Online Learning-Based Predictive Control of Crystallization Processes under Batch-to-Batch Parametric Drift

Yingzhe Zheng, Tianyi Zhao, Xiaonan Wang, Zhe Wu*

March 30, 2022

Abstract

This work considers a seeded fesoterodine fumarate (FF) cooling crystallization and presents the methodology and implementation of a real-time machine learning modeling-based predictive controller to handle batch-to-batch (B2B) parametric drift. Specifically, an autoencoder recurrent neural network-based model predictive controller (AERNN-MPC) is developed to optimize product yield, crystal size, and energy consumption while accounting for the physical constraints on cooling jacket temperature. Deviations in the kinetic parameters are considered in the closed-loop simulations to account for the B2B parametric drift, and two error-triggered online update mechanisms are proposed to address issues pertaining to the availability of real-time crystal property measurements and are incorporated into the AERNN-MPC to improve the model prediction accuracy. Closed-loop simulation results demonstrate that the proposed AERNN-MPC with online update, irrespective of the accessibility to real-time crystal property data, achieves a desired closed-loop performance in terms of maximizing product yield and minimizing energy consumption.

Keywords: Model predictive control; Machine learning; Recurrent neural networks; Autoencoder; Monte Carlo simulation; Online learning

*Yingzhe Zheng is with the Department of Chemical and Biomolecular Engineering, National University of Singapore, 117585, Singapore. Tianyi Zhao is with the Department of Chemical and Biomolecular Engineering, National University of Singapore, 117585, Singapore, and the Joint School of National University of Singapore and Tianjin University, International Campus of Tianjin University, Binhai New City, Fuzhou 350207, China. Xiaonan Wang is with the Department of Chemical and Biomolecular Engineering, National University of Singapore, 117585, Singapore, and the Department of Chemical Engineering, Tsinghua University, Beijing 100084, China. Zhe Wu is with the Department of Chemical and Biomolecular Engineering, National University of Singapore, 117585, Singapore. Emails: yzzheng@nus.edu.sg; tianyi_z@u.nus.edu; wangxiaonan@mail.tsinghua.edu.cn; wuzhe@nus.edu.sg. Corresponding authors: X. Wang and Z. Wu.

Introduction

In pharmaceutical manufacturing, batch crystallization is typically regarded as one of the crucial unit operations as more than 90% of the active pharmaceutical ingredients are synthesized in the form of crystals.[2] Crystallization is the key operation that dictates the process performance (e.g., product yield) and the physical properties of the crystalline product (e.g., crystal size distribution (CSD), purity, shape), which directly govern process productivity, downstream processing (e.g., filtration, milling, tableting), and drug performance (e.g., bioavailability). Since the dynamic behaviors of the crystal properties are primarily influenced by the intricate interplay between the crystallization kinetics of growth, nucleation, agglomeration, and breakage, which are regulated by the process operating conditions, selecting the optimal operating and control strategies is thus of great interest to pharmaceutical manufacturers in achieving a more effective and eco-friendlier attainment of the specification targets.

Modern pharmaceutical manufacturing practice employs the quality-by-control (QbC) paradigm where model-based predictive control forms the core to improving process understanding and product assurance. Model predictive control (MPC) is an optimization-based advanced control method that solves for the optimal control actions based on a predictive model of the process while accounting for inherent process characteristics. A key challenge to the implementation of MPC in batch crystallization is the development of accurate and computationally efficient process models, which has been a long-standing research problem in predictive control of dynamic processes. Recently, attributing to their exceptional modeling performance for multi-dimensional dynamic systems, recurrent neural networks (RNN) have garnered substantial attention and are gaining traction in model identification of highly nonlinear chemical processes.[7,26,29] RNN is a special type of neural networks that is especially suitable for modeling time series data. It incorporates the dependencies between data points in a sequence by sending feedback signals in a directed loop, thereby exhibiting temporal dynamic behavior.[15] Additionally, autoencoder (AE) neural network, which is an unsupervised dimensionality reduction technique that minimizes modeling complexity while still maintaining an adequate model fidelity, can be integrated with RNN to further enhance its computational efficiency. Several recent works have utilized RNN and AE-based RNN (AERNN) as surrogate process models in MPC to control chemical plants in real time and optimize process performance accounting for closed-loop stability, safety, and control actuator constraints, which demonstrated superior computational efficiency as compared to using conventional first-principles model-based MPCs in their implementation to a diffusion-reaction process,[33] a batch crystallization process,[34] and a plasma etch process.[30]

However, although pretrained machine learning models have exhibited desirable computational properties and have shown to be promising substitutions for first-principles models in MPC, uncertainty in crystallization kinetics including intrinsic and exogenous uncertainties could be of a major concern to their applications to real chemical processes, which can result in performance deterioration of these MPC schemes over time.[13,28] In reality, process kinetic parameters may vary due to external (e.g., aging equipment, unknown disturbances) and internal (e.g., equipment fouling) factors.[28] Periodical update of the machine learning models based on newly acquired process data is thus imperative in ensuring a high model fidelity, and thereby enabling a more robust control. Exploiting the repetitive nature of batch processes, which are in conformity with a set of pre-established routine recipes, previous works have sought to circumvent issues pertaining to model-plant mismatch using iterative learning-based MPC strategies (e.g., batch-to-batch (B2B) or run-to-run (R2R) controls[9,12–14,19,25]) where process data obtained from previous batches are used to update the nominal model for predictive control of the next batch. A recent work proposed an integrated B2B-MPC control scheme based on a multiway partial least-squares (MPLS) model and a just-in-time learning-extended prediction self-adaptive control (JITL-EPSAC) technique,[22] which outperformed the conventional B2B control strategy by producing a smoother and more rapid convergence to product quality set points under the influence of parametric shifts. However, while the B2B iterative learning-based MPC schemes generally foster favorable performance improvements, the effects are only manifested in the next batch as learning within the current batch is not realizable since the process model is only updated inter- instead of intra-batches.[3,31]

Ascribing to the unknown B2B parametric drift and variation (e.g., due to equipment fouling, changes in impurity concentrations and flow rates, etc), process models constructed and updated using process knowledge harvested from previous batches may still be susceptible to significant model-plant mismatch especially for the current batch run, and hence resulting in sub-optimal control profiles. It is thus vital to incorporate real-time or online update of the process model within the MPC scheme to further enhance its performance. Attributing to the burgeoning research and development in many *in situ* process analytical technologies (PATs) for real-time monitoring of crystallization processes (e.g., attenuated total reflection Fourier transform infrared spectroscopy (ATR-FTIR) for real-time measurement of solute concentration,[32] and focused beam reflectance measurement (FBRM) and particle vision measurement (PVM) for real-time tracking of crystal size and shape, respectively, during the crystallization process[4,6,20]), online learning, and hence prompt update of process models using latest process data retrieved has become feasible. A recent work proposed an online model update strategy for an RNN-based MPC of a continuous stirred

tank reactor (CSTR) subject to time-varying disturbances, which yielded improved process dynamic performance with respect to closed-loop stability, optimality, and smoothness of control actions.[28] Another recent work exploits the dominant and low-rank features of a complex, high-dimensional, and nonlinear fluid system to control its flow properties using an RNN-based MPC where real-time sensor data are collected and used to update the RNN model to improve its prediction accuracy.[5] However, while online learning has been investigated for continuous nonlinear processes, its implementation in predictive control of batch crystallization processes under B2B parametric drift, to the best of authors' knowledge, has not been previously reported.

Motivated by the above considerations, in this manuscript, we develop a general framework for constructing machine learning models and developing machine learning-based predictive control schemes with error-triggered online model update strategies for batch crystallization processes under the influence of B2B parametric drift. Specifically, considering situations where real-time measurements of crystal properties may not be readily available, two online update schemes are proposed for the AERNN-based MPC, and are applied to the predictive control of a seeded fesoterodine fumarate (FF) batch cooling crystallization under parametric drift in order to account for the presence and absence of real-time crystal property measurements, respectively. The resulting closed-loop control performance is also compared with that of an AERNN-based MPC without online update to demonstrate the effectiveness of the two proposed online update strategies.

Modeling of Seeded Batch Cooling Crystallization

This section introduces the mathematical expressions utilized to describe the dynamics of the batch cooling crystallization system employed in this work. Specifically, the mechanistic model and the basic operating mode of the batch crystallizer are first presented, followed by an overview of the surrogate modeling approach adopted to approximate the computationally intensive mechanistic model using recurrent neural networks. We will then demonstrate that the neural network models significantly outperform the mechanistic model in terms of computational efficiency while maintaining a desired model fidelity and prediction accuracy.

Population Balance

Fesoterodine fumarate (FF), marketed as TOVIAZ® by Pfizer Inc., is a muscarinic antagonist prescribed to relieve overactive bladder symptoms.[18] One of its polymorphic forms (Form I), which resembles a cubic structure, may be isolated by seeded cooling crystallization from solutions in 2-butanone,[23] and the process dynamics can be described using the discretized formulation of the one-dimensional population balance equations (PBE) presented in Ref. 23 (Eqs. 1 - 2), where the crystallization mechanism and kinetics have been well investigated.

$$\frac{dN_1}{dt} = -\frac{N_1}{2w_1}G_1 + B_N - A_{D,1} \ , \quad \frac{dN_i}{dt} = -\frac{N_i}{2w_i}G_i + \frac{N_{i-1}}{2w_{i-1}}G_{i-1} + A_{B,i} - A_{D,i} \ , \quad \frac{dN_r}{dt} = \frac{N_{r-1}}{2w_{r-1}}G_{r-1} + A_{B,r} \quad (1)$$

$$\frac{dN_1}{dt} = -\frac{N_1}{2w_1}D_1 + \frac{N_2}{2w_2}D_2 \ , \quad \frac{dN_i}{dt} = -\frac{N_i}{2w_i}D_i + \frac{N_{i+1}}{2w_{i+1}}D_{i+1} \ , \quad \frac{dN_r}{dt} = -\frac{N_r}{2w_r}D_r \quad (2)$$

Method of classes has been employed to discretize the one-dimensional PBE with respect to both the temporal and spatial domains,[23] and the resulting system of r ordinary differential equations (Eqs. 1 - 2) entails a detailed quantitative description of the dynamic behaviors of the evolution of number of crystals N_i in size class i due to the coupling of all kinetic terms involved in the crystallization process (e.g., nucleation B_N, growth G, agglomeration A, dissolution D). Specifically, Eq. 1 accounts for the rates of change of crystal number N_i within each size class i through nucleation, crystal growth and agglomeration in a supersaturated system ($C > C_s$), whereas Eq. 2 describes the rates of change of number of crystals N_i due to crystal dissolution in an undersaturated system ($C < C_s$). Following the same discretization employed in Ref. 23, the number of size classes $r = 40$ is divided between 0.1 and 1000 μm with logarithmic progression of class borders. Eqs. 1 - 2 can be collectively solved numerically using standard integration methods (e.g., Runge-Kutta methods, finite difference method).

Specifically, the nucleation rate B_N is modeled by a power law dependence on supersaturation (Eq. 3), and occurs solely in the smallest size class (e.g., $i = 1$). B_N takes only secondary nucleation (i.e., due to seeding) into account as a very wide metastable zone was observed for the FF cooling crystallization process.[23] m_{cr}, V_{sol}, k_N, and n denote crystal mass, solution volume, nucleation rate constant, and nucleation order, respectively. k_N and n are the two kinetic parameters determined empirically in Ref. 23.

$$B_N = k_N m_{cr} V_{sol} (C - C_s)^n \quad (3)$$

Size-dependent crystal growth rate G_i is written as a power-law based expression with supersaturation as the overall driving force (Eq. 4). It is modeled using a two-step growth mechanism comprising mass transfer of solute from bulk solution to solid-liquid interface followed by mass integration into the crystal lattice. k_a, k_G, ρ_{sol}, η_i, ρ_{cr}, k_v, and g denote crystal area factor, growth rate constant, solution density, growth effectiveness factor for crystals in class i, crystal density, crystal volume factor, and growth order, respectively. k_G and g are the two unknown size-independent kinetic parameters that have been determined empirically in Ref. 23. η_i (Eq. 5) describes the relative importance of mass transfer and surface integration, and is a size-dependent parameter due to the effect of crystal size on mass transfer coefficient $k_{d,i}$. η_i can be computed using standard root-finding algorithms such as Newton's method.

$$G_i = \frac{k_a k_G \rho_{sol} \eta_i}{3\rho_{cr} k_v}(C - C_s)^g \tag{4}$$

$$\eta_i = \left(1 - \frac{k_G}{k_{d,i}}(C - C_s)^{g-1}\eta_i\right)^g \tag{5}$$

Size-dependent crystal dissolution rate D_i (Eq. 6) is expressed in a similar fashion as G_i (Eq. 4), with the power-law rate exponent equal to unity, however. It is mechanistically described as the opposite of G_i, where solute molecules disengage from the crystal lattice to the crystal-liquid interface before diffusing into the undersaturated bulk solution. k_D denotes the dissolution rate constant and has been determined experimentally in Ref. 23.

$$D_i = \frac{k_a k_D k_{d,i} \rho_{sol}}{3\rho_{cr} k_v (k_D + k_{d,i})}(C_s - C) \tag{6}$$

Finally, size-dependent crystal agglomeration birth rate $A_{B,i}$ (Eq. 7) and disappearance rate $A_{D,i}$ (Eq. 8) are modeled by a binary mechanism proposed in Ref. 23, where crystals in size classes j and k ($k \geq j$) conglomerate to form crystals of size class l where $l > k, j$. S_i denotes the average crystal size in class i. $\delta_{i,j}$, $\delta_{i,k}$, and $\delta_{i,k}$ are equal to unity when $i = j$ or $i = k$ or $i = l$, and 0 otherwise. $\beta_{j,k}$ denotes the agglomeration kernel and is expressed as Eq. 9, where k_A, P, ν denotes the agglomeration rate constant, power dissipation per unit mass, and kinematic viscosity, respectively. k_A is a kinetic parameter and its value has been determined experimentally in Ref. 23.

$$A_{B,i} = \sum_{m=1}^{m_{max}} \beta_{j,k} N_j N_k V_{sol} \left(\frac{S_j^3 + S_k^3}{S_l^3}\delta_{i,l}\right) \tag{7}$$

$$A_{D,i} = \sum_{m=1}^{m_{max}} \beta_{j,k} N_j N_k V_{sol} \left(\delta_{i,j} + \delta_{i,k}\right) \tag{8}$$

$$\beta_{j,k} = k_A G_k \left(S_j + S_k\right)^3 \left(\frac{P}{\nu}\right)^{1/2} \tag{9}$$

Interested readers are referred to Ref. 23 for a detailed discussion on the PBE formulation and the computation of crystallization kinetics and physicochemical properties of the crystallization mixture in the seeded batch cooling crystallization of Form I FF.

The values of all six kinetic parameters (k_N and n for nucleation (Eq. 3), k_G and g for crystal growth (Eq. 4), k_D for dissolution (Eq. 6), and k_A for agglomeration (Eq. 9)) are determined empirically in Ref. 23: $k_N = 1.82 \times 10^{15}$ (1/m^3 kg s), $n = 3.38$ ($-$), $k_G = 9.74 \times 10^{-8}$ (m/s), $g = 1.47$ ($-$), $k_A = 1.33 \times 10^{10}$ (s/m^7), $k_D = 2.92 \times 10^{-7}$ (m/s).

Mass balance

Total mass of the crystals formed m_{cr} (Eq. 10) is calculated by summing the mass of crystals within each size class i where the total volume of crystals formed is computed as the sum of product between number of crystals N_i and its corresponding average crystal size S_i.

$$m_{cr} = k_\nu \rho_{cr} \sum_{i=1}^{r} N_i S_i^3 \tag{10}$$

The bulk solute concentration $C(t)$ (Eq. 11) at a certain time t during the crystallization process is determined by subtracting $m_{cr}(t)$ at time t from the initial mass of dissolved solute m_0 and the mass of seed crystals m_{seed}, and dividing by the mass of solvent m_{sol}.

$$C(t) = \frac{m_0 + m_{seed} - m_{cr}(t)}{m_{sol}} \tag{11}$$

Energy balance

The rate of change of crystallizer temperature during cooling or heating is modeled by the ordinary differential equation of Eq. 12. Insignificant enthalpy of crystallization and ideal mixing with homogeneous temperature distribution are assumed for the crystallization process.

$$\frac{dT_r}{dt} = \frac{UA(T_j - T_r)}{m_r c_{p,r}} \tag{12}$$

where U denotes overall heat transfer coefficient, A denotes area of crystallizer wall available for heat transfer, T_j is jacket temperature, T_r is solution temperature, m_r and $c_{p,r}$ denote mass and specific heat capacity of the crystallization mixture, respectively.

Autoencoder-based RNN formulation

The mechanistic model (Eqs. 1 - 12) presented in the preceding sections is susceptible to high computational complexity, and is hence impractical to be incorporated in a real-time MPC controller. The RNN-based modeling approach thus forms the cornerstone to constructing computationally efficient surrogate models that well capture the dynamic behavior of the batch crystallization process, and can be readily employed in real-time optimization and control. In contrary to the one-way connectivity between neurons in conventional feedforward neural network models, signals in RNNs travel in bi-directions (i.e., forward and backward) as a result of introducing loops in the network. This facilitates the dynamic feedback of information obtained from earlier inputs into the network. In

addition, attributing to the universal approximation theorem of neural networks, it is demonstrated in Ref. 21 that an RNN with an appropriate architecture is capable of approximating the dynamics of any nonlinear system on compact subsets of the state space for finite time. RNN is therefore an ideal surrogate of the mechanistic model in describing the batch system of interest. Furthermore, the integration of autoencoder (AE), which is a special type of feedforward neural network that reduces the modeling complexity, with RNN could further aid its computational efficiency at a minimal loss of prediction accuracy and is remarkably instrumental in modeling large-scale chemical processes where a sizable number of process variables is involved.

Following the method in Ref. 34, the reduced-order RNN model is constructed using the simulation data generated from the discretized PBE in the previous sections to describe the dynamics of batch FF crystallization. Specifically, ascribing to the high-dimensional input (i.e., 43 dimensions consisting of C, T_r, T_j, and N_i, $\forall i = 1, ..., 40$) and output (i.e., 42 dimensions consisting of C, T_r, and N_i, $\forall i = 1, ..., 40$) state variables generated by the crystallization system, an AE is developed as a dimensionality reduction technique to enhance the computational efficiency of the RNN model. The structure of the AERNN model is illustrated in Fig. 1. AE is constructed as a conventional feedforward neural network that comprises two primary components: encoder and decoder functions. The encoder function $f_e(\cdot)$ maps an input data $\mathbf{x} \in \mathbf{R}^{d_x}$ to a reduced representation $\mathbf{x}_r \in \mathbf{R}^{d_{x_r}}$, and is mathematically expressed by Eq. 13. The decoder function $f_d(\cdot)$ reconstructs the output data in the original high-dimensional space from the reduced representation $\tilde{\mathbf{x}}_r$ and its mathematical formulation is presented by Eq. 14.

$$\mathbf{x}_r = f_e(\mathbf{x}) = \sigma_e(W_e\mathbf{x} + b_e) \tag{13}$$

$$\mathbf{y} = f_d(\tilde{\mathbf{x}}_r) = \sigma_d(W_d\tilde{\mathbf{x}}_r + b_d) \tag{14}$$

where $W_e \in \mathbf{R}^{d_{x_r} \times d_x}$ and $W_d \in \mathbf{R}^{d_y \times d_{\tilde{x}_r}}$ are the weight matrices, and $b_e \in \mathbf{R}^{d_{x_r}}$ and $b_d \in \mathbf{R}^{d_y}$ are the bias terms for $f_e(\cdot)$ and $f_d(\cdot)$, respectively. $\sigma_e(\cdot)$ and $\sigma_d(\cdot)$ are the activation functions (e.g., hyperbolic tangent and linear functions) for $f_e(\cdot)$ and $f_d(\cdot)$, respectively.

Subsequently, the RNN model is constructed based on input and output data residing in the low-dimensional latent space, and this effectively reduces the number of layers and neurons required to fit the data, thereby further minimizing its computation time. The reduced-order RNN model can be represented as the following continuous-time nonlinear system.[27,29]

$$\dot{\mathbf{x}}_r = F'_{nn}(\mathbf{x}_r, \mathbf{u}) := A_r\mathbf{x}_r + \Theta_r^T\mathbf{z}_r \tag{15}$$

where $\mathbf{x}_r \in \mathbf{R}^{d_{x_r}}$ is the reduced-order RNN state vector and $\mathbf{u} \in \mathbf{R}^{d_u}$ is the reduced-order RNN input vector. $\mathbf{z}_r = [z_1, \ldots, z_{d_{x_r}}, z_{d_{x_r}+1}, \ldots, z_{d_{x_r}+d_u}] = [\sigma(x_1), \ldots, \sigma(x_{d_{x_r}}), u_1, \ldots, u_{d_u}] \in \mathbf{R}^{d_{x_r}+d_u}$ is a vector of both the reduced network state $\mathbf{x}_r$ and the input $\mathbf{u}$, and $\sigma(\cdot)$ is the nonlinear activation function. A_r and Θ_r are the coefficient matrices consisting of reduced-order RNN weights.

Data generation

To construct an AERNN model with a desired accuracy, extensive open-loop simulations are first performed to generate a rich data set that well captures the system dynamics of the batch crystallization process under varying operating conditions. Specifically, the system of Eqs. 1 - 12 is solved numerically using the explicit Runge-Kutta method of order 5(4) with an integration time step $h_c = 1$ min. The simulation was conducted with three different initial conditions (m_{sol} and m_0 are maintained at 0.482 kg and 0.1 kg, respectively, for all initial conditions. m_{seed} used are 0.01, 0.005, and 0.005kg, Seed d_{50} are measured to be 4.2, 6.4, and 6.4μm, Seeding T are at 35, 30, and 35°C for initial condition 1, 2, and 3, respectively) subject to a sequence of pseudo-random jacket temperatures in a sample-and-hold fashion (i.e., the jacket temperature is supplied to Eq. 12 as a piecewise constant function, $T_j(t) \in [-10, 30]^\circ C$, and $T_j(t) = T_j(t_k),\ \forall t \in [t_k, t_{k+1})$, where $t_{k+1} := t_k + \Delta$ with a sampling period Δ of 30 mins). All other operating conditions (e.g., stirring rate) are maintained constant in each batch run.

With the abovementioned initial conditions, dynamic state trajectories ($C,\ T_r,\ N_i$) are collected at an integration time step h_c of 1 min within each sampling period Δ. Simulation data from a total of 1800 batch runs (600 for every initial condition), with each comprising 30 Δ (i.e., each batch run lasts 15 hrs), are captured. As a result, state trajectories from a total of 54,000 Δ are obtained, which are subsequently partitioned into training (60%), validation (10%), and testing (30%) datasets. It is imperative to acquire a plethora of simulation data as an exhaustive representation of the dynamic batch process under disparate operating conditions and system states (e.g., $T_r,\ C,\ N_i$) is instrumental in constructing an accurate data-driven model. The resulting AERNN model is constructed with a prediction horizon equal to $\Delta = 30$ mins.

Open-loop simulation results

The AERNN model (Eqs. 13 - 15), with $\mathbf{x} \in \mathbf{R}^{42}$ and $\mathbf{u} \in \mathbf{R}$ (i.e., an input data with 43 features), is developed using the deep learning application programming interface (API) Keras, to predict future state dynamics ($C,\ T_r,\ N_i$) for $\Delta = 30$ mins given the current state measurements and the manipulated input T_j. The reduced state $\mathbf{x}_r \in \mathbf{R}^{16}$ in Eq. 13 is the minimal number of dimensions required in the latent space to represent the 43 features in their original high-dimensional space. The resulting AERNN model achieved mean squared errors (MSEs) of 3.45×10^{-5} and

3.29×10^{-5} on validation and test datasets, respectively. Figure. 2 shows the comparison of T_r, C, and N_{23} predicted by the AERNN model to those captured from first-principles model (Eqs. 1-12) simulations, under the same pseudo-random jacket temperature profiles. The stars in Figure. 2 denote the start of a sampling period Δ for which the system states (C, T_r, N_i $\forall i = 1, ..., 40$) and the manipulated variable T_j are fed to the AERNN model to predict one sampling period Δ of 30 mins forward. The prediction is carried out recursively to predict the entire trajectory for the full batch of 900 mins. It is demonstrated in Figure. 2 that the AERNN prediction results are in close agreement with the results from the first-principles model under the same pseudo-random jacket profile, and the AERNN model is thus deemed suitable for approximating Eqs. 1 - 12 in describing the dynamics of the batch FF crystallization process.

In addition, it is noted that the computation time required to solve the first-principles model for each batch run (41.309 s) is substantially longer (at least 1000 times) than that of the AERNN model (0.032 s), and this thus renders the AERNN model a promising substitution for the first-principles model in real-time predictive control.

Error-Triggered Online Learning of AERNN

Industrial crystallization processes are typically vulnerable to unknown systematic B2B parametric drifts (e.g., due to changes in impurity concentrations of raw materials[13]), the presence of significant model-plant mismatch thus greatly hinders the control performance of MPC, thereby leading to sub-optimal attainment of the specification targets. Nominal machine learning predictive models developed based on process data garnered from previous batches may still deviate appreciably from the dynamics of the current batch, and on-the-fly model update is thus essential to achieving a desired model fidelity and hence a more robust control.

In general, traditional machine learning paradigms comprise batch or offline learning where the predictive model is trained on a large set of data before deployment, and post-deployment model update is often not considered or performed.[10] These learning techniques are commonly susceptible to poor scalability for real-world applications as the models are less amenable to changing process kinetics and rapidly evolving environmental conditions.[17] On the contrary, online learning is a fundamentally different approach to offline learning as it emphasizes on-the-fly and incremental training of the predictive models based on an inflow of new training data in a sequential manner. This in turn effectively overcomes the drawbacks associated with tradition offline learning algorithms as the predictive models are now updated in real time, and hence are more capable of predicting the process dynamics of the current batch run.

A naive implementation of the online learning algorithm stores and uses all process data obtained in the current batch to update the model in real time, and this inevitably leads to prohibitively large training data sets and computational infeasibility over time. It is therefore prudent to restrict the number of active training data to a finite set, and to trigger an online predictive model update only when the model-plant mismatch exceeds a specific threshold. In this section, we introduce two error-triggered online learning strategies by considering the accessibility to real-time crystal size and number measurements.

Error-triggered online learning of AERNN

In this subsection, two online learning triggering mechanisms will be introduced to account for scenarios where real-time measurements of some system states may not be readily accessible.

Real-time crystal number and size measurements are available

In the first case (denoted as case 1 in the following sections), we consider an AERNN-based MPC controller where real-time measurement of all system states (i.e., $C,\ T_r,\ N_i$) are available. Specifically, the error-triggered online learning mechanism is incorporated into the AERNN-MPC to improve the prediction accuracy of AERNN model by updating the model using the most recent closed-loop states data garnered under the influence of B2B parametric drifts. The triggering mechanism is based on the accumulative MSE between predicted states and measured states defined by a moving horizon error metric $E_{RNN}(t_k)$ (Eq. 16).[1]

$$E_{RNN}(t_k) = \sum_{i=0}^{N_\Delta} \frac{|\tilde{\mathbf{x}}'(t_{k-i}) - \mathbf{x}'(t_{k-i})|^2}{n} \tag{16}$$

where N_Δ denotes the number of sampling periods before t_k that are included in the computation of the accumulative prediction error. $\tilde{\mathbf{x}}'(t_{k-i}),\ i = 0, ..., N_\Delta$ are the normalized past state predictions by the AERNN model, and $\mathbf{x}'(t_{k-i})$ represent the normalized past state measurements from the actual crystallization system as modeled by Eqs. 1 - 12 subject to the same sequence of control actions. $n = 42$ denotes the number of states in $\tilde{\mathbf{x}}(t_{k-i})$ and $\mathbf{x}(t_{k-i})$ for calculating the MSE. It should be noted that only the RNN component (Eq. 15) of the AERNN model is updated as the AE component has minimal impact on the prediction accuracy. The update of RNN model (Eq. 15) is triggered only if the accumulative error $E_{RNN}(t_k)$ exceeds a predefined threshold E_T (Eq. 17).

$$E_{RNN}(t_k) \geq E_T \tag{17}$$

where $E_T = 0.015$ is determined via extensive closed-loop simulations by setting $N_\Delta = 5$ such that the update is triggered at appropriate times to achieve a balance between update frequency

and data-storage burden. After each update, E_{RNN} is reset to zero, and the state measurements stored are purged. Furthermore, as the newly acquired process data is likely to be insufficient for training a new RNN model of Eq. 15 from randomly initialized weights, the update is performed based on the weights of the pre-trained model and for a small number of epochs = 250. This fine-tuning strategy (i.e., using most recent data and 250 epochs only) avoids overfitting and ensures that the RNN model can be updated in less than one sampling period Δ, and the fined-tuned model is thus more capable of making accurate predictions of the system states under the influence of B2B parametric drift. In addition, it is noted that the updated model is not guaranteed to approximate the new system dynamics (i.e., under B2B parametric drift) in the entire operating region. The error-triggered mechanism therefore activates the model update continuously until a sufficient prediction accuracy is achieved.

Real-time crystal number and size measurements are unavailable

In the second case (denoted as case 2 in the following sections), we consider the scenario where real-time measurements are only available for a limited number of system states (i.e., C, T_r), and measurements of crystal number and size N_i are available only at specific intervals of the batch process. The error-triggered online learning strategy adopted in this case thus differs slightly to the one outlined in the previous section, where the missing and unmeasured dynamics of N_i are estimated using Monte Carlo simulations of the first-principles model (Eqs. 1 - 12) based on the control action T_j output from the predictive controller. Specifically, Monte Carlo is an easy and convenient algorithm that can be readily implemented to generate a vast amount of synthetic N_i trajectories by repeatedly solving the system of Eqs. 1 - 12 under varying kinetic parameter values sampled from their respective uniform distributions,[11] as shown in Table 1. The set of kinetic parameters that yields state trajectories which most coincide with the actual process states measured (e.g., real-time measurements obtained for C, T_r, and measurements of N_i obtained at regular intervals) is assumed to be representative of the actual dynamic process under B2B parametric drift, and is used to estimate the N_i trajectories in the missing intermediate time steps of the process which will be subsequently utilized for online update of the RNN model.

Additionally, in the time frame where measurements of N_i are unavailable, a partial error-triggered online update of the RNN model is executed to improve its prediction accuracy solely with respect to C and T_r, and when measurements of N_i become available at a particular time instance, a full online update of the RNN model, using the N_i trajectories selected from the pool of artificially generated trajectories and the real-time measurements of C and T_r, will be performed. Similarly, the update triggering mechanism employed during the time intervals where N_i data are

unobtainable also pertains to the notion of accumulative MSE between the predicted and measured states C and T_r, and the calculation of the moving horizon error metric $E'_{RNN}(t_k)$ (Eq. 18) resembles that of Eq. 16.

$$E'_{RNN}(t_k) = \sum_{i=0}^{N_\Delta} \frac{\left|\tilde{\mathbf{x}}'_{partial}(t_{k-i}) - \mathbf{x}'_{partial}(t_{k-i})\right|^2}{n_{partial}} \tag{18}$$

where the notations follow those in Eq. 16, with subscript *partial* denoting the variables for calculating the MSE in the partial error-triggered online update strategy. Specifically, $\tilde{\mathbf{x}}'_{partial}(t_{k-i})$ are the normalized past C and T_r predictions by the AERNN model. $\mathbf{x}'_{partial}(t_{k-i})$ denote the normalized past state C and T_r measurements acquired from the actual crystallization process modeled by Eqs. 1 - 12, and the dynamics of N_i are excluded in the computation. $n_{partial} = 2$ denotes the number of state variables involved in computing the MSE. Furthermore, N_Δ and the error threshold E'_T (Eq. 19) are specified at 5 and 0.015, respectively, as well.

$$E'_{RNN}(t_k) \geq E'_T \tag{19}$$

The actual measurements of N_i from the crystallization system are assumed to be obtainable at a regular time interval of 5 hrs (i.e., N_i measurements are available at the 5^{th} and 10^{th} hr of the 15 hr batch run), which conform to the typical industrial practices where quality variables of batch processes are measured every few hours.[8] The full online update of the RNN model is therefore only performed at the 5^{th} and 10^{th} hr within a batch run for a moderate number of epochs = 500 since more data are now available for training (relative to case 1).

Remark 1. Although real-time measurements of C and T_r can be readily obtained from sensors, online measurements of N_i are less straightforward. While FBRM, which operates on the principle of laser backscattering,[6,20] has been widely employed for online monitoring of crystal sizes, determination of crystal size distribution can be cumbersome as the output of FBRM is in the form of particle counts and chord length distribution (CLD), and has to be converted to CSD. A substantial number of works in recent years has sought to address this challenge by proposing a variety of approaches (e.g., empirical methods to establish the relationships between FBRM measurements and CSD[16]) to convert CLD to CSD for different types of particles. We therefore only consider scenarios where online measurements of N_i may be inaccessible as attaining real-time measurements of other system states is relatively more convenient.

Remark 2. It should be noted that the synthetic T_r and C profiles generated from Monte Carlo simulations are used solely for the purpose of ascertaining the set of kinetic parameters that is most representative of the actual dynamic process under consideration, and are not used in fine-tuning

the predictive model. The artificial N_i profiles, which are used to approximate the inaccessible N_i values at the various intermediate time steps, are employed in the online update of the predictive model, however.

Remark 3. Furthermore, depending on the number of Monte Carlo simulations executed, mismatch between the selected state trajectories of Monte Carlo simulation of the mechanistic model of Eqs. 1 - 12 under varying kinetic parameters and the actual dynamics of N_i may still exist since the first-principles model may not represent the actual dynamics with 100% accuracy. Nevertheless, Monte Carlo simulation is used in this work as a way to augment the training data by estimating system states in the missing intermediate time steps due to unavailability of real-time N_i measurements. Moreover, while the kinetic parameters are varied under Monte Carlo simulations based on the probabilistic distributions shown in Table 1, their values in Eqs. 1 - 12 are not updated at the end of the batch run since the first-principles model, due to its high computation complexity, is not incorporated in solving the MPC optimization problem.

Remark 4. In addition, we only consider scenarios where changes in the time-varying system dynamics are due to B2B parametric drift only. Other contributing factors, such as external disturbances from the upstream reactor (e.g., differing concentrations, flow rates, etc), which may render some of the assumptions made in formulating the mechanistic model of Eqs. 1 - 12 invalid, and hence results from the Monte Carlo simulations may deviate significantly from the real evolution of N_i. However, since T_r and C are measured in real time and are also compared with the system state profiles from Monte Carlo simulations, the results from Monte Carlo simulations can be regarded as good approximations when full information on N_i is unattainable.

AERNN-Based Predictive Control

In this section, an AERNN-based predictive control scheme constituting an online learning mechanism is developed to optimize the performance of the batch FF crystallization process. Specifically, we first present the mathematical formulation of the predictive control scheme in optimizing a number of process performance considerations (e.g., product yield, crystal size, and energy consumption), followed by introducing the error-triggered online update strategies for improving the prediction accuracy of the AERNN model in the presence and absence of real-time crystal size and number measurements, respectively. We will demonstrate that the proposed error-triggered online learning mechanisms considerably enhance the controller performance.

AERNN-MPC formulation

The AERNN-based predictive control scheme is formulated as a real-time optimization problem that calculates the optimal control action for the manipulated input T_j in achieving a number of process performance specifications such as maximizing product yield and crystal size while minimizing energy consumption. The formulation of the AERNN-based MPC (AERNN-MPC) is presented as follows:[34]

$$\min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{\mathbf{x}}, \mathbf{u}) dt \tag{20a}$$

$$\text{s.t.} \quad \dot{\tilde{\mathbf{x}}}(t) = f_d(F'_{nn}(\tilde{\mathbf{x}}_r(t), \mathbf{u}(t))) \tag{20b}$$

$$\tilde{\mathbf{x}}_r(t_k) = f_e(\mathbf{x}(t_k)) \tag{20c}$$

$$\mathbf{u}(t) \in U, \ \forall \ t \in [t_k, t_{k+N}) \tag{20d}$$

$$|\Delta \mathbf{u}| \leq U_\Delta, \ \forall \ t \in [t_k, t_{k+N}) \tag{20e}$$

where $\tilde{\mathbf{x}} = [C, T_r, N_i] \in \mathbf{R}^{42}$, $i = 1, ..., 40$ and $\mathbf{u} = T_j$ are the predicted states and manipulated input, respectively. $S(\Delta)$ is the set of piecewise constant functions with sampling period Δ. $L(\tilde{\mathbf{x}}, \mathbf{u}) = Q_1(C(t))^2 - Q_2(N_{class}(t))^2 + Q_3(T_j(t) - T_j^r)^2$ is the objective function that seeks to minimize the solute concentration C (i.e., maximizing product yield), maximizes the number of crystals in a particular size class of interest N_{class}, and minimizes the energy consumption which is formulated as the deviation of the jacket temperature T_j from room temperature ($T_j^r = 25°C$) over the prediction horizon $t \in [t_k, t_{k+N})$, where Q_1, Q_2, Q_3 are the coefficients for the various performance specifications to indicate their relative importance in the objective function. The AERNN model of Eq. 15 with decoder $f_d(\cdot)$ of Eq. 14 is used as the constraint of Eq. 20b. Specifically, the MPC converts the state measurements $x(t_k) = [C, T_r, N_i](t_k)$ received at the beginning of each sampling period Δ to the reduced states $\mathbf{x}_r(t_k)$ using the encoder $f_e(\cdot)$ of Eq. 13 as specified in the constraint of Eq. 20c. Subsequently, the AERNN model is invoked to make predictions of the future states $\tilde{\mathbf{x}}_r(t), t \in [t_k, \ t_{k+N})$ in latent space using $\tilde{\mathbf{x}}_r(t_k)$ and manipulated input $\mathbf{u}(t_k)$ before finally calling the decoder $f_d(\cdot)$ of Eq. 14 to convert $\tilde{\mathbf{x}}_r(t)$ back to the states $\tilde{\mathbf{x}}(t) = [C, T_r, N_i](t)$, $t \in [t_k, \ t_{k+N})$ in their original high-dimensional space. It is necessary to transform $\tilde{\mathbf{x}}_r(t)$ back to $\tilde{\mathbf{x}}(t)$ as the former hardly contains any comprehensible physical information.

Specifically, the number of crystals N_{20} in size class 20 (average size = 8.97 μm) is chosen to be the performance specification in the objective function of Eq. 20a as we demonstrate the implementation of the proposed AERNN-based predictive control strategy. The AERNN model of Eq. 15 is utilized as the predictive model in Eq. 20b, and the process state measurements at

the beginning of each sampling period Δ are fed to the AERNN model to predict the evolution of the system states for one Δ (Eq. 20c). Eq. 20d is the constraint on the operating range of jacket temperature imposed by the physical limitations on coolant supply, and Eq. 20e is the rate-of-change constraint on the jacket temperature to avoid physically implausible changes in the manipulated inputs. The jacket temperature is bounded by $U = [-10,\ 30]°C$, and $|\Delta u| = |T_j(t_{k+1}) - T_j(t_k)|$, $k = 0, 1, 2, ...$, is bounded by $U_\Delta = 5°C$.

In addition, we assume that online measurements of the state variables ($C,\ T_r$) are available at each sampling time, and the availability of real-time measurements of N_i is on a case-by-case basis. The AERNN-MPC optimization problem of Eq. 20 is solved for every sampling time with the new state measurements received by the controller, and applies only the first control action $u(t)$, $t \in [t_k, t_{k+1})$ to the crystallization system, which is represented by the first-principles model in Section Modeling of Seeded Batch Cooling Crystallization. As the AERNN model is capable of predicting future states for one sampling period Δ, it is invoked recursively to predict all the future states within the prediction horizon $t \in [t_k, t_{k+N})$. The MPC is solved using PyIpopt, which is a python connector to the IPOPT software package that is well-suited for solving large scale nonlinear optimization of continuous systems.[24] IPOPT stands for interior-point optimizer which utilizes the interior-point methods in solving linear and nonlinear convex optimization problems. The initial guess of T_j for the first sampling period is specified as 30 $°C$, and the initial guesses thereafter are taken to be the optimal values determined in the previous Δ.

Remark 5. Since larger particle sizes are almost always preferred to facilitate downstream processing, such as filtration, the MPC is thus configured to maximize N_{20} (average size = 8.97 μm) from the seed crystals (average size = 4.20 μm). The maximum attainable crystal size of FF is approximately 15 μm (but at a significantly lower crystal number N as compared with smaller size classes) as observed from the experimental data reported by.[23]

Integrating online learning of AERNN with MPC

Following the two online learning triggering mechanisms outlined in the previous section, this subsection presents the algorithms and steps involved in the online updates of the predictive model within the AERNN-MPC under case 1, and case 2, respectively.

Real-time crystal number and size N_i measurements are available

The implementation strategy (Algorithm 1) of online learning, in the event that real-time measurements of all system states are available (case 1), for AERNN-based MPC with an error-triggered online update mechanism is described as follows:

1. An initial AERNN model employed in the MPC of Eq. 20 is trained using simulation data obtained from extensive open-loop simulations for the system under the kinetic parameter values (original) shown in Table 2 (Section Data generation - Section Open-loop simulation results).

2. With an initial system states $C,\ T_r,\ N_i$ specified, the crystallization system of Eq. 1 - 12 is operated under AERNN-MPC in a sample-and-hold fashion with system states being continuously monitored and collected up to N_Δ. AERNN model update will be triggered only if Eq. 17 holds true and the optimal manipulated input $u(t)$ will be computed based on the updated AERNN model from the next sampling period Δ onward. The error-triggered online learning mechanism will not be activated as long as Eq. 17 is not fulfilled.

Fig. 6 illustrates a general set up of the closed-loop control by AERNN-MPC with an error-triggered online update mechanism, as described in Algorithm 1. Real-time state measurements will be actively monitored and stored in the database, and online update of the predictive model will only be triggered if Eq. 17 is satisfied. The size of the database should be sufficient to store sensor data collected up to 5 sampling period Δ.

Algorithm 1 Error-triggered online learning mechanism with real-time measurements of full system states (case 1)

Require: Import pre-trained AERNN model into MPC
  while batch system is running **do**
    if $E_{RNN} \geq E_T$ ($Eq.$ 17) **then**
      activate AERNN model update for 250 epochs
    else
      AERNN model update is not triggered
    end if
  end while

Real-time crystal number and size N_i measurements are unavailable

The implementation strategy (Algorithm 2) of online learning, in the event that measurements of N_i are only available at regular intervals (case 2), for AERNN-based MPC with an error-triggered online update strategy is described as follows:

1. An initial AERNN model employed in the MPC of Eq. 20 is trained using simulation data obtained from extensive open-loop simulations for the system under the parameter values (original) shown in Table 2 (Section Data generation - Section Open-loop simulation results).

2. With an initial system states C, T_r, N_i specified, the crystallization system of Eqs. 1 - 12 is operated under AERNN-MPC in a sample-and-hold fashion with C and T_r being continuously monitored and collected up to N_Δ. AERNN model partial update (fine-tuning with respect to C and T_r for 250 epochs) will be triggered if Eq. 19 holds true and the optimal manipulated input $u(t)$ will be computed based on the updated AERNN model from the next sampling period Δ onward. The error-triggered partial online learning mechanism will not be activated as long as Eq. 19 is not fulfilled.

3. At the initial sampling period, Monte Carlo simulations of the first-principles model (Eqs. 1 - 12) are run in parallel based on the optimal control action computed by AERNN-based MPC to generate a considerable number of synthetic C, T_r, and N_i trajectories under kinetic parameter values sampled from their respective uniform distributions shown in Table 1. The values of the kinetic parameters corresponding to each artificially generated N_i trajectory are also recorded.

4. In the ensuing sampling periods, Monte Carlo simulations of the first-principles model are conducted using only the kinetic parameters collected from the first sampling period and the optimal manipulated input computed for that particular sampling time. Furthermore, the absolute percentage error (APE) of the dynamics of C between the synthetic trajectories and the real-time measurements are computed, and any trajectories with APE $> 10\%$ are discarded, together with their corresponding kinetic parameters, and T_r and N_i profiles, from the database. This greatly reduces the computational burden of Monte Carlo simulations after successive sampling periods as the number of simulation run decreases after each sampling time.

5. It is assumed that actual measurements of N_i can be obtained at a regular interval of 5 hrs or 2 N_Δ. Specifically, at the 5^{th} and 10^{th} hr timestamp of the batch run, absolute errors of all system states (e.g., C, T_r and N_i measurements obtained at their respective time instances) between the actual measurements and the synthetic trajectories (corresponding to each set of kinetic parameter values) are calculated, and the state trajectories, which are generated from a specific set of kinetic parameter values, that most coincide (i.e., result in the smallest error) with the actual process states measured are assumed to be representative of the actual state profiles and can be used to estimate the unmeasured dynamics of N_i in the intermediate time steps of the process. The selected artificial N_i trajectories, together with the actual real-time measurements of C and T_r, will be subsequently incorporated in the full online update of the RNN model. It should be noted that neither kinetic parameters nor their corresponding

system dynamics are discarded at the two time instances where a full online update of the predictive model is performed.

The general set up of the closed-loop control by AERNN-MPC with an error-triggered online update mechanism under case 2, as described in Algorithm 2, is similar to the one illustrated by Fig. 6 except that the N_i measurements are only accessible at specific time intervals rather than real time. Other system states will be measured in real time and the data collected are stored in the database. Partial online update of the predictive model (with respect to C and T_r) will only be triggered if Eq. 19 is satisfied. The size of the database should be sufficient to store sensor data collected up to 10 sampling periods Δ and the synthetic state trajectories generated from 5000 Monte Carlo simulations.

Remark 6. A good balance has to be achieved between the number of Monte Carlo simulations executed and the total computation time required (e.g., a higher number of Monte Carlo simulations corresponds to higher computational costs but has a better probability of yielding state profiles that are in close agreement with the actual measurements) . The number of simulations (5000) to be performed in the first sampling period is determined via extensive closed-loop simulations such that all Monte Carlo simulations, when run in parallel with MPC, can be completed within one sampling period Δ.

Closed-Loop Simulation Results Under AERNN-MPC

The closed-loop simulation results for the crystallization system of Eqs. 1 - 12 subject to a B2B parametric drift under AERNN-based MPC of Eq. 20 with an error-triggered online update mechanism are shown in Figs. 3a - 4b. Specifically, we consider two scenarios for the implementation of the online learning strategy: N_i measurements are 1) available in real-time (case 1), and 2) available only at regular time intervals (case 2). The relative importance of the process specification targets are indicated through the coefficients $Q_1 = 20$, $Q_2 = 10$, and $Q_3 = 1$ in the control objectives of Eq. 20a, where minimizing C has the highest priority while minimizing energy consumption has the least importance. It should be noted that, in simulating the impact of a B2B parametric drift, the new kinetic parameter values are deliberately chosen to deviate considerably from the original values used in open-loop simulations, as shown in Table 2, in order to make apparent the efficacy of incorporating error-triggered online update mechanisms into the AERNN-MPC. The initial conditions used in the closed-loop simulation are: $m_{sol} = 0.482$kg, $m_0 = 0.1$kg, $m_{seed} = 0.01$kg, Seed $d_{50} = 4.2\mu$m, and Seeding $T = 35°$C, at a batch time of 15 hrs or 6 N_Δ. With the aforementioned specifications, closed-loop simulations are performed for the seeded batch cooling crystallization of

Algorithm 2 Error-triggered online learning mechanism with real-time measurements of partial system states (case 2)

Require: Import pre-trained AERNN model into MPC
iteration $\leftarrow 1$
while batch system is running **do**
if iteration $== 1$ **then**

1. Execute 5000 Monte Carlo simulations of Eqs. 1 - 12 in parallel under varying kinetic parameter values sampled from the random distributions shown in Table 1 and the optimal manipulated input calculated
2. Record the artificially generated state trajectories and their corresponding kinetic parameters

else if iteration % 10 $== 0$ **then**

1. Compute the absolute errors of all system states (i.e., C, T_r and N_i) between the actual measurements and the synthetic trajectories (corresponding to each set of kinetic parameter values) at the time instances where the actual measurements are acquired.
2. Identify the set of kinetic parameters that yields the least absolute error
3. Activate full online update of RNN model for 500 epochs using N_i profiles generated from the set of kinetic parameters determined in the previous step, together with actual real-time measurements of C and T_r, and including all dynamic data collected (both the artificial N_i and actual measurements of C and T_r) in the preceding 9 sampling periods

else
if $E'_{RNN}(t_k) \geq E'_T$ $(Eq.\ 19)$ **then**
activate AERNN model partial update for
250 epochs
end if

1. Calculate the APE of the dynamics of C between the synthetic trajectories and the real-time measurements, and any trajectories with $APE > 10\%$ are discarded, together with their corresponding kinetic parameters, and T_r and N_i profiles, from the database

end if
iteration $\leftarrow$ iteration + 1
end while

FF under AERNN-MPC with the error-triggered online update mechanisms detailed in the preceding section for case 1 (Algorithm 1), and case 2 (Algorithm 2), respectively. Their control performances are compared with that of the AERNN-MPC without online update to demonstrate the effectiveness of the proposed online learning schemes.

Fig. 3a shows the closed-loop solute concentration (top figure) and the number of crystals N_{20} in size class 20 (bottom figure) profiles under 1) AERNN-MPC with online update triggered by Eq. 17 where real-time measurements of full system states are available (denoted by AERNN_update_realtime), 2) AERNN-MPC with online update triggered by Eq. 19 where measurements of N_i are available only at regular time intervals (denoted by AERNN_update_periodic), and 3) AERNN-MPC without online update mechanism (denoted by AERNN_without_update), respectively. It is demonstrated that AERNN-MPC with the online update schemes, irrespective of the availability of real-time N_i measurements, outperforms AERNN_without_update in terms of meeting the control objectives of Eq. 20a (i.e., minimizing energy consumption while contemporaneously maximizing yield and the number of crystals N_{20} in class size 20). The relative importance of the different process specifications are dictated by the magnitude of the coefficient values with minimizing C having the highest importance, and minimizing utility cost having the lowest priority. It is apparent that AERNN_update_realtime and AERNN_update_periodic yielded better control performances than that of AERNN_without_update, with the two former MPC schemes attaining lower C values at the end of the batch run, as illustrated in Fig. 3a. The final C achieved by the MPCs are 0.098, 0.106, and 0.139 for AERNN_update_realtime, AERNN_update_periodic, and AERNN_without_update, respectively. The final N_{20} attained are 4.81×10^9, 5.34×10^9, and 7.50×10^9 for AERNN_update_realtime, AERNN_update_periodic, and AERNN_without_update, respectively.

Furthermore, it is noticed in Fig. 3b that control actions T_j output from AERNN_without_update stay relatively constant at 30 $^\circ C$ for the entire duration of the batch run, and more variations are observed for the control outputs from AERNN_update_realtime and AERNN_update_periodic, respectively. This can be attributed to the prediction errors of the AERNN models embedded in the various MPCs, as shown in Fig. 4a and Fig. 4b. Fig. 4a shows the accumulative errors E_{RNN} for AERNN_update_realtime and AERNN_without_update, respectively, at every sampling period of the batch run. Specifically, at the first sampling period, the two MPCs yielded the same E_{RNN}, with AERNN_update_realtime promptly triggering an online update using the full system states collected from the first MPC iteration. Subsequently, with a few additional updates triggered for the AERNN model, the E_{RNN} of AERNN_update_realtime is almost always substantially lower

than that of AERNN_without_update, as shown in Fig. 4a. Moreover, the prediction accuracy of the AERNN model in MPC under AERNN_update_realtime improved significantly toward the latter half of the batch run as online update is only triggered after 4 to 5 MPC iterations (e.g., up to N_Δ) following the 10^{th} sampling period, which is in contrast to the triggering frequency of every 1 to 2 MPC iterations observed prior to the 10^{th} sampling period.

The optimal cooling trajectory, as observed in Fig. 3b, follows a cyclic pattern where cooling and heating are alternated. This control profile can be ascribed to the process targets specified in the objective function of Eq. 20a and the evolution of the state dynamics shown in Fig. 3a. With the onset of crystallization of FF triggered by the addition of seed crystals, an initial decrease in T_j is observed in Fig. 3a for time < 100 mins as the MPC controller attempts to maximize the driving force for crystallization, which is accompanied by a decline in C and an incipient surge in N_{20}, and these dynamic behaviors are in favor of the specification targets of Eq. 20a. As the crystallization proceeds further and with further cooling of the system below room temperature T_j^r of $25°C$, the rate of increase in N_{20} abates, and the MPC controller thus increases T_j back to T_j^r to minimize energy consumption, as observed between the 100 mins and 200 mins interval in Fig. 3a, thereby attaining an optimal balance between the different process targets specified (i.e., maximizing N_{20}, and minimizing C and deviation of T_j from T_j^r). Beyond the 200 mins mark, the crystallization system begins to establish a trade-off between minimizing C and maximizing N_{20}, where further cooling decreases C at the expense of N_{21}. A cyclic temperature profile is thus preferred to balance the rates of decline in both C and N_{20} in an effort to achieve the optimal trade-off between the two process targets. Furthermore, in contrast to T_j fluctuating around a temperature of T_j^r in the first half of the batch simulation, the cyclic temperature profile oscillates at lower temperatures toward the latter half. This can be explained by the relative significance of the process targets specified in the objective function of Eq. 20a, where minimizing C takes precedence over the other two specifications, and hence a general descending trend of T_j is observed as the MPC controller minimizes C at the expense of the other two targets. Therefore, in the scenario where real time measurements of full system states are available, MPC performance is improved considerably by incorporating the online update mechanism to the AERNN model (AERNN_update_realtime), which yielded an enhanced control performance and attainment of the control objectives as compared with MPC without the online model update (AERNN_without_update).

Similar to Fig. 4a, Fig. 4b shows the accumulative errors E'_{RNN} for AERNN_update_periodic and AERNN_without_update, respectively, at every sampling period of the batch run. The trends observed resemble the ones in Fig. 4a, where accumulative errors E'_{RNN} are identical for both

AERNN_update_periodic and AERNN_without_update up to the MPC iteration that triggers the first online update for AERNN_update_periodic (e.g., the 2$^{\text{nd}}$ iteration). Subsequently, post the first update of the AERNN model, E'_{RNN} of AERNN_update_periodic is almost always substantially lower than that of AERNN_without_update at every MPC iteration, with the update triggering frequency increases to 4 to 5 MPC iterations, as shown in Fig. 4b. Contrary to E_{RNN} calculated in case 1, which is based on real-time measurements of the full system states (C, T_r, N_i), E'_{RNN} computed in case 2 is derived from two system states (C, T_r) only. Therefore, attributing to the low number of system states involved, the prediction accuracy of the AERNN model in case 2, with respect to C and T_r, improves noticeably just after the first online update, as opposed to the AERNN model in case 1 where its prediction performance is evaluated against the full system states, and hence requiring additional updates before a significant improvement can be realized.

Moreover, it should be noted that, as observed in Fig. 3b and Fig. 4b, although it is apparent that the prediction performance of the AERNN model in AERNN_update_periodic improves with the batch run, it is measured against two system states (C, T_r) only, and the inadequacy of its prediction with respect to N_i engenders sub-optimal control actions T_j output from the MPC controller at time < 300 mins with both AERNN_update_periodic and AERNN_without_update having similar T_j profiles (Fig. 3b). It is only upon the online update of the AERNN model at the 300 mins mark using synthetic state profiles generated from Monte Carlo simulations that most match with the actual state measurements garnered, AERNN_update_periodic starts to output similar control profiles as that of AERNN_without_update, indicating that the model prediction performance, with respect to the full system states, improved tremendously. Therefore, in the scenario where full system state measurements are not readily available in real time, the error-triggered online update of the AERNN model using only partial system states (outlined in Algorithm 2) is inefficacious, and optimal control actions are only guaranteed when a high-fidelity AERNN predictive model is incorporated into the MPC controllers.

Fig. 5a and Fig. 5b illustrate the synthetic trajectories of the state variable C from Monte Carlo simulations. Specifically, Fig. 5a shows several artificial C profiles generated from Monte Carlo simulations, and only the profiles which fall consistently within a deviation (measured by APE) of 10% from the actual measurements of C will be recorded, other trajectories will be discarded immediately once they are found to lie outside of the 10% threshold. This effectively reduces the overall computational burden as the number of Monte Carlo simulations performed decreases with the batch run. Fig. 5b shows the best matching profile among the candidate trajectories (e.g., synthetic profiles that fall within the 10% threshold), which achieved the least absolute error based

on comparison with the real-time measurements of C and T_r, and periodic measurements of N_i. The synthetic N_i profiles, corresponding to the ones that most coincide with the real-time and periodic measurements of the actual system states, will be used to approximate the missing N_i at the various intermediate time steps and utilized for the online update of the RNN model at regular time intervals to enhance its prediction accuracy. Through the closed-loop simulation study, we demonstrate that the proposed error-triggered online update strategies, as outlined in Algorithm 1 and Algorithm 2 for AERNN_update_realtime and AERNN_update_periodic, respectively, are able to accommodate the various process specifications and achieve desired closed-loop performances in terms of product yield, crystal size, and energy consumption as compared to AERNN_without_update.

Conclusion

In this work, we developed an AERNN model for describing the dynamics of a batch cooling crystallization process, and designed an AERNN-based MPC with an online learning mechanism to optimize product yield, crystal size, and energy consumption of the crystallization operation in real time under the influence of B2B parametric drift. Specifically, two error-triggered online update mechanisms were proposed to account for scenarios where only a limited number of system states is accessible. Closed-loop simulation results under deviated kinetic parameters demonstrated that the proposed AERNN-MPC with online update, regardless of the accessibility to full system states, achieved a more satisfactory closed-loop control performance when compared with the MPC without online update.

Acknowledgments

Financial support from Pfizer Asia Manufacturing PTE. LTD. and NUS Start-up Grant R-279-000-656-731 is gratefully acknowledged.

Literature Cited

1. A. Alanqar, H. Durand, and P. D. Christofides. Error-triggered on-line model identification for model-based feedback control. AIChE Journal, 63(3):949–966, 2017.

2. A. J. Alvarez and A. S. Myerson. Continuous plug flow crystallization of pharmaceutical compounds. Crystal Growth & Design, 10(5):2219–2228, 2010.

3. T. Ao, X. Dong, and Z. Mao. Batch-to-batch iterative learning control of a batch polymerization process based on online sequential extreme learning machine. Industrial & Engineering Chemistry Research, 48(24):11108–11114, 2009.

4. L. Barros, M. Gim-Krumm, G. Seriche, M. Quilaqueo, C. Castillo, C. F. Ihle, R. Ruby-Figueroa, and H. Estay. In-situ and real-time aggregation size evolution of copper sulfide precipitates using focused beam reflectance measurement (FBRM). Powder Technology, 380:205–218, 2021.

5. K. Bieker, S. Peitz, S. L. Brunton, J. N. Kutz, and M. Dellnitz. Deep model predictive control with online learning for complex physical systems. arXiv preprint arXiv:1905.10094, 2019.

6. A. Chanda, A. M. Daly, D. A. Foley, M. A. LaPack, S. Mukherjee, J. D. Orr, G. L. Reid III, D. R. Thompson, and H. W. Ward. Industry perspectives on process analytical technology: tools and applications in api development. Organic Process Research & Development, 19(1):63–83, 2015.

7. E. Chee, W. C. Wong, and X. Wang. An integrated approach for machine-learning-based system identification of dynamical systems under control: application towards the model predictive control of a highly nonlinear reactor system. Frontiers of Chemical Science and Engineering, pages 1–14, 2021.

8. L. H. Chiang, R. Leardi, R. J. Pell, and M. B. Seasholtz. Industrial experiences with multivariate statistical analysis of batch process data. Chemometrics and Intelligent Laboratory Systems, 81(2):109–119, 2006.

9. M. W. Hermanto, R. D. Braatz, and M. Chiu. Integrated batch-to-batch and nonlinear model predictive control for polymorphic transformation in pharmaceutical crystallization. AIChE Journal, 57(4):1008–1019, 2011.

10. S. C. H. Hoi, D. Sahoo, J. Lu, and P. Zhao. Online learning: A comprehensive survey. Neurocomputing, 459:249–289, 2021.

11. D. P. Kroese, T. Brereton, T. Taimre, and Z. I. Botev. Why the Monte Carlo method is so important today. Wiley Interdisciplinary Reviews: Computational Statistics, 6(6):386–392, 2014.

12. J. S. Kwon, M. Nayhouse, and P. D. Christofides. Detection and isolation of batch-to-batch parametric drift in crystallization using in-batch and post-batch measurements. Industrial & Engineering Chemistry Research, 54(20):5514–5526, 2015.

13. J. S. Kwon, M. Nayhouse, G. Orkoulas, D. Ni, and P. D. Christofides. A method for handling batch-to-batch parametric drift using moving horizon estimation: application to run-to-run mpc of batch crystallization. Chemical Engineering Science, 127:210–219, 2015.

14. J. S. Kwon, M. Nayhouse, G. Orkoulas, D. Ni, and P. D. Christofides. Run-to-run-based model predictive control of protein crystal shape in batch crystallization. Industrial & Engineering Chemistry Research, 54(16):4293–4302, 2015.

15. Q. V. Le. A tutorial on deep learning part 2: Autoencoders, convolutional neural networks and recurrent neural networks. Google Brain, 20:1–20, 2015.

16. H. Li, M. A. Grover, Y. Kawajiri, and R. W. Rousseau. Development of an empirical method relating crystal size distributions and fbrm measurements. Chemical Engineering Science, 89:142–151, 2013.

17. M. Maiworm, D. Limon, and R. Findeisen. Online learning-based model predictive control with gaussian process models and stability guarantees. International Journal of Robust and Nonlinear Control, 31(18):8785–8812, 2021.

18. M. C. Michel. Fesoterodine: a novel muscarinic receptor antagonist for the treatment of overactive bladder syndrome. Expert Opinion on Pharmacotherapy, 9(10):1787–1796, 2008.

19. S. Oh and J. M. Lee. Iterative learning model predictive control for constrained multivariable control of batch processes. Computers & Chemical Engineering, 93:284–292, 2016.

20. S. Sankaranarayanan, B. Likozar, and R. Navia. Real-time particle size analysis using the focused beam reflectance measurement probe for in situ fabrication of polyacrylamide–filler composite materials. Scientific reports, 9(1):1–12, 2019.

21. A. M. Schäfer and H. Zimmermann. Recurrent neural networks are universal approximators. International Journal of Neural Systems, 17(04):253–263, 2007.

22. Q. Su, M. Chiu, and R. D. Braatz. Integrated B2B-NMPC control strategy for batch/semibatch crystallization processes. AIChE Journal, 63(11):5007–5018, 2017.

23. M. Trampuž, D. Teslić, and B. Likozar. Crystallization of fesoterodine fumarate active pharmaceutical ingredient: Modelling of thermodynamic equilibrium, nucleation, growth, agglomeration and dissolution kinetics and temperature cycling. Chemical Engineering Science, 201:97–111, 2019.

24. A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical programming, 106(1):25–57, 2006.

25. Y. Wang, F. Gao, and F. J. Doyle III. Survey on iterative learning control, repetitive control, and run-to-run control. Journal of Process Control, 19(10):1589–1600, 2009.

26. W. C. Wong, E. Chee, J. Li, and X. Wang. Recurrent neural network-based model predictive control for continuous pharmaceutical manufacturing. Mathematics, 6(11):242, 2018.

27. Z. Wu and P. D. Christofides. Process Operational Safety and Cybersecurity. Springer, 2021.

28. Z. Wu, D. Rincon, and P. D. Christofides. Real-time adaptive machine-learning-based predictive control of nonlinear processes. Industrial & Engineering Chemistry Research, 59(6):2275–2290, 2019.

29. Z. Wu, A. Tran, D. Rincon, and P. D. Christofides. Machine learning-based predictive control of nonlinear processes. part i: theory. AIChE Journal, 65(11):e16729, 2019.

30. T. Xiao, Z. Wu, P. D. Christofides, A. Armaou, and D. Ni. Recurrent neural-network-based model predictive control of a plasma etch process. Industrial & Engineering Chemistry Research, 61(1):638–652, 2021.

31. Z. Xiong and J. Zhang. A batch-to-batch iterative optimal control strategy based on recurrent neural network models. Journal of Process Control, 15(1):11–21, 2005.

32. F. Zhang, T. Liu, X. Z. Wang, J. Liu, and X. Jiang. Comparative study on ATR-FTIR calibration models for monitoring solution concentration in cooling crystallization. Journal of Crystal Growth, 459:50–55, 2017.

33. T. Zhao, Y. Zheng, J. Gong, and Z. Wu. Machine learning-based reduced-order modeling and predictive control of nonlinear processes. Chemical Engineering Research and Design, 179:435–451, 2022.

34. Y. Zheng, X. Wang, and Z. Wu. Machine learning modeling and predictive control of batch crystallization process. Industrial & Engineering Chemistry Research, in press, 2022.

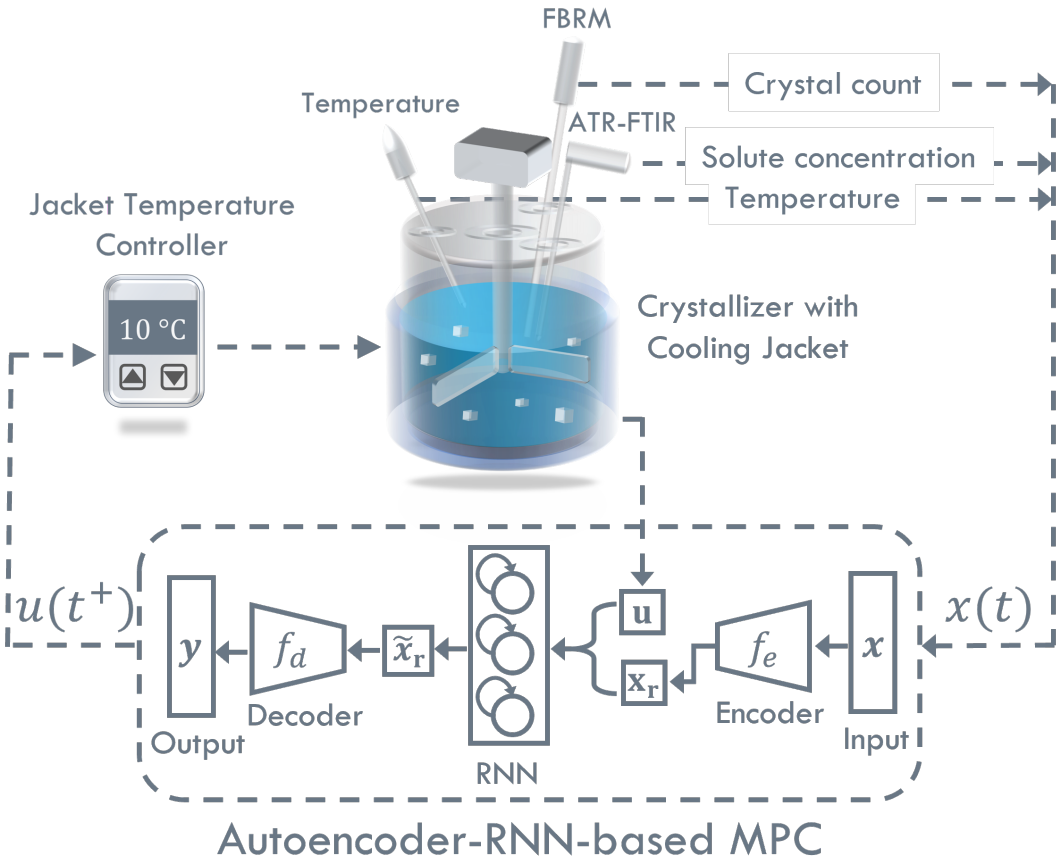


Figure 1: Autoencoder-based recurrent neural network structure employed in predictive control of a batch crystallization process under MPC.

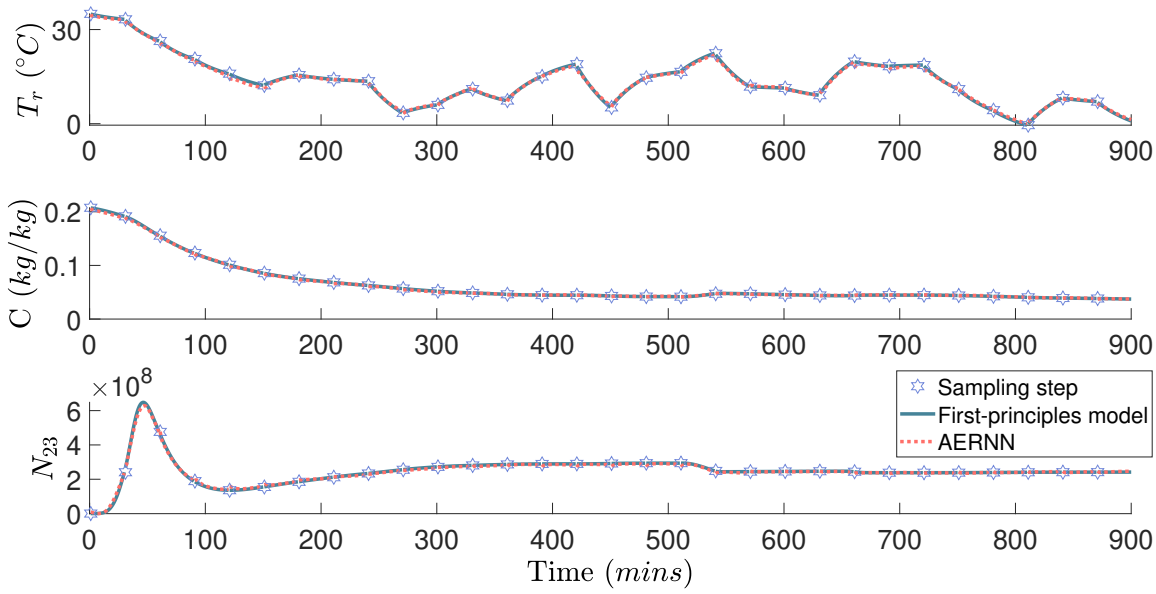
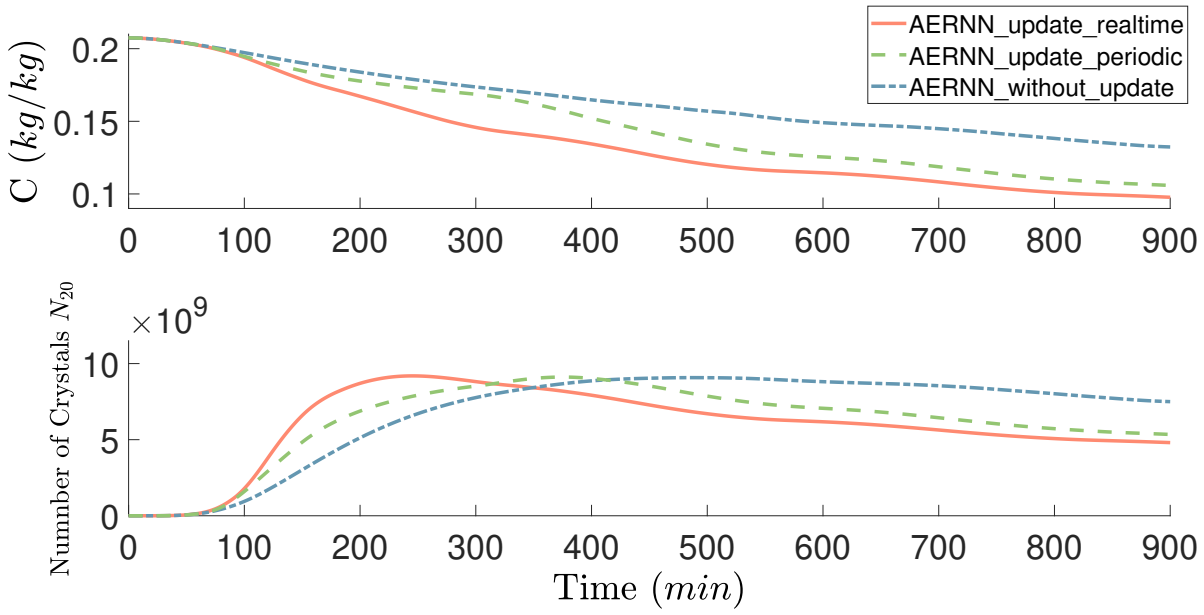
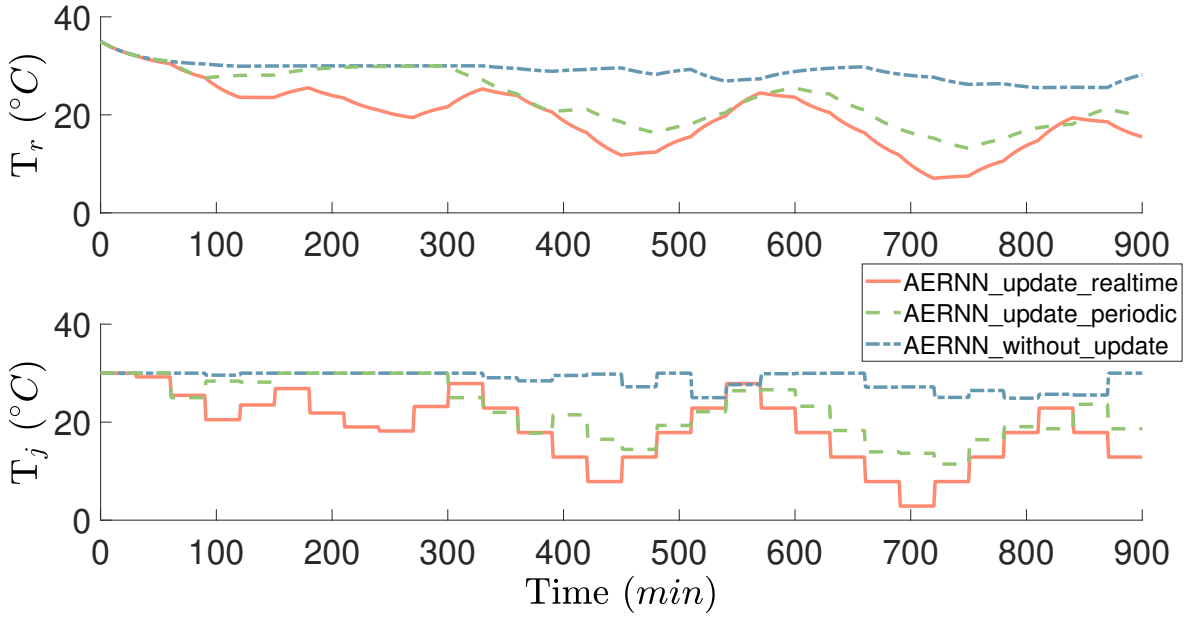


Figure 2: Comparison of crystallizer temperature T_r (top figure), solute concentration C (middle figure), and crystal number N_{23} of size class 23 (bottom figure) predicted by the AERNN model, and the first-principles model, respectively, where the stars denote the beginning of a sampling period Δ of 30 mins.

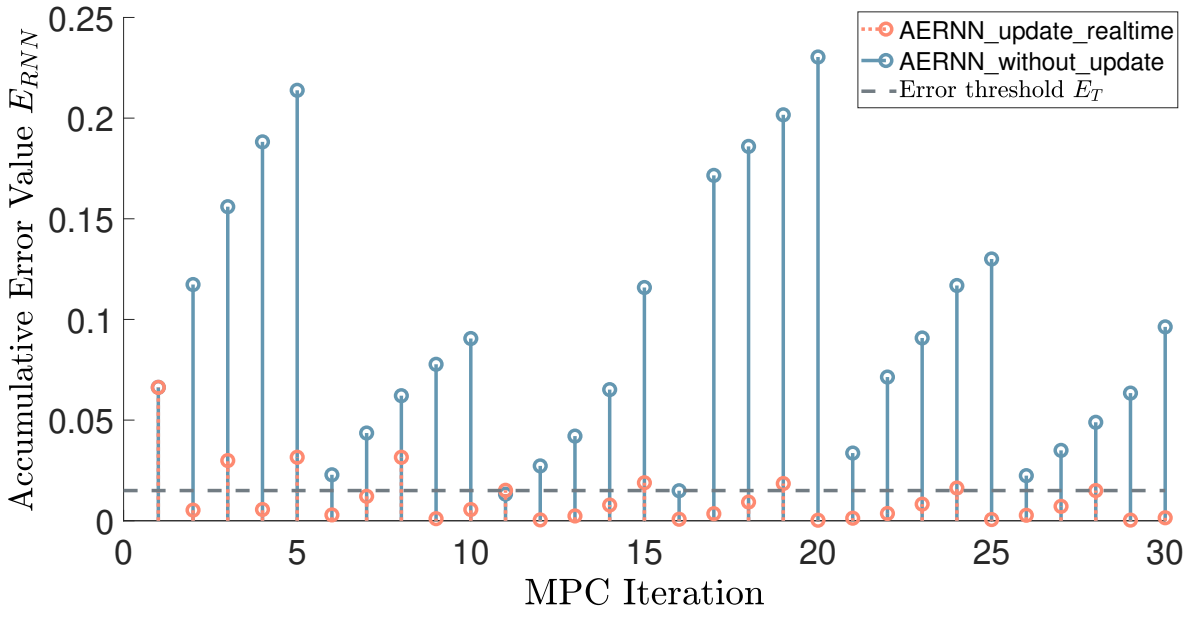


(a)

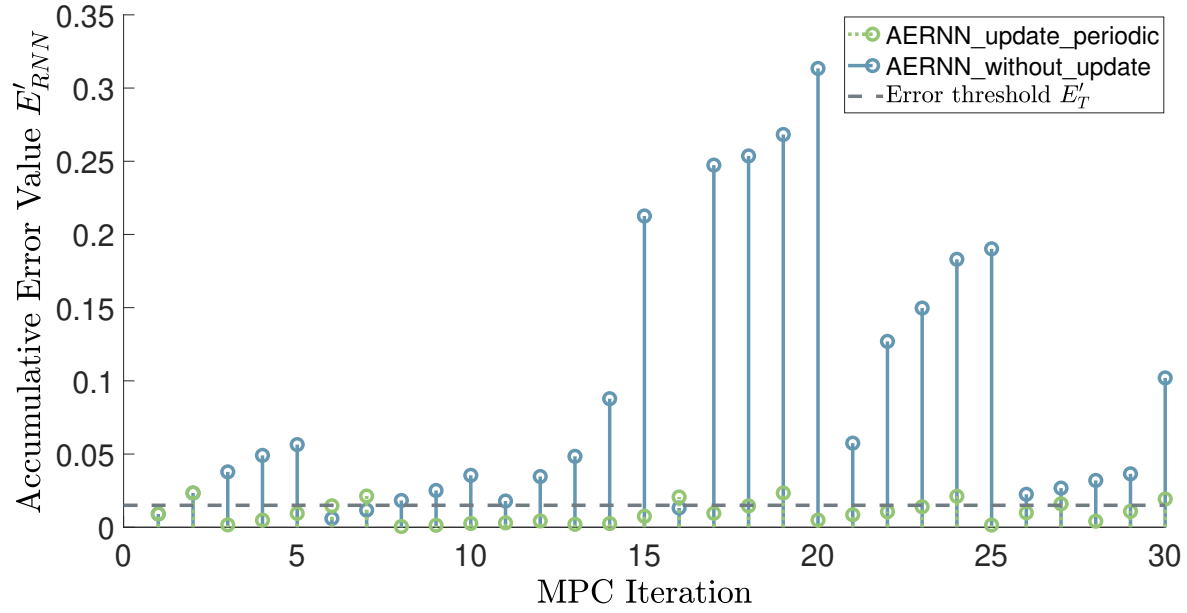


(b)

Figure 3: Comparison of closed-loop solute concentration profiles ((a) top figure), the number of crystals of average size 8.97 μm ((a) bottom figure), crystallizer temperature profiles ((b) top figure), and jacket temperature profiles ((b) bottom figure) under AERNN-MPC with error-triggering and online update mechanism with real-time full system state measurements (AERNN_update_realtime), without real-time measurements of N_i (AERNN_update_periodic), and without the online update mechanism (AERNN_without_update), respectively.

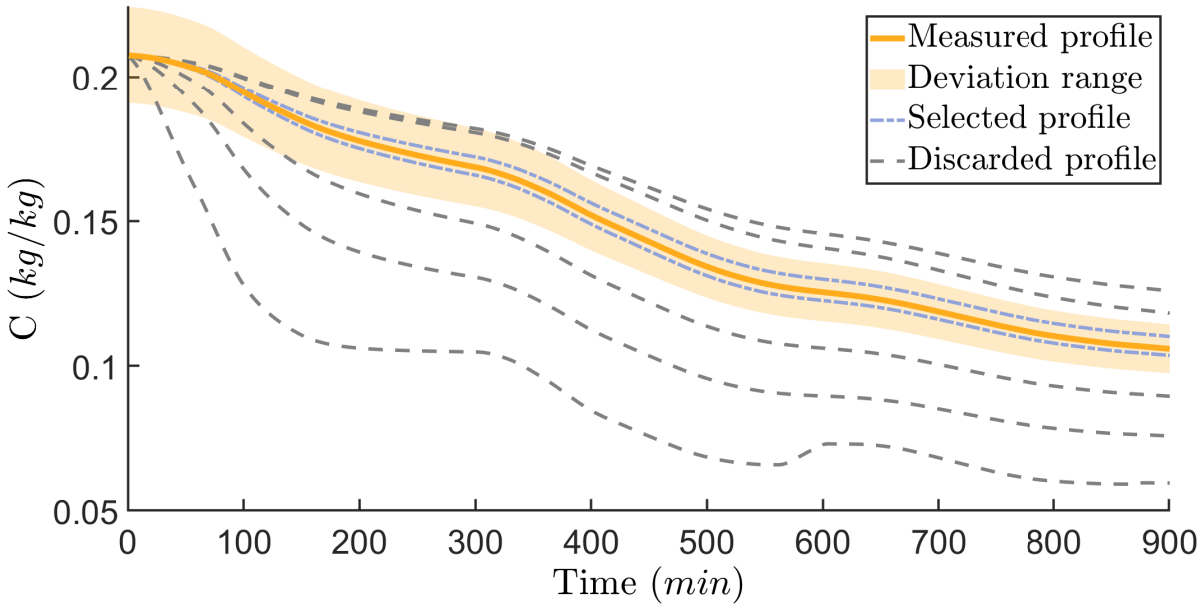


(a)

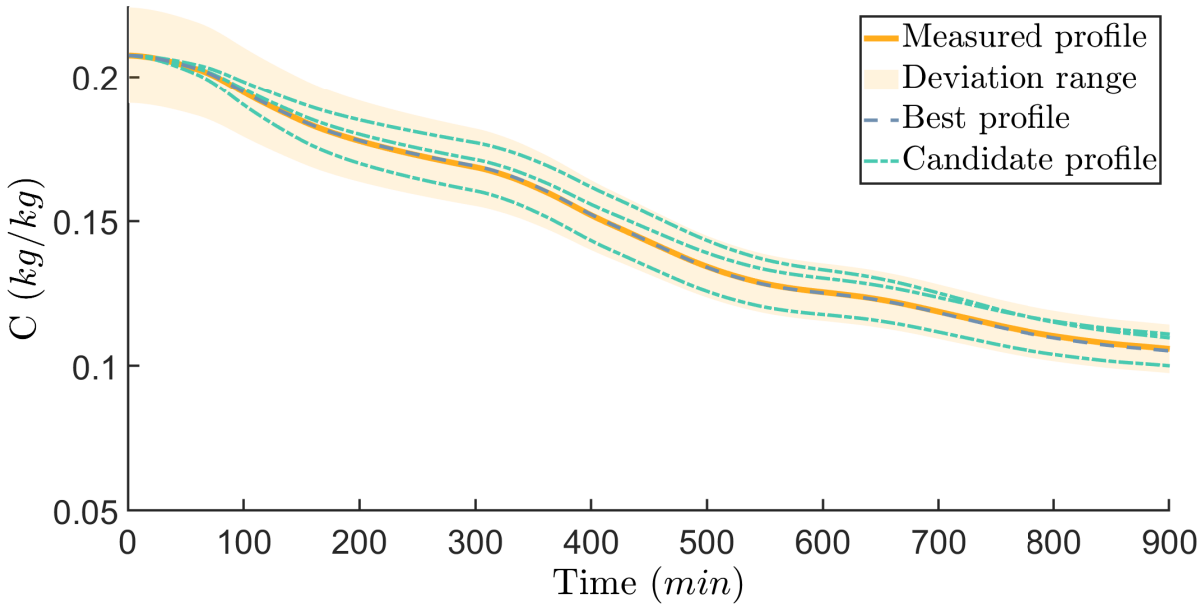


(b)

Figure 4: Comparison of the accumulative error (a) E_{RNN} and (b) E'_{RNN} at every sampling period during the batch run under AERNN-MPC with error-triggered online update mechanisms in the presence of real-time full system state measurements (AERNN_update_realtime) and in the absence of real-time measurements of N_i (AERNN_update_periodic), respectively. The AERNN-MPC without the online update mechanism is denoted by AERNN_without_update, and both the error thresholds E_T and E'_T are specified at 0.015.



(a)



(b)

Figure 5: (a) synthetic state trajectories generated from Monte Carlo simulations that are within an absolute percentage error (APE) of 10% (Deviation range) of the actual measurements of solute concentration C obtained (Measured profile) are kept (Selected profile), while others are discarded (Discarded profile). (b) the set of kinetic parameters, which has state profiles with the least absolute error as compared to the actual state measurements obtained (Best profile), is selected among a pool of plausible candidates (Candidate profile) to estimate the missing N_i at the intermediate time steps of the process.

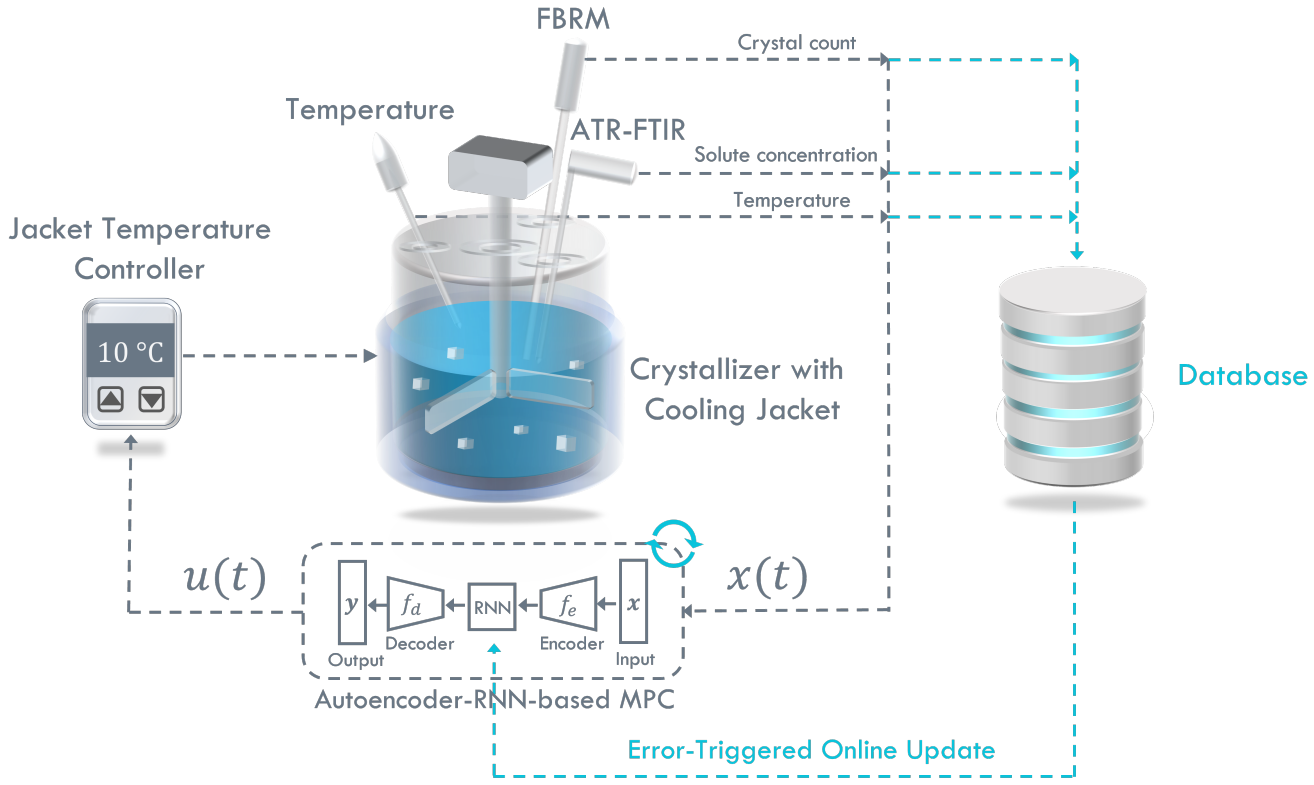


Figure 6: Illustration of the closed-loop control by AERNN-MPC with error-triggered online update: sensor data are actively collected and stored in a database, and an online update of the predictive model will be triggered if the discrepancy between the predicted and measured system states exceeds a specific threshold. The database size should be sufficient to store the sensor data collected up to 5 sampling periods Δ.

Table 1: sampling distributions of kinetic parameters

Parameter	Distribution	Deviation Range
k_N ($1/m^3$ kg s)	U(5.46×10^{14}, 2.00×10^{15})	(-70%, +10%)
n (-)	U(2.70, 6.08)	(-20%, +80%)
k_G (m/s)	U(5.84×10^{-8}, 1.46×10^{-7})	(-40%, +50%)
g (-)	U(1.18, 2.94)	(-20%, +100%)
k_A (s/m^7)	U(7.98×10^{9}, 1.73×10^{10})	(-40%, +30%)
k_D (m/s)	U(1.75×10^{-7}, 4.38×10^{-7})	(-40%, +50%)

Table 2: Deviated kinetic parameters used in closed-loop simulations

Parameter	Original Value	Deviated Value	Deviation
k_N ($1/\mathrm{m}^3$ kg s)	1.82×10^{15}	0.91×10^{15}	-50%
n (-)	3.38	5.07	+50%
k_G (m/s)	9.74×10^{-8}	7.79×10^{-8}	-20%
g (-)	1.47	2.65	+80%
k_A (s/m^7)	1.33×10^{10}	1.00×10^{10}	-25%
k_D (m/s)	2.92×10^{-7}	2.04×10^{-7}	-30%