Predictive Control of Batch Crystallization Process Using Machine Learning ⋆

Yingzhe Zheng * **Zhe Wu** **

* *Department of Chemical and Biomolecular Engineering, National University of Singapore, 117585, Singapore. (e-mail: yzzheng@nus.edu.sg)*
** *Department of Chemical and Biomolecular Engineering, National University of Singapore, 117585, Singapore. (email: wuzhe@nus.edu.sg)*

Abstract: This work develops a framework for building machine learning models and machine learning-based predictive control schemes for batch crystallization processes. We consider a seeded fesoterodine fumarate cooling crystallization and dissolution in a batch reactor and present the methodology and implementation of simulation, modeling, and controller design. Specifically, to address the experimental data scarcity problem, we first develop a population balance model (PBM) based on published kinetic parameters to describe the formation of crystals via nucleation, growth, and agglomeration. Then, recurrent neural network (RNN) models are developed using data from extensive simulations of the semi-empirical PBM under various operating conditions to capture the process dynamic behavior. The model predictive control (MPC) scheme using RNN models is developed to optimize the crystallization process in terms of product yield, crystal size, and energy consumption, while accounting for the constraints on the manipulated inputs. Through open- and closed-loop simulations, it is demonstrated that the RNN models well capture the process dynamics, and the RNN-based MPC achieves desired product yield and crystal size with significantly improved computational efficiency.

Keywords: Batch crystallization processes; Model predictive control; Machine learning; Recurrent neural networks

1. INTRODUCTION

In the pharma industry, batch crystallization is commonly regarded as an indispensable component of most pharmaceutical processes as more than 90% of active pharmaceutical ingredients are synthesized in the form of crystals (Alvarez and Myerson, 2010). The physicochemical properties of the crystalline product (e.g., crystal size distribution (CSD), purity, shape) are primarily determined by the intricate interplay between the crystallization kinetics of growth, nucleation, agglomeration, and breakage, which are in turn heavily influenced by the crystallization operating conditions. Therefore, selecting the optimal operating and control strategies is of great interest to pharmaceutical manufacturers in attaining a more efficient and greener achievement of the specification targets.

Modern pharmaceutical development approaches leverage the quality-by-control (QbC) methodology for enhanced process understanding and product assurance. QbC emphasizes the use of mathematical and knowledge-based modeling and is widely recognized as a valuable tool for improved mechanistic understanding, process optimization and control (Simone et al., 2015). However, while recent studies have sought to elucidate the optimization and control of crystallization processes (Liu et al., 2020), a majority has adopted the first-principles modeling approach (e.g., population balance model (PBM)), which is susceptible to high computation complexity and is hence less amenable for real-time optimization and control. Attributing to the burgeoning research and development in many process analytical technologies (PATs) and the pervasive nature of data in modern pharmaceutical enterprises, machine learning model-based predictive control (MPC) has emerged as the state-of-the-art optimal control strategy for cooling crystallization processes.

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. The development of accurate and computationally efficient process models has been a long-standing research problem in predictive control of dynamic processes. Recently, recurrent neural networks (RNNs) have received considerable attention in model identification of chemical processes due to their ability in modeling multi-dimensional, nonlinear dynamic systems using process time-series data. Additionally, RNNs have been utilized in MPC in many recent works to control chemical plants in real time and optimize process performance accounting for closed-loop stability, safety and control actuator constraints (Wu et al., 2019a,c). However, RNN-based predictive control of batch crystallization pro-

⋆ Financial support from the Pfizer Asia Manufacturing PTE. LTD. under Data-Driven Knowledge Discovery in Pharmaceutical Manufacturing through Batch Process Optimization (R-279-000-624-592) and NUS Start-up grant (R-279-000-656-731) is gratefully acknowledged.

cesses, using data generated from discretized PBM, has not been studied yet.

Motivated by the above considerations, in this manuscript, we develop a general framework for building machine learning models and developing machine-learning-based predictive control schemes for batch crystallization processes. Specifically, we consider a seeded fesoterodine fumarate (FF) cooling crystallization and dissolution in a batch reactor and first develop a process model based on published kinetic parameters and experimental data reported in Trampuž et al. (2019). Subsequently, an RNN-based MPC scheme is proposed to optimize product yield, crystal size, and energy consumption while accounting for the physical constraints on cooling jacket temperature. The simulation results demonstrate that the proposed RNN-based MPC achieves the desired closed-loop performance at a significantly improved computation time.

2. MODELING OF BATCH CRYSTALLIZER FOR FESOTERODINE FUMARATE CRYSTALLIZATION

2.1 Population balance

Fesoterodine fumarate (FF) is a muscarinic antagonist indicated for the treatment of overactive bladder. Its crystalline state of form I, which resembles a cubic structure, may be produced by seeded cooling crystallization from solutions in 2-butanone, where its mechanisms and kinetics have been well investigated by Trampuž et al. (2019) and summarized in the discretized population balance models (PBM), as shown in Eqs. 1 - 6:

$$\frac{dN_1}{dt} = -\frac{N_1}{2w_1}G_1 + B_N - A_{D,1} \tag{1}$$

$$\frac{dN_i}{dt} = -\frac{N_i}{2w_i}G_i + \frac{N_{i-1}}{2w_{i-1}}G_{i-1} + A_{B,i} - A_{D,i} \tag{2}$$

$$\frac{dN_r}{dt} = \frac{N_{r-1}}{2w_{r-1}}G_{r-1} + A_{B,r} \tag{3}$$

$$\frac{dN_1}{dt} = -\frac{N_1}{2w_1}D_1 + \frac{N_2}{2w_2}D_2 \tag{4}$$

$$\frac{dN_i}{dt} = -\frac{N_i}{2w_i}D_i + \frac{N_{i+1}}{2w_{i+1}}D_{i+1} \tag{5}$$

$$\frac{dN_r}{dt} = \frac{N_r}{2w_r}D_r \tag{6}$$

The system of Eqs. 1 - 6 can be solved numerically using standard integration methods (e.g., Runge-Kutta methods). Eqs. 1 - 3 collectively describe nucleation N, crystal growth G, and agglomeration A in a supersaturated system, whereas Eqs. 4 - 6 account for crystal dissolution D in an undersaturated system. Specifically, following the same discretization employed in Trampuž et al. (2019), the number of size classes $r = 40$ is divided between 0.1 and 1000 μm with logarithmic progression of class borders. N_i denotes the number of crystals in size class i, w_i denotes the width of size class i, G_i and D_i denote the size-dependent crystal growth rate and dissolution rate, respectively. $A_{B,i}$ and $A_{D,i}$ denote the size-dependent crystal agglomeration birth rate and disappearance rate, respectively. Interested readers are referred to Trampuž et al. (2019) for a detailed discussion on the PBM formulation and the computation of the various crystallization kinetics in seeded batch cooling crystallization of form I FF.

2.2 Energy balance

It is assumed that the enthalpy of crystallization is negligible and the batch crystallizer is perfectly mixed with an uniform temperature distribution. Temperature change in the crystallizer can be estimated by the following ordinary differential equation:

$$\frac{dT_r}{dt} = \frac{UA(T_j - T_r)}{m_r c_{p,r}} \tag{7}$$

where the change in crystallizer temperature is determined by the overall heat transfer coefficient U, the area of crystallizer wall A available for heat transfer, the difference between jacket temperature T_j and crystallizer temperature T_r, the mass of crystallization mixture m_r, and the specific heat capacity of the crystallization mixture $c_{p,r}$.

2.3 Mass balance

Total mass of the crystals formed is computed as the sum of masses of crystals within each class:

$$m_{cr} = k_\nu \rho_{cr} \sum_{i=1}^{r} N_i S_i^3 \tag{8}$$

where k_ν is the crystal volume factor, ρ_{cr} is the density of crystals, and the total volume of the crystals formed is determined as the sum of product between the number of crystals N_i and the average crystal size S_i in each class.

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

$$C(t) = \frac{m_0 + m_{seed} - m_{cr}(t)}{m_{sol}} \tag{9}$$

2.4 Model validation

The PBM comprising Eqs. 1 - 6 is developed based on published kinetic parameters and validated against the experimental data reported (Trampuž et al., 2019). Fig. 1 demonstrates the validity of the model developed in this work for describing the general trends of the dynamic behavior of the batch system of interest. We also compared the PBM developed in this work with the one by Trampuž et al. (2019), and found the two models are in close agreement with each other. The discrepancy can be attributed to the different techniques and values adopted in estimating the physicochemical properties of the crystallization system.

3. RECURRENT NEURAL NETWORK (RNN) DEVELOPMENT

Due to the complexity of PBM, it is computationally impractical to solve it in real-time optimization and control of the process operation. In this section, we propose a machine learning modeling approach for capturing the

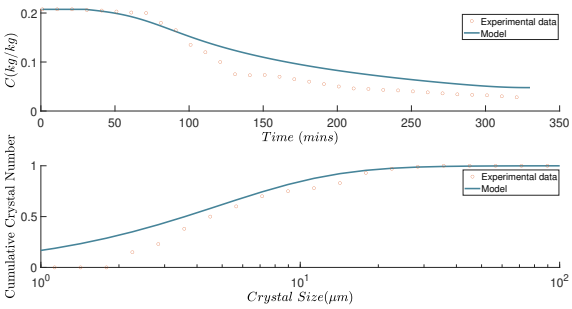

Fig. 1. Validating the PBM developed in this work against the experimental data reported in Trampuž et al. (2019).

dynamic behavior of a batch crystallization process using the data from experiments and simulations. We will demonstrate that the machine learning model significantly improves computational efficiency while maintaining a desired prediction accuracy compared to the semi-empirical PBM.

3.1 RNN formulation

RNN is utilized to model the batch crystallization of FF using the simulation data generated from the PBM in the previous section. RNN is the preferred choice of surrogate model as other deep neural networks such as conventional feedforward artificial neural networks are usually incapable of capturing the full dynamics provided by the PBM, and the use of RNN overcomes this limitation since the entire dynamics accounting for all the intermediate time steps of the PBM can be modeled using RNN.

Specifically, the RNN is constructed with input, hidden and output layers (Fig. 2), where the states in the hidden layers $\mathbf{x} \in \mathbf{R}^{d_x}$ are represented as follows:

$$\mathbf{x}_t = \sigma_h(U\mathbf{x}_{t-1} + W\mathbf{u}_t) \tag{10}$$

where $\mathbf{u}_t \in \mathbf{R}^{d_u}$ are the RNN inputs at time t, and the weight matrices $W \in \mathbf{R}^{d_x \times d_u}$, $U \in \mathbf{R}^{d_x \times d_x}$ are associated with the input and hidden state vectors, respectively. The element-wise nonlinear activation function is denoted by σ_h (e.g., ReLU). The output layer $\mathbf{y}_t$ is calculated using the following equation:

$$\mathbf{y}_t = \sigma_y(V\mathbf{x}_t) \tag{11}$$

where the activation function σ_y and the weight matrix $V \in \mathbf{R}^{d_y \times d_x}$ are associated with the output layer. In regression problems, a linear unit is generally used as the activation function in the output layer. To simplify the notation, the RNN model of Eqs. 10-11 can be collectively represented as the following continuous-time nonlinear system (Wu et al., 2019b).

$$\dot{\mathbf{x}} = F_{nn}(\mathbf{x}, \mathbf{u}) := A\mathbf{x} + \Theta^T \mathbf{z} \tag{12}$$

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

Contrary to the one-way connectivity between units in a feedforward neural networks (FNNs), one dominant advantage of RNNs is that RNNs have signals traveling in both directions (i.e., forward and backward) by introducing loops in the network. This enables feedback of information derived from earlier inputs into the network, thereby exhibiting a dynamic behavior. Furthermore, based on the universal approximation theorem for FNNs, it is shown in Billings (2013) that an RNN model with sufficient number of neurons is capable of approximating any dynamic nonlinear system on compact subsets of the state-space for finite time. This makes RNN an ideal candidate for approximating the continuous-time nonlinear systems of Eqs. 1 - 6.

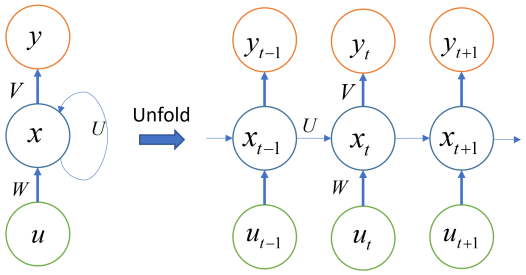

Fig. 2. Recurrent neural network structure.

3.2 Data generation

To construct an RNN model with a desired accuracy, extensive open-loop simulations are first conducted to obtain a rich data set that captures the system dynamics of the batch crystallization process. The system of Eqs. 1 - 9 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 performed using similar initial conditions reported in Trampuž et al. (2019): $m_{sol} = 0.482\ kg$, $m_0 = 0.1\ kg$, $m_{seed} = 5 \times 10^{-3}\ kg$, and $T_r = 35\ ^\circ C$, and subject to a sequence of varying jacket temperatures in a sample-and-hold fashion (i.e., the jacket temperature is fed into the system of Eqs. 1 - 9 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 $\Delta = 30$ mins indicating one sampling period). As the PBM formulation in Trampuž et al. (2019) incorporates T_j as the only manipulated input, other variables (e.g., stirring rate) could not be considered. It should be noted that the proposed RNN modeling approach in this work can be generalized to more comprehensive mechanistic models that account for all other operating conditions as manipulated inputs.

With the aforementioned configuration, dynamic state trajectories (C, T_r, N_i) are sampled at an integration time step $h_c = 1$ min within each sampling period. Simulation data from a total of 600 batch runs (each lasts for 15 hrs) are collected. The state trajectories from each batch run are then discretized, where the time-series data are separated into 30 time-series samples with a period of $\Delta = 30$ mins, which represents the prediction horizon of the RNN model. As a result, 18,000 datasets are obtained, and are partitioned into training (60%), validation (10%), and testing (30%) datasets. It is essential to obtain such a large amount of simulation data in order to guarantee a comprehensive representation

of the dynamic batch process under different operating conditions (e.g., $T_r,\ C,\ N_i$).

3.3 Open-loop simulation results

Next, the RNN model $F_{nn}(\mathbf{x}, \mathbf{u})$ is developed using the state-of-the-art application programming interface (API) Keras Chollet et al. (2015) to predict future states (i.e., $C,\ T_r,\ N_i$) for one sampling period $\Delta = 30$ mins given the current state measurements and the manipulated input (i.e., T_j). As shown in Fig. 2, in each prediction period, the time interval between two consecutive internal states x_{t-1} and x_t for the unfolded RNN is chosen to be the integration time step h_c used in open-loop simulations. Therefore, all the states between $t = 0$ and $t = \Delta$ with a step size of h_c are treated as the internal states and can be predicted by the RNN model. Finally, the RNN model is designed with two hidden recurrent layers consisting of 630 and 600 recurrent units regularized with dropouts of 0.25 and 0.15, respectively. The activation function is chosen to be the rectified linear activation function (ReLU). Furthermore, to facilitate more efficient training of the RNN, both input and output data are normalized to values between 0 and 1. The resulting RNN model is trained for 500 epochs with the Adam optimizer, and has attained mean squared errors (MSE) of 1.45×10^{-5} and 1.29×10^{-5} on validation and testing datasets, respectively. Fig. 3 compares the crystallizer temperature T_r, solute concentration C, and crystal number N_{16} of size class 16 predicted by the RNN model, and the PBM of Eqs. 1-9, respectively, under the same pseudo-random jacket temperature profile. The RNN model is developed to predict one sampling period forward, and thus, it is recursively applied to predict the entire trajectory for 900 mins. It is demonstrated in Fig. 3 that the RNN prediction results and the PBM results are in close agreement with each other under various jacket temperatures.

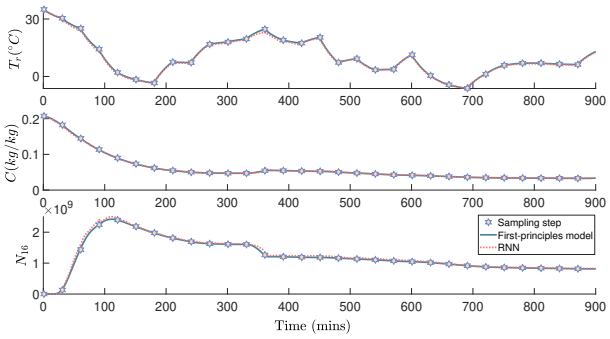

Fig. 3. Comparison of crystallizer temperature T_r (top figure), solute concentration C (middle figure), and crystal number N_{16} of size class 16 (bottom figure) predicted by the RNN model, and the first-principles model, respectively, where the stars denote the sampling time.

In the simulation, it is noted that, due to the computational complexity of PBM, each batch run took approximately 40 s, and this thus poses a great hindrance to its applicability in real-time process optimization and control. Whereas the RNN model only took approximately 0.1 s for predicting the states of each batch run, thereby demonstrating a dominant computational efficiency, and is thus considered as a better alternative to PBM for real-time process optimization and control.

4. RNN-BASED PREDICTIVE CONTROL

In this section, we develop a machine learning-based predictive control scheme to optimize the batch crystallization of FF. Specifically, the predictive control scheme is formulated as a real-time optimization problem that computes the optimal manipulated input (jacket temperature) to optimize a number of process performance considerations, for example, product yield, a particular crystal size of interest, and energy consumption. Additionally, by incorporating RNN models in MPC schemes, the MPC optimization problem achieves a desired control performance with significantly reduced computation time. The formulation of the RNN-based MPC is presented as follows:

$$\min_{u \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{\mathbf{x}}, \mathbf{u}) dt \tag{13a}$$

$$\text{s.t.} \quad \dot{\tilde{\mathbf{x}}}(t) = F_{nn}(\tilde{\mathbf{x}}(t), \mathbf{u}(t)) \tag{13b}$$

$$\tilde{\mathbf{x}}(t_k) = \mathbf{x}(t_k) \tag{13c}$$

$$\mathbf{u}(t) \in U,\ \forall\ t \in [t_k, t_{k+N}) \tag{13d}$$

$$|\Delta \mathbf{u}| \leq U_\Delta,\ \forall\ t \in [t_k, t_{k+N}) \tag{13e}$$

where $\tilde{\mathbf{x}} = [C, T, N_i] \in \mathbf{R}^{42}$, $i = 1, ..., 40$ and $\mathbf{u} = T_j$ are the predicted states and manipulated inputs, respectively. $S(\Delta)$ is the set of piecewise constant functions with sampling period Δ. $L(\tilde{\mathbf{x}}, \mathbf{u}) = Q_1(C(t))^2 - Q_2(N_{21}(t))^2 + Q_3(T_j(t) - T_j^r)^2$ is the objective function that minimizes $C(t)$ (i.e., maximizing product yield), maximizes N_{21}, and minimizes the energy consumption which is represented as the deviation of the jacket temperature T_j from its value at room temperature ($T_j^r = 25^\circ C$) over the prediction horizon $t \in [t_k, t_{k+N})$, where Q_1, Q_2, Q_3 are the coefficients chosen to balance the scale of each term. The RNN model is used as the predictive model in Eq. 13b, and the measurements of process states at every sampling time are used as the initial condition to predict the evolution of the dynamics of the batch crystallizer (Eq. 13c). Eq. 13d is the constraint on the jacket temperature that reflects the physical limitations on coolant supply, and Eq. 13e is the rate-of-change constraint on the jacket temperature to avoid extreme changes in the manipulated inputs. In this work, $Q_1 = 10^6, Q_2 = 10^{-15}, Q_3 = 0.1$ are chosen in Eq. 13a. The jacket temperature is bounded by $U = [-10,\ 30]^\circ C$, and $|\Delta u| = |T_j(t_{k+1}) - T_j(T_k)|$, $k = 0, 1, 2, ...$, is bounded by $U_\Delta = 5^\circ C$.

The RNN-MPC optimization problem of Eq. 13 is solved 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 crystallizer, which is represented by the PBM in Section 2. Since the RNN model is developed to predict one sampling period Δ, the RNN model prediction is carried out 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 (Wächter and Biegler, 2006).

Remark 1. Note that all intermediate steps within one RNN prediction step (i.e., all the integration time steps

within one sampling period) are utilized in the calculation of the integral of Eq. 13a. This improves the MPC optimization process since more dynamic information is included in the calculation. As compared to the feed-forward neural networks that predict one time step only, incorporating all the internal steps is one of the benefits of using RNNs in modeling nonlinear dynamic systems.

5. CLOSED-LOOP SIMULATION RESULTS

The batch crystallization process of FF is used to illustrate the application of the RNN-based MPC of Eq. 13 to maximize both the product yield and a particular crystal size of interest, while accounting for the energy consumption and the physical constraints of the cooling jacket. Closed-loop simualtions are carried out for the seeded cooling crystallization of FF using the RNN-MPC of Eq. 13. Additionally, the MPC, utilizing the PBM of Eqs. 1-9 as its predictive model, is also used as a benchmark case for comparison purpose.

The process states $\mathbf{x} = [C, T, N_i] \in \mathbf{R}^{42}$, $i = 1, ..., 40$, are the solute concentration C, crystallizer temperature T, and the number of crystals in 40 size classes through logarithmic discretization, respectively. The manipulated input is the jacket temperature $\mathbf{u} = T_j$. The MPC is implemented in a sample-and-hold fashion, with the sampling time $\Delta = 30$ min. The process is initially operated at $m_{sol} = 0.482\ kg$, $m_0 = 0.1\ kg$, $m_{seed} = 5 \times 10^{-3}\ kg$, $T_r = 35\ ^\circ C$, and $T_j = 35\ ^\circ C$.

Fig. 4 shows the closed-loop solute concentration profiles (top figure), and the number of crystals of average size 11.3 μm (bottom figure) under open-loop control, and the MPC using the first-principles PBM (denoted by FP-MPC), and the RNN model (denoted by RNN-MPC), respectively. The open-loop control scheme uses the maximum cooling (i.e., $T_j = -10^\circ C$) after it reaches $-10^\circ C$, and subject to the rate-of-change constraint. It is demonstrated that both MPCs maximize the product yield by decreasing the solute concentration, and achieve a desired number of crystals with the size of interest. Overall, the closed-loop performance under RNN-MPC is very close to that under the PBM-based MPC, which demonstrates that the RNN model provides a sufficiently accurate prediction of the dynamic evolution of the batch crystallization process in the MPC optimization problem as compared to the benchmark case of FP-MPC. We have also performed additional simulations with various initial conditions, and observe that the RNN-MPC results are all very similar to those under FP-MPC.

Fig. 5 shows the evolution of crystal size distribution in the closed-loop simulation with RNN-MPC. It is noticed that the number of crystals in the size class of interest (N_{21} of approximately 11.3μm) reaches an initial peak rapidly before experiencing a sharp descent. This observation can be attributed to the addition of seed crystals (corresponding to N_{18} and N_{19}), which marks the onset of the crystallization process. At the initial stage, most crystals are clustered in N_{18} and N_{19}, which are in close proximity to the target size class N_{21}. Further ascribing to the sharp decrease in crystallizer temperature T_r at this initial stage, the driving force for crystal growth is maximized, thereby engendering a rapid increase in crystals of larger size classes adjacent to the seed size class and hence the observed sharp increase in them. As the crystallization process proceeds further, the number of crystals spreads more evenly among the various size classes due to the interplay between the various crystallization kinetics (e.g., crystal growth and agglomeration), and classes close to N_{18} and N_{19} would thus experience a decline. Finally, as observed toward the end of the crystallization process, the number of crystals in all classes levels off. This is mainly due to the effect of T_r, which also stabilizes toward the end, and therefore diminishes the driving force for crystal growth. The crystal size distribution profile obtained is also similar to the ones reported by Trampuž et al. (2019).

Fig. 6 shows the closed-loop crystallizer temperature (top figure), and jacket temperature profiles (bottom figure) under open-loop control, FP-MPC, and RNN-MPC, respectively. The open-loop control initially decreases T_j and uses $T_j = -10^\circ C$ for the remaining time of simulation after it reaches the lowest jacket temperature of $-10^\circ C$. However, it is observed that both MPCs decrease the jacket temperature initially, and after around 240 min, gradually increase the jacket temperature in order to reduce energy consumption since both the solute concentration and the number of crystals in N_{21} (Fig. 5) start to vary slowly after 240 min. This phenomenon is in well agreement with the empirical results reported by Trampuž et al. (2019), and can be attributed to the size of the target class N_{21}, which is comparable to that of the crystal seeds (e.g., $\sim 6.4\mu m$ v.s.$\sim 11.3\mu m$). Ascribing to the relatively small target crystal size, a fast cooling is preferred as it is more conducive for the formation of small crystals. Through comparison with the open-loop scheme, it is demonstrated in Fig. 4 that no significant improvement in product yield and the number of crystals can be achieved by maintaining the lowest jacket temperature. Therefore, the MPCs optimize the overall performance by stopping using maximum cooling while still maintaining a desired outcome.

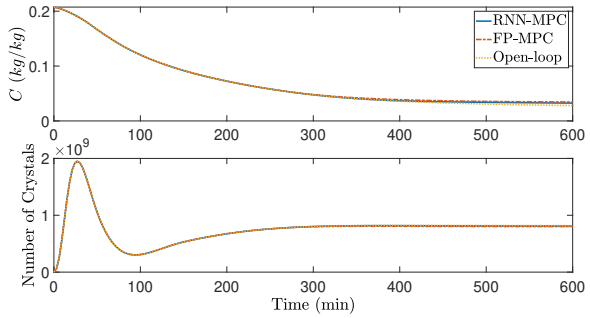

Fig. 4. Comparison of closed-loop solute concentration profiles (top figure), and the number of crystals of size 11.3 μm (bottom figure) under open-loop control (i.e., using maximum cooling after T_j reaches $-10^\circ C$), and the MPC using the first-principles model (FP-MPC), and the RNN model (RNN-MPC), respectively.

Additionally, one of the benefits of using RNN models is the computational efficiency of online prediction using the offline-trained RNN model. Therefore, we compare the computation time for solving FP-MPC and RNN-MPC under the same initial condition using the desktop with Intel Core i7-10700 CPU @ 2.90GHz. Table 1 reports the average computation time to solve the MPC optimization

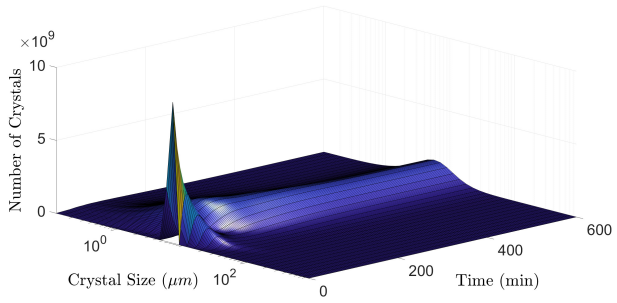

Fig. 5. Crystal size distribution in the closed-loop simulation using RNN-MPC.

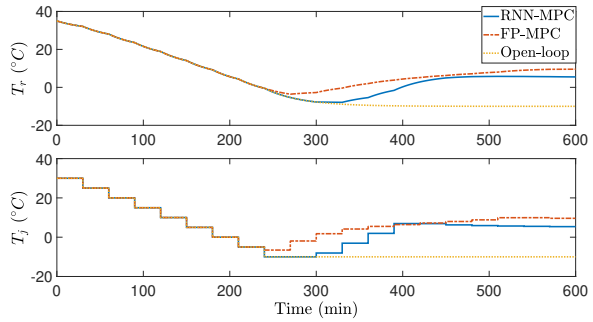

Fig. 6. Comparison of closed-loop crystallizer temperature (top figure), and jacket temperature (bottom figure) under open-loop control (i.e., using maximum cooling after T_j reaches $-10°C$), and the MPC using the first-principles model (FP-MPC), and the RNN model (RNN-MPC), respectively.

problem with different prediction horizons at each sampling step. Since the MPC is solved successively when new state measurements are received at each sampling time with $\Delta = 30$ min, we run the closed-loop simulation for 600 min, and calculate the average computation time for solving MPC problems 20 times over the entire simulation period.

Table 1. Computation time for solving MPC

MPC Prediction Horizon N	FP-MPC	RNN-MPC
3	148.0 seconds	24.7 seconds
5	3992.7 seconds	327.3 seconds
10	N/A	1023.3 seconds

From Table 1, it is shown that the RNN-MPC significantly reduces the computation time compared to the FP-MPC. Additionally, as MPC prediction horizon increases, the computation time for solving MPC optimization problem increases rapidly. Specifically, for RNN-MPC, the computation time for solving MPC at each sampling time is rendered less than one sampling period (i.e., 30 min) for all prediction horizons $N \leq 10$; however, it is noticed that for FP-MPC, the computation time for $N = 5$ has already exceeded 30 min, making it infeasible for practical implementation. Also, the FP-MPC with $N = 10$ is unable to obtain the solutions within a reasonable computation time, and thus, is shown as N/A in Table 1. Since computation time is an important performance metric that determines whether an algorithm can be implemented in practice, it demonstrates the benefit of RNN-MPC in the practical implementation of real-time control of crystallization process. Moreover, due to its data driven nature, the prediction accuracy of RNN can be readily improved by updating it with the most recent process data obtained through online learning, and thereby rendering RNN robust predictive models.

6. CONCLUSION

In this work, we developed an RNN model for a batch crystallization process using the simulation data from a PBM, and designed an RNN-MPC to control the batch crystallization process in real time in order to optimize its product yield, crystal size, and energy consumption. Through open- and closed-loop simulations, we demonstrated that the RNN model achieved a desired accuracy in predicting future states with much less computation time as compared to PBM, and the RNN-MPC achieved desired product yield and crystal size while reducing energy consumption simultaneously. Additionally, the computational efficiency of solving real-time control problems was significantly improved through the use of the RNN in MPC.

REFERENCES

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

Billings, S.A. (2013). *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Liu, Y.C., Acevedo, D., Yang, X., Naimi, S., Wu, W., Pavurala, N., Nagy, Z.K., and O'Connor, T.F. (2020). Population balance model development verification and validation of cooling crystallization of carbamazepine. *Crystal Growth & Design*, 20, 5235–5250.

Simone, E., Zhang, W., and Nagy, Z. (2015). Application of pat-based feedback control strategies to improve purity and size distribution in biopharmaceutical crystallization. *Cryst. Growth Des*, 15(6), 2908–2919.

Trampuž, M., Teslić, D., and Likozar, B. (2019). 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.

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

Wu, Z., Tran, A., Ren, Y.M., Barnes, C.S., Chen, S., and Christofides, P.D. (2019a). Model predictive control of phthalic anhydride synthesis in a fixed-bed catalytic reactor via machine learning modeling. *Chemical Engineering Research and Design*, 145, 173–183.

Wu, Z., Tran, A., Rincon, D., and Christofides, P.D. (2019b). Machine learning-based predictive control of nonlinear processes. part I: Theory. *AIChE Journal*, 65, e16729.

Wu, Z., Tran, A., Rincon, D., and Christofides, P.D. (2019c). Machine learning-based predictive control of nonlinear processes. part II: Computational implementation. *AIChE Journal*, 65, e16734.