Optimizing TPU performance: The role of mold temperature on injection molding of TPU

Grønborg, Frederik; Pedersen, David Bue; Spangenberg, Jon; Daugaard, Anders Egede; Susoff, Markus Lothar

Published in:
Journal of Applied Polymer Science

Link to article, DOI:
10.1002/app.55521

Publication date:
2024

Document Version
Publisher's PDF, also known as Version of record

I&EC research
Industrial & Engineering Chemistry Research

pubs.acs.org/IECR

Article

Semi-Mechanistic Prediction and Optimization of Residence Time Metrics of a Starve-Fed Extruder via a Hybrid Machine-Learning Convection−Diffusion Model

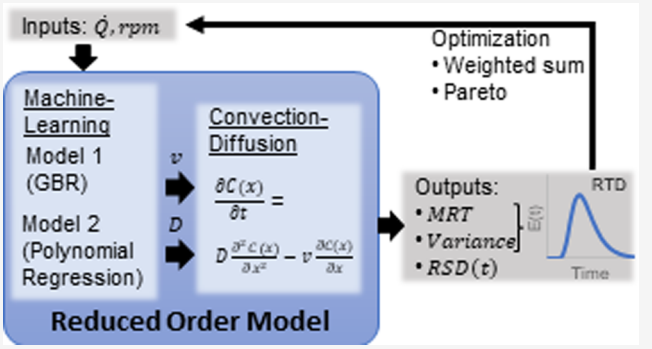
Ashley Dan, Urjit Patil, Erik Holmen Olofsson, Jesper Henri Hattel, and Rohit Ramachandran*

Cite This: *Ind. Eng. Chem. Res.* 2024, 63, 7271−7280

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: This study addresses the challenge of determining mixing dynamics, such as the residence time distribution (RTD) and relative standard deviation (RSD) of a system, which usually require tedious experimental setups or computationally intensive models. A reduced order model (ROM) in the form of a machine-learning-based convection−diffusion model was developed for the computationally efficient prediction of mixing metrics. A two-dimensional compartmental convection−diffusion model was applied to a starve-fed single screw extruder, capturing non-uniform axial velocity fluxes. The model was enhanced through machine learning algorithms to establish correlations between process parameters, velocity fluxes, and diffusion coefficients. The use of a convection−diffusion model provided a mechanistic prediction of RTD and RSD. The model accuracy was demonstrated via an *R*-squared value of above 0.9 for the prediction of different metrics evaluated, such as mean residence time and variance. The calibrated model was then optimized to identify improved input conditions that led to the desired mixing metrics. The developed ROM provided an efficient alternative to the full computational fluid dynamics (CFD) model by substantially reducing the computational time from 44 h to 2.5 s while retaining model accuracy. Source codes for the model are available on GitHub repositories at https://github.com/adan626/ROM_CFD and https://github.com/patilurjit/CFD-ML.



1. INTRODUCTION

Mixing dynamics and flow behavior of materials are important considerations in many industrial processes, including chemical reactions, polymer extrusion, catalyst extrusion, food mixing, and pharmaceutical mixing and granulation processes. The degree or extent of mixing is often expressed by residence time distribution (RTD), which represents the probability distribution of time that the material remains within a system.[1] From the RTD profiles, metrics, such as mean residence time (MRT), variance, and skewness, can be obtained from the first, second, and third moments, respectively.[2,3] It is common to compare RTDs using their moments (MRT, variance, and skewness), instead of the full RTD curve.[4] Variance and skewness have been used to quantify the nonideality of fluid flow and mixing (i.e., deviation from ideal systems, such as continuous stirred tank reactor (CSTR) and plug flow reactor (PFR) systems) within a system.[5] Another metric that is more commonly used in multicomponent powder processes, to express the level of mixedness, is the relative standard deviation (RSD), which calculates standard deviation of concentration of particles within specified areas of the processing equipment.[6−8] A low RSD is indicative of blend uniformity or good blend homogeneity,[9] which is the desired outcome of mixing and extrusion processes.

Quantifying mixing dynamics using RTD and RSD provides a means to mechanistically link the effects of input process parameters (e.g., mass throughput and rotation speed) and material attributes to product quality.[10] Several studies have effectively correlated RTDs to product quality and process reliability.[11−14] In these studies, heterogeneity in RTDs and product quality were introduced by varying the process parameters. Therefore, it can be inferred that the process parameters of a system directly impact its RTD, consequently influencing the output product quality. This inference supports the task of determining optimal process parameters via the optimization of the desired RTD and RSD, as such

Received: January 18, 2024
Revised: March 19, 2024
Accepted: March 29, 2024
Published: April 11, 2024

https://doi.org/10.1021/acs.iecr.4c00201
Ind. Eng. Chem. Res. 2024, 63, 7271−7280

optimization efforts will contribute to the design of a process to produce the targeted product quality.

RTDs are typically determined using a pulse test, where an instantaneous pulse input of a known amount of tracer is injected into the system feed, and the concentration of the tracer is measured at the outlet.[15] This pulse test can be done experimentally[14,16] or computationally using computational fluid dynamics (CFD)[17] or discrete element method (DEM).[5,18] However, there are some challenges associated with both approaches. The experimental approach requires the selection of an appropriate tracer that results in minimal interferences to the system but contains good detectability when measured,[16,19] whereas the computational approach often utilizes high-fidelity, first principle-based models that result in high computational cost.[17,20,21] To address the concern of computational cost, reduced order models (ROMs) are often utilized as potential computationally efficient solutions.[22,23] In the literature, equation-based models, such as axial dispersion (AD) or compartment models, are commonly used as a time efficient alternative of predicting RTDs of a system.[5] AD is a simplified, one-dimensional convection−diffusion model that assumes uniform velocity and flux along the length (axial direction) of the flow system.[2,24] On the other hand, compartmental models consist of multiple interconnected models of ideal systems (continuous stirred tank reactor (CSTR), plug flow reactor (PFR), and laminar flow reactor) and reactor design elements (recycle, bypass, dead, and stagnant zones).[15] A tanks-in-series (TiS) model is a commonly used compartment model for describing the mixing behavior. It consists of multiple perfectly mixed CSTRs connected in series and can be used as a one-parameter model, with the number of CSTRs (n) as the model parameter. When n = 1, the system is perfectly mixed and behaves as a CSTR, as n increases, the model will be able to describe nonideal mixing behaviors, and finally, as n approaches infinity, the system will resemble a PFR.[1]

In this study, a starve-fed single screw extruder was modeled. The starve-fed nature is an important attribute to the system, as CFD simulations from previous studies have indicated that the fill level of the extruder influences the velocity fluxes across the extruder.[17] Additionally, variation in process parameters, such as the throughput and screw speed, can impact the fill level within the extruder, and thus the velocity fluxes of the system.[25,26] Due to the nonuniform fluxes, starve-fed extruders are not suitable to be modeled using either AD or TiS models. Instead, a two-dimensional (2D) compartmental convection−diffusion model, based on Portillo et al.,[27] was used. The compartmental model captures the nonuniform velocity fluxes across the length of the extruder, and the two-dimensional coordinates account for fluxes in both the axial and theta direction and can predict mixing metrics, RTD and RSD. On its own, the convection−diffusion model requires the fluxes as inputs to the model, which has to be sourced from CFD model simulations. In order to develop a reduced order model (ROM) that can entirely replace the need of CFD modeling, the correlation between process parameters and velocity fluxes is established using machine learning algorithms. Additionally, the starve-fed nature of the system results in a long simulation time required to obtain an RTD profile from the CFD simulations, which drives the need for an ROM. The development of an ROM not only improves the usability of a model but is also necessary for model applications, such as dynamic modeling, real-time optimization, and control of the system.[23]

1.1. Objectives. The objective of this study is to develop a computationally efficient model that integrates a convection−diffusion model with machine-learning algorithms for the prediction of RTDs and RSDs. The reduced order model (ROM) accounts for variability in fluxes as a function of different process parameters (throughput and extruder screw speed). The model is calibrated and verified based on CFD simulation results generated from the model developed by Olofsson et al., which was experimentally validated.[17] The calibrated model will be used for process optimization to determine the optimal process parameters based on the desired mixing metrics.

2. METHODS

2.1. Design of Experiments (DoE). Data used for the reduced order model development was generated using CFD simulations of the Olofsson et al.[17] model. A full factorial design of experiments (DoE) with 4 levels and 2 factors was generated, which resulted in 16 total runs. The process parameters of the DoE are listed in Table 1.

Table 1. Variables and Levels of the Design of Experiment (DoE)

	levels			
factors/variables	−2	−1	+1	+2
screw speed (rpm)	45	55	65	75
feed rate $\times 10^{-7}$ (m^3/s)	3	4	5	6

2.2. Model Framework. The schematic in Figure 1 illustrates the ROM development and prediction of mixing metrics, which directly replicates the high-fidelity CFD model prediction of RTD. In the model development phase, data generation from the CFD model was necessary for training and calibration of different model components, including the convection−diffusion and machine learning (ML) models. From the CFD models, velocity fluxes are obtained; however, the convection−diffusion model requires both velocity fluxes and diffusion coefficient as its inputs, as shown in Table 2. Thus, the second step of the model development phase requires parameter fitting using the convection−diffusion model to determine the diffusion coefficient of each DoE run. Once the velocity fluxes and diffusion coefficient of each run were available, machine learning algorithms were trained to predict these parameters as a function of input variables (mass throughput and rotational speed). Velocity fluxes and the diffusion coefficient were then used as inputs in the convection−diffusion model to predict mixing metrics—RTD and RSD. The ROM consists of the serial combination of the ML and convection−diffusion model.

2.3. Convection−Diffusion Model. Convection−diffusion models are equation-based models that are commonly employed to describe transport phenomena, where physical quantities are transferred due to the simultaneous effects of convection and diffusion of a system.[28] The governing equation, expressed in eq 1, considers convection or advection resulting from the velocity fluxes (v) and diffusivity (D).

$$\frac{\partial C(x)}{\partial t} = D\frac{\partial^2 C(x)}{\partial x^2} - v\frac{\partial C(x)}{\partial x} \quad (1)$$

https://doi.org/10.1021/acs.iecr.4c00201
Ind. Eng. Chem. Res. 2024, 63, 7271−7280

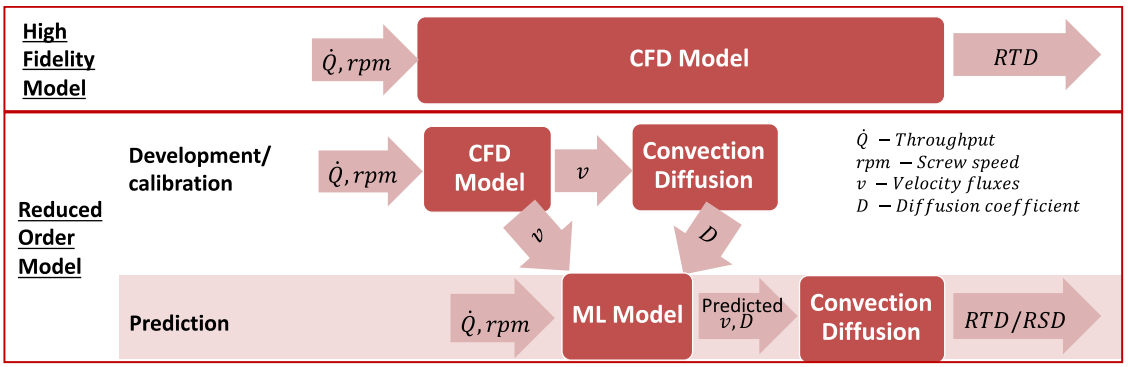


Figure 1. Comparison of the high-fidelity model and reduced order model (ROM) framework.

Table 2. Model Inputs and Outputs of the Different Model Components

model type	model inputs	model outputs
computational fluid dynamics (CFD)	mass throughput	velocity fluxes
	screw speed	residence time distribution (RTD)
convection−diffusion	velocity fluxes	residence time distribution (RTD)
	diffusion coefficient	relative standard deviation (RSD)
machine learning	mass throughput	velocity fluxes
	screw speed	diffusion coefficient
	Extruder axial length	

For this work, the extruder was discretized in both the axial and theta directions, as shown in Figure 2. A total of 181 bins in the axial direction and 10 bins in the theta direction was used. The two-dimensional convection−diffusion equation used is given by eq 2.

$$\frac{\partial C(i,j)}{\partial t} = D\left[\frac{\partial^2 C(i,j)}{\partial r^2}\right] - v_x(i)\frac{\partial C(i,j)}{\partial x} - v_r(j)\frac{\partial C(i,j)}{\partial r} \tag{2}$$

The numerical integration method used to solve the convection−diffusion equation was forward in time, centered in space for diffusion, and forward in time and backward in space for convection. Model performance was evaluated using the parity plots, as well as metrics such as the *R*-squared and normalized root-mean-squared error (NRMSE), which are given in eqs 3 and 4, respectively. *R*-squared is calculated using the sum of squared residuals (SSR) and the total sum of squares (SST).

$$R^2 = 1 - \left(\frac{\mathrm{SSR}}{\mathrm{SST}}\right) = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \tag{3}$$

$$\mathrm{NRMSE} = \sqrt{\frac{1}{n}\sum\left(\frac{|y_i - \hat{y}_i|}{y_i}\right)^2} \tag{4}$$

2.4. Machine Learning Algorithms. *2.4.1. Locally Weighted Scatterplot Smoothing (LOWESS) for Data Smoothing.* Smoothing a data set helps separate noise from the actual trends or patterns of the data.[29] Since the axial velocity data obtained from the CFD simulations contained a high level of noise, a smoothing step was required to denoise the data. Various algorithms, such as moving average, exponential moving average, kernel smoothing, B-spline smoothing, and locally weighted scatterplot smoothing (LOWESS), can be used for smoothing. Out of these options, LOWESS was selected due to its nonparametric nature, which refrained from assuming a predefined functional form of the underlying data, resulting in greater flexibility to data smoothing.[30] LOWESS fits nonparametric regression models to localized subsets of data within the data set in a moving fashion, allowing the algorithm to adapt to changing patterns and variations within the data by considering a localized context around each data point during the regression fitting process.[31]

The algorithm sequence of LOWESS is illustrated in Figure 3. In the first smoothing iteration, each data point is initially fitted with values using a localized weighted least-squares regression on the data in a subset consisting of neighboring points. The size of this subset is predefined using a tuning parameter, *f*. The span (*f*) parameter ranges from 0 to 1 and decides the fraction of the total points used for the localized regressions. The aim is to select a value for the span that is as

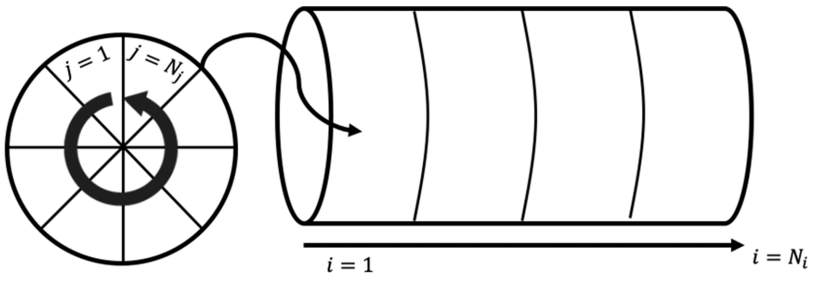


Figure 2. Schematic of the discretization scheme used in the convection−diffusion model.

https://doi.org/10.1021/acs.iecr.4c00201
Ind. Eng. Chem. Res. 2024, 63, 7271−7280

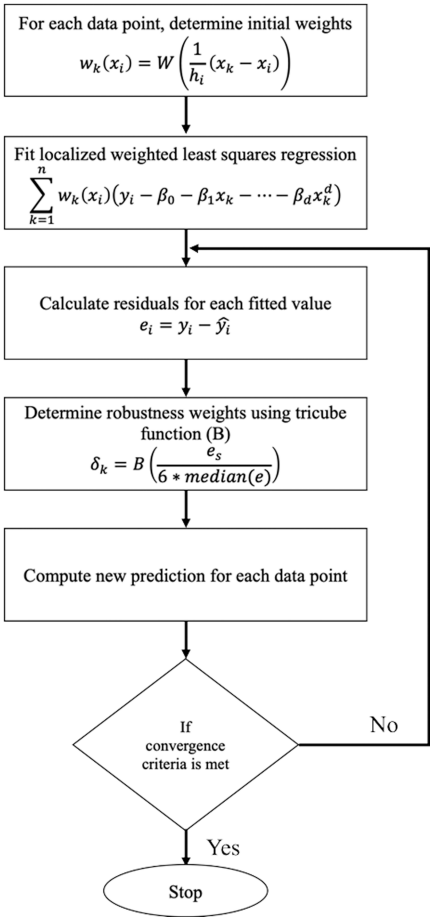


Figure 3. Schematic describing the LOWESS algorithm for data smoothing.

large as possible to minimize the variation while maintaining the underlying pattern in the data. According to Cleveland,[31] when the sole purpose of smoothing the data is to enhance the visual perception of the patterns in the plot, the choice of f is not very critical. Choosing a value between 0.2 and 0.8 should be appropriate for most instances. In contrast, for situations where the smoothed values are to be used in building a regression model, the selection of f becomes more critical. For these situations, the prediction sum of squares (PRESS) procedure by Allen[32] for the selection of independent variables in regression can be adapted for selecting an optimal value of f. PRESS assesses the predictive performance of regression models by iteratively evaluating prediction errors. These weights are determined using a weight function centered at the point of interest and gradually scaled to approach zero as the data points approach the subset boundary. This process is repeated for every data point in the data set. This weight function enables LOWESS to adaptively smooth the data, emphasizing the influence of nearby data points while diminishing the impact of those situated at a greater distance from the center.[33] In the following iterations, a new set of weights called the robustness weights are determined and used to replace the previous weights.[31] The robustness weights are determined using the tricube weight function, given in eq 5, as a function of residuals. The tricube weight function is a robust weighting function commonly used in robust regression techniques like LOWESS. It assigns weights to data points based on their residuals from the fitted regression model. Specifically, the tricube function assigns zero weight to data points with large residuals, gradually increasing the weight as the residuals decrease and reaching their maximum weight at zero residual. This function is preferred for its ability to down weight the influence of outliers more aggressively compared to other weighting functions, such as the Gaussian or bisquare functions. Its sharp decline in weight for large residuals helps LOWESS effectively mitigate the impact of outliers, thereby improving the robustness and reliability of the regression estimates. By incorporating the tricube weight function, the resilience of the method to outliers and noisy data is enhanced, ensuring more accurate and reliable regression results. Since the robustness weights are inversely proportional to the magnitude of the residuals of the fitted values, the approach is robust against outliers. Finally, weighted least-squares regression is used to determine the following predicted values. The iterative process of updating the weights and calculating the corresponding fitted values is carried out for a predetermined number of times. Once the iterations are completed, the final predicted values are connected to form the final smoothed curve.

$$W(x) = \begin{cases} (1 - |x|^3)^3, & \text{for} |x| < 1 \\ 0, & \text{for} |x| \geq 1 \end{cases} \quad (5)$$

The range and parameter values used in developing the LOWESS algorithm for smoothing of the axial velocity data are detailed in Table 3. The span parameter, which denotes the fraction of data points used in the localized regression, serves as an important factor in controlling the level of smoothing performed by LOWESS. A higher span value leads to more data points used in the regression, resulting in a smoother result. It is essential to select a span value that is sufficiently large to minimize variation while still retaining the underlying pattern inherent in the data. The value of 1 of the degree of polynomial regression (d) parameter indicates fitting a linear regression to the localized regression models.

Table 3. Parameter Range and Values Used in the LOWESS Algorithm

parameter	range	value
span (f)	0–1	0.5
degree of polynomial regression (d)	$0 - \infty$	1
number of iterations (t)	$1 - \infty$	3

2.4.2. Gradient Boosting Regression (GBR) for Prediction of Axial Velocity. Gradient boosting regression is a popular tree-based ensemble model. Ensemble algorithms are made up of multiple base models, each contributing to the final model prediction.[34] This approach mitigates overfitting by creating a final model with increased stability and accuracy compared to the individual base models.[35] Boosting is a type of ensemble method, which takes a sequential and adaptive approach to improve model performance.[36] Specifically, multiple weak models are trained iteratively, where each subsequent model is focused on rectifying the errors of its preceding model. Gradient boosting leverages boosting to create a series of decision trees. The model is initiated with the mean value as the first prediction, and pseudoresiduals, which indicate the differences between the actual and prediction values, are calculated.[37] The successive models are built to predict the pseudoresiduals that were determined in the previous step.

https://doi.org/10.1021/acs.iecr.4c00201
Ind. Eng. Chem. Res. 2024, 63, 7271−7280

Table 4. Parameter Descriptions, Cross-Validation Ranges, and Final Values Used in the GBR Algorithm Development

category	parameters	symbol used in sklearn library	range for cross-validation	value
tree-specific parameters	minimum number of data points required to split an internal node	min_samples_split	15−40	36
	maximum number of features	max_features	1−3	2
boosting parameters	learning rate	learning_rate	0.001−0.5	0.05
	number of estimators	n_esimators	20−100	100
	subsample	subsample	0.6−0.9	0.8

The final prediction of the GBR model combines the serial model predictions by adding the prediction from the successive model to the initial prediction after scaling it with a constant known as the learning rate, as shown in eq 6. Here, $\hat{y}$ indicates the predicted value, y_0 indicates the initial prediction (mean value), γ indicates the pseudoresidual predictions, and ν denotes the learning rate. Studies have shown that using a smaller learning rate value, which takes smaller steps in the right direction, results in a model with lower variance, which leads to better predictions when validating the model on new data.[38]

$$\hat{y} = y_0 + \nu\gamma_1 + \nu\gamma_2 + \cdots \nu\gamma_N \tag{6}$$

Model development of the GBR model is performed using the *GradientBoostingRegressor* function on the sklearn library in Python. The specific parameter values are determined using cross-validation, where a range is selected for each parameter, and the parameter combination with the lowest validation error is selected. Both the cross-validation range and the final values for each parameter are illustrated in Table 4.

2.4.3. Polynomial Regression for Prediction of Diffusion Coefficient. A multivariate second-order polynomial regression was used to model the diffusion coefficients containing parameters x_1, x_2, and x_3, indicating scaled mass throughput, screw speed, and average axial velocity, respectively. To avoid overfitting, L1 regularization was applied by using the Lasso function to retain only the relevant interaction features, simplifying the second-order polynomial into eq 7.

$$D = [-12.0x_1 - 5.9x_2 + 17.1x_3 + 5.2{x_1}^2 + 0.53x_1x_3 - 2.8x_2x_3] \times 10^{-6} \tag{7}$$

2.5. Optimization. Optimization was conducted to determine the optimal input process parameters required to obtain the desired mixing metrics. Two different techniques are compared, namely, the weighted sum optimization and pareto optimization. For the weighted sum, different objectives are combined as a weighted sum into the objective function, as shown in eq 8. Here, t and σ^2 denote the mean residence time and variance, respectively, and the objective function combines both the normalized sum of squared error (SSE) of t and σ^2. The optimization algorithm then iteratively searches for the process parameter values that satisfy the minimum value of the objective function. For this work, the *GlobalSearch* algorithm on MATLAB was used to conduct this task.

$$\text{minimize}_{\text{SSE}} = 0.5 \times \left(\frac{|t_{\text{Target}} - t_{\text{model}}|}{t_{\text{Target}}}\right)^2 + 0.5 \times \left(\frac{|\sigma^2_{\text{Target}} - \sigma^2_{\text{model}}|}{\sigma^2_{\text{Target}}}\right)^2 \tag{8}$$

Additionally, pareto optimization is also conducted, and its result is compared to that of the weighted sum optimization. The pareto front is a set of points in the parameter space that has noninferior function values, meaning that you can improve one function value only by degrading the other. The pareto front is created using the multiobjective function to minimize MRT and maximize variance while remaining within the stipulated process design space, as shown in eq 9. It is ideal to minimize MRT to reduce the likelihood of overmixing, which can result in material degradation, and the variance is maximized to ensure good product homogeneity. This multiobjective function was selected because both objectives pose a conflict, causing a trade-off to be required between the two objectives. The *gamultiobj* function on MATLAB is used, which utilizes the genetic algorithm technique. The hyperparameters used in the algorithm include a population size of 60 and a pareto fraction of 0.35. The combination of both parameters specifies the number of data points on the pareto front. While a lower value for both parameters can speed up convergence, it also limits the ability of the algorithm to explore the solution space.

$$\begin{aligned} &\min \text{MRT}(\text{rpm}, \dot{Q}), \max \text{Variance}(\text{rpm}, \dot{Q}) \\ &45 \le \text{rpm} \le 75 \\ &3\text{e}^{-7} \le \dot{Q} \le 6\text{e}^{-7} \end{aligned} \tag{9}$$

To compare the weighted sum optimization to the pareto optimization, a hypothetical target MRT and variance values are selected, and the results of both the optimization methods are compared.

3. RESULTS AND DISCUSSION

3.1. Parameter Estimation. As mentioned in Section 2.2, diffusion coefficients are determined through parameter estimation by using the convection−diffusion model. The parameters are determined via minimization of the sum of squared error (SSE) of the mean residence time and variance of the CFD model and ROM. The objective function that was minimized is shown in eq 10. This iterative process of determining the diffusion coefficient is repeated for every run.

$$\phi = \left(\frac{t_{\text{CFD}} - t_{\text{ROM}}}{t_{\text{CFD}}}\right)^2 + \left(\frac{\sigma^2_{\text{CFD}} - \sigma^2_{\text{ROM}}}{\sigma^2_{\text{ROM}}}\right)^2 \tag{10}$$

In addition to the diffusion coefficients, a correction factor for the axial velocities is also introduced as a calibration parameter to account for the particle forward movement as a result of the rotation of the screws. The initial value, as well as the final estimated value, is listed in Table 5. The parameter estimation was carried out using the *fminsearch* algorithm on MATLAB, which utilized the nelder-mead simplex algorithm to iteratively search for the optimal solution to minimize the objective function.

https://doi.org/10.1021/acs.iecr.4c00201
Ind. Eng. Chem. Res. 2024, 63, 7271−7280

Table 5. Estimated Parameter Values

parameter	iInitial guess value	estimated value
diffusion coefficient, D	$5e^{-6}$	$5.08e^{-6}$–$1.07e^{-5}$
v_x correction factor, x	1.5	1.4492

3.2. Model Validation and Prediction Performance.

3.2.1. Prediction of Axial Velocity. Axial velocities determined through the CFD simulations are first required to be smoothed before a prediction model could be built, as illustrated in Figure 4. The ML algorithms using hyperparameters, as listed in Section 2.4 resulted in a prediction model with good accuracy indicated by the parity plots and R-squared values for both the training and validation data sets, as shown in Figure 5.

For training the gradient boosting model for prediction axial velocities, the data set underwent a randomized split into training and validation sets, maintaining an 80:20 ratio, respectively. This randomization ensured that the division was not biased and that both sets encompassed a representative sample of the data. By allocating 80% of the data to the training set, the model could learn patterns and relationships from a substantial portion of the data set, while the remaining 20% constituted the validation set, allowing for an independent assessment of the model's performance on unseen data. This randomized partitioning strategy helps prevent overfitting and ensures the model's generalizability to new observations. However, additional care was taken to ensure that the selected validation set was not on the boundary of the process design space, which includes the low level of the DoE $\left(\text{Screwspeed} = 45\ \text{rpm}, \dot{Q} = \frac{3e^{-7}m^3}{s}\right)$ and the high level of the DoE $\left(\text{Screwspeed} = 75\ \text{rpm}, \dot{Q} = \frac{6e^{-7}m^3}{s}\right)$. This was done to avoid the need for model extrapolation outside of the DoE boundaries.

3.2.2. Prediction of Diffusion Coefficient. The model development for predicting diffusion coefficient is slightly more challenging due to the limited amount of data available (16 data points). Figure 6 illustrates the model parity plot and indicates that while the R-squared values of both the training and validation data set are not high, the NRMSE indicates reasonable prediction error. The similar R-squared value obtained for the training and validation data set suggests that overfitting is minimal. Considering that the diffusion coefficient is only an intermediate prediction, the tolerance for prediction error for this parameter is dependent on the prediction error of the final prediction, which are the RTD metrics. Since we were able to get good predictive performance in the RTD metric prediction, as shown in the next section, the higher errors for the diffusion coefficient were considered within tolerance.

3.2.3. Prediction of RTD Metrics. The predictive performance of the ROM is then evaluated based on the prediction accuracy of the first and second moments of the RTD curve: mean residence time (MRT) and variance. Figure 7 and Table 6 show the parity plots and prediction performance (R-squared and RMSE) of MRT and variance of the RTD. Based on the ability of the model to predict MRT and variance values to an R-squared value of above 0.9, this indicates good predictive accuracy of the ROM consisting of a serially connected ML and convection–diffusion model. This prediction result was considered reasonable by comparison to previous work in the literature on RTD modeling.[13,39] From Table 6, it is shown that the R^2 scores for the training data set appear smaller than the validation data set, which is an unexpected result. However, the R^2 score, while a commonly used metric to assess the goodness of fit of a regression model, can have misleading interpretations, especially when dealing with data sets of smaller sizes. Given $R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$, where SS_{res} and SS_{tot} indicate the sum of square of residuals and the total sum of squares, respectively. With fewer data points in the validation set, the ratio $\frac{SS_{res}}{SS_{tot}}$ can sometimes be smaller compared to the training set, potentially inflating the R^2 score for the validation set. Consequently, the validation R^2 may appear higher than the training R^2 due to data limitations. Thus, other additional metrics are also used to evaluate model performance, including the root-mean-squared error (RMSE) and parity plots. In this case, RMSE was a more reliable metric for model evaluation and was shown to follow the expected trend, exhibiting a lower RMSE on the training data set compared to the validation data set, indicating reasonable model performance given the available data constraints.

Figure 8 shows the comparison of the RTD curves between the CFD and ROM model predictions, and it is evident that the predictions are similar, further validating the accuracy of the ROM.

3.2.4. Prediction of RSD. Unlike the RTD curve and metrics, the RSD was not predicted in the CFD simulations. Thus, the RSD predictions from the ROM cannot be validated. However, since the velocities in both i and j directions are

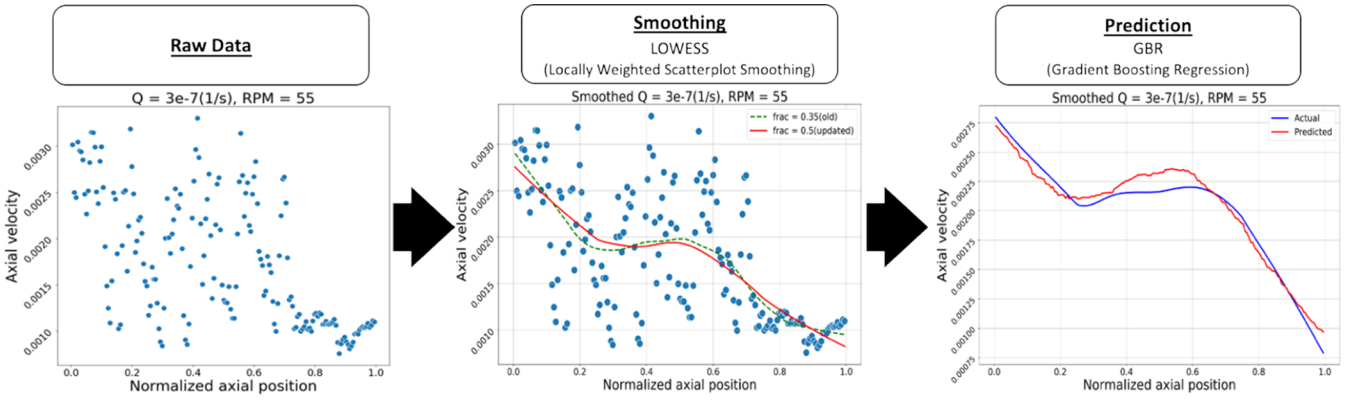


Figure 4. Process of data smoothing and model prediction of axial velocities.

https://doi.org/10.1021/acs.iecr.4c00201
Ind. Eng. Chem. Res. 2024, 63, 7271–7280

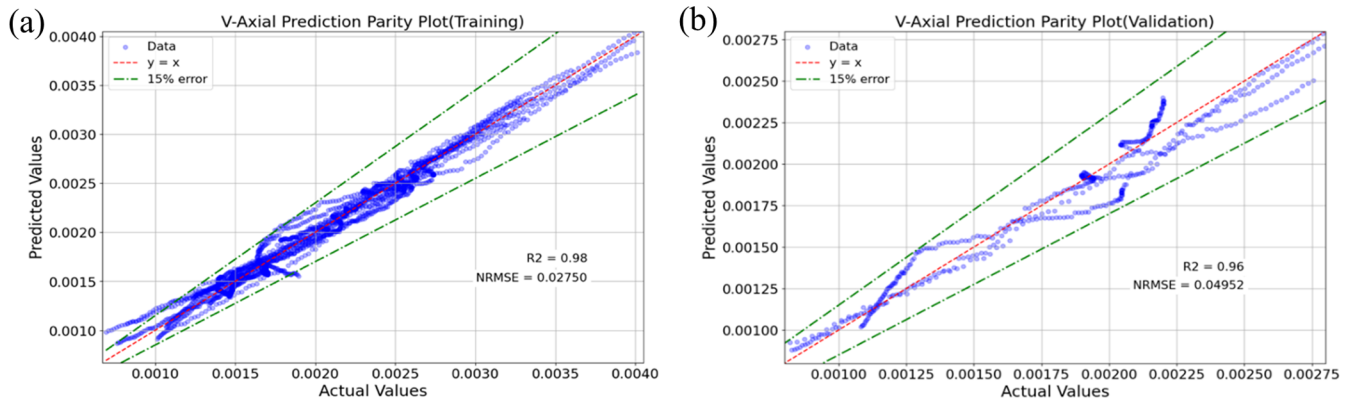


Figure 5. Parity plots for (a) training and (b) validation data sets of the model.

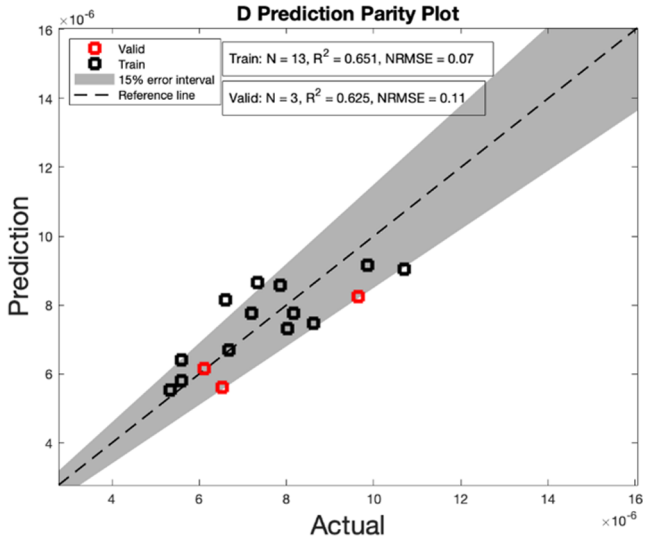


Figure 6. Parity plot for the diffusion coefficient.

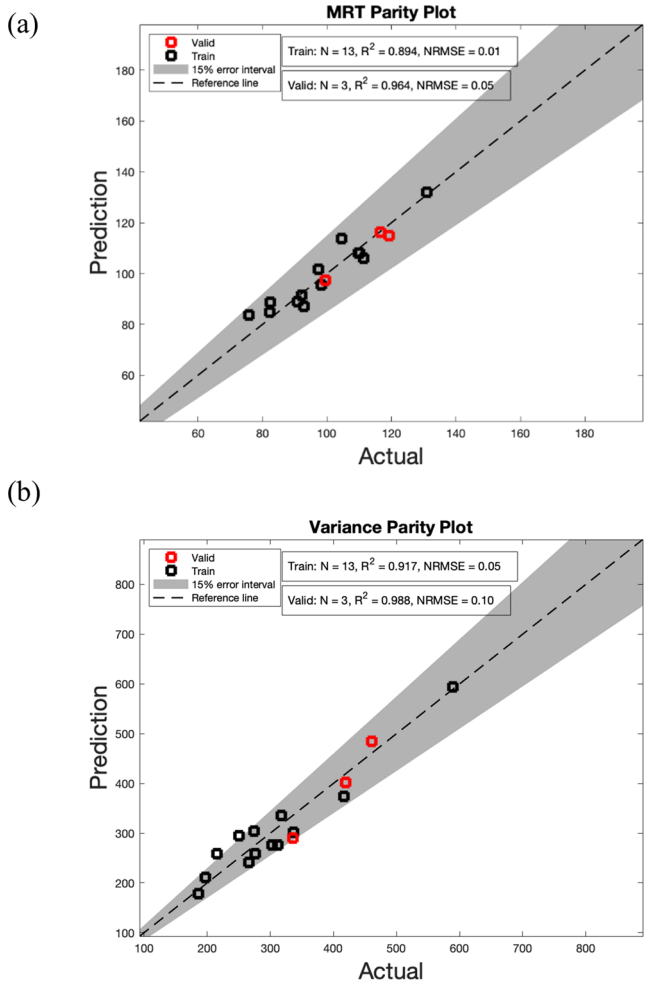


Figure 7. Parity plots for MRT and variance predictions.

determined by the CFD simulations, which are used as inputs to the convection−diffusion model, used for the prediction, it is likely that the relative differences between the RSD of each run are accurate. Figure 9 indicates the RSD at the extruder exit and the corresponding velocity (v_j) for the three validation runs specified in the above section, which are runs 6, 9, and 11. From Figure 9a, it is evident that the RSD decreases over time as the level of mixedness increases with time. This finding is in alignment with the observations of Portillo et al.[27] As for the relative differences between the three runs, it is shown that run 11, which has the lowest RSD, also has the highest velocity of the three runs. Higher velocity is attributed to the higher level of mixing, which improves uniformity of the product.

3.3. Computational Time. Significant time savings are captured when transitioning from the high-fidelity CFD model to the ROM, reducing the time taken to obtain an RTD prediction from approximately 43.5 h on a 20 core processor to 2.23 s on a one core processor. This highly supports the use of ROM for model applications, such as real-time predictions and optimization tasks. However, various tasks in the model development of the ROM require additional computational time and load. The breakdown for the duration and computational load required for different tasks are listed in Table 7. As detailed in the table, the data collection from the CFD simulation remains as the main time-consuming task, with the remainder of the model components requiring only approximately 5 min to complete.

Table 6. Model Performance Metrics for Prediction of MRT and Variance

parameter	data set	R^2	RMSE
MRT	training (13 runs)	0.895	0.01
	validation (3 runs)	0.964	0.05
variance	training (13 runs)	0.917	0.05
	validation (3 runs)	0.988	0.10

https://doi.org/10.1021/acs.iecr.4c00201
Ind. Eng. Chem. Res. 2024, 63, 7271−7280

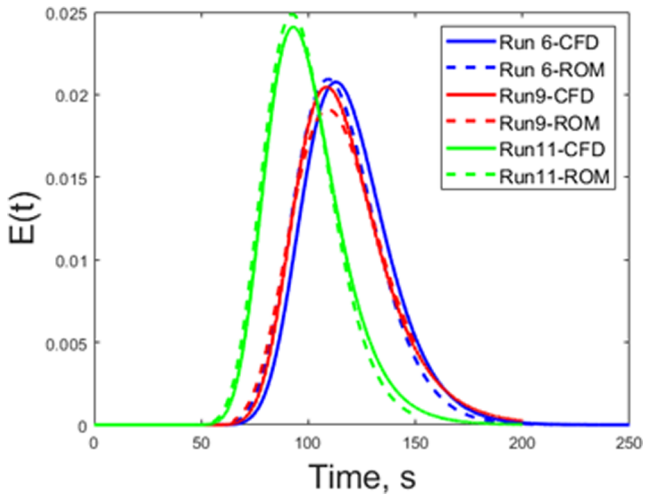


Figure 8. Comparison between the CFD and ROM predictions of the RTD curve for the three validation runs.

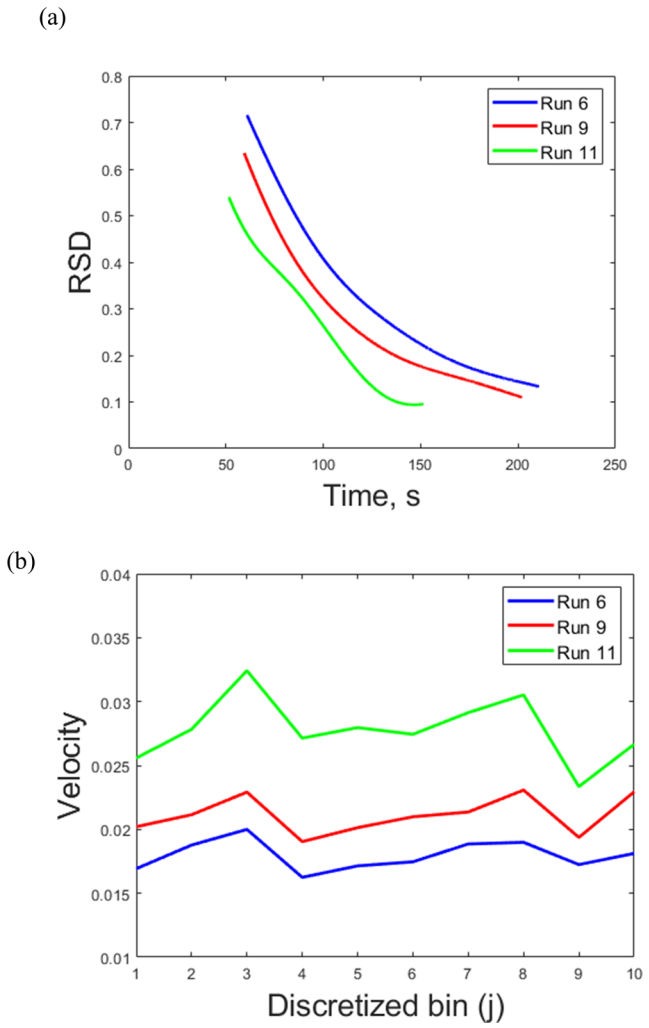


Figure 9. (a) RSD and (b) velocity in the *j*-direction for the three validation runs.

3.4. Process Design Space Using Mixing Metrics. The process design map illustrated in Figure 10 indicates the MRT and variance predictions for the design space used in this study. As shown in the figure, the highest MRT and variance correspond to the run with both the lowest throughput and screw speed. The same is true for the inverse, where the lowest MRT and variance correspond to the highest throughput and screw speed, as both parameters result in faster processing of the material in the extruder and lower mixedness. Additionally, a steady variance zone is also highlighted in red to indicate the area where variance remains unchanged despite significant changes in MRT. The data points in the steady variance zone are mostly made up of runs with a higher throughput. This can be interpreted as the high throughput runs, resulting in poor mixing (low variance), regardless of the screw speed. However, at a lower throughput, the screw speed significantly affects both the MRT and the variance of the system.

Table 7. Computational TIme Distribution between Different Model Development Tasks

task	model	duration (16 runs)	computation load
data collection	CFD	176 h	20 cores
parameter estimation	convection−diffusion	5 min	8 cores
velocity prediction	ML	1.25 s	1 core
diffusion prediction	ML	0.63 s	1 core

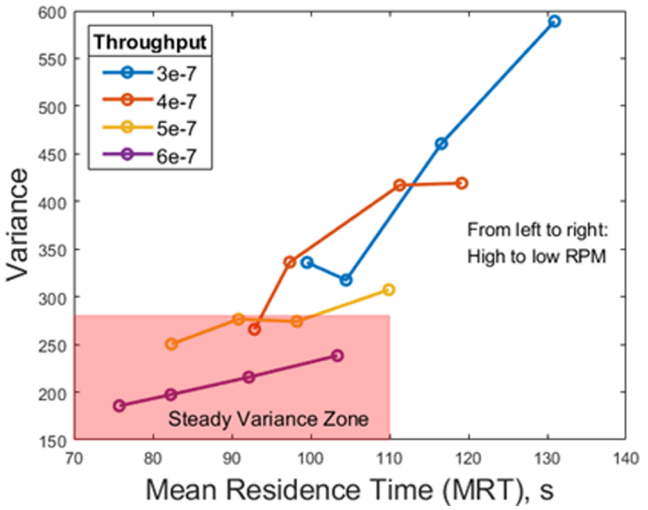


Figure 10. Process design map illustrating the MRT and variance of the design space.

3.5. Optimization. The ROM is also used for optimization to determine the optimal process parameters (mass throughput and screw speed) to obtain the desired mixing metrics. A hypothetical RTD target was set at an MRT of 90 s and a variance of 500. Table 7 shows the optimal mass throughput and screw speed that was determined via the two different optimization techniques and the corresponding MRT and variance values at those process parameters. The last column in the table details the computational time required to obtain a solution. As seen in Table 8, both optimization schemes resulted in similar results; however, the pareto optimization solution requires significantly less computational time to determine a solution. Thus, it can be used as a time efficient method to conduct process optimization. The pareto front for this work and method solution determination is illustrated in Figure 11

https://doi.org/10.1021/acs.iecr.4c00201
Ind. Eng. Chem. Res. 2024, 63, 7271−7280

Table 8. Optimization Solutions Were Based on Weighted and Pareto Optimization

optimization methods	mass throughput (1/s)	screw speed (rpm)	MRT (s)	variance	computational time (min)
weighted sum	4.5e−7	70.2	91.1	485.9	30
pareto	4.8e−7	70.5	90	425	3.5

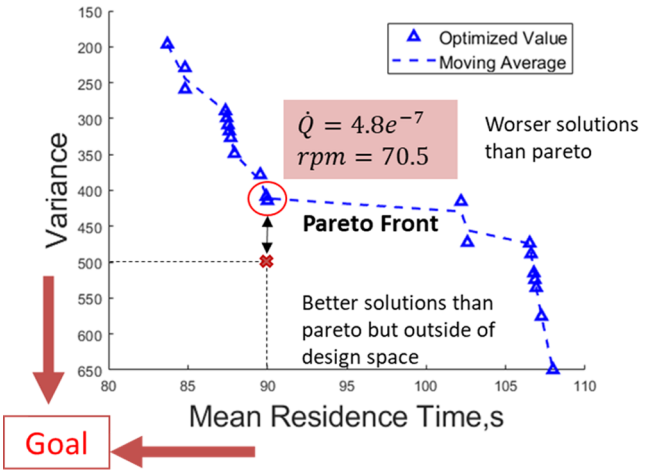


Figure 11. Pareto front to minimize MRT and maximize variance.

4. CONCLUSIONS

In conclusion, this study has detailed the methodology for developing an effective reduced order model (ROM) to predict mixing metrics of an extrusion system, specifically by employing a machine-learning-based convection−diffusion model. The model accuracy demonstrated by the ROM justifies the use of the ROM as a viable alternative to the high-fidelity CFD model. Notably, the ROM achieves a substantial reduction in computational time to produce the model prediction as compared to the CFD model, reducing the time from 44 h to a mere 2.2 s. The almost instantaneous prediction enables the use of the ROM for process design exploration and process optimization. The feasibility of utilizing the ROM for both these tasks was explored in this work to demonstrate the practicality of the ROM. Optimization on the developed ROM was shown to effectively identify the optimal process parameters for operating the extruder to ensure that the desired mixing metrics was achieved, leading to the production of the targeted product quality. This established framework can be seamlessly applied to the development of reduced order models of other systems by model training with system-specific data.

AUTHOR INFORMATION

Corresponding Author

Rohit Ramachandran − *Department of Chemical and Biochemical Engineering, Rutgers, The State University of New Jersey, Piscataway, New Jersey 08854, United States;* orcid.org/0000-0001-8086-7015; Email: rohitrr@soe.rutgers.edu

Authors

Ashley Dan − *Department of Chemical and Biochemical Engineering, Rutgers, The State University of New Jersey, Piscataway, New Jersey 08854, United States*

Urjit Patil − *Department of Statistics, Rutgers, The State University of New Jersey, Piscataway, New Jersey 08854, United States*

Erik Holmen Olofsson − *Department of Mechanical Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*

Jesper Henri Hattel − *Department of Mechanical Engineering, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.iecr.4c00201

Author Contributions

A.D., E.H.O., and R.R.: conceptualization; A.D., E.H.O., U.P., and R.R.: methodology; A.D. and R.R.: formal analysis; A.D. and R.R.: investigation; A.D. and U.P.: writing—original draft preparation; A.D., U.P., E.H.O., J.H.H., and R.R.: writing—review [TC6]and editing; A.D. and U.P.: visualization; R.R.: supervision and project administration and funding acquisition. All authors have given approval to the final version of the manuscript.

Funding

Any funds used to support the research of the manuscript should be placed here (per journal style).

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors would like to acknowledge funding support from the Department of Chemical and Biochemical Engineering, Rutgers University.

REFERENCES

(1) Gao, Y.; Muzzio, F. J.; Ierapetritou, M. G. A review of the Residence Time Distribution (RTD) applications in solid unit operations. *Powder Technol.* **2012**, *228*, 416−423.

(2) Danckwerts, P. V. Continuous flow systems: Distribution of residence times. *Chem. Eng. Sci.* **1953**, *2* (1), 1−13.

(3) Fogler, H. S. *Elements of Chemical Reaction Engineering*; Pearson, 2020.

(4) Wen, C.-Y.; Fan, L. T. *Models for Flow Systems and Chemical Reactors*; Dekker, 1975.

(5) Bhalode, P.; Tian, H.; Gupta, S.; Razavi, S. M.; Roman-Ospino, A.; Talebian, S.; Singh, R.; Scicolone, J. V.; Muzzio, F. J.; Ierapetritou, M. Using residence time distribution in pharmaceutical solid dose manufacturing−A critical review. *Int. J. Pharm.* **2021**, *610*, No. 121248.

(6) Gao, Y.; Muzzio, F. J.; Ierapetritou, M. G. Optimizing continuous powder mixing processes using periodic section modeling. *Chem. Eng. Sci.* **2012**, *80*, 70−80.

(7) Sen, M.; Karkala, S.; Panikar, S.; Lyngberg, O.; Johnson, M.; Marchut, A.; Schäfer, E.; Ramachandran, R. Analyzing the mixing dynamics of an industrial batch bin blender via discrete element modeling method. *Processes* **2017**, *5* (2), 22.

(8) Bhalode, P.; Ierapetritou, M. A review of existing mixing indices in solid-based continuous blending operations. *Powder Technol.* **2020**, *373*, 195−209.

(9) Vanarase, A. U.; Järvinen, M.; Paaso, J.; Muzzio, F. J. Development of a methodology to estimate error in the on-line measurements of blend uniformity in a continuous powder mixing process. *Powder Technol.* **2013**, *241*, 263−271.

(10) Kumar, A.; Ganjyal, G. M.; Jones, D. D.; Hanna, M. A. Modeling residence time distribution in a twin-screw extruder as a series of ideal steady-state flow reactors. *J. Food Eng.* **2008**, *84* (3), 441−448.

(11) Matić, J.; Alva, C.; Witschnigg, A.; Eder, S.; Reusch, K.; Paudel, A.; Khinast, J. Towards predicting the product quality in hot-melt extrusion: Small scale extrusion. *Int. J. Pharm.: X* **2020**, *2*, No. 100062.
(12) Vanarase, A. U.; Muzzio, F. J. Effect of operating conditions and design parameters in a continuous powder mixer. *Powder Technol.* **2011**, *208* (1), 26−36.
(13) Kumar, A.; Alakarjula, M.; Vanhoorne, V.; Toiviainen, M.; De Leersnyder, F.; Vercruysse, J.; Juuti, M.; Ketolainen, J.; Vervaet, C.; Remon, J. P.; et al. Linking granulation performance with residence time and granulation liquid distributions in twin-screw granulation: An experimental investigation. *Eur. J. Pharm. Sci.* **2016**, *90*, 25−37.
(14) Kotamarthy, L.; Ramachandran, R. Mechanistic understanding of the effects of process and design parameters on the mixing dynamics in continuous twin-screw granulation. *Powder Technol.* **2021**, *390*, 73−85.
(15) Rodrigues, A. E. Residence time distribution (RTD) revisited. *Chem. Eng. Sci.* **2021**, *230*, No. 116188.
(16) Sebastian Escotet-Espinoza, M.; Escotet-Espinoza, M. S.; Moghtadernejad, S.; Oka, S.; Wang, Y.; Roman-Ospino, A.; Schäfer, E.; Cappuyns, P.; Van Assche, I.; Futran, M.; Ierapetritou, M. Effect of tracer material properties on the residence time distribution (RTD) of continuous powder blending operations. Part I of II: Experimental evaluation. *Powder Technol.* **2019**, *342*, 744−763.
(17) Olofsson, E. H.; Roland, M.; Spangenberg, J.; Jokil, N. H.; Hattel, J. H. A CFD model with free surface tracking: predicting fill level and residence time in a starve-fed single-screw extruder. *Int. J. Adv. Des. Manuf. Technol.* **2023**, *126* (7−8), 3579−3591.
(18) Dubey, A.; Vanarase, A. U.; Muzzio, F. J. Impact of process parameters on critical performance attributes of a continuous blender—A DEM-based study. *AIChE J.* **2012**, *58* (12), 3676−3684.
(19) Razavi, S. M.; Román-Ospino, A. D.; Bhalode, P.; Scicolone, J.; Callegari, G.; Dubey, A.; Koolivand, A.; Krull, S.; Tian, G.; Xu, X.; et al. Selection of an appropriate tracer to measure the residence time distribution (RTD) of continuous powder blending operations. *Powder Technol.* **2023**, *429*, No. 118864.
(20) Sen, M.; Dubey, A.; Singh, R.; Ramachandran, R. Mathematical development and comparison of a hybrid PBM-DEM description of a continuous powder mixing process. *J. Powder Technol.* **2013**, *2013*, No. 843784, DOI: 10.1155/2013/843784.
(21) Dubey, A.; Sarkar, A.; Ierapetritou, M.; Wassgren, C. R.; Muzzio, F. J. Computational approaches for studying the granular dynamics of continuous blending processes, 1−DEM based methods. *Macromol. Mater. Eng.* **2011**, *296* (3−4), 290−307.
(22) Lang, Y.-d.; Malacina, A.; Biegler, L. T.; Munteanu, S.; Madsen, J. I.; Zitney, S. E. Reduced order model based on principal component analysis for process simulation and optimization. *Energy Fuels* **2009**, *23* (3), 1695−1706.
(23) Boukouvala, F.; Gao, Y.; Muzzio, F.; Ierapetritou, M. G. Reduced-order discrete element method modeling. *Chem. Eng. Sci.* **2013**, *95*, 12−26.
(24) Nauman, E.; Mallikarjun, R. Generalized boundary conditions for the axial dispersion model. *Chem. Eng. J.* **1983**, *26* (3), 231−237.
(25) Wilczyński, K.; Lewandowski, A.; Wilczyński, K. J. Experimental study for starve-fed single screw extrusion of thermoplastics. *Polym. Eng. Sci.* **2012**, *52* (6), 1258−1270.
(26) Wilczyński, K. J.; Lewandowski, A.; Nastaj, A.; Wilczyński, K. Modeling for Starve Fed/Flood Fed Mixing Single-Screw Extruders. *Int. Polym. Process.* **2016**, *31* (1), 82−91.
(27) Portillo, P. M.; Muzzio, F. J.; Ierapetritou, M. G. Using compartment modeling to investigate mixing behavior of a continuous mixer. *J. Pharm. Innov.* **2008**, *3*, 161−174.
(28) Bird, R. B. Transport phenomena. *Appl. Mech. Rev.* **2002**, *55* (1), R1−R4.
(29) Velleman, P. F. Definition and comparison of robust nonlinear data smoothing algorithms. *J. Am. Stat. Assoc.* **1980**, *75* (371), 609−615.
(30) Jacoby, W. G. Loess:: a nonparametric, graphical tool for depicting relationships between variables. *Elect. Stud.* **2000**, *19* (4), 577−613.
(31) Cleveland, W. S. Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.* **1979**, *74* (368), 829−836.
(32) Allen, D. M. The relationship between variable selection and data agumentation and a method for prediction. *Technometrics* **1974**, *16* (1), 125−127.
(33) Derkacheva, A.; Mouginot, J.; Millan, R.; Maier, N.; Gillet-Chaulet, F. Data reduction using statistical and regression approaches for ice velocity derived by landsat-8, sentinel-1 and sentinel-2. *Remote Sen.* **2020**, *12* (12), 1935.
(34) Zemel, R.; Pitassi, T. A Gradient-Based Boosting Algorithm for Regression Problems. In *Advances in Neural Information Processing Systems*; MIT Press, 2000; Vol. *13*, pp 696−702.
(35) Sollich, P.; Krogh, A. Learning with Ensembles: How Overfitting Can Be Useful. In *Advances in Neural Information Processing Systems*; MIT Press, 1995; Vol. *8*, pp 190−196.
(36) Natekin, A.; Knoll, A. Gradient boosting machines, a tutorial. *Front. Neurorobot.* **2013**, *7*, 21.
(37) Friedman, J. H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38* (4), 367−378.
(38) Elith, J.; Leathwick, J. R.; Hastie, T. A working guide to boosted regression trees. *J. Anim. Ecol.* **2008**, *77* (4), 802−813.
(39) Escotet-Espinoza, M. S.; Moghtadernejad, S.; Oka, S.; Wang, Z.; Wang, Y.; Roman-Ospino, A.; Schäfer, E.; Cappuyns, P.; Van Assche, I.; Futran, M.; et al. Effect of material properties on the residence time distribution (RTD) characterization of powder blending unit operations. Part II of II: Application of models. *Powder Technol.* **2019**, *344*, 525−544.

https://doi.org/10.1021/acs.iecr.4c00201
Ind. Eng. Chem. Res. 2024, 63, 7271−7280