9. Zhang, Q.; Zhou, Y. Recent advances in non-Gaussian stochastic systems control theory and its applications. *Int. J. Netw. Dyn. Intell.* **2022**, *1*, 111–119. [\[CrossRef\]](#)
10. Wang, H. *Bounded Dynamic Stochastic Systems: Modelling and Control*, 1st ed.; Springer Science & Business Media: London, UK, 2000; pp. 15–34.
11. Huang, E.; Cheng, Y.; Hu, W. Tracking control of multi-agent systems based on reset control. *Control. Eng. China* **2022**, *29*, 6.
12. Luan, X.; Liu, F. Finite time stabilization of output probability density function of stochastic systems. *Control Decis.* **2009**, *24*, 1161–1166.
13. Zhou, J.; Wang, H. Optimal tracking control of the output probability density functions: Square root B-spline model. *Control Theory Appl.* **2005**, *22*, 369–376.
14. Hansen-Estruch, P.; Kostrikov, I.; Janner, M.; Kuba, J.G.; Levine, S. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv* **2023**, arXiv:2304.10573.
15. Carmona, R.; Laurière, M.; Tan, Z. Model-free mean-field reinforcement learning: Mean-field MDP and mean-field Q-learning. *Ann. Appl. Probab.* **2023**, *33*, 5334–5381. [\[CrossRef\]](#)



16. Yaghmaie, F.A.; Modares, H.; Gustafsson, F. Reinforcement Learning for Partially Observable Linear Gaussian Systems Using Batch Dynamics of Noisy Observations. *IEEE Trans. Autom. Control.* **2024**. [\[CrossRef\]](#)
17. Nguyen, H.; Dang, H.B.; Dao, P.N. On-policy and off-policy Q-learning strategies for spacecraft systems: An approach for time-varying discrete-time without controllability assumption of augmented system. *Aerosp. Sci. Technol.* **2024**, *146*, 108972. [\[CrossRef\]](#)
18. Meyn, S. Stability of Q-learning through design and optimism. *arXiv* **2023**, arXiv:2307.02632.
19. Garg, D.; Hejna, J.; Geist, M.; Ermon, S. Extreme q-learning: Maxent rl without entropy. *arXiv* **2023**, arXiv:2301.02328.
20. Lopez, V.G.; Alsalti, M.; Müller, M.A. Efficient off-policy Q-learning for data-based discrete-time LQR problems. *IEEE Trans. Autom. Control.* **2023**, *68*, 2922–2933. [\[CrossRef\]](#)
21. Wang, R.; Zhuang, Z.; Tao, H.; Paszke, W.; Stojanovic, V. Q-learning based fault estimation and fault tolerant iterative learning control for MIMO systems. *ISA Trans.* **2023**, *142*, 123–135. [\[CrossRef\]](#)
22. 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. [\[CrossRef\]](#)
23. Xue, W.; Fan, J.; Lopez, V.G.; Jiang, Y.; Chai, T.; Lewis, F.L. Off-policy reinforcement learning for tracking in continuous-time systems on two time scales. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4334–4346. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Zha, W.; Li, D.; Shen, L.; Zhang, W.; Liu, x. Review of neural network-based methods for solving partial differential equations. *Chin. J. Theor. Appl. Mech.* **2022**, *54*, 543–556.
25. Zhang, Y.; Zhou, P.; Lv, D.; Zhang, S.; Cui, G.; Wang, H. Inverse calculation of burden distribution matrix using B-spline model based PDF control in blast furnace burden charging process. *IEEE Trans. Ind. Inform.* **2023**, *19*, 317–327. [\[CrossRef\]](#)
26. Hu, B.; Zhang, K.; Li, N.; Mesbahi, M.; Fazel, M.; Başar, T. Toward a theoretical foundation of policy optimization for learning control policies. *Annu. Rev. Control. Robot. Auton. Syst.* **2023**, *6*, 123–158. [\[CrossRef\]](#)
27. Wang, D.; Wang, J.; Zhao, M.; Xin, P.; Qiao, J. Adaptive multi-step evaluation design with stability guarantee for discrete-time optimal learning control. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 1797–1809. [\[CrossRef\]](#)
28. Gravell, B.; Esfahani, P.M.; Summers, T. Learning optimal controllers for linear systems with multiplicative noise via policy gradient. *IEEE Trans. Autom. Control.* **2020**, *66*, 5283–5298. [\[CrossRef\]](#)
29. Willems, J.L.; Willems, J.C. Feedback stabilizability for stochastic systems with state and control dependent noise. *Automatica* **1976**, *12*, 277–283. [\[CrossRef\]](#)
30. Lewis, F.L.; Vrabie, D.; Syrmos, V.L. *Optimal Control*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
31. Xiao, B.; Lam, H.K.; Su, X.; Wang, Z.; Lo, F.P.-W.; Chen, S.; Yeatman, E. Sampled-data control through model-free reinforcement learning with effective experience replay. *J. Autom. Intell.* **2023**, *2*, 20–30. [\[CrossRef\]](#)
32. Yang, Y.; Zhang, Y.; Zhou, Y. Tracking Control for Output Probability Density Function of Stochastic Systems Using FPD Method. *Entropy* **2023**, *25*, 186. [\[CrossRef\]](#)

**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.