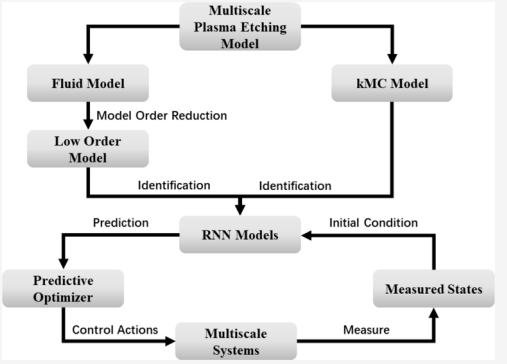
I&EC research
Industrial & Engineering Chemistry Research
pubs.acs.org/IECR
Article

Recurrent Neural-Network-Based Model Predictive Control of a Plasma Etch Process

Tianqi Xiao, Zhe Wu, Panagiotis D. Christofides, Antonios Armaou, and Dong Ni*

Cite This: *Ind. Eng. Chem. Res.* 2022, 61, 638−652
Read Online
ACCESS | Metrics & More | Article Recommendations

ABSTRACT: In this article, we propose the development of a recurrent neural network (RNN)-based model predictive controller (MPC) for a plasma etch process on a three-dimensional substrate using inductive coupled plasma (ICP) analysis. Specifically, the plasma etch process is simulated by a multiscale model: (1) A macroscopic fluid model is applied to simulate the gas flows and chemical reactions of plasma. (2) A kinetic Monte Carlo (kMC) model is applied to simulate the etching process on the substrate. Subsequently, proper orthogonal decomposition (POD) is used to derive the empirical eigenfunctions of the plasma model. Then the empirical eigenfunctions are utilized as basis functions within a Galerkin's model reduction framework to compute a low-order system capturing dominant dynamics of the plasma model. Additionally, RNN is introduced to approximate dynamics of both the reduced-order plasma system and the microscopic etch process. The training data for the RNN models are generated from discrete sampling of open-loop simulations. A probability distribution function is also involved to present the stochastic characteristic of the kMC model. The trained RNN models are then implemented as the prediction model in the development of MPC to achieve desired control objectives. Closed-loop simulation results are presented to compare the performance of the model predictive controller and a proportional-integral (PI) controller, which show that the proposed MPC framework is effective and exhibits better performance than does a PI controller.



1. INTRODUCTION

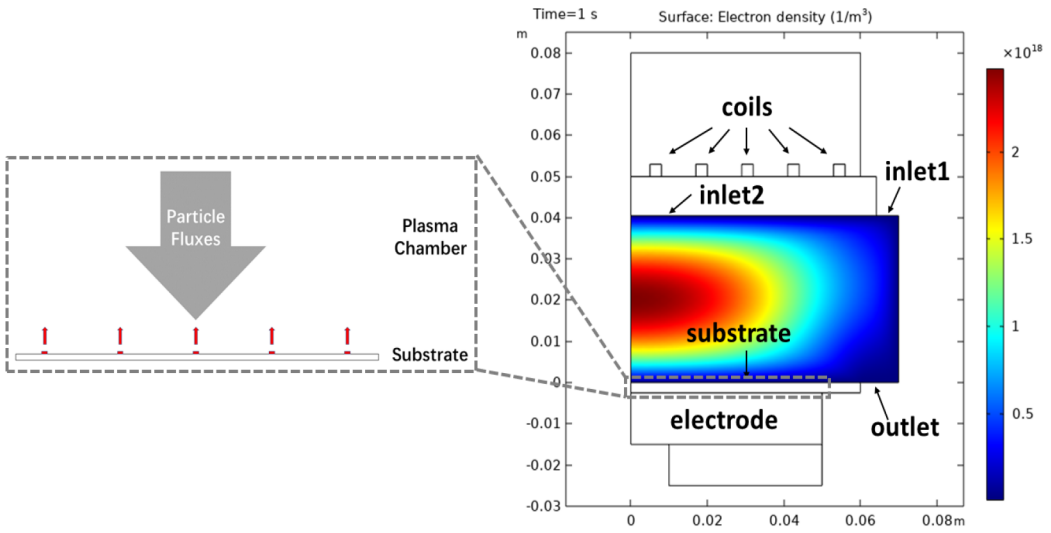
Plasma etching has been applied in integrated circuits (IC) processing since the 1960s and has became one of the core techniques due to the continuous decreases in fabricating scale.[1,2] Simulations of plasma etching are effective ways to maintain and optimize the etching process techniques.[3,4] Among all the simulation models for plasma, fluid models are commonly used because they are computationally efficient and flexible in coupling the electromagnetic fields. Additionally, the complex transport phenomena and reactions of the etching process are simulated by some quite precise approaches, like the level set method and the kinetic Monte Carlo method (kMC). kMC has the most potential to realize a high-resolution simulation of plasma etch processes while maintaining a relatively high computational efficiency.[5−7] An appropriate modeling methodology is established by kMC transforming every physical phenomena into stochastically selected events. Through simulations or experiments, the probability table of the simulated process can be attained, which is the key step for kMC.

However, the natural disparity of scales between the macroscopic plasma chamber and the microscopic etching process is a big challenge. The large temporal/spatial differences of scales prevent us to simulate the plasma behavior and the etching process concurrently. Continuous models used to simulate plasma are insufficient at microscopic scale. Meanwhile, high-resolution approaches applied to simulating etching process, like kMC, are impossible to apply on macroscopic domains considering the computational efficiency. Multiscale models are the solution to this problem and such models were developed in the crystallization process,[8] pulping,[9] and catalytic reactors.[10] Microscopic models and macroscopic models are computed concurrently in these works through coupling bridges to exchange data. Nevertheless, three-dimensional (3D) multiscale models for plasma etch processes were rarely developed in previous works. In an earlier work by our group, we have presented a 3D multiscale model for the silicon etch process using Cl_2/Ar inductive coupled plasma (ICP) analysis.[11]

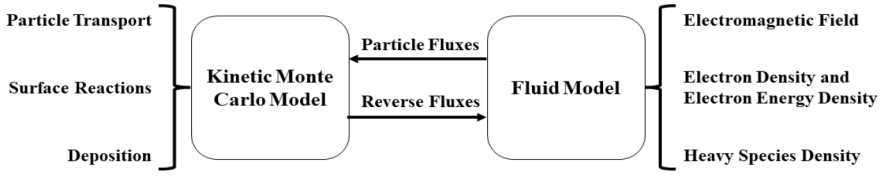
Plasma models are complex in that they are derived from conservation laws and are described by highly dissipative (typically parabolic) partial differential equation (PDE)

Received: October 26, 2021
Revised: December 10, 2021
Accepted: December 14, 2021
Published: December 28, 2021

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652



(a) The ICP equipment



(b) The Multiscale Model

Figure 1. (a) the half cross section of the ICP equipment and (b) the multiscale model.

systems. Traditional methods to solve PDEs involve temporal and spatial discretization by using a finite-difference or finite-element method,[12,13] the main drawback of which is that they require significant computational resources. A promising approach to solve this problem for dissipative PDE systems is model order reduction, in which the dominant behavior can be captured by a finite number of modes.[14,15] If global basis functions of the system are available, then the dissipative PDE can be discretized using weighted residual methods such as Galerkin projection, and then a reduced-order system of finite dimensional ordinary differential equations (ODEs) can be derived.[16] An appropriate approach to obtain the global basis functions is that of applying statistical techniques to compute empirical basis functions (EBFs) based on dominant dynamics of systems.[17,18] The empirical functions are computed by applying proper orthogonal decomposition (POD) to process the data.[19] Subsequently, the empirical eigenfunctions can be used as basis functions within a Galerkin's method and approximate an inertial manifold model reduction framework to derive low-order ODE systems that accurately describe the dominant dynamics of the PDE system.[20,21]

Machine-learning and deep-learning methods have grown rapidly in last 20 years owing to the development of advanced machine-learning algorithms, innovative neural network structures, powerful computers and open-source software libraries. Among all of these machine-learning methods, a recurrent neural network (RNN) is an outstanding method to be applied in approximating nonlinear dynamical systems. RNN was first proposed in 1980s and was utilized in pattern recognition.[22] Later, more complex networks like long short-term memory (LSTM) and gated recurrent unit (GRU) were invented to overcome the gradient vanishing problem of RNN.[23,24] What distinguishes RNN from the commonly known feed-forward neural networks is the existence of feedback cycles in the connections topology. These recurrent cycles make it possible for RNN to capture the dynamic behaviors of nonlinear systems.[25] Results in the literature showed that RNN can approximate any dynamic system to an arbitrary accuracy.[26] Several works on using RNN to approximate dynamic systems have been already published in both discrete-time systems[27] or continuous systems.[28]

As a powerful and advanced control methodology, model predictive control (MPC) has been applied in real-time operation of various industrial processes in order to optimize process performance.[29,30] While most studies on MPC pay attention to lumped-parameter process models, the application of MPC on distributed parameter systems is a large area that need to be further developed. Exceptions include refs 31 and 32. In the first work,[31] a finite-difference method was utilized to generate a high-order ODE system as a prediction model in a MPC framework. In the second work,[32] a low-order ODE model was developed as a prediction model by using Galerkin's method in an economic MPC framework. Designs of MPC for multiscale systems were proposed in previous works,[33,34] which rely on the following: (1) POD was employed to construct reduced order models of the continuum model. (2) The microscopic model was approximated by a discrete system using in situ adaptive tabulations (ISAT), in which the identified state-space of the coarse microscopic state can be tabulated.[35,36] A prediction model is one of the key elements of MPC and can be derived by first-principle models or data-driven models. In practical implementations, first-principles

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652

models are always difficult to attain, especially for systems with complex and nonlinear behaviors. Data-driven modeling is an appropriate method to address this issue and has been historically researched.[37,38] In these works, data-driven methods like subspace methods were utilized to identify input−output systems. Machine-learning models such as RNN were applied as prediction models to solve model predictive control problems for nonlinear systems in several works.[39−42] It should be noted that the computational complexity of RNN would increase if it is utilized to approximate high-order complex systems, which would lead to significant difficulties for the practical use of RNN in MPC. The combination of a low-order solution and RNN can be a reasonable method for this problem.

Motivated by the above considerations, we focus on the design of MPC for a plasma etch process on a 3D substrate using inductive coupled plasma (ICP) analysis. Specifically, the plasma etch process is simulated by a multiscale model: (1) A macroscopic fluid model is applied to simulate the gas flows and chemical reactions of plasma and is constructed in COMSOL Multiphysics (for convenience, called COMSOL from now on). (2) kMC is applied to simulate the etch process on the substrate. Subsequently, POD is used to derive empirical eigenfunctions of the plasma model. Then the empirical eigenfunctions are utilized as basis functions within a Galerkin's model reduction framework to compute a low-order system capturing the dominant dynamics of the plasma model. Additionally, RNN is implemented to approximate the dynamics of both the reduced-order plasma system and the microscopic etch process. Training data for RNN models are generated from discrete sampling of open-loop simulations. A probability distribution function is also involved in order to present the stochastic nature of the kMC model. The trained RNN models are then implemented as prediction model in the development of MPC to achieve desired control objectives. Closed-loop simulation results are presented to compare the performance of the model predictive controller and a proportional-integral (PI) controller, which show that the effectiveness of the proposed MPC framework and improved control performance compared to PI controller.

The article is organized as following: In section 2, the multiscale system and methodologies of our framework is presented. In section 3, the macroscopic model and the model reduction procedure are given. The training process of the RNN model for macroscopic model is also described in section 3. Subsequently, in section 4, we briefly show the microscopic kMC model. The training process of the RNN model for kMC model and the development of the probability distribution function are also presented in section 4. The designs of the PI and model predictive controllers are presented in section 5. Finally, the simulation results are discussed in section 6, and a conclusion is presented in section 7.

2. PRELIMINARIES

2.1. Multiscale Model. In Figure 1, we present the schematic diagram of the ICP reactor considered in this work. The reactive gases are pumped into the chamber through two inlets: the edge inlet (inlet1) and the center inlet (inlet2). Top coils generate the plasma in the plasma chamber. The bottom electrode accelerates the ions in the plasma chamber, which will impact the substrate placed on top of the electrode. Complex reactions occur on the substrate and derive the microscopic etching process.

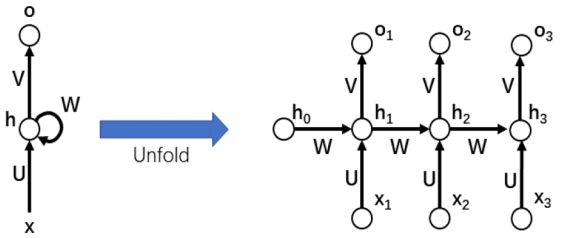


Figure 2. Recurrent neural network and its unfolded structure.

The disparity of scales of simulation models and a flow diagram of the multiscale model are shown in Figure 1. In order to capture both macroscopic plasma behaviors and microscopic etching behaviors, a multiscale model consisting of two simulation model components is developed: The continuous fluid model consisting of three modules (electromagnetic field, electron and electron energy density, and heavy species density) is established in COMSOL; the kMC model simulating three kinetic behaviors of the etching surface (particle transport, surface reaction, and deposition) is completed through C language. In addition, the macroscopic model and the microscopic model were coupled by applying a spatial-temporal discrete method. Fluxes data is exchanged between the two models in discrete time step t_s. Furthermore, the whole substrate is divided into five parts. The etching process of each part is represented by one microscopic model.

2.2. Proper Orthogonal Decomposition. We use solution data of the fluid model to construct global basis functions by utilizing POD. POD is applied to an ensemble of process solution data to derive a small set of empirical eigenfunctions which describe dominant spatial patterns of the nonlinear PDE system. In this work, the ensemble of solutions is constructed based on solutions of the fluid model for different profiles of plasma parameters and different initial conditions.[14,15] Application of POD to this ensemble of data provides an orthogonal set of basis functions (known as empirical eigenfunctions) for the representation of the ensemble, as well as a measure of the relative contribution of each basis function to the total energy (mean square fluctuation) of the ensemble. A truncated series representation of the ensemble data in terms of the dominant basis functions has a smaller mean square error than a representation by any other basis of the same dimension.[43] These computed eigenfunctions will then be used as basis functions within the Galerkin's model reduction framework. Specifically, we first generate a large ensemble set of the solutions of our PDE system. In particular, we can only consider five PDEs, which will be illustrated in detail in the following sections. We assume one set of these solutions as $\overline{v}_k$, which consists of K sampled states $\overline{v}_k(z)$ (typically called snapshots). Then we define the ensemble average as

$$\langle \overline{v}_k \rangle = \frac{1}{K} \sum_{n=1}^{K} \overline{v}_k(z) \tag{1}$$

Furthermore, the ensemble average of snapshots $\langle \overline{v}_k \rangle$ is subtracted out from the snapshots.

$$v_k = \overline{v}_k - \langle \overline{v}_k \rangle \tag{2}$$

Subsequently, we define the covariance matrix B as follows:

$$B^{\kappa k} = \frac{1}{K} \int_{\Omega} v_\kappa(z) v_k(z)\, dz \tag{3}$$

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652

where Ω denotes the domain of definition of the process, and the eigenvalues can be obtained by following equation:

$$Bc = \lambda c \tag{4}$$

The solution of the above eigenvalue problem yields the eigenvectors $c = [c_1, ..., c_K]^T$ which can be used in following equations to construct the eigenfunctions:

$$\phi(z) = \sum_k c_k \nu_k(z) \tag{5}$$

2.3. Galerkin's Method. In this section, we use Galerkin's method to derive low-order systems of nonlinear ordinary differential equations, which represent the dominant dynamics of the nonlinear PDE system. To simplify the presentation, we first present a general form of PDEs. Subsequently, we assume that an orthogonal and complete set of analytical eigenfunctions is available, which can span the entire domain of the nonlinear process. Specifically in our work, the basis functions $\phi_k(z)$ are the set of empirical eigenfunctions computed through POD.

First we present the general form of PDEs:

$$\frac{\partial x}{\partial t} = A\frac{\partial x}{\partial z} + B\frac{\partial^2 x}{\partial z^2} + Wu(t) + f(x(z, t)) \tag{6}$$

subject to the mixed-type boundary conditions:

$$q\left(x, \frac{\mathrm{d}x}{\mathrm{d}\eta}, \frac{\mathrm{d}^2x}{\mathrm{d}\eta^2}\right) = 0, \quad \text{on } \Gamma \tag{7}$$

and the initial conditions:

$$x(z, 0) = x_0(z) \tag{8}$$

where $x(z, t) = [x_1(z, t), ..., x_n(z, t)]^T \in \Omega \subset \mathbb{R}^n$ denotes the vector of state variables, $t \in [0, \infty)$ is the time, $z = [z_1, z_2] \in \mathbb{R}^2$ is the vector of spatial coordinates, $f(x(z, t))$ denotes a nonlinear vector function, $u \in \mathbb{R}^{n_u}$ denotes the manipulated input vector, A, B, and W are matrices and vectors of appropriate dimensions, $q\left(x, \frac{\mathrm{d}x}{\mathrm{d}\eta}, \frac{\mathrm{d}^2x}{\mathrm{d}\eta^2}\right)$ is a nonlinear vector function which is assumed to be sufficiently smooth, Γ is the boundary of the domain Ω, $\frac{\mathrm{d}x}{\mathrm{d}\eta}|_\Gamma$ denotes the derivative in the direction perpendicular to the boundary, and $x_0(z)$ is a smooth vector function of z.

Then, we formulate the parabolic PDE system of eq 6 in the Hilbert space $\mathcal{H}(\Omega, \mathbb{R}^n)$, where $\mathcal{H}$ is the space of n-dimensional vector functions defined on Ω that satisfy the boundary condition eq 7, with inner product and norm:

$$(\omega_1, \omega_2) = \int_\Omega (\omega_1(z), \omega_2(z))_{\mathbb{R}^n} \mathrm{d}z$$
$$\|\omega_1\|_2 = (\omega_1, \omega_1)^{1/2} \tag{9}$$

where ω_1 and ω_2 are two elements of $\mathcal{H}(\Omega, \mathbb{R}^n)$ and the notation $(\cdot, \cdot)_{\mathbb{R}^n}$ denotes the standard inner product in $\mathbb{R}^n$. Subsequently, we define the state function $\bar{x}(t) = x(z, t)$, $t > 0$, $z \in \Omega$, and the operator $\mathcal{A}$ in $\mathcal{H}(\Omega, \mathbb{R}^n)$ as

$$\mathcal{A}\bar{x} = A\frac{\mathrm{d}\bar{x}}{\mathrm{d}z} + B\frac{\mathrm{d}^2\bar{x}}{\mathrm{d}z^2}, \quad z \in \Omega \tag{10}$$

and the input operator as well as the nonlinear function operator as $\mathcal{B}\,\bar{x}(t) = Wu(t)$, $\mathcal{F}(\bar{x}(t))$. The system of eq 6 can be written in an n-dimensional semilinear form:

$$\dot{\bar{x}}(t) = \mathcal{A}\,\bar{x}(t) + \mathcal{B}\,u(t) + \mathcal{F}(\bar{x}(t)), \quad \bar{x}(0) = \bar{x}_0 \tag{11}$$

where $\bar{x}_0 = x_0(z)$. Then, we let $[\lambda_1, ..., \lambda_N]$ be the eigenvalues of $\mathcal{A}$ and $[\phi_1, ..., \phi_N]$ be the corresponding eigenvectors, and we let $\sigma(\mathcal{A}) = \{\lambda_1, ..., \lambda_N\}$ be the eigenspectrum of $\mathcal{A}$. On the basis of the assumption that most diffusion−reaction processes have a large separation between slow and fast modes, $\sigma(\mathcal{A})$ can be divided as $\sigma(\mathcal{A}) = \sigma_s(\mathcal{A}) \oplus \sigma_f(\mathcal{A})$, where $\sigma_s(\mathcal{A})$ consists of the first m slow eigenvalues and $\sigma_f(\mathcal{A})$ consists of the remaining fast eigenvalues. Then, we define the $\mathcal{H}_s$ and $\mathcal{H}_f$ modal subspaces of $\mathcal{A}$ as $\mathcal{H}_s = \mathrm{span}\{\phi_1, ..., \phi_m\}$ and $\mathcal{H}_f = \mathrm{span}\{\phi_{m+1}, ..., \phi_N\}$. We define $\epsilon = |Re\lambda_1|/|Re\lambda_{m+1}|$. Let P_s and P_f denote the orthogonal projection operators that project the state $\bar{x}$ onto the subspaces $\mathcal{H}_s$ and $\mathcal{H}_f$ of $\mathcal{A}$ as $\bar{x} = \bar{x}_s \oplus \bar{x}_f = P_s\bar{x} \oplus P_f\bar{x}$. Utilizing the decomposition of $\bar{x}$ in eq 11, the following equations are given:

$$\begin{aligned}
\dot{\bar{x}}_s(t) &= \mathcal{A}_s\,\bar{x}_s(t) + \mathcal{F}_s(\bar{x}_s(t), \bar{x}_f(t)) + \mathcal{B}_s\,u(t),\\
\bar{x}_s(0) &= P_s\bar{x}(0) = P_s\bar{x}_0\\
\dot{\bar{x}}_f(t) &= \mathcal{A}_f\,\bar{x}_f(t) + \mathcal{F}_f(\bar{x}_s(t), \bar{x}_f(t)) + \mathcal{B}_f\,u(t),\\
\bar{x}_f(0) &= P_f\bar{x}(0) = P_f\bar{x}_0
\end{aligned} \tag{12}$$

where $\mathcal{A}_s = P_s\mathcal{A}$, $\mathcal{B}_s = P_s\mathcal{B}$, $\mathcal{F}_s = P_s\mathcal{F}$, $\mathcal{A}_f = P_f\mathcal{A}$, $\mathcal{B}_f = P_f\mathcal{B}$, and $\mathcal{F}_f = P_f\mathcal{F}$. It should be noted that $\mathcal{A}_s$ is a diagonal matrix and that the operator $\mathcal{A}_f$ is an unbounded exponentially stable differential operator. To capture the dominant dynamics of the PDE system, only the slow modes of resulting ODE system is considered:

$$\begin{aligned}
\dot{\bar{x}}_s(t) &= \mathcal{A}_s\,\bar{x}_s(t) + \mathcal{F}_s(\bar{x}_s(t), \bar{x}_f(t)) + \mathcal{B}_s\,u(t),\\
\bar{x}_s(0) &= P_s\bar{x}_0
\end{aligned} \tag{13}$$

Following the approximation procedure in previous work,[16] an $O(\epsilon)$ approximation can be presented:

$$\dot{\bar{x}}_s(t) = \mathcal{A}_s\,\bar{x}_s(t) + \mathcal{F}_s(\bar{x}_s(t), 0) + \mathcal{B}_s\,u(t), \quad \bar{x}_s(0) = P_s\bar{x}_0 \tag{14}$$

2.4. RNN Model. RNN is an outstanding approach to approximate dynamic systems for that it has the memory of past states. The unfolded structure of RNN is shown in Figure 2. Specifically, **U**, **V**, and **W** represent the RNN parameters, **x** represents the model inputs, **o** represents the model outputs, and **h** represents the hidden states. The states derived from past inputs are imported into next RNN cell, which shows dynamic behaviors. With recent advancements of GPU technology and open-source software libraries, RNN can be easily developed and efficiently trained in an open-source toolkit, e.g., Pytorch or Tensorflow, which makes it proper for identifying dynamic models with high accuracy.

3. MACROSCOPIC MODEL

3.1. Fluid Model. There are three main modules in computing the macroscopic fluid model: the Maxwell equations, the drift−diffusion equations, and the heavy species transportation equations (simplified Stefan−Maxswell equations). The electromagnetic field is simulated through Maxwell equations:

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652

$$\nabla \times \mathbf{H} = \mathbf{J}$$
$$\mathbf{B} = \nabla \times \mathbf{A}$$
$$\mathbf{E} = -j\omega\mathbf{A}$$
$$\mathbf{J} = \sigma\mathbf{E} + j\omega\mathbf{D} \quad (15)$$

Second, the electron density and the electron energy density are solved through drift−diffusion equations:[44]

$$\frac{\partial}{\partial t}(n_e) + \nabla\cdot\Gamma_e = R_e - (\mathbf{u}\cdot\nabla)n_e$$
$$\Gamma_e = -(\mu_e\cdot\mathbf{E})n_e - \nabla(\mathbf{D}_e n_e)$$
$$\frac{\partial}{\partial t}(n_\varepsilon) + \nabla\cdot\Gamma_\varepsilon + \mathbf{E}\cdot\Gamma_e =$$
$$S_{en} - (\mathbf{u}\cdot\nabla)n_\varepsilon + (Q + Q_{gen})/q$$
$$\Gamma_\varepsilon = -(\mu_\varepsilon\cdot\mathbf{E})n_\varepsilon - \nabla(\mathbf{D}_\varepsilon n_\varepsilon) \quad (16)$$

where n_e is the electron density, R_e is the electron rate expression, μ_e is the electron mobility, $\mathbf{E}$ is the electric field, $\mathbf{D}_e$ is the electron diffusivity, $\mathbf{u}$ is the neutral fluid velocity vector, n_ε is the electron energy density, S_{en} is the energy loss/gain due to inelastic collisions, Q is an external heat source, Q_{gen} is a generalized heat source, μ_ε is the electron energy mobility which is either a scalar or a tensor, and $\mathbf{D}_\varepsilon$ is the electron energy diffusivity.

Third, all the heavy species densities are computed through heavy species transportation equations:[45]

$$\rho\frac{\partial}{\partial t}(w_k) + \rho(\mathbf{u}\cdot\nabla)w_k = \nabla\cdot\mathbf{j}_k + R_k \quad (17)$$

where $\mathbf{j}_k$ is the diffusive flux vector, R_k is the rate expression for species k, $\mathbf{u}$ is the mass averaged fluid velocity vector, ρ is the density of the mixture, and w_k is the mass fraction of species k. It should be noted that the drift−diffusion equations and the heavy species transportation equations (14 equations in total) can be approximated by a reduced order system via POD and Galerkin's method.

Specifically, 12 species and their corresponding reactions are taken into account. Species are divided into three classes: the ground-state species (Cl, Cl_2, and Ar), the excitation species (Cl_2V, Cl_2(B3PI), Cl_2(C1PI), Cl_2(B3PI+C1PI), Ars), and ions (Cl^-, Cl^+, Cl_2^+, and Ar^+). Three types of reactions are taken into account: electron impact reactions, heavy species reactions and wall reactions. The electron impact reactions and heavy species reactions play determinant roles in the plasma chemistry, which are considered in terms R_e, S_{en}, and R_k during the computational process of the fluid model. The wall reactions are considered in boundary conditions during the computational process. All the reaction coefficients and reaction rates can be found in the literature.[46−49] Specific and detailed illustrations on the plasma chemistry can be found in our previous work.[11] The control variables of the plasma chamber are set as the power of the top coils (P_{rf}), the bottom electrode bias (V_B), Ar/Cl_2 ratio of the input gases at edge inlet (R_1), and Ar/Cl_2 ratio of the input gases at center inlet (R_2).

3.2. Low-Order Model. Of the 12 species of plasma model, only Cl, Cl_2, Cl^+, Cl_2^+, and Ar^+ are considered capable to participate in the reactions on the substrate. Thus, five low-order approximations that capture the dominant dynamics of these five species should be generated by utilizing POD and Galerkin's method. The states of the plasma species can be written as

$$x(z, t) = \sum_{i=1}^{m} c_i(t)\,\phi_i(z) \quad (18)$$

where $[c_1(t), ..., c_m(t)]^T$ are amplitudes of the first m eigenfunctions. The computational procedure can be presented as follows:

(1) Generate an ensemble data set of the five plasma species states from extensive open-loop simulations. To capture all the dynamics of the plasma states, open-loop simulations should be operated with combinations of different initial conditions and inputs (as mentioned in above section, the inputs would be R_1, R_2, P_{rf}, and V_B).

(2) Utilize POD on the ensemble data set to construct the empirical eigenfunctions. The normalized computed eigenvalues are also measures of the relative contribution of each basis function to the total energy, or equivalently, the percentage of time that the solution of the PDE system spends along the spatial structure of the empirical functions.

(3) Arrange these empirical eigenfunctions in a descending order of their eigenvalues and generate a set of the first m eigenfunctions to store large enough (>99.9%) energy of the whole system, which means the sum of the first m eigenvalues should be greater than 99.9% of the total sum of all eigenvalues identified.

(4) Decompose the plasma states into the form of $c(t)$ by utilizing Galerkin's method and the dominant empirical eigenfunctions for model reduction.

3.3. Development of RNN Models. In this section, RNN models are implemented to approximate the low-order system. Theoretically, RNN models are able to directly learn the dynamics of the high-order fluid model with enough solution data. However, in practical implementation, it is quite difficult to construct and train an RNN model that captures the whole states of the fluid model. Therefore, training the RNN model based on low-order solutions is a more feasible and simpler approach. Five corresponding RNN models are constructed for the five low-order approximations of the plasma states. The five RNN models are named as the RNN_L model, where L represents Cl, Cl_2, Cl^+, Cl_2^+, and Ar^+, respectively. Training data comes from the low-order solutions for the open-loop simulations of the plasma model (the data set $c(t)$). The sampling period of open-loop simulations is t_s = 0.2 s. Specifically, the inputs of RNN_L are the values of R_1, R_2, P_{rf}, and V_B at every time step t_k ($t_k = kt_s$, k = 1, 2, ...) . The initial conditions of RNN_L are the corresponding coefficients $[c_1(t_k), ..., c_m(t_k)]^T$ at time t_k. The outputs of RNN_L would be the corresponding coefficients $[c_1(t_{k+1}), ..., c_m(t_{k+1})]^T$ at time t_{k+1}. The inputs are selected randomly at the beginning of every time step within the ranges $0.05 \le R_1 \le 0.5$, $0.05 \le R_2 \le 0.5$, 800 W $\le P_{rf} \le$ 1400 W, and 50 V $\le V_B \le$ 250 V. The whole data set is finally obtained from an open-loop simulation which includes 24 000 time steps.

The RNN_L model is constructed using the RNN module of pytorch with python version 3.0, torch version 1.2.0, and cuda version 10.0. The GPU used for training is GTX 1080ti. Specifically, for each sub-RNN model of RNN_L, the number of neurons and layers as well as the learning rate are determined by the grid search method. Specifically, there will be one layer in the recurrent layer and two linear layers as well as one rectified linear unit (ReLU) layer in the output layer. The

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652

Adam optimization method is applied as the parameter optimization algorithm; the mean square error (MSE) is applied as the loss function. In order to avoid the overfitting problem, the training process was terminated if the loss of the validation data set falls below a desired threshold (set as 10^{-6}), which is chosen based on pre-experiments. Normalization method is applied to the input and output values:

$$\overline{x_j} = \frac{x_j - x_{\min,j}}{x_{\max,j} - x_{\min,j}} \tag{19}$$

where x_j is the jth variable of the data set and $x_{\max,j}$ and $x_{\min,j}$ are the maximum and minimum values of the jth variable, respectively.

4. MICROSCOPIC MODEL

Simulation of the etching process is completed by using kMC method. To reduce computing resource requirements, we further implement an RNN model (called RNN_k) and a probability distribution function to approximate the kMC model outputs. Specifically, we use RNN_k to approximate the expectation of the kMC model and use the probability distribution function to describe the stochastic nature of kMC. The development of the kMC method, the RNN model, and the probability distribution function applied in this work follows closely that in our previous works.[11,50] For completeness of the modeling process, we will present the main structure of these methods.

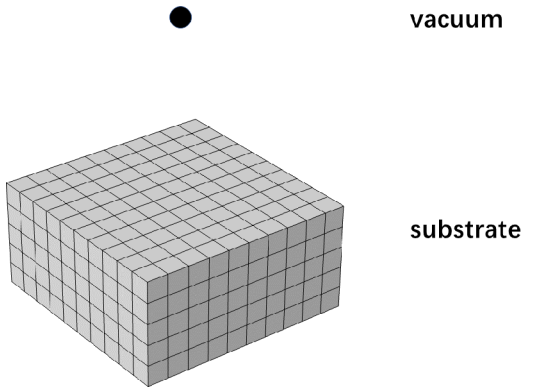


Figure 3. Virtual simulation box of the kMC model.

Table 1. All Surface Reactions and Depositions

chemical reaction of chlorine with Si	$Cl + Si_{(s)} \rightarrow SiCl_{(s)}$	0.99[49,55]
	$Cl + SiCl_{(s)} \rightarrow SiCl_{2(s)}$	0.20[49,55]
	$Cl + SiCl_{2(s)} \rightarrow SiCl_{3(s)}$	0.15[49,55]
	$Cl + SiCl_{3(s)} \rightarrow SiCl_{4(g)}$	0.0001[49,55]
	$Cl_2 + Si_{(s)} \rightarrow SiCl_{2(s)}$	0.05[49,55]
	$Cl_2 + SiCl_{(s)} \rightarrow SiCl_{2(s)} + Cl$	0.05[49,55]
	$Cl_2 + SiCl_{2(s)} \rightarrow SiCl_{3(s)} + Cl$	0.05[49,55]
	$Cl_2 + SiCl_{3(s)} \rightarrow SiCl_{4(g)} + Cl$	0.0001[49,55]
deposition of Si	$Si + Si_{(s)} \rightarrow Si_{(s)} + Si_{(s)}$	1.00[49]
	$Si + SiCl_{(s)} \rightarrow Si_{(s)} + SiCl_{(s)}$	1.00[49]
	$Si + SiCl_{2(s)} \rightarrow Si_{(s)} + SiCl_{2(s)}$	1.00[49]
	$Si + SiCl_{3(s)} \rightarrow Si_{(s)} + SiCl_{3(s)}$	1.00[49]
deposition of SiCl	$SiCl + Si_{(s)} \rightarrow SiCl_{(s)} + Si_{(s)}$	0.80[49]
	$SiCl + SiCl_{(s)} \rightarrow SiCl_{(s)} + SiCl_{(s)}$	0.50[49]
	$SiCl + SiCl_{2(s)} \rightarrow SiCl_{(s)} + SiCl_{2(s)}$	0.30[49]
	$SiCl + SiCl_{3(s)} \rightarrow SiCl_{(s)} + SiCl_{3(s)}$	0.10[49]
deposition of $SiCl_2$	$SiCl_2 + Si_{(s)} \rightarrow SiCl_{2(s)} + Si_{(s)}$	0.80[49]
	$SiCl_2 + SiCl_{(s)} \rightarrow SiCl_{2(s)} + SiCl_{(s)}$	0.50[49]
	$SiCl_2 + SiCl_{2(s)} \rightarrow SiCl_{2(s)} + SiCl_{2(s)}$	0.30[49]
	$SiCl_2 + SiCl_{3(s)} \rightarrow SiCl_{2(s)} + SiCl_{3(s)}$	0.10[49]

4.1. kMC Model. kMC uses stochastically selected events to represent all phenomena of the etching process, and realize a kinetic simulation. In Figure 3, the virtual simulation box of the kMC model is exhibited. The black sphere is the particle injected from the plasma region to the virtual box. A lattice that is composed of atomic cubic cells is implemented to simulate the substrate. One cell only has one atom in it. When particles bombard the lattice surface, the etching parameters are calculated based on the material type, injection angle, and local coverage type. The atomic kinetic simulation is realized by adding or removing atoms on the lattice. In this work, the substrate material is silicon. The resist material is set as incorruptible since the sputtering rate of etchable resist is relatively low. Specifically, the initial structure of the lattice is set as following: the lattice size is 100 × 100 × 100 monolayers^3 (ML^3); the resistive mask is on the substrate and its height is 50 ML; a 40 × 40 ML^2 surface is uncovered by the resistive mask, which is in the middle of the whole surface. The inputs to the kMC model are the particle fluxes $F_{i,k}$ (denotes the flux of ith species at jth location) and the electrode bias (V_B). The average etching depth (D) of all uncovered surface sites is defined as the etching depth of the

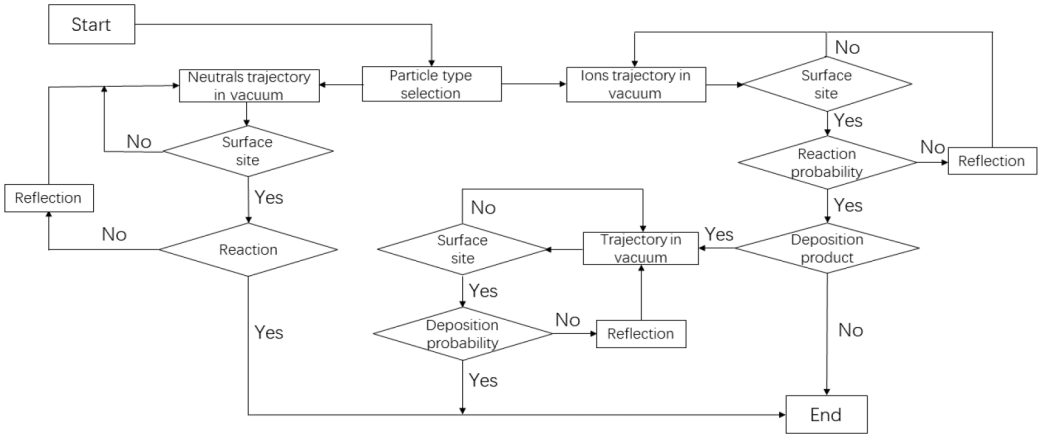


Figure 4. Numerical algorithm flowchart of the kMC model.

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652

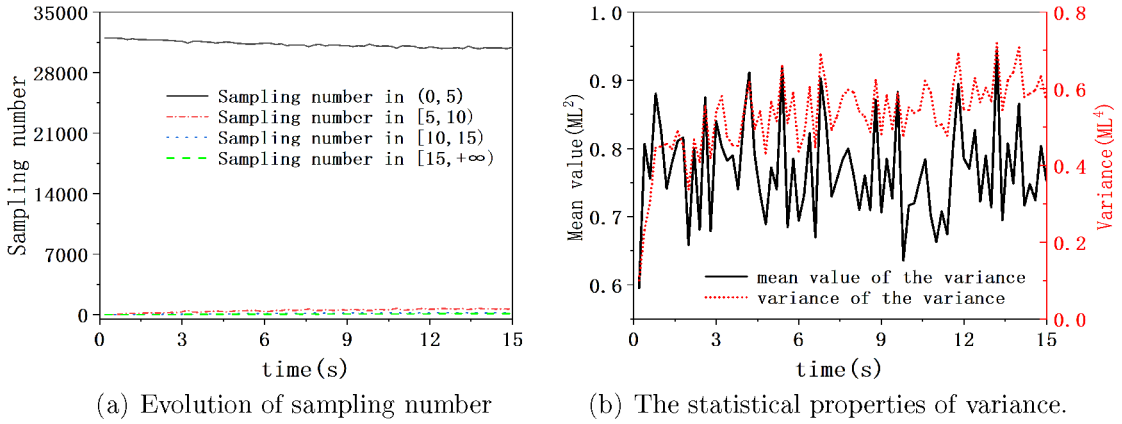


Figure 5. (a) Evolutions of variance sampling number in range (0, 5), [5, 10), [10, 15), [15, + ∞) and (b) the statistical properties of variance.

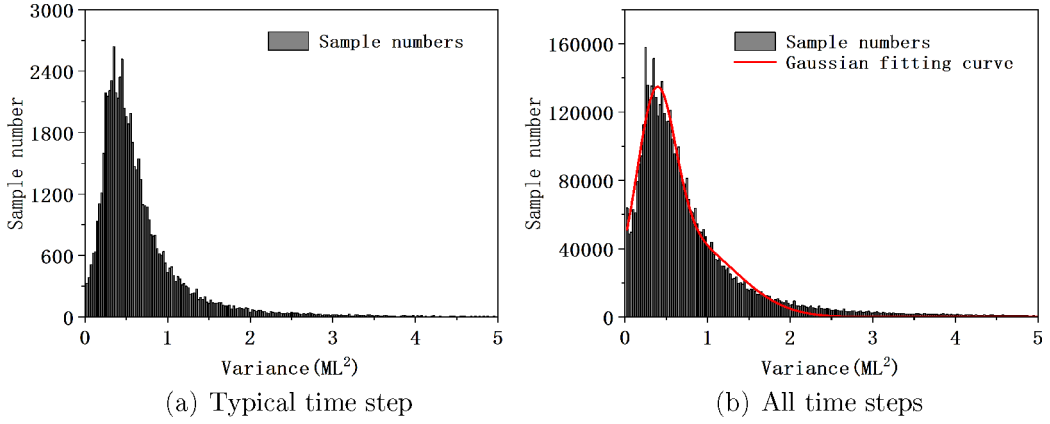


Figure 6. (a) Sampling statistical histogram of the variance at a typical time step (at 10 s) and (b) of all time steps.

substrate. Because of the placed resistive mask, the etching trench is getting narrower over time. Therefore, the bottom roughness (r_b) of the substrate is defined by MSE in a narrower surface with size of 32 × 32 ML^2.

Figure 4 presents the numerical algorithm of the microscopic kMC model. All reaction and deposition rate expressions are presented in Table 1. Species denoted with (g) are gas phase products and the species denoted with (s) are solid surface products. The flux data is used to choose the type of the injected particle at the beginning of each time step. Then the injection trajectory is computed based on selections of particle initial position and injection angle. When the selected particle bombard a surface site, it will react with the atom in this site or reflect to vacuum. The etching product will also reflect to vacuum after a etching reaction occurs. When the etching product reaches one surface site, it will deposit on this site or reflect to vacuum again. A periodic boundary condition is implemented in all sidewalls of the virtual simulation box. All particles are not computed again when they reach the interface between the simulation box and the plasma chamber. All related reaction coefficients and expressions of functions can be found in literature.[3,49,51−55]

4.2. Development of RNN Model. The data for training the RNN$_k$ model come from single-run open-loop simulations of the kMC model. Specifically, the inputs are $F_{i,k}$ and V_B; the initial conditions and outputs are the height data of the 40 × 40 ML^2 uncovered surface sites at last time step t_k and current time step t_{k+1}, which are denoted as $S_{kMC}(t_k)$ and $S_{kMC}(t_{k+1})$, respectively. The open simulation time is set as 15 s, and the sampling period is 0.2 s. Thus, 75 sampling periods are included in each simulation. With 150 open-loop simulations, the size of the whole data set is 150 × 75 = 11 250.

It should be noted that RNN$_k$ is implemented to approximate the expectation of the microscopic process. Thus, the training data should be the average data of multiple-run, open-loop simulations under the same initial condition and inputs. In pre-experiments, we utilized data from single-run, open-loop simulations as training data set and the first validation data set and utilized average data from multiple-run, open-loop simulations as the second validation data set. Experiment results showed that loss values for the training data set, first validation data set, and second validation data set are close and all can be reduced to a desired threshold (10^{-6}). Thus, the single-run, open-loop data set was directly used to train the RNN$_k$ model. Specifically, there is one layer in the recurrent layer and four linear layers as well as three rectified linear unit (ReLU) layers in the output layer. The other settings of the training process for RNN$_k$ model are the same as those in the RNN$_L$ model.

4.3. Stochastic Characteristic Analysis. Due to the stochastic nature of kMC, the height data of the lattice surface sites will stochastically oscillate around the expectation values. In order to capture this important feature, the statistical properties of the lattice surface height data should be analyzed. The variances of the height data of the lattice are computed from the sampling data of multiple-run, open-loop simulations with the same initial conditions and inputs. Specifically, one simulation was carried out 100 times with same initial

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652

conditions and inputs. One open simulation lasts 15 s, and the sampling period is 0.2 s. Thus, 75 sampling periods are included in each simulation. With 20 open-loop simulations, the size of the whole data set is 20 × 75 × 100 = 150 000. In Figure 5, we show the evolutions of sampling numbers and the statistical properties of the height data variances. At all time steps, most of the height data are in the variance range of (0, 5). Besides of the first few time steps, the mean values of the height data variance oscillate around a constant value. Compared to the mean values, the variance of the height data variance presents a trend of increasing from 1 to 15 s. However, its increasing rate is relatively low. Thus, we assume that the height variance is in the range of (0, 5) and can be considered time-invariant.

The sampling statistical histograms of the variance at a typical time step (at 10 s) and of all time steps are shown in Figure 6, in which the variance range is divided into 200 parts and the corresponding sampling numbers are given. This sampling statistical histogram of all time steps and the sampling statistical histogram at 10 s are quite similar, which suggests that our previous assumption is reasonable. Subsequently, a Gaussian Mixture fitting curve is developed to approximate the statistical histogram of all time steps, which has two terms: $y = a_1\, e^{-(x-b_1/c_1)^2} + a_2\, e^{-(x-b_2/c_2)^2}$, where a_1 = 0.02083, b_1 = 0.3785, c_1 = 0.3211, a_2 = 0.009339, b_2 = 0.7318, and c_2 = 0.8555. In practical use, we normalized the sampling statistical histogram and utilized it as a discrete probability distribution function. The RNN_k model predicts the expectation of the lattice profile $S_{RNN}(i)$ and the probability distribution function computes a variance value for each site of the lattice $\varepsilon(i)$, where i denotes ith surface sites. The prediction profile for kMC model is $S_{RNN}(i) \pm \sqrt{\varepsilon(i)}$, and the whole prediction model that approximates the kMC model is named $RNN_{k,p}$.

5. CLOSED-LOOP CONTROLLER DESIGN

In this section, we present both the PI controller and the model predictive controller for the plasma etch process by utilizing the above RNN models as the prediction model. The control objective is to achieve a desired average etching depth and average bottom roughness at the final time. Specifically, the $RNN_{k,p}$-based predictive optimizer optimizes the set points (the particle fluxes $F_{i,k}$) of critical PI control loops of R_1, R_2, and P_{rf}. Such methodologies of customized PI controller tuning were also developed in previous works, in which the parameters of the PI controller were estimated by mathematical programming.[56,57] The $RNN_{k,p}$ model and the RNN_L model are utilized as prediction model in the development of the model predictive controller. It should be noted that the prediction model developed in previous sections is designed to predict system states within a single time step $[t_k, t_{k+1}]$. By utilizing previous prediction results as initial condition for current time step, long time horizon prediction can be achieved. From pre-experiments, long-time prediction error between the prediction model and the kMC model is reduced by adding the probability distribution function in the prediction model.

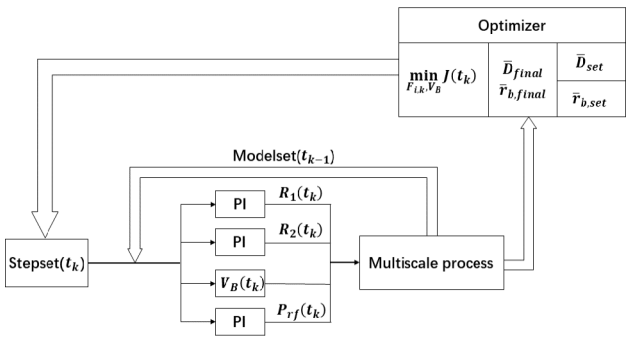


Figure 7. Block diagram of the closed-loop PI controller system.

5.1. PI Control. This PI controller is developed following our former work.[11] The block diagram of the closed-loop PI controller is developed in Figure 7. A $RNN_{k,p}$-based predictive optimizer is developed to optimize the set points of the PI controllers. Specifically, the prediction horizon is from the present time step t_k ($t_k = kt_s$) to the final time step t_{final}. The optimization problem can be formulated as

$$\min_{F_{i,k},V_B} J(t_k) = w_D(\bar{D}_{final} - \bar{D}_{set})^2 + w_r(\bar{r}_{b,final} - \bar{r}_{b,set})^2$$

$$\{\bar{D}_{final}, \bar{r}_{b,final}\} = RNN_{k,p}(F_{i,k}, V_B, \tilde{S}_{kMC})$$

$$\tilde{S}_{kMC} = F_{kMC}(t_k)$$

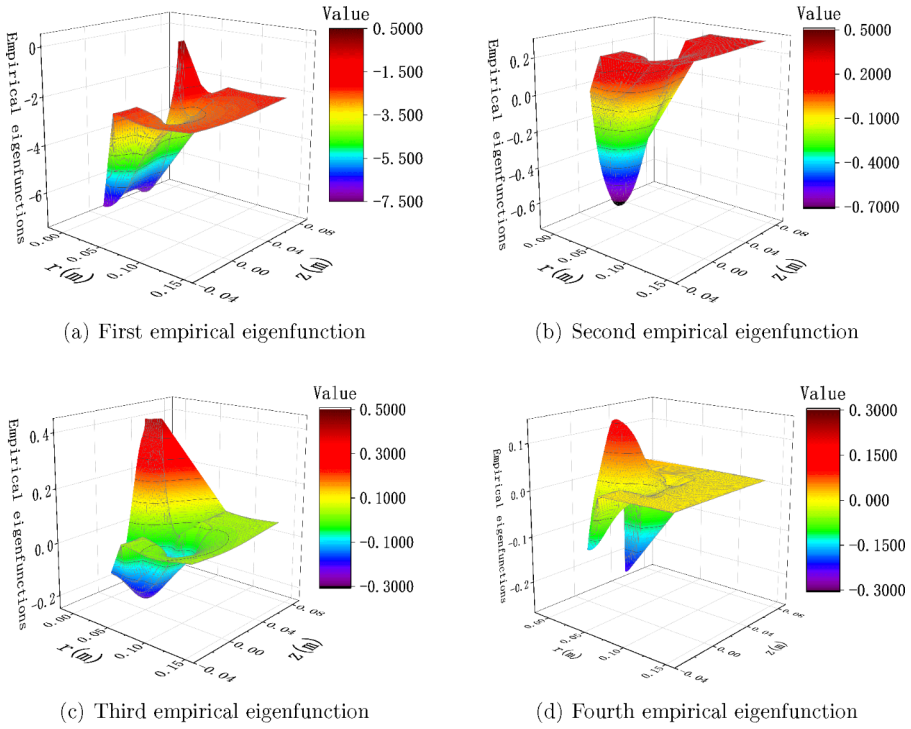
$$V_{min} \le V_B \le V_{max}$$

$$F_{i,k,min} \le F_{i,k} \le F_{i,k,max} \quad (20)$$

where w_D and w_r are the penalty weights on etching depth and bottom roughness, respectively; $\bar{D}_{final}$, $\bar{r}_{b,final}$, $\bar{D}_{set}$, and $\bar{r}_{b,set}$ are the average etching depth at final time, the average bottom roughness at the final time, the set value of the final average etching depth, and the set value of the final average bottom roughness, respectively. $F_{i,k}$ is the flux of the ith species at the kth location; $\tilde{S}_{kMC}(t_k)$ is the initial surface profile measured from kMC model. V_{min}, V_{max}, $F_{i,k,min}$, and $F_{i,k,max}$ are the lowest and the highest bounds of V_B and $F_{i,k}$, respectively. In this work, the nonlinear programming (NLP) tool box of MATLAB was implemented to solve the optimization problem. Specifically, the MultiStart method and fmincon function were used in our work. Specific settings are presented in the following sections. With the measurement data from the last time step, the optimizer computes the optimal trajectories of V_B and $F_{i,k}$, which are then used as the set points for the PI controllers to compute the control actions at next time step.

5.2. Model Predictive Control. The RNN_L model and the $RNN_{k,p}$ model are combined as the prediction model for the multiscale process. With initial states of the plasma chamber and values of R_1, R_2, P_{rf}, and V_B, this data-driven prediction model can predict the future etch profiles of the substrate. Specifically, in every time step t_k, an open-loop optimization problem is solved. The optimizing time ranges from the present time step t_k to the final time step t_{final}, and the optimized parameters are R_1, R_2, P_{rf}, and V_B. The optimization problem is presented below:

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652



(a) First empirical eigenfunction (b) Second empirical eigenfunction

(c) Third empirical eigenfunction (d) Fourth empirical eigenfunction

Figure 8. Four dominant eigenfunctions computed by POD for the obtained ensemble data of Cl species.

$$\min_{R_1,R_2,P_{rf},V_B} J(t_k) = w_D(\bar{D}_{final} - \bar{D}_{set})^2 + w_r(\bar{r}_{b,final} - \bar{r}_{b,set})^2$$

$$\{\bar{D}_{final}, \bar{r}_{b,final}\} = F_{RNN}(R_1, R_2, P_{rf}, V_B, \tilde{c}(t_k), \tilde{S}_{kMC})$$

$$\tilde{c}(t_k) = c(t_k)$$

$$\tilde{S}_{kMC} = F_{kMC}(t_k)$$

$$R_{min} \le R_1 \le R_{max}$$

$$R_{min} \le R_2 \le R_{max}$$

$$P_{min} \le P_{rf} \le P_{max}$$

$$V_{min} \le V_B \le V_{max} \quad (21)$$

where F_{RNN} is the multiscale prediction model that consists of the RNN_L and $RNN_{k,p}$ models; R_{min}, R_{max}, P_{min}, P_{max}, V_{min}, and V_{max} are the lowest and the highest bounds of the optimized parameters; the initial coefficients $\tilde{c}(t_k)$ and the initial etch profile $\tilde{S}_{kMC}(t_k)$ come from $c(t_k)$ (measurement of current plasma state) and $F_{kMC}(t_k)$ (measurement of current etch profile). Similarly, this optimization problem is solved in MATLAB with NLP tool box. The optimal inputs trajectories computed by MPC are applied to the multiscale process within the next time step. The MPC is recomputed with new measurements from the multiscale model at the next sampling time.

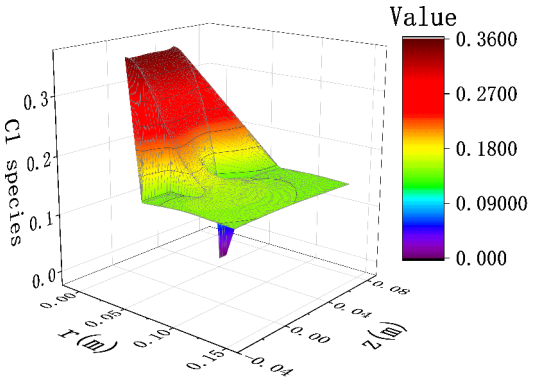
6. SIMULATION RESULTS

The plasma chemistry and microscopic structure of the etched lattice have been described in section 2. The frequency of the top coils is 13.56 MHz; the chamber temperature is 60 °C. The chamber pressure is 40 mTorr. In the following sections, we will present the computation process of the empirical eigenfunctions and Galerkin's method, the validation process of the RNN models, and the closed-loop simulation results under the PI controller as well as the model predictive controller.

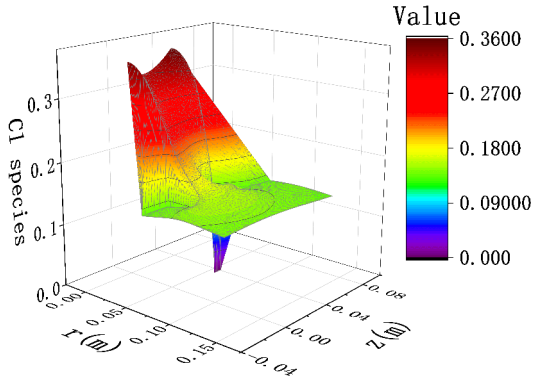
6.1. Computation of Empirical Eigenfunctions and Galerkin's Method. As we mentioned in previous sections, we computed five sets of eigenfunctions for five species in the plasma chamber, based on the ensemble of solutions data obtained by solving the plasma model in COMSOL. With combinations of different initial conditions and inputs, we generated 69 600 snapshots of the plasma states. According to pre-experiments, 11 600 snapshots with nonuniform time intervals were extracted to utilize in POD and compute the eigenfunctions. We note that in order to capture the dominant dynamics of the PDE system, the low-order ODE system should account for 99.9% energy included in the ensemble. The resulting numbers of the required empirical functions for Cl, Cl_2, Cl^+, Cl_2^+, and Ar^+ were 4, 5, 4, 5, and 5, respectively. In Figure 8, we show the four dominant empirical eigenfunctions of Cl species, which are arranged in descending order of corresponding eigenvalues.

The computed empirical eigenfunctions were then used as the basis functions and employed in Galerkin's method. Using the empirical eigenfunctions to decompose the solution of plasma model, we obtained the coefficients $c(t)$ and derived the low-order Galerkin's model. In Figure 9, a high-order solution of Cl computed by COMSOL and the corresponding four-order Galerkin's solution are compared. It can be seen that in most areas the low-order model captures the dominant dynamics of the high-order model, except the middle area of the plasma chamber, where the plasma dynamic change is the most dramatic. From validation experiments, the single time step average relative errors between the high-order systems and the low-order systems of Cl, Cl_2, Cl^+, Cl_2^+, and Ar^+ were 0.01968, 0.00748, 0.02225, 0.02192, and 0.01906, respectively. These average errors were computed based on 400 solutions,

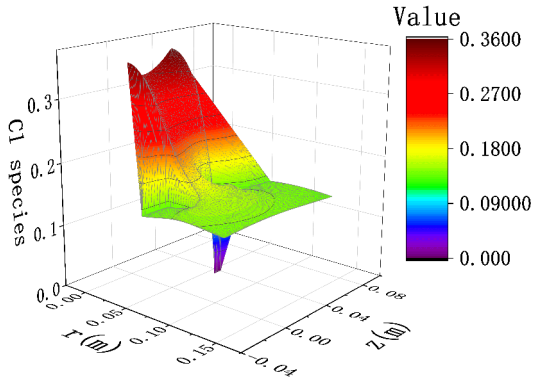
https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652



(a) High-order solution of Cl



(b) Low-order Galerkin's solution of Cl



(c) Predicted profile of Cl

Figure 9. (a) High-order solution of Cl computed by COMSOL. (b) Corresponding four-order Galerkin's solution. (c) Corresponding predicted profile computed by RNN_L model.

which were extracted randomly from our open-loop simulation results. The simulation results show that our low-order model is capable of approximating the dominant dynamics of the plasma states.

6.2. Validation of RNN Models. In this section, we present the validation process to verify the prediction performance of our RNN models. The RNN_L model, which consists of five sub RNN models, was constructed to predict the coefficients $c(t)$ of the low-order systems. In Figure 9, the low-order solution of Cl species and the corresponding predicted profile computed by RNN_L model are shown. According to the validation experiments, the errors between the low-order systems and the RNN_L model were relatively small compared to the error between the high-order systems and the low-order systems. The single time step average relative errors of Cl, Cl_2, Cl^+, Cl_2^+, and Ar^+ between the high-order fluid model and the RNN_L model were 0.01966, 0.00763, 0.02217, 0.02186, and 0.01925, respectively. As above, these average errors were also computed based on 400 solutions, which were extracted randomly from our open-loop simulation results.

Furthermore, the RNN_k model was also validated. Figure 10 shows the evolutions of $\bar{D}$ and $\bar{r}_b$ as well as the minimum and maximum values of $\bar{D}$ and $\bar{r}_b$ among the 100 simulations of the kMC model. In Figure 10a,b, the solid lines represent the $\bar{D}$ and $\bar{r}_b$ evolutions of the kMC model and the dotted lines represent the predicted $\bar{D}$ and $\bar{r}_b$ evolutions of the RNN_k model. In Figure 10c,d, the solid lines represent the maximum values, and the dotted lines represent the minimum values. The $\bar{D}$ and $\bar{r}_b$ data are sampled every 0.2 s, and the model inputs are randomly selected from the selection range at the beginning of every 0.2 s. The evolutions computed by the kMC model are the average of the 100 run simulations in order to obtain the expectation evolution. From Figure 10a,b, it can be seen that the predicted evolutions of $\bar{D}$ and $\bar{r}_b$ are close to the expectation evolutions of kMC model. From Figure 10c,d, it can be seen that the average bottom roughness is more influenced by the stochastic nature of the etching process compared to the results of the average etching depth.

6.3. Closed-Loop Simulations. In our experiments, a disturbance of P_{rf} (set to be maximum value 1400 W when disturbance occurs) was added on both open-loop simulations and closed-loop simulations from 2 to 5 s. The control objectives were the final average etching depth and final bottom roughness of a 10 s open-loop simulation (without disturbance) with the initial conditions $R_1 = 0.12$, $R_2 = 0.12$, $P_{rf} = 1000$ W, and $V_B = 150$ V. The open-loop simulations were operated 100 times, and the average results were generated to eliminate the randomness of the etching process. The finial objectives are $r_{set} = 11.0977$ ML^2 and $D_{set} = 40.5293$ ML. Two experiments were carried out with different ranges of the control variables. The optimization problems were solved by MultiStart method and fmincon function in MATLAB. The specific settings were as follows: optimization algorithm, sequential quadratic programming (SQP); difference step size, 0.01; maximum number of function evaluations, 1000; number of random initializations, 10.

The first experiment was performed with the control variable ranges $0.05 \le R_1 \le 0.2$, $0.05 \le R_2 \le 0.2$, 900 W $\le P_{rf} \le$ 1100 W, 120 V $\le$ Bias $\le$ 180 V. The evolutions of the average etching depth and average bottom roughness as well as the evolutions of control variables R_1, R_2, P_{rf}, and V_B are shown in Figure 11. It can be seen that the disturbance of P_{rf} is added at 2 s and ends at 5 s. The etching rate increases due to this disturbance, which causes the average etching depth and average bottom roughness deviate from their expected evolutions. The final average etching depth and average final bottom roughness of the open-loop simulations with disturbance finally showed about 9.73 and 25.24% deviation at the final time step, respectively. While the deviations of the final average etching depth and final bottom roughness between the desired objectives and the results under PI controller were 0.53 and 2.45%. The deviations for model predictive controller were 0.94 and 2.18%. Both the PI

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652

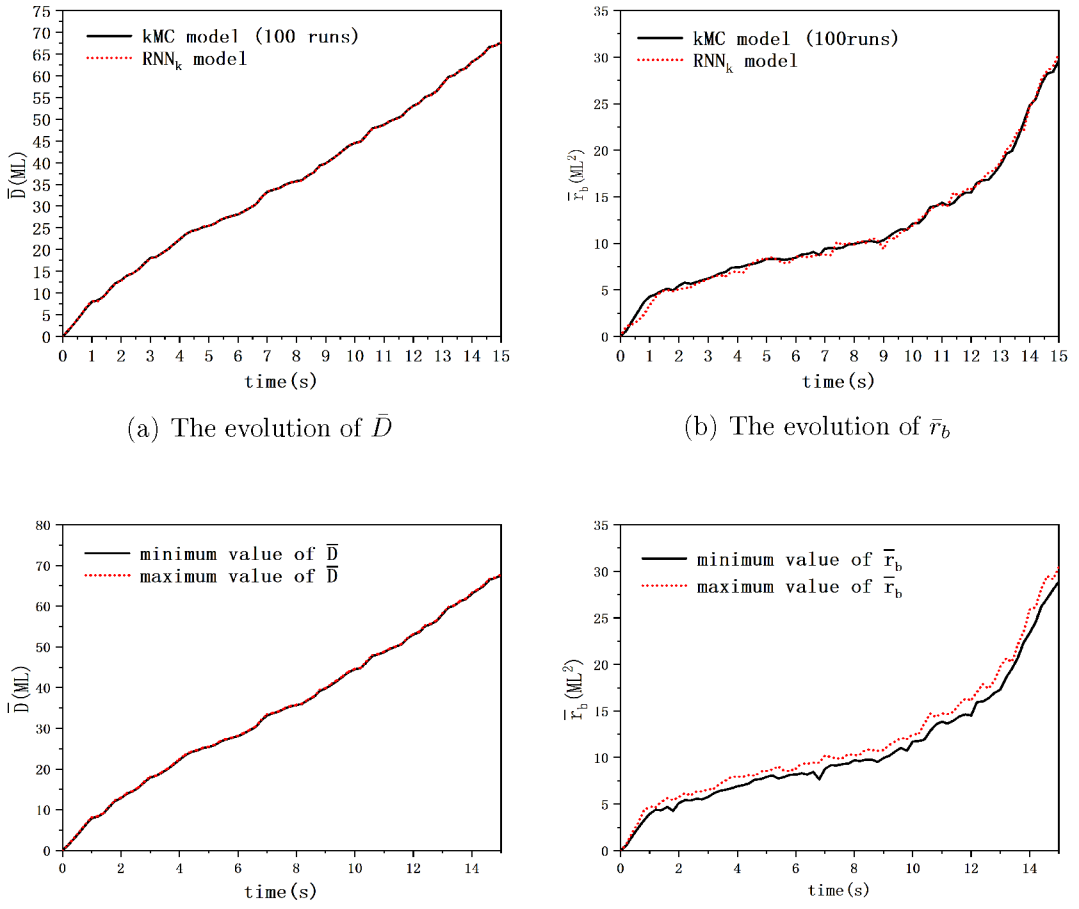


Figure 10. Evolutions of (a) $\bar{D}$ and (b) $\bar{r}_b$. Minimum and maximum values of (c) $\bar{D}$ and (d) $\bar{r}_b$.

controller and the model predictive controller presented sufficient performance and managed to complete the desired objectives by compensate the disturbance through modifying the control variables.

The second experiment was performed with a narrower control variable ranges $0.01 \leq R_1 \leq 0.15$, $0.10 \leq R_2 \leq 0.15$, $950\ \text{W} \leq P_{rf} \leq 1050\ \text{W}$, $130\ \text{V} \leq \text{Bias} \leq 170\ \text{V}$. The evolutions of the average etching depth and bottom roughness as well as the evolutions of control variables R_1, R_2, P_{rf}, and V_B are shown in Figure 12. In this experiment, the deviations of the final average etching depth and final average bottom roughness between the desired objectives and the results under PI controller were 4.78 and 10.89%, and the deviations for model predictive controller were 1.11 and 4.52%. The desired etching roughness was achieved by the model predictive controller, while the resulting error of etching depth under the PI controller was unacceptable in nanoscale etching process. Though the resulting bottom roughness deviation under the model predictive controller was still relatively large, the model predictive controller performed much better than the PI controller to reduce the influence of the disturbance. This difference can be partly explained by that the model predictive controller is able to predict the expected evolution and attain the states of the whole multiscale system, while the PI controller is implemented with an optimizer that only predicts the future profiles of the microscopic system, which makes it unable to timely adjust the plasma parameters. In the first experiment, it is possible for the PI controller to achieve the control objectives by adjusting the control variables when a wide range of actions is allowed. However, in the second experiment, the PI controller cannot approach the control objectives with narrow box constraints. Other reasons for the poor performance of the PI controller are the integral windup and chattering problem. Although an antiwindup approach was implemented in the PI controller, the integral windup problem could still lead to significance deviation when narrow box constraints were utilized. Moreover, chattering is observed from the results of the PI controller. From Figure 11a and 12a,c, we can see that control variables under PI controller oscillate during a period of time, which can lead to poor performance of the controller and is disadvantageous in manufacturing. As a contrast, the model predictive controller is able to predict the expected evolution and attain the states of the whole multiscale system, which enables it to adjust the control variables relatively stably and predictively within the narrow box constraints.

Remark 1. *In future research, we are interested in including more model order reduction techniques in our framework, such as dynamic mode decomposition method.*[58,59] *Identification techniques that do not require large amount of data, such as subspace identification method,*[60] *are also potential research directions in our group. Moreover, we consider to include research on addressing the problem of input data colinearity in developing RNN-based dynamic models for MPC, in which multiple-step approaches were proposed and closed-loop data was used to develop RNN models.*[61]

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652

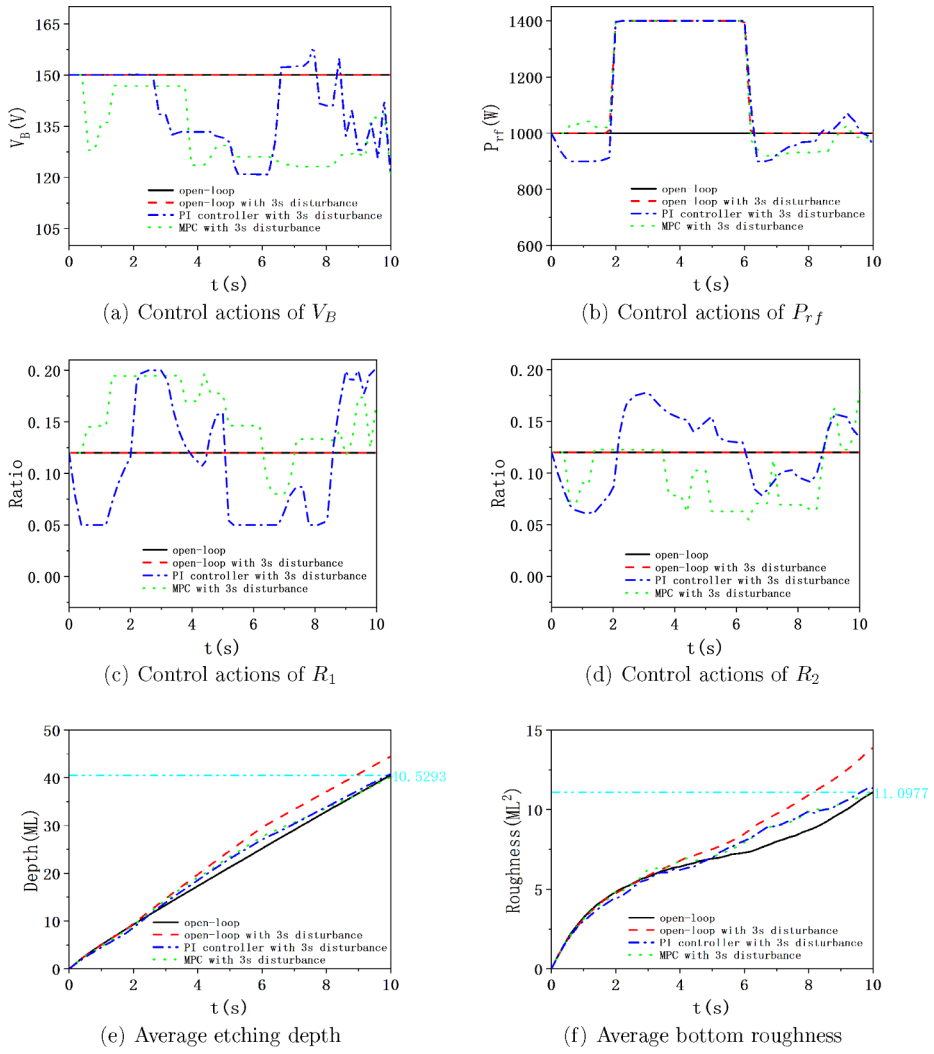


Figure 11. Profiles of (a) R_1, (b) R_2, (c) P_{rf}, and (d) V_B and the resulting evolutions of (e) $\bar{D}$ and (f) $\bar{r}_b$ when control variables are in the ranges of $0.05 \le R_1 \le 0.2$, $0.05 \le R_2 \le 0.2$, 900 W $\le P_{rf} \le$ 1100 W, 120 V $\le$ Bias $\le$ 180 V. The cyan double-dotted and dashed lines in (e) and (f) denote the control objectives.

Remark 2. *It should be noted that although the microscopic processes of the real etch process are the same given a specific etch chemistry the macroscopic dynamics and reaction kinetics of the plasma are highly equipment-dependent with respect to the accuracy needed for real applications. While such high fidelity models are not openly available, equipment vendors typically have their own proprietary plasma models for their plasma processing equipment, and these models can be easily integrated into our multiscale modeling and control framework when needed.*

7. CONCLUSIONS

An RNN-based MPC for the plasma etch process on a 3D substrate using ICP analysis was proposed. The plasma etch process was simulated by a multiscale model, in which the macroscopic plasma model was simulated by a continuous fluid model and the microscopic etching process was simulated by a kMC model. Empirical eigenfunctions were derived from the solution data of the plasma model by utilizing POD. These empirical eigenfunctions were applied in Galerkin's method as basis functions to compute a reduced order ODE system that capture the dominant dynamics of the plasma model. Subsequently, the RNN models were trained by open-loop simulation data to approximate the dynamics of both the reduced order plasma system and the microscopic etch process. A probability distribution function was analyzed to approximate the stochastic nature of the etch process. Utilizing the RNN models as the prediction model, the model predictive controller was designed to achieve desired etching depth and bottom roughness at the final time. In the simulations and discussions sections, the computation process of the empirical eigenfunctions and Galerkin's method as well as the validation process of the RNN models were shown, which suggested that the proposed model reduction framework is effective and the data-driven prediction models are sufficiently accurate. Furthermore, the closed-loop simulation results under the PI controller and model predictive controller were also presented. Though both controllers showed desired performance with wide box constraints on control variables, the model predictive controller outperformed the PI controller when a narrow range of control variables values was allowed. In both experiments, the model predictive controller prevented the chattering of the control variables, which the PI controller failed to eliminate. These results demonstrated the effectiveness of the proposed MPC framework.

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652

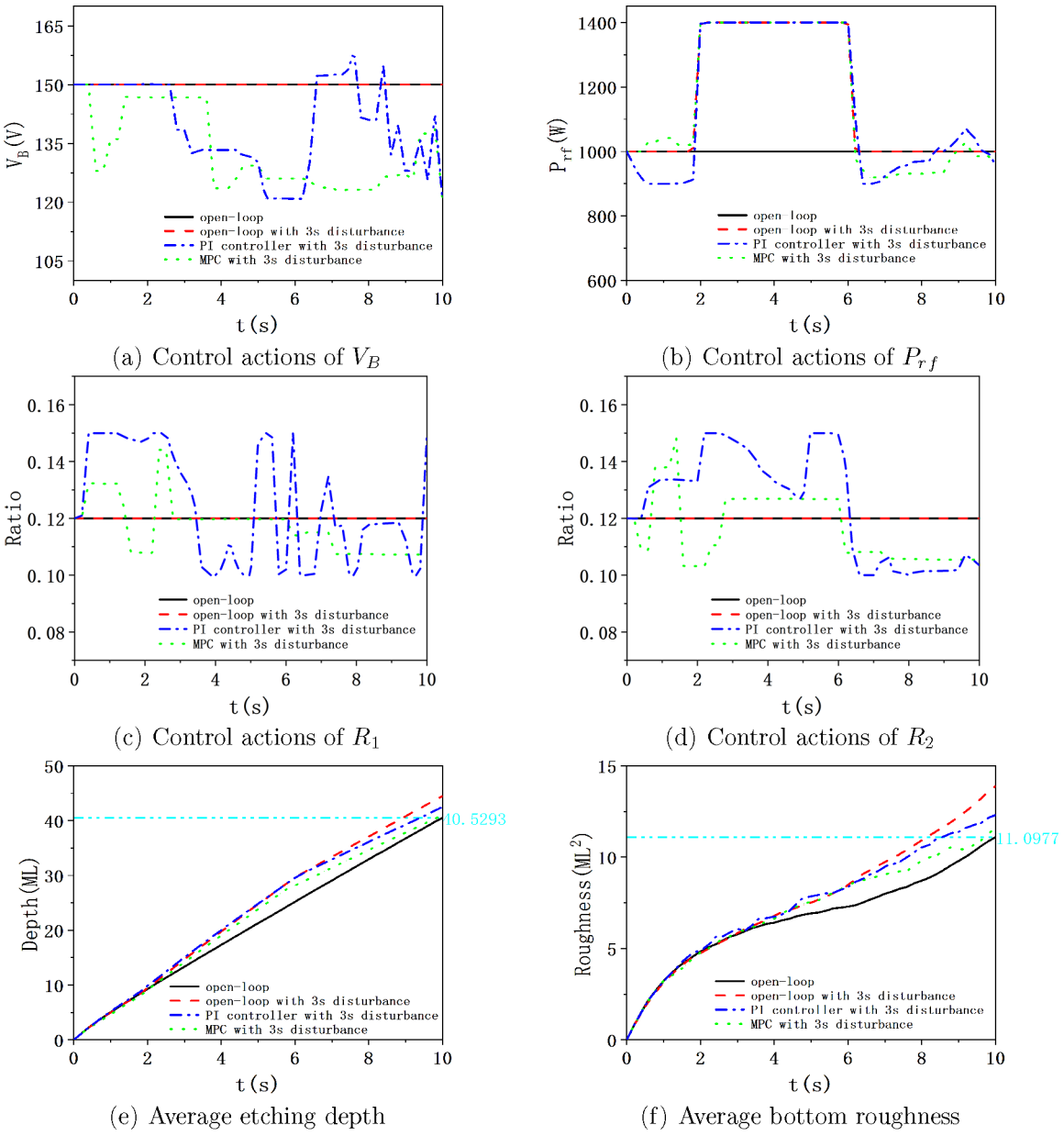


Figure 12. Profiles of (a) R_1, (b) R_2, (c) P_{rf}, and (d) V_B and the resulting evolutions of (e) $\bar{D}$ and (f) $\bar{r}_b$ when the control variables are in the following ranges: $0.10 \le R_1 \le 0.15$, $0.10 \le R_2 \le 0.15$, 950 W $\le P_{rf} \le$ 1050 W, 130 V $\le$ Bias $\le$ 170 V. The cyan double-dotted and dashed lines in (e) and (f) denote the control objectives.

AUTHOR INFORMATION

Corresponding Author

Dong Ni − *College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China;* orcid.org/0000-0002-2227-2555; Email: dni@zju.edu.cn

Authors

Tianqi Xiao − *College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China;* orcid.org/0000-0002-8624-2689

Zhe Wu − *Department of Chemical and Biomolecular Engineering, National University of Singapore, 117585, Singapore*

Panagiotis D. Christofides − *Department of Chemical and Biomolecular Engineering and Department of Electrical and Computer Engineering, University of California, Los Angeles, California 90095-1592, United States;* orcid.org/0000-0002-8772-4348

Antonios Armaou − *Department of Chemical Engineering, The Pennsylvania State University, University Park, Pennsylvania 16802, United States; Department of Mechanical Engineering, The Pennsylvania State University, University Park, Pennsylvania 16802, United States; Department of Chemical Engineering, University of Patras, 26243 Patras, Greece;* orcid.org/0000-0002-8592-7934

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.iecr.1c04251

Notes

The authors declare no competing financial interest.

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652

■ ACKNOWLEDGMENTS

The authors are grateful for the financial support from National Natural Science Foundation of China (grant no. U1609213).

■ REFERENCES

(1) Wu, B.; Kumar, A.; Pamarthy, S. High aspect ratio silicon etch: A review. *J. Appl. Phys.* **2010**, *108*, 051101.
(2) Donnelly, V.; Kornblit, A. Plasma etching: Yesterday, today, and tomorrow. *J. Vac. Sci. Technol., A* **2013**, *31*, 050825.
(3) Mahorowala, A. P.; Sawin, H. H. Etching of polysilicon in inductively coupled Cl_2 and HBr discharges. IV. Calculation of feature charging in profile evolution. *J. Vac. Sci. Technol., B: Microelectron. Process. Phenom.* **2002**, *20*, 1084−1095.
(4) Aydil, E.; Quiniou, B.; Lee, J.; Gregus, J.; Gottscho, R. Incidence angle distributions of ions bombarding grounded surfaces in high density plasma reactors. *Mater. Sci. Semicond. Process.* **1998**, *1*, 75−82.
(5) Chatterjee, A.; Vlachos, D. An overview of spatial microscopic and accelerated kinetic Monte Carlo methods. *J. Comput.-Aided Mater. Des.* **2007**, *14*, 253−308.
(6) Gosalvez, M.; Xing, Y.; Sato, K.; Nieminen, R. Atomistic methods for the simulation of evolving surfaces. *J. Micromech. Microeng.* **2008**, *18*, 055029.
(7) Guo, W.; Sawin, H. Review of profile and roughening simulation in microelectronics plasma etching. *J. Phys. D: Appl. Phys.* **2009**, *42*, 194014.
(8) Abbas, A.; Romagnoli, J. A. Multiscale modeling, simulation and validation of batch cooling crystallization. *Sep. Purif. Technol.* **2007**, *53*, 153−163.
(9) Son, S. H.; Choi, H.-K.; Kwon, J. S.-I. Multiscale modeling and control of pulp digester under fiber-to-fiber heterogeneity. *Comput. Chem. Eng.* **2020**, *143*, 107117.
(10) Raimondeau, S.; Vlachos, D. Recent developments on multiscale, hierarchical modeling of chemical reactors. *Chem. Eng. J.* **2002**, *90*, 3−23.
(11) Xiao, T.; Ni, D. Multiscale modeling and neural network model based control of a plasma etch process. *Chem. Eng. Res. Des.* **2020**, *164*, 113−124.
(12) Smith, G. D.; Smith, G. D.; Smith, G. D. S. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*; Oxford University Press, 1985.
(13) Bathe, K. J.; Saunders, H. Finite Element Procedures in Engineering Analysis. *J. Pressure Vessel Technol.* **1984**, *106*, 421−422.
(14) Christofides, P. D. *Nonlinear and Robust Control of PDE Systems: Methods and Applications to Transport-Reaction Processes*; Springer Science & Business Media, 2001.
(15) Graham, M. D.; Kevrekidis, I. G. Alternative approaches to the Karhunen-Loève decomposition for model reduction and data analysis. *Comput. Chem. Eng.* **1996**, *20*, 495−506.
(16) Baker, J.; Christofides, P. D. Finite-dimensional approximation and control of non-linear parabolic PDE systems. *Int. J. Control* **2000**, *73*, 439−456.
(17) Armaou, A.; Christofides, P. D. Dynamic optimization of dissipative PDE systems using nonlinear order reduction. *Chem. Eng. Sci.* **2002**, *57*, 5083−5114.
(18) Tarman, I.; Sirovich, L. Extensions to Karhunen-Loève based approximation of complicated phenomena. *Comput. Method Appl. M* **1998**, *155*, 359−368.
(19) Lumley, J. *Transition and Turbulence*; Elsevier, 1981; pp 215−242.
(20) Finlayson, B. A. *Nonlinear Analysis in Chemical Engineering*; Ravenaa Park Publishing, 2003.
(21) Dodhia, A.; Wu, Z.; Christofides, P. D. Machine learning-based model predictive control of diffusion-reaction processes. *Chem. Eng. Res. Des.* **2021**, *173*, 129−139.
(22) Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U. S. A.* **1982**, *79*, 2554−2558.
(23) Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput* **1997**, *9*, 1735−1780.
(24) Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder−Decoder for Statistical Machine Translation. *Proc. 2014 Conf. Empirical Methods Nat. Lang. Process.* **2014**, 1724−1734.
(25) Lukoševičius, M.; Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **2009**, *3*, 127−149.
(26) Funahashi, K.; Nakamura, Y. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Netw* **1993**, *6*, 801−806.
(27) Jaeger, H.; Haas, H. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science* **2004**, *304*, 78−80.
(28) Trischler, A. P.; D'Eleuterio, G. M. D. Synthesis of recurrent neural networks for dynamical system simulation. *Neural Netw* **2016**, *80*, 67−78.
(29) Mhaskar, P.; El-Farra, N. H.; Christofides, P. D. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *Syst. Control Lett.* **2006**, *55*, 650−659.
(30) Venkat, A. N.; Hiskens, I. A.; Rawlings, J. B.; Wright, S. J. Distributed MPC strategies with application to power system automatic generation control. *IEEE Trans Control Syst. Technol.* **2008**, *16*, 1192−1206.
(31) Dufour, P.; Touré, Y.; Blanc, D.; Laurent, P. On nonlinear distributed parameter model predictive control strategy: on-line calculation time reduction and application to an experimental drying process. *Comput. Chem. Eng.* **2003**, *27*, 1533−1542.
(32) Lao, L.; Ellis, M.; Christofides, P. D. Economic model predictive control of parabolic PDE systems: Addressing state estimation and computational efficiency. *J. Process Control* **2014**, *24*, 448−462.
(33) Varshney, A.; Armaou, A. Reduced order modeling and dynamic optimization of multiscale PDE/kMC process systems. *Comput. Chem. Eng.* **2008**, *32*, 2136−2143.
(34) Pitchaiah, S.; Armaou, A. Predictive control of PDEs via using adaptive reduced order modeling. *CDC-ECC 2011* **2011**, 2004−2009.
(35) Varshney, A.; Armaou, A. Identification of Macroscopic Variables for Low-Order Modeling of Thin-Film Growth. *Ind. Eng. Chem. Res.* **2006**, *45*, 8290−8298.
(36) Varshney, A.; Armaou, A. *Model Reduction and Coarse-Graining Approaches for Multiscale Phenomena*; Springer: Berlin Heidelberg, 2006; pp 515−533.
(37) Van Overschee, P.; De Moor, B. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica* **1994**, *30*, 75−93.
(38) Viberg, M. Subspace-based methods for the identification of linear time-invariant systems. *Automatica* **1995**, *31*, 1835−1851.
(39) Pan, Y.; Wang, J. Nonlinear model predictive control using a recurrent neural network. *Int. Jt. Conf. Neural Networks, Proc.* **2008**, 2296−2301.
(40) Xu, J.; Li, C.; He, X.; Huang, T. Recurrent neural network for solving model predictive control problem in application of four-tank benchmark. *Neurocomputing* **2016**, *190*, 172−178.
(41) Wu, Z.; Tran, A.; Rincon, D.; Christofides, P. D. Machine learning-based predictive control of nonlinear processes. Part I: Theory. *AIChE J.* **2019**, *65*, e16729.
(42) Wu, Z.; Tran, A.; Rincon, D.; Christofides, P. D. Machine-learning-based predictive control of nonlinear processes. Part II: Computational implementation. *AIChE J.* **2019**, *65*, e16734.
(43) Holmes, P.; Lumley, J. L.; Berkooz, G. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*; Cambridge Monographs on Mechanics; Cambridge Press, 1996.
(44) Hagelaar, G. J. M.; Pitchford, L. C. Solving the Boltzmann equation to obtain electron transport coefficients and rate coefficients for fluid models. *Plasma Sources Sci. Technol.* **2005**, *14*, 722−733.

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652

(45) Bogaerts, A.; Neyts, E.; Gijbels, R.; van der Mullen, J. Gas discharge plasmas and their applications. *Spectrochim. Acta, Part B* **2002**, *57*, 609−658.
(46) Pitchford, L. LXCat: A web-based, community-wide project on data for modeling low temperature plasmas. *67th Annu. Gaseous Electron. Conf. 2014 (November 2−7, 2014; Raleigh, NC)* **2014**, *59*, BAPS.2014.GEC.AM2.5.
(47) GEC ICP Reactor, Argon/OxygenChemistry. https://www.comsol.com/model/gec-icp-reactor-argon-chemistry-8649.
(48) Chanson, R.; Rhallabi, A.; Fernandez, M.; Cardinaud, C.; Bouchoule, S.; Gatilova, L.; Talneau, A. Global Model of Cl_2/Ar High-Density Plasma Discharge and 2-D Monte-Carlo Etching Model of InP. *IEEE Trans Plasma Sci.* **2012**, *40*, 959.
(49) Tinck, S.; Boullart, W.; Bogaerts, A. Investigation of etching and deposition processes on Cl_2/O_2/Ar inductively coupled plasmas on silicon by means of plasma surface simulations and experiments. *J. Phys. D: Appl. Phys.* **2009**, *42*, 095204.
(50) Xiao, T.; Ni, D. Multiscale Modeling and Recurrent Neural Network Based Optimization of a Plasma Etch Process. *Processes* **2021**, *9*, 151.
(51) Osano, Y.; Ono, K. An Atomic Scale Model of Multilayer Surface Reactions and the Feature Profile Evolution during Plasma Etching. *Jpn. J. Appl. Phys.* **2005**, *44*, 8650.
(52) Osano, Y.; Mori, M.; Itabashi, N.; Takahashi, K.; Eriguchi, K.; Ono, K. A Model Analysis of Feature Profile Evolution and Microscopic Uniformity during Polysilicon Gate Etching in Cl_2/O_2 Plasmas. *Jpn. J. Appl. Phys.* **2006**, *45*, 8157−8162.
(53) Chiaramonte, L.; Colombo, R.; Fazio, G.; Garozzo, G.; La Magna, A. A numerical method for the efficient atomistic simulation of the plasma-etch of nano-patterned structures. *Comput. Mater. Sci.* **2012**, *54*, 227−235.
(54) Chang, J. P.; Sawin, H. H. Kinetic study of low energy ion-enhanced polysilicon etching using Cl, Cl_2, and Cl^+ beam scattering. *J. Vac. Sci. Technol., A* **1997**, *15*, 610−615.
(55) Agarwal, A.; Kushner, M. Plasma atomic layer etching using conventional plasma equipment. *J. Vac. Sci. Technol., A* **2009**, *27*, 37−50.
(56) Kookos, I.K.; Lygeros, A.I.; Arvanitis, K.G. On-Line PI Controller Tuning for Integrator/Dead Time Processes. *Eur. J. Control* **1999**, *5*, 19−31.
(57) Syrcos, G.; Kookos, I. K. PID controller tuning using mathematical programming. *Chem. Eng. Process.* **2005**, *44*, 41−49.
(58) Otto, S. E.; Rowley, C. W. Linearly Recurrent Autoencoder Networks for Learning Dynamics. *Siam J. Appl. Dys Syst* **2019**, *18*, 558−593.
(59) Li, Q.; Dietrich, F.; Bollt, E. M.; Kevrekidis, I. G. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos* **2017**, *27*, 103111.
(60) Corbett, B.; Mhaskar, P. Subspace identification for data-driven modeling and quality control of batch processes. *AIChE J.* **2016**, *62*, 1581−1601.
(61) Hassanpour, H.; Corbett, B.; Mhaskar, P. Integrating dynamic neural network models with principal component analysis for adaptive model predictive control. *Chem. Eng. Res. Des.* **2020**, *161*, 26−37.

https://doi.org/10.1021/acs.iecr.1c04251
Ind. Eng. Chem. Res. 2022, 61, 638−652