IISE TRANSACTIONS
2024, VOL. 56, NO. 6, 585–599
https://doi.org/10.1080/24725854.2023.2219290

Reinforcement learning for process control with application in semiconductor manufacturing

Yanrong Li[a], Juan Du[b,c], and Wei Jiang[a]

[a]Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai, China; [b]Smart Manufacturing Thrust, Systems Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China; [c]Department of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China

ABSTRACT

Process control is widely discussed in the manufacturing process, especially in semiconductor manufacturing. Due to unavoidable disturbances in manufacturing, different process controllers are proposed to realize variation reduction. Since Reinforcement Learning (RL) has shown great advantages in learning actions from interactions with a dynamic system, we introduce RL methods for process control and propose a new controller called RL-based controller. Considering the fact that most existing run-to-run (R2R) controllers mainly rely on a linear model assumption for the process input–output relationship, we first discuss theoretical properties of RL-based controllers based on the linear model assumption. Then the performance of RL-based controllers and traditional R2R controllers (e.g., Exponentially Weighted Moving Average (EWMA), double EWMA, adaptive EWMA, and general harmonic rule controllers) are compared for linear processes. Furthermore, we find that the RL-based controllers have potential advantages to deal with other complicated nonlinear processes. The intensive numerical studies validate the advantages of the proposed RL-based controllers.

ARTICLE HISTORY
Received 12 October 2021
Accepted 7 May 2023

KEYWORDS
Process control; reinforcement learning; semiconductor manufacturing

1. Introduction

Process control is necessary for reducing variations in manufacturing processes to improve the quality and productivity of final products. For example, in semiconductor manufacturing, different unavoidable disturbances (e.g., tool-induced and product-induced disturbances) caused by various factors can influence the stability of manufacturing processes and the quality of final products (Su *et al.*, 2007). Therefore, designing an efficient control strategy that reduces variations due to various disturbances is an important research problem in semiconductor manufacturing.

In practice, sensors installed in production lines can collect various data during the manufacturing process, including the system's inputs and output quality characteristics. Control recipes based on the collected data need to be optimized to compensate for disturbances (Del Castillo and Hurwitz, 1997). In the semiconductor manufacturing process, existing research can be categorized into the following two groups: (i) theoretical and (ii) practical perspectives of controller designs to realize variation reduction and quality improvement.

In terms of the theoretical design of controllers, the existing extensive research has generally made assumptions on the process model with *predefined* disturbances followed by corresponding designs of the controller for a semiconductor manufacturing process. For example, Ingolfsson and Sachs (1993) introduced the Exponentially Weighted Moving Average (EWMA) controller for the integrated moving average (IMA) process disturbance. Based on this work, more complicated disturbance models such as Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA) processes are discussed. For example, Tsung and Shi (1999) focused on the ARMA (1,1) disturbance and proposed the Proportional-Integral-Derivative (PID) controller to deal with ARMA disturbance. Chen and Guo (2001) paid more attention to the linear drift process and proposed the age-based double EWMA controller to improve the performance of the traditional EWMA controller. Tseng *et al.* (2003) proposed the Variable-EWMA (VEWMA) controller to optimize the discount factor of EWMA in the ARIMA process and found that the VEWMA controller is easy to implement by calculating the optimal discount factor. Furthermore, more general controllers have also been proposed to deal with various kinds of disturbances. For example, He *et al.* (2009) proposed a General Harmonic Rule (GHR) controller and proved that the GHR controller could handle IMA (1,1), ARMA (1,1), and ARIMA (1,1) disturbances quite well. These theoretical controllers provide a foundation for process control in semiconductor manufacturing, which inspires many extensions in industry practice.

In terms of practical extensions, specific applications are considered in the control scheme according to the manufacturing scenarios. For example, Wang and Han (2013)

CONTACT Juan Du juandu@ust.hk
Supplemental data for this article is available online at https://doi.org/10.1080/24725854.2023.2219290

proposed a batch-based EWMA controller by modifying a K-means algorithm to group the incoming material into batches with fixed sizes while minimizing the within-batch variation. Liu *et al.* (2018) reviewed comprehensive literature on batch-based processes and summarized control principles and simulation examples of various controllers. Huang and Lv (2020) improved the EWMA controller by considering the online measurement in the batch production process. In addition to the considerations of batch-based characteristics, other cases have also been discussed. For example, Djurdjanović *et al.* (2017) proposed a robust automatic control method based on inaccurate knowledge about process noises and applied it to the lithography processes.

Existing works make influential contributions to semiconductor manufacturing, and a linear process model with certain disturbances is usually assumed. In practice, the process model may not be linear, and the disturbance can be different from the assumption. Therefore, using a *predefined* fixed linear model with one type of disturbance to describe the input–output relationship can experience difficulties in accurately fitting the entire manufacturing procedure (Wang and Shi, 2021). A more flexible controller that can be applied in various cases is desired.

Reinforcement Learning (RL) is a powerful data-driven method to learn actions in dynamic environments or systems without assuming process models or disturbances (Kaelbling *et al.*, 1996). RL is designed to minimize the total cost by learning the relationship between the input actions and outputs directly from historical control and output data (Sutton and Barto, 2018). Due to the comprehensive consideration of the real-time system output and historical control strategies, RL-based control methods are widely applied in different fields. In the literature, several review works have focused on applications using the RL-based control scheme. For example, Naeem *et al.* (2020) introduced RL methodologies and reviewed the corresponding control applications, such as robotics and pricing control. For robotics control problems, prior research has shown great performances of RL and began to maximize the reward function when applied to physical robots early in the 1990s (Mataric, 1994). With the explorations of RL in robotics over 20 years, Kober et al. (2013) summarized different RL-based methodologies in robotics control problems and categorized the corresponding algorithms according to different criteria. Recently, the deep RL methodology has become more popular in robotics control. Gu *et al.* (2017) demonstrated a deep RL algorithm based on deep Q-functions and applied it to complex 3D manipulation tasks. He *et al.* (2020) proposed an RL-based control system with a deep actor-critic structure and applied it to a flexible two-link manipulator of a vibration suppression system. For pricing control problems, RL-based methods are usually applied in the dynamic pricing control of public resources. For example, Kutschinski *et al.* (2003) examined several adaptive pricing strategies by RL methodology in electronic marketplaces. Wang *et al.* (2019) proposed RL-based approaches for optimizing pricing strategies to maximize the profits of public electric vehicle charging stations. Pandey *et al.* (2020) developed a deep RL framework for dynamic pricing on express lanes with multiple access locations and heterogeneity in travelers to minimize the total travel time.

Similar to the aforementioned applications, RL-based control methods also have the potential to handle various cases of manufacturing process control problems. Taking the semiconductor manufacturing process as an example, traditional run-to-run (R2R) controllers are designed based on *predefined* process models with disturbances. However, the assumptions of process models and disturbances may not be accurate, which brings difficulties for real applications. RL is a data-driven method that does not rely on the *predefined* model and can learn control strategies from offline data. This advantage brings opportunities for designing an RL-based controller for complex processes. However, to our best knowledge, few works deal with process control problems in semiconductor manufacturing by RL. Meanwhile, the existing RL-based control methods cannot be directly applied in semiconductor manufacturing, due to the effects of disturbances, which are autocorrelated in manufacturing processes but cannot be observed directly from data. Therefore, we fill the research gap by developing RL-based controllers for process control in semiconductor manufacturing. Following the control objective of the semiconductor manufacturing process, our RL-based controllers are also designed to minimize the errors between real-time system outputs and their target levels. From the perspective of RL, the real-time outputs reflect the system states. The mean of squared errors at each run is defined as the control cost, which corresponds to the negative reward in the RL framework.

Our contributions can be summarized as follows:

1. we first propose RL-based controllers, which can be applied in different cases. If domain knowledge is available, RL-based controllers with different approximate process models (such as linear or nonlinear) can be developed accordingly. Otherwise, if there is no evidence to assume a proper model, the RL-based controller with policy gradient search method can be applied.
2. Two computational algorithms for RL-based controllers are presented according to whether domain knowledge is available or not;
3. Theoretical properties are investigated for RL-based controllers given the assumption of the widely accepted linear process models.

The remainder of this article is organized as follows. Section 2 provides the methodology of RL-based controllers including the model formulations, algorithms, and theoretical properties. For fair comparisons with conventional controllers, Section 3 introduces two classic linear simulation case studies in the semiconductor manufacturing process. To further validate the performance of RL-based controllers, Section 4 presents two simulation cases with other complicated nonlinear process models, where traditional controllers are inapplicable. Section 5 summarizes the conclusions and future research.

2. Methodology of RL in process control

In this section, we first introduce the formulation of the process control problems. Then, the methodology of RL-based controllers is developed. Specifically, if the domain knowledge is available for process model approximation, the RL-based controller with approximate models is proposed in Algorithm 1. Otherwise, we propose the RL-based controller with policy gradient search in Algorithm 2 to find the optimal control strategies. In addition, theoretical properties are analyzed to guarantee the performance of the RL-based controller under the generalized linear assumption.

2.1. Problem definition and formulation

In this work, we consider a sequential process control problem at one manufacturing stage (such as chemical mechanical planarization or deep reactive ion etching process) in semiconductor manufacturing. The set of corresponding sampling times at each run is denoted as $\mathbf{T} = \{1, 2, \ldots t, \ldots, T\}$, where T is the total run number. The manufacturing process from run 1 to T is defined as a complete production cycle. Notably, the total run number T may vary across different production cycles. Denote y_t as the system output at run t, where y_0 is the initial system output. Following the process control model in Del Castillo and Hurwitz (1997), we have $y_t = g(u_t, d_t)$ for $t \in \mathbf{T}$, where d_t is the process disturbance of the manufacturing system that has autocorrelations, and u_t is the control action at t. Figure 1 illustrates the process control problem in the semiconductor manufacturing process. During the manufacturing process, there are unavoidable disturbance processes that can influence the system output, and control action u_t is necessary to compensate for the effect of d_t.

Our objective is to keep y_t close to the target level y^* in each run $t \in \{1, 2, \ldots, T\}$ in terms of squared loss to achieve a high-level process capability and quality (Wang and Han, 2013). Specifically, we would like to obtain the control action u_t by solving the optimization problem as follows:

$$\begin{aligned} &\min_{u_t} \sum_{t=1}^{T} (y_t - y^*)^2 \\ &\text{s.t.} \quad y_t = g(u_t, d_t). \end{aligned} \tag{1}$$

Generally, if the process model is unknown, the underlying model needs to be discussed. One of the most effective solutions is to find an approximate process model $f(\cdot)$ and minimize the difference between $f(\cdot)$ and $g(\cdot)$. In practice, $f(\cdot)$ can be obtained from the following optimization problem when a set of observations $\{y_t\}$ is available:

$$\min_{f(\cdot)} \sum_{t=1}^{T} \|y_t - f(\cdot)\|^2. \tag{2}$$

Theoretically, the approximate function $f(\cdot)$ can be any form and a proper $f(\cdot)$ is necessary to describe the real system, but difficult to be determined. Therefore, in the following subsections, we discuss two scenarios to specify $f(\cdot)$ with and without domain knowledge, respectively.

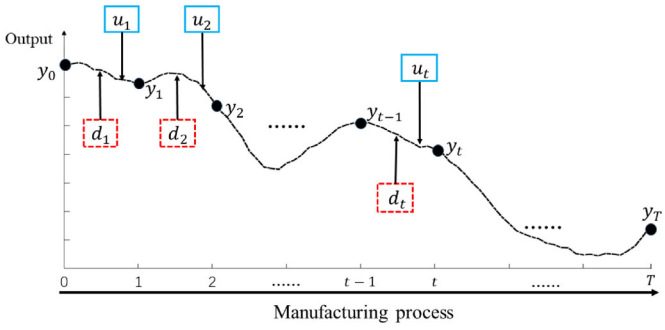

Figure 1. Illustration of semiconductor manufacturing control.

2.2. RL-based controller with domain knowledge

RL is one of the most important data-driven methods; it can interact with environments over time and select a real-time control action from action spaces based on the current system state to maximize rewards or minimize costs. Since the state transition function (i.e., process model $g(\cdot)$) is usually unknown, RL methodologies can be divided into two categories based on whether an approximate model can be fitted to denote the system dynamics. If an approximate model can arise from domain knowledge such as first principles, model-based RL algorithms can estimate parameters in the model and optimize control actions accordingly. Otherwise, model-free RL algorithms can search for control actions according to the gradient to increase the value of rewards (or decrease the costs). Considering that RL is efficient to deal with process control problems, we propose RL-based controllers and discuss the case with domain knowledge in this section. First, the general case of the process model is investigated, and the corresponding algorithm is proposed. Second, theorems are developed assuming the widely accepted linear process models. Finally, the characteristic of the RL-based controller is discussed.

When the domain knowledge is available and characterized by an unknown parameter $\boldsymbol{\Theta}$, i.e., $g(\cdot)$ can be well approximated by $f(u_t, d_t|\boldsymbol{\Theta})$. Different from traditional R2R control schemes that specify the process model and corresponding parameters by Design of Experiment (DOE) or Response Surface Methodology (RSM), a RL-based controller learns the system dynamics by updating parameter $\boldsymbol{\Theta}$ using data of historical runs in previous production cycles, which is denoted by dataset $\boldsymbol{D}$. Notably, we suppose that data in previous production cycles are collected from the same manufacturing process as the online cycle and follow the same process model. As shown in Figure 2, we have k production cycles whose run lengths are T_k to estimate the parameter $\boldsymbol{\Theta}$. The corresponding estimator is used for online control optimization in the $(k+1)$ th production cycle. For notation, we use superscript indices when referring to the index of production cycles and subscript indices when referring to the number of runs. Formally, the parameter estimator of the online control is denoted as $\hat{\boldsymbol{\Theta}}^{k+1}$. To minimize the sum of the squared loss of online system output and its target level in (1), we tend to minimize the difference between y_t and y^* at each run t. Therefore, the

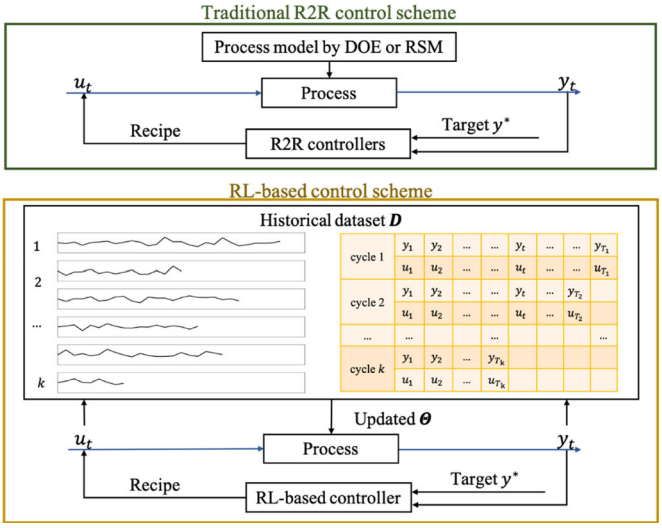

Figure 2. Methodology of traditional R2R and RL-based controllers.

online control action u_t^{k+1} is determined by solving the optimization problem

$$\min_{u_t^{k+1}} \left(f\left(u_t^{k+1}, \hat{d}_t^{k+1} | \hat{\boldsymbol{\Theta}}^{k+1}\right) - y^* \right)^2,$$

where $\hat{d}_t^{k+1}$ denotes the disturbance estimator that is usually predicted by its trajectory $(d_{t-1}^{k+1}, \ldots, d_1^{k+1})$ and the number of runs (t). As the disturbance cannot be observed directly, we represent it by

$$d_t^{k+1} = \left\{ d_t | y_t^{k+1} = f\left(u_t^{k+1}, d_t; \hat{\boldsymbol{\Theta}}^{k+1}\right) \right\},$$

where u_t^{k+1} and y_t^{k+1} are the values of control action and system outputs in online control at run t, respectively. Finally, u_t^{k+1}, y_t^{k+1} and d_t^{k+1} are collected for parameter re-estimation. This procedure of parameter estimation and online control optimization is summarized in Algorithm 1.

Since there are many choices for the form of $f(\cdot)$, it is infeasible to analyze all the possible models. Here, based on $f(\cdot|\boldsymbol{\Theta})$ and specified disturbance processes in the assumptions, we especially analyze the most widely accepted generalized linear model assumptions on parameters (Theorem 1), and control actions (Theorem 2) in the R2R control scheme applied in semiconductor manufacturing. The details are presented as follows. All the proofs are listed in the Appendix.

Assumption 2.1. Suppose that $g(\cdot)$ is well-approximated by $f(u_t, d_t|\boldsymbol{\Theta}) = \boldsymbol{\delta}^T \cdot \boldsymbol{\psi}(u_t) + d_t(\boldsymbol{\gamma})$, where $\boldsymbol{\Theta} = (\boldsymbol{\delta}; \boldsymbol{\gamma})$, and d_t can be formulated as $d_t(\boldsymbol{\gamma}) = \boldsymbol{\gamma}^T \cdot \boldsymbol{\eta}(t) + e_t$. $\boldsymbol{\psi}(\cdot)$ and $\boldsymbol{\eta}(\cdot)$ are known multi-dimensional functions on u_t and t, respectively. e_t is independent normally distributed errors with zero mean.

Algorithm 1. RL-based controller with domain knowledge

Input: function $f(\cdot)$, parameter T, y^*
Initialize: $\boldsymbol{D}$, k, T_k

$\hat{\boldsymbol{\Theta}}^{k+1} = \arg\min_{\boldsymbol{\Theta}} \left(\sum_{i=1}^{k} \sum_{t=1}^{T_k} \| f(u_t^i, d_t^i; \boldsymbol{\Theta}) - y_t^i \|^2 \right)$ based on dataset $\boldsymbol{D}$
for $t = 1 : T$ **do**

$$u_t^{k+1} = \arg\min_{u_t} \left(f\left(u_t, \hat{d}_t^{k+1}(d_{t-1}^{k+1}, \ldots, d_1^{k+1}, t); \hat{\boldsymbol{\Theta}}^{k+1}\right) - y^* \right)^2$$

execute u_t^{k+1} and record the outputs y_t^{k+1}
calculate d_t^{k+1} by $d_t^{k+1} = \left\{ d_t | y_t^{k+1} = f\left(u_t^{k+1}, d_t; \hat{\boldsymbol{\Theta}}^{k+1}\right) \right\}$
collect u_t^{k+1}, y_t^{k+1} and d_t^{k+1} in the online control process and update $\boldsymbol{D}$
end for
$k \leftarrow k + 1$

Theorem 1. Based on Assumption 2.1, let $\hat{\boldsymbol{\Theta}}$ denote the parameter estimators in $f(\cdot|\boldsymbol{\Theta})$, we have: $E(\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}) = 0$ and $var(\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}) \sim O(1/N)$, where N is the total number of historical observations (i.e., $N = \sum_{i=1}^{k} T_i$, and T_i is the run number of ith production cycle).

The proof is provided in Appendix A.

Theorem 1 guarantees the convergence order of the RL-based controller based on a generalized linear process model. Offline data in different runs are considered to estimate parameters in $f(\cdot|\boldsymbol{\Theta})$, and trigger control decisions according to $f(\cdot|\hat{\boldsymbol{\Theta}})$. According to Theorem 1, we have the variance of the difference between parameter estimators $\hat{\boldsymbol{\Theta}}$ and their ground truth $\boldsymbol{\Theta}$ is derived to be the order of $O(1/N)$. Therefore, if $g(\cdot)$ can be well-approximated by $f(\cdot|\boldsymbol{\Theta})$, with the converged $\hat{\boldsymbol{\Theta}}$, control action optimization based on $f(\cdot|\hat{\boldsymbol{\Theta}})$ will also converge to that based on $f(\cdot|\boldsymbol{\Theta})$, which is the real optimal control action. Specifically, if $\boldsymbol{\psi}(u_t)$ is also a linear function with current control action u_t, we have the following theorem.

Assumption 2.2. The approximate process model can be specified as a linear function with current control actions (i.e., $\boldsymbol{\psi}(\cdot)$ is linear on u_t), which is formulated as

$$f(u_t, d_t|\boldsymbol{\Theta}) = \boldsymbol{\delta}^T \cdot \boldsymbol{\psi}(u_t) + d_t(\boldsymbol{\gamma}) = bu_t + \boldsymbol{\gamma}^T \cdot \boldsymbol{\eta}(t) + e_t.$$

Let $\boldsymbol{\Theta}^{\mathrm{T}} = [b, \boldsymbol{\gamma}^{\mathrm{T}}]$ denote the parameter set.

By reformulating the approximate process model in Assumption 2.2, we have the expression of $f(\cdot|\boldsymbol{\Theta})$ as follows:

$$\begin{aligned} y_t = f(\cdot|\boldsymbol{\Theta}) &= bu_t + \boldsymbol{\gamma}^T \cdot \boldsymbol{\eta}(t) + e_t \\ &= bu_t + c_t + e_t \end{aligned} \tag{3}$$

where $c_t = \boldsymbol{\gamma}^T \cdot \boldsymbol{\eta}(t)$. The parameter estimators of c_t and b are denoted by $\hat{c}_t = \hat{\boldsymbol{\gamma}}^T \cdot \boldsymbol{\eta}(t)$ and $\hat{b}$, respectively. Then, we will have the following Theorem 2.

Theorem 2. In each run t, let μ_1 and μ_2, σ_1 and σ_2 denote the mean values and standard deviations of $(y^* - \hat{c}_t)$ and $\hat{b}$ respectively, and σ_{12} denotes the covariance of $(y^* - \hat{c}_t)$ and $\hat{b}$. We have the upper bound on the probability of the control errors of the weighted sample mean $\overline{u}_t = [\boldsymbol{\eta}(t)]^T (\boldsymbol{\eta}_N^T \boldsymbol{\eta}_N)^{-1} \boldsymbol{\eta}_N^T \boldsymbol{U}_N$ and its corresponding output y_t as

follows:

$$\begin{cases} P\left(\left|\overline{u}_t - u_t^*\right| > \xi\right) \le \dfrac{\sigma_2^2\sigma_1^2 - \sigma_{12}^2}{\left(\mu_2\sigma_2\xi\right)^2} + 2\Phi\left(-\dfrac{\mu_2}{\sigma_2}\right) \\ P\left(\left|E\left(y_t(\overline{u}_t)\right) - y_t^*\right| > \xi\right) \le \dfrac{\sigma_2^2\sigma_1^2 - \sigma_{12}^2}{\left(\sigma_2\xi\right)^2} + 2\Phi\left(-\dfrac{\mu_2}{\sigma_2}\right), \end{cases} \tag{4}$$

where ξ is an arbitrary positive number, $\Phi(\cdot)$ is the cumulative distribution function of standard normal distribution; $\boldsymbol{\eta}(t)$ is a known function of t; $\boldsymbol{U}_N$ is historical control actions and $\boldsymbol{\eta}_N$ is the observation of the function $\boldsymbol{\eta}(\cdot)$ in historical runs, where N is the total number of observations.

The proof is provided in Appendix B.

With the increased number of offline data, the variance of parameters will be reduced, and the ratio between variance and mean will be reduced. Therefore, the upper bound of the difference between the estimated control action (output) and the optimal control action (output) will also be reduced. Theorem 2 proposes theoretical error bounds of control actions and corresponding system outputs for the widely used linear process model in semiconductor manufacturing.

Notably, traditional statistical methods tend to estimate the parameters according to offline data at first, and then optimize the control action, i.e., the two-step procedure. Compared with the two-step procedure of traditional methods, RL-based control has a unique characteristic: "learning-by-doing". In each iteration, according to the estimated parameters in the approximate function $f(\cdot|\boldsymbol{\Theta})$, the control decision is optimized. Then, the current decision and the output observation are used to re-estimate the parameters. Therefore, the current optimal control action contributes to reinforcing the knowledge of decision-making.

Following Assumptions 2.1 and 2.2, the output of the RL-based controller in Algorithm 1 has less variation than the method of the two-step procedure. For example, if $y_t = a + bu_t + e_t$, even in a single production cycle, we have the variance of output prediction $\hat{y}_t$ given a certain control action u_t according to Altman and Gardner (1988):

$$\operatorname{var}\left(\hat{y}_t\right) = \left(\frac{1}{t-1} + \frac{(u_t - \overline{u})^2}{\sum_{i=1}^{t-1}(u_i - \overline{u})^2}\right)\sigma^2, \tag{5}$$

where $\overline{u}$ is the mean of control actions until the run $t-1$, and σ^2 is the variance of the random error e_t. With the increase of the run length from runs 1 to $t-1$, parameter estimators will converge to the real values, and the control action u_t also converges to the real optimal control action u^*, as it is optimized according to parameter estimators. Then the mean of historical control actions $\overline{u}$ also converges to u^*. Therefore, compared with random sampling in the traditional optimization after parameter estimation method, "learning-by-doing" in the RL-based controller can reduce the variance of $\hat{y}_t$ by reducing the difference of u_t and $\overline{u}$ in (5). This property makes it easy for RL-based controllers to find the optimal control action based on $\hat{y}_t$, especially when samples are limited.

In summary, Theorem 1 guarantees the convergence rate of parameters in the generalized linear process models for RL-based controllers. Theorem 2 proposes the upper bound on the probability of the control errors. In addition, an important advantage of RL-based controllers is discussed. Furthermore, if the process model is nonlinear, Recht (2019) discussed that as long as $f(\cdot)$ is consistent with the real function $g(\cdot)$, the RL-based controller with domain knowledge has good performance. However if the wrong form of $f(\cdot)$ is chosen, then the inconsistent form will lead to the bias of parameter estimation and then increases the variation of system output. Therefore, if a reasonable or accurate approximate process model is unavailable, we propose the RL-based controller without domain knowledge, as shown in Section 2.3.

2.3. RL-based controller without domain knowledge

To deal with the cases where domain knowledge is not available, we relax the assumption to approximate a process model. Instead, the RL-based controller with Policy Gradient Search (PGS) that estimates the distribution of input–output relationship from historical output data of a system is introduced in this subsection.

One typical method to solve the process control problem without domain knowledge refers to PGS, which is based on the assumption that the system output follows a distribution with control actions. We call this type of method RL-based control with PGS in process control problems. Suppose that the system output y_t follows a distribution with probability $p(y_t; u_t, y_{t-1}, \boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is the set of parameters in the distribution that need to be estimated by the offline data in historical runs denoted by $\boldsymbol{D}$. Based on the total cost in (1), the RL-based controller with PGS aims to minimize the expectation of total cost:

$$\min_{u_t} E_{p(y_t;u_t,y_{t-1},\boldsymbol{\beta})}\left[\sum_{t=1}^{T}\left(y_t - y^*\right)^2\right]. \tag{6}$$

The gradient descent method is used to minimize the cost in each run. We re-define the cost at t as $J_t(u_t) = E_{p(y_t;u_t,y_{t-1},\boldsymbol{\beta})}C_t(y_t)$, where $C_t(y_t) = (y_t - y^*)^2$. To find the gradient of the cost over control actions $\left(\nabla J_t(u_t)\right)$, the log-likelihood is used to simplify the solution. Thus, the gradient is calculated as:

$$\nabla J_t(u_t) = E_{p(y_t;u_t,y_{t-1},\boldsymbol{\beta})}\left[C_t(y_t)\nabla_{u_t}\log\left(p(y_t; u_t, y_{t-1}, \boldsymbol{\beta})\right)\right]. \tag{7}$$

Therefore, we propose the following Algorithm 2 to find the gradient descent direction and obtain the optimal solution by iterating control actions.

The RL-based controller with PGS searches a gradient descent direction of the control cost. The optimal cost value depends on the distribution of output at each run t, which needs to be estimated from historical offline data. In general, uniform and normal distributions are accepted to describe the system outputs (Recht, 2019). For example, if the system output follows the normal distribution, we can formulate the probability density function as:

$$p(y_t) = \frac{1}{\sqrt{2\pi}\beta_2} \exp\left(-\frac{\left(y_t - \left(y_{t-1} + \beta_1(u_t - u_{t-1})\right)\right)^2}{2\beta_2^2}\right), \tag{8}$$

where β_1 and β_2 are parameters that need to be estimated from the offline data. After parameter estimation, the gradient of control actions is calculated by (7), and the control actions are iterated for $n_{\max}$ times by PGS according to the gradient descent direction with step size α. Similar to Algorithm 1, we use u_t^{k+1} and y_t^{k+1} to denote the value of control action after $n_{\max}$ iterations by PGS and the corresponding system output in the online production (i.e., $(k+1)$th) cycle, respectively. Moreover, we use $u_t^{k+1(n)}$ to denote the value of u_t^{k+1} after the n th iteration of PGS. Algorithm 2 summarizes the details as follows.

In real applications, this policy gradient algorithm can deal with different kinds of disturbance processes, even though the disturbance variation is large. To verify the performance of the RL-based controller, we compare it with traditional controllers based on different simulation studies in Section 3. Moreover, nonlinear cases are elaborated in Section 4 to further show the advantages of RL-based controllers.

Algorithm 2. Algorithm of the RL-based controller with PGS in process control

Input: distribution $p(\cdot)$, parameters T, y^*, α, $\boldsymbol{\beta}$, $n_{\max}$
Initialize: y_0, k, n, and offline data $\boldsymbol{D}$, $u_1^{k+1(0)}$
Estimate parameters $\boldsymbol{\beta}$ in the distribution function $p(\cdot)$ according to data in $\boldsymbol{D}$ by

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \left[\sum_{i=1}^{k}\sum_{t=1}^{T} \log\left(p(\boldsymbol{\beta}; y_t^i, u_t^i, y_{t-1}^i)\right)\right]$$

for $t = 1 : T$ **do**
  $n = 0$
  repeat:
    $n \leftarrow n + 1$
    $g_t\left(u_t^{k+1}\right) = C_t(y_t)\nabla_{u_t}\log\left(p(y_t^{k+1}; u_t^{k+1}, y_{t-1}^{k+1}, \hat{\boldsymbol{\beta}})\right)$
    $u_t^{k+1(n)} = u_t^{k+1(n-1)} - \alpha g_t\left(u_t^{k+1(n-1)}\right)$
  Until $n \geq n_{\max}$
  $u_t^{k+1} \leftarrow u_t^{k+1(n)}$, and record corresponding output y_t^{k+1}.
  $u_{t+1}^{k+1(0)} \leftarrow u_t^{k+1}$ if $t \neq T$
end for
Add the data of complete online cycle to $\boldsymbol{D}$
$k \leftarrow k + 1$

3. Classic linear simulation cases

To compare with the traditional controllers that are usually applied in linear process models, we first simulate two linear cases in the semiconductor manufacturing process. Section 3.1 mainly shows the performance of RL-based controllers with domain knowledge in the chemical mechanical planarization process. Section 3.2 illustrates the performance of RL-based controllers without domain knowledge in the deep reactive ion etching process.

3.1. Application in the chemical mechanical planarization process

Chemical mechanical Planarization (CMP) is a crucial process in the semiconductor manufacturing process. Virtual metrology systems are often applied in the CMP process to remove the non-planar parts of the films. To simulate the CMP process, we first accept the simulation model in Ning et al. (1996) and Chang *et al.* (2006), which is defined as:

$$\boldsymbol{y_t} = \boldsymbol{A} + \boldsymbol{B}\boldsymbol{u_t} + \gamma t + \boldsymbol{e_t}, \tag{9}$$

where $\boldsymbol{y_t} \in \mathbb{R}^{2\times 1}$ is the output vector that represents the removal rate and non-uniformity, respectively. $\boldsymbol{u_t} \in \mathbb{R}^{4\times 1}$ is the control vector, that denotes the platen speed, back pressure, polish head downforce, and profile, respectively. According to Ning *et al.* (1996), the parameter matrices in (9) are assigned as

$$\boldsymbol{A} = \begin{bmatrix} -138.21 \\ -627.32 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 5.018 & -0.665 & 16.34 & 0.845 \\ 13.67 & 19.95 & 27.52 & 5.25 \end{bmatrix},$$

and $\gamma = \begin{bmatrix} -17 \\ -1.5 \end{bmatrix}$.

The white noise $\boldsymbol{e_t}$ is normally distributed with zero mean and covariance matrix

$$\boldsymbol{\Lambda} = \begin{bmatrix} 665.64 & 0 \\ 0 & 5.29 \end{bmatrix}.$$

The total number of runs $T = 30$ (i.e., $t \in \{1, 2 \ldots 30\}$) represents the length of the CMP process. The objective is to keep y_t at each time t close to the target value of output $\boldsymbol{y}^* = [1700, 150]^T$. The objective function to measure the control performance is to minimize the cost calculated by the Mean of Square Errors (MSE) from the first run to the last run, which is defined as:

$$\sum_{t=1}^{T} \left(\boldsymbol{y_t} - \boldsymbol{y}^*\right)^T \left(\boldsymbol{y_t} - \boldsymbol{y}^*\right)/T. \tag{10}$$

To verify the performance of the RL-based controller, we make comparisons for RL-based controllers with traditional controllers. Moreover, to verify the advantages of the RL-based controller, it is also compared with the method of control optimization after parameter estimation.

3.1.1. Comparison with the traditional controller

Facing the linear drift disturbances in the CMP process, Chen and Guo (2001) proposed the double EWMA (dEWMA) controller and showed its great performance. Therefore, we take the dEWMA controller as a benchmark in this case, and its control scheme is as follows:

$$\begin{cases} \boldsymbol{a_t} = w_1\left(\boldsymbol{y_{t-1}} - \hat{\boldsymbol{B}}\boldsymbol{u_t}\right) + (1 - w_1)\boldsymbol{a_{t-1}} \\ \boldsymbol{p_t} = w_2\left(\boldsymbol{y_{t-1}} - \hat{\boldsymbol{B}}\boldsymbol{u_t} - \boldsymbol{a_{t-1}}\right) + (1 - w_2)\boldsymbol{p_{t-1}}, \end{cases} \tag{11}$$

where w_1 and w_2 are two tuning parameters of $\boldsymbol{a}_t$ and $\boldsymbol{p}_t$ respectively. The control recipe of dEWMA is $\boldsymbol{u}_t = \hat{\boldsymbol{B}}^{-1}(\boldsymbol{y}^* - \boldsymbol{a}_t - \boldsymbol{p}_t)$. According to Chen and Guo (2001), the tuning parameters w_1 and w_2 are numerically optimized as 0.4 and 0.2, respectively.

To make a fair comparison, we generate the same offline data for dEWMA and RL-based controllers. For the RL-based controller, since the domain knowledge is available for an approximate model in this case, parameters ($\boldsymbol{A}$, $\boldsymbol{B}$, and γ) in this model are first estimated by offline historical data, and control decisions are made according to the estimators of parameters $\hat{\boldsymbol{A}}$, $\hat{\boldsymbol{B}}$ and $\hat{\gamma}$. Then, the online data are also collected to re-estimate parameters. According to Algorithm 1, the parameter estimation and control optimization are executed iteratively and convergent to the optimal ones.

To show more learning details for the RL-based controller, we use Figures 3 (a) to (d) to illustrate the learning process of the process gain (matrix $\boldsymbol{B}$) and disturbances ($\boldsymbol{A} + \gamma t + \boldsymbol{e}_t$). In Figure 3(a), we take four elements in the parameter matrix $\boldsymbol{B}$ as examples to illustrate the value of parameter estimators (represented by colored lines) and their ground truth (represented by black lines). As the number of iterations increases, the estimators will converge to their ground truth. The estimation error ratios evaluated by $\|\boldsymbol{B} - \hat{\boldsymbol{B}}\|_F / \|\boldsymbol{B}\|_F$ are illustrated in Figure 3(b). As the number of iterations increases, the error ratio gradually decreases, and the parameter estimators converge to their ground truth. Figure 3(c) illustrates the MSE of disturbance prediction based on the RL-based and dEWMA controllers from runs 1 to T in the iterations of the 10th, 20th, 30th, 40th, and 50th production cycles, respectively. Figure 3(d) shows the explicit values of the real disturbance and its predicted values by dEWMA and RL-based controllers from runs 1 to T in the 5th, 20th, and 50th online iterations. We find that under the approximate model, the RL-based controller provides a more reliable disturbance prediction than the dEWMA controller, and thus has better performance.

To comprehensively show the performance of two controllers, we use Figure 4 to illustrate the comparison results through 500 replications of the two controllers. In each replication, 50 production cycles are investigated, with $T = 30$ runs in each cycle. As the variation of the first cycle is much larger than the others, we provide two figures to show the results. The upper figures in Figures 4(a) and 4(b) show the entire plot and the lower figures show the details from the second to the last manufacturing cycle. We find that although the dEWMA controller has a smaller error in the first cycle, the MSE from the second to the last cycle of the RL-based controller is much lower than the dEWMA controller. The explicit values are summarized in Table 1. In this simulation case, the RL-based controller has better performance, since domain knowledge is available for an approximate model, and all offline data are used to predict more accurate process gains and disturbances.

Notably, in real applications, the manufacturing process may change in different production cycles. According to Sachs *et al.* (1995), parameters in manufacturing systems are subjected to different drifts in process models. Therefore, we focus on the case where parameters in the process model vary according to a certain distribution and compare the performance of EWMA, dEWMA, and adaptive EWMA

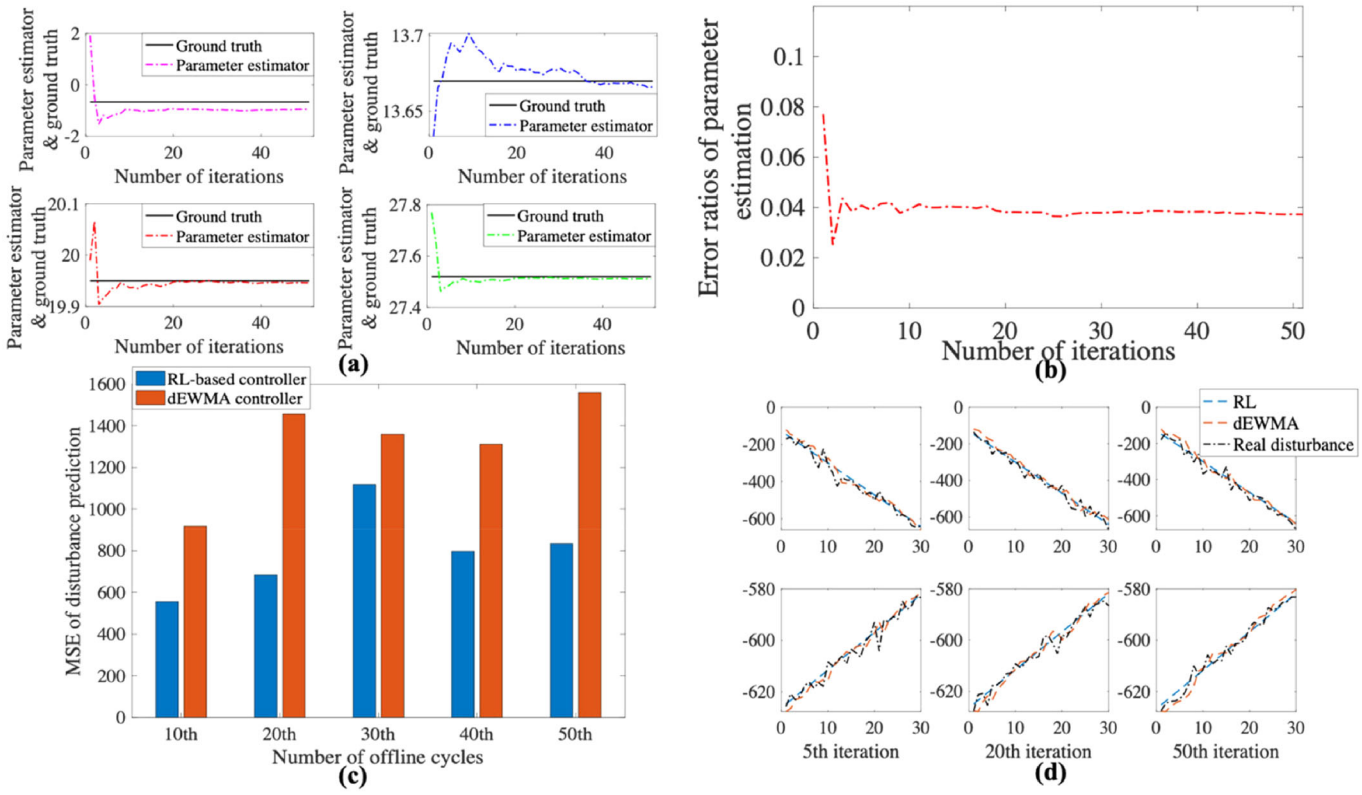

Figure 3. Learning process for RL-based controllers.

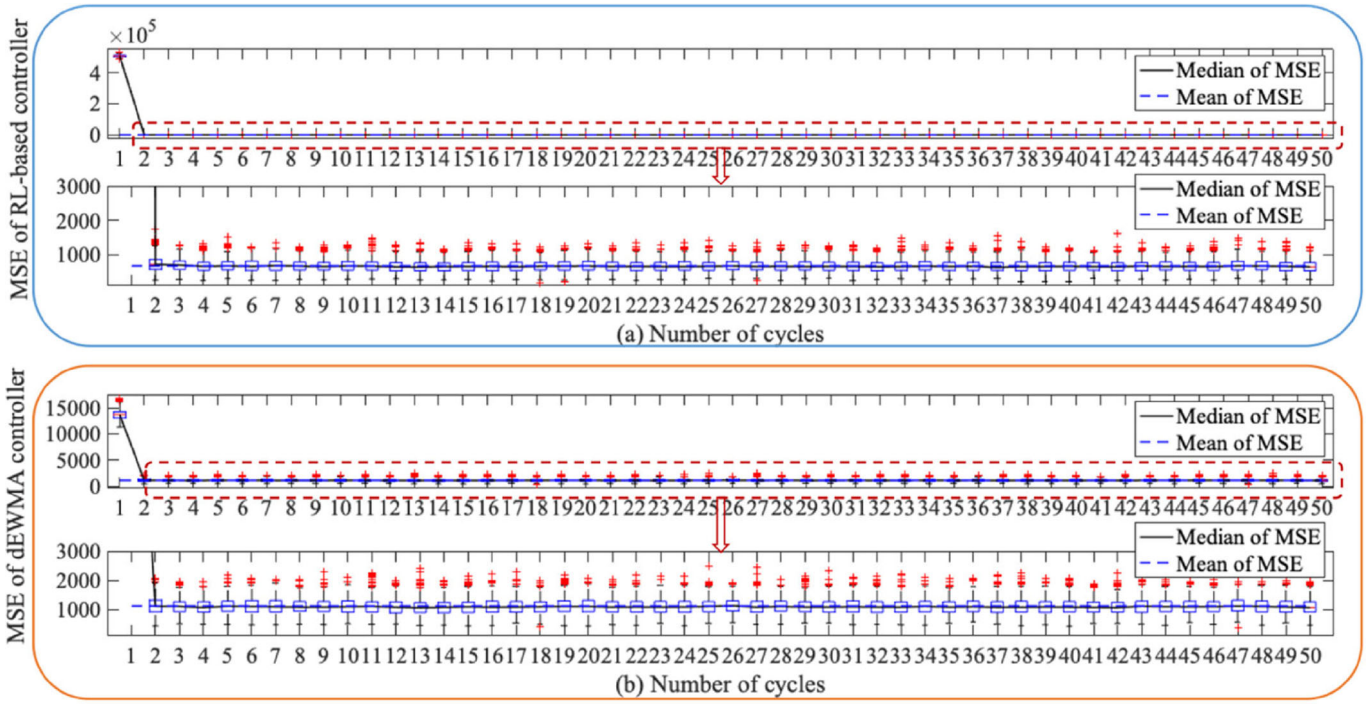

Figure 4. Boxplots of MSE under two controllers in 50 production cycles.

Table 1. MSE from the second to the last cycle.

MSE *Control method*	*Mean*	*Standard deviation*
RL-based controller	675.9774	24.3492
dEWMA controller	1.1194×10^3	37.8488

controllers with the RL-based controllers. We set the parameters in each run to i.i.d. normally distributed random variables, i.e., $\boldsymbol{A} \sim \boldsymbol{N}(\boldsymbol{E}(\boldsymbol{A}), (C_v\boldsymbol{I} \cdot \boldsymbol{E}(\boldsymbol{A}))^2)$ and $\boldsymbol{B} \sim \boldsymbol{N}(\boldsymbol{E}(\boldsymbol{B}), (C_v\boldsymbol{I} \cdot \boldsymbol{E}(\boldsymbol{B}))^2)$, where C_v is the variation coefficient for each parameter set to 0.01, and the mean of $\boldsymbol{A}$ and $\boldsymbol{B}$ are set to their values in the fixed-parameter case (Ning *et al.*, 1996). For convenience to analyze the random effects of parameters, we extended the total number of runs to $T = 500$. Then the EWMA, dEWMA, adaptive EWMA, and RL-based controllers are compared based on the same offline data and the common changing scheme of parameters in the process model.

Figure 5 compares the performance of these four controllers with changing parameters in the process model. Figure 5(a) shows the boxplots of the system outputs at early periods in the learning phase to illustrate the transient time before stability. If the difference between the output mean over 500 replications of two adjacent runs is less than the threshold value (set to 30 in this case), the control performance is considered as stable. Figure 5(b) illustrates the transient time before the stability of the four controllers. We find that the RL-based controller takes less transient time (two periods) to be stable than the other three controllers. Unlike traditional R2R controllers, which rely more on the last output to predict disturbances, the RL-based controller considers both offline and real-time online data in disturbance predictions. Based on the approximate process model provided by domain knowledge, the RL-based controller can still predict disturbances more accurately than traditional R2R controllers even though the parameters are random, resulting in a shorter transient time before stability. Figure 5(c) summarizes the boxplots for the MSE from runs 1 to T based on four controllers over 500 replications. As shown, the RL-based controller still performs well when the parameters in the process model follow stable distributions.

In summary, we conclude that the RL-based controller has more advantages when the parameters in the process model are fixed or varied according to stable distributions. All offline data are used to estimate the parameters in the approximate process model, and the performance of the RL-based controller is guaranteed. We also consider the effects of random shifts on parameters in the process model, which is discussed in Appendix D.

3.1.2. Comparison with the control optimization after parameter estimation method

As we analyzed in Section 2.2, different from traditional controllers that make control Optimization After Parameter Estimation (OAPE), the RL-based controller has a crucial characteristic "learning-by-doing". In this part, we make a comparison of these two control methods based on the CMP process. Specifically, the MSE of the N th cycle defined in (10) is used to evaluate the control performance. To describe the robustness of the performance for these two control methods, Table 2 presents the mean and standard deviation of MSE according to 50 replications.

As shown in Table 2 under the different number of manufacturing cycles, we find that the mean and standard deviation of the MSE in the RL-based controller are less than those in the OAPE method. Particularly, the RL-based controller has better performance when the number of cycles is

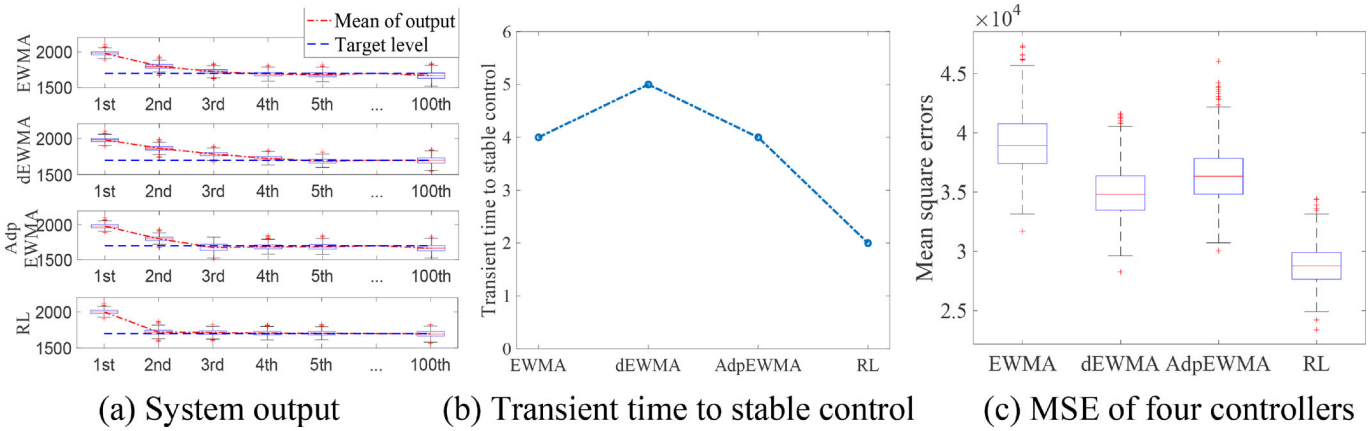

Figure 5. Comparisons of four controllers with changing parameters.
(a) System output (b) Transient time to stable control (c) MSE of four controllers

Table 2. Performance comparison of two control methods.

	Mean of MSE		Std. dev. of MSE	
Production cycle (N)	RL-based control	OAPE control	RL-based control	OAPE control
10	681.30	4228.74	173.51	6008.70
30	671.15	1670.92	143.82	1973.54
50	672.61	1019.64	182.87	552.91
100	673.02	904.67	163.08	463.24

limited. Furthermore, Figure 6 provides the boxplot of the MSE defined in (10) by using the RL-based controller and the OAPE method according to 50 replications, which shows the advantages of the RL-based controller intuitively.

3.2. Application in the deep reactive ion etching process

The Deep Reactive Ion Etching (DRIE) process is another important manufacturing process in semiconductor manufacturing. In existing research, the DRIE process is widely studied and used to compare the performance of different controllers. Due to the complex auto-correlation that exists in the disturbance process, the domain knowledge of proper correlation assumption is very limited, which hinders the use of the RL-based controller in Algorithm 1. Therefore, we propose the RL-based controllers without domain knowledge for process control in the DRIE process.

To conduct a reasonable comparison with traditional controllers that need a proper process model to describe the DRIE process, we refer to their process formulation to simulate data. Notably, the RL-based controller makes decisions solely according to the data. The widely accepted process model in the literature and also used in our simulation is as follows:

$$y_t = a + bu_t + d_t, \tag{12}$$

where d_t is the disturbance at t. By referring to the parameter setting in He *et al.* (2009), we have $a = 91.7$, and $b = -1.8$. Moreover, the total number of runs is $T = 80$, and the target value is $y^* = 90$. The most general disturbance process that is considered in the literature are IMA and ARIMA processes (He *et al.*, 2009; Wang and Han, 2013). Therefore, we focus on these two kinds of disturbances in the DRIE process.

3.2.1. IMA process

In the literature, the IMA disturbance process is one of the most widely accepted disturbances in process control problems, which is defined as $d_t = d_{t-1} + w_t - \theta w_{t-1}$, where w_t is the white noise. As it is proved that the EWMA controller with corresponding smoothing parameter is optimal for the linear process model with the IMA process (Ingolfsson and Sachs, 1993; Sachs *et al.*, 1995), we propose the EWMA controller with known parameters as the optimal benchmark to compare the performance of the RL-based controller.

For the RL-based controller, we first generate the offline data according to the process model in (12). Without the domain knowledge for an approximate process model, we try to approximate the output solely based on the distribution of the offline historical runs shown in Figure 7. In this case, the generalized Brownian motion is used to approximate. Then, the online distribution approximation of the system output y_t follows a normal distribution with mean value as: $\mu_{y_t|y_{t-1}} = y_{t-1} + \beta_1(u_t - u_{t-1})$, and variance $\text{var}_{y_t|y_{t-1}} = \nu(t)$. We assume that control actions can only influence the mean value of system outputs, and variance only depends on the operation time of the manufacturing system (constant variance can also be accepted based on real data). In our study, a time-dependent variance function $\text{var}_{y_t|y_{t-1}} = \beta_2^2 t$ is accepted. Moreover, parameters β_1 and β_2 are estimated from the offline data by the maximum likelihood estimation method. According to Algorithm 2, $\hat{\beta}_1$ and $\hat{\beta}_2$ are first estimated by 1000 historical production cycles, which are generated randomly, then the control actions are

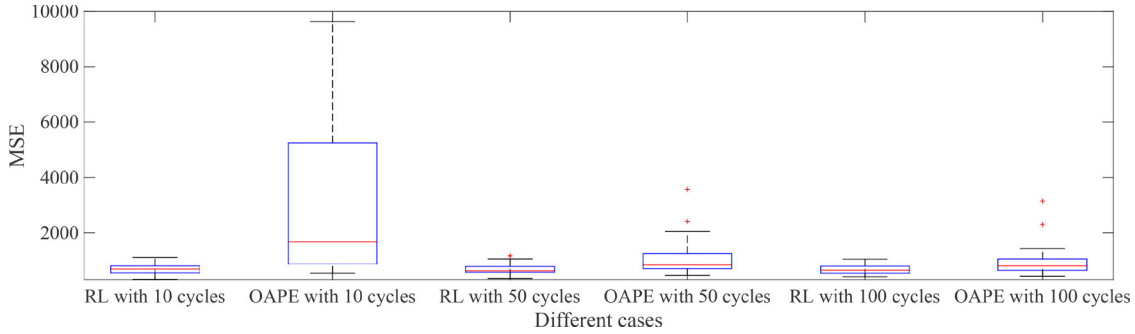

Figure 6. Comparison of RL-based controller and the OAPE method.

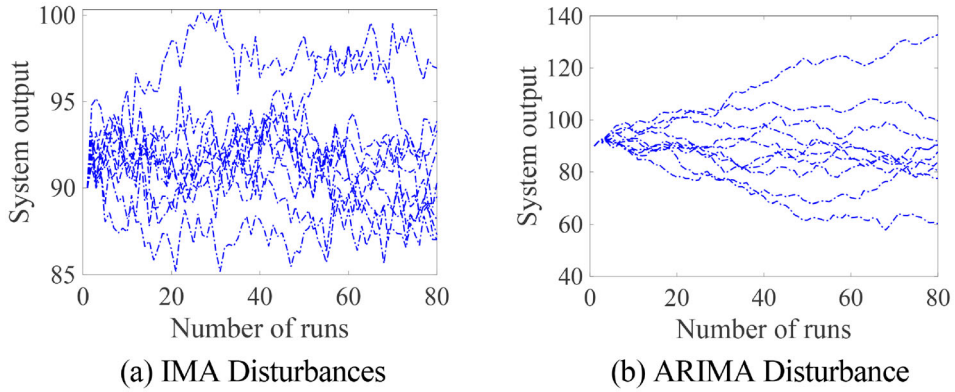

Figure 7. Offline data illustration on the system output distribution. (a) IMA Disturbances (b) ARIMA Disturbance

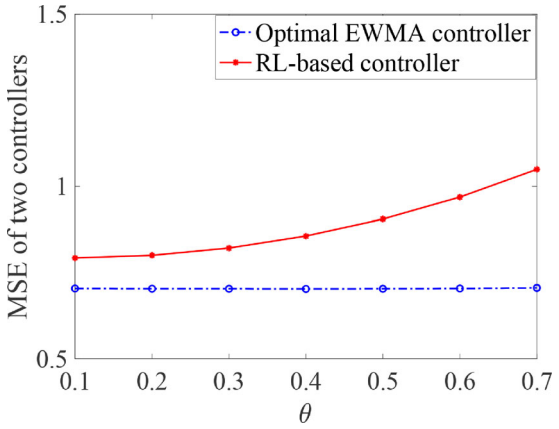

Figure 8. Performance comparison of RL-based and EWMA controllers

iterated according to policy gradient search. To compare the performance of RL-based controllers with the optimal EWMA controllers, the MSE of different runs is used to evaluate the performance, and summarized in Figure 8.

As shown in Figure 8 for different disturbance settings (i.e., parameter θ). It can be noticed that, especially when θ approaches zero, the performance of the RL-based controller gets close to the optimal EWMA controller. The reason for this behavior is that when θ approaches zero, the underlying truth of the system output distribution is close to the generalized Brownian motion we used in Algorithm 2. As θ gets larger, the gap between the estimated distribution and the underlying ground truth increases, and this decreases the performance of the RL-based controller. However, it does not mean that the RL-based controller cannot deal with the case when θ is large. In addition to the generalized Brownian motion, other distributions should also be considered according to practical applications and relevant data, which can be one of the potential research directions in the future.

3.2.2. ARIMA process

The ARIMA process is another general process in disturbance description with larger drifts than the IMA process. In this part, we focus on the ARIMA disturbance process to evaluate the performance of controllers, i.e., the formulation of d_t is as follows:

$$\begin{cases} d_t = d_{t-1} + \Delta d_t \\ \Delta d_t = \phi \Delta d_{t-1} + w_t - \theta w_{t-1}, \end{cases} \tag{13}$$

where $\phi, \theta \in (0, 1)$, and w_t is white noise.

Existing works have claimed that the traditional EWMA controller cannot deal with ARIMA disturbance very well and proposed modified controllers, such as the VEWMA controller (Tseng *et al.*, 2003) and GHR controller (He *et al.*, 2009). He *et al.* (2009) numerically showed that the GHR controller has a better control performance than the VEWMA controller and EWMA controller. Hence, we use the GHR controller with known process parameters as the quasi-optimal controller to compare with our RL-based controllers. Furthermore, we use the same parameters in (12) and (13) as He *et al.* (2009) to generate data, where $\theta = 0.5$, and $\phi = 0.6$.

For the RL-based controller, Figure 7(b) illustrates the system output distribution according to the offline data with

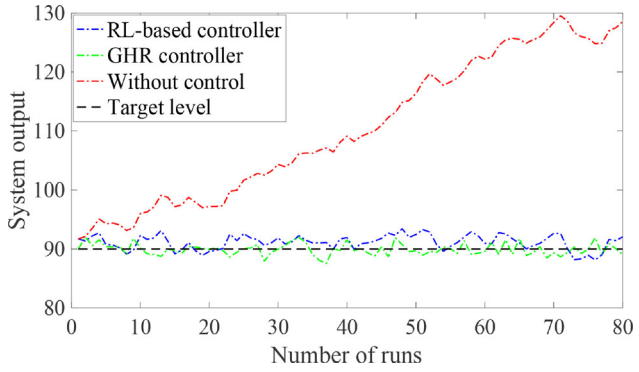

Figure 9. Results of three control methods based on ARIMA disturbance.

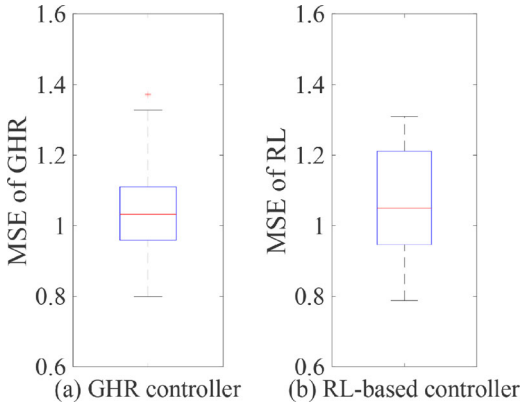

Figure 10. Distributions of MSE based on GHR and RL-based controllers.

ARIMA disturbances. The generalized Brownian motion is applied as the approximate distribution and the related parameters are estimated according to the maximum likelihood estimation method by 1000 offline production cycles of data. According to Algorithm 2, the control actions are iterated to reduce the control cost. As a result, we obtain the system output of the RL-based controller and compare it with the GHR controller, which is shown in Figure 9.

In Figure 9, the process model and parameters are exactly known for the GHR controller, whereas the RL-based controller with PGS relaxes the process model assumptions, which only estimates the output distribution from offline data. As a result, the performances of the GHR controller and RL-based controller with PGS are comparable in the ARIMA disturbance process. To show the comprehensive performance of both controllers, Figure 10 displays the boxplots of the MSE of two controllers in 30 replications.

As shown in Figure 10, the GHR and RL-based controllers have comparable performances. However, the RL-based controller only depends on historical offline data and relaxes the process model assumption, whereas the GHR controller relies on the explicit process model in (12) and (13). The superiority of the RL-based controller is obvious if the process model is unavailable.

To recognize the rationality of the RL-based controller, we make a theoretical analysis of the difference between the approximate distribution and the underlying ground truth model in this case. For the variance of these two stochastic processes, according to the linear process model with an

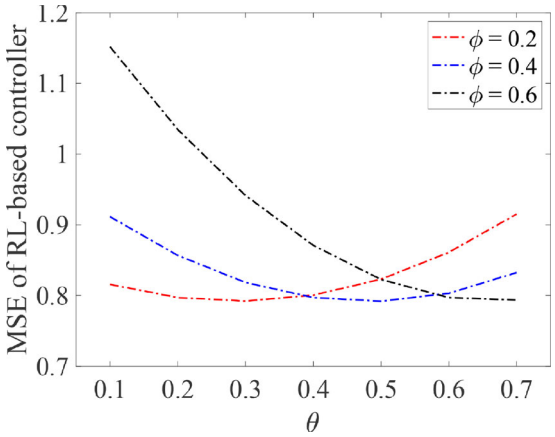

Figure 11. Sensitivity analysis of the RL-based controller.

ARIMA disturbance process, we have the variance of system output as follows:

$$\mathrm{var}_{y_t} = \left(\sum_{i=1}^{t-1} (t-i)\left(\phi^{i-1}(\phi-\theta)\right)^2 + t\right)\sigma^2, \tag{14}$$

where σ^2 is the variance of white noise w_t in (13). The variance difference $\Delta\mathrm{var}_{y_t}$ between the system output with ARIMA disturbances and traditionally Brownian motion process is

$$\Delta\mathrm{var}_{y_t} = (\phi-\theta)^2 S_t \sigma^2, \tag{15}$$

where $S_t = \sum_{i=1}^{t-1} (t-i)\left(\phi^{i-1}\right)^2$. Since S_t increases with t, and so does $\Delta\mathrm{var}_{y_t}$. Therefore, the time-dependent variance function $v(t) = \mathrm{var}_{y_t|y_{t-1}} = \beta_2^2 t$ is accepted and more accurate than the traditional Brownian motion process (i.e., variance function $v(t)$ is a constant function) to describe the manufacturing process with ARIMA disturbances.

To further verify the stability of the RL-based controller in the second case, we propose the sensitivity analysis in the DRIE process with the ARIMA disturbance. The parameters in the disturbances (i.e., θ and ϕ) are varied in each case to demonstrate the performance of the RL-based controller. As shown in Figure 11, since the ground truth of the process model is unknown, the performance of the RL-based controller has some variations with θ and ϕ. However, the variations are limited. In particular, as the values of θ and ϕ get close, the performance is better since the difference between the approximate distribution and underlying ground truth model is minimal as we analyzed in (15).

One of the highest potential advantages of RL-based controllers with PGS is they are data-driven controllers that can handle more complicated process models. For example, the RL-based controller with PGS can solve higher-order differential sequences. If d_t follows ARIMA (1, 2), a more complicated stochastic process $\tilde{d}_t \sim N\left(0, \sigma^2 \sum_{j=1}^{t} j\right)$ can be considered to approximate the second-order ARIMA disturbance. Moreover, in addition to traditional linear simulation cases, other complicated simulation studies are also considered to further verify the performance of RL-based controllers.

4. Other nonlinear simulation cases

In addition to the widely used linear process models, nonlinear models are proposed to describe the semiconductor manufacturing processes. In this section, we focus on the evaluation of RL-based controllers for different nonlinear models. In subsection 4.1, domain knowledge is available for a nonlinear approximate model.Whereas in subsection 4.2, we introduce cases with complicated nonlinear process models, which are hard to approximate by domain knowledge.

4.1. Application in a nonlinear CMP process

Based on the experiment tool presented by Khuri (1996), Del Castillo and Yeh (1998) simulated the CMP process by a nonlinear process model. In this case, three control variables are considered: back pressure downforce (u_1), platen speed (u_2), and the slurry concentration (u_3). The system outputs to reflect the manufacturing quality are removal rate (y_1) and within-wafer standard deviation (y_2). Similar to Section 3.1, the control decision aims to adjust the system output close to the target level, which is defined as $y_1^* = 2200$ and $y_2^* = 400$. We use the same model and parameters as Del Castillo and Yeh (1998), which are estimated from the results of a 32-wafer experimental design as follows:

$$\begin{aligned} y_1 &= 2756.5 + 547.6u_1 + 616.3u_2 - 126.7u_3 - 1109.5u_1^2 \\ &\quad -286.1u_2^2 + 989.1u_3^2 - 52.9u_1u_2 - 156.9u_1u_3 - 550.3u_2u_3 \\ &\quad -10t + \varepsilon_{1t} \\ y_2 &= 746.3 + 62.3u_1 + 128.6u_2 - 152.1u_3 - 289.7u_1^2 \\ &\quad -32.1u_2^2 + 237.7u_3^2 - 28.9u_1u_2 - 122.1u_1u_3 - 140.6u_2u_3 \\ &\quad +1.5t + \varepsilon_{2t}, \end{aligned} \tag{16}$$

where $\varepsilon_{1t} \sim N(0, 60^2)$ and $\varepsilon_{2t} \sim N(0, 30^2)$ are random errors.

If this nonlinear process model can be guided from domain knowledge, we use Algorithm 1 to estimate the parameters in the model and optimize the control action alternately. It takes time for an RL-based controller to learn the parameters in the approximate model before it reaches a stable performance. To fully conduct the learning process, we suppose that no offline data are available and only online data can be used to estimate the parameters in the approximate model. Therefore, the parameters are unknown and the control action in the first run is generated randomly. Then, control actions and system outputs are collected to estimate the parameters, and the control actions are optimized according to parameter estimators. More details of the learning process for parameter estimation are shown in Appendix C.

The performance of the RL-based controller is illustrated by the system output in Figure 12. As shown, in early periods, due to the limited number of historical online data, the RL-based controller exhibits large variability. As the number of historical data increases, the estimated parameters converge to their ground truth, and the control action also converges to the optimal one. As a result, the RL-based controller performs well to keep the system outputs close to their target levels.

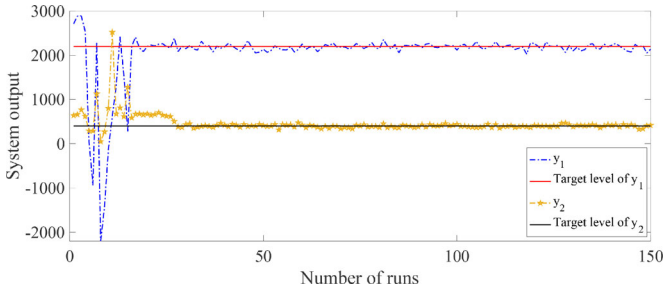

Figure 12. System output of RL-based controller for quadratic process model.

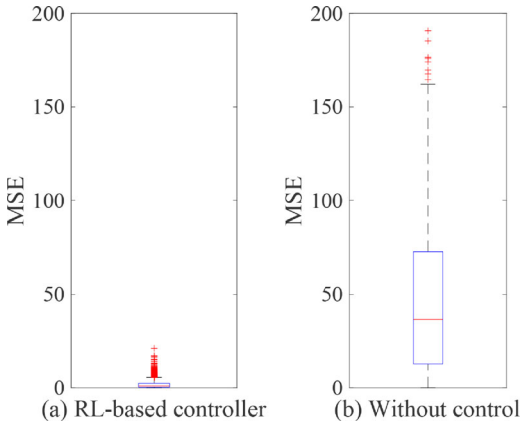

Figure 13. Distributions of MSE based on RL-based controller and the case without control.

To evaluate the performance of the RL-based controller, we compare it with the case without control. Figure 13 presents the MSE by boxplot according to 50 production cycles. It is obvious that the RL-based controller significantly decreases the MSE. In summary, as long as the domain knowledge can guide a reliable approximate model, which is not restricted to linear or polynomial models, the RL-based controller in Algorithm 1 can handle the control problem by estimating parameters and optimizing control actions.

4.2. Process model approximated by stochastic processes

When the process model is unavailable and hard to be approximated by domain knowledge, finding a proper approximate model is challenging. The RL-based controller with PGS in Algorithm 2 can analyze the relationship between control actions and system output from historical offline data and find the optimal control actions for unknown complicated nonlinear models.

To validate the performance of RL-based controllers, we consider two different stochastic processes (i.e., Wiener process and Gamma process) in the process model. In the literature, these two stochastic processes are also applied to describe disturbances in process control problems (Shao and Hou, 2006; Raouf and Michalska, 2010). It is generally accepted that the manufacturing system has a random drift without control, thus, we make two assumptions as follows:

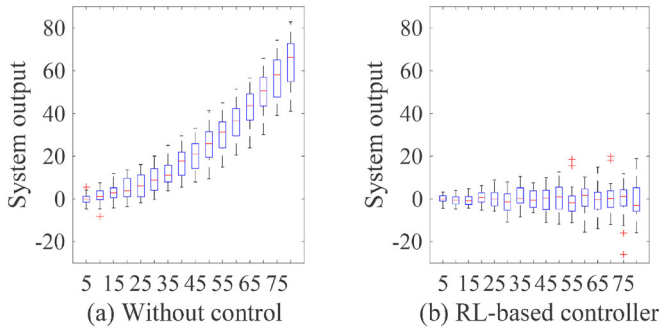

Figure 14. Errors of system output with the Wiener-process disturbance.

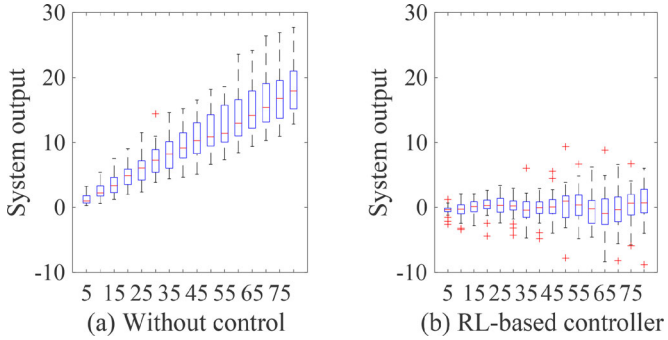

Figure 15. Errors of system output with the Gamma-process disturbance

1. The system outputs will drift following the Wiener/Gamma process when the manufacturing process is without control.
2. The control actions can adjust the mean values of system outputs directly but cannot influence the variabilities caused by disturbances.

Based on these assumptions, two stochastic processes are used to approximate the process model for data generation.

4.2.1. Wiener process

For validation purposes, we simulate the system output data by the Wiener process, which is a widely discussed stochastic process. According to Peng and Tseng (2009), we have the process model for data generation as follows:

$$y_t = y_0 + vt + \sigma B(t), \tag{17}$$

where $v = 0.02$ and $\sigma = 1$ are the drift and diffusion parameters, respectively, and $y^* = y_0 = 0$ without loss of generality.

Similar to Section 3.2, we use Algorithm 2 to optimize the control actions. Thirty replications are simulated, and Figure 14 illustrates the errors of the system output with and without control. It is obvious that the RL-based controller reduces the error and keeps the system output to the target level.

4.2.2. Gamma process

Another well-known stochastic process is the Gamma process. In this part, similar to the setting in Section 4.2.1, we also use the Gamma process to verify the performance of the RL-based controller with PGS. According to the formulation in Cheng *et al.* (2018), we have the process model for data simulation as $y_t = y_{t-1} + \Delta y_t$, where Δy_t follows a gamma distribution with probability density function:

$$f_{\Delta y}(y) = \frac{\beta^{\alpha} y^{\alpha-1}}{\Gamma(\alpha)} e^{-\beta y}, \tag{18}$$

where α is the shape parameter, β is the scale parameter, and $\Gamma(\cdot)$ is the Gamma function. We use the same parameters as in the numerical study from Cheng *et al.* (2018), i.e., $\alpha = 0.36$, and $\beta = 0.64$. Similar to the Wiener process, Algorithm 2 is also applied in the Gamma process to search for the control actions. The errors of the system output are illustrated in Figure 15 with 30 replications. We find that RL-based controllers are also efficient in terms of the gamma process.

Table 3. MSE in the two simulation cases.

Simulation cases	*Control cases*	*Mean of MSE*	*Std. dev. of MSE*
(1) Wiener process model	Without control	923.73	342.67
	RL-based controller	28.32	25.01
		(0.031)	(0.073)
(2) Gamma process model	Without control	125.80	61.93
	RL-based controller	4.85	5.46
		(0.039)	(0.088)

Table 3 summarizes the mean and standard deviation of MSE for these two stochastic simulation processes by 30 replications. To show the advantages of RL-based controllers intuitively, we calculate the mean and standard deviation of MSE, and also the ratios (shown in the bracket) of two control methods (i.e., without control and RL-based controller). We find that the RL-based controller can reduce more than 95% of MSE compared with the scenario without control. Meanwhile, the standard deviations of MSE are also reduced to 7–9% by the RL-based controller.

To further analyze the performance of RL-based controllers, we discuss the effects of different initialization conditions such as the number of offline production cycles and control iterations in Algorithm 2. Table 4 summarizes the results of control accuracy (measured by MSE) and efficiency (measured by computation time). As shown, the accuracy of the RL-based controller increases with the number of offline cycles and iterations of control actions, whereas the computation time also increases. Meanwhile, we find that the number of offline cycles has a more significant influence on the MSE, since it determines the accuracy of parameters in the approximate distribution. If the historical data are limited, increasing the number of iterations of control actions can also reduce the MSE. However, the increase in accuracy is accompanied by more computation costs, which is general for data-driven approaches. Therefore, in practice, if the historical data are sufficient, the number of iterations can be reduced for higher efficiency. Otherwise, we can increase the number of control action iterations for less control error.

4.3. Discussion

After the two simulation studies, we conclude that:

Table 4. Effects of initializations of RL-based controllers.

Simulation cases	# of historical cycles	# of control action iterations in Algo. 2	MSE	Computation time (seconds)
Wiener process model	100	1000	32.3312	251.80
		2000	27.2028	329.59
	1000	1000	29.4651	1182.04
		2000	25.1072	2037.49
Gamma process model	100	1000	14.9672	193.09
		2000	8.4722	239.33
	1000	1000	4.5166	1355.71
		2000	3.8336	1637.59

1. According to the general disturbance setting in the existing literature, RL-based controllers can solve the control problem without process model assumptions, and outperform or at least have comparable performance with traditional controllers.
2. To further verify the superiority of RL-based controllers, we use other process models in addition to the linear model to describe the input–output relationship such as quadratic models. In addition, we refer to two stochastic process models (Wiener process and Gamma process) to further validate the proposed RL-based controller with PGS. In practical applications, if the process is well-known with explicit model and disturbances series, traditional controllers with theoretical guarantees should be first applied. RL-based controllers are preferred for more complex processes, in which the explicit models or parameters are unknown.

If domain knowledge is available for approximation of unknown process models (not restricted to linear model), the RL-based controller in Algorithm 1 outperforms the traditional R2R controllers to deal with the system drift when the parameters in the process model are fixed or varied according to a stationary distribution. Even if the parameters in the approximate process model are unknown, the total cost of the RL-based controller decreases within limited samples. The RL-based controller has advantages as follows:

1. Different from the traditional R2R controller which only focuses on the estimation of process gain parameter, in the RL-based controller, all parameters in the approximate process model are estimated and will converge to their real values.
2. When domain knowledge is available for an approximate model, the RL-based controller predicts the disturbance according to it by fully considering the offline data and historical online data. In comparison, traditional R2R controllers focus more on the last system output.
3. In the RL-based controller, the parameters are estimated alternately with the control action optimization. In contrast, traditional R2R controllers make the control OAPE by offline data, and the parameter estimators are not updated. Therefore, the RL-based controller results in a lower MSE with more accurate parameter estimators.

If domain knowledge is not available for an approximate model, the RL-based controller in Algorithm 2 can still deal with various disturbances based on the output distribution approximation of the offline data. For example, based on the IMA and ARIMA processes, RL-based controllers even have comparable performance with the optimal EWMA and GHR controllers with known parameters. Moreover, more complicated cases such as nonlinear CMP simulation and other stochastic process models are also analyzed to verify the performance of the RL-based controller.

5. Conclusions

Process control is an important problem in the semiconductor manufacturing process to reduce process variation. Considering the presence of various disturbances in the manufacturing environments, our work aims to obtain the optimal control action based on historical offline data and real-time output of the system. Different from traditional control methods, which focus on building a linear process model to describe the input–output relationship, we propose RL-based controllers and relax the assumption of the process model. Compared with traditional controllers, RL-based controllers are suitable for more complicated applications, and not restricted to specific process models.

Based on the availability of domain knowledge to approximate a process model, we propose two different RL-based control algorithms and discuss some theoretical properties based on the widely accepted linear process models. Two sections of simulations prove that RL-based controllers are not only better or at least comparable with traditional controllers in linear simulation cases, but also show great potential to deal with more complicated cases. In the future, improvements of RL-based controllers can be further made, such as variations reduction of system outputs under model-free cases.

Funding

This work was supported by Guangzhou Municipal Science and Technology Program under grant No. 202201011235, Guangdong Basic and Applied Basic Research Foundation under grant No. 2023A1515011656, Foshan HKUST Projects under grant No. FSUST20-FYTRI03B, and the National Natural Science Foundation of China under Grants 71831006, 72001139 and 72122013.

Notes on contributors

Yanrong Li is a PhD candidate in management science and engineering at Antai College of Economics and Management, Shanghai Jiao Tong University. She received BE and ME degrees from Tianjin University in 2015 and 2018, respectively. Her research interests include data

analytics for process control and operational optimization in manufacturing systems.

Juan Du is currently an Assistant Professor with the Smart Manufacturing Thrust, Systems Hub, The Hong Kong University of Science and Technology (Guangzhou), China. She is also affiliated with the Department of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China, and Guangzhou HKUST Fok Ying Tung Research Institute, Guangzhou, China. Her current research interests include data analytics and machine learning for modeling, monitoring, control, diagnosis and optimization in smart manufacturing systems. Her research has received 7 best paper awards and two outstanding doctoral thesis awards.

Wei Jiang is a distinguished professor of management science at Antai College of Economics and Management, Shanghai Jiao Tong University. Prior to joining Shanghai Jiao Tong University, he worked in AT&T Labs, Stevens Institute of Technology, and Hong Kong University of Science and Technology. His research interests include big data analytics and innovation, Industry 4.0, and operations management, etc. He received the NSF CAREER award in 2006 and NSFC National Funds for Distinguished Young Scientists award in 2013.

ORCID

Juan Du http://orcid.org/0000-0002-6018-2972

References

Altman, D.G.and Gardner, M.J. (1988) Statistics in medicine: Calculating confidence intervals for regression and correlation. *British Medical Journal*, **296**(6631), 1238–1242.

Chang, Y.J., Kang, Y., Hsu, C.L., Chang, C.T. and Chan, T.Y. (2006) Virtual metrology technique for semiconductor manufacturing, in *International Joint Conference on Neural Networks*, IEEE Press, Piscataway, NJ, pp. 5289–5293.

Chen, A.and Guo, R.S. (2001) Age-based double EWMA controller and its application to CMP processes. *IEEE Transactions on Semiconductor Manufacturing*, **14**(1), 11–19.

Cheng, G.Q., Zhou, B.H. and Li, L. (2018) Integrated production, quality control and condition-based maintenance for imperfect production systems. *Reliability Engineering & System Safety*, **175**, 251–264.

Del Castillo, E. and Hurwitz, A.M. (1997) Run-to-run process control: Literature review and extensions. *Journal of Quality Technology*, **29**(2), 184–196.

Del Castillo, E. and Yeh, J.Y. (1998) An adaptive run-to-run optimizing controller for linear and nonlinear semiconductor processes. *IEEE Transactions on Semiconductor Manufacturing*, **11**(2), 285–295.

Djurdjanović, D., Jiao, Y. and Majstorović, V. (2017) Multistage manufacturing process control robust to inaccurate knowledge about process noise. *CIRP Annals*, **66**(1), 437–440.

Gu, S., Holly, E., Lillicrap, T., and Levine, S. (2017) Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates, in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE Press, Piscataway, NJ, pp. 3389–3396.

He, F., Wang, K.and Jiang, W. (2009) A general harmonic rule controller for run-to-run process control. *IEEE Transactions on Semiconductor Manufacturing*, **22**(2), 232–244.

He, W., Gao, H., Zhou, C., Yang, C. and Li, Z. (2020) Reinforcement learning control of a flexible two-link manipulator: An experimental investigation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **51**(12), 7326–7336.

Huang, D. and Lv, J. (2020) Run-to-run control of batch production process in manufacturing systems based on online measurement. *Computers & Industrial Engineering*, **141**(3), 106298.1–106298.15.

Ingolfsson, A. and Sachs, E. (1993) Stability and sensitivity of an EWMA controller. *Journal of Quality Technology*, **25**(4), 271–287.

Kaelbling, L.P., Littman, M.L. and Moore, A.W. (1996) Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, **4**, 237–285.

Khuri, A.I. (1996) Multiresponse surface methodology, in A. Ghosh and C.R. Rao (eds.), *Handbook of Statistics: Design and Analysis of Experiments*, Volume 13, Elsevier, Amsterdam, pp. 377–406.

Kober, J., Bagnell, J.A.and Peters, J. (2013) Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, **32**(11), 1238–1274.

Kutschinski, E., Uthmann, T. and Polani, D. (2003) Learning competitive pricing strategies by multi-agent reinforcement learning. *Journal of Economic Dynamics and Control*, **27**(11-12), 2207–2218.

Liu, K., Chen, Y., Zhang, T., Tian, S.and Zhang, X. (2018) A survey of run-to-run control for batch processes. *ISA Transactions*, **83**, 107–125.

Mataric, M.J. (1994) Reward functions for accelerated learning, in *Machine Learning Proceedings*, Morgan Kaufmann Publishers, pp. 181–189.

Naeem, M., Rizvi, S.T.H. and Coronato, A. (2020) A gentle introduction to reinforcement learning and its application in different fields. *IEEE Access*, **8**, 209320–209344.

Ning, Z., Moyne, J.R., Smith, T., Boning, D., Del Castillo, E., Yeh, J.Y. and Hurwitz, A. (1996) A comparative analysis of run-to-run control algorithms in the semiconductor manufacturing industry, in *IEEE/SEMI 1996 Advanced Semiconductor Manufacturing Conference and Workshop. Theme-Innovative Approaches to Growth in the Semiconductor Industry*., ASMC 96 Proceedings, IEEE, Cambridge, MA, pp. 375–381.

Pandey, V., Wang, E. and Boyles, S.D. (2020) Deep reinforcement learning algorithm for dynamic pricing of express lanes with multiple access locations. *Transportation Research Part C: Emerging Technologies*, **119**, 102715.

Peng C.Y. and Tseng, S.T. (2009) Mis-specification analysis of linear degradation models. *IEEE Transactions on Reliability*, **58**(3), 444455

Raouf, J.and Michalska, H. (2010, June) Stabilization of switched linear systems with Wiener process disturbances, in *Proceedings of the 2010 American Control Conference*, IEEE Press, Piscataway, NJ, pp. 3281–3286.

Recht, B. (2019) A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, **2**, 253–279.

Sachs, E., Hu, A. and Ingolfsson, A. (1995) Run by run process control-combining SPC and feedback-control. *IEEE Transactions on Semiconductor Manufacturing*, **8**(1), 26–43.

Shao, Y.E. and Hou, C.D. (2006) Estimation of the starting time of a step change disturbance in a gamma process. *Journal of the Chinese Institute of Industrial Engineers*, **23**(4), 319–327.

Su, A.J., Jeng, J.C., Huang, H.P., Yu, C.C., Hung, S.Y. and Chao, C.K. (2007) Control relevant issues in semiconductor manufacturing: Overview with some new results. *Control Engineering Practice*, **15**(10), 1268–1279.

Sutton, R.S. and Barto, A.G. (2018) *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA.

Tseng, S.T., Yeh, A.B., Tsung, F. and Chan, Y.Y. (2003) A study of variable EWMA controller. *IEEE Transactions on Semiconductor Manufacturing*, **16**(4), 633–643.

Tsung, F. and Shi, J. (1999) Integrated design of run-to-run PID controller and SPC monitoring for process disturbance rejection. *IIE Transactions*, **31**(6), 517–527.

Wang, A. and Shi, J. (2021) Holistic modeling and analysis of multistage manufacturing processes with sparse effective inputs and mixed profile outputs. *IISE Transactions*, **53**(5), 582–596.

Wang, K. and Han, K. (2013) A batch-based run-to-run process control scheme for semiconductor manufacturing. *IIE Transactions*, **45**(6), 658–669.

Wang, S., Bi, S. and Zhang, Y.A. (2019) Reinforcement learning for real-time pricing and scheduling control in EV charging stations. *IEEE Transactions on Industrial Informatics*, **17**(2), 849–859.