Data-Driven Model Predictive Quality Control of Batch Processes

Siam Aumi and Brandon Corbett
Dept. of Chemical Engineering, McMaster University, Hamilton, ON L8S 4L8, Canada

Tracy Clarke-Pringle
DuPont Engineering Research and Technology, Wilmington, DE, 19898

Prashant Mhaskar
Dept. of Chemical Engineering, McMaster University, Hamilton, ON L8S 4L8, Canada

DOI 10.1002/aic.14063
Published online March 27, 2013 in Wiley Online Library (wileyonlinelibrary.com)

The problem of driving a batch process to a specified product quality using data-driven model predictive control (MPC) is described. To address the problem of unavailability of online quality measurements, an inferential quality model, which relates the process conditions over the entire batch duration to the final quality, is required. The accuracy of this type of quality model, however, is sensitive to the prediction of the future batch behavior until batch termination. In this work, we handle this "missing data" problem by integrating a previously developed data-driven modeling methodology, which combines multiple local linear models with an appropriate weighting function to describe nonlinearities, with the inferential model in a MPC framework. The key feature of this approach is that the causality and nonlinear relationships between the future inputs and outputs are accounted for in predicting the final quality and computing the manipulated input trajectory. The efficacy of the proposed predictive control design is illustrated via closed-loop simulations of a nylon-6,6 batch polymerization process with limited measurements. © 2013 American Institute of Chemical Engineers *AIChE J*, 59: 2852–2861, 2013
Keywords: control, process control, batch processes, model predictive control

Introduction

Batch processes constitute a class of processes that play an important role in the production and processing of a wide range of value-added products (i.e., chemical, pharmaceuticals, bio-chemicals, etc.). They are finite duration processes with unique characteristics, such as the absence of equilibrium points and nonlinear and time-varying dynamics over a wide range of operating conditions, that preclude the direct application of control strategies designed for continuous systems. Unlike continuous systems, which are characterized by control at an equilibrium point, the primary control objective in batch processes is to reach a specified product quality by batch termination. The economic benefits from batch processing are realized from the consistent production of on-spec product. Direct control to the specified quality, however, is impractical in most cases because quality measurements are unavailable online and only made offline after batch completion.

In the past, batch-to-batch operation policy consisted of charging the reactor with a recipe and then implementing predetermined input trajectories. These trajectories were either optimized offline, determined through data-mining, or historically yielded on-spec product. This type of open-loop operation policy, however, negatively impacted the reproducibility of quality as it was susceptible to disturbances encountered during the process and in the initial conditions (i.e., raw material impurities). Motivated by the increased demands of consistent production of high-quality products, numerous batch-to-batch (offline) and within-batch (online) control strategies were adopted.

The idea behind batch-to-batch control is to refine the batch recipe and operating trajectories for the upcoming batch using past data in an attempt to bring the new batch's quality closer to the specified value.[1] Batch-to-batch control strategies range from updating the model parameters and then recomputing the batch input trajectories (and/or batch recipe) to directly updating the process variable trajectories using an optimization-based algorithm[2] or the iterative learning control framework.[3,4] The former drives the process toward a specified optimum batch-wise, whereas the latter exploits the repetitive nature of batch systems by using the error in the final quality from the last batch to update the process variable trajectories and/or initial conditions.

Batch-to-batch control represents an entirely offline strategy and lacks any real-time feedback mechanism for rejecting disturbances encountered during batch evolution, motivating the use of real-time, within-batch control approaches. The within-batch quality control problem has been investigated extensively in the literature with many

Correspondence concerning this article should be addressed to P. Mhaskar at mhaskar@mcmaster.ca.

studies assuming availability of a detailed, first-principles process model.[5,6] However, in instances where the development and maintenance of an accurate first-principles model may not be possible, control designs need to rely on empirical or data-driven modeling. Data-driven modeling methods are particularly well-suited for batch systems because they can exploit existing batch databases and also provide an opportunity to update the model following the completion of each batch. Note that because continuous processes are operated around steady-states and narrower operating ranges, there is little useful information for identifying data-based models from typical operation. Instead, the process must be excited in some fashion to collect meaningful identification data. In contrast, due to the inherent process nonlinearity and moving setpoint, databases of batch processes are comprised of trajectories of online measurements of the key process variables from previous batches and contain important information about the process states, allowing a reliable model to be identified.

Within-batch control approaches can be broadly divided into trajectory tracking and inferential quality control approaches. Trajectory tracking is used where there is expected to be very little variance in the initial conditions for the batch, and hence reference trajectories for the measurable process variables (such as temperatures and pressures) can be tracked to reliably meet the specified quality. Tracking is achieved using classical control designs[7] or advanced control designs, such as differential geometric[8,9] or predictive[10–17] controllers, which are capable of compensating for the effects of nonlinearity and tracking set-points over a wide operating range. Although trajectory tracking controllers can reject disturbances online, even with perfect tracking, there is no guarantee that the desired quality will be met if there is significant variation in the initial conditions from batch to batch. This is because disturbances encountered during the new batch could alter the relationship between the product quality and the trajectories of the process variables. Thus, implementing the same reference trajectories batch-to-batch is not guaranteed to consistently produce on-spec product.

Inferential quality control refers to the control approach aimed at controlling quality directly, and is most commonly achieved through multivariate statistical process control (SPC) approaches, particularly those using latent variable tools, such as principal component analysis (PCA) or partial least-squares (PLS) regression.[18] For batch processes, the model development for the majority of SPC applications begins with the so-called "batch-wise" unfolding of multiway batch data.[19,20] The unfolded data are regressed (commonly via PLS regression) onto a matrix of final quality measurements to obtain an inferential PLS quality model[21] that is usable for predicting the final quality prior to batch completion. For batches with multiple phases or stages with distinct dynamics, multiple phase-specific (and transition) models can also be constructed.[22,23] During the batch evolution, the final quality can be predicted (at every sampling instant or predetermined decision points) and if the prediction exceeds the control limits, appropriate remedial action can be taken to correct the batch. The nature of the corrective action may be heuristics or knowledge based or more systematic wherein the quality model is inverted (one way or another) to directly compute the future input trajectories that recover the batch. The latter approach has been classified as a midcourse correction (MCC) control strategy.[24,25] Because it requires model inversion, the effectiveness of a MCC approach is particularly dependent on the underlying quality model and in general, demands richer training data that spans a wider operating range and exhibits more input variation compared to modeling for SPC.[26]

An important issue that arises in SPC and MCC approaches is that future online measurements that are required to predict the quality are incomplete. More specifically, the data arrangement in the model building process calls for the entire batch trajectory to predict the quality of the batch. However, during a batch, measurements are only available up to the current sampling instant, and the future data are required to predict the final quality. In the literature, this has so far been treated as a "missing data" problem, with the choice of the data completion technique playing a key role in the overall performance of the control design. Prediction error in the future data is propagated to the quality prediction error, and both of these errors add uncertainty to any control action computed using the model. This problem is particularly prevalent during the early stages of the batch when most of the information is unknown. In fact, with poor prediction of the future batch behavior, inputs determined from using the model can drive the batch to a point from where good quality product cannot be produced. This characteristic is typical of methods that lack a causal relationship between the inputs and the process trajectory, and in turn, the quality, which leads to the treatment of the future trajectory as a "missing data" problem.[24]

A variety of ad-hoc approaches exist to handle this "missing data" problem. Many methods use missing data algorithms available for latent variable methods. These missing data algorithms work on the assumption that the correlation structure between the collected measurements and future measurements for the new batch is the same as in the training data. Another approach has been to build a finite set of quality models at predetermined decision points (possibly at every sampling instant), and in building each model, rather than using the entire batch trajectory, data only up to the decision point is used.[26,27] This idea of an evolving model has also been modified for improving the quality prediction in multistage batches through consideration of critical-to-quality time periods at specific phases of the batch.[28,29] One issue with these multimodel-based approaches, however, is that quality models developed at early time points may be highly inaccurate because they will not capture the effects of large periods of the batch duration toward the batch quality. Although these missing data approaches are useful for predicting the quality and monitoring applications, when the inferential model is used in a control design, the need to consider the nonlinear casual relationship between the future input-output behavior is obvious. The quality control problem, therefore, stands to gain from the use of a causal, nonlinear model that does not treat the future trajectory as a missing data problem and instead recognizes it as the problem of choosing the remaining input trajectory, which determines the rest of the measured variables trajectory and in turn the final quality.

Motivated by the above considerations, in this work, we develop a within-batch quality control strategy for batch processes that unites a single-PLS inferential quality model with a previously developed nonlinear, data-driven modeling approach. By properly representing the future behavior using a causal model, control inputs can be chosen that result in improved quality control. The rest of this article is organized

as follows: first, we describe the class of processes being considered and review the key concepts, namely the different modeling approaches used in the control design. Next, we present the details of a predictive controller that is designed to drive a batch process to a desired specified product quality by batch termination. The efficacy of the control design is then demonstrated via simulations of a nylon-6,6 batch polymerization system. Finally, we summarize our results.

Preliminaries

In this section, we first describe the class of batch processes considered. Next, we discuss how an inferential quality model can be identified from existing batch data through multiway analysis. This is followed by an overview of PLS regression, a latent variable technique used to compute the inferential quality model parameters. Finally, we review a previously developed data-based modeling approach[30,31] that serves as the basis for predicting the future state trajectory (for a candidate input profile) when using the inferential quality model for quality control.

Process description

We consider batch processes subject to input constraints and limited measurements described by

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \\ \boldsymbol{y} &= \boldsymbol{g}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{v} \\ \boldsymbol{q} &= \boldsymbol{h}(\boldsymbol{x},\boldsymbol{u}) \\ t \in [t_0, t_f], \boldsymbol{u}(\cdot) &\in \mathscr{U}, \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \end{aligned} \tag{1}$$

Note that the above model is used only to illustrate the class of processes that the proposed modeling approach is expected to work, and the proposed approach does not require a model of the form of Eq. 1 to be available. The times $t=t_0$ and $t=t_f$, are the initial and termination times, respectively. The vectors $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^p$, and $\boldsymbol{q} \in \mathbb{R}^q$ denote the state variables, measurable process variables, and quality variables, respectively. Measurements of $\boldsymbol{y}$ are assumed to be available at every sampling instant, whereas the elements of $\boldsymbol{q}$ are only measured once following batch completion. The vector $\boldsymbol{v} \in \mathbb{R}^p$ represents zero-mean, normally distributed measurement noise. The vector $\boldsymbol{u} \in \mathbb{R}^m$ consists of constrained manipulated inputs, taking values in a nonempty, convex set, $\mathscr{U} \triangleq \{\boldsymbol{u} | \boldsymbol{u}_{\min} \le \boldsymbol{u} \le \boldsymbol{u}_{\max}\} \subset \mathbb{R}^m$ where $\boldsymbol{u}_{\min}$ and $\boldsymbol{u}_{\max}$ define the minimum and maximum (respectively) allowable input values.

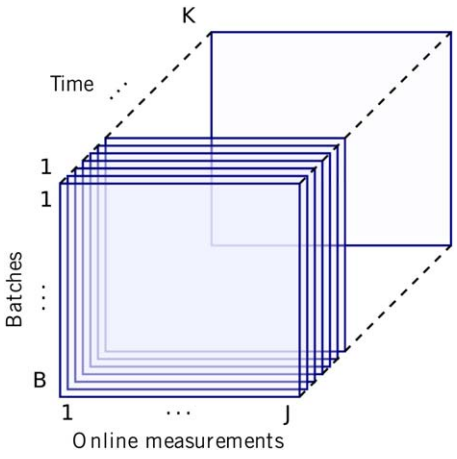

Figure 1. Process variable trajectories, X.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

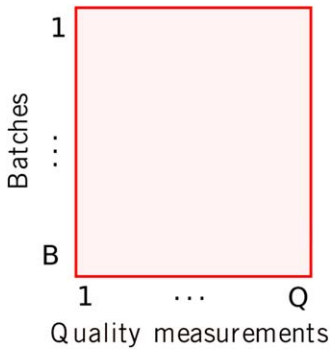

Figure 2. Quality data, Q.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Inferential quality model

To understand how to build a quality model that can be used during batch evolution to predict the quality at batch termination, we first describe the nature of data available in a batch database. Consider a typical batch run in which $j=1,2,\ldots,J=m+p$ variables are measured at $k=1,2,\ldots,K$ sampling instants. For $b=1,2,\ldots B$ batches, this data can be organized into a three-dimensional array (3-D), $\mathbf{X}$ ($B\times J\times K$), as shown in Figure 1.

Each vertical slice in this array, which has dimensions $B\times J$, represents the values of all the measurable variables for all batches at a common sampling instant. In addition to $\mathbf{X}$, measurements of $q=1,2,\ldots Q$ quality variables taken post batch can be summarized in a $B\times Q$ quality matrix, $\mathbf{Q}$, as shown in Figure 2. Finally, information about the initial conditions for each batch is also typically available (i.e., feedstock properties, measured raw material properties and compositions, charges of each ingredient, etc.), and this can be summarized in a $B\times M$ matrix, $\mathbf{Z}_0$, (see Figure 3) where M is the number of known variables related to the initial conditions.

To identify an inferential quality model that can be used to predict the batch quality, the 3-D array, $\mathbf{X}$, is first transformed into a 2-D $B\times JK$ matrix by unfolding it "batch-wise" such that each of its vertical slices is arranged side-by-side[19,20] (note that the dynamic model that we use to predict the batch process dynamics[30] does not require this kind of unfolding). Next, the initial conditions matrix, $\mathbf{Z}_0$, is

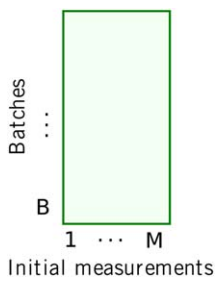

Figure 3. Initial conditions, $\mathbf{Z}_0$.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

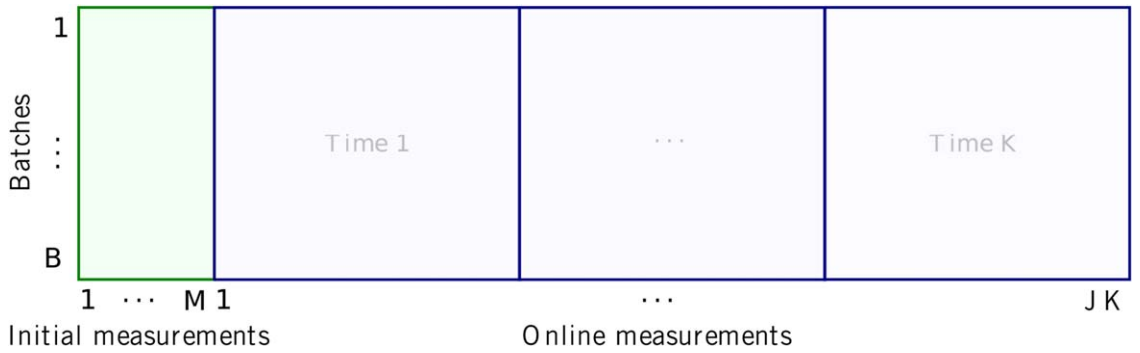

Figure 4. Rearrangement of the batch data in Figure 3 to form the regressor matrix for identifying the quality model in Eq. 2.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

concatenated to the unfolded matrix, forming a regressor matrix, $[\mathbf{Z}_0 \quad \mathbf{X}]$, as shown below in Figure 4.

The matrix in Figure 4 can be regressed onto the quality data matrix (Figure 2) using linear regression, yielding a model that relates the initial and process conditions to the quality characteristics as shown below

$$\hat{\mathbf{Q}} = [\mathbf{Z}_0 \quad \mathbf{X}]\Lambda \tag{2}$$

where $\hat{\mathbf{Q}}$ is the predicted quality and Λ is a $(M+JK)\times Q$ matrix of the quality model coefficients. Due to the high-dimensionality/multivariate nature of the regressor matrix and the likely presence of correlations among its variables, a latent variable regression technique, such as PLS regression or principal component regression, is necessary for appropriately estimating Λ.

Remark 1. An assumption made during the batch-wise unfolding scheme is that all batches are of equal length and the trajectories are synchronized (a requirement relaxed in recent results,[30] albeit for the purpose of dynamic modeling, not quality models). In practice, this assumption may not hold for raw batch data. Consequently, several methods have been proposed for addressing unequal batch lengths and to synchronize trajectories. The most common method involves choosing a monotonic indicator variable common to all batches, such as conversion, and resampling the variables with respect to this variable instead of time.[32] Additional methods for aligning batch trajectories include dynamic time-warping[33] and curve registration.[34]

PLS regression

When variables in the regressor matrix are (auto/cross) correlated, the use of ordinary least-squares regression leads to parameter estimates with large variances (due to covariance of the regressor matrix being ill-conditioned). Correlations can result from data being collected under closed-loop conditions[35] and/or from lagging the variables. One way to handle the numerical issues arising from correlations is through PLS regression. Geometrically, in PLS regression, the variables in the regressor and response matrices $[\mathbf{Z}_0 \quad \mathbf{X}]$ and $\mathbf{Q}$, are projected onto corresponding orthogonal subspaces of A-pairs of latent variables. Each pair of latent variables accounts for a certain percentage of the variance in the original matrices. Mathematically, PLS regression consists of decomposing $[\mathbf{Z}_0 \quad \mathbf{X}]$ and $\mathbf{Q}$ as the sum of the outer products of a score and loading vector

$$[\mathbf{Z}_0 \quad \mathbf{X}] = \sum_{a=1}^{A} \boldsymbol{t}_a \boldsymbol{p}'_a + \mathbf{E} \tag{3}$$

$$\mathbf{Q} = \sum_{a=1}^{A} \boldsymbol{d}_a \boldsymbol{r}'_a + \mathbf{F} \tag{4}$$

where $\boldsymbol{t}_a$ and $\boldsymbol{d}_a$ are the input and output scores representing the projections of the variables in $[\mathbf{Z}_0 \quad \mathbf{X}]$ and $\mathbf{Q}$ on the subspaces, $\boldsymbol{p}_a$ and $\boldsymbol{r}_a$ define the orientation of the corresponding subspaces, and $\mathbf{E}$ and $\mathbf{F}$ denote residual matrices. Because it is desired to obtain a useful relationship between the original data matrices, the two matrices are linked by an inner relation between their scores of the form

$$\boldsymbol{d}_a = \boldsymbol{b}_a \boldsymbol{t}_a + \boldsymbol{e}_a \qquad \forall a \in [1, A] \tag{5}$$

where $\boldsymbol{b}_a$ are the coefficients of the inner relationship and $\boldsymbol{e}_a$ are the residuals. In PLS algorithms, such as nonlinear iterative PLS,[18] the subspace orientation and scores for both matrices are determined simultaneously to maximize the correlation between $[\mathbf{Z}_0 \quad \mathbf{X}]$ and $\mathbf{Q}$ and, therefore, obtain the optimal fit for the inner relationship. The final result from PLS regression is a linear model between $[\mathbf{Z}_0 \quad \mathbf{X}]$ and $\mathbf{Q}$ where the coefficients are functions of the scores and loadings from the matrix decompositions.

Remark 2. In the PLS model, trajectories of the process variables over the entire batch duration and initial conditions are projected onto a latent variable subspace, and the values of the latent variables in this space are related to the scores of the final quality through the inner relationship. The projection essentially represents an estimation of the batch states at the end of the batch while the inner relationship is a "measurement" function relating the states at batch termination to the final quality. For a new batch, at sampling instant k, process variable trajectories are only available only up to k. More specifically, the process outputs are available up to sampling instant k and the inputs are available up to $k-1$. As a result, the vector required to make the state estimation at batch termination is incomplete. There are ways to eliminate this problem in monitoring applications (e.g., by using multiple models,[26] lookup-tables,[36] or a different unfolding scheme[19]); however, when using the model for control, the prediction of the future behavior for a given input is a necessity. Rather than eliminating the need for future data, we recognize the causal nature of the inputs in determining the future trajectory and in turn the quality.

Remark 3. In conventional PLS modeling, a common preprocessing step is to normalize the regressor and response matrices to zero mean and unit variance. Scaling to unit variance gives each variable equal importance during model identification; however, in many batch systems, there are specific periods during the batch that play a more critical role in determining the final quality than others. A simple way to account for these quality-critical periods within the PLS regression framework is to multiply the appropriate columns in the regressor matrix by weighting factors that make them more influential during the computation of the model parameters (and, therefore, during quality prediction). More formalized approaches for considering time-specific effects are also available.[27,28]

Multimodel data-driven modeling for batch systems

In this section, we review a multimodel, data-driven modeling approach that is used for predicting the future output behavior when using the inferential quality model.[30,31] Mathematically, the model for the process outputs takes the form of a weighted combination of L linear dynamic models as shown below in Eq. 7

$$\hat{\mathrm{y}}(k)=\sum_{\ell=1}^{L} w_\ell(k)\hat{\beta}_\ell \bar{\mathrm{x}}(k) \tag{6}$$

$$=\sum_{\ell=1}^{L} w_\ell(k)\hat{\beta}_\ell \left[\boldsymbol{y}'(k-1)\cdots \boldsymbol{y}'\left(k-n_y\right)\cdots \boldsymbol{u}'(k-1)\cdots \boldsymbol{u}'(k-n_u)\right]' \tag{7}$$

where $w_\ell(k)$ is model ℓ's weight at sampling instant, k, $\hat{\beta}_\ell$ defines the ℓ-th local model, and $\bar{\mathrm{x}}(k)$ is a vector of lagged inputs and outputs. The scalars, n_y and n_u, denote the number of lags in the outputs and inputs (respectively). For notational simplicity, we have assumed the same number of lags, n_y and n_u, for each output and input variable (respectively) and the same lag structure for all L models. These assumptions can be readily relaxed.

Using the following definitions

$$\hat{\beta} \triangleq \left[\hat{\beta}_1 \quad \cdots \quad \hat{\beta}_\ell \quad \cdots \quad \hat{\beta}_L\right]$$
$$\boldsymbol{h}(k) \quad \triangleq \left[w_1(k)\bar{\boldsymbol{x}}'(k) \quad \cdots \quad w_\ell(k)\bar{\boldsymbol{x}}'(k) \quad \cdots \quad w_L(k)\bar{\boldsymbol{x}}'(k)\right]'$$

Equation 7 can be rewritten in the vector form

$$\hat{\mathrm{y}}(k)=\hat{\beta}\boldsymbol{h}(k) \tag{8}$$

The model identification procedure consists of an initial clustering step followed by solving a linear regression problem. In the first step, for a given lag structure, a matrix $\bar{\mathbf{X}}$, corresponding to $\bar{\boldsymbol{x}}(k)$, is generated by sorting the plant data sample-wise (not time or critical variable wise, as is done in existing batch modeling approaches) and then, $\bar{\mathbf{X}}$ (or its equivalent latent variable space[30]) is clustered into L clusters using an appropriate clustering technique. Thus, observations of lagged inputs and outputs are clustered together, and, to illustrate our results, in this work we use fuzzy c-means clustering. In fuzzy c-means clustering, points that are mathematically "similar" according to the Euclidean 2-norm are clustered into overlapping spherical clusters with corresponding center points.[37] Each cluster represents a region in the $\bar{\mathbf{X}}$ space where an associated model has the highest degree of validity, and in this way, the cluster's center point represents the linearization point of its associated model. Using the results of the clustering step (the cluster center points), the weights $w_\ell(\cdot)$, for the training data can be computed prior to the model coefficients (to be discussed shortly). Consequently, the $\boldsymbol{h}(k)$ vector in Eq. 8 is completely specified for the training data. Thus, a regressor matrix corresponding to $\boldsymbol{h}(k)$ can be constructed, and the local linear models $\hat{\beta}$ are computable using linear regression.

Intuitively, from Eq. 1, the model weights $w_\ell(\cdot)$, should depend on the current values of the states and inputs because they define the system dynamics through $\boldsymbol{f}(\cdot)$. In other words, the local models should be weighted according to the current process conditions. In the absence of state measurements, the vector of lagged outputs and inputs $\bar{\mathrm{x}}(k)$, can be used to infer the current process conditions, and each model's weight can be assigned based on the proximity of the operating conditions to its center point. For instance, denoting model ℓ's center point as $\boldsymbol{c}_\ell$, its weight should be inversely proportional to the squared distance between $\bar{x}(k)$ and $\boldsymbol{c}_\ell$

$$w_\ell(k) \propto ||\bar{\mathrm{x}}(k)-\boldsymbol{c}_\ell||^{-2}$$

Normalizing this expression over all clusters yields the following normalized weighting function

$$w_\ell(k)=\frac{||\bar{\mathrm{x}}(k)-\boldsymbol{c}_\ell||^{-2}}{\sum_{i=1}^{L}||\bar{\mathrm{x}}(k)-\boldsymbol{c}_i||^{-2}}$$

The number of clusters is an important parameter in this approach. Well-defined criteria (based on the cluster geometry) to iteratively refine the number of clusters are available.[38] Additionally, to evaluate the goodness of the final fuzzy partitions, many validation measures have also been proposed with the most popular being the Xie–Beni index,[39] which is a ratio of the total within-cluster variance to the separation of the cluster centers (and, therefore, should be minimal for the best partition). In this work, we picked the number of clusters (iteratively) based on how well an independent validation data set was predicted. Thus, there was a balancing of the number of clusters and prediction error.

Model Predictive Quality Control

In this section, the data-driven modeling approach reviewed in the previous subsection is used in conjunction with an inferential quality model in a model predictive control (MPC) design. The quality model captures the (time-cumulative) effects of the entire batch trajectory on the final quality, whereas the multiple linear models for the measurable process variables take the causality and nonlinear relationship between the inputs and outputs into account. The benefit from this approach is the ability to account for the direct connection between the control action and the quality, something that is both expected and desired.

Given a batch database, consider the case where the quality and process variable models have been identified. The following MPC optimization problem is solved to compute the control action with the objective of achieving a desired product quality, $\boldsymbol{q}_\mathrm{des}$

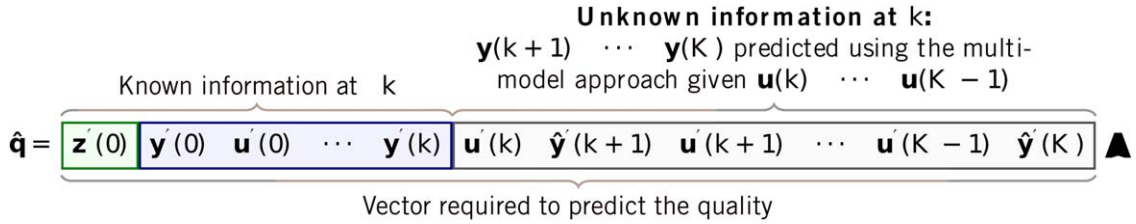

Figure 5. Schematic of how the multimodel approach is used with the inferential quality model to predict the future (unknown) output trajectories.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$$\min_{\boldsymbol{u}(k)\in\mathcal{U}} (\hat{q}-\boldsymbol{q}_{\text{des}})'\Psi(\hat{q}-\boldsymbol{q}_{\text{des}})+\sum_{i=k}^{K}\Delta\boldsymbol{u}'(i)\Phi\Delta\boldsymbol{u}(i) \tag{9}$$

$$\text{st}: \quad \hat{y}(k)=\boldsymbol{y}(t) \tag{10}$$

$$\hat{y}(k)=\sum_{\ell=1}^{L} w_\ell(k)\hat{\beta}_\ell\bar{x}(k) \tag{11}$$

$$\boldsymbol{x}'_{\text{future}}=\begin{bmatrix}\boldsymbol{u}'(k) & \hat{\boldsymbol{y}}'(k+1) & \boldsymbol{u}'(k+1) & \cdots & \boldsymbol{u}'(K-1) & \hat{\boldsymbol{y}}'(K)\end{bmatrix}' \tag{12}$$

$$\hat{q}=[\boldsymbol{x}_{\text{past}} \quad \boldsymbol{x}_{\text{future}}]\Lambda \tag{13}$$

In this optimization problem, the objective function consists of a term for minimizing the discrepancy between the target product quality and the predicted quality $\hat{q}$, and a move suppression factor. Each term's relative importance is traded-off using the positive-definite weighting matrices Ψ and Φ. Equation 10 is the MPC initialization at the current plant conditions and Eq. 11 represents the prediction of the future process variables using the data-driven model (given the current input trajectory in the optimizer). The predicted process outputs and optimizer inputs are stored appropriately in the row vector $\boldsymbol{x}_{\text{future}}$, through Eq. 12. This vector is concatenated with a vector of previous plant outputs and implemented inputs $\boldsymbol{x}_{\text{past}}$. Note that the vector $\boldsymbol{x}_{\text{past}}$, is known prior to solving the MPC optimization problem; specifically, $\boldsymbol{x}'_{\text{past}}=[\boldsymbol{z}'(0) \quad \boldsymbol{y}'(0) \quad \boldsymbol{u}'(0) \quad \cdots \quad \boldsymbol{y}'(k)]'$ where $\boldsymbol{z}\prime(0)$ denotes all the information known prior to starting the batch (i.e., the initial conditions). The concatenated vector $[\boldsymbol{x}_{\text{past}} \quad \boldsymbol{x}_{\text{future}}]$, is used to predict the quality through Eq. 13.

The way the multimodel approach is integrated with the inferential quality model is illustrated in Figure 5. At sampling instance k, the vector required to predict the quality is incomplete; the inputs and outputs beyond k are unknown. However, when the model is embedded in the MPC optimization problem (wherein the prediction horizon extends to the end of the batch), a candidate input trajectory exists in the optimizer that can be used to predict the future outputs up to batch termination. In our work, we use the multimodel approach to predict these future outputs. For a candidate input trajectory, the final product quality can be thus predicted, allowing direct control to a desired quality.

Remark 4. A distinguishing feature of this MPC design includes the use of a causal (and effectively nonlinear) model for predicting the future observations expected if a particular candidate input trajectory is implemented. The majority of control designs that have used a multivariate, PLS-based inferential quality model instead treat this as a missing data problem. Based on the data collected up to the current sampling instant, these algorithms essentially invert the linear, PLS model such that the future behavior maintains the same correlation structure as previous batches. This leads to an inherent mismatch in the sense that the predicted future behavior is based on past data [that typically uses existing, proportional integral (PI) controllers] which in turn is used to compute the current control action via a different control algorithm than in the data set. In contrast, the proposed approach recognizes that the problem is not that of missing data, because the future trajectories depend on state and input trajectories up to the current point as well as future input moves. The only "missing" part, therefore, is the part that needs to be computed by the controller—the set of future control moves. By using an appropriate model (which captures the process nonlinearity) that links the future inputs to the future process outputs and in turn to the quality, the controller then computes the set of input moves that would yield the desired quality. The problem of unequal batch lengths can also be readily handled by the proposed quality prediction approach by virtue of using a dedicated (nonlinear) model for predicting the future batch behavior. In particular, all the batch data can be used to build the dynamic model, while the quality model can be built using a common time from the end of batch for all batches (i.e., using the batch time of the shortest batch, if not significantly shorter than the other batches).

Remark 5. Other control designs have tried to eliminate the missing data problem completely through evolving quality models (at each sampling instant or a selected number of predetermined decision points), which use measurements only up to a given time. These models are designed to forecast the final quality without the future batch trajectories and inherently rely on the assumption that the same control action is implemented for the rest of the batch. Therefore, although such methods may be good to "predict" the quality under an existing controller, they are not well-suited for use in a control design aimed at computing the control action to yield the desired quality.

Remark 6. The prediction horizon for the MPC optimization problem above must extend to the end of the batch; thus, the prediction horizon, $P=K-k$, shrinks at every sampling instant. During the early stages of the batch when k is low, the MPC optimization problem may be too computationally demanding for real-time implementation. Under such circumstances, the optimization problem can be used to update the reference trajectories for local controllers rather than directly computing the inputs. Specifically, although the optimization problem is being solved, trajectory tracking controllers can be used to track the nominal reference trajectories, and on completion of the optimization problem, the

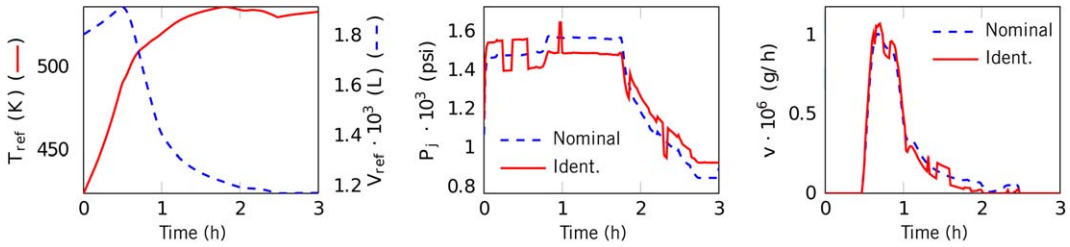

Figure 6. Trajectories used during database generation.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

trajectories of the measurable output variables from the solution can be specified as the updated nominal reference trajectories in the local trajectory tracking controllers.

Simulation Results

In this section, we demonstrate the efficacy of the proposed MPC design through closed-loop simulations of a nylon-6,6 batch polymerization process. First, we give an overview of the process and state the control objective. Next, we develop data-driven models for the measurable process variables and product quality from a database generated using a rigorous first-principles model (also used as a test bed for implementation of the control design). Finally, using these models, we implement the proposed MPC design and compare its performance against trajectory tracking control.

Process overview and control objective

For this work, we focus on nylon-6,6 production by the amidation of adipic acid and hexamethylenediamine (HMD) in a batch reactor. The reactor is initially charged with molten adipic acid and HMD (from an evaporator) in approximately stoichiometric (1:1) proportions. The polymerization reaction (a polyamidation) is summarized as follows: Carboxylic end groups (C) on adipic acid or the polymer chain react (reversibly) with amine end groups (A) on HMD or the polymer chain, producing a polymer chain link (L) and a water molecule (W)

$$\text{Polyamidation} : \text{C}+\text{A} \leftrightarrows \text{L}+\text{W} \tag{14}$$

This polymerization is typically carried out in an autoclave reactor equipped with a steam jacket for providing the heat needed for vaporization (and reaction) and a valve for venting vaporized water. The polymerization occurs in three phases: (1) initial heating phase, (2) boiling phase, and (3) finishing phase. During the initial heating phase, the vent valve is closed to prevent the loss of volatile HMD and heat is supplied through the steam jacket, driving the polymerization reaction. After a certain extent of reaction, the valve is opened, initiating the boiling phase. During this phase, excess water is removed, which is important for achieving high molecular weight of the final polymer. After venting water for an appropriate amount of time, the vent is closed, and the finishing phase begins during which the final quality characteristics of the polymer are developed.

For our studies, we used a mathematical model of this process[40] that takes the general form shown in Eq. 1. The reaction model, modeling assumptions, equations, and parameter values[40] are omitted here for brevity. The state vector is comprised of the molar amounts of each functional group and evaporated HMD, and the reaction medium mass and temperature. The states were assumed to be measured only once at the initial time, but note that many of the states were trivially zero due to the absence of any reaction. The manipulated inputs were taken to be the steam jacket pressure, P_j (psi), and vent rate, v (g/h): $\boldsymbol{u}=[P_j \quad v]'$, and the constraints were defined as follows: $\boldsymbol{u}_{\min}=[700 \quad 0]'$ and $\boldsymbol{u}_{\max}=[1.8\times10^3 \quad 2.5\times10^6]'$. All batches were assumed to be 3 h, eliminating the requirement for any trajectory synchronization, with a sampling period of 1 min.

The measurable process variables were taken to be the reaction mixture temperature, T (K), volume, V (L), and the viscosity, η (cP): $\boldsymbol{y}=[T \quad V \quad \eta]'$. Note that in practice, although the viscosity may not be directly measurable at the sampling period of 1 min, stirrer torque measurements are typically available in real time at every sampling instant. The stirrer torque is strongly correlated with the solution viscosity with a more viscous polymer resulting in higher torque (for a fixed rotations per minute stirrer motor). Thus, the torque measurements provide important information about the viscosity evolution.

The product quality of nylon-6,6 polymer is defined by the number average molecular weight, MW, and the residual amide concentration R_{NH_2} (mol/g): $\boldsymbol{q}=[\text{MW} \quad R_{NH_2}]'$. Both qualities are related to the state variables through highly nonlinear relationships.[40] The control objective considered in this work was to achieve end-point qualities of $\boldsymbol{q}_{\text{des}}=[5569 \quad 136]'$.

Inferential quality and process variable models

To develop data-driven models for the quality and process variables, an artificial batch database of the form in Figure 3 was first generated. To this end, the deterministic model was simulated 80 times from different initial conditions (30 batches were reserved as the validation data set). In generating the database, a set of reference T and V profiles presented in Figure 6a were tracked using P_j and v (respectively) via two tightly tuned PI controllers. In addition to these closed-loop trajectories, the database was supplemented with four open-loop identification batches. For these batches, low amplitude, pseudorandom-binary-sequence signals were added on top of the nominal input trajectories. In

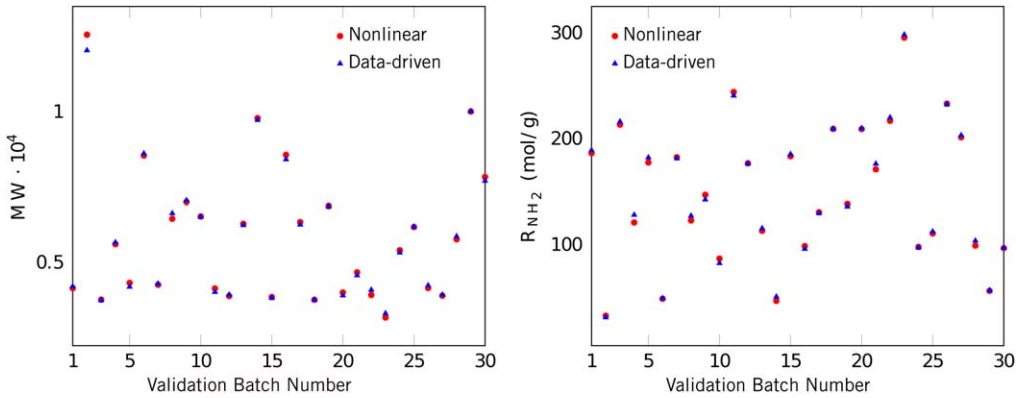

Figure 7. Comparison of the predicted quality by the inferential PLS based data-driven model with the nonlinear model.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Figure 6b, the input trajectories for one of these identification batches are shown together with the nominal trajectories (that correspond to the T and V trajectories in Figure 6a).

The final database consisted of measurements of the states at the initial time, T, V, and η at every sampling instant, and the qualities at batch termination. Prior to developing the models, the η measurements were replaced by $\ln(\eta)$, in recognition of the fact that viscosity typically satisfies a power law correlation. Using this database, an inferential quality model was developed using PLS regression. As discussed in Remark 3, quality-critical periods during the batch can be given more weight prior to computing the model parameters. For the nylon-6,6 process, the initial conditions and process behavior during the boiling and finishing phases are more influential to the final quality compared to the heating phase; consequently, columns corresponding to these conditions in the unfolded regressor matrix (see Figure 4) were given six, two, and four times more weight than the heating phase (respectively).

The motivation behind placing the lowest weight on the heating phase was because it corresponded to a limited extent of the polymerization compared to the other phases. The weights for the other phases and initial conditions were found iteratively and chosen to minimize the root mean squared error (RMSE) in the predicted qualities of the validation batches. Thus, these weights were essentially tuning parameters in the model. High weights were placed on the initial conditions to compensate for the fact that they constituted a very small portion of the regressor matrix compared to the other phases. In Figure 7, the qualities predicted by the PLS model for the 30 validation batches are displayed along with the database qualities. The number of principal components in the PLS model, 24, was selected to minimize the RMSE in the predicted qualities of the validation batches. From the discussion in Remark 2, this meant 24 latent variables were required to estimate the states at batch termination. Note that the total number of columns in the regressor matrix was over 900; thus, 24 latent variables still represent a significant reduction in the dimensionality of the process. Additionally, the training data contained four identification batches that expanded the range of typical operating conditions, calling for additional latent variables. Overall, the inferential quality model predicted the final qualities with relatively good accuracy.

Next, multiple local linear models of the form described in "Multimodel data-driven modeling for batch systems" section were developed for T, V, and $\ln(\eta)$ because these variables must be predicted (for a given input trajectory) as part of the quality prediction. To cluster the database and compute the resulting local linear models, the following parameters were required to be specified: the lag structure (n_y and n_u) and the number of clusters, L. The model identification procedure was as follows. For a given lag structure (which sets the dimensions of $\mathbf{X}$), the $\mathbf{X}$ matrix was constructed, decomposed using PCA, and clustered for a range of clusters, $L=\{1,2,\ldots,20\}$. For each cluster choice and a lag structure, a PLS model was identified. This was repeated for all possible lag structures with a lag range of 0–2 for each variable. The goodness of each fit (and the number of latent variables to retain in the PLS regression) was judged using its RMSE in predicting the validation batches. The number of input and output lags were found to be 1 (i.e., $n_y=n_u=1$)

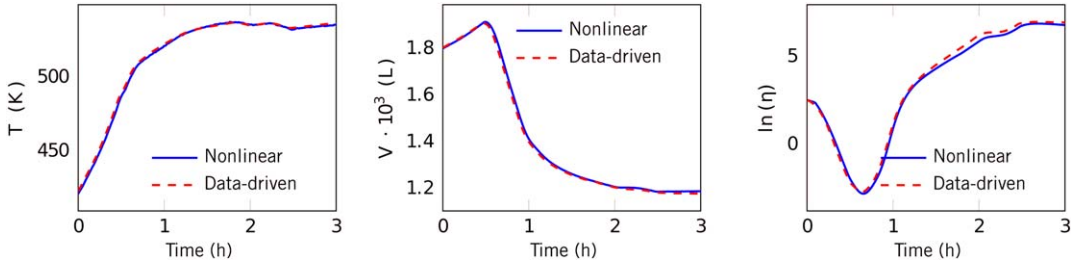

Figure 8. Comparison of the predicted (measurable process) variables by the data-driven model with the nonlinear model for a random validation batch.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

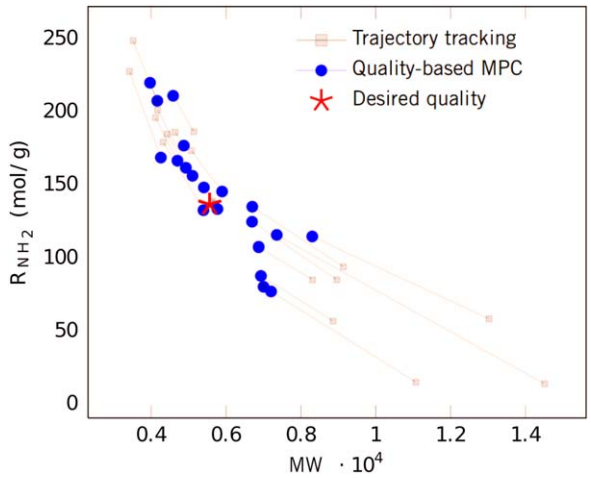

Figure 9. Comparison of the final qualities obtained from trajectory tracking and quality-based MPC design for 21 new initial conditions.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

and the number of clusters, L, was 7. One explanation for requiring only one lag is the assumption of the same lag structure for all the local models (note that this assumption can be readily relaxed if needed). With this assumption, using all first-order models minimized the possibility of over-fitting, and in this case, yielded the lowest RMSE values. In Figure 8, we compare the output of the nonlinear model with the model output from the data-driven model for a set of initial conditions in the validation data set. Overall, the multimodel approach captured the major nonlinearities and provided relatively reliable predictions.

Closed-loop simulation results

Using the models developed in the previous section, in this section, we present the results from implementing the predictive controller and compare its control performance against trajectory tracking via PI controllers. For these simulations, we considered 21 new initial conditions that were not in the training or validation data sets. The reference trajectories for the PI-based trajectory tracking simulations were those presented earlier in Figure 6a, and the loop-pairings and tunings were kept consistent with the database generation procedure. In solving the MPC optimization problem, the initial guess for the input trajectories was set to the nominal trajectories at $t=0$ and the tail of the solution at the previous sampling instant for all subsequent sampling instants. The computation time (as reported by the Matlab functions, `tic` and `toc`) for the predictive controller at $t=0$ was 1.5 s on an Intel Quad Core machine using GAMS with IPOPT as the optimization software. The computation times for all successive sampling times were lower due in part to the shrinking horizon nature of the optimization problem.

In Figure 9, the final qualities yielded from trajectory tracking are compared with those from the proposed MPC design. On average, there was a significant improvement in meeting the specified quality. With trajectory tracking, the standard deviations from the target quality were 3392 and 69.69 mol/g for MW and R_{NH_2}, respectively. These values were reduced to 1240 and 40.71 mol/g by the predictive controller.

Input trajectories (from both controllers) for one of the batches are shown in Figure 10. For these inputs, trajectory tracking yielded qualities of 6434 and 110 mol/g for MW and R_{NH_2} (respectively), whereas MPC yielded qualities of 5559 and 131 mol/g. Recall that the desired values for the final MW and R_{NH_2} were 5569 and 136 mol/g. Comparing the input trajectories in Figure 10, we observe that the MPC prescribed inputs followed the same general trends as with trajectory tracking but with sufficient refinements to significantly improve on the quality.

Conclusions

In this work, we proposed a predictive control design for batch systems designed to drive the batch to a specified quality by batch termination. The MPC design used two types of models: a linear, PLS model which relates the process conditions over the entire batch duration to the final product quality and weighted local linear models that, for a candidate input trajectory, predict the future process conditions up to batch termination. Accounting for the causality and nonlinear relationships between the future input and process measurements, trajectory through the multiple local linear models led to more effective control action. The proposed control design was demonstrated via simulations of a highly nonlinear nylon-6,6 batch polymerization system, and it significantly improved on the final quality variance over that obtained using trajectory tracking control.

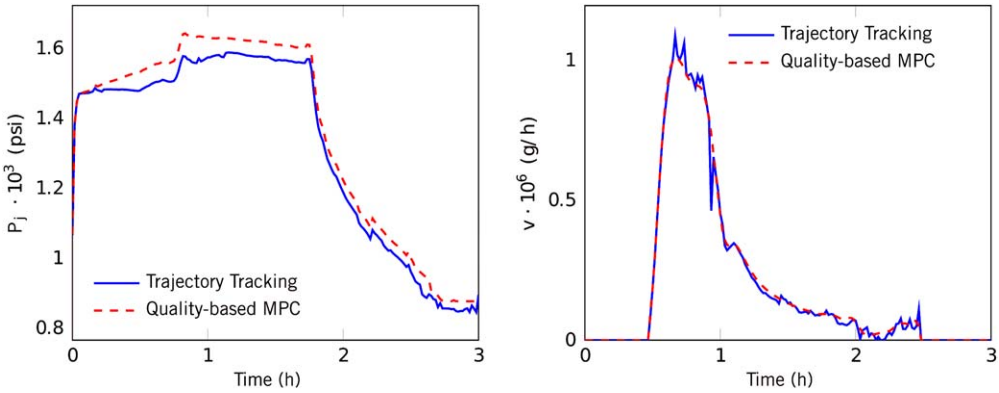

Figure 10. Inputs prescribed by the predictive and trajectory tracking controllers for one of the batches in the closed-loop simulations.

[Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Acknowledgment

Financial support from the McMaster Advanced Control Consortium and the Natural Sciences and Engineering Research Council of Canada (NSERC) through the CGS(D) award for Siam Aumi is gratefully acknowledged.

Literature Cited

1. Clarke-Pringle TL, MacGregor JF. Optimization of molecular-weight distribution using batch-to-batch adjustments. *Ind Eng Chem Res*. 1998;37(9):3660–3669.
2. Dong D, McAvoy TJ, Zafiriou E. Batch-to-batch optimization using neural network models. *Ind Eng Chem Res*. 1996;35(7):2269–2276.
3. Flores-Cerrillo J, MacGregor JF. Iterative learning control for final batch product quality using partial least squares models. *Ind Eng Chem Res*. 2005;44(24):9146–9155.
4. Lee JH, Lee KS, Kim WC. Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*. 2000;36(5):641–657.
5. Soroush M, Kravaris C. Multivariable nonlinear control of a continuous polymerization reactor: an experimental study. *AIChE J*. 1993;39(12):1920–1937.
6. Soroush M, Kravaris C. Optimal-design and operation of batch reactors. *1. Theoretical framework*. *Ind Eng Chem Res*. 1993;32:866–881.
7. Cruickshank SM, Daugulis AJ, McLellan PJ. Dynamic modeling and optimal fed-batch feeding strategies for a two-phase partitioning bioreactor. *Biotechnol Bioeng*. 2000;67:224–233.
8. Clarke-Pringle T, MacGregor JF. Nonlinear adaptive temperature control of multi-product, semi-batch polymerization reactors. *Comput Chem Eng*. 1997;21(12):1395–1409.
9. Kravaris C, Soroush M. Synthesis of multivariable nonlinear controllers by input/output linearization. *AIChE J*. 1990;36(2):249–264.
10. Peterson T, Hernandez E, Arkun Y, Schork F. Nonlinear DMC algorithm and its application to a semibatch polymerization reactor. *Chem Eng Sci*. 1992;47(4):737–753.
11. Shi D, El-Farra NH, Li M, Mhaskar P, Christofides PD. Predictive control of particle size distribution in particulate processes. *Chem Eng Sci*. 2006;61(1):268–281.
12. Shi D, Mhaskar P, El-Farra N, Christofides P. Predictive control of crystal size distribution in protein crystallization. *Nanotechnology*. 2005;16(7):562–574.
13. Golshan M, MacGregor JF, Bruwer MJ, Mhaskar P. Latent variable model predictive control (LV-MPC) for trajectory tracking in batch processes. *J Process Control*. 2010;20(4):538–550.
14. Golshan M, MacGregor JF, Bruwer MJ, Mhaskar P. Latent variable MPC for trajectory tracking in batch processes: role of the model structure. In: Proceedings of the American Control Conference. 2009;4779–4784.
15. Aumi S, Mhaskar P. Safe-steering of batch processes. *AIChE J*. 2009;55:2861–2872.
16. Aumi S, Mhaskar P. Robust model predictive control and fault-handling of batch processes. *AIChE J*. 2010;57:1796–1808.
17. Mhaskar P, Aumi S. Transition from batch to continuous operation in bio-reactors: a model predictive control approach and application. *Can J Chem Eng*. 2007;45:416–423.
18. Geladi P, Kowalski B. Partial least-squares regression: a tutorial. *Anal Chim Acta*. 1986;185:1–17.
19. Wold S, Geladi P, Esbensen K, Ahman J. Multi-way principal components-and PLS-analysis. *J Chemom*. 1987;1(1):41–56.
20. Nomikos P, MacGregor JF. Monitoring batch processes using multiway principal component analysis. *AIChE J*. 1994;40(8):1361–1375.
21. Flores-Cerrillo J, MacGregor JF. Within-batch and batch-to-batch inferential-adaptive control of semibatch reactors: a partial least squares approach. *Ind Eng Chem Res*. 2003;42(14):3334–3345.
22. Undey C, Cinar A. Statistical monitoring of multistage, multiphase batch processes. *IEEE Control Syst Mag NY*. 2002;22(5):1361–1375.
23. Ng YS, Srinivasan R. An adjoined multi-model approach for monitoring batch and transient operations. *Comput Chem Eng*. 2009;33(4):887–902.
24. Flores-Cerrillo J, MacGregor JF. Control of particle size distributions in emulsion semibatch polymerization using mid-course correction policies. *Ind Eng Chem Res*. 2002;41(7):1805–1814.
25. Yabuki Y, MacGregor JF. Product quality control in semibatch reactors using midcourse correction policies. *Ind Eng Chem Res*. 1997;36(4):1268–1275.
26. Wang D, Srinivasan R. Multimodel based real-time final product quality control strategy for batch processes. *Comput Chem Eng*. 2009;33(5):992–1003.
27. Gunther JC, Conner JS, Seborg DE. Process monitoring and quality variable prediction utilizing PLS in industrial fed-batch cell culture. *J Process Control*. 2009;19(5):914–921.
28. Lu N, Gao F. Stage-based process analysis and quality prediction for batch processes. *Ind Eng Chem Res*. 2005;44(10):3547–3555.
29. Zhao C, Wang F, Mao Z, Lu N, Jia M. Quality prediction based on phase-specific average trajectory for batch processes. *AIChE J*. 2008;54(3):693–705.
30. Aumi S, Mhaskar P. Integrating data-based modelling and nonlinear control tools for batch process control. *AIChE J*. 2012;58:2105–2119.
31. Aumi S, Corbett B, Mhaskar P, Clarke-Pringle T. Data-based modelling and control of nylon-6,6 batch polymerization. *IEEE Trans Control Sys Technol*. In press.
32. Kourti T, Lee J, Macgregor JF. Experiences with industrial applications of projection methods for multivariate statistical process control. *Comput Chem Eng*. 1996;20:745–750.
33. Kassidas A, MacGregor JF, Taylor PA. Synchronization of batch trajectories using dynamic time warping. *AIChE J*. 1998;44(4):864–875.
34. Undey C, Williams BA, Cinar A. Monitoring of batch pharmaceutical fermentations: data synchronization, landmark alignment, and real-time monitoring. In: Proceedings of the IFAC World Congress on Automatic Control, Vol. 15. Barcelona, ES, 2002.
35. Huang B, Ding XS, Qin SJ. Closed-loop subspace identification: an orthogonal projection approach. *J Process Control*. 2005;15(1):53–66.
36. Patel S, Yelchuru R, Ryaliand S, Gudi RD. Discriminatory learning based performance monitoring of batch processes. In: Proceedings of the American Control Conference, San Francisco, CA. 2011;2552–2557.
37. Bezdek J. A convergence theorem for the fuzzy isodata clustering algorithms. *IEEE Trans Pattern Anal Mach Intell*. 1980;2(1):1–8.
38. Frigui H, Krishnapuram R. Clustering by competitive agglomeration. *Pattern Recognit*. 1997;30(7):1109–1119.
39. Xie XL, Beni G. A validity measure for fuzzy clustering. *IEEE Trans Pattern Anal Mach Intell*. 1991;13:841–847.
40. Russell SA, Robertson DG, Lee JH, Ogunnaike BA. Control of product quality for batch nylon-6,6 autoclaves. *Chem Eng Sci*. 1998;53(21):3685–3702.

Manuscript received May 28, 2012, revision received Dec. 4, 2012, and final revision received Feb. 8, 2013.