electronics

MDPI

Article

Online Three-Dimensional Fuzzy Reinforcement Learning Modeling for Nonlinear Distributed Parameter Systems

Xianxia Zhang [1,*], Runbin Yan [2], Gang Zhou [1], Lufeng Wang [1] and Bing Wang [1]

[1] School of Mechatronics and Automation, Shanghai University, 99 Shangda Rd., Shanghai 200444, China; gangzhou@shu.edu.cn (G.Z.); wanglufeng@shu.edu.cn (L.W.); susanbwang@shu.edu.cn (B.W.)
[2] North Automatic Control Technology Institute, Taiyuan 030000, China; happyray288@gmail.com
* Correspondence: xianxiazh@t.shu.edu.cn

Abstract: Distributed parameter systems (DPSs) frequently appear in industrial manufacturing processes, with complex characteristics such as time–space coupling, nonlinearity, infinite dimension, uncertainty and so on, which is full of challenges to the modeling of the system. At present, most DPS modeling methods are offline. When the internal parameters or external environment of DPS change, the offline model is incapable of accurately representing the dynamic attributes of the real system. Establishing an online model for DPS that accurately reflects the real-time dynamics of the system is very important. In this paper, the idea of reinforcement learning is creatively integrated into the three-dimensional (3D) fuzzy model and a reinforcement learning-based 3D fuzzy modeling method is proposed. The agent improves the strategy by continuously interacting with the environment, so that the 3D fuzzy model can adaptively establish the online model from scratch. Specifically, this paper combines the deterministic strategy gradient reinforcement learning algorithm based on an actor critic framework with a 3D fuzzy system. The actor function and critic function are represented by two 3D fuzzy systems and the critic function and actor function are updated alternately. The critic function uses a TD (0) target and is updated via the semi-gradient method; the actor function is updated by using the chain derivation rule on the behavior value function and the actor function is the established DPS online model. Since DPS modeling is a continuous problem, this paper proposes a TD (0) target based on average reward, which can effectively realize online modeling. The suggested methodology is implemented on a three-zone rapid thermal chemical vapor deposition reactor system and the simulation results demonstrate the efficacy of the methodology.

Keywords: distributed parameter system; fuzzy modeling; reinforcement learning; online modeling; 3D fuzzy system

Citation: Zhang, X.; Yan, R.; Zhou, G.; Wang, L.; Wang, B. Online Three-Dimensional Fuzzy Reinforcement Learning Modeling for Nonlinear Distributed Parameter Systems. *Electronics* **2024**, *13*, 4217. https://doi.org/10.3390/electronics13214217

Academic Editors: Alexander Gegov, Raheleh Jafari and Farzad Arabikhan

Received: 5 September 2024
Revised: 18 October 2024
Accepted: 21 October 2024
Published: 27 October 2024

1. Introduction

Distributed parameter systems (DPSs) are prevalent in the real world, such as flexible beams [1], transport reaction processes [2] and heat processes [3]. Modeling these systems is crucial for the design, optimization, dynamic prediction and control. However, due to the complex system characters, such as time–space coupling, infinite dimension and so on, it is difficult to model these systems.

In the last several decades, many modeling methods of DPS have been proposed and these methods may be broadly classified into two distinct types. Grey/black box modeling with unknown partial differential equations [4–7] and first principles modeling with known partial differential equations (PDEs) [8–13] are the two types of modeling. Obtaining an accurate mathematical description of a system using partial differential equations is challenging due to imperfect understanding of physical and chemical processes. Compared with the first principles modeling method, the grey/black box modeling method is used more widely in practice. This study concentrates on the black box modeling method, which involves developing a model for a DPS without knowledge of the system's partial differential equation structure or its parameters.

The black box modeling method drives a model for the DPSs in a data-driven style [14]. To date, the most common data-driven modeling methods [15–18] of DPSs are based on KL decomposition, such as KL-MLP-LSTM [19], in which the infinite dimensional spatiotemporal coupling data are reduced to finite dimensional by KL decomposition, and the space base function is extracted from it, then the time series model is constructed by MLP-LSTM and finally the DPS model is completed by spatiotemporal synthesis. In Ref. [20], two neural network modeling methods are proposed. One is to use the control quantity and position information as the input and the output of the system is the system mathematical model obtained by training the neural network. The other is that singular value decomposition (SVD) is employed to separate the output data of the system in time and space, then a neural network is used to identify the relationship between the control variables and time coefficients and finally the finite low dimensional model of the system is generated via space–time reconstruction. Although the KL decomposition-based methods have been successfully employed in many applications, they neglect the influence of nonlinear space dynamics and depend on model reduction, which can lead to precision loss.

In recent years, a novel DPS modeling method has been developed: three-dimensional (3D) fuzzy modeling [21–23]. The preceding and resulting components of a 3D fuzzy system are the temporal coefficient and space base function, respectively. This is a natural implementation of time–space separation and time–space reconstruction. The time–space separation is achieved through the computation of the preceding and resulting components, while the time–space reconstruction is achieved through the union of all activated 3D rules. The 3D fuzzy modeling method has two distinct advantages: linguistic interpretability and a lack of dependence on model reduction. A novel 3D fuzzy modeling framework without model reduction was proposed in Ref. [24], where a 3D fuzzy model is established by employing the nearest neighbor clustering algorithm, the similarity analysis and multiple support vector regression. The method described in Reference [25] employed spatial multiple output support vector regression (MSVR) to construct a comprehensive 3D fuzzy rule foundation. However, the above methods are offline modeling methods, i.e., historical data-driven models. As the internal or external parameters of a distributed parameters system change over time, the performance of the offline model will become worse and worse [26].

Therefore, DPS online modeling has become a research hotspot recently. In Ref. [27], Wang and Li proposed an online modeling method based on incremental spatiotemporal learning. The methods of incremental learning iteratively update the space base function and the accompanying temporal model. In reference [28], Lu et al. introduced a nonlinear time-varying support vector machine model that depended on a spatiotemporal linear support vector machine. This model combined the adaptable space kernel functions with an online time factor model to reconstruct the DPS system. However, those online modeling methods have a complex modeling process and lack language interpretability.

Compared to the traditional learning methods, the reinforcement learning-based methods are naturally implemented in an incremental way based on immediate rewards obtained during the interaction, enabling online learning for different tasks. Reinforcement learning, as a very active algorithm in the field of machine learning, has been used in many disciplines like [29], control [30,31], recognition [32], batching [33], scheduling [34] and optimization [35,36]. Reinforcement learning algorithms learn the optimal policy directly from interactions with the environment, i.e., DPS in this paper. The fundamental components of reinforcement learning algorithms include policy, reward function, value function and environment.

Reinforcement learning algorithms may be categorized into two main types: value function-based reinforcement learning and direct policy search-based reinforcement learning, depending on the methodologies of policy updating and learning [37]. For value function-based reinforcement learning algorithms, David Sliver proposed DQN (Deep Q-network) [38] by combining deep neural network with reinforcement learning. There is also FQ-learning [39] which combines a custom fuzzy system with Q-learning. As for direct

policy search-based reinforcement learning, David Sliver et al. [33] proposed DPG (deterministic policy gradient) and proved for the first time that a deterministic policy gradient exists. DPG uses a deterministic policy and outputs deterministic actions for each state, so it avoids expectation in the action space and greatly increases the sample utilization.

In recent years, the combination of reinforcement learning (RL) and fuzzy systems has become an important research direction in the field of intelligent control and decision-making systems. This combination aims to leverage the adaptive and self-optimizing capabilities of RL and the advantages of fuzzy systems in handling uncertainty and fuzzy information, thereby creating more intelligent and robust systems. The integration of RL and fuzzy systems, often referred to as fuzzy reinforcement learning (FRL), combines the learning abilities of RL with the rule-based reasoning of fuzzy systems to effectively solve decision-making problems in complex environments. Specifically, fuzzy systems can serve as approximators for RL's policy functions or value functions, helping agents make smooth decisions in continuous spaces. Additionally, RL can automatically adjust the rules and parameters of fuzzy systems through online learning, thereby enhancing the adaptability and performance of the system. In Reference [40], a fuzzy Q-learning method was proposed to achieve the online tuning of a PID controller. By introducing a fuzzy system, it addresses the issue that the traditional Q-learning algorithm cannot handle continuous state–action spaces. The paper proposed a fuzzy Q-learning multi-agent system algorithm, where three agents were used to control the three variables of the PID controller and experimental results demonstrate the effectiveness of this method. In Reference [41], a method based on fuzzy actor–critic reinforcement learning was proposed to address the tracking problem. By using a fuzzy system, it enhanced the interpretability of network parameters, making the mapping structure easier to trace and understand. Compared with an artificial neural network, a fuzzy system is more explainable and it allowed the incorporation of necessary human knowledge to construct the inference rules. In Reference [42], an Intelligent Traffic Control Decision-Making method based on Type-2 Fuzzy and Reinforcement Learning was proposed. In this method, the output action of the Type-2 fuzzy control system replaced the action of selecting the maximum output Q-value of the target network in the DQN algorithm, reducing the error caused by the max operation in the target network. In Reference [43], a dynamic fuzzy Q-learning method was proposed, which enabled the automatic learning of the structure and parameters of the fuzzy system based on Q-learning. The effectiveness of the proposed method was validated through the wall-following task of mobile robots.

However, there is very little research on applying reinforcement learning and a fuzzy system to modeling distributed parameter systems. As far as we know, Wang Z. et al. [44] introduced a modeling method for DPSs based on reinforcement learning algorithms. The fundamental concept is to represent the arrangement of measurement sensors in the DPS as a Markov Decision Process model and then use a reinforcement learning method based on a value function to ascertain the most advantageous sensor placement. However, this method employs an offline modeling method, meaning that once the sensor positions are determined, they remain fixed. When the dynamic characteristics of the distributed parameter system change, the initially optimal sensor placement may no longer be optimal, resulting in a substantial decrease in the accuracy of the model.

This study introduces a novel reinforcement learning-based method for online 3D fuzzy modeling. Based on the actor–critic framework, using the deterministic strategy gradient theory, two 3D fuzzy systems are used to represent the actor function and the critic function, respectively, and the critic function and actor function are updated alternately. The critic function uses TD (0) target and is updated by the semi-gradient method; the actor function changes via incorporating the chain derivation rule to the behavior value function. The actor function serves as the established DPS online model. Because DPS online modeling is a continuous problem, it has been in progress without termination. Thus, the average reward return is used and the actor can update itself for each data in real time, so as to realize online modeling.

The main contribution are as follows:

1. The actor–critic framework is combined with 3D fuzzy systems to construct a novel online 3D modeling method using the deterministic policy gradient reinforcement learning algorithm.
2. Three-dimensional fuzzy deterministic policy gradient algorithm with average reward set is proposed to solve the continuing modeling problem of DPS. The chosen reward set is the average rate of reward every time step, which defines the performance.
3. This online modeling method has the ability to build an online model adaptively from scratch without human knowledge.

The subsequent sections of this work are structured in the following manner. Section 2 outlines the concept of the DPS modeling issue. Section 3 provides a comprehensive description of the suggested method for modeling a 3D fuzzy system using reinforcement learning in an online setting. The suggested modeling technique is used in a rapid thermal chemical vapor deposition system and the outcome of the experiment is shown in Section 4. Section 5 provides the final conclusion.

2. Problem Formulation

DPS commonly appears in industrial manufacturing processes. Taking the rapid thermal chemical vapor deposition (RTCVD) system in the semiconductor industry as an example, we introduce the characteristics and mathematical description of a DPS.

Figure 1 displays the schematic of an RTCVD system. A 6-inch silicon wafer is placed on a rotating platform inside the chamber of the system and exposed to heat from a heating apparatus. There are three lamp banks that make up the heating system. The first lamp bank warms the wafer's surface evenly. The second lamp bank heats only the wafer's edges and the third lamp bank warms the wafer evenly all over. Figure 2 shows the heating lamps' output incident radiation flux. The reactor is fed 10% concentration of silane gas (SiH_4). After breaking down, SiH4 yields silicon (Si) and hydrogen gas (H_2). After about one minute, the wafer is covered with a 0.5 μm thick layer of polysilicon that has been deposited at temperatures of at least 800K. The support is rotated to ensure that the temperature is distributed evenly in the azimuthal direction when the wafer is processing. Because the wafer is very thin, the temperature change in the $Z-axis$ direction is ignored. Therefore, providing a uniform temperature distribution along the length of the wafer radius becomes the decisive factor for depositing a uniform layer of polysilicon on the wafer surface. The uniform distribution of the temperature can be controlled by adjusting the power of the heating lamp group in the three zones. This thermal dynamic characteristic can be described by a one-dimensional spatiotemporal coupled PDE model, as follows:

$$\partial T_f'/\partial t' = k_0[(1/f')\partial T_f'/\partial f' + \partial^2 T_f'/\partial f'^2] + \sigma_0(1 - T_f'^4) + w_f q_1(f')u_1 + w_f q_2(r')u_2 \quad + w_f q_3(r')u_3 \tag{1}$$

The above PDE is constrained by the following boundary conditions:

$$\text{s.t.} \begin{cases} T_f'/\partial f' = \sigma_{ed}\left(1 - T_f'^4\right) + q_{ed}u_b & \text{when} \quad f' = 1 \\ \partial T_f'/\partial f' = 0 & \text{when} \quad f' = 0 \end{cases} \tag{2}$$

T_f' represents the dimensionless wafer temperature, expressed as T_f/T_{amb}. Here, T_f represents the actual wafer temperature and T_{amb} represents the ambient temperature. In this study, T_{amb}=300K. t' represents the dimensionless time, expressed as $\frac{t}{\tau}$. Here, t represents the actual time and τ is 2.9s. f' represents the dimensionless radius position, expressed as $\frac{f}{R_w}$. Here, f represents the actual radius position and R_w is 7.6 cm. u_1, u_2 and u_3 represent the power of three heating lamp groups, respectively. $q_1(f')$, $q_2(f')$, $q_3(f')$

represent the radiation flow of the three zone heating lamp group incidents on the wafer, respectively. The parameters in Equations (1) and (2) are listed as follows:

$$k_0 = 0.0021, \sigma_0 = 0.0012, \sigma_{ed} = 0.0037, q_{ed} = 4.022, w_f = 0.0256.$$

where k_0 is the thermal conductivity of the wafer, σ_0 is the emissivity of the quartz chamber, σ_{ed} is the is the emissivity of the wafer, q_{ed} is the incident radiation flux at the edge of the wafer and w_f is the density of the wafer.

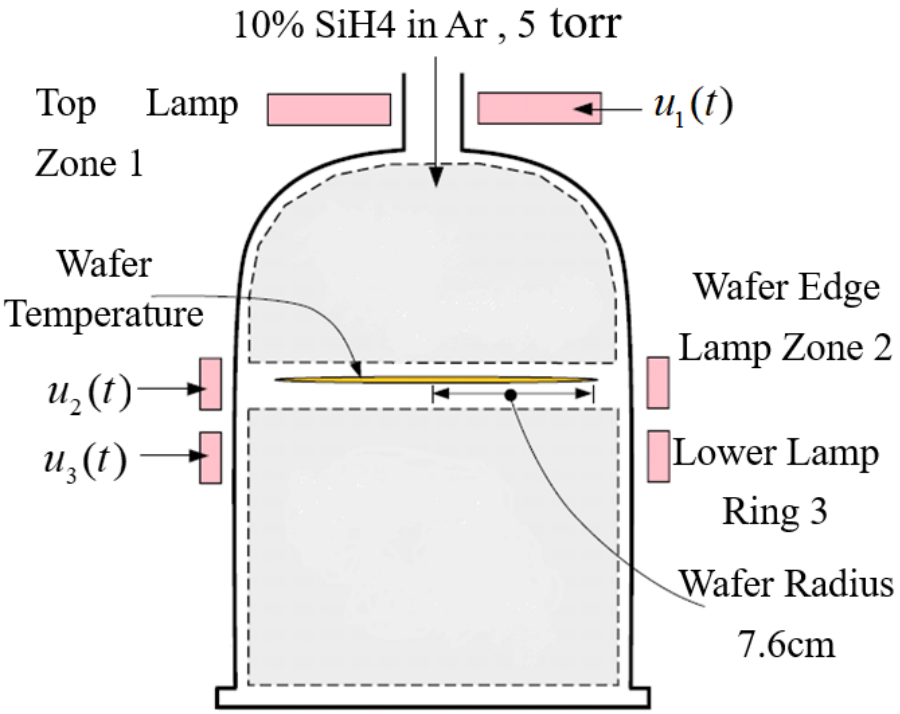

Figure 1. Structure of the RTCVD.

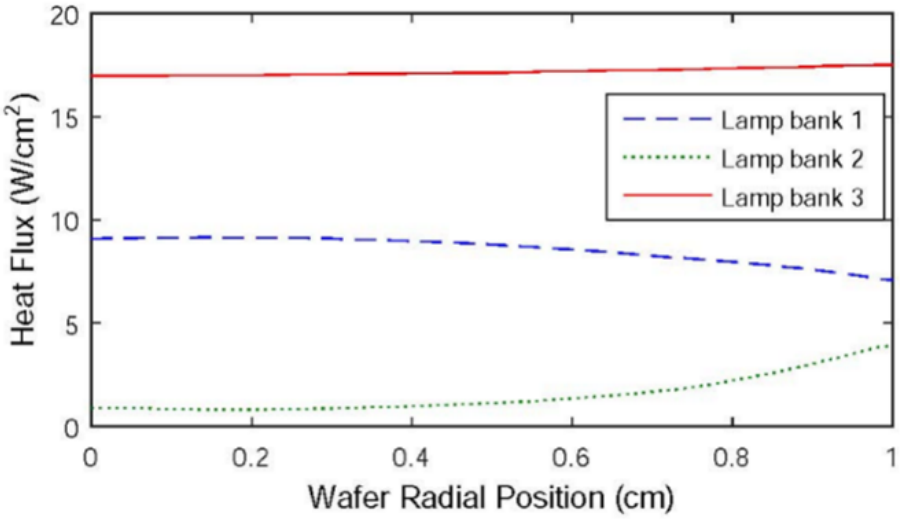

Figure 2. The distribution of radiation flux from a bank of heating lamps with three zones.

The RTCVD system is an infinite dimensional system with spatiotemporal coupling properties and is an infinite dimensional system. $T_f(f,t)$ can be spatiotemporally separated into an infinitely weighted summation of space base functions and time factors.

$$T_f(f,t) = \sum_{i=1}^{\infty} y_i \omega_i(f) \tag{3}$$

where y_i denotes a time coefficient and $\omega_i(f)$ denotes a space base function. In practice, this means that only a limited number of sensors at $f_1, f_2, \cdots, f_p$ may be used. Let $\bar{Z}$ denote the space domain, i.e., $\bar{Z} = [f_1, f_2, \cdots, f_p]$. $T(\bar{Z},t) = [T_f(f_1,t), T_f(f_2,t), \cdots, T_f(f_p,t)]$ denotes the spatial output.

Then, $T_f(f,t)$ may be essentially represented by Equation (4).

$$T_f(f,t) = \sum_{i=1}^{n} y_i \omega_i(f) \tag{4}$$

In this process, $T_f(f,t)$ is simplified from Equation (3) to Equation (4), which is called dimension reduction. In the absence of the mechanism model delineated in Equations (1) and (2), the time coefficient in Equation (4) may be determined using conventional methods. And the space base function in Equation (4) can be estimated by the KL decomposition technique. In fact, most current DPS modeling methods utilize the dimension-reduction technique. However, it has a limit, ignoring dynamic aspects and ambiguities in the identified model because of the reduced dimensionality.

The 3D fuzzy modeling method is a novel modeling technique that has been established in the last few years. This method naturally integrates spatiotemporal separation and spatiotemporal reconstruction into 3D fuzzy rules without dimension reduction and has language interpretability. This paper incorporates reinforcement learning into 3D fuzzy modeling and presents a novel method for online 3D fuzzy modeling using reinforcement learning.

3. Reinforcement Learning-Based Online 3D Fuzzy Modeling

This section first introduces the Markov Decision Process (MDP) model as the basis of DPS modeling utilizing reinforcement learning. Then, the core content of this paper, 3D fuzzy deterministic policy gradient (3DFDPG) reinforcement learning, is described in detail, including the deterministic policy gradient algorithm, 3D fuzzy modeling and the fusion of them: 3DFDPG modeling method. Finally, under an actor–critic framework, it describes how to update the parameters of a 3D fuzzy system, including the update of actor function parameters and critic function parameters and the calculation process of the 3DFDPG modeling method.

3.1. MDP Model of the DPS Online Modeling Problem

Given the input and output data, we can derive the Markov Decision Process (MDP) model for the DPS, which is represented as a 5-tuple $< S, A, R, T, \gamma >$, where S and A, respectively, represent the sets of states and actions, R represents the reward function ($R(s,a|s \in S, a \in A)$ denotes the expected reward when taking action a in state s), T is the transition function ($T(s' \in S|s \in S, a \in A)$ denotes the probability of transitioning to state s' from s when taking action a and $\gamma \in [0,1]$ is the discount factor.

The output $y(\bar{Z},t)$ of a DPS is determined by the past outputs $[y(\bar{Z},t-1), y(\bar{Z},t-2), ..., y(\bar{Z},t-K)]$ and input $u(t-1), u(t-2), \cdots, u(t-G)$, where $u(t) = [u_1(t), \cdots, u_m(t)]$. For simplicity and without loss of generality, we consider the one-order case in this study, i.e., $K = 1$, $G = 1$, where $y(\bar{Z},t)$ depends on $y(\bar{Z},t-1)$ and $u(t-1)$. The DPS modeling problem can be expressed as an MDP. The state $s_t \in S$ is written as follows.

$$s_t = [u(t), y(\bar{Z},t)] \tag{5}$$

The action $a_t \in A$ is defined as $\hat{y}(\bar{Z}, t+1)$ which is the predicted DPS output at time step $t+1$, i.e.,

$$a_t = \hat{y}(\bar{Z}, t+1) \tag{6}$$

The reward function $R(s,a)$ is given by Equation (7).

$$R(s,a) = -\sqrt{(\hat{y}(\bar{Z}, t+1) - y(\bar{Z}, t+1))^2} \tag{7}$$

The transition function $T(s,a,s')$ is determined by the inherent dynamics of the DPS, which include its physical attributes and control inputs. γ is a fixed discounted factor.

Figure 3 illustrates the architecture of the suggested online modeling method, where two 3D fuzzy systems are used. One is taken as the function of actor and the other is viewed as the role of a critic. The actor 3D fuzzy system serves as the DPS model, which takes s_t as input and exports the predicted value $\hat{y}(\bar{Z}, t+1)$ as action at every time step t, then the environment exports reward R according to Equation (7). The critic-3D fuzzy system takes the state s_t and the action a_t as input and exports the action value $Q(s_t, a_t)$. The main process of the proposed algorithm is interleaving evaluation and improvement of the actor's policy. This work employs the temporal-difference method to assess the action value function $Q(s,a)$ (also known as the critic function) and thereafter enhance the actor's policy by adjusting the policy along the gradient of the function.

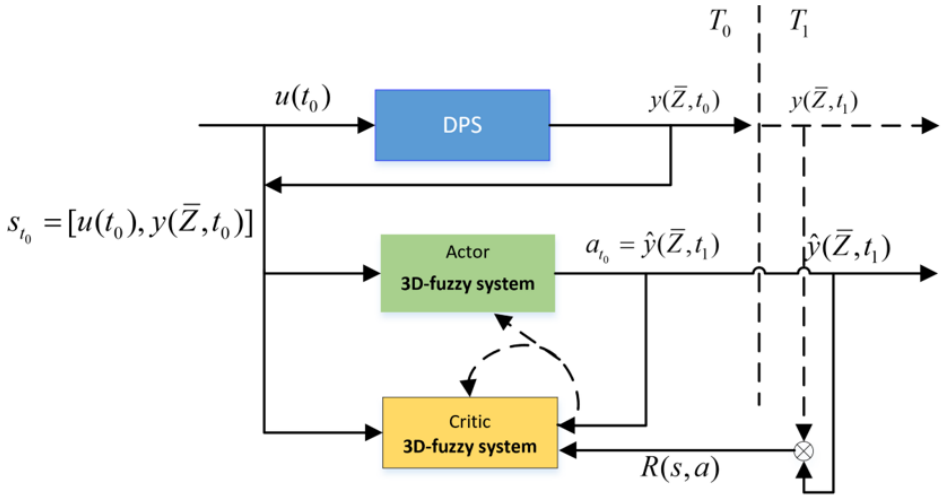

Figure 3. The structure of online 3D-fuzzy modeling using reinforcement learning.

3.2. 3D Fuzzy Deterministic Policy Gradient Reinforcement Learning for DPS Online Modeling

Given the MDP model of the DPS modeling problem, the fusion of reinforcement learning and the 3D fuzzy system produces a novel 3D fuzzy reinforcement learning algorithm for DPS online modeling.

In order to describe the 3D fuzzy reinforcement learning algorithm well, we first introduce the principle of the deterministic policy gradient algorithm and 3D fuzzy modeling.

3.2.1. Deterministic Policy Gradient Algorithm

Deterministic policy gradient (DPG) is a specific type of policy gradient method that aims to optimize $J(\theta)$. The fundamental concept underlying the DPG is to modify the policy parameters θ in accordance with the gradient $\nabla_\theta J(\mu_\theta)$ of the performance measure. The DPG typically relies on the actor–critic architecture, as depicted in Figure 4.

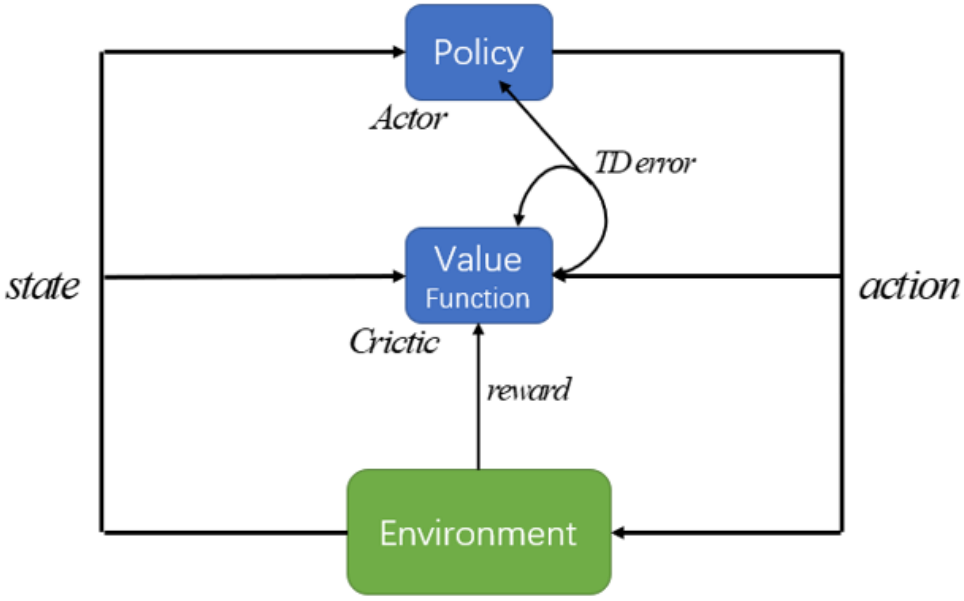

Figure 4. The actor–critic structure.

The deterministic policy gradient theorem examines a policy function $\mu_\theta(s) = \mu(s|\theta)$ that is deterministic and has a parameter vector θ for the policy. The term 'deterministic' indicates that the action is chosen based on a deterministic policy $\mu_\theta : S- > A$ with parameter $\theta \in R^n$, rather than a policy represented by a parametric probability distribution, which is known as SPG. The existence of the DPG is demonstrated in Reference [45] and the DPG theorem is established.

Assuming that the MDP fulfills the conditions $\mu_\theta(s)$, $\nabla_\theta \mu_\theta(s)$, $r(s,a)$ and $\nabla_a r(s,a)$, which are continuous in all parameters and variables s, a, s', it can be inferred that the conditions $\nabla_\theta u(s|\theta)$ and $\nabla_a Q^\mu(s,a)$ also exist and consequently, the DPG exists. Then:

$$\nabla_\theta J(\mu_\theta) = E_{s\sim\rho^\mu}[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a)|_{a=\mu_\theta(s)}] \tag{8}$$

where $Q^\mu(s,a)$ represents the action value function , $\rho^\mu(s)$ represents the discounted state distribution (analogous to the stochastic case) and $J(\mu_\theta)$ represents the performance objective. These variables are defined with regard to a deterministic policy μ and parameter θ. Empirical evidence shows that the DPG algorithm can achieve superior performance compared to stochastic algorithms when dealing with high-dimensional action spaces [45]. Additionally, DPG is able to circumvent the challenges associated with integrating throughout the entire action space.

3.2.2. 3D Fuzzy Modeling

3D fuzzy modeling [21] is an innovative intelligent modeling method designed for nonlinear DPSs in recent years. This method presents a novel fuzzy model equipped to handle and articulate time/space-coupled information. The 3D fuzzy model is distinct from conventional fuzzy models as it is based on the concept of 3D fuzzy sets [21].

As demonstrated in Figure 5, the 3D fuzzy model provides a cohesive structure for effectively combining time–space separation with time–space reconstruction. Therefore, it can be inferred that DPS modeling is achievable regardless of the necessity of model reduction within the context of the 3D fuzzy model.

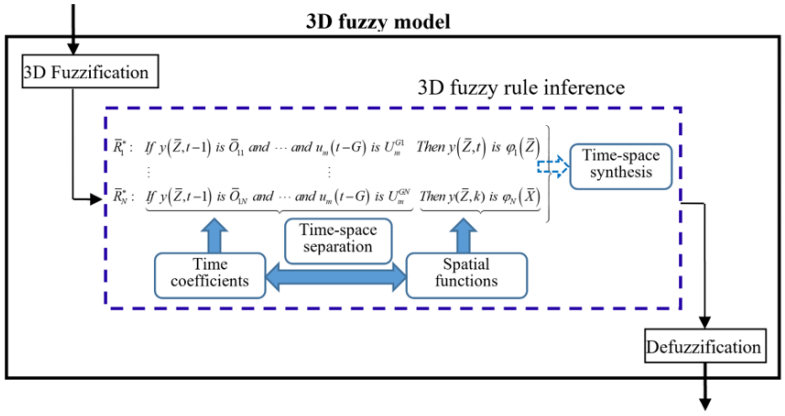

Figure 5. Framework of 3D fuzzy modeling.

The 3D fuzzy rule in Figure 5 is rewritten as in Equation (9).

$$\begin{cases} \bar{R}_1^*: \ If\ y(\bar{Z},t-1)\ is\ \bar{O}_{11}\ and\ \cdots\ and\ u_m(t-G)\ is\ U_m^{G1}\ Then\ y(\bar{Z},t)\ is\ \varphi_1(\bar{Z}) \\ \vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ \bar{R}_N^*: \ If\ y(\bar{Z},t-1)\ is\ \bar{O}_{1N}\ and\ \cdots\ and\ u_m(t-G)\ is\ U_m^{GN}\ Then\ y(\bar{Z},t)\ is\ \varphi_N(\bar{Z}) \end{cases} \tag{9}$$

where $\varphi_l(\bar{Z})$ denotes a space base function, $\bar{O}_{sl}$ denotes a 3D fuzzy set, U_g^{hl} denotes a conventional fuzzy set, $h = 1,2,...,G, g = 1,2,...,m, s = 1,2,...,K, l = 1,2,...,N$, N indicates the quantity of fuzzy rules.

The preceding component in $\bar{R}_l^*$ is utilized to calculate temporal coefficients, while the resulting component in $\bar{R}_l^*$ is utilized to depict space functions. The rule $\bar{R}_l^*$ intrinsically accomplishes the role of separating time and space, similar to the traditional modeling of time and space in DPS. The process of time–space reconstruction is achieved by combining active 3D fuzzy rules. It has been shown that the 3D fuzzy model has the ability to approximate any function universally [46].

Like the neural network, the 3D fuzzy model also has the layer structure shown in Figure 6.

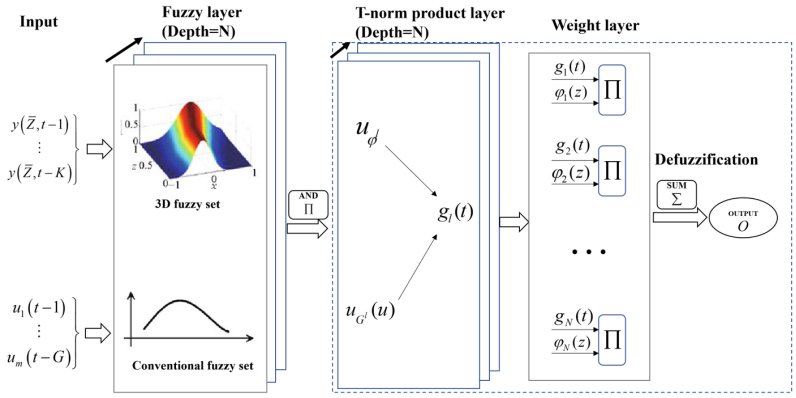

Figure 6. Framework of 3D fuzzy modeling.

The 3D fuzzy model is presumed to possess two distinct categories of inputs. One is the measured space–time coupled data $[y(\bar{Z},t-1), y(\bar{Z},t-2), ..., y(\bar{Z},t-K)]$ with $y(\bar{Z},t) = [y(f_1,t), \cdots, y(f_p,t)]$; the other is traditional data $u = [u_1(t-1), \ldots, u_m(t-G)]$. Within the fuzzy layer, there exist conventional fuzzy sets and 3D fuzzy sets that correspond to two distinct types of inputs. Suppose a Gaussian-type membership function is

used. The "and" operation of multiple 3D fuzzy memberships and the "and" operation of multiple conventional memberships of the inputs are shown as Equations (10) and (11).

$$\mu_{\varphi^l} = \sum_{j=1}^{p} a_j \prod_{i=1}^{K} \exp(-((y(f_j, t-i) - c_{ij}^l)/\sigma_{ij}^l)^2) \tag{10}$$

$$\mu_{G^l} = \prod_{i=1}^{m} \prod_{k=1}^{G} \exp(-((u_i(t-k) - d_{ik}^l)/\delta_{ik}^l)^2) \tag{11}$$

where c_{ij}^l and σ_{ij}^l denote the centroid and spread of the Gaussian 3D fuzzy set $\bar{O}_{il}$ at the jth sensor position; d_{ik}^l and δ_{ik}^l represent the centroid and spread of the conventional Gaussian fuzzy set U_i^{kl}; a_j represents the weight of the jth sensor position.

In the t-norm product layer, the active intensity of the rule is determined via combining the membership of each input in each rule, as in Equation (12).

$$g_l(t) = \mu_{\varphi^l} \mu_{G^l} \tag{12}$$

In the weight layer, the active rules are normalized and output is given as Equation (13).

$$O = \frac{\sum_{i=1}^{N} g_l(t) \varphi_l(\bar{Z})}{\sum_{i=1}^{N} g_l(t)} \tag{13}$$

where O represents the output generated by the 3D fuzzy model. Weight layer is known as the weight average defuzzification method. Up to now, 3D fuzzy modeling belongs to the offline category. In the next subsection, we will introduce the online 3D fuzzy modeling method based on reinforcement learning in detail.

3.2.3. 3DFDPG Modeling Method

The fusion of deterministic policy gradient algorithm and 3D fuzzy system generates 3D fuzzy deterministic policy gradient reinforcement (3DFDPG) learning. When it is used as DPS online modeling, we call it 3DFDPG modeling for short. Figure 7 illustrates the detailed structural framework of 3DFDPG modeling.

As seen in Figure 7, the environment is composed by a DPS. $T(t)$ is the DPS's output at time step t, i.e., $T(t) = y(\bar{Z}, t)$. $u(t)$ is the DPS's input. The state s_t, action a_t, reward r_t and next state s_{t+1} of the environment are set as $s_t = (u(t), y(\bar{Z}, t))$, $a_t = \hat{y}(\bar{Z}, t+1)$, $r_t = -\sqrt{(\hat{y}(\bar{Z}, t+1) - y(\bar{Z}, t+1))^2}$ and $s_{t+1} = (u(t+1), y(\bar{Z}, t+1))$, respectively. The 3DFDPG algorithm utilizes an actor–critic structure as its foundation. The actor receives an environmental state as input and generates an action to be applied to the environment. The environment provides a subsequent state and a reward based on the action taken. The critic receives the actor's action and the current state of the environment as input and produces an action value (also known as Q-value) for the specific state–action combination. The critic's parameters are updated using the TD error. On the other hand, the actor's parameters are updated by moving in the direction of the gradient of the action value function. Reinforcement learning may be prone to instability or be divergent when the action value function is represented by a nonlinear function approximator. To address this issue, the use of a replay buffer and target value method [38] is recommended.

Online DPS modeling is a continuous problem where the ongoing interaction between the actor and the environment is perpetual, lacking any termination states; therefore, the average reward setting [47] is needed. The average reward rate is defined as Equation (14).

$$\begin{aligned} R(\mu) &= \lim_{h\to\infty} \frac{1}{h} \sum_{t=1}^{h} \mathbb{E}[R_t | S_0, A_{0:t-1\sim\pi}] \\ &= \lim_{t\to\infty} \mathbb{E}[R_t | S_0, A_{0:t-1\sim\pi}] \\ &= \sum_{s\in S} \tau(s) \sum_{a\in A} \mu(a|s) \sum_{s'\in S, r} p(s', r|s, a) r \end{aligned} \tag{14}$$

where τ represents a stable distribution given the condition of μ, S is a set of states, A is a set of actions, p: $S \times A \times S \times R \to [0, 1]$ is the state transition probability which implies the dynamics of the environment and r is the reward at every time step. In the average reward setting, returns are defined in terms of differences between rewards and the average reward, as in Equation (15).

$$G_t \doteq R_{t+1} - R(\mu) + R_{t+2} - R(\mu) + R_{t+3} - R(\mu) + \cdots \tag{15}$$

where G_t is the long time return, R_{t+1} is the reward at time step $t+1$ and $R(\mu)$ is the average reward under the policy μ.

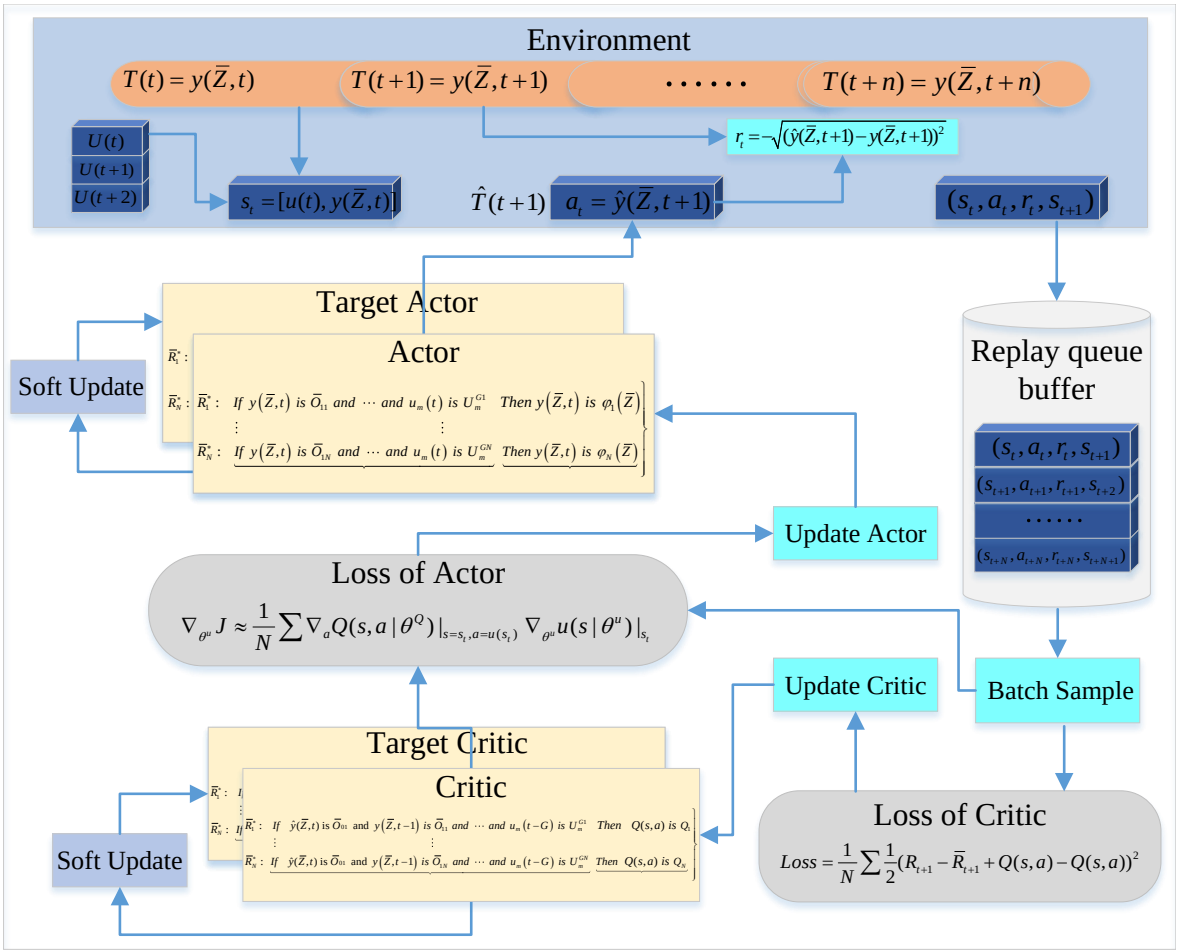

Figure 7. Structural framework of 3DFDPG in detail.

Then, the action value functions $Q(s,a)$ for all $s \in S, a \in A$ are defined as in Equation (16).

$$\begin{aligned} Q(s,a) &= \mathbb{E}_{\mu}[G_t|S_t = s, A_t = a] \\ &= \mathbb{E}[R_{t+1} - R(\mu) + G_{t+1}|S_t = s, A_t = a] \\ &= \mathbb{E}[R_{t+1} - R(\mu) + Q(S_{t+1}, A_{t+1})|S_t = s, A_t = a] \\ &= \sum_{a} \mu(a|s) \sum_{s',r} p(s', r|s, a)[r - R(\mu) + v(s')] \end{aligned} \tag{16}$$

The Bellman equation of the action value function $Q(s,a)$ for all $a \in A, s \in S$ is defined as Equation (17).

$$Q(s,a) = \mathbb{E}[R_{t+1} - R(\mu) + Q(S_{t+1}, A_{t+1})|S_t = s, A_t = a] \tag{17}$$

Then, the TD error is defined as Equation (18).

$$\delta_t \doteq R_{t+1} - \bar{R}_{t+1} + \hat{Q}(S_{t+1}, A_{t+1}) - \hat{Q}(S_t, A_t) \tag{18}$$

where $\bar{R}_{t+1}$ is an estimate at time $t+1$ of the average reward $R(\mu)$ and $\hat{Q}(S_t, A_t)$ is an estimate at time t of the action value function.

The average reward is updated by a soft style as in Equation (19).

$$\bar{R}_{t+1} = \beta \bar{R}_t + (1 - \beta)\delta_t \tag{19}$$

where β represents the update factor.

The mathematical description of the actor and critic is shown below. The actor function $\mu(s)$'s structure is constructed as in Equation (9), which is rewritten as Equation (20)

$$\begin{cases} \bar{R}_1^* : \ If\ y(\bar{Z}, t-1)\ is\ \bar{O}_{11}\ and\ \cdots\ and\ u_m(t-G)\ is\ U_m^{G1}\ Then\ y(\bar{Z}, t)\ is\ \varphi_1(\bar{Z}) \\ \vdots \qquad\qquad\qquad\qquad\qquad \vdots \\ \bar{R}_N^* : \ If\ y(\bar{Z}, t-1)\ is\ \bar{O}_{1N}\ and\ \cdots\ and\ u_m(t-G)\ is\ U_m^{GN}\ Then\ y(\bar{Z}, t)\ is\ \varphi_N(\bar{Z}) \end{cases} \tag{20}$$

If the Gaussian-type member function, singleton 3D fuzzification, average defuzzification and product t-norm are selected for a 3D fuzzy model, the following nonlinear mathematical statement is provided as Equation (21).

$$\begin{aligned} \mu(s) &= \hat{y}(\overline{Z}, t) \\ &= \frac{\sum_{i=1}^{N} \left\{ \begin{array}{l} \sum_{j=1}^{p} a_j^l \prod_{i=1}^{K} \exp(-((y(f_j, t-i) - c_{ij}^l)/\sigma_{ij}^l)^2) \\ \times \prod_{i=1}^{m} \prod_{k=1}^{G} \exp(-((u_i(t-k) - d_{ik}^l)/\delta_{ik}^l)^2) \end{array} \right\} \varphi_l(\overline{Z})}{\sum_{i=1}^{N} \left\{ \begin{array}{l} \sum_{j=1}^{P} a_j^l \prod_{i=1}^{J} \exp(-((y(f_j, t-i) - c_{ij}^l)/\sigma_{ij}^l)^2) \\ \times \prod_{i=1}^{m} \prod_{k=1}^{G} \exp(-((u_i(t-k) - d_{ik}^l)/\delta_{ik}^l)^2) \end{array} \right\}} \\ &= \frac{\sum_{i=1}^{N} g_l(t) \varphi_l(\bar{Z})}{\sum_{i=1}^{N} g_l(t)} \end{aligned} \tag{21}$$

where

$g_l(t) = \mu_{\varphi^l} * \mu_{G^l}$,

$\mu_{\varphi^l} = \sum_{j=1}^{p} a_j \prod_{i=1}^{K} \exp(-((y(f_j, t-i) - c_{ij}^l)/\sigma_{ij}^l)^2)$,

$\mu_{G^l} = \prod_{i=1}^{m} \prod_{k=1}^{G} \exp(-((u_i(t-k) - d_{ik}^l)/\delta_{ik}^l)^2)$.

$\varphi_l(\bar{Z})$ is the Fourier base function, which can be given as Equation (22).

$$\varphi_l(\bar{Z}) = a_l sin(b_l \bar{Z}) + c_l cos(b_l \bar{Z}) + d_l \tag{22}$$

The action value function $Q(s,a)$ is constructed with the structure as expressed in Equation (23).

$$\begin{cases} \bar{R}_1^*: \ If\ \hat{y}(\bar{Z},t) \text{ is } \bar{O}_{01} \text{ and } y(\bar{Z},t-1)\ is\ \bar{O}_{11}\ and\ \cdots\ and\ u_m(t-G)\ is\ U_m^{G1} \\ Then\ Q(s,a)\ is\ Q_1 \\ \vdots \qquad\qquad\qquad\qquad \vdots \\ \bar{R}_N^*: \ If\ \hat{y}(\bar{Z},t) \text{ is } \bar{O}_{0N} \text{ and } y(\bar{Z},t-1)\ is\ \bar{O}_{1N}\ and\ \cdots\ and\ u_m(t-G)\ is\ U_m^{GN} \\ Then\ Q(s,a)\ is\ Q_N \end{cases} \tag{23}$$

where U_g^{hl} represents a conventional fuzzy set, $\bar{O}_{sl}$ represents a 3D fuzzy set, Q_l represents a constant, $g = 1,2,...,m$, $s = 0,1,...,K$, $l = 1,2,...,N$, $h = 1,2,...,G$ and N indicates the quantity of fuzzy rules.

Similar to the actor function, Equation (24) is provided for the mathematical expression of the action value function $Q(s,a)$.

$$\begin{aligned} Q(s,a) &= \frac{\sum_{l=1}^{N}\left\{\sum_{j=1}^{P} a_j^l \exp(-((\hat{y}(f_j,t)-c_{0j}^l)/\sigma_{0j}^l)^2)\sum_{j=1}^{P} a_j^l \prod_{i=1}^{K}\exp(-((y(f_j,t-i)-c_{ij}^l)/\sigma_{ij}^l)^2) \times \prod_{\mathrm{i}=1}^{m}\prod_{k=1}^{G}\exp(-((u_i(t-k)-d_i^{kl})/\delta_i^{kl})^2)\right\}}{\sum_{l=1}^{N}\left\{\sum_{j=1}^{P} a_j^l \exp(-((\hat{y}(f_j,t)-c_{0j}^l)/\sigma_{0j}^l)^2)\sum_{j=1}^{P} a_j^l \prod_{i=1}^{K}\exp(-((y(f_j,t-i)-c_{ij}^l)/\sigma_{ij}^l)^2) \times \prod_{\mathrm{i}=1}^{m}\prod_{k=1}^{K}\exp(-((u_i(t-k)-d_i^{kl})/\delta_i^{kl})^2)\right\}} Q_l \\ &= \frac{\sum_{i=1}^{N} h_l(t) Q_l}{\sum_{i=1}^{N} h_l(t)} \end{aligned} \tag{24}$$

where

$$\begin{aligned} &h_l(t) = \mu_{d^l}\mu_{\varphi^l}\mu_{G^l}, \\ &\mu_{d^l} = exp(-((\hat{y}(f_i,t)-c_{0j}^l)/\sigma_{0j}^l)^2), \\ &\mu_{\varphi^l} = \sum_{j=1}^{p} a_j \prod_{i=1}^{K}\exp(-((y(f_j,t-i)-c_{ij}^l)/\sigma_{ij}^l)^2), \\ &\mu_{G^l} = \prod_{i=1}^{m}\prod_{k=1}^{G}\exp(-((u_i(t-k)-d_{ik}^l)/\delta_{ik}^l)^2). \end{aligned}$$

3.3. Update Process of Parameters for 3D Fuzzy Systems

For a discrete state–action space with restricted options, the action value of each state may be enumerated, that is, tabular reinforcement learning. In the tabular reinforcement learning, the action value $Q(s,a)$ is updated by the Bellman equation [47]. Then, the policy optimization is completed by taking the maximum action a corresponding to different states. As the action value function $Q(s,a)$ is a continuous function, it is challenging to determine the utmost value when the state action space is continuous. Instead, a straightforward and computationally appealing alternative is to shift the policy in the direction of the gradient of $Q(s,a)$, rather than globally maximizing $Q(s,a)$.

The action value function is specified by the following Equation (25):

$$Q(s,a) = \mathbb{E}[G_t | S_t = s, A_t = a] \tag{25}$$

Equation (25) shows that $Q(s,a)$ is the expected return after taking action a and following strategy μ from state s. The goal of reinforcement learning is to find an optimal strategy to maximize $Q(s,a)$ for each state action pair, which is equivalent to maximizing the total expected return G_t. Equation (26) gives the return G_t as the sum of future rewards:

$$G_t = R_{t+1} - R(\mu) + R_{t+2} - R(\mu) + R_{t+3} - R(\mu) + \cdots + R_{t+N} - R(\mu) \tag{26}$$

where $R_{t+1}, R_{t+2}, R_{t+3}, \cdots$ is the immediate reward for the future time step and $R(\mu)$ is the average reward under strategy μ. Since $R(\mu)$ remains unchanged under the same strategy μ, in order to maximize G_t, it is necessary to maximize every reward $R_{t+1}, R_{t+2}, \cdots, R_{t+N}$ in the future. According to Equation (7), it is established that when the immediate reward R_t reaches its maximum value, the discrepancy between the predicted output of the online 3DFDPG model and the actual system output is zero. If the immediate reward R_t is not at its peak value, then a bigger R_t results in increased model accuracy, but a lower R_t leads to decreased accuracy. Therefore, the 3DFDPG algorithm maximizes the action value function $Q(s,a)$ by updating the strategy μ, which is equivalent to maximizing the expected return G_t by maximizing the immediate reward.

So the performance objective $J(\mu_\theta)$ is defined as in Equation (27).

$$J(\mu_\theta) = E_{s\sim\mu_\theta}[Q(s,a)] \tag{27}$$

By the chain rule, the update of policy's parameters is shown as Equation (28).

$$\theta^{\mu}_{k+1} = \theta^{\mu}_{k} + \alpha \frac{1}{N} \sum \nabla_a Q(s,a|\theta^Q)|_{s=s_t, a=\mu(s_t)} \nabla_{\theta^{\mu}_k} \mu(s|\theta^{\mu})|_{s_t} \tag{28}$$

where θ^{μ}_{k} is the parameters of the policy μ, N is the number of samples and α is the learning rate.

The critic function takes TD(0) as the target; the loss function is shown as Equation (29).

$$Loss = \frac{1}{N} \sum \frac{1}{2} (y_t - Q(s,a)|\theta^Q)^2 \tag{29}$$

where y_t is the $TD(0)$ target defined as Equation (30).

$$y_t = r_t - \bar{R}_t + Q'(s_{t+1}, \mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'}) \tag{30}$$

The update of the action value function's parameters is shown as Equation (31).

$$\theta^{Q}_{k+1} = \theta^{Q}_{k} + \alpha \frac{1}{N} \sum (y_t - Q(s,a)|\theta^Q) \nabla_{\theta^Q} Q(s,a|\theta^Q) \tag{31}$$

where θ^{Q}_{k} is the parameters of the action value function; the detail of the parameter's updating is given in the following subsection.

In this paper, because of the instability that directly uses the 3D fuzzy system as the nonlinear approximators, the target critic and the target actor are adopted. Rather than updating the target parameters by directly copying the original 3D fuzzy model's parameters, Equation (32) is utilized to modify the parameters of the target 3D fuzzy model.

$$\begin{aligned} \theta^{Q'} &= \gamma\theta^{Q} + (1-\gamma)\theta^{Q'} \\ \theta^{\mu'} &= \delta\theta^{\mu} + (1-\delta)\theta^{\mu'} \end{aligned} \tag{32}$$

where γ and δ are the update factors.

3.3.1. The Parameter Updating of the Actor Function in Detail

The actor described in Section 3.2.3 is updated by the performance objective as Equation (27). Equation (27) is subjected to the chain rule, resulting in the derivation of Equation (33).

$$\nabla_\theta J(\mu_\theta) = E[\nabla_a Q(s,a)|_{a=\mu(s)} \nabla_\theta \mu(s)] \tag{33}$$

where $a = (\hat{y}(f_1,t), \hat{y}(f_2,t), \cdots, \hat{y}(f_p,t))$. Then, the partial derivative with respect to a yields Equation (34).

$$\nabla_a Q(s,a) = (\nabla_{\hat{y}(f_1,t)} Q(s,a), \nabla_{\hat{y}(f_2,t)} Q(s,a), \ldots, \nabla_{\hat{y}(f_p,t)} Q(s,a)) \tag{34}$$

$$\nabla_{\hat{y}(f_j,t)} Q(s,a) = \sum_{l=1}^{N} \frac{\partial Q}{\partial h_l(t)} \frac{\partial h_l(t)}{\hat{y}(f_j,t)} \tag{35}$$

From Equation (24), the term $\partial Q / \partial h_l(t)$ and the term $\partial h_l(t) / \hat{y}(f_j,t)$ can be determined by Equations (36) and (37).

$$\frac{\partial Q}{\partial h_l(t)} = \frac{Q_l - Q}{\sum_{i=1}^{N} h_l(t)} \tag{36}$$

$$\frac{\partial h_l(t)}{\hat{y}(f_j,t)} = \mu_{\varphi^l} \mu_{G^l} a_j \exp(-((\hat{y}(f_j,t) - c_{0j}^l)/\sigma_{0j}^l)^2) \frac{-2(\hat{y}(f_j,t) - c_{0j}^l)}{(\sigma_{0j}^l)^2} \tag{37}$$

Substituting Equations (36) and (37) into (35), the gradient of Q in relation to a is given as in Equation (38).

$$\nabla_{\hat{y}(z_j,t)} Q(s,a) = \sum_{l=1}^{N} \frac{Q_l - Q}{\sum_{i=1}^{N} h_l(t)} \mu_{\varphi^l} u_{G^l} a_j \exp(-((\hat{y}(z_j,t) - c_{0j}^l)/\sigma_{0j}^l)^2) \frac{-2(\hat{y}(z_j,t) - c_{0j}^l)}{(\sigma_{0j}^l)^2} \tag{38}$$

According to the same process as above, we can obtain $\partial \mu(s) / \partial \theta^\mu$. θ^μ is the parameter vector of the actor function, described by Equation (39).

$$\theta^\mu = (c_{ij}^l, \sigma_{ij}^l, d_{ik}^l, \delta_{ik}^l, a_l, b_l, c_l, d_l)(j = 1, \ldots, p; i = 1, \ldots, m; l = 1, \ldots, N) \tag{39}$$

Then, the partial derivative of the policy with respect to the parameter θ is shown in Equation (40).

$$\nabla_\theta \mu(s) = (\frac{\partial \mu}{c_{ij}^l}, \frac{\partial \mu}{\sigma_{ij}^l}, \frac{\partial \mu}{d_{ik}^l}, \frac{\partial \mu}{\delta_{ik}^l}, \frac{\partial \mu}{a_l}, \frac{\partial \mu}{b_l}, \frac{\partial \mu}{c_l}, \frac{\partial \mu}{d_l}) \tag{40}$$

where c_{ij}^l and σ_{ij}^l denote the centroid and spread of the Gaussian 3D fuzzy set $\bar{O}_{il}$ at the jth sensor position; d_{ik}^l and δ_{ik}^l represent the centroid and spread of the conventional Gaussian fuzzy set U_i^{kl}, a_l, b_l, c_l, d_l denote the coefficients of the base function in Fourier space.

From Equations (21) and (22), we can derive the partial derivative of the policy $\frac{\partial \mu}{c_{ij}^l}$, $\frac{\partial \mu}{\sigma_{ij}^l}$, $\frac{\partial \mu}{d_{ik}^l}$, $\frac{\partial \mu}{\delta_{ik}^l}$, $\frac{\partial \mu}{a_l}$, $\frac{\partial \mu}{b_l}$, $\frac{\partial \mu}{c_l}$, $\frac{\partial \mu}{d_l}$, shown as Equations (41)–(48).

$$\frac{\partial \mu}{c_{ij}^l} = \frac{\varphi_l(\bar{Z}) - \mu(s)}{\sum_{i=1}^{N} g_l(t)} u_{G^l} a_j exp(-((y(f_j,t-1) - c_{ij}^l)/\sigma_{ij}^l)^2) \frac{2(y(f_j,t-1) - c_{ij}^l)}{(\sigma_{ij}^l)^2} \tag{41}$$

$$\frac{\partial \mu}{\sigma_{ij}^l} = \frac{\varphi_l(\bar{Z}) - \mu(s)}{\sum_{i=1}^{N} g_l(t)} \mu_{G^l} a_j \prod_{i=1}^{K} exp(-((y(f_j,t-i) - c_{ij}^l)/\sigma_{ij}^l)^2) \frac{2(y(f_j,t-i) - c_{ij}^l)^2}{(\sigma_{ij}^l)^3} \tag{42}$$

$$\frac{\partial \mu}{d_{ik}^{l}} = \frac{\varphi_l(\bar{Z}) - \mu(s)}{\sum_{i=1}^{N} g_l(t)} \mu_{\varphi^l} \prod_{i=1}^{m} \prod_{k=1}^{G} exp(-((u_i(t-k) - d_{ik}^{l})/\delta_{ik}^{l})^2) \frac{2(u_i(t-k) - \delta_{ik}^{l})}{(\delta_{ik}^{l})^2} \tag{43}$$

$$\frac{\partial \mu}{\delta_{ik}^{l}} = \frac{\varphi_l(\bar{Z}) - \mu(s)}{\sum_{l=1}^{N} g_l(t)} u_{\varphi^l} \prod_{i=1}^{m} exp(-((u_i(t-1) - d_{ik}^{l})/\delta_{ik}^{l})^2) \frac{2(u_i(t-1) - d_{ik}^{l})^2}{(\delta_{ik}^{l})^3} \tag{44}$$

$$\frac{\partial \mu}{a_l} = \frac{g_l(t)}{\sum_{l=1}^{N} g_l(t)} sin(b_l \bar{Z}) \tag{45}$$

$$\frac{\partial \mu}{b_l} = \frac{g_l(t)}{\sum_{l=1}^{N} g_l(t)} a_l \bar{Z} cos(b_l \bar{Z}) - c_l \bar{Z} sin(b_l \bar{Z}) \tag{46}$$

$$\frac{\partial \mu}{c_l} = \frac{g_l(t)}{\sum_{l=1}^{N} g_l(t)} cos(b_l \bar{Z}) \tag{47}$$

$$\frac{\partial \mu}{d_l} = \frac{g_l(t)}{\sum_{l=1}^{N} g_l(t)} \tag{48}$$

3.3.2. The Updating of the Parameters of the Critic Function in Detail

The critic function adopts the same structure as the actor function, as shown in Figure 6. Different to the actor function's performance objective, the loss function of the critic function is as in Equation (18). To describe this clearly, here we write the equation again.

$$\theta_{k+1}^{Q} = \theta_{k}^{Q} + \alpha \frac{1}{N} \sum (y_t - Q(s,a)|\theta^Q) \nabla_{\theta^Q} Q(s,a|\theta^Q) \tag{49}$$

where, θ^Q is the parameter vector of the critic function. N is the sample batch size. α is the learning rate. y_t is the $TD(0)$ target defined as Equation (30), and θ^Q is defined as Equation (50).

$$\theta^Q = (c_{0j}^{l}, \sigma_{0j}^{l}, c_{ij}^{l}, \sigma_{ij}^{l}, d_{ik}^{l}, \delta_{ik}^{l}, Q_l)(j = 1, \ldots, p; i = 1, \ldots, m; l = 1, \ldots, N) \tag{50}$$

Then the partial derivative of $Q(s,a)$ with respect to θ^Q is shown in Equation (51); the detailed formulas are shown in Equations (52)–(58).

$$\nabla_{\theta^Q} Q(s,a|\theta^Q) = (\frac{\partial Q}{c_{0j}^{l}}, \frac{\partial Q}{\sigma_{0j}^{l}}, \frac{\partial Q}{c_{ij}^{l}}, \frac{\partial Q}{\sigma_{ij}^{l}}, \frac{\partial Q}{d_{ik}^{l}}, \frac{\partial Q}{\delta_{ik}^{l}}, \frac{\partial Q}{Q_l}) \tag{51}$$

$$\frac{\partial Q}{c_{0j}^{l}} = \frac{Q_l - Q}{\sum_{i=1}^{N} h_l(t)} \mu_{G^l} a_j \exp(-((\hat{y}(f_j,t) - c_{0j}^{l})/\sigma_{0j}^{l})^2) \prod_{i=1}^{K} exp(-((y(f_j,t-i) - c_{ij}^{l})/\sigma_{ij}^{l})^2) \frac{2(\hat{y}(z_j,t) - c_{0j}^{l})}{(\sigma_{0j}^{l})^2} \tag{52}$$

$$\frac{\partial Q}{\sigma_{0j}^{l}} = \frac{Q_l - Q}{\sum_{i=1}^{N} h_l(t)} \mu_{G^l} a_j \exp(-((\hat{y}(f_j,t) - c_{0j}^{l})/\sigma_{0j}^{l})^2) \prod_{i=1}^{K} exp(-((y(f_j,t-i) - c_{ij}^{l})/\sigma_{ij}^{l})^2) \frac{2(\hat{y}(z_j,t) - c_{0j}^{l})^2}{(\sigma_{0j}^{l})^3} \tag{53}$$

$$\frac{\partial Q}{c_{ij}^{l}} = \frac{Q_l - Q}{\sum_{i=1}^{N} h_l(t)} \mu_{G^l} a_j \exp(-((\hat{y}(f_j,t) - c_{0j}^{l})/\sigma_{0j}^{l})^2) \prod_{i=1}^{K} exp(-((y(f_j,t-i) - c_{ij}^{l})/\sigma_{ij}^{l})^2) \frac{2(y(f_j,t) - c_{ij}^{l})}{(\sigma_{ij}^{l})^2} \tag{54}$$

$$\frac{\partial Q}{\sigma_{ij}^l} = \frac{Q_l - Q}{\sum_{i=1}^{N} h_l(t)} \mu_{G^l} a_j \exp(-((\hat{y}(f_j,t) - c_{0j}^l)/\sigma_{0j}^l)^2) exp(-((y(f_j,t-1) - c_{ij}^l)/\sigma_{ij}^l)^2) \frac{2(y(f_j,t) - c_{ij}^l)^2}{(\sigma_{ij}^l)^3} \tag{55}$$

$$\frac{\partial Q}{d_{ik}^l} = \frac{Q_l - Q}{\sum_{i=1}^{N} h_l(t)} \mu_{\varphi^l} \prod_{i=1}^{m} exp(-((u_i(t-1) - d_{ik}^l)/\delta_{ik}^l)^2) \frac{2(u_i(t-1) - d_{ik}^l)}{(\delta_{ik}^l)^2} \tag{56}$$

$$\frac{\partial Q}{\delta_{ik}^l} = \frac{Q_l - Q}{\sum_{i=1}^{N} h_l(t)} \mu_{\varphi^l} \prod_{i=1}^{m} exp(-((u_i(t-1) - d_{ik}^l)/\delta_{ik}^l)^2) \frac{2(u_i(t-1) - d_{ik}^l)^2}{(\delta_{ik}^l)^3} \tag{57}$$

$$\frac{\partial Q}{Q_l} = \frac{h_l(t)}{\sum_{i=1}^{N} h_l(t)} \tag{58}$$

3.3.3. The Process of the Proposed 3DFDPG

The proposed 3DFDPG is executed as follows.

Input: Batch size N, number of 3D fuzzy rules of policy $\mu(s)$ and action value function $Q(s,a)$, replay buffer size N', actor's learning rate α_a and critic's learning rate α_c.

Step 1: Initialize critic $Q(s,a|\theta^Q)$ and policy $\mu(s|\theta^u)$ with parameters $\theta^Q = (c_{0j}^l, \sigma_{0j}^l, c_{ij}^l, \sigma_{ij}^l, d_{ik}^l, \delta_{ik}^l, Q_l)$ and $\theta^\mu = (c_{ij}^l, \sigma_{ij}^l, d_{ik}^l, \delta_{ik}^l, a_l, b_l, c_l, d_l)$ randomly.

Step 2: Initialize target μ' and Q' with parameters $\theta^{\mu'} = \theta^\mu$ and $\theta^{Q'} = \theta^Q$. The average reward $\bar{R} = 0$.

Step 3: Initialize replay buffer RB.

Step 4: Loop forever:

Step 5: Perform action a_t on the DPS and analyze the resulting reward r_t and subsequent state s_{t+1} of the DPS.

Step 6: Preserve the transition (s_t, a_t, r_t, s_{t+1}) to RB.

Step 7: Randomly select a small quantity of N' transitions from RB.

Step 8: Set $y_i = r_i - \bar{R} + Q'(s_{t+1}, \mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'})$.

Step 9: Set $\bar{R} = \beta\bar{R} + (1-\beta)\delta$.

Step 10: Minimize the loss function $Loss = 1/N\sum(y_t - Q(s,a)|\theta^Q)^2$ to update the critic.

Step 11: Utilize the policy gradient that was sampled to update the policy.

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum \nabla_a Q(s,a|\theta^Q)|_{s=s_t, a=\mu(s_t)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s_t}$$

Step 12: If the estimate error is less than a threshold δ, update the policy parameters and directly use the error via stochastic gradient descent.

$$Error = \frac{1}{2}(y(\bar{Z},t) - \hat{y}(\bar{Z},t))^2$$

Step 13: Update the target $\theta^{Q'} = \gamma\theta^Q + (1-\gamma)\theta^{Q'}, \theta^{\mu'} = \delta\theta^\mu + (1-\delta)\theta^{\mu'}$.

Step 14: Output the policy $\mu(s_t|\theta^\mu)$ as the model of the DPS at every time step.

4. Applications

The typical distributed parameter system (RTCVD) depicted in Section 2 is considered to assess the efficacy of the suggested algorithm 3DFDPG. To obtain sufficient dynamic knowledge about the system, the system manipulates the input variable u_1, u_2 and u_3 to

add the interference signal whose amplitude is not more than 10%. Therefore, three input variables with interference signals can be given in the following ways.

$$\begin{aligned} u_1(t) &= u_{10} + \Delta_1 * u_{10} * normrnd(0,1) \\ u_2(t) &= u_{20} + \Delta_2 * u_{20} * normrnd(0,1) \\ u_3(t) &= u_{30} + \Delta_3 * u_{30} * normrnd(0,1) \end{aligned} \tag{59}$$

where u_{10}, u_{20} and u_{30} are the steady-state input when the furnace temperature is 1000K, Δ_1, Δ_2 and Δ_3 are the disturbance amplitudes of u_{10}, u_{20} and u_{30}. In this study, u_{10}, u_{20} and u_{30} are 0.2028, 0.1008 and 0.2245, respectively; Δ_1, Δ_2 and Δ_3 are set to 10%. Eleven sensors are arranged equally along the radial direction. To replicate the effect of noise, eleven sets of separate random noise signals are then included into the data gathered by those eleven sensors. The sample period was set as $\Delta t = 0.1s$ and the experiment lasted for 500 s. The data generated from the experiment are shown in Equation (60).

$$S = \{x^k, y_z^k\} = \{(x_z^k, x_u^k), y_z^k | x_z^k \in R^{11\times 1}, x_u^k \in R^{1\times 3}, y_z^k \in R^{P\times 1}, k = 1, \cdots, 5000\} \tag{60}$$

where,

$x^k = (x_z^k, x_u^k)$;
$x_z^k = [T(\bar{Z}, k-1)]$;
$x_u^k = [u_a(k-1), u_b(k-1), u_c(k-1)]$;
$y_z^k = T(\bar{Z}, k) = [T_f(f_1, k), \cdots, T_f(f_{11}, k)]$.

Unlike offline modeling algorithms, online modeling algorithms use data generated in real-time by the DPS simulation system. At a given time t, the 3DFDPG model predicts the output at time t based on the output and input of the DPS system at time $t-1$, as shown in Equation (61).

$$y(t) = f(y(z, t-1), u(t-1), \theta) \tag{61}$$

The hyperparameters for the 3DFDPG are specified in Table 1.

In this paper, we use Matlab2023b software to realize the simulation of RTCVD, the realization of 3D fuzzy modeling, and the comparison of different modeling methods.

Table 1. The hyperparameters of the 3DFDPG.

Parameters	Value	Description
Mini-batch Size	20	Mini batch Size
Buffer Size	40	Replay buffer size
Num of the Q value function 3D fuzzy rules	20	The number of the Q value 3D fuzzy rules
Num of the policy rules	20	The number of policy 3D fuzzy rules
α_1	0.001	Critic's learning rate
α_2	0.001	Actor's learning rate
β_1	0.9	The hyperparameter for exponential decay in the Adam optimizer
β_2	0.999	The hyperparameter for exponential decay in the Adam optimizer
ϵ	1×10^{-8}	Adam's stability constant

4.1. Simulations

Ideally, an off-line model based on the historical data with sufficient information can show the dynamic characteristics of the actual system well. However, when the dynamic characteristics of the actual system change, such as equipment aging, local equipment damage and large changes in operating conditions, the off-line model cannot express the dynamic characteristics of the change. However, an online model makes up for this

deficiency. In this study, two scenarios of the RTCVD system are investigated to simulate external or internal environmental changes, including change in the system's radiation coefficient σ_0 and change in wafer density w_0.

(1) The parameters σ_0 are changed

Parameters σ_0 are reduced by 10% after 350 s to simulate the external change of the real system environment. To illustrate the modeling performance in response to changes in the DPS's circumstances, we show the 400 data between 330 s and 370 ts.

As shown in Figure 8, when the system's radiation coefficient σ_0 decreases, heat diffusion in the furnace slows down, causing a sudden temperature increase in the system, After a certain period, the system's temperature stabilizes, that is the DPS's actual output $y(\bar{Z}, t)$. As the predicted output $\hat{y}(\bar{Z}, t)$ illustrated in Figure 9, the model established in this paper is able to follow this change in the system. Figure 10 shows that the higher the immediate reward, the higher the model's accuracy, with the peak reaching 0 at around 60 s, indicating that the proposed online modeling method has high modeling efficiency. Figures 11–13 illustrate the real and anticipated values from sensors s_3, s_5 and s_7, respectively, to elucidate the modeling accuracy of 3DFDPG under varying system conditions. It can be seen from the figures that the 3DFDPG model can promptly track changes in the system, proving the effectiveness of the proposed modeling method. Figures 14 and 15 illustrate the prediction error and relative error of the 3DFDPG method, with the prediction error spanning from $[-6, +6]$ and the relative error measuring 0.006. It indicates that the proposed online modeling method has good model accuracy.

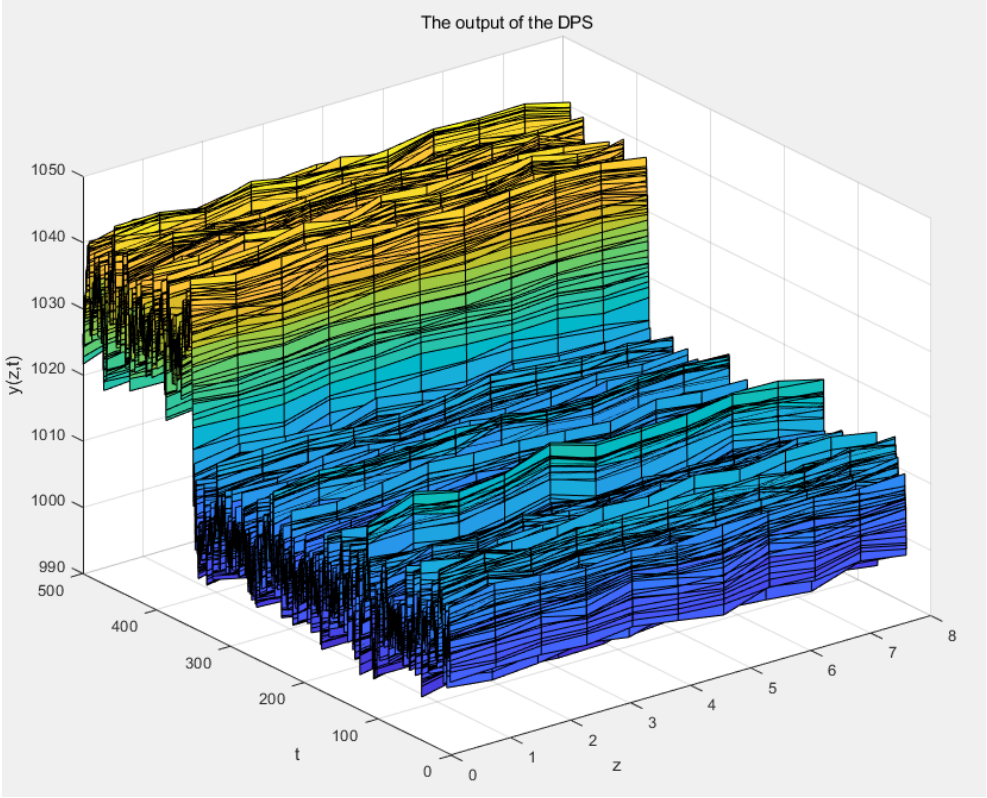

Figure 8. The actual output of the DPS under σ_0 changed. t is the sample time, z is the spatial position and $y(z, t)$ is the DPS's output.

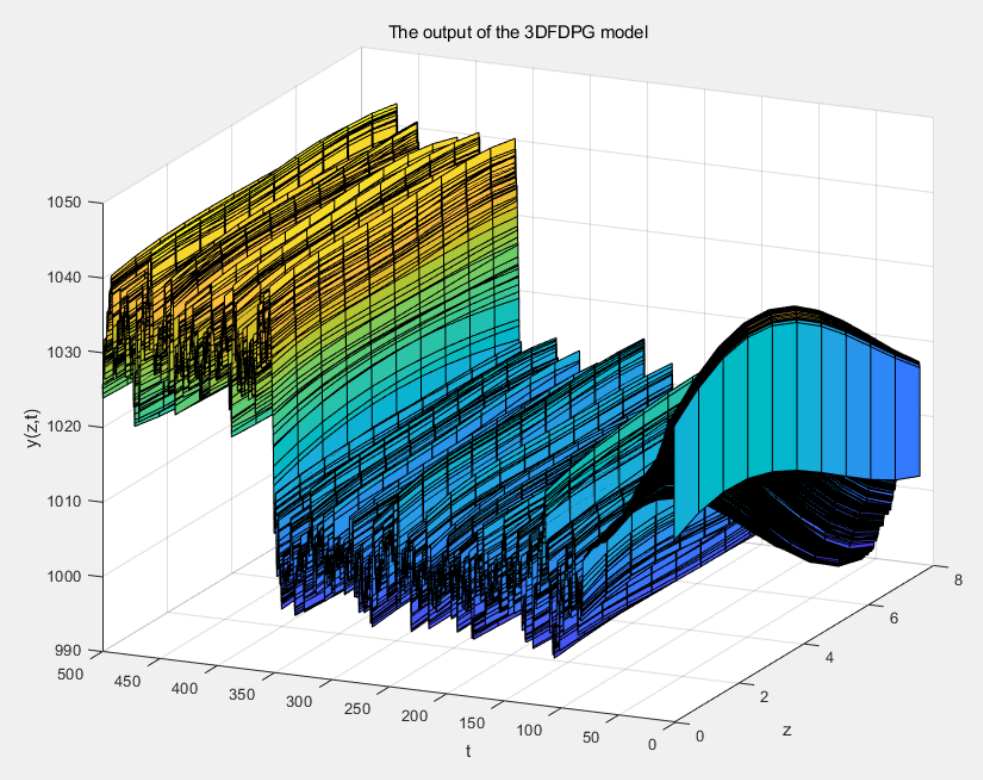

Figure 9. The predict output under σ_0 changed. t is the sample time, z is the spatial position and $y(z,t)$ is the predict output.

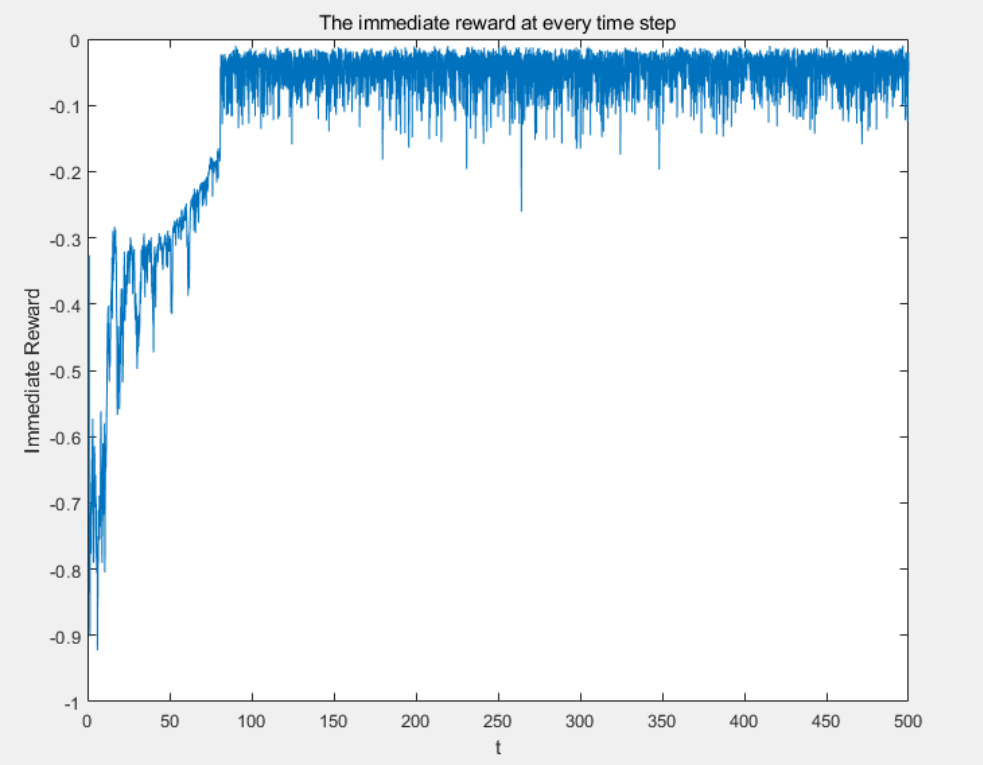

Figure 10. The immediate reward under σ_0 changed. t is the sample time.

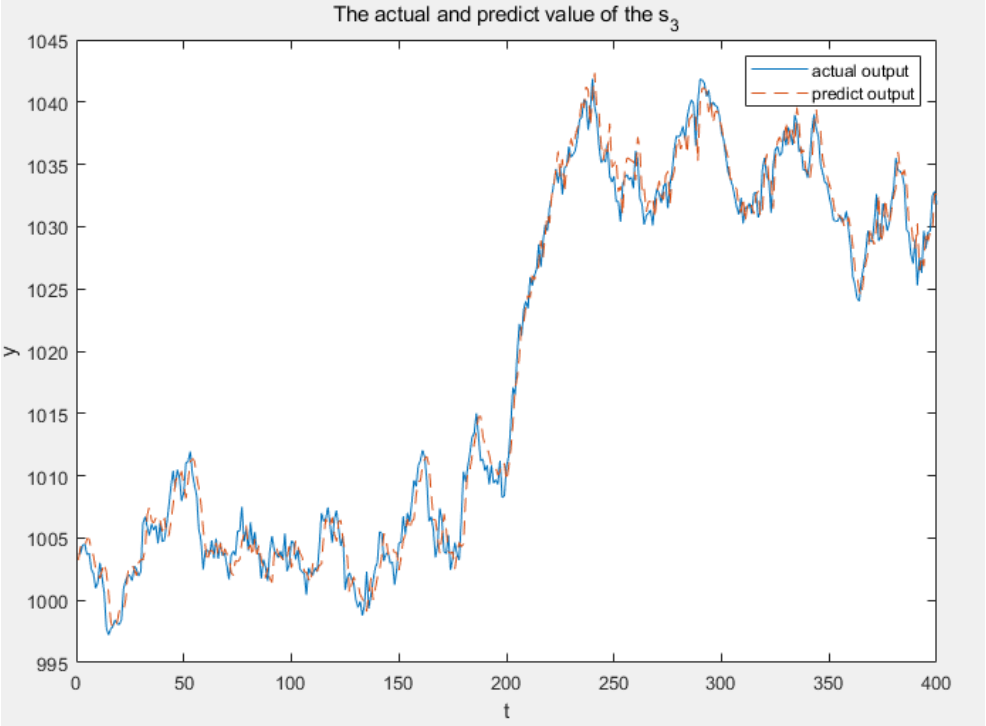

Figure 11. Actual and predict value of S_3 under σ_0 changed. t is the sample time and y is the output of the model and DPS.

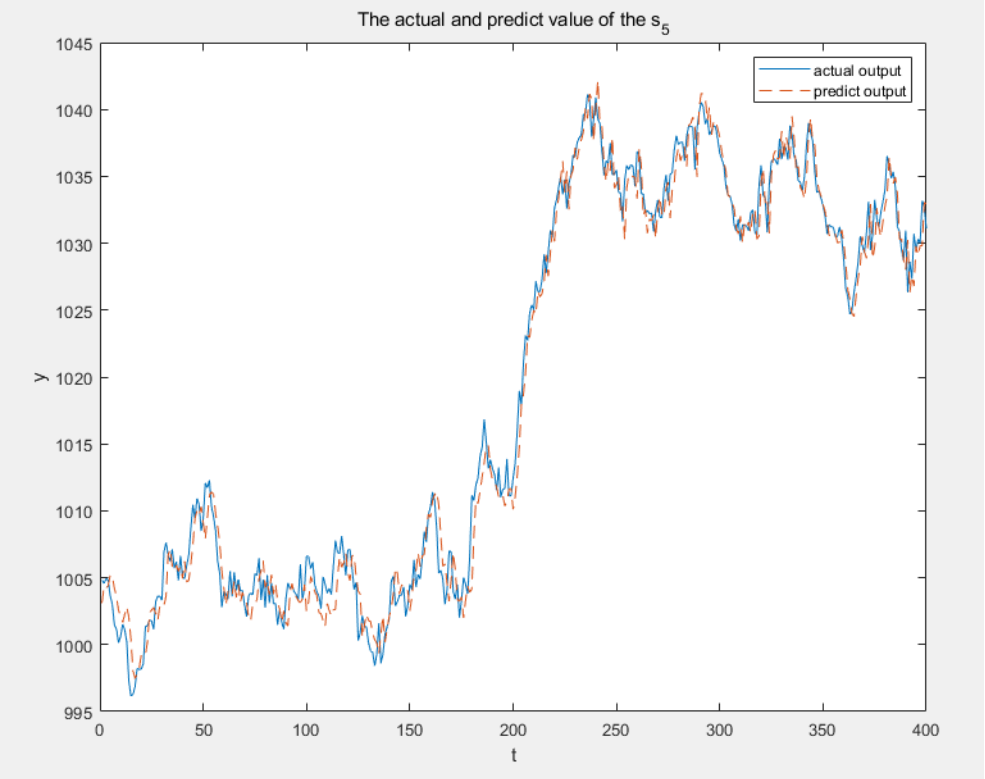

Figure 12. Actual and predict value of S_5 under σ_0 changed. t is the sample time and y is the output of the model and DPS.

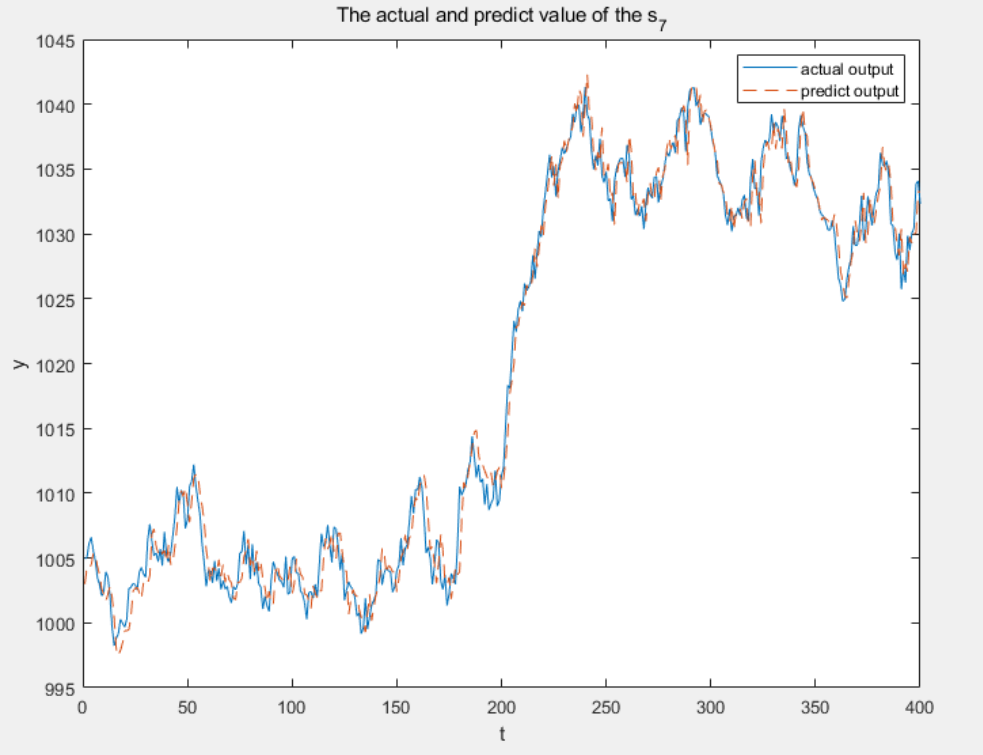

Figure 13. Actual and predict value of S_7 under σ_0 changed. t is the sample time and y is the output of the model and DPS.

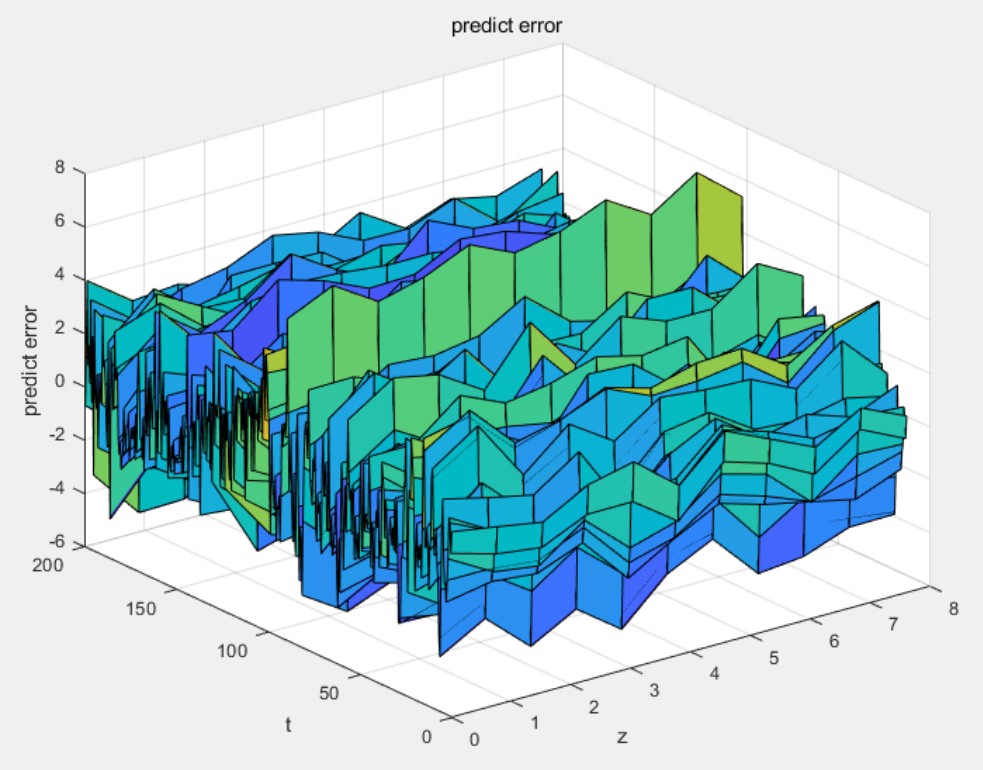

Figure 14. The predict error of the 3DFDPG under σ_0 changed. t is the sample time and z is the spatial position.

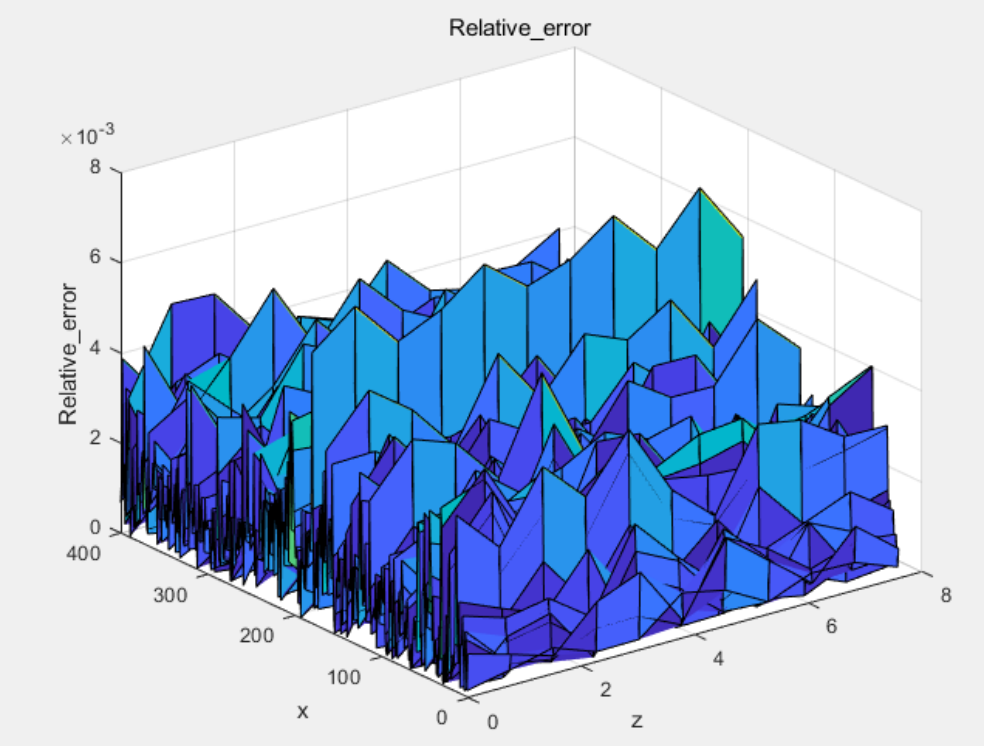

Figure 15. The relative error of the 3DFDPG under σ_0 changed. x is the time step and z is the spatial position.

(2) The parameters w_0 are changed

Parameters w_0 are reduced by 10% after 350s to simulate the internal change of the real system environment. To illustrate the modeling performance in response to changes in the DPS's circumstances, we show the 400 data between 330s and 370s.

As shown in Figure 16, when the wafer density w_0 decreases, the wafer temperature drops, causing a sudden decrease in the system's temperature, which then stabilizes after reaching a certain level. As illustrated in Figure 17, the model established in this paper is able to follow this change in the system. Figure 18 illustrates that an increase in immediate reward correlates with enhanced model accuracy, peaking at 0 around 60 s. This indicates that the proposed online modeling method has excellent modeling efficiency. Figures 19–21 correspondingly illustrate the actual and expected values from sensors s_3, s_5 and s_7. These figures indicates that the 3DFDPG model can swiftly monitor system changes, hence validating the efficacy of the proposed modeling method. Figures 22 and 23 illustrate the prediction error and relative error of the 3DFDPG algorithm, with the prediction error spanning from $[-6, +6]$ and the relative error at 0.006. This indicates that the proposed online modeling method maintains commendable model accuracy despite variations in wafer density w_0.

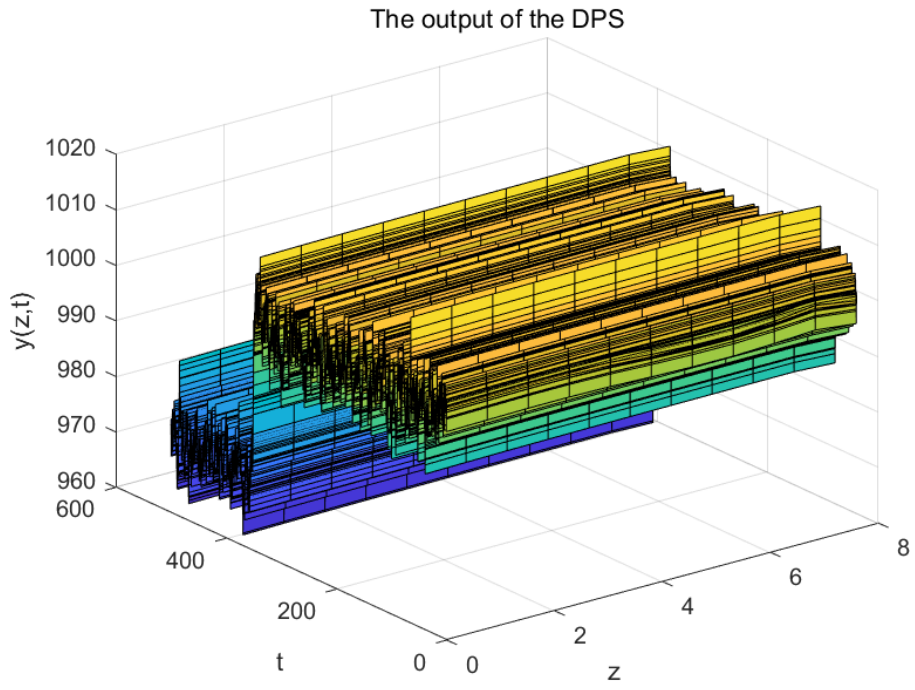

Figure 16. The actual output of the DPS under w_0 changed. t is the sample time, z is the spatial position and $y(z,t)$ is the DPS's output.

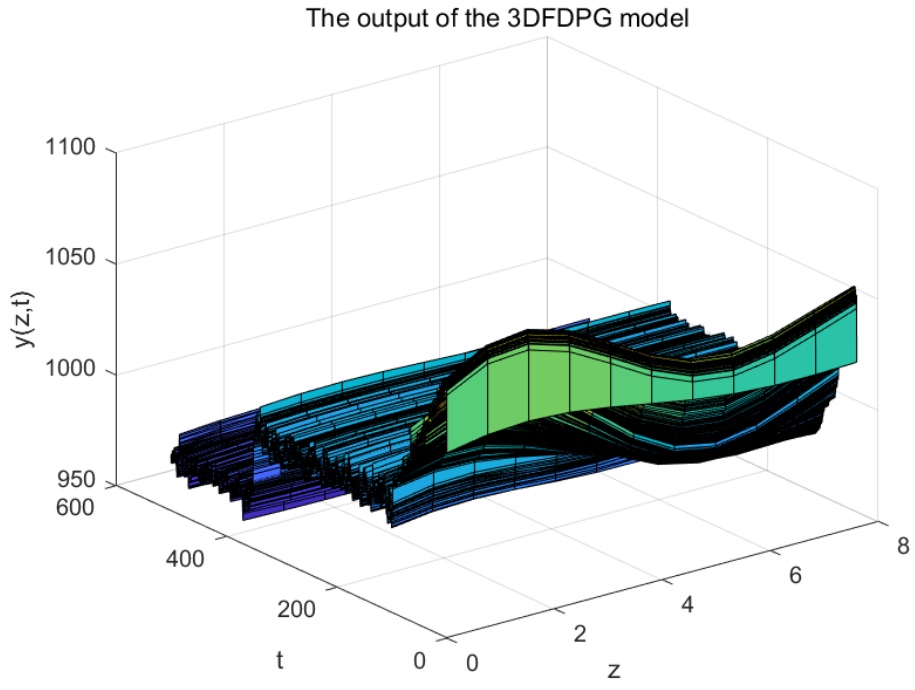

Figure 17. The predict output under w_0 changed. t is the sample time, z is the spatial position and $y(z,t)$ is the predict output.

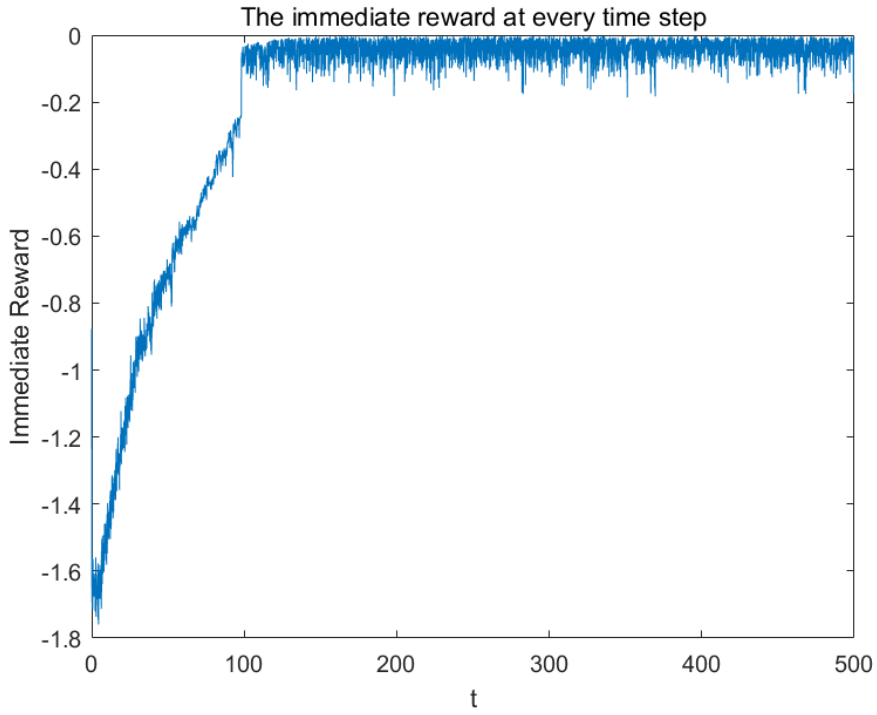

Figure 18. The immediate reward under w_0 changed. t is the sample time.

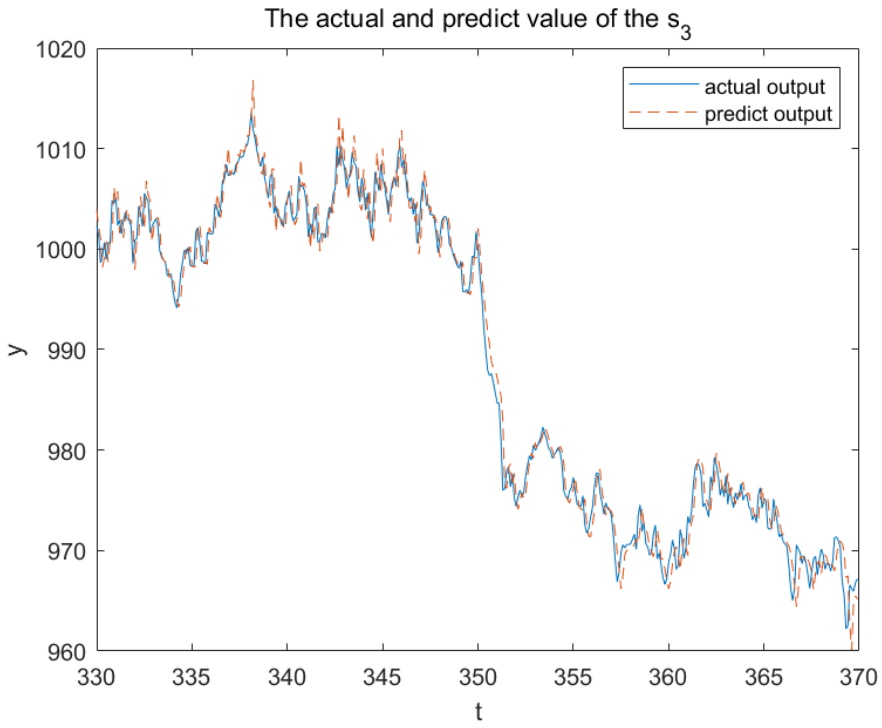

Figure 19. Actual and predict value of S_3 under w_0 changed. t is the sample time and y is the DPS's output.

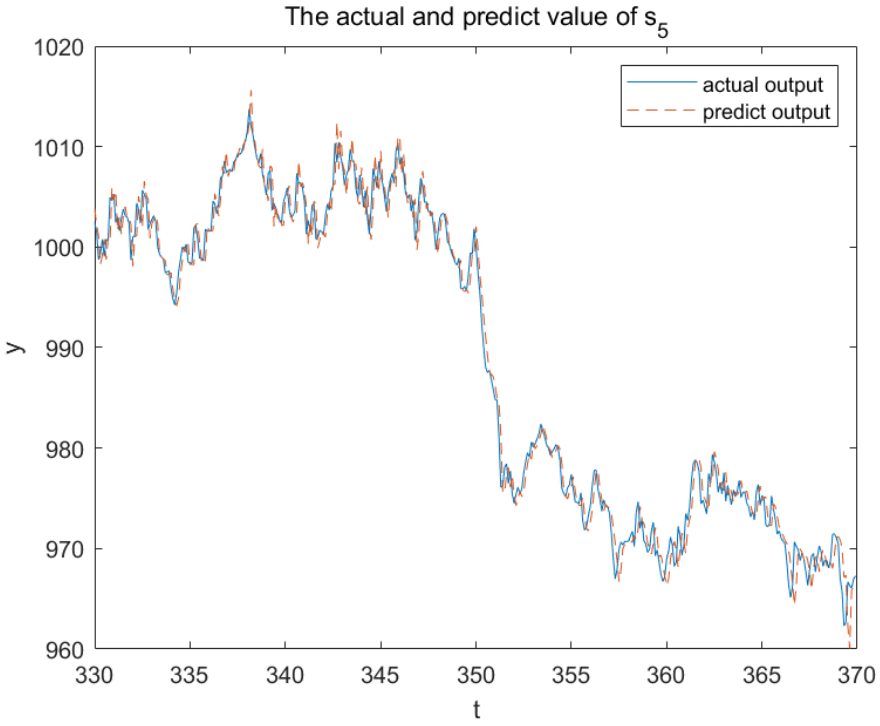

Figure 20. Actual and predict value of S_5 under w_0 changed. t is the sample time and y is the DPS's output.

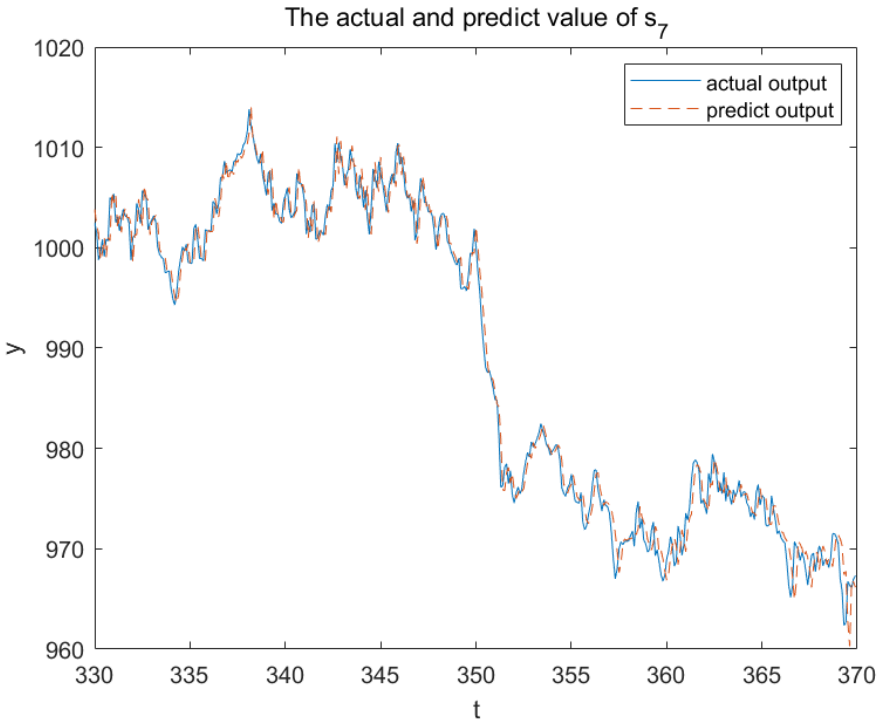

Figure 21. Actual and predict value of S_7 under w_0 changed. t is the sample time and y is the DPS's output.

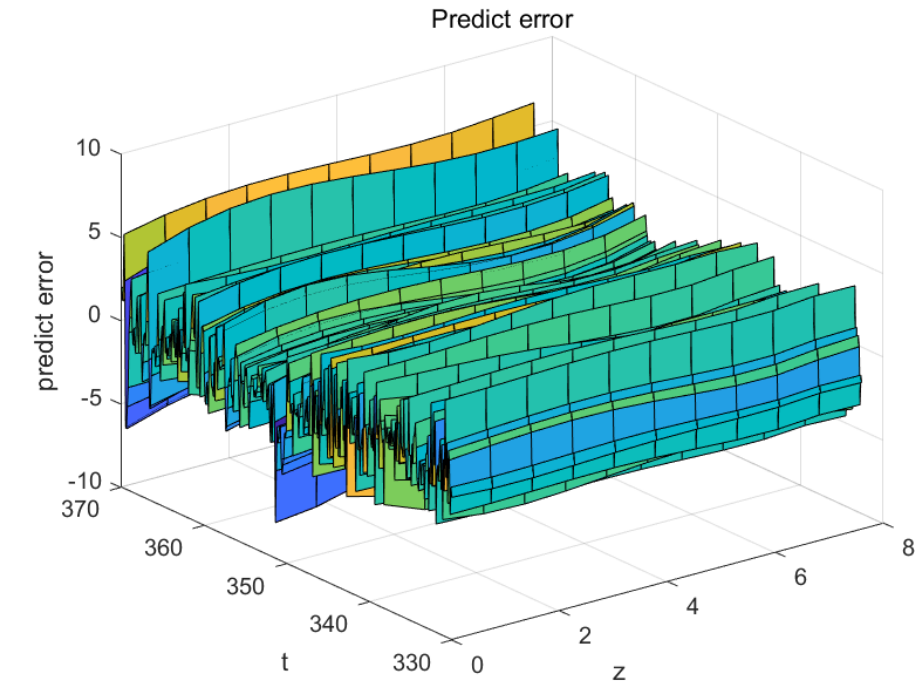

Figure 22. The predict error of the 3DFDPG under w_0 changed. t is the sample time and z is the spatial position.

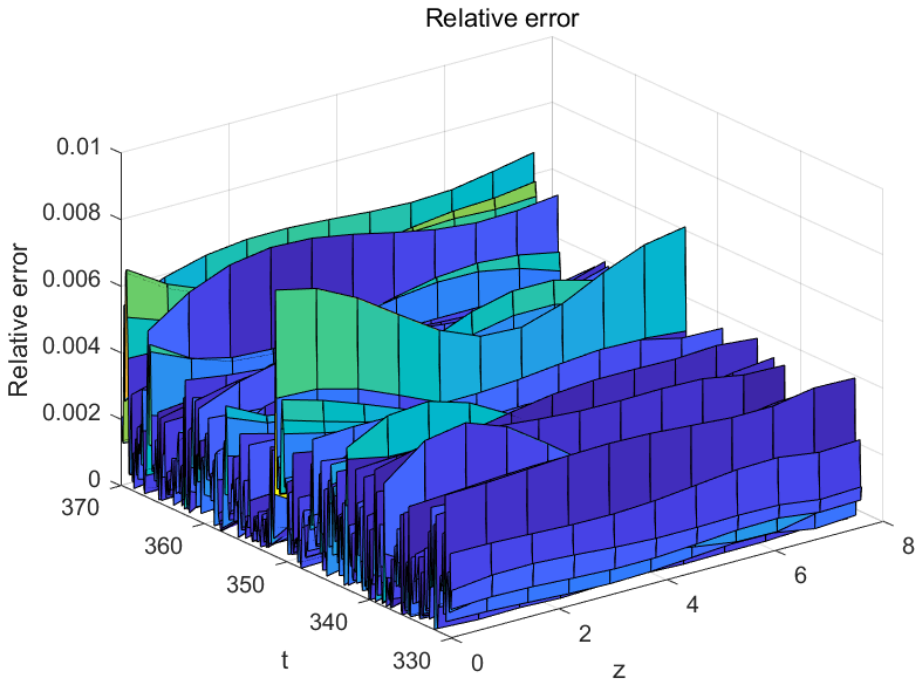

Figure 23. The relative error of the 3DFDPG under w_0 changed. t is the sample time and z is the spatial position.

4.2. Comparison and Analysis

The accuracy of the model is measured by relative error, the time normalized absolute error (TNAE) and the relative L2 standard error (RLNE) as Equations (62), (63) and (64), respectively.

$$Relative \quad error = |\frac{\hat{y}(\bar{Z},t) - y(\bar{Z},t)}{y(\bar{Z},t)}| \times 100\% \tag{62}$$

$$TNAE(\bar{Z}) = \sum |\hat{y}(\bar{Z},t) - y(\bar{Z},t)| / \sum \Delta t \tag{63}$$

$$RLNE(t) = (\int (\hat{y}(\bar{Z},t) - y(\bar{Z},t))^2 d\bar{Z})^{1/2} / (\int y(\bar{Z},t)^2 d\bar{Z})^{1/2} \tag{64}$$

Then, we compared the proposed 3DFDPG with KL-LS-SVM [48] as a traditional offline DPS modeling method and online LS-SVM [28] as a newly proposed online DPS modeling method. Figures 24 and 25 compare the TANE obtained from the two online modeling methods. Figures 26 and 27 compare the RLNE obtained from the two modeling methods.

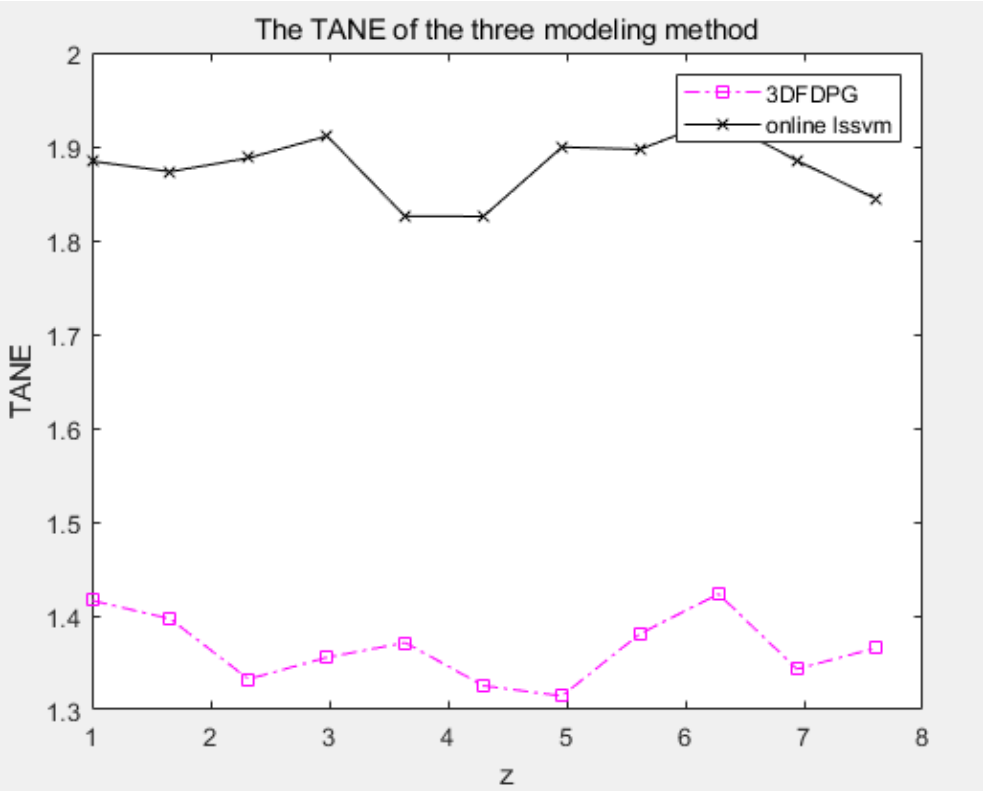

Figure 24. The TANE of the two modeling methods under σ_0 changed. z is the spatial position.

Table 2 presents the mean RMSE and the standard deviation of the three models by changing the random seed and conducting 10 trials. In both cases, the modeling error of the offline model (KL-LS-SVM) is much larger than that of the other two online models. The offline model is unable to cover the data in the new scenario when the DPS's circumstances change. But since the online modeling method uses incremental learning, it can still appropriately assess the DPS in different scenarios. In addition, we conduct t-test and p-value analyses for the two online modeling methods. The T-test calculation equation for the two online models is as follows:

$$T = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{65}$$

where $\bar{X}_i$, s_i and n_i represent the mean of RMSE, the standard deviation of RMSE and the number of samples from the ith online model, respectively, $i = 1$ represents the 3DFDPG model and $i = 2$ represents the online LS-SVM model. Substituting the data from Table 2 into Equation (65) yields the T value. Then, we obtain the p value by consulting the corresponding statistical distribution table, that is T = -35.27, $p < 0.01$ with σ_0 changed and T = -13.47, $p < 0.01$ with w_0 changed. Since the p value is less than 0.05, we reject the null hypothesis and conclude that there is a significant difference between the means of the two groups. This demonstrates that the proposed 3DFDPG model outperforms the Online LS-SVM model under the two application scenarios.

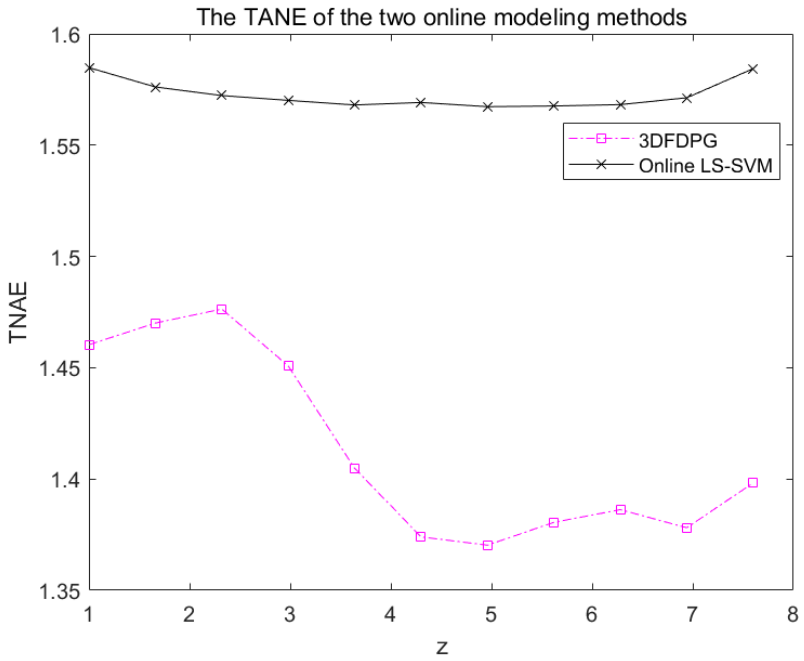

Figure 25. The TANE of the two modeling methods under w_0 changed. z is the spatial position.

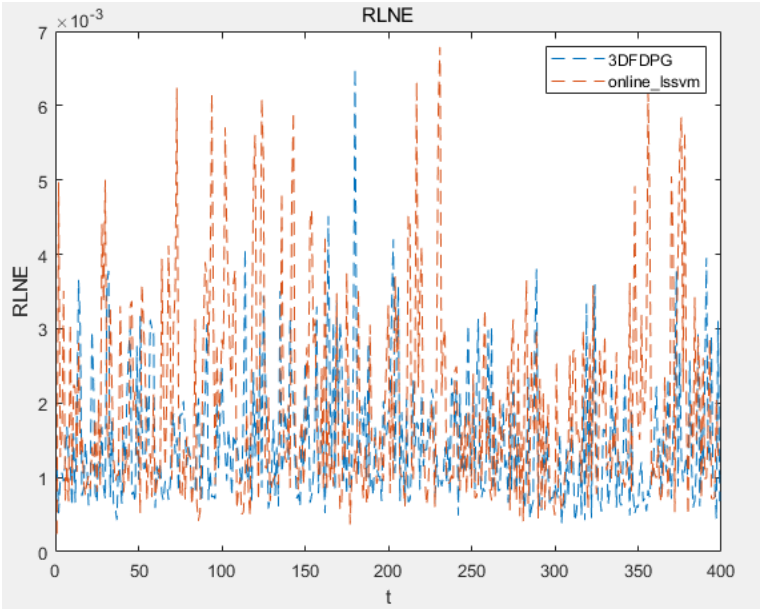

Figure 26. The RLNE of the two modeling methods under σ_0 changed. t is the sample time.

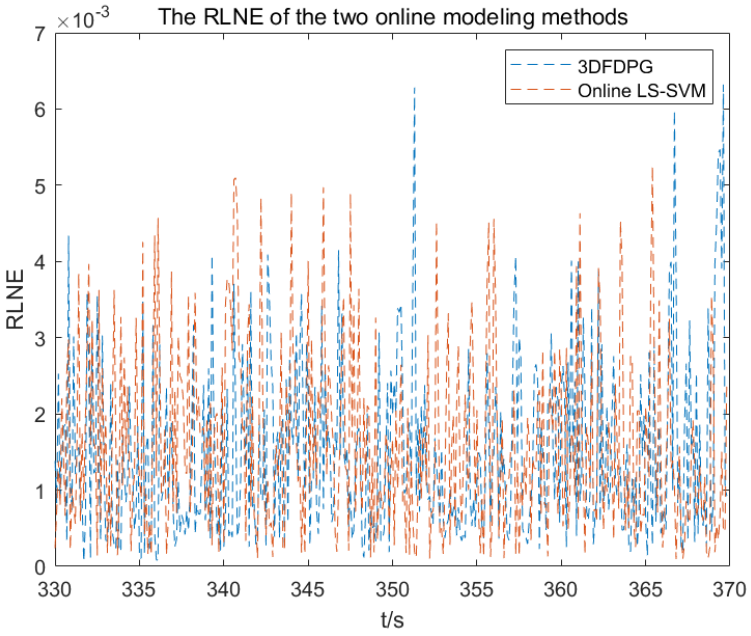

Figure 27. The RLNE of the two modeling methods under w_0 changed. t is the sample time.

Table 2. Mean RMSE and its standard deviation of the three models.

Modeling Method	Mean RMSE with σ_0 Changed	Standard Deviation of RMSE with σ_0 Changed	Mean RMSE with w_0 Changed	Standard Deviation of RMSE with w_0 Changed
KL-LS-SVM	28.4719	0.03	29.9498	0.03
Online LS-SVM	2.31787	0.0417	2.0164	0.0454
3DFDPG	1.7686	0.0550	1.7915	0.0270

TNAE and RLNE represent the prediction errors of the 3DFDPG model and the online LS-SVM model in spatial and temporal dimensions, respectively. As shown in Figures 24–27, in both cases, the errors of the 3DFDPG model are smaller than those of the online LS-SVM model. The reason is that the 3D fuzzy system naturally achieves temporal-spatial separation and temporal-spatial synthesis, allowing for the spatiotemporal modeling of DPS without the need for model reduction. Therefore, the 3DFDPG modeling method outperforms the online LS-SVM model. The online LS-SVM updates its model from an initial offline model. In contrast, the proposed 3DFDPG starts from scratch, relying on the interaction between the agent and the environment, with model parameters being updated adaptively.

5. Conclusions

In this paper, we proposed a 3DFDPG modeling method for online modeling of the DPSs. Reinforcement learning is learning from interactions between an environment and an agent; the 3D fuzzy system has the ability to deal with spatiotemporal data and has the universal approximation property. The reinforcement learning algorithm and the 3D fuzzy system were combined to realize the online learning. The average reward set was utilized to implement the online modeling of the DPS, so that the modeling process can run forever and have the ability to catch the situation change of the DPS. The simulation results confirmed the efficacy of the proposed 3DFDPG algorithm and its exceptional modeling performance is further emphasized by comparison with existing DPS modeling methods.

Due to practical constraints, this paper applied the proposed 3DFDPG modeling method to an RTCVD simulation system, aiming to simulate the real system as closely as possible by varying parameters and adding disturbances. When applied to the real system, the benefits are as follows:

1. It develops a very accurate model for the actual system to serve as a reference for system optimization, control and cost reduction.
2. This work proposes an online modeling technique capable of swiftly indicating internal or external component failures inside the system, hence furnishing engineers with accurate fault-localization data.
3. The proposed online modeling method can model the real system from scratch without human intervention, facilitating system modeling.

Although the proposed online modeling algorithm achieved good results, there are two limitations, which need to be solved in future work. Firstly, the 3D fuzzy system used in this paper has a fixed structure and lacks flexibility. In future work, a dynamic variable structure 3D fuzzy system will be investigated. Secondly, the reinforcement learning algorithm based on DPG used in this paper is somewhat insufficient in terms of action exploration and may become trapped in local optimal policies. In future work, the feasibility of using reinforcement learning algorithms based on stochastic policies will be explored. Finally, we plan to apply the proposed online modeling method to a real industrial system, for instance a rotary hearth furnace system.

Author Contributions: Conceptualization, X.Z. and R.Y.; Data curation, G.Z., L.W. and B.W.; Formal analysis, R.Y.; Funding acquisition, X.Z.; Investigation, G.Z., L.W. and B.W.; Methodology, X.Z. and R.Y.; Software, R.Y.; Supervision, B.W.; Validation, R.Y.; Visualization, G.Z. and L.W.; Writing—original draft, R.Y.; Writing—review & editing, X.Z. and R.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Science Foundation of China under Grant 62073210 and 62173219.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author. Dataset available on request from the authors. The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: Author Runbin Yan was employed by the company North Automatic Control Technology Institute, R.P. China. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Nomenclature

Symbol	Describtion	Symbol	Describtion
S	The states set.	A	The actions set.
$T(s,a,s')$	The probability of transitioning to state s' from s when taking action a.	γ	The discount factor.
s_t	The state at time step t.	a_t	The action at time step t.
r_t	Immediate reward at time step t.	$R(s,a)$	The reward function.
$\mu_\theta(s)$	The policy function with the parameter θ.	$Q^\mu(s,a)$	The action value function under the policy μ.
$J(\mu_\theta)$	The performance objective under the policy μ.	ρ^μ	The discounted state distribution.
$y(\bar{Z},t)$	The output of the DPS.	$u(t)$	The input variables of the DPS.
$\bar{O}_{sl}$	Denotes a 3D fuzzy set.	U_g^{hl}	Denotes a conventional fuzzy set.
$\bar{R}_l^*$	The fuzzy rules.	a_j	The weight of the jth sensor position.
c_{ij}^l	Centroid of the Gaussian 3D fuzzy set $\bar{O}_{sl}$.	σ_{ij}^l	Spread of the Gaussian 3D fuzzy set $\bar{O}_{sl}$.
d_{ik}^l	Centroid of the conventional Gaussian U_g^{hl}.	δ_{ik}^l	Spread of the conventional Gaussian U_g^{hl}.
a_j	Represents the weight of the *jth* sensor position.	$R(\mu)$	The average reward under the policy μ.
G_t	The long time return.	G_t	The long time return.
$v(s)$	The state-value function.	$Q(s,a)$	The action value function.
$\bar{R}_t$	The estimate of the average reward at time step t.	$s \sim \mu$	The distribution of state under the policy μ.
θ_k^μ	The parameters of the policy μ.	θ_k^Q	The parameters of the action value function.

References

1. von Flotow, A.H.; Schaefer, B.E. Wave absorbing. controllers for a flexible beam. *J. Guid. Control Dyn.* **1986**, *9*, 673–680. [CrossRef]
2. Christofides, P.D.; Chow, J. Nonlinear and robust control of PDE systems: Methods and applications to transport-reaction processes. *Appl. Mech. Rev.* **2002**, *55*, B29–B30. [CrossRef]
3. Chen, Y.; Zhu, Y.; Wang, Z.; Li, Y.; Wang, L.; Ding, L.; Gao, X.; Ma, Y.; Guo, Y. Application studies of activated carbon derived from rice husks produced by chemical-thermal process—A review. *Adv. Colloid Interface Sci.* **2011**, *163*, 39–52. [CrossRef] [PubMed]
4. Abonyi, J.; Babuska, R.; Szeifert, F. Fuzzy modeling with multivariate membership functions: Gray-box identification and control design. *IEEE Trans. Syst. Man Cybern. Part B* **2001**, *31*, 755–767. [CrossRef]
5. Xu, Y.Y.; Xu, K. Hammerstein model for distributed parameter system of micro-cantilever in atomic-force microscope. *Kongzhi Lilun Yu Yingyong/Control Theory Appl.* **2015**, *32*, 304–311. [CrossRef]
6. Shao-yuan, L. Time–space ARX modeling and predictive control for distributed parameter system. *Control Theory Appl.* **2011**, *28*, 1711–1716.
7. Wang, M.L.; Li, N.; Li, S.Y. Model-based predictive control for spatially-distributed systems using dimensional reduction models. *Int. J. Autom. Comput.* **2011**, *8*, 1–7. [CrossRef]
8. Varshney, A.; Pitchaiah, S.; Armaou, A. Feedback Control of Dissipative PDE Systems Using Adaptive Model Reduction. *AIChE J.* **2009**, *55*, 906–918. [CrossRef]
9. Zheng, D.; Hoo, K.A. Low-order model identification for implementable control solutions of distributed parameter systems. *Comput. Chem. Eng.* **2002**, *26*, 1049–1076. [CrossRef]
10. Bellamine, F.H.; Elkamel, A. Numerical characterization of distributed dynamic systems using tools of intelligent computing and generalized dimensional analysis. *Appl. Math. Comput.* **2006**, *182*, 1021–1039.
11. Park, H.M.; Cho, D.H. The use of the Karhunen-Loève decomposition for the modeling of distributed parameter systems. *Chem. Eng. Sci.* **1996**, *51*, 81–98. [CrossRef]
12. Deng, H.; Li, H.X.; Chen, G. Spectral-approximation-based intelligent modeling for distributed thermal processes. *IEEE Trans. Control Syst. Technol.* **2005**, *13*, 686–700. [CrossRef]
13. Erickson, M.; Smith, R.; Laub, A. Finite-dimensional approximation and error bounds for spectral systems with partially known eigenstructure. *IEEE Trans. Autom. Control* **1994**, *39*, 1904–1909. [CrossRef]
14. Jiang, Y.; Yin, S.; Kaynak, O. Data-Driven Monitoring and Safety Control of Industrial Cyber-Physical Systems: Basics and Beyond. *IEEE Access* **2018**, *6*, 47374–47384. [CrossRef]
15. A, M.W.; B, C.Q.; A, H.Y.; A, H.S. Hybrid neural network predictor for distributed parameter system based on nonlinear dimension reduction. *Neurocomputing* **2016**, *171*, 1591–1597.
16. Zhang, R.; Tao, J.; Lu, R.; Jin, Q. Decoupled ARX and RBF Neural Network Modeling Using PCA and GA Optimization for Nonlinear Distributed Parameter Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *29*, 457–469. [CrossRef]
17. Wang, Y.; Li, H.X.; Yang, H. Adaptive spatial-model-based predictive control for complex distributed parameter systems. *Adv. Eng. Inform.* **2024**, *59*, 102331. [CrossRef]
18. Chen, L.; Shen, W.; Zhou, Y.; Mou, X.; Lei, L. Learning-based sparse spatiotemporal modeling for distributed thermal processes of Lithium-ion batteries. *J. Energy Storage* **2023**, *69*, 107834. [CrossRef]
19. Fan, Y.; Xu, K.; Wu, H.; Zheng, Y.; Tao, B. Spatiotemporal Modeling for Nonlinear Distributed Thermal Processes Based on KL Decomposition, MLP and LSTM Network. *IEEE Access* **2020**, *8*, 25111–25121. [CrossRef]
20. Aggelogiannaki, E.; Sarimveis, H. Nonlinear model predictive control for distributed parameter systems using data driven artificial neural network models. *Comput. Chem. Eng.* **2008**, *32*, 1225–1237. [CrossRef]
21. Li, H.X.; Zhang, X.X.; Li, S.Y. A Three-Dimensional Fuzzy Control Methodology for a Class of Distributed Parameter Systems. *IEEE Trans. Fuzzy Syst.* **2007**, *15*, 470–481. [CrossRef]
22. Zhang, X.X.; Jiang, Y.; Li, H.X. 3-d fuzzy logic controller for spatially distributed dynamic systems: A tutorial. In Proceedings of the 2009 IEEE International Conference on Fuzzy Systems, Jeju Island, Republic Korea, 20–24 August 2009; pp. 854–859. [CrossRef]
23. Zhang, X.X.; Zhao, L.R.; Li, H.X.; Ma, S.W. A Novel Three-Dimensional Fuzzy Modeling Method for Nonlinear Distributed Parameter Systems. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 489–501. [CrossRef]
24. Zhang, X.X.; Fu, Z.Q.; Li, S.Y.; Zou, T.; Wang, B. A time/space separation based 3D fuzzy modeling approach for nonlinear spatially distributed systems. *Int. J. Autom. Comput.* **2018**, *15*, 52–65. [CrossRef]
25. Zhang, X.; Yuan, H.Y.; Li, H.X.; wei Ma, S. A spatial multivariable SVR method for spatiotemporal fuzzy modeling with applications to rapid thermal processing. *Eur. J. Control* **2020**, *54*, 119–128. [CrossRef]
26. Kadlec, P.; Grbić, R.; Gabrys, B. Review of adaptation mechanisms for data-driven soft sensors. *Comput. Chem. Eng.* **2011**, *35*, 1–24. [CrossRef]
27. Wang, Z.; Li, H.X. Incremental Spatiotemporal Learning for Online Modeling of Distributed Parameter Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2612–2622. [CrossRef]
28. Lu, X.; Yin, F.; Huang, M. Online Spatiotemporal Least-Squares Support Vector Machine Modeling Approach for Time-Varying Distributed Parameter Processes. *Ind. Eng. Chem. Res.* **2017**, *56*, 7314–7321. [CrossRef]
29. Zhan Chen, P.; Pei, J.; Lu, W.C.; Li, M. A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance. *Neurocomputing* **2022**, *497*, 64–75. [CrossRef]

30. Gupta, S.; Sangeeta, R.; Mishra, R.S.; Singal, G.; Badal, T.; Garg, D. Corridor segmentation for automatic robot navigation in indoor environment using edge devices. *Comput. Netw.* **2020**, *178*, 107374. [CrossRef]
31. Chen, Y.; Pan, Y.; Lu, Q. Reinforcement learning-based adaptive predefined-time optimal tracking control for strict-feedback nonlinear systems. *Int. J. Adapt. Control Signal Process.* **2023**, *38*, 492–512. [CrossRef]
32. Duan, X. Abnormal Behavior Recognition for Human Motion Based on Improved Deep Reinforcement Learning. *Int. J. Image Graph.* **2023**, *24*, 2550029:1–2550029:16. [CrossRef]
33. Cheng, B.; Wang, L.; Tan, Q.; Zhou, M. A deep reinforcement learning hyper-heuristic to solve order batching problem with mobile robots. *Appl. Intell.* **2024**, 1–23. [CrossRef]
34. Wang, X.; Zhong, P.; Liu, M.; Zhang, C.; Yang, S. A novel method-based reinforcement learning with deep temporal difference network for flexible double shop scheduling problem. *Sci. Rep.* **2024**, *14*, 9047. [CrossRef] [PubMed]
35. Peng, L.; Yuan, Z.; Dai, G.; Wang, M.; Tang, Z. Reinforcement learning-based hybrid differential evolution for global optimization of interplanetary trajectory design. *Swarm Evol. Comput.* **2023**, *81*, 101351. [CrossRef]
36. Stavrev, S.; Ginchev, D. Reinforcement Learning Techniques in Optimizing Energy Systems. *Electronics* **2024**, *13*, 1459. [CrossRef]
37. Wang, X.; Wang, S.; Liang, X.; Zhao, D.; Huang, J.; Xu, X.; Dai, B.; Miao, Q. Deep reinforcement learning: A survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *35*, 5064–5078. [CrossRef]
38. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.A.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
39. Berenji, H.R. Fuzzy reinforcement Learning and dynamic programming. In *Proceedings of the Fuzzy Logic in Artificial Intelligence*; Ralescu, A.L., Ed.; Springer: Berlin, Heidelberg, 1994; pp. 1–9.
40. Kofinas, P.; Dounis, A.I. Online Tuning of a PID Controller with a Fuzzy Reinforcement Learning MAS for Flow Rate Control of a Desalination Unit. *Electronics* **2019**, *8*, 231. [CrossRef]
41. Wang, X.; Ma, Z.; Mao, L.; Sun, K.; Huang, X.; Fan, C.; Li, J. Accelerating Fuzzy Actor–Critic Learning via Suboptimal Knowledge for a Multi-Agent Tracking Problem. *Electronics* **2023**, *12*, 1852. [CrossRef]
42. Bi, Y.; Ding, Q.; Du, Y.; Liu, D.; Ren, S. Intelligent Traffic Control Decision-Making Based on Type-2 Fuzzy and Reinforcement Learning. *Electronics* **2024**, *13*, 3894. [CrossRef]
43. Er, M.J.; Deng, C. Online tuning of fuzzy inference systems using dynamic fuzzy Q-learning. *IEEE Trans. Syst. Man Cybern. Part B* **2004**, *34*, 1478–1489. [CrossRef] [PubMed]
44. Wang, Z.; Li, H.X.; Chen, C. Reinforcement Learning-Based Optimal Sensor Placement for Spatiotemporal Modeling. *IEEE Trans. Cybern.* **2020**, *50*, 2861–2871. [CrossRef] [PubMed]
45. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. In Proceedings of the International Conference on Machine Learning, Beijing, China, 22–24 June 2014; Pmlr: Beijing, China, 2014; pp. 387–395.
46. Zhang, X.X. *A Three-Domain Fuzzy Controller with Spatial Information Fusion*; Publishing House of Electronics Industry: Beijing China, 2017.
47. Sutton, R.S.; Barto, A.G. Reinforcement Learning: An Introduction. *IEEE Trans. Neural Netw.* **1998**, *9*, 1054. [CrossRef]
48. Qi, C.; Li, H.X.; Zhang, X.; Zhao, X.; Li, S.; Gao, F. Time/Space-Separation-Based SVM Modeling for Nonlinear Distributed Parameter Processes. *Ind. Eng. Chem. Res.* **2011**, *50*, 332–341. [CrossRef]