Abstract: Concentration control (C-control) strategy for (semi-)batch antisolvent crystallization processes has been recently developed with the aid of new sensors that measure in situ process variables. This control strategy gives better robustness over traditional flowrate/temperature control in the presence of process disturbances. However, the setpoint value for the existing C-control is determined through trial-and-error procedure and hence gives sub-optimal product quality in most of the cases. This motivates the current study to develop a two-staged integrated data-based approach to facilitate the determination of setpoint value for the implementation of C-control strategy in anti-solvent crystallization processes. In the first stage of the proposed design, a k-nearest neighborhood LSSVM (k-NN LSSVM) is used to select a subset of the training database that resembles the current batch dynamics as the relevant training database, which is subsequently used in the second stage for determination of the setpoint value of C-control. Simulation results show that the performance of proposed design is near to the best achievable performance of C-control strategy obtained with a known first-principles model for controlling the semi-batch antisolvent crystallization process.

Keywords: Concentration control; Least squares support vector machine; Pattern classification; Nonlinear partial least squares

1. INTRODUCTION

Traditionally, the optimal antisolvent addition rate as a function of time obtained from an offline process model is given as setpoint for the regulatory level controller to ensure its tight tracking. However, this approach has been found to be highly sensitive to process disturbances like variations in kinetics due to foreign contaminants, inhibitors and admixtures in the solvent. With the recent advancements in sensor technology, online process monitoring and control of pharmaceutical processes has become possible in the bench and even industrial scale. Consequently, model-free direct design approaches like C-control and nucleation control strategies, which use solute concentration and in situ chord length distribution, respectively, as feedback have been developed (Patience and Rawlings [2001], Woo et al. [2009]). These approaches were found to be relatively less sensitive to process disturbances and variations, due to their closed-loop nature. However, the direct design approaches suffer from being operated at the sub-optimal level, as they completely rely on trial and error procedure to determine the optimal setpoint, which is not only tedious and costly to perform the required experiments, but also requires considerable engineering efforts. Furthermore, these strategies have varying batch times, which may sometimes pose as a bottleneck for the smooth operations at industrial scale and may also lead to batch-to-batch variability, if not adapted after each run. Hence, tuning of the supersaturation setpoint for optimal control is crucial to steer the process to obtain the desired product specifications (Zhou et al. [2006]).

To alleviate the drawback of existing C-control strategy, an integrated two-staged procedure is developed to facilitate the determination of setpoint for the C-control strategy. In the first stage, a k-nearest neighborhood LSSVM (k-NN LSSVM) is used for pattern classification in order to select the training database which resembles the current process dynamics as the relevant training database, which is used in the second stage for determination of the setpoint value to obtain optimal product quality. Furthermore, the Just-in-Time Learning (JITL) technique is employed for predicting solute concentration values, while a LSSVM inner relationship based multiway partial least squares (LSSVM-MPLS) model is used for predicting the product quality. Simulation results show that the performance of proposed design is near to the best achievable performance of C-control strategy obtained with a known first-principle model for controlling the semi-batch antisolvent crystallization process.

2. OVERVIEW OF TOOLS USED FOR PROPOSED DESIGN

2.1 Overview of LSSVM for multiclass classification

Typically, multiclass problem with M classes is solved by reformulating it into a set of L binary classification problems (van Gestel et al. [2004]). In this respect, the methodology of binary classification using LSSVM is briefly discussed here to facilitate the ensuing discussions.

LSSVMs for binary classification: The LSSVM algorithm proposed by Suykens and Vandewalle [1999] is used...
in this study. Consider a model in the primal weight space given by:
\[ y(x) = \text{sign}(w^T \phi(x) + b), \] (1)
Given a set of \( N \) training samples \( \{x_k, y_k\}_{k=1}^N \), where \( x_k \in \mathbb{R}^p \) is the \( k \)th input and \( y_k \in \{-1, +1\} \) is the corresponding class label, the SVM classifier satisfies the following conditions:
\[ \begin{cases} w^T \phi(x_k) + b \geq +1, & \text{if } y_k = +1 \\ w^T \phi(x_k) + b \leq -1, & \text{if } y_k = -1 \end{cases} \] (2)
which is equivalent to
\[ y_k[w^T \phi(x_k) + b] \geq 1 \] (3)
where the nonlinear function \( \phi(\cdot) \) maps the input \( x \) to a high dimensional feature space. However, in order to evaluate \( y(x) \) using Eq. (1), the parameters \( w \) and \( b \) must be obtained. Hence, an optimization problem is formulated as given in Eq. (4).
\[ \min_{w, b, e} J_F(w, b, e) = \frac{1}{2} w^T w + \gamma \sum_{k=1}^N e_k^2 \] (4)
subject to the equality constraints
\[ y_k \{w^T \phi(x_k) + b\} = 1 - e_k, \quad k = 1, \ldots, N. \] (5)
The solution of Eq. (4) is obtained through the Lagrangian:
\[ L(w, b, e; \alpha) = J_F(w, b, e) - \sum_{k=1}^N \alpha_k \{y_k \{w^T \phi(x_k) + b\} - 1 + e_k\}, \] (6)
where \( \alpha_k \in \mathbb{R} \) are the Lagrange multipliers that can be positive or negative in the LSSVM formulation. From the conditions for optimality, the Karush-Kuhn-Tucker (KKT) system of equations lead to the elimination of \( w \) and \( e \) to obtain
\[ \begin{bmatrix} 0 \\ y^T \Omega + \gamma \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 1_v \end{bmatrix} \] (7)
where \( y = [y_1, \ldots, y_N], 1_v = [1, \ldots, 1], e = [e_1, \ldots, e_N], \alpha = [\alpha_1, \ldots, \alpha_N] \). Thus, by applying the Mercer’s condition within the \( \Omega \) matrix, we get
\[ \Omega_{ij} = y_i y_j \phi(x_i)^T \phi(x_j) = y_i y_j K(x_i, x_j). \] (8)
In this study, Radial Basis Function (RBF) kernel \( K(\cdot, \cdot) \) given in Eq. (9) is used
\[ K(x, x_k) = \exp \left( -\frac{\|x - x_k\|^2}{\sigma^2} \right) \] (9)
where \( \sigma \) is a constant, which takes care of the scaling of the inputs in the RBF kernel function. As shown in Eq. (7), the set of linear equations are solved in the dual space (large scale algorithm) in order to obtain the parameters of the LSSVM classifier as shown in Eq. (10).
\[ y(x) = \text{sign} \left[ \sum_{l=1}^{N} \alpha_l y_l K(x, x_l) + b \right]. \] (10)

There exists different approaches to construct the set of binary classifiers. In this study, one-versus-one (1-vs-1) output coding approach is used, during which \( L = M(M-1)/2 \) binary classifiers are plugged in, where each of which discriminates between two opposing classes. Each binary classifier \( f(l)(x) \), \( l = 1, \ldots, L \), is inferred on the training set \( D(l) = \{(x_k, y_k(l)) \mid k = 1, \ldots, N \text{ and } y_k(l) \in \{-1, +1\}\} \), consisting of \( N(l) \leq N \) training points, by solving
\[ \begin{bmatrix} 0 \\ y(l)^T \Omega(l) + \gamma(l) \end{bmatrix} \begin{bmatrix} b(l) \\ \alpha(l) \end{bmatrix} = \begin{bmatrix} 0 \\ 1_v \end{bmatrix} \] (11)
where \( \Omega_{ij,l} = K_l(x_i, x_j) \). The binary classifier \( f(l)(x) \) is then obtained as \( f(l)(x) = \text{sign}(\sum_{k=1}^{N(l)} \alpha_k(l) K(l)(x, x_k) + b(l)) \). Thus, each of these \( L \) classifiers assign an output bit \( y(l) = \text{sign}(f(l)(x)) \), such that the unique codeword \( c \) to a new input vector \( x \) can be reconstructed to predict its corresponding class (van Gestel et al. [2004]).

For this study, LS-SVMLab: a MATLAB® / C toolbox is used for implementing the LSSVM algorithms (Peleckmans et al. [2002]). Furthermore, motivated to improve the existing LSSVM method, a k-nearest neighborhood based LSSVM (k-NN LSSVM) algorithm is used in this study. In a slightly modified form, the k-NN LSSVM selects the subset of training data from the entire database based on the similarity criterion with respect to the query vector, trains the model and uses it for classifying that specific query data alone. This procedure repeats for the next query data.

2.2 Unfolding the data matrix: multiway approach

The historical data of the process variables collected for all the batches were used to form the 3-D matrix of the form \( X \) (I batches \( \times \) J variables \( \times K \) sampling instances). Also, the information regarding the corresponding class of dynamics that each of these batch data belongs to \( Y_{\text{class}} \) (I batches \( \times \) C class of the dynamics) and the product quality data \( Y \) (I batches \( \times \) M product qualities) for each batch data are also collected, which are later used during the training of the pattern classifier and nonlinear regression model, respectively.

After the data acquisition step, the batch data to be classified is preprocessed using MPCA to reduce the dimensionality of the variables (Nomikos and MacGregor [1994]). For the application of MPCA, the input data \( X \) should be unfolded by treating each of the variable value at each sampling instant as a separate variable as shown in Eq. (12). Thus, \( X \) can now be represented as \( X = (I \text{ batches } \times \text{ J } \text{ variables at each sampling instance}) \). Now, each of these \( JK \) variables (treated as independent variables) for all the \( I \) batches are autoscaled using the corresponding values of mean and standard deviation. Based on the variance captured, the first \( h \) principal components are retained in the lower dimensional space. The training data set for the pattern classification algorithm is obtained for the \( I \) batches whose input data consists of the first \( h \) scores, while the output is the vector containing the information regarding the class of dynamics which each of these \( I \) batches belong to. For new batch data whose dynamics is unknown, the query data vecor consisting of \( JK \) input
variables is autoscaled using the mean and standard deviation values determined earlier and projected onto the lower dimensional space. The corresponding scores obtained for this batch are treated as test data. The LSSVM based pattern classifier is trained using the training data and the trained LSSVM model is used to determine the specific dynamics of the test data.

\[
X = \begin{bmatrix}
i = 1, k = 1 & x_1 x_2 \ldots x_j & x_1 x_2 \ldots x_j & \ldots & x_1 x_2 \ldots x_j \\
i = 1, k = 2 & x_1 x_2 \ldots x_j & x_1 x_2 \ldots x_j & \ldots & x_1 x_2 \ldots x_j \\
i = 2, k = 1 & x_1 x_2 \ldots x_j & x_1 x_2 \ldots x_j & \ldots & x_1 x_2 \ldots x_j \\
i = 2, k = 2 & x_1 x_2 \ldots x_j & x_1 x_2 \ldots x_j & \ldots & x_1 x_2 \ldots x_j \\
i = 1, k = K & x_1 x_2 \ldots x_j & x_1 x_2 \ldots x_j & \ldots & x_1 x_2 \ldots x_j \\
i = K, k = 1 & x_1 x_2 \ldots x_j & x_1 x_2 \ldots x_j & \ldots & x_1 x_2 \ldots x_j \\
i = K, k = K & x_1 x_2 \ldots x_j & x_1 x_2 \ldots x_j & \ldots & x_1 x_2 \ldots x_j \\
\end{bmatrix}
\]  

(12)

2.3 Nonlinear dynamic modeling of batch processes

The second stage of the proposed framework makes use of the predictions of the pattern classifier to find the relevant samples in the complete database for prediction of process variables during the batch and product quality variables at the end of batch. Thus, the subset of the complete database of the training data that follow the same dynamics as that of the predicted class for the test data is chosen as the relevant training data for calculating the dynamics for the next batch. In this paper, the Just-In-Time-Learning (JITL) modeling method (Cheng and Chiu [2004]) is adopted to predict solute concentration during the batch. The JITL method is briefly reviewed as follows; given a database \((y_i, x_i)\), the parameters \(k_{min}, k_{max}\) and weight parameter \(\gamma\), and a query data \(x_q\). The first step is to determine the relevant data based on a similarity measure using distance and angular metrics. The next step is to predict the output variables throughout the batch using the relevant data and the following ARX model:

\[
g(k) = z^T(k - 1) \Psi
\]  

(13)

where \(\hat{g}(k)\) is the model output at the \(k\)th sampling instant, \(z(k-1)\) is the regression vector, and \(\Psi\) is the model parameter vector as given by

\[
z(k - 1) = [y(k - 1), \ldots, y(k - n_y), u(k - n_d - 1), \ldots, u(k - n_d - n_u)]^T,
\]  

(14)

\[
\Psi = [\psi_1, \ldots, \psi_{n_y}, \psi_{n_y+1}, \ldots, \psi_{n_y+n_u}]^T,
\]  

(15)

where \(n_y\) and \(n_u\) are integers related to the model’s order, and \(n_d\) is the process time delay.

2.4 Nonlinear product quality modeling

As explained earlier in section 2.3, during the dynamic modeling of the batch, the process was approximated as locally linear using a first order ARX model in the JITL framework. However, the linear relationship does not hold for product quality prediction. Hence, a nonlinear regression model is often built for this purpose. The most simple yet effective nonlinear model is the quadratic regression model (Lee and Lee [2003]), of the form

\[
q = A_1 + B_1 u + B_2 u \odot u + \tau
\]  

(16)

where \(\odot\) denotes the element-wise multiplication. Note that \(B_2 u \odot u = B_2 \text{diag}(u)u\). However, multivariate statistical models like linear PLS models are often used for product quality predictions and on-line monitoring of batch processes. Also, nonlinear PLS modeling techniques have been introduced by considering different nonlinear inner relationship between the scores of the inputs and the outputs like quadratic regression models (Wold et al. [1989]) and artificial NNets (Qin and McAvoy [1992]).

During this study, the inner relationship of the scores of the input and output scores are modeled using a series of LSSVM Regression (LSSVR) models, one for each of the scores. To justify the use of this nonlinear relationship, comparison between the linear and LSSVR inner relationship based MPLS models is also presented in the case study discussed later. The details of the implemented algorithm can be found in (Li et al. [2006]).

LSSVM for nonlinear regression: LSSVM is not only used for multiclass classification as discussed earlier in the section 2.1, but also for nonlinear regression during this study. The approach is very similar as explained earlier, however the model in the primal weight space will be now of the form shown in Eq. (17).

\[
y(x) = w^T \varphi(x) + b
\]  

(17)

The parameters of Eq. (19) result from the solution of the optimization problem Eq. (4), subject to the equality constraints given by Eq. (18).

\[
y_k = w^T \varphi(x_k) + b + \epsilon_k, \quad k = 1, \ldots, N.
\]  

(18)

\[
y(x) = \sum_{k=1}^{N} \alpha_k K(x, x_k) + b.
\]  

(19)

2.5 Integrating the pattern classifier and the nonlinear modeling framework

The flowchart shown in Figure 1 describes the integration of the pattern classification tool with the nonlinear dynamic and product quality modeling technique.

3. CASE STUDY: SEMI-BATCH ANTISOLVENT CRYSTALIZATION PROCESS

3.1 Process model

In this study, a laboratory scale, isothermal, seeded antisolvant crystallization process model for paracetamol in acetone-water mixture is considered (Granberg and Rasmuson [2000, 2005], Granberg et al. [2001], Woo et al. [2009]). The corresponding expressions for the crystal kinetics and solubility are summarized as follows

\[
B = k_0 \Delta C^6
\]  

(20)

\[
G = k_9 \Delta C^9
\]  

(21)

and
\[ C_{sat} = 1.0559 - 2.048 \times 10^{-2} m_w + 1 \times 10^{-4} m_w^2, \]
\[ (60\% \leq m_w \leq 80\%) \]
\[ k_b = 4.338 \times 10^{58} \exp(-1.374m_w) \]
\[ b = 1.997 \times 10^{-3} m_w^2 - 6.237 \times 10^{-1} m_w + 40.42 \]
\[ k_g = -9.63 \times 10^{-11} m_w^3 + 3.3558 \times 10^{-4} m_w^2 \]
\[ -1.2606 \times 10^{-6} m_w + 3.6852 \times 10^{-5} \]
\[ g = -1.108 \times 10^{-4} m_w + 1.024 \times 10^{-2} m_w + 1.427 \]

where \( B \) denotes the nucleation rate, \( G \) growth rate, \( m_w \) the antisolvent mass percent on the solute-free basis, \( k_b \) and \( b \) the nucleation kinetics, \( k_g \) and \( g \) the growth kinetics, \( C \) the solute concentration, \( C_{sat} \) the saturation concentration, and \( \Delta C = (C - C_{sat}) \) supersaturation.

### C-control strategy

The C-control developed for semi-batch antisolvent crystallization processes is reviewed in this section (Yu et al. [2006], Woo et al. [2009]). Denote the solute concentration \( C \) at instant \( k \) by

\[ C_k = \frac{M_{\text{solute},k}}{M_{\text{solvent}} + M_{w,k}} \]

where \( M_{\text{solute},k} \) is the mass of solute at instant \( k \), \( M_{\text{solvent}} \) is the mass of solvent, and \( M_{w,k} \) is the mass of antisolvent at instant \( k \).

Figure 2 depicts the closed-loop C-control strategy. The idea of C-control is to determine the setpoint values for the antisolvent mass percent, \( m_{w,k+1}^{\text{set}} \), according to the Eq. (24) so as to maintain constant relative supersaturation (Rel SS), which is specified by trial and error.

\[ \frac{C_{k+1}}{C_{sat}(m_{w,k+1}^{\text{set}})} - 1 = \left( \frac{\Delta C}{C_{sat}} \right)^{\text{set}} \]

(24)

By taking the dilution effect into account, Eq. (24) can now be reformulated as

\[ C_{sat}(m_{w,k+1}^{\text{set}}) \left( 1 + \frac{\Delta C}{C_{sat}} \right)^{\text{set}} + \]
\[ M_{\text{solute},k} \frac{m_{w,k+1}^{\text{set}}}{100} - 1 = 0 \]

where \( M_{w,k} \) is the antisolvent flowrate to be implemented at the current sampling instant.

As a case study, five different dynamics are considered assuming that the nucleation and growth kinetic parameters given in Eqs. (20) and (21) are subject to the following perturbations.

\[ g^* = g(1 + \theta_1), \]
\[ \ln(k_b^*) = \ln(k_g)(1 + \theta_2), \]
\[ b^* = b(1 + \theta_3), \]
\[ \ln(k_g^*) = \ln(k_b)(1 + \theta_4), \]
\[ C_{sat}^* = C_{sat}(1 + \theta_5), \]

where \( \theta_1 \) and \( \theta_2 \) are the uncertainties in the growth kinetics; \( \theta_3 \) and \( \theta_4 \) are the uncertainties in the nucleation kinetics; \( \theta_5 \) is the uncertainty in the solubility curve of paracetamol in acetone-water system. The five perturbations considered along with the range of setpoint values considered for C-control for each specific dynamics are summarized in Table 1.

### Table 1 Different dynamics of the process

<table>
<thead>
<tr>
<th>Case</th>
<th>Dynamics</th>
<th>Perturbations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Nominal</td>
<td>( \theta_1 )</td>
</tr>
<tr>
<td>1</td>
<td>(0.077 0.12)</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>High nucleation and low growth (0.117 0.2)</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>Low nucleation and high growth (1e-5 0.035)</td>
<td>-0.2</td>
</tr>
<tr>
<td>4</td>
<td>High nucleation and high growth (1e-5 0.035)</td>
<td>-0.2</td>
</tr>
<tr>
<td>5</td>
<td>Low nucleation and low growth (0.117 0.2)</td>
<td>0.2</td>
</tr>
</tbody>
</table>

The two-staged framework developed in this study integrates pattern classification and product quality prediction.
for optimal operation even during the presence of process variations. For both classification and nonlinear regression, LSSVM acts as the backbone. The details of the framework is explained as follows:

(i) The database consists of process data of antisolvent mass percent and solute concentration at each sampling time corresponding to 30 batches for each of the 5 dynamics listed above. Thus, the database consists of 150 batch data points. The data is partitioned into training and testing data by random selection, with 121 batch data forming the training dataset while the rest 29 batch data form the testing dataset.

(ii) Before the LSSVM is applied onto the testing data in order to classify each of the batch data into one of the 5 dynamics, the data is preprocessed using multi-way principal component analysis (MPCA). Eq. (27) shows the way in which the process data is unfolded. By considering the percentage of variance captured in the data, the first 10 principal components are retained in order to train the LSSVM for pattern classification. The testing data is autoscaled using the mean and standard deviation of the training data and is projected onto the first 10 principal components.

(iii) Using the training data with reduced dimensionality, LSSVM is trained for multi-class classification. In order to improve the performance of the standard LSSVM, an improved k-nearest neighborhood LSSVM is used, as explained earlier in section 2.1. The k (here, set as 30) nearest neighbors of the test data among the training data are selected and sorted based on their distance measure. This subset of relevant batch data is used as the training data for multi-class classification using k-NN LSSVM. This procedure is repeated to determine the specific class of dynamics that each of the test batch data will follow. Thus, once the dynamics or the specific class of the test data is determined, all the batch data in the training dataset that fall into this class are selected as highly relevant data set for subsequent modeling. This forms the first-stage of the framework, where a subset of training samples are selected for the prediction of solute concentration and product quality.

(iv) Now, in order to determine the value of the relative supersaturation setpoint that results in optimal product quality for the given class or dynamics, a global optimization algorithm is initiated using random initial guesses. The bounds on the decision variable are considered based on the minimum and maximum value of the relative supersaturation setpoint value found in the highly relevant data set.

(v) For the initial guess (if iteration = 1) or updated decision variable (if iteration > 1) value of relative supersaturation setpoint, the process data is predicted using a first order ARX model. In order to improve its predictive ability, JITL framework is used. Thus, a JITL based first order ARX model provides the predictions of the solute concentration. The antisolvent mass percent profiles for the entire batch are thus obtained as a result of the implementation of the C-control strategy.

For the simulation of the process, a first order ARX model based on JITL framework is used to predict the concentration values ($\hat{C}(k)$) at each sampling instant, $k$, given the values of $C(k-1)$ and $(m_w(k-1))$. Thus, the first order ARX model is given as

$$\hat{C}(k) = \alpha \hat{C}(k-1) + \beta m_w(k-1)$$

Note, that the initial conditions for $C(0)$ and $m_w(0)$ are known at the start of each batch. The values of $k_{min}$, $k_{max}$ and $\gamma$ were chosen to be 8, 60 and 1, respectively. The database is constructed using the relevant training data. Using this JITL based first order ARX model to predict the solute concentration values at each sampling time, the closed loop implementation of the C-control strategy is carried out until the end of the batch. Thus, given the initial conditions and the relative supersaturation setpoint value, the JITL based model predicts the solute concentration at each sampling instant until the end of the batch. As the C-control strategy finds the antisolvent mass percent values through the solution of Eq. (25), the batch data as represented in the matrix $X$ (of Eq. (27)) is obtained.

(vi) Using the highly relevant dataset as the training data, a LSSVM-MPLS model is built in order to predict the product quality (i.e., yield and volume-weighted mean size of the product crystals). The predicted concentration and antisolvent mass percent profiles in Step (v) are used as the test data for the trained LSSVM-MPLS model. With an objective to maximize the volume-weighted product quality and constrained by the minimum product yield and the bounds on the relative supersaturation setpoint, the optimization algorithm initiated at Step (iv) iterates until the stopping criteria is satisfied. Set, $iteration = iteration + 1$ and go to Step (v). Iterate until the convergence criterion is satisfied.

This forms the second stage of the framework, where the relative supersaturation setpoint is determined in order to obtain optimal product quality for any specific dynamics of the process. For the product quality prediction, the solute concentration and antisolvent mass percent values at each sampling instant until the end of the batch serve as the input, while the product yield and volume-weighted mean size of the product crystals serve as model output.

4. RESULTS AND DISCUSSIONS

This section presents the results obtained for the determination of the relative supersaturation setpoint values for optimal product quality through the proposed integrated framework. The process data for the semi-batch crystallization process is generated by implementing C-control strategy by giving different values for the relative supersaturation setpoint within a pre-defined range for each batch and for each specific dynamics. The batch
time is 120 minutes and the sampling time for solute concentration measurements and antisolvent mass percent \( (m_w) \) and solute concentration \( (C) \) is 30 seconds. Thus, at the end of each batch, 241 data points for both \( m_w \) and \( C \) are obtained. Besides, the product quality measurements in terms of both product yield and volume-weighted mean size of the product crystals are obtained from each batch. The constraints on the decision variables are chosen to be the bounds considered for each of the dynamics as shown in Table 1. Additionally, the constraint on minimum product yield is considered, i.e., yield \( \geq 40\% \). Population size and maximum number of generation were selected to be 100 and 9, respectively.

### Table 2 Summary of optimization study

<table>
<thead>
<tr>
<th>Case</th>
<th>Proposed design</th>
<th>Optimal control</th>
<th>% deviation of product quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rel SS setpoint</td>
<td>Product quality (( \mu m ))</td>
<td>Rel SS setpoint</td>
</tr>
<tr>
<td>2</td>
<td>0.099</td>
<td>381.49</td>
<td>0.101</td>
</tr>
<tr>
<td>3</td>
<td>0.119</td>
<td>38.99</td>
<td>0.117</td>
</tr>
<tr>
<td>4</td>
<td>0.020</td>
<td>655.05</td>
<td>0.035</td>
</tr>
<tr>
<td>5</td>
<td>0.456</td>
<td>440.38</td>
<td>0.46</td>
</tr>
<tr>
<td>6</td>
<td>0.007</td>
<td>62.22</td>
<td>10^{-5}</td>
</tr>
</tbody>
</table>

Table 2 summarizes the results obtained for all the case studies. For comparison, the actual optimal setpoint values resulting in optimal product quality were also obtained by solving optimal control problem with a precisely known process model for each case study. It can be seen that the product quality values and corresponding setpoint values obtained through the proposed approach are very close to the true values. Thus, the proposed framework helps in providing a learning control framework that not only adapts to variations in the process, but also provides optimal product quality at the end of each batch.

### 5. CONCLUSIONS

Recognizing the grave necessity of robust control and operation of pharmaceutical (semi-)batch crystallization processes, a data-based framework is proposed in this paper. A two-staged framework which incorporates pattern classification and nonlinear modeling for product quality is presented. In the first stage, LSSVM-based pattern classifier that selects its training data based on the \( k \) nearest neighborhood criterion is used during this study. Furthermore, JITL modeling method is used to predict solute concentration, while LSSVM-based MPLS model is used for the product quality predictions. Simulation results show that setpoints determined by the proposed approach helps in the optimal operation of the semi-batch antisolvent crystallization processes in the presence of variations in the process dynamics.

### REFERENCES


