


MFRL-BI: Design of a Model-free Reinforcement Learning Process Control Scheme by Using Bayesian Inference

Yanrong Li, Juan Du, Wei Jiang & Fugee Tsung


To cite this article: Yanrong Li, Juan Du, Wei Jiang & Fugee Tsung (26 Jul 2024): MFRL-BI: Design of a Model-free Reinforcement Learning Process Control Scheme by Using Bayesian Inference, IIE Transactions, DOI: [10.1080/24725854.2024.2384965](https://doi.org/10.1080/24725854.2024.2384965)

To link to this article: <https://doi.org/10.1080/24725854.2024.2384965>

 View supplementary material 

 Accepted author version posted online: 26 Jul 2024.

 Submit your article to this journal 

 Article views: 32

 View related articles 

 View Crossmark data 

MFRL-BI: Design of a Model-free Reinforcement Learning Process Control Scheme by Using Bayesian Inference

Yanrong Li^a, Juan Du^{b,c,*}, Wei Jiang^a, and Fugee Tsung^{d,e}

^aAntai College of Economics and Management, Shanghai Jiao Tong University, Shanghai, China

^bSmart Manufacturing Thrust, Systems Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

^cDepartment of Mechanical and Aerospace Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China

^dDepartment of Industrial Engineering and Decision Analytics, The Hong Kong University of Science and Technology, Hong Kong SAR, China

^eInformation Hub, Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

*juandu@ust.hk

Abstract

Design of process control scheme is critical for quality assurance to reduce variations in manufacturing systems. Taking semiconductor manufacturing as an example, extensive literature focuses on control optimization based on certain process models (usually linear models), which are obtained by experiments before a manufacturing process starts. However, in real applications, pre-defined models may not be accurate, especially for a complex manufacturing system. To tackle model inaccuracy, we propose a model-free reinforcement learning (MFRL) approach to conduct experiments and optimize control simultaneously according to real-time data. Specifically, we design a novel MFRL control scheme by updating the distribution of disturbances using Bayesian inference to reduce their large variations during manufacturing processes. As a result, the proposed MFRL controller is demonstrated to perform well in a nonlinear chemical mechanical planarization (CMP) process when the process model is unknown. Theoretical properties are guaranteed when disturbances are additive. The numerical studies also demonstrate the efficiency of our methodology.

Keywords: model-free reinforcement learning; process control; Bayesian inference; design of experiments.

1. Introduction

1.1 *Background and motivations*

Process control is critical to keep the stability of manufacturing processes and guarantee the quality of final products, especially when a manufacturing process is complex. For example, in a semiconductor manufacturing process, two types of factors influence the stability of the manufacturing system. First, internal factors from manufacturing equipment and environments, mainly refer to process dynamics and disturbances during the manufacturing process (Tseng and Chen, 2017). Second, external factors refer to control recipes designed by the manufacturer, which aim to compensate for disturbances and adjust the system output to its desired target.

Traditional run-to-run (R2R) control schemes in semiconductor manufacturing processes can be divided into two phases. In Phase I, a process model is specified to describe the relationship between control input and process output through domain knowledge, design of experiments (DOE), or response surface methodology (RSM), followed by control recipe optimizations in Phase II (Tseng et al., 2019). A detailed literature review is provided in Section 1.2. However, in practical applications, when manufacturing processes are too complex to be described by specific models accurately, traditional R2R controllers may encounter significant challenges in accurate quality control. For example, the chemical mechanical planarization (CMP) process is one of the most important steps in semiconductor manufacturing to remove excess materials from the surface of silicon wafers. In literature, CMP processes are often controlled with explicit assumptions of process models (Castillo and Yeh, 1998). However, such models cannot fully capture the relationship between system outputs, control recipes, and disturbances, thereby leading to unavoidable model errors, which affect the accuracy of control optimization.

To tackle model inaccuracy in complex manufacturing processes, model-free reinforcement learning (MFRL) approaches (Recht, 2019) have been developed to learn manufacturing environments from real-time experimental data and directly search optimal control recipes without process model assumptions. Therefore, MFRL provides unprecedented opportunities for control optimization, especially in complex manufacturing processes. However, current MFRL approaches need to be improved as disturbances are hidden unstable factors that affect system outputs

significantly (Nian et al., 2020). Take CMP process as an example, Figure 1 illustrates the system outputs based on the MFRL controller in Recht (2019) (defined as a basic MFRL controller). In the basic MFRL controller, the effects of disturbances are ignored and control recipes are directly optimized based on system outputs. As shown in Figure 1, compared with the case without control, the basic MFRL controller can roughly keep the system output close to the target level. However, the controlled process still experiences significant deviations during some periods, which leads to invalid control. Therefore, it is highly desired to design a new control methodology to improve the basic MFRL controller by updating real-time distributions of disturbances to reduce the variations.

1.2 Literature review

In this subsection, we review different process control methods for complex manufacturing systems, especially for semiconductor manufacturing. Since the control mechanism or process model is important for controller design (Bastiaan, 1997), we classify the literature into two main categories based on whether the process model is available/predefined or not: (1) model-based controllers and (2) data-driven or model-free controllers.

Both linear and nonlinear process models have been considered in existing process control methodologies. Extensive pioneer works considered linear process models with disturbances that follow different stochastic time series. For example, Ingolfsson and Sachs (1993) analyzed the stability and sensitivity of the exponentially weighted moving average (EWMA) controller in compensating for the integrated moving average (IMA) disturbance process. Ning et al. (1996) formulated the process model as a linear transfer function with time-dependent drifts and developed a time-based EWMA controller. Tsung and Shi (1999) designed a proportional-integral-derivative (PID) controller for linear process models with autoregressive moving average (ARMA) disturbances and integrated the PID-based control scheme with statistical process control. Chen and Guo (2001) proposed an age-based double EWMA controller, which performs better than the EWMA controller in dealing with time-dependent drifts. Tseng et al. (2003) designed a new controller to improve the traditional EWMA controller by optimizing its discount factor and defined it as the variable-EWMA (VEWMA) controller, which has great performance in linear process models with ARIMA disturbance. Tseng et al. (2007) showed that the VEWMA controller has better performance than

double EWMA numerically. He et al. (2009) proposed a new controller named general harmonic rule (GHR) and theoretically proved its performance for a wide range of stochastic disturbances. Tseng et al. (2016) focused on the effects of previous process recipes and output responses on current outputs in semiconductor manufacturing processes, and proposed a multivariate EWMA controller for this linear dynamic process. Tseng and Chen (2017) proposed a generalized quasi-minimum mean square error (q-MMSE) controller to deal with a general-order dynamical model with added noises and guaranteed a long-term stability condition. Tseng et al. (2019) extended the q-MMSE controller to deal with more complicated disturbances such as high-order ARIMA processes. Ma and Pan (2024) optimized the tuning parameters in double-EWMA controllers using deep reinforcement learning, which performs well in linear process models with high-order ARIMA disturbances.

Besides linear process models, nonlinear process models are also widely studied. Hankinson et al. (1997) introduced a polynomial function to approximate a process model in deep reactive ion etching. Del Castillo and Yeh (1998) reviewed different polynomial process models for approximation of the CMP process and proposed adaptive R2R controllers according to these polynomial models. Kazemzadeh et al. (2008) extended the EWMA and VEWMA controllers in quadratic process models. In addition to polynomial models, more complicated nonlinear process models are introduced by differential equations. For example, Bibian and Jin (2000) considered a digital control problem in a second-order system and proposed two practical control schemes to deal with the time delay. Chen et al. (2012) focused on the deterministic as well as stochastic process models with measurement delay and proposed a new controller that integrates deterministic and stochastic components with applications in chemical vapor deposition (CVD) processes. Clerget et al. (2016) proposed a nonlinear sampled model-based controller for a nonlinear system considering model uncertainties and measurement delays. Moya et al. (2023) proposed an adaptive feedforward control scheme for a nonlinear electromechanical system. Zhou et al. (2024) focused on nonlinear batch-based systems with constraints and proposed an integration methodology of model predictive control and iterative learning control. In summary, model-based controllers depend crucially on explicit process formulations and are suitable for cases where the focused process models are well-validated.

When an explicit process model is not available, data-driven or model-free controllers are directly designed based on historical or offline data. For example, neural networks (NN) are widely used to approximate the unknown process model according to control recipes and system outputs. Park et al. (2005) approximated the real process model by an NN and designed an NN-based controller to reduce overlay misalignment errors significantly in semiconductor manufacturing processes. Wang and Chou (2005) proposed a neural-Taguchi-based control strategy to reach the desired material removal rate through an NN-simulated CMP process. Chang et al. (2006) developed a virtual metrology system using different NNs to describe the process model and optimized the control recipes accordingly. Kim et al. (2020) proposed a controller based on a least square generative adversarial network (GAN) and applied it to the CMP process. Tom et al. (2022) combined artificial NN methodology into an R2R control scheme and applied it in a spatial thermal atomic layer etching reactor. However, the NN-based controller also has limitations such as nonstationary control results and poor interpretations (Liu et al., 2018). Therefore, when controlling dynamic manufacturing systems characterized by unstable disturbances, existing NN-based approaches also encounter challenges in accurately approximating the manufacturing process.

Compared with NN-based control methods, reinforcement learning (RL) is another efficient data-driven control method to learn system dynamics and optimize control recipes by interacting with real-time system states. Given the definition of system state, control policy, and cost or reward function, RL can optimize control recipes based on real-time system states (Wang et al., 2018). For example, Recht (2019) introduced two basic policy-based algorithms for MFRL methods, policy gradient and pure random search (PRS). The policy gradient method optimizes control strategies based on the distribution of system outputs (Li et al., 2024), while the PRS method is more general and directly optimizes control strategies by stochastic gradient descent. However, as pointed out by Nian et al. (2020), these MFRL controllers cannot be directly applied in complex manufacturing systems due to large variations caused by unknown process models and unstable disturbances. Therefore, Khamaru et al. (2021) explored an effective variance reduction method based on an instance-dependent function in Q-learning.

In summary, the above data-driven methods share a common limitation: variations are relatively large. As process models are unknown, hidden unstable disturbances are hard to recognize, thereby bringing difficulties in optimizing control recipes compensating for them. To tackle the challenges, in this article, we design a new process control scheme to improve the basic MFRL controller (e.g., PRS-based MFRL controller) by updating the distribution of disturbances through Bayesian inference. We define it as a model-free reinforcement learning controller with Bayesian inference (MFRL-BI).

As disturbances can be reflected by system outputs, we use Bayesian inference to update the real-time distribution and integrate it into current MFRL control schemes. Figure 2 illustrates the difference between the control schemes of existing R2R and the proposed MFRL-BI controllers in terms of process assumptions and control optimization. Following the design steps of the process control scheme in Figure 2 (Del Castillo and Hurwitz, 1997), we divide the MFRL-BI controller into two phases: the optimization phase for controller learning (Phase I) and the implementation phase in real-time manufacturing (Phase II). In Phase I, we design experiments by virtual metrology (VM) to provide extensive data (Chang et al., 2006; Kang et al., 2009) for searching control recipes using MFRL algorithms. Considering the fact that disturbance can be inferred by system outputs, we update its distribution through Bayesian inference using real-time outputs. Finally, the input control recipes, system outputs, and disturbance inference data are collected and used for online real-time control in Phase II.

The main contributions of our work are summarized as follows: (1) a new model-free control scheme called MFRL-BI is proposed for efficient variation reduction by updating disturbance processes through Bayesian inference. (2) The corresponding algorithms of the MFRL-BI controller that combine Bayesian inference with the current PRS-based MFRL methodology are presented. (3) The proposed MFRL-BI controller is theoretically shown to guarantee optimality.

The remainder of this paper is organized as follows. Section 2 introduces the basic MFRL methodology in an R2R control scheme. Section 3 provides the design procedure of the MFRL-BI control scheme and interprets the related theoretical principles in Phases I and II. Section 4 demonstrates the performance of our method numerically and compares it with other benchmark

controllers with the application in a nonlinear CMP process control. Finally, Section 5 concludes the paper with remarks on future research directions.

2. Basic MFRL controller

In this section, we first present formulations of the process control problem in Section 2.1, and then discuss the methodology and corresponding algorithms of the basic MFRL in Section 2.2.

2.1 Process control formulation

We consider a multiple input-multiple output (MIMO) R2R process control problem that aims to reduce variations in a manufacturing system. At run $t \in \{1, 2, \dots, T\}$, a control recipe $\mathbf{u}_t \in \mathbb{R}^{m \times 1}$ is optimized to keep the system output $\mathbf{y}_t \in \mathbb{R}^{n \times 1}$ close to its target level $\mathbf{y}^* \in \mathbb{R}^{n \times 1}$, where T is the total number of runs. m and n are the dimensions of input control recipes and system outputs, respectively. The squared errors of process outputs are used to measure the control cost (Wang and Han, 2013). Furthermore, as control actions also bring extra costs in the manufacturing process, the cost function at run t is:

$$C_t(\mathbf{y}_t, \mathbf{u}_t) = (\mathbf{y}_t - \mathbf{y}^*)^T \mathbf{Q} (\mathbf{y}_t - \mathbf{y}^*) + \mathbf{u}_t^T \mathbf{R} \mathbf{u}_t, \quad (1)$$

where \mathbf{Q} and \mathbf{R} are positive definite weighted matrices. According to Del Castillo and Hurwitz (1997), the system output \mathbf{y}_t is affected by the control recipes \mathbf{u}_t as well as disturbances in manufacturing environments. Therefore, we define the underlying truth of the unknown process model as $\mathbf{y}_t = h(\mathbf{u}_t, \mathbf{d}_t)$, where $\mathbf{d}_t \in \mathbb{R}^{n \times 1}$ is the disturbance at run t . Combining with the cost function in Equation (1), we have the process control problem in T runs as:

$$\begin{aligned} \min_{\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_T\}} E_{\{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_T\}} [\sum_{t=1}^T ((\mathbf{y}_t - \mathbf{y}^*)^T \mathbf{Q} (\mathbf{y}_t - \mathbf{y}^*) + \mathbf{u}_t^T \mathbf{R} \mathbf{u}_t)] \\ \text{s. t. } \mathbf{y}_t = h(\mathbf{u}_t, \mathbf{d}_t). \end{aligned} \quad (2)$$

Note that the process model $h(\mathbf{u}_t, \mathbf{d}_t)$ is general and not specified.

In semiconductor manufacturing, it is widely recognized that process disturbances are used to describe the unmodeled process dynamics, which could source from manufacturing tools, products, or proceeding processes (Su et al., 2007) and be compensated by control recipes. As disturbances are hidden variables that cannot be directly observed, an additive model is widely used to quantify their effects on quality variables (Box and Kramer, 1992; Zhong et al, 2010; Wang and Han, 2013). Therefore, we have Assumption 2.1 for the process model.

Assumption 2.1: *The manufacturing process outputs can be separated into two additive parts related to control recipes and disturbances respectively, i.e.,*

$$\mathbf{y}_t = h(\mathbf{u}_t, \mathbf{d}_t) = g(\mathbf{u}_t) + \mathbf{d}_t. \quad (3)$$

where $g(\mathbf{u}_t)$ and \mathbf{d}_t are assumed to be independent.

In semiconductor manufacturing systems, disturbance processes exhibit general autocorrelations due to manufacturing environments such as aging effects (Del Castillo and Hurwitz, 1997). Therefore, in a manufacturing cycle from runs 1 to T , the disturbance \mathbf{d}_t can be inferred from its historical trajectory $\mathbf{D}_{t-1} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_{t-1}]$. We define the conditional probability density function of the disturbance at run t as $p(\mathbf{d}_t | \mathbf{D}_{t-1})$ with mean vector $\boldsymbol{\mu}_t$ and covariance matrix $\boldsymbol{\Sigma}_t$.

For control recipes to compensate for the disturbances, as shown in Equation (3), their effects on the system output are modeled by a function $g(\cdot)$, which is often assumed as a linear function in literature (Chen and Guo, 2001; Tseng et al., 2003; 2007). Considering the potential inaccuracy, we relax formulation assumptions of $g(\cdot)$ in our model. Although the effects of control recipes and disturbances on the system output are separated according to Assumption 2.1, there still exists a significant challenge in quantifying the effects of control recipes and disturbances as $g(\cdot)$ is unknown and \mathbf{d}_t cannot be observed directly.

2.2 Methodology of basic MFRL with PRS

In the control methodology of a basic MFRL controller, the expectation of control cost over disturbances \mathbf{d}_t is minimized by optimizing control recipe \mathbf{u}_t . Due to the unknown process model $g(\cdot)$, the cost function is also an unknown function over \mathbf{u}_t . According to Recht (2019), the objective function in Equation (2) can be reformulated as $J(\mathbf{u}) = \mathbf{E}_{\{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_T\}}[\sum_{t=1}^T C_t(\mathbf{y}_t(\mathbf{u}_t, \mathbf{d}_t), \mathbf{u}_t)]$, where $\mathbf{u} = [\mathbf{u}_1, \dots, \mathbf{u}_t, \dots, \mathbf{u}_T]$. Before optimizing the function $J(\mathbf{u})$, suppose the following assumption holds.

Assumption 2.2: *The function $J(\mathbf{u}) = \mathbf{E}_{\{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_T\}}[\sum_{t=1}^T C_t(\mathbf{y}_t(\mathbf{u}_t, \mathbf{d}_t), \mathbf{u}_t)]$ achieves a minimum at an unknown point \mathbf{u}^* .*

To minimize $J(\mathbf{u})$, the basic MFRL controller in Recht (2019) uses a PRS-based method to optimize the control recipes by stochastic gradient descent (SGD). If Assumptions 2.1 and 2.2 hold, the optimization problem in Equation (2) can be solved via the SGD algorithm as follows.

SGD Algorithm: *There are two steps in the SGD algorithm for the basic MFRL controller. First, the gradient of $J(\mathbf{u})$ is approximated by a finite difference along the direction $\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \in \mathbb{R}^{m \times T}$ is a random vector consisting of 0 or 1. Then, we can write the gradient of $J(\mathbf{u})$ as:*

$$\nabla_{\mathbf{u}} J(\mathbf{u}) = \frac{J(\mathbf{u} + \iota \boldsymbol{\epsilon}) - J(\mathbf{u} - \iota \boldsymbol{\epsilon})}{2\iota} \boldsymbol{\epsilon}, \quad (4)$$

where $\iota \rightarrow 0$ and $\mathbf{u} \mp \iota \boldsymbol{\epsilon}$ denote the neighborhood of the control strategy \mathbf{u} . Second, the control recipe moves along the gradient descent direction with step size α . If $\mathbf{u}^{[k]}$ is used to denote the value of control recipes in the k th iteration, we have

$$\mathbf{u}^{[k+1]} = \mathbf{u}^{[k]} - \alpha \nabla_{\mathbf{u}} J(\mathbf{u}^{[k]}). \quad (5)$$

These two steps are executed alternately until \mathbf{u} converges (i.e., the difference between successive iterated values of $\mathbf{u}^{[k+1]}$ and $\mathbf{u}^{[k]}$ is smaller than a pre-defined threshold η).

Following the SGD algorithm, Algorithm 1 presents the aforementioned control search procedure to minimize the unknown function $J(\cdot)$.

Algorithm 1. MFRL with PRS Algorithm

Function: MFRL_PRS(\cdot)

Input: hyper-parameters $\boldsymbol{\epsilon}$, ι , α , η

Initialize: $k = 0$, control recipe $\mathbf{u}^{[0]}$

Repeat:

Execute two initial control strategies $\mathbf{u}^{[k]} + \iota \boldsymbol{\epsilon}$ and $\mathbf{u}^{[k]} - \iota \boldsymbol{\epsilon}$

$$\nabla_{\mathbf{u}} J(\mathbf{u}^{[k]}) = \frac{J(\mathbf{u}^{[k]} + \iota \boldsymbol{\epsilon}) - J(\mathbf{u}^{[k]} - \iota \boldsymbol{\epsilon})}{2\iota} \boldsymbol{\epsilon}$$

$$\mathbf{u}^{[k+1]} = \mathbf{u}^{[k]} - \alpha \nabla_{\mathbf{u}} J(\mathbf{u}^{[k]})$$

$$k \leftarrow k + 1$$

Until $\|\mathbf{u}^{[k]} - \mathbf{u}^{[k-1]}\| < \eta$

$$\hat{\mathbf{u}} = \mathbf{u}^{[k]}$$

Output: $\hat{\mathbf{u}}$

According to the asymptotic analysis of SGD algorithm in Kiefer and Wolfowitz (1952), if disturbances satisfy the condition $\mathbf{E}(\mathbf{d}_t) = \mathbf{0}$, the control recipe searched in Algorithm 1 will converge to the optimal value. However, in practice, the disturbance process is not stable, its fluctuations and drifts are inevitable and may even increase as time goes by. For example, in CMP process in Figure 1, the basic MFRL controller encounters large variations, as it focuses on

minimizing the expected control cost $J(\mathbf{u})$ but ignores the variations and drifts of disturbance \mathbf{d}_t . To overcome this limitation, we propose the MFRL-BI controller to further reduce the variations of system outputs by dynamically updating the distribution of disturbances in Section 3.

3. The MFRL-BI controller

In this section, the MFRL-BI controller is proposed to improve the performance of basic MFRL by updating the distribution of disturbance via Bayesian inference. Following Figure 2, we introduce methodologies of the proposed MFRL-BI controller in two phases in Sections 3.1 and 3.2 respectively. As shown in Figure 3, in Phase I, control recipes are searched in the inner loop using the MFRL algorithm with PRS. After taking the convergent control recipe, the distribution of disturbance is updated in the outer loop. Meanwhile, the control recipes, system outputs, and estimated disturbances are collected, which are used for real-time control optimization in Phase II.

As introduced in Section 2.2, disturbances are unobservable, we define the prior distribution of \mathbf{d}_t condition on its trajectory as

$$\mathbf{d}_t | \mathbf{D}_{t-1} \sim p(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), \quad (6)$$

where $p(\cdot)$ is the probability distribution function. The observations of system output \mathbf{y}_t can reflect the disturbance process and be used to update the posterior distribution of \mathbf{d}_t . However, \mathbf{y}_t is also affected by the control recipe \mathbf{u}_t , which brings challenges for disturbance inference. Therefore, in Figure 3, we separate the effects of \mathbf{d}_t and \mathbf{u}_t , make inference of \mathbf{d}_t in the outer loop, and optimization of \mathbf{u}_t in the inner loop.

Specifically, to separate the effects of \mathbf{d}_t and \mathbf{u}_t , we reformulate the process model in Equation (3) as $\mathbf{y}_t = g(\mathbf{u}_t) + \mathbf{d}_t = g(\mathbf{u}_t) + \boldsymbol{\mu}_t + \boldsymbol{\delta}_t$, where $\boldsymbol{\mu}_t$ is the mean vector of \mathbf{d}_t and $\boldsymbol{\delta}_t = \mathbf{d}_t - \boldsymbol{\mu}_t$ is a random vector with $E(\boldsymbol{\delta}_t) = \mathbf{0}$. Since the process model $g(\mathbf{u}_t)$ is unknown, the variability of searched control recipe via Algorithm 1 using PRS is unavoidable, especially when the number of iterations is limited and the step size is fixed (Kiefer and Wolfowitz, 1952). We use $\mathbf{v}_t = \hat{\mathbf{u}}_t - \mathbf{u}_t^*$ to denote this variability, where $\hat{\mathbf{u}}_t$ is control recipe searched by PRS and \mathbf{u}_t^* is the underlying optimal control recipe. In summary, we reformulate the optimization problem in Equation (2) as follows at each run t :

$$\begin{aligned} & \min_{\mathbf{u}_t} \mathbf{E}_{\delta_t, \nu_t} [C_t(\mathbf{y}_t, \mathbf{u}_t)] \\ & \text{s.t. } \mathbf{y}_t = g(\mathbf{u}_t) + \boldsymbol{\mu}_t + \boldsymbol{\delta}_t. \end{aligned} \quad (7)$$

By incorporating the constraints into the objective function, we have:

$$\mathbf{E}_{\delta_t, \nu_t} [C_t(\mathbf{y}_t, \mathbf{u}_t)] = \text{tr}(\mathbf{Q}\boldsymbol{\Sigma}_t) + M(\mathbf{u}_t | \boldsymbol{\mu}_t), \quad (8)$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix, and

$$M(\mathbf{u}_t | \boldsymbol{\mu}_t) := \mathbf{E}_{\nu_t} [(g(\mathbf{u}_t) + \boldsymbol{\mu}_t - \mathbf{y}^*)^T \mathbf{Q} (g(\mathbf{u}_t) + \boldsymbol{\mu}_t - \mathbf{y}^*)] + \mathbf{u}_t^T \mathbf{R} \mathbf{u}_t. \quad (9)$$

Detailed derivations are presented in Appendix A.1. Then the total cost can be divided into two parts: $M(\mathbf{u}_t | \boldsymbol{\mu}_t)$ and $\text{tr}(\mathbf{Q}\boldsymbol{\Sigma}_t)$. As shown in Equation (9), the control optimization depends on the mean vector of the disturbances. However, due to the dynamics of disturbance, it is necessary to continuously infer its current distribution based on real-time outputs to guarantee the accuracy of control optimization. As a result, the separation in Equation (8) allows us to optimize $M(\mathbf{u}_t | \boldsymbol{\mu}_t)$ by MFRL algorithm with PRS and update the value of $\text{tr}(\mathbf{Q}\boldsymbol{\Sigma}_t)$ and $\boldsymbol{\mu}_t$ by Bayesian inference. The methodology and corresponding algorithms of control optimization and disturbance inference in Phase I will be elaborated in Section 3.1.

3.1 Control optimization in Phase I

To separate the effects of \mathbf{u}_t and \mathbf{d}_t , we divide the control process at each run into two steps: (i) at the beginning of run t , given the prior distribution of \mathbf{d}_t , control recipe \mathbf{u}_t is searched to minimize the control cost $M(\mathbf{u}_t | \boldsymbol{\mu}_t)$; (ii) the posterior distribution of \mathbf{d}_t is updated when the system output \mathbf{y}_t is observed and the prior distribution of \mathbf{d}_{t+1} is inferred according to the posterior distribution of \mathbf{d}_t . These two steps correspond to the inner and outer loops in Figure 3, respectively, and are presented as follows.

A. Inner loop: search for control recipes

In this part, we design an experiment searching for control recipes to minimize the expected control cost $M(\mathbf{u}_t | \boldsymbol{\mu}_t)$. According to its definition in Equation (9), we can separate $M(\mathbf{u}_t | \boldsymbol{\mu}_t)$ as:

$$M(\mathbf{u}_t | \boldsymbol{\mu}_t) := H(\mathbf{u}_t | \boldsymbol{\mu}_t) + \mathbf{u}_t^T \mathbf{R} \mathbf{u}_t, \quad (10)$$

where $H(\mathbf{u}_t | \boldsymbol{\mu}_t) = [(g(\mathbf{u}_t) + \boldsymbol{\mu}_t - \mathbf{y}^*)^T \mathbf{Q} (g(\mathbf{u}_t) + \boldsymbol{\mu}_t - \mathbf{y}^*)]$. As $\mathbf{u}_t^T \mathbf{R} \mathbf{u}_t$ is a deterministic convex function of \mathbf{u}_t , it is necessary to search the gradient of $H(\cdot)$, and we have $\nabla_{\mathbf{u}_t} M(\mathbf{u}_t | \boldsymbol{\mu}_t) = \nabla_{\mathbf{u}_t} H(\mathbf{u}_t | \boldsymbol{\mu}_t) + 2\mathbf{R}\mathbf{u}_t$. Before searching for \mathbf{u}_t , we suppose that $H(\cdot)$ also satisfies Assumption 2.2, i.e., $H(\cdot)$ is an unknown function that has a minimum at an unknown point $\tilde{\mathbf{u}}_t$

($\tilde{\mathbf{u}}_t = \arg \min_{\mathbf{u}_t} H(\mathbf{u}_t | \boldsymbol{\mu}_t)$). Then, similar to the basic MFRL controller, we implement Algorithm 1 to optimize the unknown function $M(\cdot)$ using PRS. Particularly, to further guarantee the stability of control recipes and reduce the variability of \mathbf{v}_t , after the convergence of \mathbf{u}_t based on Algorithm 1, we execute another N iterations of control recipes, which are denoted as $\hat{\mathbf{u}}_t(1)$ to $\hat{\mathbf{u}}_t(N)$. The final recipe is chosen as the mean of control recipes after convergence (i.e., $\bar{\mathbf{u}}_t = \frac{1}{N} \sum_{i=1}^N \hat{\mathbf{u}}_t(i)$). Algorithm 2 presents the details of the control optimization in the MFRL-BI controller.

Algorithm 2 has two procedures: first, control recipes are searched to minimize the cost function $M(\cdot)$ given the distribution of disturbances. Second, after the convergence of control recipes, we use another N samples to reduce the variations of control resulting from stochastic gradient approximation for the unknown function $H(\cdot)$. Before examining the properties of searched control recipes in Algorithm 2, we introduce two assumptions about function $H(\cdot)$ as in Mandt et al. (2017).

Algorithm 2. Control optimization given disturbance distribution

Function: Control_Search

Input: parameter $\boldsymbol{\mu}_t$, hyper-parameters $\epsilon \in \mathbb{R}^{m \times 1}$, α , N , ι

Output: $\bar{\mathbf{u}}_t$

Initialize: control recipe $\mathbf{u}_t^{[0]}$

Calculate $\hat{\mathbf{u}}_t(1)$ using Algorithm 1 based on function $M(\cdot | \boldsymbol{\mu}_t)$

For $i = 1$ to $N - 1$ **do**

Execute control strategies $\hat{\mathbf{u}}_t(i) + \iota\epsilon$ and $\hat{\mathbf{u}}_t(i) - \iota\epsilon$

$$\nabla_{\mathbf{u}_t} M(\hat{\mathbf{u}}_t(i) | \boldsymbol{\mu}_t) = \frac{H(\hat{\mathbf{u}}_t(i) + \iota\epsilon | \boldsymbol{\mu}_t) + H(\hat{\mathbf{u}}_t(i) - \iota\epsilon | \boldsymbol{\mu}_t)}{2\iota} \epsilon + 2\mathbf{R}\hat{\mathbf{u}}_t(i)$$

$$\hat{\mathbf{u}}_t(i + 1) = \hat{\mathbf{u}}_t(i) - \alpha \nabla_{\mathbf{u}_t} M(\hat{\mathbf{u}}_t(i) | \boldsymbol{\mu}_t)$$

End for

$$\bar{\mathbf{u}}_t = \frac{1}{N} \sum_{i=1}^N \hat{\mathbf{u}}_t(i)$$

Assumption 3.1: *The stochastic gradient in Algorithm 2 can be expressed as the underlying truth gradient value plus a random gradient noise. The noise can be approximated as Gaussian, whose variance is independent of control recipes. i.e., $\nabla_{\mathbf{u}_t} H(\mathbf{u}_t | \boldsymbol{\mu}_t) \approx \nabla_{\mathbf{u}_t} H^*(\mathbf{u}_t | \boldsymbol{\mu}_t) + \boldsymbol{\varepsilon}$ and $\nabla_{\mathbf{u}_t} M(\mathbf{u}_t | \boldsymbol{\mu}_t) \approx \nabla_{\mathbf{u}_t} M^*(\mathbf{u}_t | \boldsymbol{\mu}_t) + \boldsymbol{\varepsilon}$, where $\nabla_{\mathbf{u}_t} H^*(\mathbf{u}_t | \boldsymbol{\mu}_t)$ and $\nabla_{\mathbf{u}_t} M^*(\mathbf{u}_t | \boldsymbol{\mu}_t)$ denote the underlying*

truth gradients of functions $H(\cdot)$ and $M(\cdot)$, respectively. It is obvious that $\nabla_{\mathbf{u}_t} M^*(\mathbf{u}_t | \boldsymbol{\mu}_t) = \nabla_{\mathbf{u}_t} H^*(\mathbf{u}_t | \boldsymbol{\mu}_t) + 2\mathbf{R}\mathbf{u}_t$ according to their definition. $\boldsymbol{\varepsilon}$ follows a multi-normal distribution with zero mean vector and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}$.

Assumption 3.2: The finite-difference equation of control iterations can be approximated by the stochastic differential equation. Specifically, the difference equation between two successive control iterations searched by Algorithm 2 ($\Delta \mathbf{u}_t = -\alpha \nabla_{\mathbf{u}_t} M(\mathbf{u}_t | \boldsymbol{\mu}_t)$) can be approximated by $d\mathbf{u}_t = -\alpha \nabla_{\mathbf{u}_t} M(\mathbf{u}_t | \boldsymbol{\mu}_t) dt$. Combining with Assumption 3.1, we have $d\mathbf{u}_t = -\alpha \nabla_{\mathbf{u}_t} M^*(\mathbf{u}_t | \boldsymbol{\mu}_t) dt + \alpha \mathbf{B} dW_t$, where $\mathbf{B}^T \mathbf{B} = \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}$ and W_t is a standard Wiener process.

Assumption 3.1 indicates that the gradient calculated by the SGD algorithm can approximate the truth gradient well with a normal error. This assumption is generally adopted in deep learning or large-scale model optimization (Stephan et al., 2017; Bottou et al., 2018; Wu et al., 2020), and we also numerically verify it in our CMP case study in Appendix B.1. Assumption 3.2 holds when the step size in the control iteration approaches 0, where discrete differences can be approximated by continuous differentials. According to Assumptions 3.1 and 3.2 on the unknown functions $H(\cdot)$, Theorem 1 shows the theoretical property of the searched control recipes in Algorithm 2.

Theorem 1: The control recipes obtained using Algorithm 2 converge to the optimal recipes.

The proof is provided in Appendix A.2.

Theorem 1 guarantees the optimality of Algorithm 2 when process models are unknown for complex manufacturing processes in general. In practical applications, Algorithm 2 can be further improved to reduce the number of iterations. Two main aspects can be considered. First, finding a better initialization of control recipes (i.e., $\mathbf{u}_t^{[0]}$) based on historical runs. When $t > 1$, the distribution of disturbances from historical runs can provide information for current disturbance prediction, and the initial control recipe is chosen as the optimized control at the last run (i.e., \mathbf{u}_{t-1}^*) especially when disturbances are highly correlated. Second, choosing a dynamic step size (α) can also accelerate the convergent rate. For example, according to Castera et al. (2022), if the unknown function $M(\cdot)$ is twice-differentiable, a dynamic step size can be chosen as:

$$\alpha^k(\mathbf{u}_t) = \begin{cases} \alpha_0 \cdot \frac{\|\Delta \mathbf{u}_t^{[k]}\|^2}{\langle \Delta \mathbf{u}_t^{[k]}, \Delta \mathbf{g}_M^{[k]} \rangle} & \text{if } \langle \Delta \mathbf{u}_t, \Delta \mathbf{g}_k \rangle > 0 \\ \nu & \text{otherwise,} \end{cases}$$

where $\Delta \mathbf{u}_t^{[k]} = \mathbf{u}_t^{[k]} - \mathbf{u}_t^{[k-1]}$, $\Delta \mathbf{g}_M^{[k]} = \nabla M(\mathbf{u}_t^{[k]}) - \nabla M(\mathbf{u}_t^{[k-1]})$, and $\alpha_0, \nu > 0$ are constant hyper-parameters in the algorithm that represent the scaling factor and large step-size used in locally concave regions, respectively. We also present more discussions on the improvement of Algorithm 2 in Appendix B.2.

Specifically, if the function $H(\mathbf{u}_t | \boldsymbol{\mu}_t)$ can also be approximated by its second-order Taylor expansion, which implies that the process model $g(\mathbf{u}_t)$ can be approximated by linear, piecewise-linear or local linear functions, more theoretical properties are obtained related to the closed-form solution (Proposition 1), the stochastic searching process (Theorem 2), and the stationary distribution (Theorem 3) of the control recipes.

Proposition 1: *If function $H(\mathbf{u}_t | \boldsymbol{\mu}_t)$ has a minimum at an unknown point $\tilde{\mathbf{u}}_t$, i.e., $\tilde{\mathbf{u}}_t := \arg \min_{\mathbf{u}_t} H(\mathbf{u}_t | \boldsymbol{\mu}_t)$, the optimal control recipe to minimize the cost C_t is $\mathbf{u}_t^* =$*

$$(\mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{R})^{-1} \mathbf{G}^T \mathbf{Q} \mathbf{G} \tilde{\mathbf{u}}_t, \text{ where } \mathbf{G} = \begin{bmatrix} \frac{\partial g_1}{\partial \tilde{u}_1} & \dots & \frac{\partial g_1}{\partial \tilde{u}_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n}{\partial \tilde{u}_1} & \dots & \frac{\partial g_n}{\partial \tilde{u}_m} \end{bmatrix}_{n \times m} \text{ is the gradient matrix of function } g(\cdot).$$

The proof is provided in Appendix A.3.

Theorem 2: *The control search process for \mathbf{u}_t^* in Algorithm 2 can be approximated by an Ornstein-Uhlenbeck process, i.e., $d\mathbf{u}_t = \boldsymbol{\Psi}(\mathbf{u}_t^* - \mathbf{u}_t)dt + \boldsymbol{\sigma}dW_t$, where $\boldsymbol{\Psi} = 2\alpha[\mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{R}]$, $\boldsymbol{\sigma} = \alpha \mathbf{B}$ and $\mathbf{B}^T \mathbf{B} = \boldsymbol{\Sigma}_\varepsilon$.*

The proof is provided in Appendix A.4.

Theorem 3: *The stationary distribution of the control recipe searched in Algorithm 2 can be approximated by a multi-normal distribution, which is expressed as*

$$\mathbf{u}_t \sim MN\left(\mathbf{u}_t^*, \frac{1}{2} \boldsymbol{\sigma}^T \boldsymbol{\Psi}^{-1} \boldsymbol{\sigma}\right), \quad (11)$$

where $\boldsymbol{\Psi} = 2\alpha[\mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{R}]$ and $\boldsymbol{\sigma} = \alpha \mathbf{B}$.

The proof is provided in Appendix A.5.

In summary, Theorem 1 guarantees the control searched in Algorithm 2 can converge to the underlying optimal one in general. Specifically, if the unknown function $H(\cdot)$ can be approximated by its second-order Taylor expansion, Theorems 2 and 3 propose the explicit formulations of the search process and stationary distribution of control recipes, respectively. Furthermore, from the distribution of control recipes in Equation (11), we find that smaller step sizes can reduce the variations of \mathbf{u}_t .

B. Outer loop: Bayesian inference of disturbances

In Section 2.1, the prior probability of disturbance \mathbf{d}_t is defined as $p(\mathbf{d}_t|\mathbf{D}_{t-1})$ depending on its trajectory $\mathbf{D}_{t-1} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_{t-1}]$. After making control decisions and observing the system output \mathbf{y}_t , we can update the posterior probability of disturbance \mathbf{d}_t using Bayesian inference as follows:

$$p(\mathbf{d}_t|\mathbf{y}_t) = \frac{p(\mathbf{d}_t|\mathbf{D}_{t-1})p(\mathbf{y}_t|\mathbf{d}_t)}{p(\mathbf{y}_t)} \propto p(\mathbf{d}_t|\mathbf{D}_{t-1})p(\mathbf{y}_t|\mathbf{d}_t), \quad (12)$$

where the prior distribution $p(\mathbf{d}_t|\mathbf{D}_{t-1})$ is typically determined based on prior domain knowledge with a known formulation. In theory, any type of distribution can be used based on the amount of available information, while in practice, uniform and normal distributions are most commonly adopted (Lye et al, 2021). However, if the probability density function is too complicated to be analytically derived or directly simulated, Monte Carlo methods can be used in disturbance simulation. The conditional probability $p(\mathbf{y}_t|\mathbf{d}_t)$ is obtained based on the system outputs after the convergence of control recipes in Algorithm 2. If the existing known distributions can be used to approximate the data, maximum likelihood estimation is used to estimate parameters in the distribution. Otherwise, if the probability density function is not easily parametrizable by an explicit formulation, other nonparametric methods such as kernel density or orthogonal series density estimation can be used to approximate $p(\mathbf{y}_t|\mathbf{d}_t)$ (Agarwal et al., 2016).

In semiconductor manufacturing processes, the disturbance \mathbf{d}_t is generally supposed to be normally distributed given its historical trajectory (Teng et al, 2007; Wang and Han, 2013; Tseng et al, 2019). Specifically, if $p(\mathbf{y}_t|\mathbf{d}_t)$ can also be approximated by a normal distribution, we have Proposition 2 for the posterior distribution of the disturbance using Bayesian inference theory as follows.

Proposition 2: *If the prior distribution of the disturbance follows multi-normal distribution as $\mathbf{d}_t|\mathbf{D}_{t-1} \sim MN(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, the explicit expression of the posterior distribution disturbances after observing the system output \mathbf{y}_t is given by:*

$$p(\mathbf{d}_t|\mathbf{y}_t) \propto \exp \left\{ -\frac{1}{2} \left(\left(\mathbf{y}_t - \frac{1}{N} \sum_{i=1}^N \hat{\mathbf{y}}_t(\hat{\mathbf{u}}_t(i)) \right)^T \frac{1}{N} \boldsymbol{\Sigma}_y^{-1} \left(\mathbf{y}_t - \frac{1}{N} \sum_{i=1}^N \hat{\mathbf{y}}_t(\hat{\mathbf{u}}_t(i)) \right) + \right. \right. \\ \left. \left. (\mathbf{d}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_t^{-1} (\mathbf{d}_t - \boldsymbol{\mu}_t) \right) \right\},$$

where $\boldsymbol{\Sigma}_y$ is the sample variance matrix of system output after the convergence of control recipes.

Notably, other distributions of disturbances can also be updated by Bayesian inference methods using Monte Carlo methods. By analyzing the posterior probability of disturbances, we obtain a more reliable prior distribution to reduce variations of disturbances in the next run. Algorithm 3 presents the Bayesian update procedure of disturbance as follows.

Algorithm 3. Update distributions of disturbances

Initialize $t, \mathbf{u}_1^{[0]}$, the prior distribution of disturbance $p(\cdot)$, initial disturbance \mathbf{d}_0 .

For $t = 1:T$

$$\boldsymbol{\mu}_t = \int_{-\infty}^{+\infty} \mathbf{d}_t \cdot p(\mathbf{d}_t | \mathbf{D}_{t-1}) d\mathbf{d}_t$$

$\bar{\mathbf{u}}_t \leftarrow \text{Control_Search}(\boldsymbol{\mu}_t)$

*/*Algorithm 2*/*

Take control $\bar{\mathbf{u}}_t$, and record the system output \mathbf{y}_t .

Update the disturbance according to:

$$p(\mathbf{d}_t | \mathbf{y}_t) = \frac{p(\mathbf{d}_t | \mathbf{D}_{t-1}) p(\mathbf{y}_t | \mathbf{d}_t)}{p(\mathbf{y}_t)} \propto p(\mathbf{d}_t | \mathbf{D}_{t-1}) p(\mathbf{y}_t | \mathbf{d}_t)$$

Update $p(\mathbf{d}_{t+1} | \mathbf{D}_t)$.

End for

3.2 Real-time control in Phase II

In real applications of semiconductor manufacturing processes, after control optimization by VM systems in Phase I, real-time control recipes need to be directly determined in practical manufacturing processes. Therefore, in this section, we propose a real-time control algorithm used in Phase II.

Suppose that manufacturing environments and process models keep stable in Phases I and II, and it is reasonable that the control recipes searched in Phase I can be applied in Phase II. We denote the offline experimental database collected in Phase I as $\{D_off\}$. Each sample in $\{D_off\}$ consists of the control recipes, system output, and the distribution of disturbances, i.e., $[\mathbf{u}_t, \mathbf{y}_t, \mathbf{d}_t] \in \{D_off\}$.

Due to the optimality of searched control recipes in the offline database $\{D_off\}$, it can be used as a “memory buffer” for online real-time control. Since the key hidden variables in manufacturing processes are disturbances, real-time control decisions can be implemented by matching the closest offline disturbance in $\{D_off\}$ with the real-time inferred disturbance and choosing the corresponding control recipe as the real-time recipe. Specifically, we can divide the real-time control

process into three steps: (a) update the real-time distribution of the online disturbance. (b) Match closest offline disturbance \mathbf{d}_s by:

$$\mathbf{d}_s := \arg \min_{\mathbf{d} \in \{D_off\}} \mathbb{D}_{KL}(p(\mathbf{d}) || q(\mathbf{d}_t^{on} | \mathbf{D}_{t-1}^{on})), \quad (13)$$

where \mathbf{d}_t^{on} is online real-time disturbance and $\mathbb{D}_{KL}(\cdot || \cdot)$ is Kullback-Leibler divergence. To distinguish the online disturbance, we use $q(\cdot)$ to denote its prior distribution. (c) Choose the control recipe \mathbf{u}_s corresponding to \mathbf{d}_s as the real-time control strategy. Figure 4 illustrates this real-time decision process in detail. Notably, as the size of database $\{D_off\}$ increases, the divergence between the online and offline disturbance becomes smaller, and the control performs better. Further numerical discussions on the size of $\{D_off\}$ are provided in Section 4.2 and a detailed algorithm for the real-time control scheme is presented in Algorithm 4.

Algorithm 4. Real-time control in Phase II

Input: Historical offline database $\{D_off\}$, initial system output y_0 , prior distribution of online disturbance $q(\cdot)$

For $t = 1:T$

$$\mathbf{d}_s := \arg \min_{\mathbf{d} \in \{D_off\}} \mathbb{D}_{KL}[p(\mathbf{d}) || q(\mathbf{d}_t^{on} | \mathbf{D}_{t-1}^{on})]$$

Take the control recipe \mathbf{u}_s corresponding to \mathbf{d}_s , and collect the output \mathbf{y}_t .

Update the disturbance according to

$$q(\mathbf{d}_t^{on} | \mathbf{y}_t) = \frac{q(\mathbf{d}_t^{on} | \mathbf{D}_{t-1}^{on}) p(\mathbf{y}_t | \mathbf{d}_t^{on})}{p(\mathbf{y}_t)} \propto q(\mathbf{d}_t^{on} | \mathbf{D}_{t-1}^{on}) p(\mathbf{y}_t | \mathbf{d}_t^{on}).$$

Calculate $q(\mathbf{d}_{t+1}^{on} | \mathbf{D}_t^{on})$.

End for

4. Numerical study and comparison

To show the performance of the proposed MFRL-BI control scheme, we propose numerical studies based on a nonlinear chemical mechanical planarization (CMP) process in semiconductor manufacturing. Section 4.1 numerically verifies the improvement by using Bayesian inference in the MFRL-BI controller compared with the basic MFRL controller. More sensitivity analysis of the control performance for the MFRL-BI controller is conducted in Section 4.2. In Section 4.3, we focus on performance comparisons between the MFRL-BI controller and other control benchmarks.

4.1 Improvement of MFRL-BI controller

Due to the privacy of real CMP data, Khuri (1996) proposed an experiment tool and designed a nonlinear process model to describe the CMP process, which is widely used in CMP data simulation (Del Castillo and Yeh, 1998). In this section, we also follow their simulation for data generation. The control recipe \mathbf{u}_t consists of three dimensions (i.e., $\mathbf{u}_t = [u_t^{(1)}, u_t^{(2)}, u_t^{(3)}]^T$), which represent the backpressure downforce, platen speed, and slurry concentration, respectively. The two dimensions of the system outputs ($\mathbf{y}_t = [y_t^{(1)}, y_t^{(2)}]^T$) to reflect the manufacturing quality are removal rate and within-wafer standard deviation with target levels as $\mathbf{y}^* = [2200, 400]^T$. Without loss of generality, the initial system output is set as the target levels.

Specifically, following the nonlinear model proposed by Del Castillo and Yeh (1998), we use the following formulation to simulate data in the CMP process at each run t

$$\mathbf{y}_t = \mathbf{C}\mathbf{X}_t + \mathbf{d}_t, \quad (14)$$

where \mathbf{C} is the parameter matrix defined as

$$\mathbf{C} = \begin{bmatrix} 2756.5 & 547.6 & 616.3 & -126.7 & -1109.5 & -286.1 & 989.1 & -52.9 & -156.9 & -550.3 & -10 \\ 746.3 & 62.3 & 128.6 & -152.1 & -289.7 & -32.1 & 237.7 & -28.9 & -122.1 & -140.6 & 1.5 \end{bmatrix},$$

\mathbf{X}_t consists of constant, linear, and quadratic terms of control recipes at run t

$$\mathbf{X}_t = [1, u_t^{(1)}, u_t^{(2)}, u_t^{(3)}, [u_t^{(1)}]^2, [u_t^{(2)}]^2, [u_t^{(3)}]^2, u_t^{(1)}u_t^{(2)}, u_t^{(1)}u_t^{(3)}, u_t^{(2)}u_t^{(3)}, t]^T.$$

$\mathbf{d}_t = [d_t^{(1)}, d_t^{(2)}]^T$ are two dimensions of disturbances that follow two independent IMA(1,1) processes, and the total number of runs T is 50. Based on this setting, we analyze the performance of the proposed MFRL-BI controller and compare it with the basic MFRL controller.

We first consider a special case where there is no extra cost associated with control actions, i.e., $\mathbf{R} = \mathbf{0}$, the control cost is $C_t(\mathbf{u}_t) = (\mathbf{y}_t - \mathbf{y}^*)^T \mathbf{Q}(\mathbf{y}_t - \mathbf{y}^*)$. Under this setting, the basic MFRL and MFRL-BI controllers are applied for real-time control, and the corresponding system outputs are used to evaluate the performances of these two controllers. To make a fair comparison, we search control recipes for 2000 iterations at each run in both Algorithms 1 and 2 in the basic MFRL and MFRL-BI controllers, respectively. After collecting data from 1000 production cycles in $\{D_off\}$, we make the online real-time control by matching the disturbances in $\{D_off\}$ with the real-time one using Algorithm 4. Figure 5 illustrates the boxplot of system outputs in Phase II with 100 replications. The two panels in Figures 5(a) and 5(b) correspond to the two dimensions of \mathbf{y}_t . As shown, system

outputs based on the basic MFRL controller have relatively large variations and significant deviations when dealing with system drifts, while the proposed MFRL-BI controller can keep the system outputs well close to their desired targets, even though the process model is unknown.

Generally, executing control has extra control cost during the manufacturing process, the total cost is: $C_t(\mathbf{u}_t) = (\mathbf{y}_t - \mathbf{y}^*)^T \mathbf{Q}(\mathbf{y}_t - \mathbf{y}^*) + \mathbf{u}_t^T \mathbf{R} \mathbf{u}_t$, where $\mathbf{R} \neq \mathbf{0}$. For example, we set $\mathbf{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\mathbf{R} = \begin{bmatrix} 10 & & \\ & 10 & \\ & & 5 \end{bmatrix}$. The mean control cost (MCC) at each run (defined as $\sum_{t=1}^T C_t(\mathbf{y}_t, \mathbf{u}_t) / T$) is used as performance criteria. Table 1 summarizes the mean and standard deviation of MCC in basic MFRL, MFRL-BI controllers, and without control under 100 replications.

As shown in Table 1, in comparison to without control, the basic MFRL controller presented in Algorithm 1 substantially reduces the control cost. Nonetheless, the performance of the basic MFRL does not fulfill the accuracy specifications for semiconductor manufacturing. Upon updating the distribution of disturbances by Algorithms 2 to 4, it is observed that the mean of control cost reduces by 97% in comparison to the basic MFRL controller. Table 1 demonstrates the efficient performance of the MFRL-BI controller in further reducing the control cost during the manufacturing process.

4.2 Sensitivity analysis of MFRL-BI controller

To further analyze the sensitivity of the performance of the MFRL-BI controller and verify its stability, in this section, we mainly focus on two factors that can affect the control performance, i.e., parameters in the process model and the size of offline data required for online real-time control.

As described in Equation (14), the coefficient matrix \mathbf{C} is used to clarify the impact of linear and quadratic forms. We divided the coefficient matrix \mathbf{C} into three segments corresponding to linear terms, quadratic terms and others, which are defined as \mathbf{C}_1 , \mathbf{C}_2 and \mathbf{C}_0 respectively. Therefore, we have:

$$\begin{aligned} \mathbf{C}_1 &= \begin{bmatrix} 547.6 & 616.2 & -126.7 \\ 62.3 & 128.6 & -152.1 \end{bmatrix}, \\ \mathbf{C}_2 &= \begin{bmatrix} -1109.5 & -286.1 & 989.1 & -52.9 & -156.9 & -550.3 \\ -289.7 & -32.1 & 237.7 & -28.9 & -122.1 & -140.6 \end{bmatrix}, \\ \mathbf{C}_0 &= \begin{bmatrix} 2756.5 & -10 \\ 746.3 & 1.5 \end{bmatrix}. \end{aligned}$$

With this segmentation, our primary focus lies in examining the performance sensitivity of the proposed MFRL-BI model under various values of \mathbf{C}_1 and \mathbf{C}_2 . An additional scale coefficient (i.e.,

$\rho = 0.5, 1, \text{ and } 1.5$) is applied to C_1 and C_2 , respectively to modify their values. We take the case with $R = \begin{bmatrix} 10 & & \\ & 10 & \\ & & 5 \end{bmatrix}$ as an example, to summarize the corresponding performance of the MFRL-BI controller in Table 2.

Table 2 presents the mean and standard deviations of the real-time control costs in 100 replications. It is evident that the MFRL-BI controller demonstrates inevitable variability with parameter modifications. Nevertheless, it consistently achieves superior performance than the basic MFRL controller with substantially much lower control cost, which is acceptable in practical applications.

Another important factor influencing the MFRL-BI method is the size of offline database, which is defined as $\{D_{off}\}$ and used to collect the experimental data related to control recipes, system output, and disturbances. $\{D_{off}\}$ serves as a crucial foundation for the data-driven control optimization in phase II. Therefore, the data size of $\{D_{off}\}$ has significant impacts on the accuracy of real-time control. To further validate the effects of $\{D_{off}\}$, we conduct a sensitivity analysis on the size of $\{D_{off}\}$. The accuracy is evaluated by the results of real-time control in Phase II, which consists of the mean and standard error of MCC. Additionally, the efficiency of data collection is evaluated by computation time in Phase I. Table 3 provides more details on the effects of sample size of $\{D_{off}\}$.

As shown in Table 3, the accuracy of online real-time control converges when the size of $\{D_{off}\}$ exceeds 500. While the computation time gets larger with the increase of data size. We can conclude from Table 3 that although performance convergence is not attained until the size of offline database $\{D_{off}\}$ reaches 500, satisfactory results are already observed when the size equals 100.

4.3 Comparison with other controllers

As the MFRL-BI addresses the control optimization when process models are unknown, we primarily consider a nonparametric-based benchmark, which can approximate the process model and optimize the control recipes. In this CMP process, due to the multiple dimensions of control recipes, we first apply the multivariate adaptive regression spline (MARS) method for the process model

approximation and control optimization. Secondly, to demonstrate the superiority of the proposed MFRL-BI controller, we also introduce other benchmarks, i.e., process model-based and theoretical optimal controllers for further performance comparison.

A. MARS-based controller

MARS is one of the nonparametric regression methods proposed by Friedman (1991) for multivariate independent variables, which is used to approximate the unknown CMP process model with multiple control variables. Similar to the real-time control in MFRL-BI, we use 1000 production cycles ($N = 1000$) of offline data with 50 runs ($T = 50$) in each cycle to train the MARS-based approximate process model.

Specifically, MARS uses basis functions based on addition and multiplication operations of piecewise linear hinge functions formulated as $(x - t)_+ := \max(x - t, 0)$ and $(x - t)_- := \max(t - x, 0)$ to approximate the unknown function. Therefore, the approximated process model can be formulated as:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^p a_i B_i(\mathbf{x}),$$

where a_i is constant coefficient, $B_i(\mathbf{x})$ is a basis function that can be formulated as a constant, a hinge function or a product of two or more hinge functions, and p is the total number of basis searched by MARS. To emphasize the dynamics of the process model, we specify the independent multivariate \mathbf{x} as $\mathbf{x}_{n,t} = [\mathbf{u}_{n,t}, t]$, where t is the index of runs, $\mathbf{u}_{n,t} \in \mathbb{R}^3$ is the control recipe and n is the index of production cycles.

MARS aims to search basis functions that give the maximum reduction in sum-of-squares residual errors calculated by $\sum_{n=1}^N \sum_{t=1}^T (y_{n,t} - \hat{f}(\mathbf{x}_{n,t}))^2$. The total number of basis functions are determined by general cross validation (GCV) to avoid overfitting, which is calculated by $\frac{\sum_{n=1}^N \sum_{t=1}^T (y_{n,t} - \hat{f}(\mathbf{x}_{n,t}))^2}{(1 - \frac{K(\lambda)}{N \cdot T})^2}$, and $K(\lambda)$ is the effective number of parameters. As a result, training by offline data, we obtain the basis functions in Table 4, where $u^{(1)}$, $u^{(2)}$ and $u^{(3)}$ denote the three dimensions of $\mathbf{u}_{n,t}$, respectively.

Based on basis functions in Table 4, the corresponding approximated process model is formulated as:

$$\begin{aligned}
y^{(1)} &= 2573.8 - 1217.8 * BF1_1 - 57.908 * BF1_2 + 2643.4 * BF1_3 - 976.59 * BF1_4 - 639.01 \\
&\quad * BF1_5 - 274.38 * BF1_6 + 628.37 * BF1_7 + 907.16 * BF1_8 - 9.9129 * BF1_9 \\
&\quad + 9.9731 * BF1_10 - 467.05 * BF1_11 + 1670.4 * BF1_12 - 1445.5 * BF1_13 \\
&\quad - 101.07 * BF1_14 + 180.35 * BF1_15 - 1117.5 * BF1_16 + 648.27 * BF1_17, \\
y^{(2)} &= 533.62 - 1154.8 * BF2_1 + 823.85 * BF2_2 + 418.12 * BF2_3 + 66.818 * BF2_4 - 147.51 \\
&\quad * BF2_5 + 136.74 * BF2_6 - 127.45 * BF2_7 + 37.34 * BF2_8 - 134.2 * BF2_9 + 77.12 \\
&\quad * BF2_10 + 1.4915 * BF2_11 - 1.5282 * BF2_12 + 137.54 * BF2_13 - 205.18 \\
&\quad * BF2_14 + 115.55 * BF2_15 - 200.39 * BF2_16 - 312.22 * BF2_17 - 240.83 \\
&\quad * BF2_18.
\end{aligned} \tag{15}$$

Then the online control recipe is optimized based on Equation (15) to minimize the mean control cost (MCC): $\sum_{t=1}^T C_t(\mathbf{y}_t, \mathbf{u}_t) / T$. Following the aforementioned setting, we have the system outputs in Figure 6. It is apparent that the MARS-based controller may experience challenges with local performance due to its piecewise basis. Therefore, the nonparametric method may also have difficulties in estimating the approximate model, thereby it may not be suitable for dealing with unstable and unobservable disturbances.

To further improve the performance of MARS controllers, we add an EWMA scheme to predict disturbances. The control recipe is calculated by:

$$\begin{cases} \hat{\mathbf{d}}_{t+1} = \omega \hat{\mathbf{d}}_t + (1 - \omega)(\mathbf{y}_t - \mathbf{y}^*) \\ \mathbf{u}_{t+1} = \min_{\mathbf{u}} (\hat{f}(\mathbf{u}) + \hat{\mathbf{d}}_{t+1} - \mathbf{y}^*)^2. \end{cases}$$

The online control process is replicated 100 times, and the MCC of the MARS, MARS-EWMA, and the proposed MFRL-BI controllers are compared in Table 5. As shown, although the MARS-EWMA improves the performances of the MARS controller significantly, the MCC is still much larger than MFRL-BI.

B. Process model-based controller

Due to the effect of hidden and unstable disturbances, MARS-based nonparametric method cannot approximate the process model accurately, thereby leading limited control performance. To further clarify the effects of the model accuracy and disturbance variation, we propose a nonlinear process model-based controller as another benchmark. Specifically, the process model is supposed to be

known in this setting. We generate the same offline data with MARS-based controllers, which are used to estimate the parameters (θ) in the process model, and the control recipes are optimized by:

$$\mathbf{u}_t = \min_{\mathbf{u}} (f(\mathbf{u}|\hat{\theta}) - \mathbf{y}^*)^2.$$

Actually, in Section 4.1, the nonlinear CMP process is simulated by a multivariate quadratic model with linear drift and IMA disturbances, which has an underlying optimal control strategy. According to Sachs et al. (1995), if the drift can be approximated by a known model, the IMA disturbances can be effectively compensated for by an optimal EWMA controller with the corresponding coefficient. To demonstrate the performance of different control methods, we also propose the theoretical optimal controller for comparison. Table 5 presents the results of different control methods.

From Table 5, we find that the performance of MARS-related (i.e., MARS and MARS-EWMA) controllers is limited due to the accuracy of the approximate process model trained by offline data. Furthermore, even if the formulation of the process model is known, it remains challenging to predict the unobservable disturbances. Therefore, by integrating the MFRL with Bayesian inference, the advantages of the MFRL-BI controller are sufficiently verified, whose performance approaches to the theoretical optimal controller.

5. Conclusion

This work designs a new process control scheme by model-free reinforcement learning to reduce the system variations in semiconductor manufacturing when the process model is unknown and complex. Due to unstable and unobservable disturbances, the basic MFRL controller usually suffers from large variations. To overcome this challenge, we update the distribution of disturbances during manufacturing processes using Bayesian inference. The algorithms of control recipe optimization and real-time control implementation are presented, and corresponding theoretical properties are also guaranteed. Through performance comparisons of the proposed MFRL-BI with basic MFRL, MARS-based and process model-based controllers, we observe that the MFRL-BI controller exhibits superior performance, particularly when underlying process models are unknown, nonlinear and complex.

Along with our research direction, several extensions can be further investigated. First, how to develop an RL-based process control model when the effects of control recipes and disturbances are correlated and not additive. Second, the constraints of control recipes can also be considered in process control optimization in future studies.

Acknowledgements

The authors acknowledge the generous support from the National Natural Science Foundation of China Grant (No. 72001139, No. 72371219, No. 72371271, and No. 52372308), Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515011656), Guangzhou-HKUST(GZ) Joint Funding Program (No. 2023A03J0651), the Guangzhou Industrial Information and Intelligent Key Laboratory Project (No. 2024A03J0628), the Nansha Key Area Science and Technology Project (No. 2023ZD003), and Project No. 2021JC02X191.

References

- Agarwal, R., Chen, Z., & Sarma, S. V. (2016). A novel nonparametric maximum likelihood estimator for probability density functions. *IEEE transactions on pattern analysis and machine intelligence*, 39(7), 1294-1308.
- Bastiaan, H. K. (1997). Process model and recipe structure, the conceptual design for a flexible batch plant. *ISA Transactions*, 36(4), 249-255.
- Bibian, S., & Jin, H. (2000). Time delay compensation of digital control for DC switchmode power supplies using prediction techniques. *IEEE Transactions on Power Electronics*, 15(5), 835-842.
- Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM review*, 60(2), 223-311.
- Box, G., & Kramer, T. (1992). Statistical process monitoring and feedback adjustment—a discussion. *Technometrics*, 34(3), 251-267.
- Castera, C., Bolte, J., Févotte, C., & Pauwels, E. (2022). Second-order step-size tuning of SGD for non-convex optimization. *Neural Processing Letters*, 54(3), 1727-1752.
- Chang, Y. J., Kang, Y., Hsu, C. L., Chang, C. T., & Chan, T. Y. (2006, July). Virtual metrology technique for semiconductor manufacturing. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings* (pp. 5289-5293). IEEE.
- Chen, A., & Guo, R. S. (2001). Age-based double EWMA controller and its application to CMP processes. *IEEE Transactions on Semiconductor Manufacturing*, 14(1), 11-19.
- Chen, J., Munoz, J., & Cheng, N. (2012). Deterministic and stochastic model based run-to-run control for batch processes with measurement delays of uncertain duration. *Journal of Process Control*, 22(2), 508-517.
- Clerget, C. H., Grimaldi, J. P., Chebre, M., & Petit, N. (2016). Run-to-run control with nonlinearity and delay uncertainty. *IFAC-PapersOnLine*, 49(7), 145-152.
- Del Castillo, E., & Hurwitz, A. M. (1997). Run-to-run process control: Literature review and extensions. *Journal of Quality Technology*, 29(2), 184-196.
- Del Castillo, E., & 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.
- Hankinson, M., Vincent, T., Irani, K. B., & Khargonekar, P. P. (1997). Integrated real-time and run-to-run control of etch depth in reactive ion etching. *IEEE Transactions on Semiconductor Manufacturing*, 10(1), 121-130.
- He, F., Wang, K., & Jiang, W. (2009). A general harmonic rule controller for run-to-run process control. *IEEE Transactions on Semiconductor Manufacturing*, 22(2), 232-244.
- Ingolfsson, A., & Sachs, E. (1993). Stability and sensitivity of an EWMA controller. *Journal of Quality Technology*, 25(4), 271-287.
- Kang, P., Lee, H. J., Cho, S., Kim, D., Park, J., Park, C. K., & Doh, S. (2009). A virtual metrology system for semiconductor manufacturing. *Expert Systems with Applications*, 36(10), 12554-12561.
- Kazemzadeh, R. B., Karbasian, M., & Moghadam, M. B. (2008). Design and investigation of EWMA and double EWMA with quadratic process model in R2R controllers. *Quality & Quantity*, 42(6), 845-857.
- Khamaru, K., Xia, E., Wainwright, M. J., & Jordan, M. I. (2021). Instance-optimality in optimal value estimation: Adaptivity via variance-reduced Q-learning. *arXiv preprint arXiv:2106.14352*.
- Khuri, A. (1996, April). Response surface methods for multiresponse experiments. In *13th SEMATECH Statistical Methods Symposium*.
- Kiefer, J., & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 462-466.
- Kim, S., Jang, J., & Kim, C. O. (2021). A run-to-run controller for a chemical mechanical planarization process using least squares generative adversarial networks. *Journal of Intelligent Manufacturing*, 32, 2267-2280.
- Li, Y., Du, J., & Jiang, W. (2024). Reinforcement Learning for Process Control with Application in Semiconductor Manufacturing. *IIEE Transactions*, 56(6), 585-599.
- Liu, K., Chen, Y., Zhang, T., Tian, S., & Zhang, X. (2018). A survey of run-to-run control for batch processes. *ISA transactions*, 83, 107-125.

- Lye, A., Cicirello, A., & Patelli, E. (2021). Sampling methods for solving Bayesian model updating problems: A tutorial. *Mechanical Systems and Signal Processing*, 159, 107760.
- Ma, Z., & Pan, T. (2024). Deep reinforcement learning-assisted extended state observer for run-to-run control in the semiconductor manufacturing process. *Transactions of the Institute of Measurement and Control*, 01423312241229492.
- Mandt, S., Hoffman, M. D., & Blei, D. M. (2017). Stochastic gradient descent as approximate Bayesian inference. *arXiv preprint arXiv:1704.04289*.
- Moya-Lasheras, E., Ramirez-Laboreo, E., & Serrano-Seco, E. (2023). Run-to-Run Adaptive Nonlinear Feedforward Control of Electromechanical Switching Devices. *IFAC-PapersOnLine*, 56(2), 5358-5363.
- Nian, R., Liu, J., & Huang, B. (2020). A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, 139, 106886.
- Park, S. J., Lee, M. S., Shin, S. Y., Cho, K. H., Lim, J. T., Cho, B. S., & Park, C. H. (2005). Run-to-run overlay control of steppers in semiconductor manufacturing systems based on history data analysis and neural network modeling. *IEEE Transactions on Semiconductor Manufacturing*, 18(4), 605-613.
- Recht, B. (2019). A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2, 253-279.
- Stephan, M., Hoffman, M. D., & Blei, D. M. (2017). Stochastic gradient descent as approximate Bayesian inference. *Journal of Machine Learning Research*, 18(134), 1-35.
- Tom, M., Yun, S., Wang, H., Ou, F., Orkoulas, G., & Christofides, P. D. (2022). Machine learning-based run-to-run control of a spatial thermal atomic layer etching reactor. *Computers & Chemical Engineering*, 168, 108044.
- Tseng, S. T., Yeh, A. B., Tsung, F., & Chan, Y. Y. (2003). A study of variable EWMA controller. *IEEE Transactions on Semiconductor Manufacturing*, 16(4), 633-643.
- Tseng, S. T., Tsung, F., & Liu, P. Y. (2007). Variable EWMA run-to-run controller for drifted processes. *IIE Transactions*, 39(3), 291-301.
- Tseng, S. T., & Chen, P. Y. (2017). A generalized quasi-mmse controller for run-to-run dynamic models. *Technometrics*, 59(3), 381-390.
- Tseng, S. T., Mi, H. C., & Lee, I. C. (2016). A multivariate EWMA controller for linear dynamic processes. *Technometrics*, 58(1), 104-115.
- Tseng, S. T., Tsung, F., & Wu, J. H. (2019). Stability conditions and robustness analysis of a general MMSE run-to-run controller. *IIE Transactions*, 51(11), 1279-1287.
- Tsung, F., & 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, G. J., & Chou, M. H. (2005). A neural-Taguchi-based quasi time-optimization control strategy for chemical-mechanical polishing processes. *The International Journal of Advanced Manufacturing Technology*, 26(7), 759-765.
- Wang, K., & Han, K. (2013). A batch-based run-to-run process control scheme for semiconductor manufacturing. *IIE Transactions*, 45(6), 658-669.
- Wang, Y., Velswamy, K., & Huang, B. (2018). A novel approach to feedback control with deep reinforcement learning. *IFAC-PapersOnLine*, 51(18), 31-36.
- Wu, J., Hu, W., Xiong, H., Huan, J., Braverman, V., & Zhu, Z. (2020). On the noisy gradient descent that generalizes as sgd. In *International Conference on Machine Learning*, 10367-10376, PMLR.
- Zhong, J., Shi, J., & Wu, J. C. (2009). Design of DOE-based automatic process controller with consideration of model and observation uncertainties. *IEEE Transactions on Automation Science and Engineering*, 7(2), 266-273.
- Zhong, J., Liu, J., & Shi, J. (2010). Predictive control considering model uncertainty for variation reduction in multistage assembly processes. *IEEE Transactions on Automation Science and Engineering*, 7(4), 724-735.
- Zhou, S., Ding, Y., Chen, Y., & Shi, J. (2003). Diagnosability study of multistage manufacturing processes based on linear mixed-effects models. *Technometrics*, 45(4), 312-325.

Zhou, Y., Tang, X., Li, D., Lai, X., & Gao, F. (2024). Combined Iterative Learning and Model Predictive Control Scheme for Nonlinear Systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Accepted Manuscript

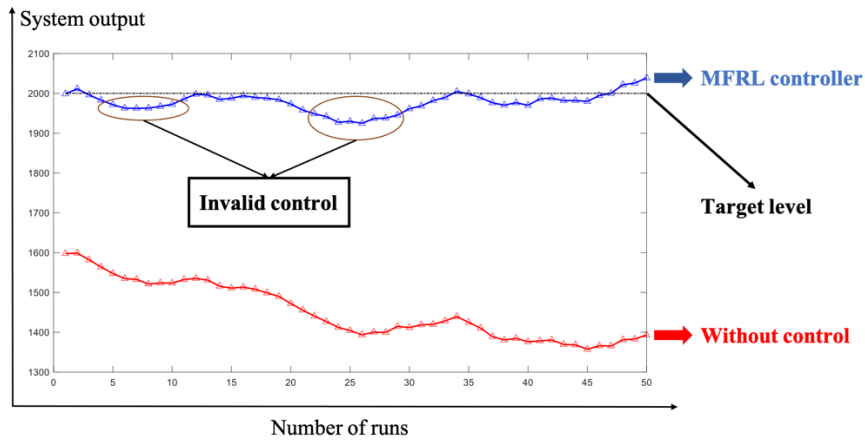


Figure 1. An example of basic MFRL controller in a CMP process

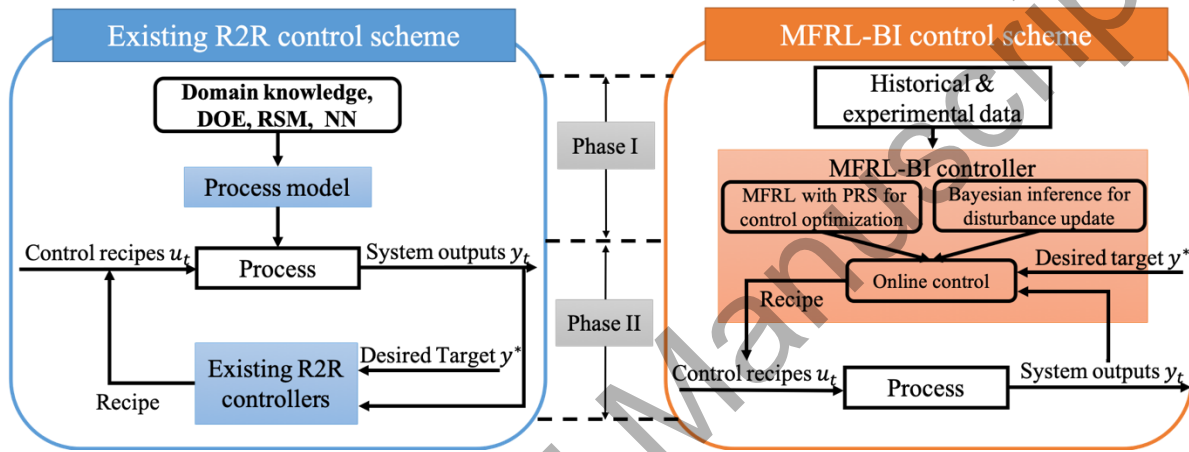


Figure 2. Difference between existing R2R and MFRL-BI control schemes

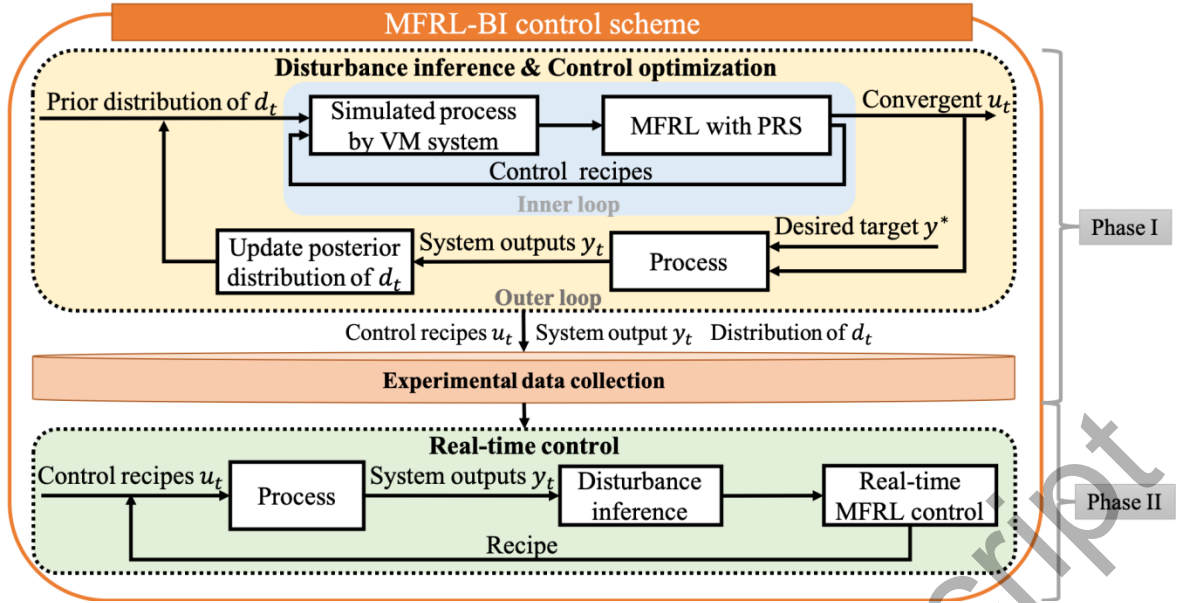


Figure 3. The methodology of the MFRL-BI controller

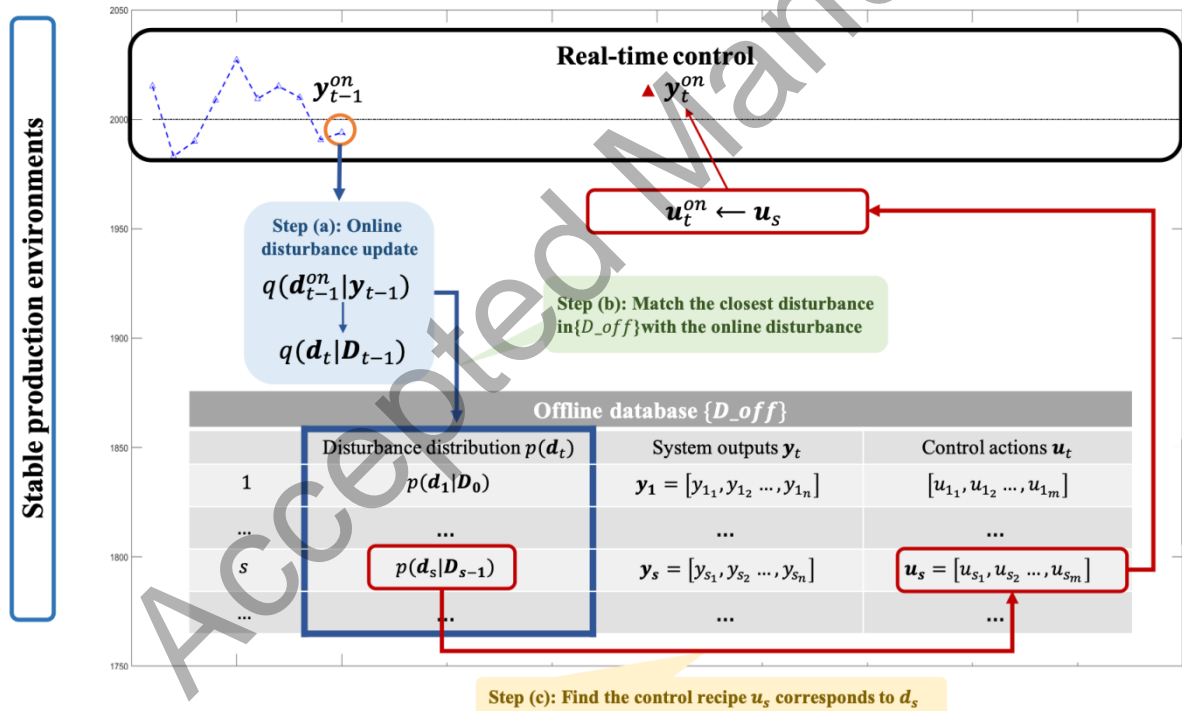


Figure 4. Illustration of the real-time control in Phase II

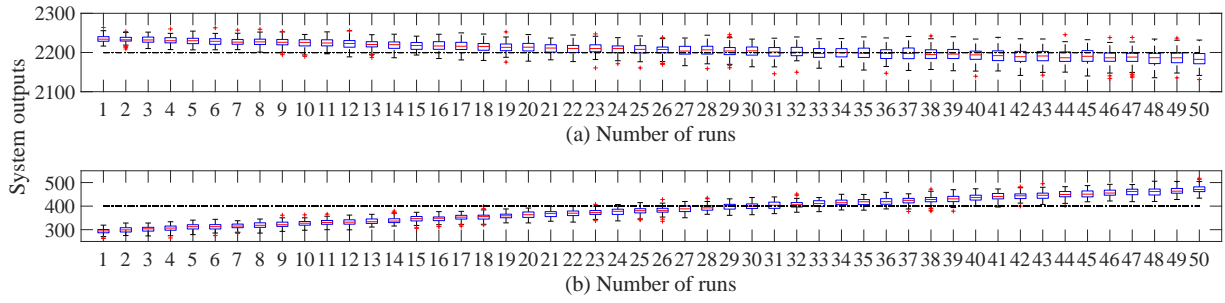


Figure 5(a). Real-time control results based on the basic MFRL controller

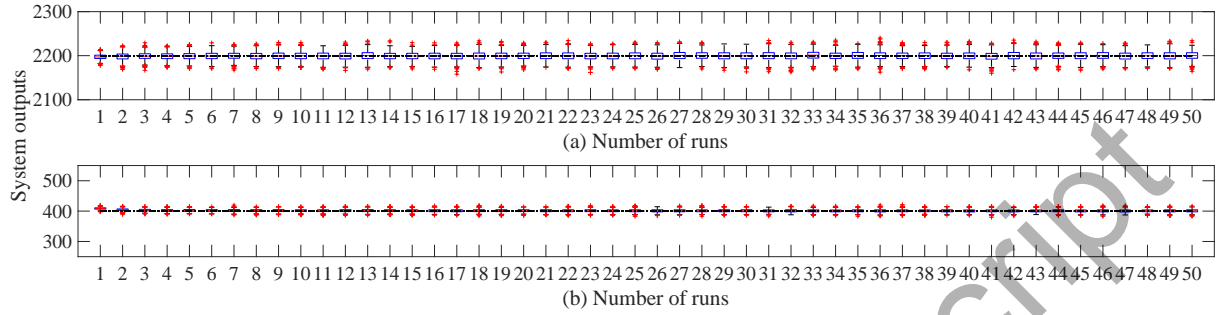


Figure 5(b). Real-time control results based on the MFRL-BI controller

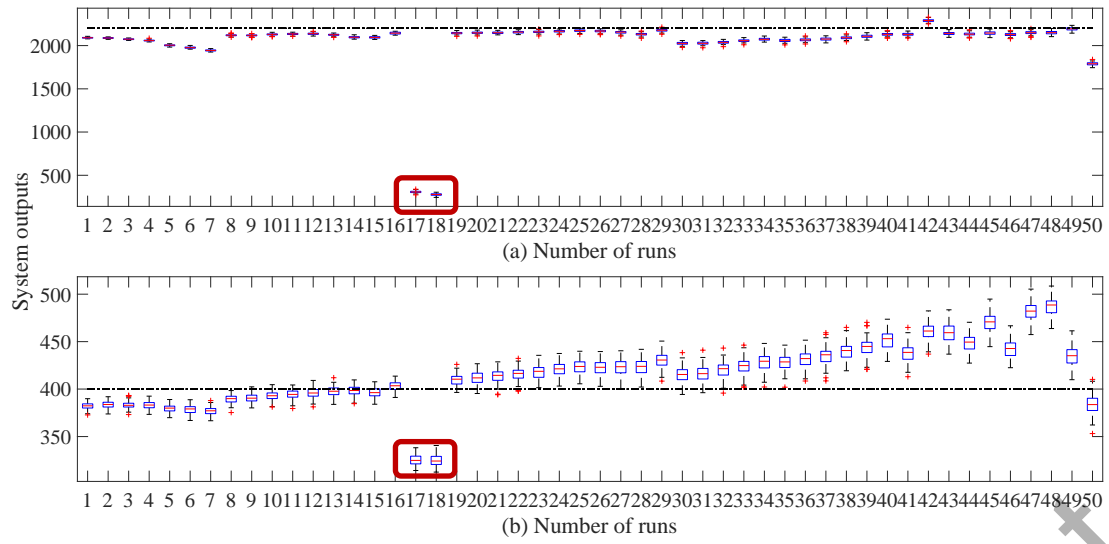


Figure 6. The performance of the MARS-based controller

Table 1. Comparisons of basic MFRL and MFRL-BI controllers

Different cases	MCC	Without control	Basic MFRL controller in Algo.1	MFRL-BI controller in Algo.2-4
$R = 0$	Mean	2.5989×10^5	3.7054×10^3	116.4702
	Std.	6.9650×10^3	382.4001	21.3797
$R \neq 0$	Mean	2.5989×10^5	5.1766×10^3	135.8367
	Std.	6.9650×10^3	386.9175	22.2550

Table 2 Sensitivity analysis on the MFRL-BI controller

Different coefficient parts	ρ	Mean of MCC	Std. of MCC
C_1	50%	251.2846	26.6052
	100%	135.8367	22.2550
	150%	128.8154	20.2492
C_2	50%	190.5863	23.9526
	100%	135.8367	22.2550
	150%	133.1384	20.9100

Table 3. Effects on offline data size

Data size of $\{D_{off}\}$	Accuracy		Efficiency (computation time)
	Mean of MCC	Std. of MCC	
100	142.0545	23.9734	148.72s
200	139.4296	23.7201	376.34s
500	135.2964	21.1445	908.17s
1000	135.5655	21.4516	1696.14s
2000	136.4871	19.4987	3579.19s
3000	134.6417	20.4617	5518.34s

Table 4. Basis functions determined by MARS

$y^{(1)}$	BF1_1	BF1_2	BF1_3	BF1_4	BF1_5
	$(u^{(1)} - 1.0246)^+$	$(1.0246 - u^{(1)})^+$	$(u^{(3)} - 0.9971)^+$	$(0.9971 - u^{(3)})^+$	$(u^{(2)} - 1.0124)^+$
	BF1_6	BF1_7	BF1_8	BF1_9	BF1_10
	$(1.0124 - u^{(2)})^+$	BF1_6 \times $(u^{(3)} - 0.2748)^+$	BF1_6 \times $(0.2748 - u^{(3)})^+$	$(t - 49)^+$	$(49 - t)^+$
	BF1_11	BF1_12	BF1_13	BF1_14	BF1_15
	BF1_5 \times $(u^{(3)} - 0.3324)^+$	BF1_5 \times $(0.3324 - u^{(3)})^+$	$(u^{(1)} - 0.4223)^+$	BF1_13 \times $(u^{(3)} - 1.6594)^+$	BF1_13 \times $(1.6594 - u^{(3)})^+$
	BF1_16	BF1_17			
	$(u^{(1)} - 1.5321)^+$	$(1.5528 - u^{(3)})^+$			
$y^{(2)}$	BF2_1	BF2_2	BF2_3	BF2_4	BF2_5
	$(u^{(1)} - 1.0267)^+$	$(1.0267 - u^{(1)})^+$	$(u^{(3)} - 0.9971)^+$	$(0.9971 - u^{(3)})^+$	BF2_3 \times $(u^{(2)} - 1.1789)^+$
	BF2_6	BF2_7	BF2_8	BF2_9	BF2_10
	BF2_3 \times $(1.1789 - u^{(2)})^+$	BF2_3 \times $(u^{(1)} - 0.3227)^+$	BF2_3 \times $(0.3227 - u^{(1)})^+$	$(u^{(2)} - 1.0496)^+$	$(1.0496 - u^{(2)})^+$
	BF2_11	BF2_12	BF2_13	BF2_14	BF2_15
	$(t - 22)^+$	$(22 - t)^+$	BF2_4 \times $(u^{(2)} - 0.2556)^+$	BF2_4 \times $(0.2556 - u^{(2)})^+$	BF2_4 \times $(u^{(1)} - 0.3524)^+$
	BF2_16	BF2_17	BF2_18		
	BF2_4 \times $(0.3524 - u^{(1)})^+$	$(1.5760 - u^{(1)})^+$	$(0.5331 - u^{(1)})^+$		

Table 5. MCC of different controllers

Controllers	Mean of MCC	Std. of MCC
MFRL-BI controller	116.4702	21.3797
MARS controller	1.6108×10^5	3.2499×10^3
MARS-EWMA controller	6.4886×10^4	1.9897×10^3
Process model-based controller	246.6173	221.1187
Theoretical optimal controller	93.0040	58.6906