Digital Chemical Engineering 14 (2025) 100206

Contents lists available at ScienceDirect

Digital Chemical Engineering

journal homepage: www.elsevier.com/locate/dche

Original article

Integration of on-line machine learning-based endpoint control and run-to-run control for an atomic layer etching process

Henrik Wang [a], Feiyang Ou [a], Julius Suherman [a], Gerassimos Orkoulas [c], Panagiotis D. Christofides [a,b,*]

[a] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, 90095-1592, USA
[b] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA
[c] Department of Chemical Engineering, Widener University, Chester, PA 19013, USA

ARTICLE INFO

Keywords:
Process control
Multiscale computational fluid dynamics modeling
Semiconductor manufacturing processes
Atomic layer etching

ABSTRACT

Control methods for Atomic Layer Etching (ALE) processes are constantly evolving due to the increasing level of precision needed to manufacture next-gen semiconductor devices. This work presents a novel, real-time Endpoint-based (EP) control approach for an Al_2O_3 ALE process in a discrete feed reactor. The proposed method dynamically adjusts the process time of both ALE half-cycles to ensure an optimal process outcome. The EP controller uses a machine learning-based transformer to take in variable-length, time-series pressure profiles to identify when the ALE process is complete. However, this model requires a large amount of process data to ensure that it will perform well even when under a variety of kinetic and pressure disturbances that mimic common issues in a real-world manufacturing environment. Thus, this work uses a multiscale modeling method that integrates a macroscopic Computational Fluid Dynamics (CFD) and a mesoscopic kinetic Monte Carlo (kMC) simulation to generate process data and test the proposed controllers. After testing the performance of the EP controller on individual runs, various combinations of ex-situ Run-to-Run (R2R) and EP controllers are examined in order to determine the strongest control strategy in a manufacturing environment. The final results show that the EP controller is highly accurate when trained on conditions that are representative of its implementation environment. Compared to traditional EWMA controllers, it has significantly fewer misprocesses, which enhances the overall control performance and efficiency of the ALE process.

1. Introduction

Electronic devices are an integral part of our modern society. They are used in everything from personal computers and mobile devices to smart vehicles and medical equipment. While there are many different types of electronic devices, all of them share a key feature: they are all made of densely connected integrated circuits, which are in turn composed of semiconductor transistors. Besides just the growing demand for the raw quantity of semiconductor chips, these chips are also becoming more and more compact (Singh et al., 2023; Voas et al., 2021). Moore's Law continues to hold true as the semiconductor chips used in these electronic devices shrink and become more sophisticated (Ajayan et al., 2021). Many new semiconductor device architectures, such as the gate-all-around structure, require the development of novel, high-precision manufacturing processes (Mukesh and Zhang, 2022).

A major factor in the industry's ability to continuously manufacture more and more sophisticated chips is the usage of advanced equipment and processes, including extreme ultraviolet (EUV) lithography (Wang et al., 2020). Additionally, other critical process steps necessitate 3D, nanoscale precision, contributing to the ongoing semiconductor supply shortages (Voas et al., 2021). Other processes capable of achieving this high level of precision are Atomic Layer Etching (ALE) and Atomic Layer Deposition (ALD). These two processes are similar to traditional etching and deposition processes, except they use half-reactions to etch or deposit a single atomic layer of material at a time (Kanarik et al., 2015; George, 2010). This is an extremely high precision process, which is required to manufacture modern semiconductor devices with tight process specifications (Ajayan et al., 2021; Tseng, 2022; Shauly, 2023). For the purposes of this work, the ALE of bulk Al_2O_3 is considered. This process consists of two steps: first, a precursor of hydrogen fluoride (HF) is used to prepare the Al_2O_3 surface by fluorinating it. Then, an etching reagent of trimethylaluminum (TMA) is used to etch the fluorinated surface (George, 2020; Kondati Natarajan and Elliott, 2018). This

* Corresponding author at: Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, 90095-1592, USA.
E-mail address: pdc@seas.ucla.edu (P.D. Christofides).

https://doi.org/10.1016/j.dche.2024.100206
Received 20 November 2024; Received in revised form 3 December 2024; Accepted 3 December 2024
Available online 10 December 2024

cycle removes a single layer of the Al_2O_3 substrate and can be repeated to remove multiple layers with a high level of precision.

Besides advanced manufacturing methods, advanced process control methods are also vital in the improvement of the manufacturing efficiency of semiconductor chips due to the high sensitivities of these processes. For older etch processes, such as plasma-enhanced etch, a popular, high-precision feedback control method that is still used today is endpoint (EP) detection (Roland et al., 1985; Wan et al., 2014). By measuring the ratio of gases in the outlet stream or the voltage of the plasma, the end time of the process can be automatically determined in real time. The advantages of EP-based process control are twofold: product quality is maximized, as an accurate EP control system ensures that the reaction is carried out to completion. Secondly, reagent wastage is minimized, as the process will terminate soon after the reaction is completed. This is also an in-situ feedback control system, which means that there are no ex-situ variables that require several iterations to tune (Moyne, 2015). Another popular control strategy is Run-to-Run (R2R) control, which is a widely used ex-situ control method and has been actively studied in the context of ALE processes (Yun et al., 2022b). R2R controllers operate on a batch-to-batch basis, adjusting process parameters between batches by using the measured output values from previous runs as feedback, unlike real-time feedback controllers, which make continuous adjustments. Rather, this batch-based approach enables consistent process optimization, helping to manage variations and disturbances that may arise during production.

While EP control systems are common for standard semiconductor etch processes, they have yet to be implemented for ALE processes. Most likely, this can be attributed to the high difficulty in relating a real-time measurable parameter to the completion of the process and the fact ALE processes focus on manufacturing cutting-edge devices that make traditional endpoint detection methods difficult to implement (Sun et al., 1994). In a previous work, the authors demonstrated that machine-learning methods can be used in conjunction with process simulations to train a transformer model that uses real-time wafer surface pressure data as an input to predict whether a wafer is fully processed as an output for the ALE process described above (Wang et al., 2024a). This work continues on by using this transformer as a basis for a real-time endpoint feedback controller.

To minimize the financial and time-based costs of real-world testing, multiscale computational fluid dynamics simulations are widely used to model semiconductor manufacturing processes, including plasma-enhanced chemical vapor deposition (Crose et al., 2018; Zhang et al., 2020), atomic layer deposition (Pan et al., 2014), and atomic layer etching (Yun et al., 2022b). In this work, a multiscale simulation approach is applied, which combines macroscopic CFD simulations of a discrete feed reactor with mesoscopic kinetic Monte Carlo simulation. This integrated method provides a detailed and accurate representation of the actual physical processes.

This work explores the integration of both a real-time endpoint (EP) feedback control system and an ex-situ run-to-run (R2R) controller to ensure the optimal operation of an atomic layer etching (ALE) process in an industrial manufacturing environment with process disturbances. First, Section 2 summarizes the ALE process and how the process is simulated. Next, Section 3 describes the formulation and implementation of the EP control system. Section 4 does the same for the R2R control system. Finally, Sections 5 and 6 analyze how effective various combinations of EP and R2R controllers are at mitigating the effects of a kinetic process disturbance across multiple process runs.

2. Process description

2.1. Atomic layer etching

Atomic layer etching (ALE) is a modern semiconductor fabrication technique that uses two alternating half-reactions to achieve atomic-level control over the etching process (Kanarik et al., 2015). Crucially, both of these half-reactions are self-limiting; this means that the reaction naturally slows down as it approaches completion. For a well-designed ALE process, overprocessing will only result in wasted reagents and will not misprocess the wafer.

The specific ALE process that this work examines is the etching of Al_2O_3 by the following reactions:

$$Al_2O_3\ (s) + 6HF\ (g) \rightarrow 2AlF_3\ (s) + 3H_2O\ (g) \quad \text{(Reaction A)}$$

$$2AlF_3\ (s) + 4Al(CH_3)_3\ (g) \rightarrow 6AlF(CH_3)_2\ (g) \quad \text{(Reaction B)}$$

In Reaction A, the gaseous hydrofluoric acid (HF) precursor fluorinates the Al_2O_3 surface. Then, in Reaction B, the gaseous trimethylaluminum (TMA) precursor etches away the fluorinated ALF_3 surface created in Reaction A, releasing a gaseous byproduct of dimethylaluminum fluoride (DMAF) (Kondati Natarajan and Elliott, 2018).

The two half-reactions can each be split into multiple elementary reactions (Yun et al., 2022a). Generally speaking, these elementary reactions can be sorted into one of two categories: adsorption/desorption reactions and nonadsorption reactions. The kinetic rate constant of the former is modeled by the Collision Theory equation shown in Eq. (1) and that of the latter is modeled by the Transition State Theory equation shown in Eq. (2).

$$k_{ads}(P_a, T) = \frac{\sigma_a P_a A_{site}}{Z_a \sqrt{2\pi m_a k_B T}} \tag{1}$$

where a is the adsorbate (HF, TMA), σ_a is the sticking coefficient between the adsorbate and the Al_2O_3 surface, P_a is the adsorbate's partial pressure on the wafer surface, A_{site} is the surface area of a single Al_2O_3 binding site, Z is the adsorbate's coordination number, m_a is the adsorbate's atomic mass, k_B is the Boltzmann constant, and T is the surface temperature of the wafer.

$$k_{nonads}(T) = \nu \exp\left(-\frac{E_A}{RT}\right),\ \nu = \frac{k_B T}{h} \tag{2}$$

where ν is the pre-exponential factor, h is Planck's constant, E_A is the activation energy, and R is the ideal gas constant.

With a kinetic rate constant equation for each elementary reaction, the overall reaction progression can be simulated with a kinetic Monte Carlo (kMC) algorithm. The algorithm takes place in a 300 × 300 grid that represents a larger reaction zone. Each point on the grid represents a reaction site, and the algorithm evaluates how the 90,000 reaction sites progress through time. The specifics of the kMC simulation method can be found in an earlier work by the authors, but a brief summary of the algorithm's implementation is given below (Wang et al., 2024b).

1. Randomly select a reaction site
2. Calculate $k_{tot} = \sum_{i-1}^{n} k_i$
3. Find j such that $\sum_{i=1}^{j-1} k_i \le \gamma_1 k_{tot} \le \sum_{i=1}^{j} k_i$
4. Calculate $\delta t_k = \frac{-\ln \gamma_2}{k_{tot} A}$

where n is the number of possible reactions for the selected reaction site, k_i is a reaction rate constant for a possible reaction, k_{tot} is a site-specific constant, j is the selected reaction, γ_1, γ_2 are randomly generated numbers that are evenly distributed within the range $(0, 1]$, A is the total number of active sites, and δt_k is the time elapsed for that reaction. Another way to interpret Steps 2 and 3 is that they are randomly selecting a reaction for the site selected in Step 1, with the probability of each possible reaction being weighted by the reaction rate constant of that reaction. And Step 4 is calculating how long the reaction takes to occur within the context of the entire grid, with δt_k ~1e−6 s. Thus, as the kMC algorithm is repeated, it will simulate the surface reaction progression at a very fine resolution.

2.2. Discrete feed reactor

The ALE process takes place inside a Discrete Feed Reactor (DFR), pictured in Fig. 1. This reactor operates at a constant temperature and

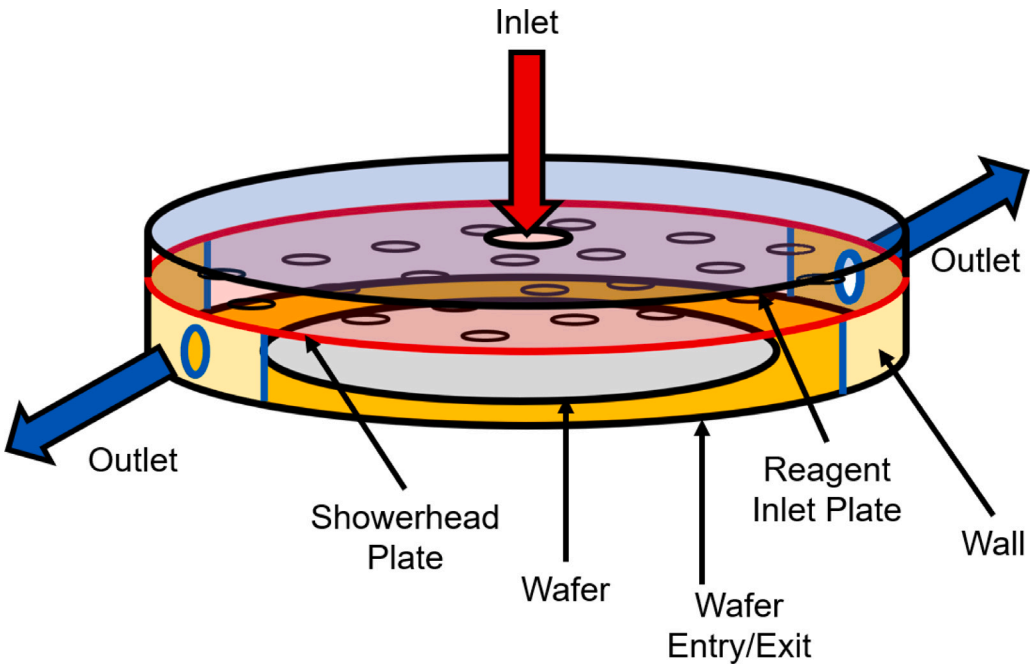


Fig. 1. 3D representation of the discrete feed reactor and its components.

pressure, and it allows reagents to continually flow into the reactor and byproducts to be flushed from the reactor. These characteristics all help ensure precise control over the process, which enables the process control techniques discussed later in this work.

The precursor for the reaction, along with a carrier gas of N_2, enters the reactor from the inlet at the top. These gases are dispersed by a showerhead plate, which improves the precursor distribution on the wafer at the bottom of the reactor. Finally, any unused precursor and byproducts evacuate the chamber through the outlet at the sides (Wang et al., 2024b).

The reactor is simulated through a Computational Fluid Dynamics (CFD) software called ANSYS Fluent, which calculates the unsteady-state pressure evolutions within the reactor. All the boundaries of the reactor are simulated as inert walls, except for three areas that are shown in Fig. 1: the inlet has a set mass flowrate as its boundary condition, the outlet has a set negative pressure differential as its boundary condition, and the wafer surface is modeled as a reaction zone where the mass source fluxes are calculated by the mesoscopic kMC simulation described in Section 2.1. Of note, the wafer surface is separated into 5 sections as shown in Fig. 2; this allows the collected process data to contain information regarding the reaction progression across the wafer itself, which is a crucial process metric.

Given these boundary conditions, the fluid volume of the reactor is then divided into a mesh so that the characteristic mass, momentum, and energy transport equations shown in Eqs. (3) to (5) can be solved numerically.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{v}\right) = S_m \tag{3}$$

$$\frac{\partial}{\partial t}\left(\rho \vec{v}\right) + \nabla \cdot \left(\rho \vec{v}\vec{v}\right) = -\nabla P + \nabla \cdot \left(\bar{\bar{\tau}}\right) + \rho \vec{g} + \vec{F} \tag{4}$$

$$\frac{\partial}{\partial t}\left(\rho E\right) + \nabla \left(\vec{v}\left(\rho E + P\right)\right) = -\nabla \left(\Sigma h_j \vec{J}_j\right) + S_h \tag{5}$$

where ρ is the gas-phase species density, $\vec{v}$ is the velocity of said species, S_m is the source generation and consumption flux of that species, P is the operating pressure of the reactor, $\bar{\bar{\tau}}$ is the normal two-rank stress tensor, $\vec{g}$ is the gravitational acceleration constant, $\vec{F}$ is the force acting on the system, E is the accumulated rate of system energy, S_h is the energy source generation or consumption, h_j is the sensible enthalpy flux of gas species j, and $\vec{J}_j$ is the mass diffusion flux of gas species j.

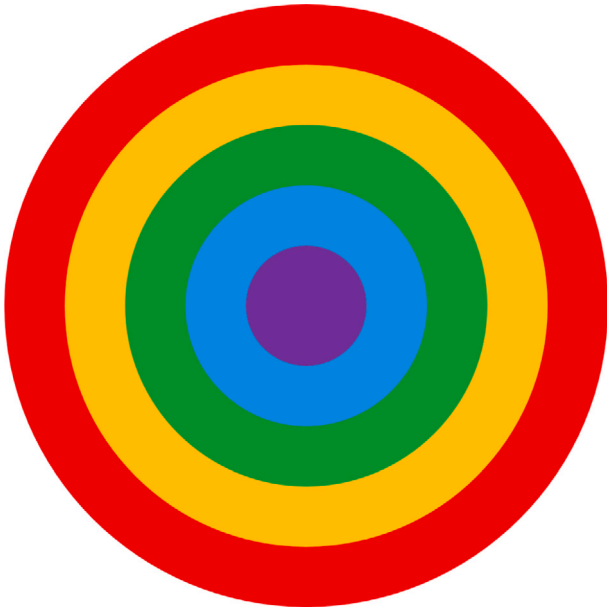


Fig. 2. Top-down view of the various reaction zones considered in the overall simulation.

The simulation is run with an integration timestep of 0.001 s, which is how often Eqs. (3) to (5) are solved. Additionally, it was found that the operating conditions listed in Table 1 result in both half-reactions being fully processed within 2 s (Yun et al., 2022a).

2.3. Multiscale model

Due to the self-limiting nature of the ALE half-reactions, the reaction rate is not constant; it gradually slows down as the reaction approaches completion. Thus, the mesoscopic kMC and macroscopic CFD simulations cannot be run independently as the former affects the latter, and vice versa. To link the two simulations and increase the accuracy of the overall simulation, a multiscale framework is used.

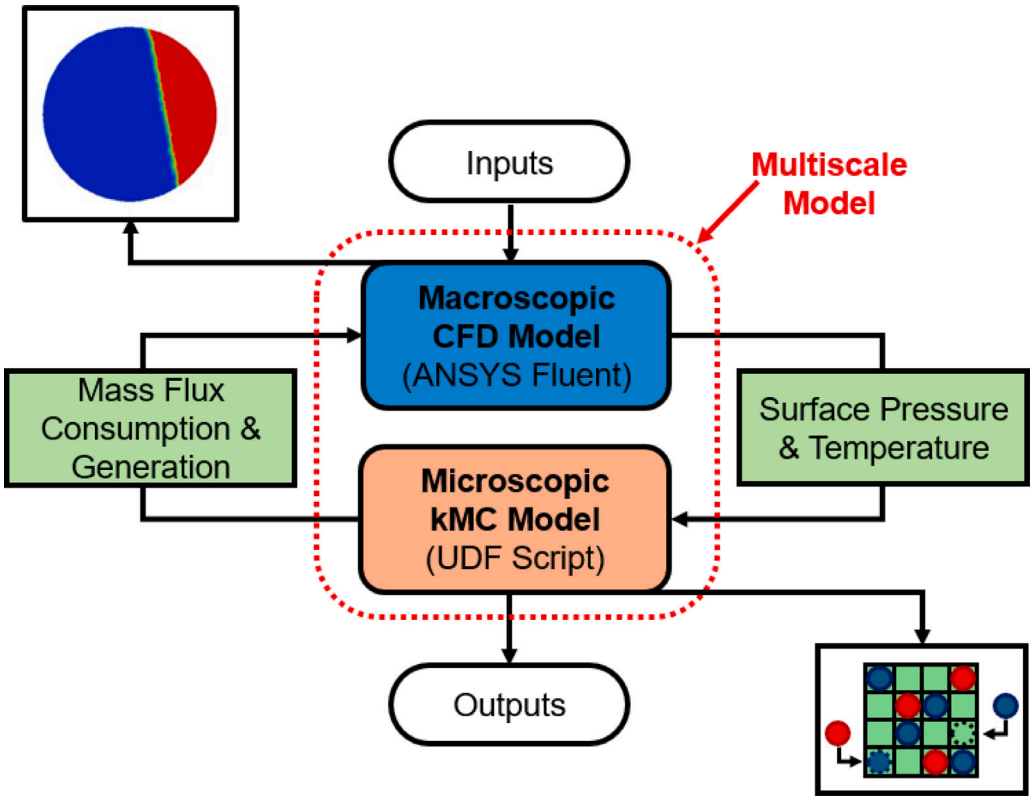


Fig. 3. Graphical representation of the information flow in the multiscale model.

Table 1
Operating conditions used for all the ALE simulations in this work.

Variable	Value
Operating temperature	573 K
Operating pressure	300 Pa
Inlet mass flowrate	1e−5 kg/s
Outlet pressure	−200 Pa
Reaction A HF inlet mole fraction	0.1
Reaction B TMA inlet mole fraction	0.5

The multiscale coupling method that conjoins the CFD and kMC simulations is shown in Fig. 3. The simulation starts by loading a steady-state simulation of the CFD model where the input is pure N_2; as there is no reagent, the kMC simulation is inactive. Then, the unsteady-state CFD model with the inlet parameters listed in Table 1 is ran for a single timestep of 0.001 s. Once the CFD simulation converges, the partial pressures and temperature at each wafer section are sent to their own kMC simulation, which makes for 5 independent kMC simulations. Each simulation then calculates the reaction rate constants and extents of reaction for the next 0.001 s that the CFD simulation is about to simulate. These extents of reaction are converted into mass source generation and consumption fluxes, S_m, and used in the CFD simulation of the following timestep. This constant flow of information persists throughout the entire multiscale simulation to ensure both an accurate macroscopic CFD simulation and an accurate mesoscopic kMC simulation.

Each simulation represents a full process of a half-reaction, producing two important time-series datasets: the absolute pressure on the wafer surface, and the reaction completion percentage, or coverage, on the wafer surface. Both datasets consist of 5 data points at each timestep due to the 5 reaction zones, and an example of the former is shown in Fig. 4(b). However, the latter is processed to yield the coverage mean and the coverage standard deviation, as seen in Fig. 4(a), because these two metrics directly determine if a wafer was successfully processed or not.

3. Endpoint controller methods

3.1. Endpoint controller description

Due to the sensitive nature of bleeding edge manufacturing techniques, process control methods are inherent assumptions. For example, proportional–integral (PI) controllers are commonly used to control the temperature and pressure of etching and deposition reactors (Ou et al., 2024). While PI controllers work well for on-line measurable process variables to drive them to the requested setpoints, they are not suitable when the primary control objective is to control a key process parameter that cannot be measured in real-time. For example, the reaction coverage is one such parameter that can only be measured once the processing step is complete. Rather, a common process control method for well-characterized processes such as plasma-enhanced etch is endpoint (EP) control (Roland et al., 1985). An EP controller uses some sort of signal, such as a voltage change, as a flag to end the desired process. However, for the ALE process examined in this work, there is no such indicator. Rather, this work uses a data-driven transformer model to act as the indicator common in other EP controllers.

The real-time endpoint detector developed in this work is based on a binary classifier model that uses real-time pressure data to determine whether the given ALE half-reaction has reached completion. If so, the controller ends the reaction and initiates chamber purging; otherwise, the process continues. The endpoint controller is activated 0.5 s into the process, as it is impossible for the reaction to finish any earlier than this. Once the termination signal is received, the endpoint controller is also deactivated as the reaction cannot be restarted. An optimal detector will ensure full wafer processing without wasting any precursors or time, enhancing cost-effectiveness and efficiency. However, creating such a detector is challenging, as its performance depends on accurately

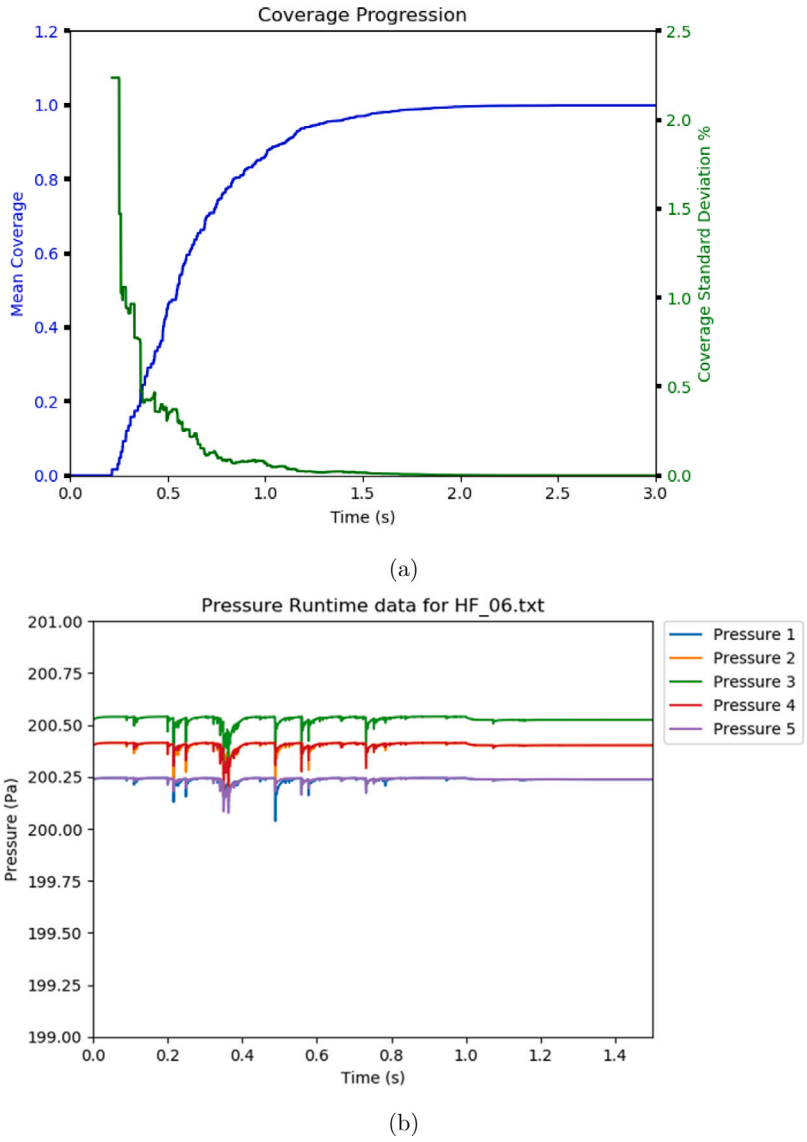


Fig. 4. Example of the coverage progression data (a) and the wafer surface pressure progression data (b).

correlating inputs to outputs. Due to inherent limitations of the process data, it may not achieve perfect control under all conditions, leading to trade-offs discussed in later sections.

3.2. Transformer model description

To train the transformer model for the classification task described above, a total of 240 process runs with unique run conditions were simulated. Each dataset can be defined by a naming scheme with three components: Reaction, Variable, and Number. There are two possible choices for Reaction: Reaction A or Reaction B. Variable describes the general category of process condition the run was simulated under. For example, "TST" affects the pre-exponential factor, or ν, of Eq. (2); "CT" refers to the sticking coefficient, or σ, of Eq. (1); "INV" has σ and ν in an inverse relationship; "PRESS" refers to the operating pressure inside the reactor. Finally, Number describes the exact process conditions according to Table 2. For example, "A_CT_23" refers to a process run for Reaction A with run parameters of $\sigma = 1.05,\ \nu = 1.0,\ P = 300$.

Finally, each process run is broken down into smaller input sequences, which is simply the pressure dataset fed to the transformer model. These input sequences represent the variable-length nature of the actual data that the EP controller is to be used on. Because each process run is simulated for 3.0 s, multiple input sequences can be extracted per process run for the purpose of model training. Specifically, the entire 3.0 s run can be separated into 26 input sequences of varying length that evenly span from 0.5 s to 3.0 s. Note that the first input sequence considered is for $t = 0.5$ s because it is considered impossible for either reaction to reach full coverage before then. Each of these variable-length input sequences have an associated output of whether the run is complete or not. In this manner, two transformer models are trained, one for each reaction, each on 1920 input sequences.

The ultimate goal of the EP controller is to take in input sequences of variable-length, time-series data and output a binary classification

Table 2
List of the ranges for each variable. Each one consists of 40 points, which evenly span the range of each process condition denoted in the columns. σ represents the sticking coefficient found in Eq. (1), ν represents the pre-exponential factor in Eq. (2), and P is the operating pressure.

Variable	σ range	ν range	P range
TST	[1.0,1.0]	[0.5,1.475]	[300,300]
MIX	[0.5,1.475]	[0.5,1.475]	[300,300]
INV	[1.5,0.525]	[0.5,1.475]	[300,300]
PRESS	[1.0,1.0]	[1.0,1.0]	[200,395]

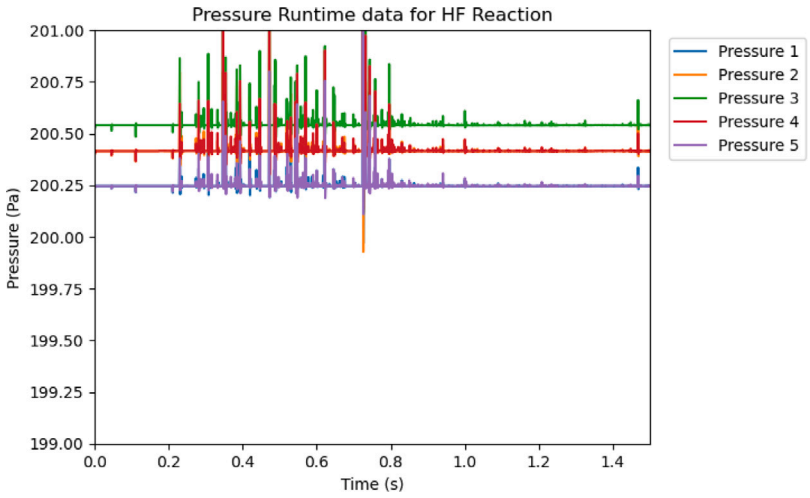


Fig. 5. Example of the raw wafer surface pressure input data. The large pressure spikes and sharp changes make it nonideal for training a transformer model.

of whether the process is complete. There are a variety of data-driven machine learning model architectures that can handle this task, such as recurrent neural networks (RNNs), long short term memory (LSTM) networks, and transformers. In a previous work, it was found that transformers perform the best when handling time-series input data (Wang et al., 2024a). While both RNNs and LSTM networks can handle time-series data through proper padding and masking operations, they still have their own challenges. RNNs cannot contextualize data across a long time period, "forgeting" about earlier data. On the other hand, LSTM networks can only process data in a sequential manner, making it impossible to form any long-term correlations. In comparison, the transformer's encoder/decoder structure allows it to retain information across long time periods and extract complex relationships. Transformer networks are also trained faster, as they have a parallel structure that naturally lends itself towards graphical processing units (GPUs). As RNNs and LSTM networks have sequential structures, they cannot take advantage of a GPU's powerful processing capabilities. Thus, the EP controller's process model is based on a data-driven transformer.

3.3. Transformer model training

Before discussing the transformer architecture, it is important to understand the data used to train it. The input data sequence is the variable-length, time-series pressure profile described in Section 2.3, and the output data is whether that pressure profile would result in a completed reaction. While the output data is stable, the raw input data has large, sudden spikes that are a natural result of the numerical solving process, as seen in Fig. 5. These spikes occur because the numerical solution method is trying to minimize the overall error of the entire reactor's pressure profile, not just that of the wafer surface pressure. While a transformer model can still be trained on such data, it is obviously nonideal as the noise will reduce the model's predictive ability.

To clean the wafer pressure input data, two steps are taken. First, all outliers are dropped. For this problem, the pressure is generally confined to 200 ± 1 Pa; thus, all pressure spikes/drops of more than 1 Pa were labeled as outliers and dropped from the data. As an example, after this step is taken, the raw input data shown in Fig. 5 becomes the data shown in Fig. 6(a). Second, the data is further smoothed by applying a rolling average. A window of 3 data points was used to avoid removing critical information, and the results can be seen in Fig. 6(b). Once the input data sequence is successfully cleaned, it can be used to train and test the transformer model.

The EP controller examined in this work uses a transformer model, and its encoder–decoder architecture is used to correlate the wafer surface pressure to process completion. The overall structure of the transformer is shown in Fig. 7. Specifically, each block has a multi-head self-attention mechanism (Vaswani et al., 2017) that relates each element of the input sequence data to each other element; this allows the model to capture process behavior that varies over time.

Table 3
Hyperparameters for the transformer model.

Model hyperparameter	Value
Input dimension	5
Embedding dimension	8
Number of heads	2
FNN neurons	64
Dropout ratio	0.1
Encoder layer number	2
Final FNN layer number	2
Final FNN neurons	64
Output dimension	1

The real-time pressure data is fed into the model through an input layer. Following that, it is embedded by a dense layer with dimension 8 that performs a positional encoding operation that provides information regarding how the time-series elements are ordered. Inside the encoder block, there is an internal FNN with a hidden layer of 64 neurons. Two encoder blocks with multi-head attentions are stacked together in a serial manner. The outputs of the last encoder block are combined through a global average pooling operation that makes the output vector have the same dimension regardless of the length of the input sequence. This output is then fed into the final FNN, which has 2 layers of 64 neurons each, and it outputs the final decision with a sigmoid activation function. The hyperparameters of the soft sensor transformer model are summarized in Table 3.

4. Run-to-Run controller methods

4.1. Run-to-Run controller description

The Run-to-Run (R2R) controller is an ex-situ controller, which means that it can only apply control actions after a process run is completed. Though it lacks real-time precision, it has access to higher-quality process data. For the ALE process, while the endpoint controller uses surface pressure as input, the R2R controller can use the final coverage as its input; this is the most important process metric, which allows the R2R controller to make finer adjustments.

Because the R2R controller only takes place after the process is completed, the final coverage that it uses as its input is actually the

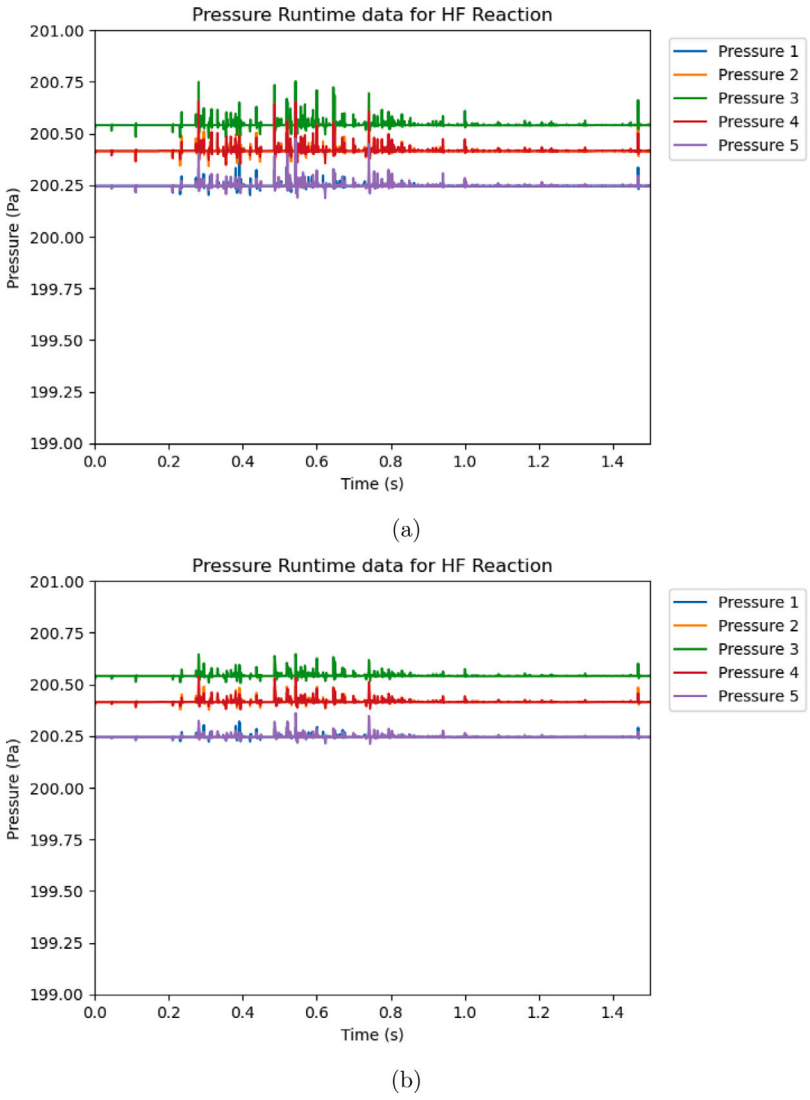


Fig. 6. Example of the wafer surface pressure input data after points more than 0.5 Pa have been removed (a) and a rolling average of 3 points is implemented (b).

product of Reaction A's coverage and Reaction B's coverage; this final coverage represents the percentage of the wafer that is fully processed (underwent both Reaction A and Reaction B). While there are many parameters that the R2R controller can adjust, one of the goals of this work is to evaluate its efficacy when used in conjunction with the EP controller described in Section 3. Thus, all the R2R controllers examined in this work only adjust the process time in response to the final coverage.

4.2. Run-to-Run process model

The R2R controller's control actions are generally based on a process model that describes the relationship between the process outcome and the control variable. Most such models assume a linear relationship between the outcome and the control variable, as this assumption is generally valid for small control actions. Such a linear model has the following general form:

$$y = a \cdot x + b$$

where y represents the target output, x is the control variable, a is the slope, and b is the intercept. Note that a and b are parameters that relate the behavior of the control variable to the output variable.

In this work, both the mean and the standard deviation (std.) of the final coverage must be controlled, as uniformity is a critical process metric in semiconductor manufacturing. Thus, there are two R2R controllers, one for each process metric. However, the calculation of the process time, which is the input, requires some nuance. Because the final coverage is a process metric that is indicative of whether *both* half-reactions were successfully completed, it is impossible for the R2R controller to independently adjust the two half-reactions' process times. Thus, the R2R controllers' process models use a process time offset term, δ, as a shared input that determines the process times for both half-reactions.

Another challenge for implementing R2R control of ALE processes is that both the mean and std. profiles exhibit highly nonlinear behavior, which poses a challenge for traditional linear control models. A poor process model can cause the model to deviate significantly from the actual physical process and result in poor control performance (Yun et al., 2022c). A solution to this issue involves applying nonlinear transformations to the input and output parameters so that the transformed input and output have a more linear relationship. Thus, the two models will both have the form shown below.

$$\psi_c = \alpha_c \cdot \chi_c + \beta_c, \ c = m, s \tag{6}$$

where c is a subscript that can be either m for the final coverage mean or s for the final coverage std., ψ_c is the transformed output metric c of the process, α_c is the slope for process metric c, χ_c is the transformed

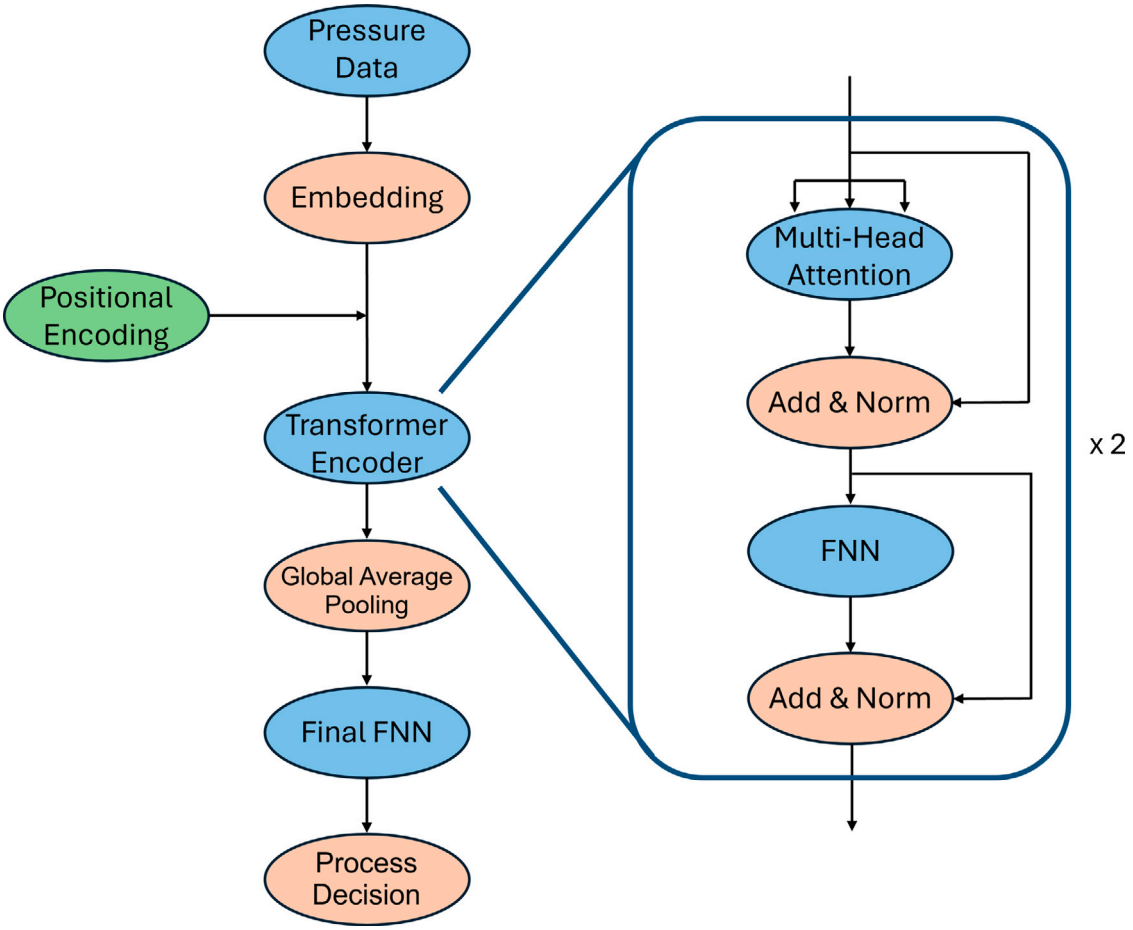


Fig. 7. Structure of the soft sensor transformer network.

input for the process metric c, and β_c is the intercept for process metric c.

This study uses a median effect function (Yun et al., 2022c) to transform the final coverage mean and a simple exponential function to transform the final coverage std. It was empirically determined that these transformed ouputs are linearly related to $\ln \delta_m$ and δ_s, respectively. The equations of the transformed terms are shown below:

$$\psi_m = \ln \frac{cov_m}{1 - cov_m}, \quad \chi_m = \ln(\delta_m + 0.75) \tag{7a}$$

$$\psi_s = \ln cov_s, \quad \chi_s = \delta_s + 0.75 \tag{7b}$$

where cov_m is the final coverage mean, cov_s is the final coverage std., δ_m is the process time offset for the final coverage mean equation, and δ_s is the process time offset for the final coverage std. equation. Both the nonlinear and linear fits for the final coverage mean are shown in Figs. 8(a) and 8(b), and the fits for the final coverage std. are shown in Figs. 8(c) and 8(d). The R^2 score of the fitting for the final coverage mean and final coverage std. are 0.997 and 0.96, respectively. These scores show that the fitted curve is highly accurate and can be used to build a process model.

The kink in Fig. 8(c) is a result of the difference in magnitude of the kinetic rate constants. For the HF reaction, there is one surface reaction that is 1000 times larger than the rest. Thus, when the substrate reaches that step, large parts of the wafer will effectively pause at the slow reaction while the remainder finishes reacting. This causes the standard deviation to momentarily increase before settling back down.

4.3. Estimated weight moving average method

Even with a highly accurate process model, inherent noise within the system or process disturbances that shift the process model may affect the R2R controller's ability to maintain the system at the desired setpoint. One widely used methodology to mitigate these challenges is the Exponentially Weighted Moving Average (EWMA) method, which updates the α_c, β_c tuning parameters in Eq. (6) by taking the exponentially weighted moving average of its past values. This effectively gives the controller information regarding its past error, which allows it to adjust and overcome the above challenges.

In real-world applications, the slope α_c is typically assumed to remain constant, even under various disturbances, while the intercept β_c is set to be adjustable (Ingolfsson and Sachs, 1993). Thus, the updating mechanism for the intercept β for process metric c is defined by the following equation (Del Castillo and Hurwitz, 1997):

$$\beta_{c,i+1} = (1 - \lambda)\beta_{c,i} + \lambda(\psi_{c,i} - \alpha_c \chi_{c,i}) \tag{8}$$

where $\beta_{c,i+1}$ is the updated intercept, $\beta_{c,i}$ is the intercept used in the previous run, $\psi_{c,i}$ is the transformed output of the previous run, α_c is the slope, and $\chi_{c,i}$ is the transformed input of the previous run. As all of the terms for the previous run are already known, $\beta_{c,i+a}$ can be easily solved for each controller as follows.

$$\beta_{m,i+1} = (1 - \lambda)\beta_{m,i} + \lambda \left(\ln \frac{cov_{m,i}}{1 - cov_{m,i}} - \alpha_m \ln(\delta_{m,i} + 0.75) \right)$$

$$\beta_{s,i+1} = (1 - \lambda)\beta_{s,i} + \lambda \left(\ln cov_{s,i} - \alpha_s(\delta_{s,i} + 0.75) \right)$$

Then, $\beta_{c,i+1}$ can be plugged into Eqs. (6) and (7) to find $\chi_{c,i}$ and subsequently $\delta_{c,i+1}$.

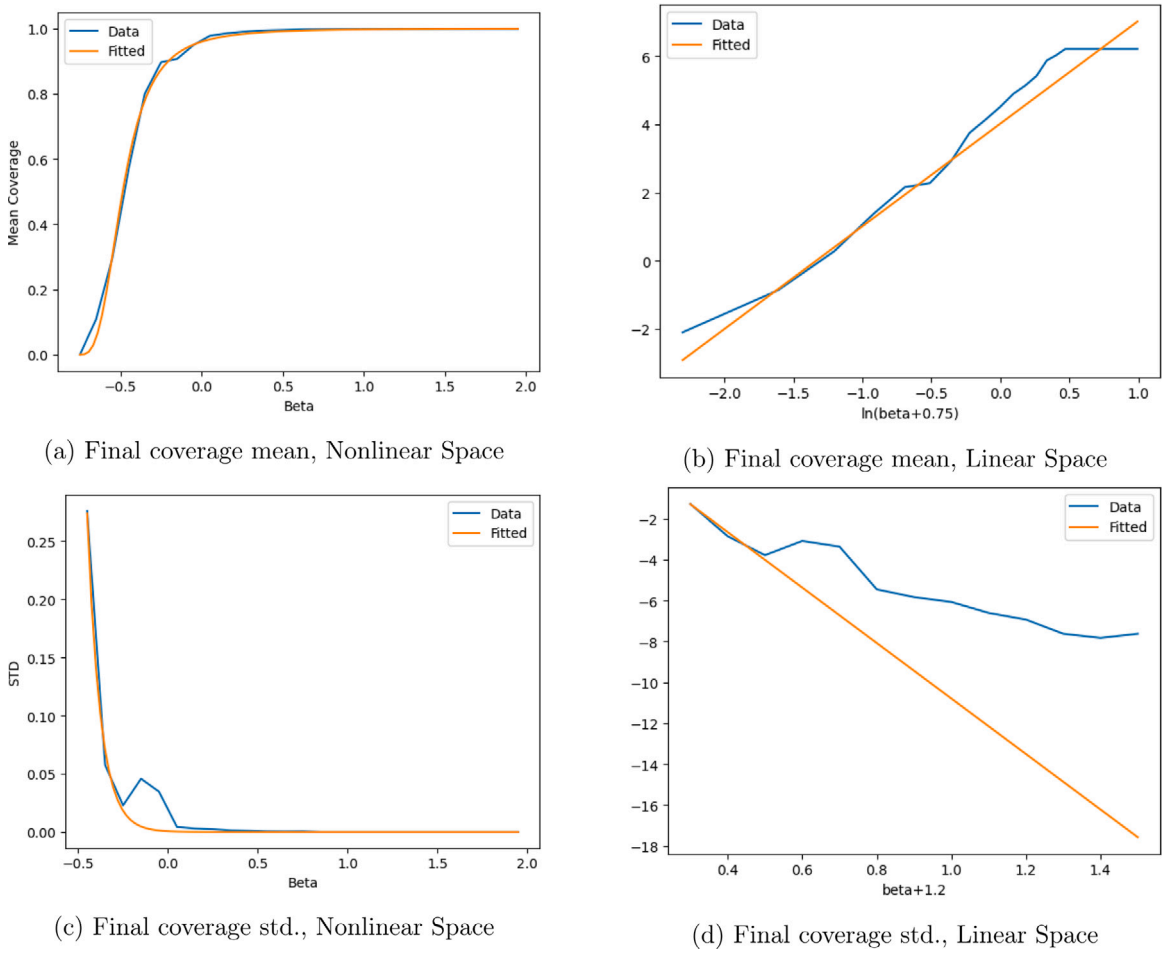


Fig. 8. Nonlinear fittings of the two coverage criteria vs. process time. The orange line is the predicted coverage, and the blue line is the actual coverage. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The exact solution for the final coverage mean ($c = m$) is shown below.

$$\ln \frac{cov_{m,d}}{1 - cov_{m,d}} = \alpha_m \gamma + \beta_{m,i+1}, \ \gamma = \ln(\delta_{m,i+1} + 0.75) \tag{9a}$$

$$\gamma = \left(\ln \frac{cov_{m,d}}{1 - cov_{m,d}} - \beta_{m,i+1} \right) / \alpha_m$$

$$\delta_{m,i+1} = e^{\gamma} - 0.75 \tag{9b}$$

where Eq. (9a) is the full form of Eq. (6) with the nonlinear m terms from Eq. (7) substituted in, Eq. (9b) is the final equation that the R2R controller uses to find the process time offset, $cov_{m,d}$ is the desired final coverage mean, γ is a placeholder variable as defined in Eq. (9a), and $\delta_{m,i+1}$ is the process time offset for the next run as determined by the final coverage mean R2R controller. Similarly, the exact solution for the final coverage std. ($c = s$) is as follows:

$$\ln cov_{s,d} = \alpha_s(\delta_{s,i+1} + 0.75) + \beta_{s,i+1} \tag{10a}$$

$$\delta_{s,i+1} = \frac{\ln cov_{s,d} - \beta_{s,i+1}}{\alpha_s} - 0.75 \tag{10b}$$

where Eq. (10a) is the full form of Eq. (6) with the nonlinear s terms from Eq. (7) substituted in, Eq. (10b) is the final equation that the R2R controller uses to find the process time offset, $cov_{s,d}$ is the desired final coverage std., and $\delta_{s,i+1}$ is the process time offset for the next run as determined by the final coverage mean R2R controller.

Once both controllers have found their respective δ_c, the final δ_f is set as the largest of the two to minimize underprocessing.

$$\delta_{f,i+1} = \max(\delta_{m,i+1}, \delta_{s,i+1})$$

where $\delta_{f,i+1}$ is the final process time offset that is used to determine the process times of the next run, $\delta_{m,i+1}$ is the process time offset found by the final coverage mean controller, and $\delta_{s,i+1}$ is the process time offset found by the final coverage std. controller. Then, Eq. (11) is used to find the process times for the next run.

$$t_{A,i+1} = t_{A,0} + \delta_{f,i+1} \tag{11a}$$

$$t_{B,i+1} = t_{B,0} + \delta_{f,i+1} \tag{11b}$$

where $t_{A,i+1}$ is the process time of the next HF reaction, $t_{A,0}$ is the initial process time for the HF reaction, $t_{B,i+1}$ is the process time of the next TMA reaction, and $t_{B,0}$ is the initial process time for the TMA reaction.

5. Endpoint controller results and analysis

5.1. Endpoint controller testing dataset

The EP controller's main objective is to accurately stop the process. To evaluate its ability to do so, the EP control system is tested on complete runs with different parameters from the training simulations; these runs are called testing runs to differentiate them from the training/validation data used to develop the transformer model. The main difference between the testing data and the training/validation data is that the testing data is examined in real time rather than separated into

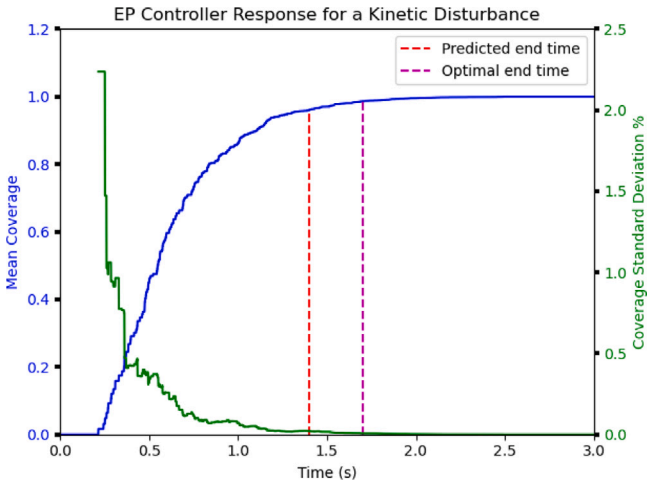


Fig. 9. The blue line represents the final coverage mean throughout the run, the green line is the final coverage std., the vertical dotted red line is t_{ep}, and the vertical dotted purple line is t_{tr}. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

multiple input sequences. Specifically, every 0.1 s, the EP controller receives the real-time wafer surface pressure data and makes a decision about whether to terminate or continue the process. The point at which the EP controller first decides to terminate the process is referred to as t_{ep}, and it represents the end time as determined by the EP controller. For the sake of analysis, each process run is simulated to 3.0 s regardless of the t_{ep}. This is so that the optimal end time, t_{tr}, can be determined and compared to t_{ep}.

An important note is that undershooting the predicted end time ($t_{ep} < t_{tr}$) is much worse than overshooting ($t_{ep} > t_{tr}$). When the model overshoots, it effectively makes the process run longer than what is necessary. While this wastes some reagent and time, the wafer is still successfully processed. Conversely, when the model undershoots, the wafer is underprocessed and will most likely have to be thrown away. Thus, undershooting is much worse than overshooting. For this reason, the error metric is weighted so that undershooting is more heavily penalized.

$$e = \begin{cases} |t_{ep} - t_{tr}| & \text{if } t_{ep} > t_{tr} \\ 2|t_{ep} - t_{tr}| & \text{otherwise} \end{cases} \tag{12}$$

where e is the error metric used to evaluate the EP system's performance, t_{ep} is the end time predicted by said model, and t_{tr} is the optimal end time. For example, in Fig. 9, the predicted end time is 1.4 s while the true end time is 1.7 s. Thus, the error associated with this run is 0.6.

Eq. (12) will be used to evaluate the EP controller's efficacy at mitigating various process disturbances in two ways: first is its robustness, or how changing the training data themselves affects the model performance. Then, the controller is evaluated on its consistency, which is how noise in the training data affects the EP control system.

5.2. Robustness

To understand how the training data affects the model performance when under various disturbances, multiple EP controller are first trained on the datasets described in Section 3.2 that have datasets with either pure kinetic, pure pressure, or both kinetic and pressure disturbances. Note that the kinetic disturbance is directly applied to the reaction rate k rather than to any individual constant, which was the case for the training data. Then, each EP controller is run on testing datasets with the same set of disturbances, and the error as described in Eq. (12) is calculated for each run. The results for Reaction A are shown in Table 4 and the results for Reaction B are shown in Table 5.

Table 4
Error comparison for Reaction A with a kinetic/pressure spread.

Training data	Validation data		
	Kinetic	Pressure	Both
Kinetic	0.467	1.060	0.915
Pressure	0.593	0.313	0.285
Both	0.613	0.413	0.440

Table 5
Error comparison for Reaction B with a kinetic/pressure spread.

Training data	Validation data		
	Kinetic	Pressure	Both
Kinetic	0.153	1.707	1.790
Pressure	0.733	0.164	0.250
Both	0.373	0.267	0.310

Generally speaking, the EP controller improves when it is trained on the types of disturbances that it is tested on. For example, when the system is trained on a kinetic disturbance from Reaction A, it performs better on the kinetic test ($e = 0.467$) compared to the pressure test ($e = 1.060$), and vice versa. Of note, while it is vital to train the model on the disturbances it is expected to face, the pressure disturbance has a much larger impact than the kinetic disturbance. For both Reaction A and Reaction B, when examining the column where the validation data has both kinetic and pressure disturbances, the model trained on only pressure disturbance data outperforms all others. This indicates that the pressure disturbance plays a much larger role in the model's ability to understand the process than the kinetic disturbance, to the point where a model trained on only the pressure disturbance outperforms a model trained on both, even when validated on a dataset with both disturbances. This shows that the optimal strategy for training an EP controller is to simply use training data similar to what the model will be used on. In industry, this is trivial as each process has many years of data (Zhang et al., 2021).

It is worth mentioning that the impact of the pressure disturbances may not come from it affecting the actual kinetics of the reactions; usually, kinetic disturbances have a greater effect on the actual reaction rates and t_{tr}. Rather, this phenomenon is most likely due to how the model uses the surface pressure of the wafer as its input. Additionally, the range of the surface pressure, ±5 Pa, is relatively narrow for each run compared to the scale of pressure disturbance, ±100 Pa. This means that a good model must necessarily adapt to the wide range of pressure disturbances. This explains the poor performance of the models trained on kinetic data that are tested on runs with pressure disturbances; the pressure disturbances shifts the wafer surface pressure far beyond the pressure range that the model is used to, which makes all the input sequence's values seem abnormal.

5.3. Consistency

It is also important to understand how well the EP controller can handle noise, which is referred to as its consistency. Each reaction was run ten times, each with a randomly selected ν and σ, which are the same variables used in Table 2, and no other process disturbances. The value of these two variables were selected from a Gaussian distribution centered around 1.0 with a standard deviation of 0.1 because this distribution follows the industrial standard of an average fail rate of 2%. The results of the twenty total runs are shown in Tables 6 and 7.

Reaction A has a higher average error ($e = 0.46$) compared to Reaction B ($e = 0.08$), which suggests that the model for Reaction A is not as accurate as the model for Reaction B. This result is corroborated by Tables 4 and 5, as the average error of a model when tested on the same dataset it was trained on is 0.407 for Reaction A and 0.209 for Reaction B. Both average errors are comparable to the ones in Tables 6

Table 6
Evaluation of Reaction A with no disturbances and moderate noise. Average error is 0.47.

No.	ν	σ	t_{tr}	t_{ep}	Error
Run 1	0.939	1.020	1.3	1.3	0.00
Run 2	1.120	0.969	1.0	1.5	0.50
Run 3	0.922	1.030	1.0	1.5	0.50
Run 4	0.930	0.999	1.1	2.1	1.00
Run 5	0.947	0.923	1.1	1.7	0.60
Run 6	1.010	0.910	1.0	1.5	0.50
Run 7	1.160	0.938	1.8	2.0	0.20
Run 8	1.150	0.819	1.1	1.5	0.40
Run 9	0.934	1.140	1.2	2.1	0.90
Run 10	0.888	0.882	1.2	1.3	0.10

Table 7
Evaluation of Reaction B with no disturbances and moderate noise. Average error is 0.08.

No.	ν	σ	t_{tr}	t_{ep}	Error
Run 1	1.080	0.916	1.2	1.2	0.00
Run 2	0.979	0.856	1.4	1.3	0.20
Run 3	1.090	1.050	1.2	1.3	0.10
Run 4	1.070	1.070	1.2	1.3	0.10
Run 5	0.877	1.000	1.4	1.6	0.20
Run 6	0.954	1.230	1.3	1.3	0.00
Run 7	1.060	1.080	1.2	1.3	0.10
Run 8	0.964	0.913	1.3	1.3	0.00
Run 9	0.946	0.995	1.3	1.4	0.10
Run 10	1.170	1.090	1.2	1.2	0.00

and 7; thus, the EP controller for Reaction A performs worse than that of Reaction B. But regardless of the model's baseline prediction ability, the results for Reaction B show that the EP method is resistant to noise when the model is trained on datasets where those parameters are varied.

6. Run-to-Run controller results and analysis

6.1. Run-to-Run environment

The EP controller shows great potential for mitigating common disturbances in an industrial manufacturing environment. However, the controller may not be possible to implement for all processes, as the controller for Reaction A was not as consistent as that of Reaction B even though they were trained on similar datasets. Thus, there is still motivation to design more complex control schemes for processes that are hard for data-driven machine learning models to learn.

A common control system for ALE processes is ex-situ run-to-run (R2R) control, which adjusts process parameters after directly measuring the process outcome after the process is complete. Thus, this section examines how EP and R2R control systems can work together under various frameworks. Specifically, all the control systems will be tested under the same conditions; the process will experience a sudden shift where all the kinetic activity is lowered by 40%. This is a relatively large in comparison to the shifts examined for the pure EP controller, and any failure to account for this shift will result in scrapped material and wasted time. The ideal combination of control systems will quickly adjust the process time so that the final coverage criteria are met with minimal over-processing.

The control systems will be evaluated on how many runs they take to return to the target final coverage criteria and how much overprocessing occurs. Because there are two final coverage criteria, a total of three error calculations are made:

$$\epsilon_m = \sum_{i=1}^{L} c_{m,i} \cdot \frac{|cov_{m,i} - 0.96|}{L}, \quad \text{where } c_{m,i} = \begin{cases} 1 & \text{if } cov_{m,i} \geq 0.96 \\ 2 & \text{if } cov_{m,i} < 0.96 \end{cases} \tag{13a}$$

$$\epsilon_s = \sum_{i=1}^{L} c_{s,i} \cdot \frac{|cov_{s,i} - 0.02|}{L}, \quad \text{where } c_{s,i} = \begin{cases} 1 & \text{if } cov_{s,i} \leq 0.02 \\ 2 & \text{if } cov_{s,i} > 0.02 \end{cases} \tag{13b}$$

$$\epsilon_t = \sum_{i=1}^{L} 0.01 \cdot \frac{t_{A,i} + t_{B,i}}{L} \tag{13c}$$

where ϵ_m is the error term associated with the final coverage mean criterion, $c_{m,i}$ is a scaling factor based on if the final coverage mean criterion was met for run i, $cov_{m,i}$ is the final coverage mean of run i, L is the number of process runs, ϵ_s is the error term associated with the final coverage std. criterion, $c_{s,i}$ is a scaling factor based on if the final coverage std. criterion was met for run i, $cov_{s,i}$ is the final coverage std. of run i, ϵ_t is the error term associated with overprocessing, $t_{A,i}$ is the process time for the HF reaction for run i, and $t_{B,i}$ is the process time for the TMA reaction for run i. The final, comprehensive error term is found by simply summing up the three error terms of Eq. (13) as shown below:

$$\epsilon_f = \epsilon_m + \epsilon_s + \epsilon_t \tag{14}$$

where ϵ_f is a comprehensive error term used to evaluate the various control systems presented in this work.

Of these control systems, first is a standalone EWMA-R2R system that is meant to establish a baseline expectation for the following control systems. Second, the EP system will be evaluated on its own to better compare real-time and ex-situ controllers. Finally, combined systems will be examined: an EP+SCC system, and an EP+EWMA system. "SCC" stands for Standard Case Corrector, which is a newly developed R2R ex-situ controller. All of these systems will be compared through two metrics. First and foremost, they will be evaluated on how many wafers are scrapped, or thrown away; this occurs when the final coverage criteria is insufficient. If two control systems have the same number of scrapped wafers, then they will be evaluated on how much time they waste on overprocessing. With these metrics, the best control system among the aforementioned four systems can be determined.

6.2. Pure EWMA controller

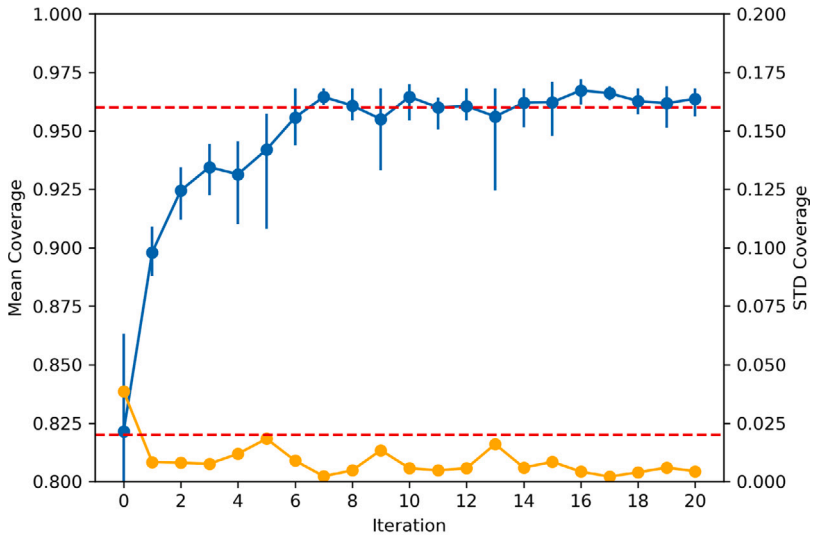
The EWMA-R2R controller processes coverage data by first converting the measured mean and standard deviation to nonlinear forms, then applying the EWMA algorithm to determine the process time. While Eqs. (9) and (10) describe how the EWMA-R2R controller updates the run parameters after each run, they do not describe the initial starting point of the process system. In this work, the initial process times, $t_{A,0} = 0.75$ s and $t_{B,0} = 1.05$ s, are set to achieve a final coverage whose mean is over 96% and std. is less than 2% when there are no disturbances. Although the ALE process has two half-reactions, only the final etch per cycle (EPC) is measurable, preventing the R2R controller from adjusting each half-reaction individually. Instead, both t_A and t_B are updated simultaneously using a process time offset δ, adjusted by the controller as shown in Eq. (11) and restated here.

$$t_A = t_{A,0} + \delta$$
$$t_B = t_{B,0} + \delta$$

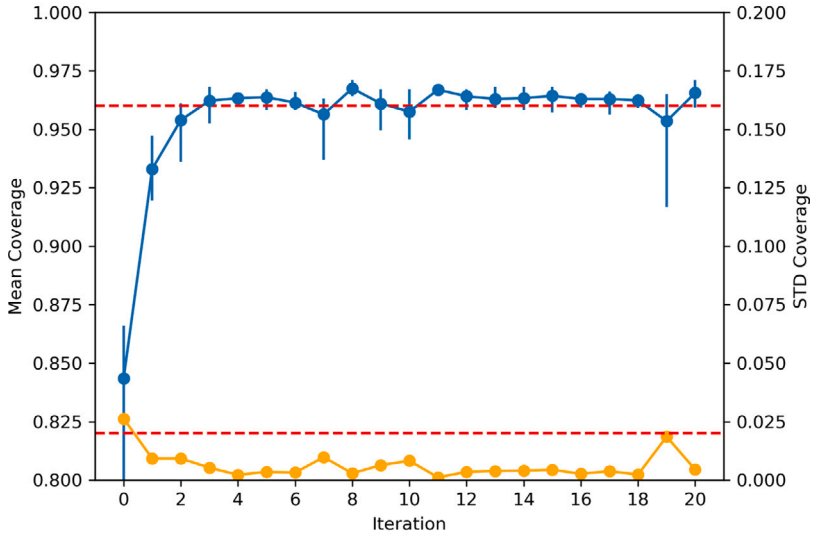
where t_A and t_B are the process times for the HF and TMA reactions, respectively, $t_{A,0}$ and $t_{B,0}$ are their initial process times, and δ is the time offset determined by the R2R controller.

The performance of the EWMA-R2R controller depends on two factors: the accuracy of the process model used in Eq. (6), and the value of λ in Eq. (8), which is a tunable factor that determines how much weight is given to recent measurements. A larger λ value makes the controller more responsive to recent batches, enabling quicker, more aggressive corrections, while a smaller λ emphasizes historical data, leading to a more conservative, stable response. Although aggressive settings can correct shifts faster, conservative settings reduce the risk of oscillations or divergence. This study uses λ values of 0.7 for aggressive and 0.3 for conservative control.

Performance results for two pure EWMA-R2R controllers with $\lambda = 0.3, 0.7$ are shown in Figs. 10(a) and 10(b). These plots show that



(a) Pure EWMA, $\lambda = 0.3$; 8 scrapped wafers, $\epsilon_f = 0.074$



(b) Pure EWMA, $\lambda = 0.7$; 5 scrapped wafers, $\epsilon_f = 0.061$

Fig. 10. Control results of the EWMA-R2R controller. The blue line is the mean coverage, the orange line is the std. coverage, the high red dashed line is the mean coverage target, and the low red dashed line is the std. coverage target. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the EWMA-R2R controller can drive the system back to the desired setpoint even with a large process shift, where all kinetic rates are reduced by 40%. The more aggressive $\lambda = 0.7$ controller in Fig. 10(b) results in fewer scrapped wafers compared to the conservative $\lambda = 0.3$ controller in Fig. 10(a) because the aggressive EWMA controller reaches the final coverage criteria faster, indicating that it is better suited for this ALE process. This is also supported by the ϵ_f criterion, as the more aggressive EWMA has a lower ϵ_f of 0.061 compared to the conservative controller's 0.074. However, even though the aggressive EWMA controller performs well, it still has a key limitation in its inefficiency at the initial stages; several wafers are misprocessed before the process fully corrects.

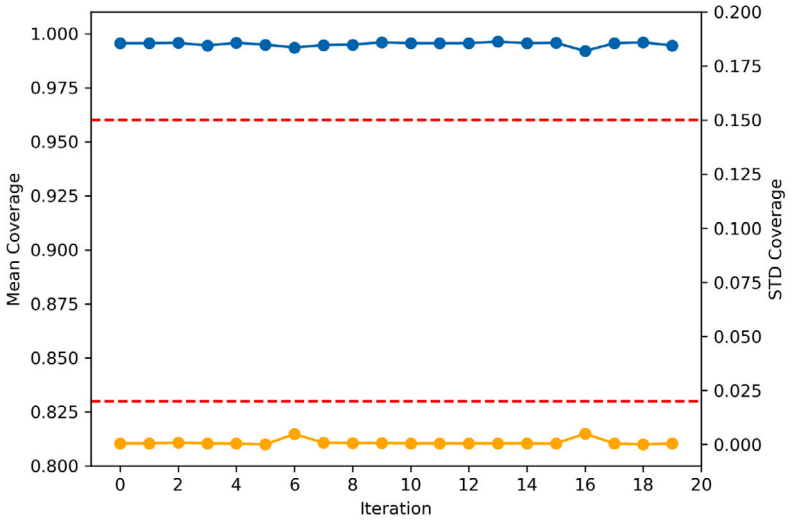
6.3. Pure endpoint controller

The EP controller is a real-time controller, which means that there is no parameter updating or changing in the controller in between each run. Thus, the process times are represented by the following equations:

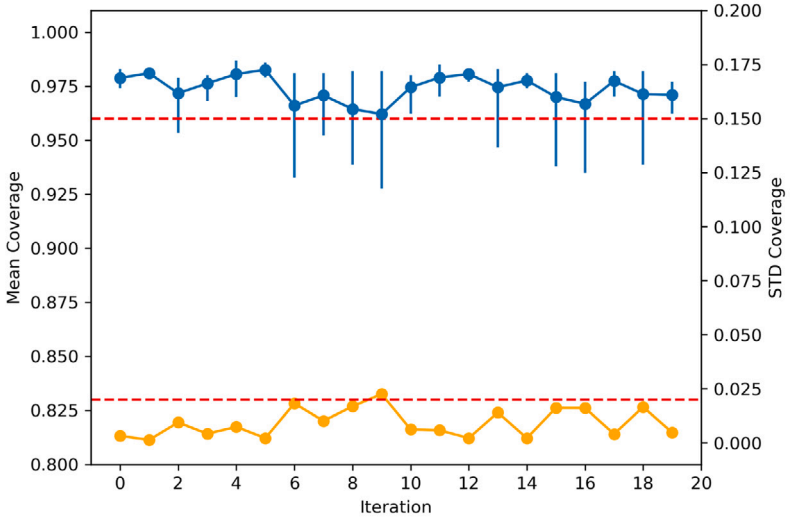
$$t_A = EP_A$$
$$t_B = EP_B$$

where t_A and t_B are the process times for the HF and TMA reactions, respectively, and EP_A and EP_B are the process times as determined by the EP controller for the HF and TMA reactions, respectively.

Rather, all the differences in between the runs stem from the stochastic nature of the multiscale simulations and the Transformer model in the EP controller. The EP controller used for this simulation is the same one described in Section 3, but the threshold of the final sigmoid function is adjusted; this represents the transformer's confidence that the process has terminated. Two thresholds were considered: a conservative EP controller with a sigmoid threshold of 0.96 for Reaction A and 0.99 for Reaction B, and an aggressive EP controller with a threshold of 0.5 for both reactions. Fig. 11(a) shows the performance of the conservative EP controller and Fig. 11(b) that of the aggressive controller.disturbances.



(a) Conservative EP Controller, Threshold = 0.95 (HF)/0.99 (TMA); 0 scrapped wafers, $\epsilon_f = 0.099$



(b) Aggressive EP Controller, Threshold = 0.5 (HF)/0.5 (TMA); 0 scrapped wafers, $\epsilon_f = 0.056$

Fig. 11. Control results for pure EP controllers. The lines are the same as in Figs. 10(a) and 10(b).

The conservative and aggressive pure EP controllers in Figs. 11(a) and 11(b) both resulted in 0 scrapped wafers. However, the conservative EP controller consistently resulted in large amounts of overprocessing, causing its ϵ_f of 0.099 to be above even that of the pure EWMA controllers. In comparison, the aggressive EP controller has the lowest ϵ_f of all four controller systems at 0.056. This means that it is best suited for handling sudden, large process shifts, and its performance here highlights its ability to reduce precursor usage and improve manufacturing efficiency.

Both EP controllers have some overprocessing, but they are still able to prevent all misprocesses, even when the disturbance first appeared. In comparison, the pure EWMA-R2R controller requires several batches to adjust to the disturbance before there are no more misprocesses. Despite its advantages, the pure EP controller also has other weaknesses; it relies on time-series pressure profiles, which are influenced by the stochastic nature of surface reactions and noise in the measuring equipment. This causes it to predict different endpoint times even when the process conditions are identical. Thus, combining EP and R2R controllers can make up for their individual shortcomings. For the combined control systems shown next, the aggressive EP controller is used as it performed better than the conservative EP controller.

6.4. Standard case corrector

As both the EP and R2R controllers use the process time as their control variable, there are many ways to combine the two systems. As mentioned earlier, while the EP controller does prevent misprocessing, it can be volatile when used for multiple runs. Thus, one combined EP+R2R method is to use a Standard Case Corrector (SCC) Controller. This controller assumes that the process time required to adjust one set of coverage criteria to another is the same regardless of if there are any disturbances.

As illustrated in Fig. 12, after each run, the SCC controller uses the final coverage mean and std. progression curves of a standard case without any disturbances to find two key values for each curve: t_0, which is the time needed to reach the target set point, and t_m, which is the time needed to reach the measured output of the most recent run. The controller then calculates the sum $t_d = t_m - t_0$. Like $\beta_{f,i+1}$

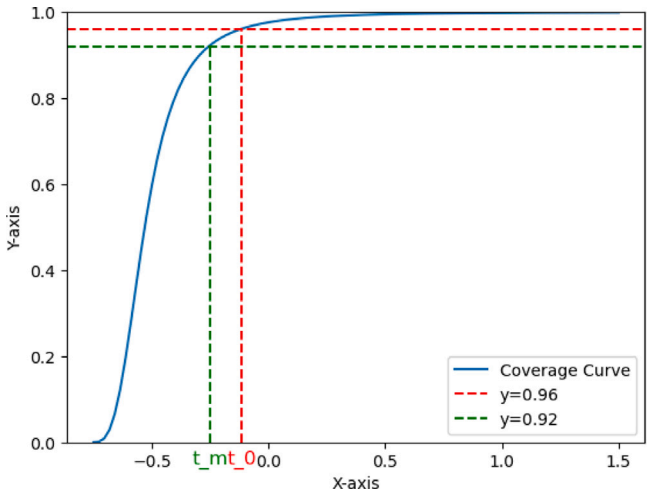


Fig. 12. Representation of how the SCC controller calculated t_d for the final coverage mean.

in Eq. (11), the final t_d is the maximum between the one calculated from the final coverage mean curve and the one calculated from the final coverage std. curve. Additionally, the fitted progression curves of Fig. 8 are used as the standard case examples. This is because the smoother curves prevent the controller from forming any offsets. If the true final coverage criteria progression curves were used, their non-monotonic nature could cause issues in the SCC controller.

When a disturbance is sensed, the system is "reset" with a run that only uses the endpoint controller. This effectively sets $t_{A,0} = EP_A$ and $t_{B,0} = EP_B$. Then, the process time continues to be updated by modifying Eq. (11) as follows:

$$t_d = t_m - t_0 \tag{17a}$$

$$t_{A,i+1} = t_{A,i} + t_d \tag{17b}$$

$$t_{B,i+1} = t_{B,i} + t_d \tag{17c}$$

where t_d is the value found in Fig. 12, $t_{A,i}$ and $t_{B,i}$ are the most recent process times for reactions A and B, and $t_{A,i+1}$ and $t_{B,i+1}$ are the next set of process times for reactions A and B. Note that, unlike the EWMA-R2R controller, the SCC-R2R controller does not rely on a linear model. Thus, it avoids using any nonlinear transformations, making it easier to implement.

The result of combining the EP and SCC controllers is shown in Fig. 13. Run 0 is where the disturbance is first introduced, resulting in a misprocess. Run 1 is the pure EP run, and Runs 2 and onwards are the SCC-controlled runs as defined by Eq. (17). Even though the EP+SCC control system results in a scrapped wafer in Run 0, it successfully brings the process back within control by Run 1 and continues to tightly control the final coverage criteria around the target set point due to the unstable nature of the process. Thus, the EP+SCC is insufficient to control the process.

While the EP+SCC controller may be worse than the pure EP controller when it comes to sudden, unexpected process disturbances, that is not always the case. In manufacturing environments, it is common practice to have qualifying test runs after a major equipment cleaning in order to detect any process shifts. In this scenario, Runs 0 and 1 of the EP+SCC controller can be thought of as qualifying runs that do not count towards the misprocessing rate. In that case, the ϵ_f should only span Runs 2–20. When evaluated in this context, the EP+SCC controller's ϵ_f becomes 0.047, which is lower than the aggressive EP controller's ϵ_f of 0.056. Thus, in the right circumstances, the EP+SCC controller can outperform the pure EP controller.

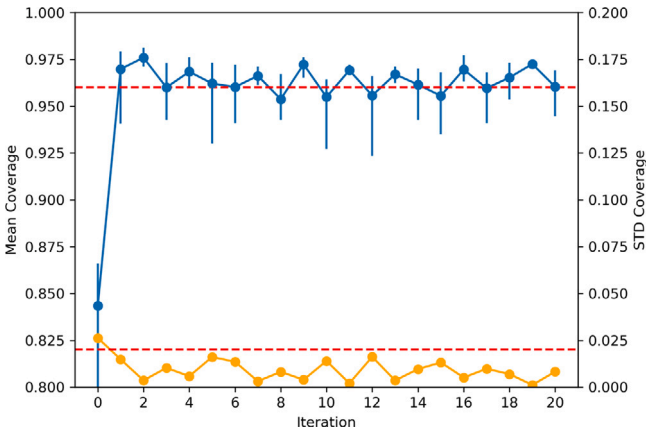


Fig. 13. Control results for the EP+SCC controller; 6 scrapped wafers, $\epsilon_f = 0.057$.

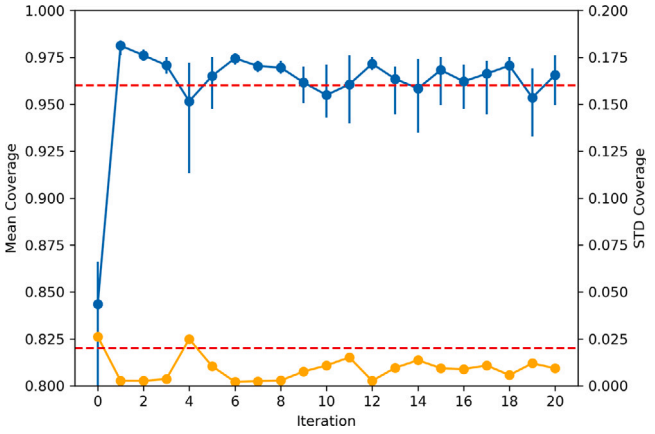


Fig. 14. Control results for the EP+EWMA controller; 4 scrapped wafers, $\epsilon_f = 0.060$.

6.5. EWMA and EP controller

While the EP+SCC controller has a better performance than either of the controllers on their own in a controlled manufacturing environment, many of the batch runs did not meet the final coverage criteria. Thus, the combined controller can still be improved upon, and other combinations of R2R and EP controllers must be explored. In this section, the EWMA-R2R controller in Section 6.2 is combined with the EP controller in Section 6.3 to create another hybrid approach.

While both the EWMA-R2R and EP controllers use the process time as their control variable, there is no issue as the combined controller functions similarly to the SCC controller; once a disturbance is detected, the process times are reset after a run that only has an EP controller. Once that run is completed, the control system reverts to the classical EWMA-R2R equations of Eqs. (9) to (11). The results of combining a EWMA-R2R controller with $\lambda = 0.7$ and an aggressive EP controller are shown in Fig. 14. Run 0 is where the disturbance is first introduced, resulting in a misprocess. Run 1 is the pure EP run, and Runs 2 and onwards are controlled by the same EWMA-R2R controller discussed earlier in Section 6.2.

Like the EP+SCC controller, the EP+EWMA controller system performs worse than the pure EP system when it comes to reacting to a sudden process shift. However, when evaluated in a well-controlled manufacturing environment, its ϵ_f becomes = 0.049. While this is still larger than the EP+SCC controller's ϵ_f of 0.047, which indicates that the EP+SCC controller is ideal, it is nonetheless higher than the pure EP controller's ϵ_f of 0.056.

Table 8
Summary of the R2R evaluation criteria.

	ϵ_f, Runs 0–20	ϵ_f, Runs 2–20
EWMA, $\lambda = 0.3$	0.074	–
EWMA, $\lambda = 0.7$	0.061	–
EP, aggressive	0.056	–
EP, conservative	0.099	–
EP+SCC	0.057	0.047
EP+EWMA	0.060	0.049

The final results of all the different control systems are summarized in Table 8. For manufacturing environments that are poorly controlled with a potential for unexpected process shifts, the pure EP control system is the best option as it has the lowest ϵ_f when all the runs are considered. However, for a manufacturing environment that is well controlled and where process shifts only occur at known events, the combined systems are better. Of the two, the EP+SCC controller has a slightly lower ϵ_f when only Runs 2–20 are considered. Thus, the pure EP controller is best suited for poorly controlled manufacturing environments, and the EP+SCC controller is best suited for well controlled environments.

7. Conclusions

This work presents an integrated control strategy combining a real-time endpoint (EP) feedback controller, based on a transformer machine learning architecture, with an ex-situ Run-to-Run (R2R) controller for an Al_2O_3 atomic layer etching (ALE) process. The EP controller was trained with simulated process data and then tested on a different set of simulated process data for two different metrics: robustness and consistency. This novel controller enables real-time detection of process indicators for the ALE process, effectively handling kinetic and pressure disturbances. For R2R control, this work introduced a new EWMMA strategy involving nonlinear transformations to create a linear relationship and a novel standard case corrector (SCC) simplified the overall implementation by eliminating the need for complex nonlinear modeling.

Various combinations of EP and R2R controllers were applied to the ALE process under a severe negative kinetic disturbance. Two manufacturing environments were considered: a poorly controlled environment where process shifts occur randomly and without warning, and one where process shifts are expected (e.g., after maintenance is done on the etching tool). For the former case, the pure EP controller performed best, as it had the lowest error metric, $\epsilon_f = 0.056$, when considering all runs, including the initial disturbance run. But for the latter case, only Runs 2–20 are used to calculate ϵ_f as Runs 0 and 1 are considered to be qualifying test runs used to adjust the process parameters. In that case, the EP+SCC controller performed the best as it had the best performance of $\epsilon_f = 0.047$ at maintaining the system at the desired setpoint after the kinetic disturbance was implemented. This hybrid approach leverages the strengths of both controllers, offering a significantly improved performance over the traditional pure EWMA and pure EP controllers.

The controllers developed in this work only use the surface wafer pressure in their machine-learning models, but in reality, the amount and variety of process data that is available in a high-volume manufacturing environment is many times larger. Whether it be incorporating multiple data streams, aggregating these large datasets, or using bleeding edge machine-learning models, industrial manufacturing represents a space with abundant opportunities for innovation.

CRediT authorship contribution statement

Henrik Wang: Writing – original draft, Methodology, Investigation, Conceptualization. **Feiyang Ou:** Writing – original draft, Methodology, Investigation, Conceptualization. **Julius Suherman:** Writing – original draft, Methodology, Investigation, Conceptualization. **Gerassimos Orkoulas:** Writing – review & editing, Supervision, Methodology. **Panagiotis D. Christofides:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Financial support from the National Science Foundation, United States is gratefully acknowledged. This work used computational and storage services associated with the Hoffman2 Shared Cluster provided by UCLA Institute for Digital Research and Education's Research Technology Group.

References

Ajayan, J., Nirmal, D., Tayal, S., Bhattacharya, S., Arivazhagan, L., Fletcher, A.A., Murugapandiyan, P., Ajitha, D., 2021. Nanosheet field effect transistors-A next generation device to keep Moore's law alive: An intensive study. Microelectron. J. 114, 105141.

Crose, M., Zhang, W., Tran, A., Christofides, P.D., 2018. Multiscale three-dimensional CFD modeling for PECVD of amorphous silicon thin films. Comput. Chem. Eng. 113, 184–195.

Del Castillo, E., Hurwitz, A.M., 1997. Run-to-run process control: Literature review and extensions. J. Qual. Technol. 29 (2), 184–196.

George, S.M., 2010. Atomic layer deposition: An overview. Chem. Rev. 110, 111–131.

George, S.M., 2020. Mechanisms of thermal atomic layer etching. Acc. Chem. Res. 53 (6), 1151–1160.

Ingolfsson, A., Sachs, E., 1993. Stability and sensitivity of an EWMA controller. J. Qual. Technol. 25 (4), 271–287.

Kanarik, K.J., Lill, T., Hudson, E.A., Sriraman, S., Tan, S., Marks, J., Vahedi, V., Gottscho, R.A., 2015. Overview of atomic layer etching in the semiconductor industry. J. Vac. Sci. Technol. A 33 (2), 020802.

Kondati Natarajan, S., Elliott, S.D., 2018. Modeling the chemical mechanism of the thermal atomic layer etch of aluminum oxide: A density functional theory study of reactions during HF exposure. Chem. Mater. 30 (17), 5912–5922.

Moyne, J., 2015. Run-to-run control in semiconductor manufacturing. In: Baillieul, J., Samad, T. (Eds.), Encyclopedia of Systems and Control. Springer London, London, pp. 1248–1254.

Mukesh, S., Zhang, J., 2022. A review of the gate-all-around nanosheet FET process opportunities. Electronics 11 (21).

Ou, F., Abdullah, F., Wang, H., Tom, M., Orkoulas, G., Christofides, P.D., 2024. Sparse identification modeling and predictive control of wafer temperature in an atomic layer etching reactor. Chem. Eng. Res. Des. 202, 1–11.

Pan, D., Li, T., Chien Jen, T., Yuan, C., 2014. Numerical modeling of carrier gas flow in atomic layer deposition vacuum reactor: A comparative study of lattice Boltzmann models. J. Vac. Sci. Technol. A 32, 01A110.

Roland, J.P., Marcoux, P.J., Ray, G.W., Rankin, G.H., 1985. Endpoint detection in plasma etching. J. Vac. Sci. Technol. A 3, 631–636.

Shauly, E.N., 2023. Design rules in a semiconductor foundry. In: Eitan N. Shauly (Ed.), Design Rules in a Semiconductor Foundry. Jenny Stanford Publishing, Singapore.

Singh, M., Sargent, J.F., Sutter, K.M., 2023. In: Singh, Manpreet., Sargent, Jr., John F., Sutter, Karen M. (Eds.), Semiconductors and the Semiconductor Industry, Library of Congress public ed. Congressional Research Service, Washington, D.C.

Sun, H., Patel, V., Singh, B., Ng, C., Whittaker, E., 1994. Sensitive plasma etching endpoint detection using tunable diode laser absorption spectroscopy. Appl. Phys. Lett. 64, 2779–2781.

Tseng, S.-H., 2022. CMOS MEMS design and fabrication platform. Front. Mech. Eng. 8, 894484.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Vol. 30, Curran Associates, Inc., Long Beach, CA, USA, pp. 1–11.

Voas, J., Kshetri, N., DeFranco, J.F., 2021. Scarcity and global insecurity: The semiconductor shortage. IT Prof. 23, 78–82.

Wan, J., McLoone, S., English, P., O'Hara, P., Johnston, A., 2014. Predictive maintenance for improved sustainability — An ion beam etch endpoint detection system use case. In: Intelligent Computing in Smart Grid and Electrical Vehicles. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 147–156.

Wang, H., Ou, F., Suherman, J., Tom, M., Orkoulas, G., Christofides, P.D., 2024a. Data-driven machine learning predictor model for optimal operation of a thermal atomic layer etching reactor. Ind. Eng. Chem. Res. 63 (45), 19693–19706.

Wang, H., Tom, M., Ou, F., Orkoulas, G., Christofides, P.D., 2024b. Multiscale computational fluid dynamics modeling of an area-selective atomic layer deposition process using a discrete feed method. Digit. Chem. Eng. 10, 100140.

Wang, C.-P., Tsai, Y.-P., Lin, B.J., Liang, Z.-Y., Chiu, P.-W., Shih, J.-R., Lin, C.J., King, Y.-C., 2020. On-wafer FinFET-based EUV/eBeam detector arrays for advanced lithography processes. IEEE Trans. Electron Devices 67, 2406–2413.

Yun, S., Tom, M., Luo, J., Orkoulas, G., Christofides, P., 2022a. Microscopic and data-driven modeling and operation of thermal atomic layer etching of aluminum oxide thin films. Chem. Eng. Res. Des. 177, 96–107.

Yun, S., Tom, M., Ou, F., Orkoulas, G., Christofides, P.D., 2022b. Multiscale computational fluid dynamics modeling of thermal atomic layer etching: Application to chamber configuration design. Comput. Chem. Eng. 161, 107757.

Yun, S., Tom, M., Ou, F., Orkoulas, G., Christofides, P.D., 2022c. Multivariable run-to-run control of thermal atomic layer etching of aluminum oxide thin films. Chem. Eng. Res. Des. 182, 1–12.

Zhang, Y., Ding, Y., Christofides, P.D., 2020. Multiscale computational fluid dynamics modeling and reactor design of plasma-enhanced atomic layer deposition. Comput. Chem. Eng. 142, 107066.

Zhang, C., Yella, J., Huang, Y., Qian, X., Petrov, S., Rzhetsky, A., Bom, S., 2021. Soft sensing transformer: Hundreds of sensors are worth a single word. In: 2021 IEEE International Conference on Big Data (Big Data). pp. 1999–2008.