I&EC research
Industrial & Engineering Chemistry Research

pubs.acs.org/IECR
Article

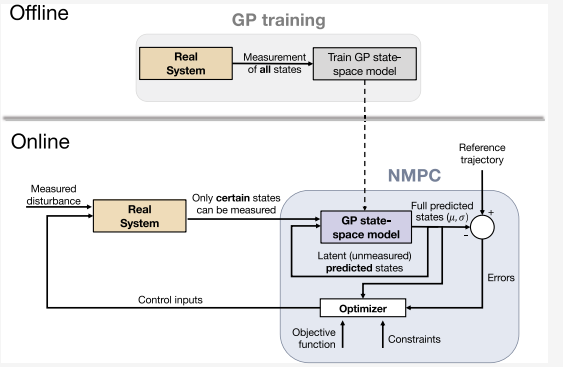
Gaussian Process Nonlinear Model Predictive Control for Online Partially Observable Systems: An Application to Froth Flotation

Published as part of Industrial & Engineering Chemistry Research special issue "Celebrating Undergraduate Research in Chemical Engineering 2024".

Yicong Wang, Ehecatl Antonio del Río Chanona,* and Paulina Quintanilla*

Cite This: *Ind. Eng. Chem. Res.* 2025, 64, 13307−13322
Read Online

ABSTRACT: This paper presents a nonlinear model predictive control (NMPC) framework employing Gaussian processes (GPs) for application in froth flotation processes under online partial observability. Froth flotation, a critical process in mineral processing, involves complex, nonlinear dynamics and unmeasured variables, making traditional control methods challenging to implement. In this work, we build a data-driven control strategy by training GP-based state-space models on data generated from a physics-based model. These GP models are then integrated into an NMPC architecture where only a subset of the process states is observable online. The GP-MPC controller accounts for uncertainty and disturbances by predicting both the mean and variance of system dynamics, enabling robust optimization over a finite horizon. Results demonstrate a 20% reduction in the concentration of valuable minerals in the tailings compared to traditional control methods, while achieving consistent operation throughout the process. The framework effectively handles noise and disturbances, with the partially observable GP-MPC achieving consistent set point tracking within a 5% error margin, and reduces deviations from target values by approximately 15%. The framework also meets real-time constraints, making it well suited for complex, data-driven process control applications.



1. INTRODUCTION

Model predictive control (MPC) is an advanced control strategy widely applied in various industries for its ability to handle multivariate systems and explicitly manage process constraints.[1] MPC uses a dynamic model of the process to optimize a sequence of control actions over a time horizon at each sampling step.[2−4] Although linear MPC has matured into a well-established strategy, many industrial processes exhibit significant nonlinear behavior, motivating the development of nonlinear MPC (NMPC) strategies.[5] With the advancement of optimization algorithms capable of handling nonconvex problems, NMPC has gained increasing popularity, especially in chemical engineering applications.

One of the critical factors influencing the performance of MPC is the accuracy of the model used to represent the system.[6] As a result, the development of accurate models often constitutes a substantial portion of the effort required for MPC implementation. For example, previous studies have estimated that up to 80% of the commissioning effort is devoted to model development.[7] Traditional approaches typically rely on mechanistic or first-principles models, which, while accurate, are often too complex for real-time application and expensive to develop.[8] This has prompted interest in data-driven methods, where models are derived from process data instead of detailed physical principles. Data-driven approaches can learn the dynamics of the system directly from process data. Among these, Gaussian processes (GPs) have emerged as a powerful tool to learn system dynamics from limited data, providing both predictive accuracy and a measure of uncertainty.[9,10] Their flexibility and minimal reliance on prior process knowledge make GPs well-suited for various applications, including state estimation, reinforcement learning,[11,12] and MPC.[2,13,14] The ability of GPs to provide probabilistic predictions is particularly valuable in MPC, where it enables efficient exploration of unknown regions and avoidance of unsafe areas with high uncertainty, thus improving both learning efficiency and robustness.[15]

GPs have been successfully integrated into MPC frameworks to model nonlinear dynamic systems (e.g.[16−18]), with recent

Received: February 14, 2025
Revised: June 10, 2025
Accepted: June 11, 2025
Published: June 23, 2025

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

works demonstrating their efficacy in areas such as disturbance rejection[19,20] and fault tolerance.[21,22] However, most GP-based MPC applications assume that the system is fully observable,[23,24] where all relevant state variables can be measured directly. In many real-world processes, this assumption does not hold. One such example is the froth flotation process, the most extensive mineral separation process used in the mining industry.[25] Froth flotation effectively separates valuable minerals from gangue, but its efficiency depends on the control of various process variables such as froth depth, air flow rate, and tailings composition.[8,26,27] While MPC has been successfully applied in other chemical industries, its implementation in the mineral processing sector, particularly for froth flotation, remains limited.[28,29] This is largely due to the difficulties in accurately modeling the complex dynamics of flotation processes, which restrict the application of traditional model-based control approaches.[8,30]

1.1. Related Work. Previous works on modeling for flotation control have focused on the use of simplified kinetics models (e.g.[31−33]), which are limited in handling the process's inherent complexities, including nonlinear dynamics, disturbances, and measurement noise. For example, Maldonado et al. (2007)[31] proposed dynamic programming approaches for rougher flotation that have shown promise, but their high computational complexity restricts real-time applications. Putz and Cipriano (2015)[32] developed a hybrid control system that integrates continuous and discrete system variables through logical rules, combined with MPC, to address these challenges. However, they rely heavily on static models and do not adequately capture the inherent uncertainties of the process. Some other studies, such as Brooks and Koorts. (2017),[34] have focused more on regulatory control than process optimization. Quintanilla et al. (2023)[26,35] presented an economic MPC strategy, achieving an 8−22% improvement in recovery while maintaining a concentrate grade. Other recent advancements in flotation modeling and control are discussed in.[30,36−39] Although these advances enable performance improvements, there remains a need to fully account for process uncertainties and disturbances, especially in the presence of unmeasured process variables.

In this work, we develop a nonlinear GP-MPC framework to optimize an economic objective while accounting for system uncertainties and disturbances. Our approach addresses the limitations of previous studies by incorporating stochastic process modeling via GPs, which explicitly accounts for prediction uncertainty. This enables our GP-MPC to dynamically balance exploration (reducing uncertainty) and exploitation (optimizing control actions) based on real-time measurements, which is crucial for maintaining robust control in the presence of noise and disturbances. By introducing a partially observable framework, our approach can predict latent variables, such as mineral masses in flotation processes, that are not directly measurable online and can typically only be measured offline, significantly improving control effectiveness in scenarios where conventional models struggle with incomplete state information.[31,32] We demonstrate that the partially observable GP-MPC remains robust under varying levels of noise and disturbances, making it highly applicable in industrial settings where measurement variability and process uncertainty are critical factors. This integration of stochastic modeling, uncertainty quantification, and partial observability renders this GP-MPC framework a suitable and reliable solution for flotation processes. The framework is designed to be applicable to other systems.

The rest of this paper is organized as follows: Section 2 introduces the theoretical background, including GP and MPC. Section 3 details the proposed methodology, presenting the development of the GP state-space model, the partially observable GP state-space model, and the GP-MPC framework. Section 4 describes the mineral froth flotation process, covering the flotation model, problem setup, and implementation details, including initial data set generation. Section 5 presents the results and discussion, evaluating the performance of both full-state and partially observable GP models, followed by an in-depth analysis of the GP-MPC approach in different scenarios, including trajectory tracking, optimization with the objective function, and disturbance handling. Finally, Section 6 summarizes the key findings and discusses potential future research directions.

2. PRELIMINARIES

2.1. Gaussian Processes. In this section, we give an introduction to GP regression. For a more general overview, see.[40] GP regression aims to describe an unknown function $f:\mathbf{R}^{n\_x} \rightarrow \mathbf{R}$ using noisy observations, $y = f(\boldsymbol{x}) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is Gaussian distributed measurement noise with zero mean and (possibly unknown) variance σ_ε^2.

GPs consider a distribution over functions, and they can be seen as a generalization of multivariate Gaussian distributions

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)) \tag{1}$$

where the mean function $m(\cdot)$ can be interpreted as the "average" shape of the function, while the covariance function $k(\cdot,\cdot)$ describes the correlations between function values.

Our focus herein is on a constant mean function, $m(\boldsymbol{x}) := 0$, alongside the squared-exponential (SE) covariance function[40]

$$k(\boldsymbol{x}, \boldsymbol{x}') := \sigma_n^2 \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^{\mathrm{T}} \boldsymbol{\Lambda} (\boldsymbol{x} - \boldsymbol{x}')\right) \tag{2}$$

where σ_n^2 is the covariance magnitude; and $\Lambda := \mathrm{diag}(\lambda_1 \cdots \lambda_{n\_x})$ is a scaling matrix. The basis for the choice of the SE covariance function is the assumption that the inferred function f is smooth and stationary.

Maximum likelihood estimation is commonly applied to infer the unknown hyperparameters $\boldsymbol{\Psi} := [\sigma_n \sigma_\varepsilon \lambda_1 : \cdots : \lambda_{nx}]^{\mathrm{T}}$, including σ_ε in case the measurement noise variance is also unknown. Consider n_d (noisy) function evaluations, denoted by $\boldsymbol{y} := [y^{(1)} : \cdots : y^{(n\_d)}]^{\mathrm{T}} \in \boldsymbol{R}^{n\_d}$, with corresponding inputs collected in the matrix $\mathbf{X} := [\boldsymbol{x}^{(1)} : \cdots : \boldsymbol{x}^{(n_d)}] \in \boldsymbol{R}^{n_x \times n_d}$. The log-likelihood of the observed data, ignoring constant terms, is given by

$$\mathcal{L}(\boldsymbol{\Psi}) := -\frac{1}{2}\ln(|\mathbf{K}(\mathbf{X})|) - \frac{1}{2}\boldsymbol{y}^{\mathrm{T}} \mathbf{K}(\mathbf{X})^{-1} \boldsymbol{y} \tag{3}$$

with $\mathrm{K}_{ij}(\mathbf{X}) := k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}) + \sigma_\varepsilon^2 \delta_{ij}$ for each pair $(i,j) \in 1 \cdots n_d^2$; and the Kronecker delta function δ_{ij}.

The predicted distribution of $f(\boldsymbol{x})$ at an arbitrary input point $\boldsymbol{x}$, given the input−output data $(\boldsymbol{X}, \boldsymbol{y})$ and the maximum-likelihood estimates of $\boldsymbol{\Psi}$, follows the Gaussian distribution.

$$f(\boldsymbol{x})|\mathbf{X}, \boldsymbol{y} \sim \mathcal{N}(\mu_f(\boldsymbol{x}), \sigma_f^2(\boldsymbol{x})) \tag{4}$$

with

$$\mu_f(\boldsymbol{x}) := \boldsymbol{r}(\boldsymbol{x}, \mathbf{X}) \mathbf{K}(\mathbf{X})^{-1} \boldsymbol{y}$$

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

$$\sigma_f^2(\mathbf{x}) := \sigma_n^2 - \boldsymbol{r}(\boldsymbol{x}, \mathbf{X})\mathbf{K}(\mathbf{X})^{-1}\boldsymbol{r}(\boldsymbol{x}, \mathbf{X})^{\mathrm{T}}$$

$$\text{and } \boldsymbol{r}(\boldsymbol{x}, \mathbf{X}) := [k(\boldsymbol{x}, \boldsymbol{x}^{(1)}) : \cdots : k(\boldsymbol{x}, \boldsymbol{x}^{(n_d)})]$$

The mean μ_f in this context is the prediction made by the GP at $\boldsymbol{x}$, while the variance σ_f^2 provides a measure of the uncertainty around this predictor.

2.2. Model Predictive Control. In this section, we provide an introduction to model predictive control (MPC) in the context of discrete time. MPC is an optimization-based control strategy that predicts future system behavior over a finite time horizon using a model of the system. Originally developed in the 1970 s for chemical process control, MPC has since been widely adopted in various industries,[41] including automotive, aerospace, robotics, and chemical processing, due to its ability to handle multivariable control problems and constraints effectively. In chemical processes, such as froth flotation, MPC is particularly valuable for optimizing operations where precise control is necessary to achieve desired outcomes under complex, nonlinear dynamics.[42]

Discrete-time MPC is particularly useful for real-world applications. It allows the controller to update information at discrete intervals rather than continuously, making it well-suited for digital implementations where control strategies are executed at specific time steps. This approach ensures compatibility with digital processors and systems that operate on sampled data, while also reducing computational complexity.[43] At each discrete time step k, MPC solves an optimization problem to determine the optimal sequence of control actions that minimize an objective (or cost) function while satisfying system dynamics and constraints.

In this work, we extend the traditional MPC strategy by integrating GP models, which allow for explicit handling of uncertainty in nonlinear systems. The GP-MPC framework predicts future states and incorporates the uncertainty of these predictions into the optimization problem, enhancing robustness and performance under uncertain conditions. The cost function for this optimization is typically defined as

$$\min_{\boldsymbol{u}_{0:N-1}} J(\boldsymbol{x}_{1:N}, \boldsymbol{u}_{0:N-1}) = \sum_{k=0}^{N-1} (w_1\|\boldsymbol{x}_k - \boldsymbol{x}_{SP}\|^2 + w_2\|\Delta\boldsymbol{u}_k\|^2) \tag{5}$$

subject to

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k),\ k = 0, \ldots, N-1 \tag{6}$$

$$\boldsymbol{u}_{\min} \le \boldsymbol{u}_k \le \boldsymbol{u}_{\max},\ k = 0, \ldots, N-1 \tag{7}$$

$$\boldsymbol{x}_{\min} \le \boldsymbol{x}_k \le \boldsymbol{x}_{\max},\ k = 0, \ldots, N-1 \tag{8}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}(0) \quad \text{(initial condition)} \tag{9}$$

where $\boldsymbol{x}_k \in \mathrm{R}^{n-x}$ represents the state vector, $\boldsymbol{u}_k \in \mathrm{R}^{n-u}$ is the control input, and $\mathbf{f}$: $\mathbf{X} \times \mathbf{U} \to \mathbf{X}$ can ideally model the uncertainties and disturbances within the system. The weights w_1, w_2, and w_3 are used to balance the contributions of each term in the objective function: w_1 corresponds to the state tracking error, w_2 penalizes the control effort, and w_3 accounts for the predicted uncertainty in the system. Since MPC is a model-based strategy, developing an accurate model is essential. In the context of nonlinear systems, the state evolution is not simply described by a linear relationship but rather by a nonlinear function $\boldsymbol{f}(\cdot)$, as expressed in eq 6.

The solution of the optimization problem is a sequence of control inputs over the prediction horizon N, represented as a column vector

$$\boldsymbol{u}_{k:k+N-1} = [\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \ldots, \boldsymbol{u}_{k+N-1}]^{\mathrm{T}} \tag{10}$$

Only the first control input $\boldsymbol{u}_k$ is implemented, after which the optimization process is repeated at the next time step with updated state information, following the receding horizon strategy. This approach allows MPC to continuously adapt to changes in the system and external disturbances, making it robust to model inaccuracies and uncertainties.

3. METHODOLOGY

3.1. Gaussian Processes State-Space Model. We consider the nonlinear discrete-time system as described in Section 2.2

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_d(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{\varepsilon}_k \tag{11}$$

$\boldsymbol{x}_k \in \mathrm{R}^{n-x}$ denotes the state vector, $\boldsymbol{u}_k \in \mathrm{R}^{n-u}$ is the control input, ϵ_k represents process noise, and k marking the time-step where $k \in \boldsymbol{N}^0$ (meaning that k can take positive integer real numbers including zero). The function $\mathbf{f}_d$:$\mathrm{R}^{n-x} \times \mathrm{R}^{n-u} \to \mathrm{R}^{n-x}$ is differentiable but unknown. The GP model provides probabilistic predictions for the mean and variance of the state at the next time step, incorporating these probability estimates into control decisions.

We assume that we can fully measure the current state; hence, we have a data set of states and control measurements $\{\boldsymbol{x}^{(i)}, \boldsymbol{u}^{(i)}\}$ with its corresponding consecutive state noisy measurement $\boldsymbol{y}_k^{(i)} = \boldsymbol{f}_d(\boldsymbol{x}_k^{(i)},\boldsymbol{u}_k^{(i)})+\varepsilon_k^{(i)}$, where the noise term is modeled as $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathrm{diag}([\sigma_{d,x_1}^2, \sigma_{d,x_2}^2, \ldots, \sigma_{d,x_{n_x}}^2]))$.

We can then state our data set as

$$\mathcal{D} = \{\{\boldsymbol{x}^{(i)}, \boldsymbol{u}^{(i)}\}, \boldsymbol{y}^{(i)}\}_{i=1}^{n_d} \tag{12}$$

For ease of notation, we group the variables as $\boldsymbol{\xi}_k = [\boldsymbol{x}_k^T, \boldsymbol{u}_k^T]^T$ such that $\boldsymbol{\xi} \in \mathcal{X} \subseteq \boldsymbol{R}^{n_x+n_u}$. To model dynamical systems we will use a GP as a state-space model of the dynamical system. For this, we use n_d samples (our data set) to construct our GP state-space model. We interpret the GP as a distribution over functions as follows

$$f_\psi(\boldsymbol{\xi}) \sim \mathcal{GP}(m_\psi(\boldsymbol{\xi}), k_\psi(\boldsymbol{\xi}, \boldsymbol{\xi}')) \tag{13}$$

which is fully characterized by a mean function $m_\psi(\boldsymbol{\xi})$: $\mathcal{X} \to \mathbf{R}$ and a covariance function $k_\psi(\boldsymbol{\xi}, \boldsymbol{\xi}')$: $\mathcal{X} \times \mathcal{X} \to \boldsymbol{R}$. The input ξ' refers to an arbitrary point in the input space at which the GP prediction is evaluated. It is important to note that the covariance function $k_\psi(\boldsymbol{\xi}, \boldsymbol{\xi}')$ models the GP's predictive uncertainty due to data sparsity or extrapolation, and not the process noise, which is assumed to be additive and input-independent during data generation.

Note that both the mean and covariance function produce a scalar. This means that they map control and state inputs $\boldsymbol{\xi}$ to a single output. If we want to produce n_x outputs (say, one for each state at a subsequent time), we must have n_x GPs, one for each state. Therefore, in this context, we will use data set $\mathcal{D}$ to train a GP, which given an input ξ_k, will aim to predict the mean and variance of a future state $\boldsymbol{x}_{k+1}$. We then concatenate n_x GPs to model such dynamics

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

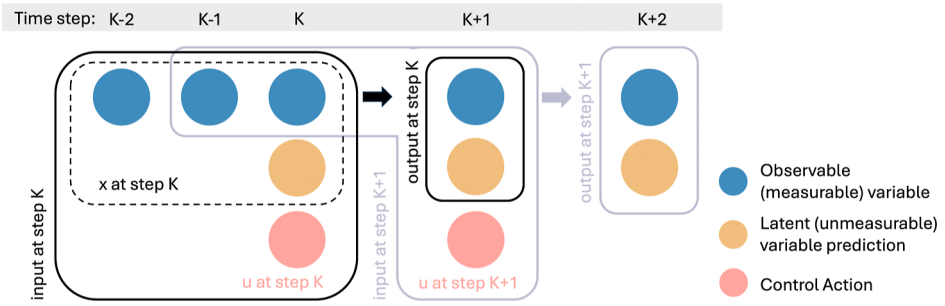


Figure 1. Schematic representation of the partially observable input−output structure used for GP model training and prediction. At time step $k = 0$, full state information (both observable and latent variables) is assumed to be available for model initialization. For $k > 0$, only observable variables (blue) are measured, and the GP model is used to predict the latent (unmeasurable) states (yellow) based on a time window of past measurements and control actions.

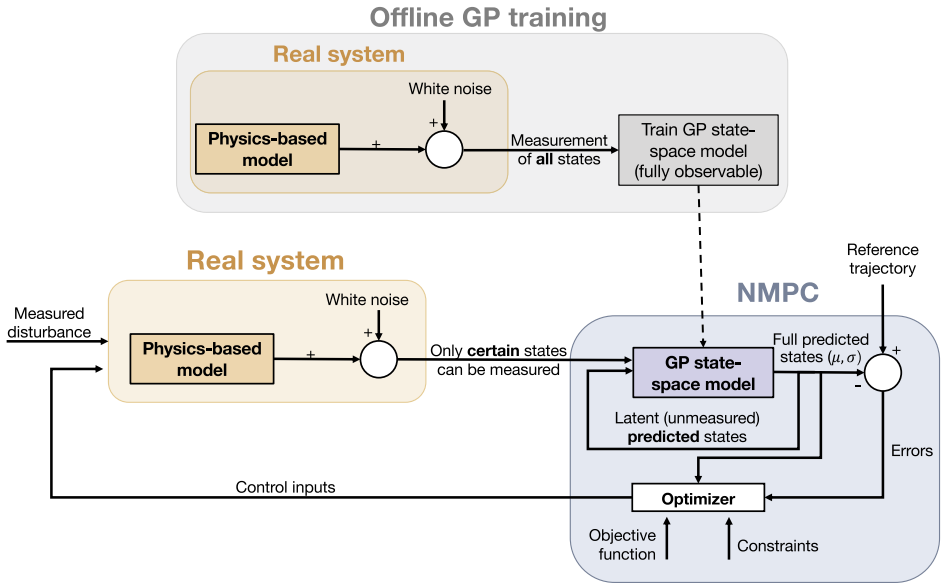


Figure 2. Schematic of the proposed GP-MPC framework. A physics-based model generates full-state data offline to train a GP model. Online, the GP model predicts full states from partial measurements, which are used by the NMPC optimizer to compute control actions subject to constraints. Disturbances and noise are added at the process level to reflect real-world conditions.

$$\boldsymbol{f}_{\Psi}(\boldsymbol{\xi}) = \begin{cases} f_{\psi_1}(\boldsymbol{\xi}) \sim \mathcal{GP}(0, k_{\psi_1}(\boldsymbol{\xi}, \boldsymbol{\xi}')) \\ \vdots \\ f_{\psi_{n_x}}(\boldsymbol{\xi}) \sim \mathcal{GP}(0, k_{\psi_{n_x}}(\boldsymbol{\xi}, \boldsymbol{\xi}')) \end{cases} \tag{14}$$

We can then concatenate all parameter vectors $\Psi = [\psi_1, \cdots, \psi_{nx}]$ and all covariance functions $\boldsymbol{k}_{\Psi}(\cdot,\cdot) = [k_{\psi 1}(\cdot,\cdot), \cdots, k_{\psi nx}(\cdot,\cdot)]$ to describe the concatenation of all kernel functions into

$$\boldsymbol{f}_{\Psi}(\boldsymbol{\xi}) \sim \mathcal{GP}(0, \boldsymbol{k}_{\Psi}(\boldsymbol{\xi}, \boldsymbol{\xi}')) \tag{15}$$

This representation of the dynamic system, via GP state-space models, is used in Section 3.3 to formulate a data-driven stochastic MPC.

3.2. Gaussian Processes Partially Observable State-Space Model. In real-world systems, not all state variables can be easily measured online. We distinguish these variables as observed variable $\boldsymbol{x}_{\text{observed}}$ (measurable) and latent variable $\hat{x}_{\text{latent}}$ (unmeasurable), as shown in eq 16.

$$\boldsymbol{x}^{(i)} = \{\hat{\boldsymbol{x}}^{(i)}_{\text{latent}}, \boldsymbol{x}^{(i)}_{\text{observed}}\}_{i=l}^{n_d} \tag{16}$$

To address this challenge, it is essential to develop a partially observable GP model that can predict all state variables based on the measurable ones.

A time window of the observed variables is introduced to compensate for the lack of direct measurements of the latent variables and to enhance the robustness of the predictions. We can consider the system as

$$\boldsymbol{x}_{k+1} = \hat{f}_d(\boldsymbol{x}_k, \boldsymbol{x}_{k,\text{history}}, \boldsymbol{u}_k) \tag{17}$$

where $\boldsymbol{x}_{k,\text{history}} = [\boldsymbol{x}_{\text{obs},k}, \ldots, \boldsymbol{x}_{\text{obs},k-l}]$ and l defines the time-window size.

Figure 1 illustrates the operation of the GP model at each time step. At time step k, the GP model takes a time window of l (with $l = 3$ in this example) previous observable states, along with estimated latent variables and nominated control actions, to predict the next state. This prediction includes both observable and latent variables. For the next prediction at $k + 1$, the observable states at $k + 1$ are measured and used to update the time window, ensuring that the GP model incorporates the most recent data for accurate predictions in subsequent steps.

The training data set for the partially observable GP model is constructed using data generated from the first-principles

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

MATLAB model. Since the latent variables are measurable offline, they can also be included in the training data set alongside the observable variables. By leveraging the comprehensive data set generated by the first-principles model, the GP model captures both observable and latent dynamics, improving its ability to represent the system. Consequently, this helps maintain predictive reliability and improve the accuracy of the control strategy by accounting for the cumulative effects of past actions.[4]

$$\mathcal{D} = \{\{\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(i)}_{\text{history}}, \boldsymbol{u}^{(i)}\}, \boldsymbol{y}^{(i)}\}_{i=l}^{n_d} \tag{18}$$

To simplify notation, we group the variables $\boldsymbol{x}_k^{(i)}$, $\boldsymbol{x}_{k,\text{history}}^{(i)}$, and $\boldsymbol{u}_k^{(i)}$ into $\hat{\boldsymbol{\xi}}_k^{(i)}$, where $\hat{\boldsymbol{\xi}} \in \hat{\mathcal{X}} \subseteq \boldsymbol{R}^{n_{x_{\text{latent}}}+n_{x_{\text{observed}}}\cdot l+n_u}$. Using a similar approach as derived in eqs 13 to 15, we develop a second GP state-space model for data-driven stochastic MPC under partially observable scenarios. This model enables learning complete system dynamics from comprehensive experimental measurements while ensuring reliable control during online operation with partial observations.

3.3. Gaussian Processes Model Predictive Control (GP-MPC). Figure 2 illustrates the proposed GP-based nonlinear model predictive control (GP-MPC) framework under online partial observability. The methodology consists of two main components: an offline training phase and an online control loop. In the offline phase, a high-fidelity physics-based model is used to simulate full-state trajectories, which are then used to train a GP state-space model. This fully observable training data set enables the GP to learn a surrogate model that captures nonlinear process dynamics with associated uncertainty.

During online implementation, only a subset of the process states is directly measurable. The trained GP model is used within the NMPC controller to predict the evolution of the full state vector from recent measurements, based on a fixed-length observation window. The predicted states are used to compute tracking errors relative to the set point, which are passed to the optimizer. The optimizer then solves the constrained finite-horizon optimal control problem using the GP predictions, the objective function, and the input constraints. Only the first control input is applied to the process. To account for real-world uncertainty, process measurements include additive white noise, and disturbances are explicitly added to the simulation environment.

We defined two types of objective functions for the GP-MPC framework. The first objective function aims to minimize the sum of a generic performance and/or economic term ($\boldsymbol{z}_k$), the change in control input ($\Delta\boldsymbol{u}_k = \boldsymbol{u}_k - \boldsymbol{u}_{k-1}$), the control input magnitude ($\boldsymbol{u}_k^T\mathbf{R}\boldsymbol{u}_k$), and the predicted variance ($\boldsymbol{\sigma}_k^T\mathbf{Q}_3\boldsymbol{\sigma}_k$)

$$J_1 = \sum_{k=0}^{N-1} (w_1\mathbf{z}_k^T\mathbf{Q}_1\mathbf{z}_k + w_2(\Delta\boldsymbol{u}_k)^T\mathbf{Q}_2(\Delta\boldsymbol{u}_k) + w_3\boldsymbol{u}_k^T\mathbf{R}\boldsymbol{u}_k + w_4\boldsymbol{\sigma}_k^T\mathbf{Q}_3\boldsymbol{\sigma}_k) \tag{19}$$

where $\boldsymbol{z}_k$ represents a generic performance and/or economic term that we aim to minimize or control, which will later be defined in the context of the specific application. Each term in the objective functions is penalized with a corresponding weight w_i. Note that we assumed that the first control variables are known; thus, $\boldsymbol{u}_{-1} = \boldsymbol{u}_0$.

We also propose a second objective function used for a trajectory-based MPC strategy. This second objective function focuses on minimizing the deviation of the performance and economic term from a set point ($\boldsymbol{z}_{SP}$), alongside the same terms as in the first objective function

$$J_2 = \sum_{k=0}^{N-1} (w_1(\mathbf{z}_k - \mathbf{z}_{SP})^T\mathbf{Q}_1(\mathbf{z}_k - \mathbf{z}_{SP}) + w_2(\Delta\boldsymbol{u}_k)^T \mathbf{Q}_2(\Delta\boldsymbol{u}_k) + w_3\boldsymbol{u}_k^T\mathbf{R}\boldsymbol{u}_k + w_4\boldsymbol{\sigma}_k^T\mathbf{Q}_3\boldsymbol{\sigma}_k) \tag{20}$$

Note that the term σ_k in eqs 19 and 20 corresponds to the predicted standard deviation of the system's state at time step k, which is obtained from the GP model. This predicted variance reflects the model's confidence at each step and is penalized in the objective to encourage reliable predictions and robust control performance, particularly under partial observability and disturbances. Specifically, σ_k^2 is defined as

$$\sigma_k^2 = \sigma_f^2(\boldsymbol{\xi}_k) = \sigma_n^2 - \boldsymbol{r}(\boldsymbol{\xi}_k, \mathbf{X})\mathbf{K}(\mathbf{X})^{-1}\boldsymbol{r}(\boldsymbol{\xi}_k, \mathbf{X})^{\mathrm{T}} \tag{21}$$

where $\boldsymbol{\xi}_k = [\boldsymbol{\mu}_k^{\mathrm{T}}, \boldsymbol{u}_k^{\mathrm{T}}]^{\mathrm{T}}$ is the GP input at time step k, σ_n^2 is the covariance magnitude from the squared-exponential kernel in eq 2, $\boldsymbol{K}(X)$ is the covariance matrix with entries $\mathrm{K}_{ij} = k(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})+\sigma_\varepsilon^2\delta_{ij}$, $\boldsymbol{r}(\boldsymbol{\xi}_k, \mathbf{X}) = [k(\boldsymbol{\xi}_k, \boldsymbol{x}^{(1)}), ..., k(\boldsymbol{\xi}_k, \boldsymbol{x}^{(n_d)})]$ is the vector of covariances between the test input ξ_k and training inputs.

The GP-MPC optimization problem is then formulated as

$$\begin{aligned}
&\min_{\mathbf{u}_{0:N}} \quad J(\boldsymbol{z}_{1:N}, \boldsymbol{u}_{0:N-1}, \boldsymbol{\sigma}_{1:N})\\
&\qquad\qquad \text{subject to:}\\
&\boldsymbol{\mu}_0 = \boldsymbol{x}(0) \quad \text{(initial condition)}\\
&\qquad\qquad \boldsymbol{z}_k = \mathcal{J}_z(\boldsymbol{\mu}_k)\\
&\qquad\qquad \boldsymbol{\xi}_k = [\boldsymbol{\mu}_k^T, \boldsymbol{u}_k^T]^T\\
&\qquad\qquad \boldsymbol{k}_\psi(\boldsymbol{\xi}_k, \Xi) := [k_\psi(\boldsymbol{\xi}_k, \boldsymbol{\xi}^{(1)}), ..., k_\psi(\boldsymbol{\xi}_k, \boldsymbol{\xi}^{(n_d)})]\\
&\qquad\qquad \boldsymbol{\mu}_{k+1}, \boldsymbol{\sigma}_{k+1} \leftarrow \mathcal{GP}(m_\psi(\boldsymbol{\xi}_k), \boldsymbol{k}_\psi(\boldsymbol{\xi}_k, \Xi))\\
&\qquad\qquad \boldsymbol{u}_{\min} \le \boldsymbol{u}_k \le \boldsymbol{u}_{\max}\\
&\text{for } k = 0, ..., N-1
\end{aligned} \tag{22}$$

Eq 22 is parametrized by the current state estimate $\boldsymbol{x}_0 \in \boldsymbol{R}^n_x$, analogous to the standard NMPC formulation in eq 5. The prediction horizon is initialized as $\mu_0 := \boldsymbol{x}_0$, and the subsequent GP-based state predictions μ_{k+1} are recursively computed using the GP posterior mean.

In our framework, two objective functions are defined to accommodate different control strategies: J_1 for economic optimization and J_2 for set point tracking. The choice between these objectives depends on the specific control scenario. The term $\boldsymbol{z}_k = \mathcal{J}_z(\boldsymbol{\mu}_k)$ maps the predicted state mean μ_k to an economic or performance-related variable used in the objective function to evaluate system performance over the control horizon. At each time step k, ξ_k represents the concatenated vector of the predicted state mean μ_k and the control input $\boldsymbol{u}_k$, with N denoting the control horizon. The initial state $\mathbf{Z}_0$ includes both observable and latent variables, where latent variables are assumed to be measurable offline and available for the first MPC step, ensuring an accurate initial condition before transitioning to predictive estimates in subsequent steps.

In addition, μ_{k+1} and σ_{k+1} represent the predicted mean and variance of the output, respectively, obtained from the GP

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

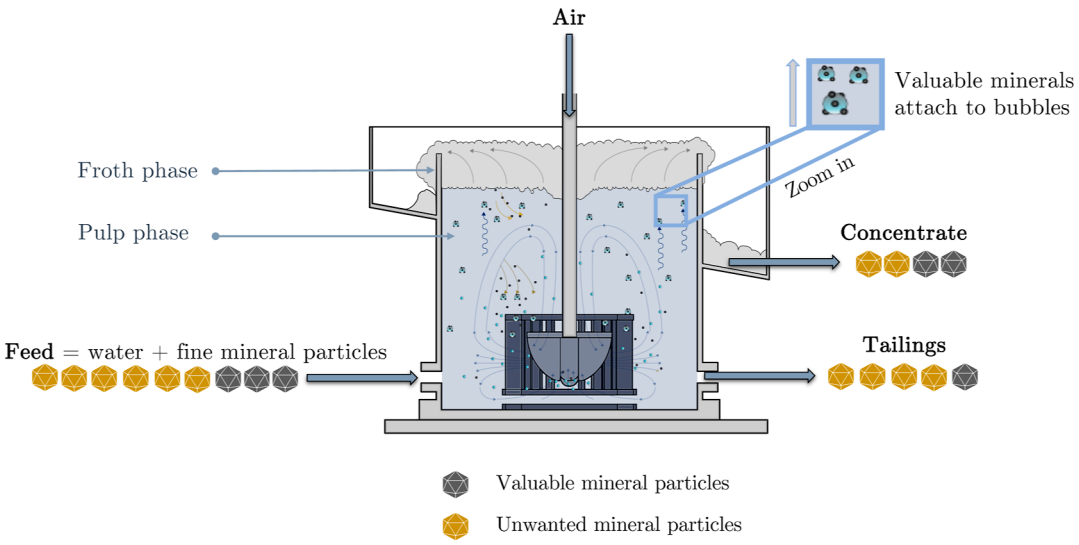


Figure 3. Schematic of the mineral froth flotation process. Note that the proportion of valuable and unwanted mineral particles (also known as gangue) in this schematic is illustrative and does not represent the actual proportion of a flotation operation, as the proportion is typically below 1% of valuable mineral.

distribution described by the mean function $m_\psi(\xi_k)$ and the covariance function $\boldsymbol{k}_\psi(\xi_k, \Xi)$, where Ξ represents the set of training data inputs. The control inputs are bounded by $\boldsymbol{u}_{\min}$ and $\boldsymbol{u}_{\max}$. It is important to note that the notation used in this formulation corresponds to the fully observable case, where ξ_k represents the concatenated vector of the predicted state mean μ_k and the control input $\boldsymbol{u}_k$. For the partially observable case, the previously defined notation $\hat{\boldsymbol{\xi}}_k$ should be used, which accounts for both observable and estimated latent states within the GP-based state-space model.

Although uncertainty propagation can be considered in GP state-space models,[44] we do not consider it in this work. This decision comes from the fact that available approaches typically rely on linearization or statistical moment-matching approximations, and most efficient methods focus on open-loop propagation of uncertainties. However, open-loop propagation often leads to overly conservative results due to the unchecked growth of uncertainties. Instead, we adopt an alternative strategy: limiting process uncertainty by penalizing the uncertainty considered within the MPC scheme.[18]

4. SYSTEM DESCRIPTION: MINERAL FROTH FLOTATION

The methodology proposed in Section 3 was implemented for a mineral froth flotation process. Mineral froth flotation is a separation process to extract valuable minerals, such as copper, from impurities (unwanted mineral particles, also known as gangue). This process involves mixing chemicals and air into stirred tanks, which modifies the surface properties of mineral particles, making them hydrophobic and causing them to attach to air bubbles. The resulting mixture generates a froth at the surface of the tank, which overflows as a mineral-rich concentrate. The unattached particles remain in the tank's pulp phase and exit from the bottom of the tank as tailings. A schematic of this process is depicted in Figure 3.

4.1. Flotation Model. Modeling mineral froth flotation presents complexities due to its multiphase nature, involving solid (minerals), liquid (water) and gas (air) components. In particular, modeling the performance of the froth phase, which defines the quality of the final product and overall efficiency, depends on a number of complex subprocesses such as bubble coalescence, particle-bubble attachment, and froth stability.[45−47] Significant research has focused on understanding the role of these subprocesses and their interactions in flotation performance, generally resulting in complex nonlinear models composed of differential equations that are unsuitable for online control purposes. For a comprehensive review of flotation modeling for control, please refer to.[8]

In this study, we used a dynamic physics-based model that incorporates froth physics[37] to generate training data for the GP model and to evaluate the impact of control actions, serving as a surrogate for the real system. The model has been validated at a laboratory scale,[38,48] ensuring its reliability in representing the flotation process dynamics. It comprises a nonlinear differential-algebraic equation (DAE) system with $2 + MC + B$ ordinary differential equations. Here, MC denotes the number of mineralogical classes, and B represents the number of bubble size classes. More information on bubble size classes can be found in.[49] In this study, we assumed $B = 5$ and $MC = 2$, corresponding to chalcopyrite (valuable mineral, $mc = 1$) and quartz (gangue (unwanted mineral), $mc = 2$). Presents an overview of the states and control variables of the model.

4.2. Problem Setup. A prediction horizon of 30 steps with a 5 min sampling interval was selected, as this duration is sufficient for the system to reach a new steady state under the tank dimensions and operating conditions considered in this study.[35,37,38] The training data set consists of six simulated trajectories designed to capture representative system dynamics while ensuring computational feasibility. This is particularly important to maintain manageable computational complexity for the GP training, which scales as $O(n_d^3)$ with the number of training points.[50]

The flotation process is modeled with nine states: mineral masses ($M_{\min}$, M_{gangue}), gas holdup for each bubble size class[49] (ε_1, ε_2, ε_3, ε_4, ε_5), pulp height (h_p), and tailings flow rate (Q_{tails}). In the full-state GP model, predictions focus on changes from the current state rather than directly forecasting the next state. This approach enhances accuracy and generalization by concentrating on localized changes while mitigating the

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

accumulation of prediction errors over time, which is critical for maintaining robustness in MPC applications.[4]

In contrast, the partially observable GP model accounts for the fact that only three variables (h_p, Q_{tails}, and the total gas holdup ($\sum_{b=1}^{5} \varepsilon_b$)) are directly measured, while the remaining variables are latent (unmeasured). These latent variables, however, can be realistically measured offline and incorporated into the initial condition of the model to improve prediction accuracy. This division reflects the typical availability of observed and latent variables in industrial-scale operations with existing instrumentation. This information is summarized in Table 1. The manipulated variables are the pulp level set point (h_p) and air flow rate (Q_{air}). These are the typical manipulated variables found at the industrial scale.

Table 1. States for the Flotation Process

symbol	unit	description	equation	GP observability	
				fully observable	partially observable
M_m	kg	mass of mineral class m	eq S1	yes	no
ε_b		gas holdup for bubble size b	eq S2	yes	only $\sum_{b=1}^{5} \varepsilon_b$
h_p	m	pulp level	eq S3	yes	yes
Q_{tails}	m^3/s	tailings flow rate	eq S4	yes	yes

The time window size (l) for the partially observable GP (eq 16) is set to 3. This time window size was selected based on empirical testing and domain-specific process knowledge. We evaluated window lengths ranging from $l = 1$ to $l = 5$ and found that $l = 3$ offers a practical trade-off between model accuracy, generalization, and computational efficiency. This window provides sufficient temporal context for the GP model to infer the latent states from sequences of observable outputs without introducing overfitting or unnecessary computational overhead during training or online control.

To define the objective function in the MPC (eqs 19 and 20), $C_{1,tails}$ from eq 23 is selected as the economic term (z_k), because it represents the concentration of valuable mineral particles (i.e., $m = 1$) in the tailing flow rate, a critical factor closely related to the economic efficiency of the flotation process. In other words, by minimizing the concentration of valuable particles in the tailings, we aim to maximize the concentration of valuable particles in the concentrate flow rate. This idea has also been adopted by other studies (e.g.[31]).

$$C_{1,tails} = \frac{M_{min}}{h_p(1 - \sum_{b=1}^{5} \varepsilon_b)A_{cell}} \tag{23}$$

The weight coefficients in the objective function were tuned to $[w_1, w_2, w_3, w_4] = [60, 0.3, 0.01, 0.03]$. The weights were tuned empirically to balance control performance, model confidence,

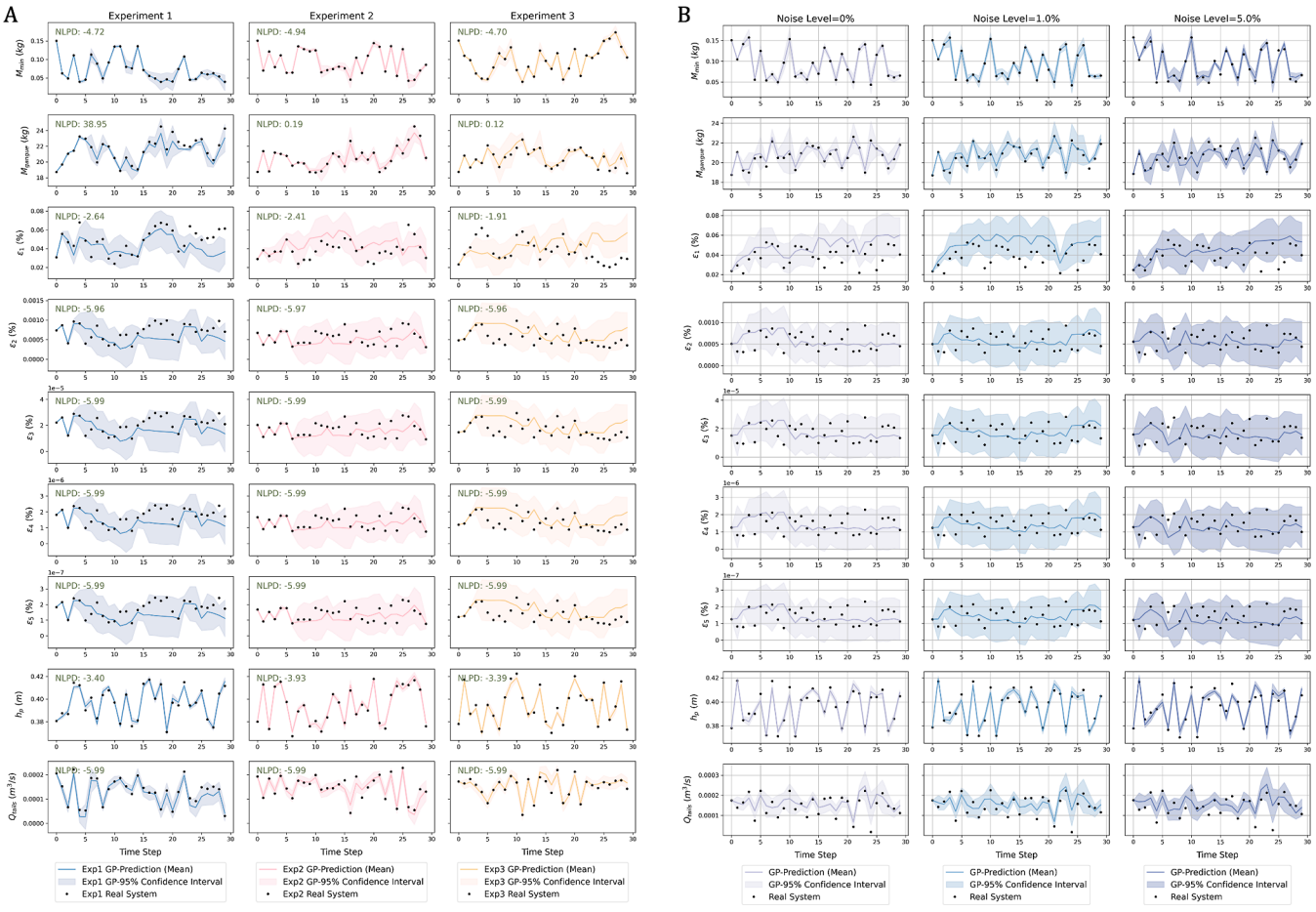


Figure 4. (A) Predictions of the full-state GP model for each state variable across three simulated experiments used for training. The solid lines represent the GP mean predictions, shaded regions indicate the 95% confidence intervals, and black dots denote the true state trajectories. (B) Cross-validation results under three levels of Gaussian measurement noise (0%, 1.0%, and 5%).

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

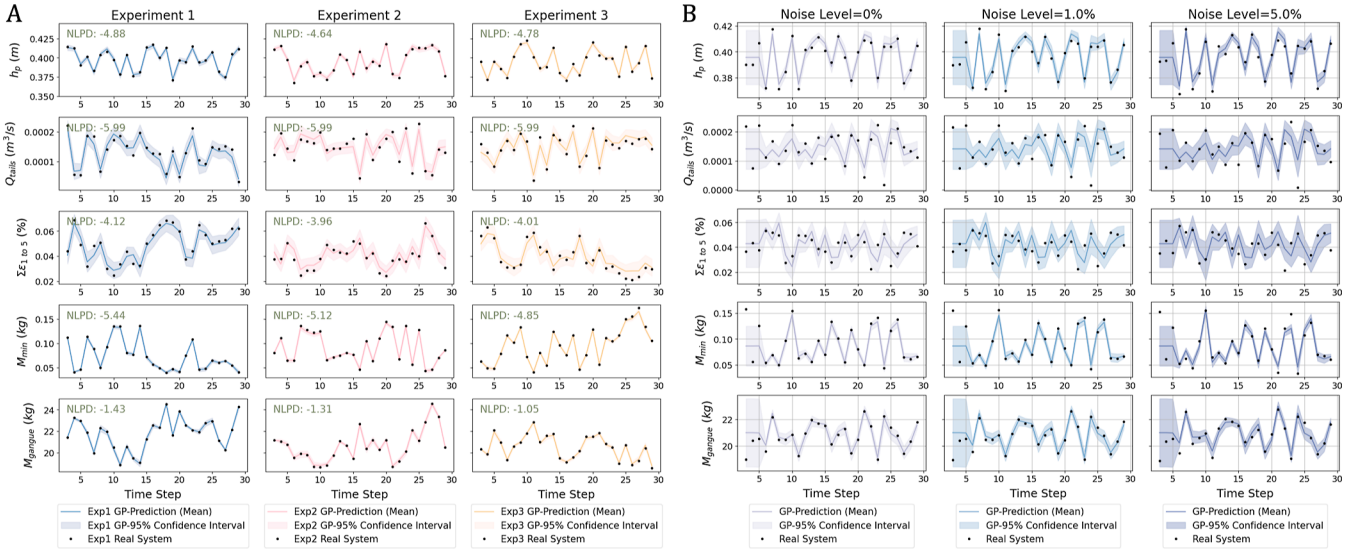


Figure 5. (A) Predictions of the partially observable GP model for each state variable across three simulated experiments used for training. The solid lines represent the GP mean predictions, shaded regions indicate the 95% confidence intervals, and black dots denote the true state trajectories. (B) Cross-validation results under three levels of Gaussian measurement noise (0%, 1.0%, and 5%).

and computational tractability. The prediction horizon in the MPC is set to 5 time steps. It is important to note that $C_{1,\text{tails}}$ is calculated from the final values of each variable, as optimizing it is the ultimate objective, while the other terms in the objective function are calculated from subsequent steps, given that they relate to the actual execution of the control actions.

4.3. Implementation Details and Initial Data Set Generation. The GP model was trained using three data sets generated by model simulations in MATLAB R2022b. During these simulations, the control action was adjusted every 300 s within the bounds $h_p \subseteq [0.37, 0.42]$ and $Q_{\text{air}} \subseteq [9\times10^{-4}, 3\times10^{-3}]$. The GP model was then constructed in Python 3.8.19 using an RBF kernel, with 10 hyperparameters, using a multistart approach.

For training the full-state model, $X_{\text{data set}}$ was constructed by extracting the state values at every 300 s, while $Y_{\text{data set}}$ was obtained by calculating the difference between consecutive time steps. For the partially observable model, although latent variables are not directly measurable in real-time applications, they were included in the training stage by leveraging offline measurements obtained from the first-principles model. This approach enhances the model's understanding of the system's underlying dynamics.

To establish the initial time window for the partially observable model, the system was operated under a preset control action from $k = 0$ to $k = 2$ to collect measurements for the observable state vector, defined as $\mathbf{X}_{\text{obs}} = [h_p, Q_{\text{tails}}, \sum_{b=1}^{5} \varepsilon_b]$. Latent variables, represented by the state vector $\mathbf{X}_{\text{latent}} = [M_{\text{min}}, M_{\text{gangue}}, \varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4, \varepsilon_5]$, are assumed to be available initially through offline measurements. These initial values are used to initialize the model for the first MPC step, ensuring accurate state estimation. During real-time operation, the latent variables are replaced by their predicted values for the subsequent steps to accommodate partial observability. The fitted GP model, integrated within the objective function, predicts state variables over the control horizon based on the current state and proposed control actions, enabling the computation of the objective value.

The optimization of control actions is performed using a constrained nonlinear solver (SLSQP) that minimizes the GP-MPC objective function defined in eq 22. At each control step, an initial guess of the control sequence is refined iteratively by the solver, subject to predefined input bounds. The solver evaluates the objective function by simulating the GP model over a finite control horizon, set to $h_c = 5$. To reduce oscillations, the optimized sequence takes the form $[\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{u}_{k+2}, \boldsymbol{u}_{k+2}, \boldsymbol{u}_{k+2}]$, instead of a fully independent sequence $[\boldsymbol{u}_k, ..., \boldsymbol{u}_{k+4}]$. After each control step, the first control action is applied, and the MATLAB simulation provides a measurement of the resulting system state. Noise is added after each one-step MATLAB simulation. This noise reflects measurement uncertainty and is added to the observable outputs after each MATLAB simulation step. The resulting noisy measurements are then used to construct the GP input vector ξ_k for the subsequent prediction. In disturbance scenarios, the feed flow rate Q_{feed} is no longer treated as constant. Instead, it is supplied as a time-varying input to the MATLAB simulator to enable flexible and more realistic disturbance by modifying Q_{feed} at particular time steps.

5. RESULTS AND DISCUSSIONS

5.1. GP-Model: Full-State. To evaluate the model's fit, particularly how well the predicted distributions align with the true distribution of the data, we extensively used the negative log predictive density (NLPD) metric, a standard for assessing GP models.[51] As shown in Figure 4A, the NLPD values for nearly all fittings are close to zero or slightly negative, indicating a well-calibrated model with high accuracy and appropriately estimated uncertainty. Notably, for variables such as M_{min} and h_p, which are later used in calculating the objective value ($C_{1,\text{tails}}$), the full-state GP model provides highly accurate predictions with relatively small standard deviations that are no larger than ±10% of the state value. This level of precision is crucial for optimizing $C_{1,\text{tails}}$, underscoring the model's significance in achieving the controller's objectives.

The cross-validation performance of the full-state GP model under different noise levels, as presented in Figure 4B, highlights the GP model's robustness. The results demonstrate that the

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

Table 2. Predictive Performance of GP Models Evaluated by Negative Log Predictive Density (NLPD)

variable		full-state GP				partially observable GP			
	experiment	1	2	3	average	1	2	3	average
h_p		−3.40	−3.93	−3.39	−3.57	−4.88	−4.64	−4.78	−4.77
Q_{tails}		−5.99	−5.99	−5.99	−5.99	−5.99	−5.99	−5.99	−5.99
ε_1		−2.64	−2.41	−1.91	−2.32				
ε_2		−5.96	−5.97	−5.96	−5.96				
ε_3		−5.99	−5.99	−5.99	−5.99				
ε_4		−5.99	−5.99	−5.99	−5.99				
ε_5		−5.99	−5.99	−5.99	−5.99				
$\Sigma\varepsilon_{1\text{ to }5}$						−4.12	−3.96	−4.01	−4.03
M_{min}		−4.72	−4.94	−4.70	−4.79	−5.44	−5.12	−4.85	−5.14
M_{gangue}		38.95	0.19	0.12	13.09	−1.43	−1.31	−1.05	−1.26

model effectively handles small perturbations in the data with minimal performance degradation. This robustness is particularly beneficial in dynamic or highly variable environments, where the input data may be noisy or subject to sudden fluctuations. By ensuring consistent performance even under these conditions, the model enables the MPC to make accurate and reliable predictions, thereby improving overall system performance. This ability to retain predictive accuracy despite data imperfections makes the model well-suited for real-world applications, where such variability is often unavoidable.

5.2. GP-Model: Partially Observable. Despite the increased training burden associated with the time window approach, it did not result in significantly longer training times than the full-state model. As illustrated in Figure 5A,B, the partially observable model effectively captures overall trends and demonstrates robustness in the presence of noise. The model exhibited negative NLPD values for all variables (see Table 2), indicating that it remains a reliable tool for practical applications despite the inherent challenges of measuring certain state variables (see Table 3).

Table 3. Final Concentration of $C_{1,tails}$ (kg/m^3) Achieved by Different GP-MPC Strategies and Measurement Noise Levels[a]

model	trajectory tracking	end-point optimization			disturbance scenario
		0% noise	1% noise	5% noise	(with 5% measurement noise)
full-state	0.840	0.620	0.595	0.579	0.592
partially observable	0.896	0.922	0.924	0.921	0.925

[a]Note: 0%/1%/5% noise indicate the level of white noise added to the measurement data.

Compared to the full-state model, the partially observable model achieved a higher level of accuracy with a smaller confidence interval by predicting $\sum_b \varepsilon_b$ as a combined variable rather than individually. Since $\sum_b \varepsilon_b$ is also used to calculate $C_{1,tails}$, its accurate prediction has the potential to improve the implementation of MPC, thus improving the overall control performance.

5.3. Gaussian Process Model Predictive Control. *5.3.1. Trajectory Tracking with GP-MPC.* After developing reliable GP models, we implemented them within an MPC framework to achieve efficient control. The first step involved trajectory tracking to ensure consistent system behavior and the feasibility of the control actions generated by the MPC. This procedure is essential to avoid the optimization process suggesting solutions that are theoretically optimal but practically unfeasible.

As shown in Figure 6A, the set point for $C_{1,tails}$ was initially 1.2 kg/m^3 and later adjusted to 0.6 kg/m^3. The GP-MPC controller responded efficiently, solving each NMPC problem in under 30 s on average using an Apple M2 CPU laptop (8 GB RAM), which fits within the 5 min sampling period. The final predicted mean was 0.840 kg/m^3, and the actual state reached 0.847 kg/m^3, indicating a steady-state offset. This offset can be attributed to the saturation of the h_p control input, which reached its lower bound and limited further corrective action. Although the Q_{tair} input remained within bounds, its influence on the output may have been underweighted in the cost function, reducing its effectiveness in achieving the target set point. This suggests that further tuning of control weightings or rebalancing of the economic vs variance penalties may help improve performance. The rest of the process variables remained within the 95% confidence interval and showed strong agreement with the predicted trajectories, as illustrated in Figure 6B.

For the partially observable model, the results show that it can respond to the set point change within one time step and achieve a similar level of trajectory tracking as the full-state model, with a final predicted $C_{1,tails}$ = 0.896 kg/m^3. Although M_{min} and M_{gangue} were assumed to be latent (unmeasured), their simulation results were recorded to assess the performance of the GP-MPC. As shown in 6(D) (red cross), the model provided accurate predictions for both variables, with deviations smaller than 3%.

To compare the performance of both models (see Figure 6A and C), although both models reached a similar level of tracking, the confidence interval of the partially observable model was significantly larger than that of the full-state model. This difference is likely due to the lack of direct observation of M_{min}. Furthermore, the control variables h_p and Q_{air} exhibited consistent control behavior in both scenarios, ensuring smooth system operation.

5.3.2. End-Point Optimization with GP-MPC. Since our design aims to solve optimization in real-time, data acquisition is required, where measurement noise is unavoidable. To assess the impact of measurement noise, we evaluated the GP-MPC performance under three conditions: 0% noise (no noise), 1% noise, and 5% noise. For the full-state model, as can be seen from Figure 7A−C, the actual state values of $C_{1,tails}$ generally align closely with the GP prediction across all scenarios. This close alignment is attributed to the accurate state value predictions, which exhibit a consistent 95% confidence interval (see Figures 7D, and S3D,E in the Supporting Information). However, as the noise level increases, while the optimization performance does

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

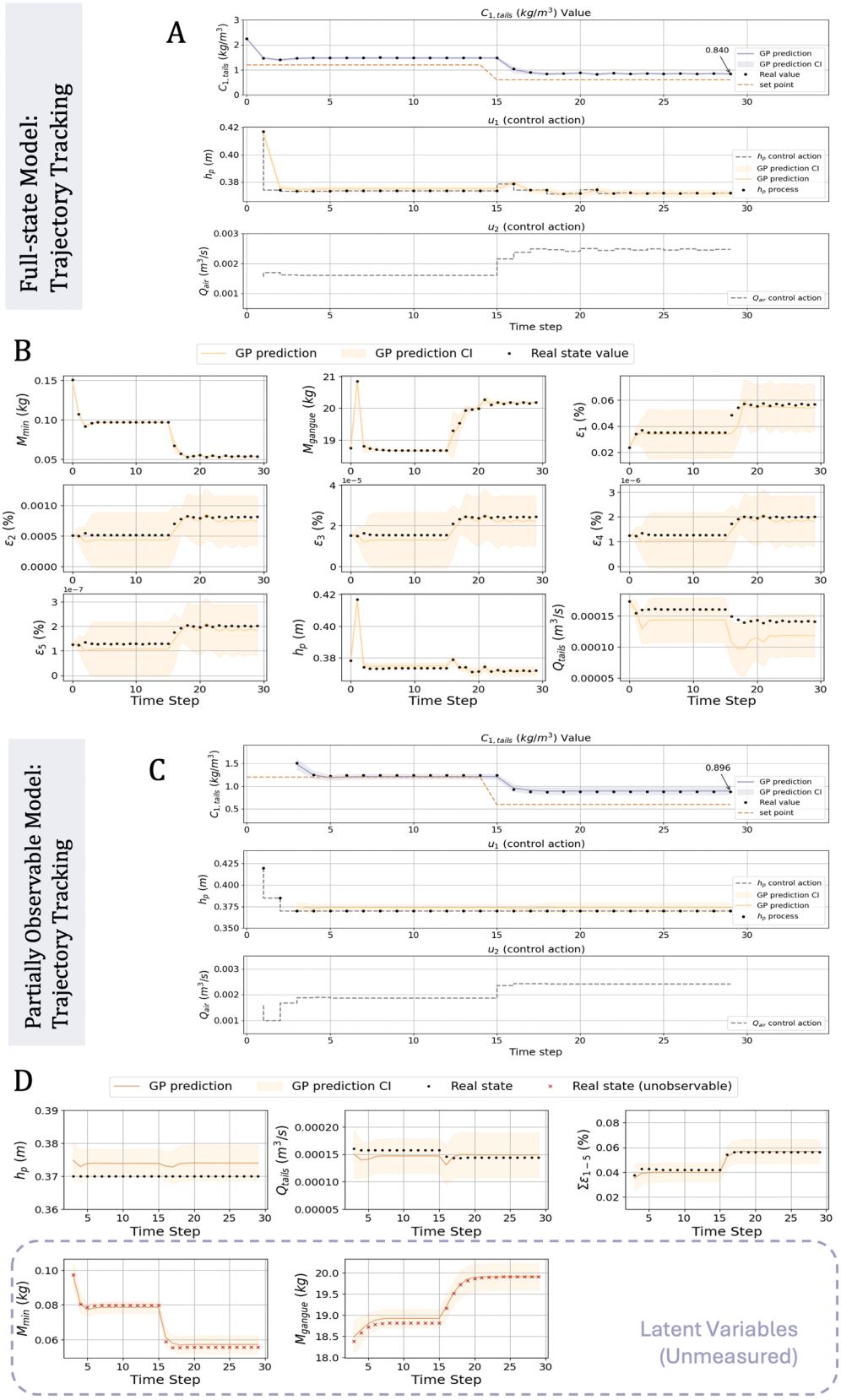


Figure 6. Closed-loop trajectory tracking performance of the proposed GP-MPC framework. (A) Full-state model: control performance using GP-MPC under full observability. (B) Full-state predictions for all state variables. (C) Partially observable model. (D) GP-based predictions of the latent (unmeasurable) state variables under partial observability. The shaded areas represent the 95% confidence intervals for GP predictions.

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

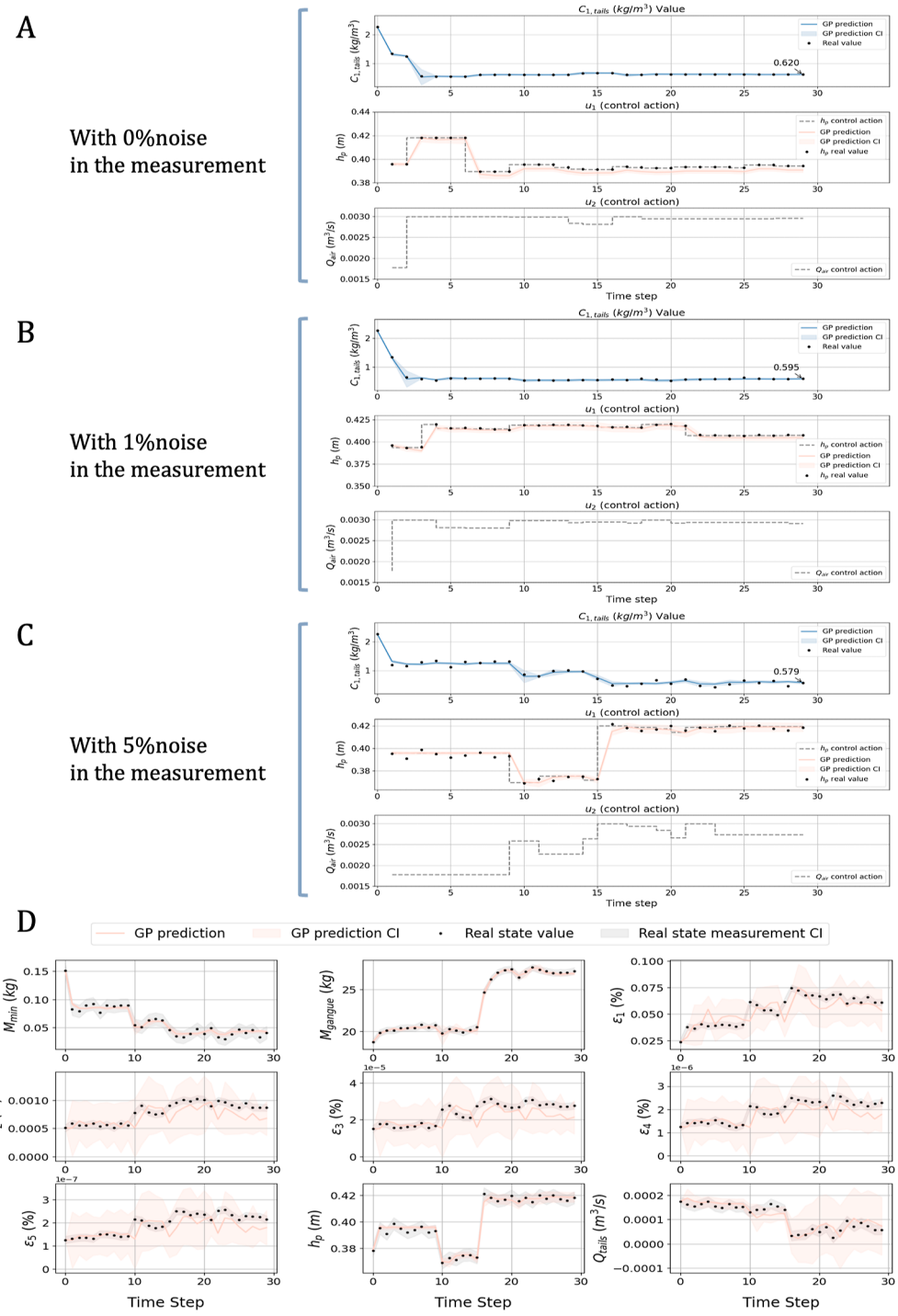


Figure 7. Closed-loop end-point optimization performance of the proposed GP-MPC framework using the fully observable GP model. (A) Without measurement noise. (B) With 1% measurement noise. (C) With 5% measurement noise. (D) GP-based predictions of the state variables under full observability. The shaded areas represent the 95% confidence intervals for GP predictions.

not significantly degrade, a noticeably higher level of fluctuation is observed, indicating the influence of measurement noise on the consistency of control actions.

In the case of the partially observable model, Figure 8A–C demonstrate that, due to the partially observable nature where fewer measurements are available, the impact of noise in the measurements is less pronounced. For instance, the final predicted value of $C_{1,tails}$ for the 0% noise, 1% noise and 5% noise scenarios are 0.922, 0.924, and 0.921 kg/m^3, respectively, which are almost identical. Although the MPC did not minimize $C_{1,tails}$ to the same extent as the full-state model, this consistency shows that the partially observable model has significantly higher robustness, which is crucial for real-world operations. This conclusion is further supported by the convergence of all objective function terms to a relatively steady level in this model, as seen in Figure S4A–C in the Supporting Information. Moreover, as detailed in Figure 8C, all state variables, regardless of their availability for online observation, exhibit strong alignment with the GP predictions, indicating the model's capability to accurately capture system dynamics (Figure S4D,E

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

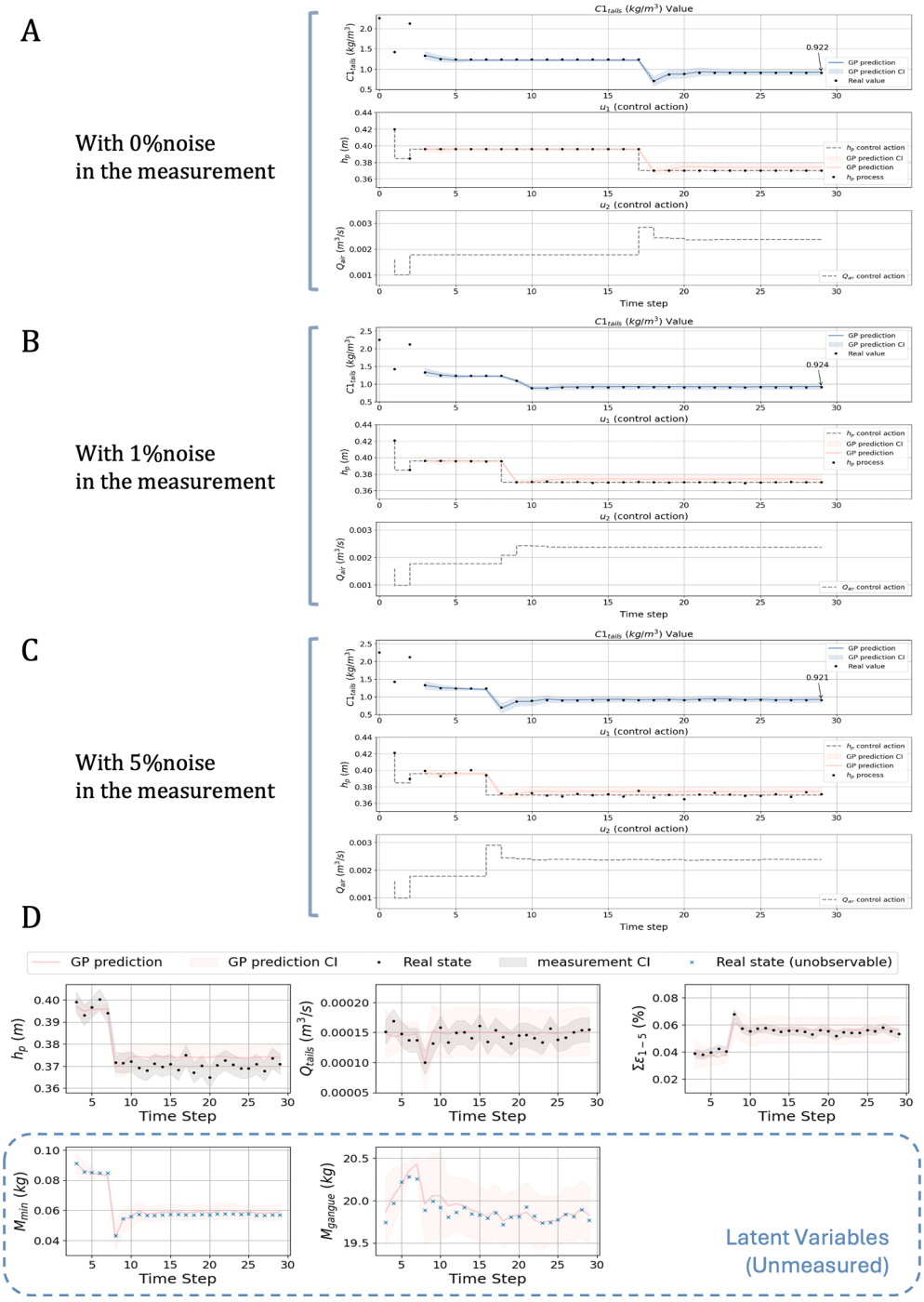


Figure 8. Closed-loop end-point optimization performance of the proposed GP-MPC framework using the partially observable GP model. (A) Without measurement noise. (B) With 1% measurement noise. (C) With 5% measurement noise. (D) GP-based predictions of the state variables under partial observability. The shaded areas represent the 95% confidence intervals for GP predictions.

in the Supporting Information). Furthermore, the optimization problem for each 5 min time step took no longer than 30 s on average in both cases, confirming the practical feasibility of implementing our strategy in real-world processes.

5.3.3. Disturbance Scenario: Feed Flow Rate Disturbances. To ensure its robustness and reliability in real-world applications, it is also crucial to evaluate the controller's performance under disturbance conditions. As a case study, we tested the GP-MPC's ability to handle disturbances in the feed flow rate (Q_{feed}), a common process variable that frequently experiences fluctuations and can significantly impact the final product quality.

In the disturbance scenario, feed flow rates at 90% and 110% of the nominal value of $Q_{feed} = 1.662 \times 10^{-4}$ m^3/s were tested sequentially, as illustrated in Figure 9A. For the full-state model, both the GP model's predictions and the actual state values of the objective variable exhibited slightly higher levels of fluctuation compared to the undisturbed case. However, there remained a strong alignment between the predicted and actual values, indicating that even though the GP model was trained on

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

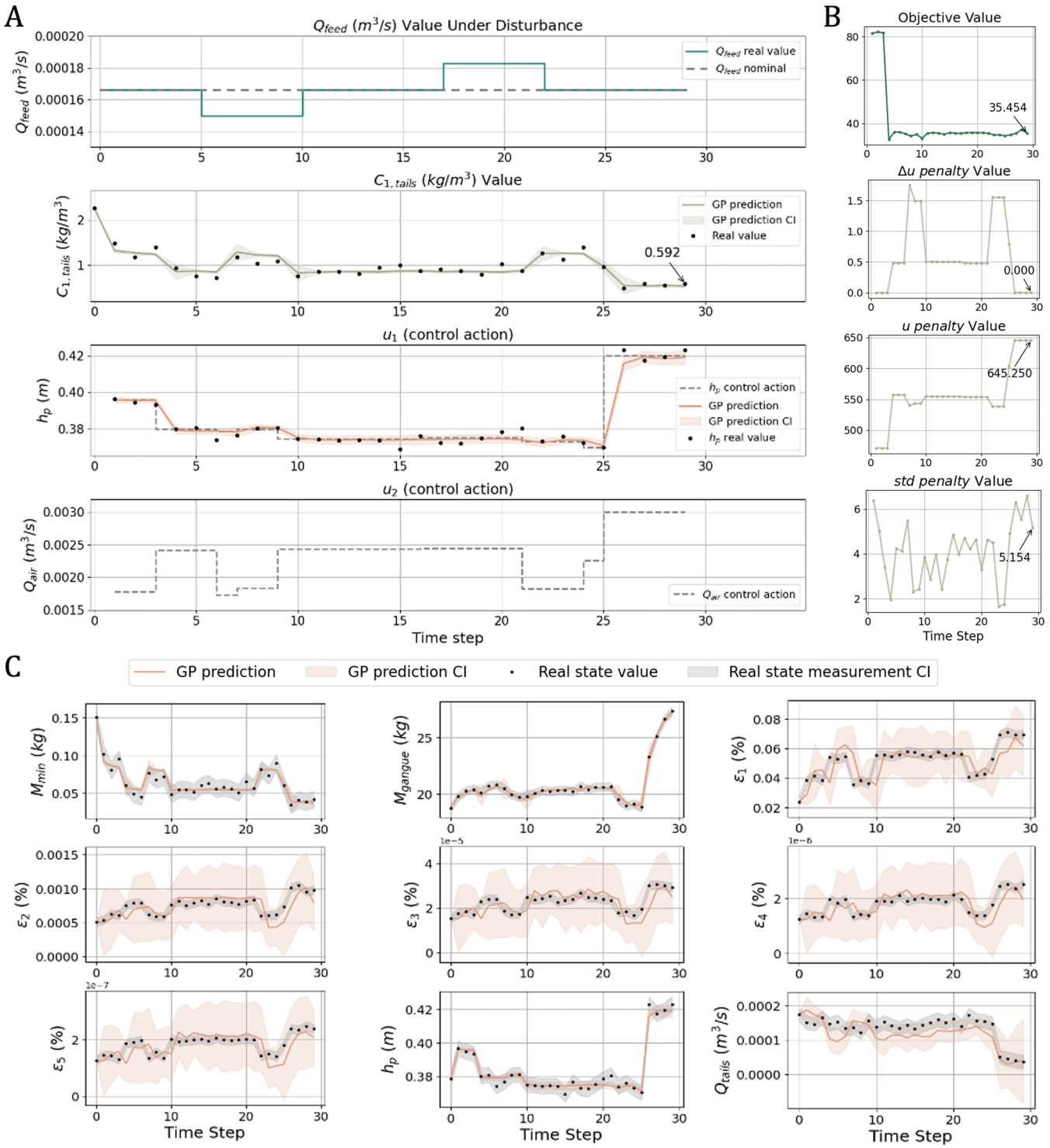


Figure 9. Optimization performance of the full-state GP-MPC under feed flow disturbance and 5% measurement noise. (A) Closed-loop tracking under step disturbances in the feed flow rate (Q_{feed}). (B) Decomposition of the objective function into its components: total cost, Δu penalty, control effort (u) penalty, and predicted standard deviation (uncertainty) penalty. (C) GP model predictions of all state variables with shaded 95% confidence intervals.

undisturbed data, it retains sufficient robustness to effectively manage moderate disturbances.

For the full-state model, the final state achieved was $C_{1,tails}$ = 0.592 kg/m^3, with both the GP predictions and the actual state values of the objective variable exhibiting higher levels of fluctuation compared to the undisturbed case. During disturbances, the state values occasionally fell outside the confidence interval (see Figure 9A). However, unlike the full-

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

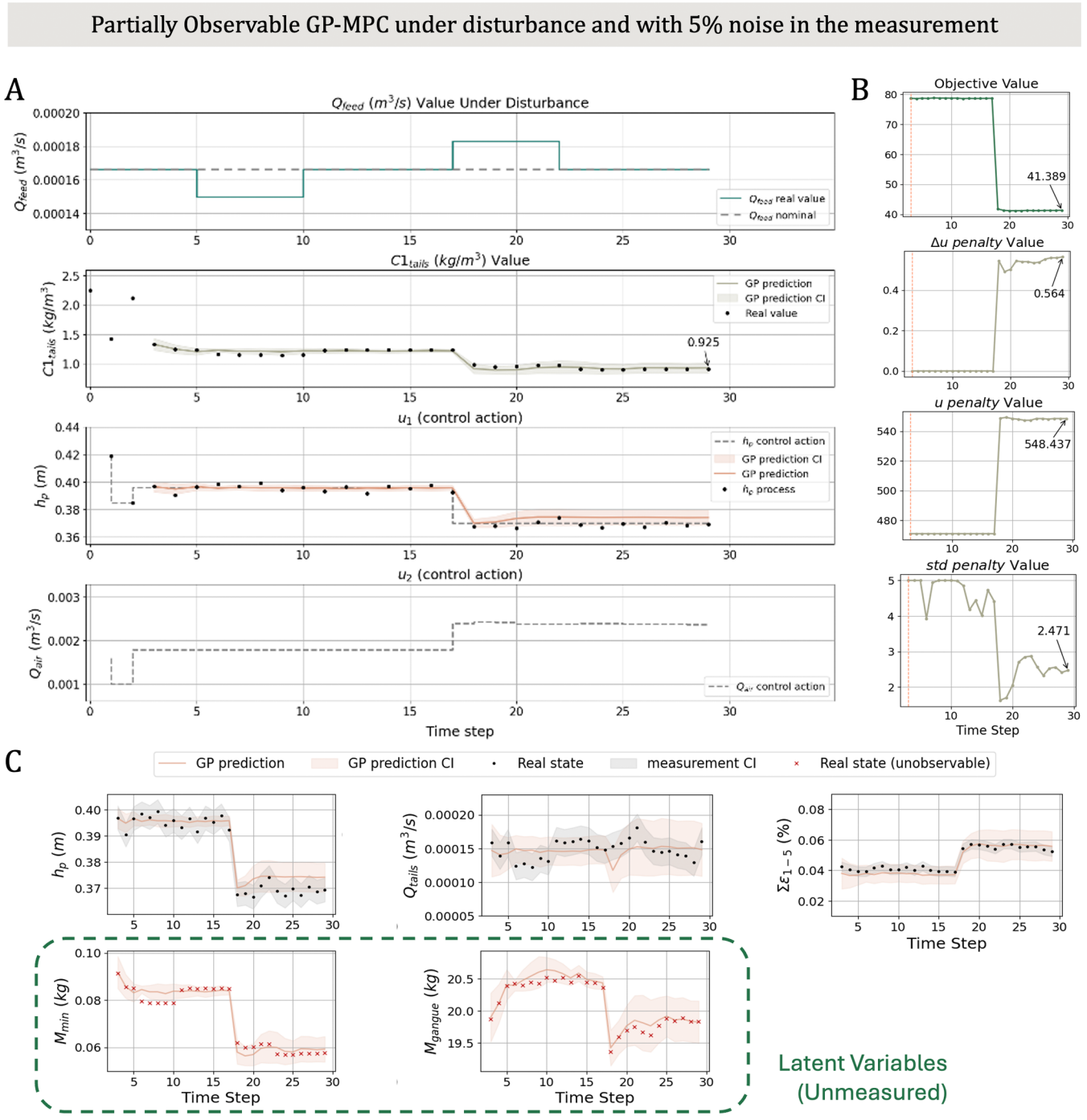


Figure 10. Optimization performance of the partially observable GP-MPC under feed flow disturbance and 5% measurement noise. (A) Closed-loop tracking under step disturbances in the feed flow rate (Q_{feed}). (B) Decomposition of the objective function into its components: total cost, Δu penalty, control effort (u) penalty, and predicted standard deviation (uncertainty) penalty. (C) GP model predictions of all state variables with shaded 95% confidence intervals.

state model, when the same disturbance test was conducted on the partially observable model, which reached a final optimization level of $C_{1,tails} \approx 0.92$ kg/m^3 similar to the undisturbed case, the larger confidence interval provided greater tolerance to disturbances. As shown in Figure 10A, even when disturbances occurred between time steps 17 and 22, the actual state values of $C_{1,tails}$ remained within the confidence interval.

A similar trend was observed for the latent variables (see Figure 10C), with almost all state values remaining within the CI at all time steps. Furthermore, the MPC optimization process, as detailed in Figure 10B, converged to values comparable to those observed in the undisturbed case (Figure S4C in the Supporting Information). This consistency demonstrates that the GP model, even when trained on undisturbed data, retains sufficient robustness to effectively manage moderate disturbances.

6. CONCLUSIONS

In this work, we developed and demonstrated the application of a fully and partially observable Gaussian process-based model predictive control (GP-MPC) framework for froth flotation,

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322

targeting the minimization of mineral tailings concentration ($C_{1,tails}$), as a proxy to maximize mineral recovery, under varying operational conditions. By incorporating GP models, we were able to account for system uncertainties and disturbances, significantly enhancing the controller's robustness compared to traditional control strategies. Our GP-MPC framework, particularly the partially observable model, demonstrated an ability to handle noise and disturbances effectively, maintaining accurate predictions of state variables even when some were unmeasurable in real-time.

The results from both the full-state and partially observable models confirmed the potential of our approach to providing reliable control in dynamic environments, with the partially observable GP-MPC offering superior robustness under noise and disturbances. Additionally, including historical data through a time window further improved the accuracy of predictions, ensuring better control performance in real-world applications.

This study contributes to the broader application of data-driven control strategies in industrial processes, specifically addressing the challenges of partial observability and process uncertainty. Our work highlights the advantages of integrating stochastic process modeling into MPC frameworks, paving the way for future improvements in the economic and operational efficiency of froth flotation and other complex industrial processes. Future research will focus on extending the model to other complex, multivariable systems and exploring its scalability in large industrial applications.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.iecr.5c00660.

Dynamic flotation model equations and normalization parameters; nomenclature and DAE model structure;[37,38] additional figures showing control performance and state prediction under varying observability and noise levels (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Ehecatl Antonio del Río Chanona − *Department of Chemical Engineering, Imperial College London, London SW7 2AZ, U.K.; Sargent Centre for Process Systems Engineering, Imperial College London, London SW7 2AZ, U.K.*; Email: a.del-rio-chanona@imperial.ac.uk

Paulina Quintanilla − *Department of Chemical Engineering, Brunel University of London, Uxbridge UB8 3PH, U.K.*; orcid.org/0000-0002-7717-0556; Email: paulina.quintanilla@brunel.ac.uk

Author

Yicong Wang − *Department of Chemical Engineering, Imperial College London, London SW7 2AZ, U.K.*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.iecr.5c00660

Notes

The authors declare no competing financial interest.

■ REFERENCES

(1) Maciejowski, J. M.; Huzmezan, M. *Robust Flight Control: A Design Challenge*; Springer, 2007; pp 125−134.

(2) Ostafew, C. J.; Schoellig, A. P.; Barfoot, T. D. Robust Constrained Learning-based NMPC enabling reliable mobile robot path tracking. *Int. J. Robot. Res* **2016**, *35*, 1547−1563.

(3) Rawlings, J. B.; Mayne, D. Q. *Model Predictive Control: Theory and Design*; Nob Hill Publishing: Madison, WI, USA, 2009.

(4) Kocijan, J. *Modelling and Control of Dynamic Systems Using Gaussian Process Models*; Springer Science & Business Media, 2016.

(5) Allgower, F.; Findeisen, R.; Nagy, Z. K. Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Eng* **2004**, *35*, 299−316.

(6) Lucia, S. *Robust multi-stage Nonlinear Model Predictive Control*; Shaker Dortmund, 2015.

(7) Sun, Z.; Qin, S. J.; Singhal, A.; Megan, L. Performance monitoring of model-predictive controllers via model residual assessment. *J. Process Control* **2013**, *23*, 473−482.

(8) Quintanilla, P.; Neethling, S. J.; Brito-Parada, P. R. Modelling for froth flotation control: A review. *Miner. Eng.* **2021**, *162*, 106718.

(9) Hewing, L.; Arcari, E.; Fröhlich, L. P.; Zeilinger, M. N. On Simulation and Trajectory Prediction with Gaussian Process Dynamics. In *Proceedings of Machine Learning Research: 2nd Annual Conference on Learning for Dynamics and Control*, 2020; pp 1−11.

(10) Bradford, E.; Imsland, L.; del Rio-Chanona, E. A. Nonlinear model predictive control with explicit back-offs for Gaussian process state space models. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019; pp 4747−4754.

(11) Savage, T.; Zhang, D.; Mowbray, M.; Río Chanona, E. A. D. Model-free safe reinforcement learning for chemical processes using Gaussian processes. *IFAC-PapersOnLine* **2021**, *54*, 504−509.

(12) Deisenroth, M.; Rasmussen, C. PILCO. A Model-Based and Data-Efficient Approach to Policy Search *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011; pp 465−472.

(13) Kamthe, S.; Deisenroth, M. P. Data-Efficient Reinforcement Learning with Probabilistic Model Predictive Control. *International conference on artificial intelligence and statistics*, 2018.

(14) Hewing, L.; Wabersich, K. P.; Menner, M.; Zeilinger, M. N. Learning-Based Model Predictive Control: Toward Safe Learning in Control. *Annu. Rev. Control Robot. Auton. Syst* **2020**, *3*, 269−296.

(15) Zheng, Y.; Li, S.; Hou, B. A distributed model predictive control with neighborhood state feedback invariant set for reconfigurable networked systems. *Int. J. Robust Nonlinear Control* **2022**, *32*, 5600−5618.

(16) Girard, A.; Rasmussen, C.; Candela, J. Q.; Murray-Smith, R. Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting. *Advances in neural information processing systems* 2002, *15*.

(17) Kocijan, J.; Murray-Smith, R.; Rasmussen, C. E.; Girard, A. Gaussian process model based predictive control. In *Proceedings of the 2004 American control conference*, 2004; pp 2214−2219.

(18) Bradford, E.; Imsland, L.; Zhang, D.; del Rio Chanona, E. A. Stochastic data-driven model predictive control using gaussian processes. *Comput. Chem. Eng.* **2020**, *139*, 106844.

(19) Graßhoff, J.; Männel, G.; Abbas, H. S.; Rostalski, P. Model Predictive Control using Efficient Gaussian Processes for Unknown Disturbance Inputs. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019; pp 2708−2713.

(20) Zhang, F.; Wang, L. Disturbance rejection design for Gaussian process-based model predictive control using extended state observer. *Comput. Chem. Eng.* **2024**, *186*, 108708.

(21) Yang, X.; Maciejowski, J. M. Fault tolerant control using Gaussian processes and model predictive control. *Int. J. Appl. Math. Comput. Sci* **2015**, *25*, 133−148.

(22) Klenske, E. D.; Zeilinger, M. N.; Scholkopf, B.; Hennig, P. Gaussian process-based predictive control for periodic error correction. *IEEE Trans. Control Syst. Technol.* **2016**, *24*, 110−121.

(23) McAllister, R. T.; Rasmussen, C. E. Data-efficient reinforcement learning in continuous state-action Gaussian-POMDPs; *31st International Conference on Neural Information Processing Systems (NIPS'17)*: Red Hook, NY, USA, 2017; pp 2037−2046.

(24) Park, S.-S.; Park, Y.-J.; Min, Y.; Choi, H.-L. Online Gaussian process state-space model: Learning and planning for partially observable dynamical systems. *Int. J. Control Autom. Syst* **2022**, *20*, 601−617.
(25) Wills, B. A.; Finch, J. *Wills' Mineral Processing Technology: An Introduction to the Practical Aspects of Ore Treatment and Mineral Recovery*; Butterworth-heinemann, 2015.
(26) Quintanilla, P.; Navia, D.; Neethling, S.; Brito-Parada, P. Evaluation of Changes in Feed Particle Size within an Economic Model Predictive Control Strategy for Froth Flotation. *IFAC-PapersOnLine* **2023**, *56*, 2317−2322.
(27) González, R. A.; Quintanilla, P. Grey-box Recursive Parameter Identification of a Nonlinear Dynamic Model for Mineral Flotation. In *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2024; pp 2967−2972.
(28) Olivier, L. E.; Craig, I. K. A survey on the degree of automation in the mineral processing industry. *IEEE AFRICON: Science, Technology and Innovation for Africa, AFRICON 2017* 2017, 404−409..
(29) Pérez-García, E.; Bouchard, J.; Poulin, E.. Integrating online mineral liberation data into process control and optimization systems for grinding−separation plants. *J. Process Control* **2021**, *105*, 169−178.
(30) Oosthuizen, D.; le Roux, J.; Craig, I. Non-linear Model Predictive Control to Improve the Mineralogical Efficiency of Flotation Circuits. *IFAC-PapersOnLine* **2024**, *58*, 1−6.
(31) Maldonado, M.; Desbiens, A.; Del Villar, R.; Quispe, R. Towards the optimization of flotation columns using predictive control. *IFAC-PapersOnLine* **2007**, *40*, 75−80.
(32) Putz, E.; Cipriano, A. Hybrid model predictive control for flotation plants. *Miner. Eng.* **2015**, *70*, 26−35.
(33) Riquelme, A.; Desbiens, A.; Del Villar, R.; Maldonado, M. Predictive control of the bubble size distribution in a two-phase pilot flotation column. *Miner. Eng.* **2016**, *89*, 71−76.
(34) Brooks, K.; Koorts, R. Model Predictive Control of a Zinc Floation Bank Using Online X-ray Fluorescence Analysers. *IFAC-PapersOnLine* **2017**, *50*, 10214−10219.
(35) Quintanilla, P.; Navia, D.; Neethling, S. J.; Brito-Parada, P. R. Economic model predictive control for a rougher froth flotation cell using physics-based models. *Miner. Eng.* **2023**, *196*, 108050.
(36) Oosthuizen, D.; Le Roux, J. D.; Craig, I. K. A dynamic flotation model to infer process characteristics from online measurements. *Miner. Eng.* **2021**, *167*, 106878.
(37) Quintanilla, P.; Neethling, S. J.; Navia, D.; Brito-Parada, P. R. A dynamic flotation model for predictive control incorporating froth physics. Part I: Model development. *Miner. Eng.* **2021**, *173*, 107192.
(38) Quintanilla, P.; Neethling, S. J.; Mesa, D.; Navia, D.; Brito-Parada, P. R. A dynamic flotation model for predictive control incorporating froth physics. Part II: Model calibration and validation. *Miner. Eng.* **2021**, *173*, 107190.
(39) Brooks, K.; Ramodike, K.; Higginson, A. Controlling a Flotation Cell at the Peak Air Recovery Point. *IFAC-PapersOnLine* **2024**, *58*, 13−17.
(40) Rasmussen, C. E.; Williams, C. K. I. *Gaussian Processes for Machine Learning*; MIT Press, 2016.
(41) Darby, M. In *Encyclopedia of Systems and Control*; Baillieul, J., Samad, T., Eds.; Springer: London, 2013.
(42) Camacho, E.; Alba, C. *Model Predictive Control*; Springer, 2013.
(43) Dai, L.; Xia, Y.; Fu, M. S. M. *Advances in Discrete Time Systems*; InTech, 2012.
(44) Hewing, L.; Kabzan, J.; Zeilinger, M. N. Cautious Model Predictive Control Using Gaussian Process Regression. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 2736−2743.
(45) Neethling, S. J.; Cilliers, J. J. Modelling flotation froths. *Int. J. Miner. Process.* **2003**, *72*, 267−287.
(46) Neethling, S. J. Simple approximations for estimating froth recovery. *Int. J. Miner. Process.* **2008**, *89*, 44−52.
(47) Neethling, S. J.; Cilliers, J. J. The entrainment factor in froth flotation: Model for particle size and other operating parameter effects. *Int. J. Miner. Process.* **2009**, *93*, 141−148.
(48) Quintanilla, P.; Navia, D.; Moreno, F.; Neethling, S. J.; Brito-Parada, P. R. A methodology to implement a closed-loop feedback-feedforward level control in a laboratory-scale flotation bank using peristaltic pumps. *MethodsX* **2023**, *10*, 102081.
(49) Mesa, D.; Quintanilla, P.; Reyes, F. Bubble Analyser - An open-source software for bubble size measurement using image analysis. *Miner. Eng.* **2022**, *180*, 107497.
(50) Wang, F.; Chen, J. E. Efficient modeling of random fields by using Gaussian process inducing-point approximations. *Comput. Geotech* **2023**, *157*, 105304.
(51) GPyTorch Team, Metrics for Gaussian Process Models. https://docs.gpytorch.ai/en/v1.8.1/examples/00_Basic_Usage/Metrics.html, 2021; Accessed Sep 23, 2024.

https://doi.org/10.1021/acs.iecr.5c00660
Ind. Eng. Chem. Res. 2025, 64, 13307−13322